

Martin Wighus Holtmon
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Automated Quality Assurance of Digitization in the Digital Archives

Bachelor's thesis in Computer Science
Supervisor: Rune Hjelsvold and Johanna Johansen
May 2022

Martin Wighus Holtmon
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Automated Quality Assurance of Digitization in the Digital Archives

Bachelor's thesis in Computer Science
Supervisor: Rune Hjelsvold and Johanna Johansen
May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Department of Computer Science

Abstract

The Digital Archives is Norway's joint solution for receiving, preserving, and publishing digitized/media-converted historical archives from state, municipal and private actors. The archives are digitized/media-converted by each archive institution, with both different equipment and results. The client for the project is one of these archive institutions, Inter-municipal archive of Møre og Romsdal IKS (IKAMR). They needed an application that could grade images at a quality level in accordance with ISO 19264-1:2021. The application is necessary to ensure the quality of the images uploaded to the Digital Archives. The result of the project is a split application: a backend REST API module and a web-based frontend module that serves as a usable example of how the API can be integrated into an existing system. The backend module conducts analysis, grades the images, validates the images, and embeds the results as metadata in the images. The solution consists of an efficient way of conducting quality evaluation of all uploaded images and is usable by anyone with basic experience in image quality assurance. Furthermore, the solution gives IKAMR an application that can be integrated into their workflow to ensure more consistent and faster quality assurance. The group has focused on delivering a high-quality product developed using professional work environments and the agile methodology Scrum.

Sammen drag

Digitalarkivet er Norges felle sløsning for mottak, bevaring og publisering av digitaliserte/mediekonverterte historiske arkiv fra statlige, kommunale og private aktører. Arkivene blir digitalisert/mediekonvertert av hver enkelt arkivinstitusjon, både med ulikt utstyr og resultat. Oppdragsgiveren for prosjektet er en av disse arkiveinstitusjonene, Interkommunnalt arkiv for Møre og Romsdal (IKAMR). De hadde behov for en applikasjon som kunne gjennomføre sortering av bilder på kvalitetssnivå i henhold til ISO 19264-1:2021. Denne applikasjonen er nødvendig for å kvalitetssikre bilder som blir lastet opp til Digitalarkivet. Resultatet av prosjektet er en todelt applikasjon: en backend REST API modul og et web-basert brukergrensesnitt som fungerer som et eksempel på hvordan API'en kan bli integrert i et eksisterende system. Backend modulen gjennomfører analyser, graderer bildene, validerer bildene og legger ved resultatet som metadata i bildene. Løsningen er en effektiv måte å gjøre kvalitetssikring av all opplastede bilder og er brukbar av alle som har grunnleggende erfaring innen bildekvalitetssikringsfeltet. Løsningen gir IKAMR en applikasjon som kan bli integrert i deres arbeidsflyt for å sikre en raskere og mer konsekvent kvalitetssikring. Gruppen har fokusert på å levere et produkt med høy kvalitet og som har blitt utviklet ved bruk av profesjonelle arbeidsmiljøer og den smidige metodologien Scrum.

Preface

The thesis is written by Martin Wighus Holtmon, Jakob Frantzvåg Karlsmoen and Mikael Falkenberg Krog at the Department of Computer Science (IDI) at NTNU in Gjøvik.

We would like to express our gratitude to anyone who has helped us during the project. First off, we are incredibly grateful to our supervisors, Rune Hjelsvold and Johanna Johansen at NTNU in Gjøvik. Their guidance through our weekly meetings has been highly beneficial. Next off, we would like to thank our client, Inter-municipal archive of Møre og Romsdal IKS (IKAMR) whom Ottar André Breivik Anderson represented. Their representative has shown great interest in the project and has supported us during the project's entirety. Furthermore, the project would not have been possible without the help of Image Engineering and Zeuschel and their respective image quality evaluation software. An additional thanks go to Image Engineering for providing a free trial of their software.

We would also like to thank our family, friends, and fellow students for their help proofreading the report. Lastly, we would like to thank each other for our great collaboration throughout the project and for having an excellent educational environment.

Project Repository

The code for the project can be found in the public Git repository on GitLab:
https://gitlab.com/bachelor_group9_2022/kvalitetssikring-av-digitisering.

Contents

Abstract	iii
Sammendrag	v
Preface	vii
Project Repository	ix
Contents	xi
Figures	xv
Tables	xvii
Code Listings	xix
Acronyms	xxi
1 Introduction	1
1.1 Background	1
1.2 Task definition	3
1.3 Target group	4
1.3.1 Software	4
1.3.2 Report	4
1.4 The group's background	4
1.4.1 Academic background	4
1.4.2 Technical background	5
1.5 Goals and Limitations	5
1.5.1 Project goals	5
1.5.2 Limitations	6
1.6 Role distribution	7
1.7 Report structure	8
2 Requirements	11
2.1 Functional requirements	11
2.2 Non-functional requirements	12
2.3 Open Source	12
2.4 Use cases	13
3 Theory and Technologies	17
3.1 Task field	17
3.1.1 Importance of quality assurance	17
3.2 Theory	19
3.2.1 ISO 19264-1:2021	19

3.2.2	FADGI	20
3.2.3	Test targets	21
3.3	Technologies	24
3.3.1	Python	24
3.3.2	Pytest	24
3.3.3	Flask	24
3.3.4	Pyexiv2	24
3.3.5	React	24
3.3.6	Jest	25
3.3.7	SonarQube	25
3.3.8	IQ-Analyzer-X	25
3.3.9	OS QM-Tool	25
3.3.10	JHOVE	25
3.3.11	XMP	26
4	Development Process	27
4.1	Methodology	27
4.1.1	Scrum board	29
4.1.2	The columns	29
4.1.3	Sprints	30
4.2	Git workflow	32
4.2.1	Pipeline	33
5	Technical Design	35
5.1	Program Flow	36
5.2	Software Architecture	37
5.3	Backend	38
5.3.1	Choice of language	39
5.4	Frontend	39
6	Implementation	41
6.1	Backend	41
6.1.1	Endpoints	42
6.1.2	Validation and Analysis	43
6.1.3	Data structure	47
6.1.4	Embedding metadata	48
6.1.5	JSON Results file	49
6.2	Frontend	50
6.2.1	Uploading images for analysis	51
6.2.2	Receiving results	53
6.2.3	Downloading images	53
7	Quality Assurance	55
7.1	Code review	55
7.2	Testing	56
7.3	Development Environment	57
7.4	CI/CD Pipeline	58
7.4.1	SonarQube	58

8 Discussion	61
8.1 Group environment	61
8.2 Time management	61
8.3 Scrum process	63
8.4 Choosing image quality evaluation software	63
8.5 Report	64
8.6 Product discussion	65
8.6.1 Licenses	65
8.6.2 Quirks	66
9 Conclusion	67
9.1 Results	67
9.2 Future work	68
Bibliography	71
A Cooperation agreement	75
B Project Description	83
C Project plan	87
D Sprint 1 - Review and Retrospective	113
E Sprint 2 - Review and Retrospective	117
F Sprint 3 - Review and Retrospective	123
G Sprint 4 - Review and Retrospective	129
H Sprint 5 - Review and Retrospective	135
I Backend Documentation	141
J Wireframe of the frontend	157
K Meeting Notices and Minutes with Client	163
L Meeting Notices and Minutes With Supervisors	193
M Manual for KAD	235
N Timelog	253

Figures

1.1	The two different types of target used in the digitization process . .	2
1.2	Role distribution	7
2.1	Use case diagram	15
3.1	Information loss in a digitized map	18
3.2	Unwanted artifacts in a digitized blueprint	19
3.3	GoldenThread/DICE target	22
3.4	Image including object level target	23
4.1	The Scrum Process	27
4.2	Scrum board in the middle of sprint 5	29
4.3	Example of branching	32
4.4	Merge Request from the project	33
5.1	Flow Chart	36
5.2	MVC architecture	37
5.3	REST API structure	38
6.1	API documentation	41
6.2	Figma wireframe	51
6.3	Result overview page with download button	54
7.1	Example of code review	56
7.2	Pipeline jobs	58
7.3	SonarQube Project Overview	59
7.4	Example of SonarQube Security Hotspot	60
7.5	Example of SonarQube code duplication warning	60
8.1	Excerpt from the time logs	62
8.2	Image Quality Evaluation Software comparison	64

Tables

2.1	Use case 1: Analyze device level targets	13
2.2	Use case 2: Analyze object level targets	14
2.3	Use case 3: Download analyzed images	14
3.1	Comparison of FADGI stars and ISO 19264-1:2021 levels	20

Code Listings

6.1	Results endpoint example	42
6.2	Analyze request examples	43
6.3	Result request example	43
6.4	Download request example	43
6.5	JHOVE validation	44
6.6	IQX subprocess spawning	45
6.7	OS QM-Tool example result file	46
6.8	File system structure in a session	47
6.9	Metadata embedding	48
6.10	Example of results file JSON structure.	49
6.11	Uploading images to the frontend	51
6.12	Uploading images to the API	52
6.13	Fetching results from API	53
6.14	Downloading images from API	54

Acronyms

API application programming interface. iii, v, 5, 6, 11, 32, 38–42, 52, 66–68

CIELAB International Commission on Illumination L* a* b*. 22

FADGI Federal Agencies Digital Guidelines Initiative. 2, 5, 20, 21, 25

IKAMR Inter-municipal archive of Møre og Romsdal IKS. iii, vii, 1–3, 6, 8, 13, 17, 38, 65, 67, 69

IQX iQ-Analyzer-X. 25, 31, 36, 43, 45, 46, 64, 65

JSX JavaScript XML. 24, 39

OCR Optical Character Recognition. 20, 21

OQT OS QM-Tool. 25, 31, 36, 43, 45, 46, 64, 65

REST Representational state transfer. iii, v, 5, 12, 38, 39, 42, 67

TDD test driven development. 56, 57

UTT universal test target. 25

XML Extensible Markup Language. 26, 31, 38, 39, 46

XMP Extensible Metadata Platform. 6, 12, 24, 26, 49

Chapter 1

Introduction

1.1 Background

Inter-municipal archive of Møre og Romsdal IKS (IKAMR) with County center for digitization of cultural heritage (SEDAK) is responsible for digitizing archives for libraries, museums, and cultural preservation institutions in *Møre og Romsdal*. SEDAK was established by *Møre og Romsdal* county in 2013. Digitizing is the conversion from analog to digital information without adding or changing functionality with digital tools. [1] This work allows the analog material to be protected from damage while still providing digital access to the material. The digital files can be secured for the future through long-term digital preservation, and the contents can be digitally publicized for dissemination and research. [2]

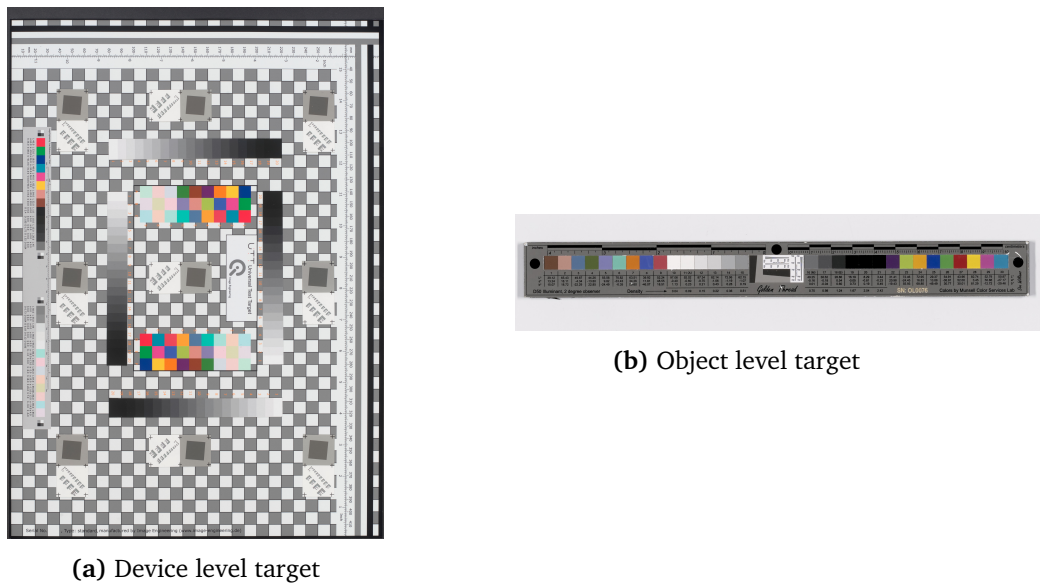


Figure 1.1: The two different types of target used in the digitization process of still media.

(a) Placed in the entire scene

(b) Placed alongside image in the scene

IKAMR's current digitization process is manual and includes multiple steps. When they receive the material, it is first captured using either scanners or cameras. To be able to assure the quality of the captured material, a sample test pattern called a test target is included in the captured scene, either alongside the material or separately. The two different types of targets are shown in figure 1.1. These reference targets include known values that can be used to validate the digitized material. There are two main types of target: device level (figure 1.1a) and object level (figure 1.1b). Object level targets are smaller and can be placed next to objects in images, while device level targets cover the entire image. The targets are further explained in section 3.2.3.

IKAMR currently uses software called GoldenThread¹ and ImageZebra² to analyze the captured targets. IKAMR's digitization workload allows them to repeat this process until they have a result that satisfies three stars or higher in the *Technical Guidelines for Digitizing Cultural Heritage Materials* by Federal Agencies Digital Guidelines Initiative (FADGI). The guidelines are explained in section 3.2.2 with a description of the different levels. After this threshold is reached, they will upload the digitized material to the Digital Archives and keep the raw files stored locally.

¹http://www.imagescienceassociates.com/mm5/merchant.mvc?Screen=CTGY&Store_Code=ISA001&Category_Code=GT, visited 19.04.2022

²<https://imagezebra.com/>, visited 20.04.2022

The Digital Archives is a website first created in 1998 for publishing digitized archival material and is now developed by The National Archives of Norway. The Digital Archives makes it possible to search through structured data across archives. It is also possible to view photos and videos and hear digitized sounds. [3]

The Digital Archives is Norway's joint solution for receiving, preserving, and publishing digitized and media-converted historical archives from state, municipal, and private actors. Archive institutions throughout Norway publish their archives to the Digital Archives. [4]

When the archives are digitized, quality assurance is crucial, especially considering the digitized images are considered cultural heritage. The group has been asked to follow the ISO 19264-1:2021 standard, which defines specific requirements for resolution, noise, distortion, scanning errors, and others. [5] An ISO standard is a standard that is internationally agreed upon and defined by experts. [6] Currently, the image quality analysis is done manually by the employees who upload the material. If this process could be automated, it would save time, and human error during quality assessment could be reduced. Therefore, SEDAK, in collaboration with The National Archives, wished to have an automated quality assurance process developed.

1.2 Task definition

The project aims to define and develop an automated quality assurance module for digitized still media, like images of documents, maps, items, and others. Initially, the module was to be integrated with the Digital Archive's upload system, but this was changed due to complications explained in section 8.6. Therefore, the group split the project into two standalone modules: a backend and a frontend.

The backend would be the main part of the project, as it should fulfill the goals of the client, IKAMR. It should be able to analyze images using a chosen image quality evaluation software, parse the results, embed the results in the image file, and make it available to the user. Furthermore, the backend is accessible over the network so that it can be integrated into applications running on the same network. It is important to note that the backend is not a file management system, so the goal of the application is not to be able to find, delete and manage the data but rather to perform an analysis, get the results, and download the images with the results embedded into the images.

Embedding the result in the images themselves has not been done within the field previously and is a new method of saving the results from the analysis.

The frontend should consist of a user interface and serve as a usable example of how the backend can be utilized. The user should be able to select images to upload, choose which image quality evaluation software to do the analysis, and

see the analysis results.

1.3 Target group

This section is divided into two parts; the target group for the software itself and the target group for the report.

1.3.1 Software

- **Archive institutions and their employees:**

The software will primarily be used within an archive institution, and users related to these are therefore natural target groups. Currently, archive institutions and their employees have to manually perform the quality analysis using the various software they have available. Their current workflow leads to a few problems and time sinks that the group hopes to solve:

- Wrong configuration of software
- Only one person can use the software at a time
- Human error when manually conducting quality assurance

- **Developers working on quality analysis software, both internal and external, in the Digital Archives:**

The groups application was designed around two common workflows, but it might not fit the needs of everyone. Therefore, developers are also a target group. They can use the project as a base to build their required functionality. If they do not want to use the project, they can use the concepts to create something that fits their needs better.

1.3.2 Report

- **People working in archive institutions and the general image quality analysis industry:**

One of the target groups for the report is archive institutions. They may wish to use the information described in this report to develop their own automatic quality assurance module for their use case.

- **Students in their third year of studies towards a bachelor's degree:**

The report is also written with students in mind, considering they might wish to use this project as a reference for their own in the future.

1.4 The group's background

1.4.1 Academic background

The group consists of three students studying for a degree in Bachelor of Engineering in computer science at NTNU in Gjøvik. The most relevant courses the

students have taken are Cloud Technologies (PROG2005)³ and Web Technologies (PROG2053)⁴. Cloud Technologies teaches about how application programming interface (API)s are used to connect computer or computer programs, which was actively used during the design and implementation phase of the project. Web Technologies teaches mostly about frontend development with React, which incentivized us to continue using React during the development of the frontend. The students have taken a course in user-centered design⁵ which was useful during the design of the frontend.

1.4.2 Technical background

The group has previous experience in software development and programming. The group's most valuable knowledge was their previous experience in Python and JavaScript. However, the group did not have any experience in quality analysis of scanned material, and therefore some time was spent learning about this subject.

There were a few concepts and utilities that the group members learned about during the project. One of them was the use of image targets and the difference between them. There are multiple image targets (section 3.2.3) available, some cover the entire frame, and some are included with the picture. Those targets also might only comply with one standard. During the project, the group members learned about two image analyzing standards, *Technical Guidelines for Digitizing Cultural Heritage Materials* by FADGI and ISO 19264-1:2021. The group also had to research and evaluate different image quality evaluation software.

1.5 Goals and Limitations

1.5.1 Project goals

The project goals are divided into three categories: result, impact, and learning objectives. Result goals are the objectives the project should achieve, the impact goals are what the project should achieve for the client, and the learning objectives are what the students should learn during the project.

Result goals

- Develop a web-based, automatic quality assurance module that can be integrated with the Digital Archives backend.
- The backend should be an API that fulfills Representational state transfer (REST) constraints.

³<https://www.ntnu.edu/studies/courses/PROG2005>, visited 13.05.2022

⁴<https://www.ntnu.edu/studies/courses/PROG2053>, visited 13.05.2022

⁵<https://www.ntnu.edu/studies/courses/IDG1362#tab=omEmnet>, visited 13.05.2022

- Develop a backend that analyzes and validates digitized images by the ISO 19264-1:2021 standard.
- Save analysis result as embedded Extensible Metadata Platform (XMP) metadata.
- Be able to download the images after analysis is complete with the metadata embedded.
- Develop a user-friendly frontend that showcases how the API can be used.

Impact goals

- Automate quality assurance for digitizing analog material, which will save time for the employees.
- Reduce human error that occurs as a result of the current manual quality assurance workflow.
- Streamline the quality assurance assessment, enabling all content uploaded to the Digital Archives to be quality assured.
- Streamlining the quality assurance will reduce workflow, enabling the client to upload more material to the Digital Archives.

Learning objectives

- Learn how to develop a user interface that is intuitive and user-friendly.
- Experience in using Scrum in a larger-scale project.
- Further improve the programming and software development competency of the group members.
- Learn how to use and integrate third party software in a project.
- How to perform quality assurance of the codebase by using a CI/CD pipeline.

1.5.2 Limitations

The group's responsibility was to develop a module for quality assurance. The digitization process, where the images are scanned and converted into digital images, was not a part of the project scope. This product is developed for internal use at IKAMR and the Digital Archives. The project description recommended using PHP and Python because they are the languages used internally at the Digital Archives. The group decided to use Python for the backend and React for the frontend. These choices are further explained in section 5.3.1 and 5.4. Initially, the plan was to integrate the solution into the Digital Archives. However, the plan changed to make an API that can be easily set up anywhere and can be integrated into a frontend.

A significant limitation for the project was the image quality evaluation software. These software systems have expensive licenses and are not designed to be automated outside of the existing functionality. They include some support for using the applications through a command line, but this functionality seems to be an afterthought. It also took some time before the group got access to the software.

The test targets could have minor inconsistencies which need to be accounted for. The inconsistencies are accounted for by calibrating the analysis software using custom reference data instead of the default reference data. However, this functionality is not a part of the groups application and has not been considered due to time limitations.

1.6 Role distribution

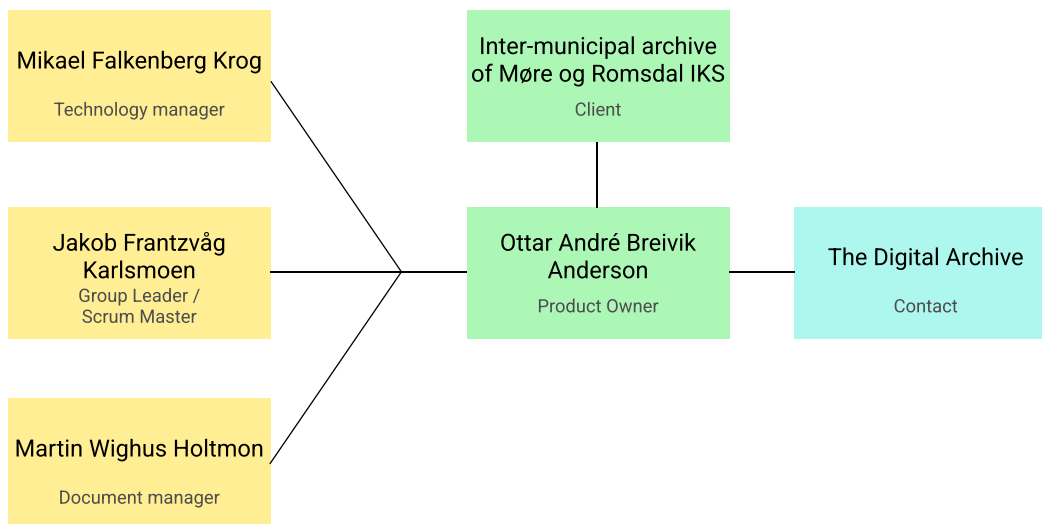


Figure 1.2: Role distribution

During the project, some responsibilities were assigned to different group members. A graph of the role distribution and how they are connected is shown in figure 1.2 and further explained below:

- Document manager - Martin Wighus Holtmon
 - The document manager was responsible for writing meeting summons and meeting minutes. There was also the responsibility of assuring the documents produced during the project were in good condition.
- Group leader and Scrum master - Jakob Frantzvåg Karlsmoen
 - The responsibilities of the group leader and scrum master were to ensure the group worked efficiently and steadily. Furthermore, the group leader was responsible for structuring and administrating the Scrum process.
- Technology manager - Mikael Falkenberg Krog
 - The technology manager was responsible for ensuring all developers were using a good Git workflow and for researching technology that

would be relevant to the project.

- Developer - Martin Wighus Holtmon, Jakob Frantzvåg Karlsmoen, Mikael Falkenberg Krog
 - The developers were responsible for following the group's code standards to ensure all code was written with equal quality.

In addition to the group members, other people took part in the project:

- Product owner - Ottar André Breivik Anderson
 - The product owner was the group's contact at IKAMR and had the responsibility of choosing which direction the development should take.
 - Informed the group about existing image quality evaluation software
 - Provided access to the full ISO 19264-1:2021 standard.
 - Informed the group about the user's needs and requirements for the automatic quality analysis module.
- Supervisors - Rune Hjelsvold, Johanna Johansen
 - The supervisors had the responsibility to help the group through the project and provide the group with help and feedback on their work.

1.7 Report structure

Chapter 1 - Introduction

Chapter 1 introduces the project by providing a project description and background on the topic. This chapter also introduces the group's background and organization during the project.

Chapter 2 - Requirements

The requirements chapter defines the project requirements and use cases for the software.

Chapter 3 - Theory and Technologies

A deeper explanation of the task field and the objective for the project. It also contains a walkthrough of theory and technologies used as a basis in the project. Finally, this chapter provides background that might be necessary to understand future chapters.

Chapter 4 - Development Process

Chapter 4 details how the group planned and performed the software development and what methods were chosen, and why.

Chapter 5 - Technical Design and Implementation

The technical design and implementation chapter details the program flow and architecture of the application.

Chapter 6 - Implementation

Chapter 6 describes how the group planned and developed the two modules and how they function and process data.

Chapter 7 - Quality Assurance

Chapter 7 details how the group did quality assurance of the code and what tools were used.

Chapter 8 - Discussion

The discussion chapter contains around the solution itself and the choices made during the project are described in this chapter.

Chapter 9 - Conclusion

A conclusion detailing whether the project has achieved the goals set at the start of the project, and further work that group noted, but did not do during the project.

Chapter 2

Requirements

In this chapter, the requirements for the system are defined. The requirements are categorized into two groups, functional and non-functional requirements. Functional requirements define what the system must do, and non-functional requirements describe how the system works. The requirements must be well thought out and documented to have a successful project and deliver a product that performs as expected. Furthermore, it will ensure that the system works as anticipated because it will help the developer understand the system's and the product owner's needs. [7]

The group defined a set of requirements based on the project description in collaboration with the product owner, as well as the research in sprint 1 (section 4.1.3). However, the project description has loose guidelines, so the requirements were revised multiple times. For instance, it was defined in the project description that the solution was to be integrated into the Digital Archives backend, but this was later discarded. The project requirements were then changed, and it was decided that the application should be split into a backend API and a frontend to demonstrate the API's functionality.

2.1 Functional requirements

The functional requirements were decided on by the group collectively. The requirements were based on input from the product owner, the task definition, and what the group deemed achievable in the project's duration.

The functional requirements for this project are defined as:

- The application uses image quality evaluation software to analyze and classify the images as level A, B, or C according to ISO 19264-1:2021 using image quality evaluation software.
- The application validates the files and ensures they are not corrupt or broken.

- The application embeds the analysis results as XMP metadata in the image.
- The backend of the application is a RESTful API.
- The application has a web-based user interface.
- The user can choose the type of analysis performed, either device level or object level.
- The user can specify the target and image quality evaluation software used to perform the analysis.
- The user can upload image files in JPEG or TIFF format, which contains test targets, and view the analysis results.
- The user can download the images with results embedded as metadata.

2.2 Non-functional requirements

Non-functional requirements define how a system is expected to operate or behave, rather than specific functional operations defining what the system is supposed to do. [8] Usually, non-functional requirements are quality attribute criteria for evaluating how a software system should perform, for example: portability, security, usability, maintainability, testability, scalability, and extensibility. [9] It essentially describes how a system should behave and what limits there are on its functionality.

The reason non-functional requirements are defined is to ensure that the program meets user or stakeholder expectations. If the expectations are not satisfied, it could give a negative user experience which implies poor product quality. [10]

The non-functional requirements for this project are defined as:

- The application should allow to run multiple analyses at once to fully utilize the systems resources.
- If many analyses are initiated, the system should queue the analyses to not overload the system.
- The frontend should be easy to use, even for people with low technical experience.
- The applications architecture should allow for easy implementation of additional quality analysis software.

2.3 Open Source

The client wished for the project to be open source, and the group agreed that this was a good idea. The primary purpose behind the decision to make the project open source was to allow for information sharing and collaboration between archive institutions and other entities in the image quality analysis industry, as described in section 1.3.

To have a good open source codebase, some formal guidelines had to be decided. The guidelines will provide standardization in the codebase and improve the codebase's quality. The defined guidelines for the codebase are that:

- Everything is written in English.
- The code should be cleanly formatted and easy to read.
- Everything should be documented.
- Follows the same code standard, for example, snake case naming convention for variable names.
- It should be easy for others to continue the development.

2.4 Use cases

These use cases describe specific tasks the user will be able to perform using the software. The goal of the project is to analyze scanned images, view the results, and download the images with the results embedded as mentioned in the task definition (section 1.2). Therefore, there are no use-cases defining file management, as for instance, deleting or managing the results.

The reason use cases were chosen instead of user stories or scenarios was that the functionality itself is more important than the intent and motive of the user. Furthermore, the context of the situation is also not very important, considering the group does not know precisely how the the backend will be integrated by IKAMR.

The list of use cases is as follows:

- Device level target analysis
- Object level target analysis
- Download analyzed images with analysis information as embedded metadata

Use-case:	Perform device level target analysis.
Actor:	Employee at archive institution
Goal:	Be able to view analysis result
Description:	Uploads the start target, end target and corresponding images and is able to view the results as the targets are analyzed.

Table 2.1: Use case 1: Analyze device level targets

Use-case:	Perform object level target analysis.
Actor:	Employee at archive institution
Goal:	Be able to view analysis result
Description:	Uploads images with an object level target and is able to view the results as the targets are analyzed.

Table 2.2: Use case 2: Analyze object level targets

Use-case:	Download analyzed images.
Actor:	Employee at archive institution
Goal:	Be able to download analyzed images with the analysis result as embedded metadata
Description:	After the images have been analyzed, they can be downloaded using a button on the frontend. One can either download all images analyzed or a single image.

Table 2.3: Use case 3: Download analyzed images

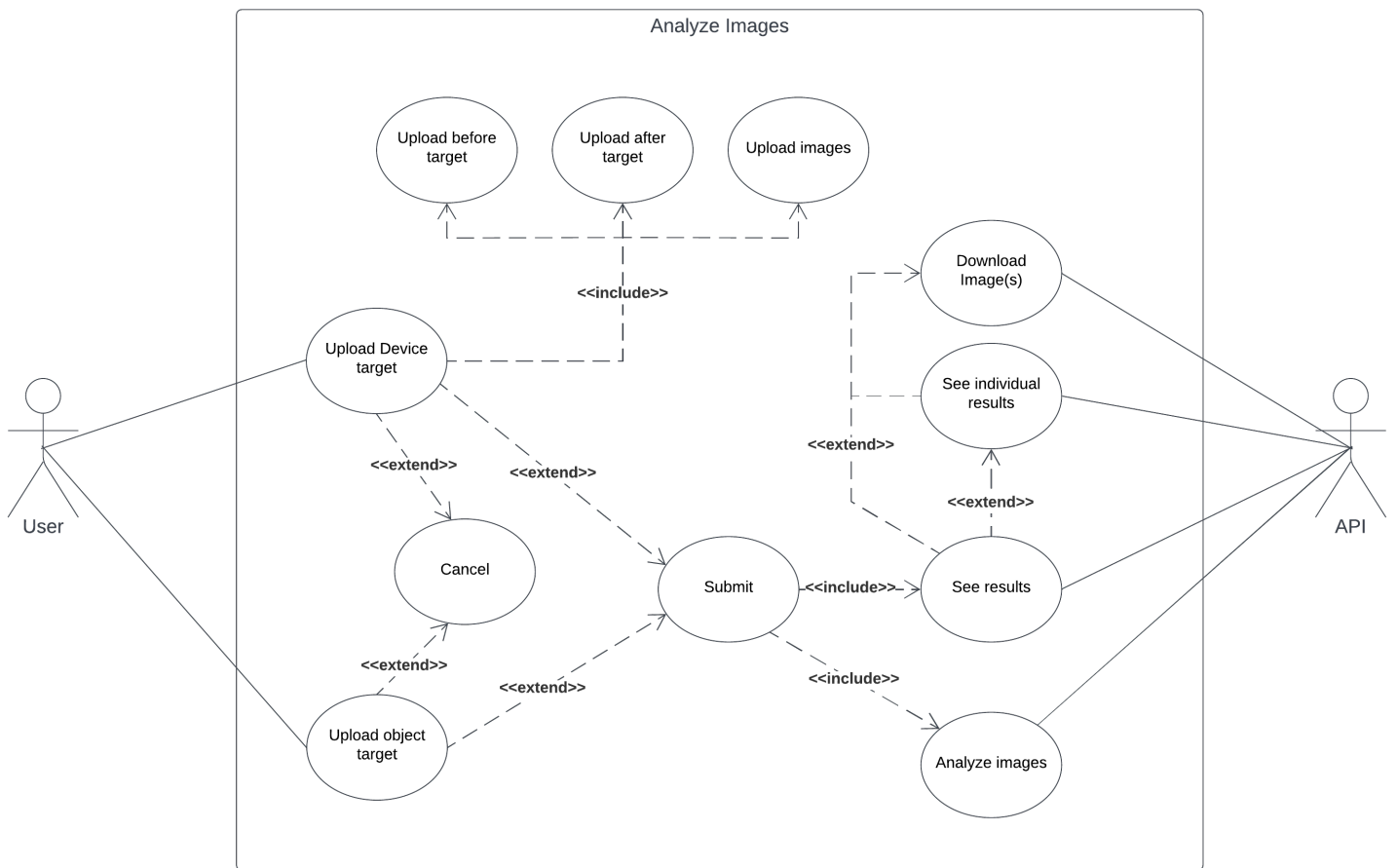


Figure 2.1: Use case diagram

Chapter 3

Theory and Technologies

3.1 Task field

The task field for this project is quality management of cultural heritage. Certain laws in Norway detail the quality requirements for digitized material. [11] It is, therefore, essential that the archive intuitions perform quality assurance on the digitized material to ensure that the cultural heritage is protected. Furthermore, to ensure no data is lost or modified during the digitization process.

In 2017, an audit was performed by the Office of the Auditor General of Norway¹ on the digitization process of cultural heritage. This audit revealed a few problems within the field. For instance, many collections in the Norwegian cultural institutions were still not digitized, and the already digitized material did not have the quality it was expected to have. The consequence is that cultural heritage could be lost. [12]

In 2019, the Digital Archives was used as a national solution for storing digitized cultural heritage. The goal of the Digital Archives project was to enable all users to search and find digitized material from national, municipal, and private archives. [13]

In 2020 IKAMR started a preliminary project on the necessity of quality assurance. This project was also meant to detail how quality assurance should be performed. The archive institutions needed to take responsibility for quality assurance to ensure further trust and development within the sector. [14]

3.1.1 Importance of quality assurance

Digitization of cultural heritage is essential as it allows all kinds of collections to be digitized, thus capturing, preserving, and communicating knowledge to a

¹<https://www.riksrevisjonen.no/om-riksrevisjonen/>, visited 16.05.2022

broader audience. However, in the process of conserving cultural heritage, the digital version must represent the original object to be of any informational value, especially for researchers. Therefore the digitization process must be quality assured. [15]



Figure 3.1: Information loss in a digitized map. The left side is a properly scanned version and the right side has a loss of information.

Figure 3.1 shows an example of why quality assurance is essential. In the left part of the figure, the elevation lines are clearly visible but are not present in the right part of the figure. The loss of information creates a false representation of the area, and by looking at the map, one would be led to believe it is much flatter than it is in reality.

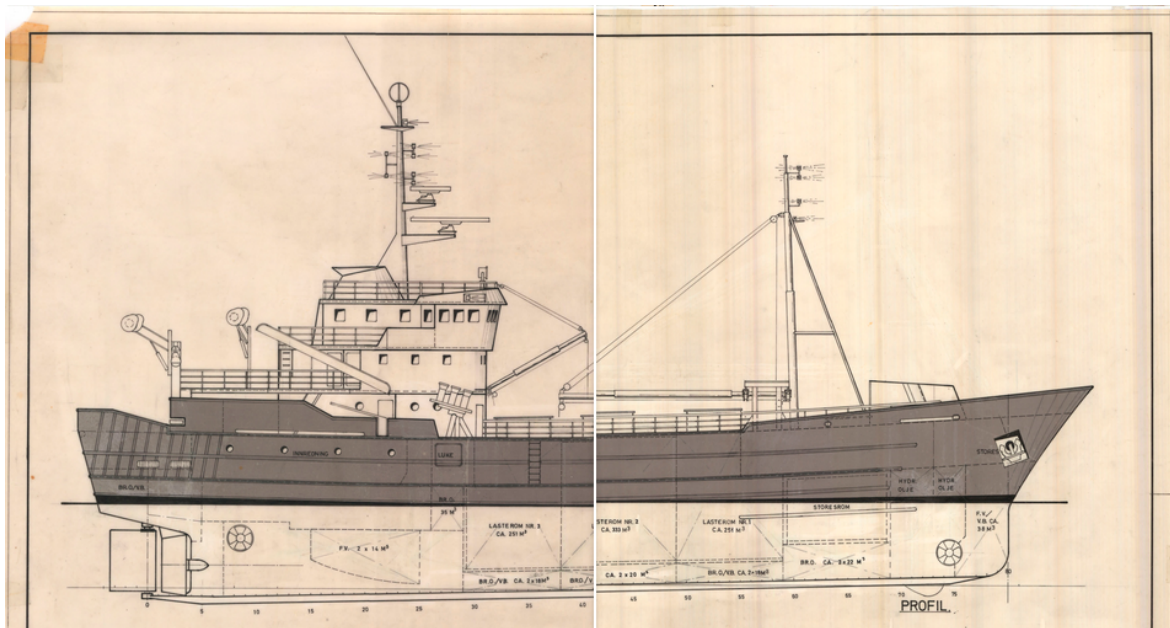


Figure 3.2: Unwanted artifacts in a digitized blueprint. The left side is a properly scanned version and the right side has artifacts described as lines in the image.

The second example in figure 3.2 shows an image with many artifacts. In this case, the artifacts are vertical stripes that span the entire image. These artifacts can easily be confused with actual lines in the original blueprint. Furthermore, the artifacts may lead to the digitized version not being a proper representation of the original version.

3.2 Theory

This section will introduce all the essential theory to build a basis for understanding later chapters. An explanation of what a target is and the different types of targets will be given. The focus will be on the theory around the digitization of still media and the different standards and guidelines most commonly used in the sector.

3.2.1 ISO 19264-1:2021

Digital imaging systems are used for digitizing physical material, such as documents, photos, and maps. The digital versions can have varying accuracy regarding color reproduction, details, distortion, and others. These variations are natural in this process but can be minimized by analyzing the imaging systems.

Several ISO standards define performance characteristics for imaging systems, but the ISO 19264-1:2021 standard combines multiple of them into a document for

digitizing cultural heritage. These performance characteristics include tone range, details, colors, light reflection, and light absorption. [5]

Based on the values from the measurements, the images are graded as A, B, C, or none based on tolerances set in the standard. [16]

- Level C should only be considered to have informational value and may not be suited for Optical Character Recognition (OCR). This level is the minimal accepted level in the ISO 19264:2021 standard.
- Level B is acceptable and usable for OCR and media.
- Level A specifies the best possible quality in practice today. These images are suitable for almost all applications.

3.2.2 FADGI

Federal Agencies Digital Guidelines Initiative (FADGI) is a collaborative effort by federal agencies to create a set of technical guidelines, methods, and practices for digitized and historical, archival, and cultural content. [17] FADGI provides guidelines for audio-visual, still images, and general digital content. However, the relevant standard for this report is the *Technical Guidelines for Digitizing Cultural Heritage Materials*. [18] This guideline will be referred to as the FADGI standard in this report.

The FADGI standard consists of technical guidelines, best practices, and digital imaging conformance evaluation (DICE) application². A combination of these three elements will provide a FADGI compliant digitization environment. [16].

Description	FADGI	ISO 19264-1:2021
Very low quality	One star	No equivalent
Low quality	Two stars	Level C
Good quality	Three stars	Level B
Best quality	Four stars	Level A

Table 3.1: Comparison of FADGI stars and ISO 19264-1:2021 levels

The FADGI standard will provide stars depending on the quality of the scanned images. There are four quality levels, ranging from one star to four stars, where four is the best quality. One can see a comparison between the different quality levels in FADGI standard and ISO 19264:2021 standard in Table 3.1

- One star is generally very low quality and should only be considered informational and is not useful for OCR or other information processing tech-

²<https://www.digitizationguidelines.gov/term.php?term=dice>, visited 13.05.2022

niques.

- Two stars is considered low quality where there is no expectation of achieving three or four stars. This level should still only be considered informational but may be used for OCR.
- Three stars are considered good quality in a professional setting. It can be used for almost all use cases, for example, OCR and could be re-printed as a representation of the original.
- Four stars is considered the best possible quality. This level is the best representation of the original content in a digital form according to the FADGI standard.

[18]

3.2.3 Test targets

Test targets are reference cards that are scanned to evaluate the quality of the scanning system or a single scan depending on the type of test target. The targets contain different reference values, for instance, a known set of colors. When the target is scanned, one can measure the color values of the scanned image against the known values. Multiple standards exist that define thresholds for the maximum amount of deviation between the known and measured values. The standard used in this project is ISO 19264-1:2021. Other standards can be used, for instance, FADGI standard.

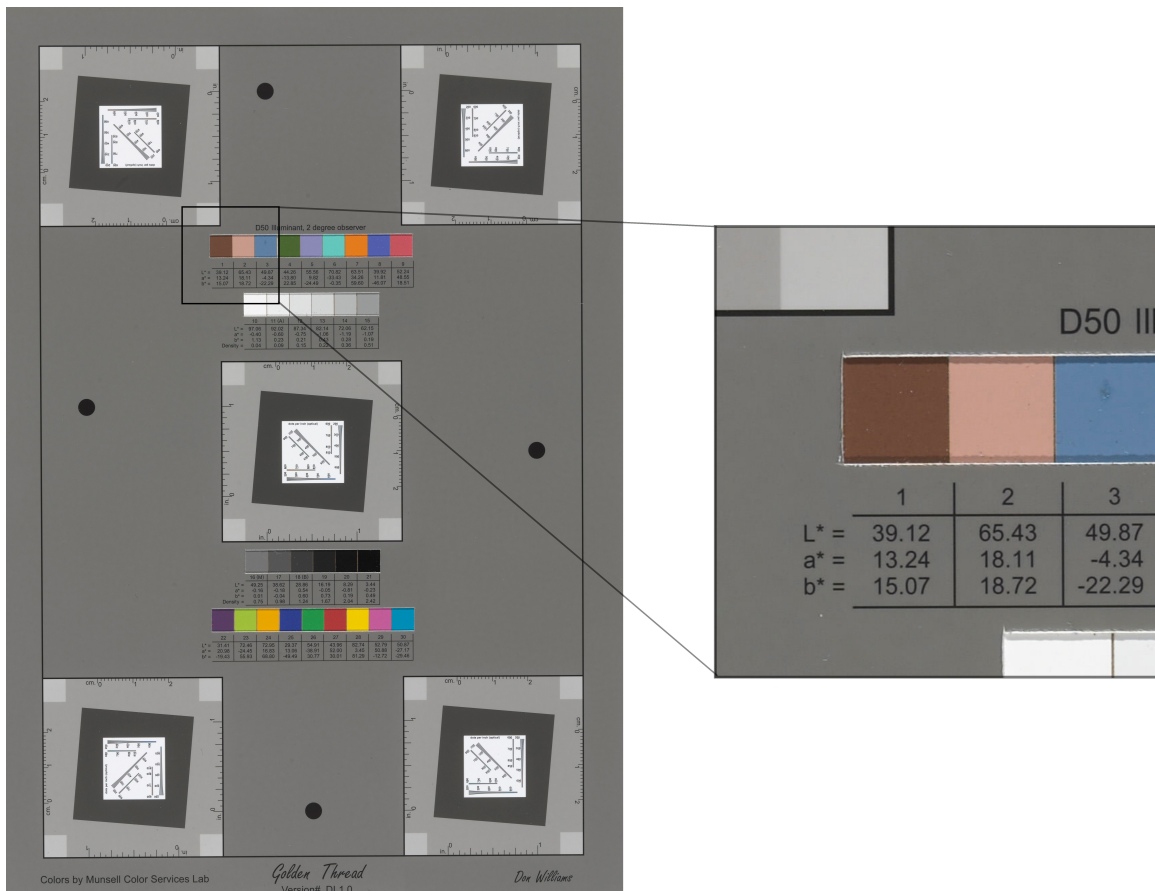


Figure 3.3: GoldenThread/DICE target

Figure 3.3 is an example of a test target. It shows what the reference colors on the targets might look like. Underneath the color itself are the color values. These are represented in the International Commission on Illumination L^* a^* b^* (CIELAB) color space. It is a three-dimensional color space that consists of three axes. The L^* -axis represents lightness with values ranging from 0 (black) to 100 (white), a^* is the red and green axis where positive and negative a^* values describe red and green, and b^* is the yellow and blue axis where positive and negative b^* values describe yellow and blue [19]. This color space is just one example of how these known values can be represented on a test target.

This project uses two types of test targets: Device level and object level targets.

Device Level Target

GoldenThread/DICE (Figure 3.3) is an example of a test target for a system. This target covers the entire scene such that comprehensive visual measurements can be taken for the analysis. For example, a target can have many color palettes, full

grayscale, a dimensional reference, a ruler to indicate scale, and several identical patches scattered across the target to evaluate the consistency of the image. This target is not intended to be added alongside an image but rather be scanned independently. Therefore, this type of measure is best suited for validating the system. The validation is done by including one scan of the target at the beginning and end of a session. [20]

Object Level Target



Figure 3.4: Image including object level target (top)

Object level target is a target that is included alongside an image in the scene (Figure 3.4). The advantage of using an object level target is that the scanned image will have a reference with known values to measure against. A negative side is that it does not have as many references as a device level target due to its size. Object level targets are usually used to only validate colors, tone, DPI, and SFR. [20]

3.3 Technologies

This section introduces the technologies used in this project. Then, the implementation of such technologies is further explained in Chapter 6.

3.3.1 Python

Python is a programming language that was first developed by Guido van Rossum in 1989. Python was originally an object-oriented language, but later it also got support for writing procedural programs. Python is known for its extensive standard library and is often found to be one of the most popular programming languages in the world. [21]

3.3.2 Pytest

Pytest³ is a test framework that allows small and readable tests to be written but can also scale to support complex systems and code.

3.3.3 Flask

Flask⁴ is a micro web framework written in Python that can be used to write both simple and complex web applications.

3.3.4 Pyexiv2

Pyexiv2⁵ is a library which can read and modify metadata of files, including XMP. Pyexiv2 has a few restrictions, like only being able to embed metadata in images with a size less than 1Gb.

3.3.5 React

React⁶ is a JavaScript library for building user interfaces. React makes use of JavaScript XML (JSX), which is a syntax extension to JavaScript that allows HTML to be written intertwined with JavaScript code.

³<https://docs.pytest.org/en/7.0.x/>, visited 24.02.2022

⁴<https://flask.palletsprojects.com/en/2.1.x/>, visited 02.05.2022

⁵<https://github.com/LeoHsiaol/pyexiv2>, visited 02.05.2022

⁶<https://reactjs.org/>, visited 29.03.2022

3.3.6 Jest

Jest⁷ is a testing framework for JavaScript which works with React. The framework is simple to use and makes frontend testing simple.

3.3.7 SonarQube

SonarQube⁸ is an open source platform for continuous code quality inspection. SonarQube offers reports on code quality and security aspects, for instance, code duplication, coding standards, unit tests, code coverage, code complexity, comments, bugs, and security recommendations.

3.3.8 IQ-Analyzer-X

iQ-Analyzer-X (IQX)⁹ is a software for image quality evaluation developed by Image Engineering. It can perform image analysis on many compatible targets. It also has an integrated workflow for performing analysis and classification according to the ISO 19264-1:2021 standard. However, at the time of writing, this feature is only available for the universal test target (UTT). IQX has functionality for performing analysis through a command line.

3.3.9 OS QM-Tool

OS QM-Tool (OQT)¹⁰ is a software for image quality evaluation. The software supports many image targets and offers good integration with other Zeutschel software and scanners. OQT supports creating custom *parameter files* that define thresholds for analyses, and there exist parameter files for standards like ISO 19264-1:2021 and FADGI. OQT also has functionality for running the analysis through a command line, but it is an undocumented and not officially supported feature of the software.

3.3.10 JHOVE

JHOVE¹¹ is an open-source software for file identification, validation and characterization of digital objects. This validation ensures that files are valid and not corrupt or broken. JHOVE has a command line interface, in addition to a graphical user interface. [22]

Other alternatives for file validation were also tested. A few examples are MediaC-

⁷<https://jestjs.io/>, visited 29.04.2022

⁸<https://www.sonarqube.org/>, visited 05.04.2022

⁹<https://www.image-engineering.de/products/software/iq-analyzer-x>, visited 05.04.2022

¹⁰<https://www.zeutschel.de/en/produkte/os-qm-tool-3/>, visited 05.04.2022

¹¹<https://jhove.openpreservation.org/>, visited 09.02.2022

onch¹², KOST-Val¹³ and Bad Peggy¹⁴. However, these programs were not as simple and easy to integrate as JHOVE, as JHOVE can easily be run through a command line.

3.3.11 XMP

Extensible Metadata Platform (XMP) is an ISO standard originally made by Adobe Systems Inc for processing and storing metadata in files. The standard standardizes a data model, a serialization format, and core properties for the definition and processing of metadata. XMP is based on Extensible Markup Language (XML).

¹²<https://mediaarea.net/MediaConch>, visited 11.02.2022

¹³<https://github.com/KOST-CECO/KOST-Val>, visited 11.02.2022

¹⁴<https://github.com/coderslagoon/BadPeggy>, visited 11.02.2022

Chapter 4

Development Process

This chapter will detail the chosen development methodology and framework used during the project, how it was utilized, and the benefits it provided. The chapter will also explain the type of Git workflow the group used and why it was chosen.

4.1 Methodology

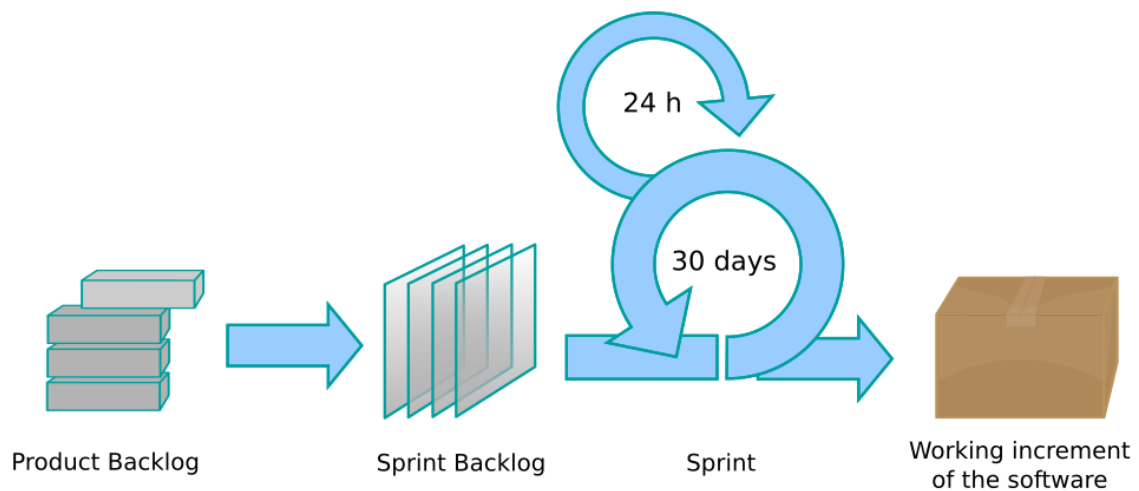


Figure 4.1: The Scrum Process, Lakeworks, CC BY-SA 4.0 <https://creativecommons.org/licenses/by-sa/4.0>, viaWikimediaCommons

The framework the group chose was Scrum; it is an agile method where the development team works in sprints. Each sprint has set goals that should be achieved. These goals are decided during the sprint planning phase, which happens at the start of each sprint. A sprint ends with a review and retrospective, which helps the developers reflect on their work and learn from their experiences. A Scrum team consists of a Scrum Master, a product owner, and developers. An important part of the Scrum process is the Scrum board, which contains all the *issues* sorted into different categories; the scrum board is detailed in section 4.1.1. Every *issue* which should be handled during the project is described in the product backlog. Some of these *issues* are moved into the sprint backlog during the sprint planning. The sprint backlog consists of *issues* that should be solved during the current sprint. [23] The scrum process is shown in figure 4.1.

Kanban is another agile framework that was considered. The Kanban framework visualizes the work using kanban boards and limits the amount of "work in progress" (WIP) to maximize a team's efficiency. Limiting WIP will reduce the amount of nearly done features and force a team to focus on a minor set of tasks. If the team encounter blockages or problems, the whole team can focus on the *issue* to get it resolved instead of pausing the *issue* and moving on to something else. Moving between tasks is inefficient because time will be lost to context switching¹. All the work a Kanban team does revolve around a Kanban board. This board will visualize the team's work and usually include three sections, "To do", "In Progress", and "Done". However, at the team's discretion, more sections can be added to meet the team's needs. [24]

The difference between Kanban and Scrum is that Kanban is continuous and more fluid, while Scrum focuses on short structured sprints with achievable goals aiming to deliver something at the end of each sprint. Both frameworks try to make the development process more efficient. Kanban focuses on optimizing the development flow by limiting WIP, while Scrum tries to improve each sprint by learning through experiences and reflections done after each sprint. Kanban also operates without roles, while Scrum defines project roles. [25]

Scrum was chosen over Kanban mainly because the group members had worked with Kanban before and wanted to try something different to get familiar with more frameworks. In addition, the development period was relatively short, so it was essential to show good progression and receive feedback from the product owner often. Therefore, the meetings were done weekly with the product owner and project supervisors. Another reason for choosing scrum is that reflection is built into the framework through the sprint retrospective. The group has not worked on a project at this scale before, so it is beneficial to have bi-weekly predefined retrospectives to reflect on the process and get input from the supervisors.

The group is a small team and therefore decided not to set specific development

¹<https://blog.rescuetime.com/context-switching/>, visited 06.05.2022

roles. This decision was made to increase each member’s learning outcome and decrease the risk of the group being negatively affected if one of the group members were unable to work.

The scrum board, daily scrums, and weekly meetings gave the group the advantage of having everyone know what is happening at all times and what needs to be done in the current sprint. With the help of a scrum board, it is also possible to see who is working on what. It makes it easy for anyone to pick up a task from the scrum board and work on it.

4.1.1 Scrum board

A Scrum board is a visual way of keeping track of the current state of a project. It consists of *issues* and columns. *Issues* are tasks that the group members can work on, and the columns represent the states an *issue* can have. The board provides an overview of the current *issues*. For instance, if they are active, completed, or pending.

The group further sorted the *issues* by using tags, and the most useful tag was the ”High Priority” tag. It was used for any functionality that needed to be implemented as soon as possible. The tags made it easier for the group members to pick an *issue* to work on. Other tags that were used includes ”Frontend”, ”Backend”, ”Enhancement” and ”Bug Fix”.

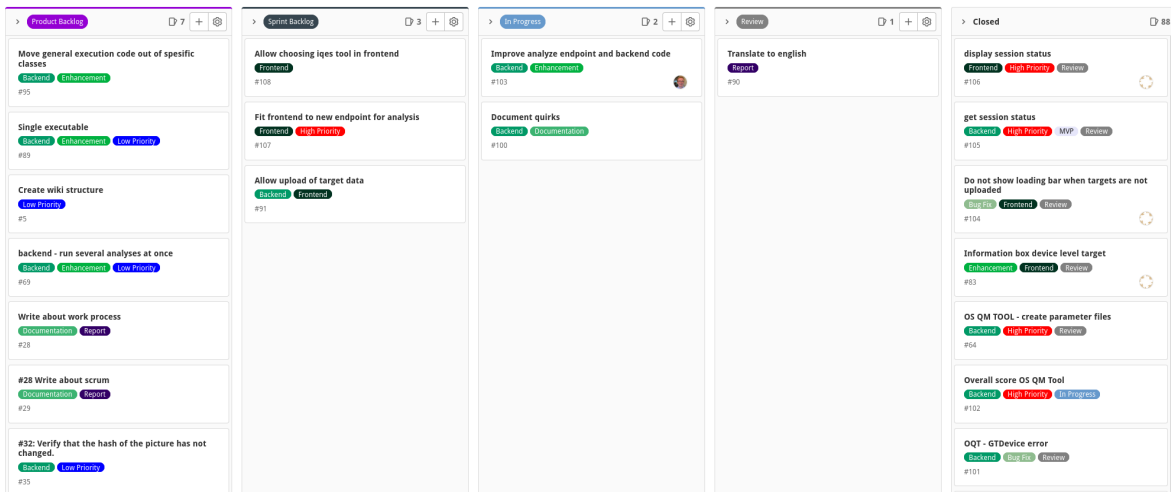


Figure 4.2: Scrum board in the middle of sprint 5

4.1.2 The columns

Product Backlog

The *product backlog* is a list of all technical requirements, business requirements,

and features that must be done to complete the project. This list was continuously updated and expanded throughout the project.

Sprint Backlog

The *sprint backlog* is a prioritized list of *issues* that are to be worked on during a current sprint.

In Progress

The *in progress* column contains the current *issues* that are worked on by the group members.

Review

When the work on an *issue* has been completed, it will be moved to the *review* column, and an MR (merge request) is created. Another group member must review the merge request and its changes before approving it. When approved, the issue moves into the *closed* column, and the merge request gets merged and closed.

Closed

All *issues* that have been completed and reviewed are put in the *closed* column.

4.1.3 Sprints

Sprints are fixed-length events of one month or less to create consistency. A new Sprint starts immediately after the conclusion of the previous sprint. During the sprint, the group works towards the sprint goals set beforehand and works on the *issues* in the sprint backlog. [26]

The group decided on five sprints, which are detailed in section 4.1.3. [27] After every sprint, a sprint retrospective and review were written to reflect on how the sprint went. The review was a summary of what was done during the sprint, and the retrospective was a team discussion of their performance and included ways to improve it. [28] [29]

The group had two weekly meetings; one with the supervisors and one with the client. After each sprint was finished, the next meeting with the supervisors was a sprint meeting where the retrospective and review were presented. The group received feedback on the current workflow and progress in the meetings. All the meeting minutes can be found in appendices K and L.

Sprint 1 (Week 5 - Week 6)

See appendix D for the full review and retrospective report from Sprint 1.

The main goal of the first sprint was to decide on which image quality evaluation software to integrate into the backend module and decide on what software to use for file validation. JHOVE was chosen as the image validation software after testing a few alternatives. However, the group did not have much to do in terms

of development yet, and instead started to work on the main report and defined a suitable MVP for the project. In addition, much research was done within the field. The research was necessary because the group had no previous experience in the field of cultural heritage digitization. The group did not manage to decide on an image quality evaluation software. However, after testing and evaluating a few software, the choice had been narrowed down to four options: GoldenThread, iQ-Analyzer-X, OS QM-Tool, and ImageZebra.

Sprint 2 (Week 7 - Week 8)

See appendix E for the full review and retrospective report from Sprint 2.

The group continued to try out free versions of image quality evaluation software. The goal of this sprint was to start developing code around one or more quality evaluation software. Unfortunately, licenses for the quality evaluation software were challenging to acquire. It turned out not to be possible to get the licenses for the software during this sprint, so instead, the group started designing a prototype for the frontend.

Time was also spent setting up the development environment and configuring the group's quality assurance tools.

Sprint 3 (Week 9 - Week 10)

See appendix F for the full review and retrospective report from Sprint 3.

The majority of the frontend was developed during this sprint. In the middle of the sprint, the group got access to iQ-Analyzer-X and OS QM-Tool, and started to develop code to perform analyses with code.

Sprint 3 was the sprint where the development of the backend started. In addition, parsers for the outputs from the software were also created.

Sprint 4 (Week 11 - Week 12)

See appendix G for the full review and retrospective report from Sprint 4.

The goal of sprint 4 was to finish all planned functionality for the MVP and connect all the pieces to create a working MVP.

In this sprint, the group started to develop the code for the second quality analysis tool, OS QM-Tool, which was integrated into the solution.

Code to embed the quality analysis results as XML metadata in the images was developed in this sprint.

A refactor of the backend was also a part of this sprint.

Sprint 5 (Week 13 - Week 14)

See appendix H for the full review and retrospective report from Sprint 5.

In the final sprint, the goals were stability, quality of life improvements, and documentation. To list a few: JHOVE file validation before and after analysis, consistent logging, allowing selection of target and image quality evaluation software, and a fix for duplicate filenames overwriting each other. In the documentation part of the sprint, the API documentation was updated. Additionally, READMEs were added for the frontend and backend.

4.2 Git workflow

Git was used for version control management, and the group used a trunk-based workflow. This type of workflow is common practice among DevOps teams. These are teams that use DevOps practices to automate and integrate processes. [30] In this type of workflow, minor updates are frequently merged into the main branch. Frequent merging makes it easy to perform code reviews with each merge request since each merge request is relatively small. This workflow is typical when using a CI/CD pipeline because the pipeline is more useful with frequent minor updates rather than large and infrequent ones. [31]

The main branch should always be stable and ready to be deployed. Code reviews and CI/CD pipelines were used to ensure stability. Each merge request had to be reviewed and approved by another person in the group before it was merged. This process is to ensure good code quality.

All branches should be based on the main branch; they are called feature branches. These branches are short-lived and should be merged into the main branch.

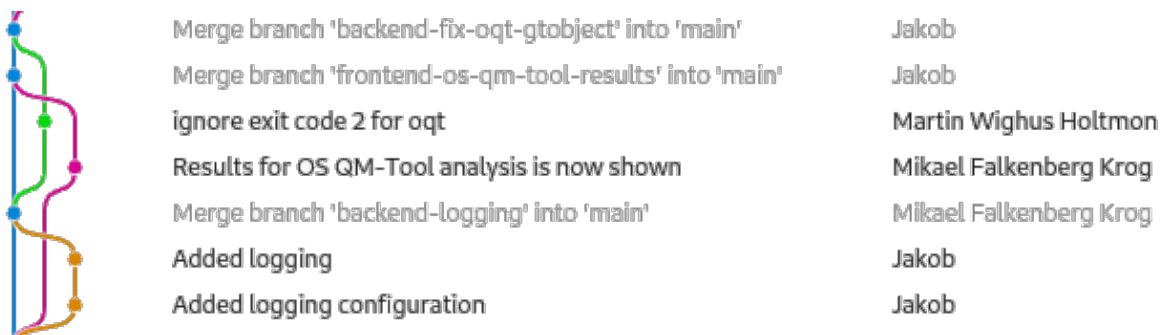


Figure 4.3: Example from the group's git repository showing how branching was used

An example from the project is shown in figure 4.3. The blue branch is the main branch, while the others are feature branches that are later merged into the main branch.

bachelor_group9_2022 > Kvalitetssikring av digitisering > Merge requests > 166

Merged Created 1 month ago by Mikael Falkenberg Krog Owner Code Edit ⋮

Added download buttons for session and individual images

Overview **3** Commits **3** Pipelines **2** Changes **3** 🟢 All threads resolved

Closes #92 (closed)

Request to merge frontend-download-buttons into main

🟢 Pipeline #504780166 passed for 0b5ab7ad on frontend-download-buttons 1 month ago » 🟢

👤 Approved by

🔗 Merged by Martin 1 month ago Revert Cherry-pick

The changes were merged into main with 33ff058f

The source branch has been deleted

Closed #92 (closed)

🟢 Pipeline #504785806 passed for 33ff058f on main 1 month ago » 🟢 🟢 🟢 🟢

👍 1 👎 0 😊 Oldest first Show all activity

Figure 4.4: Merge Request from the project

Figure 4.4 shows a merge request from this project. It was connected to an issue, which is closed when the branch is merged. The figure shows that one other member from the group has approved the request and that the pipeline has passed, then the merge request was merged.

4.2.1 Pipeline

All tests had to pass before merging the request into the main branch. The CI/CD pipeline automatically verifies this. The pipeline runs all tests, and when the branch has been merged, SonarQube runs on the main branch to analyze the code quality, and Jenkins² deploys the frontend. The testing and CI/CD is explained further in 7.2 and 7.4.

²<https://www.jenkins.io/>, visited 03.05.2022

Chapter 5

Technical Design

This chapter will focus on explaining the design, architecture and program flow of the application. This includes both the backend- and frontend module.

5.1 Program Flow

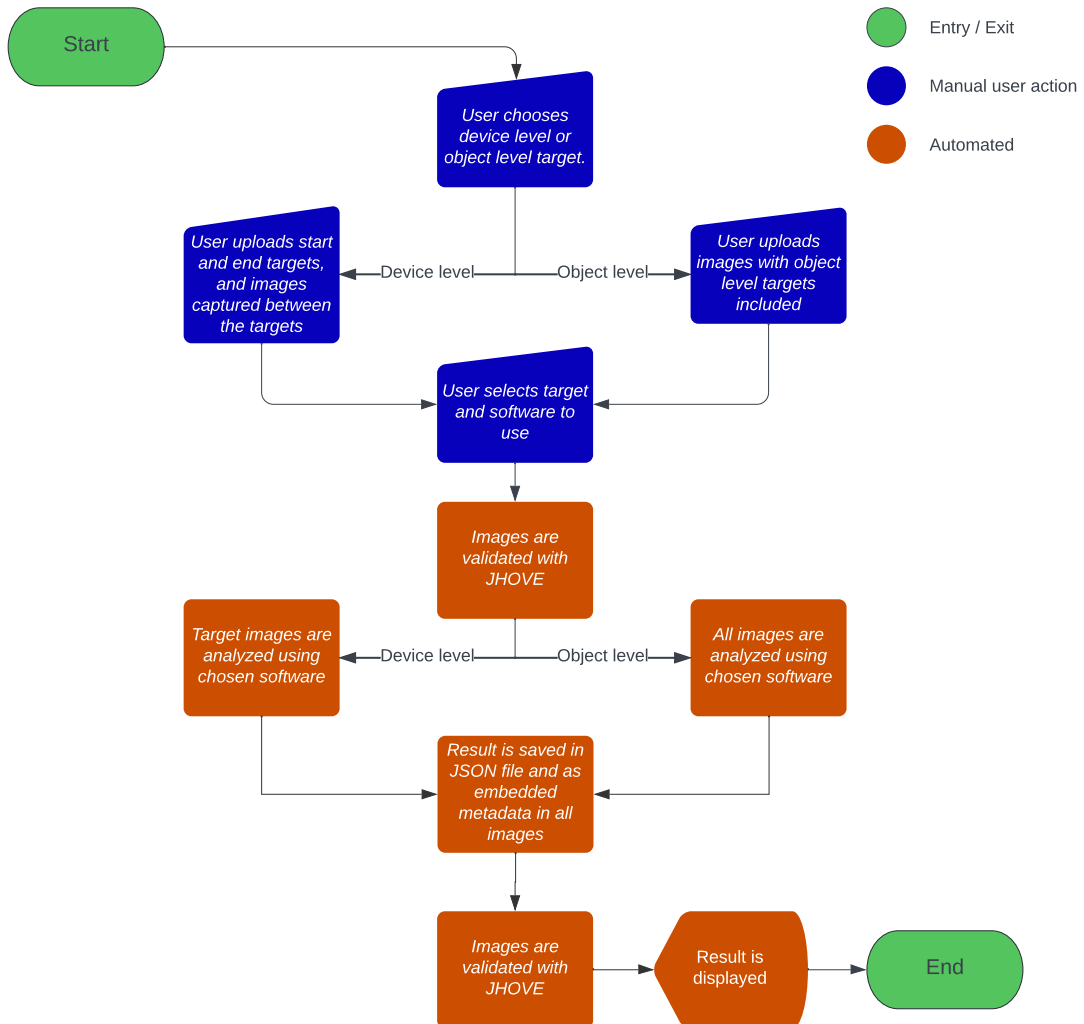


Figure 5.1: Flow Chart

A flow chart was created to make it easier for the group to understand what the overall architecture should look like and to get an understanding of how the application works. In addition, the flow chart makes it easier to communicate how the application works to people without a technical background. The flow chart is primarily determined by user inputs and starts with the user choosing device level or object level as their target type. After choosing a target type, the user uploads the correct targets and images. Afterward, all images are validated with JHOVE and then analyzed using iQ-Analyzer-X or OS QM-Tool based on the user

input. The results are saved in a JSON¹ file and embedded as metadata in the images. Lastly, the images are validated again, and the result is displayed to the user.

5.2 Software Architecture

A clearly defined software architecture is essential in software development, as it helps the developers maintain the software and makes the code easier to understand for external developers.

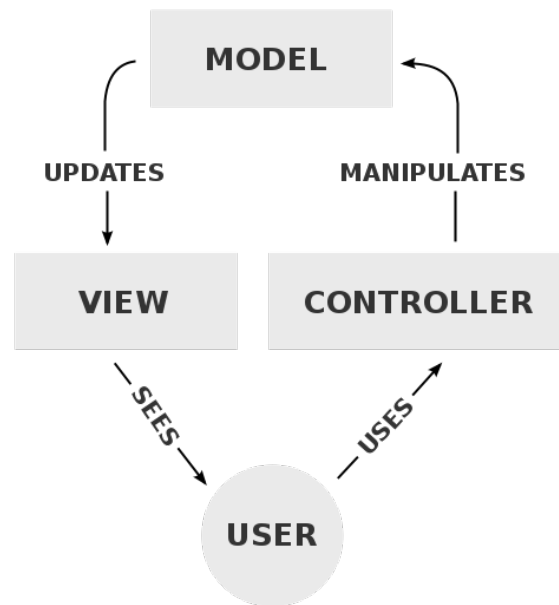


Figure 5.2: MVC architecture, RegisFrey, Public domain, via Wikimedia Commons

The overall architecture of the project is a layered architecture. Each layer has its own responsibility. The type of layered architecture used in the project is Model-View-Controller. It consists of a controller, which the user interacts with, a view the user sees, and a model which implements the domain logic. This architecture is shown in figure 5.2. In an MVC architecture, it is important that the view and model never interact directly but rather through the controller.

In the project, the backend logic itself is the model; this model performs the analysis on images and saves the results. The backend endpoint logic is the controller, and the frontend built using React is the view.

¹<https://www.json.org/json-en.html>, visited 19.05.2022

5.3 Backend

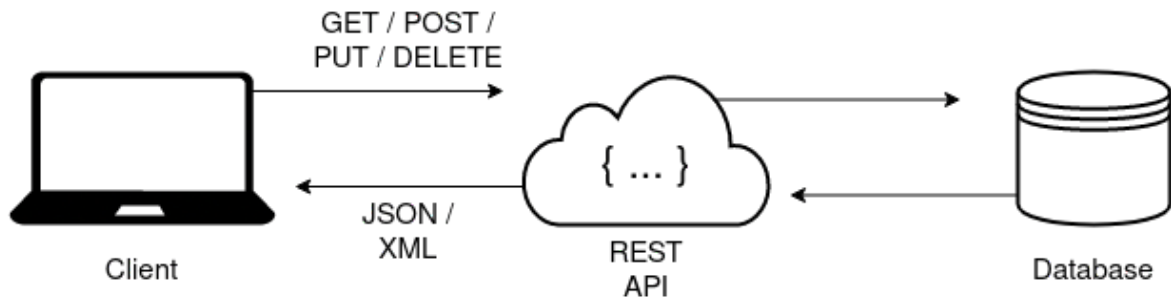


Figure 5.3: REST API structure

The backend module exposes an application programming interface (API) which enables two software components to communicate with each other. The API's scope of use is private, meaning it is intended only to be used internally to connect systems and data within the business. [32] The reason behind making an API was that IKAMR might wish to integrate it into their existing solution for quality assurance. By creating an API, the backend can run on a server and be integrated into an existing frontend by making HTTP requests to the server.

The backend module provides a RESTful API. It means that the API follows the Representational state transfer (REST) constraints. These APIs are based on HTTP methods to access resources using parameters encoded in the URL. The response is typically in JSON or XML format. In this API, all data received is in JSON format, and form data is used to upload files to the API. [33]

The REST architectural style defines six constraints:

- **Client-server architecture**
In this project, the frontend is a client, and the server is the backend. A client exchanges information with the server through the HTTP request and -response protocol.
- **Statelessness**
The server does not store any information about the client session (state of the client) on the server-side. Each request sent to the server must contain all the necessary information to process the request. All the information about the session is kept on the client-side.
- **Cacheability**
A client that makes requests to the API should be able to cache the response. For example, GET requests are automatically defined as cacheable, while POST requests are not.
- **Layered system**
The layered system architecture was implemented by separating the primary concerns in the application.

- **Uniform interface**

This constraint defines the interface between the clients and the servers. For instance, the endpoints should be resource-based, meaning that the URIs contain resource identifiers. The server does not send the database but rather a representation of the database using JSON, XML, or HTML.

- **Code on Demand (Optional)**

This constraint was optional, and the group has not followed this constraint.

5.3.1 Choice of language

When deciding what language to use for the backend, the two languages that were considered were Go² and Python. The group had some previous experience with both languages, and both languages are effective for developing REST APIs. PHP³ was also considered but was quickly ruled out due to the group preferring Python or Go.

After some discussion, Python was chosen as the language for the backend. This was because the group wanted a versatile language with a comprehensive library because of the uncertainty regarding how the image quality evaluation software would operate.

5.4 Frontend

The frontend module was written using React, a frontend library for JavaScript. The group had previous experience using React. Therefore, since the frontend module was meant as a small-scale usable example of how the backend module can be integrated into a system, the group chose a library they had experience using. The group liked JSX and React's flexibility. React is the most favored frontend library among the group. Vue.js and Angular were also discussed, but React was chosen.

The frontend module is a single-page application that switches out components instead of reloading the page. Items that are reused on the page are made into components. These components can either be defined as classes or functions.

In newer versions of React, functional components have almost all the functionality that a class-based component does through the use of hooks. [34] The group decided to use functional components because they are simpler and easier to read, and the group liked writing functional components more than class-based components.

When developing the frontend module, the group took a mobile-first approach. This approach is a design principle where the application is first designed with

²<https://go.dev/>, visited 19.05.2022

³<https://www.php.net/>, visited 19.05.2022

mobile devices in mind. It allows the developers to focus on the most important elements of the user interface. [35] The application will most likely not be primarily used on mobile devices. However, the mobile-first approach is still valuable because of the focus on simplicity and usability.

The frontend module will interact with the backend module by making API requests to it and then displaying the received results.

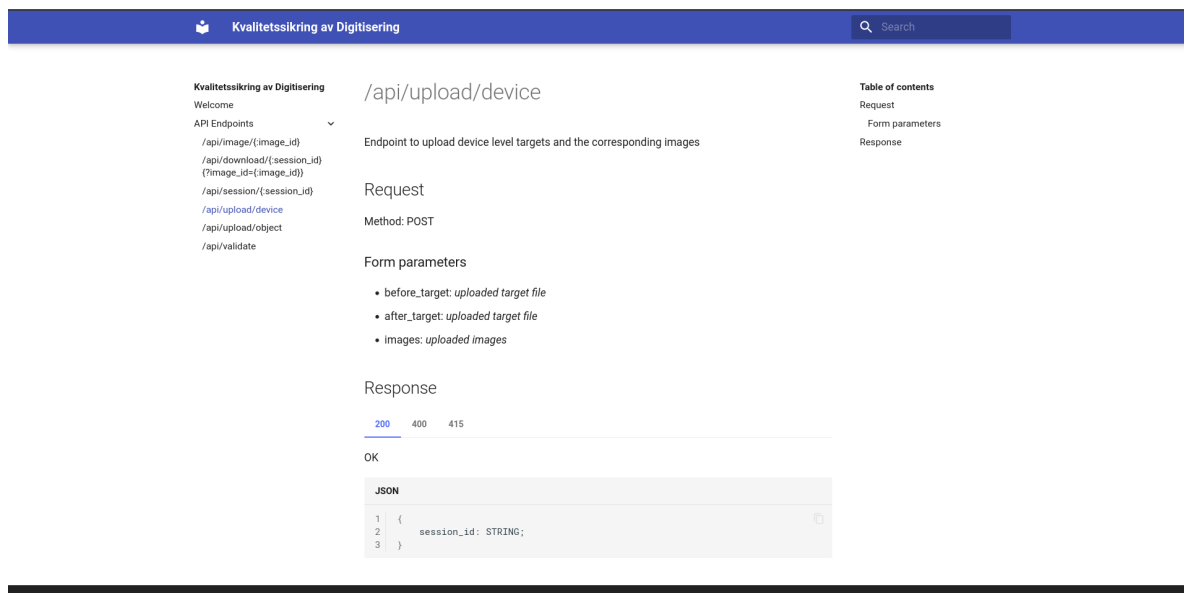
Chapter 6

Implementation

This chapter will present how the backend and frontend were implemented in the project. Some code examples will be provided to explain the implementation of the modules.

6.1 Backend

Before implementing any functionality, the group documented what endpoints the API should have. The backend endpoints can perform image validation, image quality analysis, retrieve the results from the quality analysis and download the images with the results embedded as metadata.



The screenshot shows the API documentation for the endpoint `/api/upload/device`. The page has a dark blue header with the text "Kvalitetssikring av Digitisering" and a search bar. On the left, there is a sidebar with a "Kvalitetssikring av Digitisering" logo and a list of API endpoints: `/api/image/{image_id}`, `/api/download/{session_id}({image_id})`, `/api/session/{session_id}`, `/api/upload/device` (highlighted), `/api/upload/object`, and `/api/validate`. The main content area displays the following information:

- Endpoint:** `/api/upload/device`
- Description:** Endpoint to upload device level targets and the corresponding images
- Request:** Method: POST
- Form parameters:**
 - `before_target`: uploaded target file
 - `after_target`: uploaded target file
 - `images`: uploaded images
- Response:** Status codes 200, 400, 415 are listed. The response is "OK" and a JSON object is shown:

```
1 { session_id: STRING;
2 }
3
```

On the right side, there is a "Table of contents" section with links for "Request", "Form parameters", and "Response".

Figure 6.1: API documentation

Figure 6.1 shows what the API documentation looks like when it has been deployed. MkDocs¹ was used to create the documentation pages. The full endpoint documentation can be found in appendix I. This documentation was written before the endpoints were made and were used as a reference when developing the backend. The API was developed using Flask, a web application framework for Python. The group's previous experience from developing REST APIs was useful when developing the backend.

```
@results_endpoint.route("/api/results/<session_id>", methods=["GET"])
def results(session_id):
    """Method for retrieving the overall results of a session.

    Returns:
        A HTTP Response with corresponding status code and data
    """

    session_results_file = get_session_results_file(session_id)

    if not check_session_exists(session_id):
        return Response(json.dumps({"error": "session does not exist"}), status=404)

    with open(session_results_file, "a+", encoding="UTF-8") as json_file:
        json_file.seek(0)

        if os.path.getsize(session_results_file) > 0:
            json_data = json.load(json_file)

            return Response(json.dumps(json_data), status=200)

        return Response(json.dumps({"error": "no results"}), status=404)
```

Code listing 6.1: Results endpoint example

After all the functionality had been documented, the group started developing the endpoints themselves. The example in code listing 6.1 shows how the results endpoint was implemented. This endpoint took a session id as a parameter and returned the analysis results by reading it from the results JSON file for that specific session.

6.1.1 Endpoints

An endpoint is one end of a communication channel between two systems. [36] The endpoints include a URL of a server or service, and each endpoint is the location of a resource used to serve different functions given a request. In this project, there are four endpoints: starting the analysis of the device level target, starting the analysis of an object level target, retrieving the results, and downloading the images with embedded results. The documentation for the endpoints is shown in appendix I.

¹<https://www.mkdocs.org/>, visited 19.05.2022

```

POST http://[IP:PORT]/api/analyze/device?iqes=IQX&target=UTT
POST http://[IP:PORT]/api/analyze/device?iqes=OQT&target=GTDevice
POST http://[IP:PORT]/api/analyze/object?iqes=OQT&target=GTObject

```

Code listing 6.2: Analyze request examples

The endpoint `/api/analyze/device` is used to analyze device level targets. Two queries must be specified to get a successful request, `iqes` and `target`. The `iqes` query is used to specify the image quality evaluation software, and the `target` query is used to specify the target you are analyzing (see Code listing 6.2 for example requests). Supported parameter values are documented in the endpoint documentation in appendix I.

To analyze object level targets, the endpoint `/api/analyze/object` is used. The endpoint is similar to the device level target endpoint. Queries for the quality evaluation software and the target used must be specified here to get a successful request.

```

GET http://[IP:PORT]/api/results/43d7e711-2548-4ee0-a8b0-916dff7a9513

```

Code listing 6.3: Result request example

After the analysis is done, the endpoint `/api/results/{:session_id}` is used to retrieve the results. The response is a JSON object in the form described in section 6.1.5. An example of the `GET` request to retrieve the results is displayed in Code listing 6.3

```

GET http://[IP:PORT]/api/download/43d7e711-2548-4ee0-a8b0-916dff7a9513
GET http://[IP:PORT]/api/download/
43d7e711-2548-4ee0-a8b0-916dff7a9513?file_name=UTT.tif

```

Code listing 6.4: Download request example

The endpoint `/api/download/{:session_id}` (see Code listing 6.4) is used to download the images from the analysis with the results embedded as metadata. A zip of all the images will be downloaded. The endpoint also enables downloading of a single image with the use of the query `file_name={:file_name}`.

6.1.2 Validation and Analysis

The validation and analysis is the most important part of the backend. JHOVE is used for file validation, and IQX and OQT are used for image quality analysis. The implementation of such tools will be explained in this section.

Validation with JHOVE

JHOVE is used to validate images. JHOVE was detailed in section 3.3.10. If images are corrupt or broken, this will be detected, and the analysis process will be stopped. The images are validated twice, once before the analysis and once after. The second validation ensures that the files are still valid after the metadata has been embedded into the files.

```
def jhove_validation(url):
    """File validation using JHOVE.

    Args:
        url (str): the path of the image which should be validated

    Returns:
        str: output with all the metadata
        bool: True if successful, False otherwise
    """

    # Check if operating system is posix or windows
    if os.name == "posix":
        jhove_command = os.path.join(jhove_path, "jhove")
    else:
        jhove_command = os.path.join(jhove_path, "jhove.bat")
        url = url.replace("/", "\\")

    # Runs the file validation and saves the output to a variable
    if url.lower().endswith(".jpg") or url.lower().endswith(".jpeg"):
        logging.getLogger().info("Validating JPEG file %s with JHOVE", url)

        stream = os.popen(jhove_command + " -m JPEG-hul -kr " + "'" + url + "'")
    elif url.lower().endswith(".tiff") or url.lower().endswith(".tif"):
        logging.getLogger().info("Validating TIFF file %s with JHOVE", url)

        stream = os.popen(jhove_command + " -m TIFF-hul -kr " + "'" + url + "'")
    else:
        logging.getLogger().warning("File %s can not be validated with JHOVE", url)

        return "Filetype not valid", False
    output = stream.read()

    # Checks if the validation was successful
    if "Status: Well-Formed and valid" not in output:
        status = False
    else:
        status = True

    return output, status
```

Code listing 6.5: JHOVE validation

Code listing 6.5 shows how the file validation works. It runs JHOVE with different arguments depending on the file type and checks that the status is "Well-Formed and valid".

Analysis using Image Quality Evaluation Software

An important part of the application is how the image quality evaluation software was used. The currently integrated image quality evaluation software are iQ-Analyzer-X (IQX) and OS QM-Tool (OQT). Parser functions were made for both software to get the analysis result as a Python dictionary. This is explained further in section 6.1.2.

When an analysis is performed, the software scans the given images. What images are scanned is determined by what type of analysis the user requested. For example, if the analysis type is object level, each image is scanned, and if the type is device level, only the two given targets are scanned. The images are scanned three times because the software has to do an individual scan for each level according to the ISO 19264-1:2021 standard. The data of all the scans will be saved, but the overall score for an image will be the best passing score achieved.

```
subprocess.run(
    [
        iqx_executable,
        image_file,
        settings_id,
        reference,
        utt,
        specification,
        exif,
    ],
    timeout=int(
        Config.config().get(section="IQ ANALYZER X", option="SessionTimeout")
    ),
    check=True,
)
```

Code listing 6.6: IQX subprocess spawning

The backend module uses the command line features of the supported image quality evaluation software to perform analyses. The available command line arguments make it possible to specify all the needed parameters for performing an analysis. When an analysis begins, the software is spawned as a subprocess by the backend module process. Once the analysis has been initiated, the application has no control over the image quality evaluation software. If the image quality evaluation software were to freeze or not finish, the subprocess for that analysis would be killed after a set amount of time. Code listing 6.6 shows how this is implemented in code. The timeout of the subprocess is configurable by the user and is specified in the configuration file.

Parser

The different quality evaluation software provides extensive and detailed result reports when an analysis is completed. However, the programs provide different reports, and some are structured, others are not. For this reason, a parser needed

to be written for each program that is added. The parser's job is to extract essential information from the analysis results and provide it in a structured way using dictionaries in Python. According to the requirements in section 2.1, the only information that the program needs to extract is the passed or failed status of each type of analysis that was executed, for example, noise. The goal of the analysis is to classify the images as level A, B, or C according to ISO 19264-1:2021 and thus, does not require extensive information about the different values and deviations reported by the quality analysis software.

```

Check started at: Thu Mar  3 14:11:01 2022
Measurement title:
Used parameter file: C:\Users\Jakob\Desktop\OSQMT00L\runs\UTT\UTT.qmp
Used image file: C:\Users\Jakob\Desktop\OSQMT00L\runs\UTT\UTT.tif

*****
Delta E check

Analyse Delta E (CIE 1976)
Profile: WT25-650
Reference data : C:\ProgramData\Zeutschel\OS QM-Tool\TargetData\UTT_Lab_DIN_A3.txt
Limits
Max Delta E: 25.00
Mean Delta E: 12.00
Result: Passed
Measured Values:
Delta E - max. Delta E Target 1: 14.99
Delta E - min. Delta E Target 1: 1.26
Delta E - Mean Delta E Target 1: 5.33

*****
Noise check

Analyse Noise (Grayscale)
Profile: WT25-650
Result: Passed
Measured Values:
L* UpperHorizontal (Target ID1): 0.92 -> max: 5.00
L* RightVertical (Target ID1): 0.91 -> max: 5.00
L* LowerHorizontal (Target ID1): 1.05 -> max: 5.00
L* LeftVertical (Target ID1): 0.90 -> max: 5.00

*****

```

Code listing 6.7: Example of the result from OS QM-Tool analysis

The results file provided by IQX is a structured XML document consisting of close to 30 000 lines. Although this file includes a lot more information than is needed in the project, it was relatively easy to extract the types of analyses done with the passed or failed status because the results are structured.

Code listing 6.7 displays a short extraction of the summary result file provided by OQT. This result file is not structured, so the process of parsing this was more challenging than parsing the results from IQX. Each line is evaluated, and string manipulation is done to extract only the wanted information when relevant lines are located. If the result file format is changed by the software developers, the

parser will need to be updated. The documentation specifies which version of the software the application works with, but there is no guarantee it will work with future versions of the third party software.

6.1.3 Data structure

The images and analysis results must be stored so they can be accessed. One problem was that none of the image quality evaluation software allowed the result file to be saved in a user-defined location. Instead, the programs always saved the result file in the same directory as the analyzed image. This created problems considering each image had to be scanned three times, one for each quality level according to ISO 19264-1:2021. The solution was to hardlink the image to three separate folders, one for each quality level. Hardlinking enables us to have more than one path reference a single file in a volume [37], which will enable space-saving, reduce unnecessary strain on the storage device, and reduce I/O operations.

```

./
|-- images/
|   |-- IMAGE-1.tif
|   '-- IMAGE-2.tif
|-- outputs/
|   |-- IMAGE-1.tif-analysis/
|   |   |-- A/
|   |   |   |-- IMAGE-1.tif
|   |   |   |-- result_summary.txt
|   |   |   '-- result.txt
|   |   |-- B/
|   |   |   |-- IMAGE-1.tif
|   |   |   |-- result_summary.txt
|   |   |   '-- result.txt
|   |   '-- C/
|   |       |-- IMAGE-1.tif
|   |       |-- result_summary.txt
|   |       '-- result.txt
|   '-- IMAGE-2.tif-analysis/
|       |-- A/
|       |   |-- IMAGE-2.tif
|       |   |-- result_summary.txt
|       |   '-- result.txt
|       |-- B/
|       |   |-- IMAGE-2.tif
|       |   |-- result_summary.txt
|       |   '-- result.txt
|       '-- C/
|           |-- IMAGE-2.tif
|           |-- result_summary.txt
|           '-- result.txt
|-- images.zip
|-- results.json
'-- state.json

```

Code listing 6.8: File system structure in a session

Each session has its own folder, with the folder name being the session id. This id is a randomly generated UUID4². Each session has an images folder with all the images, an outputs folder, a zip file with all the images with embedded metadata, and two JSON files: results and state. The results file saves the overall result for each image, while the state file saves the session's state. For instance, while the session is analyzing the images, the state is "running". The outputs folder contains folders for each image, and each of these folders has three folders: A, B, and C. The results from the analysis programs are saved in these folders and later parsed and saved in the results JSON file. See figure 6.8 for an overview of the folder structure.

6.1.4 Embedding metadata

The metadata is embedded using a python library called pyexiv2. This library is detailed in section 3.3.4.

```
def add_metadata_to_file(file_path, result_dict: dict):
    """Adds json result as metadata to a given file

    Args:
        file_path (str): Path to the image
        result_dict (dict): A dict containing the results
    """
    try:
        img = pyexiv2.Image(file_path)
        data = result_dict
        tags = ["Xmp", "xmp"]
        metadata = {}
        get_xmp_metadata(tags, data, metadata)
        img.modify_xmp(metadata)
        img.close()

        logging.getLogger().info("Added metadata to file %s", file_path)

    except Exception:
        logging.getLogger().warning("Unable to add metadata to file %s", file_path)

def get_xmp_metadata(tags, result_data, metadata: dict):
    """A recursive function which adds all data to their own xmp tags

    Args:
        tags (list[str]): a list of the current xmp tags
        result_data (dict): the value of the current tags
        metadata (dict): the full xmp data which is to be added to the image
    """
    for key in result_data:
        # Appends the current key as a tag
        tags.append(
            key.replace("_", "-").replace("*", "").replace("/", "").replace(":", "")
        )
        value = result_data.get(key)
```

²<https://datatracker.ietf.org/doc/html/rfc4122>, visited 03.05.2022


```

if type(value) == dict:
    get_xmp_metadata(tags, value, metadata)
else:
    metadata.update({".".join(tags): value})
# Deletes the last tag from the list
del tags[-1]

```

Code listing 6.9: Metadata embedding

Code listing 6.9 shows the metadata Python module. It takes the path to an image and a Python dictionary containing the results (see Code listing 6.10). The module first uses a recursive function to format the results dictionary to a dictionary that can be converted to XMP metadata. This conversion is necessary because the `pyexiv2` library requires a dictionary with the keys in a certain format. When the dictionary is converted, the metadata is added to the image.

6.1.5 JSON Results file

```

{
  "GTObject.tif": {
    "file_validation": {
      "before": true,
      "after": true
    },
    "A": {
      "completed": true,
      "results": {
        "passed": true,
        "delta_e": {
          "result": true,
          "limits": {
            "max": "10.00",
            "mean": "4.00"
          },
          "values": {
            "max": "6.59",
            "min": "2.21",
            "mean": "3.78"
          }
        },
        "noise": {
          "result": true,
          "limits": {},
          "values": {
            "l*: 0.81": "1.60"
          }
        }
      },
      ...
    }
  },
  "B": {
    "completed": true,
    "results": {
      "passed": true,
      ...
    }
  },
  ...
}

```

```
    "C": {  
      "completed": true,  
      "results": {  
        "passed": true,  
        ...  
      }  
    },  
    "overall_score": "A"  
  }  
}
```

Code listing 6.10: Example of results file JSON structure. It has been shortened to improve readability.

When an analysis is initiated, all the generated results are parsed and stored in a *results.json* file in the root of the session folder. This file has a defined structure, and this structure is displayed in code listing 6.10. The overall score is generated after the analyses are done and will display the highest quality level that the analysis passed. In addition, in each analysis, one will get information about each type of quality analysis performed. For instance, distance or difference between two colors and noise in the image, and if that type of analysis passed.

6.2 Frontend

The frontend module development started by making a wireframe in Figma³. Figma is an excellent tool for developing wireframes and prototypes. It allows the group to collaborate in real-time on a design, and functional prototypes can be made easily. This tool makes it possible to perform user testing on the application before the design is implemented in code. The group has experience with using this tool from the user-centered design course, as detailed in section 1.4

³<https://www.figma.com/>, visited 26.04.2022

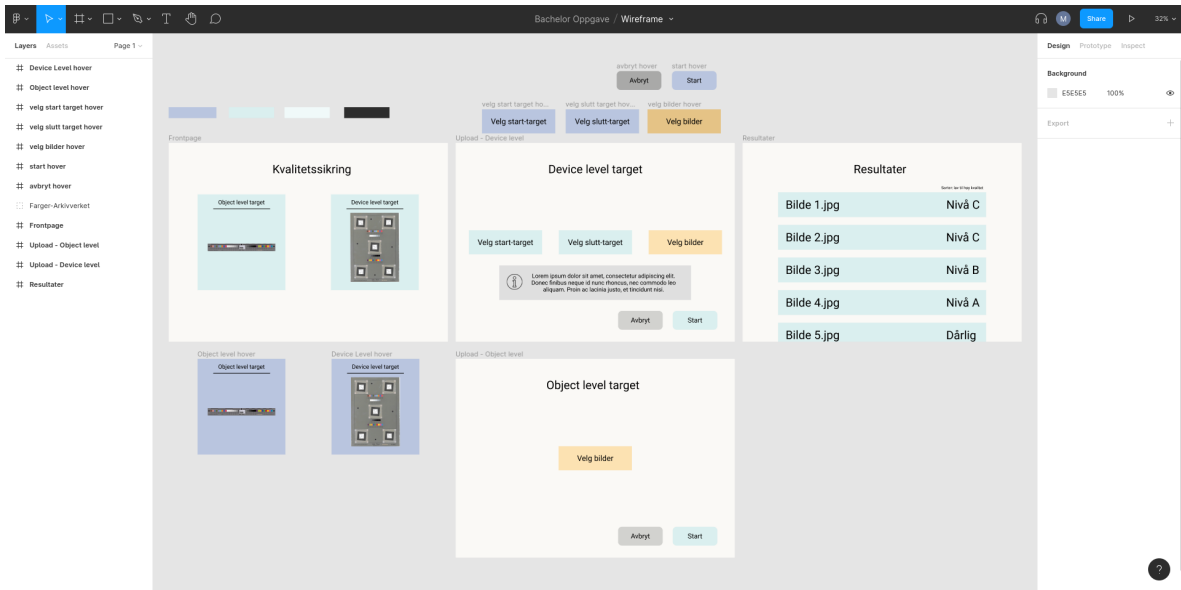


Figure 6.2: Figma wireframe

Figure 6.2 shows the Figma wireframe that was made. Each wireframe page can be seen in detail in Appendix J. It consists of a front page, where the user can choose either device level or object level analysis. There are separate upload pages for each analysis type. After the images have been uploaded, the user is taken to a results page.

The end product was quite similar to the wireframe. However, some minor changes were made since additional functionality needed new buttons that were not in the wireframe, like downloading the images. The most significant change from the wireframe is the addition of a new page for a single result. When the user clicks on an image in the result list, the user is taken to this new page.

6.2.1 Uploading images for analysis

When navigating to either the device level or object level upload pages, the user can upload their images.

```
const [startTarget, setStartTarget] = useState({});
const [endTarget, setEndTarget] = useState({});

// handles start target upload and saves it to state variable
const handleStartTargetUpload = (event) => {
  document.getElementById("start-target-text").style.visibility = "visible"
  setStartTarget(event.target.files[0])
};

// handles end target upload and saves it to state variable
const handleEndTargetUpload = (event) => {
```

```

document.getElementById("end-target-text").style.visibility = "visible"
setEndTarget(event.target.files[0])
};

```

Code listing 6.11: Uploading images to the frontend

Code listing 6.11 shows how the images are saved in state variables when the user selects them using the buttons on the upload page.

```

// handles submit for device level analysis
const handleDeviceSubmit = () => {
// checks if both targets have been uploaded
if (startTarget && Object.keys(startTarget).length === 0 &&
Object.getPrototypeOf(startTarget) === Object.prototype) {
  alert("Start target has not been selected");
  return;
}
if (endTarget && Object.keys(endTarget).length === 0 &&
Object.getPrototypeOf(endTarget) === Object.prototype) {
  alert("End target has not been selected");
  return;
}

// adds the targets as files in a form
const formData = new FormData();
formData.append('before_target', startTarget);
formData.append('after_target', endTarget);
for (const file of props.files) {
  formData.append('files', file);
}
// shows a loading circle
document.getElementById('loader-container').style.visibility = "visible";

// finds software acronym based on the software name
// the acronym is used in the api call
const softwareAcronym = softwareAcronyms
.find(item => item.software === software).acronym;

// makes the request to the api
props.fetchAnalyzePostWrapper(formData,
'/api/analyze/device?iqes=${softwareAcronym}&target=${deviceTarget}')
}

```

Code listing 6.12: Uploading images to the API

The logic in the frontend module uses the Fetch API⁴ to make requests to the API.

Code listing 6.12 shows how the images are uploaded to the API. A wrapper function for the post request was made because this code was used in multiple places.

⁴https://developer.mozilla.org/en-US/docs/Web/API/Fetch_API, visited 21.04.2022

6.2.2 Receiving results

When the images have been uploaded, the user is automatically taken to the result page. This page is where the overall results for each image are shown.

```
let params = useParams();
useEffect(() => {
  let interval;

  const fetchData = async () => {
    console.log("fetching results");
    // Fetches the session id
    fetch(Config.API_URL + '/api/session/status/${params.session_id}')
      .then(resp => resp.json())
      .then(data => {
        setStatus(data.session_status);
        setDownloadReady((data.session_status === "finished"));
      })
      .catch(err => console.log(err));

    // Fetches the results from the api
    fetch(Config.API_URL + '/api/results/${params.session_id}')
      .then(resp => resp.json())
      .then(data => setResults(data))
      .catch(err => console.log(err));
  }

  fetchData();
  if (status !== "finished") {
    interval = setInterval(() => {
      fetchData();
    }, 5 * 1000)
  }

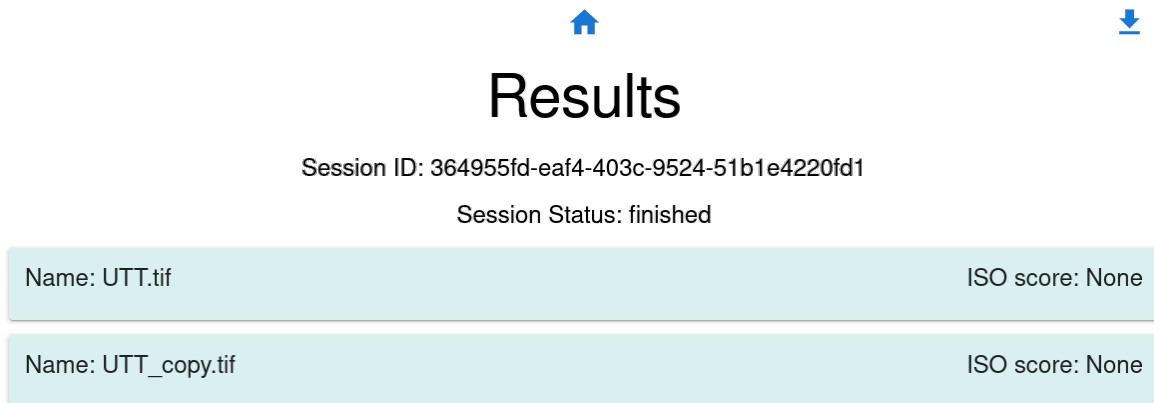
  return () => {
    clearInterval(interval);
  }
}, [status, params.session_id]);
```

Code listing 6.13: Fetching results from API

Code listing 6.13 shows how the results are fetched and how they are saved in the results state variable. This example will auto fetch the results every 5 seconds until the analysis is done. The communication architecture between the frontend and the backend is described in Section 5.2.

6.2.3 Downloading images

The images can be downloaded when the user is on the result page. The downloading is done by clicking the download icon in the top right. The button is only available after the analysis is completed.



Name: UTT.tif	ISO score: None
Name: UTT_copy.tif	ISO score: None

Figure 6.3: Result overview page with download button

All images are downloaded when the user clicks the download button while on the result overview page. A specific image can be downloaded by clicking on the image in the overview and then clicking the download button. Figure 6.3 shows the download button in the top right corner.

```
// Downloads the images.zip file from a session
const handleDownloadAll = () => {
  fetch(Config.API_URL + '/api/download/${params.session_id}')
    .then(resp => {
      if (resp.status !== 200) {
        throw Error("download is not ready");
      }

      return resp.blob();
    })
    .then(blob => {
      const url = window.URL.createObjectURL(blob);
      const a = document.createElement('a');
      a.style.display = 'none';
      a.href = url;
      a.download = 'images.zip';
      document.body.appendChild(a);
      a.click();
      window.URL.revokeObjectURL(url);
    })
    .catch(err => console.log(err))
};
```

Code listing 6.14: Downloading images from API

Code listing 6.14 shows the code for how the images are downloaded. It uses the download endpoint detailed in section 6.1.1

Chapter 7

Quality Assurance

This chapter will explain how the group performed quality assurance during the development. This was done using code reviews, testing, and a CI/CD pipeline. A common code style was also ensured by using the same auto formatters and linters in the group members' development environments.

7.1 Code review

Code reviews were performed regularly throughout the project. Before merging a branch into the main branch, the merge request required approval from at least one other member. Before a member could approve the request, they had to review the code. When reviewing a merge request, a few things were taken into consideration: code readability, bugs, and documentation. Therefore, the code reviews ensured more secure and consistent code.

Martin @martinwh started a thread on an old version of the diff 2 months ago

Resolved by Mikael Falkenberg Krog 2 months ago

^ Hide thread

frontend/src/pages/result-page/ResultPage.jsx

```

20 23 // TODO: make home button on this page
21 24 return (
22 25   <div className="container" >
26 26 +   <HomeButton></HomeButton>
23 27   <Typography variant="h2">
24 28     Resultat: {imageResult.name}
25 29   </Typography>
30 30 +   <div>
31 31 +     <p>Result goes here</p>
32 32 +   </div>
33 33 +   <div className="backButtonContainer">
34 34 +     <IconButton color="primary" aria-label="upload picture" component={Link} to={`/${results
/${imageResult.session_id}`} size="large">

```

Martin @martinwh · 2 months ago

Strange aria label to go back :)

Owner ✔️ 😊 ✎ ⋮

Mikael Falkenberg Krog @Mikaelfk · 2 months ago

Fixed :)

Author Owner 😊 ✎ ⋮

Reply...

Unresolve thread

Figure 7.1: Example of code review

Figure 7.1 shows an example of how code reviews helped the team during the process. In this example, the aria-label for the back button is wrong. This bug was found, reported, and fixed during the code review.

7.2 Testing

Testing is an essential part of any software development project and has been important during this project. Testing ensures that the application behaves the way the developers expect. The group used unit testing to test different functions in the application. Pytest was used to test the backend module, while the frontend module tests were made using Jest. These technologies were detailed in section 3.3.2 and 3.3.6

Testing was used during the entire development of the project. Some parts of the backend module, like the utilities, were developed by using test driven development (TDD). In TDD, the tests are written before the functionality is developed. It was easier to use TDD because the group had already created an overview of what functionality was set to be implemented.

The group wanted to use TDD where possible. However, because of the third party software requiring licenses, some of their integration could not be tested

and was therefore developed without TDD. The endpoints themselves were also not developed using TDD. However, most of the minor functionality like metadata adding was developed using TDD.

Most of the tests were unit tests for the backend module since this was the most important part of the project. Unit tests ensure that individual units of code work as expected; this is done by running the units of code with a set of data and checking that the code returns the expected results. [38]

The group also made a couple of frontend tests. These tests perform specific actions on the frontend elements, and then the tests check if the frontend responds expectedly.

Manual testing was used a great deal when developing the analysis endpoints. It was utilized because the analysis could only be run on a computer with the licenses for the image quality evaluation software. To make it easier to run the tests, a separate computer was used to host the backend, each member of the group could connect to this computer to test their changes and run analyses. The manual tests were performed by making manual HTTP requests to the API.

All automated tests were run automatically in a CI/CD pipeline on each commit made to the Git repository. Every test was required to pass before merging a branch into the main branch. This was to ensure good code quality and stability.

7.3 Development Environment

It is a common issue when several developers are working on a project that they have different personal code style preferences and development environment configurations. Since the code is supposed to be merged with existing code, it is better if all code follows the same style. A unified configuration makes commit logs cleaner and avoids unnecessarily complicated merge conflicts. To keep a consistent code style, the tools *Black*¹, *Pylint*² and *ESLint*³ were used by all the group members.

Black is a general Python code formatter. It has strict and deterministic code style rules and will apply them to entire Python code files at a time. For example, it will take care of line-length constraints, consistent newlines, and removal of superfluous whitespace.

Pylint is a Python linting tool that, in comparison to *Black*, will not format code but rather give warnings about code smells and offer details on how to reduce code complexity.

ESLint is a tool for analyzing JavaScript code to find problems and improvements.

¹<https://github.com/psf/black>, visited 11.05.2022

²<https://pylint.pycqa.org/en/latest/intro.html>, visited 11.05.2022

³<https://eslint.org/>, visited 11.05.2022

It will give warnings about anything it finds and, in many cases, offer solutions. It also supports automatic formatting, which the group members used to keep a consistent JavaScript style.

7.4 CI/CD Pipeline

Continuous Integration / Continuous Deployment pipelines are used to automatically build, test, and deploy software. The pipeline has been an important tool in the development process during this project. The pipeline makes finding problems easier and ensures that none of the previous functionality gets broken when implementing new functionality.

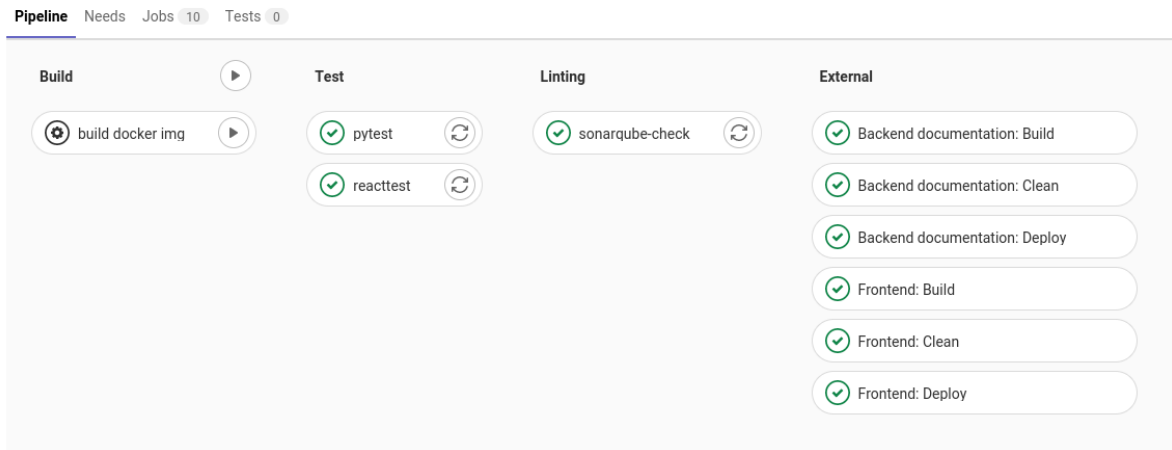


Figure 7.2: Pipeline jobs

Figure 7.2 shows all the jobs that were run in the CI/CD pipeline. The *build docker img* job was used to build the docker image, which is used in the other jobs. This job is manual and only needs to run when the docker image is altered. There are two test jobs: *pytest* and *reacttest*. These run all the tests for Python and React, respectively. The next job is the *sonarqube-check* job, which uses SonarQube to analyze and lint the code. The final external jobs use Jenkins to build and deploy the documentation and the frontend module.

7.4.1 SonarQube

SonarQube is detailed in section 3.3.7. SonarQube is a part of the project's CI/CD pipeline. However, due to the limitations of the free plan, it was only executed when the code was merged into the main branch.

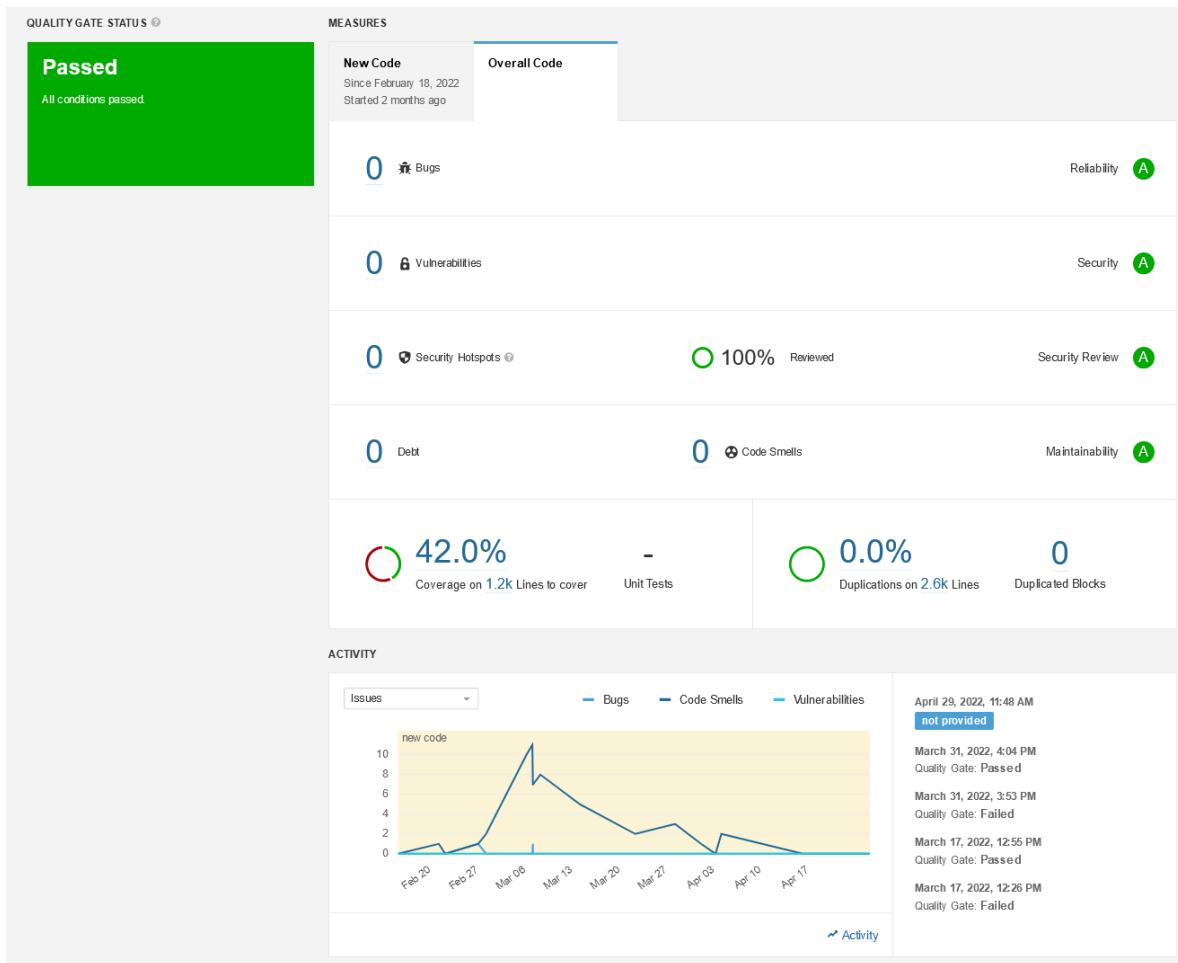


Figure 7.3: SonarQube Project Overview

Figure 7.3 displays an overall report of the current measures in the project. In the top left, there is an indication of whether or not the analysis passed the quality gate, which essentially describes if the project is production-ready. The center column displays a breakdown of different code quality and security factors. Each section provides information on how many issues there are and a rating. In cases where SonarQube registered bugs, code smells, security hotspots, etc., it is reported here. The report gives a detailed description of where the issue occurred and, in some cases, offers explanations and possible fixes. In this example, the project passed the quality gate and got an "A" score on "Reliability", "Security", "Security Review" and "Maintainability".

At the bottom of figure 7.3 there is an activity graph showing the number of bugs, code smells, and vulnerabilities in the project over time.

The screenshot displays the SonarQube Security Hotspot interface. At the top, filters show 'Assigned to me' and 'All' status, with 'Reviewed as safe' and 'Overall code' dropdowns. A progress indicator shows 'Security Hotspots Reviewed' at 100%. The main content area is titled 'Make sure that this logger's configuration is safe.' and includes buttons for 'Add Comment', 'Open in IDE', and 'Get Permalink'. The issue details show a 'Log Injection' category, a 'LOW' review priority, and an 'Assignee' of 'Not assigned'. The status is 'Safe', with a note: 'The code is not at risk and doesn't need to be modified.' Below this, a code snippet from 'backend/kad/main.py' is shown, featuring logging configuration for Flask. The bottom section, 'What's the risk?', 'Are you at risk?', and 'How can you fix it?', provides a detailed explanation of the risk, listing CVEs (CVE-2018-0285, CVE-2000-1127, CVE-2017-15113, CVE-2015-5742) and explaining that logs are useful for security incidents but can be targeted by attackers. It concludes with a note that this rule flags review code that initiates loggers configuration.

Figure 7.4: Example of SonarQube Security Hotspot

In figure 7.4, one can see that SonarQube registered a security hotspot detailing the specific vulnerabilities and offers a detailed explanation of how you may fix this hotspot under the tab 'How can you fix it?'.

The screenshot shows a SonarQube code duplication warning. The main message is 'Update this function so that its implementation is not identical to the one on line 67. Why is this an issue?' with a timestamp of '1 month ago'. Below the message, there are icons for 'Code Smell', 'Major', 'Closed (Fixed)', and 'Not assigned', along with '15min effort' and a 'Comment' button. At the bottom, a list of tags includes 'confusing, duplicate, suspicious'.

Figure 7.5: Example of SonarQube code duplication warning

The group was able to find and fix multiple bugs and code smells using SonarQube, for instance, refactoring code duplications and other bad coding practices. The group was also able to view code coverage and see what parts of the code needed tests. Therefore, SonarQube was a good addition to this project's CI/CD pipeline.

Chapter 8

Discussion

8.1 Group environment

In general, the group environment has been excellent. Every group member has participated and worked hard. No major problems have arisen during the project, the only exception being time pressure, and the group has managed to complete all the goals that were set at the beginning. This can be seen in the sprint reviews and retrospectives in appendices D through H. The group has tried to work together as much as possible, with scheduled work sessions almost every day.

There were a couple of measures the group members took to ensure a good group environment. These measures were: a clear division of labor, frequent communication, predefined group rules, and scheduled work sessions.

The division of labor was done by delegating specific issues through Scrum, and frequent communication was achieved through making a shared communication channel. The predefined rules were set at the start and are described in the project plan in appendix C. Lastly, the scheduled work sessions were decided on at the start of the project, and these work sessions were primarily in-person. The group felt that these sessions helped improve motivation and ensure that every group member was working.

Overall the group has worked well together, and the group is happy with the quality of the product.

8.2 Time management

At the start of the project, the group had four scheduled sessions each week because the group members had a course running simultaneously with this project from February through the end of March. After March, the group had five scheduled work sessions each week. These sessions were usually between the time

10:00 to 15:00.

	Jakob	Kommentar
30. Mars		5 Oppdaterd backend dokumentasjon, parkoding
31. Mars		6.5 Oppdatert parameter filer for OS QM Tool, kranglet med pipeline grrrr, endret navn på backend modul fra <i>kvalitetssikring av digitisering</i> til <i>kad</i> , lagt til logging i hele backend modul
1. April		5 Startet å jobbe på nye, bedre, kulere analyze endpoints
4. April		4.5 Også blitt syk >:(Gjorde ferdig nye, bedre, kulere analyze endpoints. Møte med veiledere
5. April		5.25 Skrevet videre på utviklingsprosess i rapporten, sett over kapittel 1 slik at den er klar for tilbakemelding fra veiledere. Møte med oppdragsgiver
2. Mai		5.5 Gjort ferdig dokumentasjon for nytt endpoint. Fikset error i results page frontend. Parkodet
7. April		4 Download fix for backend og frontend. Skrevet videre på development process i rapporten.
8. April		2 Skrevet mer på rapporten. Sprint 5 retrospective og review, møteinkallinger
11. April		1.5 Møte med veiledere og arbeidsgiver
19. April		5 Fysisk møte med Ottar, tatt opp demovideo :d

Figure 8.1: Excerpt from the time logs, which were written in Norwegian

Every group member was responsible for logging their work hours and what they had worked on that day. The logging was done in a spreadsheet, and an excerpt from the logs is shown in Figure 8.1.

Throughout the project, the group felt some time pressure, especially at the start when the group had not acquired any licenses for any image quality evaluation software. This time pressure was due to a couple of reasons: it took some time for the group to evaluate and decide on which programs would suit the groups needs, and the licenses had to be sent through the postal service. In addition, there was pressure towards the end of the project due to the deadline approaching and the fact that the group had never written a thesis with such a large scope.

The group was unable to follow the project plan and the Gantt diagram to the letter. For instance, the group had initially planned to have an MVP ready by the end of sprint 3. However, this milestone was not reached on time due to the licenses taking longer to acquire than anticipated. Apart from this, the group managed the time well and finished the application by the end of sprint 5.

8.3 Scrum process

The Scrum framework combined with the git workflow described in section 4.2 worked well for the group. The group benefited greatly from the review and retrospectives at the end of each sprint. The reviews forced the group to reflect on the work done, which allowed the group to improve and change things that did not work well. Furthermore, it allowed the group to continue doing the things that did work well. Scrum also made the group focus on smaller fixed goals in short periods, which made it easy to see progress during the development and helped motivate the group. Smaller goals are also easier to complete than a larger overall goal.

An agile methodology like Scrum ended up being a perfect fit for this project. The project requirements changed during the project, and the limited access to image quality evaluation software made it hard to progress initially. However, the agile methodology allowed us to find ways to be productive without wasting time having to re-work and backtrack on what had already been done.

A part of the group's Scrum process was the weekly meetings with the client. This process ensured the group had consistent communication with the client. The group could show their work to the client, get feedback, and make sure the product covered the client's needs. This feedback was a great resource for the group and helped a lot during the development process.

8.4 Choosing image quality evaluation software

A large part of the project was choosing which software to use to perform the image quality evaluation. This choice was made in collaboration with the product owner.

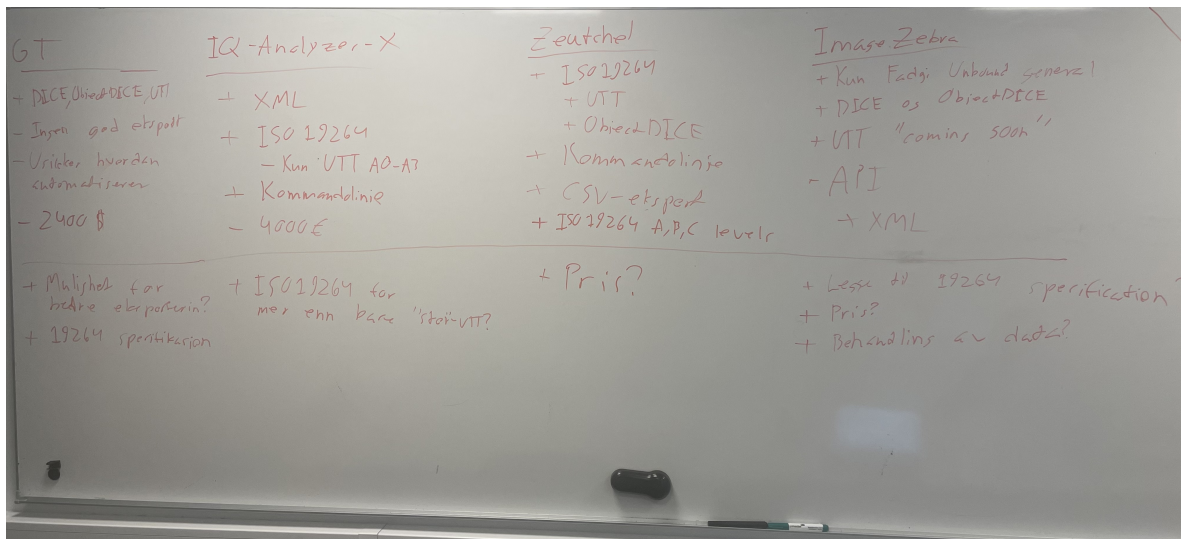


Figure 8.2: A comparison of the different image quality evaluation software assessed in the project

Figure 8.2 shows the pros and cons list the group made about each software that was in consideration. The four software that were in consideration were Golden-Thread, iQ-Analyser-X (IQX), OS QM-Tool (OQT) and ImageZebra. After some discussion with the product owner, the group evaluated that the best option would be IQX. The group then decided to ask ImageEngineering, the creators of IQX, for a trial license of the software. OQT also seemed like a good option, so the product owner decided to purchase to license to OQT, which the group could use during development.

8.5 Report

The group got started on the report fairly early in the project time span and did some work on the report during the development process. The reason for starting the report early in the process was that the group did not have much work to do yet because the group had not received the image quality evaluation software licenses. Later in the process, the group scheduled a day each week to work on the report. This decision was made to ensure that some work was done on the report while developing the main application.

The group decided early on when the software development phase of the project should end. The date that was set was the 20th of April. A few changes were made to the code after this, but no significant functionality was added after this date. This limitation made it easier for the group to focus on the report.

The group took sufficient notes during the development process and could easily look back and find out why a particular decision was made. In addition, the notes

made the report writing phase easier and more manageable.

The only problem that occurred was that the group faced a slight lack of motivation during the report writing compared to the development phase. However, the group overcame this by having consistent work hours and good teamwork.

8.6 Product discussion

The client originally wanted a module that automated their current image analysis workflow and could be integrated into their current solution. However, the integration requirement was changed. The change was a wish from the client due to a lack of communication with the Digital Archives. Therefore, an independent application for automated quality analysis was developed instead. However, the application was split into two modules, a backend, and a frontend. This modularization means that any archive institution could easily use the application standalone or only integrate the backend into their current workflow.

Splitting the application into two modules ended up being an excellent idea. If the application had been one module with its own GUI and integrated logic, it would have been more challenging to integrate the application into an already existing workflow.

The group managed to finish all of the requirements that were set at the start. The final product works fine but could use some optimization and refinement. Primarily because each image is analyzed three times, this is not necessary if the code could compare the analysis results instead itself instead of letting the image quality evaluation software do it.

The group enjoyed working on the product itself with Python, but the students felt they would have enjoyed working with a language like Go slightly more. However, this is simply due to the group members' personal preferences.

8.6.1 Licenses

All the quality evaluation software that the group acknowledged as applicable required a license. The group evaluated the software with the product owner, as detailed in section 8.4, but it took significantly more time to acquire the licenses than anticipated. The plan was to make the request to IKAMR at the end of sprint one and to receive and begin to develop the backend module in sprint 2. Instead, the group got access to IQX and OQT in the middle of sprint 3, so there were essentially two and a half sprints left to finish the project.

The group remained productive and delegated the time for other tasks. For example, the frontend prototype was designed, and the development for the frontend was initiated while waiting on the licenses. Time was also spent evaluating

different file validation software and designing the first version of the backend API endpoints.

8.6.2 Quirks

While working with the third party image quality evaluation software, it became clear that the command line support was lacking. For example, starting an analysis through the command line will still bring up the GUI even though it is technically unnecessary.

Another issue that the group noted was that the image quality evaluation software had no support for performing a single analysis to grade an image. Therefore, an analysis had to be executed for all three ISO 19264-1:2021 levels, even though the analysis returns the same result each time. The only difference is that the thresholds that the results are compared to are different. In theory, this could be worked around by performing the result and threshold comparison in the groups' application, but it was deemed outside this project's scope.

Chapter 9

Conclusion

The task given was to develop an automated quality assurance application for digitization in the Digital Archives for IKAMR. The project's overall goal was to grade images according to the ISO 19264-1:2021 standard and embed the results as metadata automatically. The group made a backend API that performs the analysis and embeds the metadata into the images. A frontend module was created to serve as an example of how the API can be implemented. This chapter explains the project's goals, whether or not they were met, and details how the application could be improved in the future.

9.1 Results

The resulting modules work as intended. The backend module fulfills the goals of analyzing and validating images according to the ISO 19264-1:2021 standard by using two image quality evaluation software. After the images are analyzed, the result is embedded as metadata in the image files, and they can be downloaded. The backend API also follows the RESTful API constraints. The frontend module fulfills the goal of showcasing how the API can be used. With this, the application fulfills all the result goals described in section 1.5.1

It is hard to evaluate whether or not the impact goals described in section 1.5.1 have been fulfilled because the application has not been tested in production yet. Therefore, the group can not reasonably conclude that the impact goals are fulfilled, but it is safe to assume that the application has the potential to fulfill them. Although, it might require some more development to do so. However, the product owner seems optimistic that the application will improve the digitization process at IKAMR.

The group learned how to develop a single-page user-friendly interface using React throughout the project. The group has successfully used Scrum, see section 8.3. During the project, the group members became more confident in using Py-

thon and JavaScript, and the members learned how to integrate third party applications into a structured workflow. Setting up a CI/CD pipeline has helped the group learn about DevOps principles and quality assurance of code, see section 7.4.

The project also provided the group members with experience in working with a client that does not have deep technical knowledge, and how to communicate concepts and ideas to the client in a good way. The experience of working with an outside person and building something based on set requirements was something none of the group members had done before. The project in its entirety was an excellent example of how a project like this gets done in a professional work environment and was a great learning experience for the group members.

The group's application has succeeded in fulfilling the result goals and learning objectives, and it has the potential to fulfill the impact goals when put into production. Therefore, the project has been successful, and a working application that fulfills the requirements has been developed. The overall process has gone very smoothly, with only some minor problems, which were discussed in section 8.5 and 8.6. The group had a good experience using the scrum process, and it worked well for a team of this size.

9.2 Future work

The most important feature the group wished to continue working on was optimizing how the image quality evaluation software are used. In theory, the software do not need to perform three analyses on each image when the results are the same each time. While this is an inherent problem in the image quality evaluation software themselves, it would be possible to optimize it with external code.

A functionality discussed during the development process was using a database to store the analysis results and program state instead of JSON files. Using a database would have made updating and managing the data easier. However, this was not done due to time constraints, and it would not have changed how the application works. Therefore, the group decided to focus on other functionality instead.

Authorization is also something that could be implemented. However, the product is meant to be used internally within the business and not open to the general public, so user authentication was not required. It will only be accessible on their internal network and not to the general public.

Another functionality that could be implemented is a method for stopping an ongoing analysis. Currently, the only way to stop an analysis is to restart the backend API manually. This is not practical and would interrupt the workflow.

The backend module was designed to be flexible on which image quality evaluation software to use when performing analyses. While it is possible to add support for more software without much hassle in its current state, there is room

for improvement. This improvement would include generalizing the methods for running analyses and removing coupling between the endpoints and the code for starting an analysis.

Test coverage of both the backend and frontend modules should be improved before adding any significant features. While the group relied on manual testing during the development process, more test coverage would have reduced the time spent manually testing changes. In addition, stress testing was not performed during the project since this will be handled by IKAMR in the future.

Bibliography

- [1] M. S. Gupta. 'What is digitization, digitalization, and digital transformation?' (2020), [Online]. Available: <https://www.arcweb.com/blog/what-digitization-digitalization-digital-transformation> (visited on 19/05/2022).
- [2] IKAMR. 'Om sedak / digitisering.' (2020), [Online]. Available: <https://www.ikamr.no/om-sedak-digitisering> (visited on 19/05/2022).
- [3] Arkivverket. 'Historia til digitalarkivet.' (n.d.), [Online]. Available: <https://www.digitalarkivet.no/content/55/historia-til-digitalarkivet> (visited on 19/05/2022).
- [4] Arkivverket. 'Digitalarkivet som nasjonal publiseringsplattform for skanna arkivmateriale.' (n.d.), [Online]. Available: <https://www.digitalarkivet.no/content/442/digitalarkivet-som-nasjonal-publiseringsplattform-for-skanna-arkivmateriale> (visited on 19/05/2022).
- [5] P Technical Committee ISO/TC 42, 'Photography — archiving systems — imaging systems quality analysis — part 1: Reflective originals,' en, Standard ISO 19264-1:2021, 2021. [Online]. Available: <https://www.iso.org/standard/79172.html>.
- [6] International Organization for Standardization. 'Standards.' (), [Online]. Available: <https://www.iso.org/standards.html> (visited on 19/05/2022).
- [7] P Gorbachenko. 'What are functional and non-functional requirements and how to document these.' (n.d.), [Online]. Available: <https://enkonix.com/blog/functional-requirements-vs-non-functional/> (visited on 19/05/2022).
- [8] L. Chen, M. A. Babar and B. Nuseibeh, 'Characterizing architecturally significant requirements,' *IEEE Software*, vol. 30, no. 2, pp. 38–45, Mar. 2013. DOI: 10.1109/ms.2012.174. [Online]. Available: <https://doi.org/10.1109/ms.2012.174>.
- [9] Scaled Agile Framework. 'Nonfunctional requirements.' (2021), [Online]. Available: <https://www.scaledagileframework.com/nonfunctional-requirements/> (visited on 19/05/2022).

- [10] ReQtest. 'Why is the difference between functional and non-functional requirements important?' (2012), [Online]. Available: <https://reqtest.com/requirements-blog/functional-vs-non-functional-requirements/> (visited on 19/05/2022).
- [11] Lovdata. 'Forskrift om utfyllende tekniske og arkivfaglige bestemmelser om behandling av offentlige arkiver (riksarkivarens forskrift).' (2022), [Online]. Available: https://lovdata.no/dokument/SF/forskrift/2017-12-19-2286/KAPITTEL_5-4#%C2%A75-15 (visited on 18/05/2022).
- [12] Riksrevisjonen. 'Undersøkelse av digitalisering av kulturarven.' (2017), [Online]. Available: <https://www.riksrevisjonen.no/rapporter-mappe/no-2016-2017/digitalisering-av-kulturarven/> (visited on 16/05/2022).
- [13] Arkivverket. 'Undersøkelse av digitalisering av kulturarven.' (2019), [Online]. Available: <https://www.arkivverket.no/nyheter/alle-landets-arkiv-et-tasteklikk-unna?q=grande> (visited on 16/05/2022).
- [14] IKAMR. '722 000 til arkivprosjekt.' (2020), [Online]. Available: <https://www.ikamr.no/nyhende/722-000-til-arkivprosjekt> (visited on 16/05/2022).
- [15] M. Laine-Zamojska, 'Digitization of cultural heritage,' *Nordisk Museologi*, vol. 1, pp. 125–134, 2011. [Online]. Available: <https://journals.uio.no/museolog/article/download/3151/2727/9853>.
- [16] A. TI, 'Quality controlled scanning,' en, Oct. 2021. [Online]. Available: https://support.imageaccess.de/downloads/product_manuals/FAQ/FAQ-Quality-Controlled-Scanning.pdf.
- [17] Federal Agencies Digital Guidelines Initiative. 'About.' (2017), [Online]. Available: <https://www.digitizationguidelines.gov/about/> (visited on 26/04/2022).
- [18] Still Image Working Group, 'Technical guidelines for digitizing cultural heritage materials,' en, Federal Agencies Digital Guidelines Initiative, Technical guidelines, 2016. [Online]. Available: <https://www.digitizationguidelines.gov/guidelines/digitize-technical.html>.
- [19] B. C. K. Ly, E. B. Dyer, J. L. Feig, A. L. Chien and S. D. Bino, 'Research techniques made simple: Cutaneous colorimetry: A reliable technique for objective skin color measurement,' *Journal of Investigative Dermatology*, vol. 140, no. 1, 3–12.e1, Jan. 2020. DOI: 10.1016/j.jid.2019.11.003. [Online]. Available: <https://doi.org/10.1016/j.jid.2019.11.003>.
- [20] DT Cultural Heritage. 'Numerically evaluating image quality.' (n.d.), [Online]. Available: <https://heritage-digitaltransitions.com/numerically-evaluating-image-quality/> (visited on 19/05/2022).
- [21] Python Software Foundation. 'About python.' (n.d.), [Online]. Available: <https://www.python.org/about/> (visited on 19/05/2021).

- [22] Open Preservation Foundation. 'Getting started with jhove.' (2015), [Online]. Available: <https://jhove.openpreservation.org/getting-started/> (visited on 19/05/2021).
- [23] Scrum.org. 'What is scrum?' (n.d.), [Online]. Available: <https://www.scrum.org/resources/what-is-scrum> (visited on 19/05/2022).
- [24] D. Radigan. 'Kanban.' (n.d.), [Online]. Available: <https://www.atlassian.com/agile/kanban> (visited on 19/05/2022).
- [25] M. Rehkopf. 'Kanban vs. scrum: Which agile are you?' (n.d.), [Online]. Available: <https://www.atlassian.com/agile/kanban/kanban-vs-scrum> (visited on 19/05/2022).
- [26] Scrum.org. 'What is a sprint in scrum?' (n.d.), [Online]. Available: <https://www.scrum.org/resources/what-is-a-sprint-in-scrum> (visited on 19/05/2022).
- [27] 'What is sprint planning?' (n.d.), [Online]. Available: <https://www.scrum.org/resources/what-is-sprint-planning> (visited on 19/05/2022).
- [28] 'What is a sprint retrospective?' (n.d.), [Online]. Available: <https://www.scrum.org/resources/what-is-a-sprint-retrospective> (visited on 19/05/2022).
- [29] 'What is sprint review?' (n.d.), [Online]. Available: <https://www.visual-paradigm.com/scrum/what-is-sprint-review> (visited on 19/05/2022).
- [30] Atlassian. 'Devops: Breaking the development-operations barrier.' (n.d.), [Online]. Available: <https://www.atlassian.com/devops> (visited on 19/05/2022).
- [31] K. Zettler. 'Trunk-based development.' (n.d.), [Online]. Available: <https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development> (visited on 19/05/2022).
- [32] Amazon Web Services. 'What is an api?' (n.d.), [Online]. Available: <https://aws.amazon.com/what-is/api/> (visited on 19/05/2022).
- [33] Red Hat. 'What is a rest api?' (2020), [Online]. Available: <https://www.redhat.com/en/topics/api/what-is-a-rest-api> (visited on 19/05/2022).
- [34] Meta. 'Introducing hooks.' (n.d.), [Online]. Available: <https://reactjs.org/docs/hooks-intro.html> (visited on 19/05/2022).
- [35] J. Morales. 'Mobile first design strategy: The when, why and how.' (2021), [Online]. Available: <https://xd.adobe.com/ideas/process/ui-design/what-is-mobile-first-design/> (visited on 19/05/2022).
- [36] Smartbear. 'Api endpoints - what are they? why do they matter?' (n.d.), [Online]. Available: <https://smartbear.com/learn/performance-monitoring/api-endpoints/> (visited on 19/05/2022).

- [37] Microsoft. 'Hard links and junctions.' (2021), [Online]. Available: <https://docs.microsoft.com/en-us/windows/win32/fileio/hard-links-and-junctions> (visited on 19/05/2022).
- [38] Smartbear. 'What is unit testing.' (n.d.), [Online]. Available: <https://smartbear.com/learn/automated-testing/what-is-unit-testing/> (visited on 19/05/2022).

Appendix A

Cooperation agreement

Fastsatt av prorektor for utdanning 10.12.2020

STANDARDAVTALE

om utføring av studentoppgave i samarbeid med ekstern virksomhet

Avtalen er ufravikelig for studentoppgaver (heretter oppgave) ved NTNU som utføres i samarbeid med ekstern virksomhet.

Forklaring av begrep

Opphavsrett

Er den rett som den som skaper et åndsverk har til å fremstille eksemplarer av åndsverket og gjøre det tilgjengelig for allmennheten. Et åndsverk kan være et litterært, vitenskapelig eller kunstnerisk verk. En studentoppgave vil være et åndsverk.

Eiendomsrett til resultater

Betyr at den som eier resultatene bestemmer over disse. Utgangspunktet er at studenten eier resultatene fra sitt studentarbeid. Studenten kan også overføre eiendomsretten til den eksterne virksomheten.

Bruksrett til resultater

Den som eier resultatene kan gi andre en rett til å bruke resultatene, f.eks. at studenten gir NTNU og den eksterne virksomheten rett til å bruke resultatene fra studentoppgaven i deres virksomhet.

Prosjektbakgrunn

Det partene i avtalen har med seg inn i prosjektet, dvs. som vedkommende eier eller har rettigheter til fra før og som brukes i det videre arbeidet med studentoppgaven. Dette kan også være materiale som tredjepersoner (som ikke er part i avtalen) har rettigheter til.

Utsatt offentliggjøring

Betyr at oppgaven ikke blir tilgjengelig for allmennheten før etter en viss tid, f.eks. før etter tre år. Da vil det kun være veileder ved NTNU, sensorene og den eksterne virksomheten som har tilgang til studentarbeidet de tre første årene etter at studentarbeidet er innlevert.

1. Avtaleparter

Norges teknisk-naturvitenskapelige universitet (NTNU) Institutt: Fakultet for informasjonsteknologi og elektroteknikk, Institutt for datateknologi og informatikk
Veileder ved NTNU: Johanna Johansen E-post og tlf.: johanna.johansen@ntnu.no
Veileder ved NTNU: Rune Hjelsvold E-post og tlf.: rune.hjelsvold@ntnu.no, +47 61 13 51 84
Ekstern virksomhet: Interkommunalt arkiv for Møre og Romsdal IKS v/Fylkessenter for digitisering av kulturarven (SEDAK) i samarbeid med Arkivverket Ekstern virksomhet sin kontaktperson, e-post og tlf.: Ottar A. B. Anderson ottar.anderson@ikamr.no, +47 90 91 82 91
Student: Martin Wighus Holtmon Fødselsdato: 17.02.2000
Student: Jakob Frantzvåg Karlsmoen Fødselsdato: 07.08.2000
Student: Mikael Falkenberg Krog Fødselsdato: 18.02.2000

Partene har ansvar for å klarere eventuelle immaterielle rettigheter som studenten, NTNU, den eksterne eller tredjeperson (som ikke er part i avtalen) har til prosjektbakgrunn før bruk i forbindelse med utførelse av oppgaven. Eierskap til prosjektbakgrunn skal fremgå av eget vedlegg til avtalen der dette kan ha betydning for utførelse av oppgaven.

2. Utførelse av oppgave

Studenten skal utføre: (sett kryss)

Masteroppgave	
Bacheloroppgave	x
Prosjektoppgave	
Annen oppgave	

Startdato: 19.01.2022
Sluttdato: 20.05.2022

Oppgavens arbeidstittel er: Kvalitetssikring av digitisering i Digitalarkivet
--

Ansvarlig veileder ved NTNU har det overordnede faglige ansvaret for utforming og godkjenning av prosjektbeskrivelse og studentens læring.

3. Ekstern virksomhet sine plikter

Ekstern virksomhet skal stille med en kontaktperson som har nødvendig faglig kompetanse til å gi studenten tilstrekkelig veiledning i samarbeid med veileder ved NTNU. Ekstern kontaktperson fremgår i punkt 1.

Formålet med oppgaven er studentarbeid. Oppgaven utføres som ledd i studiet. Studenten skal ikke motta lønn eller lignende godtgjørelse fra den eksterne for studentarbeidet. Utgifter knyttet til gjennomføring av oppgaven skal dekkes av den eksterne. Aktuelle utgifter kan for eksempel være reiser, materialer for bygging av prototyp, innkjøp av prøver, tester på lab, kjemikalier. Studenten skal klarere dekning av utgifter med ekstern virksomhet på forhånd.

Ekstern virksomhet skal dekke følgende utgifter til utførelse av oppgaven:
Nødvendige reiser, programlisenser og utstyr

Dekning av utgifter til annet enn det som er oppført her avgjøres av den eksterne underveis i arbeidet.

4. Studentens rettigheter

Studenten har opphavsrett til oppgaven¹. Alle resultater av oppgaven, skapt av studenten alene gjennom arbeidet med oppgaven, eies av studenten med de begrensninger som følger av punkt 5, 6 og 7 nedenfor. Eiendomsretten til resultatene overføres til ekstern virksomhet hvis punkt 5 b er avkrysset eller for tilfelle som i punkt 6 (overføring ved patenterbare oppfinnelser).

I henhold til lov om opphavsrett til åndsverk beholder alltid studenten de ideelle rettigheter til eget åndsverk, dvs. retten til navngivelse og vern mot krenkende bruk.

Studenten har rett til å inngå egen avtale med NTNU om publisering av sin oppgave i NTNUs institusjonelle arkiv på Internett (NTNU Open). Studenten har også rett til å publisere oppgaven eller deler av den i andre sammenhenger dersom det ikke i denne avtalen er avtalt begrensninger i adgangen til å publisere, jf. punkt 8.

¹ Jf. Lov om opphavsrett til åndsverk mv. av 15.06.2018 § 1

5. Den eksterne virksomheten sine rettigheter

Der oppgaven bygger på, eller videreutvikler materiale og/eller metoder (prosjektbakgrunn) som eies av den eksterne, eies prosjektbakgrunnen fortsatt av den eksterne. Hvis studenten skal utnytte resultater som inkluderer den eksterne sin prosjektbakgrunn, forutsetter dette at det er inngått egen avtale om dette mellom studenten og den eksterne virksomheten.

Alternativ a) (sett kryss) Hovedregel

<input checked="" type="checkbox"/>	Ekstern virksomhet skal ha bruksrett til resultatene av oppgaven
-------------------------------------	--

Dette innebærer at ekstern virksomhet skal ha rett til å benytte resultatene av oppgaven i egen virksomhet og/eller i samarbeid med Arkivverket. Retten er ikke-eksklusiv.

Alternativ b) (sett kryss) Unntak

<input type="checkbox"/>	Ekstern virksomhet skal ha eiendomsretten til resultatene av oppgaven og studentens bidrag i ekstern virksomhet sitt prosjekt
--------------------------	---

Begrunnelse for at ekstern virksomhet har behov for å få overført eiendomsrett til resultatene:

6. Godtgjøring ved patenterbare oppfinnelser

Dersom studenten i forbindelse med utførelsen av oppgaven har nådd frem til en patenterbar oppfinnelse, enten alene eller sammen med andre, kan den eksterne kreve retten til oppfinnelsen overført til seg. Dette forutsetter at utnyttelsen av oppfinnelsen faller inn under den eksterne sitt virksomhetsområde. I så fall har studenten krav på rimelig godtgjøring. Godtgjøringen skal fastsettes i samsvar med arbeidstakeroppfinnelsesloven § 7. Fristbestemmelsene i § 7 gis tilsvarende anvendelse.

7. NTNU sine rettigheter

De innleverte filer av oppgaven med vedlegg, som er nødvendig for sensur og arkivering ved NTNU, tilhører NTNU. NTNU får en vederlagsfri bruksrett til resultatene av oppgaven, inkludert vedlegg til denne, og kan benytte dette til undervisnings- og forskningsformål med de eventuelle begrensninger som fremgår i punkt 8.

8. Utsatt offentliggjøring

Hovedregelen er at studentoppgaver skal være offentlige.

Sett kryss

x	Oppgaven skal være offentlig
---	------------------------------

I særlige tilfeller kan partene bli enige om at hele eller deler av oppgaven skal være undergitt utsatt offentliggjøring i maksimalt tre år. Hvis oppgaven unntas fra offentliggjøring, vil den kun være tilgjengelig for student, ekstern virksomhet og veileder i denne perioden. Sensurkomiteen vil ha tilgang til oppgaven i forbindelse med sensur. Student, veileder og sensorer har taushetsplikt om innhold som er unntatt offentliggjøring.

Oppgaven skal være underlagt utsatt offentliggjøring i (sett kryss hvis dette er aktuelt):

Sett kryss

Sett dato

Sett kryss	Sett dato
<input type="checkbox"/>	ett år
<input type="checkbox"/>	to år
<input type="checkbox"/>	tre år

Behovet for utsatt offentliggjøring er begrunnet ut fra følgende:

Dersom partene, etter at oppgaven er ferdig, blir enig om at det ikke er behov for utsatt offentliggjøring, kan dette endres. I så fall skal dette avtales skriftlig.

Vedlegg til oppgaven kan unntas ut over tre år etter forespørsel fra ekstern virksomhet. NTNU (ved instituttet) og student skal godta dette hvis den eksterne har saklig grunn for å be om at et eller flere vedlegg unntas. Ekstern virksomhet må sende forespørsel før oppgaven leveres.

De delene av oppgaven som ikke er undergitt utsatt offentliggjøring, kan publiseres i NTNUs institusjonelle arkiv, jf. punkt 4, siste avsnitt. Selv om oppgaven er undergitt utsatt offentliggjøring, skal ekstern virksomhet legge til rette for at studenten kan benytte hele eller deler av oppgaven i forbindelse med jobbsøknader samt videreføring i et master- eller doktorgradsarbeid.

9. Generelt

Denne avtalen skal ha gyldighet foran andre avtaler som er eller blir opprettet mellom to av partene som er nevnt ovenfor. Dersom student og ekstern virksomhet skal inngå avtale om konfidensialitet om det som studenten får kjennskap til i eller gjennom den eksterne virksomheten, kan NTNUs standardmal for konfidensialitetsavtale benyttes.

Den eksterne sin egen konfidensialitetsavtale, eventuell konfidensialitetsavtale den eksterne har inngått i samarbeidprosjekter, kan også brukes forutsatt at den ikke inneholder punkter i motstrid med denne avtalen (om rettigheter, offentliggjøring mm). Dersom det likevel viser seg at det er motstrid, skal NTNUs standardavtale om utføring av studentoppgave gå foran. Eventuell avtale om konfidensialitet skal vedlegges denne avtalen.

Eventuell uenighet som følge av denne avtalen skal søkes løst ved forhandlinger. Hvis dette ikke fører frem, er partene enige om at tvisten avgjøres ved voldgift i henhold til norsk lov. Tvisten avgjøres av sorenskriveren ved Sør-Trøndelag tingrett eller den han/hun oppnevner.

Denne avtale er signert i fire eksemplarer hvor partene skal ha hvert sitt eksemplar. Avtalen er gyldig når den er underskrevet av NTNU v/instituttleder.

Signaturer:

Instituttleder:
Dato:
Veileder ved NTNU:
Dato:
Veilder ved NTNU: <i>Johanna Johansen</i> <i>Rune Hjeltnes</i>
Dato: <i>28. januar 2022</i> <i>31. januar 2022</i>
Ekstern virksomhet:
<i>Ottar Andre Andersen</i>
Dato: 27.01.2022
Student: <i>Martin W. Holman</i>
Dato: <i>28.01.2022</i>
Student: <i>Jacob Frantzen og Karlsmoen</i>
Dato: <i>28.01.2022</i>
Student: <i>Mikael F. Krog</i>
Dato: <i>28.01.2022</i>

Appendix B

Project Description

Arbeidstittel: Kvalitetssikring av digitisering i Digitalarkivet

Oppdragsgjevar: Interkommunalt arkiv for Møre og Romsdal IKS
v/Fylkessenter for digitisering av kulturarven (SEDAK)
i samarbeid med Arkivverket

Kontaktperson: Ottar A. B. Anderson

Adresse: Kirkegata 10B, 6004 ÅLESUND

Telefon: 90 91 82 91

E-post: ottar.anderson@ikamr.no

Bakgrunn

Digitalarkivet www.digitalarkivet.no er Noreg si fellesløyising for mottak, bevaring og publisering av digitiserte/mediekonverterte historiske arkiv frå statlege, kommunale og private aktørar.

Arkivinstitusjonar frå heile Noreg publiserer sine arkiv, både offentlege og private. Offentlege som stat, fylke/region og kommune, og privat som bedrifter, lag, foreiningar og privatpersonar.

Ein kan søke i strukturerte data på tvers av arkiv, lese fulltekstavskrifter, sjå fotografi, videoar og/eller høyre digitisert lyd.

Digitalarkivet er og blir utvikla av Arkivverket, første versjon var tilgjengeleg så tidleg som i 1998. Nokre av prinsippa til Digitalarkivet er at brukarane sine behov står i sentrum og at tenester er i kontinuerleg endring.

Oppgåva sitt mål

Oppgåva sitt mål er å greie ut og bygge ein webbasert, automatisert kvalitetssikringsmodul og kople denne til Digitalarkivet si back-end løysing/portal «Arkivdigitalisering».

Digitalarkivet er bygd med nødvendig grunnleggande funksjonalitet for å ta imot og bevare historiske data for æva og gjere dei søkbare og tilgjengelege for sluttbrukaren. Arkivverket ønsker at Digitalarkivet går inn i eit økosystem, og stimulerer derfor til å etablere nye løysingar og tenester.

Arkiva blir digitisert/mediekonvertert av kvar enkelt arkivinstitusjon, både med ulikt utstyr og resultat. Det er derfor svært viktig å sikre kvaliteten på dei digitiserte/mediekonverterte filene som vert lasta opp og publisert i Digitalarkivet. Arkivmaterialet er skapt på papir, som bøker, dokument, kart, teikningar, notar, illustrasjonar og filmmateriale og fotografi i mange ulike størrelsar og format.

Modulen skal kunne gjennomføre sortering på kvalitetsnivå i samsvar med ISO 19264-1:2021 og filer validert ved bruk av JHOVE, samt å sikre at informasjon om dette følger i metadataen til filene via XMP (Extensible Metadata Platform).

Det finst ferdigutvikla programvare som kan brukast for sjølve utføringa av kvalitetsanalysen, både kommersielle og open kjeldekode. Oppdragsgjever vil gjere desse tilgjengelege. Det er likevel ein del av oppgåva å greie ut kva programvare, eventuelt kva kombinasjon av desse som vil fungere best, eventuelt om det er behov for vidareutvikling. Tilgang til eigne datasett med digitisert/mediekonvertert arkivmateriale og nødvendige brukartilgang stilles til rådighet i prosjektperioden.

Det er via Digitalarkivet si back-end løysing / portal at alle filene blir lasta opp, administrert og kvar enkelt arkivinstitusjon styrer si eiga digitisering/mediekonvertering. Det er også ein del av oppgåva å sjå på moglegheita for å ta i bruk IIF (International Image Interoperability Framework) for betre visuell kontroll av filene både i back-end og front-end.

Modulen vil gi arkivsektoren og sluttbrukaren moglegheit til å samanlikne arkiv (datasett) for analyse og forskning på likt grunnlag (same kvalitetsnivå), uavhengig av kva for ein arkivinstitusjon som har digitisert/mediekonvertert materialet. Ved suksessfull utføring av oppgåva vil modulen kunne bli i permanent bruk.

Oppgåva sitt krav

Fordel å ha kjennskap til webutvikling med både forståing for front-end og back-end, samt kjennskap til bildebehandling og XMP/metadadataintegrasjon.

Kodespråk brukt i Digitalarkivet

PHP og Python, moglegheit for kopling av modulen via eit API vil bli vurdert.

Ressursar: XMP: <https://docs.goobi.io/docs/>

<https://github.com/Kitodo> <https://www.kitodo.org/en>

<https://www.intranda.com/en/digiverso/goobi/goobi-overview/>

<https://www.adobe.com/products/xmp.html>

https://docs.goobi.io/goobi-workflow-plugins-en/step/intranda_step_xmp?fbclid=IwAR1wFspZk2-fQinh3g5Un2exFmVdcpRDgDas-JW4NrwpK7tHGJdjXL2ysmA

Kvalitetsanalyseverktøy: <https://www.image-engineering.de/products/software/iq-analyzer-x> <https://deltae.picturae.com/wiki>
http://www.imagescienceassociates.com/mm5/software/GoldenThread_Users_Manual.pdf <https://jhove.openpreservation.org/>

IIF: <https://github.com/IIF/awesome-iif>

ISO 19264-1:2021: <https://www.iso.org/obp/ui/#iso:std:iso:19264:-1:ed-1:v1:en>

Digitalarkivet: <https://www.digitalarkivet.no/content/879/veikart-for-digitalarkivet>

Appendix C

Project plan

Prosjektplan

Martin Wighus Holtmon
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Januar 2022

Innhold

1	Mål og Rammer	6
1.1	Bakgrunn	6
1.2	Prosjektmål	6
1.2.1	Resultatmål	6
1.2.2	Effekt mål	7
1.2.3	Læringsmål	7
1.3	Rammer	7
2	Omfang	8
2.1	Fagområde	8
2.2	Avgrensning	8
2.3	Oppgavebeskrivelse	8
3	Prosjektorganisering	10
3.1	Organisasjonskart	10
3.2	Ansvarsforhold og Roller	10
3.3	Rutiner og Regler	11
3.3.1	Brudd på regler	12
4	Planlegging, oppfølging og rapportering	12
4.1	Hovedinndeling	12
4.2	Valg av metode	12
4.2.1	Scrum-prosessen	13
4.2.2	Scrum Board	14
4.2.3	Sprintplanlegging	14
4.2.4	Sprinten	14
4.2.5	Sprintmøte og retrospektiv	15

4.3	Møteplan	15
5	Plan for gjennomføring	16
5.1	Gantt-skjema	16
5.1.1	Gantt-skjema i tabellform	16
5.2	Milepæler	17
6	Organisering av kvalitetssikring	18
6.1	Dokumentasjon, standardbruk og kildekode	18
6.1.1	Kodestruktur	18
6.1.2	SonarLint og Pylint	19
6.2	Plan for testing og produksjonssetting	19
6.2.1	Git arbeidsflyt og GitLab	19
6.2.2	Testing og Pipelines	20
6.3	Verktøy	20
6.4	Risikoanalyse og tiltak	21
6.4.1	Risikoanalyse	21
6.4.2	Tiltak	22
	Referanseliste	23

Figurer

1	Ansvarsforhold	10
2	Scrum Prosessen	13
3	Gantt-skjema	16

Tabeller

1	Gantt-skjema i tabellform	17
---	-------------------------------------	----

2	Risikoanalyse	21
3	Tiltakliste	22

Akronym

CI/CD Continuous Integration / Continuous Delivery.

XMP Extensible Metadata Platform.

SEDAK Fylkessenter for Digitalisering av Kulturarven.

ISO International Organization for Standardization.

MR Merge Request

1 Mål og Rammer

1.1 Bakgrunn

Interkommunalt arkiv for Møre og Romsdal IKS v/Fylkessenter for digitisering av kulturarven (SEDAK) ble opprettet av Møre og Romsdal fylkeskommune i 2013. De har ansvar for å digitisere arkiver, bibliotek og museum, og kulturverninstitusjonene i Møre og Romsdal. Digitisering er overgangen fra analog til digital informasjon uten å tilføye og/eller endre funksjonalitet vha. digitale verktøy. [1] Dette arbeidet er viktig fordi det bidrar til å unngå slitasje av analogt skapt materiale gjennom å mediekonvertere det. Digitale filer blir sikret for fremtiden gjennom digital langtidsbevaring og innholdet blir digitalt publisert for formidling og forskning. [2]

Digitalarkivet er Norges fellesløsning for mottak, bevaring og publisering av digitiserte og mediekonverterte historiske arkiv fra statlige, kommunale og private aktører. Arkivinstitusjoner i hele Norge publiserer sine arkiv til Digitalarkivet. Dette prosjektet ble satt i gang i 2018. [3]

Digitalarkivet er utviklet av Arkivverket, og gjør det mulig å søke i strukturerte data på tvers av arkiv, fulltekstavskrifter, se fotografi, videoer og høre digitisert lyd om mulig. Første versjon av Digitalarkivet var en nettside utviklet som et plattform til publisering av kildemateriale for Arkivverket og andre i 1998. [4]

Når arkivene bli digitisert er kvalitetssikring nødvendig. Vi har blitt bedt om å følge en ISO standard som definerer visse krav til blant annet oppløsning, bildestøy og feil ved skanning. Hittil er dette gjort manuelt av de som legger inn materialet. Hvis dette kan bli automatisert vil det kunne blitt spart mye tid og menneskelig feilvurdering av kvaliteten på materialet vil kunne bli redusert. SEDAK, i samarbeid med Arkivverket, ønsker derfor å få utviklet en automatisert kvalitetssikringsmodul.

1.2 Prosjektmål

Vi har delt prosjektmålene i tre kategorier: resultatmål, effektmål og læringsmål. Resultatmål er de målene prosjektet skal oppnå. Effektmål er det prosjektet skal oppnå for oppdragsgiver. Læringsmål er det studentene ønsker å lære gjennom utførelse av prosjektet.

1.2.1 Resultatmål

- Lage en web-basert, automatisert kvalitetssikringsmodul og koble det til Digitalarkivets back end løsning ”Arkivdigitalisering”. Enten vil vi lage en API som kan brukes eller direkte integrere løsningen vår.

- Lage et brukervennlig grafisk brukergrensesnitt.
- Lage en back end som integrer et bildekvalitetssikringsprogram til å vurdere bilder.

1.2.2 Effektmål

- Automatisere kvalitetssikring for digitisering av analogt materiale, som sparer tid for de ansatte.
- Forbedre kvalitetssikringen av filer ved å unngå menneskelige feil.
- Spare arbeidstid for de ansatte som kun må vurdere materiale hvor den automatiske modulen feiler eller at materialet ikke består minimumskravet til kvalitetsanalysen.

1.2.3 Læringsmål

- Lære hvordan vi lager et brukergrensesnitt som er intuitivt og brukervennlig.
- Hvordan bruke Scrum i et reelt prosjekt.
- Viderutvikle programmerings- og programvareutviklingskompetansen i PHP og Python.
- Lære hvordan å bruke og integrere tredjeparts kvalitetssikrings programmer i vårt prosjekt.
- Kvalitetssikring av kode ved bruk av en *CI/CD pipeline*.

1.3 Rammer

- Programmet skal kjøre på Windows og/eller Linux. MacOS kompatibilitet vurderes, men er ikke prioritert.
- Skal ha et web-basert brukergrensesnitt.
- Programmet skal bli integrert i Digitalarkivet sin eksisterende back-end.

2 Omfang

2.1 Fagområde

Hver arkivinstitusjon i Norge må digitisere sine egne arkiv som skal bli lastet opp til Digitalarkivet, med både ulikt utstyr og resultat. Det er viktig å kvalitetssikre alt materiell som blir lastet opp til Digitalarkivet. Arkivmaterialet er hovedsaklig papirbasert, for eksempel bøker, dokumenter, kart, og tegninger osv. Dette materialet er i mange ulike størrelser og format. Modulen vi skal lage må gjennomføre sortering på kvalitetsnivå i samsvar med ISO 19264-1:2021[5]. ISO-standarden definerer tre kvalitetsnivåer, A, B og C. Filene skal valideres ved bruk av JHOVE eller et annet valideringsprogram bestemt av gruppen, og sikre at informasjon om dette følger i metadataen til filene via XMP.

Teknologiene som skal brukes for å automatisere denne prosessen er:

- Python og/eller PHP.
- JHOVE for filvalidering.
- XMP for metadata.
- Tredjeparts kvalitetsmålingsprogramvare.

Disse er ønsket av oppdragsgiver, og vi vil prøve å bruke disse, men vi kan selv vurdere om andre teknologier som oppfyller samme krav vil være mer hensiktsmessig å bruke.

2.2 Avgrensning

Vårt ansvar er å utvikle en modul for kvalitetssikring og organisering. Dokumentene og digitaliseringsprosessen er ikke en del av prosjektomfanget. Dette produktet lages for intern bruk hos Interkommunalt arkiv for Møre og Romsdal IKS, og for bruk av Arkivverket i Digitalarkivet. Vi vil mest sannsynlig bruke programmeringsspråkene PHP og Python, fordi de brukes i Digitalarkivet. Hvis vi lager en selvstendig API har vi muligheten til å velge dette selv. Utover dette har vi i stor grad valgfrihet for hva slags teknologier vi vil bruke.

2.3 Oppgavebeskrivelse

Målet med prosjektet er å definere og utvikle en automatisert kvalitetssikringsmodul, og koble den til Digitalarkivets backend ”Arkivdigitalisering”. Brukergrensesnittdelen av modulen burde tillate ulike arkivinstitusjoner å enkelt laste opp sine

digitiserte filer til det sentrale arkivet. Backend-delen av modulen skal organisere disse digitiserte filene basert på kvaliteten og passe på at de lagres riktig sammen med informasjonen om kvaliteten.

Funksjonelle krav:

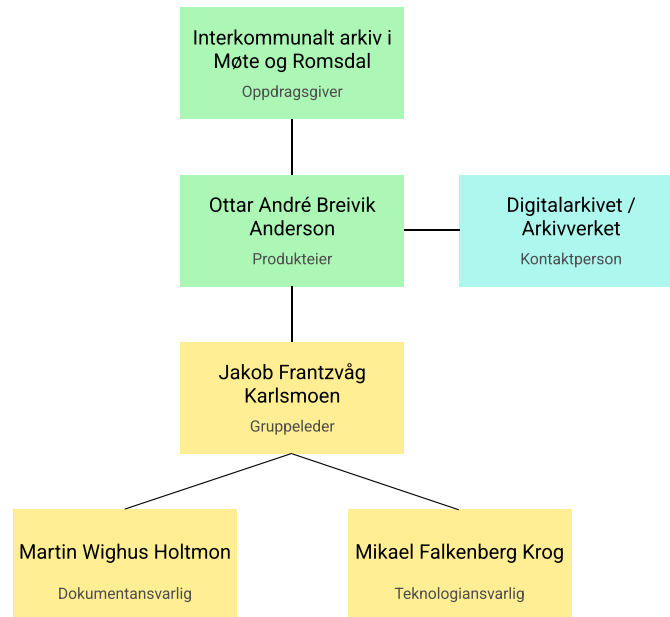
- Brukeren skal kunne laste opp digitale filer.
- Programmet skal ha et web-basert brukergrensesnitt.
- Programmet skal sortere bildene etter A, B, og C gradering i følge ISO 19264-1:2021.
- Resultatet fra kvalitetsmålingsprogramvare om bildet skal lagres i bildets metadata.
- Filene skal bli validert med JHOVE eller et liknende program.
- Metadataen skal være i XMP format eller et annet hensiktsmessig format.

Ikke-funksjonelle krav:

- Programmet skal ikke krasje.
- Testene skal være hensiktsmessige og gode.
- Kodebasen skal være ryddig og forståelig for en utenforstående utvikler.
- Koden og bruken av programmet skal være godt dokumentert.
- Brukergrensesnittet skal være brukervennlig.

3 Prosjektorganisering

3.1 Organisasjonskart



Figur 1: Ansvarsforhold

3.2 Ansvarsforhold og Roller

- **Produkteier - Ottar André Breivik Anderson**
 - Vår kontaktperson hos Interkommunalt arkiv i Møre og Romsdal v/Fylkessenter for digitisering av kulturarven.
 - Kommer med innspill slik at vi lager et produkt som oppfyller kravene deres.
- **Veiledere - Rune Hjelsvold, Johanna Johansen**
- **Gruppeleder - Jakob Frantzvåg Karlsmoen**
 - Har ansvar for å koordinere gruppen.
 - Passer på at gruppen jobber jevnt og effektivt, og ikke faller bak tidsskjema.
 - Gruppelederen er også Scrum master, og har ansvar for strukturering og administrering av Scrum prosessen.

- Ansvarlig for dokumentering av arbeidsprosessen og fremgangen i prosjektet.
- Dokumentansvarlig - Martin Wighus Holtmon
 - Ansvarlig for å skrive møtereferat.
 - Forberede agenda, tid, sted for alle møter, samt å sende ut møteinvitasjoner.
 - Passe på at nødvendige dokumenter blir skrevet og lagret, som for eksempel se til at ukeslogg og sprint review blir dokumentert.
- Teknologiansvarlig - Mikael Falkenberg Krog
 - Passer på at alle følger god Git arbeidsflyt, og at alle tar i bruk pipeline for testing.
 - Ansvarlig for å holde seg oppdatert på nye relevante teknologiske verktøy.
- Utviklere - Martin Wighus Holtmon, Jakob Frantzvåg Karlsmoen, Mikael Falkenberg Krog
 - Utvikle programvaren etter gruppens kodestandarder som har blitt avtalt på forhånd.

3.3 Rutiner og Regler

1. Hvert medlem er ansvarlig for å notere ned dens egen arbeidstid samt en ukesrapport med hva dem har gjort i løpet av uken.
2. Gruppen har som mål å ha et minimum med to ukentlige møter, enten fysisk eller digitalt, hvis ikke annet blir avtalt.
3. Hvert medlem må gi beskjed i god tid om dem ikke kan møte til planlagte aktiviteter så langt det lar seg gjøre.
4. Alle må holde tidsfrister som er satt av gruppen.
5. Alle må gjøre arbeid som er tildelt til dem.
6. Gruppen må følge samme kode- og dokumentasjonsstandard.
7. Hvis det er uenigheter i gruppen skal vi forhøre oss med en ekstern person for å få et nytt synspunkt på problemet. Hvis dette ikke lar seg gjøre, så faller vi tilbake på at det er flertallet som bestemmer.

3.3.1 Brudd på regler

Vi skal prøve å ta tak i problemer som oppstår tidlig, slik at vi kan løse de på en god og rask måte. Gruppemedlemmer som har brutt regler vil få klare leveransekrav som må bli oppfylt for medlemmets videre delaktighet i prosjektet. Hvis dette kravet ikke blir møtt, og det er et samtykke blant resten av gruppe-medlemmene samt veiledere, kan medlemmet bli bedt om å forlate gruppen.

Løsningen vår for å minimere konsekvensene av at et medlem må forlate gruppen er å ha god kommunikasjon gjennom prosjektet slik at alle på gruppen vet hva de andre arbeider med. Dette gjør at et annet medlem kan fortsette arbeidet til den som forlater gruppen.

4 Planlegging, oppfølging og rapportering

4.1 Hovedinndeling

Prosjektet er delt i fire faser som vil bli utført i løpet av prosjektets livsløp. Disse fire fasene er planleggingsfasen, utviklingsfasen, rapportfasen, og presentasjonsfasen. Det blir viktig å følge disse fasene så godt vi klarer siden tidsrammen vi har på prosjektet er kort. Vi har planlagt at de to første ukene skal brukes til prosjektplanlegging hvor vi skriver en prosjektplan. De neste 20 ukene blir fylt av 5 sprinter hvor vi utvikler løsningen. Planen er at vi skal skrive på rapporten i løpet av hele perioden, men de 4 siste ukene blir dedikert til rapportskrivning. For en nøyere oversikt, se 5 Plan for gjennomføring.

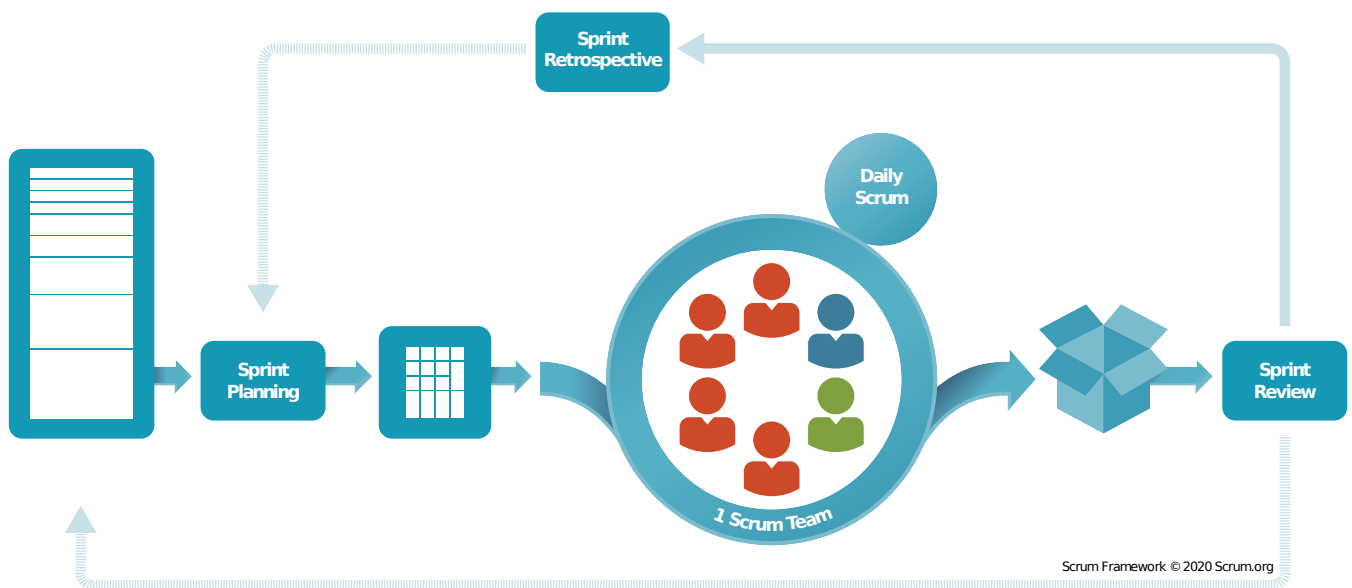
4.2 Valg av metode

De metodikkene vi vurderte var hovedsakelig Scrum og Kanban, altså smidige metoder. Oppgaven vi fikk er tredelt, men den burde gjøres i rekkefølge, og vi må ha muligheten til å gå tilbake et eller flere steg hvis det blir noen endringer i planene. Vi er i tett kontakt med oppdragsgiver og derved har mulighet til å få regelmessig tilbakemelding noe som gjør at en smidig metode passer godt. Oppgaven er åpen for endringer, for eksempel må vi vurdere hvilken kvalitets-sikrings programvare vi skal bruke, eventuelt videreutvikle denne, eller utvikle vår egen løsning. Det er også åpnet opp for at vi kan ta i bruk flere rammeverk for å øke visuell kontroll av filene. En annen metodikk som fossefallmetoden gir ikke like god mulighet for endringer da den er en lineær metode.

Den smidige metoden gruppen landet på er Scrum. Grunnen til at vi valgte dette over Kanban er at vi har jobbet med Kanban før og ønsket å prøve oss på noe nytt. Perioden som er satt av til utvikling er relativt kort, så det er viktig å kunne vise progresjon og få tilbakemelding fra produkteier ofte. Dette gjøres

i et sprintmøte på slutten av hver sprint. Vi er også et lite utviklings team, så å dele ansvarsområder er nok lite relevant eller spesielt lurt med tanke på sykdom, det er viktig at alle kan jobbe med alt. Scrum gir muligheten til å holde god oversikt over hvilke aktiviteter som må gjøres og hva som skal gjøres i pågående sprint. Ved hjelp av et Scrum board kan vi se hvem som jobber med hva. Dette gir oss muligheten til å prioritere oppgavene som må gjøres i rekkefølge da de kan bli plassert i forskjellige sprinter. Vi har bestemt at vi skal ha fem sprinter som varer ti arbeidsdager. Etter hver sprint vil det holdes et sprintmøte hvor vi ser på hva vi fikk til og hva vi ikke fikk til i løpet av sprinten.

4.2.1 Scrum-prosessen



Figur 2: Scrum Prosessen

4.2.2 Scrum Board

Fase	Beskrivelse
Product Backlog	En liste med alle forretningsmessige tekniske krav og funksjoner som må bli gjort for å gjøre ferdig prosjektet. Denne listen blir vedlikeholdt av produkteier og Scrum master.
Sprint Backlog / To do	En prioritert kø med funksjoner som skal bli implementert i løpet av en sprint.
In progress	Liste med funksjoner som holder på å bli implementert av en av utviklerene.
Review (MR)	Når en funksjon eller et krav har blitt implementert, så vil utviklerene lage en <i>MR (merge request)</i> hvor implementasjonen blir satt til <i>review</i> . Et annet medlem på gruppen må se over og godkjenne dette før den blir merget inn i <i>main branch</i> .
Done	Alle funksjoner og implementasjoner som er ferdig implementert.

4.2.3 Sprintplanlegging

Arbeidet som skal gjøres i en sprint blir planlagt under planleggingsfasen. Sprintplanen blir laget sammen med hele Scrum teamet og svarer på spørsmål om hva som skal bli levert i denne sprinten og hvilket arbeid som er nødvendig for å oppnå dette inkrementet. Scrum master, i samarbeid med produkteier, presenterer målet for sprinten og hva som ønskes utviklet for å nå målet. Ut ifra det velger utviklingsteamet elementer fra release backlog for å tilfredsstille disse målene og som dem vurderer som oppnåelig i løpet av en sprint. Sprint planleggingen skal maksimalt vare fire timer for en to ukers sprint [6].

4.2.4 Sprinten

Vi har bestemt at hver sprint skal vare i 10 arbeidsdager hvor vi kjører planlegging på starten av hver sprint, og et møte på slutten fulgt av retrospektiv refleksjon av sprinten. Dette er perioden hvor teamet utvikler det som ble definert under Sprintplanleggingen. Dersom et sprintmål ikke lenger har noen verdi, vil målet revurderes underveis i sprinten. Dette kan skje hvis prosjektplanen endres eller ny teknologi blir tilgjengelig eller oppdaget. Siden en sprint er kun over en kort periode er det veldig sjeldent den blir kansellert. Hvis det skulle skje vil produkteier og Scrum master vurdere produktelementene som ble ferdig i løpet av sprinten om dem er leveranseklart eller om dem skal legges tilbake i produktkøen.

4.2.5 Sprintmøte og retrospektiv

Etter hvert Sprint så gjennomfører teamet et sprintmøte. I dette møtet skal det gjennomgås hva som har blitt gjort, ikke blitt gjort, nytt arbeid som har oppstått og arbeid som har blitt fjernet. Det åpnes for muligheter for å stille spørsmål og gi tilbakemelding på arbeidet til de andre gruppe-medlemmene. [7]

I tillegg til gjennomgang av arbeid skal teamet også gjøre et tilbakeblikk for å finne forbedringsområder. I motsetning til å fokusere på produktet, skal fokuset plasseres på teamet. Dette starter ved at teamet identifiserer hva som gikk bra og hva som kunne blitt forbedret, samt lage en plan for forbedringer som skal implementeres i neste sprint. [8]

Møtet burde ikke vare mer enn én og en halv time. Det skrives også rapporter fra disse møtene, for å dokumentere arbeidsprosessen og fremgangen i prosjektet.

4.3 Møteplan

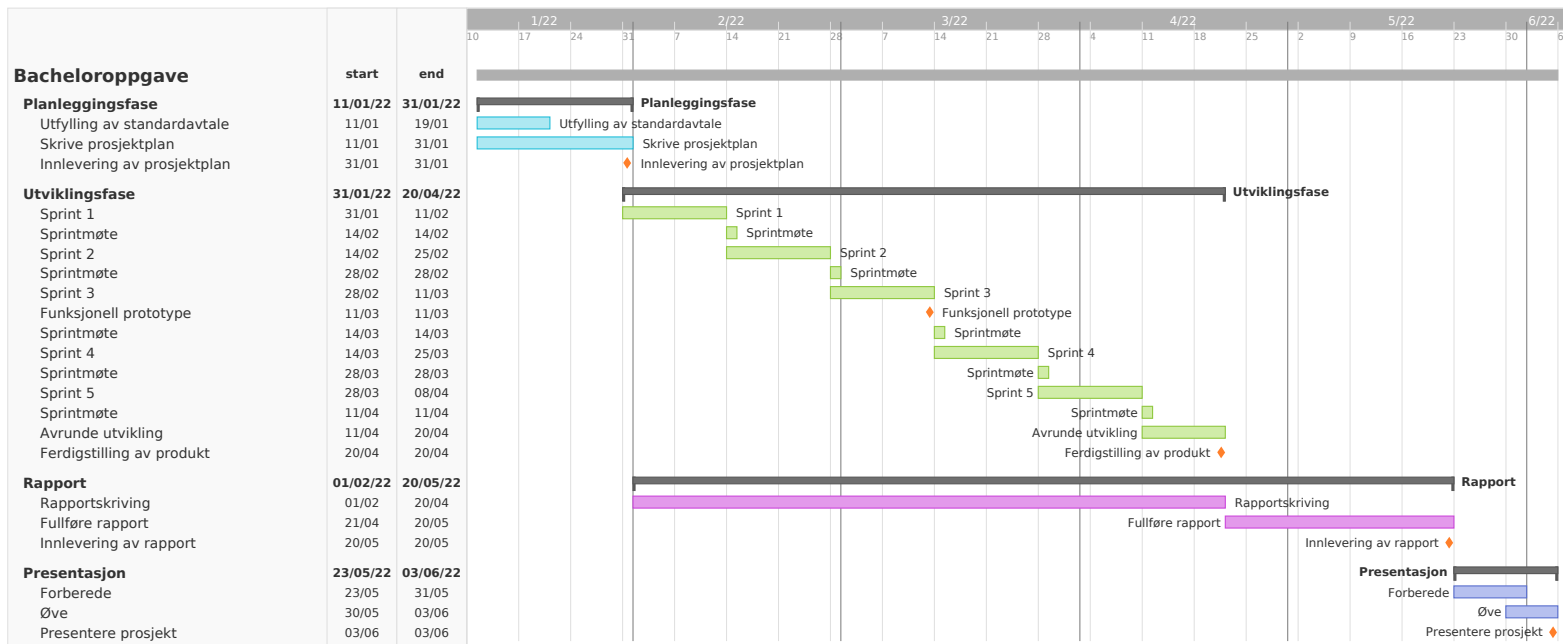
Det er blitt laget en plan for faste hendelser hver uke. Denne planen er åpen for endringer etterhvert som prosjektet utvikler seg, men er viktig base for å holde struktur på arbeidet. Planen definerer felles arbeidstimer for gruppe-medlemmene, og møter med veiledere og oppdragsgivere.

Planen er som følger:

- Arbeidstimer
 - Mandag kl 10.15 - kl 14.00
 - Torsdag kl 10.15 - kl 14.00
 - Fredag kl 10.15 - kl 14.00
- Møte med oppdragsgiver
 - Mandag kl 13.00 - kl 14.00
- Møte med veiledere
 - Mandag kl 14.00 - kl 15.00

5 Plan for gjennomføring

5.1 Gantt-skjema



Figur 3: Gantt-skjema

Gantt-skjemaet i Figur 3 viser en grov oversikt over faser, oppgaver og milepæler til prosjektet.

5.1.1 Gantt-skjema i tabellform

I Tabell 1 er det listet opp eksakte datoer for innholdet i Gantt diagrammet i Figur 3.

WBS #	Tittel	Type	Startdato	Sluttdato
1	Bacheloroppgave	Prosjekt	11. januar	3. juni
1.1	Planleggingsfase	Fase	11. januar	31. januar
1.1.1	Utfylling av standardavtale	Oppgave	11. januar	19. januar
1.1.2	Skrive prosjektplan	Oppgave	11. januar	31. januar
1.1.3	Innlevering av prosjektplan	Milepæl	31. januar	
1.2	Utviklingsfase	Fase	31. januar	21. april
1.2.1	Sprint 1	Oppgave	31. januar	11. februar
1.2.2	Sprintmøte	Oppgave	14. februar	
1.2.3	Sprint 2	Oppgave	14. februar	25. februar
1.2.4	Sprintmøte	Oppgave	28. februar	
1.2.5	Sprint 3	Oppgave	28. februar	11. mars
1.2.6	Funksjonell prototype	Milepæl	11. mars	
1.2.7	Sprintmøte	Oppgave	14. mars	
1.2.8	Sprint 4	Oppgave	14. mars	25. mars
1.2.9	Sprintmøte	Oppgave	28. mars	
1.2.10	Sprint 5	Oppgave	28. mars	8. april
1.2.11	Sprintmøte	Oppgave	11. april	
1.2.12	Avrunde utvikling	Oppgave	11. april	20. april
1.2.13	Ferdigstilling av produkt	Milepæl	20. april	
1.3	Rapport	Fase	1. februar	20. mai
1.3.1	Rapportskriving	Oppgave	1. februar	20. april
1.3.2	Fullføre rapport	Oppgave	21. april	20. mai
1.3.3	Innlevering av rapport	Milepæl	20. mai	
1.4	Presentasjon	Fase	23. mai	3. juni
1.4.1	Forberede	Oppgave	23. mai	31. mai
1.4.2	Øve	Oppgave	30. mai	3. juni
1.4.3	Presentere prosjekt	Milepæl	3. juni	

Tabell 1: Gantt-skjema i tabellform

5.2 Milepæler

I dette prosjektet har vi definert 5 milepæler:

- WBS 1.1.3 - Innlevering av prosjektplan
 - Prosjektplanen skal være ferdigstilt, godkjent og levert inn slik at arbeid på prosjektet kan begynne.
- WBS 1.2.6 - Funksjonell prototype
 - Vi ønsker at etter 3. sprinten skal vi ha kommet oss til et punkt der hovedfunksjonaliteten i modulen skal fungere.
- WBS 1.2.13 - Ferdigstilling av produkt
 - Når utviklingsfasen er over skal produktet ferdigstilles slik at det kan tas i bruk.
- WBS 1.3.3 - Innlevering av rapport
 - I slutten av arbeidsperioden skal rapporten leveres inn. Den må være helt ferdig, og ha med alle dokumentene fra prosjektet som vedlegg.
- WBS 1.4.3 - Presentere prosjekt
 - Presentasjonen skal inneholde en oversikt over prosessen og sluttresultatet fra prosjektarbeidet.

6 Organisering av kvalitetssikring

6.1 Dokumentasjon, standardbruk og kildekode

I løpet av prosjektet har gruppen blitt enige om å følge en rekke standarder for kode- og dokumentasjonsstruktur, i tillegg skal vi også bruke utvidelser som SonarLint og Pylint for å forsikre oss om at koden standarden blir fulgt. Der hvor automatisk sjekk ikke mulig må vi etter beste evne følge og verifisere at vi følger kodestrukturene selv.

Funksjonalitet skal bli forklart i koden, men det skal også bli dokumentert i vår GitLab wiki med eksempler på bruk samt en presis forklaring av funksjonaliteten. GitLab wikien skal være en brukerhåndbok for løsningen som forklarer all relevant informasjon om hvordan den kan brukes og implementeres.

6.1.1 Kodestruktur

For Python kode har vi tenkt å bruke Google's kodenstandard for Python. Dette er for å ha en konsekvent struktur på koden og dokumentasjonen [9]. Det er viktig at alle klasser, moduler og skripter blir kommentert ved bruk av *docstrings*,

og kompliserte kodesnutter burde kommenteres med block og inline kommentarer som forklarer litt nøyere hva som skjer. Dette vil hjelpe både oss som utviklere og sluttbrukeren. Ved å skrive gode *docstring* kommentarer som forklarer funksjonaliteten av en kommando, så kan dette bli printet når programmet er i bruk med *help()* funksjonen i Python. For PHP har vi tenkt å bruke PHPDoc som standard for dokumentering og SonarLint sine standard regler for kodestruktur.

6.1.2 SonarLint og Pylint

Utvidelsene vi kommer til å ta i bruk for forsikre oss at kodestandardene blir holdt er SonarLint og Pylint. Disse blir lagt rett inn i utviklingsmiljøet vårt som en utvidelse og vil hjelpe oss med å indentifisere og håndheve kodestrukturen mens vi koder i sanntid. Ved å bruke disse utvidelsene unngår vi unødvendig rot i *commits* ved at kodeformateringen er konsekvent mellom alle utviklerne. De vil også gi oss tips til hvordan vi kan forbedre kodekvaliteten eller gjøre koden sikrere.

6.2 Plan for testing og produksjonssetting

6.2.1 Git arbeidsflyt og GitLab

Vi har planlagt å bruke en *trunk-based workflow*.^[10] Denne typen arbeidsflyt er standard praksis hos moderne utviklere. I denne arbeidsflyten vil små oppdateringer bli *merget* til en *main branch* ofte. Dette gjør det lett for andre å se over ny kode som har blitt skrevet. Denne arbeidsflyten er også standard når man skal ta i bruk *CI/CD pipelines*, noe som vi har planlagt å bruke. Når man har hyppige og små oppdateringer vil *CI/CD* være mer til nytte enn hvis vi hadde hatt store og sjeldne oppdateringer.

Main branchen skal alltid være stabil og fungere. Utifra *main branchen* lages det *feature branches*, som er kortlevde *branches* som skal *merges* inn i *main branchen*. Hver *merge request* må godkjennes av minst en annen person på gruppen for at den skal kunne bli *merget*. Dette er for å sikre kvaliteten på koden. Alle andre *branches* skal være basert på *main branchen*, og *branches* skal *merges* ofte. Dette hjelper å redusere *merge conflicts*.

Vi skal bruke GitLab sine verktøy for å håndtere *issues*. Vi vil sette dette opp som en Scrum tavle, slik at vi lett har oversikt over hva som skal gjøres.

6.2.2 Testing og Pipelines

For hver ny *merge request* må man forsikre seg om at koden kjører og at alle tester har blitt kjørt vellykket. Dette gjør vi ved hjelp av en *CI/CD Pipeline*.

Vi skal lage tester til kritiske funksjoner. Vi skal prøve å lage disse testene så tidlig som mulig slik at de kan bli brukt i en *CI/CD Pipeline*. Vi vil også bruke en *linter* i *pipelinen* slik at vi ikke må gjennomgå koden manuelt. En mulighet er at vi tar i bruk SonarQube.

6.3 Verktøy

Navn	Beskrivelse
SonarLint	Kontrollerer kodestruktur mens vi koder.
Pylint	Kontrollerer kodestruktur mens vi koder.
SonarQube	Kontrollerer kodestruktur kontinuerlig ved at den blir integrert inn i vår <i>CI/CD pipeline</i> .
GitLab	Plattform for kildekode og versjonskontroll.
Overleaf	LaTeX editor som brukes til å skrive prosjektrapport og prosjektplan.
Discord	Kommunikasjonsplattform for samarbeid med gruppen.
Google Docs	Skriveprogram som brukes til møtereferat og andre relevante dokumenter.
Google Sheets	Regnearkprogram som brukes til loggføring av timer.
Google Drive	Fillagringstjeneste for dokumenter og regneark.
Microsoft Teams	Kommunikasjonsplattform for møter med veiledere og produkteier.
OpenStack/SkyHiGh	<i>Hosting</i> tjeneste hvor vi kan sette opp virtuelle servere.
Team Gantt	Program for å lage Gantt skjema for prosjektet.

6.4 Risikoanalyse og tiltak

6.4.1 Risikoanalyse

Nummer	Beskrivelse	Sannsynlighet	Konsekvens	Tiltak
1	Ta for oss for mye arbeid.	Usannsynlig	Problematiske	Ja
2	Langvarig sykdom i gruppen.	Usannsynlig	Kritisk	Ja
3	Et medlem av gruppen mister motivasjon og forlater gruppen.	Svært usannsynlig	Kritisk	Ja
4	Tap av kode eller rapport.	Usannsynlig	Kritisk	Ja
5	Endring av prosjektbeskrivelse eller kravspesifikasjoner.	Sannsynlig	Problematiske	Nei
6	Vanskeligheter med integrering av funksjonalitet.	Svært Sannsynlig	Problematiske	Ja
7	Funksjonalitet vi ikke klarer å implementere.	Sannsynlig	Problematiske	Ja
8	Konflikt innad i gruppen.	Svært Usannsynlig	Problematiske	Ja

Tabell 2: Risikoanalyse

6.4.2 Tiltak

Nummer	Tiltak
1 og 7	Hvis vi ser at vi har tatt for oss for mye arbeid eller at vi ikke klarer å implementere en funksjonalitet, vil vi ha møter med veiledere og oppdragsgiver og sannsynligvis kutte ned på arbeidet eller finne et alternativ. For å oppdage disse problemene må gruppe-medlemmene gi regelmessig tilbakemelding til resten av gruppen på de planlagte fellesmøtene, og hvis et medlem ikke får til noe er det viktig å gi beskjed.
2 og 3	Vi sørger for at en langvarig sykdom ikke vil påvirke oss i stor grad ved å ha god kommunikasjon med hverandre, slik at vi vet hva de andre gruppe-medlemmene gjør. Vi vil også dokumentere godt slik at det er mulig å fortsette på andre sitt arbeid. Dette vil også hjelpe hvis et av gruppe-medlemmene forlater gruppen.
4	Vi forsikrer oss om at tap av rapporten ikke vil skje ved å lagre den både i Overleaf og på GitLab. Koden vil også være lagret på GitLab, i tillegg til at den er lagret lokalt hos alle gruppe-medlemmene.
6	Hvis vi har vanskeligheter med å integrere funksjonalitet vil vi først bruke litt tid på å prøve å få det til å funke, hvis dette ikke går vil vi ta møte med veiledere for å få hjelp eller diskutere alternative løsninger. Vi vil også prøve å integrere løsningene våre regelmessig slik at vi kan oppdage integrasjonsproblemer tidlig
8	Hvis det blir konflikter innad i gruppen vil vi ha møte med veiledere for å bestemme veien videre og eventuelt løse konflikten, se 3.3.1 Brudd på regler for mer utdypning.

Tabell 3: Tiltaksliste

Referanseliste

- [1] Paal Leveraas. *Fra digitisering til digitalisering*. URL: <https://www.dataforeningen.no/fra-digitisering-til-digitalisering/> (sjekket 31. jan. 2022).
- [2] *Om SEDAK / Digitisering*. URL: <https://www.ikamr.no/om-sedak-digitisering> (sjekket 12. jan. 2022).
- [3] *Digitalarkivet som nasjonal publiseringsplattform for skanna arkivmateriale*. URL: <https://www.digitalarkivet.no/content/442/digitalarkivet-som-nasjonal-publiseringsplattform-for-skanna-arkivmateriale> (sjekket 20. jan. 2022).
- [4] *Historia til Digitalarkivet*. URL: <https://www.digitalarkivet.no/content/55/historia-til-digitalarkivet> (sjekket 31. jan. 2022).
- [5] *Photography — Archiving systems — Imaging systems quality analysis — Part 1: Reflective originals*. URL: <https://www.iso.org/standard/79172.html> (sjekket 25. jan. 2022).
- [6] *What is Sprint Planning?* URL: <https://www.scrum.org/resources/what-is-sprint-planning> (sjekket 20. jan. 2022).
- [7] *What is Sprint Review?* URL: <https://www.visual-paradigm.com/scrum/what-is-sprint-review> (sjekket 25. jan. 2022).
- [8] *What is a Sprint Retrospective?* URL: <https://www.scrum.org/resources/what-is-a-sprint-retrospective> (sjekket 20. jan. 2022).
- [9] *Google Python Style Guide*. URL: <https://github.com/google/styleguide/blob/gh-pages/pyguide.md#38-comments-and-docstrings> (sjekket 18. jan. 2022).
- [10] Kev Zettler. *Trunk-based development*. URL: <https://www.atlassian.com/continuous-delivery/continuous-integration/trunk-based-development> (sjekket 18. jan. 2022).

Appendix D

Sprint 1 - Review and Retrospective

Sprint 1: Review og Retrospektiv

Sprint review

Hva har vi gjort:

- Sett på og prøvd ut IQ-Analyser-X, ImageZebra, OpenDICE
 - Alle tre er lovende for analyse av enhetsnivå bildemål
 - Ønsker å bruke et program som kjører lokalt, typ IQ-Analyser-X og OpenDICE
 - Men typ ImageZebra har nok noe lettere implementasjon mtp. dårlig måte å bruke lisens på
- Sett på og prøvd ut JHOVE, Bad Peggy, MediaConch og KOST-val
 - Landet på JHOVE
- Har testet en python package som heter pyexiv2 for å lagre metadata i filer.
- Definert MVP og startet på use cases
- Kommet i gang med hovedrapporten
- Forbedret gruppens kompetanse
 - Bedre forståelse for problemstilling og teori
 - Fått oppklaring i hvordan vi gjør analyse av scan/systemet (Object/device level target)
- Revidert timeplan med mer arbeidstimer

Hva vi planla, men ikke har gjort:

- Ikke fått prøvd ut OS QM TOOL eller GoldenThread pga. Lisenser
 - GoldenThread vil vi kunne se på neste sprint vha. Ottar
- Ingen bilder med objektnivå bildemål, så ingen mulighet til å teste med det

Nåværende status og plan fremover:

- MVP definert
 - Trenger å lage wireframe for UX/UI
 - Lage use case diagrammer
- Har fått oversikt over noen alternativer for analyseringsprogram
 - Trenger å velge et slik at man kan begynne å utvikle rundt det.
- Valgt JHOVE som integritetssjekker
- Bestemt filtyper som TIFF og JPEG
 - Trenger å finne ut god prosedyre for å lagre metadataen med/i filene
- Begynne å forme programmet og utvikle prosessen i kode
 - Bruke mock-data der deler mangler (f. eks. analyseprogram)
- Skrive på rapport:
 - Argumentere rundt programmene og hvorfor vi velger det vi velger osv.
 - Arbeidsprosessen.

Sprint Retrospektiv

Hva gikk bra denne sprinten:

- Klart å gjøre arbeid selv om vi har støtt på problemer
 - Manglende tilgang til programmer
 - ImageZebra
 - Uklarheter i oppgaven
- Godt samarbeid mellom medlemmene
 - Jobbet godt i lag
- Kommet godt i gang

Usikkerheter / Spørsmål:

Hva som ikke fungerer så bra:

- Mye miks av digital og fysisk jobbing

Forbedringer:

- Prøve å ha mer av de faste arbeidstimene fysisk

Appendix E

Sprint 2 - Review and Retrospective

Sprint 2: Review og Retrospektiv

Goals

- Develop code and start to integrate quality analysis tool(s).
 - Input an image -> output image with quality analysis
- Develop code to extract & validate metadata and insert quality analysis.

Sprint review

Hva har vi gjort:

Hva	Notater	Hvorfor
Laget filvalideringsmodul	Filvalideringsmodul som bruker JHOVE.	
SonarQube & CI/CD Runner	CI/CD for testing med pytest og react tests. SonarQube for linting og analyse av koden. Kontinuerlig inspeksjon av koden for å øke kodekvalitet.	Hver gang vi merger kode inn i main, så vil SonarQube analysere koden for å se etter feil/bugs osv. Python testene våre blir også kjørt.
Møtt og jobbet med Ottar	Kartlagt programvarene, kontaktet produsenter for lisenser, diskutert rundt prosjektet for å få en bedre forståelse for hva vi jobber med og hvorfor	
Prøvd å ta i bruk OpenDICE	Konkludert med at det ikke var brukbart for våre formål pga. resultatet som eksponeres ikke var analyserbart	Selv om OpenDICE ikke i utgangspunktet så ut som en god kandidat, var planen å lage noe enkelt som viser funksjonaliteten i prosjektet fra start til slutt
API	Kjørbar kode som kan ta imot data og returnere. Gjøres på denne måten slik at frontend kan være helt uavhengig fra backend.	

Prototype av brukergrensesnitt	Lagde en prototype i Figma for brukergrensesnittet. Dette kommer til å være veldig enkelt for å vise funksjonaliteten til API'en.	Begynte på dette siden vi ikke fikk tilgang på programvarene vi trenger.
Frontend utvikling	Begynte å lage en web frontend i Reactjs. Hittil har vi laget startsiden.	

Hva vi planla, men ikke har gjort:

Hva	Hvorfor	Videre
Utvikling av bildeanalysemodul	Vi hadde håpet at vi fikk tilgang på programvarene i løpet av sprint 2, men det tok litt lenger tid enn ventet. Håper på å få tilgang i løpet av sprint 3	Vi har bekreftelse på de, må bare vente på lisensnøklerne blir sendt. Det er en fysisk dongle (usb). Burde også få tilgang til ImageZebra, men venter på svar fra dem.
Utvikling av metadata modul som legger ved bildeanalysen som metadata til bildene.	Siden vi ikke vet hvilken programvare vi skal bruke vet vi ikke helt hvordan responsen fra bildeanalysen vil se ut, og det ble derfor vanskelig å lage en metadata modul.	Selve modulen burde ikke være så mye jobb å utvikle, vi må bare vite hva slags format bildeanalysen vil være på.

Nåværende status og plan fremover:

Hva	Videre
Frontend: Fått implementert fremside for applikasjonen	<ul style="list-style-type: none"> - Ta hensyn til skalering - Begynne å se på accessibility (aria-label & tabindex...) - Implementer opplasting side og resultat side.
Har bekreftet tilgang til 3 kvalitetsanalyseprogrammer (ImageZebra, OS QM Tool, IQ Analyser X)	Når vi mottar lisensene, begynne å utvikle rundt ihvertfall ett av programmene.
CD	<p>Continous Delivery for minst frontend</p> <p>Backend er noe vanskeligere pga windows krav</p>

Windows maskin	Ordne istand Windows maskin som programmene kan kjøre på og som kan <i>hoste</i> backend løsningen.
----------------	---

Sprint Retrospektiv

Hva gikk bra denne sprinten

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre
Samarbeid med Ottar	Fikk diskutert mye og drøftet bruken av de forskjellige bildeanalyseprogramvarene	Det er kanskje lurt å ha flere slike arbeidsøker med Ottar
Klarte fortsatt å finne ting å gjøre selv om vi ikke fikk tilgang på programvarene.	Fordi vi har ganske mange arbeidsoppgaver som kan gjøres utenom bildeanalysen.	
Bruke tiden produktivt selv om det ble noen korte dager	Vi måtte bruke tiden på Mandag til presentasjoner, og klassens time tok mesteparten av tiden på Onsdag.	

Usikkerheter / Spørsmål

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre

Hva som ikke fungerer så bra

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre
Ingenting :)		

Appendix F

Sprint 3 - Review and Retrospective

Sprint 3: Review og Retrospektiv

Goals

- Develop the frontend
- Create the image quality analysis module
- Create prototype

Lower priority goals:

- Write about our current progress in the report

Sprint review

Hva har vi gjort:

Hva	Notater
Laget en fungerende frontend	Mangler bare api calls til backenden for å full funksjonalitet
Begynt å lage analyze endpoints til API'en	Laget endpoint for å starte analysen, begynt på å lage endpoint for å få ut resultatet. Har også endpoints for å lage sessions. Hver batch med bilder som blir analysert får sin egen session med en unik id, det blir automatisk laget mappestruktur for sessions
Laget resultat parsers for outputs til IQ-Analyzer X og OS QM-Tool	Laget funksjoner for å få dataen vi trenger på en strukturert måte for å kunne brukes og vises i framtiden.
Dokumentasjon for API	Dokumentert endpoints og fremtidige endpoints
Lært mye om IQ-Analyzer X	Etter å ha fått brukt programvaren har vi møtt på mange av programmets særheter og kan videre i utvikling arbeide rundt disse siden vi nå har mer erfaring.

Hva vi planla, men ikke har gjort:

Hva	Hvorfor	Videre
Har ikke fått koblet sammen frontend og backend ennå, siden backenden tok lengre tid enn antatt å lage.	Fikk lisensene senere enn vi hadde håpet.	Det vi lærte av dette var at ting som regel tar lengre tid enn antatt. For eksempel har XML dataen vi får fra et av programmene en feil som gjorde at vi måtte lage en funksjon som fikser dette. Dette førte til ekstra tidsbruk. Vi tar med oss dette videre og i fremtiden vil vi anta at flere problemer vil oppstå som fører til mer arbeid.
Fungerende MVP	Analyseprogrammene oppfører seg ikke konsekvent, noen ganger finner den targetet, andre ganger ikke.	

Nåværende status og plan fremover:

Hva	Videre
Ligger litt bak det vi hadde håpet på, men har gjort svært god fremgang denne uken.	Vi kommer til å jobbe iherdig selv om vi forventer at eksamensforberedning kommer til å ta en del tid av neste sprint. Vi kommer også til å få besøk av Ottar igjen mot slutten av sprint 4
Har ikke fått koblet sammen frontend og backend enda	Grunnet at backend ikke er helt klar, så er planen videre å begynne å koble disse sammen i løpet av sprint 4.

Sprint Retrospektiv

Hva gikk bra denne sprinten

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre
Har gjort svært god fremgang	Har en fungerende frontend, og analyse av bilder fungerer	Vi har lært at vår nåværende arbeidsstrategi fungerer godt.
Gruppearbeidet fungerer godt, med god fordeling av arbeidsoppgaver	Alle er motiverte og ønsker å arbeide. Derfor er det enkelt å fordele arbeid. Vi har vært mer motiverte fordi vi har fått tilgang til programmer som vi kan bruke i utviklingen.	Vi vil fortsette med samme metode å arbeide på, siden det fungerer godt.
All arbeidet har blitt gjort fysisk		Vi fortsetter med dette

Hva som ikke fungerer så bra

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre
Denne uken var et av gruppemedlemmene våre opptatt i deler av sprinten grunnet studentassistent arbeid.	DevOps workshop i uke 10.	Det gruppemedlemmet vil arbeide mer i fremtiden. Det har ikke vært et stort problem at de har arbeidet litt mindre, siden de har arbeidet når de hadde mulighet.

Appendix G

Sprint 4 - Review and Retrospective

Sprint 4: Review og Retrospektiv

Goals

- Få ferdig funksjonalitet som var planlagt til MVP
- Skrive på rapporten

Sprint review

Hva har vi gjort:

Hva	Notater	Hvorfor
Skrevet om (refactor) backend		Backendfunksjonene var for spesifikke, så de måtte skrives om for bedre funksjonalitet
Implementert kode for å kjøre OS QM Tool	Dette gikk mye raskere enn implementasjonen av IQ Analyser-X. Grunnen til dette er at vi har fått mye erfaring fra utviklingen med IQX.	For å begynne å ta i bruk programmet til å gjøre Object Level analyse
Frontend scaling	Use-casen for å analysere fra mobil er kanskje ikke der, men det er ikke mer arbeid å bare designe mobile-first og tilpasse større skjermer	Gjøre nettsiden brukbar på alle skjermstørrelser.
Skrive ut individuelle resultatet på frontend	Koden er ikke perfekt, men må skrives om uansett nå som OS QM Tool også er implementert pga. ulik format på resultat	
Lagd endpoint for OS QM Tool		For å kunne koble den med frontend
Laget endpoint for nedlasting av alle bilder og individuelle bilder	Alle bilder i en session blir nå zippet når analysen er ferdig, slik at den fort kan lastes ned senere. Man kan også bruke	

	en query for å laste ned enkelte bilder fra en session.	
Håndtering av eventuelle like filnavn ved opplasting	Tar å genererer et unikt filnavn for alle filer som lastes opp. Lagrer det originale filnavnet i et map/json	Hvis filnavnene er like, så vil disse overskrive hverandre
Koblet sammen object level frontenden med backenden, og filer i en device level scan blir også lastet opp via frontend nå.	Videre må vi fikse resultat siden til object level. Backenden for object level analyse må også fikses, fordi nå fungerer ingen av object level targets vi har med OS QM Tool.	
Metadata blir nå lagret i hvert bilde i en session.	I device level analyse vil alle bilder ha informasjon om både before target og after target som embedded xmp metadata. I object level analyse vil hvert bilde ha informasjon om det spesifikke bilde som embedded xmp metadata.	Vi følte det ga mening å inkludere informasjonen om begge targets i device level analysen som metadata i hvert bilde.

Hva vi planla, men ikke har gjort:

Hva	Hvorfor	Videre
Skrive på rapport	Ble dårlig med tid pga eksamen, og viderutvikling av produktet ble prioritert.	Vi setter av tirsdagene fremover til rapportskrivning
Object Level Target scanning ved bruk av OS QM Tool	Får error mens vi gjør scanning av TE263. GTDevice/GTObject får exit error 2 selv om den fullfører scanningen.	Finne ut av hvorfor vi får disse errorene. Vi skal ta kontakt med produsenten

Nåværende status og plan fremover:

Hva	Videre

Fått parameterfiler for ISO standard (A, B, C) for OS QM Tool	Bruke disse filene til å lage parameterfiler for targetene
Mye kode som gjøre mye av det samme, f.eks. Håndtering av opplastede filer	Generalisere kode / gjøre koden mer utvidbar
Forbedre hvordan de individuelle resultatene blir vist på frontend	Legge til mulighet for å se resultatene ved siden av hverandre
Kodekvalitet	Gjøre kode så sikker som mulig
Mye av analyse prosessen er ferdig, men mangler fil validering	Fil validerings modulen skal bli integrert i analysen.

Sprint Retrospektiv

Hva gikk bra denne sprinten

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre
Fått implementert alt vi ønsket av funksjonalitet til denne sprinten og ekstra som download endpoint og å legge til metadata i filene	Effektiv bruk av tiden som vi satte av til prosjektet.	Gikk mye fortere å implementere kode for OS QM Tool siden vi har mye erfaring etter å implementert kode for IQX.
Mitigere stress	Eksamen midt i sprinten	

Usikkerheter / Spørsmål

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre

Hva som ikke fungerer så bra

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre

--	--	--

Gruppe status:

- Motivasjonsnivå: Fortsatt høyt
- Arbeidsmengde: God arbeidsfordeling så arbeidsmengden er ok.
- Progresjon: Vi er i rute for å bli ferdig innen tidsfristen (20. April)

Appendix H

Sprint 5 - Review and Retrospective

Sprint 5: Review og Retrospektiv

Goals

- Write proper documentation for the code
 - READMEs for backend and frontend
 - Install
 - Usage
 - Troubleshooting
 - Requirements
 - Info
 - Update backend endpoint documentation
- Stability / Quality development
 - No new big features
 - Improve quality and security of code
 - Improved feedback from running code (logging)
- Write on main report

Sprint review

Hva har vi gjort:

Hva	Notater	Hvorfor
JHOVE Filvalidering av filer før og etter analysen	Kjøre JHOVE på alle filene før og etter modifikasjon	Ønsker å validere filene før og etter vi legger til metadata så vi kan verifisere at vi ikke har gjort filene korrupert
Hente session status og vise dem på frontend	Leser av status.json filen i en session	For å vise progresjonen/status i prosessen
Forbedre analyseringsendpoint og backend kode	Er nå egne funksjoner for device og object analyse. Nye progammer kan legges inn enkelt i switch/case og bestemmes av parameter.	Funksjonene var for avhengige av program, vanskelig å legge til nye program

Logging	Man får nå like og konsekvente tlbakemeldinger når programmet kjører	Det var lite tilbakemelding når programmet kjørte, og de loggmeldingene som var der var ukonsekvente
Unike filnavn	Hvis to filer har like filnavn, så legger vi til et tall på slutten, f.eks. fil_1.jpeg	Vi bruker filnavn til å identifisere filer i programmet vårt, så alle filer må ha et unikt filnavn.
Oversatt frontend til engelsk	Overskrifter, info osv...	Alt i repo skal være på engelsk
API Dokumentasjon / READMEs	La til READMEs for både frontend og backend med generell informasjon om dependencies, installasjon, dokumentasjon og bruk. Trenger fortsatt litt om troubleshooting.	Det var ingen READMEs
Information box på frontend	Gir litt informasjon til brukerne om mulighetene/bruksområdet	Man må ikke laste opp bilder, men man må laste opp targets så vi la til en informasjonsboks om dette.
Refactored frontend kode	Logikk for opplasting av targets på device og object nivå ble flyttet til sine egne components istedenfor i deres parent component.	Mye av frontend logikken var i en parent component når det kunne vært separert.
Valg av software og target på frontend	La til drop down menyer hvor man kan velge hvilke program man kan bruke for analysen.	Siden OS QM-tool kan også gjøre analyse på device nivå ville vi legge til funksjonalitet for dette i frontenden.
Rapportskriving	Kapittel 1, 2 og 3 er for det meste ferdig. Begynt på 4 og 5.	

Hva vi planla, men ikke har gjort:

Hva	Hvorfor	Videre
Brukermanual + installasjons manual		Lage brukermanual + installasjons manual

Nåværende status og plan fremover:

Hva	Videre
Skrevet litt på rapport, ferdig med kapittel 1, 2 og 3 skrevet endel på 4 og 5	Mer rapport skriving
Dokumentasjon	Brukermanual + installasjonsmanual
Auto fetch av resultater	Implementere en form for auto fetch / loading page (mulig skjule resultater frem til status = finished / err)
Frontend forbedringer	Når du trykker submit, blurre background og ha loading animasjonen midt på skjermen

Sprint Retrospektiv

Hva gikk bra denne sprinten

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre
Jobbet videre selv om det har vært sykdom		Det går sakte
Vi har vært veldig produktive denne sprinten, og vi har kommet i mål med all funksjonaliteten.	God arbeidsfordeling, og bra definerte arbeidsoppgaver slik at alle visste hva de skulle gjøre.	Prøve å fordele arbeidsoppgaver godt når det gjelder rapportskrivning også. Lese over det de andre skriver.

Usikkerheter / Spørsmål

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre

Hva som ikke fungerer så bra

Hva	Hvorfor	Hva lærte vi av dette / Grep vi må ta videre

Gruppe status: 200

Appendix I

Backend Documentation

Welcome

This is the documentation for the API endpoints. Check out the left sidebar for all the endpoints

/api/analyze/device

Endpoint for performing an analysis on two targets. Targets are referred to as *before target* and *after target*, as that is the usecase the endpoint is modeled after.

Request

Method: POST

Query Parameters

```
?iqes={:iqes_name}
```

```
?target={:target_name}
```

Supported Parameter Values

Supported iqes values:

- OQT
- IQX

Supported target values for OQT:

- GTDevice
- UTT

Supported target values for IQX:

- UTT

Response

200 OK

```
1      {
2      "session_id": STRING
3      }
```


400 Bad Request

```
1    {  
2    "error": STRING  
3    }
```

Sample Response

200 OK

```
1    {  
2    "session_id": "f811367b-6646-4df0-9303-ad850bb36c54"  
3    }
```

400 Bad Request

```
1    {  
2    "error": "before_target not specified"  
3    }
```

/api/analyze/object

Endpoint for performing analysis on all images uploaded.

Request

Method: POST

Query Parameters

```
?iqes={:iqes_name}
```

```
?target={:target_name}
```

Supported Parameter Values

Supported iqes values:

- OQT

Supported target values for OQT:

- GTOBJECT
- TE263

Request

POST

Multipart form

Fields:

```
files: [FILE, ...]
```

Response

200 OK

```
1      {  
2      "session_id": STRING  
3      }
```

400 Bad Request

```
1      {  
2      "error": STRING  
3      }
```

Sample Response

200 OK

```
1      {  
2      "session_id": "f811367b-6646-4df0-9303-ad850bb36c54"  
3      }
```

400 Bad Request

```
1      {  
2      "error": "missing file(s). Check request"  
3      }
```

/api/download/{:session_id}

Endpoint for downloading a single image or a ZIP with all images in a session.

Request

Method: GET

Optional Query Parameters

?file_name={:file_name}

Response

200 OK

FILE

404 Not Found

```
1    {  
2      "error": STRING  
3    }
```

Sample Response

200 OK

FILE

404 Not Found

```
1    {  
2      "error": "session does not exist"  
3    }
```


/api/results/{:session_id}

Endpoint that gives complete results of current results of a session.

Request

Method: GET

Response

200 OK

```
1      {
2          STRING: {
3              CHAR: {
4                  "completed": BOOL,
5                  "results": {
6                      "passed": BOOL,
7                      STRING: {
8                          ...
9                      }
10                     ...
11                 }
12             },
13             "image_tag": STRING,
14             "overall_score": CHAR
15         },
16         ...
17     }
```

404 Not Found

```
1      {
2          "error": STRING
3      }
```

Sample Response

200 OK

```
1      {
```



```

59     },
60     "Resolution": {
61         "MTF10": "succeeded",
62         "MTF50": "succeeded",
63         "Max_Misregistration": "succeeded",
64         "Max_Modulation": "succeeded",
65         "Min_Sampling_Efficiency": "succeeded",
66         "Obt_Sampling_Rate": "succeeded"
67     },
68     "Color": {
69         "Max_Delta_C": "succeeded",
70         "Max_Delta_E": "succeeded",
71         "Max_Delta_H": "succeeded",
72         "Max_Delta_L": "succeeded",
73         "Mean_Delta_C": "succeeded",
74         "Mean_Delta_E": "succeeded",
75         "Mean_Delta_H": "succeeded",
76         "Mean_Delta_L": "succeeded"
77     },
78     "Shading": {
79         "Max_Delta_Gray": "succeeded",
80         "Max_Delta_White": "succeeded"
81     },
82     "Distortion": {
83         "Max_Distortion": "succeeded"
84     },
85     "Lines": {
86         "Max_Out_Black": "succeeded",
87         "Max_Out_Gray": "succeeded",
88         "Max_Out_White": "succeeded"
89     },
90     "Noise": {
91         "Visual_Noise": "succeeded"
92     }
93 }
94 },
95 "A": {
96     "completed": true,
97     "results": {
98         "passed": true,
99         "TonalReproduction": {
100             "Delta_C": "succeeded",
101             "Delta_E": "succeeded",
102             "Lab_L": "succeeded",
103             "Max_Gain_Modulation_L": "succeeded",
104             "Min_Gain_Modulation_L": "succeeded"
105         },
106         "Resolution": {
107             "MTF10": "succeeded",
108             "MTF50": "succeeded",
109             "Max_Misregistration": "succeeded",
110             "Max_Modulation": "succeeded",
111             "Min_Sampling_Efficiency": "succeeded",
112             "Obt_Sampling_Rate": "succeeded"
113         },
114         "Color": {
115             "Max_Delta_C": "succeeded",

```



```

116         "Max_Delta_E": "succeeded",
117         "Max_Delta_H": "succeeded",
118         "Max_Delta_L": "succeeded",
119         "Mean_Delta_C": "succeeded",
120         "Mean_Delta_E": "succeeded",
121         "Mean_Delta_H": "succeeded",
122         "Mean_Delta_L": "succeeded"
123     },
124     "Shading": {
125         "Max_Delta_Gray": "succeeded",
126         "Max_Delta_White": "succeeded"
127     },
128     "Distortion": {
129         "Max_Distortion": "succeeded"
130     },
131     "Lines": {
132         "Max_Out_Black": "succeeded",
133         "Max_Out_Gray": "succeeded",
134         "Max_Out_White": "succeeded"
135     },
136     "Noise": {
137         "Visual_Noise": "succeeded"
138     }
139 }
140 },
141 "image_tag": "before_target",
142 "overall_score": "A"
143 },
144 "e86a-UTT.tif": {
145     "C": {
146         "completed": true,
147         "results": {
148             "passed": true,
149             "TonalReproduction": {
150                 "Delta_C": "succeeded",
151                 "Delta_E": "succeeded",
152                 "Lab_L": "succeeded",
153                 "Max_Gain_Modulation_L": "succeeded",
154                 "Min_Gain_Modulation_L": "succeeded"
155             },
156             "Resolution": {
157                 "MTF10": "succeeded",
158                 "MTF50": "succeeded",
159                 "Max_Misregistration": "succeeded",
160                 "Max_Modulation": "succeeded",
161                 "Min_Sampling_Efficiency": "succeeded",
162                 "Obt_Sampling_Rate": "succeeded"
163             },
164             "Color": {
165                 "Max_Delta_C": "succeeded",
166                 "Max_Delta_E": "succeeded",
167                 "Max_Delta_H": "succeeded",
168                 "Max_Delta_L": "succeeded",
169                 "Mean_Delta_C": "succeeded",
170                 "Mean_Delta_E": "succeeded",
171                 "Mean_Delta_H": "succeeded",
172                 "Mean_Delta_L": "succeeded"

```

```

173     },
174     "Shading": {
175         "Max_Delta_Gray": "succeeded",
176         "Max_Delta_White": "succeeded"
177     },
178     "Distortion": {
179         "Max_Distortion": "succeeded"
180     },
181     "Lines": {
182         "Max_Out_Black": "succeeded",
183         "Max_Out_Gray": "succeeded",
184         "Max_Out_White": "succeeded"
185     },
186     "Noise": {
187         "Visual_Noise": "succeeded"
188     }
189 }
190 },
191 "B": {
192     "completed": true,
193     "results": {
194         "passed": true,
195         "TonalReproduction": {
196             "Delta_C": "succeeded",
197             "Delta_E": "succeeded",
198             "Lab_L": "succeeded",
199             "Max_Gain_Modulation_L": "succeeded",
200             "Min_Gain_Modulation_L": "succeeded"
201         },
202         "Resolution": {
203             "MTF10": "succeeded",
204             "MTF50": "succeeded",
205             "Max_Misregistration": "succeeded",
206             "Max_Modulation": "succeeded",
207             "Min_Sampling_Efficiency": "succeeded",
208             "Obt_Sampling_Rate": "succeeded"
209         },
210         "Color": {
211             "Max_Delta_C": "succeeded",
212             "Max_Delta_E": "succeeded",
213             "Max_Delta_H": "succeeded",
214             "Max_Delta_L": "succeeded",
215             "Mean_Delta_C": "succeeded",
216             "Mean_Delta_E": "succeeded",
217             "Mean_Delta_H": "succeeded",
218             "Mean_Delta_L": "succeeded"
219         },
220         "Shading": {
221             "Max_Delta_Gray": "succeeded",
222             "Max_Delta_White": "succeeded"
223         },
224         "Distortion": {
225             "Max_Distortion": "succeeded"
226         },
227         "Lines": {
228             "Max_Out_Black": "succeeded",
229             "Max_Out_Gray": "succeeded",

```

```

230         "Max_Out_White": "succeeded"
231     },
232     "Noise": {
233         "Visual_Noise": "succeeded"
234     }
235 },
236 },
237 "A": {
238     "completed": true,
239     "results": {
240         "passed": true,
241         "TonalReproduction": {
242             "Delta_C": "succeeded",
243             "Delta_E": "succeeded",
244             "Lab_L": "succeeded",
245             "Max_Gain_Modulation_L": "succeeded",
246             "Min_Gain_Modulation_L": "succeeded"
247         },
248         "Resolution": {
249             "MTF10": "succeeded",
250             "MTF50": "succeeded",
251             "Max_Misregistration": "succeeded",
252             "Max_Modulation": "succeeded",
253             "Min_Sampling_Efficiency": "succeeded",
254             "Obt_Sampling_Rate": "succeeded"
255         },
256         "Color": {
257             "Max_Delta_C": "succeeded",
258             "Max_Delta_E": "succeeded",
259             "Max_Delta_H": "succeeded",
260             "Max_Delta_L": "succeeded",
261             "Mean_Delta_C": "succeeded",
262             "Mean_Delta_E": "succeeded",
263             "Mean_Delta_H": "succeeded",
264             "Mean_Delta_L": "succeeded"
265         },
266         "Shading": {
267             "Max_Delta_Gray": "succeeded",
268             "Max_Delta_White": "succeeded"
269         },
270         "Distortion": {
271             "Max_Distortion": "succeeded"
272         },
273         "Lines": {
274             "Max_Out_Black": "succeeded",
275             "Max_Out_Gray": "succeeded",
276             "Max_Out_White": "succeeded"
277         },
278         "Noise": {
279             "Visual_Noise": "succeeded"
280         }
281     }
282 },
283 "image_tag": "after_target",
284 "overall_score": "A"
285 }
286 }

```

404 Not found

```
1  {  
2    "error": "session does not exist"  
3  }
```

Appendix J

Wireframe of the frontend

Kvalitetssikring

Object level target



Device level target



Device level target

Velg start-target

Velg slutt-target

Velg bilder



Lorem ipsum dolor sit amet, consectetur adipiscing elit. Donec finibus neque id nunc rhoncus, nec commodo leo aliquam. Proin ac lacinia justo, et tincidunt nisi.

Avbryt

Start

Resultater

Sorter: lav til høy kvalitet

Bilde 1.jpg

Nivå C

Bilde 2.jpg

Nivå C

Bilde 3.jpg

Nivå B

Bilde 4.jpg

Nivå A

Bilde 5.jpg

Dårlig

Object level target

Velg bilder

Avbryt

Start

Appendix K

Meeting Notices and Minutes with Client

Meeting Minute

Møte med oppdragsgiver

Time/Location: 14-01-2022 13:00 - Teams (**Impulsivt møte, så ikke noen innkalling**)

Present: Martin Wighus Holtmon, Jakob Karlsmoen, Mikael Falkenberg Krog, Ottar André Breivik Anderson

Absent:

Moderator: Ottar André Breivik Anderson

Case no. 1/2022 - Intro

Send mail til Ottar med rådgivere.

Få kontroll på datoer - innlevering + presentasjon

Mulighet for videre lønnet arbeid for å ferdigstille prosjektet i senere tid.

Løsningen kan være en **API** som kan integreres inn i arkivverket.

Prosjektbeskrivelse - åpen for endringer.

Vi ønsker tilgang til datasett - inkluderer ground truth.

Case no. 2/2022 - Kontrakter

- Trenger ikke konfidensialitetsavtale på *løsningen* - mer taushetsbelagt avtale på *materiale* vi får tildelt. **Konsekvent holde taushetsbelagt materiale som vedlegg i rapport.**
- Litt usikker på opphavsrett/bruksrett/eiendomsrett - Trenger vel opphavsrett for å vise det frem i en akademisk sammenheng (ntnuopen)? Både vi og arkivverket ønsker at publikasjonen skal være offentlig.
 - Så lenge arkivverket har bruksrett burde det holde, vi beholder eiendomsretten - må avklares.
 - Eventuelle unntak?
 - Arkivverket har lyst til å vidreutvikle løsningen - kanskje lurt å fraskrive seg rettigheten for at det kan bli holdt oppdatert? Eventuelt gjøre det etter oppgaven.
 - Hva er blitt gjort tidligere (eksempler)? **Spør veileder**
- Usikkerhet rundt *Overføring av eiendomsrett* - **diskuter med veileder.** Tanken er at løsningen skal være åpen og brukbart.
- Patent? Ser ikke helt hvordan det er relevant
- Ta i bruk oppdragsgiver til støtte under rapportskrivning
- Mulig vi trenger lisenser - dekkes av eksterne virksomheten.
 - Hvordan er det med lisenser til prgrammer mpt. at det er i en akademisk sammenheng, men produktet kan bli brukt i en kommersiell sammenheng?
- Dekking av eventuelle reiser for studenter? Noe system? **Spør veileder**

Taushetsskjema fra IKAMR:

- ikke noen tidsramme på taushetsplikt
- Materialet vi får tildelt er taushetsbelagt

Case no. 2/2022 - Møteplan

Ukentlige møter - Mandager 13:00 frem til 25. April

Mulighet for å møtes fysisk for å låne en maskin (Komme ned til Gjøvik i et par dager).

- Vi er opptatt tirsdag-onsdag fra uke 4 til 11 med annet fag.

Sette opp møte med oppdragsgiver og veileder på tirsdag 18. Januar.

14.01.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 24-01-2022 15:00 - Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

- sak no. 01/2022** Funksjonelle krav og ikke funksjonelle krav
- sak no. 02/2022** Programmering kontaktperson
- sak no. 03/2022** Standardavtalen

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 21.01.2022

Møtereferat

Tid/Sted: 24-01-2022 15:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Funksjonelle krav og ikke funksjonelle krav

IIIF (alternativ)

JHOVE

Ferdige programmer for kvalitetsnivå analyse - resultatet er sorteringen skal brukes

- Sammenlikne programmer
- Helst ISO Standard
- Si ifra hvis filer ikke går igjennom / ikke består kvalitetsanalysen

Resultatet + Infoen av analysen skal inn i filene som XMP metadata

Sak no. 2/2022 - Programmering kontaktperson

Få tilgang til digitalarkivet

- Sette opp møte med kontaktperson når Ottar får svar.

Sak no. 2/2022 - Standardavtalen

Signering av standardavtalen blir ordnet til Onsdag 26. Januar 2022.

24.01.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 31.01-2022 13:00 - Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 Se kort over prosjektplan
- Kart

sak no. 02/2022 Gå igjennom sprintplanlegging + mål

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 28.01.2022

Møtereftrat

Tid/Sted: 31-01-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson
Martin Wighus Holtmon
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Prosjektplan

Organisasjonskart

- Legge til arkivverket

Ottar vil se over prosjektplan. Vil gi et kort svar i kveld. 31.01.2022

Ottar sender over into ISO standard introduksjon.

Sak no. 2/2022 - Sprint plan

Programmer:

- Golden thread - cli
- ImageZebra
- Har "troa" på iQ-Analyzer-X

Sak no. 3/2022 - Ekstra

16-18 Feb - leie grupperom når Ottar kommer på besøk

31.01.2022, Martin Wighus Holtmon

Møtereferat

Tid/Sted: 08-02-2022 14:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022

Hvis vi bruker API (tredje-parts ekstern service), kanskje en skal bare laste opp Target (device level) og ikke object level med selve bildet som kan være taushetsbelagt.

In house kan bruke alt.

Få datasettet kan vi bruke på alle løsningen vi tester ut.

Ottar og dem eier/har lisens til GoldenThread, men er låst til en pc. Ordner plan videre.

08.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 14-02-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

- sak no. 01/2022** Kontaktperson til Arkivverket
- sak no. 02/2022** Datasett
 - Bildemål på objektnivå
 - Analyse av et bilde med GoldenThread
- sak no. 03/2022** Begynne utvikling rundt et av programvarene

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 11.02.2022

Møtereferat

Tid/Sted: 14-02-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Kontaktperson til Arkivverket

Tar tid

Sak no. 2/2022 - Datasett

Image Target - Referanse Kort (evt. fargekart) på norsk (usikker) (butikk.foto.no)

Skal sende over et eksempel på object level

Sak no. 3/2022 - Begynne utvikling rundt et av programvarene

Sende mail til produsent/utvikler av de forskjellige programvarene: Nevne at vi tester ut dems programvare mot de andre

14.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 28-02-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 Demonstrere frontend

sak no. 02/2022 Planlegge neste workshop

sak no. 03/2022 ImageZebra

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 25.02.2022

Møtereftrat

Tid/Sted: 28-02-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Demonstrere frontend

Ser OK ut.

Object - FLERE ulike (amerikanske og europeiske)

- Velge hvilket object level

Sak no. 2/2022 - Planlegge neste workshop

Dato: 24-25.03.2022

Sak no. 3/2022 - ImageZebra

Ikke noe svar enda. Venter litt lengre før vi purrer.

Ekstra

Arkivarforeningens vårseminar. 27-28.05.2022

Laste opp flere bilder (zip).

28.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 07-03-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 OS QM Tool Parameter file (ISO Standard)

sak no. 02/2022 ImageZebra

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 07.03.2022

Møtereferat

Tid/Sted: 07-03-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - OS QM Tool Parameter File (ISO Standard)

Ottar sender purring på parameter filen

Sak no. 2/2022 - ImageZebra

Ottar tar kontakt med dem

Ekstra

Presentere forskjellene mellom en god/dårlig scan

(<https://image-compare-viewer.netlify.app/>)

Nettside

Ottar sender 6 bilder + evt. tekst

Bestille grupperom til workshop session (24-25 Mar)

07.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 14-03-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 Demo av det vi har til nå

sak no. 02/2022 OS QM Tool (A, B, C)

sak no. 03/2022 Lav prioritet: ImageZebra

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 11.03.2022

Møtereferat

Tid/Sted: 14-03-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Demo

nice

Sak no. 2/2022 - OS QM Tool (A, B, C)

Ikke har svart

Sak no. 3/2022 - Lav prioritet: ImageZebra

ottar spør igjen

laster opp utt target

- også på objekt nivå (lite brukt)
- A3

14.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 21-03-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

- sak no. 01/2022** Noen spørsmål angående Device Level Target:
1. Hva slags metadata skal lagres, before eller after target. Er fullt mulig at disse gir forskjellig resultat
 2. Ønskes det å laste den targets eller kun bilder med resultatene integrert som XMP metadata?

Hva er kompetansenivået til brukeren? Må vi forklare noe når vi viser resultatene?

- sak no. 02/2022** Kan vi få et scanning av UTT device level target (TE262). Den vi har nå består ingen av ISO nivåene

- sak no. 03/2022** Parameter filer (ISO standard) for OS QM Tool

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 18.03.2022

Møtereferat

Tid/Sted: 21-03-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Device level target + kompetansenivået

Kan godt ta med before og after target

Sak no. 2/2022 - UTT Target

Skal få tilgang etter møtet, blir lastet opp 2 stk.

Sak no. 3/2022 - OS QM TOOL

Har ikke fått svar enda - Ottar skal sende purring

Kan møte oss på torsdag ettermiddag.

Viser en demo av OS QM Tool på fredag.

21.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 28-03-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 Manual OS QM Tool

Gi beskjed om du ikke kan delta i møtet.

Hilsen,

Martin Wighus Holtmon

NTNU 25.03.2022

Møtereftrat

Tid/Sted: 29-03-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022

Gjøre 3 målinger med de 3 targetene som er lastet opp i hightail med bruk av te263

Digitalt webinar hvor ottar og dem får prøvd ut programvaren (slutten av April)

Når vi skriver XMP til filene

- Trenger revisjon funksjon ??
 - Lese XMP og skille A , B og C.

-

Sak no. 2/2022

...

29.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 04-04-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

- sak no. 01/2022** Behandle dupliserte filer
- sak no. 02/2022** Planlegging av innlegg på Arkivarforeningens vårseminar
- sak no. 03/2022** Planlegg gjennomgang av bruk av OS-QM Tool og IQ Analyzer
- sak no. 04/2022** Arbeidsmøte 19. april, innlegget 28. april

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 03.04.2022

Møtereferat

Tid/Sted: 05-04-2022 13:00 - Teams (**Ble flyttet pga. Ottar ikke kunne møte til 04-04**)

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Behandle dupliserte filer

Metoden vi håndterer dette på er OK for dem

Sak no. 2/2022 - Planlegging av innlegg på Arkivarforeningens vårseminar

Ta opptak av prosessen (demo av modulen)

Sak no. 3/2022 - Planlegg gjennomgang av bruk av OS-QM Tool og IQ Analyser

Ottar vi ha gjennomgang av OS QM Tool + IQ Analyser X

- Eventuelt team viewer (Helst) eller annen VNC tilgang

Sak no. 4/2022 - Arbeidsmøte 19. april, innlegget 28. april

Blir diskutert senere

04.04.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 11-04-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 Demo

sak no. 02/2022 Tilgang til verktøyene (ta dem med til Ålesund)

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 08.04.2022

Møtereftrat

Tid/Sted: 11-04-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Demo

Opptak klar til 25. April

Sak no. 2/2022 - Ta med nøkler til Ålesund

ok - Ottar tar det med ned igjen 19. April

Ekstra

Brukertesting:

- Bergen byarkiv
- Folk hos Ottar

11.04.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 25-04-2022 13:00 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022

Spørsmål:

1. Hva vil du kalle oppgavens fagfelt?
2. Er Digitalarkivet "The Digital Archive" eller "The Digital Archives" på engelsk?

sak no. 02/2022

Deres nåværende digitiserings prosess (Ref. tidligere mail)

sak no. 03/2022

Arkivarforeningens vårseminar plan

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 22.04.2022

Møtereftrat

Tid/Sted: 25-04-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Spørsmål

Fagfelt: Quality management (evt. Cultural Heritage)

IKAMR Sektor: ABM - arkiv, bibliotek og museum (Engelsk: GLAM - galleries, libraries, archives, and museums) - kulturvårnsektor er inkludert i ABM

- Offentlig / kommunalt

Digitalarkivet på engelsk: The Digital Archives

Sak no. 2/2022 - Nåværende digitiseringsprosess

De bruker også NXT, results.

Verifisert utkastet vårt der vi beskriver prosessen.

Sak no. 2/2022 - Arkivarforeningen

Jakob presenterer litt av det vi har gjort.

25.04.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 06-05-2022 13:30 -Teams

Møteinnkalling er sendt til:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Agenda

sak no. 01/2022 Lisens for prosjektet ([MIT License](#))

sak no. 02/2022 Plan for resten av perioden.

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 05.05.2022

Møtereferat

Tid/Sted: 10-05-2022 14:00 - Teams

Notis: Dag og tid ble endret da Ottar ikke kunne delta på tidligere avtalt tid..

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Lisens

MIT så OK ut

Sak no. 2/2022 - Plan for resten av perioden

Ottar sender over bakgrunn til fagfeltet med før-rapport fra Arkivverket.

Ekstra

Mulig vi kan få et "Letter of recommendation" fra Zeutchel

10.05.2022, Martin Wighus Holtmon

Møterefereat

Tid/Sted: 18-05-2022 13:00 - Teams

Til stede:

Ottar André Breivik Anderson

Martin Wighus Holtmon

Jakob Frantzvåg Karlsmoen

Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Fiksing på rapport

Illustrasjon i bakgrunn- Device level target. Bytte fra FADGI til UTT (ligger på hightail)

Prosessen i arkivverket:

- gjøre det tydeligere, FADGI three stars or higher
- "complications" til "time limitation"
- Object level target - Bytte det store bildet med et mindre ett. - Ottar laster opp til Hightail
- IMKAR vil ta over og kjøre stresstesting selv.
- Zeutchel har oppdatert nettsidene sine. Lurt å se over dem på nytt og oppdatere datoene på kildene.
- Limitation, referanse data (må uansett oppdateres hvert år)

Resten ble fikset på "live"

Sak no. 2/2022

Blir mest sansynelig skrevet en artikkel om prosjektet. En redaktør har tatt kontakt med Ottar. Vi kan svare på spørsmål. Frist 8. August.

18.05.2022, Martin Wighus Holtmon

Appendix L

Meeting Notices and Minutes With Supervisors

Meeting Summon

Initial meeting with the supervisors

Time/Location: 13-01-2022 12:00 - Teams

The following persons are called for:

Rune Hjelsvold

Martin Wighus Holtmon

Johanna Johansen

Jakob Karlsmoen

Mikael Falkenberg Krog

Agenda

- case no. 01/2022 Introduction / Say hello
- case no. 02/2022 Additional documentation outside of the standard agreement for academic collaboration that we must sort out?
- case no. 03/2022 Set-up weekly meetings
- case no. 04/2022 Rapport/project plan: Written in Norwegian or English?

Please contact me if you are unable to attend the meeting.

Regards,
Martin Wighus Holtmon

NTNU 12.01.2022

Meeting Minute

Initial meeting with the supervisors

Time/Location: 13-01-2022 12:00 - Teams

Present: Rune Hjelsvold, Martin Wighus Holtmon, Johanna Johansen, Jakob Karlsmoen, Mikael Falkenberg Krog

Absent:

Moderator: Jakob Karlsmoen

Case no. 1/2022 - Introduction

Tips:

- Faste rutiner
- Sette klare **mål**
- Ha god kommunikasjon
- Prate med oppdragsgiver:
 - Hva skal vi gjøre for at løsningen faktisk blir tatt i bruk?
 - Jo mer vi vet om forventninger jo bedre
- Hva som skiller en A, B og C.
 - Selvstendighet (friere valg, mindre direksjon)
 - Analytisk evne - hvorfor, utfordringer, alternative metoder (svakheter osv)
 - Diskutere og reflektere - hvorfor er den god? Hvorfor er den bedre enn noe annet?
 - A: Bedre enn forventet - finne med problemstillinger som er krevende (demonstrer hvor flinke vi er)

Case no. 2/2022 - Additional documentation

Avklare rettigheter. Eierrettigheter til det som lages (eventuelt IPR avtale). Ønsker å ha det åpen for muligheter til å vise det i en akademisk sammenheng (ntnuopen).

Case no. 3/2022 - weekly meetings

Weekly meetings: Vi bestemmer agenda, **dagen før møte sende (status)melding** om

1. Hva har vi gjort
2. Noe spesielt vi ønsker å diskutere
3. Hva vi tenker å gjøre frem til neste møte

Hvorfor:

Selvstendigheter

Enklere å kommentere på prosessen

Bedre kvalitet på tilbakemeldinger

14:00-15:00 Hver mandag - sette opp teams møte.

Eksakte når vi spør om hjelp med tekst (del overleaf - skriv kommentarer rett i prosjektet). **Hva er verdifult for oss?** (Små inkremitter/deler om gangen).

Case no. 4/2022 - Rapport/project plan language

Skriv i det språket som vi fungerer best for oss mtp. å uttrykke seg.

Lag heller et sammendrag på engelsk hvis det er ønskelig.

13.01.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 24-01-2022 14:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Jobbet med prosjektplanen

Agenda

sak no. 01/2022 Godkjenne grupperegler (3.3 Regler og rutiner)

Plan for arbeid frem til neste møte

- Få standardavtalen ferdig
- Sette opp Scrum board og sette inn milestones
- Skrive ferdig prosjektplan

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 23.01.2022

Møtereftrat

Tid/Sted: 24-01-2022 14:00 - Teams

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog.

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Grupperegler

Proessen hvis noen ikke greier å levere (sette opp mål og trekke seg hvis dem ikke når dem). Si opp i steden for å bli "sparket".

Bør ha med:

- Hva hvis det er **uenighet**? Hva er prosessen (flertall bestemmer eller få hjelp fra noen eksterne)?
-

Hvordan skal vi dokumentere arbeidsprosessen og progresjon?

- Skrive rapport i begynnelsen av **hver** sprint
 - Skjermlog av sprint backlog + produkt backlog
 - Få med milepæler - hvor ligger vi i forhold til milpæler
 - Skrive rapport etter vi har hatt retrospektiv møte om hva som ble gjort osv. Brukes til å finne ut hva som ikke fungerer, hvem som har bidratt/ikke bidratt osv.
- **Få med det viktigste, ikke noe lang rapport.**
 - Dette var det vi planla
 - Dette observerte vi (hva gikk bra, hva gikk dårlig)
 - Dette gjorde vi

Sak no. 2/2022 - Ekstra

Bruke veileder mer aktivt (se på tekst)

Prosjektplan

- Hjelpe hovedprosjektet
- **Viktig informasjon kommer tidlig**, hvor lang er sprinten, hvor mange sprinter burde skrives ned tidlig. Gi leseren det leseren forventer. Gjør det lett for leseren å få tak i informasjonen.
- Gir en god historiefortelling.
- Forklare detaljer for andre, ikke for oss.

Gå litt dypere inn i kvalitetskriteriene, begynne å grave tidlig.

- Hva skal det tilfredsstillende
- Hva er tilgjengelig

- **Forklare litt bedre i prosjektplanen hva dette er.** Gjør det lett for leser å få med hovedpoengene, skal være lett å tolke. Gi et overordnet beskrivelse. **Oppsummering av kvalitetssjekking.** Forklare målestokken til kvalitetssjekking. Konkrete eksempler av bilder + kvalitetsvurderinger.

Skrive på rapport underveis

- Valg vi gjør, hva de forskjellige tingene er gode dårlig på osv
- Definere hvem som er leseren og si dette i rapporten. Hvem skriver vi for? Hvor mye teknisk informasjon skal vi gå inn i?

Bøker

The craft of research - Hvordan bruke kilder, hvordan ta notater, hvordan argumentere, hvordan ta valg osv.

Skriveboka - Merete M. Andersen

24.01.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 31-01-2022 14:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Ferdigstilt prosjektplan
- Skrevet under Taushetsavtale og standardavtale
- Begynt å se på forskjellige kvalitetsanalyse programmer
- Startet med å sette opp Scrum Board

Agenda

sak no. 01/2022 Flere kommentarer til prosjektplan?

sak no. 02/2022 Se over sprint plan

Plan for arbeid frem til neste møte

- Gjøre oppgaver mot sprintmål

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 28.01.2022

Møtereftrat

Tid/Sted: 31-01-2022 14:00 - Teams

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Prosjektplan

Definere MVP - minimum viable product (hva vi prioriterer)

- utelater brukergrensesnitt
- Følger standard
-

Hva er spennende med prosjektet?

- Argumentere for at hvorfor dette kan bli et A prosjekt

Hvorfor valgt de programmene vi har valgt. (Etter prosjektplan)

Forklare tidlig forskjellen på digitalisering og digitisering, hvorfor vi bruker digitisering. Forklare både i rapport og prosjektplan.

Sak no. 2/2022 - Sprint Plan

Fullstendig beskrivelse om hva som er MVP og få det inn i prosjekt backlog.

Får noe funksjonalitet - få testet integrasjon. Integrere mer og mer etterhvert, ofte der problemer oppstår. Få de viktigste "småbitene" til å spille sammen.

Hvordan ta gode notater - dokumentere hva vi leser teknisk. Være god på dokumentasjon - vi følger jo en ISO standard! Beskriv standarden, forklare hvordan vi følger standarden.

Må forstå prosessen i "black box" og vise forståelse om den. Burde starte å koble sammen noen "biter". Testing av programvare til vårt bruk med integrasjon. Se å andre rapporter / eksempler fra tidligere år.

31.01.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 07-02-2022 14:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Prøvd ut IQ-Analyser
- Filformat validering med JHOVE (Lagd script)
- Sett på metadata + modifisering (XMP)
- Sett over ImageZebra, men fikk ikke lagd bruker
- Testet ut Delt.ae
- Fått oppklaring i hvordan vi gjør analyse av scan/systemet (Object/device level target)

Agenda

sak no. 01/2022 Gi en statusrapport

Plan for arbeid frem til neste møte

- Gjøre mer testing av programvare med datasett
 - Se om vi får automatisert analysen
- Prøve å få tilgang til GoldenThread
- Fullføre sprint 1

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 04.02.2022

Møtereftrat

Tid/Sted: 07-02-2022 14:00 - Teams

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Statusrapport

...

Sak no. 2/2022 - Ekstra

Er det OK å laste opp disse bildene til en tredjeparts tjeneste som ImageZebra / delt.ae

- **Lage sekvensdiagram / flow diagram**
- Les igjennom Automatisk rapportskrivning og testing av digitale arkiv
- Se hvor lenge dem har vært ute? Hva vil skje om dem slutter å utvikle/funke.
 - Vurdere hvor solide leverendørene er.
- God retrospektiv med ønskelige justeringer frem mot neste sprint.

07.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 14-02-2022 14:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Begynt på hovedrapporten
 - Beskrevet ISO 19264-1:2021 standarden
 - Startet på brukermønster
 - Image level target
- Evaluert noen flere filvalideringsprogrammer
- Sett på OpenDICE
- Gjennomført sprint review og retrospektiv

Agenda

- sak no. 01/2022** Sprint 1 review + retrospektiv (*vedlegg*)
- sak no. 02/2022** Sprint 2 plan
- sak no. 03/2022** Fotnote vs. sluttnote. Når skal vi bruke hva?

Plan for arbeid frem til neste møte

- Jobbe med Ottar og få tilgang/prøve ut GoldenThread
- Starte å utvikle
- Lage wireframes til modulen
- Fortsette på rapporten

Vedlegg

- Sprint 1: Review og Retrospektiv

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 11.02.2022

Møtereferat

Tid/Sted: 14-02-2022 14:00 - Teams

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Sprint 1 review + retrospektiv

Fysiske møter er ok for veiledere

Viktig å vite hvorfor et program ikke fungerer før en eventuelt velger å ikke jobbe videre med det.

<https://github.com/Chlara> - sende dem mail for å prøve å få programmet til å funke.

Viktig å nevne i sprint review/retrospektiv

Hva skjedde

Hvorfor skjedde / hvorfor er dette et problem?

(Hvordan) **Hva** kan vi gjøre videre / **Hva** lærte vi av dette / **Grep** vi må ta

Gå hakket dypere!!!!

Definere prosessen

Må forklare hvorfor

Ta hensyn så det er relativt enkelt å bytte ut "bibliotek"/program

Se på grupperegler/prosessen - fulgte vi planne? Noe vi burde endre?

Retrospektiv:

Actionplan - Hva gikk bra / dårlig

Hvordan kan vi bruke mer av den oppsatte arbeidstiden

Sak no. 2/2022 - Sprint 2 plan

<https://www.ntnu.no/sekom/stadier-i-en-skriveprosess>

- Hvordan jobbe med oppgaven - **se over før neste sprint!**
- Gjøre hele tiden
- Kreative delen av skrivingen
- Skrive notater, lage tankekart

Ha en skrivegruppe (annen gruppe) for å dele tekstene med hverandre for å få kommentarer.

Sak no. 3/2022 - Fotnote vs. sluttnote

Hvis vi bruker kilder som ikke kommer til å være der i mange år, kan være i fotnotater.

Det som vi bygger teksten på (referanseliste) - grunnlaget (oversikt ov hva vi har basert oss på). Tilgjengelig litteratur osv.

Fotnote: Foklare verktøy (noe brukeren egentlig ikke trenger å vite om) som ikke er grunnlaget for teksten. Ikke bruk fotnoter til forklaringer, ha det heller i teksten.

14.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 21-02-2022 14:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Ottar har vært på besøk
 - Mye diskutering rundt programvarene - kartlegging av funksjonalitet, fordeler og ulemper.
 - Skrevet mail til produsenter av programvare for bildeanalyse. Har nesten fått lisens/tilgang fra to stk. Vi får nok tilgang veldig fort til en av dem (ImageZebra).
- Frisket opp python kunnskapene
- Laget fil validerings modulen og laget tester til modulen
- Sett opp testing & linting jobs til CI/CI pipelines

Agenda

sak no. 01/2022 Avtale fysiske møter

Plan for arbeid frem til neste møte

- Begynne utvikling av bildeanalysemodulen rundt ImageZebra og OS QM-Tool hvis vi får tilgang til den i løpet av uken.

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 18.02.2022

Møtereftrat

Tid/Sted: 21-02-2022 14:00 - Teams

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Avtale fysiske møter

Ta møtet på kontoret (**A215**)

- 28. Man digitalt

Ekstra

Trekke frem code review med konkrete eksempler på hva dette førte til (forbedringer)

Vise demoer, gjerne ved hver sprint. Også til Ottar.

Passe på at Ottar ikke presser på mer funksjonalitet en det som er bestemt.

21.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 28-02-2022 14:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Satt opp enkel API for å sende inn et bilde og verifisere filen med filvalideringsmodulen
- Figma prototype til frontend. Begynte å utvikle frontend i react.

Agenda

- sak no. 01/2022** Demonstrere API & Figma (+react) prototype
- sak no. 02/2022** Sprint 2 review & retrospektiv
- sak no. 03/2022** Sprint 3

Plan for arbeid frem til neste møte

- Jobbe videre med frontend til vi får tak i lisensene. Skal purre på ImageZebra hvis dem ikke svarer snart. De andre lisensene får vi i posten.
- Sette opp en Windows maskin som vi kan ha alle programmene på som trenger lisenser.
- Jobbe videre på API'en.

Vedlegg

- Sprint 2: Review og Retrospektiv

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 25.02.2022

Møtereferat

Tid/Sted: 28-02-2022 14:00 - Teams

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Demonstrere API & Figma (+react) prototype

Vise arbeidsflyten ved hjelp av use cases.

Prototype som viser HELE prosessen.

Kilder til accessibility:

- <https://www.w3.org/TR/wai-aria-1.1/>
- <https://www.uutilsynet.no/>
- <https://lovdata.no/dokument/SF/forskrift/2013-06-21-732>
- https://developer.mozilla.org/en-US/docs/Web/HTML/Element/link#technical_summary
- <https://www.w3.org/WAI/fundamentals/accessibility-intro/>
- <https://www.w3.org/WAI/standards-guidelines/wcag/>

Sak no. 2/2022 - Sprint 2 review & retrospektiv

Skrive litt mer når vi skrive review og retrospektiv:

- Hva gjør vi videre
- Hva lærte vi nå, noe vi kan gjøre få å unngå dette/optimalisere dette
 - Refleksjon: hva lærte vi her.
 - Hva kunne vi gjort annerledes

Mer generelt om gruppen og prosessen (hvordan vi jobber)

- Gruppearbeid
 - Hvordan vi har fordelt arbeidet
 - Hvordan forholder vi oss til Gruppereglene
 - Hvordan er kommunikasjonen
 - Hvordan er prosessen?
- tidsmessig
- verktøy vi bruker
- En slags generell overvåkning av gruppen.

Sak no. 3/2022 - Sprint 3

OK

Ekstra

Bruke møtene til en slags øving for spørsmålene som kommer til presentasjonen.

28.02.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 14-03-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Agenda

- sak no. 01/2022** Sprint 3 retrospektiv
- sak no. 02/2022** Demo av det vi har til nå
- sak no. 03/2022** Plan for sprint 4

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 11.03.2022

Møtereferat

Tid/Sted: 14-03-2022 14:00 - A2

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Sprint 3 retrospektiv

Avklare med Ottar om parameter filene, hva som er viktigst

Sak no. 2/2022 - Demo av det vi har til nå

Feedback på upload (progresjons bar)

- .

Referere til standarden i frontend.
Hva er kompetansenivået brukerne.

Sak no. 3/2022 - Plan for sprint 4

Implementere ferdig MVP funksjonalitet, ikke noe ekstra.
Progresjons bar

Sak no. 4/2022 - ekstra

Spørre om oversikt / rapport / statistikk -> resultatdata
Invalid XMP i IQX

- Rapporter til utviklerne av programmet
- Kan godt skrive om det i rapporten også

Dokumentere hvilken versjon parseren fungerer på

Skrive ned beslutninger + begrunner.
Dokumentere erfaringer/særheter til oppdragsgiver. Brukerinfo.

Greit med en brukertest på ansatte fra IMKAR eller folk fra colourlab

- Peter
- Skriv en slags guide / minimalistisk brukermanual

Statistikk på mengder med bilder fra IMKAR

14.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 21-03-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Frontend laster opp bilder
- Skrevet om mye av backend kode
- La til funksjonalitet for å legge til metadaten i de individuelle bildene
- Funksjonalitet for å laste opp de individuelle bildene

Agenda

sak no. 01/2022 Statusrapport

Plan for arbeid frem til neste møte

- Fullføre funksjonalitet som er planlagt/definert til MVP

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 18.03.2022

Møtereferat

Tid/Sted: 21-03-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Statusrapport

“Option” i programmet for å velge hva som blir skrevet ut (mtp. resultater / image target). Hva er det brukeren trenger?

Beskrive produksjonsprosessen.

.... pylint / eslin

Postman verktøy (OpenAPI / Swagger)

Beskrive generelle konsepter (MVC)

Velger rammeverk - liste

Rapport :

- Tenke illustrasjoner / Modeller
 - Gjør teksten lettere å lese / forstå
 - Kan godt bruke det til å illustrere kildekoden
 - Illustrere godt for å gjøre rapporten mer lettlest

Definere Generelle kriterier

- Performance
- Sikkerhet

Dokumenter hvilke test/program som ble kjørt i metadata

21.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 28-03-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Lagd OS QM Tool Endpoint
- Lagd download endpoint
- Lagd kode for å kjøre OS QM Tool
- Litt frontend scaling fixes

Agenda

- | | |
|------------------------|-------------------------|
| sak no. 01/2022 | Prøve å vise demo igjen |
| sak no. 02/2022 | Sprint 4 Retrospektiv |
| sak no. 03/2022 | Sprint 5 plan |

Plan for arbeid frem til neste møte

- Debugge OS QM Tool
 - Ta kontakt med produsent for å få litt hjelp/tips til hvorfor programmet oppfører seg sånn som det gjør (se retrospektiv)

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 28.03.2022

Møtereftrat

Tid/Sted: 28-03-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Demo

Brukeren får litt mer informasjon om hva dem kan trykke på osv.

Ta utgangspunkt i brukeren

Sak no. 2/2022 - Sprint 4 Retrospektiv & sprint 5

Hvordan håndtere dupliserte filnavn (xxxx_1.png eller 1_xxx.png osv)

- Hør med ottar hvordan dem ønsker det

Litt kort retrospektiv, kan fint utbedre litt mer

Ekstra

Saksnummerering i møteinnkalling.

Vanlig med kontinuerlig saksnummerering.

Skrive om / diskutere erfaringene og hva produsentene kunne gjort fra dem sin side for å gjøre prosessen bedre for vårt bruksområde (automatisjon).

- Kjøre programmet en gang, gi svarene og la oss teste disse verdiene mot A, B og C istedenfor å måtte kjøre programmet 3 ganger.

“Vi følte” - burde heller skrive at kunden ønsket (med diskusjon)

Skriv en plan for refactoring

- Lage en plan (oversikt)
- Estimere og prioritere (verdien, ikke sikkert alle svakheter må forbedres, hvilke er kritiske?)
- Issues
- Do the thing

Rapport:

- utkast ferdig til før påske
- Sette av dedikert tid
 - Gjerne litt lenge bolker med tid, gjerne 2 dager
- Fokus på hva oppdragsgiver kan få ut av det

- Hva har vi gjort for å gjøre koden utvidbar (hvorfor, hvordan med eksempler)
- Skrive tilbakemelding til produsentene (brev vi kan ha som vedlegg i rapporten)

Veiledning til rapporten:

- Jo **tidligere** ut med store biter (av rapporten), jo bedre sannsynlighet er det å få gode tilbakemeldinger
- Vær **spesifikk** med hva vi ønsker å ha tilbakemelding på
 - **Mange små biter** er lettere å gi tilbakemelding på enn en stor del
 - Side x-x, figur x-x, diskusjon x-y osv. Være spesifikk

beskrive prosessen

- Beslutninger med "low priority"
- Bestemmer oss for å stoppe (e.g. ImageZebra)
- Prioriteringer

Se på emnes beskrivelsen, hva skal vi demonstrere (hva skal vi kunne)

- sørg for at det er lett for leseren å finne dette i rapporten. (kan godt merke dette i teksten)
- Kan også se på vurderingskriteriene (UiS lenke, krasj kurs 2)

Glad sensor er en god sensor:

- Lite jobb for å krysse av fra vurderingslisten

28.03.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 04-04-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Integrert filvalidering inn i prosessen
- Skrevet om rapporten til engelsk
- Laget kode for å håndtere duplisert filnavn, skal høre med Ottar om han liker løsningen på mandag
- Begynt å skrive dokumentasjon for løsningen
- Begynt å refactor kode for å gjøre den mer ryddig
- Laget info knapp til frontend
- Lage funksjon for å zippe alle bildene
- Input checking (bare laste opp png og tiff) på frontend og backend
- Dokumentere særheter vi har støtet på / oppdaget i programmene

Agenda

sak no. 01/2022 Statusrapport

Plan for arbeid frem til neste møte

- Refactor kode, er litt mye duplisering.
- Skrive ferdig dokumentasjon (brukermanual / installasjonsguide / api dokumentasjon)
- Fullføre applikasjonen/løsningen

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 01.04.2022

Møtereftrat

Tid/Sted: 04-04-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022

Mulig det tar litt lengre til å skrive refleksjonsdelen siden vi har byttet til Engelsk.

informasjon om hvilke filer som kan lastes opp (Er ikke det noe man kan gjøre i browser, si at type = file, input = ...)

Brukerteste løsningen.

Dokumentere brukertesten i rapporten (spørsmål)

Ekstra

Dele rapport med veiledere.

Les emnebeskrivelse / oppgave + tidligere rapport (se på strukturen).

- Ethiske problemstillinger
- Samfunnsmessige problemstillinger
- Bærekraft
- Eks: hvorfor rapporten og repo er på engelsk

Fremover: Prioritere rapport og planlegge brukertesting.

- Hvis ikke tid til brukertest, ta utgangspunkt i informasjonen vi har fått fra Ottar om kunnskapsnivået til en bruker.

Mulig møtet utgår på Mandag 11. April.

04.04.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 11-04-2022 10:00 - Teams

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Refactored front end kode
- Kan velge hvilket program man ønsker å bruke på frontend
- Hente session status og vise det på frontend
- Frontend: Disable/enable nedlastingsknapp
 - Backend: Kan ikke last ned zip før den er helt klar
- Skrevet på rapporten

Agenda

sak no. 01/2022 Sprint 5 review og retrospective

sak no. 02/2022 Demo :)

sak no. 03/2022 Rapport Kapittel 1.1 - 1.6

Plan for arbeid frem til neste møte

- Påske
- Muligens skrive litt på rapport

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 08.04.2022

Møtereftrat

Tid/Sted: 11-04-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Sprint 5

Bok til brukertesting: A practical guide to user testing av Joseph S. Dumas

Kan godt diskutere i rapporten om hvor robust programvaren er, og hva konsekvensene er hvis det krasjer. Taper arkivverket mye tid osv..

Selv om resultatet evt. har gått veldig bra, er det lurt å diskutere hvorfor dette gikk bra og hva vi gjorde for å unngå problemer. Lurt å prøve å gå dypt i denne analysen. Hvordan gjennomføringen har gått i forhold til planen. Sensor synes det er interessant å vite hvorfor det har gått bra fra en pedagogisk side.

Sak no. 2/2022 - Demo

Skrive om forskjellene mellom programmene de kan velge mellom.

Høre med Ottar: Rekkefølge på target eller programvare. Hva burde man velge først?

Cancel og submit sitter litt for seg selv Forklarende tekst på session id

- Endre session status til "Running" osv. mer forklarende
- Hva er arbeidsflyten
- Hvorfor vises session id - er det viktig, kanskje bare vise frem de siste 5 chars

Sak no. 3/2022 - Rapport 1.1 - 1.6

Overordnet og små kommentarer.

Viktigste: Får med alt på en måte som at det er forståelig for leseren.

Det kommer kommentarer på språket selv om det ikke nødvendigvis er det viktigste.

Overordnet:

- Tenk på leseren. Tegn bildet i introduksjonen
 - Hva gjør arkivverket i praksis? Hva skjer hos arkivverket.
 - Hvorfor vil de digitalisere det? Tilgang
 - Hvordan ser den manuelle prosessen ut i dag? Hvordan gjør de kvalitetskontrollen i dag?
- Kjenner ikke til ISO, arkiv, osv
 - Hvis leseren må lese mange sider før den får vite hva dette handler om så blir den sur.

- E.g. Hva er et target?
- Gi den et **bilde** tidlig om hva det handler om, ISO standard osv.
 - Gi leseren en forståelse / helhetsbilde **tidlig**.
 - Forklar temaet lett tidlig.
- Rapporten er skrevet til en hvilken som helst student, ikke Ottar eller en ekspert. De skal forstå tidlig hva bidraget er tidlig.
- En tekst må **skrives 3 ganger** før det blir bra :)
 - Må vurdere selv hvilke kommentarer vi burde følge opp (mtp. tid), en tekst kan alltid bli bedre.
 - Hvis vi ikke forstår kommentarene, spør, gjerne med eksempler på forbedringer (mente dere x, y eller z?)
- Bruk verktøy for å øke nivået på teksten
 - Finne synonymer:
 - merriam webster dictionary
 - the free dictionary
 - Sekom / skrivetips fra NTNU: <https://www.ntnu.no/sekom/skrivetips>

Skrivestil:

- Bedre overgang mellom avsnittene, en slags into, konklusjon -> få flyt mellom avsnittene og seksjonene
- Hvorfor er jeg her, hvordan kom jeg hit, hva er veien videre.
- Sikter høyt: Jo mer presis, jo bedre
 - E.g. "this is" -> være presise i steden for
- FORKLAR MER. Gjerne bruke mer ord på å forklare, med figurer eksempler. Gjør det lettere å forstå hva vi prater om. Kutt ut til slutten hvis det blir for mye.
- Teksten er litt blandet, kanskje lag et tankekart for å ha en ide hvor ting kan passe inn. **En oversikt!**
- Kan godt endre kapittel 2 og 3. Gjennom kravspek kan gi leseren en forståelse for hvorfor teorien er viktig. Ellers må kapittel 1 utdypes en del.
 - Intro
 - Krav
 - Teori
 - Design
 - Prosess
 - ...

Sosiale faktorer:

- Tar vi noen jobber?
- Er dette noe de kommer til å ta i bruk
- Hvordan skanningen passer inn.

Innledning

- Hva er omgivelsen, hva er det vi gjør.
- La leseren få lov til å forstå hva det snakkes om før en går inn i detaljer.
- Få en leser som ikke har noe relevanse til arkivverket til å lese over - introduksjonen skal være forståelig for alle.

Figure 1.1: Role distribution

- Rune syntes ikke det passer at vi er under arkivverket

11.04.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 25-04-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Møte med Ottar
- Skrevet på rapport
- Endrett sånn at en velger target før program på frontend
- Auto fetching av resultater på frontend

Agenda

sak no. 01/2022 Kommentar til Kapittel 2 & 3

Plan for arbeid frem til neste møte

- Lage en fremdriftsplan for rapportskrivning

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 21.04.2022

Møterefereat

Tid/Sted: 25-04-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Kommentarer til rapport (kapittel 2 & 3)

Husk verdien av ordene vi skriver -> må komme frem i rapporten

Viktig med rapporten

- Sette det vi har gjort i et bredt perspektiv. Hvorfor har det verdi?
- skrive mye mer av det som er opplagt for oss, men ikke for leseren
 - Alternativer?
 - sette det i Sammenheng
 - Gi nok informasjon / kunnskap om det vi har gjort.
- Mer detaljer / gå mer i dybden
 - Refere til hva andre gjorde, hvilke alternativer forkastet vi
 - eks. use cases: forklar hvorfor noe ble kuttet ut

Generelt: Få det opp på middeelveien :)

- Bruk mer litteratur. e.g. open source. og bruk teorien på å forklare metodene og argumentere hvorfor vi valgte den.
- Ikke være redd for å bruke et avsnitt på å forklare fagord.
 - illustrere (bilder, figurer, sette det i perspektiv) istedenfor å copy past definisjonen.
- En referanse til hvordan man skriver rapporter
- En 5. semester idi student skal kunne forstå alt i rapporten
- Gjøre rede for alternative ord der det blir introdusert.
- Bare skriv, gjør ikke noe om det er dårlig eller urelevant, det kan flyttes på eller slettes

Eks. bilbiotekene vi har brukt

- Hvordan skal vi bruke det
- Hvilke krav setter vi?
-

Godt arbeid, lett å forstå hva vi har gjort.

25.04.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 02-05-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

Rapport skiving, mest kapittel 1-5, prøvde å få ferdig første utkast til disse kapitlene.

Agenda

- | | |
|------------------------|---|
| sak no. 01/2022 | Kommentarer til rapport, fokus på kapittel 4 & 5. Hvis dere har tid, så kan dere skimme igjennom kapittel 1-3 og se om forbedringene er gode nok. |
| sak no. 02/2022 | Se på fremdriftsplanen |

Plan for arbeid frem til neste møte

1. Rapport skiving, mest kapittel 5-10.

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 27.04.2022

Møtereferat

Tid/Sted: 02-05-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Rapport kapittel 4 & 5

Seksjon 3 - Hvordan vi bruker disse teknologiene.

- Kan godt referere til seksjonen vi skriver om det.

Generelt referere mer til litteratur / utfylle mer

Øke leseopplevelsen:

- Bruk foroverreferanse for å si ifra "det kommer mer info om dette"
 - Kan godt si kort hvordan vi har brukt det med en referanse.
- Gjør det lettere for leseren, reduser hvor mange ting som er ubesvart (maks 5-7)
- Avrunde diskusjon med et sammendrag, og dra det opp igjen i diskusjonskappitlet.
- Bruk det mest passende ordet
- Gjør det tydelig at eksemplene er fra vårt kappitel

Generelt, skriv mer utfyllende.

Sak no. 2/2022 - Ekstra

For mange ideer i et avsnitt, ha en ide per avsnitt og heller del teksten opp i flere avsnitt: <https://www.sokogskriv.no/skriving/akademisk-sprak-og-stil.html#a-skape-flyt>

Intoduser ideen (temasetning) -> argumenter -> konkluder

Mangler i teksten som gjør at det kan være vanskelig å forstå hva vi mener. Vi har bakgrunnsinformasjonen i hodet, men det gjør ikke leseren så viktig å få ned alt relevant i teksten.

Flytte scrum board opp til Scrum

Nevne forskjeller mellom scrum / kanban - evt. ta det bort.

Evt. Feil i tekst:

- MVC / RESTful
- Se på WWW undervisning

Git board / issues -> prat om taggene. Hvordan har vi brukt backlog osv.

Introduksjon:

- Forklar mer detlasjert behovet som kunden har ut i fra prosessen dems / det de allerede gjør.
- De har bilder, de har scennere, de har programvare, men hva er fordelen med det vi gjør?
- Kan lage en brukerprofil / persona på Ottar for å beskrive behov. Få han til å beskrive arbeidsflyten, hvordan vil løsningen passe inn i hans arbeidsflyt...

En vanlig leser bruker ikke mye tid å vedleggene, er mest hvis de ønsker å vite mer, men det skal ikke være nødvendig for at leseren skal forstå sammenhenget/det vi prøver å formidle.

Beskriv teknologi

Hvordan vi har brukt det

Eksempler

02.05.2022, Martin Wighus Holtmon

Møteinnkalling

Tid/Sted: 16-05-2022 14:00 - A215

Møteinnkalling er sendt til:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Arbeid siden forrige møte

- Vurdert tilbakemeldinger fra veiledere
- Revidert kapittel 1-4
- Satt inn flere illustrasjoner
- Skrevet et første utkast og sendt det til Ottar
- Lagt til open source lisens, [MIT](#)

Agenda

- | | |
|------------------------|---|
| sak no. 01/2022 | Hvordan fungerer innleveringen (Hvordan er prosessen? Autogenerering av forside osv.) |
| sak no. 02/2022 | Om veiledere har noen ting vi burde fokusere på/viktige mangler. |

Plan for arbeid frem til neste møte

- Revidere rapport
- Legge til gjennstående vedlegg (møtereferat, timelog)

Gi beskjed om du ikke kan delta i møtet.

Hilsen,
Martin Wighus Holtmon

NTNU 13.05.2022

Møtereferat

Tid/Sted: 16-05-2022 14:00 - A215

Til stede:

Rune Hjelsvold
Martin Wighus Holtmon
Johanna Johansen
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

Ikke til stede:

Moderator: Jakob Frantzvåg Karlsmoen

Sak no. 1/2022 - Prosessen ved innlevering

gå inn i grafisk senter -> forside

- mal til trykk -> studentoppgave
- Vi må betale for trykk (står sum på siden)
- Laste ned omslag

NTNU Open (se *Infoskriv april om innlevering og presentasjon*)

- Kan publiseres med en gang, **gå godkjennes i inspera**
- Gjøres i inspera når vi leverer der
- Kan gjøre endringer frem til fristen
- Tittel på oppgaven: Titelen som skal være på diplommet står der. Engelsk.
- Vedlegg i inspera: e.g. zip av koden

Sak no. 2/2022 - Noe spesielt vi burde fokusere på i innspurten

Gå igjennom oppgavebeskrivelse og se om det har blitt lagt nok vekt på alle punktene.

Viktigste er starten og slutten.

Starten:

-

Slutten:

- Hva lærte vi, hovedkonklusjon
- Konkluder med 2-3 hovedpunkter som leseren sitter igjen med.
- Analytisk evne / Diskusjon: Analyse -> syntese
 - Dele analysen opp i mange deler
 - Hva sitter vi igjen med
 - Hva er anbefalingene videre + forbedringer
 - Mangler slutten på diamanten ;)

Abstract -> sammendrag av konklusjon, mangler prosessen (konkluder prosessen, e.g. smidig prosess)

Ekstra:

- referer til timeplan i rapporten, skriv et sammendrag av timeloggen og referer til vedlegget
- alt av vedlegg må bli referert til i rapporten
- Underseksjon hvor møtereferat/info om møter blir samlet.
- Legg avtale som vedlegg i hovedrapporten
- Nevne hva som er lagt ved samme sted som git repo er nevnt
- Sende et eksemplar til Ottar.

16.05.2022, Martin Wighus Holtmon

Appendix M

Manual for KAD

Manual for KAD

Martin Wighus Holtmon
Jakob Frantzvåg Karlsmoen
Mikael Falkenberg Krog

April 2022

NOTE: This manual was written for the version of KAD that was finalized in the authors bachelor thesis. It does not reflect further development after the delivery date of the thesis.

Contents

1	Repository	4
2	Backend	4
2.1	Installation	4
2.1.1	Platform	4
2.1.2	Python 3.10	4
2.1.3	Image Quality Evaluation Software	4
2.1.4	JHOVE	4
2.2	Setup	5
2.2.1	Configuration file	5
2.2.2	Dependencies	6
2.3	Usage	7
2.3.1	Running	7
2.3.2	Interacting	7
2.4	Adding support for new software	7
3	Frontend	9
3.1	Installation	9
3.1.1	Platform	9
3.1.2	Node.js v17.6.0	9
3.2	Setup	9
3.2.1	Configuration file	9
3.2.2	Dependencies	9
3.3	Usage	10
3.3.1	Serving locally	10
3.3.2	Building	10
3.3.3	Page descriptions	11

1 Repository

The project can be downloaded from the repository on GitLab at https://gitlab.com/bachelor_group9_2022/kvalitetssikring-av-digitisering

2 Backend

2.1 Installation

2.1.1 Platform

The KAD backend is only designed to run on Microsoft Windows, as the currently supported image quality evaluation (IQE) software are Windows-based.

2.1.2 Python 3.10

The recommended way of installing Python is through the Windows store¹. Python 3.10 is the required version for running KAD.

2.1.3 Image Quality Evaluation Software

Currently the backend supports two IQEs:

- IQ-Analyzer-X by Image Engineering²
- OS QM-Tool by Zeutschel³

The backend can work with only one of the IQEs. It can work with none, but then there would be no functionality for performing analysis, only retrieving results of previous analyses.

2.1.4 JHOVE

JHOVE must be installed for file validation. It can be downloaded from the official Open Preservation Foundation webpage.⁴

¹<https://apps.microsoft.com/store/detail/python-310/9PJPW5LDXLZ5>

²<https://image-engineering.de/products/software/iq-analyzer-x/>

³<https://www.zeutschel.de/en/software/qm-os/>

⁴<https://jhove.openpreservation.org/>

2.2 Setup

2.2.1 Configuration file

The configuration file is vital for the backend to work, and must be properly populated before usage. A detailed explanation of all the fields in the configuration file can be found below. If the IQEs and JHOVE were installed at their default locations, the default values should work without any modification.

The configuration file must be saved as config.ini in the directory the backend is run from.

- **API**

This is configuration relating to the API.

- **Port**

This is the port the API will be listening on.

- **StorageFolder**

This is where sessions will be stored and files uploaded.

- **JHOVE**

This is configuration relating to JHOVE.

- **JhoveInstallPath**

This is the path to where JHOVE is installed.

- **IEQS**

This is configuration that is common for all IQEs.

- **ConcurrentSessions**

This is the amount of analyses that can run at once.

- **IQ ANALYZER X**

This is the configuration relating to IQ-Analyzer-X

- **InstallPath**

This is the path to where IQ-Analyzer-X is installed.

- **SessionTimeout**

This is the value (in seconds) of how long an analysis can be ongoing before the analysis is cancelled. This is mostly relevant if the IQE software has produced an error or is frozen.

- **OS QM-Tool**

This is the configuration relating to OS QM-Tool

- **InstallPath**

This is the path to where OS QM-Tool is installed.

- **SessionTimeout**

This is the value (in seconds) of how long an analysis can be ongoing before the analysis is cancelled. This is mostly relevant if the IQE software has produced an error or is frozen.

```
[API]
Port = 5000
StorageFolder = C:\Users\%USERPROFILE%\KAD

[JHOVE]
JhoveInstallPath = C:\Users\%USERPROFILE%\jhove

[IQES]
ConcurrentSessions = 3

[IQ ANALYZER X]
InstallPath = C:\Program Files\Image Engineering\iQ-Analyzer-X 1.5.1
SessionTimeout = 120

[OS QM-Tool]
InstallPath = C:\Program Files\Zeutschel\OS QM-Tool
SessionTimeout = 60
```

Listing 1: Configuration file with default values

2.2.2 Dependencies

To install the required dependencies, locate the requirements.txt file in the backend directory and run this command from a Powershell or CMD window:

```
pip install -r requirements.txt
```

2.3 Usage

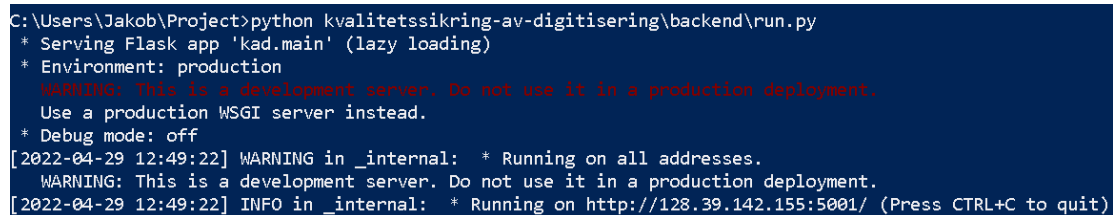
2.3.1 Running

To run KAD locate the file `run.py` in the backend folder and use this command from Powershell or CMD:

```
python run.py
```

Remember that the config file must be in the directory you are running the command from.

If successful, you should see a printout like in Figure 1.



```
C:\Users\Jakob\Project>python kvalitetssikring-av-digitisering\backend\run.py
* Serving Flask app 'kad.main' (lazy loading)
* Environment: production
  WARNING: This is a development server. Do not use it in a production deployment.
  Use a production WSGI server instead.
* Debug mode: off
[2022-04-29 12:49:22] WARNING in _internal: * Running on all addresses.
  WARNING: This is a development server. Do not use it in a production deployment.
[2022-04-29 12:49:22] INFO in _internal: * Running on http://128.39.142.155:5001/ (Press CTRL+C to quit)
```

Figure 1: Output when starting backend

2.3.2 Interacting

To interact with the backend there are two options currently:

- Use the frontend described later in this document
- Make manual requests according to the documentation⁵

2.4 Adding support for new software

The group wanted to make it easy for future developers to add new quality evaluation software to the program. To add support for a new software, you would first write the code for running analysis using the new software and writing the results to the `results.json` file. The JSON file has a specific structure.

The next step is to execute this code in the match case in the `analysis_run.py` file. There is one match case for device level analysis and one match case for object level analysis.

```
match iqes:
    case "IQX":
        ix.run_iso_analysis(before_target_filename, session_id)
        ix.run_iso_analysis(after_target_filename, session_id)
    case "OQT":
        oqt.run_iso_analysis(before_target_filename, target, session_id)
        oqt.run_iso_analysis(after_target_filename, target, session_id)
```

⁵https://bachelor_group9_2022.gitlab.io/kvalitetssikring-av-digitisering/docs/

Listing 2: Match case for device level analysis

Code listing 2 shows what the match case for device level analysis looks like. Here you add another case for the keyword for the software added which runs the code for the new software. The keyword used here will be the keyword for the iqes query in the analyze POST request.

```
@analyze_endpoint.route("/api/analyze/device", methods=["POST"])
def analyze_device():
    """An endpoint for creating a session and starting analysis on device level.

    Returns:
        Response: a JSON response which contains the session id.
    """

    # for easy checking before start
    supported_iqes = ["IQX", "OQT"]
    supported_targets = {
        "IQX": ["UTT"],
        "OQT": ["UTT", "GTDevice"],
    }
```

Listing 3: Lists for what iqes and targets are supported

The final step is to add the keyword for the software to the supported_iqes and supported_targets in the analyze_endpoints.py file shown in code listing 3.

3 Frontend

3.1 Installation

3.1.1 Platform

The frontend can be built on and served from any platform (Windows, Linux, MacOS). The described steps in this document is for Windows.

3.1.2 Node.js v17.6.0

Node.js can be downloaded from the official website⁶, and installed using the installer downloaded from there. The recommended version for running the KAD frontend is v17.6.0.

3.2 Setup

3.2.1 Configuration file

The configuration file for the frontend only specifies the ip and port for where to find the backend. The default value will only work if the backend is running with it's default values on the same system as the frontend and the frontend is accessed in the web browser of said system.

The configuration file must be saved as config.js in frontend/src/ directory before serving and/or building.

- **API_URL**

This is the url where the backend is accessible from. Must include port.

```
{
  "API_URL": "http://127.0.0.1:5000"
}
```

Listing 4: Configuration file with default values

3.2.2 Dependencies

To install the required dependencies, locate the package.json file in the frontend directory. In this directory, run the following command from a Powershell or CMD window:

```
npm install
```

⁶<https://nodejs.org/en/download/>

3.3 Usage

3.3.1 Serving locally

In the directory for the frontend, run the following command in a Powershell or CMD window to serve the frontend.

```
npm run start
```

3.3.2 Building

To build the frontend for production use, run the following command in a Powershell or CMD window in the frontend directory:

```
npm run build
```

This will create a production-ready version of the frontend in a new build/ directory. This version is meant to be served with a web-server like Nginx, but this is outside the scope of this manual.

3.3.3 Page descriptions

Main Page

This is the landing page for the frontend. Here you can select between uploading object level targets or device level targets by clicking the respective buttons.

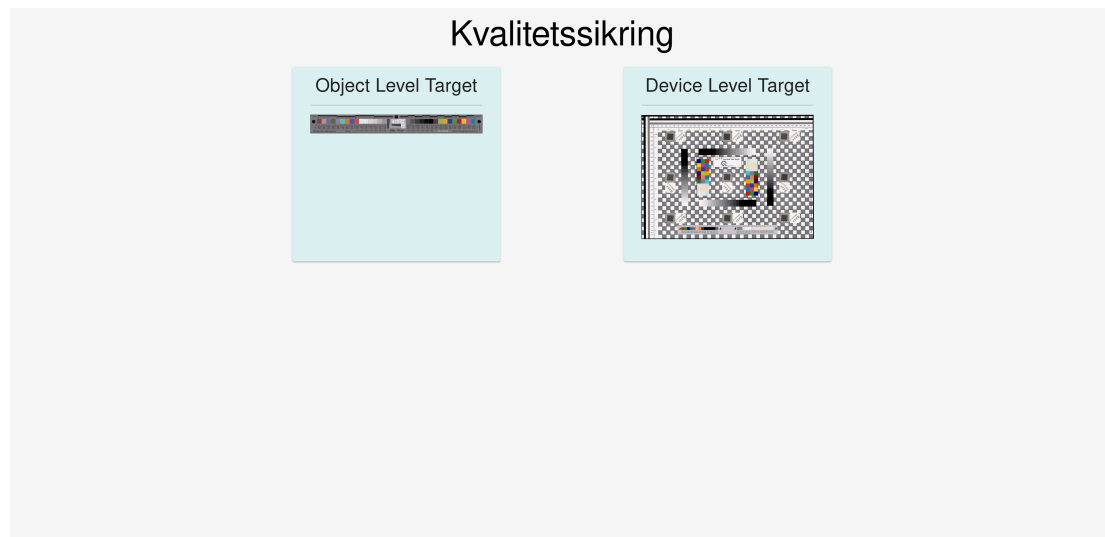


Figure 2: Screenshot of main page

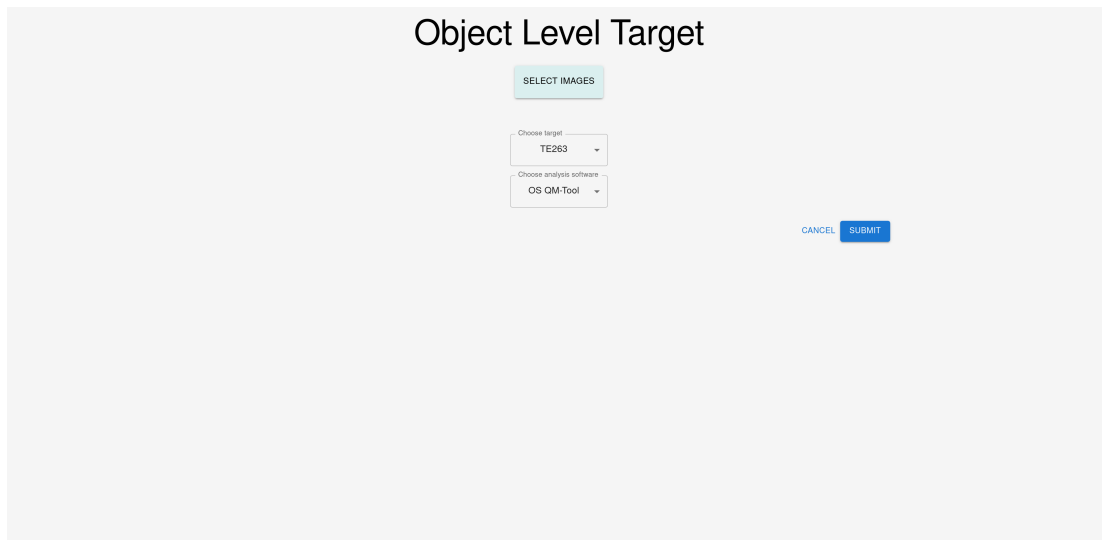
Object level target upload page

This is the page where you can upload images containing object level targets. The "SELECT IMAGES" button will prompt you to select however many images you want for upload.

By first using the dropdown for "Choose target", only the available image quality evaluation software for the selected target will be available in "Choose analysis software".

Clicking "Submit" will start the upload and analysis, and redirect you to the results page.

Clicking "Cancel" will take you back to the main page.



The screenshot shows a web interface titled "Object Level Target". At the top center is a light blue button labeled "SELECT IMAGES". Below it are two dropdown menus. The first is labeled "Choose target" and has "TE263" selected. The second is labeled "Choose analysis software" and has "OS GM-Tool" selected. To the right of these dropdowns are two buttons: "CANCEL" and "SUBMIT". The "SUBMIT" button is blue, while the "CANCEL" button is light blue.

Figure 3: Screenshot of object level target upload page

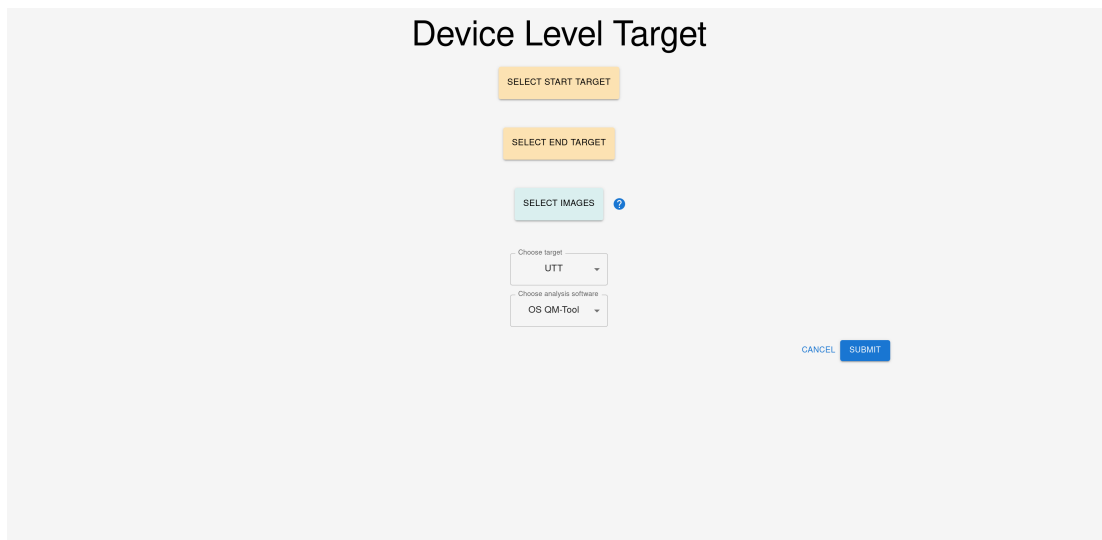
Device level target upload page

This is the page where you can upload images containing device level targets, and the images that were digitized between the targets. The "SELECT START TARGET" and "SELECT END TARGET" button will prompt you to upload the start and end target image files respectively. The "SELECT IMAGES" button will prompt you to upload the images digitized between the targets, but it is optional and not required to analyze the target images.

By first using the dropdown for "Choose target", only the available image quality evaluation software for the selected target will be available in "Choose analysis software".

Clicking "Submit" will start the upload and analysis, and redirect you to the results page.

Clicking "Cancel" will take you back to the main page.



The screenshot shows a web interface titled "Device Level Target". It contains the following elements:

- Three orange buttons stacked vertically: "SELECT START TARGET", "SELECT END TARGET", and "SELECT IMAGES" (with a small blue help icon to its right).
- A dropdown menu labeled "Choose target" with "UTT" selected.
- A dropdown menu labeled "Choose analysis software" with "OS QM-Tool" selected.
- At the bottom right, two buttons: "CANCEL" (light blue) and "SUBMIT" (dark blue).

Figure 4: Screenshot of device level target upload page

Results page

This page shows the results of an analysis. It will automatically update as the analysis progresses, and display the name and final score of each image analyzed.

Clicking on one of the images will take you to the detailed analysis results of that image.

Clicking on the home button at the top of the page will take you back to the main page.

Clicking on the download button in the top right of the page will download a collection of all images in the session. The images in this collection will also have their analysis results embedded in their metadata.

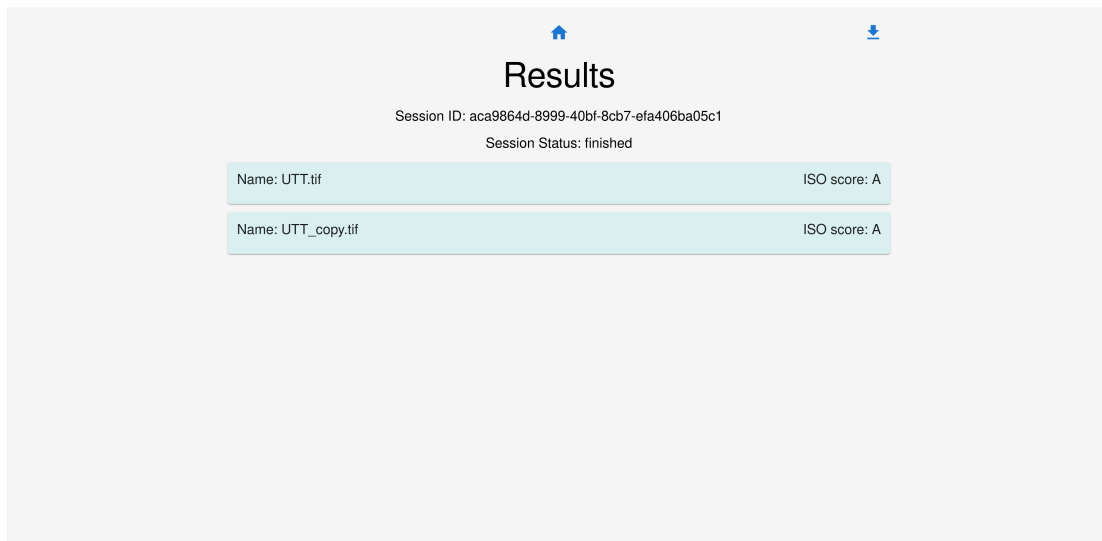


Figure 5: Screenshot of results page

Single image result page

This page shows the detailed result of an analysis on an image. It will show the results for each specification level, and if it passed that level. Clicking on one of the dropdowns will allow you to see the analysis categories and individual results.

Clicking on the back button in the top left of the page will take you back to the results page.

Clicking on the home button at the top of the page will take you back to the main page.

Clicking on the download button in the top right of the page will download the analysed image. This file will also have the analysis results embedded in the metadata.

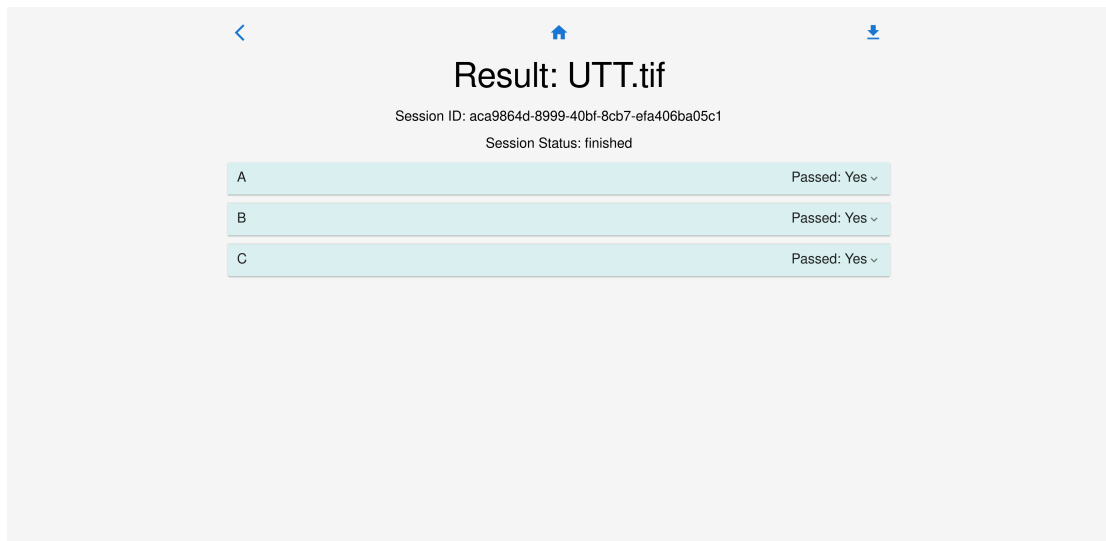


Figure 6: Screenshot of single image result page

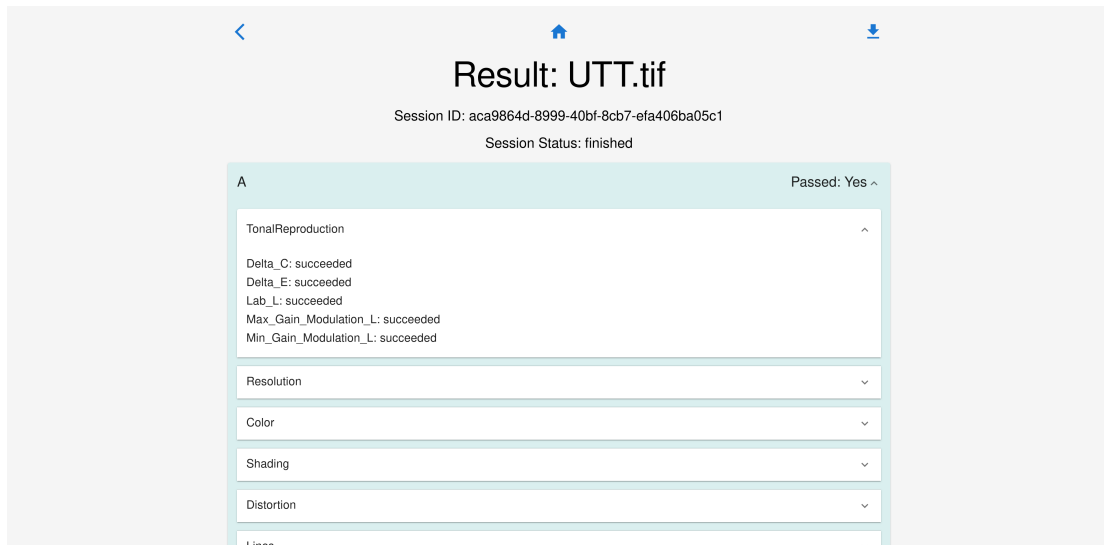


Figure 7: Screenshot of single image result page with expanded details

Appendix N

Timelog

Martin		
Dato	Timer	Kommentar
17. desember	1.5	Bli-kjent møte med oppdragsgiver
11. januar	4.5	Lynkurs, spørretime, Startet arbeid på prosjektplan. Satte opp tidsskjema og git gruppe.
12. januar	3	Fortsettet med arbeid på prosjektplan. Jobber med det som kan fylles ut før møter med oppdragsgiver og veiledere. Skrive møte template, summon, minute
13. januar	0.75	Oppstartsmøte med veiledere
14. januar	3	Møte med oppdragsgiver og jobbing på prosjektplan. Skrev møteinnkalling
17. januar	4	Skriving på prosjektplan og lagd gantt-diagram
18. januar	4	Skriving på prosjektplan
19. januar	4	Skriving på prosjektplan og møte med veiledere og oppdragsgiver
20. januar	3	Skriving på prosjektplan
21. januar	4	Skriving på prosjektplan og skriving på møteinnkalling
24. januar	2	Møte med oppdragsgiver og veiledere
25. januar	4	Jobbet med prosjektplan
27. januar	2	Jobbet med prosjektplan, standardavtale og taushetsavtale
28. januar	4	Jobbe med Sprint board og Sprint 1 forberedning. Skrev møteinnkalling
31. januar	5	Jobbe med sprint 1, møte med veiledere og oppdragsgiver. Leverte prosjektplan
3. Februar	4.25	Definert MVP, Sett på ImageZebra og Delt.ae, eksprementert med metadata validering
4. Februar	3	Møte med Ottar for å få forklaring i object og device level target. Skrev møteinnkalling
7. Februar	3	Møte + panel discussion
8. Februar	2.25	Møte med ottar, fordele arbeid til rapport, sette opp timeplan
9. Februar	3	research/rapportskriving om Bildemål
10. Februar	4	Flow Chart, Image target research, (var sliten etter vaksine)
11. Februar	3.5	Jobbet med image target, Sprint review & retrospektiv
14. Februar	5	Planlagt sprint 2, frisket opp python kunnskapene, møter med veileder og arbeidsgiver
16. Februar	3.5	Fullført image target, sett på opendice, review code
17. Februar	9	Jobbet med å sette opp SonarQube, sett på forskjellige image quality tools med Ottar, komponert mail til de forskjellige aktørene, sett på GoldenThread programmet som Ottar tok med
18. Februar	4.5	Mer mail skriving, fikset SonarQube, ordnet med "ntnu-global" så vi har public ip til OpenStack, notater fra besøket av Ottar
21. Februar	0.5	Møte med veileder
23. Februar	0.5	Svarte mail fra Ottar. Klassens time tok all tiden (11:00-14:30)
24. Februar	6	Startet å designe frontend i figma, startet nodejs prosjekt
25. Februar	5.25	Designet videre på prototypen i Figma. Satt opp front-page til frontend. Sprint 2 retrospektiv. Jobbet litt på rapporten
28. Februar	5	Planlagt sprint 3, fortsatt arbeid på frontend, møter med oppdragsgiver og veileder
02. Mars	5.5	Jobbet videre med frontend, var på lynkurs 2
03. Mars	6	Prøvd ut OS QM TOOL, startet å skrive en parser for resultatet den skriver ut
04. Mars	4.5	Jobbet videre med parser for å hente ut infoen fra de forskjellige seksjonene
07. Mars	4.5	Jobbet videre med parser for summary

Martin		
Dato	Timer	Kommentar
09. Mars		6 Skrev ferdig parser for OS QM Tool, lagde tester, fix cors security hotspot
10. Mars		6 Jobbet med API Dokumentasjon, bug fixes, få analysen til å kjøre ved å kalle på en API
11. Mars		6.75 Få IQX til å analysere ved å sende en request til API, parse resultatet til IQX
14. Mars		6 Eksportere output fra iqx parser til fil og integrere dette inn i metadata
16. Mars		2 Jobbet litt med IQX parser, startet å modellere use cases
17. Mars		5.75 Jobbet med IQX så den kan kjøre flere concurrent sessions, hjelp med refactor av prosjekt struktur, use case diagram, litt debug, hjelp til å skrive tester til session_manager
18. Mars		5.25 Hjelp med refactored koden, Hjelp mens vi laget funksjonalitet for å lagre metadata i bildene som ble lastet opp, skrev møteinnkalling
21. Mars		4.5 Prøvd å lagre metadata på korrekt form
24. Mars		6.5 Jobbet litt med OQT parser, lagre resultatet til result fil og hjelp med å sette opp kode/endpoint for å kjøre OQT
25. Mars		8.5 møte med Ottar, la til funksjonalitet for unique filename når en laster opp filer, skrev sprint review og møteinnkalling
28. Mars		5 Lagde kode for å håndtere dupliserte/like filnavn. Møte med veiledere
29. Mars		6 Satt opp rapport struktur i henhold til lunkurs 2, startet å skrive på rapporten
30. Mars		7 integrert file validation i prosessen, litt bug fixing
31. Mars		6.5 Satt overall score for OQT, lar OQT fullføre selv om den får exit code 2, oppdatert kode for å håndtere duplisert filnavn
1. April		0.5 Var syk, skrev møteinnkalling
4. April		4.5 fortsatt syk, lagde metode for å hente image session status, oversatt litt på rapport, møter
5. April		5.25 Skrevet på rapporten: vår bakgrunn. Gått over chapter 1, hatt møte med Ottar
2. Mai		5.5 Mest sett over merge requests. hjelp med client side filetype check, så over "Improve analyze endpoint and backend code", hjelp litt når Mikael implementerte muligheten til å velge iqes tool i frontend. Sett på gitlab jobs kjøre i en time mens vi prøvde å få til frontend coverage report :(
7. April		4 Operational requirements
8. April		4.25 skrevet litt om sprints i rapporten, skrevet sprint 5 retrospektiv og møteinnkallinger.
11. April		1.5 Møter med veiledere og arbeidsgiver
19. April		5 Fysisk møte med Ottar
20. April		4.75 Fikset på innledning, endret ansvarsforhold kart, skrevet på background (nåværende prosess for digitsering hos imkar) og role distribution
21. April		4 Rapport skrivning, vært med på å satt opp og forklare rapport strukturen
22. April		4.5 Skrevet litt på rapporten om backend / reslt file / parser
25. April		2 Møter med veiledere og arbeidsgiver
26. April		5.25 Rapportskrivning: FADGI Sandard, lest over kapittel 4, skrevet om non-functional requirements
27. April		5 Rapportskrivning: Skrevet introduksjon til functional requirements, lest over kapittel 1-5
28. April		9 Arkivarforening i Lillestrøm

Martin		
Dato	Timer	Kommentar
29. April	5.25	Skrev på backend integration om endpoint documentation, lest over installation manual.
2. Mai	5	rapport: Skrev om SonarQube og parser, og var på møte med veiledere
3. Mai	5	skrev litt om parser og lest over tidligere kapitler
4. Mai	4.25	Skrevet på discussion
5. Mai	5	Skrevet om REST constraints, ordnet med fadgi referanser, listet opp ting som kan bli videreutviklet i applikasjonen og skrevet litt om det, gjort generell review av introduksjonen og abstract.
6. Mai	4.5	skrevet om kanban og kanban vs scrum
9. Mai	5	Skrevet om SonarQube, lagt ved eksempler osv
10. Mai	4.5	Gått igjennom kommentarer fra veiledere, møte med Ottar
11. Mai	5	Gikk igjennom kommentarer fra veiledere, satt inn illustrasjoner og noen kilder
12. Mai	4.25	Skrev/fikset innledninger i forskjellige sections
13. Mai	5	Skrevet om FADGI standard. Revidert kapittel 1-4
16. Mai	5	Lest over kapittel 4-6, møte med veiledere
18. Mai	6	Konklusjon, møte med Ottar, gått igjennom introduksjon og konklusjon med grammarly
19. Mai	9	Fikser på rapport, spesielt kapittel 6-7 og gjør alt klart til å bli levert

Jakob		
Dato	Timer	Kommentar
17. desember	1.5	Bli-kjent møte med oppdragsgiver
11. januar	4.5	Lynkurs, spørretime, Startet arbeid på prosjektplan. Satte opp tidsskjema og git gruppe.
12. januar	2	Fortsettet med arbeid på prosjektplan. Jobber med det som kan fylles ut før møter med oppdragsgiver og veiledere. Skrive møte template, summon, minute
13. januar	0.75	Oppstartsmøte med veiledere
14. januar	3	Møte med oppdragsgiver og jobbing på prosjektplan
17. januar	4	Skriving på prosjektplan og lagd gantt-diagram
18. januar	4	Skriving på prosjektplan
19. januar	4	Skriving på prosjektplan og møte med veiledere og oppdragsgiver
20. januar	2	Skriving på prosjektplan
21. januar	4	Skriving på prosjektplan og skriving på møteinkalling
24. januar	2	Møte med oppdragsgiver og veiledere
25. januar	4	Arbeidet med å fullføre prosjektplanen
27. januar	2	Revidere prosjektplan og avtaler
28. januar	4	Forberedet til sprint 1
31. januar	5	Møter med veildere og oppdragsgiver, fortsatt planlegging til sprint 1, levert inn prosjektplan
3. Februar	4.25	Undersøkt iQ-Analyzer-X, kontaktet Ottar for oppklaringsmøte, parkoding
4. Februar	3	Ødela ImageZebra, undersøkt alternativer for filvalidering, møte med Ottar
7. Februar	3.5	Panel discussion og afterparty, møte med veiledere
8. Februar	2.25	Møte med Ottar, ordnet rapporten, ordnet felles kalender
9. Februar	4.5	Begynt på hovedrapport, kranglet med testet ut openDICE, laget repo på GitLab for hovedrapporten også
10. Februar	4	Skrevet om ISO standard, laget flow-chart, generell rapportskrivning
11. Februar	3.5	Sprint reviw og retrospektiv, rapportskrivning (python, brukermønster)
14. Februar	5	Planlagt sprint 2, ordnet pc vi kan bruke for lisens, møte med arbeidsgiver og veileder
16. Februar	3.5	kranglet enda mer med prøvd å kartlegge måter å automatisere openDICE på, reviewet kode
17. Februar	9	Diskutert med Ottar, sendt mailer til firmaer som lager kvalitetsanalyseverktøy, fått innføring i GoldenThread av Ottar
18. Februar	5	Skrive mail, notert fra Ottar besøk, fikset SonarQube, møteinkallinger
21. Februar	0.5	Møte med veiledere
23. Februar	0.5	Svarte mail fra Ottar. Klassens time som uventet tok 4 timer var en stopper for session i dag
24. Februar	5.5	Mailt mer med Ottar, jobbet litt på frontend wireframe, satt opp api i python kode, laget et endpoint for apien
25. Februar	5.25	Finalisert api-koden jeg startet på i går, skrevet sprint review og retrospektiv
28. Februar	7.5	Satt opp Jenkins for CD av frontend med GitLab integrasjon, møte med veileder og oppdragsgiver
02. Mars	5.5	Kranglet med Jenkins, begynt på dokumentasjon for backend
03. Mars	6	Utforsket OS QM Tool og scripts for bruk, CD for backend
04. Mars	4.5	Utforsket IQ Analyzer X og scripts for bruk, begynt på kode for sessions, fikset Jenkins
07. Mars	4.5	Jobbet videre med session management

Jakob		
Dato	Timer	Kommentar
09. Mars	4	Fullført sessions, begynt på IQ Analyzer X kode. TA jobb tok vekk noe tid i dag
10. Mars	2.5	Jobbet videre med å få IQ Analyzer X til å kjøre med python kode. TA jobb og annet fag tok bort mye tid i dag
11. Mars	4.5	Arbeidet videre med IQ Analyzer X koden, kom nesten i mål
14. Mars	6	Møte med arbeidsgiver og veiledere. Jobbet med IQ Analyzer X kjøring av analyse og samling av resultat
16. Mars	4.5	Feilsøket CI pipeline, jobbet mer med IQ Analyzer X koden
17. Mars	5.75	Sett på individual result page, konkluderte med at backend må fikses før det er meningsfullt å begynne på den. Startet på refactoring av mye av backenden :)
18. Mars	5.25	Jobbet med refactoring av koden, begynt på frontend single result page
21. Mars	4.5	Jobbet med frontend single result page, møte med veiledere og oppdragsgiver
24. Mars	6.5	Jobbet på single result view på frontenden. Skrev også om cssen slik at frontenden skalerer skikkelig til alle skjermstørrelser (just in case noen vil analysere fra mobil :))
25. Mars	7.5	Møte med Ottar, bytte ut bilde på nettside
28. Mars	5	Dokumentasjon, skrive readmes
29. Mars	6	Rapportskriving
30. Mars	5	Oppdatert backend dokumentasjon, parkoding
31. Mars	6.5	Oppdatert parameter filer for OS QM Tool, kranglet med pipeline grrrr, endret navn på backend modul fra <i>kvalitetssikring av digitalisering</i> til <i>kad</i> , lagt til logging i hele backend modul
1. April	5	Startet å jobbe på nye, bedre, kulere analyze endpoints
4. April	4.5	Også blitt syk >:(Gjorde ferdig nye, bedre, kulere analyze endpoints. Møte med veiledere
5. April	5.25	Skrevet videre på utviklingsprosess i rapporten, sett over kapittel 1 slik at den er klar for tilbakemelding fra veiledere. Møte med oppdragsgiver
2. Mai	5.5	Gjort ferdig dokumentasjon for nytt endpoint. Fikset error i results page frontend. Parkodet
7. April	4	Download fix for backend og frontend. Skrevet videre på development process i rapporten.
8. April	2	Skrevet mer på rapporten. Sprint 5 retrospective og review, møteinkallinger
11. April	1.5	Møte med veiledere og arbeidsgiver
19. April	5	Fysisk møte med Ottar, tatt opp demovideo :d
20. April	4.75	Skrevet på rapport, oppdatert ansvarsforhold kart
21. April	4	Skrevet på target group i rapport, lest over kapittel 2 og 3
22. April	4.5	Rapportskriving
25. April	2	Møter med veiledere og arbeidsgivere
26. April	5.25	Rapportskriving: Kapittel 4, litt rund om kring
27. April	5	Rapportskriving: finpussing for forbereding til veiledere å lese gjennom. Førsteutkast av 1-5
28. April	9	Konferanse i Lillestrøm :O

Jakob		
Dato	Timer	Kommentar
29. April	5.25	Skrev manual
2. Mai	5	Skrev flere setninger og møtte med veiledere
3. Mai	5	Generell fiksing på rapporten
4. Mai	4.75	Skrev på scrum prosess
5. Mai	5	Generell skriving, skrev på konklusjon, la til lisens for koden
6. Mai	4.5	Rapporskriving
9. Mai	5	Fikset på tilbakemeldinger fra veiledere
10. Mai	4.5	Møte med Ottar, skrevet om open source og rettet etter tilbakemeldinger
11. Mai	5	Skrev om development environment
12. Mai	4.25	Fikset på kommentarer fra veiledere
13. Mai	5	Lest gjennom og fikset på kapittel 1-4
16. Mai	5	Møte med veiledere, fiksing på teksten
18. Mai	6	Møte med Ottar, ordnet vedlegg, jobbet på introduksjon og konklusjon
19. Mai	9	Crunch time

Mikael		
Dato	Timer	Kommentar
17. desember	1.5	Bli-kjent møte med oppdragsgiver
11. januar	4.5	Lynkurs, spørretime, Startet arbeid på prosjektplan. Satte opp tidsskjema og git gruppe.
12. januar	2	Fortsettet med arbeid på prosjektplan. Jobber med det som kan fylles ut før møter med oppdragsgiver og veiledere. Skrive møte template, summon, minute
13. januar	0.75	Oppstartsmøte med veiledere
14. januar	3	Møte med oppdragsgiver og jobbing på prosjektplan
17. januar	4	Skriving på prosjektplan og lagd gantt-diagram
18. januar	4	Skriving på prosjektplan
19. januar	4	Skriving på prosjektplan og møte med veiledere og oppdragsgiver
20. januar	3	Skriving på prosjektplan
21. januar	4	Skriving på prosjektplan og skriving på møteinkalling
24. januar	2	Møte med oppdragsgiver og veiledere
25. januar	4	Skriving på prosjektplan
27. januar	2	Skriving på prosjektplan og standardavtale.
28. januar	4	Planlagt Sprint 1 og oppdatert Sprint board
31. januar	5	Møter med oppdragsgiver og veileder. Mer planlegging til sprint 1, gått over prosjektplan med oppdragsgiver og litt mer tilbakemelding fra veileder.
3. Februar	4.25	Evaluerte JHOVE som en mulighet til fil format validering og laget et python script som bruker JHOVE til å validere en gitt fil.
4. Februar	3	Undersøkte og prøvde ut hvordan man legger til xmp metadata til et bilde. Møte med oppdragsgiver for mer informasjon angående ISO standaren og filtyper brukt i Digitalarkivet
7. Februar	3	Så panel diskusjon angående imaging standarization og møte med veiledere.
8. Februar	2.25	Møte med veileder, fordelt arbeid, fikset template til hovedrapport, satt opp kalender.
9. Februar	4	Begynt på hovedrapport og sett på alternativer til JHOVE
10. Februar	4.5	Evaluert BadPeggy, skrevet om XMP og JHOVE og fikset formatering på hovedrapporten. Skrev også litt på introduksjonen på hovedrapporten
11. Februar	3.25	Sprint retrospektiv, litt rapportskriving.
14. Februar	5	Planlagt sprint 2, møte med oppdragsgiver og veiledere, begynt på python modul for fil validering
16. Februar	3.5	Skrev python tests til fil validering, og fikset kode strukturen i repoet.
17. Februar	9	Satt opp CI/CD for testing av JHOVE validering av filer. Diskutering av hvilken image analysis software vi skal bruke. Var med å komponere mail til to selskaper om trial licenses.
18. Februar	5	Mer mail skriving, fikset SonarQube, fått tilgang til ntnu-global, dokumentasjon for besøk av Ottar, nesten fått tilgang til ImageZebra og OS QM-Tool, skrive møteinkallinger
21. Februar	0.5	Møte med veiledere
23. Februar	0.5	Sendte mail til Ottar, ble lite jobbing pga klassens time.
24. Februar	5.5	Satt opp frontend struktur, fikset litt på pipeline testing, laget wireframe for frontend.
25. Februar	5.25	Lagde hovedsiden til frontenden, satt opp automatisk frontend testing i CI/CD pipelinen. Skrevet sprint review og retrospective.
28. Februar	5	Forsatte arbeid på frontend, møte med oppdragsgiver og veiledere.
02. Mars	5.5	Mer arbeid på frontend (mest rydding av css) og lynkurs i rapportskriving
03. Mars	6	Har laget struktur for opplasting av bilder på frontend.
04. Mars	4.5	Kranglet med react :/
07. Mars	4.5	Lagde result og results sider og satt opp api fetching som bare mangler riktig url'er.

Mikael		
Dato	Timer	Kommentar
09. Mars	6	Laget home og tilbake knapper på resultat sidene. Fikset layout og feedback til bruker når bilder blir lastet opp. Hjulpet til med testing av OS QM-tool parser.
10. Mars	6	Jobbet med API dokumentasjon, og fått analyse endpointet til å faktisk kjøre analyse programmet.
11. Mars	6.75	Jobbet med analyse og resultat endpoints og parser for outputs fra IQ-Analyzer X.
14. Mars	6	Jobbet med eksportering av output fra IQ Analyzer X til json format, og integrering med metadata insetting i bildene. Har også hatt møter med arbeidsgiver og veiledere.
16. Mars	4.5	Koblet frontenden til backend api'en slik at man kan laste opp device level targets og se resultatene fra dem.
17. Mars	5.75	Laget nettside for bilde sammenlikning som Ottar spurte om. Laget use cases. Hjulpet med Result page på frontenden. Skrev tester til session manager
18. Mars	5.25	Hjulpet til å refactor backend, implementert metadata adding til bilder som er lastet opp. Jobbet på analyserings endpointet.
21. Mars	4.5	Lagt til overall score når targets blir analysert. Hjulpet til på frontend for result page. Hatt møte med oppdragsgiver og veiledere.
24. Mars	6.5	Laget endpoint til OS QM Tool, laget funksjoner for å kjøre analyse på bilder med OS QM Tool. Fikset xmp metadata funksjonen slik at den funker for både OS QM Tool og IQX
25. Mars	8.5	Laget endpoint for nedlasting av bilder i en session. Koblet object level frontend til backend. Filopplasting i device level fungerer. Møte med Ottar.
28. Mars	5	Laget tester for json helpers og image zipper. Laget input checking for analysene Møte med veiledere
29. Mars	6	Rapportskriving, mest oversettelse til engelsk
30. Mars	8	Mer input checking og andre backend fixes, frontend aria labels og on hover tekst. Prøvde å oppdatere metadata python package brukt
31. Mars	6.5	Fikset frontend for os qm tool resultater, la til fil type checks når filer blir lastet opp til analyse, fikset litt kode duplikasjon, og noen små frontend fikser. Prøvde å få py3xiv2 til å funke med ci/cd testene, men ga opp og byttet tilbake til pyxiv2.
1. April	5	Info knapp og input checking på frontend. La til test coverage for sonarqube
4. April	4.5	Session status endpoint, og vise session status på frontend. Møte med veiledere.
5. April	5.25	Skrevet på rapport, gått over chapter 1. Begynt å skrive på technical design og oppdatert flow chart til programmet. Skrev litt mer på hvilke teknologier brukt i chapter 2. Har også hatt møte med oppdragsgiver.
2. Mai	5.5	Frontend refactor og valg av software og target. Fikset code coverage report for backend, ga opp på frontend coverage report.
7. April	4	Skrevet på technical design på rapporten, research om forskjellige arkitekturer.
8. April	4.25	Skrevet mer på technical design og teknologier. Skrevet sprint review og retrospective for sprint 5. Laget møte inkallinger.
11. April	1.5	Møte med veiledere og arbeidsgiver
19. April	5	Fysisk møte med Ottar, frontend fix, rapport
20. April	0	Var opptatt
21. April	4	Skrev på rapport, lagde auto fetching av resultater.
22. April	4.5	Rapportskriving og fikset struktur på rapporten, testet auto fetching av resultater.
25. April	2	Møter med veileder og oppdragsgiver
26. April	5.25	Rapportskriving, skrev på implementation og en del fiksing på kapittel 2 og 3.
27. April	5	Rapportskriving, skrev på implementasjon og technical design, og om git workflow.
28. April	9	Arkivarforening Lillestrøm

Mikael		
Dato	Timer	Kommentar
29. April	5.25	Skrevet på rapport, skrev på kapittel 7 Quality Assurance
2. Mai	5	Skrev på implementation, møte med veiledere
3. Mai	5	Fikset på technical design, litt på technologies og git workflow.
4. Mai	4.75	Skrev på hovedrapport, mest på discussion kapittelet.
5. Mai	5	Skrev på conclusion og discussion.
6. Mai	4.5	Skrevet om development process (Scrum) kapittelet.
9. Mai	5	Fikset litt på kapitler 6-9. Mest fiksing på quality assurance kapitler
10. Mai	4.5	Skrev på frontend implementation kapittelet. Møte med oppdragsgiver.
11. Mai	5	Fikset sammendrag, fikset på discussion og implementation kapitlene.
12. Mai	4.25	skrevet om Abstract og fikset på innledninger til kapitler.
13. Mai	5	Lest gjennom kapittel 1-4 og revidert de.
16. Mai	5	Skrev på task field på kapittel 3. Møte med veiledere.
18. Mai	6	Fikset Abstract, litt på konklusjonen og gått gjennom introduksjon og konklusjon med Grammarly. Møte med oppdragsgiver
19. Mai	9	Siste innspurt, gått gjennom alle kilder, lest gjennom kapitlene. Fikset på ting.

