Aksel Skaar Leirvaag
Simen Ramberg
William Eide Seiner

Facial authentication service with a privacy-preserving focus

Bachelor's thesis in Digital Infrastructure and Cybersecurity
Supervisor: Raghavendra Ramachandra
May 2022

Bachelor's thesis

NTNU
Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Aksel Skaar Leirvaag
Simen Ramberg
William Eide Seiner

Facial authentication service with a privacy-preserving focus

Bachelor's thesis in Digital Infrastructure and Cybersecurity
Supervisor: Raghavendra Ramachandra
May 2022

Norwegian University of Science and Technology
Faculty of Information Technology and Electrical Engineering
Dept. of Information Security and Communication Technology

NTNU
Norwegian University of
Science and Technology

Kunnskap for en bedre verden

Department of Information Security and Communication Technology

DCSG2900 - Bachelor Thesis Bachelor of Science in Digital Infrastructure and Cyber Security

Bachelor Report

Author:

Aksel Skaar Leirvaag, William Eide Seiner, Simen Ramberg

May, 2022

Abstract

Mobai wanted to leverage their biometric services to build a OpenID Connect Provider with persisted storage of biometric data which is both secure and adheres to privacy by design principles. Throughout the project, the Norwegian Data Protection Authority privacy by design guide and Microsoft Security Development Lifecycle were utilized to provide a holistic privacy focus throughout all development stages. The system uses a microservice architecture and utilizes Kubernetes for orchestrating the containers. In order to secure and protect the users biometric data, a biometric template protection algorithm based on bloom filters was implemented. Finally, a risk assessment of the system was conducted, which identified current risks and potential mitigation. The result is a prototype which demonstrates the opportunities of using an authentication system which leverages biometrics as the method of authentication, in addition to fulfilling the strict privacy regulations imposed by General Data Protection Regulation (GDPR).

Sammendrag

Mobai ønsket å utnytte sine biometriske tjenester til å bygge en OpenID Connect-leverandør med vedvarende lagring av biometriske data som både er sikkert og overholder prinsipper for personvern. Gjennom hele prosjektet ble Datatilsynets prinsipper for innebygd personvern og Microsoft Security Development Lifecycle benyttet for å gi et helhetlig personvernfokus gjennom alle utviklingstrinn. Systemet bruker en mikrotjeneste-arkitektur og bruker Kubernetes for å orkestrere containerne. For å sikre og beskytte brukernes biometriske data, ble en biometrisk malbeskyttelsesalgoritme basert på bloomfiltre implementert. Til slutt ble det foretatt en risikovurdering av systemet, som identifiserte aktuelle risikoer og potensielle tiltak for reduksjon av risikoen. Resultatet er en prototype som demonstrerer mulighetene av å bruke et autentiseringssystem som benytter biometri som autentiseringsfaktor, i tillegg til å oppfylle de strenge personvernforskriftene som er pålagt av General Data Protection Regulation (GDPR).

Preface

We would like to thank Martin Stokkenes from Mobai and our supervisor Raghavendra Ramachandra for providing us with valuable input during our weekly meetings and for giving us the opportunity to work on an engaging and challenging bachelor's thesis. We are also grateful to Gaute Bjørklund Wangen for providing feedback on the risk assessment. Finally, we would like to thank the group for their excellent cooperation throughout the project.

Table of Contents

List of Figures

List of Tables

Glossary

Go A programming language. 33, 35, 36, 44, 46, 50

Godoc Go documentation standard and tool. 33

Helm Package manager for Kubernetes. 35

Kind A tool for running local Kubernetes clusters using Docker container. 35

Kubernetes Automated container orchestration platform. i, 2, 25, 35, 36, 44, 53, 75

OAuth 2.0 An authorization framework. xi, 10, 12, 13, 26, 27, 36

Acronyms

API Application Programming Interface. 18, 53

APT Advanced Persistent Threat. 59, 64–67, 129, 130

BTP Biometric Template Protection. 13, 14, 78

CERN European Organization for Nuclear Research. 3

CIA Confidentiality, Integrity, and Availability. 56, 58

CLI Command-line Interface. 36

CSRF Cross-Site Request Forgery. 72

CSS Cascading Style Sheets. 44, 50

DNS Domain Name System. 35

DPA Norwegian Data Protection Authority. i, viii, 5, 21, 26, 45, 77, 79

DPIA Data Protection Impact Assessment. viii, 78, 79

ECDSA Elliptic Curve Digital Signature Algorithm. 12

EU European Union. 13

FAQ Frequently Asked Questions. 23

FERET Facial Recognition Technology. 48

FQDN Fully Qualified Domain Name. 35

GDPR General Data Protection Regulation. i, 8, 13, 16, 23, 56, 58, 64, 78

HD Hamming Distance. 48, 49

HE Homomorphic Encryption. 78

HMAC Hash-based Message Authentication Code. 12

HTML HyperText Markup Language. 44, 50

HTTP Hypertext Transfer Protocol. 36, 42, 43

IEC International Electrotechnical Commission. 13, 24, 46

IP Internet Protocol. 35

ISO International Organization for Standardization. 13, 24, 46

JWT JSON Web Token. 12, 13, 53

KSN Kaspersky Security Network. 54

NFR Non-Functional Requirements. 15, 16

NTNU Norwegian University of Science and Technology. 1, 28, 37, 51, 56, 57, 64, 75, 83

OIDC OpenID Connect. i, x, 1, 2, 4, 12, 13, 17, 24, 26, 27, 36, 50–52, 77, 79

OP OpenID Provider. 17, 36, 77

OWASP Open Web Application Security Project. 14, 63

PII Personal Identifiable Information. 24, 64–67

PKCE Proof Key for Code Exchange. 72

PNG Portable Network Graphics. 46

RP Relaying Party. 79

RSA Rivest–Shamir–Adleman. 12

SDK Software Development Kit. 1, 26, 27

SDL Microsoft Security Development Lifecycle. i, 5, 20, 77

SPA Single-Page Application. 44

SQL Structured Query Language. 44

SSR Server-side Rendering. 44

SWOT Strengths, Weaknesses, Opportunities and Threats. x, 54, 55

TLS Transport Layer Security. 24

ToS Terms of Service. 36

UI User Interface. 7, 27–30, 36, 38, 40, 41, 44, 50, 75, 79

URI Uniform Resource Identifier. 11, 36

URL Uniform Resource Locator. 6

UUID Universal Unique Identifier. 45

UX User Experience. 44, 50

WIP Work In Progress. 4, 5, 75

1 Introduction

1.1 Background

Weak passwords are responsible for over 80% of successful data breaches, therefore, passwords pose a significant risk for both companies and individuals[1]. Another issue related to passwords is that over 40% reuse their passwords across websites[2]. The consequence of this is that users might use the same password for sensitive services and this is especially concerning when passwords are reused on websites which lacks security measures. A single point of failure for a user's security would be a breach on one website, since an attacker would have access to all of the user's registered accounts through the reused password. Using biometrics as the authentication factor, instead of passwords, can mitigate these risks[3]. Additionally, as compared to passwords, it will improve end-user usability since biometrics are something that you are, while passwords are something that you know. What you recall can be unwillingly forgotten, and even be shared with unwanted parties.

Mobai is a spin-off company from the Norwegian University of Science and Technology (NTNU), and has several biometric components that are used for facial recognition and attack detection. Their mission is to democratize biometrics, enhance security for all, and protect users' privacy. Currently they have developed Software Development Kits (SDKs) for Android and iOS, but are lacking in other technologies which can more easily commercialize their biometric products. By implementing a widely accepted standard, other businesses can more easily incorporate biometrics as the authentication factor, reducing the risk associated with passwords while improving security and end-user experience.

1.2 Project description

Mobai wants a face authentication prototype as well as to study and experiment with techniques that will contribute to good security and privacy. The work entails developing a prototype that leverages face biometrics as the authentication method, and uses the OpenID Connect protocol, or something similar, to share user's data to 3rd parties. To accomplish this, the solution must manage the storage of a

[1] FIDO Alliance. *What is FIDO?*. URL: https://fidoalliance.org/what-is-fido/ (visited on 13/01/2022).

[2] Philip Nyblom et al. *The Root Causes of Compromised Accounts at the University*. July 2019.

[3] FIDO Alliance. *The Case for Replacing Passwords with Biometrics*. URL: https://fidoalliance.org/wp-content/uploads/2014/12/3.pdf (visited on 13/01/2022).

reference template. Other goals will be particular implementation mechanisms and architecture to promote security and privacy.

1.3 Project goals

Main goals

- Develop a prototype that implements the OpenID Connect (OIDC) protocol using the Authorization Code Flow.
- Develop a demo showcasing the OIDC server capabilities.
- The authentication process is intuitive and fast to use.
- Minimize technical implementation for customers.
- Secure and privacy preserving by default.

Stretch goals

- Create secure Kubernetes configuration for deployment.
- Implement readiness and liveliness probes for Kubernetes deployment.
- Research other privacy preserving technical techniques.
- Penetration test of the services.

1.4 Constraints

The user will be authenticated using a selfie image, and comparing it with an image saved from a first-time registration. The service for handling this registration will be outside of the scope of the project. The process of registering the customer that wants to utilize Mobai's OIDC service will also not be considered.

1.5 Group background

The group consists of three students at the Institute for Information Security and Communication Technology, who studies Digital Infrastructure and Cybersecurity. Two of the members followed the same course for the entire three years, whereas one spent one year at European Organization for Nuclear Research (CERN) between year two and three. In addition, the person has also worked two years part-time for Mobai and has experience with the programming language Go and the React. The remaining group members have a deeper understanding of databases and risk management.

1.6 Project Organization

In order to create a common understanding of the project's organization and structure, the team has outlined specific roles and domains for each member.

1.6.1 Project roles

The team members are responsible for certain areas to ensure quality is maintained throughout the project. All members are considered to have the role as a developer and contributor to the bachelor thesis. The following roles and its responsibilities are defined below:

Project leader : Aksel Skaar Leirvaag

- Plan meetings and agenda.
- Handle external and internal communication.
- Ensure overall project progression.

Secretary : Simen Ramberg

- Make sure that the development model and workflow is followed.
- Write meeting minutes.

Organizer : William Seiner

- Responsible for documentation.
- Organize and maintain thesis related documents.

1.7 Domain responsibilities

Each member is expected to have an equal understanding of each domain, however, in case of confusion or disagreement within the team, the member responsible for the domain under confusion must provide facts to clear up the lack of understanding. In addition, the domain expert must organize knowledge sharing sessions, if any of the other team members lack knowledge about a domain.

- **Risk analysis** : Simen Ramberg
- **OIDC** : William Seiner
- **Go programming language** : Aksel Skaar Leirvaag

1.8 Development process

1.8.1 Development model

The duration of the project is spanning over five months, therefore, it is important to use a flexible development model which enables us to handle unexpected issues or changes. Additionally, creating incremental features will be helpful to receive early feedback from Mobai regarding if our implementation is aligned with the expected product and requirements. These traits can be found in several of the frameworks under the agile methodologies.

Scrum is a popular agile framework, however, it requires a lot of ceremonies and artifacts which can be unnecessary with regards to the size of our team. That being said, there are several of the processes, such as stand-up, retrospective and backlog refinement, which would be advantageous.

Kanban is another framework which provides a great workflow visualization tool to enhance collaboration within our team. The kanban board will be used during the stand-ups to display the tasks which are being worked on, and highlight any pains or issues together. The Work In Progress (WIP) limit will encourage us to finish each issue without starting a new one, and visualize if any tasks have been stuck in a column for too long.

Scrumban is a framework which incorporates features from both scrum and kanban. This suits the team well since we want to use the workflow mentions from scrum with a combination of the kanban board and WIP limits.

The system will process privacy sensitive data, therefore, it is important that the privacy focus is holistic. It can not be an afterthought and must be incorporated from the planning and requirements phase until final delivery. Datatilsynet, Norwegian Data Protection Authority (DPA), have created a guideline for software development with built-in privacy[4]. In the guideline they recommend using a software framework such as Microsoft Security Development Lifecycle (SDL) which will be used throughout the project[5].

Development model implementation

The kanban board will consist of the following columns. The number to the right of the column name is the defined WIP limit for the column.

Column name	**Description**
Backlog	The task is fully defined and ready for development.
Blocked (2)	The task has been started, but is blocked by either internal or external dependency.
In Progress (3)	The task is actively being worked on.
In Review (1)	All requirements are fulfilled and the task is ready to be reviewed by the team members.
Done	The task has been approved by all team members.

Table 1: Kanban Board Columns

The WIP limit for the "Blocked" column is kept at two to avoid starting a task without reflecting on any potential unknown dependencies. The dependencies of a task should be defined before beginning on the task, and unexpected scenarios should be kept at a minimum.

Asynchronous work should be avoided since it results in decreased efficiency and quality. Therefore, the limit for "In Progress" is set to three which is the amount of people in the team.

The limit for "In Review" is set to one to avoid any issues being stuck in review for too long, and unnecessary context switching between old tasks in review and the

[4] Datatilsynet. *Programvareutvikling med innebygd personvern.* 20th Aug. 2019. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/innebygd-personvern/programvareutvikling-med-innebygd-personvern (visited on 13/01/2022).

[5] Microsoft. *Simplified Implementation of Microsoft Security Development Lifecycle.* 4th Oct. 2010. URL: https://www.microsoft.com/en-us/securityengineering/sdl (visited on 13/01/2022).

task that is being worked on by the team member.

1.8.2 Meetings

Standup

Every day from Monday to Friday at 09:30 the team will conduct the stand-up. The kanban board will be used where each issue on the board will be discussed by the team member which has the ticket assigned. The duration of the meeting should be kept at 15 minutes. If any issue requires further discussion the team should agree upon a new time slot in order to conclude the conversation.

Progress meetings

There will be a weekly meeting with the supervisor Raghavendra Ramachandra and Martin Stokkenes as a representative from Mobai each Thursday at 14:00 to 14:45. During the meeting the team must present the progress and current status of the bachelor thesis.

Retrospective

Every two weeks the team will have a retrospective together. During the meeting every member will have the option to highlight achievements, frustrations and things that have gone well. Pain-points will be discussed, followed by solutions for the addressed problems.

1.9 Report layout

Footnotes are used in this report for referencing, and includes a clickable URL, which appears in the first footnote of a reference. If the reference is used more than once in the report the name of it will link to the section of the report where the reference was first used. All acronyms used in the report will have a clickable link to the list of acronyms which contains information of where an acronym is used. There are underlined words or numbers that link to the referenced sections, figures, and tables to help the reader navigate the report.

1.9.1 Report structure

Below is an overview of what the different sections contains.

Section 1: Introduction

Introduces the project by presenting the background, goals, constraints and a description of the project. In addition, it presents the group, how it is organized and a section about the development process.

Section 2: Theory

The theory presented in this part is important for gaining a better grasp of the project.

Section 3: Requirements

Describes the requirements in order to fulfil the tasks of the prototype.

Section 4: Design

Presenting the design of the software architecture, the OpenID provider and the User Interface.

Section 5: Quality assurance

A summary of how documentation and testing will be utilized in this project, as well as the tools used to assure good code quality.

Section 6: Implementation

Presents the different services developed by the group.

Section 7: Risk assessment

Introduction of identified risks, vulnerabilities and possible threat actors, including an evaluation of the assets and possible mitigations.

Section 8: Conclusion

Concluding the report with the result of the project and further work.

2 Theory

2.1 Privacy

The right to privacy includes a number of fundamental rules for protecting the privacy of users who are registered for the service. The term privacy by design ensures that the information systems deployed conform with privacy standards and protect the rights of user's data. Accountability, transparency, and risk tolerance are three factors that must be considered while developing a secure product. Privacy by design must be holistic in all development phases including requirements, design and implementation. This is central in order to cover all aspects required to build both confidence with the user and comply with regulations. In addition, businesses that value privacy foster trust, and as a result, having good privacy is a competitive advantage[6].

The Norwegian Personal Data Act, *Personopplysningsloven*, and thus also General Data Protection Regulation (GDPR) contains privacy standards all businesses must comply with. Accountability is an important aspect of the act, which forces the corporation to have a thorough knowledge of its personal data processing, as well as the implementation of technological and organizational safeguards to guarantee that the law is obeyed. This implies that each organization must do a series of assessments before collecting and using personal data in order to gain a better understanding of the information gathered. The organization must also document that they are in compliance with the law. Violation of these rules may result in punishment such as warnings, reprimands, bans or fines[7].

Transparency is an important phrase in the legislation, and it is also essential for incorporating privacy into software. Transparency in the use of personal data entails responding to inquiries about what is processed, who processes it, why, how, where, and for how long. Companies must be transparent about how they process personal data in order for the user to exercise their rights. The data subjects can therefore participate in the decision-making process, ensuring the data controller's legitimacy[8].

Privacy cannot be accomplished without good security, therefore, it is critical to

[6] Datatilsynet. *Virksomhetens plikter*. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/ (visited on 26/01/2022).

[7] Datatilsynet, *Virksomhetens plikter*.

[8] Datatilsynet. *Programvareutvikling med innebygd personvern*. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/innebygd-personvern/programvareutvikling-med-innebygd-personvern/innebygd-personvern---hva-er-det/ (visited on 26/01/2022).

have a thorough grasp of the various risks with the use of a risk assessment in order to act in accordance with the privacy obligations. To evaluate risk tolerance, the firm must conduct a risk assessment in addition to determine the impact of various accidents or scenarios in order to calculate how likely or easy it is for an unwanted event to occur. It is up to management to decide how much of a risk appetite they are willing to accept, this is also known as the risk tolerance level. The tolerance level regulates the required actions with the aim of mitigating the risk to ensure that the software does not exceed the defined tolerance level[9].

Anchoring with management is critical for making the decisions to incorporate privacy by design. In addition, management must make certain actions and allocate resources for these activities. Considering privacy throughout the development lifecycle is both less expensive and more efficient than updating pre-existing software. Businesses who fail to comply with privacy standards risk incurring considerable expenses, both in the terms of fines and in the loss of their reputation. As a result, it is essential to integrate privacy by design, as well as transparency, accountability, and risk tolerance in decision-making[10].

2.2 Authentication

Authentication is essential for protecting the confidentiality of the data. Passwords are the most common way to identify a user, but this method of identification has several weaknesses that can be exploited. An authentication system compares given data to validated user information recorded in a database. In conventional systems, this information are usernames and passwords. Compared to authentication by passwords, biometric information is characterized as physical or behavioral features[11].

Biometric authentication is accomplished by comparing the physical aspect the user presents against a reference stored in the database. Types of different biometrics are retina, iris, facial, fingerprint, or palm prints. One concern with biometrics is the security features behind the sensitivity levels of the comparison. This is based on how similar the facial recognition has to be compared to the stored sample. It can be difficult to set the correct sensitivity because a too low sensitivity level may match several other physical features, while a sensitivity level that is too high may reject requests from otherwise genuine match[12].

[9] Datatilsynet, *Programvareutvikling med innebygd personvern.*

[10] Datatilsynet, *Programvareutvikling med innebygd personvern.*

[11] OneLogin. *Biometric authentication, the good, the bad, and the ugly.* URL: https://www.onelogin.com/learn/biometric-authentication (visited on 25/01/2022).

[12] Derrick Rountree. *What Is Federated Identity?* URL: https://www.sciencedirect.com/topics/

2.3 OAuth 2.0

OAuth 2.0 was created to solve the issue of resource sharing with third parties without giving away their credentials to the site that contains the resource they would like to share. A specific use-case of this would be that a person wants to share their contacts they have on a social media platform with a third party application. With OAuth 2.0 the user can set a scope, limit resource access, for the third party's privileges. The third party service can, for example, only read the user's contacts, but is denied any further access.

2.3.1 Authorization Code Flow

A flow is used to resolve an access token, which can be used to access a resource, and OAuth 2.0 supports several different types of flows to claim an access token. The prototype will focus on the authorization code flow, as described in the project goals section 1.3. In order to comprehend the flow, a set of OAuth 2.0 terminology must first be understood.

Name	**Description**
Resource Owner	The end-user who owns the protected resource.
User agent	Typically a web browser which the resource owner interacts with.
Client	A service that wants to access the resource owner's data.
Resource server	The server which can return the protected resource.
Authorization server	Authenticates the resource owner, displays scope request from client and issues access token on approval.

Table 2: OAuth 2.0 terminology

The flow of the authorization code grant is depicted in the diagram below, and the capitalized letters are further described below the illustration.

computer-science/biometric-authentication (visited on 25/01/2022).

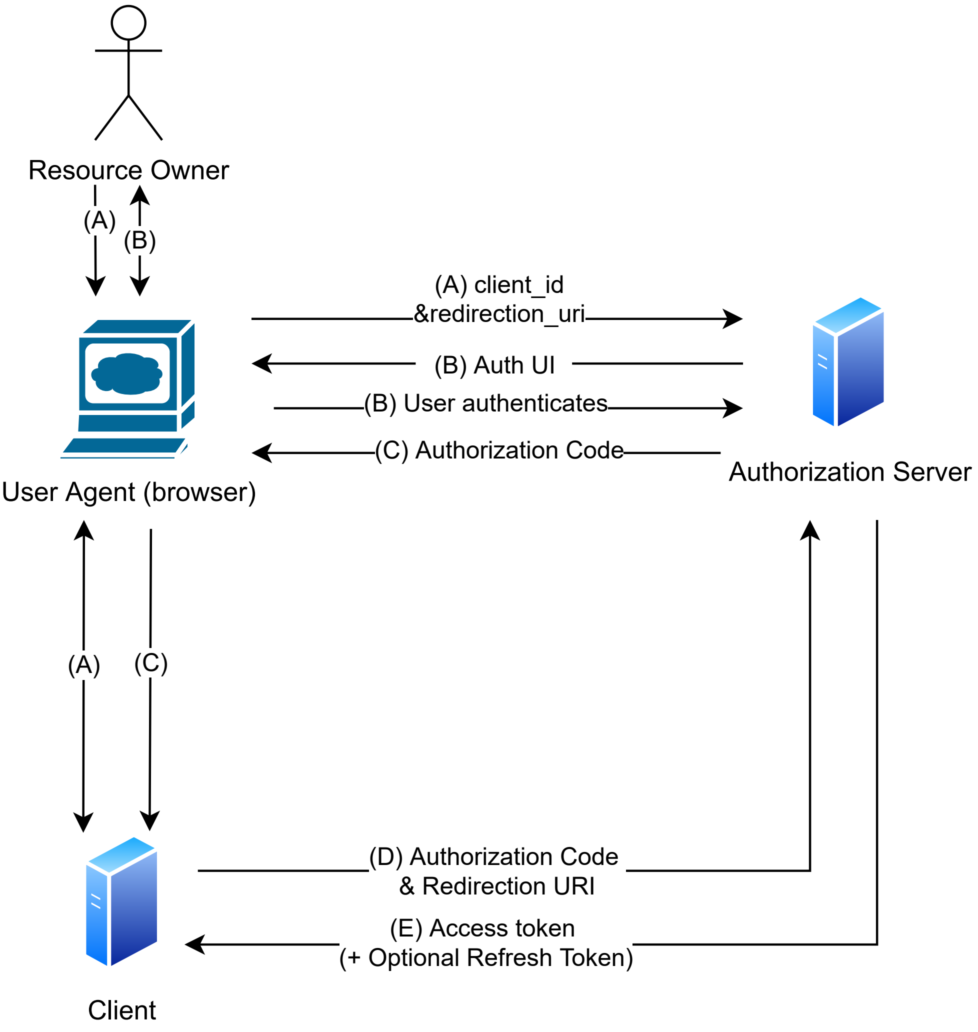


Figure 1: Authorization Code Flow

(A) The resource owner utilizes the user agent to send a request to the client. The client sets the client id and Uniform Resource Identifier (URI) in the query parameters, and then redirects the user-agent to the authorization server.

(B) The resource owner is presented with the authentication and consent screen. The authorization server verifies the authentication and if the claims were accepted by the resource owner.

(C) The authorization server redirects the user-agent with the redirection URI given by client at step (A) back to the client.

(D) The client exchanges the authorization code with the authorization server. The redirection URI used to retrieve the authorization code is included in the request, and also authenticates itself with the authorization server.

(E) The authorization server validates that the authorization code is valid. Verifies that the redirect URI from step (A) is the same that the code was issued for, and lastly authenticates the client. On success, it returns the access token and the optional refresh token.

2.4 OpenID Connect

While OAuth 2.0 solves the delegated authorization problem very well, it does not handle authentication. To solve the authentication issue, OpenID Connect was built as a thin layer on top of OAuth 2.0. OpenID Connect incorporates the authorization features from OAuth 2.0 inside the protocol itself. The specification consists of the *Core* and *OAuth 2.0 Multiple Response Types* documents which are required and 8 other documents which are optional[13]. A common use case of OIDC is to handle single sign-on across different applications, which is often used by social networks providing sign-on buttons to conveniently provide authentication on a third-party application.

The main features used by OIDC to solve the identity layer, and differentiate it from OAuth 2.0 are the ID token, user info endpoint and a standard set of scopes. The ID token provides information about the user that has been authenticated. If a service requires more information than what is stored in the ID token, then it can request that at the user info endpoint. The only difference of the authorization code flow in OAuth 2.0 and OIDC is that in the scope value the *OIDC* keyword is included and in addition of receiving the access token it also includes the ID token.

2.5 JSON Web Token

JSON Web Token (JWT) is an open standard for securely transferring information between parties. JWTs can be signed using a secret, with the Hash-based Message Authentication Code (HMAC) algorithm, or a public/private key pair utilizing Rivest–Shamir–Adleman (RSA) or Elliptic Curve Digital Signature Algorithm (ECDSA)[14]. Signed tokens can verify the integrity of the claims contained within it, while encrypted tokens hide those claims from other parties[15].

[13] OpenID-Foundation. *OpenID connect welcome page.* URL: https://openid.net/connect/ (visited on 22/01/2022).

[14] Auth0. *Introduction to JSON Web Tokens.* URL: https://jwt.io/introduction (visited on 24/01/2022).

[15] Auth0, *Introduction to JSON Web Tokens*.

In the OAuth 2.0 specification the format of the access token is outside of the specification scope, but JWT is the most common format and is also described in the companion specification[16][17]. The ID Token data structure is the most important addition that OpenID Connect provides to OAuth 2.0 in order to allow end-users to be authenticated[18]. The ID Token is a token that holds claims regarding authentication of an end-user when utilizing a client, and also other potentially desired claims[19]. JWTs are used as this because of their simplicity and portability.

2.6 Biometric template protection

The system will process and store biometric data that will be used to authenticate users. Since biometric information is used for identification, it can be utilized by criminals, among other things, to steal identities. Furthermore, it is categorized as sensitive data by the European Union (EU) under the GDPR 2016/679. As a result, it is critical to protect the subjects' privacy rights.

In order to create secure biometric systems and comply with the regulation, the International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) 24745 has defined irreversibility and unlinkability as two essential criteria for protecting biometric information[20]. If an illicit actor manage to capture a biometric template they should not be able to revert the template back to the original biometric sample. Unlinkability refers to the ability for a single biometric sample to generate different protected templates, making it impossible to tell whether they belong to the same person. In addition, if an actor gains possession of one biometric template, it cannot be utilized to identify a person in another biometric system. Both of them are important characteristics to have in case of the template is leaked or intercepted.

The main Biometric Template Protection (BTP) approaches are biometric cryptosystems, homomorphic encryption, cancelable biometrics, and hybrid methods. Sandhya et al.[21] conducted a thorough literature study that examined the current state

[16] IETF-Trust and D. Hardt. *The OAuth 2.0 Authorization Framework.* URL: https://datatracker.ietf.org/doc/html/rfc6749 (visited on 24/01/2022).

[17] M.B. Jones and D. Hardt. *The OAuth 2.0 Authorization Framework: Bearer Token Usage.* URL: https://datatracker.ietf.org/doc/html/rfc6750 (visited on 24/01/2022).

[18] N. Sakimura et al. *OpenID Connect Core 1.0 incorporating errata set 1.* URL: https://openid.net/specs/openid-connect-core-1_0.html (visited on 24/01/2022).

[19] Sakimura et al., *OpenID Connect Core 1.0 incorporating errata set 1*.

[20] ISO/IEC. *24745:2011, Security Techniques, Biometric information protection.* 2011.

[21] Mulagala Sandhya and Munaga V. N. K. Prasad. 'Biometric Template Protection: A Systematic Literature Review of Approaches and Modalities'. In: *Biometric Security and Privacy: Opportunities & Challenges in The Big Data Era.* Ed. by Richard Jiang et al. Springer International

of BTP approaches.

2.6.1 Bloom filters

One of the reliable BTP that was investigated was from the *Multi-biometric template protection based on bloom filters* paper[22]. The paper present a new methodology for estimating the parameters used for a Bloom filter-based protection scheme. Because the hash function and the parameters set will produce collisions, the Bloom filters provide irreversibility. Furthermore, it achieves unlinkability by employing structure-preserving feature re-arrangement via data regrouping and permutation.

Mobai's face recognition uses a deep learning approach which produces an array of 512 floating point numbers. The bloom filter template protection algorithm requires the input data to be in a binarized format, which implies that the output from Mobai's biometric engine must be converted from a floating point number to a binary value. A consequence of this transformation, will result in loss of information since a binarized format only has two potential values compared to floating points.

2.7 OWASP top 10

Open Web Application Security Project (OWASP) top 10 is an awareness document that lists the most well-known web application vulnerabilities, and it is something that most web developers should utilize to develop safe apps. OWASP not only lists the vulnerabilities, but also the most typical causes for exploitation and how to prevent them[23].

Publishing, 2017, pp. 323–370. ISBN: 978-3-319-47301-7. DOI: 10.1007/978-3-319-47301-7_14. URL: https://doi.org/10.1007/978-3-319-47301-7_14.

[22] Marta Gomez-Barrero et al. 'Multi-biometric template protection based on bloom filters'. In: *Information Fusion* 42 (2018), pp. 37–50. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2017.10.003. URL: https://www.sciencedirect.com/science/article/pii/S1566253516301233.

[23] Open Web Application Security Project(OWASP) foundation. *OWASP Top Ten*. Oct. 2021. URL: https://owasp.org/www-project-top-ten/ (visited on 08/04/2022).

3 Requirements

3.1 Non-functional requirements

The utility of a software system is defined primarily by its functionality aswell as non-functional features such as usability, privacy, performance, maintainability, and security[24]. The importance of Non-Functional Requirements (NFR) is to gain a broader overview of which NFR's that have to be implemented through developing the functional requirements. The NFR's should serve as the foundation for determining which functional requirements that must be addressed.

The diagram below shows a variety of NFRs that will be evaluated for the development of this prototype. It will explain the significance of the many NFRs that will be assessed using the traffic light approach. After discussing the importance of the NFRs with Mobai, it was defined that the green requirements are the most significant, followed by yellow, and finally red.

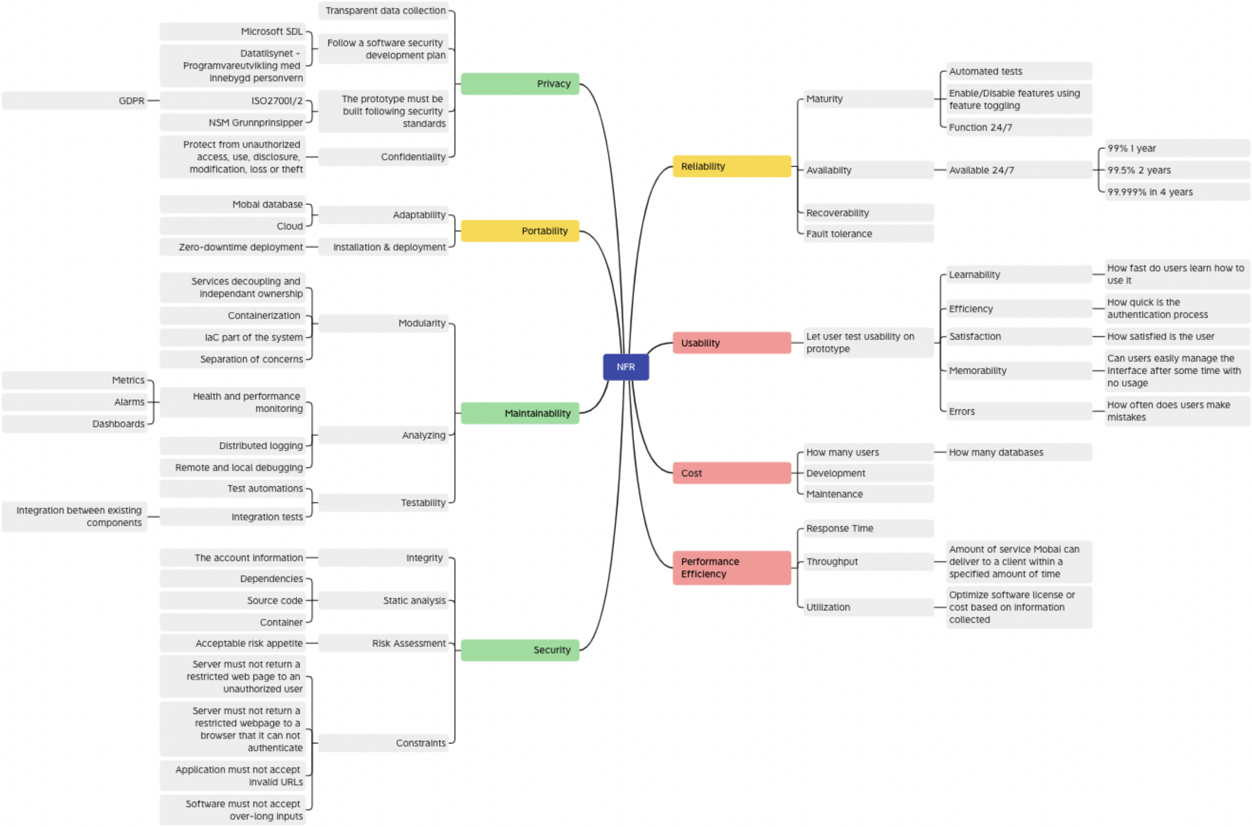

NFR
Privacy
Transparent data collection
Follow a software security development plan
Microsoft SDL
Datatilsynet - Programvareutvikling med innebygd personvern
The prototype must be built following security standards
ISO27001/2
NSM Grunnprinsipper
GDPR
Confidentiality
Protect from unauthorized access, use, disclosure, modification, loss or theft
Portability
Adaptability
Mobai database
Cloud
Installation & deployment
Zero-downtime deployment
Maintainability
Modularity
Services decoupling and independant ownership
Containerization
IaC part of the system
Separation of concerns
Analyzing
Health and performance monitoring
Metrics
Alarms
Dashboards
Distributed logging
Remote and local debugging
Testability
Test automations
Integration tests
Integration between existing components
Security
Integrity
The account information
Static analysis
Dependencies
Source code
Container
Risk Assessment
Acceptable risk appetite
Constraints
Server must not return a restricted web page to an unauthorized user
Server must not return a restricted webpage to a browser that it can not authenticate
Application must not accept invalid URLs
Software must not accept over-long inputs
Reliability
Maturity
Automated tests
Enable/Disable features using feature toggling
Function 24/7
Availabilty
Available 24/7
99% 1 year
99.5% 2 years
99.999% in 4 years
Recoverability
Fault tolerance
Usability
Let user test usability on prototype
Learnability
How fast do users learn how to use it
Efficiency
How quick is the authentication process
Satisfaction
How satisfied is the user
Memorability
Can users easily manage the interface after some time with no usage
Errors
How often does users make mistakes
Cost
How many users
How many databases
Development
Maintenance
Performance Efficiency
Response Time
Throughput
Amount of service Mobai can deliver to a client within a specified amount of time
Utilization
Optimize software license or cost based on information collected

Figure 2: Non-functional requirements

[24] L. Chung and J.C.S.D.P Leite. *On Non-Functional Requirements in Software Engineering.* URL: http://ce.sharif.edu/courses/96-97/1/ce475-1/resources/root/NonFunctionalRequirements.pdf (visited on 26/01/2022).

Mobai emphasized that the most significant components of these concepts are security and privacy, followed closely by maintainability. This is due to the fact that it incorporates various crucial components for developing a functional prototype that complies with privacy and GDPR standards, as well as designing a modular application with an emphasis on separation of concerns. This entails creating a software architecture with specific responsibilities in order to avoid a monolithic design. This enables Mobai to work on the delivered prototype more efficiently in the future since each component will have independent duties that can be changed without affecting other components.

With privacy as a NFR, the prototype and its functional requirements will be focused on complying to the necessary set of NFRs in order to manage sensitive personal information as well as biometric information. The security aspects of the NFRs ensures the user that their data is secure and appropriate precautions have been made to identify threats against Mobai that could damage the integrity of the service. The prototype must therefore follow the best standards in both cryptography and other security practices.

The third most essential NFR for this prototype is maintainability, as this includes a number of aspects that Mobai has to follow in order to be able to release the final product. It addresses various key factors such as a modular architecture, documentation and test suites to make refactors and rewrites easier in the future. The documentation will make the hand-over of the prototype easier as it will help to explain both various design decisions and how to use the software. Mobai will most likely need to extend or rewrite parts of the application, therefore, having tests to verify that the expected behavior did not change will be essential for taking the product to market.

Because portability and reliability does not influence the prototype in the way the green categorized NFRs does, they are categorized as yellow. On the other hand, it will ensure quick deployment and product maturity, making it easier to develop the product for Mobai later.

Finally, the NFRs categorized as red had little influence and were deemed insignificant by Mobai for the prototype. These are usability, cost, and performance efficiency, as this will be a proof of concept application which requires further work in the future and investing time into the usability and performance optimization is seen as less important than the other previously mentioned aspects. It is worth noting that they are still highly important for Mobai's future development.

3.2 Functional requirements

The functional requirements are a collection of definitions for the various behaviors that the system must fulfill. The developed prototype must have the following functional requirements:

Functional requirement	**Description**
OIDC compliant	The prototype must be compliant with the OIDC specification.
Facial biometric as authentication factor	Mobai's face based biometric services must be used for authenticating a user.
1 : 1 comparison (verification)	The user authentication must compare a probe image to a stored reference image.
Read persisted data	A user has the ability to read what kind of information Mobai has stored for them.
Delete persisted data	A user can delete their own data.
Resource sharing to third-parties	If a resource owner grants access to a third party, the prototype can share the requested resources.
Client management	A client needs to be registered for a company to use Mobai's OpenID Provider (OP).
Email as the user identifier	The user's email address will be utilized to locate the correct reference image.

Table 3: Mandatory Functional requirements

Mobai has specified a list of recommended requirements in addition to the mandatory functional requirements described above.

Functional requirement	**Description**
Biometric template protection	The biometric data persisted should be saved as a protected biometric template to increase the privacy and security of users.
Fuzz testing	The API endpoints and appropriate functions should be fuzz tested.
Monitoring	The prototype should be instrumented with a Prometheus metric endpoint.
Automated image capturing	The images required to authenticate the user should be captured without any user interaction.
Liveness and readiness probes	The prototype should have endpoints for liveness and readiness probes sent by Kubernetes.

Table 4: Functional requirements - Recommended from Mobai

3.2.1 Use Case Diagram

The use case diagram describes how the flow operates as well as providing the opportunity to identify the actors and what functionality each of them have access to.

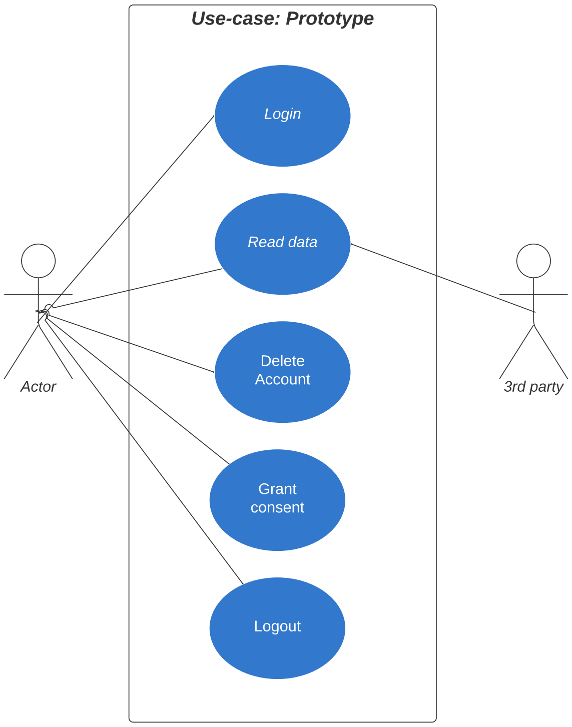


Figure 3: Use Case Diagram

The purpose of this diagram is to demonstrate how the system works from a high-level perspective. The model enables gathering system demands in addition to obtaining a wider knowledge of the prototype's external view. The use case diagram has been created by analyzing the system in order to obtain all the functionality required to implement the various use cases. Mobai is not included in this diagram as an actor because it is a prototype and no admin page has been developed. As a result, Mobai does not have any functionalities represented in this use case model.

3.2.2 User Stories

User stories are an Agile software development method that allows for communicating requirements in a more informal way. They are composed of one or two sentences written from the perspective of the user and characterize the user's desire. As a result, it allows the requirements to be communicated more effectively[25].

User story 1

The police wants to issue a renewed passport and needs to verify an identity.

[25] Visual Paradigm. *What is User Story?* URL: https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/ (visited on 31/01/2022).

User story 2

A user wishes to disclose their nationality with a third-party app that requires complete nationality verification.

User story 3

A social media platform needs to verify an identity in order to reset a locked account.

User story 4

Cryptocurrency exchange needs to comply with regulations and identity proof its customers without storing the user's passports.

User story 5

The user wants to control their sensitive health data and explicitly grant health personnel access to it.

3.3 Security requirements

The second practice of the Microsoft Security Development Lifecycle (SDL) is a security requirements analysis, which defines the application's minimal security needs for the intended environment. After a conversation with Mobai, the following set of security requirements were established:

- A risk assessment will be conducted for this project, which can be read in section 7.
- To secure the quality and security of dependencies and source code, static analysis tools will be used.
- OWASP top 10 will be covered by implementing measures to protect against the most popular web application security risks.
- Biometric template protection is not a hard requirement for this project, but is needed before the product goes into production.

3.4 Quality gates and bug bars

The third practice of the Microsoft Security Development Lifecycle is a definition of the quality gates which defines the accepted amount of security and privacy risk.

Bug bars will act as an enforcer of the established quality gates. Mobai did not prefer to set a defined test coverage percentage, but required tests where it is necessary. For the prototype the following pipeline jobs will be utilized, and all of them must pass in order to merge to main branch. These jobs are an effort to avoid any low-hanging vulnerabilities in the code.

- Go tests
- React tests
- Golangci-lint
- Gosec
- ESLint

3.5 Privacy requirements

The Norwegian Data Protection Authority (DPA) has numerous requirements related to privacy in their *"Programvareutvikling med innebygd personvern"* guide[26]. One of the criteria is the application's justification for obtaining sensitive information. This prototype will not handle user enrollment, but one possibility is providing a justification at that stage by asking for the user's approval on registration. In this scenario, it is important to notify them in plain language, and the user must comprehend what they are agreeing to. Furthermore, if the user wishes to withdraw the consent, it must be as simple as when they gave consent in the first place. However, this is not a necessity if the application is used in a circumstance where the gathering of biometric data is permitted by law. This ultimate judgment must be evaluated and made by someone who has legal expertise.

Another important facet of user registration is correctly informing the user about the type of data that Mobai handles and stores. It could be a good idea to describe the biometric template protection properties to users in order to increase understanding about how biometric data is stored, and build trust between Mobai and the user. Furthermore, in terms of trust and regulatory obligations, a thorough explanation to both users and customers of Mobai's reasons of collecting sensitive data is crucial.

Mobai must also address additional challenges, such as the long-term storage of sensitive information. Because of the nature of the application, the system must keep the data for an extended period of time and does not remove it by default.

[26] Datatilsynet, *Programvareutvikling med innebygd personvern*.

They could investigate options such as deleting inactive users to decrease the risk of unnecessary storage and automatically removing individuals with expired passports.

Additionally, evaluating where they will operate the software is important, since they must ensure that the provider and other vendors are in compliance with data processing agreements. Choosing a Norwegian infrastructure supplier would also assist to comply with legal regulations. When sensitive information is modified or removed, sufficient logging must be in place to achieve traceability in the event of a security breach. These logs, however, cannot include sensitive information and must rely on using a pseudo anonymous identifier that can be traced back to the user.

4 Design

4.1 Privacy

4.1.1 Informing

Informing the users about how the application works and how personal information is handled is an important part of privacy design. This will be accomplished through the use of terms of service that informs the users of how their data will be processed and what data will be saved. When the prototype is ready for production, Mobai should make a Frequently Asked Questions (FAQ) page and provide contact information to make it easy for users to ask questions about their privacy concerns. In addition, the prototype will feature icons, as well as a plain language to make it clear what kind of data the users are sharing.

During the registration and consent flow, the user must actively acknowledge that they have read and agreed to the terms of service. This will be enforced on the consent page by requiring the user to click on a checkbox, as this is considered an active action. To make it easier for users to understand what data they have shared, the prototype will consist of a view which lists the third-parties and which data they have granted them access to.

4.1.2 Control

Users should be in control of their own data, therefore, the prototype will have features which gives the user the ability to view, update and delete data. Because the information comes from passports, the capacity to modify it will be limited. Users will also be able to withdraw their permission to a third party once they have granted it, prohibiting them from accessing the data.

4.1.3 Data minimization

An important aspect in good software privacy design is data minimization, and is mentioned in Article 5(1)(c) of the GDPR[27]. It states that the data collection must be "adequate, relevant and limited to what is necessary in relation to the

[27] European Union. *General Data Protection Regulation.* URL: https://gdpr.eu/ (visited on 21/02/2022).

purposes for which they are processed"[28]. As previously stated, this prototype will not manage user enrollment and thus will not decide on what data that is acquired during registration. A passport will most likely be the primary data collecting asset for Mobai on a registration. However, this does not mean that Mobai has to collect all the data available in a passport.

They should be mindful of the data minimization principle, and the quantity of data retained should be dependent on the context. An option could be to allow the user to control what data they would like to share. Mobai, for example, can use the user's passport to do identity verification, but only store their face template and email in the database. As a result, it would minimize the privacy impact for the user. The trade off, of this approach is the increased complexity, as Mobai can not guarantee that all users have a set of data to a third party application. Having said that, the outward statement of Mobai's privacy commitment to customers may be worth the complexity cost.

4.1.4 Transfer and storage

The biometric data will be stored in a biometric template protection format which follows the principles described in ISO/IEC 24745 and further discussed in section 2.6. However, the system will store other Personal Identifiable Information (PII) and sensitive data such as name, gender, birthday and social security number. PostgreSQL supports numerous types of encryption options which can all contribute to minimize the consequences in case of a database breach[29]. When sending data over the network it will automatically be encrypted by Transport Layer Security (TLS) when using the PostgreSQL Operator by CrunchyData[30]. When the server is receiving the data it will then decrypt it and potentially leak the information. To solve this issue, client-side encryption, in which data is encrypted and decrypted at the user's device, would be beneficial to the user's privacy. This, however, will not function since it contradicts the data flow of the OIDC specification and would require persisted storage at the client. A middle ground is encrypting the persisted data at rest and is supported by most large infrastructure vendors, but this will not be done for this prototype as it is highly dependent on the underlying storage provider in the Kubernetes cluster.

[28] Union, *General Data Protection Regulation*.

[29] PostgreSQL. *Encryption Options*. URL: https://www.postgresql.org/docs/current/encryption-options.html (visited on 21/02/2022).

[30] Crunchydata. *PostgreSQL Operator*. URL: https://github.com/CrunchyData/postgres-operator (visited on 21/02/2022).

4.2 Software architecture

Figure 4 depicts a high-level overview of all the various components that are necessary to operate the system. All of these components must be deployed in order for the system to function; however, only Hydra Companion, Clients Overview, Template Protector, OpenID Relay, and User Store will be developed for this project, as indicated by the yellow color in the diagram below.

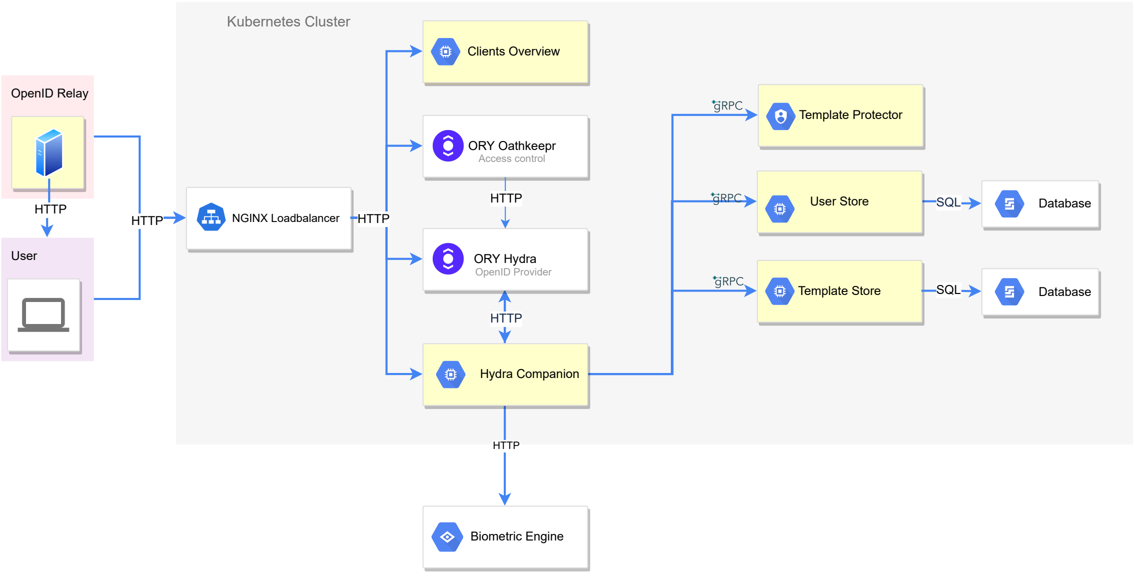


Figure 4: Software Architecture

A microservice architecture is used in order to build a loosely coupled system that facilitates modifications. This is especially important because the system will be a prototype, and new features and changes will be added in the future. The communication protocol utilized by each service is also shown. Wherever feasible, gRPC has been used for inter-server communication in order to make the contracts between each service simpler to comprehend and comply with.

Since each service is independent of each other it makes scaling horizontally much easier, especially considering Mobai uses Kubernetes internally and all the services will be containerized to make them compatible with Kubernetes. However, the downside would be the complexity of deploying each service as it can quickly result in duplication across the services. To combat this issue, a monorepo will be used to store all the different services as it can make code and configuration changes easier[31].

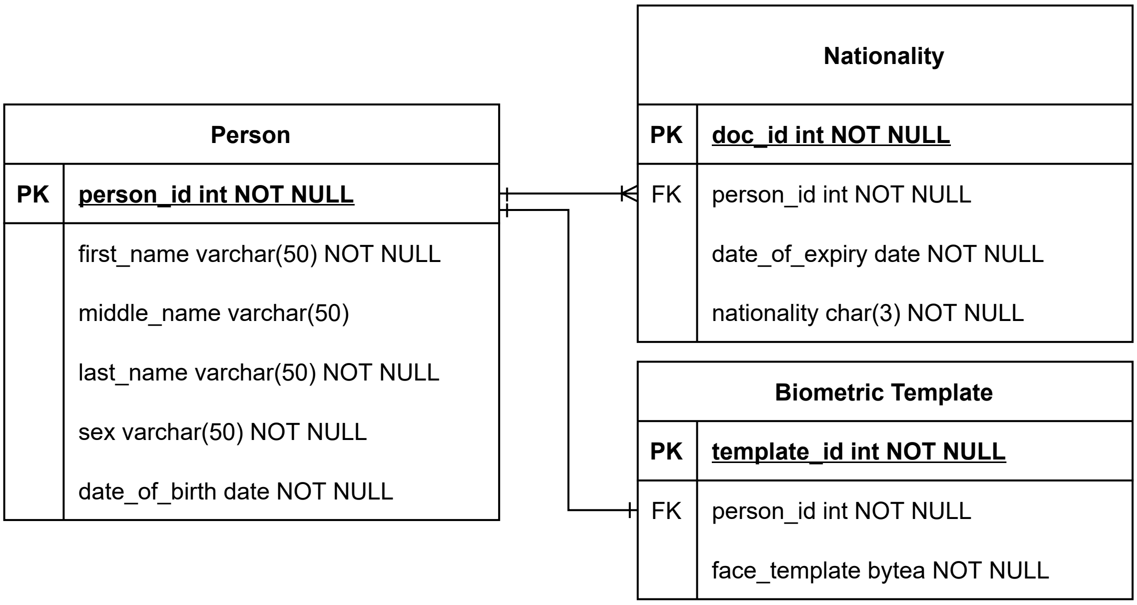
An OpenID Relay will be developed to provide a complete demonstration of the

[31] Ron Powell. *Benefits and challenges of using monorepo development practices*. URL: https://circleci.com/blog/monorepo-dev-practices/ (visited on 18/02/2022).

system's capabilities. In a production setting, this service will be developed by a third-party vendor that want to access the data of a Mobai registered user.

4.2.1 Entity relationship diagram

The entity relationship diagram illustrates how the various entities in the database model are related to one another. A person may have many nationalities while only having one biometric template. The biometric template will be stored in separate databases to further reinforce the biometric owner's security and privacy, as advised by the Norwegian Data Protection Authority (DPA)[32].



Person	
PK	**person_id int NOT NULL**
	first_name varchar(50) NOT NULL middle_name varchar(50) last_name varchar(50) NOT NULL sex varchar(50) NOT NULL date_of_birth date NOT NULL

Nationality	
PK	**doc_id int NOT NULL**
FK	person_id int NOT NULL date_of_expiry date NOT NULL nationality char(3) NOT NULL

Biometric Template	
PK	**template_id int NOT NULL**
FK	person_id int NOT NULL face_template bytea NOT NULL

Figure 5: Entity Relationship Diagram

4.3 OpenID provider

Both OpenID Connect and OAuth 2.0 are regarded as standards that are both complex and difficult to understand in terms of implementation, both from the perspective of the developer who consumes the provider and those who design them. The group had three alternatives for implementing the protocol. Either implement everything from scratch, leverage a SDK or use a decoupled OIDC provider.

The first option was to write everything from scratch, but this would take a considerable amount of time. In addition to the complexity of the standard, the system would process sensitive data, which meant that security and bugs must be kept to

[32] Datatilsynet, *Programvareutvikling med innebygd personvern*.

a minimum, which the timeframe did not allow.

The second option was to leverage a SDK which already implement most of the features, and the group could tailor the provider to match our requirements. The requirements, however, did not correspond to the necessity for a customized implementation of the OIDC protocol. The prototype's key objective was to use biometrics as an authentication factor while keeping the end-user's privacy in focus. The group investigated two different SDKs, Fosite and CAOS's oidc, and found that while both would minimize the complexity, it would still take some time to implement it correctly[33][34].

Initially the group started working on implementing the standard using a SDK, and then found a product named Hydra created by ORY, which is a decoupled OIDC and OAuth 2.0 provider[35]. It met many of the requirements since it was used in production already, OIDC certified and security was already taken care of[36]. It decoupled Hydra's login, consent, and logout responsibilities, allowing for the creation of a new service that used Mobai's biometric services to implement the authentication factor.

4.4 User interface

Hydra requires to implement three separate User Interfaces (UIs) in order to complete the authentication and authorization process. Because the product's target audience will be diverse, it is necessary that the application is minimalistic and self-explanatory to use. Furthermore, the interface must be dynamic because Mobai's customer can select their own grants, logo, and client name.

The design drafts below were created to align and confirm the expectations with Mobai and make the implementation phase easier. The final design may deviate from the initial draft. Figma, a vector graphics editor and prototyping tool, was used to create the drafts to quickly make changes and test ideas.

[33] CAOS. *Certified OpenID Connect Library (client and server) for Go.* URL: https://github.com/caos/oidc (visited on 20/02/2022).

[34] ORY. *Extensible security first OAuth 2.0 and OpenID Connect SDK for Go.* URL: https://github.com/ory/fosite (visited on 20/02/2022).

[35] ORY. *OpenID Certified™ OpenID Connect and OAuth Provider.* URL: https://github.com/ory/hydra (visited on 20/02/2022).

[36] OpenID Foundation. *OpenID Certification.* URL: https://openid.net/certification/#OPs (visited on 26/01/2022).

4.4.1 Login

The login UI consists of two different screens. The first screen will ask the user to enter their email to identify who they are. It is not necessary to use the email, as the primary goal is to have a user identifier. Other options could be a phone number or a social security number. The client which is requesting the access is displayed at the bottom of the page in addition to their logo. NTNU is used as an example in all the drafts.

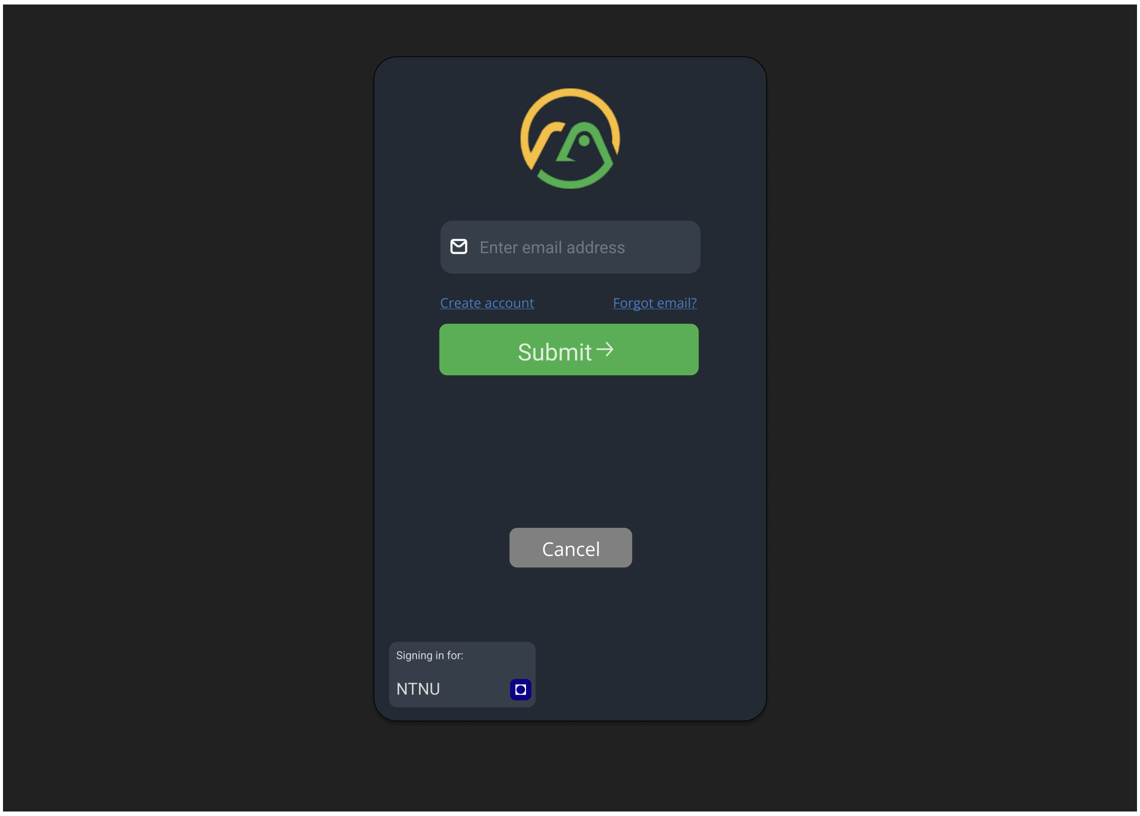


Figure 6: Login UI design - Input Email

The second screen in the login process is the selfie capture. Here the user will click the camera button and several images will be automatically taken. It is important that the user is aligned in the frame, therefore, a circle is used to navigate the user to the correct position.

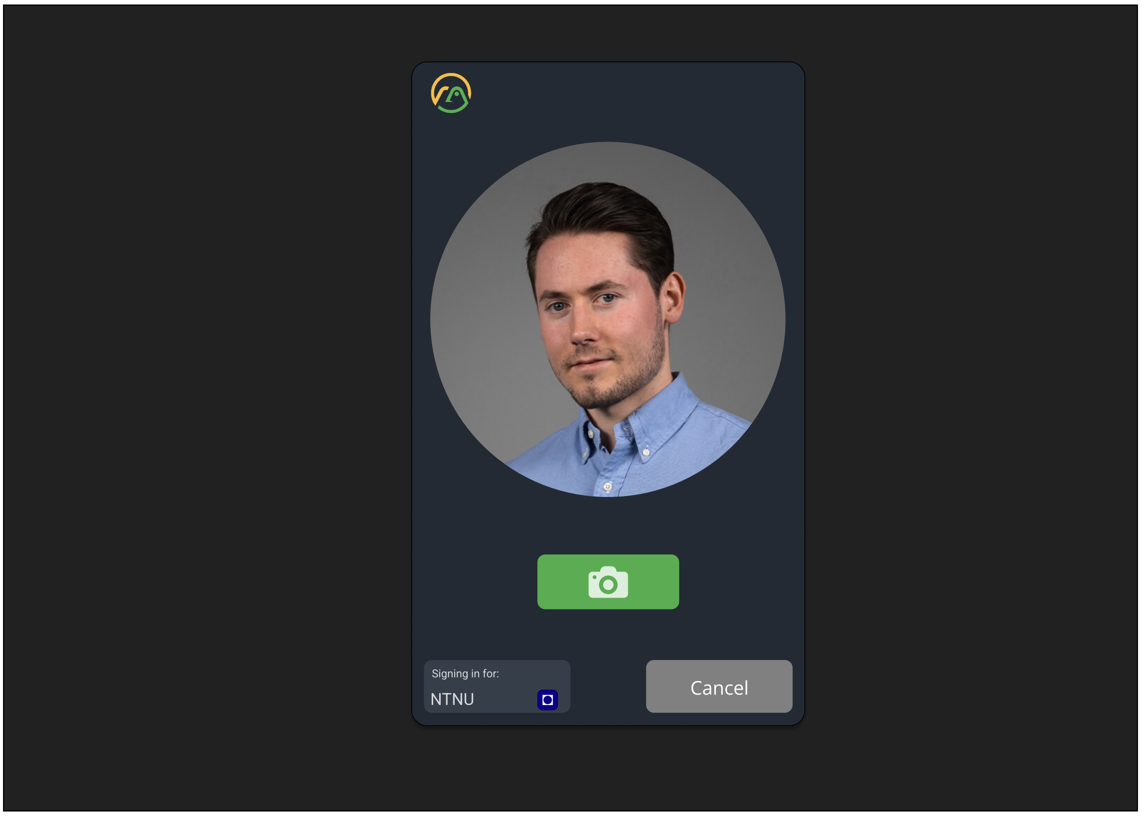
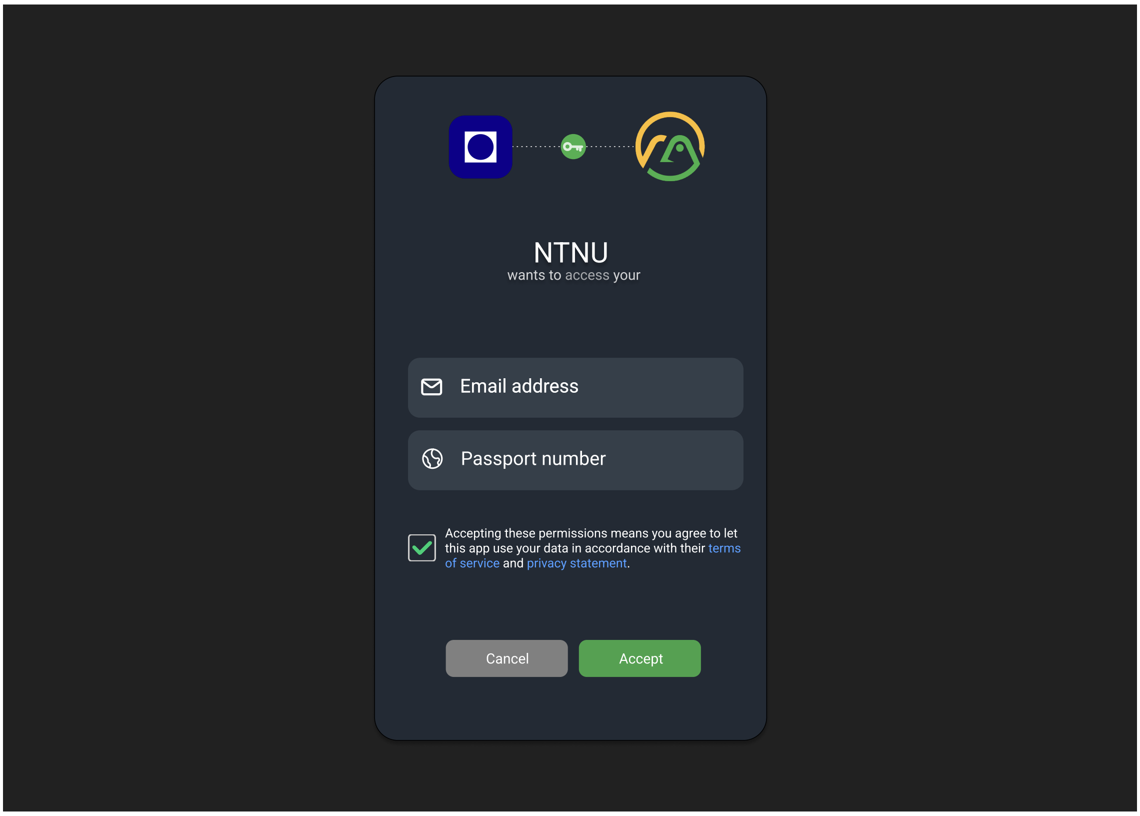


Figure 7: Login UI design - Selfie Capture

4.4.2 Consent

After a successful authentication the user will be presented with the consent UI. The logo of both the client owner and Mobai will be shown at the top. Following that, information about which clients seek access is supplied, as well as which grants they request from the resource owner. There is also included a text informing the user about the client's terms of service and privacy statement. Finally, the user has the option to accept or cancel the consent request.



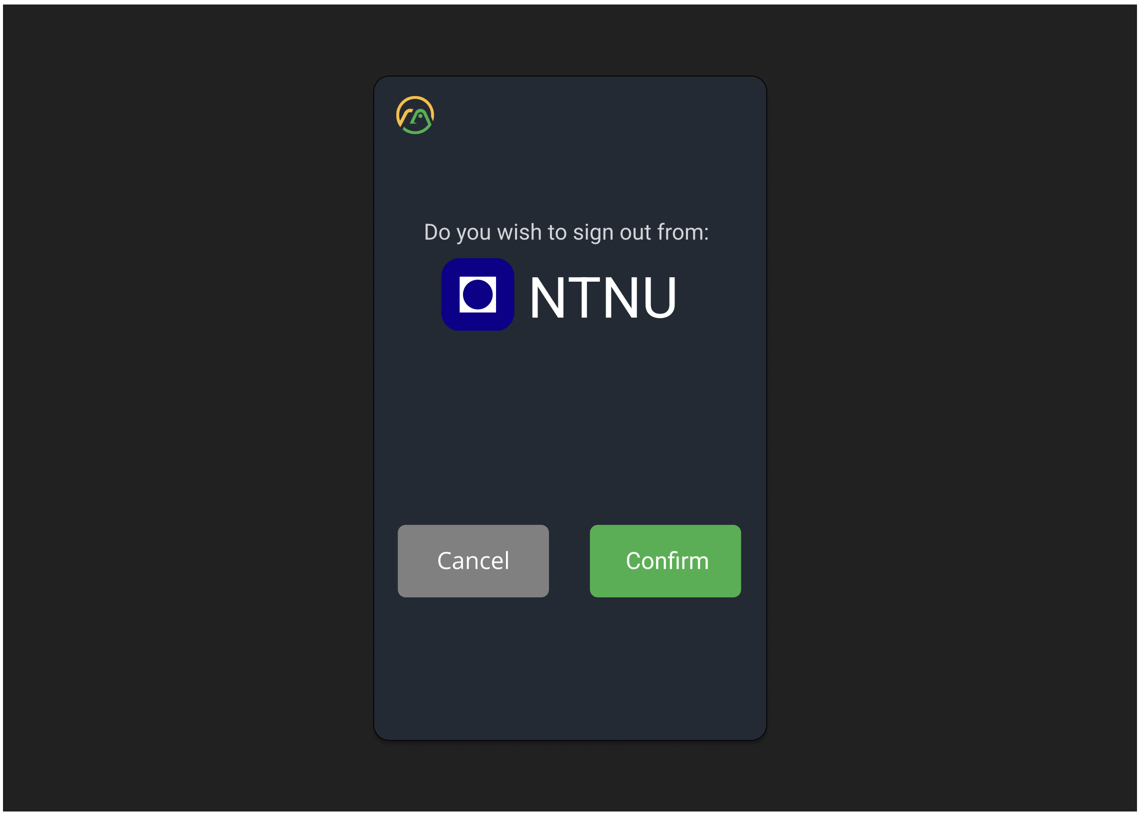
NTNU
wants to access your

☑ Accepting these permissions means you agree to let this app use your data in accordance with their terms of service and privacy statement.

Figure 8: Consent UI design

4.4.3 Logout

On sign-out the user is presented with a simple UI which asks for confirmation of signing out from the given client.


Do you wish to sign out from:

NTNU

Figure 9: Sign-out UI design

4.4.4 Client overview

As seen in figure 10, the screen lists the different third parties that the user have decided to share their data with.The next figure 11 depicts the page that appears when a user clicks the view button on one of their clients. They can see what sort of information they have shared with the client on this page, and they may revoke access if they no longer want to share their data with a third-party application.

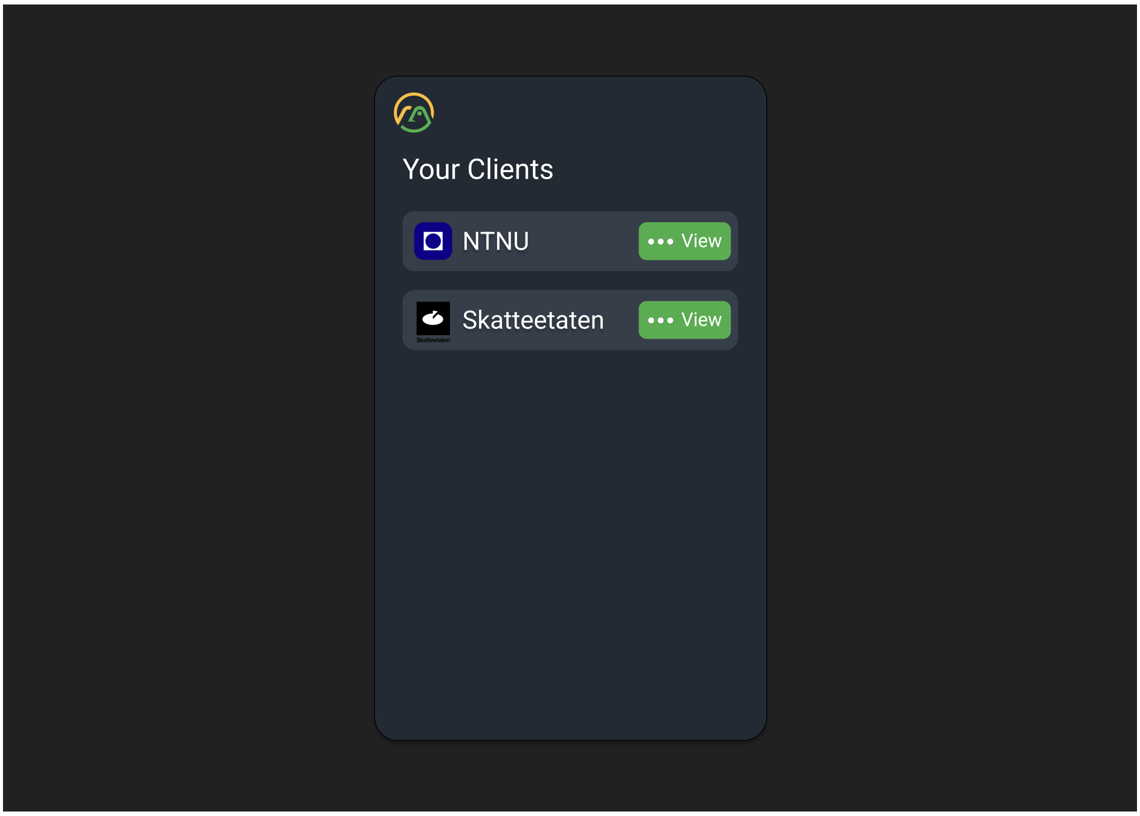


Figure 10: Client overview UI design

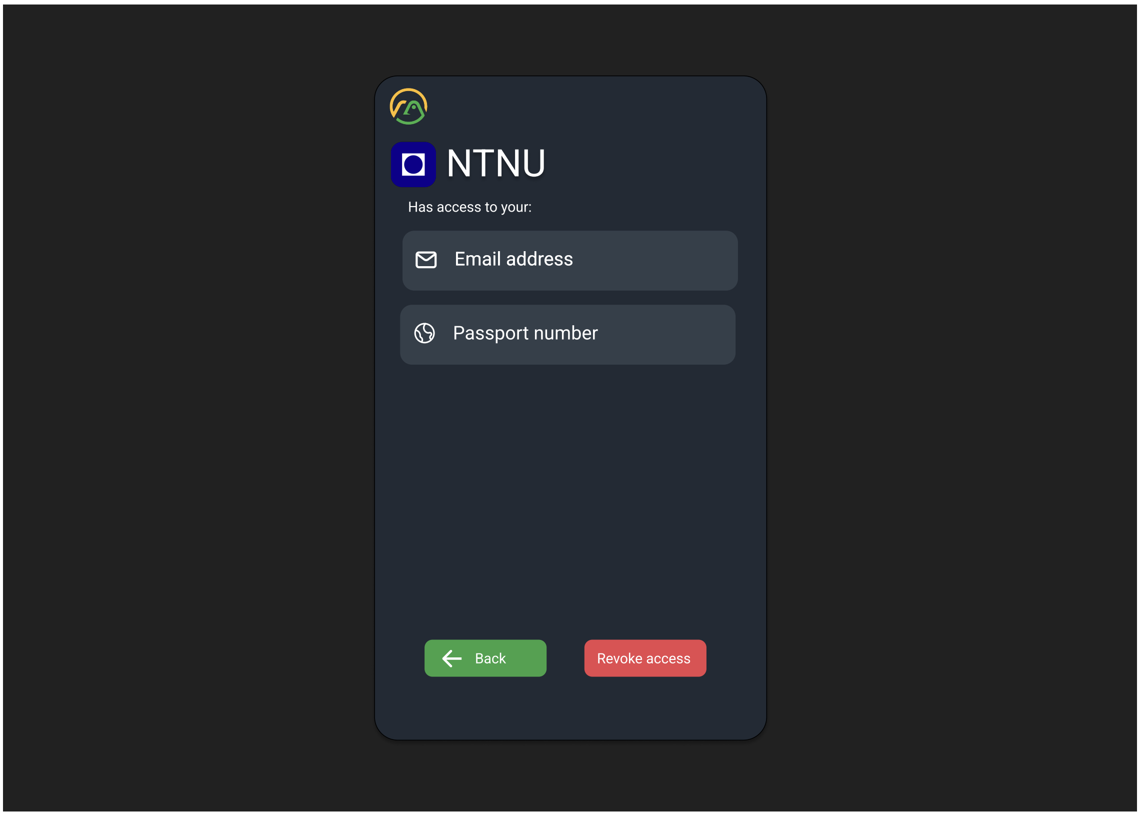


Figure 11: Client info UI design

5 Quality assurance

The end product will be used in a production environment and additional features will be added to further extend the capabilities of the solution. Good documentation, standardization and testing is therefore imperative for a smooth transition after we have concluded our defined work.

5.1 Documentation

Documentation should be produced during the development and be revised before the hand-off to Mobai. The application will be written in Go, and the team will use Godoc as the documentation tool and follow its guidelines as outlined in the article[37].

5.2 Code quality

Gofmt will be used for formatting and increasing readability of the code. A Github pipeline will be created for the repository and will use static analysis tools such as Golangci-lint, Gosec and ESLint in order to catch mistakes at an early stage of the development cycle. Figure 12 below displays a successful pipeline run.

[37] Andrew Gerrand. *Godoc*. 31st Mar. 2011. URL: https://go.dev/blog/godoc (visited on 12/01/2022).

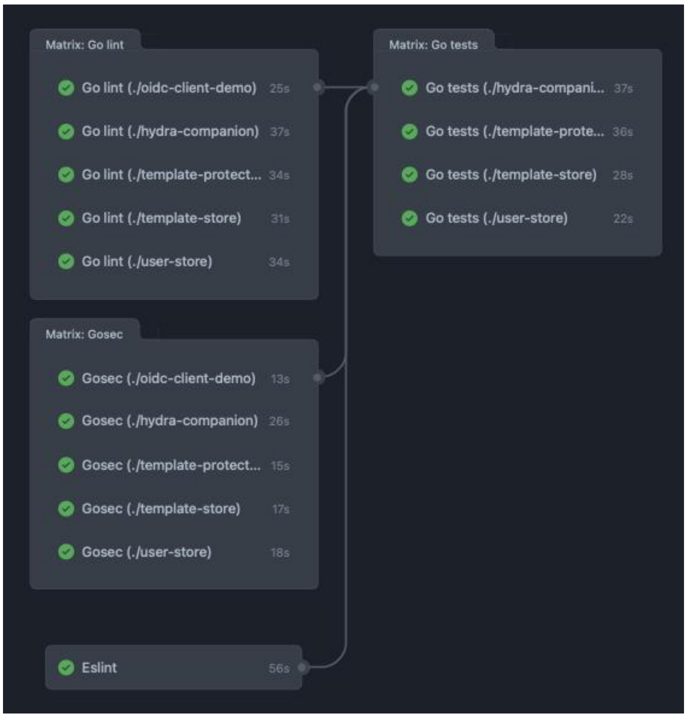

Matrix: Go lint
Go lint (./oidc-client-demo) 25s
Go lint (./hydra-companion) 37s
Go lint (./template-protect... 34s
Go lint (./template-store) 31s
Go lint (./user-store) 34s
Matrix: Go tests
Go tests (./hydra-compani... 37s
Go tests (./template-prote... 36s
Go tests (./template-store) 28s
Go tests (./user-store) 22s
Matrix: Gosec
Gosec (./oidc-client-demo) 13s
Gosec (./hydra-companion) 26s
Gosec (./template-protect... 15s
Gosec (./template-store) 17s
Gosec (./user-store) 18s
Eslint 56s

Figure 12: Github Actions pipeline

5.3 Testing

Good quality tests and coverage are essential to safeguard against any unwanted behaviours both during the development and in the future iterations. The test automation pyramid is used to focus our test coverage to where it provides the most value and efficiency[38]. The majority of our tests will be unit tests which will assert correct behavior at the function-level without too many dependencies on other parts of our code. Next the integration tests will run expectations if our different modules are working correctly together. Lastly, the acceptance tests will be used to ensure that our requirements are met. Since we are using an agile development model, features should be incremental and presentable during the meetings with Mobai. This provides us with the opportunity for early feedback.

[38] Mike Cohn. *Succeeding with Agile*. Presidio Press, 2009.

6 Implementation

6.1 Local Development

Because Mobai's container orchestration platform is Kubernetes, it was agreed upon that the configuration for third-party applications and deployment files for the created applications should be written in Kubernetes configuration from the start. The advantages of this approach are that it would make the transition from development to production smoother since it would not need any major work of recreating correct settings from, for example, using Docker Compose as our local development setup. Helm, a package manager for Kubernetes, was used to install the third-party application required for the system to function, and made it easier to adjust the apps for our needs while preserving appropriate default settings.

Mage, a Make build tool using Go, was used to create a simple abstraction around the other tools used to setup the development environment, and made it possible to deploy the environment with one simple command. Using Mage compared to Make was a decision based on that Go was already used for writing the backend services and the team was more comfortable writing Go compared to Bash. Kind was used to create the local Kubernetes cluster as it was a simple, fast and a lightweight solution. The local environment also required certain additional needs in order to function properly. Because the system utilizes a microservice architecture, as illustrated in figure 4, a method for composing the services into a coherent system was required.

As a result, an NGINX ingress controller was used to loadbalance the ingress traffic across the services using the Ingress object in Kubernetes. The Ingress configuration further required the use of a Fully Qualified Domain Name (FQDN) for the incoming traffic and the /etc/hosts file on the machine was configured to correctly resolve the Domain Name System (DNS) queries to the external Internet Protocol (IP) address of the cluster. In order for the ingress controller to receive an external IP address by Kubernetes, MetalLB, a loadbalancer implementation for baremetal clusters, was deployed. The logic of configuring it with the correct IP-range was abstracted away by the use of Mage.

Finally, Tilt was used for managing the rebuilds and deployments when doing changes to the code and configuration files. Tilt watches for changes and trigger rebuilds with the use of caching to speed up the build process. After each build the container image is pushed to a local container registry to avoid sending unnecessary traffic outside of the computer used for development. The fast build speed provided by Go was very beneficial for the development process due to the rebuilds on every change.

6.2 Hydra

Hydra was used for handling the OAuth 2.0 and OIDC logic for fulfilling the task of an OP and the decisions for choosing it can be read in section 4.3. Some of its responsibilities are handling request validation based on the provided clients configuration, issuing tokens and the cryptographics used in the protocols. Due to Hydra's architecture it required us to implement the login, consent and logout flow and which also enabled us of using biometrics as the authentication method. The application responsible for handling this logic was named Hydra Companion. On a request Hydra leverages HTTP redirection in order to correctly send the user to the correct endpoints on each respective flow, which was defined in Hydra's Helm configuration. Furthermore, a challenge was added to each request to the query parameters to make it possible for Hydra Companion to lookup information about the request and inject the information about the client into the UI.

In the beginning, a Hydra operator was used to manage the different clients configurations in order to leverage Kubernetes's reconcillation loop. However, defining a clients' ToS, privacy and logo-URI was not supported and the Command-line Interface (CLI) was used instead. This does not scale well in a production setting, and recommendation for handling the clients management can be read in section 9.2.3. Since this was only a prototype, a version of Hydra which used SQLite was used instead of a full-fledged production database, and needs to be a part of the future work before going to production.

6.3 Hydra Companion

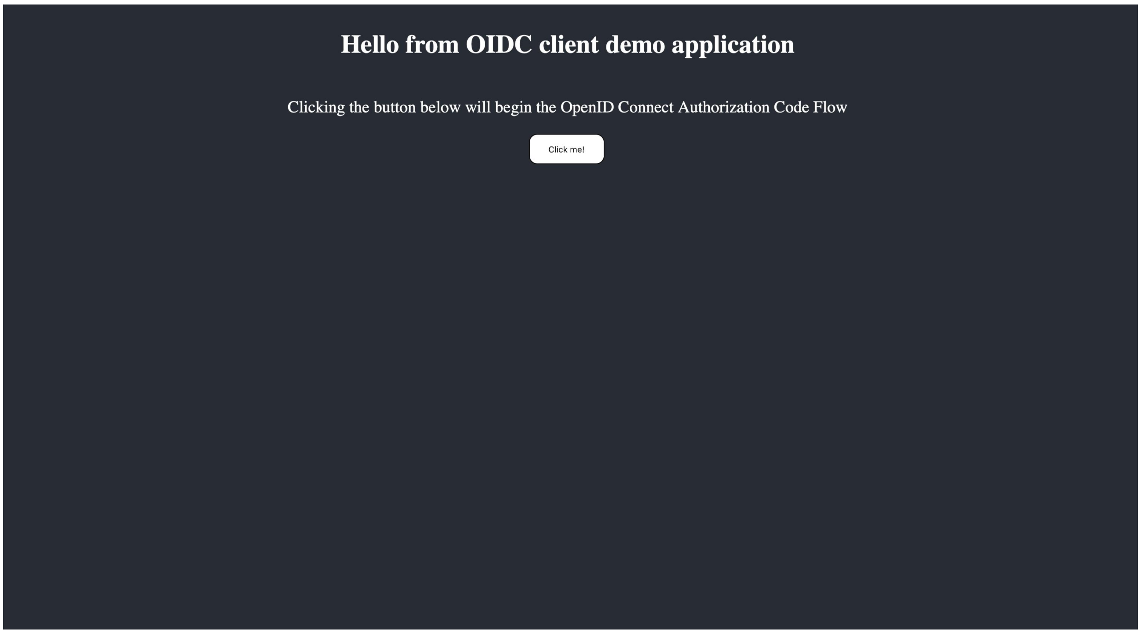
Purpose: As mentioned in section 6.2, Hydra Companion was created to implement the login, consent and logout flow as required by Hydra. It is a Go server that handles incoming HTTP redirections from Hydra, serves the UI, and responds to the POST requests from the frontend.

6.3.1 Demonstration

The images below are screenshots from the client demo created to demonstrate the Authorization Code Flow. In order to run the demo the client needs to be registered in Hydra, mock data of user and a protected biometric template which is referenced to the user needs to be stored in the databases.

Demo Welcome Page

Figure 13 is a screenshot of the welcome page of the developed client demo application. The Authorization Code Flow will be initiated when the user clicks the white button.



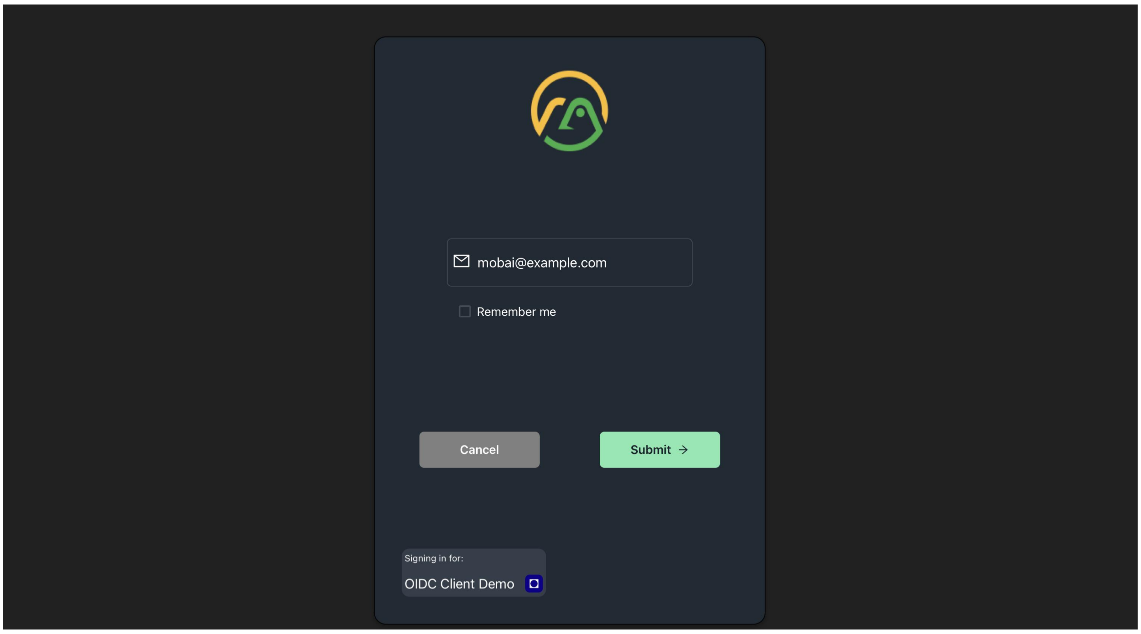
Hello from OIDC client demo application

Clicking the button below will begin the OpenID Connect Authorization Code Flow

Figure 13: Demo - Welcome Page

Email input

On click the Hydra Companion service serves the React files and injects client specific information such as the name and logo. In this case the NTNU logo is used for demonstration purposes, but can be changed based on the client registration. The user needs to enter their registered email and can also choose to be remembered which lets the user skip the authentication process for any upcoming login attempts.



mobai@example.com

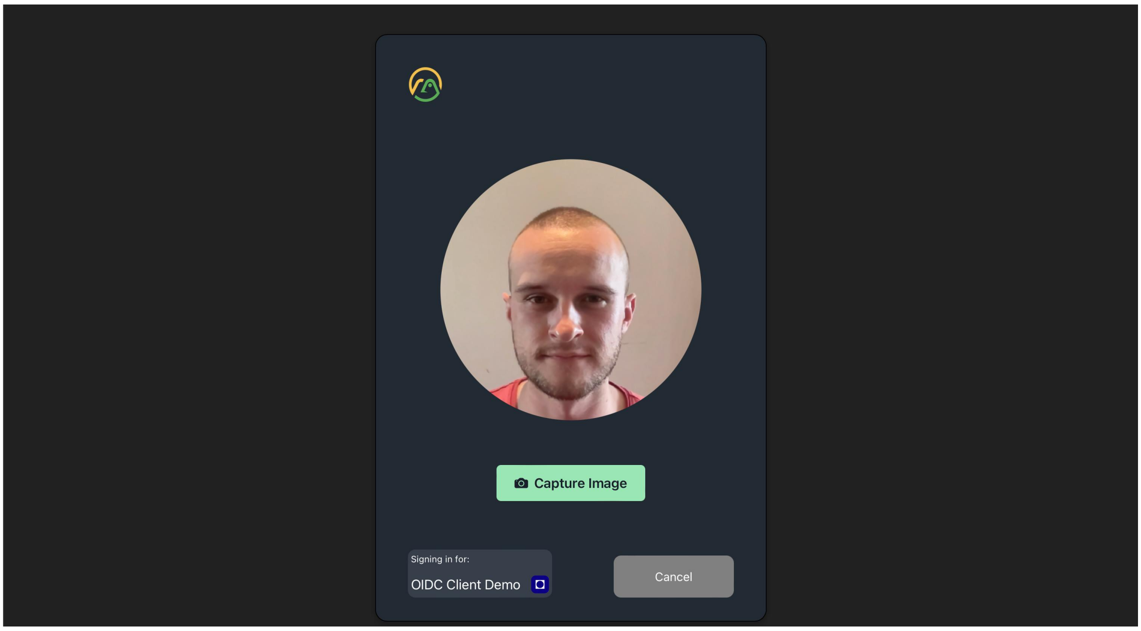
□ Remember me

Signing in for:
OIDC Client Demo

Figure 14: Demo - Email Input

Selfie capture

The last step of the authentication is to capture the selfie which will be compared to the stored reference template. The capture UI consists of a circle that is used to correctly position the user in the center of the image.

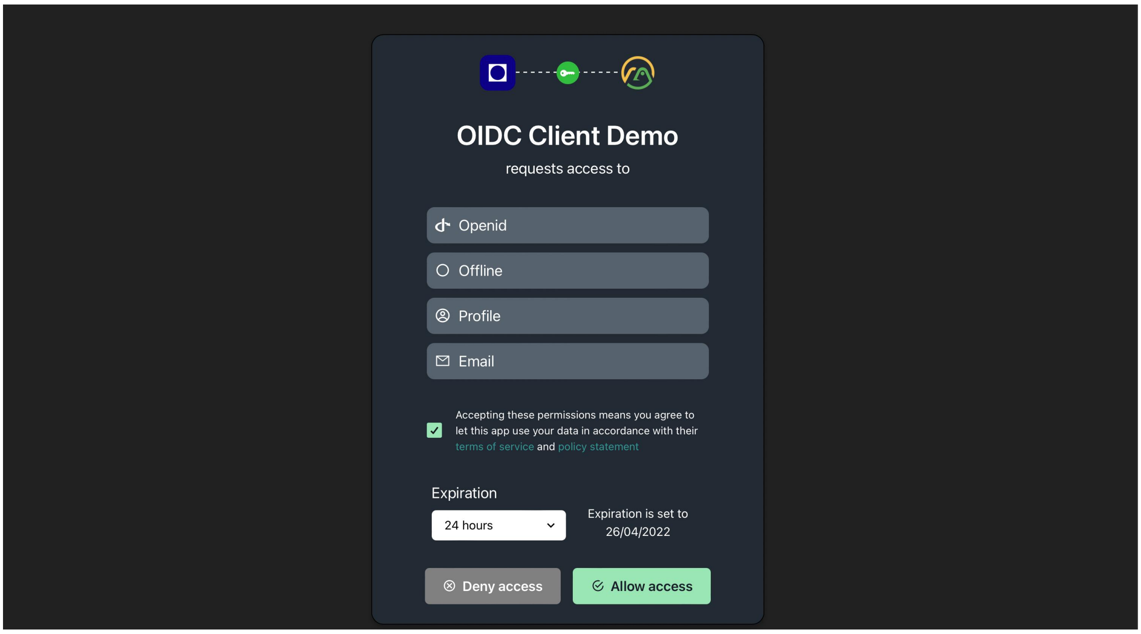


Signing in for:
OIDC Client Demo

Figure 15: Demo - Selfie Capture

Consent

The main deviation from the design, as seen in figure 8, versus the implementation is the expiration feature. This is helpful for users as it enables them to automatically remove a client's access based on their defined date. The expiration date can be one of the predefined options, or a custom date which is set using a calendar.



OIDC Client Demo

requests access to

- Openid
- Offline
- Profile
- Email

☑ Accepting these permissions means you agree to let this app use your data in accordance with their terms of service and policy statement

Expiration: 24 hours

Expiration is set to 26/04/2022

Figure 16: Demo - Consent

6.3.2 Login flow

The diagram below describes the login flow where the goal is to authenticate the user. This is accomplished by getting the user's identifier, in this prototype an email, as well as a selfie of the individual, which is then compared to the saved reference template. Both the User Store and the Template Store communicate with their respective databases, however, this is not included in the diagram for simplicity sake. Furthermore, the figure only illustrates the happy path in which no errors occur.

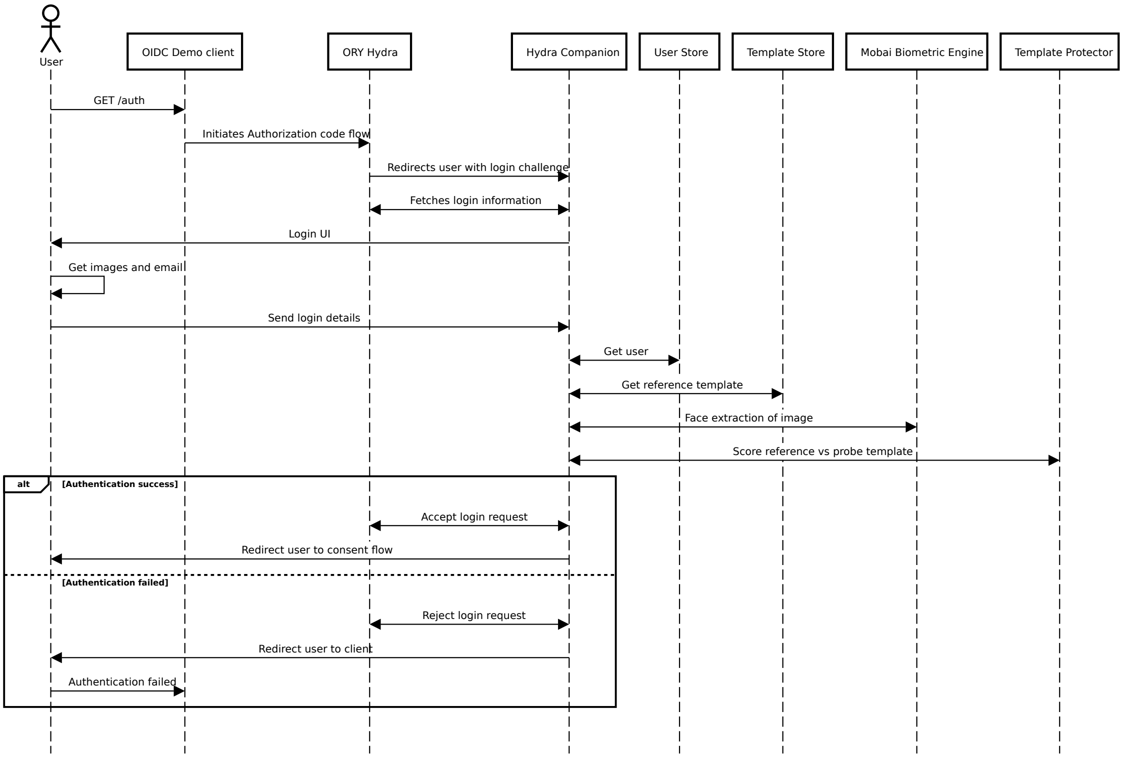


Figure 17: Sequence diagram for Login flow

6.3.3 Consent flow

The consent flow begins similarly to the login flow by obtaining some client information and injecting it into the UI. When the grants are approved, the User Store queries the database for the user's information, and the requested data, depending on the grants, is gathered and stored in the ID token, which is obtained by the client at the end of the flow.

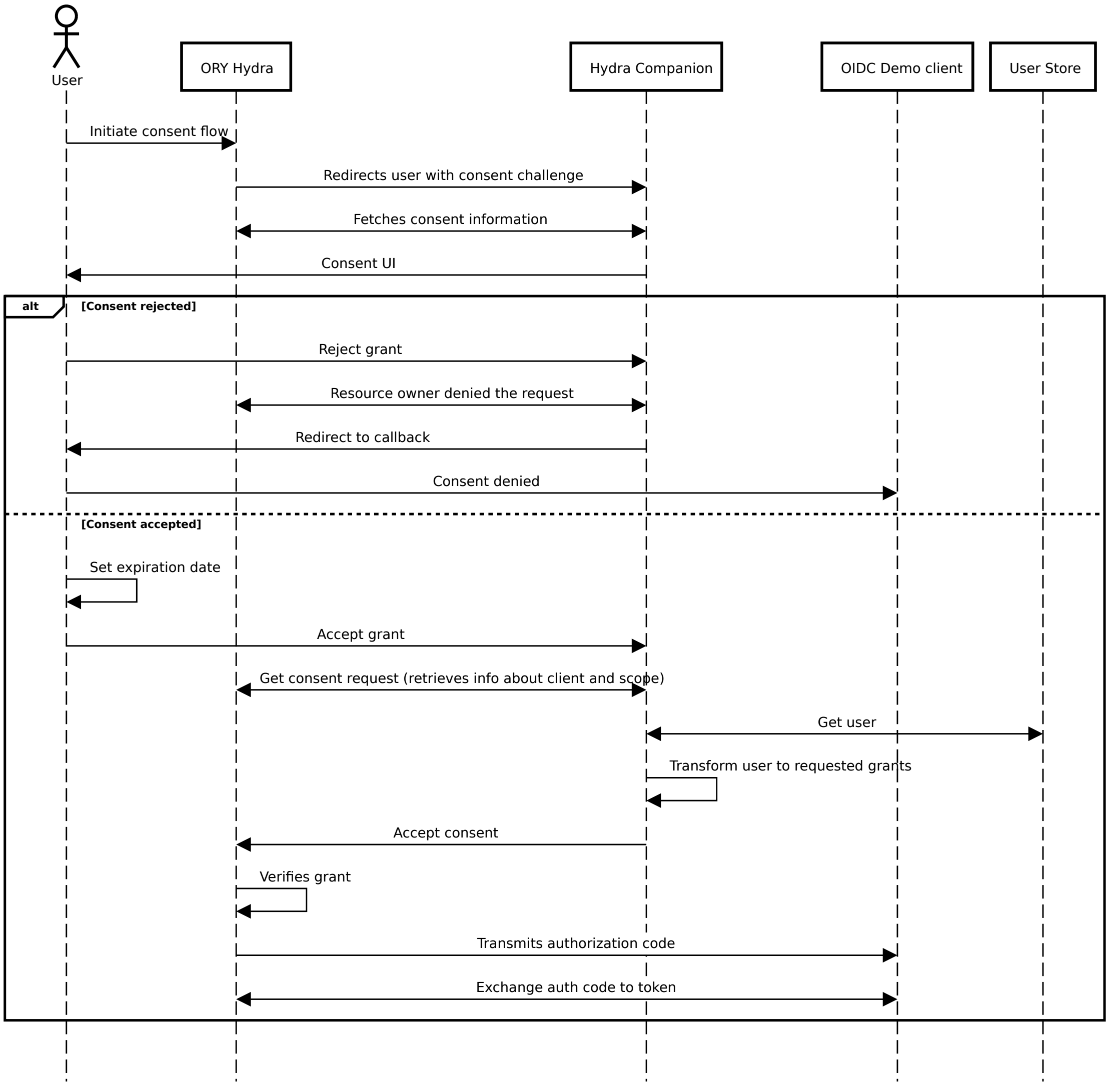

User
ORY Hydra
Hydra Companion
OIDC Demo client
User Store
Initiate consent flow
Redirects user with consent challenge
Fetches consent information
Consent UI
alt
[Consent rejected]
Reject grant
Resource owner denied the request
Redirect to callback
Consent denied
[Consent accepted]
Set expiration date
Accept grant
Get consent request (retrieves info about client and scope)
Get user
Transform user to requested grants
Accept consent
Verifies grant
Transmits authorization code
Exchange auth code to token

Figure 18: Sequence diagram for Consent flow

6.3.4 Logout flow

The logout flow starts similarly to the previous two flows, with Hydra Companion obtaining logout request information from Hydra and injecting it into the UI. If the user does not wish to logout, they are routed to their original page; otherwise, Hydra Companion signs them out and notifies Hydra. If the client has defined a post-logout callback, the user will be redirected to the specified site.

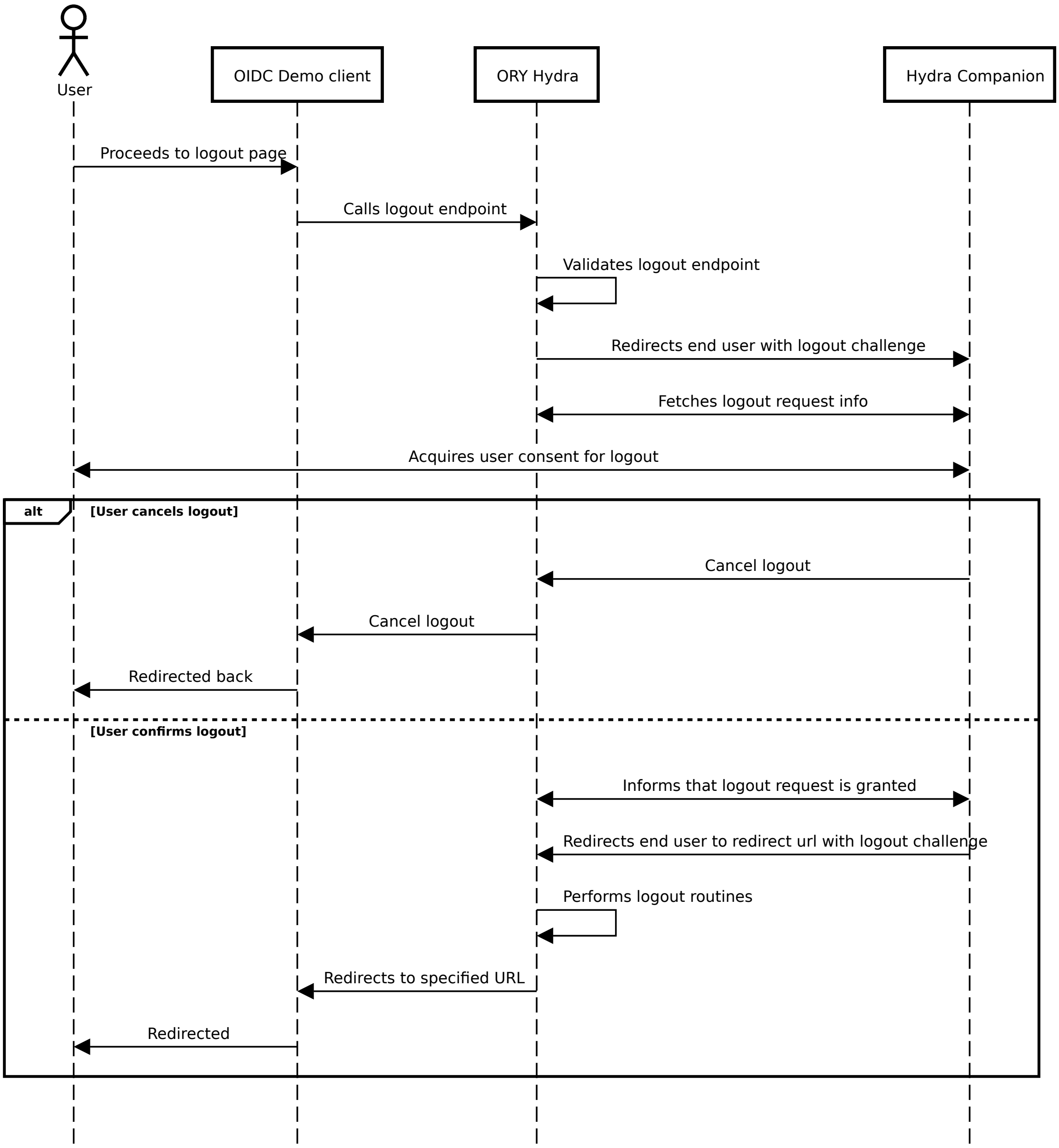

User
OIDC Demo client
ORY Hydra
Hydra Companion
Proceeds to logout page
Calls logout endpoint
Validates logout endpoint
Redirects end user with logout challenge
Fetches logout request info
Acquires user consent for logout
alt
[User cancels logout]
Cancel logout
Cancel logout
Redirected back
[User confirms logout]
Informs that logout request is granted
Redirects end user to redirect url with logout challenge
Performs logout routines
Redirects to specified URL
Redirected

Figure 19: Sequence diagram for Logout flow

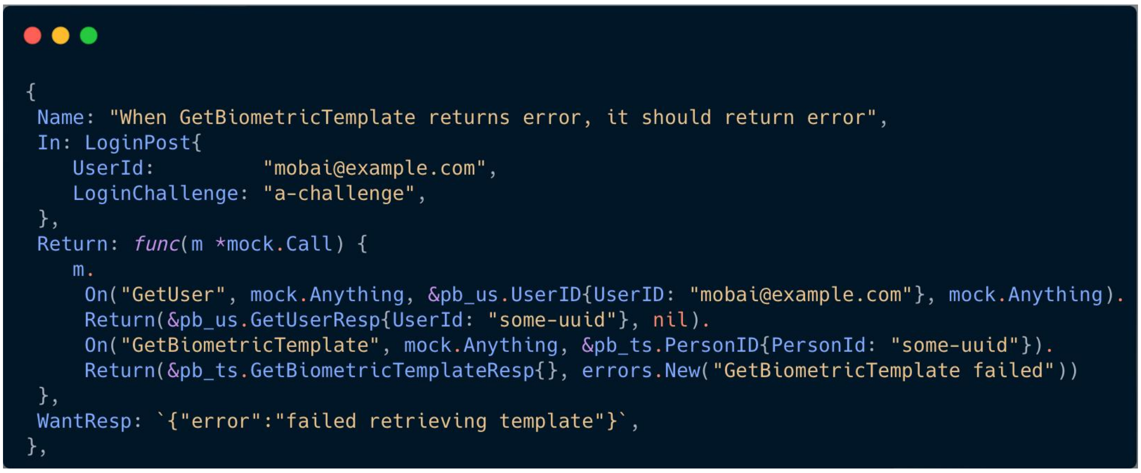
6.3.5 Face extraction

The Hydra companion sends an HTTP request to Mobai's biometric engine with a Base64 encoded picture of the individual attempting to authenticate themselves. The response is a 1x512 feature vector which is encoded as a Mat object from the OpenCV library which Mobai uses as part of the extraction. The decoding and further processing will be discussed in the Template Protector section.

An important note is that Hydra Companion does not call any endpoints on the biometric engine to do a presentation attack detection, which would attempt to detect biometric spoofing. This is due to the fact that the service does not expose any endpoint for this without performing a full face verification scoring which would add unnecessary processing time to each request. This is something that has to be included in future work, before it can be used in a production setting.

6.3.6 Testing

Hydra Companion handles most of the business logic in the system as displayed by the sequence diagrams and the software architecture figure 4, hence it is critical that it functions correctly. It has been thoroughly unit tested to ensure that the service fulfills its claims while also minimizing the number of bugs. The majority of our tests, as stated in section 5.3, should be unit tested, and in order to do this, the interfaces Hydra Companion uses to communicate with other services have been mocked using the popular testing library Testify[39]. Furthermore, table-driven tests have been used since it is the preferred way for writing tests in Go because it improves test readability while reducing boilerplate[40]. The number of unit tests in Hydra Companion totals in as 29, and an example of how it looks like can be seen in figure 20. The example is one of the tests for the function that handles POST requests for the login flow.



```
{
 Name: "When GetBiometricTemplate returns error, it should return error",
 In: LoginPost{
    UserId:         "mobai@example.com",
    LoginChallenge: "a-challenge",
 },
 Return: func(m *mock.Call) {
    m.
     On("GetUser", mock.Anything, &pb_us.UserID{UserID: "mobai@example.com"}, mock.Anything).
     Return(&pb_us.GetUserResp{UserId: "some-uuid"}, nil).
     On("GetBiometricTemplate", mock.Anything, &pb_ts.PersonID{PersonId: "some-uuid"}).
     Return(&pb_ts.GetBiometricTemplateResp{}, errors.New("GetBiometricTemplate failed"))
 },
 WantResp: `{"error":"failed retrieving template"}`,
},
```

Figure 20: Table-driven tests example

Firstly, the name of the test is defined, then the LoginPost struct, which will be used as the HTTP body. The mock is then stated in the Return field with what to anticipate as parameters to the function calls as well as what to return after each call. The method GetBiometricTemplate will return an error in this case. Finally,

[39] Stretchr. *A toolkit with common assertions and mocks that plays nicely with the standard library*. URL: https://github.com/stretchr/testify.

[40] Dave Cheney. *Prefer table driven tests*. URL: https://dave.cheney.net/2019/05/07/prefer-table-driven-tests.

at the bottom, the expected assertion is expressed. Following the definition of all tests, each test case is executed using the specified inputs, mocks, and outputs.

6.3.7 User interface

As mentioned earlier, the Go server serves the React compiled files and injects the required information for the frontend on each request. In the ORY Hydra docs it is mentioned that Single-Page Application (SPA) should not be used, and this limitation spawns from that the application cannot request additional information from the server after being rendered in the browser. This means only frontends which uses Server-side Rendering (SSR) should be used. However, Hydra Companion leverages Go's template injection functionality to insert the required data into the global window object in the index.html file on each request. As a result, it makes it possible to serve all the frontend screens using React.

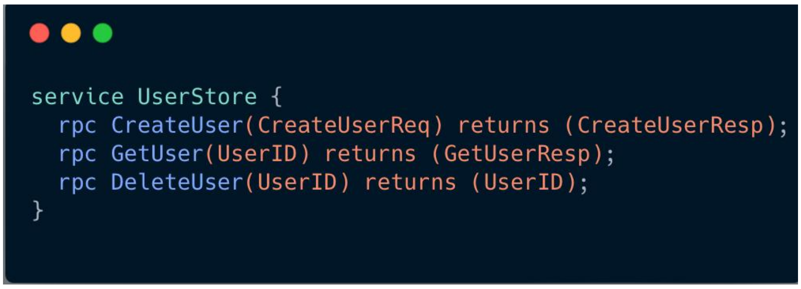
In the beginning of the implementation phase Go HTML templates was used for writing the logout and consent UI as the complexity of these screens did not require the use of React. However, it was discovered that having a consistent UI design was costly as the login UI used a component library compared to traditional Cascading Style Sheets (CSS). In the end, all the screens used Chakra UI as the component library and leveraged React to reuse the shared components across the screens. This enhanced the User Experience (UX) by using components that had a good UI design and accessibility out of the box in addition to increase the developer experience.

6.4 User Store

Purpose: The User Store service handles create, read and delete operations of user related data and persists it to a PostgreSQL database.

The required database tables as defined by the entity relationship in diagram 5 is created by an init-container in Kubernetes which executes the Structured Query Language (SQL) queries with the use of a popular database migration tool called Migrate[41]. It ensures the existing of the correct schema and also adds the mock data used for the demonstration. gRPC is used as its communication protocol and the proto file defines the following procedures.

[41] Golang-migrate. *Database migrations. CLI and Golang library.* URL: https://github.com/golang-migrate/migrate.



```
service UserStore {
  rpc CreateUser(CreateUserReq) returns (CreateUserResp);
  rpc GetUser(UserID) returns (GetUserResp);
  rpc DeleteUser(UserID) returns (UserID);
}
```

Figure 21: User Store proto file

The same proto file is used by Hydra Companion to generate the client used for communicating to the User Store server, and makes it easy to guarantee the contract between the two services. The User Store server does not expose any update functionality as such an operation needs to be carefully considered. A user should not change any of its data without the data being fully verified. The present method of updating certain data is to re-enroll, which is beyond the scope of this project.

6.5 Template Store

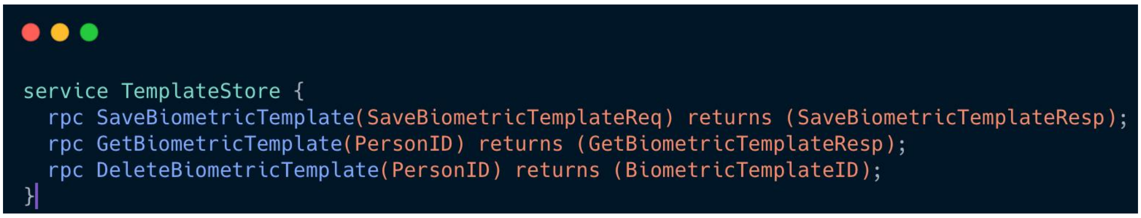
Purpose: The Template Store service handles create, read and delete operations of the protected biometric templates and persists them to a PostgreSQL database.

The only difference between it and the User Store service is what is stored. The rationale for dividing it into two services is to improve user's privacy, as recommended by the DPA[42]. Because the user's name is stored in the User Store database, determining which template belongs to a user will be more challenging. This also adheres to the principle of defense in depth by making an attacker's life more difficult with the goal of making the target less appealing[43].

The user identifier in the Template Store database is a random generated Universal Unique Identifier (UUID), which means that Hydra Companion needs to first query for the user in User Store and use the UUID retrieved there to find the corresponding template in the Template Store. The proto file defines the following procedures.

[42] Datatilsynet, *Programvareutvikling med innebygd personvern*.

[43] The National Institute of Standards and Technology. *Computer security resource center*. URL: https://csrc.nist.gov/glossary/term/defense_in_depth.



```
service TemplateStore {
  rpc SaveBiometricTemplate(SaveBiometricTemplateReq) returns (SaveBiometricTemplateResp);
  rpc GetBiometricTemplate(PersonID) returns (GetBiometricTemplateResp);
  rpc DeleteBiometricTemplate(PersonID) returns (BiometricTemplateID);
}
```

Figure 22: Template Store proto file

Template Store does not either expose an update procedure based on the same argument as mentioned in the User Store section.

6.6 Template Protector

Purpose: The Template Protector decodes the output from the Biometric Engine's feature extraction, creates protected biometric templates from a feature vector and computes the similarity score of two protected templates.

The algorithm used to generate the protected templates is based on MATLAB code provided by our supervisor, which again is based on the work discussed in theory section 2.6.1. The algorithm fulfills the ISO/IEC 24745 criterias as stated in section 2.6. Unlinkability is accomplished by specifying an integer that serves as a unique key for the system and therefore must be updated for each database in order to ensure unlinkability between them. The algorithm's irreversibility is accomplished via a series of steps which is described in detail in section 6.6.2.

6.6.1 Decoding

The output from Biometric Engine needs to be decoded before further processing. The Mat object is first Base64 decoded, then parsed as a Portable Network Graphics (PNG) with the use of Go's standard library and typecasted to the underlying type. From here the PNG's raw data is accessible without any encoding and the underlying bytes can be extracted. A float in C++, the language used for Mobai's Biometric Engine, consists of 4 bytes. Therefore, Template Protector needs to loop over the extracted bytes and break them into slices of four bytes, before finally converting each of the slices to its corresponding floating point number. After these steps the feature vector is fully decoded.

6.6.2 Bloom filter algorithm

Because a bloom filter can only function with binary data, the 512-float array must be binarized. A simple algorithm is used in which the average number is first determined and then utilized to evaluate whether a value should be a one or a zero. If the number is greater than the average, it is set to one; if it is less than the average, it is put to zero. This method will result in a data loss, which will be discussed more in the benchmark section 6.6.4.

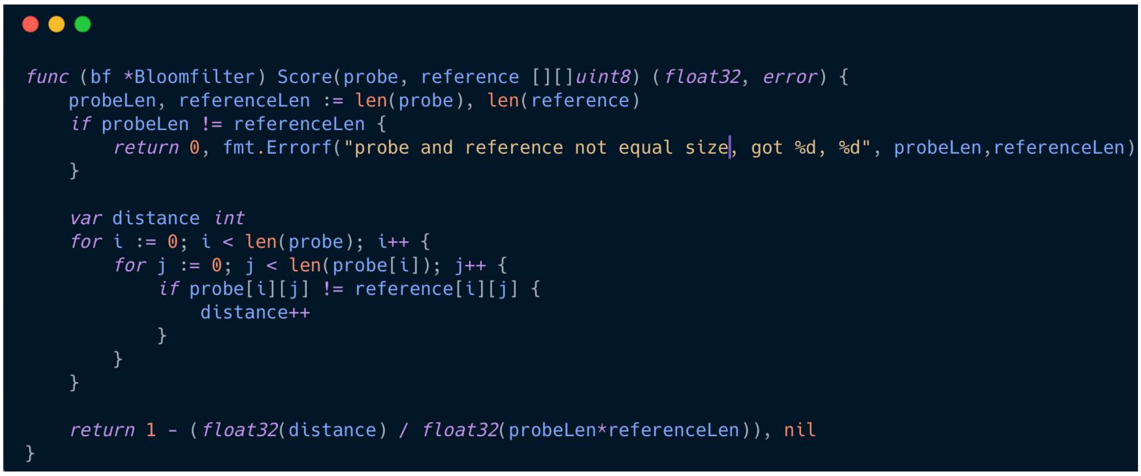
The bloom filter can be parameterized with three different values. Firstly, it is the key which ensures the unlinkability, then nWords which defines the number of elements which are inserted in the filter. The last parameters is nBits which determines the length of the template. The benchmark was used to find the most optimal values for nBits and nWords.

The bloom filter technique begins by reshaping the feature vector from its initial size of 1x512 to 32x16. Then a template with only zeros is allocated, and the size of this template is determined by the nBits and nWords parameters. The reshaped array, which now resembles a rectangle in form, is divided into smaller blocks, and the columns of each subblock is extracted. Next the bits in this column is flipped and converted to its decimal value.

To further increase the irreversbility of the biometric data the newly created decimal is used to perform bitwise XOR with the key parameter. The last operation is to perform modulus on the value from the previous operation with the length of the template as the right operand. This is done to ensure that the final value does not become larger than the max length of the template. Finally, this value is used together with the index of the current block to set the element at that location to 1.

6.6.3 Calculating similarity score

Normalized hamming distance is used to calculate the similarity score between the probe and reference template. It works by increasing distance by one for each element that is different and is normalized by dividing the distance by length of the template which can be seen in figure 23.



```go
func (bf *Bloomfilter) Score(probe, reference [][]uint8) (float32, error) {
    probeLen, referenceLen := len(probe), len(reference)
    if probeLen != referenceLen {
        return 0, fmt.Errorf("probe and reference not equal size, got %d, %d", probeLen,referenceLen)
    }

    var distance int
    for i := 0; i < len(probe); i++ {
        for j := 0; j < len(probe[i]); j++ {
            if probe[i][j] != reference[i][j] {
                distance++
            }
        }
    }

    return 1 - (float32(distance) / float32(probeLen*referenceLen)), nil
}
```

Figure 23: Code for calculating similarity score

The normalized hamming distance result is subtracted by one to align with the scale used by Mobai's Biometric Engine. As a result, two identical templates would score a 1 and templates with no similarities would score a 0.

6.6.4 Benchmark

A benchmark was conducted to increase the understanding of the consequences of data loss, performance, and establishing the system's decision threshold. In the benchmark, two front-facing images from the Facial Recognition Technology (FERET) database were utilized, totaling 146 subjects[44]. The scores were computed using three distinct techniques. First, Mobai's Biometric Engine calculates its score using floating point numbers, making it the most accurate of the techniques. This is useful for understanding how well the other approaches perform. The normalized hamming distance technique was used to score non-protected templates in order to evaluate the impact of the binarization of the float values. Lastly, the same method was used to score the protected biometric templates.

When benchmarking biometric systems it is important to test against both mated and non-mated images. A mated match is the act of comparing two images of the same subject, and non-mated is two images of two different individuals. In the figure 24, hamming distance is abbreviated to HD. A score of 1 indicates a 100% match, whereas a score of 0 indicates 0% match.

[44] P.Jonathon Phillips et al. 'The FERET database and evaluation procedure for face-recognition algorithms'. In: *Image and Vision Computing* 16.5 (1998), pp. 295–306. ISSN: 0262-8856. DOI: https://doi.org/10.1016/S0262-8856(97)00070-X. URL: https://www.sciencedirect.com/science/article/pii/S026288569700070X.

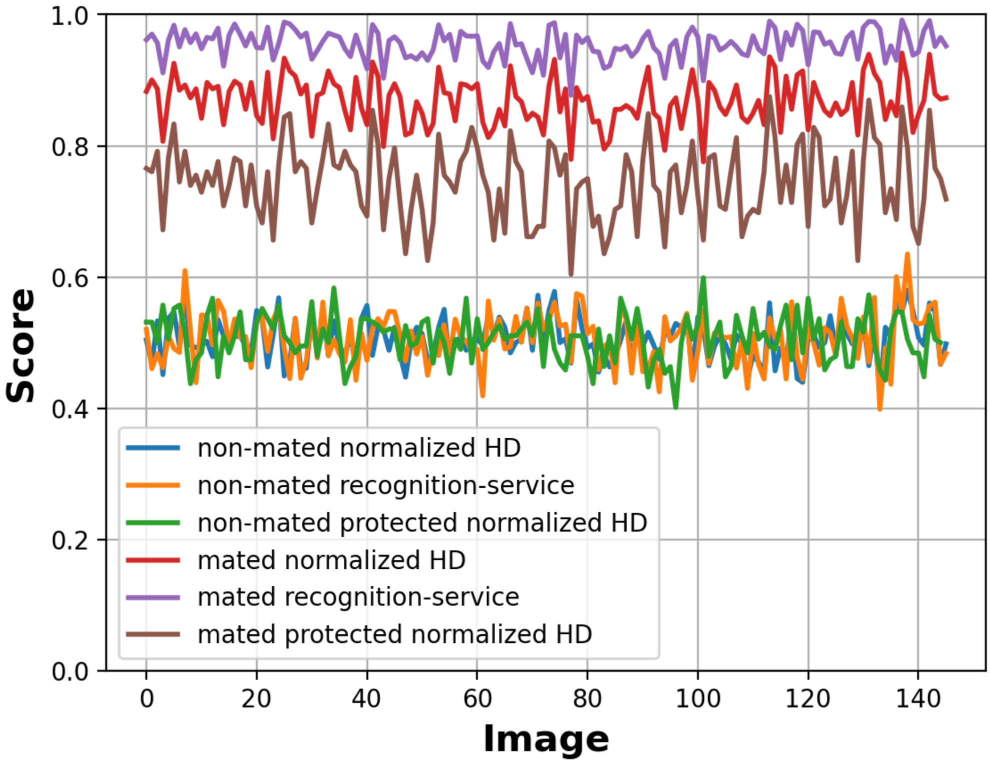


Figure 24: Biometric template protection benchmark

As expected the Mobai's Biometric Engine, here named recognition-service, is at the top of the graph for the mated matches. Not far below is the scoring of the non-protected binarized feature vector with an acceptable accuracy loss considering the rudimentary binarization algorithm. Next, we have the mated protected normalized HD, which continues to lose accuracy; nonetheless, this is to be anticipated since obfuscation necessitates data scrambling, leading to data degradation. The lowest scoring of a mated match is 0.6, which is close to the non-mated scores, however, the other techniques have also suffered a significant loss on the same subject, but this observation needs to be taken into consideration when defining the decision threshold.

The non-mated comparison scores for all the techniques are in the same range that indicates that face recognition systems are highly accurate in differentiating the impostors. Therefore the privacy preserving techniques did not influence the variation of the imposter scores while the mated or genuine scores are highly influenced.

It is critical in a biometric system to minimize the amount of false positives since it would allow a person to access the prototype without being the genuine person. On the other side, having too many false negatives will make the system more secure with the consequence of decreasing the UX for the user. The decision threshold was selected at 0.65 based on this reasoning and the observations from the benchmark. As a result, some matches will be denied access, however, it is necessary for the overall security of the system. In order to reduce the number of false negatives, measures such as assisting the user in being correctly positioned in the capture with a suitable lighting can further enhance the UX by reducing the false negatives. Those types of features must be considered in the future work.

6.7 OIDC Client

Purpose: The OIDC Client demonstrates the authorization code flow for a client.

The code used for this service was heavily inspired by the example provided for the go-oidc library by CoreOS[45]. The service presents the user with a button to begin the authorization code flow, and the library's OpenID Connect Discovery 1.0 transfers the user to Hydra Companion's login page. The access, refresh, and ID tokens are dumped when the user has authenticated themselves and accepted the grants, indicating that the flow was successful. Given the number of adjustments necessary to make this work from an example obtained online, the project's success is reinforced, since simplicity of integration with Mobai's systems was one of the project's goals.

6.8 Clients Overview

Purpose: The Clients Overview application enables the end-user to view and delete clients which they have given access to.

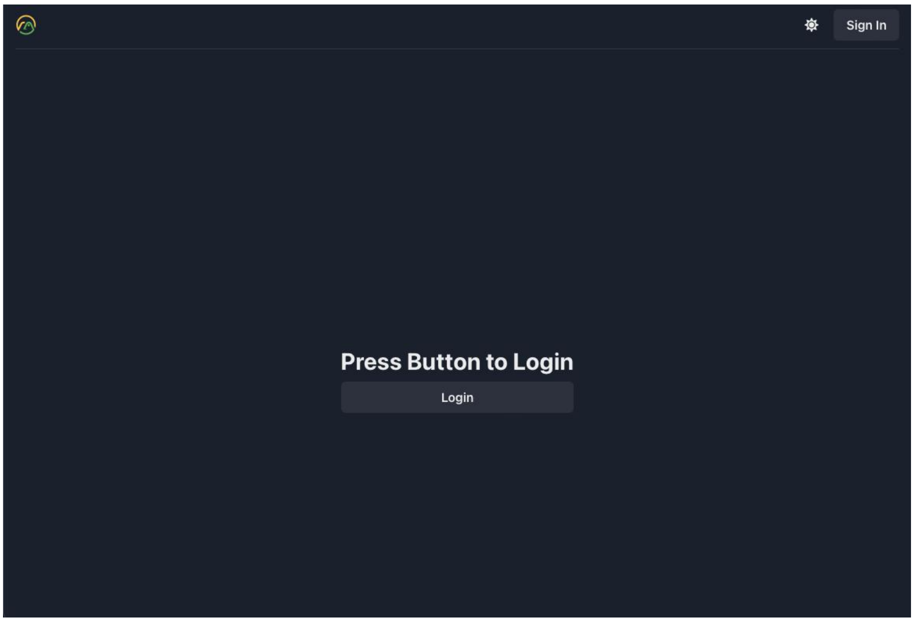
It is developed using React and uses the same UI component library as Hydra Companion. The design deviates a bit from the initial design as seen in section 4.4.4 where the idea was to use Go HTML templates with CSS. However, after transitioning to Chakra UI, the utilization of more sophisticated components and a modern feel to the UX became possible.

Login screen

The login screen is displayed as the application does not find any information in the browser that the person has signed in. The user can start the login process by

[45] CoreOS. *Go OIDC client example.* URL: https://github.com/coreos/go-oidc/blob/v3/example/idtoken/app.go.

clicking the login button.



Press Button to Login

Figure 25: Clients overview login

Overview screen

After completing the login flow and accepted the grants requested by Clients Overview, the user is presented with the different clients they have given access to. In this case the user has also granted access to the OIDC Client Demo application. As seen in the screenshot 26, NTNU and Mobai's logos are used for demonstration, and these logos are configured per client configuration. The name of the user is also extracted from the ID token, and other possible information to retrieve is based on the accepted grants. Another important feature is enabling the user to revoke the access of a client, and this can simply be done by clicking the revoke button.

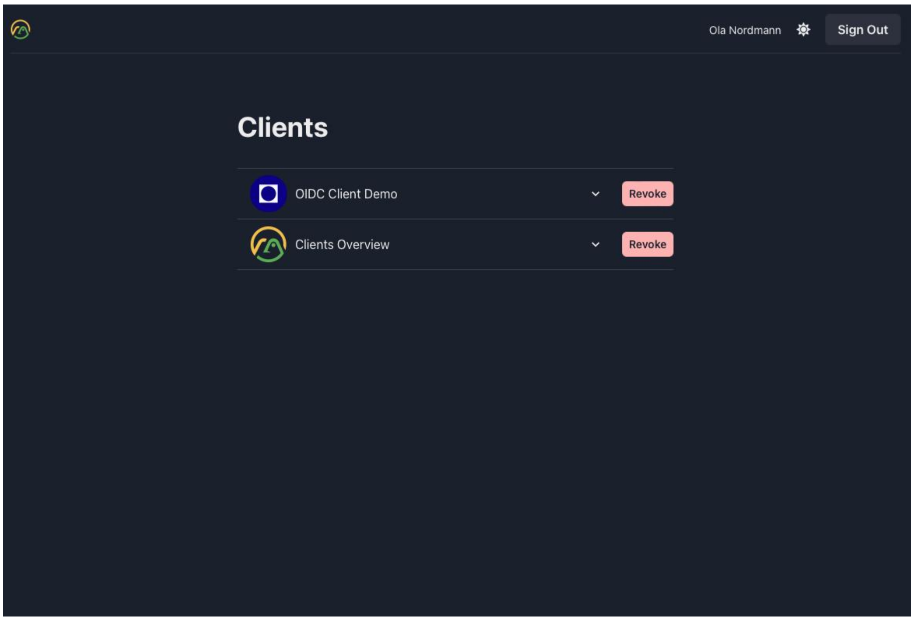
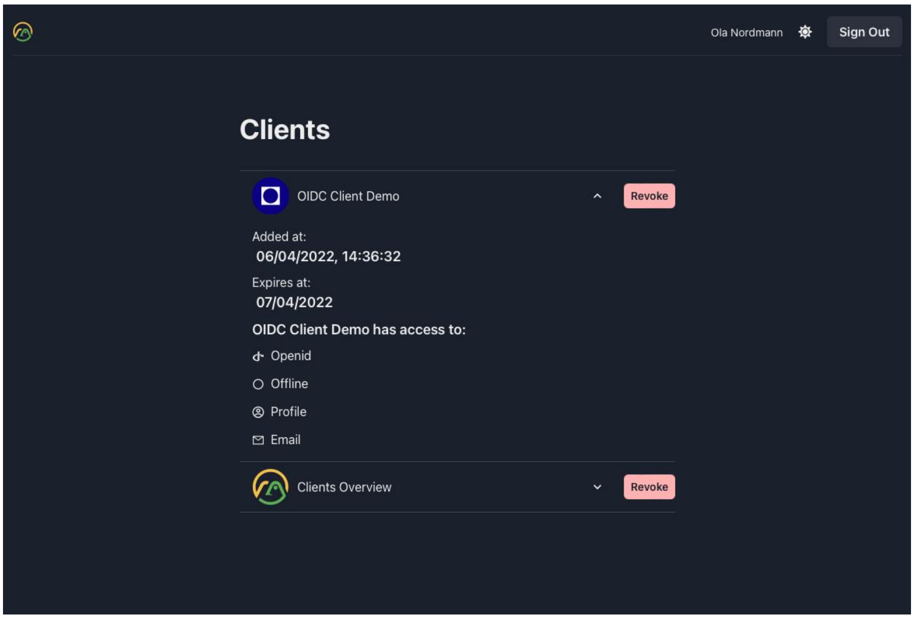


Figure 26: Clients overview

Overview screen expanded

The same view also has the option to expand each client and read more detailed information such as when it was added, expiration date and the grants that were accepted on consent.



Added at:
06/04/2022, 14:36:32

Expires at:
07/04/2022

OIDC Client Demo has access to:

- Openid
- Offline
- Profile
- Email

Figure 27: Clients overview expanded

The application uses OIDC implicit flow to perform the authentication of the user. This flow is similar to the authorization code flow, which is used by OIDC Client, with the difference being skipping the step of exchanging the authorization code in the back-channel. The access and ID token are available in the browser which can pose a security risk, and will be further discussed in risk 7.7.5[46].

The clients presented are retrieved by sending a request to Hydra Companion which again talks to Hydra. The Ingress configuration in Kubernetes is configured to use Oathkeeper as the decision API. When Oathkeeper receives the request from NGINX it sends an introspection request to Hydra, as per configuration, with the JWT token provided by Clients Overview and finally responds back to NGINX with the decision of allowing or denying the request. This approach scales well in a microservice architecture as each service does not need to implement the authentication and authorization logic.

[46] Internet Engineering Task Force Trust. *OAuth 2.0 - Implicit Grant.* URL: https://datatracker.ietf.org/doc/html/rfc6749#section-4.2.

7 Risk assessment

7.1 Introduction

Attacks on biometric systems are prevalent according to Kaspersky Security Network (KSN). According to their data, malware was prevented on 37% of the computers where they run their malware protection that collect, process, and store biometric data in Q3 2019 - that is, one computer in every three was at risk of malware infection[47]. The risk assessment is based on the assumption that the prototype exists in a production setting and used by actual users. Therefore the risk assessment will have privacy as its primary focus.

The goal of the risk assessment is to identify certain vulnerabilities and determine how they can affect the users privacy. The risk assessment will map the assets, threat actors, vulnerabilities, scenarios, probabilities and consequences. Treatments will be applied to the probability and consequences to certain events in order to reduce the risk to an acceptable level. A risk matrix indicating the risk before and after treatments will then display the severity of the risks once treatments are applied.

7.2 SWOT analysis

The purpose of the SWOT analysis is to map the prototype's strengths and weaknesses, as well as to acquire an understanding of the opportunities and threats that may effect it.

[47] Kirill Kruglov. *Biometric data processing and storage system threats.* URL: https://ics-cert.kaspersky.com/media/Threats_to_Biometrics_FINAL_ENG.pdf (visited on 02/02/2022).

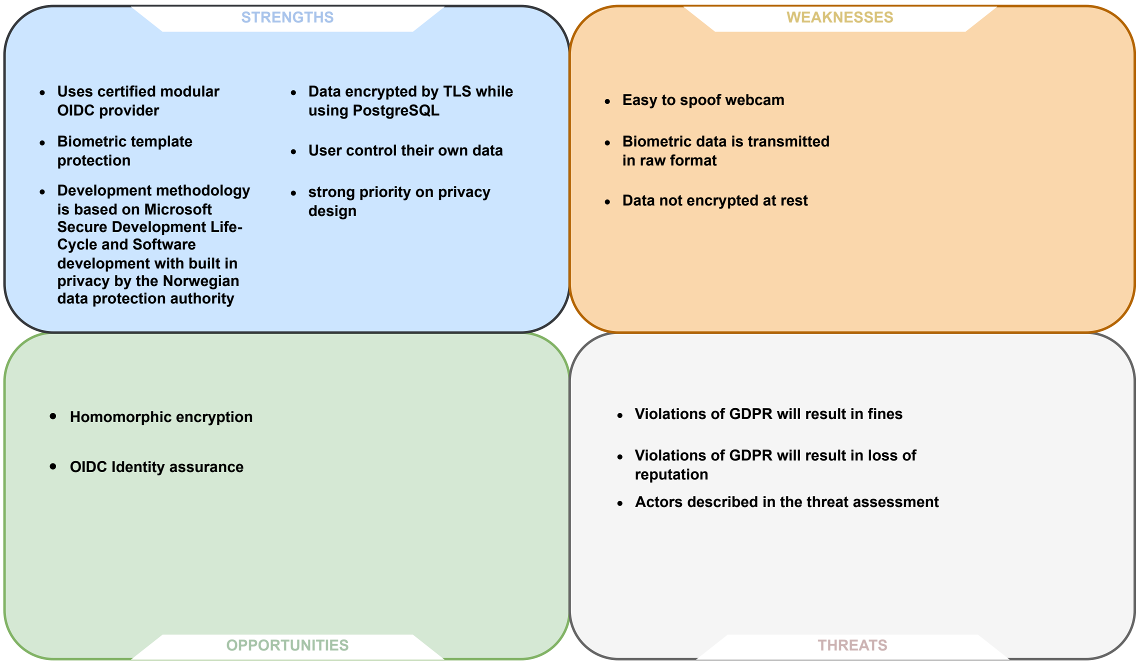

STRENGTHS
Uses certified modular OIDC provider
Biometric template protection
Development methodology is based on Microsoft Secure Development Life-Cycle and Software development with built in privacy by the Norwegian data protection authority
Data encrypted by TLS while using PostgreSQL
User control their own data
strong priority on privacy design
WEAKNESSES
Easy to spoof webcam
Biometric data is transmitted in raw format
Data not encrypted at rest
Homomorphic encryption
OIDC Identity assurance
OPPORTUNITIES
Violations of GDPR will result in fines
Violations of GDPR will result in loss of reputation
Actors described in the threat assessment
THREATS

Figure 28: SWOT Analysis

While strengths provide a suggestion to what is positive, weaknesses offers a pinpoint as to what requires further work. Opportunities and threats may suggest that the prototype requires a set of adjustments, since threats can have a negative impact, and opportunities can give guidelines on topics that could improve the prototype.

As seen in figure 28, strengths are features that have already been implemented in the prototype, which enhances the development cycle and privacy standards. These have previously been discussed. Opportunities such as homomorphic encryption, which can be read in section 9.2.1, is an example of a prospective implementation that can strengthen the prototype. Weaknesses are characteristics in the prototype that actors might exploit for personal gain or economic benefits. One example is that the prototype does not control the camera input, which implies that a malicious actor can take advantage of this. Lastly, threats are external risks that may arise to the prototype as a result of flaws or other characteristics that might negatively impact the prototype. The threats are the actors mentioned in the threat assessment, as well as potential fines and infractions as a result of exploited weaknesses.

7.3 Asset classification

To proceed with a risk assessment, it is essential to collect an inventory of the assets for further examination. In order to establish the risk analysis, the assets must be evaluated as the foundation. This is due to the fact that the assets are the data that must be protected, which is important when evaluating the overall system critically and security requirements. For the asset evaluation, a information classification approach will be used. This method is based on the pillars of information security and assesses the requirements for Confidentiality, Integrity, and Availability (CIA).

If anything occurs to an assets of the product in production, the information classification provides information to understand the repercussions. As a result, classification is used to determine system criticality and prioritization of tasks. Four security levels will be used as guidelines with descriptions that you can use to analyze the CIA values, which are based on the NTNU classification[48].

When it comes to confidentiality, information is restricted based on who has the right to access it, this is especially important due to the processing of sensitive biometric data. The prototype must be available in order to serve the authentication requests to ensure users access to their respective service. Finally, for the integrity of the data, data must be supplied in the manner intended, with no unauthorized or erroneous alterations.

The NTNU guide for classification will be used to further develop the risk assessment based on the information that is stored within the system[49]. It is essential that the appropriate standards for confidentiality, integrity, and availability is established, which is something this classification description covers. Another justification for using NTNU's classification guide is because NTNU has substantial experience in information security and risk assessment, it was deemed unnecessary to produce a classification description that NTNU had previously created.

The level of confidentiality differs according to the information on the corresponding systems to which the user has access. Access to certain information will be prohibited if the level of confidentiality for certain users exceeds a predefined threshold. Because fewer users will have access to confidential information, the chance of information leakage is reduced. This is crucial because if sensitive information were to be leaked, the consequences for Mobai would be severe, resulting in a loss of reputation and financial damage due to GDPR violations.

GDPR has a significant impact on the confidentiality of stored data. Users might lose access to their stored images if there was downtime or a malfunctioning system. As a result, assets classified as any level on confidentially must require immediate

[48] Norwegian University of Science and Technology. *Informasjonsklassifisering - informasjonssikkerhet.* URL: https : / / i . ntnu . no / wiki/ - /wiki / Norsk / Informasjonsklassifisering+ - +informasjonssikkerhet.

[49] Norwegian University of Science and Technology, *Informasjonsklassifisering - informasjonssikkerhet*.

measures to address the system faults. This is because Mobai processes highly sensitive biometric data, which necessitates fast attention if something were to happen to this information. These precautions are required, especially given the sensitive data handled and the fact that Mobai is still in a startup phase which requires a strong reputation.

The assets that are relevant to the prototype are mapped in order to detect threats, vulnerabilities, and existing controls related with the biometric authentication solution. The different assets were collected through an internal discussion with the group and with Mobai. A description of all the assets is provided, as well as an overall assessment of their importance.

7.4 Asset evaluation

The asset evaluation [5] is evaluated on the basis of the asset classification description provided by NTNU. The table below includes two columns labeled asset owner and asset user, which provide information about who owns and utilizes the assets to further increase the overall understanding of the system.

Asset	Description	Asset owner	Asset user	C	I	A	Impact
Session Cookies	Cookies used to track a user's session and avoid relogin on page refresh.	Resource Owner	ORY Hydra	4	4	2	4
Refresh Tokens	A token that refreshes the the access token.	Resource Owner	ORY Hydra, resource owner, OIDC Client	4	4	4	4
ID Token	The token that enables 3rd parties to access user's identities.	Resource Owner	OIDC Client	4	4	4	4
Access Token	The Access Token is used to aquire access to the resource owners data.	Resource Owner	OIDC Client	3	4	4	4
Template Database	Database that stores the biometric templates.	Mobai	Mobai, Resource Owner	4	4	4	4
Template Protector	Generates the biometric template from an image and calculates comparison scores between two templates.	Mobai	Resource Owner	3	4	4	4
Webcam	Webcam used by end-users to take face images to authenticate.	Resource Owner	Resource Owner	4	2	2	4
ORY Hydra	OAuth 2.0 and OpenID Connect provider.	Mobai	Mobai, Resource Owner, OIDC Client	4	4	4	4
PII Database	Database that stores the Personal Identifiable Information	Mobai	Mobai, Resource Owner, OIDC Client	4	4	4	4
Face Images	Face images taken by end-users to authenticate with the prototype.	Resource Owner	Mobai	4	4	2	4
Hydra Companion	Handles login, consent and logout management.	Mobai	Mobai, Resource Owner, OIDC Client	3	3	4	4
Biometric Engine	Mobai's web-service, recognition-service and pad-service.	Mobai	Mobai, Resource Owner, OIDC Client	3	4	4	4
OIDC Credentials	The credentials needed by a customer in order to use Mobai's OIDC solution. Client ID, client secret.	OIDC Client	Mobai, OIDC Client				
Personal Identifiable Information	The information that is stored which can be used to identify the end-user.	Resource Owner	Mobai, Resource Owner, OIDC Client	4	4	4	4
Biometric Template	The obscured facial biometric information.	Resource Owner	Mobai	1	2	4	4

Table 5: Asset evaluation

Every asset is characterized based on CIA triad. The assets' score is increased by the fact that they are part of a system that is either processing or storing biometric data and other sensitive information. For the assets evaluated to a lower score is due to them not impacting the whole system, but rather the user without violating the GDPR. The table in appendix E provides a detailed explanation of why each asset received its corresponding score in the asset assessment.

7.5 Threat Assessment

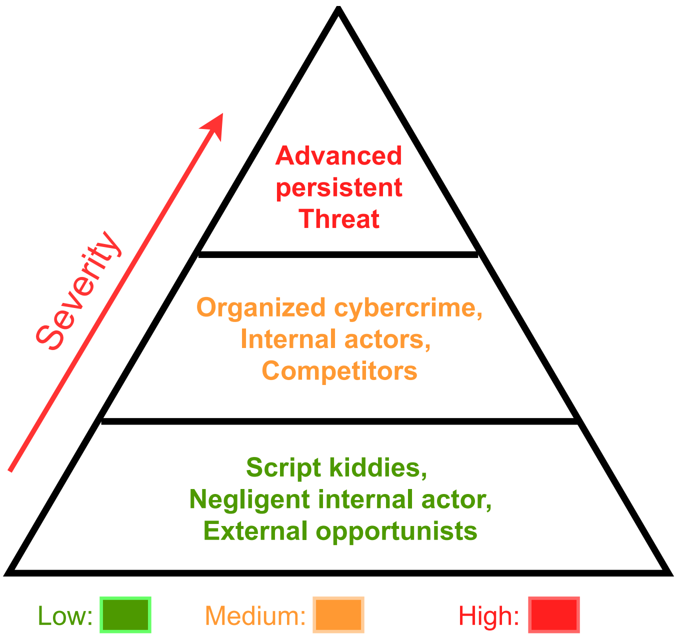

Severity
Advanced persistent Threat
Organized cybercrime, Internal actors, Competitors
Script kiddies, Negligent internal actor, External opportunists
Low: 🟩
Medium: 🟧
High: 🟥

Figure 29: Actors that pose a threat to Mobai

The threat pyramid will show how much impact an assault will do to the victim based on their severity level. Because actors with a low severity rating will attack more frequently, the number of potential attacks decreases as the severity level increases. The investment in an attack will increase as the severity of the attack increases since more advanced actors have more resources to spend on an attack. Because there are less advanced organizations than APT or organized cybercrime groups that will use less advanced tools and methods which are easier to create, there will be a rise in the frequency of attacks on the lower end of the pyramid compared to the higher end.

A threat assessment will provide an overview of the threat actors 31 who may have an impact on the assets for this prototype. It is critical to map the various actors and their motivations for their activities and to discover and analyze risks in order to determine the attack vectors and obtain knowledge of which security breach the actor could conduct. As seen on figure 31 the severity the different actors pose to Mobai can be categorized as low, medium, and high (highlighted by different colors for improved visualisation).

Capacity, capability, and frequency are used to evaluate the severity of the actors, which are also categorized as low, medium, and high. Capacity is assessed by how many resources the actor has available. For example, an attack that requires several weeks of effort by a larger team with the resources needed to conduct the operation.

Capability is based primarily on the actor's skill set, where an exceptional hacker provides a set of abilities that provides a high capability. Finally, frequency refers to how often an attack might occur, where an organized cybercrime group may have greater resources to conduct attacks more frequently than for instance, an external opportunist. Frequency is also a good measure of how motivated the attacker is, because an actor who strikes numerous times over a period of days would be more interested in reaching their goal than an actor who conducts a single assault. The threat assessment result is used in the risk analysis, which accounts for case-specific situations with their associated probability and consequences.

A detailed description of each threat actor is found in appendix D.

7.6 Vulnerability Assessment

The vulnerability assessment will identify the various vulnerabilities in the prototype, which will be evaluated and mitigated in section 7.8. Exploitability and exposure are used to determine the threat level of a vulnerability. Exploitability is described as the difficulty for an actor in gaining access to sensitive information, as well as the amount of capacity and capability required to exploit vulnerability. The term exposure refers to how accessible the assets are once the vulnerability has been exploited.

The table 6 contains a brief description of how exploitable the vulnerabilities are. It will be used in the vulnerability assessment to assess how difficult it will be for an actor to acquire access to the sensitive information, as well as how much capacity and capability necessary to exploit vulnerability. There is also a description of exposure, which is also used in the 7 to assess how accessible the assets are as after exploiting the vulnerability.

	Exploitability	Exposure
Low	The required capability and capacity of the actor is high	The assets is not very accessible through the vulnerability
Medium	The required capability and capacity of the actor is medium	The assets is to some degree accessible through the vulnerability
High	The required capability and capacity of the actor is low	The assets is very accessible through the vulnerability

Table 6: Description of Exploitablity and Exposure

Table 8 illustrates several vulnerabilities present in the prototype, which are analyzed based on an estimate of exploitability and exposure. The vulnerability is listed in the table, along with a brief description. It also estimates how exploitable and exposed the vulnerability is, using a scale of low, medium, and high. This is done to estimate how probable an actor is to exploit the various vulnerabilities before implementing mitigation methods.

Vulnerability	**Description**	**Exploitability**	**Exposure**
Camera input to browser	The prototype can not verify that the camera device is an actual camera.	High	Medium
Client ID og client secret	Mobai can not guarantee the security of the users as the client / customer must handle client ID and secret. If this is leaked, others will have access to grants that have been given.	Low	High
Information stored in database is not encrypted	Personal information can get leaked because it is not encrypted in the database.	Low	High
Flawed CSRF protection	Missing or static state paramater in the authorization code flow resulting in an attacker being able to commence an OAuth flow before deceiving a user's browser into finishing it which enables an attacker to bind their own social media account to the victims account.	Medium	Low
Implicit grant type	A less secure method than authorization code flow, where communication occurs between the browser redirects. This leaves the access token and user's data vulnerable to attack since there is no secure back-channel in the authorization code flow.	Medium	Medium
Flawed scope upgrade: Authorization code flow	An attacker can upgrade the permission request scope received by a malicious client application when a user initially just wished to share, for example, their e-mail address, but the attacker can add scope arguments to the code/token to gain further information about the user.	Low	Medium
Fake OIDC client	The client application implicitly assumes that the information saved by the OAuth 2.0 provider is accurate, which can be dangerous. An attacker can exploit this by registering with the OAuth provider using the same information as the target user, such as a known email address. This enables the attacker to sign in as the victim with the fake OAuth 2.0 account on the client app.	Low	Medium

Vulnerability	Description	Exploitability	Exposure
Unprotected endpoints when inside the Kubernetes cluster network	When a user wants to access Clients Overview the user gets authenticated by Ory Oathkeeper with a Bearer token. When the authentication is confirmed the user get redirected to Hydra companion. If an actor has access to the internal kubernetes network they would be able send requests to both hydra companion without any authorization or authentication	Low	High
No request throttling	Actor have an unlimited time of login attempts and this could be exploited with for example brute forcing and other automated attacks	Medium	Medium
Sharing of sensitive data to 3rd parties	The user has given access to their PII to a client and wants to revoke it. This does not remove the data from the client since they could have simply stored it when they had the access before the revoke.	Medium	Medium

Table 7: Vulnerability assessment

Some of the vulnerabilities discovered in the vulnerability assessment are linked to the OWASP top ten, which are discussed in section 2.7. The vulnerability identified as no request throttling is related to identification and authentication failures, because of the ability to make an unlimited number of sign in requests. Cryptographic failures and injections are also frequent OWASP weaknesses, and the prototype's discovered weakness of the database not being encrypted might be exploited to extract information about registered users. Even if the database is well-protected, it is possible that a skilled hacker may get access to it.

7.7 Risk analysis

The risk analysis will cover several risks associated with the prototype that might emerge as a result of flaws. Each risk will have one or several scenarios linked to the affected assets, threat actors, and vulnerabilities. Consequence and probability will also be ranked on a scale of 1 to 4, based on the NTNU classifications, with a brief description on why it was given a specific rank[50]. Only the criteria that are important for the prototype of this project will be considered.

7.7.1 Risk 1: Camera input

Assets

Webcam, PII

Threat actors

Script kiddie, competitor, organized cybercriminal or external opportunist, Advanced Persistent Threat (APT)

Vulnerabilities

Camera input to browser

Scenario 1: Cybercriminal

A cybercriminal has knowledge of a users' email and spoofs the camera input which enables them to gain unauthorized access in order to steal and/or sell data.

Scenario 2: Competitor

A competitor creates an account and spoofs the camera input during authentication to demonstrate weakness in the system resulting in a damaged reputation.

Consequence: 4

The consequence of an actor spoofing the camera can be serious because of the possibility to gain full unauthorized access to a user's account. Furthermore, it could lead to fines and sanctions from the government because of violation of GDPR, and the market may convert to Mobai's competitor since they see weaknesses in Mobai's solution.

Probability: 4

The probability of the camera input being spoofed is 4 because of the high exploitability, and the small amount of resources and skills that are needed.

[50] Norwegian University of Science and Technology. *Risiko og Sårbarhetsanalyse på NTNU*. URL: https://i.ntnu.no/documents/1306938287/1307171093/Presentasjon+ROS-VS.pdf/e0614d01-f2ff-46f3-84dc-516870be5166?t=1536924541534&status=0 (visited on 08/04/2022).

7.7.2 Risk 2: Unprotected endpoints in cluster

Assets

Hydra Companion, PII, ORY Hydra, Template Protector, Template Database

Threat actors

Organized cybercriminal, APT or external oppurtunist

Vulnerabilities

Unprotected endpoints in the internal Kubernetes network.

Scenario

A threat actor gains access to the internal Kubernetes cluster and extracts data due unprotected endpoints.

Consequence: 4

The consequence is set to 4 because an actor with access to the cluster may acquire a large amount of sensitive data. Data leaks will result in government sanctions and fines, as well as a loss of reputation since it will be regarded as risky to use.

Probability: 2

The probability is set to 2 since the vulnerability is difficult to exploit and the actor requires a diverse mix of skills and resources.

7.7.3 Risk 3: Sharing of sensitive data to 3rd parties

Assets

Personal Identifiable Information (PII)

Threat actors

Organized cybercriminal, APT or external opportunist

Vulnerabilities

Sharing of sensitive data to 3rd parties

Scenario

A user shares their data with a 3rd party client which has not implemented proper security measures. As a consequence, the PII provided by the user is stolen as a result of an illegal actor successfully breaching the 3rd party system.

Consequence: 3

Because clients handle data differently, as well as the quantity of sensitive data given, the consequence might vary a lot. In some circumstances, the user may simply allow access to their email, meanwhile in other cases, they would allow access to all of their sensitive data. Despite the fact that the customers are to blame for the leak, Mobai will suffer a reputational loss as well as potential sanctions. The ultimate consequence greatly depends on the circumstances and Mobai's irresponsibility in the situation.

Probability: 2

The probability can also vary because the clients has different security and routines. The chances of this scenario occurring will also increase correspondingly with the amount of customers that are using Mobai's system.

7.7.4 Risk 4: Brute forcing

Assets

Hydra Companion, PII, Webcam

Threat actors

Script kiddie, organized cybercriminal or external opportunist, APT

Vulnerabilities

No request throttling, Camera input to browser

Scenario

A threat actor with a video of a user and uses it as the camera input while bruteforcing e-mail addresses, because there is no limit to how many login attempts you can make.

Consequence: 4

The consequence is set to 4 because the actor who exploits the weakness of unlimited login attempts can abuse the user's account and get access to particular data of the user if the breach is successful. By doing so, the malicious actor has access to every client on which the user is registered, as well as personal data. Data loss will result in financial losses as well as a loss of reputation.

Probability: 3

Since the vulnerability can be easily exploited, the probability is set to 3. This is because criminal actors require a video of the person, which may be obtained in a variety of locations on the internet, including social media platforms. It also requires some set of skills to create a brute force script, which can either be off the shelf, or a more sophisticated program.

7.7.5 Risk 5: Implicit grant type

Assets
Access Token, ID Token, Refresh Token

Threat actors
Organized cybercriminal, external opportunist

Vulnerabilities
Implicit grant type

Scenario
An organized cybercriminal or an external opportunist steals tokens in the browser to obtain access to private data by exploiting the implicit grant type.

Consequence: 3
This risk has the consequence set to 3, because if a malicious actor exploited the flaws in the implicit grant type, it would result in various negative consequences such as financial losses and loss of reputation because the actor would acquire access to the user and its sensitive information by using their private tokens.

Probability: 2
The reason it is set to 2 and not higher is that the actor must have access to the user's PC or sniff out network data, both of which take a high level of expertise. It is only recommended to use this type in specific occasions and not recommended for this prototype.

7.8 Mitigation plan

The mitigation plan is created to provide treatments for the various risks. Every measure will include a description of the mitigation strategy, the cost of implementing it, and the benefit of doing so. Below is a table that shows the levels of cost.

High	500000 NOK+ or 2500 hours+
Moderate	100000 to 500000 NOK or 500 to 2500 Hours
Low	0 to 100000 NOK or 0 to 500 hours

Table 8: Cost levels of mitigations

7.8.1 Mitigation 1: Camera app

Mitigates risk:

Risk 1: Camera input

Risk 4: Brute force

Strategy

Reduce the probability of the camera input being manipulated.

Description

Create a separate mobile app that ensures the camera input for face authentication is authentic. The drawback of the mitigation is a reduced user friendliness as it complicates the login flow, and it may also be challenging to persuade consumers to install the app in addition to restricting the system to require a phone to work.

Cost

High

Risk after mitigation - Benefit(CxP):

Risk 1: 4x4 to 4x1

Risk 4: 4x3 to 4x1

Residual risk

Risk 1: Acceptable

Risk 4: Acceptable

7.8.2 Mitigation 2: Kubernetes network policies

Mitigates risk:

Risk 2: Unprotected endpoints in cluster

Strategy

Reduce the probability of exploited pods having access to other entities in the cluster.

Description

Create network policies that controls which entities the pods have access to communicate with. This will control that infected pods does not have access to other entities that the threat actors can steal data from[51].

Cost

Low

Risk after mitigation - Benefit(CxP):

From 4x2 to 4x1

Residual risk

Acceptable

7.8.3 Mitigation 3: Kubernetes pod security admission

Mitigates risk:

Risk 2: Unprotected endpoints in cluster

Strategy

Restrict unauthorized access to and from pods to reduce the probability of a data being accessed and leaked.

Description

The pod security admission will allow for the definition of different isolation levels for pods, allowing the restriction of particular pod behaviors.[52]

Cost

Low

Risk after mitigation - Benefit(CxP):

From 4x2 to 4x1

Residual risk

Acceptable

[51] The Kubernetes Authors. *Network Policies.* Apr. 2022. URL: https://kubernetes.io/docs/concepts/services-networking/network-policies/ (visited on 20/04/2022).

[52] The Kubernetes Authors. *Pod Security Admission.* Jan. 2022. URL: https://kubernetes.io/docs/concepts/security/pod-security-admission/ (visited on 20/04/2022).

7.8.4 Mitigation 4: Strict client policy

Mitigates risk:

Risk 3: Sharing of sensitive data to 3rd parties

Strategy

Reduce the consequence if a clients gets hacked.

Description

Create a strict policy for customers so that user data is handled securely and the danger of data leakage is minimized. This policy may include a risk analysis of a customer, however, this might be costly. Another approach could be a policy the customers must comply with. The disadvantage of this is that Mobai cannot ensure that the clients' security is within the required tolerance.

Cost

Moderate

Risk after mitigation - Benefit(CxP):

From 3x2 to 2x2

Residual risk

Acceptable

7.8.5 Mitigation 5: Clients management

Mitigates risk:

Risk 3: Sharing of sensitive data to 3rd parties

Strategy

Reduce both consequence and probability of data being leaked from clients.

Description

Only provide clients with data such that it conforms with the data minimization principle, and ensure that clients are serious and reliable before they can register and receive access to user information. Mobai must also be critical when customers want to become clients, and must ensure that the customer is a serious organization with a strong privacy emphasis.

Cost

Moderate

Risk after mitigation - Benefit(CxP):

From 3x2 to 2x1

Residual risk

Acceptable

7.8.6 Mitigation 6: Limit for login attempts

Mitigates risk:

Risk 4: Brute force

Strategy

Implement a limit for the amount of login attempts a user can attempt, to reduce the probability of an hacker to access the user's account.

Description

This flaw allows an attacker to get access to an account by bruteforcing it. To avoid the potential of a bruteforce assault, a fixed restriction on login attempts must be imposed.

Cost

Low

Risk after mitigation - Benefit(CxP):

From 4x3 to 4x2

Residual risk

Unacceptable

7.8.7 Mitigation 7: Disallow implicit grant flow

Mitigates risk:

Risk 5: Implicit grant type

Strategy

Disallow the use of Implicit Grant Flow, and recommend the use of Authorization Code Flow by using Proof Key for Code Exchange (PKCE) instead for public clients to reduce the probability and consequence of leaking sensitive information from the browser[53].

Description

Authorization Code Flow with the use of PKCE will protect against CSRF and authorization code injection. The creator of the client must also be conscious when sharing the tokens to the public client as it can more easily be intercepted by an illicit actor. Another approach that could be considered is the use of for example NextAuth.js's library which uses the authorization code flow in the back-channel and only provides the browser with the necessary information extracted from the access or ID token[54].

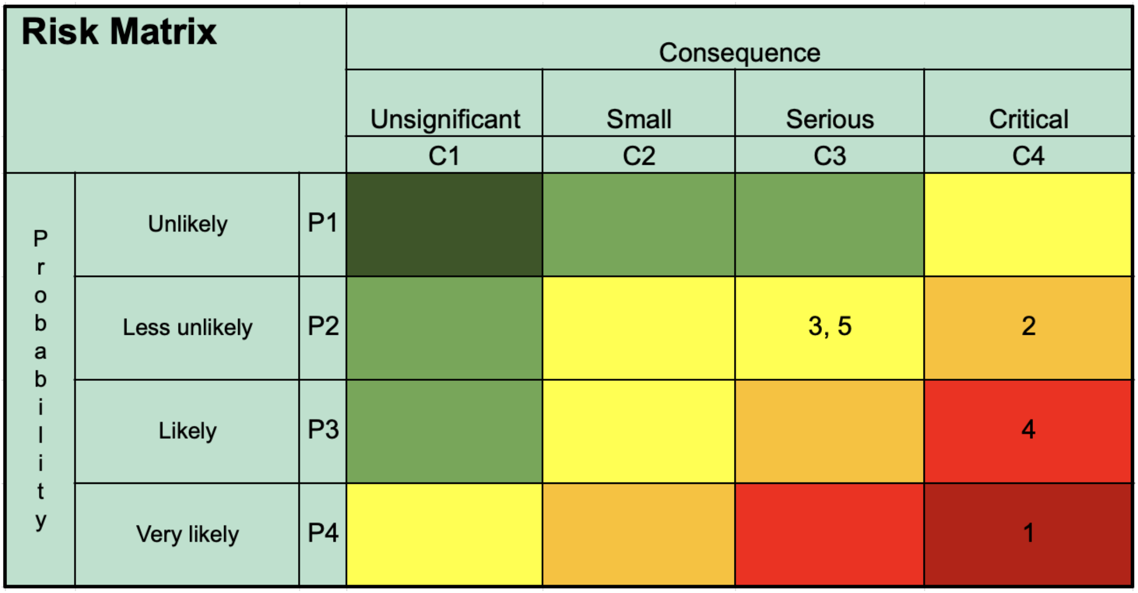
[53] N. Sakimura et al. *Proof Key for Code Exchange by OAuth Public Clients.* URL: https://datatracker.ietf.org/doc/html/rfc7636 (visited on 26/04/2022).

[54] Vercel. *Authentication for Next.js.* URL: https://next-auth.js.org (visited on 26/04/2022).

Cost
Moderate
Risk after mitigation - Benefit(CxP):
From 3x2 to 1x1
Residual risk
Acceptable

7.9 Conclusion of risk assessment

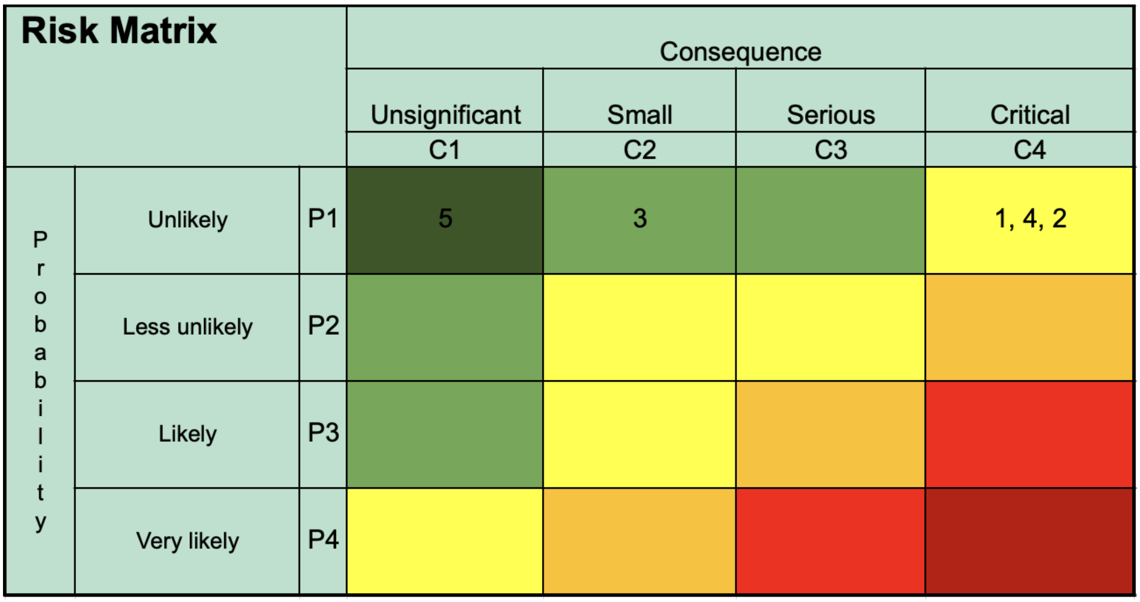
In conclusion, there are certain major risks in the prototype that should be reduced before it goes into production, according to this risk assessment. Risks that potentially constitute a significant threat were found through the identification of possible vulnerabilities and threat actors, as well as an evaluation of the assets. Recommended treatments to mitigate the risks were created, and the group defined a total risk acceptance level of 4 which is calculated by multiplying consequence and probability. The risk matrix from before mitigation is shown below.



Risk Matrix			Consequence			
			Unsignificant	Small	Serious	Critical
			C1	C2	C3	C4
Probability	Unlikely	P1	🟩	🟩	🟩	🟨
	Less unlikely	P2	🟩	🟨	🟨 3, 5	🟧 2
	Likely	P3	🟩	🟨	🟧	🟥 4
	Very likely	P4	🟨	🟧	🟥	🟥 1

Figure 30: Risks before mitigations

As seen, there are certain serious risks that might have disastrous consequences. As a result, minimizing them will be critical. To lower the risks to an acceptable level risk number 4 requires two mitigations, number 1 and 6. The risk matrix after implementation of the treatments is displayed below, and it indicates that all risks might potentially achieve an acceptable residual risk threshold if the mitigations are implemented.



Risk Matrix			Consequence			
			Unsignificant	Small	Serious	Critical
			C1	C2	C3	C4
Probability	Unlikely	P1	🟩 5	🟩 3	🟩	🟨 1, 4, 2
	Less unlikely	P2	🟩	🟨	🟨	🟧
	Likely	P3	🟩	🟨	🟧	🟥
	Very likely	P4	🟨	🟧	🟥	🟥

Figure 31: Risks after mitigations

Because the system handles very sensitive data, reducing the severity of the consequences is extremely difficult. Therefore, the primary goal of risk mitigation was to lower the probability of the risks.

8 Discussion

8.1 End-To-End tests

End-to-end tests were first developed in Cypress and introduced to the pipeline, where the system was built before the tests were ran. This required a significant amount of resources and time, resulting in long-running pipelines. The public Github Actions runners were likewise limited in terms of the resources they had available, making it impossible to use them for end-to-end testing as the number of services increased. The UI also changed, making it harder to maintain the tests, therefore they were removed from the pipeline. This came with the risk of incorporating potentially unstable code in the codebase, but the group was nevertheless confident since the code had been thoroughly tested using unit tests.

8.2 System complexity

As already mentioned, the system started to use more resources as the complexity increased. The group had access to NTNU's infrastructure and utilized it to develop the majority of the system; nevertheless, rebuilds were lengthy even when utilizing their computer with 16 CPUs and 32 GB RAM. After switching the local Kubernetes cluster from Minikube to Kind, it was feasible to run it on local machines that had Apple's new M1 Pro. This significantly improved the speed of rebuilds and the developer experience. Having said that, the complexity cost was still noticeable as tricky bugs were discovered as a result of the complicated system.

8.3 Development process

The group found that using a Kanban board and outlining the job at hand using Github issue templates was quite helpful, resulting in an efficient workflow. It ensured that everyone understood what everyone else was working on and made it easier to review each other's work. The issues that were resolved in the previous seven days were also reviewed at our weekly meeting with Mobai and our supervisor, which made it easy to visually display our overall progress. The defined WIP limits where sometimes not respected, and would also be easier to obey if Github supported it in their Kanban board.

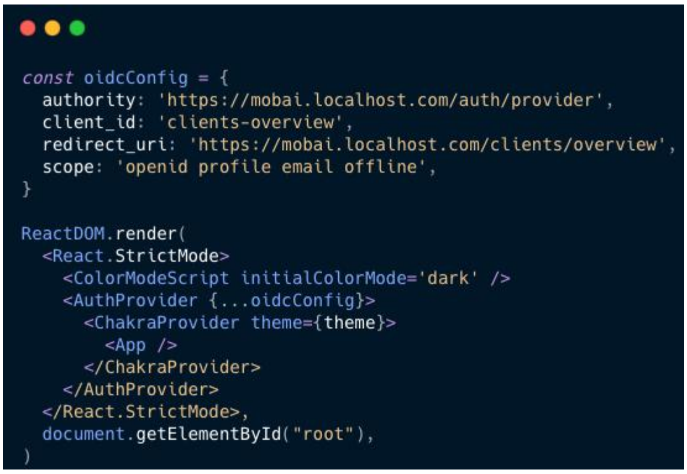
Initially, the group had bi-weekly retrospective meetings to review what worked as well as other pain points, but this was discontinued. In hindsight, this was something that may have aided collaboration but was unfortunately overlooked since no one took the responsibility of organizing the meetings. Despite this, the cooperation within the group was excellent, and any issues that arose were addressed on the spot.

In appendix F there is a summary of the timenotes. As seen there was some difference between the different group members, especially because of part-time work besides studying.

9 Conclusion

9.1 Results

The main objective of this project was to develop an OIDC provider and demonstrate its capabilities with the use of the Authorization Code flow. This was accomplished by using ORY's Hydra solution, which substantially reduced the amount of necessary code required to comply with the OIDC specification while boosting the system's security by utilizing battle-tested code. With the use of the prototype, customers of Mobai can more easily use their face as the authentication factor as seen by the code snippet in figure 32 which is from the Clients Overview application. The application uses an open source solution and only requires the OIDC configuration together with a higher-order component named AuthProvider before the application leverages the OP.



```
const oidcConfig = {
  authority: 'https://mobai.localhost.com/auth/provider',
  client_id: 'clients-overview',
  redirect_uri: 'https://mobai.localhost.com/clients/overview',
  scope: 'openid profile email offline',
}

ReactDOM.render(
  <React.StrictMode>
    <ColorModeScript initialColorMode='dark' />
    <AuthProvider {...oidcConfig}>
      <ChakraProvider theme={theme}>
        <App />
      </ChakraProvider>
    </AuthProvider>
  </React.StrictMode>,
  document.getElementById("root"),
)
```

Figure 32: Clients Overview OIDC integration for React

To achieve the project's security and privacy goals, the DPA's privacy by design and SDL was used as part of the development process. This outlined important privacy considerations which affected the different phases of the project. Furthermore, the risk assessment was performed to minimize risks and assist Mobai in understanding the residual risk, which will aid them in deciding what is necessary before proceeding with the system.

A template protection algorithm was developed to safely store biometric data and strengthen the user's privacy. The consequence of this approach is the accuracy loss as seen in figure 24, and a different template protection algorithm is proposed in section 9.2.1 which may reduce the accuracy loss at the cost of increased computation time.

9.2 Future work

The recommendation below are tasks which can be done to further improve the system together with the mitigation strategies as described in section 7.8.

9.2.1 Homomorphic encryption

Homomorphic Encryption (HE) is a type of encryption that enables computation to take place in the encrypted domain. This allows users provide encrypted data to a service, and the service does not need to decrypt it in order to perform its operations. The output from the computation will also be sent back to the user encrypted, and only they can decrypt it. As a result, the users can own their own data and does not need to give up their privacy to use a service which are beneficial traits for the a user's privacy. However, the main drawback of HE has been its heavy computational cost which among other things explains its less popular choice for usage as a BTP scheme[55].

9.2.2 Data Protection Impact Assessment

A Data Protection Impact Assessment (DPIA) is an analysis of privacy implications that should ensure that registered users' privacy is holistically present in the decisions and the end product. The DPIA should explain how personal information is processed, and to ensure any unnecessary storage of personal sensitive information. It will also assist in assessing the dangers that the personal information processing poses to users' rights and freedoms by evaluating the risks, as well as developing risk-reduction measures. When a system uses sensitive personal information such as biometrics, a DPIA is always required by Norwegian law and GDPR, therefore, a DPIA must be conducted[56].

From a high-level perspective a DPIA will assess the system by understanding its nature, scope, context, and purposes. This will answer the questions of how, why, what, and who. The necessity and proportionality of the data saved will also be evaluated. In addition, it will identify the data processing risks and devise strategies to mitigate them. Lastly, the remaining risks will be analyzed after they have been

[55] Sandhya and Prasad, 'Biometric Template Protection: A Systematic Literature Review of Approaches and Modalities'.

[56] Datatilsynet. *Vurdering av personvernkonsekvenser (DPIA)*. 17th July 2019. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/vurdere-personvernkonsekvenser/vurdering-av-personvernkonsekvenser/ (visited on 11/02/2022).

mitigated, measure the overall risk, and decide if the risk is below the established risk appetite[57]. A DPIA will not be done as part of this project as it is a time-consuming and resource-intensive process with regards to the scope. However, Mobai is required to do a DPIA before the product reaches a production environment.

9.2.3 Client and user management

The management of clients and users who wants to utilize the system is a requirement before being able to take the product to production. There are numerous user management solutions available, but ORY, the company behind Hydra, provides an identity and user management system called Kratos[58]. Because this system is already part of the ORY ecosystem, using their solution may be a viable option. It works similarly to Hydra in that Mobai may design their own UI while Kratos handles all of the business logic required in a management system. This may be highly cost efficient since the security and development costs of such a system are already done while still allowing for a great degree of customization.

9.2.4 OpenID Connect for Identity Assurance 1.0

If Mobai decides to enroll users using their passports, the passport's authenticity must be verified. The passport's data can then be considered verified and be stored in the database. If a RP requires the use of verified passport data, this information may be shared by conforming to the OpenID Connect Identity Assurance 1.0 specification[59]. It is an extension of OIDC which defines how to share identity information together with the verification status of the requested claims. This specification would be especially useful if Mobai's customer is a business that requires a high degree of assurance, such as a bank that must comply with regulatory regulations.

9.2.5 Complete DPA guide

The group did not have the time to fully complete the last three phases of the DPA's *Programvareutvikling med innebygd personvern*. Security testing such as dynamic,

[57] Information Commissioner's Office. *Data Protection Impact Assessments (DPIAs)*. URL: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/data-protection-impact-assessments-dpias/ (visited on 11/02/2022).

[58] ORY. *Kratos*. URL: https://www.ory.sh/kratos/.

[59] T. Lodderstedt et al. *OpenID Connect for Identity Assurance 1.0*. URL: https://openid.net/specs/openid-connect-4-identity-assurance-1_0-ID3.html (visited on 25/04/2022).

fuzz and penetration testing are something that must be done before production. Mobai must also develop an incident response plan in case of unwanted scenarios occurring while the system is running. Other activities that must be completed include establishing procedures for future security testing, upgrading dependencies for the various services, and re-evaluating that the established sensitive data processing is being followed.

References

Auth0. *Introduction to JSON Web Tokens*. URL: https://jwt.io/introduction (visited on 24/01/2022).

Burt, Chris. *FaceTec alleges stolen biometric liveness technology in lawsuit against iProov*. URL: https://www.biometricupdate.com/202201/facetec-alleges-stolen-biometric-liveness-technology-in-lawsuit-against-iproov (visited on 22/03/2022).

CAOS. *Certified OpenID Connect Library (client and server) for Go*. URL: https://github.com/caos/oidc (visited on 20/02/2022).

Cheney, Dave. *Prefer table driven tests*. URL: https://dave.cheney.net/2019/05/07/prefer-table-driven-tests.

Chung, L. and J.C.S.D.P Leite. *On Non-Functional Requirements in Software Engineering*. URL: http://ce.sharif.edu/courses/96-97/1/ce475-1/resources/root/NonFunctionalRequirements.pdf (visited on 26/01/2022).

Cohn, Mike. *Succeeding with Agile*. Presidio Press, 2009.

CoreOS. *Go OIDC client example*. URL: https://github.com/coreos/go-oidc/blob/v3/example/idtoken/app.go.

Crunchydata. *PostgreSQL Operator*. URL: https://github.com/CrunchyData/postgres-operator (visited on 21/02/2022).

Datareportal. *DIGITAL AROUND THE WORLD*. URL: https://datareportal.com/global-digital-overview (visited on 22/03/2022).

Datatilsynet. *Programvareutvikling med innebygd personvern*. 20th Aug. 2019. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/innebygd-personvern/programvareutvikling-med-innebygd-personvern (visited on 13/01/2022).

— *Programvareutvikling med innebygd personvern*. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/innebygd-personvern/programvareutvikling-med-innebygd-personvern/innebygd-personvern---hva-er-det/ (visited on 26/01/2022).

— *Virksomhetens plikter*. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/ (visited on 26/01/2022).

— *Vurdering av personvernkonsekvenser (DPIA)*. 17th July 2019. URL: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/vurdere-personvernkonsekvenser/vurdering-av-personvernkonsekvenser/ (visited on 11/02/2022).

FIDO Alliance. *The Case for Replacing Passwords with Biometrics*. URL: https://fidoalliance.org/wp-content/uploads/2014/12/3.pdf (visited on 13/01/2022).

— *What is FIDO?* URL: https://fidoalliance.org/what-is-fido/ (visited on 13/01/2022).

Gerrand, Andrew. *Godoc*. 31st Mar. 2011. URL: https://go.dev/blog/godoc (visited on 12/01/2022).

Golang-migrate. *Database migrations. CLI and Golang library.* URL: https://github.com/golang-migrate/migrate.

Gomez-Barrero, Marta et al. 'Multi-biometric template protection based on bloom filters'. In: *Information Fusion* 42 (2018), pp. 37–50. ISSN: 1566-2535. DOI: https://doi.org/10.1016/j.inffus.2017.10.003. URL: https://www.sciencedirect.com/science/article/pii/S1566253516301233.

Huang, Faqun. 'Post-completion Error in Software Development'. In: (2016). URL: http://homepage.tudelft.nl/6c92j/CitingMyWork/ASME%202012/1%20conference%20paper/p108-huang.pdf (visited on 22/03/2022).

Huang, Fuqun. *Human Error Analysis in Software Engineering.* 2016. URL: https://www.intechopen.com/chapters/54996 (visited on 22/03/2022).

IETF-Trust and D. Hardt. *The OAuth 2.0 Authorization Framework.* URL: https://datatracker.ietf.org/doc/html/rfc6749 (visited on 24/01/2022).

Information Commissioner's Office. *Data Protection Impact Assessments (DPIAs).* URL: https://ico.org.uk/for-organisations/guide-to-data-protection/guide-to-the-general-data-protection-regulation-gdpr/data-protection-impact-assessments-dpias/ (visited on 11/02/2022).

Internet Engineering Task Force Trust. *OAuth 2.0 - Implicit Grant.* URL: https://datatracker.ietf.org/doc/html/rfc6749#section-4.2.

ISO/IEC. *24745:2011, Security Techniques, Biometric information protection.* 2011.

Jones, M.B. and D. Hardt. *The OAuth 2.0 Authorization Framework: Bearer Token Usage.* URL: https://datatracker.ietf.org/doc/html/rfc6750 (visited on 24/01/2022).

Kruglov, Kirill. *Biometric data processing and storage system threats.* URL: https://ics-cert.kaspersky.com/media/Threats_to_Biometrics_FINAL_ENG.pdf (visited on 02/02/2022).

Lodderstedt, T. et al. *OpenID Connect for Identity Assurance 1.0.* URL: https://openid.net/specs/openid-connect-4-identity-assurance-1_0-ID3.html (visited on 25/04/2022).

Mandiant. *Advanced Persistent Threat Groups.* URL: https://www.mandiant.com/resources/apt-groups (visited on 15/03/2022).

Mazzarolo, Guerrino and Anca Delia Jurcut. *Insider threats in Cyber Security: The enemy within the gates.* URL: https://arxiv.org/ftp/arxiv/papers/1911/1911.09575.pdf (visited on 22/03/2022).

Microsoft. *Simplified Implementation of Microsoft Security Development Lifecycle.* 4th Oct. 2010. URL: https://www.microsoft.com/en-us/securityengineering/sdl (visited on 13/01/2022).

Norwegian University of Science and Technology. *Informasjonsklassifisering - informasjonssikkerhet.* URL: https://i.ntnu.no/wiki/-/wiki/Norsk/Informasjonsklassifisering+-+informasjonssikkerhet.

— *Risiko og Sårbarhetsanalyse på NTNU.* URL: https://i.ntnu.no/documents/1306938287/1307171093/Presentasjon+ROS-VS.pdf/e0614d01-f2ff-46f3-84dc-516870be5166?t=1536924541534&status=0 (visited on 08/04/2022).

Nyblom, Philip et al. *The Root Causes of Compromised Accounts at the University.* July 2019.

OneLogin. *Biometric authentication, the good, the bad, and the ugly.* URL: https://www.onelogin.com/learn/biometric-authentication (visited on 25/01/2022).

Open Web Application Security Project(OWASP) foundation. *OWASP Top Ten.* Oct. 2021. URL: https://owasp.org/www-project-top-ten/ (visited on 08/04/2022).

OpenID Foundation. *OpenID Certification.* URL: https://openid.net/certification/#OPs (visited on 26/01/2022).

OpenID-Foundation. *OpenID connect welcome page.* URL: https://openid.net/connect/ (visited on 22/01/2022).

ORY. *Extensible security first OAuth 2.0 and OpenID Connect SDK for Go.* URL: https://github.com/ory/fosite (visited on 20/02/2022).

— *Kratos.* URL: https://www.ory.sh/kratos/.

— *OpenID Certified™ OpenID Connect and OAuth Provider.* URL: https://github.com/ory/hydra (visited on 20/02/2022).

Phillips, P.Jonathon et al. 'The FERET database and evaluation procedure for face-recognition algorithms'. In: *Image and Vision Computing* 16.5 (1998), pp. 295–306. ISSN: 0262-8856. DOI: https://doi.org/10.1016/S0262-8856(97)00070-X. URL: https://www.sciencedirect.com/science/article/pii/S026288569700070X.

PostgreSQL. *Encryption Options.* URL: https://www.postgresql.org/docs/current/encryption-options.html (visited on 21/02/2022).

Powell, Ron. *Benefits and challenges of using monorepo development practices.* URL: https://circleci.com/blog/monorepo-dev-practices/ (visited on 18/02/2022).

Rountree, Derrick. *What Is Federated Identity?* URL: https://www.sciencedirect.com/topics/computer-science/biometric-authentication (visited on 25/01/2022).

Sakimura, N. et al. *OpenID Connect Core 1.0 incorporating errata set 1.* URL: https://openid.net/specs/openid-connect-core-1_0.html (visited on 24/01/2022).

Sakimura, N. et al. *Proof Key for Code Exchange by OAuth Public Clients.* URL: https://datatracker.ietf.org/doc/html/rfc7636 (visited on 26/04/2022).

Sandhya, Mulagala and Munaga V. N. K. Prasad. 'Biometric Template Protection: A Systematic Literature Review of Approaches and Modalities'. In: *Biometric Security and Privacy: Opportunities & Challenges in The Big Data Era.* Ed. by Richard Jiang et al. Springer International Publishing, 2017, pp. 323–370.

ISBN: 978-3-319-47301-7. DOI: 10.1007/978-3-319-47301-7_14. URL: https://doi.org/10.1007/978-3-319-47301-7_14.

Stretchr. *A toolkit with common assertions and mocks that plays nicely with the standard library*. URL: https://github.com/stretchr/testify.

The Kubernetes Authors. *Network Policies*. Apr. 2022. URL: https://kubernetes.io/docs/concepts/services-networking/network-policies/ (visited on 20/04/2022).

— *Pod Security Admission*. Jan. 2022. URL: https://kubernetes.io/docs/concepts/security/pod-security-admission/ (visited on 20/04/2022).

The National Institute of Standards and Technology. *Computer security resource center*. URL: https://csrc.nist.gov/glossary/term/defense_in_depth.

Trzeciak, Randall F., Andrew Preston Moore and Dawn M. Cappelli. *The CERT Guide to Insider Threats: How to Prevent, Detect, and Respond to Information Technology Crimes (Theft, Sabotage, Fraud)*. Addison-Wesley, 2012.

Tumkevič, Agnija. *CYBERSECURITY IN CENTRAL EASTERN EUROPE: FROM IDENTIFYING RISKS TO COUNTERING THREATS*. 2016. URL: https://www.journals.vu.lt/BJPS/article/view/10337/8246 (visited on 22/03/2022).

Tundis, Andrea et al. *CHALLENGES AND AVAILABLE SOLUTIONS AGAINST ORGANIZED CYBER-CRIME AND TERRORIST NETWORKS*. 2018. URL: https://www.witpress.com/elibrary/wit-transactions-on-the-built-environment/174/36535 (visited on 22/03/2022).

Union, European. *General Data Protection Regulation*. URL: https://gdpr.eu/ (visited on 21/02/2022).

Vercel. *Authentication for Next.js*. URL: https://next-auth.js.org (visited on 26/04/2022).

Visual Paradigm. *What is User Story?* URL: https://www.visual-paradigm.com/guide/agile-software-development/what-is-user-story/ (visited on 31/01/2022).

Appendix

A Project plan

NTNU
Kunnskap for en bedre verden

Department of Information Security and Communication Technology

DCSG2900 - Bachelor Thesis Bachelor of Science in Digital Infrastructure and Cyber Security

Project Plan

Author:
Aksel Skaar Leirvaag, William Eide Seiner, Simen Ramberg

January, 2022

Table of Contents

List of Figures

List of Tables

Glossary

Docker A tool for building containers. 7

GitHub Website for hosting source code and issue management. 4, 6, 7, 9

Go A programming language. 6

Godoc Go documentation standard and tool. 6

Kubernetes Automated container orchestration tool. 1

Acronyms

Diri Digitized intelligent risk identification. 7

NTNU Norwegian University of Science and Technology. 1

OIDC OpenID Connect. 1–3, 9

OP OpenID Provider. 9

RP Relaying Party. 9

SDK Software Development Kit. 1

SDL Microsoft Security Development Lifecycle. 5

WIP Work in Progress. 5

1 Goals

1.1 Background

Weak passwords are responsible for over 80% of successful data breaches, therefore, passwords pose a significant risk for both companies and individuals.[1] Another issue related to passwords is that over 40% reuse their passwords across websites.[2] The consequence of this is that users might use the same password for sensitive services and on sites that does not have strong security. Using biometrics as the authentication factor, instead of passwords, can mitigate these risks[3]. In addition, it will increase usability for the end-user compared to passwords, which are required to be remembered and can easily be forgotten.

Mobai is a spin-off company from Norwegian University of Science and Technology (NTNU), and has several biometric components that are used for facial recognition and attack detection. Their mission is to democratize biometrics, enhance security for all, and protect users' privacy.

They have developed two software development kits (SDK) for Android and iOS, but are lacking in other technologies which can more easily commercialize their biometric products. By implementing a widely accepted standard, other businesses can more easily incorporate biometrics as the authentication factor, reducing the risk associated with passwords while improving security and end-user experience.

1.2 Project goals

Main goals

- Develop a prototype that implements the OpenID Connect (OIDC) protocol using the Authorization Code Flow.
- Develop a demo showcasing the OIDC server capabilities.
- The authentication process is intuitive and fast to use.
- Minimize technical implementation for customers.
- Secure and privacy preserving by default.

Stretch goals

- Create secure Kubernetes configuration for deployment.
- Implement readiness and liveliness probes for Kubernetes deployment.
- Research other privacy preserving technical techniques.
- Penetration test of the services.

2 Scope

2.1 Project description

The project's goal is to develop a prototype that authenticates a user using facial biometrics as the authentication factor. Mobai has suggested implementing the OIDC authentication protocol, as it is a battle-tested and popular standard used by many large enterprises. [4]

The end-user will authenticate themselves by submitting an image of their face, which will be compared to a registered reference image saved as a face-template. Privacy is an essential part of Mobai's values, therefore, the development of the authentication service must embed a strong security and privacy focus from the requirement phase until production.

2.2 Constraints

The user will be authenticated using an selfie image, and comparing it with an image saved from a first-time registration. The service for handling this registration will be outside of the scope of the project. The logic of registering the customer that wants to utilize Mobai's OIDC service will also not be considered.

3 Project Organization

In order to create a common understanding of the project's organization and structure, the team has outlined specific roles and domains for each member.

3.1 Project roles

The team members are responsible for certain areas to ensure quality is maintained throughout the project. All members are considered to have the role as a developer and contributor to the bachelor thesis. The following roles and its responsibilities are defined:

Project leader : Aksel Skaar Leirvaag

- Plan meetings and agenda.
- External and internal communication.
- Ensure overall project progression.

Secretary : Simen Ramberg

- Development model and workflow is followed.
- Write meeting minutes.

Organizer : William Seiner

- Responsible for documentation.
- Organize and maintain thesis related documents.

3.2 Domain responsibilities

Each member is expected to have an equal understanding of each domain, however, in case of confusion or disagreement within the team, the member responsible for the domain under confusion must provide facts to clear up the lack of understanding. In addition, the domain expert must organize knowledge sharing sessions, if any of the other team members lack knowledge about a domain.

- **Risk analysis** : Simen Ramberg
- **OIDC** : William Seiner
- **Go programming language** : Aksel

3.3 Routines and rules

It is important for the group to have good routines and a continuous workflow during the writing of the bachelor thesis. Therefore, the group has agreed in having stand-up sessions every day at 09:30. This is done to review the team members work and what they have achieved since last time, and their next task. GitHub's Kanban board will be used to delegate issues that are currently under progress.

Routines:

- Rooms are booked 2 weeks ahead.
- During stand-up each issue on the Kanban board is discussed.
- Take a backup of the LaTeX document.
- Follow group regulations 5.2

3.4 Group guidelines

As a group it is crucial to follow certain guidelines to improve our workflow and build mutual respect for each other. Therefore, a set of guidelines have been created to fulfil these requirements.

Column name	**Description**
Respect	Every member should have equal respect for each other. Interruptions should not occur when a member is making a comment during discussions. Every member's opinion should count, as it contributes to a constructive discussion. Lastly it is crucial that every member is reliable and honest.
Equal amount of work	The group has a shared responsibility of the tasks assigned. This means even though you are signed an issue, everyone should take part of seeing it through. Feedback should be constructive and enable the person to finish their task by themselves.
Be open to compromise	Be a good listener, be open minded and willing to cooperate with other members ideas. If disagreements occur, a vote on the outcome should be held, and the team leader has the final word.
Communication	Share all your ideas, even though your idea might sound off, as it can clear up confusion or shed some light on the task at hand. Everyone's voice should be heard, both about their ideas and challenges.
Time management	Everyone are required to attend to all group meetings unless something more important makes them unable to attend. If a member is unavailable it should be communicated as soon as possible. During meetings everyone should be present in the moment and avoid sidetracks.

Table 1: Group guidelines

3.5 Group contract

Contract found at page 13, signed by the project members.

4 Planning

4.1 Development model

The duration of the project is spanning over five months, therefore, it is important to use a flexible development model which enables us to handle unexpected issues or changes. Additionally, creating incremental features will be helpful to receive early feedback from Mobai regarding if our implementation is aligned with the expected product and requirements. These traits can be found in several of the frameworks under the agile methodologies.

Scrum is a popular agile framework, however, it requires a lot of ceremonies and artifacts which can be unnecessary with regards to the size of our team. That being said, there are several of the processes, such as stand-up, retrospective and backlog refinement, which would be advantageous.

Kanban is another framework which provides a great workflow visualization tool to enhance collaboration within our team. The kanban board will be used during the stand-ups to display the tasks which are being worked on, and highlight any pains or issues together. The Work in Progress (WIP) limit will encourage us to finish each issue without starting a new one, and visualize if any tasks have been stuck in a column for too long.

Scrumban is a framework which incorporates features from both scrum and kanban. This suits the team well since we want to use the workflow mentions from scrum with a combination of the kanban board and WIP limits.

The system will process privacy sensitive data, therefore, it is important that the privacy focus is holistic. It can not be an afterthought and must be incorporated from the planning and requirements phase until final delivery. Datatilsynet, The Norwegian Data Protection Authority, have created a guideline for software development with built-in privacy [5]. In the guideline they recommend using a software framework such as Microsoft Security Development Lifecycle (SDL) which will be used throughout the project [6].

4.2 Development model implementation

4.2.1 The kanban board

The kanban board will consist of the following columns. The number to the right of the column name is the defined WIP limit for the column.

Column name	Description
Backlog	The task is fully defined and ready for development.
Blocked (2)	The task has been started, but is blocked by either internal or external dependency.
In Progress (3)	The task is actively being worked on.
In Review (1)	All requirements are fulfilled and the task is ready to be reviewed by a team member.
Done	The task has been approved by all team members.

Table 2: Kanban Board Columns

The WIP limit for the “Blocked” column is kept at two to avoid starting on a task without reflecting on any potential unknown dependencies. The dependencies of a task should be defined before beginning on the task, and unexpected scenarios should be kept at a minimum.

Asynchronous work should be avoided since it results in decreased efficiency and quality. Therefore, the limit for “In Progress” is set to three which is the amount of people in the team.

The limit for “In Review” is set to one to avoid any issues being stuck in review for too long, and unnecessary context switching between old tasks in review and the task that is being worked on by the team member.

4.2.2 Standup

Every day from Monday to Friday at 09:30 the team will conduct the stand-up. The kanban board will be used where each issue on the board will be discussed by the team member which has the ticket assigned. The duration of the meeting should be kept at 15 minutes. If any issue requires further discussion the team should agree upon a new time slot in order to conclude the conversation.

4.2.3 Retrospective

Every two weeks the team will have a retrospective together. During the meeting every member will have the option to highlight achievements, frustrations and things that have gone well. Pain-points will be discussed, followed by solutions for the addressed problems.

4.2.4 Supervisor meeting

There will be a weekly meeting with the supervisor on each Thursday at 14:00 to 14:45. During the meeting the team must present the progress and current status of the bachelor thesis.

4.3 Quality assurance

The end product will be used in a production environment and additional features will be added to further extend the capabilities of the solution. Good documentation, standardization and testing is therefore imperative for a smooth transition after we have concluded our defined work.

4.3.1 Documentation

Documentation should be produced during the development and be revised before the hand-off to Mobai. The application will be written in Go, and the team will use Godoc as the documentation tool and follow its guidelines as outlined in the article [7].

4.4 Workflow

For every development task an issue will be created in GitHub. Each issue must be fully descriptive of what the requirements are. A milestone will be created to gain an overview over issues that needs to be finished to fulfill the requirements of the milestone.

When moving the task to “In Progress” a merge request will be created which links to the given issue. On task completion it will be moved to “In Review”, and each team member will review the code changes. After the members have confirmed that the code fulfills the requirements for the issue, the issue will then be approved and merged to the main branch.

Pushing directly to the main branch will be prohibited by a GitHub configuration. This guarantees that the code on the main branch is fully functional and satisfies our code quality requirements.

On each push to a non-main branch, the GitHub pipeline will execute the lint and test stage to continuously check our acceptance criterias. When a branch is merged to the main branch all the previous stages will be executed in addition to building the Docker image. It is important that the automated processes are leveraged to make the code review easier, since the static analysis tool and tests will catch any low-hanging fruits.

4.5 Code quality

Gofmt will be used for formatting and increasing readability of the code. A Github pipeline will be created for the repository and golangci-lint, a static analyzer tool, will be added as a job to catch mistakes at an early stage of the development cycle. The formatter will also be added to the same stage.

4.6 Testing

Quality tests and good coverage are essential to safeguard against any unwanted behaviours both during the development and in the future iterations. The test automation pyramid is used to focus our test coverage to where it provides the most value and efficiency [8]. The majority of our tests will be unit tests which will assert correct behavior at the function-level without too many dependencies on other parts of our code. Next the integration tests will run expectations if our different modules are working correctly together. Lastly, the acceptance tests will be used to ensure that our requirements are met. Since we are using an agile development model, features should be incremental and presentable during the meetings with Mobai. This provides us with the opportunity for early testing by them and feedback.

4.7 Risk assessment

Digitized intelligent risk identification (Diri) was used to create the risk assessment for the project. The assessment can be found as an appendix at page 15. The risk assessment produces a complete analyses of the project assignment using a bow-tie model to connect causes, events and consequences together. Treatments are then applied to the consequences and causes of an event to reduce the risk of these events occurring. There will also be produced a risk matrix from before and after the treatments are implemented.

4.8 Tools

The table below lists tools that will be used during the project.

Name	**Description**
Github	Version control
Github Actions	Continuous integration and continuous delivery (CI/CD) platform.
Timenotes	Time management
Messenger/Discord	Internal communication
Overleaf	Report writing
Microsoft Teams	External communication
Goland	Integrated development environment (IDE) for Go
Golangci-lint	Go lint aggregator
Gofmt	Go formatting
Gosec	Go security static analysis
Trivy	Container vulnerability scanner

Table 3: Tools

5 Implementation Plan

5.1 Tasks

The project consists of the following high-level tasks:

1. Risk analysis.
2. OIDC Relaying Party (RP) demo.
3. Frontend demo which uses RP for authentication.
4. OpenID Provider (OP).

5.2 Timeline

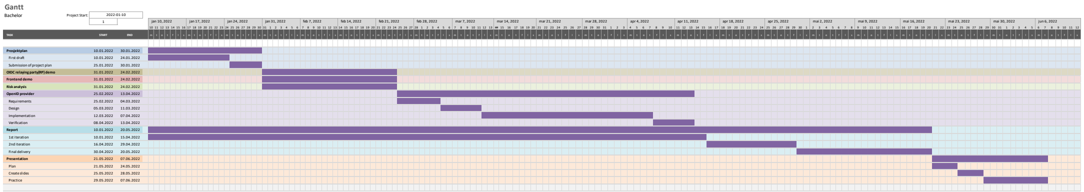
The project will be organized into milestones that will be created on GitHub, and each milestone will be further subdivided into issues. The required research to accomplish a task is included in the timeline of each milestone.

The following milestones are included in the project:

1. **1st draft of project plan 24.01.2022**
2. **Project plan 30.01.2022**: Finalized and delivered to supervisor.
3. **OIDC RP demo 24.02.2022**: Implements the authorization code flow.
4. **Frontend demo 24.02.2022**: Uses the RP demo to login a user.
5. **Risk analysis 24.02.2022**: Risk assessment of the OP.
6. **Requirements 04.03.2022**: Define requirements for the OP.
7. **Design 11.03.2022**: Architecture and design of the OP.
8. **Implementation 07.04.2022**: Minimum viable product of the OP.
9. **Verification 13.04.2022**: Verify that the implementation matches the requirements.
10. **1st draft of report 15.04.2022**
11. **2nd draft of report 29.04.2022**
12. **Deliver report 20.05.2022**
13. **Planning of presentation 24.05.2022**
14. **Create slides 28.05.2022**
15. **Practice for presentation 07.06.2022**

5.3 Gantt chart

Zoom to enlarge the chart.



Gantt

Bachelor

Project Start: 2022-01-10

1

TASK	START	END
Prosjektplan	10.01.2022	30.01.2022
First draft	10.01.2022	24.01.2022
Submission of project plan	25.01.2022	30.01.2022
OIDC relaying party(RP) demo	31.01.2022	24.02.2022
Frontend demo	31.01.2022	24.02.2022
Risk analysis	31.01.2022	24.02.2022
OpenID provider	25.02.2022	13.04.2022
Requirements	25.02.2022	04.03.2022
Design	05.03.2022	11.03.2022
Implementation	12.03.2022	07.04.2022
Verification	08.04.2022	13.04.2022
Report	10.01.2022	20.05.2022
1st iteration	10.01.2022	15.04.2022
2nd iteration	16.04.2022	29.04.2022
Final delivery	30.04.2022	20.05.2022
Presentation	21.05.2022	07.06.2022
Plan	21.05.2022	24.05.2022
Create slides	25.05.2022	28.05.2022
Practice	29.05.2022	07.06.2022

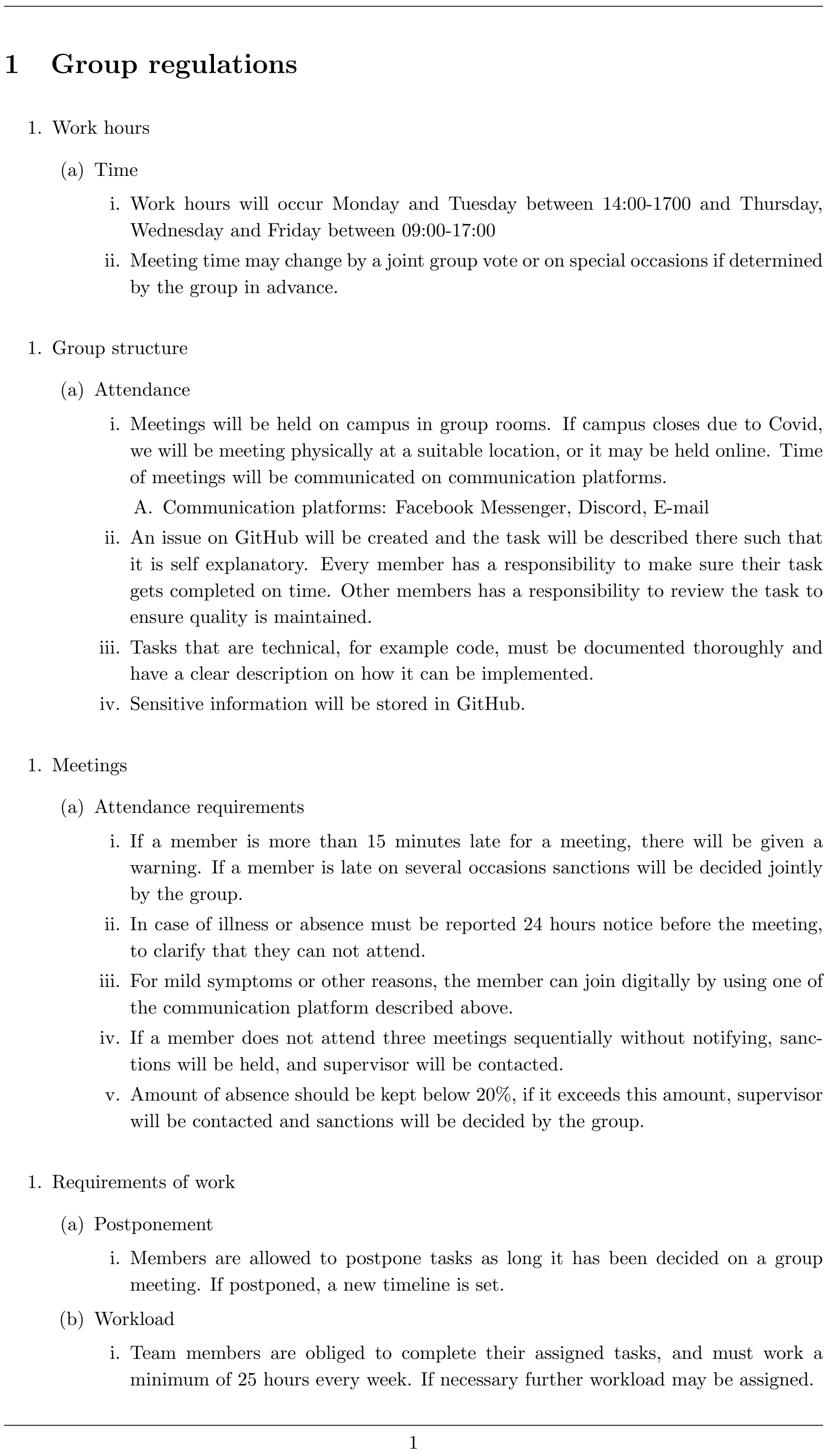
Figure 1: Gantt chart

References

1. Alliance F. What is FIDO? Available from: https://fidoalliance.org/what-is-fido/ [Accessed on: 2022 Jan 13]
2. Nyblom P, Wangen GB, Kianpour M and Østby G. The Root Causes of Compromised Accounts at the University. 2019 Jul
3. Alliance F. The Case for Replacing Passwords with Biometrics. Available from: https://fidoalliance.org/wp-content/uploads/2014/12/3.pdf [Accessed on: 2022 Jan 13]
4. OpenID-Foundation. OIDC FAQ. Available from: https://openid.net/connect/faq/ [Accessed on: 2022 Jan 16]
5. Datatilsynet. Programvareutvikling med innebygd personvern. 2019 Aug 20. Available from: https://www.datatilsynet.no/rettigheter-og-plikter/virksomhetenes-plikter/innebygd-personvern/programvareutvikling-med-innebygd-personvern [Accessed on: 2022 Jan 13]
6. Microsoft. Simplified Implementation of Microsoft Security Development Lifecycle. 2010 Oct 4. Available from: https://www.microsoft.com/en-us/securityengineering/sdl [Accessed on: 2022 Jan 13]
7. Gerrand A. Godoc. 2011 Mar 31. Available from: https://go.dev/blog/godoc [Accessed on: 2022 Jan 12]
8. Cohn M. Succeeding with Agile. Presidio Press, 2009

Appendix

A Group contract



1 Group regulations

1. Work hours
   (a) Time
      i. Work hours will occur Monday and Tuesday between 14:00-1700 and Thursday, Wednesday and Friday between 09:00-17:00
      ii. Meeting time may change by a joint group vote or on special occasions if determined by the group in advance.

1. Group structure
   (a) Attendance
      i. Meetings will be held on campus in group rooms. If campus closes due to Covid, we will be meeting physically at a suitable location, or it may be held online. Time of meetings will be communicated on communication platforms.
         A. Communication platforms: Facebook Messenger, Discord, E-mail
      ii. An issue on GitHub will be created and the task will be described there such that it is self explanatory. Every member has a responsibility to make sure their task gets completed on time. Other members has a responsibility to review the task to ensure quality is maintained.
      iii. Tasks that are technical, for example code, must be documented thoroughly and have a clear description on how it can be implemented.
      iv. Sensitive information will be stored in GitHub.

1. Meetings
   (a) Attendance requirements
      i. If a member is more than 15 minutes late for a meeting, there will be given a warning. If a member is late on several occasions sanctions will be decided jointly by the group.
      ii. In case of illness or absence must be reported 24 hours notice before the meeting, to clarify that they can not attend.
      iii. For mild symptoms or other reasons, the member can join digitally by using one of the communication platform described above.
      iv. If a member does not attend three meetings sequentially without notifying, sanctions will be held, and supervisor will be contacted.
      v. Amount of absence should be kept below 20%, if it exceeds this amount, supervisor will be contacted and sanctions will be decided by the group.

1. Requirements of work
   (a) Postponement
      i. Members are allowed to postpone tasks as long it has been decided on a group meeting. If postponed, a new timeline is set.
   (b) Workload
      i. Team members are obliged to complete their assigned tasks, and must work a minimum of 25 hours every week. If necessary further workload may be assigned.

1. Sanctions
   (a) Consequences
      i. If the rules described above are not followed, it may result in a loss of their place in the group after discussing the situation with the supervisor.

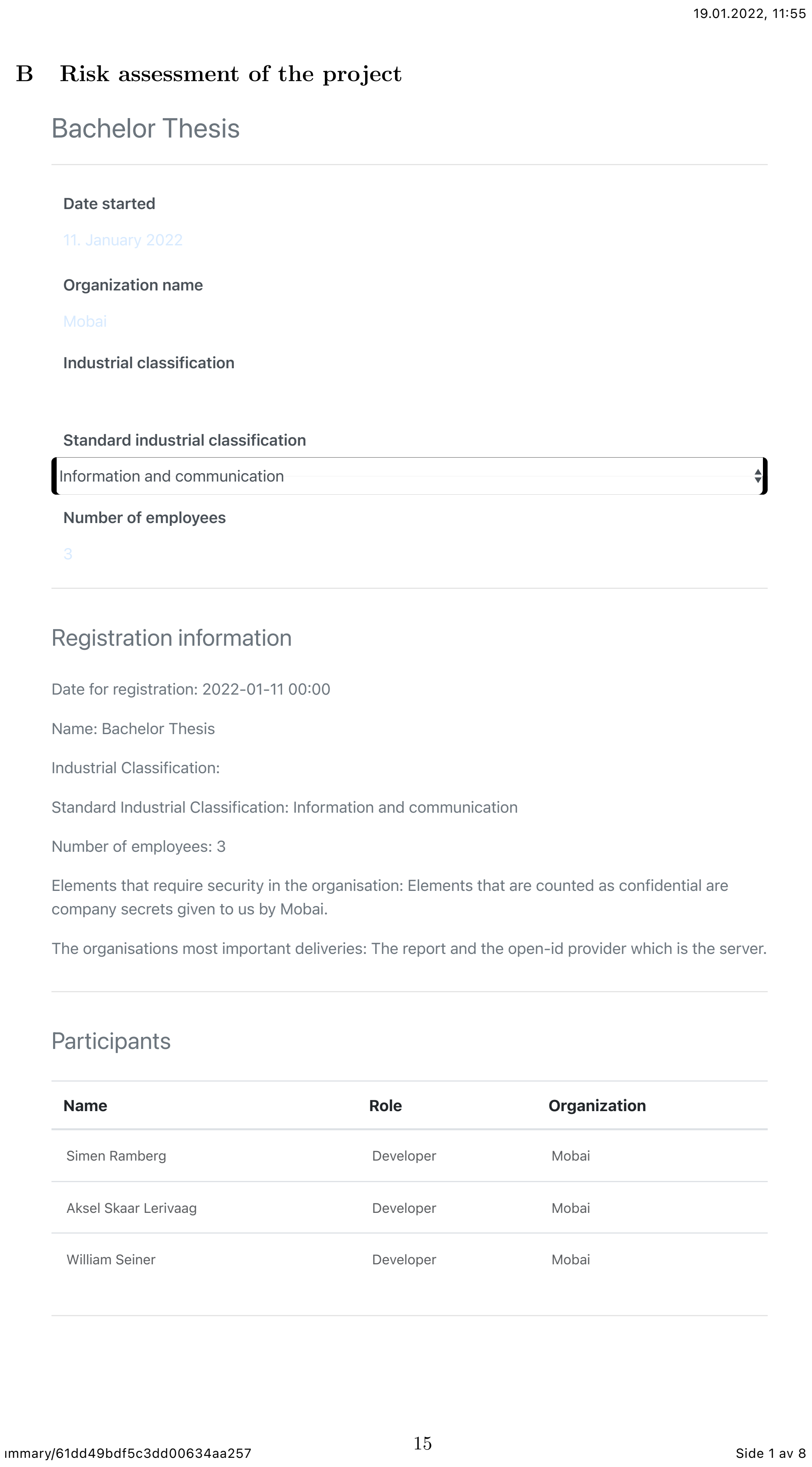
1. Change of rules
   (a) Rules may be changed by agreement of the group. This will happen through voting.

Signature and date William Seiner 18.01.2022

Signature and date Simen Ramberg 18.01.2022

Signature and date Aksel Skaar Leirvaag 18.01.2022

B Risk assessment of the project



Bachelor Thesis

Date started

11. January 2022

Organization name

Mobai

Industrial classification

Standard industrial classification

Information and communication

Number of employees

3

Registration information

Date for registration: 2022-01-11 00:00

Name: Bachelor Thesis

Industrial Classification:

Standard Industrial Classification: Information and communication

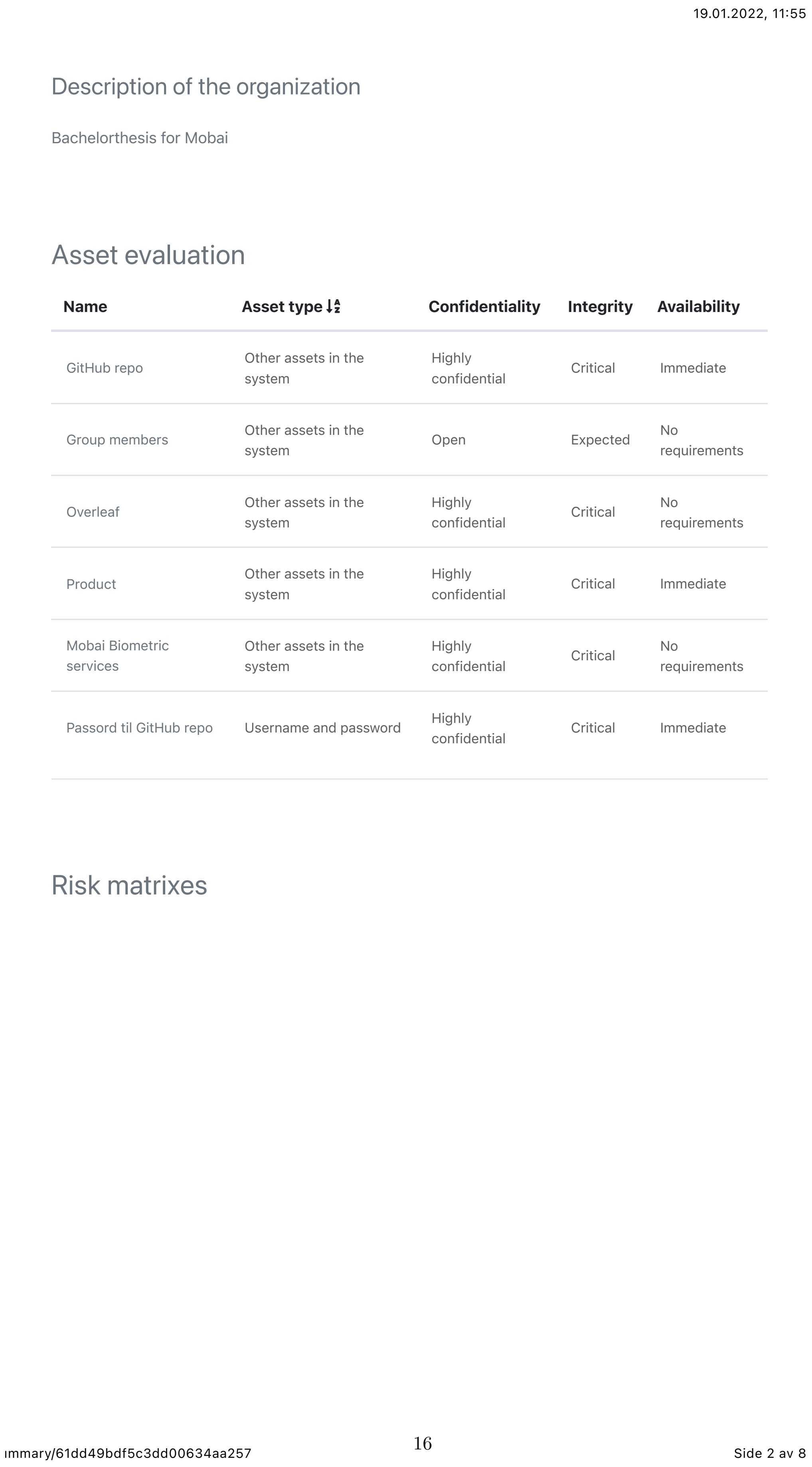
Number of employees: 3

Elements that require security in the organisation: Elements that are counted as confidential are company secrets given to us by Mobai.

The organisations most important deliveries: The report and the open-id provider which is the server.

Participants

Name	Role	Organization
Simen Ramberg	Developer	Mobai
Aksel Skaar Lerivaag	Developer	Mobai
William Seiner	Developer	Mobai



Description of the organization

Bachelorthesis for Mobai

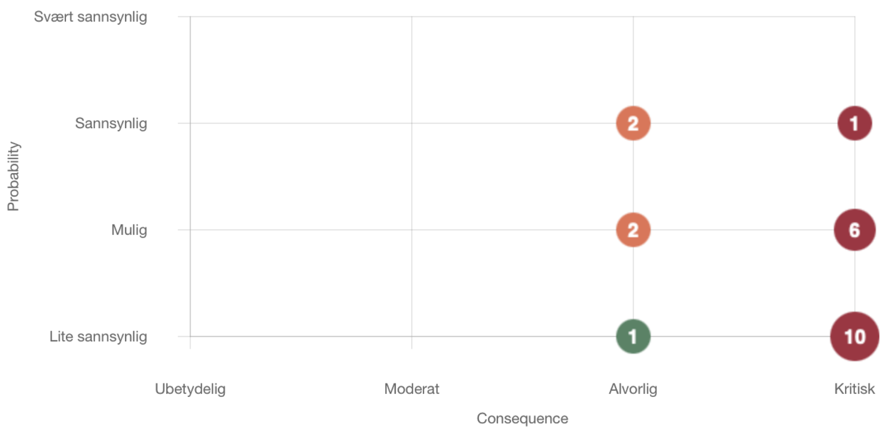
Asset evaluation

Name	Asset type ↓	Confidentiality	Integrity	Availability
GitHub repo	Other assets in the system	Highly confidential	Critical	Immediate
Group members	Other assets in the system	Open	Expected	No requirements
Overleaf	Other assets in the system	Highly confidential	Critical	No requirements
Product	Other assets in the system	Highly confidential	Critical	Immediate
Mobai Biometric services	Other assets in the system	Highly confidential	Critical	No requirements
Passord til GitHub repo	Username and password	Highly confidential	Critical	Immediate

Risk matrixes

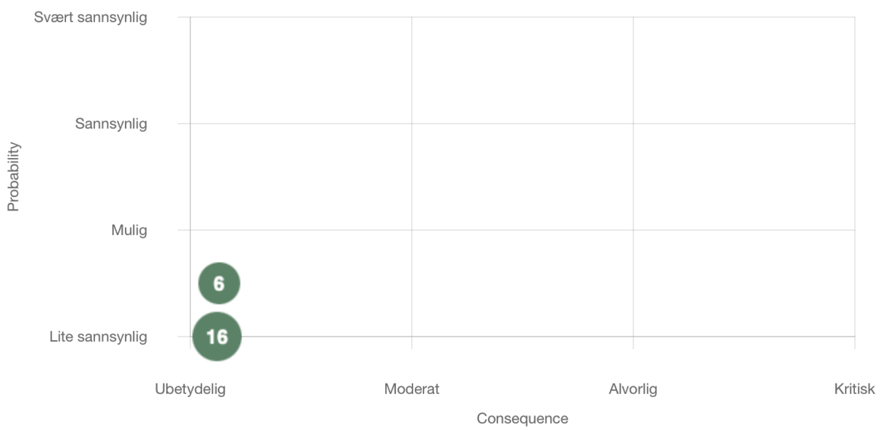
Before treatments

Risk Matrix



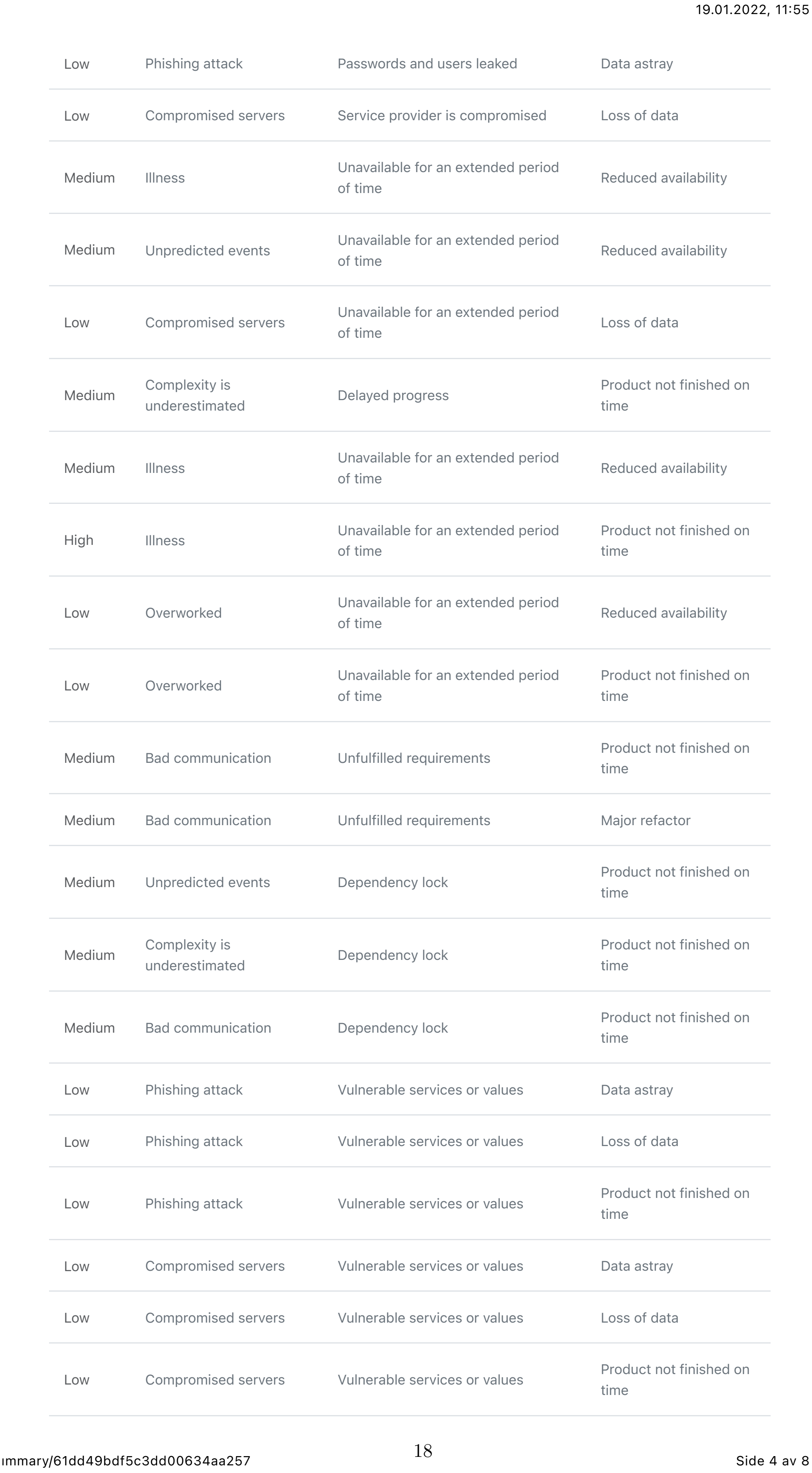
After treatment

Risk Matrix



Risk assessment

Risk	Cause	Event	Consequence

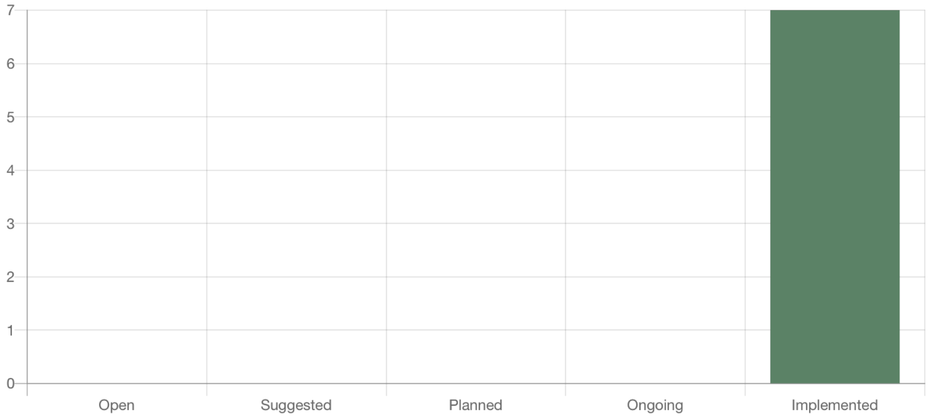


Low	Phishing attack	Passwords and users leaked	Data astray
Low	Compromised servers	Service provider is compromised	Loss of data
Medium	Illness	Unavailable for an extended period of time	Reduced availability
Medium	Unpredicted events	Unavailable for an extended period of time	Reduced availability
Low	Compromised servers	Unavailable for an extended period of time	Loss of data
Medium	Complexity is underestimated	Delayed progress	Product not finished on time
Medium	Illness	Unavailable for an extended period of time	Reduced availability
High	Illness	Unavailable for an extended period of time	Product not finished on time
Low	Overworked	Unavailable for an extended period of time	Reduced availability
Low	Overworked	Unavailable for an extended period of time	Product not finished on time
Medium	Bad communication	Unfulfilled requirements	Product not finished on time
Medium	Bad communication	Unfulfilled requirements	Major refactor
Medium	Unpredicted events	Dependency lock	Product not finished on time
Medium	Complexity is underestimated	Dependency lock	Product not finished on time
Medium	Bad communication	Dependency lock	Product not finished on time
Low	Phishing attack	Vulnerable services or values	Data astray
Low	Phishing attack	Vulnerable services or values	Loss of data
Low	Phishing attack	Vulnerable services or values	Product not finished on time
Low	Compromised servers	Vulnerable services or values	Data astray
Low	Compromised servers	Vulnerable services or values	Loss of data
Low	Compromised servers	Vulnerable services or values	Product not finished on time

Medium	Underestimated scope	Scope too big	Product not finished on time

Treatment status

Show treatment statuses



Diri Control Matrix

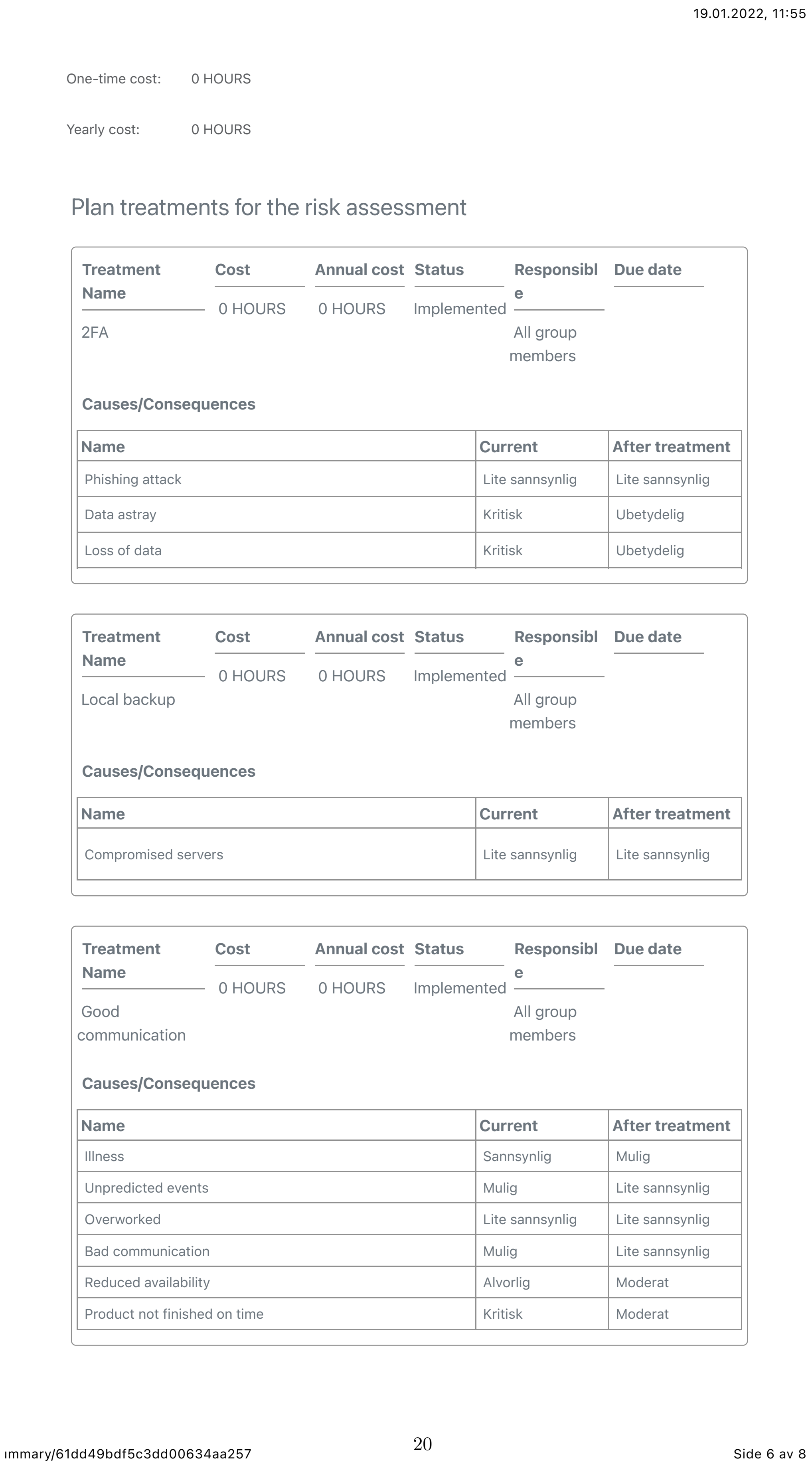
The table shows the distribution of the treatments from the risk assessment within the defined classes. Click "Expand" to see the distribution of treatment types. The colour distribution in the pie chart for each square illustrates the status of treatments. The numbers in the pie chart summarize how many of the proposed treatments have been introduced.

	Identify	Protect and maintain	Detect	Handle and recover	Uncategorized	Summary
Summary	⊘	2/2	⊘	5/5	⊘	7/7

Risk treatment plan

Costs

This is a summary of the costs estimated in each of the selected treatments.



One-time cost:	0 HOURS
Yearly cost:	0 HOURS

Plan treatments for the risk assessment

Treatment Name	Cost	Annual cost	Status	Responsible	Due date
2FA	0 HOURS	0 HOURS	Implemented	All group members	

Causes/Consequences

Name	Current	After treatment
Phishing attack	Lite sannsynlig	Lite sannsynlig
Data astray	Kritisk	Ubetydelig
Loss of data	Kritisk	Ubetydelig

Treatment Name	Cost	Annual cost	Status	Responsible	Due date
Local backup	0 HOURS	0 HOURS	Implemented	All group members	

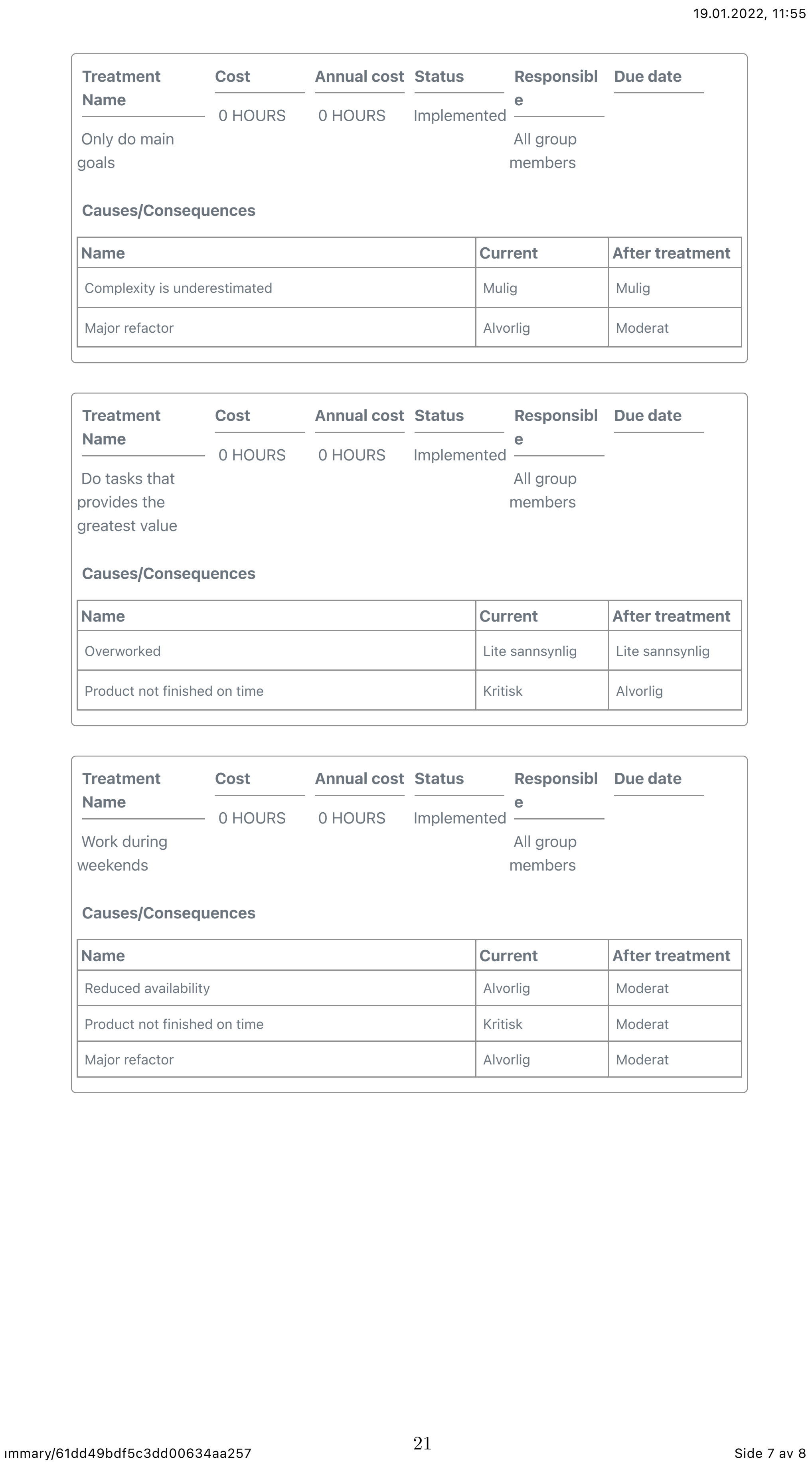
Causes/Consequences

Name	Current	After treatment
Compromised servers	Lite sannsynlig	Lite sannsynlig

Treatment Name	Cost	Annual cost	Status	Responsible	Due date
Good communication	0 HOURS	0 HOURS	Implemented	All group members	

Causes/Consequences

Name	Current	After treatment
Illness	Sannsynlig	Mulig
Unpredicted events	Mulig	Lite sannsynlig
Overworked	Lite sannsynlig	Lite sannsynlig
Bad communication	Mulig	Lite sannsynlig
Reduced availability	Alvorlig	Moderat
Product not finished on time	Kritisk	Moderat



Treatment Name	Cost	Annual cost	Status	Responsible	Due date
Only do main goals	0 HOURS	0 HOURS	Implemented	All group members	

Causes/Consequences

Name	Current	After treatment
Complexity is underestimated	Mulig	Mulig
Major refactor	Alvorlig	Moderat

Treatment Name	Cost	Annual cost	Status	Responsible	Due date
Do tasks that provides the greatest value	0 HOURS	0 HOURS	Implemented	All group members	

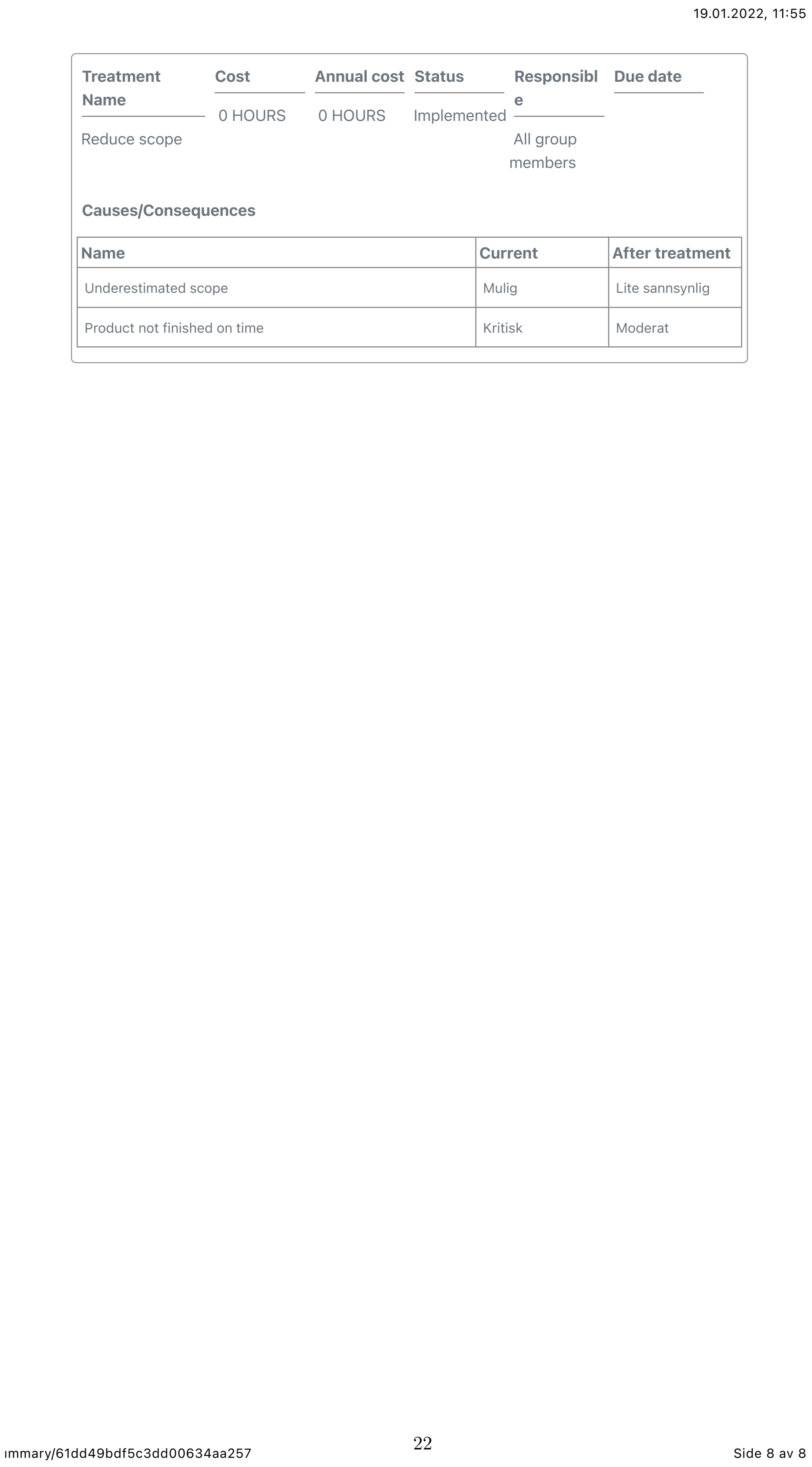
Causes/Consequences

Name	Current	After treatment
Overworked	Lite sannsynlig	Lite sannsynlig
Product not finished on time	Kritisk	Alvorlig

Treatment Name	Cost	Annual cost	Status	Responsible	Due date
Work during weekends	0 HOURS	0 HOURS	Implemented	All group members	

Causes/Consequences

Name	Current	After treatment
Reduced availability	Alvorlig	Moderat
Product not finished on time	Kritisk	Moderat
Major refactor	Alvorlig	Moderat

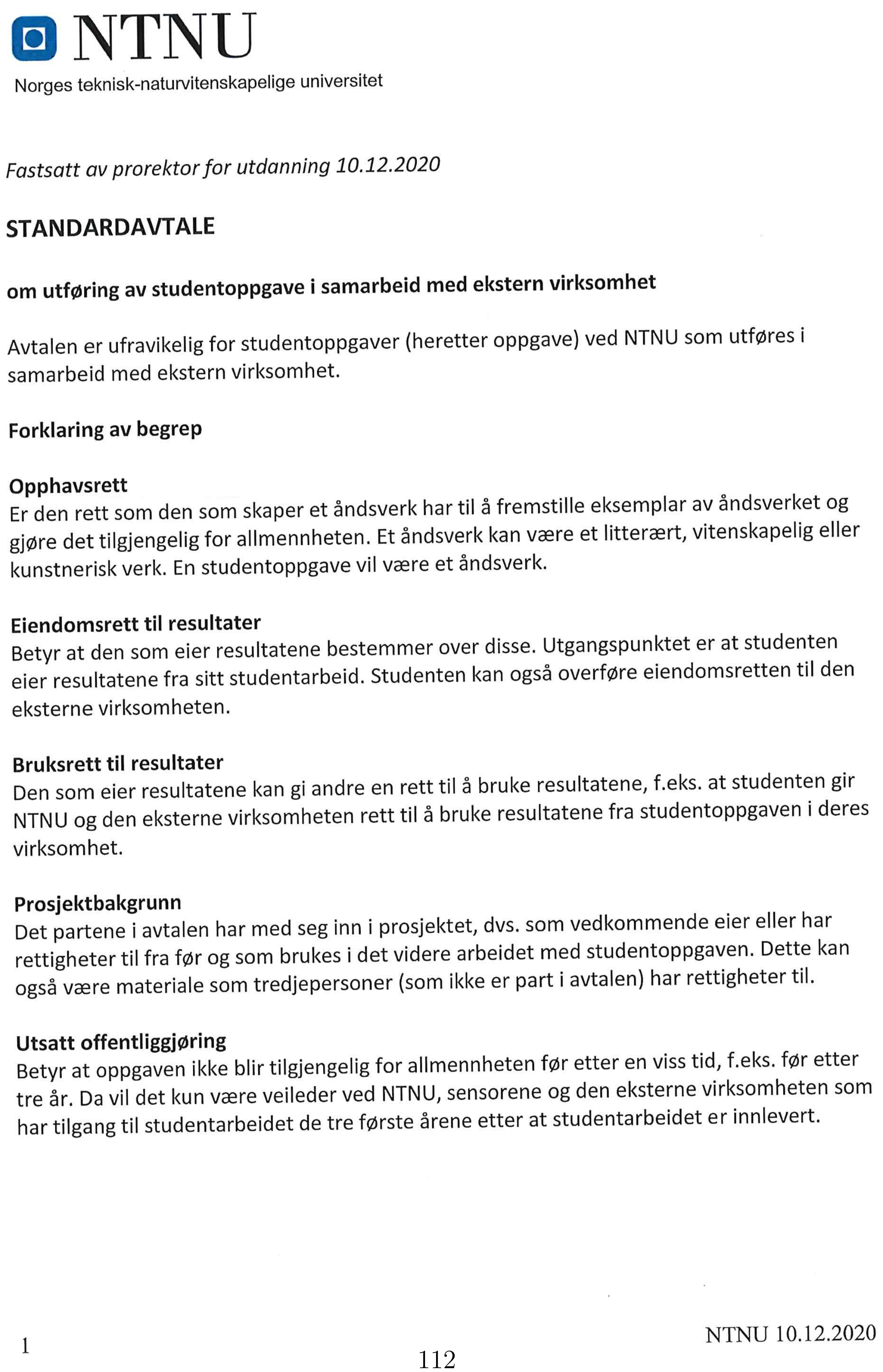


Treatment Name	Cost	Annual cost	Status	Responsible	Due date
Reduce scope	0 HOURS	0 HOURS	Implemented	All group members	

Causes/Consequences

Name	Current	After treatment
Underestimated scope	Mulig	Lite sannsynlig
Product not finished on time	Kritisk	Moderat

B Signed collaboration agreement with Mobai



NTNU
Norges teknisk-naturvitenskapelige universitet

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 eksemplar 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:
Informasjonssikkerhet og kommunikasjonsteknologi (IIK)

Veileder ved NTNU: Prof. Raghavendra Ramachandra
e-post og tlf. raghavendra.ramachandra@ntnu.no

Ekstern virksomhet:
Ekstern virksomhet sin kontaktperson, e-post og tlf.:
Brage Strand, Brage@mobai.bio, +4740490411

Student: Aksel Skaar Leirvaag
Fødselsdato: 02.07.1997

Ev. flere studenter[1] Simen Ramberg

William Seiner

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:	10.01.2022
Sluttdato:	20.05.2022

Oppgavens arbeidstittel er:

Facial authentication service with a privacy-preserving focus.

[1] Dersom flere studenter skriver oppgave i fellesskap, kan alle føres opp her. Rettigheter ligger da i fellesskap mellom studentene. Dersom ekstern virksomhet i stedet ønsker at det skal inngås egen avtale med hver enkelt student, gjøres dette.

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:

Ingen planlagte innkjøp. Eventuelle behov skal godkjennes av kontaktperson før innkjøpet

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 oppgaven[2]. 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.

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

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

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

	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. Retten er ikke-eksklusiv.

Alternativ b) (sett kryss) Unntak

X	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: Løsningene vil være tett knyttet på Mobais produktområde. Vederlagersfri overføring av alt eierskap til alle resultater er en forusetning for samarbeidet.

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
	ett år	
	to år	
	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.

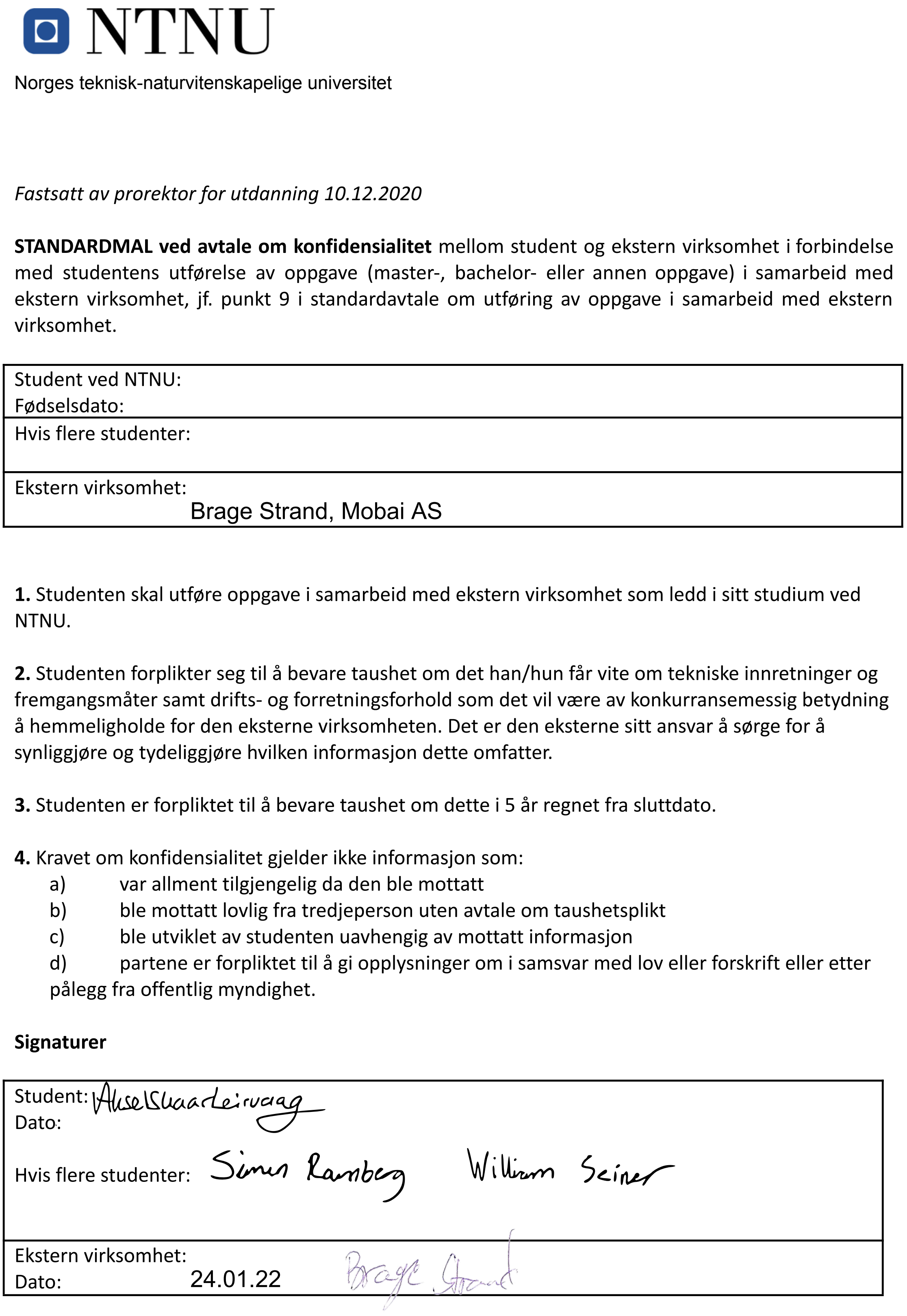
Eventuell uenighet som følge av denne avtalen skal søkes løst ved forhandlinger. Hvis dette ikke fører frem, er partene enige om at tvisten avgjøres ved voldgift i henhold til norsk lov. Tvisten avgjøres av sorenskriveren ved Sør-Trøndelag tingrett eller den han/hun oppnevner.

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: 15/01/2022
Ekstern virksomhet: Brage Strand Dato: 24.01.2022
Student: AkselSkaarLeirvaag Dato: 18.02.2022
Ev. flere studenter William Seiner 18.02.2022
Simen Ramberg 18.02.2022

C Signed confidentiality agreement with Mobai



NTNU

Norges teknisk-naturvitenskapelige universitet

Fastsatt av prorektor for utdanning 10.12.2020

STANDARDMAL ved avtale om konfidensialitet mellom student og ekstern virksomhet i forbindelse med studentens utførelse av oppgave (master-, bachelor- eller annen oppgave) i samarbeid med ekstern virksomhet, jf. punkt 9 i standardavtale om utføring av oppgave i samarbeid med ekstern virksomhet.

Student ved NTNU:
Fødselsdato:

Hvis flere studenter:

Ekstern virksomhet:
Brage Strand, Mobai AS

1. Studenten skal utføre oppgave i samarbeid med ekstern virksomhet som ledd i sitt studium ved NTNU.

2. Studenten forplikter seg til å bevare taushet om det han/hun får vite om tekniske innretninger og fremgangsmåter samt drifts- og forretningsforhold som det vil være av konkurransemessig betydning å hemmeligholde for den eksterne virksomheten. Det er den eksterne sitt ansvar å sørge for å synliggjøre og tydeliggjøre hvilken informasjon dette omfatter.

3. Studenten er forpliktet til å bevare taushet om dette i 5 år regnet fra sluttdato.

4. Kravet om konfidensialitet gjelder ikke informasjon som:

a) var allment tilgjengelig da den ble mottatt

b) ble mottatt lovlig fra tredjeperson uten avtale om taushetsplikt

c) ble utviklet av studenten uavhengig av mottatt informasjon

d) partene er forpliktet til å gi opplysninger om i samsvar med lov eller forskrift eller etter pålegg fra offentlig myndighet.

Signaturer

Student: Aksel Skaar Leirvaag
Dato:

Hvis flere studenter: Simen Ramberg William Seiner

Ekstern virksomhet:
Dato: 24.01.22 Brage Strand

D Threat actors

Threat Actor

Script Kiddies

Motivation:

Financial, vandalism, acknowledgement, curiosity

Intention:

Steal and sell data, gain access to data and delete or leak it, abuse, amusement, practice.

Attack method

Phising, credential stuffing, off the shelf malware

Capacity

Low

Capability

Low

Frequency

Low

Description

Actors with limited expertise of information technology, but exploit well-designed tools and malware created by experienced developers. They utilize zero-day or older vulnerabilities which can be found in older systems and software that lacks updates, using off the shelf toolkits and scripts. The actor is generally an individual who works alone, but they may be part of a smaller team. Their goal is usually not targeted, and may affect anyone.

Severity explanation

They lack capacity and capability because they have little or no knowledge and often limited resources, and they frequently do not completely comprehend what the tool or script they use does to the counterpart. Therefore it has been evaluated as low. The frequency is also low which is reflected on their capabilities as they lack an understanding of detecting vulnerabilities in a system as they rely on leveraging others knowledge.

Impact

For Mobai this means that they can easily prevent an attack like this by keeping their software and machines up to date. If an attack was to be successful, they would probably be exposed to loss of data. Fines for violating privacy rules would occur, leading in a loss of reputation and economic deterioration.

Threat Actor

Negligent internal actor

Motivation:

No motivation as any scenarios spawns from ignorance and carelessness

Intention:

Careless actor with no genuine intention who may be ignorant or makes an error.

Attack method

Human error, lack of cybersecurity hygiene

Capacity

Low

Capability

Medium

Frequency

Low

Description

A careless internal actor makes a mistake that could lead to information disclosure, which might potentially pose a security risk.

Severity explanation

The participants on this thesis are in the category of negligent internal actors since they were responsible for writing the source code. They have limited experience creating complex systems, resulting in a plausibility for mistakes, therefore, capability is set to medium. An example of a mistake can be introducing bugs which can have a severe negative impact on the system. The capacity is set to low due to the team consisting of only three people which enables them to have finer control over the development process. The frequency is set to low since it is unlikely that something harmful would occur multiple times which could have a detrimental impact on the system. Negligent internal actors has a low severity level because they do not pose a strong enough threat with their current intentions and motivations in combination with the frequency.

Impact

The human factor in software development is the key to understanding how errors in software systems occur. A type of negligent act is an erroneous pattern called ”post-completion error”[60]. A post-completion error occurs when a participant does not do a sub-task that is expected to be done at the end of a task, however, it is not a prerequisite for completing the primary sub-task[61]. An example of this is not implementing less important security measures for the main task, since it is not a requirement for delivering the described features. This can often happen when the developer is in a rush to complete the task they have been assigned. An experimental study on software engineers during a programming contest revealed that 41.82% of the participants made the same post-completion error where they forgot to implement the requirements which was not necessary for the main task[62]. As a result, bugs in the system may appear at a later time since the specific sub-task was not implemented as intended. These bugs can range from tiny flaws with no effects to major bugs that might cause information breaches or provide a route into the system for a possible malevolent actor.

Threat Actor
Internal actors
Motivation:
Financial, revenge, save your own skin from blackmail
Intention:
Stealing industry secrets and gaining a competitive advantage in the market, damage reputation, promote competitors.
Attack method
Phishing, social manipulation, exploiting human error, hacking, hardware theft, data leakage, blackmail
Capacity
Medium
Capability
Medium
Frequency
Low
Description
An internal actor is a malicious insider threat that has insight in what information is stored and may change, delete, or share information. The actors can be a current or a former employee, contractor, or business partner that either has or had access to the system. Malicious internal actors are known for intentionally misusing their access in a manner that negatively affected the confidentiality, integrity, or availability of the organization's information or information systems[63].

Severity explanation
Based on internal knowledge of the source code and open access to the systems, they have been assessed with medium capacity. The capacity can vary based on their access level in the organization, therefore, it is set to medium since it is the average of the possible levels. The capability is evaluated to medium because most of the employees have competence in information technology, but as mentioned, this can vary based on their expertise. The frequency is set to low because it is unlikely to happen due to the size of the company, and employees hold shares.
Impact
Internal actors are frequently underestimated as a threat, which can be harmful since they often cause incidents that are more expensive and difficult to recover from than external ones[64]. One of the most serious threats are actors attempting to obtain personal identities in order to conduct fraud and other crimes[65]. Be-

cause 4.95 billion individuals worldwide utilize the internet as of January 2022, potentially everyone on the internet may be a victim[66]. For Mobai, this means that if their mitigation efforts are inadequate, they will constitute an indirect threat to their users. If an inside actor gained access to the users' identities, they could sell them, causing Mobai to lose its customers' confidence and suffer significant financial damages as a result of GDPR violations.

Threat Actor
External opportunists
Motivation:
Financial, amusement.
Intention:
Quick financial gains, confirm knowledge, .
Attack method
Computer-based methods, password leakage, DoS, DDoS, malware, hacking, phishing, social manipulation, data leakage, blackmail
Capacity
Low
Capability
Medium
Frequency
Low
Description
External actor often with a knowledge in information technology, who takes advantage of a certain event or vulnerability in order to harm the victim. Opportunists do not consider the consequences for others, but rather take advantage of the circumstance and act mainly for self-interest.

Severity explanation
External opportunists has capacity set to low, because it is likely that they are operating on their own trying to benefit them self on trying to damage the system, or getting secret information that could be extracted from Mobai. Capability is set at medium since they are expected to have some understanding of the topic and comprehend what they are doing while looking for vulnerabilities to exploit. The frequency is set to low since it is unlikely that an opportunist will attack multiple times without a great opportunity. They are therefore evaluated as medium severity.

Impact
An external opportunist would use several types of attack methods depending on their set of skills. This could be attacks in the form of infecting internal systems with malware, or using phishing, social manipulation, and other methods that could harm Mobai´s reputation through violation of privacy regulations. It is therefore important that there are certain mitigation strategies in place to ensure that these opportunists does not get the chance to exploit any weaknesses.

Threat Actor
Organized Cybercrime
Motivation:
Financial
Intention:
Financial gains
Attack method
Computer-based methods, DoS, DDoS, Phising, social manipulation, data leakage, malware, hacking
Capacity
Medium
Capability
Medium
Frequency
Medium
Description
Organized cybercrime groups are organizations of hackers that utilize their diverse set of skills to commit crimes against victims, frequently in order to profit from the vulnerabilities they discover along the process. A group typically has a top-down hierarchy with a leader who has considerable authority over the individuals under them. The leader will issue commands and direct the members[67]. Because organized cybercrime frequently consists of smaller organizations without the resources and frequently sufficient hardware to execute attacks, it has been assigned a capacity rating of medium. Capability has also been set at medium because cybercrime groups are more likely to have decent expertise in information technology, however, they may lack in-depth knowledge required to perform complex attacks. The frequency is also set to medium because the criminals are likely to undertake this as a full-time profession, giving them the time and effort to perform cyberattacks consistently. The lack of capacity and capability reduces the frequency of assaults and necessitates the group to be organized and structured.

Impact
Because organized cybercrime groups have a solid set of capabilities for exploiting vulnerabilities, Mobai must have robust mitigation mechanisms in place. Attack techniques will frequently take the shape of sophisticated computer-based approaches, such as infecting machines with malware using advanced tools and rootkits. It is also critical that they do not obtain an edge over Mobai, since this

could result in loss of data. A breach would likely result in a violation of the GDPR and privacy regulations, resulting in financial losses and reduced trust from their customer base.

Threat Actor
Competitors
Motivation:
Financial
Intention:
Gain an competitive advantage by stealing intellectual property or damage reputation
Attack method
Computer-based methods, DDoS, phising, social manipulation, data leakage, malware, hacking, steal biometric technology, gain access to data
Capacity
Medium
Capability
Medium
Frequency
Medium
Description
Competitors are other companies that operate in the biometric domain, and may try to achieve a competitive advantage or weaken their reputation by exploiting weaknesses and vulnerabilities. Their goal is to obtain a larger market share, and as a consequence it may affect other businesses.

Severity explanation
Because there are other companies in the same industry as Mobai, this is an actor who might represent a risk. Competitors have therefore been evaluated with medium severity since they may have the purpose and incentive to destroy Mobai's reputation to gain an advantage in the market. They have been assigned a medium capacity since they typically have resources such as hardware and software, as well as the finance to recruit highly educated individuals who are professionals in their area; as a result, their capability is also set to medium.

Impact
A rival searching for a means to tarnish Mobai's reputation will look for particular vulnerabilities that they may use against Mobai to induce insecurity among their userbase causing them to switch to the competitor as their biometric service provider. Competitors that can attack Mobai are unlikely, but because the industry is small, it is possible that some other firms may attempt to out compete Mobai in order to obtain their consumer base.

A real life example where a competitor tried to exploit a demo version of a biometric system to find vulnerabilities, was used to find weaknesses and strengthen their own product. This resulted in a lawsuit against the competitor for patent infringement, which is something Mobai has to be aware of since both parties of the lawsuit will have a negative impact for their reputation and possible financial costs[68].

Threat Actor
APT - Advanced persistent threat
Motivation:
Political, financial, industrial, access, military, intelligence
Intention:
Steal biometric technology, gain access to data,
Attack method
Phishing, social manipulation, malware, hacking, theft of machines, password leakage, data leakage, rootkits
Capacity
High
Capability
High
Frequency
Low
Description
An actor that usually is within a nation state, or state-authorized group that uses advanced methods to gain unauthorized access to either do malicious damage or promote political agendas. Sophisticated tools, malware and rootkits will often be used, but also methods as back doors to escalate privileges on a system to gain access to the network and admin rights which can have a destructive impact on the target.

Severity explanation
APT is an actor assessed with the highest severity rating, since they have both high capacity and capability to carry out a longer and highly advanced attack on the target, which also result in the possibility of more frequent attacks. The reasoning is because they are often found within a nation state or a state-authorized group that is sponsored by the treasury[69]. Personnel in APT organizations are usually highly educated people with a normal 08-16 job, where as attacks rarely occur during the weekends. Generally attacks are also difficult to detect because they normally resemble regular user activity in order to avoid being detected by defensive mitigation mechanisms. Their goal is typically to steal data and/or disrupt the services given to the user, but unlike other actors they also want to stay in the targeted network for as long as possible to gather information.

Impact
It is common for an APT to have a clear intention backed up by a motivator

that will increase their incentive to reach their goal. For Mobai, a company that processes biometric data, this means that they could be a victim of biometric data theft or leakage, or a target for gathering biometric information on the users for the longer term. The frequency of the attacks will differ based on the customers of Mobai, for example, if an important governmental institute were to use Mobai´s system, the interest from an APT will increase accordingly. Fines for violating a user's privacy in this degree can often mean the end of a start-up firm, thus it is critical that Mobai has adequate mitigation strategies in place.

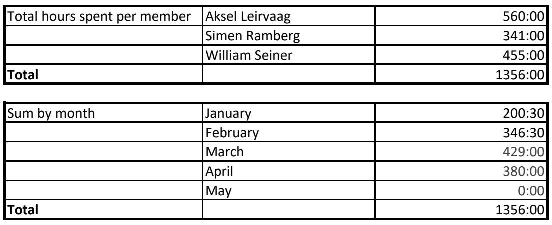
E Asset classification description

Asset	C	I	A
Session Cookies	The confidentiality is evaluated as 4 since it controls the user's session control. As a result, the actor will not require verification and will be able to obtain the identity of the user. You can access all other tokens if you get a session cookie and the consumer has pressed ”remember me” on both login and consent. There is also no central cookie storage. A cookie misconfiguration could result in a leak.	The integrity is evaluated as 4 since cookie issues might result in a poor user experience since the prototype will not not perform properly. In the worst-case situation, the system allows unauthorized requests access to the data.	The availability is evaluated as 2 since if the session cookies are not accessible, a user will have to log in and approve the grant each time. The ”remember me” button will not function properly.
Refresh Tokens	This confidentiality is evaluated as 4 since the refresh token allows you to get as many access tokens as you wish. The access token also has an expiration date.	The integrity is evaluated a s 4 because it regulates access to sensitive data	The availability is evaluated as 4 because the entire system is dependent on tokens to function, the entire system will be rendered inoperable during the time this is unavailable.
ID Token	This confidentiality is evaluated as 4 since the ID Token contains personally sensitive information.	The integrity is evaluated as 4 because it regulates access to sensitive data	The availability is evaluated as 4 because the entire system is dependent on tokens to function, the entire system will be rendered inoperable during the time this is unavailable.
Access Token	This confidentiality is evaluated as 3 because the access token is opaque, meaning that OAuth 2.0 Access Tokens represent internal state but are publicly known. It's also because JSON Web Tokens can't contain secrets, which means that personal information isn't available as clear text to those that shouldn't have access to it.	The integrity is evaluated as 4 because it regulates access to sensitive data.	The availability is evaluated as 4 because the entire system is dependent on tokens to function, the entire system will be rendered inoperable during the time this is unavailable.
Template Database	This confidentiality is evaluated as 4 because it stores biometrically sensitive data. A rumor about something that has been leaked will have disastrous implications.	The integrity is evaluated as 4 because it is crucial that the information is correct and not been tampered with.	The availability is evaluated as 4 because if this information is not available the system is deemed unaccessible for any users.
Template Protector	This confidentiality is evaluated as 3 because the key used to ensure unlinkability, which means that the relationship between two templates will not be linked. In the template protection algorithm, various keys will provide different results. If an actor gets a protected template in database A, it will be unable to determine whether or not users exist in database B.	The integrity is evaluated as 4 because errors in data (biometric template and comparison) can be catastrophic since they allow actors to gain unauthorized access to users data.	The availability is evaluated as 4 because downtime render the service inaccessible to users.

PII Database	This confidentiality is evaluated as 4 because the database contains personal identifiable information that is highly sensitive and strictly confidential.	The integrity is evaluated as 4 because other components of the system rely on the information being correct and not tampered with.	The availability is evaluated as 4 because the system will be rendered useless without this information.
Webcam	This confidentiality is evaluated as 4 because an actor with access to the webcam can inspect the input and take pictures of the user through their webcam which may result in unauthorized access.	The integrity is evaluated as 2 because faults in the camera and the data it generates can cause the system to fail because the image is inaccurate. This will prevent the user from gaining access to the system, and will only be with the user.	The availability is evaluated as 2 because the issue will only persist with the user, and can be easily resolved with software upgrades, or by purchasing a new camera.
ORY Hydra	This confidentiality is evaluated as 4 because if an unauthorized user has access to Hydra the entire system will compromised and personal information leaked.	The integrity is evaluated as 4 because ORY Hydra handles tokens, which can cause unauthorized users to gain access to secret information.	The availability is evaluated as 4 because ORY Hydra is a main component of the prototype, and without it the system will not function.
Face Images	This confidentiality is evaluated 4 because it is sensitive biometric data that can be used to identify the users of the system.	The integrity is evaluated as 4 because any faults in the picture will prevent the user from gaining access. As a result, accurate information is required.	The availability is evaluated as 2 because this will only effect the person who is attempting to acquire access, which is an issue on their end.
Hydra Companion	This confidentiality is evaluated as 3 because all communication / data passes through the companion, unauthorized access can be very harmful, but no configuration changes can be as devastating as with Hydra. Code completion will be required for fatal modifications.	The integrity is evaluated as 3 since it controls communication between all parties. As a result, some persons may be denied access.	The availability is evaluated as 4 because the Hydra Companion is an essential part of the system, without it the system is rendered useless.
Biometric Engine	This confidentiality is evaluated as 3 because it processes all biometric data, however everything is done in memory. Without doing some sort of memory dump, no one has access to or insight into what kind of data is traveling there. It doesn't generate any data.	The integrity is evaluated as 4 because it processes biometric data, and it is therefore critical that the integrity of that processing is correct.	The availability is evaluated as 4 because the Biometric Engine is an essential part of the system, without it the system is rendered useless.
OIDC Credentials	This confidentiality is evaluated as 3 because if someone were to gather the necessary information, an actor may pose as a third party. It will provide them access to information from people who utilize this third party.	The integrity is evaluated as 3 because invalid credentials will prevent a third party from using Hydra.	The availability is evaluated as 3 because incorrect credentials will render the system useless to third parties. This may affect the partnership between Mobai and the affected third parties.
Personal Identifiable Information	This confidentiality is evaluated as 4 because it contains sensitive personal information that should only be seen by those with access privileges.	The integrity is evaluated as 4 because other systems rely on the accuracy of this data.	The availability is evaluated as 4 because the Personal Identifiable Information is an essential part of the system, without it the system is rendered useless.
Biometric Template	This confidentiality is evaluated as 1 because outside of the system, this should be meaningless because the template is protected by the template protector which implies the usage of being irreversible and unlinkability.	The integrity is evaluated as 2 because it is necessary for a user to gain access to the system. As a consequence, a user has to reenroll to produce a new template.	The availability is evaluated as 4 because the Biometric Template is an essential part of the system, without it the system is rendered useless.

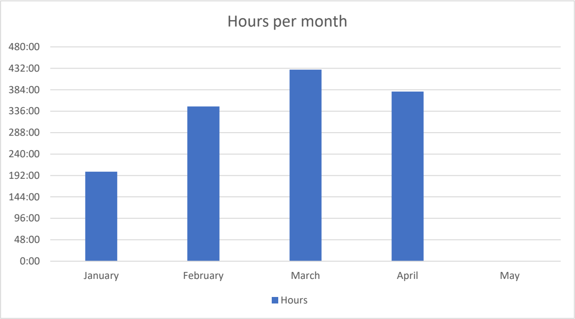
Table 9: Asset description

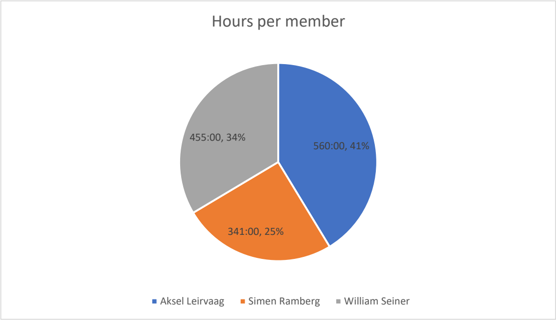
F Summary of timenotes



Total hours spent per member	Aksel Leirvaag	560:00
	Simen Ramberg	341:00
	William Seiner	455:00
Total		1356:00

Sum by month	January	200:30
	February	346:30
	March	429:00
	April	380:00
	May	0:00
Total		1356:00





NTNU

Norwegian University of
Science and Technology