NTNU
Norwegian University of
Science and Technology

Personalized Learning in Radiology

An Adaptive Learning Strategy for Teaching
Chest X-ray Interpretation

Sindre Osmundsen Rasmussen

Master of Science in Computer Science
Submission date: June 2018
Supervisor: Trond Aalberg, IDI

Norwegian University of Science and Technology
Department of Computer Science

Abstract

An adaptive learning system is a software system used in education that analyzes a student's interactions in the system and adapts and presents learning material based on this analysis. The student gets a personalized learning experience that is tailored to focus on knowledge the student is lacking, and the knowledge is presented in a way that the specific student best understands. There are many commercial adaptive learning systems available, but most of them focus on the most common areas in education. These are areas like mathematics, language, physics, chemistry, biology or history. There are still some more specialized areas where there are little or no available adaptive learning systems.

This master's thesis is investigating how adaptive learning can be used to create a learning system in radiology. More specifically a learning system that teaches students how to interpret chest X-ray images, so that students learn how to detect serious and possibly life threatening conditions. There are many different adaptive learning techniques, and a key question for this thesis is to identify which of these techniques that are applicable in the teaching of chest X-ray interpretation. Chest X-ray interpretation can in general be classified as a form of image analysis. Techniques that work here might therefore also be applicable in teaching other forms of image analysis.

In order to identify the appropriate adaptive learning techniques for chest X-ray interpretation, a prototype system was developed. The development was done in cooperation with experts in radiology, who gave feedback during the development, and contributed by creating the learning material the system used as base material when it adapted to students. Three different adaptive methods were developed and implemented in the prototype system: adaptive case selection, adaptive follow-ups and adaptive task type. After the development the methods had to be tested to see if they managed to adapt to the students. The testing was done on students of medicine through two user testing sessions, where the students were observed as they tested the system. The students were also asked to answer a series of questions in a survey, and the system automatically collected user data while the students tested the system. Analysis of these data showed that adaptive content techniques seem to be the preferred category of adaptive learning techniques when developing an adaptive learning system for chest X-ray interpretation. Also task design should have a problem-solving focus, where educational instructions are presented, followed by a problem where the teaching from the instructions can be applied.

Sammendrag

Et adaptivt læringssystem er et programvaresystem for undervisning som analyserer en students interaksjon med systemet og tilpasser læringsmateriale, og presentasjonen av dette, basert på denne analysen. Studenten får en personalisert læringsopplevelse som er skreddersydd slik at det fokuseres på kunnskap studenten mangler, og kunnskapen blir presentert på en slik måte at den er lettest mulig å forstå for den spesifikke studenten. Det finnes mange kommersielle adaptive læringssystemer, men de fleste av disse fokuserer først og fremst på de mest typiske områdene innenfor undervisning. Dette er områder som matematikk, språk, fysikk, kjemi, biologi og historie. Det er fortsatt mange spesialiserte undervisningsområder hvor det er få eller ingen adaptive læringssystemer å finne.

Denne masteroppgaven undersøker hvordan adaptiv læring kan benyttes for å utvikle et læringssystem innenfor radiologi. Mer spesifikt, et læringssystem som skal lære studenter hvordan man tolker røntgenbilder av brystet, slik at studenter lærer hvordan de oppdager alvorlige og potensielt livstruende tilstander. Det finnes mange forskjellige teknikker for adaptiv læring. Et sentralt spørsmål for denne masteroppgaven blir å finne ut hvilke av disse eksisterende teknikkene som kan anvendes for å undervise tolkning av røntgenbilder (bildediagnostikk). Bildediagnostikk kan generelt bli klassifisert som en form for bildeanalyse. Teknikker som fungerer for bildediagnostikk vil muligens derfor også være overførebare til undervisning av andre former for bildeanalyse.

Det ble utviklet et prototype-system for å komme fram til den eller de adaptive læringsteknikkene som egner seg best for undervisning av bildediagnostikk. Utviklingen av systemet ble gjort i samarbeid med spesialister innenfor radiologi. Disse gav verdifulle tilbakemeldinger underveis, samt bidro med å lage læringsmaterialet som systemet baserte seg på da det tilpasset seg til studentene. Tre forskjellige adaptive metoder ble utviklet: adaptive case selection, adaptive follow-ups og adaptive task type. Etter at utviklingen av systemet var ferdig ble disse metodene testet for å se at de klarte å tilpasse læringsopplevelsen til studentene. Testingen ble gjort på medisinstudenter gjennom to brukertestingssesjoner. Underveis i brukertestingen ble studentene observert. Etterpå ble de bedt om å svare på en rekke spørsmål gjennom en brukerundersøkelse. I tillegg samlet systemet automatisk inn brukerdata underveis mens studentene testet systemet. Analyse av de innsamlede dataene viste at adaptive content teknikker ser ut til å være den foretrukne formen for adaptiv læring når et adaptivt læringssystem skal utvikles. Når det gelder oppgavedesign så bør oppgaver ha et såkalt problem-solving fokus, hvor undervisende instruksjoner presenteres, etterfulgt av et problem hvor man anvender det man lærte.

Preface

This is a master's thesis in the field of software engineering, as part of my degree in Computer Science at the Norwegian University of Science and Technology (NTNU). My specialization is in Software Engineering, at the Department of Computer Science (IDI), in the faculty of Information Technology and Electrical Engineering (IE).

The thesis is a continuation of the work from the Computer Science Specialization Project (TDT4501) at NTNU. The most relevant findings from this specialization project are included in chapter 2 in this thesis, and form the theoretical background for the master's thesis project.

I would like to thank my supervisor, associate professor Trond Aalberg for valuable guidance and feedback throughout the process. I also would like to give a special thanks to associate professor Børge Lillebo at the department of Circulation and Medical Imaging at NTNU for giving valuable insight into radiology and medical imaging, creating and making available material necessary for the project and helping to recruit medical students for user testing sessions.

Trondheim, June 11th, 2018
Sindre Osmundsen Rasmussen

Table of Contents

List of Tables

List of Figures

Abbreviations

REST	=	REpresentational State Transfer
API	=	Application Programming Interface
GUI	=	Graphical User Interface
ICT	=	Information- and Communication Technology
CPM	=	Category Performance Measure
UDM	=	User Difficulty Measure
TTPM	=	Task Type Performance Measure.

Chapter 1

Introduction

1.1 Motivation

Education is vital to our society, and needs to be diverse in order to be effective for everyone, since people learn in different ways. The importance of adapting teaching methods to each individual is therefore instrumental for creating a good education system. As described by Bloom [1], one-to-one interaction between a teacher and the learner is the most effective way to adapt teaching to create a personalized learning experience. But a teacher is only one person, and in some cases the teacher is faced with a situation where there are too many students and not enough time to adapt the teaching to each one of them. The teacher must instead lecture in a general way that most students, but not all, will find understandable, as Krokan [2] describes more in detail. But what about those individuals that do not find the general lecture educational? There is a risk that these individuals do not get to realize their potential, since the education system is not using the teaching approaches that they need in order to learn.

Adaptive learning technology is a possible solution to this problem. An adaptive learning system is a system that utilizes ICT in order to recreate the personalized learning experience a person would need and should get from one-to-one interaction with a teacher. There are many adaptive learning systems out there, for instance ALEKS[1], Knewton[2] or SmartSparrow[3]. They all have unique approaches to how the teaching can adapt to the student, but they all focus on typical domains like mathematics, physics, history or geography. There are still many domains where there are few or none adaptive learning systems that have been developed. One of these domains is radiology, where students among many things learn how to interpret X-ray images in order to discover potential life threatening conditions.

[1] https://www.aleks.com/ Accessed: 24.05.2018

[2] https://www.knewton.com/ Accessed: 24.05.2018

[3] https://www.smartsparrow.com/ Accessed: 24.05.2018

1.2 Project Goal

The main goal for this project is to test how adaptive learning techniques and methods can be applied in the teaching of chest X-ray interpretation, through an adaptive learning system. To manage that, new unique adaptive learning methods, tailored for teaching chest X-ray interpretation, were developed. These methods had a basis in, and were inspired by, existing general adaptive learning techniques. The developed techniques were then implemented in a prototype for an adaptive learning system. The system contained image- and text-based task material created by radiologists and experts in the domain of radiology, where the main focus was on so-called thorax (chest) X-ray images. The final result was a prototype system that applied adaptive learning techniques specifically designed and adjusted for teaching chest X-ray interpretation to students of medicine. This system was then tested on medical students, in order to determine if the applied adaptive learning techniques managed to create the intended personalized learning experience. The testing consisted of user testing sessions where the students were observed, a survey where students were asked questions about the prototype system, and automatic collection of data about how the students used the system.

1.3 Research Questions

- **How can adaptive learning techniques be applied in order to teach students how to interpret chest X-ray images? (RQ1)**
  As will be described later, there exist some general techniques and methods for creating adaptive learning systems. These techniques have been applied and tested in numerous commercial and non-commercial learning systems, but none of these systems focus on radiology. There exists very few or none examples of how these general adaptive learning techniques can be applied in chest X-ray interpretation. Therefore this thesis will suggest an approach to how this can be done, and afterwards test the approach. An important goal here is to identify which techniques are suitable for teaching chest X-ray interpretation, and which techniques that are not suitable. The techniques that are identified as suitable here might also have a more general applicability in other similar forms of image analysis outside chest X-ray interpretation.

- **How can knowledge from learning theories and learning technology be applied in the design and presentation of task material for chest X-ray interpretation? (RQ2)**
  In the design of a learning system in general, understanding learning theories and learning technology is crucial. What is important here is to understand the material that is to be taught. It is important to make sure that the system in fact promotes learning. This will be possible through combining knowledge from general learning theories with insight into the learning material and how the teaching of chest X-ray interpretation is usually done. This thesis will based on acquired insight into radiology and chest X-ray interpretation suggest and test different task designs, in order to identify learning theories and approaches that are suitable for this domain.

- **How can the student's knowledge/abilities in chest X-ray interpretation be measured and modelled? (RQ3)**
  In order for an adaptive learning system to adapt, it should manage to understand the needs of the student. It is therefore important to get an overview and understanding of what skills that are required from a student, and how these skills can be measured and used to trigger adaptive responses like task recommendations or explanations. Based on insight into radiology and chest X-ray interpretation a suggestion for such a model will be tested in the system that is to be developed, in order to answer this question.

1.4 Thesis Outline

Prestudy (chapter 2): The prestudy presents vital background theory about adaptive learning technology, learning theory and radiology, that the project is built upon.

Design (chapter 3): The design chapter describes the developed idea for an adaptive learning strategy for chest X-ray interpretation, how this strategy was designed and how the prototype system for testing the strategy was designed.

Implementation (chapter 4): The implementation chapter describes how the designed adaptive learning strategy was implemented in the prototype system.

Evaulating the System through User Testing (chapter 5): The evaluation chapter describes the testing method that was applied in the testing of the adaptive learning system that had been implemented. This to determine if the chosen adaptive learning strategy was a viable strategy for teaching chest X-ray interpretation.

Results and Discussion (chapter 6): The result chapter presents the results of the test presented in chapter 5, and afterwards discusses what the results are showing.

Conclusion (chapter 7): The conclusion chapter summarizes the entire project and presents the findings. Limitations and future work for the project are also described.

Chapter 2

Prestudy

Before developing an adaptive learning system it is crucial to have sufficient background knowledge about adaptive learning, learning theories and insight into the domain that the system is going to teach the student.

2.1 Adaptive Learning

2.1.1 Defining Adaptive Learning Technology

The exact definition of adaptive learning technology could be formulated in many different ways depending on what focus and learning scenario the definer choose to consider. Even so, a good definition of adaptive learning technology is the definition formulated by the EdSurge-team [3] in their study of adaptive learning systems:

”Education technologies that can respond to a student’s interactions in real-time by automatically providing the student with individual support”

To elaborate this short, but precise, definition; one can say that adaptive learning technology is a type of educational methods that utilize ICT in order to adapt learning aids and tasks, so that they are best suited for a specific individual’s needs when it comes to learning. An example could be that the system analyzes how the student solves tasks in different topics, in order to determine the skill level for the student in that specific topic. Based on detected skill level the system can then adapt and suggest tasks that are suitable for the skill level of that specific student. The system could also analyze what type of mistakes the student makes, and choose appropriate hints that could help the student understand why he/she made exactly that kind of mistake.

2.1.2 Categorizing Different Methods for Adaptive Learning

Based on searches done in relevant literature on the topic of adaptive learning, there seems to be no universally agreed terms or definitions on how adaptive learning systems can be categorized. Both Paramythis & Loidl-Reisinger [4] and the organization Edsurge [3] suggest possible ways to categorize adaptive learning systems, and their categories have many commonalities. The term ”Intelligent Tutoring Systems” is also frequently used to refer to learning systems that have some sort of adaptive learning capability, but it is not a category within adaptive learning systems.

The organization called EdSurge[1] has come up with some category definitions [3] that were found to be very relevant and useful for this project. EdSurge is a commercial and independent organization focusing on informing teachers, schools and other relevant stakeholders about the latest research and news concerning the use of technology in education. Their three categories is a result of their own studies of many of the latest commercial adaptive learning systems. The categories are ”adaptive content”, ”adaptive assessment” and ”adaptive sequence”. The categories differ from each other in what approach they have to adaptive learning. An understanding of these three categories is useful in order to analyze an adaptive learning system, and can help to see similarities between different learning systems, and also makes it easier to understand how a specific learning system can fit a specific learning scenario.

Adaptive content

The adaptive content approach is an approach to adaptive learning where the system makes specific adaptations to the content it delivers to the student, based on what mistakes the student did when solving the tasks in the content already presented to the student by the system.
There are typically three main ways to use the adaptive content method:

- Feedback and hints: The system has a pre-implemented set of different kinds of hints that can be provided to the user. Each hint has a specific type of student-mistake that triggers it. For example, if the student had the task ”Solve: 5+2*2”, and answered 14 (which is wrong), the hint ”Remember to do multiplication before addition” would be triggered, since this is the mistake that caused the wrong answer. Different kinds of mistakes in the same task would trigger other types of hints.

- Additional learning resources: The system has a set of learning resources available. A learning resource could be a video, an article, a figure or some other representation of knowledge. When the student does a mistake, that specific mistake triggers a specific learning resource that explains and demonstrates knowledge needed in order to avoid making the same mistake again.

- Scaffolding and branching: Branching is a form of adaptive content where all the tasks a student should go through are ordered along a main branch, where every task represents a specific topic. If the student makes a mistake in one of the tasks,

[1] https://www.edsurge.com/ Accessed 07.06.2018

he/she is diverted from the main task-branch and over to a side-branch with additional tasks, which only covers that specific topic that the student failed in on the main branch. The student will remain on the side-branch until he/she shows, through correct answers, that he/she has understood the topic. After that the student is put back on the main branch and continues through the original tasks. Scaffolding is, as Azevedo et al. [5] describes, a category of methods that calibrates and fine-tunes the level of content support given to the student based on the student's current performance. Branching can based on this definition then be seen as a coarse method of scaffolding.

The implementation of any of the adaptive content approaches is a simple matter of defining conditions which trigger specific adaptation-events as a consequence of actions conducted by the student. These events could be feedback, hints, learning resources or task tutorials. The main idea is that the conditions that evaluate to true function as an indirect description of how the content should adapt to react to the student's performance. Figure 2.1 shows an example of how adaptive hints and branching could be implemented.

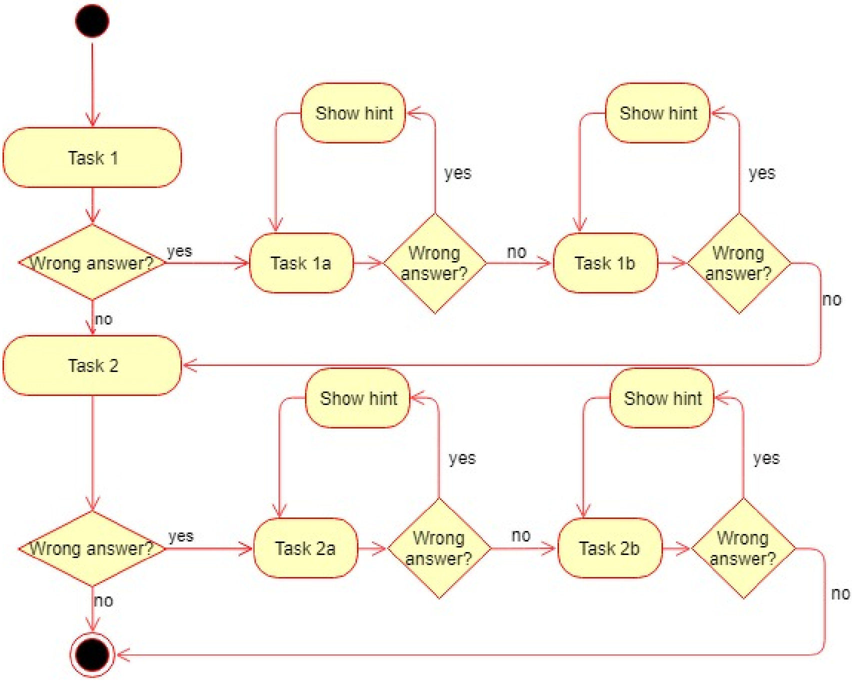


Figure 2.1: Example of general approach to adaptive content. In this example the adaptive hint approach is combined with the branching approach.

There are many different ways one could implement an adaptive content approach. An example is how Grawmeyer et al. [6] implements adaptive feedback and hints into the LIBE VLE intelligent tutoring system. Instead of pre-programming one hint to one specific mistake, the system continuously calculated the so called attainment-level of the student, and adaptively shaped the feedback and hints to match the attainment-level. Another way adaptive content can be implemented is how Thyagharajan and Nayak [7] talks about adapting the visual presentation of the learning material based on the user's preferences.

There has been done some research on the effectiveness of the adaptive content approach. One example is an experiment where a scaffolding/branching technique was used to see if it could help students learn excel [8]. The experiment was conducted on students from 60 different universities, where all the students were divided into two groups; one control group and one group that was exposed to the scaffolding approach. The conclusion of the experiment was that an adaptive e-learning system which used scaffolding could function as a good replacement for personalized teaching given by a teacher to a single student.

Adaptive assessment

The adaptive assessment approach is an approach to adaptive learning where the system adapts the difficulty of the tasks based on how well the student is performing. The system should always manage to give a challenge to the student, but at the same time not make it too difficult, since this could have a negative effect on the student's motivation.
There are typically two main ways to use the adaptive assessment method:

- Practice engine: In this approach the system has a large pool of tasks available. Each task belongs to a specific topic and has a value for how difficult it is considered to be. The system picks tasks one at the time from the pool, and based on the student's performance on a task, the system can make a more informed decision for the next task to choose from the pool. The system will continue to choose new tasks from the task pool until the student has shown a sufficient level of knowledge. A description of the flow for this approach can be seen in figure 2.2.

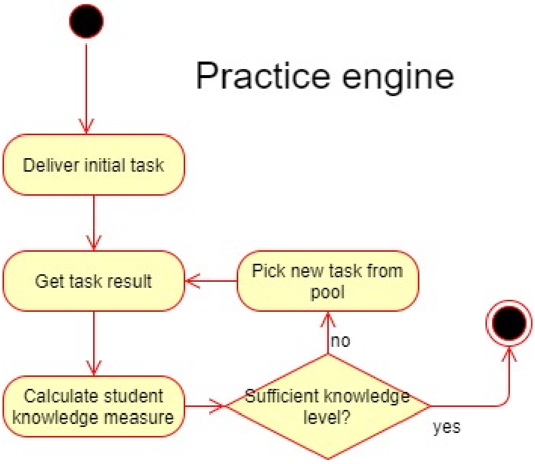


Figure 2.2: A general description of the practice engine approach

- Benchmark assessment: In this approach the student is taken though a loop. First step in the loop is to take an assessment test. In this test the system evaluates the current skill level of the student. Based on the evaluation, a set of suitable educational resources and tasks are generated. The next step for the student is now to go through the educational resources and tasks in order to improve his/her skills. After this the loop starts again, and a new evaluation test is taken. The results of this test will lead to the generation of new educational resources and tasks. The loop continues until the system detects a sufficient skill level in all topics. A good example of the benchmark assessment approach is the pie-chart in the commercial adaptive learning system ALEKS[2]. A general description of the flow for this approach can be seen in figure 2.3.

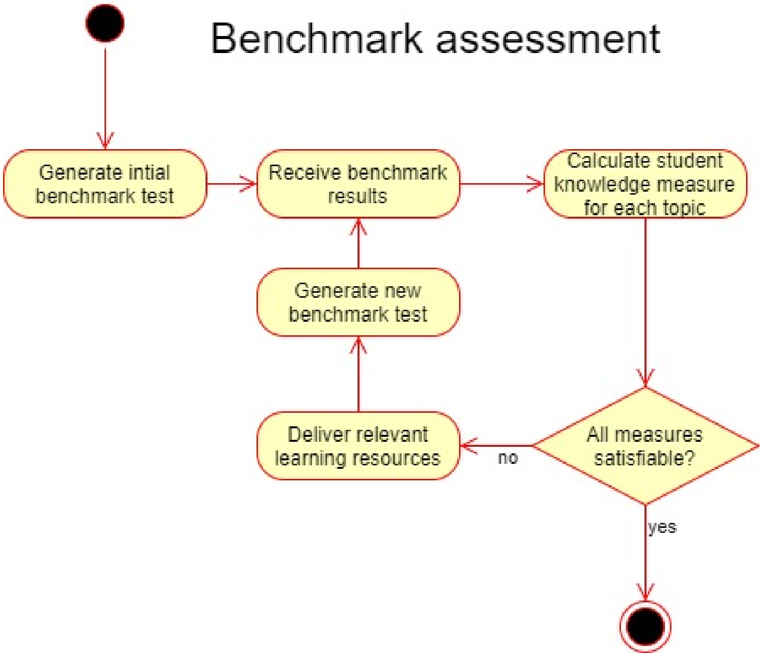

Benchmark assessment
Generate intial benchmark test
Receive benchmark results
Calculate student knowledge measure for each topic
Generate new benchmark test
Deliver relevant learning resources
no
All measures satisfiable?
yes

Figure 2.3: A general description of the benchmark assessment approach

Adaptive assessment is an approach where the measuring or modelling of the knowledge level of the student is crucial for success. Solutions that utilize this approach should therefore have some kind of student model, which contains data for describing the knowledge of the student. In the case of adaptive assessment the student model will only focus on domain dependant data, which means that the model only collects and contains data that is relevant to what knowledge the student has in the domain being learned. Such a model can then be utilized by some sort of adaptive assessment algorithm to modify the difficulty of the tasks given to the student.

[2] https://www.aleks.com/ Accessed: 24.05.2018

There are many examples on how a student's knowledge can be modelled. Many of them are inspired, or can trace their roots back to, the Bayesian Knowledge Tracing algorithm developed by Corbett & Anderson [9]. Another way to create a student model can be through a matrix representation, like the SBTS model [10] is doing. An example of such a knowledge matrix can be seen in figure 2.4. Here the rows represent categories in the domain that is going to be learned, while the columns represent difficulty. Each cell in the matrix thereby represents one possible task the system can give to the student. The column of the cell specifies the task difficulty and the row of the cell specifies the category of the task. By monitoring the student's performance on tasks from the matrix, the system can estimate a suitable position in the matrix and will choose tasks that are placed near this position. This ensures that the system suggests tasks for the student that have a suitable difficulty and a suitable category.

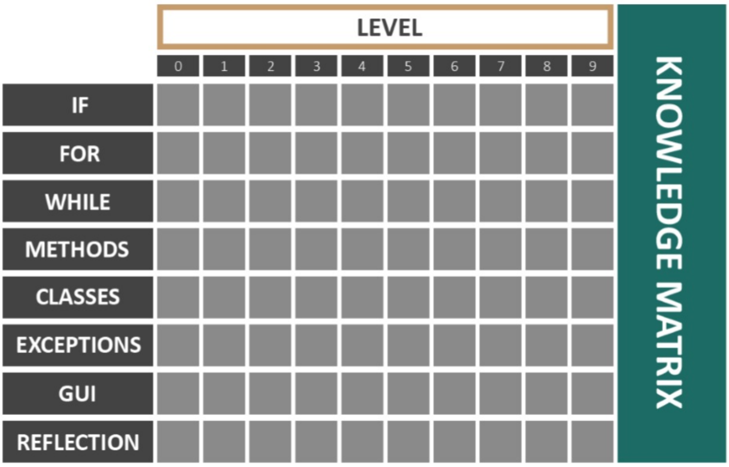


Figure 2.4: The knowledge matrix in the SBTS [10]. This specific matrix represents tasks related to teaching basic programming.

Adaptive sequence

Adaptive sequence is an approach to adaptive learning where an algorithm of some kind collects different data about the student's actions in the system and analyzes it in order to create the most suitable selection and sequence of tasks for the student to go through. The data collected could for example be data about where the student has clicked, answer data or time spent on different pages. Exactly how the algorithm makes decisions and creates the sequence will vary for every learning system, but the implementations of adaptive sequence often consist of the following three steps:

- Collect: Data about the student's behaviour is collected. The data can reflect the student's interests, skill level, motivation, patience or preferred learning strategies and learning resource types.
- Analyze: In the analysis step the algorithm takes all the collected data, and starts

with a learner analysis in order to detect the student's skill level in different topics. Based on the learner analysis the algorithm then does a skill selection, where it decides the most suitable next topic in the sequence. After the topic is chosen, a content analysis is done to find the task in that topic that is most suitable for the student and his/her current skill level.

- Adjust: In this step the actual content of the task chosen in the analysis step is adjusted, so that it is presented and visualized in a way that is most suitable for the way this specific student learns and understands best.

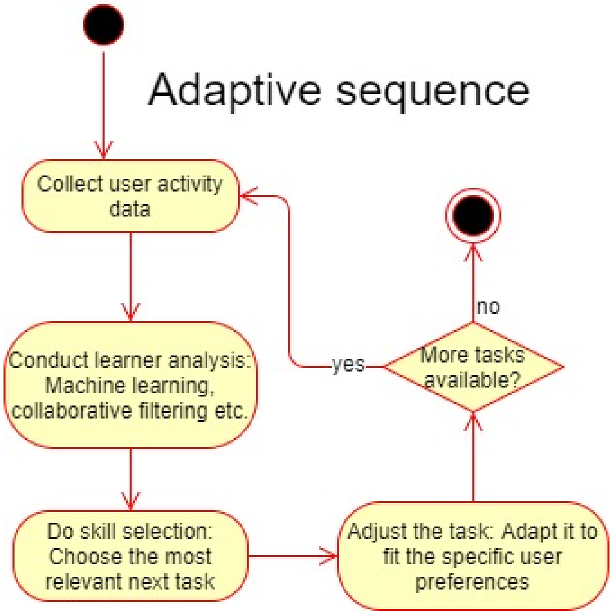

Adaptive sequence
Collect user activity data
Conduct learner analysis: Machine learning, collaborative filtering etc.
Do skill selection: Choose the most relevant next task
Adjust the task: Adapt it to fit the specific user preferences
More tasks available?
yes
no

Figure 2.5: A general description of the adaptive sequence approach

In adaptive sequence the collection of data for analysis is crucial. So just like in adaptive assessment, the concept of student modelling is a relevant concept. In adaptive assessment domain dependant data, which describes what knowledge the student has in the domain being learned, is important. Domain dependant data is just as important in adaptive sequence, but here domain independent data also plays an important role. This is data that describes the student more generally. For example gender, personality or other preferences that are not directly connected to the learning domain. An example of the differentiation between domain dependant data and domain independent data can be seen in the PCMAT-system [11], which is a typical example of an adaptive sequence system.

Other typical approaches to adaptive sequence are:

- Domain modelling: Making a model describing the learning domain. The domain model will function as the system's map of the entire topic or course that it is going to teach the student. A typical example of such a model can be seen in the Protus-system [12]. The knowledge graph used in the adaptive learning system Knewton[3], as can be seen in figure 2.6, is also a clear example of this.

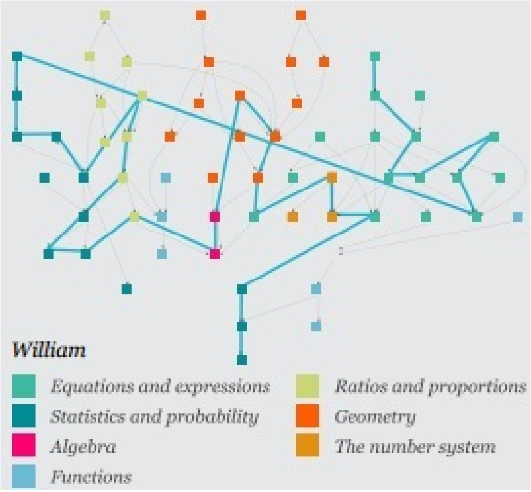


Figure 2.6: Example of how a knowledge graph could look like [13].

- Classification/clustering: Comparing all the students of the system in order to group similar students. The system will then give recommendations to a student based on what is popular in the group the student is assigned to. In classification groups are pre-defined, while in clustering the groups are defined as users are compared to each other. Typical methods are HAC or K-means [14].

- Collaborative filtering [14]: Here a matrix is used to describe all user preferences. A user is compared to all the other users, and recommendations are given to that user based on the preferences of those users with the most similar preferences.

2.2 Learning Theory and Task Design

2.2.1 The Multimedia Effect

A central focus when designing tasks for a learning system should be how images, text and sound should be put together in order to promote learning. The multimedia effect [15] provides good guidelines for how the most educational design should be achieved. Briefly explained, the effect says that text and image should be presented at the same time (temporal contiguity) and integrated in the same view (spatial contiguity). This in order to achieve maximal learning effect from the content that is presented.

[3] https://www.knewton.com/ Accessed: 24.05.2018

2.2.2 Instructional Software and Task Types

Instructional software is an application or program designed specifically to give instructions or assist a student in learning a specific subject. An adaptive learning system falls into this category of software. According to Roblyer, Edwards & Havriluk [16] instructional software can be grouped based on what kind of tasks/instructions it provides to the user. The categories are:

- Drill and practice: Tasks requiring answers, then feedback is given based on the answer.
- Tutorials: Step by step instructions and activities.
- Simulations: A model of a system or a process that can be explored in order to learn.
- Instructional games: The use of game effects to increase user motivation.
- Problem-solving programs: Instructions and activities followed by a problem where the newly instructed knowledge can be applied.

2.2.3 The Power of Comparison in Learning

One tool that can be utilized in order to achieve learning is comparison techniques. The main idea here is that the user gets to compare two cases, solutions, methods etc. and from this comparison manage to learn how and why a solution is as it is, and when a certain method should be applied or not. There are different methods for how comparison can be used in learning. Rittle-Johnson & Star [17] have come up with a useful framework for categorizing different methods of comparison in learning. It divides all comparison methods into five different types: problem comparison, problem category comparison, correct method comparison, incorrect method comparison and concept comparison.

Problem comparison

Problem comparison methods is about comparing two different problems that have been solved using the same method. From the comparison the user should manage to learn when and when not a general method of problem solving can be applied, and also how the general method is applied on different types of problems.

Problem category comparison

Problem category comparison is about comparing two different problems which are solved using different methods. The idea is that by comparing the two solutions, the user can see why one method worked on the first problem and why it was not applicable on the second problem, which needed a different method.

Correct method comparison

Correct method comparison is about comparing two different methods by solving the same problems with both of the methods. The main reason for this comparison of methods is to learn why one method is better than another.

Incorrect method comparison

Incorrect method comparison is about comparing an incorrect method of solving a problem with a correct method of solving the same problem. The reason for this is to learn why the incorrect method did not work, and why the correct method worked instead.

Concept comparison

Concept comparison is about comparing different concepts, and by understanding one concept, this concept can be used to explain a concept one do not understand yet. For example one can learn how to use the greater than sign, by comparing it to the equal sign.

Comparison methods are useful in the context of this project, since a key way to learn how to analyze an X-ray image is to see examples from other X-ray images and compare an image with images that have certain anomalies one would want to rule out or confirm [18].

2.3 Radiology and Medical Imaging

2.3.1 Domain Experts

In order to create a viable adaptive learning system for chest X-ray interpretation, there is a need for domain knowledge. Throughout the development of the adaptive learning system the required insight into radiology was obtained through communication with domain experts. The experts gave advice and feedback throughout the development, and also contributed directly by creating and quality assuring the learning material the system was going to depend on. The experts that contributed with their knowledge were:

- Børge Lillebo, physician with experience in lung medicine (primary contact for advice and feedback).
- Arve Jørgensen, specialist in radiology.
- Andreas Sjøli, specialist in radiology.

These experts will from now on be referred to as ”domain experts” or ”the expert panel”.

2.3.2 What is Medical Imaging and Radiology?

Medical imaging refers to techniques and processes used to create images of various parts of the human body for diagnostic and treatment purposes within digital health[4]. Medical imaging can be seen as the tool used in radiology, which is the science that uses medical imaging to diagnose and treat the body. There are various radiological imaging techniques for different purposes. Among these are:

- X-ray radiography: An image of the internal part of the body is created by transmitting X-rays (ionizing radiation) through a patient's body. Based on what the rays hit

[4] https://innovatemedtec.com/digital-health/medical-imaging Accessed: 21.02.2018

they will either be absorbed or pass through the body. Those that pass through will hit a film and an image is formed on the film.

- Ultrasound: Uses high frequency sound waves instead of ionizing radiation to form a real time image of the inside of the body, but the image quality is not as good as X-ray radiography.
- Computed Tomography (CT): X-rays are used in conjunction with computing algorithms to form an image of the entire body that can be viewed in slices.
- Magnetic Resonance Imaging (MRI): Applies strong magnetic fields to align protons within the body tissue and then observes the radio frequencies of signals generated as they return to their baseline state. These frequencies can then be used to construct an image of the body tissue.

The focus in this report will be on images that are generated using plain X-ray radiography with focus on so-called thorax-images (chest-images). All conclusions in this report should therefore not be interpreted as general for the entire domain of radiology, but primarily only for the task of analyzing X-ray images, even though ”radiology” might be used as a term to refer to the domain that the adaptive learning system is teaching.

2.3.3 Teaching Radiology - The ABCDE Approach

It was strongly recommended by the domain experts that the system in some way should be based upon the ABCDE-principle. The ABCDE-principle is a well established principle in medicine, and is taught to all students of medicine. It is applicable in many different disciplines of medicine. For each discipline there might be some adjustments to the general principle, like for instance radiology [19]. The ABCDE-principle can be described as a rule of thumb stating in which order different examinations of a patient should be done. The examinations that discover the most life threatening conditions should be done first. So by remembering what A, B, C, D and E stands for, a student knows what order to do things in. That is: check Airways first, then Breathing, then Circulation, then Disability and lastly Environment and exposure.

The general ABCDE-principle

Each of the steps in the ABCDE-approach [20] can be summed up like this:

Airways: Check for any kind of obstruction of the airways. If airways are obstructed these should be freed with the proper technique or equipment in order to avoid cardiac arrest.

Breathing: If the airways are clear it should be checked that the breathing is sufficient, and examinations should be done focusing on identifying common causes for insufficient breathing.

Circulation: The next step is to check for circulatory problems by looking for color changes, sweating, pulse, and checking blood pressure.

Disability: The next step is to determine the patient's level of consciousness to get a sense of the patient's condition.

Environment and exposure: After the level of consciousness and condition is determined, the last step is to identify what the cause or source of the patient's condition is. For example if the patient is in pain, then one should determine what the reason for this pain is.

Customizing the ABCDE-principle for use in chest X-ray interpretation

By reading the general ABCDE-principle one can see that this is firstly intended for a typical emergency situation where one might not have all the background information, and where the availability of an X-ray image is not necessarily the case. Even so, the importance of the order described by the ABCDE-principle still applies when analyzing an X-ray image. There is just a need to adapt the principle to the situation, like Crausman [21] for instance is doing. Therefore the domain experts created the learning material for the system based on a version of the ABCDE-principle that was modified for the task of analyzing an X-ray image. The result of this was 13 different categories (excluding the category ”normal”), which all could be linked to either airways, breathing, circulation, disability or environment/exposure. The categories can be seen in table 2.1.

Table 2.1: ABCDE-principle and categories for chest X-ray interpretation

Airways	- Luftveier er forskjøvet (trachea, carina)? - Okkluderte hovedbronkier?
Breathing	- Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)? - Pleuravæske (ensidig, bilateral)? - Hyperinflasjon? - Pneumothorax?
Circulation	- Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)? - Lungehilii forstørret (bilateralt eller ensidig)? - Hjerte forstørret?
Disability and Diafragma	-Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)? -Diagframa uskarpt avgrenset eller avflatet?
Exposure, Extrathoratic or anything Else	-Fri luft under diafragma? -Hiatus hernie?

Chapter 3

Design

As mentioned in section 1.2, the main goal for this project is to create a learning system that applies a unique approach to adaptive learning, which is specifically tailored for teaching students to analyze chest X-ray images. A crucial part of the work in achieving this goal was a design phase where the unique needs and characteristics of the domain were considered, as the different aspects of the system were defined and concretized. In this phase general knowledge about adaptive learning and learning theory was applied in order to come up with new approaches to how tasks, user interface and system flow could be designed, in order to create a good personalized learning experience that takes the unique characteristics of chest X-ray interpretation into consideration.

The adaptive learning strategy that was formed was an idea consisting of three developed methods: Adaptive case selection, which is an adaptive assessment method that tries to measure the skills of the student, adaptive follow-ups, which is an adaptive content method that reacts to the mistakes made by the student, and adaptive task type, which is an adaptive content method that adapts the presentation and looks of the material presented to the student.

3.1 Requirements

3.1.1 Stakeholders

Even though the focus for this project is to test how adaptive learning can be applied in chest X-ray interpretation, there are still many different stakeholders that may have other interests in the system. The success of this project can be linked to these different stakeholders' interests, so the system design should therefore consider these interests, but at the same time not loose focus on the main goal.

- **System users:** The direct users of the systems will be medical students who are taking courses for learning how to interpret X-ray images. Their interest in the system will be connected to usability and task relevancy. It is an evaluation based on opinions from the users that determines if the application of adaptive learning in the system is a success or not.

- **NTNU, department of circulation and medical imaging (ISB):** This stakeholder represents the other kind of users the system can have. They are indirect users like teachers and others connected primarily to the department of circulation and medical imaging. Their interests in the system are focused on the content and learning material the system provides. They would want the content to be correct and comply with their way of teaching. Another aspect of this stakeholder-group is their interest in learning analytics, which is how well they can monitor the progress of their students.
- **System creator/master-student:** The master-student for this project will be doing all the development of the system. The interest in the system for this stakeholder is related to how the system is relevant for the project goal, which is to apply and test adaptive learning in a new domain. The system could be perfect for the user interests, but fail to be relevant for the project goal, and thereby end with a bad result on the master-thesis. Therefore the interests of the system creator should also be taken into consideration when designing the system. Another aspect relevant for this stakeholder is that the project manages to create opportunities for using new and relevant technology.
- **NTNU, department of computer science (IDI):** This stakeholder is represented by the supervisor of this project, which was the initiator of the entire project. The interests in the system for this stakeholder is primarily focused on what new knowledge the system can contribute with to the field of adaptive learning.

3.1.2 Functional Requirements

The functional requirements presented in table 3.1 are created using the following requirement boilerplates:

- <system> must have <system functionality>
- <system> should have <system functionality>

The priority of each requirement is determined based on how relevant the requirement is for the main goal of the project.

Table 3.1: Functional requirements

ID	Requirement	Stakeholder	Priority
FR1	The learning system must have adaptive learning capabilities	System creator, IDI	high
FR2	The learning system should have login functionality	System users, ISB, System creator	medium
FR3	The learning system must have tasks showing images	System users, ISB	high
FR4	The learning system should have functionality for enlarging images	System users, ISB	low

3.1.3 Non-functional Requirements

In addition to functionality, the system must comply with some restrictions in order to be relevant for the stakeholders. There are also some laws and regulations which govern the use of the learning material needed. The system must also comply with these. The non-functional requirements presented in table 3.2 are created using the following requirement boilerplates:

- <system> must comply with <restriction>
- <system> should comply with <restriction>
- <system> must comply with <restriction> when <system activity>

For many of these requirements the stakeholder ”society” will be used. This stakeholder has not a direct interest in the system like the other stakeholders, but has an indirect interest in the system through the laws and regulations that affect the system.

The priority for these requirements are set based on how damaging effect it would have for the project and its goals if the system fails to fulfill the requirement.

Table 3.2: Non-functional requirements (restrictions)

ID	Requirement	Stakeholder	Priority
NFR1	The learning system must comply with national and international laws and regulations concerning privacy protection when handling medical learning material.	System creator, Society	high
NFR2	The learning system must comply with national and international laws and regulations concerning copyrights when handling medical learning material	System creator, Society	high
NFR3	The learning system should comply with general teaching practices used for chest X-ray interpretation	ISB	low
NFR4	The learning system must comply with professionally correct knowledge sources in radiology.	ISB, System users	high
NFR5	The learning system should comply with the teaching practices of the teachers that will use the system.	ISB, System users	low

3.2 Deciding the Adaptive Learning Approach

As described in the prestudy (see section 2.1), adaptive learning algorithms can roughly be divided into three different categories: adaptive content, adaptive assessment and adaptive sequence. A first step in developing an adaptive learning approach tailored for chest X-ray interpretation, is to identify which of the three categories that are suitable for the domain and scope of this project. The project scope is given by the requirements in section 3.1 and the available time and resources.

The first category to be considered was adaptive sequence. This category would involve creating some kind of model that collects data about users or compares all the users to find commonalities between them, and based on this divide them into groups, where users in the same group would be given the same recommendations for task material. Possible algorithms here could be clustering algorithms like HAC or K-means, classification algorithms or collaborative filtering algorithms, as mentioned in section 2.1.2. Unfortunately a big drawback with these methods is the cold start problem [14], and a general need for data about a user in advance. This kind of data is called training data. The cold start problem will cause the system to give bad recommendations in the beginning and gradually get better, which is not ideal, since the bad recommendations could have a negative effect on user motivation and user opinion of the system. There are many attempts and solutions for reducing the effect of the cold start problem [22; 23], but none of these were found applicable in this case. An alternative is to have a session with students first for gathering training data that could be used for later. A problem with such a session is that there is a limited amount of students available for testing. In order for an adaptive sequence approach to work properly, there should be a large amount of available users for supplying training data. Also some training data collected might be sensitive data, which must be stored and managed in a specific way, which causes more work. After taking all these factors into consideration, it was decided not to go for an adaptive sequence approach for the system.

Secondly adaptive assessment was considered. An algorithm in this category would mean that the system would rank all tasks with a value for how difficult the task is, and at the same time try to measure or record a difficulty level for each user, and based on this recommend tasks that match the user's difficulty level. The recommendations could be done similar to the SBTS knowledge matrix [10]. The advantage with this approach is that it does not need a large amount of learning material to be effective, and it does not suffer from the issues with cold start and need for training data as adaptive sequence does. However, to quantify the difficulty level of an X-ray image on a scale from 1 to 10 was considered by the domain experts to be not reliable. Rather the expert panel recommended quantifying the images in only three difficulty levels. This complicates an approach that recommends task material based on difficulty level. However an X-ray image does always fall into one or several different categories. The theory here is then that a student might struggle more with one category than another. The difficulty of an image might therefore vary for each student depending on which categories the image falls into. By measuring the user performance for each category, it might thereby be possible to get a difficulty value for an X-ray image based on what categories the image falls into. It was decided that this might be something worth investigating further, and this adaptive assessment approach was chosen to be a part of the adaptive learning strategy for the system.

Even though it was decided to go with an adaptive assessment approach, it was soon clear that just the adaptive assessment strategy might be too simple. The system could potentially be too monotone with the same type of task all the time and no variation, except for the variable difficulty. And if the adaptive assessment approach should fail to adapt to the user, there would be little to learn from the system. Therefore adaptive content elements were also considered in combination with the adaptive assessment approach. There are lots of opportunities in adaptive content, but in this project the amount of available learning material limited the possibilities. It was therefore decided to try and adapt the way a task is presented. In other words, different looking tasks are based on the same learning material, but presented in different ways. The adaptive aspect here is that the system should be able to discover which task types the user likes the most or learns the most from, and thereby giving one type of task more often than another. It was also suspected that adaptive feedback might be ideal for teaching chest X-ray interpretation, but creating personalized or detailed feedback is a time consuming task that also would have to depend heavily on the domain experts. Therefore it was difficult to come up with an idea for how adaptive feedback could be tested.

Based on the discussion above, an adaptive learning strategy tailored for chest X-ray interpretation was developed and implemented into a prototype system. Three different methods were developed, where each method took inspiration from one of the three possible adaptive learning categories described in section 2.1.2. The adaptive learning strategy that was implemented and tested in the prototype system consisted of the following methods:

- Adaptive case selection: An adaptive assessment method that measures the difficulty of an image/case by looking at which categories the case falls into, and how the user has performed previously on those categories. It then suggests cases based on these difficulty values.
- Adaptive follow-ups: An adaptive content method that based on mistakes the user does in one task, called a main task, creates follow-up tasks that focus on just those categories of knowledge that the user struggled with in the main task.
- Adaptive task type: An adaptive content method that adapts the presentation and look of the task based on the performance of the user. The idea is that each task type should focus on a specific set of learning theories or approaches, so by applying different task types, the system indirectly applies different learning theories and approaches.

Each of these methods will be explained more in detail in section 3.4.

3.3 Concept Overview

The strategy described in section 3.2 was a result of an iterative design- and development process where ideas were constantly changing due to new requirements and general feedback from the domain experts. Figure 3.1 shows a simplified description of how the adaptive learning approach was implemented in the prototype system.

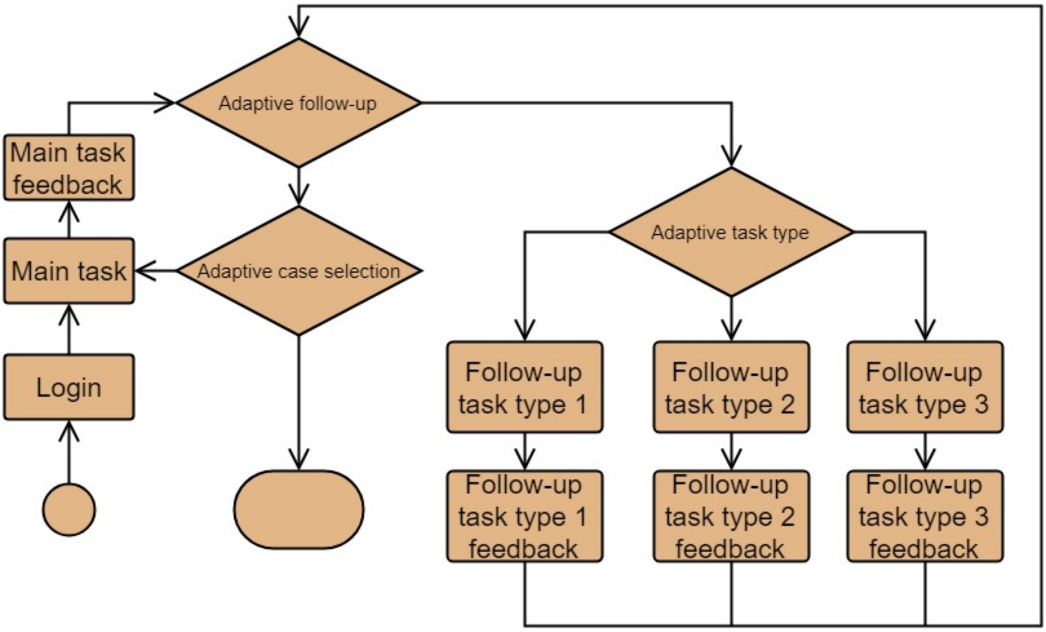
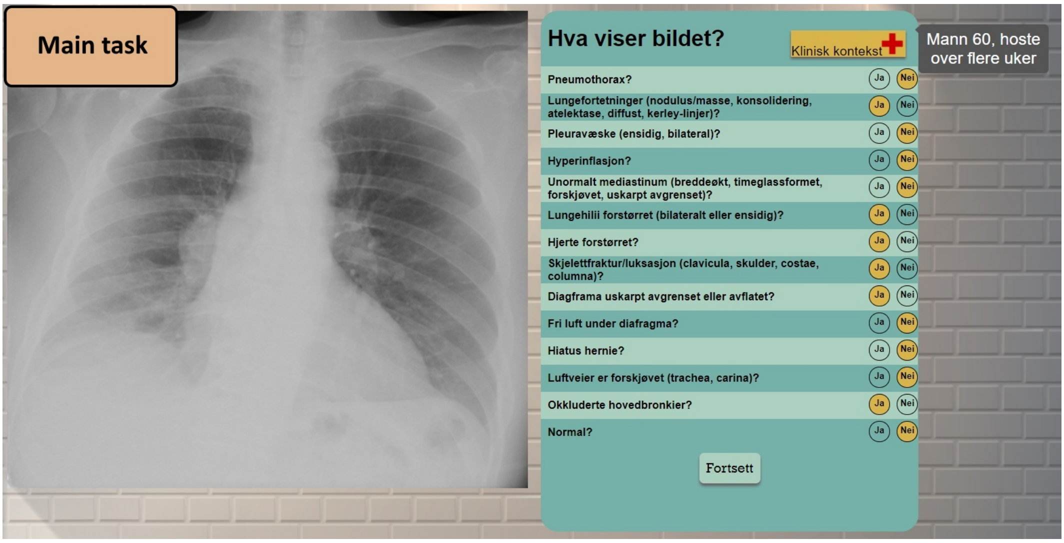


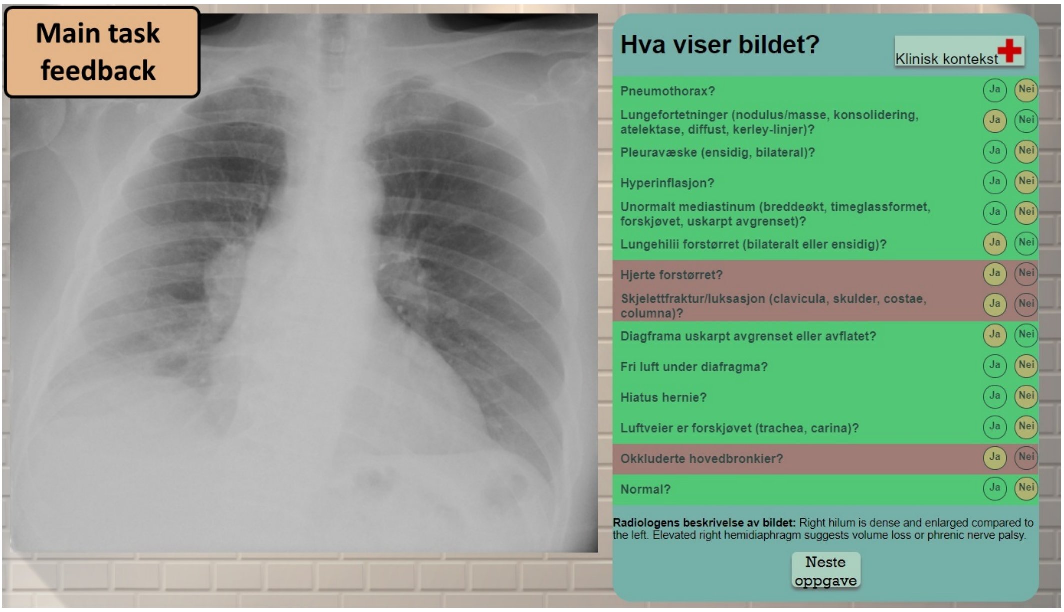
Figure 3.1: This diagram shows a simplified representation of the system flow. The simplification is done in order to show how the different user interface views relate to each other and to the three adaptive learning algorithms in the system. The algorithms are represented by the diamond-symbols, and redirects the system flow to the correct user interface view.

Each of the squares in the system flow diagram in figure 3.1 represents a user interface view. How each of these views looks like can be seen in figures 3.2, 3.3, 3.4, 3.5 and 3.6. A more proper description of the task designs and the algorithms behind the system flow will be given later in sections 3.4 and 3.5.



Hva viser bildet?

Pneumothorax?	○ Ja	● Nei
Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)?	● Ja	○ Nei
Pleuravæske (ensidig, bilateral)?	○ Ja	● Nei
Hyperinflasjon?	○ Ja	● Nei
Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)?	○ Ja	● Nei
Lungehilii forstørret (bilateralt eller ensidig)?	● Ja	○ Nei
Hjerte forstørret?	● Ja	○ Nei
Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)?	● Ja	○ Nei
Diagframa uskarpt avgrenset eller avflatet?	● Ja	○ Nei
Fri luft under diafragma?	○ Ja	● Nei
Hiatus hernie?	○ Ja	● Nei
Luftveier er forskjøvet (trachea, carina)?	○ Ja	● Nei
Okkluderte hovedbronkier?	● Ja	○ Nei
Normal?	○ Ja	● Nei

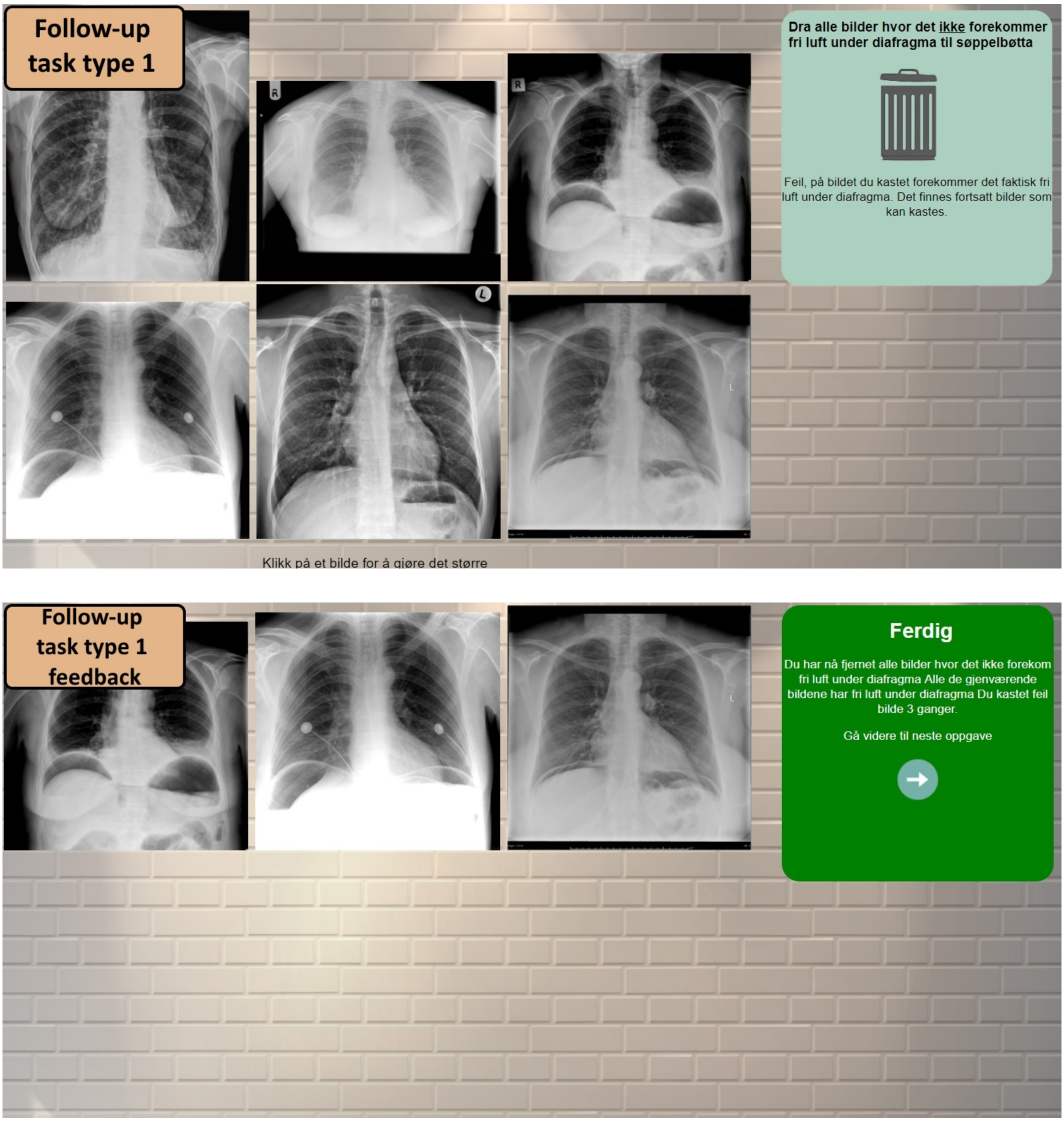


Hva viser bildet?

Pneumothorax?	Ja	Nei
Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)?	Ja	Nei
Pleuravæske (ensidig, bilateral)?	Ja	Nei
Hyperinflasjon?	Ja	Nei
Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)?	Ja	Nei
Lungehilii forstørret (bilateralt eller ensidig)?	Ja	Nei
Hjerte forstørret?	Ja	Nei
Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)?	Ja	Nei
Diagframa uskarpt avgrenset eller avflatet?	Ja	Nei
Fri luft under diafragma?	Ja	Nei
Hiatus hernie?	Ja	Nei
Luftveier er forskjøvet (trachea, carina)?	Ja	Nei
Okkluderte hovedbronkier?	Ja	Nei
Normal?	Ja	Nei

Radiologens beskrivelse av bildet: Right hilum is dense and enlarged compared to the left. Elevated right hemidiaphragm suggests volume loss or phrenic nerve palsy.

Figure 3.2: Main task



Dra alle bilder hvor det ikke forekommer fri luft under diafragma til søppelbøtta

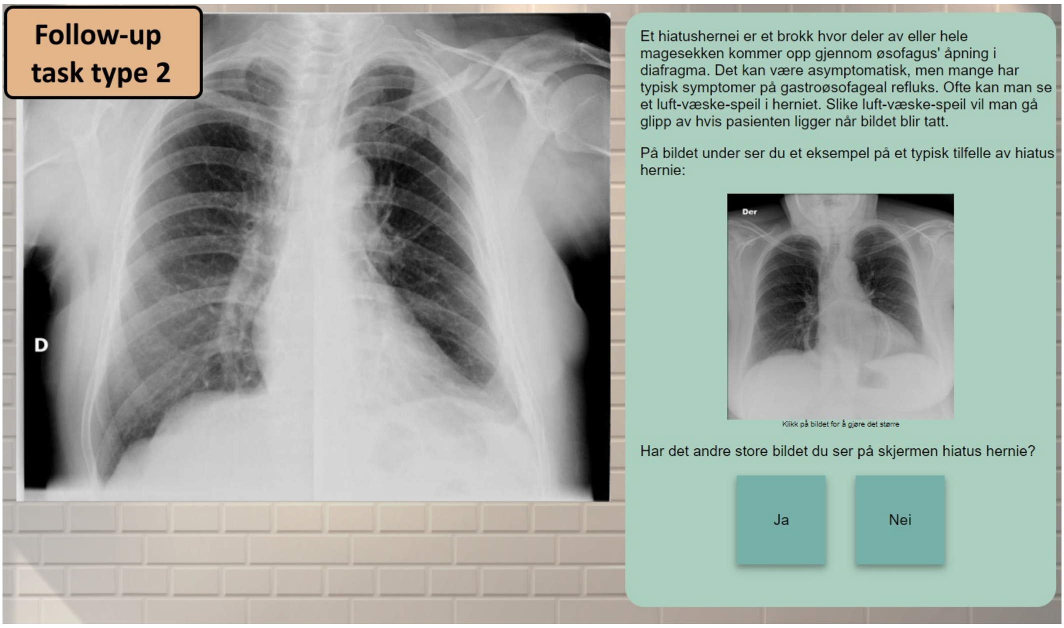
Feil, på bildet du kastet forekommer det faktisk fri luft under diafragma. Det finnes fortsatt bilder som kan kastes.

Klikk på et bilde for å gjøre det større

Du har nå fjernet alle bilder hvor det ikke forekom fri luft under diafragma Alle de gjenværende bildene har fri luft under diafragma Du kastet feil bilde 3 ganger.

Gå videre til neste oppgave

Figure 3.3: Follow-up task type 1

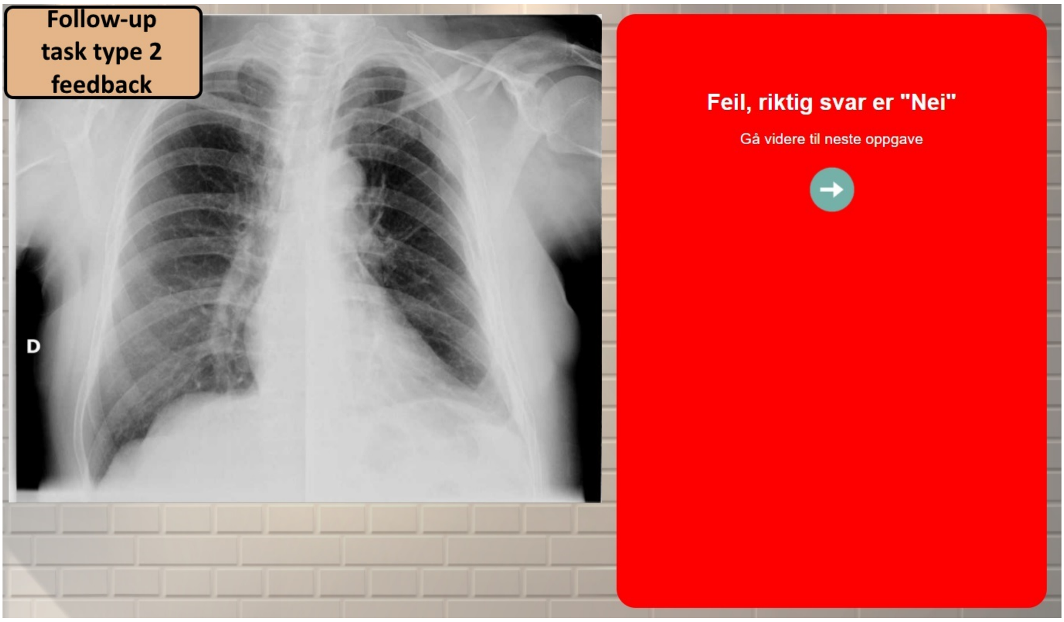


Et hiatushernei er et brokk hvor deler av eller hele magesekken kommer opp gjennom øsofagus' åpning i diafragma. Det kan være asymptomatisk, men mange har typisk symptomer på gastroøsofageal refluks. Ofte kan man se et luft-væske-speil i herniet. Slike luft-væske-speil vil man gå glipp av hvis pasienten ligger når bildet blir tatt.

På bildet under ser du et eksempel på et typisk tilfelle av hiatus hernie:

Klikk på bildet for å gjøre det større

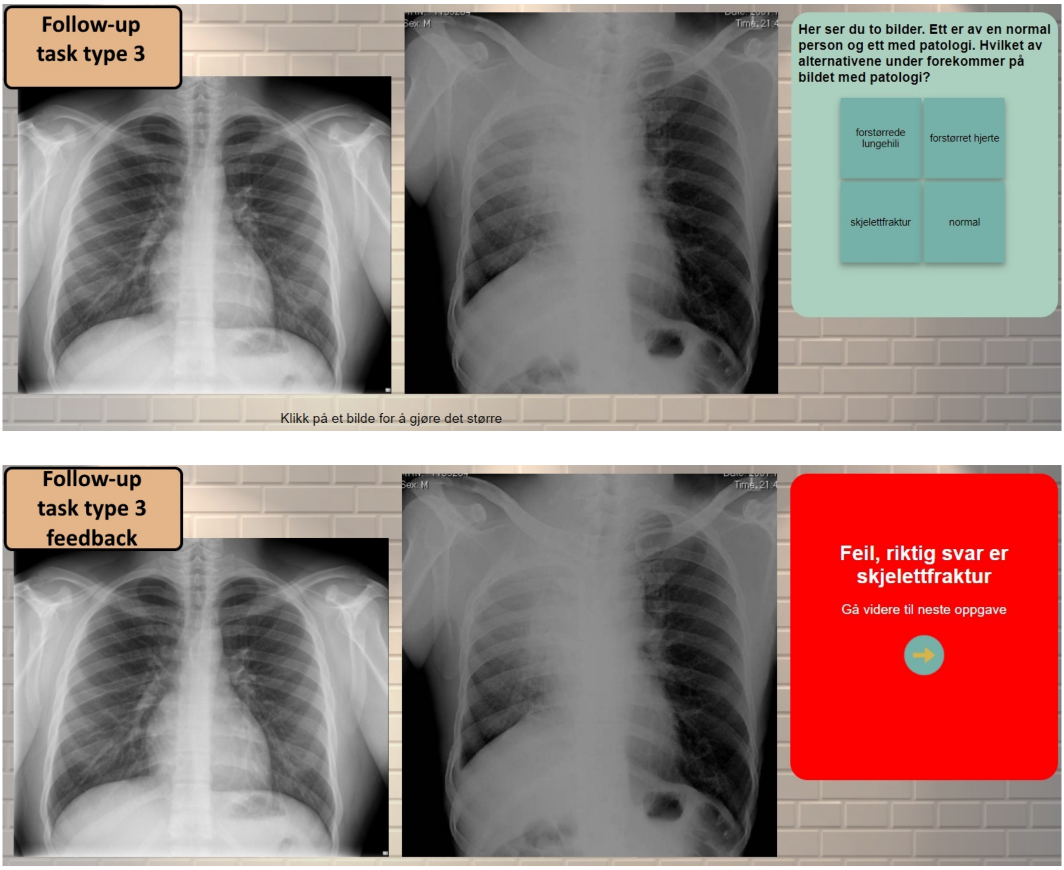
Har det andre store bildet du ser på skjermen hiatus hernie?



Feil, riktig svar er "Nei"

Gå videre til neste oppgave

Figure 3.4: Follow-up task type 2



Her ser du to bilder. Ett er av en normal person og ett med patologi. Hvilket av alternativene under forekommer på bildet med patologi?

Klikk på et bilde for å gjøre det større

Feil, riktig svar er skjelettfraktur

Gå videre til neste oppgave

Figure 3.5: Follow-up task type 3

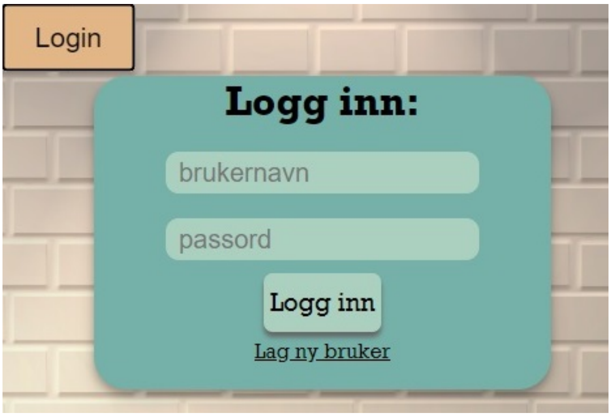


Figure 3.6: Login

3.4 Adaptive Learning Algorithm

As mentioned in section 3.2, the developed adaptive learning strategy for the system primarily consists of three methods: adaptive case selection, adaptive follow-ups and adaptive task type. These three methods worked together as one algorithm to create a personalized learning experience for the user.

3.4.1 Case Description

The algorithm was supplied with 95 cases created by the domain experts. A case consisted of one thorax frontal X-ray image, clinical context, difficulty level, category data and a comment from the domain expert about the case. The category data consisted of 14 boolean values. Each value stated if a specific kind of pathology was present in the image or not. In this context the word ”pathology” means a deviation from what is assumed to be a normal condition. The categories the boolean values represent are the same as mentioned earlier in table 2.1. The difficulty level was a number between 1 and 3 stating how difficult it is to diagnose the image correctly, where 3 is hardest and 1 is easiest. Clinical context is a description of what information the doctor has available before seeing the X-ray image. These data were the main data available for basing the algorithm on. All the images were found on the page Radiopedia[1]. When solving a case, the user has to manage to input the right boolean value for every category (answer yes or no on 14 questions about an X-ray image).

Table 3.3: Example case

ID	1
Image	img23.jpg
Clinical context	Man 25 years old. Hit by a car. Complaining about upper left chest pain.
Difficulty	2
Category data	[0,0,0,1,0,1,1,1,0,0,0,0,0,0]
Comment	Left sided tension pneumothorax.

[1] https://radiopaedia.org/ Accessed: 24.05.2018

3.4.2 Adaptive Case Selection

Each time the system needs to show the user a main task (see figure 3.2) it is in need of retrieving a case (see section 3.4.1) to build that task from. The adaptive case selection algorithm's job is to pick a case among the 95 available cases, that is suitable for the specific user's needs. This means finding a case that has a category that the user needs more practice in.

The adaptive case selection is an adaptive assessment approach, but due to the way it defines case difficulty it has some similarities with adaptive content. As mentioned earlier, the adaptive assessment approach chosen should measure the difficulty of a case through which of the 14 categories the case has, and how the current user has performed previously in those categories. This because the cases provided only have three levels of difficulty, which is too few for other adaptive assessment approaches discovered in the prestudy, like for example SBTS [10]. After calculating a measure of difficulty for each of the 95 available cases, the algorithm should choose one of those cases that scores high on difficulty, which indicates that the task has categories that the user struggles with, and thereby needs more practice in. This approach assumes that learning is achieved through focusing more on those categories that need improvement, and less on those categories that do not need improvement. It also relies on the tasks to have educational effect, and by focusing on difficult cases, the teaching is focused towards those difficult categories.

Another important aspect is that even though the algorithm discovers one subset of the categories that the user is struggling with, it should not lock into just recommending only cases with these categories and no other cases. A risk then is that the system never manages to measure the user's performance level for all the categories, since it only recommends cases containing categories from a smaller subset. Also Rozenshtein et al. [24] recommends an interleaved teaching method where categories are mixed over a massed teaching method where categories are presented in bunches. The solution to this potential issue is that the algorithm should treat the difficulty measures for the tasks as probabilities. Higher measure of difficulty equals that it is more likely that the case is chosen, but at the same time there is a chance that the algorithm will choose one case that contains new categories, and thereby making it possible to measure the user's performance level for these categories as well. This also ensures that categories are interleaved instead of bunched. Another effect from this strategy is that some easy tasks might be chosen from time to time, and thereby keeps the student's motivation up. The adaptive case selection algorithm, which tries to take all the aspects mentioned into consideration, can be described with the following steps showed in figure 3.7, and is further described in the following paragraphs.

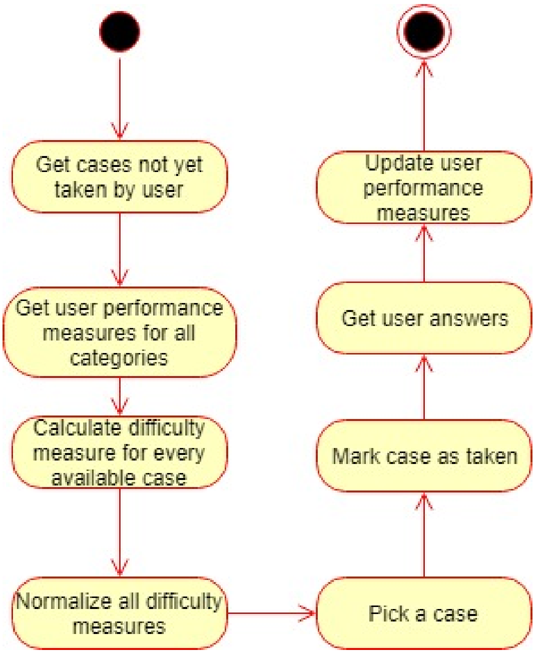

Get cases not yet taken by user
Get user performance measures for all categories
Calculate difficulty measure for every available case
Normalize all difficulty measures
Pick a case
Mark case as taken
Get user answers
Update user performance measures

Figure 3.7: This diagram shows all the steps the adaptive case selection algorithm goes through in order to adaptively choose a case that is suitable for the user

Get cases not yet taken by user

To avoid repeating the same case, the algorithm should keep track of which cases each user has already taken. Before starting to calculate difficulty for the cases, the already taken cases should therefore be filtered out to avoid unnecessary computations and repetition of the same case.

Get user performance measures for all categories

The algorithm should save and keep track of each user's scores for every category. That is, each user has 14 saved score numbers, from now on called Category Performance Measures (CPM). One for each of the 14 possible categories a case can have. The scores are updated each time the user fails or succeeds in a category when he/she is solving a case.

Calculate difficulty measure for every available case

When all available cases and saved category performance measures have been collected for the user, the algorithm will iterate through all the available cases. For each case the CPMs for the categories that are present in the case will be summed. This sum gives us the User Difficulty Measure (UDM) for the case. Note that this difficulty measure should not be confused with the general case difficulty, which is only a value between 1 and 3.

The UDM is a measure of how difficult the case will be for this specific user based on the user's previous performance on other cases.

$$UDM(u,y) = \sum_{i=0}^{i=13} CPM(u,x_i), \text{where category } x_i \text{ is present in case y} \tag{3.1}$$

- i here runs from 0 to 13 because there are 14 possible categories a case can have. Only those categories that are present in the case y will be summed.
- UDM(u, y) is the user difficulty measure of the case y for user u
- CPM(u, x) is the currently stored category performance measure for user u of category x

Normalize all difficulty measures

After the previous step the algorithm has calculated a set of UDMs. One for each case the user has not yet encountered. The UDMs will be sums that could be on a pretty large scale, from the lowest being a large negative number, to the highest being a large positive number. Low values indicate cases containing categories the user needs to improve on, and high values indicate cases containing categories the user does not need to improve on. The system should therefore prioritize cases that have low UDM values. But, as mentioned earlier, we do not want the algorithm to only pick cases we know for certain that the user struggles with. There should be a chance that the system also picks a case with new categories or categories the user is a bit good at, but could still need some more practice on. The algorithm will therefore not just pick the case with the lowest UDM value. It will use the UDM values to create probability values for how likely it is that a case will be picked. This is done by normalizing each UDM value. In this context the concept "normalize" means transforming the value to a value on the scale between 0 and 1, and thereby making it easier to compare the different cases up against each other. Since the lowest UDM values represent those cases that there should be highest probability of picking, these should get values closest to 1, and those with the highest UDM values should be transformed to values close to 0. Through some experimentation with different functions, function 3.2 ended up looking promising for the purpose of normalizing the UDM values.

$$normalizedUDM(u,y) = -\frac{e^{\frac{3*UDM(u,y)}{maxAbsUDM(u)}}}{e^{\frac{3*UDM(u,y)}{maxAbsUDM(u)}}+1} + 1 \tag{3.2}$$

- normalizedUDM(u, y) refers to the normalized value of the UDM-value for case y for user u.
- maxAbsUDM(u) refers to the maximal absolute UDM value for the user u. This will be the absolute value of the UDM-value for either the case that the user would struggle the most with, or the case that the user would find easiest. The maxAbsUDM functions as a border for when the function should converge to either 0 or 1.

As can be seen in figure 3.8 the function will give each case a value between 0 and 1, where cases with the lowest UDM-value will get a value closer to 1, and those cases having high UDM-values will get values closer to 0.

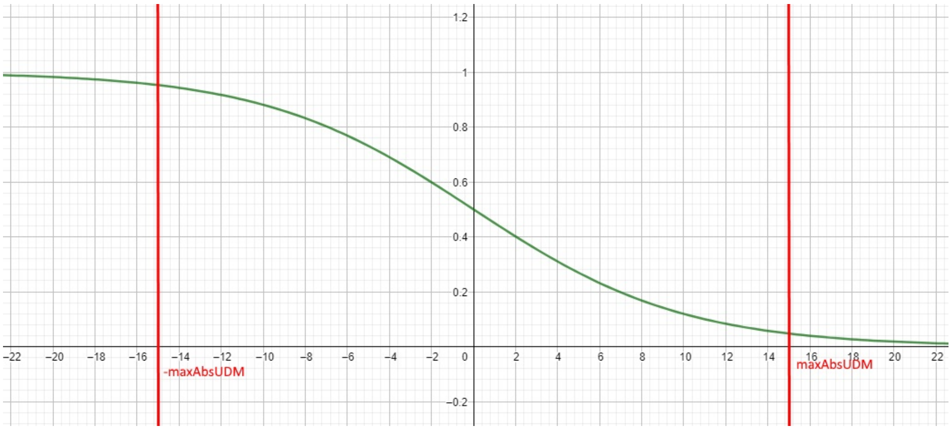


Figure 3.8: The figure shows an example of the function for normalizing the UDM-values. As can be seen, the maxAbsUDM value defines the border for when the function value starts to converge towards 1 or 0. The x-axis here refers to the input UDM-value for the case. The y-axis refers to the normalized value for the case. In this example maxAbsUDM is set to 15.

Pick case and mark case as taken

When all cases have received a value between 0 and 1, it is time to pick one case. Here it is important to say that the normalized value for each case is not the direct probability for that case to be chosen. This since many cases might have for example 0.9 as a value, and all of them can not at the same time have 90 percent probability of beeing picked. The value for each case might instead be viewed as a value saying how likely one case is compared to another case. For example, a case having the value 0.8 has two times higher probability of beeing chosen than a case having the value 0.4. The picking process can be illustrated as a lottery bowl containing notes for all the cases. The case with value 0.8 will have 80 notes in the bowl, and the case with value 0.4 will have 40 notes in the bowl. From the bowl only one note is picked to be the case that should be given to the user. After the case is picked and solved by the user, it is marked as taken, and will not be given to that specific user again.

Get user answers and update user performance measures

When the user has solved a case, the results will be used to update the saved CPMs for the user. Note that one case can have multiple categories at the same time. So one case can yield a negative score for some categories and a positive score for other categories. For each of the 14 categories a case can possibly have, the score is rewarded based on the following function:

$$score(u,x,y) = \begin{cases} \text{-(4-caseDifficulty(y))} & \text{, when category x is failed for case y} \\ \text{caseDifficulty(y)} & \text{, when category x is succeeded and present in case y.} \\ 0 & \text{, when category x is succeeded, but not present in case y} \end{cases} \quad (3.3)$$

- score(u, x, y) refers to the score added to the CPM(u, x) based on the performance of user u on case y.
- caseDifficulty(y) refers to the difficulty of the case y as described in section 3.4.1

The result of the function is added to the users's currently saved score for the category (CPM(u, y)). This means that when failed, the saved CPM will get lower. And when success, the saved CPM will increase. Also note that a saved category-score could become negative if the user repeatedly fails on the same category in different cases. The variable ”caseDifficulty(y)” refers to the case difficulty mentioned earlier in section 3.4.1. This value is between 1 and 3. The function is constructed so that if a user fails an easy case, the punishment will be high, and the reward of success will be low. On a difficult case it is the opposite. Here success will yield high reward, while failure will yield low punishment. On medium difficulty the reward is equally large as the punishment. The difference between success rewards where some successes yield 0 points and some yield caseDifficulty(y) is because it is generally more likely that a case does not have a category than having it. Among the 14 possible categories, a case usually does not have more than five at the same time. Therefore to avoid that the user gets score points by only guessing the answer ”No” (category not present in current case) for all the categories, score is only rewarded when a category is present in the case and is succeeded.

3.4.3 Adaptive Follow-ups

The adaptive follow-ups is an adaptive content functionality that is based on the principle of branching (see section 2.1.2). After the user has solved a case, by answering yes or no on 14 different questions about an image, there will most likely be some questions that are answered incorrectly. It is here this adaptive follow-up functionality will kick in. The tasks given to the user can be divided into two different types. The main task (see figure 3.2), which is to solve a case by answering the 14 questions, and follow-up tasks (see figures 3.3, 3.4 and 3.5), which are tasks that focus on only one of the 14 categories at the same time. After the main task, the user will be given one follow-up task for each of the categories that were answered incorrectly. There will be different kinds of follow-up tasks in order to create variation. The follow-up tasks will be based on the same data as the cases, but they will not use the entire case as the main task is doing, but just pick out what they need. Mostly this is just the image itself and the information that confirms that the image has the category the follow-up task is targeting. More details about these follow-up tasks will be described in section 3.5. When all the follow-up tasks have been answered, the user is given a new main task. That is, a new case is adaptively picked as described in section 3.4.2.

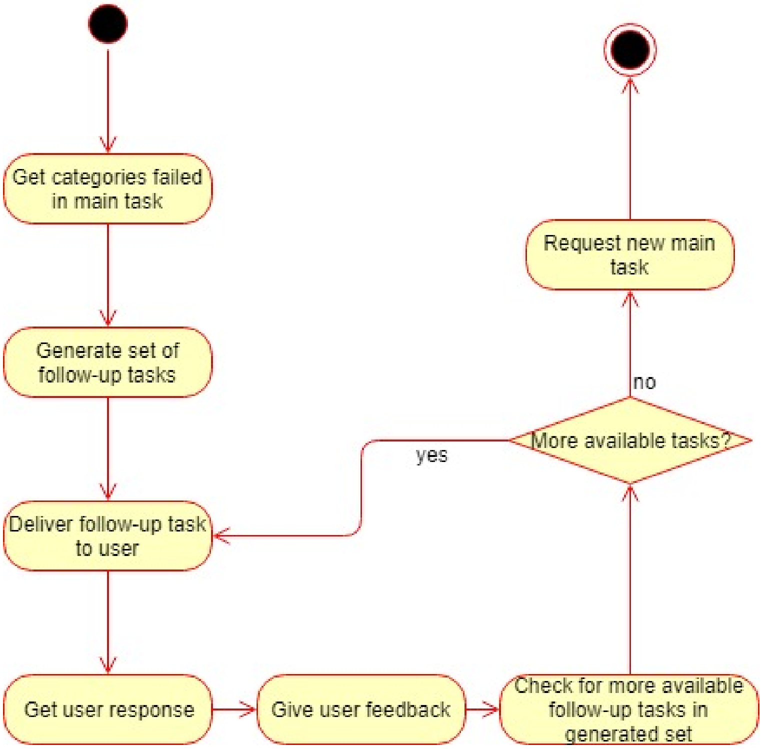

Get categories failed in main task
Generate set of follow-up tasks
Deliver follow-up task to user
Get user response
Give user feedback
Check for more available follow-up tasks in generated set
More available tasks?
yes
no
Request new main task

Figure 3.9: The diagram shows the steps taken by the system in order to give follow-up tasks based on the incorrect answers from the main task.

3.4.4 Adaptive Task Type

The last main element of the adaptive learning algorithm is adaptive task type, which can be described as a form of adaptive content. The adaptive task type algorithm will kick in at the ”Generate set of follow-up tasks” in figure 3.9. As mentioned earlier this is a functionality in the system that tries to identify which form of follow-up task that fits best to a specific user’s preferences. The system has three variations of follow-up tasks. They target different ways of analyzing the X-ray images and are based on different learning theories and principles. The idea is that by having one follow-up task covering one principle, and another covering a completely different principle, the system can apply the principle that works best for each student by selecting the right follow-up task. More details about the differences between the follow-up tasks will be described in section 3.5.

Just as with the adaptive case selection in section 3.4.2 we do not want the system to only recommend one task type all the time just because it is identified as the best task type for the user. This would create a monotone user experience, and close the possibility for other task types to be considered for the user. The algorithm should just make sure that the probability for some task types are higher than for others so that the best task type occurs most of the time, but not always. The method of deciding which task type is best, assumes that the task type where the user succeeds the most is the best task type. A risk here is that if one task type is generally easier than the others, every user will get the same task type recommended. Figure 3.10 describes the steps taken by the algorithm in order to adaptively pick a task type for a follow-up task

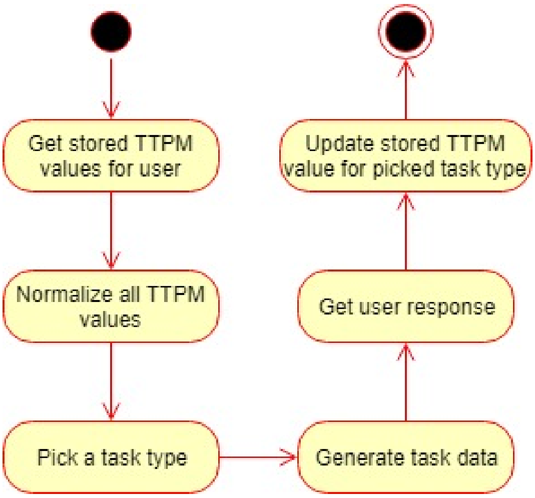

Get stored TTPM values for user
Normalize all TTPM values
Pick a task type
Generate task data
Get user response
Update stored TTPM value for picked task type

Figure 3.10: The figure shows the flow of the algorithm when it selects task type for one follow-up task. This flow is repeated for each follow-up task the user is given.

Get stored TTPM values for user

For each user there is a set of performance measures stored. The set contains one performance measure for each possible task type. This measure can be seen as a score, telling how well the user has performed on a specific task type. In this case the system has three different task types, so three performance measures are stored for each user. These measures will from now on be referred to as Task Type Performance Measures (TTPM).

Normalize all TTPM values

Since the needs of the adaptive task type algorithm are so similar to the needs of the adaptive case selection in section 3.4.2, a variant of the same function for normalization was used. Again the concept of normalization was to get the stored performance measures on a scale between 0 an 1 to better be able to compare them. Note that in this case a large positive performance measure should get a normalized value closer to 1, unlike in the adaptive case selection where a large positive performance measure would get a value closer to 0 instead.

$$normalizedTTPM(u,y) = \frac{e^{\frac{3*TTPM(u,y)}{maxAbsTTPM(u)}}}{e^{\frac{3*TTPM(u,y)}{maxAbsTTPM(u)}} + 1} \tag{3.4}$$

- TTPM(u, y) refers to the currently stored TTPM-value for user u concerning the task type y
- maxAbsTTPM(u) refers to the highest absolute value among the three stored task type performance measures for user u. It functions as a border just like in equation 3.2.
- normalizedTTPM(u, y) refers to the normalized value for the TTPM value for user u on task type y, and is a value between 0 and 1, where values closest to 1 represent task types that are best suited for the user, and values closest to 0 represent unsuitable task types for the user.

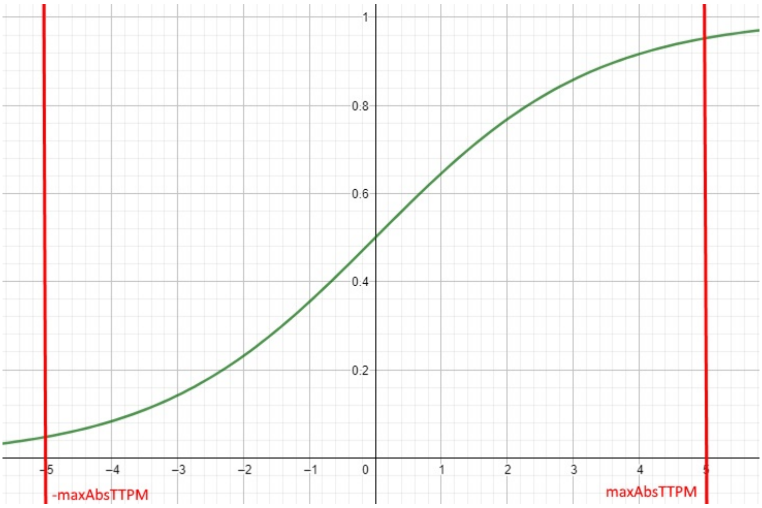


Figure 3.11: The figure shows an example of a function for normalizing the TTPM-values. The x-axis refers to the input TTPM-value. The y-axis refers to the normalized TTPM-value. The maxAbsTTPM functions as an upper and lower border for when the function value should begin to converge towards 0 or 1. In this example maxAbsTTPM is set to 5. This means that either 5 is the highest TTPM stored for the user or -5 is the lowest stored TTPM for the user.

Pick a task type

When picking a task type, the algorithm again uses the same principle as the adaptive case selection. If a task type has a normalized value closer to 1, it has more notes in the lottery bowl than a task type with normalized value closer to 0, thereby making it more likely that the task types with the highest values get picked.

Generate task data

When a task type is picked the system has to get the necessary data in order to construct the follow-up task according to the task type.

Get user response and update stored TTPM value for picked task type

When a user solves a follow-up task he/she could succeed or fail. Based on the result the system will grant the user points. How points are granted varies for each task type, and will be further described in section 3.5. But generally if a user succeeds on a follow-up task, the stored TTPM will be increased for the task type of that specific follow-up task. However if the user fails, the system will decrease the stored TTPM value for the task type of that specific follow-up task.

3.4.5 Overview

Figure 3.12 shows the flow of the entire system. The conditional ”All cases taken by user?” breaks the continuous flow of the system if all the cases have been used up. Also note the conditional ”Category set empty?”. When this evaluates to ”yes” it means that the user has answered correct on all the 14 questions of the case in the main task. Follow-up tasks are thereby skipped and a new case and main task is retrieved instead.

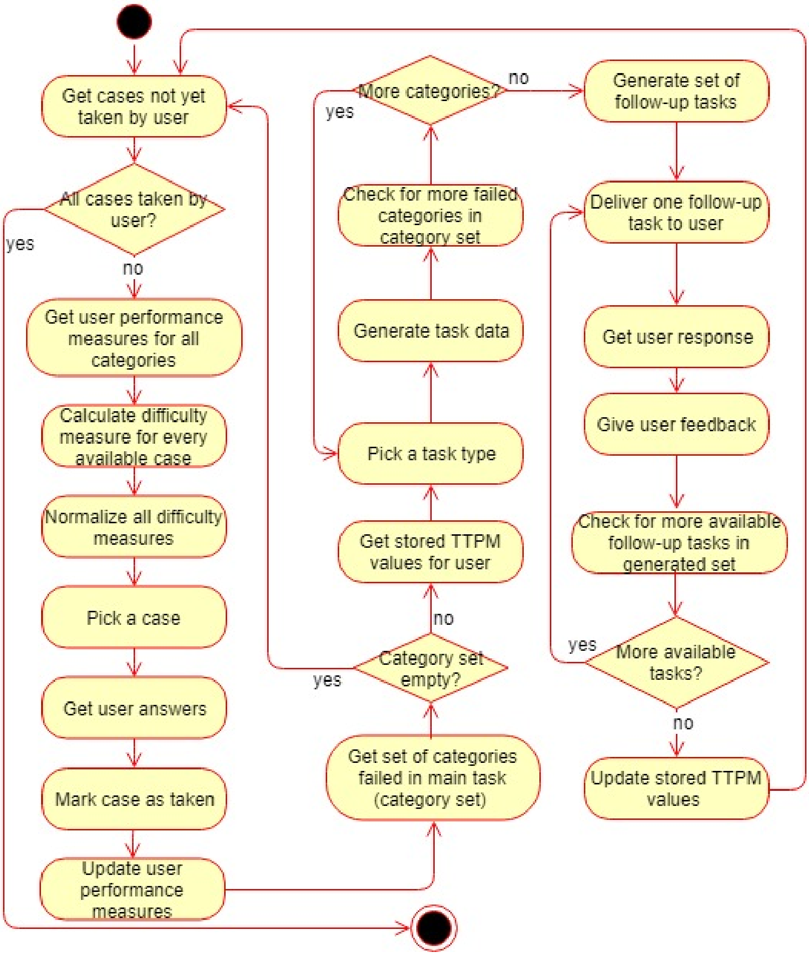

Get cases not yet taken by user
All cases taken by user?
yes
no
Get user performance measures for all categories
Calculate difficulty measure for every available case
Normalize all difficulty measures
Pick a case
Get user answers
Mark case as taken
Update user performance measures
More categories?
yes
no
Check for more failed categories in category set
Generate task data
Pick a task type
Get stored TTPM values for user
no
Category set empty?
yes
Get set of categories failed in main task (category set)
Generate set of follow-up tasks
Deliver one follow-up task to user
Get user response
Give user feedback
Check for more available follow-up tasks in generated set
yes
More available tasks?
no
Update stored TTPM values

Figure 3.12: The diagram shows the flow of the entire system when the three main adaptive elements ”Adaptive case selection”, ”Adaptive follow-ups” and ”Adaptive task type” are put together.

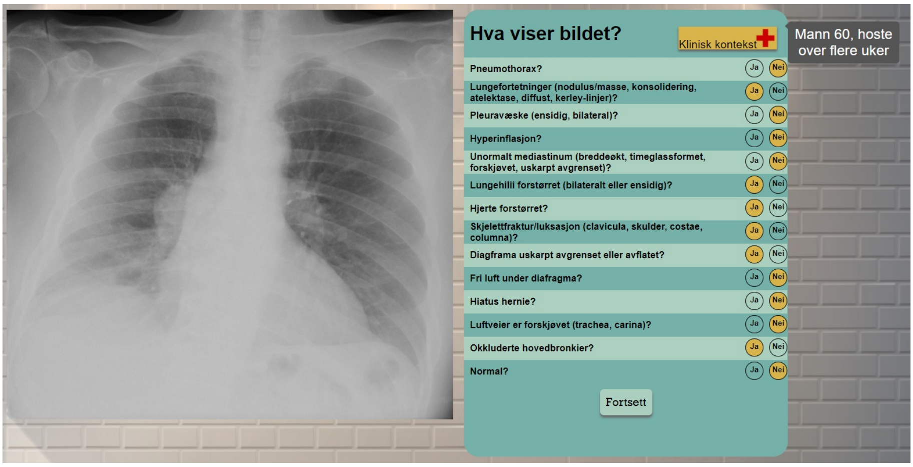
3.5 Task Design

As mentioned earlier, the system has four different kinds of task types. The main task is always given for each case. The follow-up tasks are only given for those categories the user failed in the main task. Which one of the three follow-up types that is chosen for a category is decided by the adaptive task type algorithm, as described in section 3.4.4. Each of the four different task types represent different approaches and principles for how to teach chest X-ray interpretation. Each task also affects the algorithm's choice of the next task types and cases, since stored measures it uses (CPM and TTPM) are increased or decreased based on the user's performance on the tasks. How this manipulation of the stored measures should be done proved to be the most challenging part of the design, since it was hard to find the right balance between adaptation and variation of task material.

3.5.1 Main Task

Description

The main task presents all the available data found in a case (see section 3.4.1 for more details about the case structure). The task presents one large X-ray image and asks the question ”What does the image show?”. Then it lists all the 14 categories a case could possibly have. The user then has to interpret the image and answer ”Yes” or ”No” for each category. It is intentional that the user has to answer ”No” instead of just ignoring those categories that he/she thinks are not present in the image. This to force the user to actually consider each category, instead of just focusing on those categories that are obvious. The task also has a button for showing clinical context, that when clicked shows a speech bubble with the clinical context data for the case. This is intended as an aid for the user. The continue-button only appears when all categories are answered.



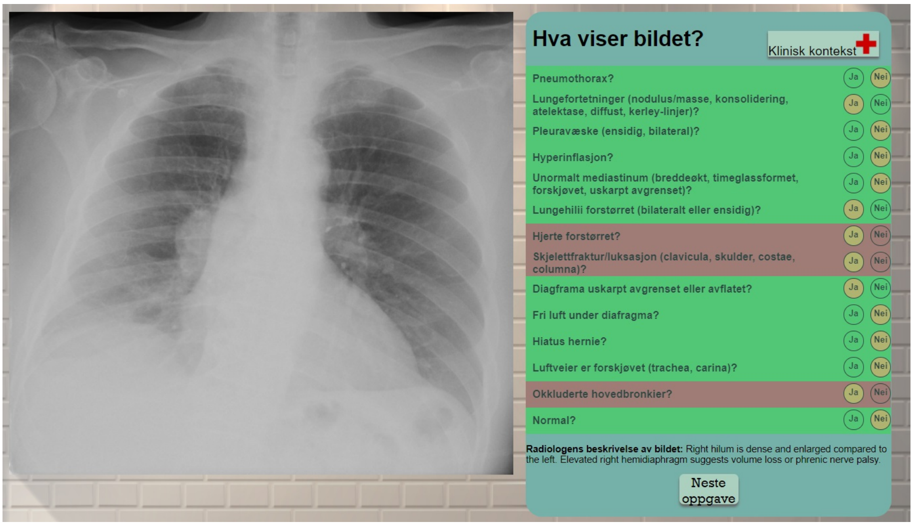
Hva viser bildet?

Mann 60, hoste over flere uker

- Pneumothorax? Ja [✓ Nei]
- Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)? [✓ Ja] Nei
- Pleuravæske (ensidig, bilateral)? Ja [✓ Nei]
- Hyperinflasjon? Ja [✓ Nei]
- Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)? Ja [✓ Nei]
- Lungehilii forstørret (bilateralt eller ensidig)? [✓ Ja] Nei
- Hjerte forstørret? [✓ Ja] Nei
- Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)? [✓ Ja] Nei
- Diagframa uskarpt avgrenset eller avflatet? [✓ Ja] Nei
- Fri luft under diafragma? Ja [✓ Nei]
- Hiatus hernie? Ja [✓ Nei]
- Luftveier er forskjøvet (trachea, carina)? Ja [✓ Nei]
- Okkluderte hovedbronkier? [✓ Ja] Nei
- Normal? Ja [✓ Nei]

Figure 3.13: The main task consists of 14 categories, an X-ray image and the clinical context button. The user has to mark each category with either "Yes" or "No" after analyzing the image.

When the continue-button is clicked, the task will generate feedback to the user. This is done by marking correct categories with green and incorrect categories with red. The task will also show a short text called "The radiologist's description of the image". This text is the commentary that the domain experts gave to each image when creating the task material. Mostly these texts are the same commentary texts that were found on the source page Radiopedia[2].



Hva viser bildet?

- Pneumothorax? Ja Nei
- Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)? Ja Nei
- Pleuravæske (ensidig, bilateral)? Ja Nei
- Hyperinflasjon? Ja Nei
- Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)? Ja Nei
- Lungehilii forstørret (bilateralt eller ensidig)? Ja Nei
- Hjerte forstørret? Ja Nei
- Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)? Ja Nei
- Diagframa uskarpt avgrenset eller avflatet? Ja Nei
- Fri luft under diafragma? Ja Nei
- Hiatus hernie? Ja Nei
- Luftveier er forskjøvet (trachea, carina)? Ja Nei
- Okkluderte hovedbronkier? Ja Nei
- Normal? Ja Nei

Radiologens beskrivelse av bildet: Right hilum is dense and enlarged compared to the left. Elevated right hemidiaphragm suggests volume loss or phrenic nerve palsy.

Figure 3.14: The user is given feedback on which categories he/she answered correctly and also gets the description of the image from the radiologist.

[2] https://radiopaedia.org/ Accessed: 24.05.2018

Applied principles and theories

The main task is the part of the system that applies the ABCDE-principle as described in section 2.3.3. In the main task all the 14 categories created by the domain experts based on the ABCDE-approach are included. By solving this type of task, the user gets used to analyzing the X-ray image through focusing on these 14 categories. Note that the task did not make any attempt to present the categories in the order given by the ABCDE-approach. The main focus of this task was to indirectly teach the user the correct method for analyzing an X-ray image, by forcing the user to think in terms of the 14 categories. The way the task is presented is also according to the Socratic way of teaching, which is found to be preferred among radiology students, compared to a didactic teaching approach [25].

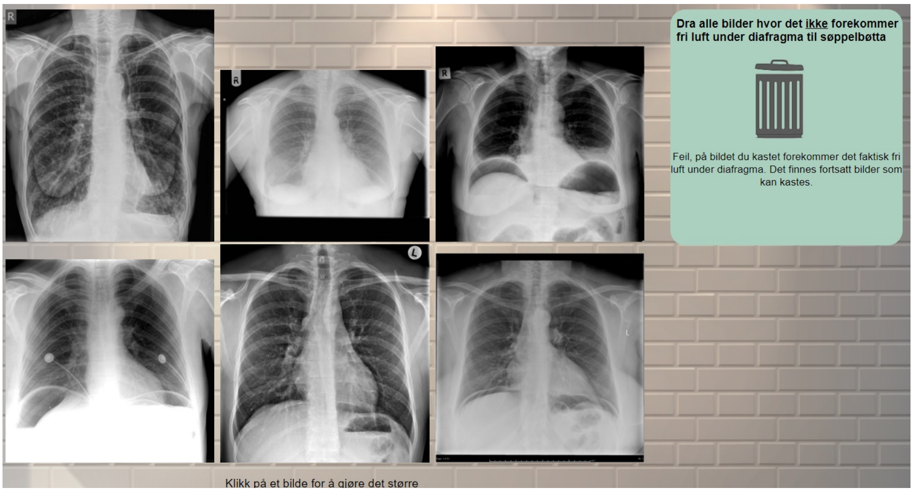
Effect on adaptive learning algorithm

The main task is the most important task when it comes to data collected by the adaptive learning algorithm. Each time the user finishes a case through the main task, all the 14 categories in the case are evaluated and generate scores which either increase or decrease the stored CPM-values for the user. These CPM-values are then later used by the adaptive case selection. Exactly how this is done is described earlier in the score-equation (equation 3.3). In addition to the use of the score-equation, the clinical context button also has an effect. It is intended as an aid, and thereby makes it easier to answer correctly. The idea therefore was that when this button is clicked, the possible rewarded score for each category, given by equation 3.3, should be halved. Meaning that if equation 3.3 for example outputs 3, the actual result value would be 1.5 instead. However this effect from the clinical context button proved not to be a good idea. It will be explained in later discussions why it was not a good idea. The main task also functioned as the base that told the system which categories it should construct follow-up tasks for. This means that after a main task, a series of follow-up tasks appeared. One follow-up task for every category that got marked red in the feedback described in figure 3.14.

3.5.2 Follow-up Task Type 1

Description

The first type of follow-up task presents the user with 6 different X-ray images. The user is then asked to remove all images that do not have a specific category. The removing of an image is done by dragging it towards a garbage bin and dropping it. The image will then disappear if it is correctly removed. At the same time a message appears saying that the user removed the image correctly, and that there are still some images left to remove. However if the image actually had the category in focus, the user will be told that he/she removed the image incorrectly, and the image will pop back to its place. Since the images are very small, the task has a functionality where the user can click an image so that a larger version of it will be shown as an overlay filling the entire screen. Since this is a follow-up task, it revolves around only one of the 14 categories at the same time. Which category it focuses on is given by the result of the last main task, so one of the failed categories from that main task will be the focus for this follow-up task. The task can have up to 3 correct images and minimum 1 correct image. The correct images are retrieved by scanning for cases in the database having the category in focus, while the incorrect images are randomly picked among those cases that do not have the category in focus.



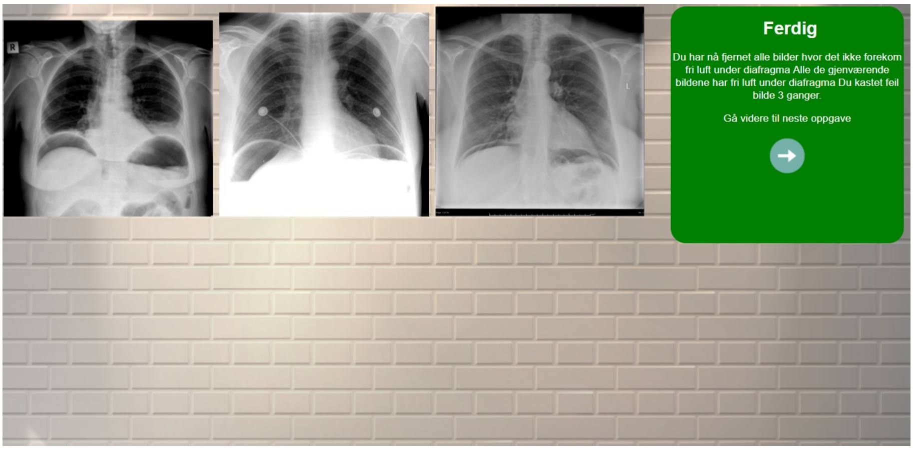
Dra alle bilder hvor det ikke forekommer fri luft under diafragma til søppelbøtta

Feil, på bildet du kastet forekommer det faktisk fri luft under diafragma. Det finnes fortsatt bilder som kan kastes.

Klikk på et bilde for å gjøre det større

Figure 3.15: The first type of follow-up task presents 6 images. The user has to sort out images by dragging them to the garbage bin.

When all the incorrect images are removed, a message appears telling the user how many times he/she did a mistake. Then the user can click on a next-button, which either takes the user to the next follow-up task, or a new main task, if all follow-up tasks have been completed.



Ferdig

Du har nå fjernet alle bilder hvor det ikke forekom fri luft under diafragma Alle de gjenværende bildene har fri luft under diafragma Du kastet feil bilde 3 ganger.

Gå videre til neste oppgave

Figure 3.16: When all images have been sorted out correctly, the system gives the user feedback

Applied principles and theories

This task type is designed based on a drill and practice philosophy (see section 2.2.2). This means that it presents a problem that should be solved, and then presents simple feedback based on the solution the user provides.

The problem this task type presents the user with primarily applies the principles of comparison as described in section 2.2.3. The idea is that by presenting all these 6 images at the same time, the system invites the user to do comparisons between them. By dragging and dropping, the user can get confirmation if an image is wrong or not. Based on this the user can compare the image that was found correct/wrong up against the other images in order to make a better decision on these images. The task thereby invites to the possibility of using methods like problem comparison, correct method comparison and incorrect method comparison as described in section 2.2.3.

Effect on adaptive learning algorithm

This follow-up task affects both the stored CPM-values and the stored TTPM-values. The TTPM-values affect the future prioritizations of which types of follow-up tasks the user should be given, while the CPM-values have an effect on which cases that should be given to the user in the main tasks. Both the TTPM and CPM get manipulated by the same score-value from the task, which is calculated using equation 3.5.

$$score(u, x, this) = \begin{cases} \text{1-correctProportion} & \text{, numberOfMisses} < \text{numberOfCorrectImages} \\ \\ \text{-correctProportion} & \text{, numberOfMisses} > \text{numberOfCorrectImages} \end{cases} \tag{3.5}$$

- score(u, x, this) refers to the value that will affect the stored CPM-value for user u for category x. The same value will also affect the stored TTPM-value for this task type for user u.

- correctProportion refers to the proportion of all the images that have the category in focus, compared to how many that do not have the category in focus. For example, if two of the 6 images have the category in focus, correctProportion would be 1/3.

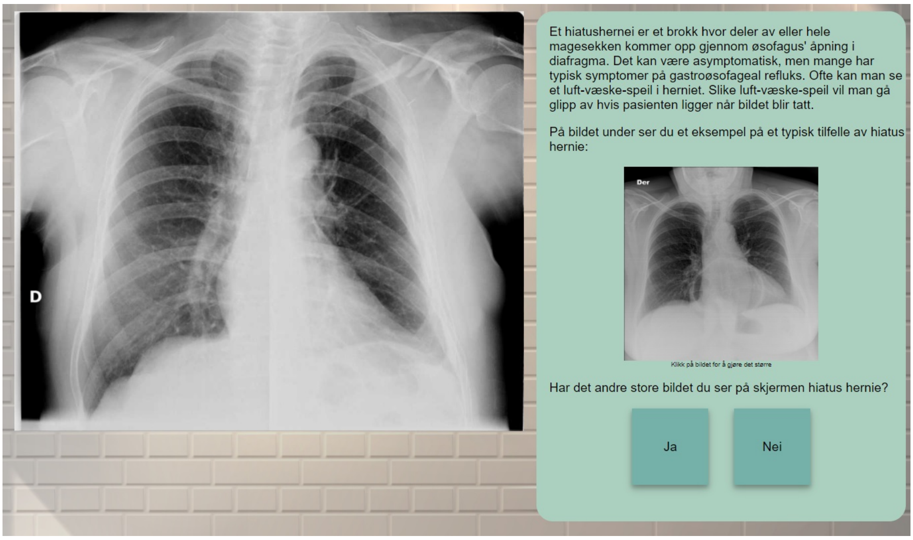
- numberOfMisses refers to how many times the user drags the wrong image to the garbage bin.

- numberOfCorrectImages refers to how many of the 6 images that have the category in focus.

3.5.3 Follow-up Task Type 2

Description

As with all the follow-up tasks, follow-up task type 2 only focuses on one category at the time. It presents the user with a text that explains the category in focus. Then it presents an image that shows a typical case for the category. The intent here is that the text and the image should teach the user how to analyze an image in order to detect the specific category. Then the user gets to try out what he/she has learned from the explanation, through analyzing a second image, and determining if it has the previously explained category or not. There is a 50 % chance that the image has the category. Since the example image is so small, there is functionality to make it bigger by clicking on it.


Et hiatushernei er et brokk hvor deler av eller hele magesekken kommer opp gjennom øsofagus' åpning i diafragma. Det kan være asymptomatisk, men mange har typisk symptomer på gastroøsofageal refluks. Ofte kan man se et luft-væske-speil i herniet. Slike luft-væske-speil vil man gå glipp av hvis pasienten ligger når bildet blir tatt.

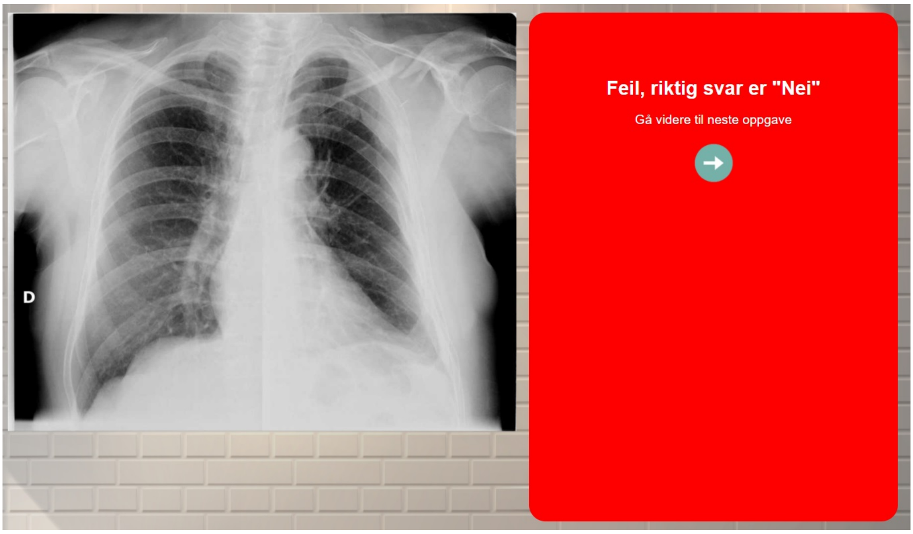
På bildet under ser du et eksempel på et typisk tilfelle av hiatus hernie:

Klikk på bildet for å gjøre det større

Har det andre store bildet du ser på skjermen hiatus hernie?

Figure 3.17: Follow-up task type 2 first shows an explanatory text and an example image for one category. Then it asks if the large image on the left side has the category.

After the user has answered, the task will show feedback to the user by telling if the answer was correct or wrong. The user can then click a next-button, which either takes him/her to the next follow-up task, or to a new main task.



Feil, riktig svar er "Nei"

Gå videre til neste oppgave

Figure 3.18: When the user has answered, the task will show feedback to the user. Red means wrong answer, green means correct answer.

Applied principles and theories

This task type is designed according to a problem-solving philosophy (see section 2.2.2). This means that it first gives general instructions for how to solve a problem, followed by a specific problem where the knowledge from the instructions can be applied. The instructions it gives were made by the domain expert, who was asked to write an explanatory text for each of the 14 categories, focusing on what the category was and how one could detect it on an image. The domain expert also picked out images showing obvious cases for each category.

This task type also tries to mimic the effect of adaptive feedback (see section 2.1.2). As mentioned earlier in section 3.2, the amount of learning material and time available made it difficult to develop proper adaptive feedback. However the instructions the task gives function as feedback, since the instructions are triggered by a mistake the user has done in the main task. Each time the user was given follow-up task type 2 as a response to his/her mistake, the task acted as feedback trying to teach the user how to detect the category he/she had missed. Unfortunately the task was only able to use the same instructions every time, since it only had one set of instructions per category. This meant that for example each time the user failed on the category ”hyperinflation” on different images, the same instructions would be shown every time. To be called proper adaptive feedback, the feedback instructions should have been specifically written for the image that the user failed on, instead of a general instruction text for the category. But one could say that the task type mimics an adaptive feedback approach, and could be valuable in order to establish if adaptive feedback is the right adaptive learning approach for teaching chest X-ray interpretation.

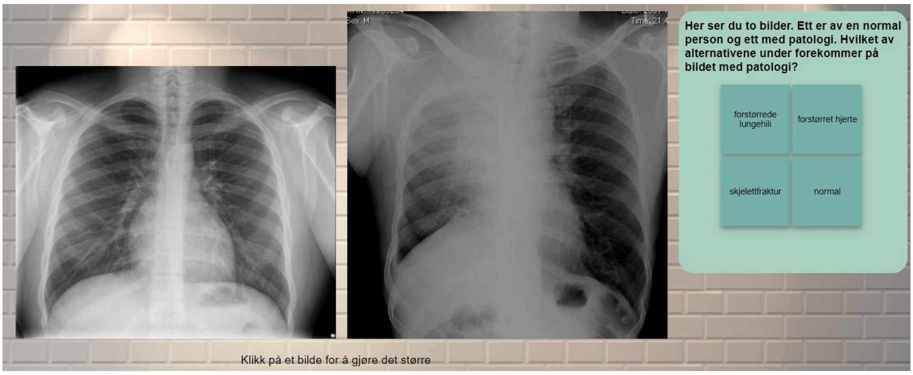
Effect on adaptive learning algorithm

A challenge with the different follow-up task types is that one task type might be generally easier than another. Follow-up task type 2 for instance, gives a 50 % chance of correct answer just by guessing. This could cause task type 2 to get higher priority by the algorithm than the other task types, and all the users end up getting only follow-up tasks of type 2. This is not a wanted situation, since the idea behind the adaptive task type algorithm is that the task type that fits best to a user is given most often to that user. The differences in difficulty through guessing must therefore be leveled out by how the task affects the stored TTPM-values. For follow-up task type 2 this is done by setting the reward for correct answer to have the same absolute value as the punishment. It should thereby be leveled out if the user just guesses. The value chosen was the probability of guessing correct, which is 0.5. So if the user answers correctly, the stored TTPM-value (value that affects adaptive task type) and the stored CPM-value (value that affects adaptive case selection for main task) is increased by 0.5. If the user answers wrong, the same stored values are decreased by 0.5 instead. Through some simple simulations by answering randomly on tasks, it was discovered that this way for the task to affect the stored values, together with the ways the other follow-up tasks affected the values, leveled out the differences in difficulty when the user answered randomly. However it was difficult to predict how this could change when the tasks were to be given to students with pre-knowledge in analyzing X-ray images.

3.5.4 Follow-up Task Type 3

Description

Follow-up task type 3 presents the user with four categories and two images. One of the images is of a normal person, and the second image has one of the four categories. The user has to pick the correct category for the second image. The reason for the normal image, is that the user should be able to compare the image where there is something wrong up against an image where there is nothing wrong, and through discovering the differences, the user might be able to make a correct decision about which of the four categories it is. The correct category is always one that the user has answered wrong in the last main task. The task type has functionality for making the images larger by clicking on them.

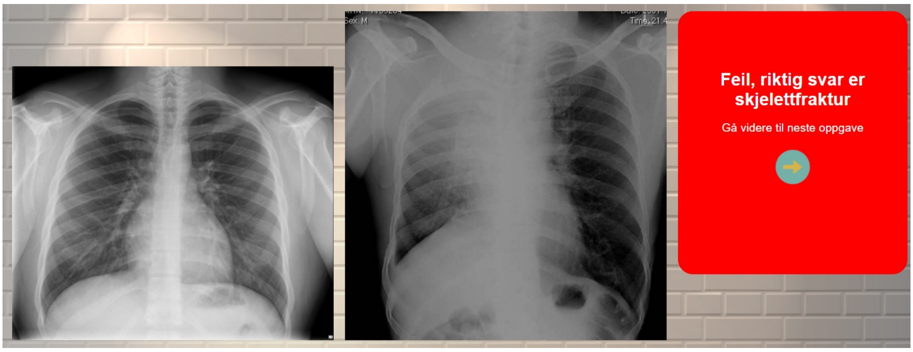


Her ser du to bilder. Ett er av en normal person og ett med patologi. Hvilket av alternativene under forekommer på bildet med patologi?

Klikk på et bilde for å gjøre det større

Figure 3.19: Follow-up task type 3 presents two images, one normal and one with pathology. The user has to select the correct category for the image with pathology.

When the user has chosen one of the four categories, the system gives feedback to the user. Red color means wrong answer, and at the same time the user is told which answer was the correct one. Green color means correct answer. The user can then, when ready, click on a next-button that takes him/her to the next follow-up task, or a new main task.



Feil, riktig svar er skjelettfraktur

Gå videre til neste oppgave

Figure 3.20: After selecting a category, the user is told if the answer was correct. If the user answers wrong, the system tells what the correct answer was.

Applied principles and theories

This task type is designed based on a drill and practice philosophy (see section 2.2.2). This means that it presents a problem that should be solved, and gives simple feedback to the user based on the solution the user provided.

Also here the focus is on comparing images, but unlike follow-up task type 1 (see section 3.5.2) the user is not able to experiment and compare against previous correct methods and wrong methods. The idea of this task type came after a discussion with the domain expert, who stated that distinguishing a normal image from an image with pathology is an important and basic ability to have. The task focuses on teaching the user to see how one category makes an image look different from a normal case. Also, by answering wrong the user might learn to distinguish the correct category from those three others.

Effect on adaptive learning algorithm

As with follow-up task type 2, this task also had some chance of guessing the correct answer, and also if the user managed to see through the workings of the system, he/she would detect that only categories that he/she answered incorrectly in the main task can be correct answers to follow-up task type 3. The user could from this knowledge of the system easily rule out some of the four alternatives given, and thereby greatly increase the probability of guessing correct. It was found that if the task rewarded and punished with the probability of guessing correct, which is 0.25. The balance between the task types was approximately maintained in the case when a user just gives random answers. As with follow-up task type 1 and 2, type 3 also affected both the stored TTPM-values for the user, and the stored CPM-values for the user for the category the task focused on.

Chapter 4

Implementation

The system described in chapter 3 was implemented through an iterative development process. The architecture, data model and other elements of the system were constantly changing as a consequence of insightful conversations with the domain expert. The domain expert also tested early versions of the system. The feedback on these early versions resulted in ideas being thrown away, and new ideas being formed. The discussions with the domain expert also resulted in the discovery of new requirements for the system. Each time a new functionality or component in the system was implemented, it was unit-tested right away after implementation. When all units in a new iteration were implemented and unit-tested, they were integration-tested before more feedback was obtained from the domain expert. The last and final version of the system was tested on real students of medicine, as will be described in chapter 5.

4.1 Chosen Technologies

The following were the most central technologies and tools that were used in the implementation:

- **React**[1]
  In the implementation of the frontend part of the system, the React library was used. React is a JavaScript library that makes it possible to make declarative user interface components. A component can be any type of element found in a user interface. The special thing about React is that with the component-concept one can make a general description of a GUI-element, containing both information about how the element should appear and also logic describing how the element should react to different events. A React-component can be described as a class definition for a GUI-element. It is defined by input parameters, it has internal methods which are triggered by different events, and it can contain other react components. A React-component also has an internal state which can be altered based on events or what

[1] https://reactjs.org/ Accessed: 24.05.2018

inputs the component is given. When the user interface is shown, all its components are first mounted by setting their state based on initial input parameters. After the mounting they are rendered. The rendering is the process where HTML and CSS code describing the appearance of the component is read in order to convert it to a visual representation. React and JavaScript were chosen since they can run in a web-browser, thereby making it easy to develop a system that can be run on many different devices and operating systems. The needs of the system that had been planned also fitted very well with the component structure that React introduces to the implementation.

- **Node.js[2] and Express[3]**
  Node.js is an open-source JavaScript runtime environment that executes JavaScript code server-side. It has the package system npm, which currently is the largest ecosystem of open source libraries in the world. One of these libraries is the Express-framework. Express is a ligthweight framework for developing Node.js applications. Node.js and Express were used for developing an API which could be used by the React frontend to request data and computations. Through investigations, Express was found to be both robust and easy to work with, and was therefore chosen for the task of developing the API.

- **MySQL[4]**
  MySQL is an open-source relational database management system. It is a vital part of the LAMP open-source web application software stack, and is used by many well known companies and applications. MySQL was used in this project for developing a database, where all the necessary data for the system was stored. This included user-data, case-data and data used by the adaptive learning algorithm. In the development of the database, the visual tool MySQL Workbench 6.3[5] was used to make the work of defining and filling the database easier.

4.2 Architecture

The architecture of the system is implemented following a client-server architecture pattern, and is described by figures 4.1 and 4.2. On the client a React application is running in the user's web browser. This application is however just a shell, and does not do any computations related to the adaptive learning algorithm. The React application only serves as a GUI that displays the task data given to it. It receives the data it needs by sending requests to the server through a REST API made available by the Node Express application running on the server. This means that all computations are done by adaptive learning algorithms on the server, and then the results of these computations are returned to the client. More details about the API will be described in section 4.4.

[2] https://nodejs.org/en/ Accessed: 24.05.2018

[3] https://expressjs.com/ Accessed: 24.05.2018

[4] https://www.mysql.com/ Accessed: 24.05.2018

[5] https://www.mysql.com/products/workbench/ Accessed: 31.05.2018

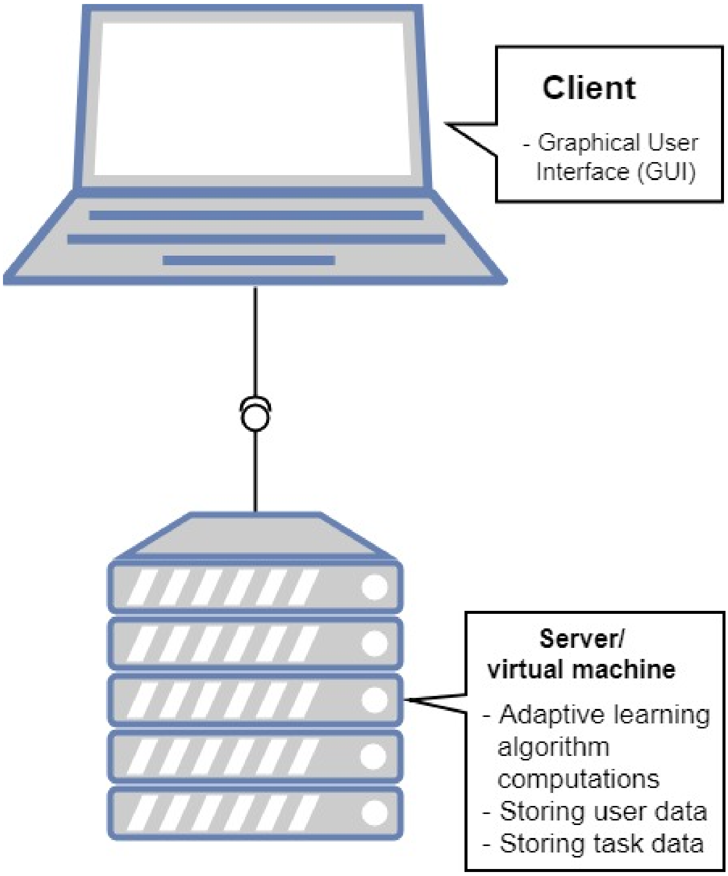

Client
- Graphical User Interface (GUI)
Server/ virtual machine
- Adaptive learning algorithm computations
- Storing user data
- Storing task data

Figure 4.1: In the client a React web application is running, which takes care of all GUI related functions. This React application is of course hosted on a web server, which is not shown in this diagram. The server this application sends REST-calls to takes care of all data storage and algorithm computations needed to produce the data displayed in the client application.

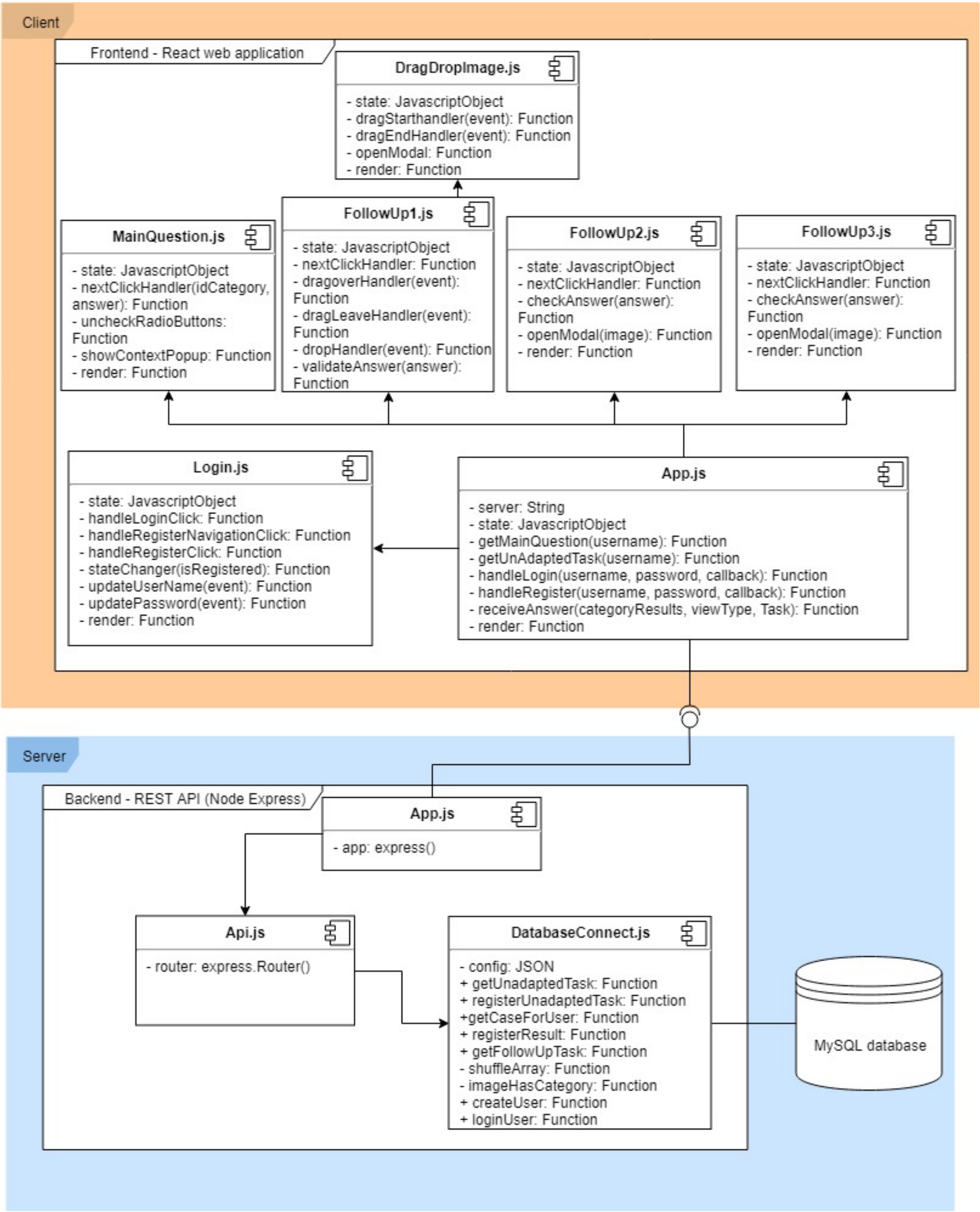

Client
Frontend - React web application
DragDropImage.js
- state: JavascriptObject
- dragStarthandler(event): Function
- dragEndHandler(event): Function
- openModal: Function
- render: Function
MainQuestion.js
- state: JavascriptObject
- nextClickHandler(idCategory, answer): Function
- uncheckRadioButtons: Function
- showContextPopup: Function
- render: Function
FollowUp1.js
- state: JavascriptObject
- nextClickHandler: Function
- dragoverHandler(event): Function
- dragLeaveHandler(event): Function
- dropHandler(event): Function
- validateAnswer(answer): Function
FollowUp2.js
- state: JavascriptObject
- nextClickHandler: Function
- checkAnswer(answer): Function
- openModal(image): Function
- render: Function
FollowUp3.js
- state: JavascriptObject
- nextClickHandler: Function
- checkAnswer(answer): Function
- openModal(image): Function
- render: Function
Login.js
- state: JavascriptObject
- handleLoginClick: Function
- handleRegisterNavigationClick: Function
- handleRegisterClick: Function
- stateChanger(isRegistered): Function
- updateUserName(event): Function
- updatePassword(event): Function
- render: Function
App.js
- server: String
- state: JavascriptObject
- getMainQuestion(username): Function
- getUnAdaptedTask(username): Function
- handleLogin(username, password, callback): Function
- handleRegister(username, password, callback): Function
- receiveAnswer(categoryResults, viewType, Task): Function
- render: Function
Server
Backend - REST API (Node Express)
App.js
- app: express()
Api.js
- router: express.Router()
DatabaseConnect.js
- config: JSON
+ getUnadaptedTask: Function
+ registerUnadaptedTask: Function
+getCaseForUser: Function
+ registerResult: Function
+ getFollowUpTask: Function
- shuffleArray: Function
- imageHasCategory: Function
+ createUser: Function
+ loginUser: Function
MySQL database

Figure 4.2: The diagram gives an overview of the system architecture. There is a main separation between the server and the client. On the server both a Node Express application and a database is running. On the client a React application is running.

The most important advantages of using this client-server architecture are:

- Modularity: The clear separation between computational components and user interface components gives a system with high cohesion and low coupling, making it easy to do changes to one component without causing problems in other parts of the system.

- Modifiability: By making the adaptive learning computations available through the REST API, and not implementing them directly into the react application, it makes it easy to make changes to the user interface, and also opens up for creating a completely different user interface, which can use the exact same methods made available by the REST API.

- Performance: All the algorithm-related computations are done on the server, making it easy to scale for possibly more users by just increasing the server resources. If the computations had been done on the client instead, large amounts of data would have had to be sent from the server to the client. All the rendering is done on the client, thereby relieving the server of having to do rendering for every user. Rendering the user interface for a user is a task that has a constant need for resources, therefore it is not affected by a rise in number of users unlike the algorithm computations.

4.2.1 Client

The client is implemented as a React application. As can be seen in figure 4.2 this application consists of a set of components. Each React component always has a JavaScript object describing the component state and a render-function describing what the component should render. What the render-function should render is often decided based on the component state or input given to the component. These inputs are called ”props”. The components of this specific application can be divided into three different groups. The app-component, the login-component and the task components.

App.js

The app-component is the main component of the application. It takes care of all the REST-calls to the server and decides which of the other components that should be rendered. The getMainQuestion-function is called each time there is a need for a new main task to be requested from the server. The receiveAnswer-function gets the results from the last main task or follow-up task and registers these results on the server, and after that decides what should be done next. The getUnadaptedTask-function is a function used in the testing of the system where one control group was not given adapted tasks, but random tasks. The function does the same as getMainQuestion, except that the mainQuestion it gets is randomly picked and not adapted to the user. The handleRegister-function and the handleLogin-function are related to the creation and login of users. The actual authentication of a user is done on the server. The app-component just sends the username and password to the server and gets a reply. The same goes for the creation of new users. The actual creation is done on the server, the app-component just forwards the user data to the server. What the app component renders is decided by the state, and the state is manipulated by all the functions mentioned. In the rendering the app component always renders one of the other React components shown in figure 4.2.

Login.js

The login-component is rendered by the app-component if the app-component does not detect that there is a user logged in. The login-component has functions and user interfaces for getting data needed to both create a user and log a user in. That means that the app-component renders the login-component in order to get the necessary data for the handleRegister-function or the handleLogin-function. When a user is logged in, the state of the app-component contains the username. The app-component will then know that there is a user logged in, and will render task components instead of the login-component.

Task components

The task components are components representing the different task types described in section 3.5. There is one component for each task, except for follow-up task type 1, which has an extra component (DragDropImage.js) it uses to render each of the six draggable images. The task components have functions for handling button-clicks, typically a nextClickHandler-function, which is called when a task is finished and the results should be sent to the app-component. Those tasks that have functionality for making images larger have a function called ”openModal”, which renders an image as a large overlay covering the entire screen.

4.2.2 Server

The server is implemented as a Node Express application. This application should make some specific services available to the client through a REST API. The server-application consists of three components. The App.js-component is the main component which sets up the application and listens for requests on a specific port. The Api.js-component is used by App.js and defines routes in the REST API. This means that it defines the functions behind each of the possible REST-paths. Each path is defined by saying what type of REST call it is (GET or POST), which inputs the function needs, the actual function definition and what the function should return. In this specific server application the Api.js was set to just forward the requests to a new component called DatabaseConnect.js. This component has functionality for connecting to the database, which runs on the same physical server as the Node Express application. For each path in Api.js, there is a function in DatabaseConnect.js made available. These functions represent all the logic operations needed to implement the adaptive learning algorithm, create users and authenticate users. All these operations rely on multiple queries to the database.

The getCaseForUser-function represents all logic operations related to the adaptive case selection described in section 3.4.2. The getFollowUpTask-function represents all logic operations related to the adaptive task type selection described in section 3.4.4. The registerResult-function updates the CPM- and TTPM-values stored in the database based on results received from the client. The getUnadaptedTask-function and registerUnadaptedTask-function are related to the testing of the system, where one control group was going to just get random tasks instead of adapted tasks. The createUser-function evaluates new user credentials and stores them in the database. The loginUser-function is called by a client in order to authenticate a user. The remaining functions are helper-functions used by the functions already mentioned.

4.3 Data Model

Figure 4.3 describes all the tables found in the MySQL database running on the server. The model was developed through an iterative process, where new tables and attributes were added as new needs arose after consulting with the domain expert and after early testing of the system.

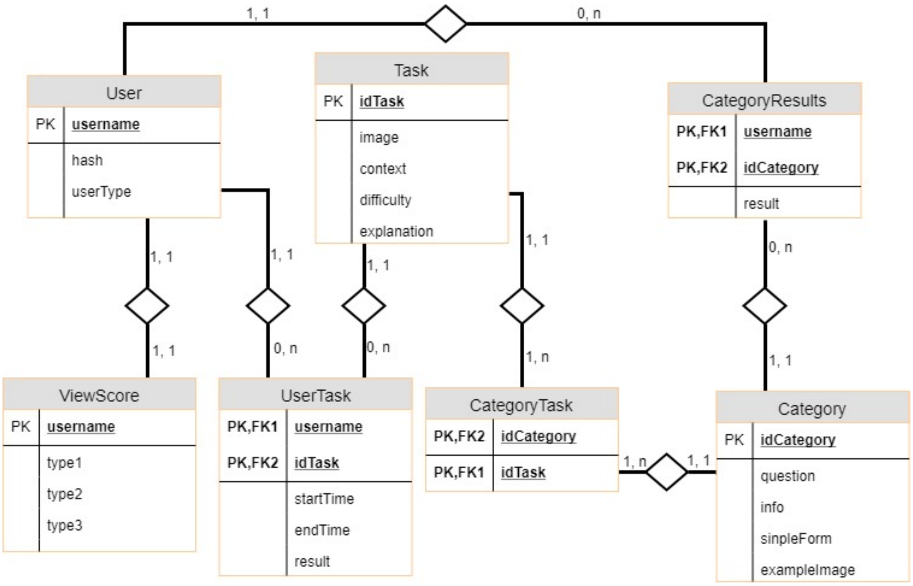


Figure 4.3

4.3.1 The Tables Explained

User

The User-table represents a user of the system. It holds the attributes username, hash and userType. Username and hash are used to authenticate a user when logging in. The userType-attribute says what type of user it is. This field could be used to distinguish users from each other and thereby give users different functionality in the system. A typical example is the distinction between admin and normal users. For this project this field was only used to distinguish between standard users and control users related to the experiment conducted on the system. Those users that had userType ”control” were not given adaptive functionality like the users having userType ”standard”.

Task

The Task-table is a constant table which is not changed by the user’s interactions with the system. A row in the table represents one case, as described in section 3.4.1. Each case has a unique identifier-attribute, idTask. The image-attribute contains the filename for the X-ray image connected to the case. The context-attribute refers to the clinical context information for the case, while the explanation-attribute holds the comment given by the domain expert about the case. The difficulty-attribute holds a value between 1 and 3 for how difficult the case is.

Category

The category-table is a constant table that always holds the same data. It does not get new rows, or the data in it does not get changed. The data in the table represents information about all the 14 possible categories a case can have. Each category is uniqly identified by the idCategory-attribute. The question-attribute is the name of the category. The info-attribute represents the explanatory text about the category made by the domain expert. This text was used by follow-up task type 2, as described in section 3.5.3. The task type also presented an example image for the category. The filename for this image was stored under the exampleImage-attribute. The simpleForm-attribute was used to store a simpler version of the name of the category. For example, the question-attribute stored a long question-phrase like ”Are the airways shifted (trachea, carina)?”. In some follow-up tasks however there was a need to refer to a category with only one or two words, so in those cases the name stored in the simpleForm-attribute was used instead.

CategoryTask

The CategoryTask-table holds information about which categories a case has or not. It is a constant table, and is therefore not modified or altered as a consequence of the user’s interactions with the system.

CategoryResults

The CategoryResults-table holds the CPM-values. CPM is explained in section 3.4.2. In this table each user, identified by the username-attribute, has one row for each category, identified by the idCategory-attribute. This means that each user has 14 rows stored in this table. For each row the result-attribute holds the user’s current score for that category. This value is, as mentioned earlier called Category Performance Measure (CPM).

ViewScore

The ViewScore-table holds the TTPM-values. TTPM is explained in section 3.4.2. In this table each user has one row. A username-attribute uniqly identifies which row belongs to which user. The three attributes ”type1”, ”type2” and ”type3” represent each of the three possible follow-up task types as described in section 3.4.4. Each hold a score for how well the user has performed in follow-up tasks of that specific type. As mentioned earlier, these values are called Task Type Performance Measures (TTPM).

UserTask

The UserTask-table holds information about which cases a user has been given in the past. This to avoid that the system gives the same case multiple times to the same user. Each row in the table represents one case, identified by the idTask-attribute, taken by one specific user, identified by the username-attribute. The remaining attributes in the table were used in relation to the experiment conducted on the system. For each time a user finished a task, the task start time and end time were saved. Also how many of the categories the user answered correctly in the case was saved. This information was valuable in order to evaluate if users improved after some time using the system.

4.4 REST API

All the adaptive learning functions of the system were made available as services through a REpresentational State Transfer Application Programming Interface (REST API). Functionality for user authentication and user creation was also made available through the API. In the following tables the different services of the API will be described.

Table 4.1: Register task results

Title	Register task results
URL	/api/registerResults
Method	POST

Input parameters	{
	”user”: ”fred”,
	”task”: 25,
	”categoryResults”:[{”idCategory”: 5, ”points”: 2}]
	”viewType”: ”type2”,
	”startTime”: ”2018-04-07 12:53:43”,
	”endTime”: ”2018-04-07 12:59:43”
	}

Response format	**Code:** 200 **Response:** {”registered”: true}
Comment	This service is called in order to update the stored CPM-values and TTPM-values. The results are given as input by the client, and based on these results, all the stored values for the user in the database are updated such that future adaptations can be more accurate.

Table 4.2: Get case for user

Title	Get case for user
URL	/api/getCaseForUser
Method	POST

Input parameters	{
	”user”: ”fred”
	}

Response format	**Code:** 200 **Response:** { ”idTask”: ”34”, ”difficulty”: 1, ”image”: ”img40.jpg”, ”context”: ”Mann 31, kortpustethet”, ”explanation”: ”Right tension pneumothorax with complete collapse of the right lung, over expansion of the right hemithorax, depression of the right hemidiaphragm and mediastinal shift to the left. ”, ”categories”: [{ ”idCategory”: 1, ”question”: ”Pneumothorax?”, ”correct”: 1 }, { ”idCategory”: 2, ”question”: ”Lungefortetninger, (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)?”, ”correct”: 1 }, { ”idCategory”: 3, ”question”: ”Pleuravæske (ensidig, bilateral)?”, ”correct”: 0 }, { ”idCategory”: 4, ”question”: ”Hyperinflasjon?”, ”correct”: 0 }, { ”idCategory”: 5, ”question”: ”Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)?”, ”correct”: 1 }, { ”idCategory”: 6, ”question”: ”Lungehilii forstørret (bilateralt eller ensidig)?”, ”correct”: 0 }, { ”idCategory”: 7, ”question”: ”Hjerte forstørret?”, ”correct”: 0 }, { ”idCategory”: 8, ”question”: ”Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)?”, ”correct”: 0 }, { ”idCategory”: 9, ”question”: ”Diagframa uskarpt avgrenset eller avflatet?”, ”correct”: 1 }, { ”idCategory”: 10, ”question”: ”Fri luft under diafragma?”, ”correct”: 0 }, { ”idCategory”: 11, ”question”: ”Hiatus hernie?”, ”correct”: 0 }, { ”idCategory”: 12, ”question”: ”Luftveier er forskjøvet (trachea, carina)?”, ”correct”: 1 }, { ”idCategory”: 13, ”question”: ”Okkluderte hovedbronkier?”, ”correct”: 0 }, { ”idCategory”: 14, ”question”: ”Normal?”, ”correct”: 0 }] }
Comment	Which task-ID that is chosen is decided with the adaptive case selection, as described in section 3.4.2. The example response above describes all information needed to construct a main task.

Table 4.3: Get follow-up tasks

Title	Get follow-up tasks
URL	/api/getFolowUpTask
Method	POST

Input parameters	{
	”user”: ”fred”,
	”categories”: [2, 4, 7]
	}

Response format	**Code:** 200
	Response: [
	{
	”idCategory”: 2,
	”viewType”: ”type2”,
	”image”: ”img13.jpg”,
	”info”: ”Lungefortetninger kan være enhver økt tetthet i lungevevet. Det er en således en uspesifikk betegnelse som bør spesifiseres. Nodulus eller knute er en 3-30 mm fortetning. Oppfylning er over 30 mm. Diffuse fortetninger kan ikke avgrenses på samme måte som nodulus og oppfylning, men oppstår f.eks. ved interstitielt ødem. Kerley-linjer oppstår når interlobære septa blir ødematøse.”,
	”exampleImage”: ”img54.jpg”,
	”category”: ”lungefortetninger”,
	”correct”: 1
	},
	{
	”idCategory”: 4,
	”viewType”: ”type3”,
	”alternatives”: [”hyperinflasjon”, ”forstørret hjerte”, ”unormalt mediastinum”, ”forstørrede lungehili”],
	”normalImage”: ”img9.jpg”,
	”pathologyImage”: ”img113.jpg”,
	”correct”: 0
	},
	{
	”idCategory”: 7,
	”viewType”: ”type1”,
	”images”: [{ ”filename”: ”img76.jpg”, ”isCorrect”: 1 },
	{ ”filename”: ”img25.jpg”, ”isCorrect”: 0 },
	{ ”filename”: ”img47.jpg”, ”isCorrect”: 1 },
	{ ”filename”: ”img13.jpg”, ”isCorrect”: 0 },
	{ ”filename”: ”img64.jpg”, ”isCorrect”: 1 },
	{ ”filename”: ”img113.jpg”, ”isCorrect”: 0 }],
	”correctCategory”: ”forstørret hjerte”
	}
	]

Comment	Note that which type each follow-up task gets is decided by the TTPM-values. Since the user in the response example above has the same TTPM-value for each task type, the pick of task type in this example is equivalent with a random pick of task type. The example above contains all the possible three task types. Note also that how the ”categories”-paramter is specified decides how many follow-up tasks that should be returned, and which categories these should have.

Table 4.4: Register user

Title	Register user
URL	/api/registerUser
Method	POST

Input parameters	{
	”username”: ”fred”,
	”password”: ”fredspassword1234”
	}

Response format	**Code:** 200
	Response: {
	”success”: true,
	”message”: ”User successfully created”
	}

Comment	This service is used when a new user is to be created. The service will validate the username and password. If the username does not exist already, the user will be created in the database.

Table 4.5: Authenticate user

Title	Authenticate user
URL	/api/loginUser
Method	POST

Input parameters	{
	"username": "fred",
	"password": "fredspassword1234"
	}

Response format	**Code:** 200
	Response: {
	"success": true,
	"message": "User successfully created",
	"token": sdfhj32iueh&jf2#4iuhr%sdefo$jin",
	"userType": "standard"
	}

Comment	This service is called to authenticate a user. The username and password will be checked up against the stored usernames and hashes in the database.

The system also has two additional services. The first of these is "/api/registerUnadaptedTask", which is a simpler version of the service described in table 4.1. The difference is that this service ignores updating the TTPM-values and CPM-values, and only just marks a task as taken by the user. The second of these additional services is "/api/getUndadaptedTask", which is a simpler version of the service described in table 4.2. Both of these were included in order to be able to have the experiment where two different groups tested the system. One control group and one standard group. Those in the control group were only given random tasks instead of adapted tasks, and in order to give such random tasks, these two functions had to be included. More details about this experiment will be given in chapter 5.

4.5 Security

The system that was implemented was implemented to be a test system that should be used primarily only for controlled user testing sessions. Even so it was expressed by the domain expert that if the system proved to be a good solution for the students, it could be taken in as part of the teaching of medicine students. The system does not store any sensitive data about users or anything else important, so there is not much that motivates an attacker to break into the system, but even so, any deployed system should at least have a minimum of security functions, even though the risks of an attack are low. In the architecture, considerations to security were therefore taken. Password hashing was implemented by using the package bcrypt[6] found in the npm library. This package also includes a functionality for automatic salting when passwords are hashed.

The remaining planned security functions in the architecture were not implemented though. The reason for this was to minimize risk of failures during the testing sessions, by minimizing the number of functions that could introduce errors that could ruin the testing session. Since the security functions were not necessary for the testing session, they were not prioritized for implementation. The planned security functions that should be implemented before the system is deployed:

- Password policy: When creating new users the passwords should be required to have a minimum length, contain capitalization, numbers and regular letters. This to ensure resistance against brute force attacks.

- Authentication for every method in the REST API: People that have not created a user account and logged in should not be able to get response on their calls to the API. It was planned that all API calls were to require a valid token for authentication. This token should only be valid for a restricted time period. The JSON Web Tokens[7] is a suitable solution for implementing this functionality.

- Securing the REST API against DoS attacks: There is no restriction for how many times the same API call can be made in a row. This might make the system vulnerable to a Denial of Service attack where the attacker continuously requests data from the server, and thereby locks other users out by taking up most of the server's resources. This can be avoided by using the Node Express middleware called express-brute[8], which is available through the package library npm.

- Also when the system is deployed it should use HTTPS so that communication between the server and the client is encrypted. This to avoid a man-in-the-middle attack where the password gets stolen.

[6] https://www.npmjs.com/package/bcrypt Accessed: 30.02.2018

[7] https://jwt.io/ Accessed: 24.05.2018

[8] https://www.npmjs.com/package/express-brute Accessed: 24.05.2018

4.6 Testing

The client and the server were tested separately from each other, and when both the client and the server had been tested individually, the integration between them was tested by running the system as it would be used by the users. A more thorough system- and acceptance test was done later on real users through user testing sessions, as described in chapter 5.

4.6.1 Testing of the React Client Application

The React application was set up using create-react-app[9], which is found in the package library npm[10]. Through create-react-app a useful testing environment was automatically configured, so that the application could be run locally, and changes could be viewed and tested instantly.

The testing of the application was first done through unit-testing, where each component was tested separately, by giving it different inputs to see if the input was rendered correctly. Key focus in the testing here was to make sure that each component could handle both large amounts of data and small amounts of data to render. An important part of the testing was also to check the rendering of the component in different window-sizes and web-browsers, in order to make sure that the component would be compatible with as many devices as possible. Since no computational functions were present in the React application, there was no need for extensive testing of each function in each component. These functions were more relevant for integration testing, since they were mostly related to navigation and transfer of data between different components.

When all components had been individually tested, and were found to render in a satisfactory way, integration testing to test the interaction between them was initiated. This testing was basically to test all possible navigation paths possible in the application, and make sure that every button navigated to the right place at the right time, that data from the server was requested at the right times, and that rendering of a component was not done before all data was received from the server. To rule out possible test failures as a consequence of errors on the server, the server was set to return dummy data.

4.6.2 Testing of the Node Express Server Application

In the testing of the Node Express application the components that were automatically generated by the Express setup were not tested. The focus was primarily on the functions created in the DatabaseConnect-component.

Unit-testing was conducted on each of the functions in the DatabaseConnect-component. Since the application is a REST API that should mostly just update data stored in the database, it should be flexible in which input values it sets as illegal, since different clients

[9] https://www.npmjs.com/package/create-react-app Accessed: 27.05.2018

[10] https://www.npmjs.com/ Accessed: 24.05.2018

could have different ways of using the same services provided by the REST API. The only clear illegal values are those where one requests data using a username that does not exist, or if one try to create a user that already exists. For all functions it was established that giving wrong username as input had no negative effects, and creating a user with a username that already existed was not possible. The most important part of the unit testing of each function was to establish that the return-format was correct according to what the clients expect. When a service is called it goes through a set of steps, where each step either generates updates in the database or generates temporary calculated values. By monitoring the updates in the database and the temporary values calculated by each step, it was possible to detect if the function did the intended steps correctly or not, and to quickly identify which step, and thereby in which part of the code the error was located.

4.7 Deployment

For deployment the institute of computer science at NTNU made a virtual machine available. This was used to deploy both the React client application and the Node Express server application. The virtual machine was running Ubuntu version 16.04.3. It was necessary to deploy the system in order to be able to conduct the user testing sessions described in chapter 5.

4.7.1 Deploying the React Client Application

Since the client application had been configured using the "create-react-application"-package a finished build script was available, and by running the command "npm run build" a deployable build-repository was generated. On the virtual machine a web-server was set up to run on port 80. This was done using the open-source HTTP web server software NGINX[11]. NGINX was chosen since it was easy to find good documentation and tutorials for how to configure and setup the web-server correctly. The web-server was configured to point to the generated build-folder. The virtual machine was now hosting the application, and by pointing their browser to the URL of the virtual machine, the users were able to get access to the application, as long as they were connected to the university network.

4.7.2 Deploying the Node Express Server Application

In order for the client-application to be able to get any data, the Node express application and the database also had to be deployed on the virtual machine. For the database setup MySQL was installed on the virtual machine, and the creation-script for the database was run in order to generate all the database tables. The Node Express application (the REST API) was deployed using a process manager software called PM2[12]. PM2 keeps the application as a running process, monitors it, and restarts it if some kind of failure causes the process to shut down. By using PM2 the system gets better availability, since the

[11] https://www.nginx.com/resources/wiki/ Accessed: 24.05.2018

[12] http://pm2.keymetrics.io/ Accessed: 24.05.2018

REST API is always up and ready to receive requests from the clients. Both the Node Express application and the MySQL server were set to run on their own separate ports. The Node Express application had to refer to the port of the MySQL database when updating the database, and all requests from the clients had to use the correct port when sending requests to the REST API.

Chapter 5

Evaluating the System through User Testing

The system presented in chapter 3 and chapter 4 can be seen as a hypothesis for one possible way to answer the research questions described in section 1.3. In order to test this hyphotesis, some kind of experiment or test was needed in order to evaluate if the chosen methods and designs actually were good answers to the research questions or not. At the same time there was a need to also test how good usability the system had and generally how relevant and helpful the system would be for the students. The goal for the evaluation was to see if the applied adaptive learning methods worked as intended, and that they managed to create a positive learning effect compared to a non-adaptive alternative. Another goal was to see how well the user preferences matched the adaptation the system had done. In order to test the system, the following actions were taken:

- Two user testing sessions were held, where in total 33 students of medicine attended and were observed as they tested the system.
- A survey was issued to the students asking for feedback on the system.
- User data describing user actions in the system were collected and analyzed.

5.1 Testing the Prototype System on Students

In order to evaluate the system, two sessions of user testing were planned. With help from the domain expert, invitations were sent out to the most relevant students. All the students who had replied on the invitation were invited to join a user testing session. These sessions were held at the laboratory center in auditorium LA21 at campus Øya, NTNU. When participants arrived, they were given a note with instructions on what to do. By only following the instructions the participants were able to log into the system and start solving tasks. The students were observed as they were working, and interesting observations about their use of the system were noted down. The observation was done by sitting in the back of the room, so it was possible to see all the computer screens. The participants were allowed to test the system for three hours, but could leave whenever they wanted to. However they were encouraged to sit at least one hour. It was noted down each time a participant left the room.

In the handed out instructions each participant was given a username and a password. The reason for pre-creating user accounts was that the participants were to be divided into two groups. One test group and one control group. The participants in the test group were given full access to every functionality the system had. Participants in the control group were given an un-adaptive version of the same system as participants in the test group. This meant that the follow-up tasks were disabled, and the main tasks were given randomly instead of adaptively. The feedback-functionality called ”The radiologist’s description of the image” was also disabled for the control group. The intention of this group division was to see if there was a notable difference between the group that was given adaptive functions and the group that did not get adaptive functions.

Table 5.1: Functionality available for the different groups of the experiment

	Test group	**Control group**
Main task	Yes	Yes
Correct answer confirmation on all tasks	Yes	Yes
Adaptive follow-ups	Yes	No
Adaptive case selection	Yes	No
Adaptive task type	Yes	No
The radiologists description of the image as feedback	Yes	No

5.2 Collecting User Opinions through a Survey

The most effective way to get participants’ opinions about the system was to create a survey. The survey consisted of questions from the categories usability, adaptive learning, system acceptance and system feedback. Each question was created with an underlying intention of revealing possible areas of improvement for the system. Many questions also were focused directly on issues that could help to answer the research questions. The survey was made through Google Forms[1], and it was made clear to the participants that

[1] https://www.google.com/forms/about/ Accessed: 24.05.2018

the survey was anonymous. All questions were given in norwegian, but are translated into english for this report (See appendix 1 for original norwegian question texts). Note that most questions had alternatives on a scale ranging from negative to positive, similar to a Likert scale[2]. Some questions were free-text-questions where the user could formulate their own answer. In order to make sure every participant took the survey, it was clearly mentioned as one of the instructions on the note the participants were given before entering the room.

5.2.1 System Acceptance and other Questions

This category of questions focused on getting information about the participants' general opinion about the system and other information like the participants' pre-knowledge. The category had questions primarily focusing on determining overall success of the concept without the focus on adaptive learning.

Questions about the system acceptance and their underlying intentions:

- **Would you have used this system even if it was not a mandatory learning activity?**: The intention of this question was to see if the users found the system helpful and educational. If most users answer negatively on this question, the system is not useful and the whole concept should be reevaluated.
- **Would you have preferred to have this system as part of the mandatory learning activities?**: A student is often very busy and making an activity mandatory force more work upon the student. So if a student answers yes on this question, it is a good indicator that the system is very useful for the student, since students do not want more mandatory activities unless they feel the activity is very useful.
- **In total, how would you rank your experience of the system on a 1-10 scale, where 10 is best?**: The intention of this question was to measure the general acceptance of the entire system and to see if there was a difference between the two groups.
- **Which description do you feel fits best to the system you have now tested?**: The intention of this question was to determine if the participants perceived the system as a test-system that only tested the knowledge without teaching it, or if they perceived it as a system trying to teach them new knowledge. The ideal positive answer should be that they felt that it was a system that tried to teach them new knowledge, since this was the intention of the system.
- **How was your pre-knowledge in analyzing thorax X-ray images?**: The intention of this question is to get an overview of the pre-knowledge of the participants, and to see if one of the two groups accidentally got more participants with higher pre-knowledge than the other.

[2] https://snl.no/Likert-skala Accessed: 24.05.2018

- **What was your username when you tested the system?**: This question is very important. By getting the participant's username, it is possible to compare his/her answers in the survey with the data that was stored in the database when he/she used the system. More details about this will be given in section 5.3. It is important to say that the usernames the participants had were pre-created anonymous usernames. It was impossible to identify a person from the username or any of the other stored data. All participants also gave their consent that their answers could be used in this report.

5.2.2 Usability

This category of questions consisted of four questions meant to detect how user friendly the system was. It was important to establish the level of usability both for possible future work and improvements, but also to rule this out as a factor that could have affected the other questions negatively if the usability proved to be bad.

Questions about usability and their underlying intentions:

- **Do you think the system was user friendly and easy to use?**: Here the user is asked directly about the usability of the entire system. This will give an impression of the participants' overall satisfaction.
- **What do you think about the size of the images?**: The system presents many images to the users. Some bigger than others. The domain expert had some concerns about the images being too small. Therefore this question was included in the survey.
- **Did you experience any technical issues during your testing of the system?**: The intention of this question was to measure the implementation quality. The other reason for this question is to be able to rule out technical issues as a reason for negative response in later questions. (given that this question gets mostly positive responses)
- **Do you otherwise have anything else you want to mention about the system usability?**: This question was a free-text question and was maybe the most important question for usability. Here participants could give feedback on anything they missed in the system and could go into detail about why they did not like certain parts of the design. The information from this question could be very valuable for future improvements on the system.

5.2.3 The System's Ability to Adapt to the User

This category was the most important category of questions for gathering information that could help answering the research questions. It consisted of 14 questions about the tasks the user had been solving. Some tasks were irrelevant for those participating in the control group, but others were specifically focused on detecting differences between the test group and the control group.

Questions about the system's ability to adapt and their underlying intentions:

- **How did you perceive the difficulty of the tasks?**: The intention here was to see if those in the control group felt that tasks were much more easy than those in the test-group. The ideal answer would be that users found the tasks not too hard and not too easy.
- **Two questions: Did you often get tasks that were too easy/difficult?**: These two questions had the intention of testing to see if the adaptive case selection, described in section 3.4.2 managed to create a difference in perceived difficulty between the two groups.
- **Did you find the content of the tasks relevant for what you needed to be better at?**: The intention of this question was to see if the participants in the test-group felt that they got more relevant tasks than those in the control group.
- **Did you feel that the system tried to adapt the tasks while you used the system?**: The intention of this question was to see if users felt that the system adapted to them. The discovery of a clear difference between the two groups on this question would be important for the evaluation of the system, since participants in the control group should ideally give much more negative answers than the participants in the test group. Alternatives here were either ”No, tasks seemed random”, ”Yes, some tasks felt adapted based on my previous mistakes” and ”Yes, the tasks felt customized for me”.
- **Four questions of the type: Did you find this task educational?**: For each of the four different task types in the system the participants were shown an image describing the task type and then to answer how educational they felt the task type was. Note that for for these four questions there was an alternative for answering that they did not get the task type at all. Those participating in the control group used that alternative on those questions, since they did not get follow-up tasks. The intention of these questions was to see if the task design managed to be educational, and to compare the different task types up against each other. Another intention was to be able to establish if the adaptive task type algorithm, as described in section 3.4.4, managed to identify the most educational task type for a user.
- **Which of the task types did you like best?**: The participants were shown images describing each of the three possible follow-up tasks and was asked to pick the one they liked best. This question was only meant for participants in the test-group. The intention was to see if the adaptive task type algorithm, as described in section 3.4.4, managed to identify the task the user liked best by comparing the survey answers with automatically collected data from the system.
- **Which of the task types did you get most often?**: The participants were shown images describing each of the three follow-up tasks, and were asked to pick the one that they felt that they had been given most often by the system. This question was only meant for participants in the test-group. The intention was to see if the adaptive task type algorithm, as described in section 3.4.4, managed to deliver the task type

the user liked the most or found most educational more often than the other task types.

- **Was it possible to guess the correct answer on tasks?**: It was a concern that the way the adaptive follow-ups were generated could make the questions more easily guessable. To determine if that was the case, this question was included.

- **Did you find the clinical context hint helpful for finding the correct answers?**: The intention of this question was to identify how helpful the clinical context was for the users.

- **Do you otherwise have anything you want to add about the tasks?**: This question was a free-text question giving the participants an opportunity to give feedback on anything else they felt could be better concerning the tasks. A nice way to get tips for future work and improvements, should someone choose to continue the development of the system.

5.2.4 System Feedback

In total two user testing sessions were held. After the first session it was discovered that many of the participants missed better feedback from the system. It was not enough to only know what they answered wrong, but also get customized feedback on what they had done wrong. Unfortunately it was not enough time to create such feedback functions, but the functionality ”The radiologist’s description of the image” was added for the next session. Also more questions about the system’s feedback to the user were added in the survey in order to investigate how feedback could be improved.

Questions added after the first user testing session and their underlying intentions:

- **After finishing a task the system usually gave you some feedback. How helpful did you find this feedback?**: This question intended to measure how satisfied the participants generally were with the feedback-functionality of the system.

- **On some tasks you got a message called ”The radiologist’s description of the image”. How educational were these messages?**: The intention of this question was to see how helpful the new feedback functionality added since last time actually was.

- **A potential improvement of the system is that the system gives feedback by marking the aberrations on the image itself. How much better would this have made the system?**: An idea that was too extensive to be done in this project, due to much work for the domain experts, was to mark the X-ray images with lines showing where the different aberrations/categories could be detected in the image, as shown in figure 5.1. Even though it was not a functionality that was part of the system, this question was included in the survey to check if it could have been a good idea and something that should be done in the future.

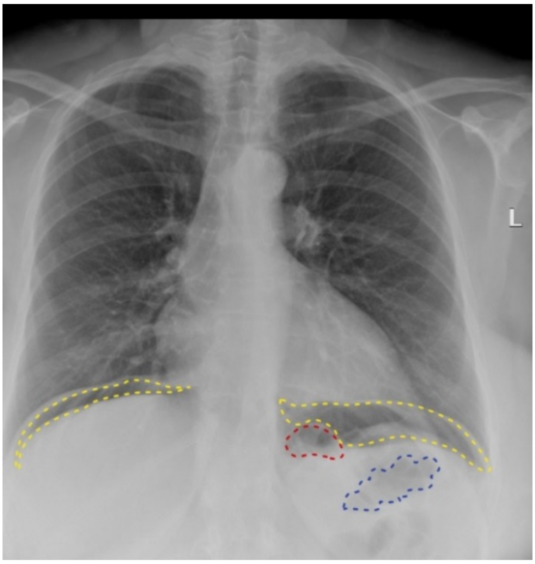


Figure 5.1: Here all the aberrations in the image are marked with lines. This could possibly be very good feedback for the students

- **Do you otherwise have anything to add about the system's feedback to you?**: This question was a free-text question and was meant to be an opportunity for the participants to mention anything they missed or thought of, concerning the feedback from the system.

5.3 Automatic Collection of User Performance Data

As mentioned briefly earlier, the system stores data about the user activity in order to have information that can say something about how well the system managed to adapt to the user. These data are data like time spent on each task and the task results. Also the stored measures that are used by the algorithm, CPM and TTPM, can have value in drawing a picture of how well the system managed to adapt to each user. The best way to measure the effect of the adaptive learning techniques would of course have been to actually test the participants' knowledge before and after the user testing session. This was not possible due to the extensiveness and need for external experts in order to create and evaluate such a test. Therefore the analysis of automatically collected data was seen as the next best way to say something about the effect of the adaptive learning techniques.

5.3.1 Task Time and User Results

Every time a user started a new main task the start time of the task was recorded. When the user finished the same task, a new timestamp for end time was recorded. With these two values it was possible to calculate how much time the user spent on a task. The intention with storing this information was to see if there would be a difference in how much time a participant in the user testing spent on tasks in the beginning compared to the end of the session. If participants spent less time on each task in the end of the session compared to the beginning, this could be an indicator that they had become better at analyzing the images, and thereby needed less time on each image. If the task time is found to be drastically reduced however, this could instead maybe say something about the participant's motivation or concentration. The results for a task were also recorded. The results were how many of the categories in the main task that were answered correctly. The intention with this information was also to see if there would be an improvement in the end of the session compared to the beginning of the session. Note that these time- and result data were only recorded for the main task, not the follow-up tasks.

5.3.2 Measures Used by the Adaptive Learning Algorithm

A Category Performance Measure (CPM) is a value stored for the user which says how well the user has performed in one specific category. The user has 14 of these values stored, and they are constantly updated for each task the user finishes. By analysing the end-values of the stored CPMs one can get an impression of the user's performance on each of the categories, and to see if there actually is a difference between the categories or not. If all the categories have the same values, future adaptive learning approaches for chest X-ray interpretation should consider other alternatives.

The Task Type Performance Measures (TTPM) are three values stored for each user. The values tell which task type the user has performed best and worse on. By comparing the end-values for the stored TTPMs with the participants' answers on the survey, one can determine how well the system managed to identify which task types the user preferred. If the TTPM values do not match the answers the participants have given in the survey, another approach for adaptive task type should be considered.

Chapter 6

Results and Discussion

In total there were 50 students that replied on the invitations and were invited to one of the two user testing sessions. Of these 50 students, 33 actually showed up. The first session had 19 participants and the second had 14 participants. In both sessions there were participants in the control group and the test group. Since there were questions in the survey that only applied to the test group, it was important to fill this group first in the sessions. Since it was difficult to predict how many people who would show up, this resulted in a bit more people in the test group than in the control group. Figure 6.1 shows the distribution of participants between the control group and the test group.

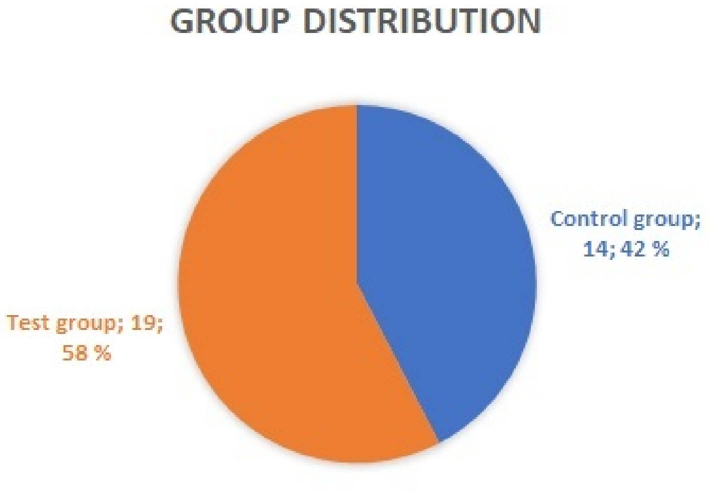


Figure 6.1: Group distribution

6.1 System Acceptance and Other Questions

6.1.1 Results

- **Would you have used this system even if it was not a mandatory learning activity?**
  All 33 participants (100 %) answered ”Yes” on this question.

- **Would you have preferred to have this system as part of the mandatory learning activities?**

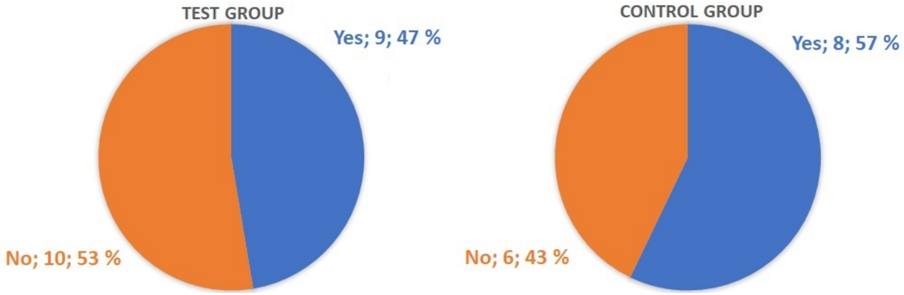


Figure 6.2: Results for system as preferred mandatory learning activity

- **In total, how would you rank your experience of the system on a 1-10 scale, where 10 is best?**

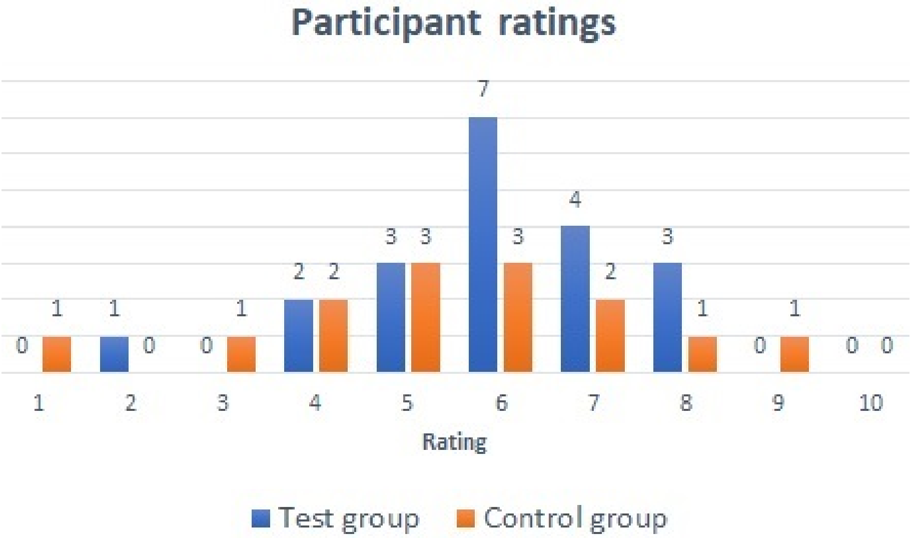


Figure 6.3: User ratings for both participant groups

- **Which description do you feel fits best to the system you have now tested?**

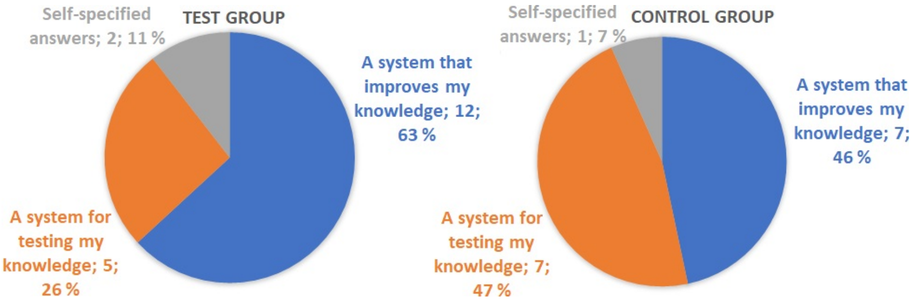


Figure 6.4: Results for system perception among the users. Those who self-specified their answer mostly answered that they perceived that the system fitted both of the two other alternatives at the same time.

- **How was your pre-knowledge in analyzing thorax X-ray images?**

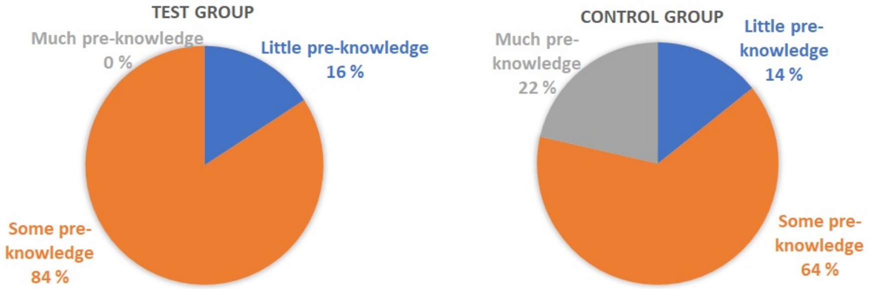


Figure 6.5: Results for participants' pre-knowledge

6.1.2 Discussing the System Acceptance Results

By going through the results it is evident that the system in total managed to achieve a relatively high level of acceptance among the participants. The strongest indicator of this is the results from the question ”Would you have used this system even if it was not a mandatory learning activity?”, where all participants said yes. This result however says more about their general opinion about having it as a possibility. Figure 6.2 can give a more accurate picture of their opinion about this exact system. It shows that around half of the participants would not have preferred this as a mandatory learning activity. This gives an indicator that this exact system still has some room for improvement, but at the same time it shows that roughly half of the participants accepted the system as good enough to compete with their current learning activities. Figure 6.3 shows that the system managed to get an average rating close to 6 for both participant groups, where the ratings were a little bit higher for the test group, but not notably higher. This again shows a general high acceptance among the participants, at the same time as it indicates room for improvement.

The two other questions in this section were not directly linked to system acceptance. The first one, described in figure 6.4, had the intention of seeing if the system achieved its goal of being a learning system, and not a test system. From these results, it can be seen that participants in the test group to a much greater extent felt that the system tried to improve their knowledge instead of just testing them. It looks like the adaptive functionality contributed to creating a learning experience instead of just a test experience. The last question was about pre-knowledge in the two participant groups. It can be seen that the pre-knowledge was a bit higher in the control group, which is nice to have in mind when analyzing the rest of the results.

6.2 Usability

6.2.1 Results

- **Do you think the system was user friendly and easy to use?**

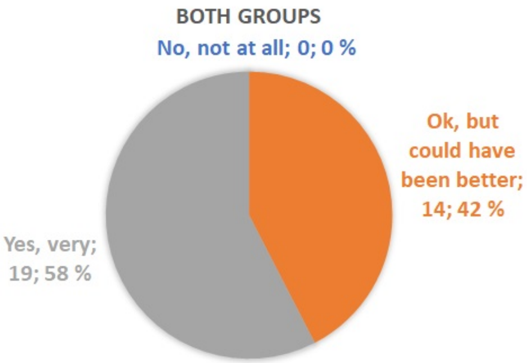


Figure 6.6: Results for usability

- **What do you think about the size of the images?**

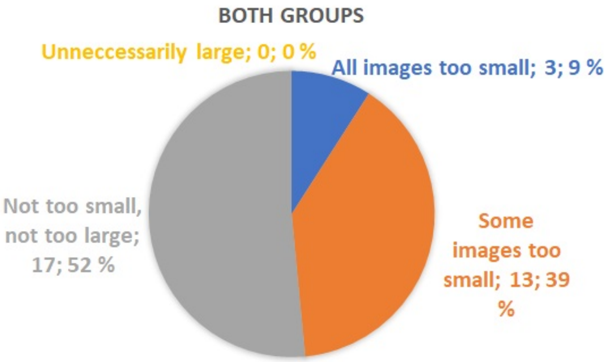


Figure 6.7: Results for image size

- **Did you experience any technical issues during your testing of the system?**

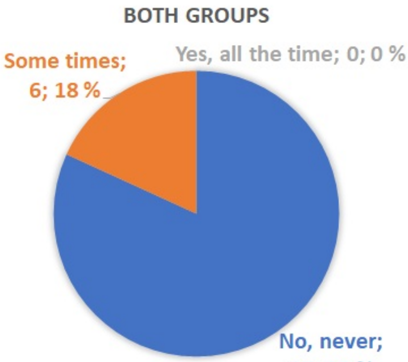


Figure 6.8: Results for technical issues

- **Do you otherwise have anything else you want to mention about the system usability?**
  On this question participants could write whatever they wanted concerning the system usability. The following sums up what participants were missing:
  - Be able to navigate back to previous tasks.
  - Tutorials explaining the user interface.
  - Possibility to zoom on images. The functionality for making the images larger, which was present in the system, should be on every image, not some of them as was the case.

- Stressful to have to answer yes/no on every category in the main task.
- Present both frontal- and side image for every case (The system only presented frontal X-ray images).
- Follow-up task type 1 should not remove images when they are thrown in the bin, but instead mark them as correct or incorrect.

6.2.2 Discussing the Usability Results

Based on the results shown in figure 6.6 the system was by the participants considered to be user friendly. Some felt that there was room for improvement, but the majority thought it was very user friendly. When it comes to the image size, figure 6.7 shows that most people found the image size OK, but as the individual free-text answers also revealed, all images should be possible to make larger, not just the smallest ones. The free-text answers also revealed many good suggestions for improvement. Many that, if there had been more time, would have been excellent to include in the implementation. The intended requirement that users had to answer yes/no on every question, as described in section 3.5, and not just yes on those that were relevant, seemed to have a negative effect on the perceived usability of the system, based on the free-text answers. Figure 6.8 shows that a large majority of the participants had no technical issues, thereby indicating good implementation quality. It is worth mentioning that most of the ”Some times” answers came on the second session. That day the network connection felt a bit slower than usual, and may have had an effect on the results.

6.3 The System’s Ability to Adapt to the User

6.3.1 Results

- **How did you perceive the difficulty of the tasks?**

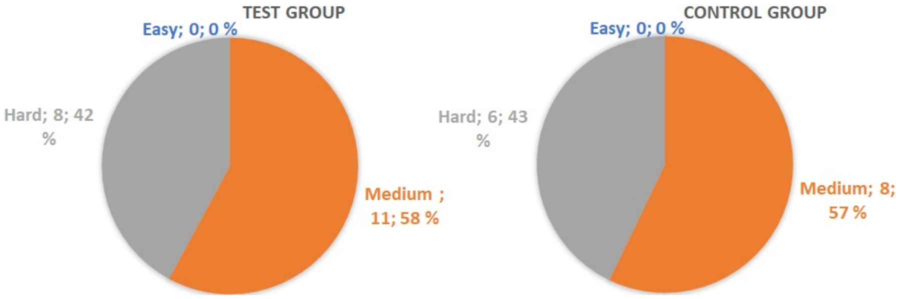


Figure 6.9: Results for perceived task difficulty

- **Did you often get tasks that were too difficult?**

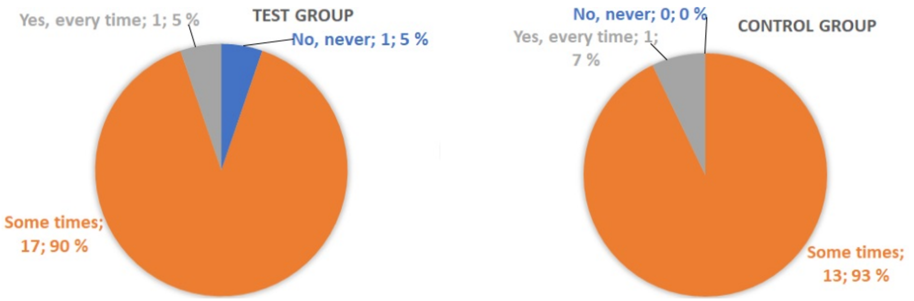


Figure 6.10: Results for how often participants got very difficult tasks

- **Did you often get tasks that were too easy?**

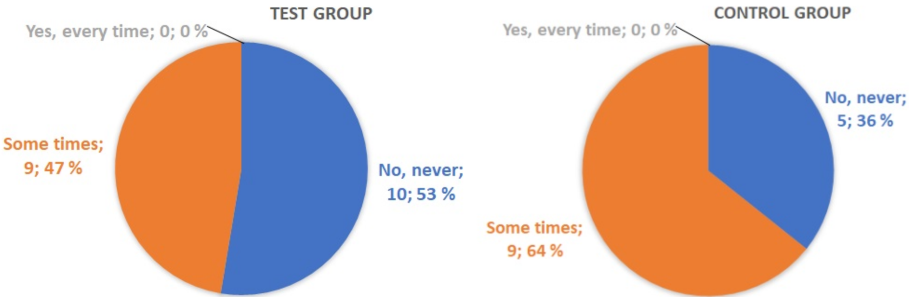


Figure 6.11: Results for how often participants got very easy tasks

- **Did you find the content of the tasks relevant for what you needed to be better at?**

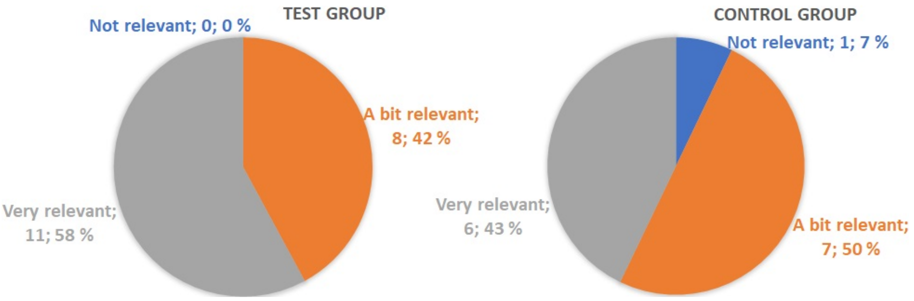


Figure 6.12: Results for perceived task relevancy

- **Did you feel that the system tried to adapt the tasks while you used the system?**

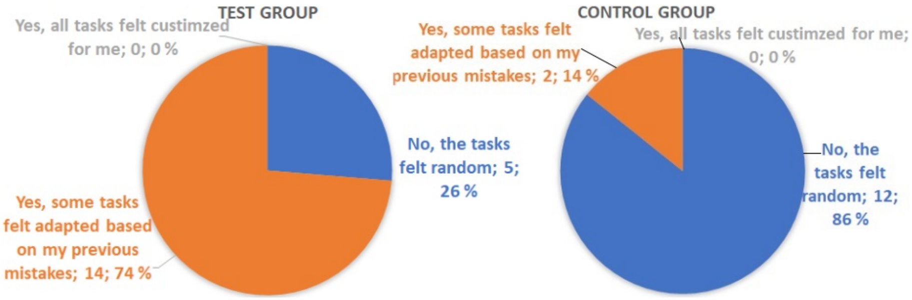


Figure 6.13: Results for how adapted the participants felt the tasks were

- **Four questions of the type: Did you find this task educational?**

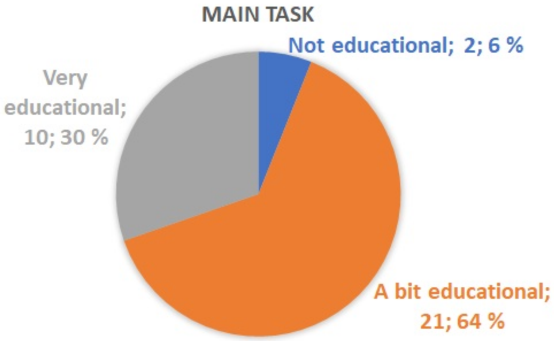


Figure 6.14: Results for how educational the main task was. This question was answered by both groups

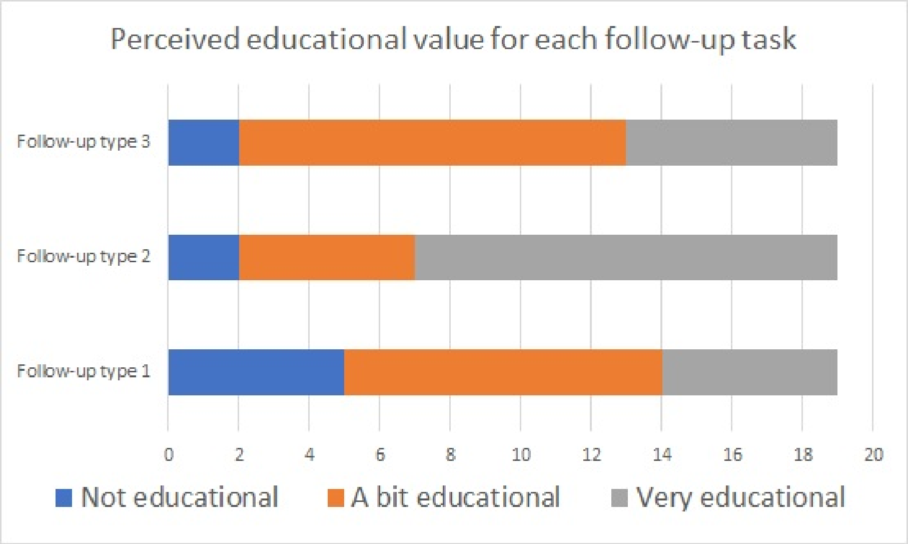


Figure 6.15: Results for how educational the follow-up tasks were. This question was only answered by participants in the test group.

- **Which of the task types did you like best?**

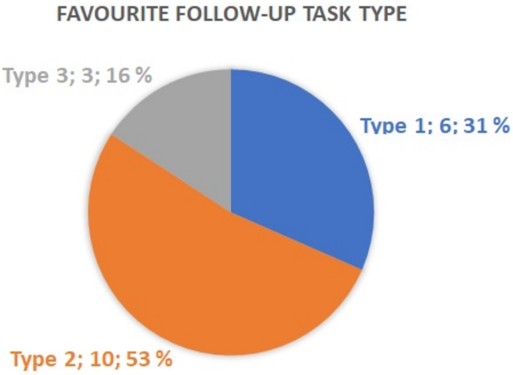


Figure 6.16: Results for favourite task type. This question was only answered by participants in the test group.

- **Which of the task types did you get most often?**

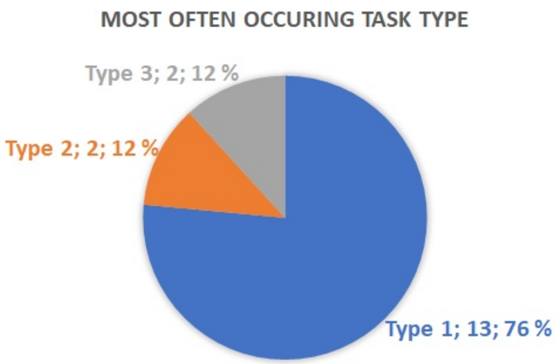


Figure 6.17: Results for most often occurring task type. This question was only answered by participants in the test group.

- **Was it possible to guess the correct answer on tasks?**

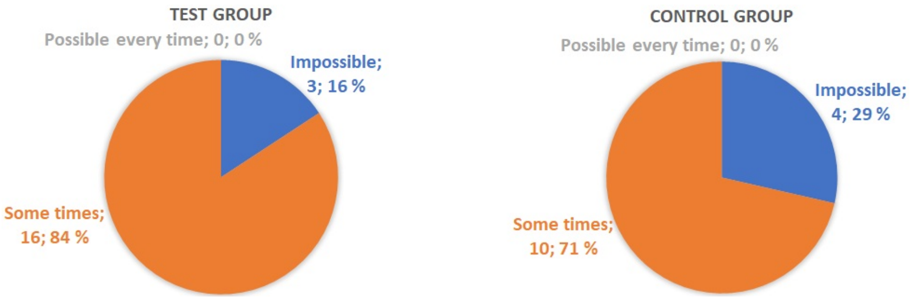


Figure 6.18: Results for how guessable participants found the tasks

- **Did you find the clinical context hint helpful for finding the correct answers?**

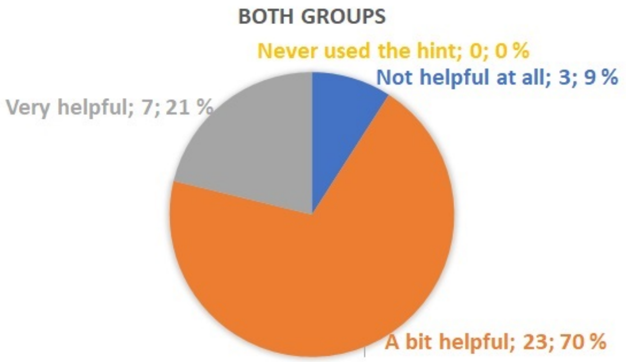


Figure 6.19: Results for clinical context helpfulness

- **Do you otherwise have anything you want to add about the tasks?** This question was a free-text question were the participants could answer anything they wanted related to the tasks. There were many answers here, where a large majority of them requested more detailed feedback on the tasks. The following sums up what participants were missing:
  - Some kind of explanation to why an answer was wrong, but also explanation for why an answer was correct. Many people suggested to use arrows and visually show on the image in order to explain answers.
  - Better and more detailed clinical context information.
  - Missing tasks about some typical kinds of pathology.

6.3.2 Discussing the Adaptive Learning Results

As can be seen in figures 6.9, 6.10 and 6.11 the tasks were perceived by a large fraction of the participants to be generally hard. This applies to both the test group and the control group, so the reason is not necessarily that the adaptive learning algorithm is bad. The reason is more likely that most cases in the learning material the system uses are to be considered as hard for the students. Unfortunately the adaptive case selection, as described in section 3.4.2, is built under the assumption that the task material has a balanced difficulty. Since most cases are considered to be hard for the students, as the results show, it will be hard for the algorithm to identify categories that the student needs practice in. This because the students will perform bad in all categories, as a consequence of the generally too high difficulty. Even so, it is expected that the algorithm will manage to detect the differences between the categories, since even though the user fails in all categories, it is likely that some categories will be failed more severely than others. Another possible solution could be to set the algorithm to prioritize cases that are easier in order to counteract the generally high difficulty. One area where it is possible to see an effect of the adaptive case selection algorithm is in filtering out too easy tasks, which can be seen in the differences between the participant groups in figure 6.11. Also figure 6.12 gives an indication of an effect, since the test group has a larger fraction saying that they got relevant tasks than the control group, but it is hard to prove that this is due to the adaptive case selection algorithm. The question is also a bit too open for interpretation by the participants. All this indicates that the adaptive case selection needs more adjustment and should have been created more tightly together with the task material.

Based on the results the adaptive follow-ups had more success. It is suspected that the very big difference between the participant groups in figure 6.13 is due to the follow-up tasks, which clearly adapted based on errors made in the main task. And since a large majority of participants in the test group answered ”Yes, some tasks felt adapted based on my previous mistakes” it is most likely the adaptive follow-ups that caused this perception among the participants. Also as figures 6.14 and 6.15 show, all task types were perceived as either a bit educational or very educational. One task type that stands out is follow-up task type 2. As figure 6.15 shows, more than half of the participants in the test group said that it was very educational, and figure 6.16 shows that follow-up task type 2 was the clear favourite of the three possible follow-up tasks. Since this task type had the intention of mimicking an adaptive feedback functionality, this could indicate that adaptive feedback is a strategy an adaptive learning system for teaching chest X-ray interpretation should focus on. This impression is also further strengthened through the free-text answers from the participants, where most participants requested more and better feedback from the system.

The fact that figure 6.16 shows that not everyone of the participants has the same favourite task shows that the idea of having adaptive task types, as described in section 3.4.4, might be a good idea. Unfortunately as figure 6.17 shows, follow-up task type 1 was perceived as the most often occurring task type, even though it was not the most popular or most educational task type. This could be due to the need for some adjustments in the adaptive task type algorithm, but it could also be due to the fact that task type 1 took much more time to solve than the two others. So even though the participants did not get task type 1 most often, they felt so since they spent most of their time solving tasks of type 1. There was a tendency that participants in the test group felt that it was easier to guess answers, as figure 6.18 shows. This was expected due to the fact that the follow-up tasks were generally more easily guessable than the main task, but the difference between the participant groups is not very large, so it is not possible to conclude anything for certain. When it comes to the clinical context we see that most people found it a bit helpful, but participants specifically mentioned the need to improve the clinical context in the free-text answers.

6.4 The System's Feedback to the User

In the second testing session some questions about system feedback were added to the survey. In total 14 participants attended this session, so these questions were only answered by them.

6.4.1 Results

- **After finishing a task the system usually gave you some feedback. How educational did you find this feedback?**

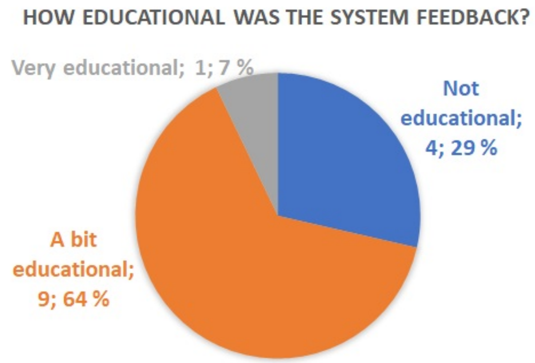


Figure 6.20: Results for how educational the system feedback was perceived

- **On some tasks you got a message called "The radiologist's description of the image". How educational were these messages?**

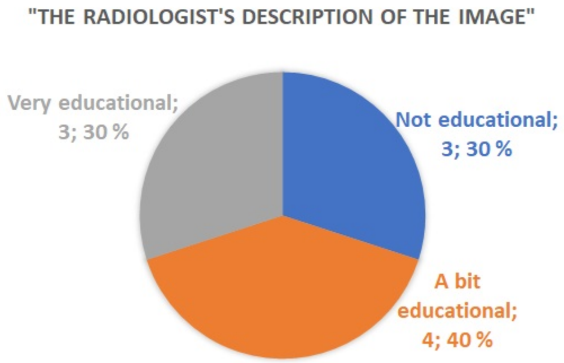


Figure 6.21: Results for how educational the radiologists's description was perceived. Note that this question was only relevant or the test group.

- **A potential improvement of the system is that the system gives feedback by marking the aberrations on the image itself. How much better would this have made the system?**
  All 14 participants that attended the second user testing session answered ”Much better” on this question.

6.4.2 Discussing the System Feedback Results

It is clear from the results in figure 6.20 that the system needs to come up with better ways to give feedback to the user. The new feedback ”The radiologist’s description of the image” did not create much more difference. As the results on the last question clearly show, a future improvement of the system should be to create adaptive feedback which marks the images to show where the different aberrations are on the image.

6.5 Task Time, User Results and Stored Measures

6.5.1 Results

By analyzing the automatically collected data in the database the following results were obtained:

Table 6.1: Results from automatically collected data

	Test group	**Control group**	**Comment**
Average amount of cases solved	23	43	This number is the average amount of main tasks a user managed to do during the three hours he/she had available. Note that this is expected to be higher for the control group, since they did not have to do follow-up tasks.
Average task duration, all three hours	149 sec	120 sec	Task duration is how much time a user spent to solve a main task.
Average task duration, first hour	173 sec	159 sec	Average task duration for tasks solved in the first hour of the sessions.
Average task duration, second hour	111 sec	108 sec	Average task duration for tasks solved in the second hour of the sessions.
Average task duration, third hour	196 sec	68 sec	Average task duration for tasks solved in the third hour of the sessions.
Average task score, all three hours	0.85	0.86	Fraction of all the 14 questions in the main task answered correctly. This will be a high value, since most questions would have the answer ”no”.
Average task score, first hour	0.85	0.84	Average task score for tasks solved in the first hour of the sessions.
Average task score, second hour	0.86	0.86	Average task score for tasks solved in the second hour of the sessions.
Average task score, third hour	0.88	0.87	Average task score for tasks solved in the third hour of the sessions.

By comparing answers from the survey with the stored TTPM values, the following results were obtained:

Table 6.2: Results from comparing the TTPM-values with the survey answers

	Ranked highest by the system	**Ranked second highest by the system**	**Lowest rank by the system**
Favourite task type	9 out of 18	4 out of 18	5 out of 18
Perceived as most occurring task type	10 out of 18	7 out of 18	1 out of 18

Table 6.2 shows how well the TTPM-values match the participants' preferences. It is desirable that when the system has given a task type high TTPM ranking, then that task type should be the task type the participant has answered is his/her favourite task type. When it comes to most occurring task type, it should be so that if a task type has high TTPM ranking, it should be perceived as the most occurring task type by the participant. If not, this could be an indicator of some implementation error. Note that these comparisons were only applicable for the participants in the test group, since only they got follow-up tasks. Note also that the total number here is 18 instead of 19 as previously. The reason for this is that one participant had entered wrong username in the survey, so a comparison was not possible for this participant.

All the results from the stored CPM-values are too extensive to present in a readable way, but just to get a picture of what kind of values one can expect, table 6.3 shows the CPM-results for two randomly selected users from the user testing sessions:

Table 6.3: CPM-results for usernames ”user47” and ”user35”

Category ID	**CPM-results for user47**	**CPM-results for user35**
1	1.50	1.00
2	-9.58	4.67
3	-6.00	-11.33
4	-2.25	-3.50
5	-21.09	-6.50
6	-0.25	-8.84
7	-7.83	-1.75
8	-3.25	-4.75
9	-16.09	-4.25
10	0.50	0.00
11	-2.25	-2.25
12	-7.83	-3.00
13	0.00	-13.00
14	-16.00	-3.00

6.5.2 Discussing the Results from the Stored User Data

From the data in table 6.1 we see that the average task duration changed differently for the two participant groups. For the test group there is no clear pattern, while for the control group the average task duration goes down for each hour passing. What the reason for this is, is hard to say. One could speculate that the continuous repetitive tasks the control group were given had a negative effect on their motivation, causing them to stop caring if they answered wrong or not. If that was to be the case, that would explain why the test group did not show this same tendency, since they got variation in task types due to the follow-up tasks. But there are many factors that could have affected the average task duration, so no certain conclusion should be drawn from these data. The average task score increased for both groups from the start of the session to the end. The increase was 3 % for both of the groups. Again one should be careful to conclude anything from this, since the difference is so small and many factors could have had affected the result, but at the same time it should be mentioned that the participants only used the system for 2-3 hours and the 3 % represents an improvement where the participants in average managed to get 0.5 (a half question) more of the 14 questions in each task correct.

When it comes to the results presented in table 6.2 they show that the adaptive task type algorithm performed well. For half of the participants it managed to give the highest

TTPM rating to the task type that the participant had said was his/her favourite task type. And since 4 out of 18 participants got their favourite task type rated second highest, the algorithm only completely missed in 5 out of 18 cases. Since the results are good, but not overwhelmingly good the experiment should ideally be repeated several times before a certain conclusion can be drawn. For the perception of most occurring task type the numbers are as expected, which means that the algorithm functions as it is intended.

For the CPM-results table 6.3 is a representative example for how results turned out for the participants. Most categories ended up getting a negative CPM-value, which indicates that the user is struggling with the category. The reason so many categories got negative values is the high difficulty of the learning material the system had available, as discussed previously in section 6.3.2. Even though most values were negative, there are still big differences between each of the categories, which indicates that some categories were more challenging for a user than others. Since there are differences between the category values, the adaptive case selection algorithm should have managed to work as intended. It is also positive to see that users scored very different on the same categories. For example user35 did very well on category 2, user47 on the other hand performed bad at category 2. Based on these results the adaptive case selection algorithm most likely gave user47 more cases containing category 2, while user35 did not get so many cases with category 2.

6.6 Observations During the Sessions

6.6.1 Results

During both the user testing sessions participants were observed as they tested the system. The observations were done by taking position in the back of the room so that it was possible to see everyone's screen. The following sums up the most interesting observations:

- Participants are spending more time on each task than anticipated.
- Dragging and dropping images in follow-up task type 1 looks a bit challenging for some.
- Clinical context is used every time, not just as a hint function as intended.
- A bug is discovered: Clinical context is not closed if the user answers everything correct. This is not good considering that the use of clinical context is set to affect the adaptive learning algorithm.
- The ”make image larger”-functionality is used very often.
- People with touch screens use the inbuilt zoom-function very often.
- Some people take screenshots of every task they do and write comments to the screenshots. This reduces how many tasks they have time to do.
- In the beginning some people treated the user testing session as a test-situation. After a while the same people realized that it was not to be considered as a test-situation.

- Some participants work in teams or discuss the tasks with others. This is not ideal, since the system is supposed to adapt to one single person, the success of the adaptation might be reduced if people help each other.
- A large majority of the participants are women.
- After the session participants asked if the system would be available outside the testing session. This is a positive indicator, since it shows that they actually found the system helpful.

It was also noted down each time a participant left the testing session. As can be seen in figure 6.22, most people spent two hours testing the system (the sessions started at 16:00 and ended 19:00). Note that the participants were encouraged to test the system for minimum one hour, but they had three hours available.

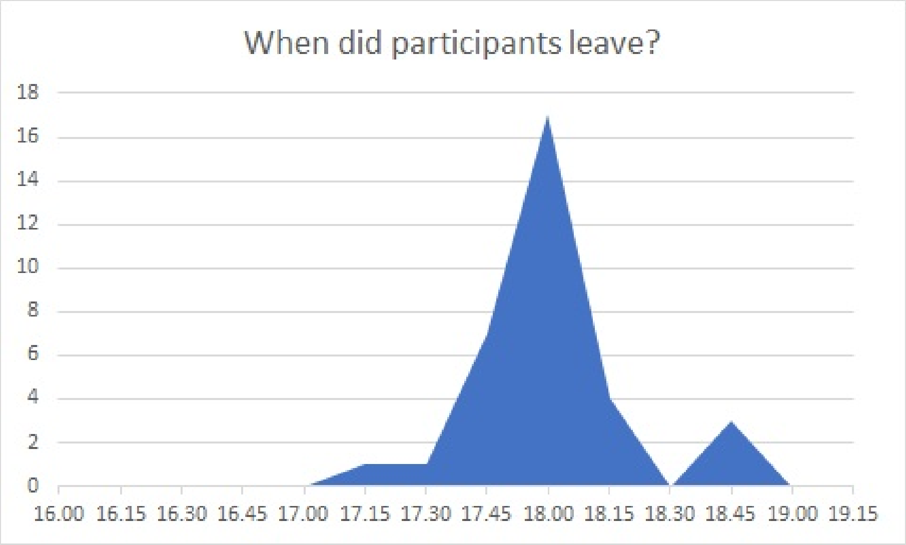


Figure 6.22: Results describing when people left the user testing session

6.6.2 Discussing the Results from the Observations

The most positive observation was that the participants took an interest in the system through asking if it was available to them after the session, and generally that they were focused and spent more time using the system than they were encouraged to do. Some observations one would assume did not have any effect on the experiment, but some might have contributed to affecting the results. Those participants that worked together instead of individually might have gotten tasks that were not individually adapted to them. This might have affected their answers negatively in the survey. Before the session started participants should have been informed that it was meant to be an individual task. The error that was discovered concerning the clinical context hint function might also have contributed to why the CPM-values described in table 6.3 became mostly negative, but the error did not occur that often, so it is assumed that it had little effect on the results. Since it was discovered that the clinical context was used all the time, and not just as a hint, it should not affect the adaptive learning algorithm's choices as it did. The fact that not everyone left at the same time might have had an effect on the average task duration and average task results presented in table 6.1.

Chapter 7

Conclusion

7.1 Summary

In this project three adaptive learning methods have been developed: adaptive case selection, adaptive follow-ups and adaptive task type. These methods were implemented together in a prototype for an adaptive learning system for teaching chest X-ray interpretation. As a consequence of the adaptive task type algorithm, the system contained different task types. Each task type had a unique design which applied a unique learning approach. The learning approaches that were mostly used by the tasks were drill-and-practice, problem-solving and image comparison techniques. Some tasks were also based on already established principles for teaching chest X-ray interpretation. One task type also tried to mimic an adaptive feedback approach. The evaluation of the system was made possible by data collected from three different activities: user testing sessions, user survey and automatic user data collection in the system itself.

7.2 Findings

Based on the results from the evaluation of the system, it was possible to draw the following conclutions:

- **How can adaptive learning techniques be applied in order to teach students how to interpret chest X-ray images? (RQ1)**

  In this project techniques for adaptive learning tailored for teaching chest X-ray interpretation were developed. These techniques were either adaptive content techniques or adaptive assessment techniques.

  The adaptive assessment technique that was developed managed to get a measure of user knowledge and give users tasks based on these measures. It proved to be very extensive to develop and was relying on many different factors in order to succeed.

From survey results it is hard to see a very clear effect of this algorithm. This might be due to the way the algorithm defined task difficulty based on categories. Also the case material it had available was a bit too much on the difficult side, which contributed to undermining the potential effect of the algorithm. Since the difficulty of task material for chest X-ray interpretation proved to be hard to predict and measure, and also since it was not possible to clearly verify any effect from the adaptive assessment algorithm that was used, adaptive assessment methods are not recommended as the first choice when an adaptive learning system for chest X-ray interpretation is to be developed.

Adaptive follow-ups and adaptive task types were the adaptive content methods that were developed. Results from the survey showed that these methods had potential. The survey also revealed that participants had different opinions about the different follow-up tasks, this shows that people learn differently, and supports the idea behind the adaptive task type algorithm. Having a variation in the tasks seemed to keep up the participants' motivation and made the system more engaging to use. The direct follow-up responses also made the users aware of the fact that the system adapted to them. The task type that mimicked adaptive feedback through problem-solving instructions to the user (follow-up task type 2), was the most popular task type. This together with free-text responses in the survey show that in teaching a student to analyze an X-ray image, giving detailed feedback to the user is the most effective way to promote learning. In the development of an adaptive learning system for chest X-ray interpretation one should therefore prioritize adaptive content techniques that analyze the user's answers and customize feedback to the user by explaining why the user answered wrong, but also why the user answered correct.

- **How can knowledge from learning theories and learning technology be applied in the design and presentation of task material for chest X-ray interpretation? (RQ2)**

  The approach that was applied in order to be able to answer this question was that each of the task types in the system applied different learning approaches. Each task type represented one learning theory or concept in learning technology, or a task type had a variation of the same approach as another task type. By looking at the results from the user testing survey, it was possible to see how participants responded to the different task types, and thereby how they responded to the learning approach the task type applied.

  Since the task type that applied a problem-solving philosophy was the most popular task type among the user testing participants, this could indicate that task design in chest X-ray interpretation should consider to base the design on the problem-solving philosophy (see section 2.2.2). The two other task types were primarily based on the drill and practice philosophy. The difference between those philosophies is primarily about drill and practice going straight to the problem, instead of giving instructions first. Since the task types are follow-up tasks they should not present new problems, but educational instructions instead. Even though the intention with the drill and practice tasks was to learn through solving a problem, this did not have as strong effect as the problem-solving philosophy had. The two other task types also primarily focused on applying comparison (see section 2.2.3) as a mean to promote learning. Even though sources [18] present comparison as a natural way to teach how to analyze the images, it seems that without direct textual instructions, comparing images does not have as strong learning effect as a problem-solving philosophy. But there were many participants that supported the two other tasks as well, and since everyone learns differently, a learning system for chest X-ray interpretation should embrace many different approaches. These findings support the idea behind the adaptive task type algorithm that was tested, and it is therefore recommended as a viable adaptive learning approach for teaching students how to analyze X-ray images.

- **How can the student's knowledge/abilities in chest X-ray interpretation be measured and modelled? (RQ3)**

  The measuring of a student's abilities in chest X-ray interpretation was primarily done through the adaptive case selection which measured knowledge by dividing the domain in 14 categories and giving a score (CPM-value) for each category. The set of CPM-values would therefore function as a model describing a student's knowledge. The model managed to show that there were differences between categories for each student, and that different students had different categories they struggled with. Unfortunately it was difficult to test if these results actually were accurate descriptions of the student knowledge. In order to answer that, students would have had to take a test in advance before they tested the system, which was not possible. There were also some issues with the model since the task material proved to have a higher difficulty than anticipated. All in all, the results are not sufficient enough to either confirm or debunk the suggested model as a viable method for measuring student knowledge in chest X-ray interpretation.

To summarize the most important findings:

- Adaptive content methods, and especially adaptive feedback, seem to be the preferred adaptive learning methods for teaching chest X-ray interpretation.

- A problem-solving approach which focuses on instructions combined with a problem to solve, that has both textual and visual feedback to the user, should be the basis for task design in the development of an adaptive learning system for chest X-ray interpretation, instead of a drill and practice approach.

- Measuring student knowledge in chest X-ray interpretation is challenging due to the challenges of establishing difficulty levels for task material, and may not necessarily be the most effective adaptive learning approach compared to an adaptive content approach.

7.3 Limitations

- **Limited knowledge about chest X-ray interpretation, and expert dependency:** The biggest challenge in this project was the lack of own expertise and knowledge about chest X-ray interpretation and how it should be taught. When coming up with ideas for how tasks or algorithms could be designed, it helps a lot when one already has knowledge about the subject that the system should teach. This was not the case in this project, and the development of ideas were always dependant on information provided by a third party in form of one or more domain experts. Also, adaptive learning often relies heavily on the availability of a large amount of learning material. All of this material had to be provided/created by the domain experts. This ensures quality material, of course, but at the same time limits the amount of material, since domain experts do not have unlimited amount of time, and producing good quality learning material is time consuming. Depending on experts, and not by one self being able to create the task material, therefore excludes those adaptive techniques that are extra content dependant. This also limits the possibility for experimenting with different ways of creating material and task designs. The experts were not easily available for physical meetings, which limited the communication to e-mail.

- **No or very few similar projects to get inspiration from:** When designing tasks for an adaptive learning system it helps to see how it has been done in other learning systems for radiology. Unfortunately searches for such learning systems or task material yielded little results. There seems to be a lack of learning resources for radiology, and thereby little previous experiences to learn from. This meant that more effort had to be put into actually identifying what kinds of general tasks that would work, and which tasks that would not work. This stole time that could have been used specifically for finding and developing the right adaptive learning technique.

7.4 Future Work

The results from the user testing sessions revealed that the prototype system achieved relatively high acceptance among the participants, and it was also perceived as very user friendly and stable. The system incorporated several approaches to both task design and adaptive learning. From the survey it is clear that some of these approaches had a more positive effect than others. Since the system itself managed to get so well accepted among the users, it is recommended to keep the system as a framework to build upon.

7.4.1 Recommended Future Adaptive Learning Improvements

The adaptive case selection algorithm (see section 3.4.2) was not possible to confirm as a viable method. A reason for this was that the learning material was too difficult for the users. A future improvement that may make the adaptive case selection better is to create learning material in closer cooperation with students and teachers. When it comes to the material, it should also be extended to include data that can be used to build customized feedback, since adaptive feedback was identified as the probably most effective adaptive learning technique for teaching chest X-ray interpretation. An example of such an improvement could be the improvement suggested in section 5.2.4. The system should not just give a generalized feedback, but feedback specifically based on the answers the user gave, and it should be a combination of both textual instructions and visual markings on the image.

The adaptive follow-ups and adaptive task type seemed to be viable methods. If more time and research is put into the design of the different task types, these methods will have an even greater positive effect.

7.4.2 Recommended Future Task Design Improvements

Follow-up task type 2 received good response from the participants, and this task and the problem-solving philosophy should therefore be kept. Even so, it is recommended to make follow-up task type 2 more interactive and engaging to solve, and it should have improved feedback to the user. The two other task types showed potential, but these tasks did not have sufficient feedback to the user. Some redesign should therefore be done so that the tasks better instruct the user about what he/she did wrong. Again most of these changes depend on improving the learning material the system is based on.

7.4.3 Other Necessary Improvements

This system was designed for the user testing sessions that were held. Some necessary functions that a finished deployed system should have were therefore not prioritized. Before the system is deployed and made available to students, the security functions mentioned in section 4.5 should be implemented. The small error concerning the clinical context, that was discovered, should be fixed. The clinical context should also not affect the adaptive learning algorithm's choices. It is also recommended that more and proper testing of all the system functions is done.

Bibliography

[1] B. S. Bloom, “The 2 sigma problem: The search for methods of group instruction as effective as one-to-one tutoring,” *Educational Researcher*, vol. 13, no. 6, pp. 4–16, 1984.

[2] A. Krokan, “Adaptiv lææring og læringsanalyse for raskere og bedre læring,” *Dialog*, 2015.

[3] EdSurge, “Decoding adaptive,” tech. rep., Pearson, 2016. `https://www.pearson.com/content/dam/one-dot-com/one-dot-com/global/Files/about-pearson/innovation/Pearson-Decoding-Adaptive-v5-Web.pdf` (Accessed: 13.09.2017).

[4] A. Paramythis and S. Loidl-Reisinger, “Adaptive learning environments and e-learning standards,” in *Second european conference on e-learning*, vol. 1, pp. 369–379, 2003.

[5] R. Azevedo, J. G. Cromley, D. C. Moos, J. A. Greene, and F. I. Winters, “Adaptive content and process scaffolding: A key to facilitating students’ self-regulated learning with hypermedia,”

[6] B. Grawemeyer, K. Karoudis, G. Magoulas, M. Pinto, and A. Poulovassilis, “Design and evaluation of adaptive feedback to foster ict information processing skills in young adults,” in *Proceedings of the 26th International Conference on World Wide Web Companion*, WWW ’17 Companion, (Republic and Canton of Geneva, Switzerland), pp. 369–377, International World Wide Web Conferences Steering Committee, 2017.

[7] K. Thyagharajan and R. Nayak, “Adaptive content creation for personalized e-learning using web services,” *Journal of Applied Sciences Research*, vol. 3, no. 9, pp. 828–836, 2007.

[8] C. H. Wu, T. C. Chen, Y. H. Yan, and C. F. Lee, “Developing an adaptive e-learning system for learning excel,” in *2017 International Conference on Applied System Innovation (ICASI)*, pp. 1973–1975, May 2017.

[9] A. T. Corbett and J. R. Anderson, "Knowledge tracing: Modelling the acquisition of procedural knowledge," *User Modeling and User-Adapted Interaction*, vol. 4, no. 4, pp. 253–278, 1994.

[10] P.-A. Andersen, C. Kråkevik, M. Goodwin, and A. Yazidi, "Adaptive task assignment in online learning environments," in *Proceedings of the 6th International Conference on Web Intelligence, Mining and Semantics*, WIMS '16, (New York, NY, USA), pp. 5:1–5:10, ACM, 2016.

[11] C. Martins, L. Faria, M. Fernandes, P. Couto, C. Bastos, and E. Carrapatoso, *PCMAT – Mathematics Collaborative Educational System*, pp. 183–212. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013.

[12] A. Klašnja-Milićević, B. Vesin, M. Ivanović, Z. Budimac, and L. C. Jain, *Design, Architecture and Interface of Protus 2.1 System*, pp. 185–212. Cham: Springer International Publishing, 2017.

[13] "Knewton adaptive learning." `https://www.knewton.com/wp-content/uploads/knewton-adaptive-learning-whitepaper.pdf`. Accessed: 20.09.2017.

[14] C. C. Aggarwal, *Recommender Systems: The Textbook*. Springer Publishing Company, Incorporated, 1st ed., 2016.

[15] R. E. Mayer, "Multimedia learning," , vol. 41, pp. 27–29, 2002.

[16] M. D. Roblyer, J. Edwards, and M. A. Havriluk, *Chapter 4: Integrating Instructional Software into Teaching and Learning*. Integrating educational technology into teaching, 2002.

[17] B. Rittle-Johnson and J. R. Star, *The Power of Comparison in Learning and Instruction: Learning Outcomes Supported by Different Types of Comparisons*, vol. 55 of *Psychology of Learning and Motivation*. San Diego, Calif.: Academic Press, 2011.

[18] K. J. Macura, R. T. Macura, and B. D. Morstad, "Digital case library: a resource for teaching, learning, and diagnosis support in radiology.," *Radiographics*, vol. 15, no. 1, pp. 155–164, 1995.

[19] D. R. Kool and J. G. Blickman, "Advanced trauma life support®. abcde from a radiological point of view," *Emergency Radiology*, vol. 14, pp. 135–141, Jul 2007.

[20] T. Thim, N. H. V. Krarup, E. L. Grove, C. V. Rohde, and B. Løfgren, "Initial assessment and treatment with the airway, breathing, circulation, disability, exposure (abcde) approach," *Int J Gen Med*, vol. 5, pp. 117–121, Jan 2012. ijgm-5-117[PII].

[21] R. S. Crausman, "The abcs of chest x-ray film interpretation," *Chest*, vol. 113, no. 1, pp. 256–257, 1998.

[22] X. N. Lam, T. Vu, T. D. Le, and A. D. Duong, "Addressing cold-start problem in recommendation systems," in *Proceedings of the 2nd international conference on Ubiquitous information management and communication*, pp. 208–211, ACM, 2008.

[23] J. Bobadilla, F. Ortega, A. Hernando, and J. Bernal, “A collaborative filtering approach to mitigate the new user cold start problem,” *Knowledge-Based Systems*, vol. 26, pp. 225–238, 2012.

[24] A. Rozenshtein, G. D. Pearson, S. X. Yan, A. Z. Liu, and D. Toy, “Effect of massed versus interleaved teaching method on performance of students in radiology,” *Journal of the American College of Radiology*, vol. 13, no. 8, pp. 979–984, 2016.

[25] L. Zou, A. King, S. Soman, A. Lischuk, B. Schneider, D. Walor, M. Bramwit, and J. K. Amorosa, “Medical students’ preferences in radiology education: a comparison between the socratic and didactic methods utilizing powerpoint features in radiology education,” *Academic radiology*, vol. 18, no. 2, pp. 253–256, 2011.

Appendix

Appendix 1: User Testing Survey in Original Language (Norwegian)

Brukervennlighet

- Synes du systemet var brukervennlig og enkelt å bruke?
  - Nei, ikke i det hele tatt
  - Helt greit, men kunne vært bedre
  - Ja, veldig
- Hva synes du om størrelsene på bildene?
  - Alle bildene var for små
  - Noen bilder var for små
  - Passe størrelse
  - Unødvendig store
- Opplevde du tekniske problemer underveis?
  - Nei, aldri
  - Noen ganger
  - Ja, hele tiden
- Har du ellers noe du vil legge til angående systemets brukervennlighet?

Oppgavenes relevans og vanskelighetsgrad

- Totalt sett, hvordan opplevde du oppgavenes vanskelighetsgrad?
  - Lett
  - Middels
  - Vanskelig
- Synes du oppgavenes innhold var relevant for det du trengte å bli bedre på?
  - Oppgavene var ikke relevante
  - Oppgavene var litt relevante

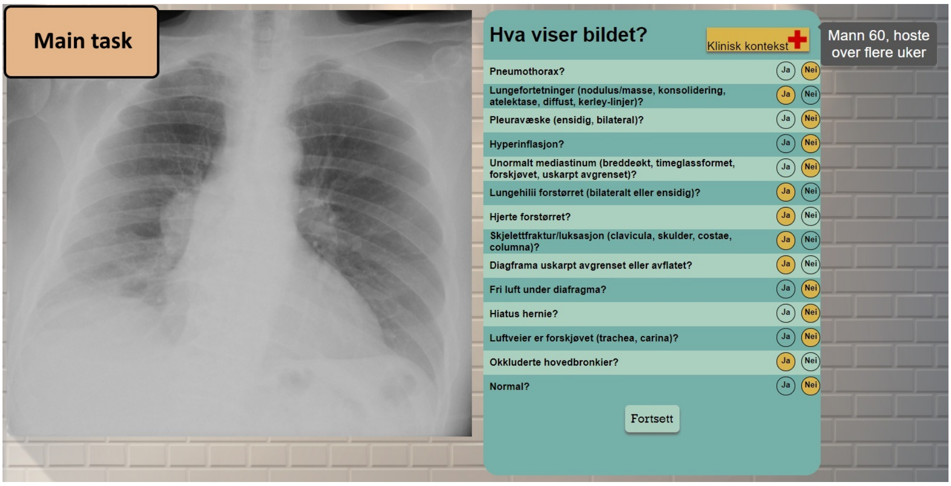
  - Oppgavene var veldig relevante

- Fikk du ofte oppgaver du anså som svært enkle?
  - Nei, aldri
  - Noen ganger
  - Ja, hver gang

- Fikk du ofte oppgaver du anså som svært vanskelige?
  - Nei, aldri
  - Noen ganger
  - Ja, hver gang

- Følte du at systemet tilpasset oppgavene underveis?
  - Nei, jeg opplevde at oppgavene var tilfeldige
  - Ja, noen av oppgavene virket å være tilpasset basert på tidligere feil jeg hadde gjort
  - Ja, alle oppgavene virket tilpasset til meg

- Synes du oppgavetypen beskrevet på bildet under var lærerik?



Hva viser bildet?

Mann 60, hoste over flere uker

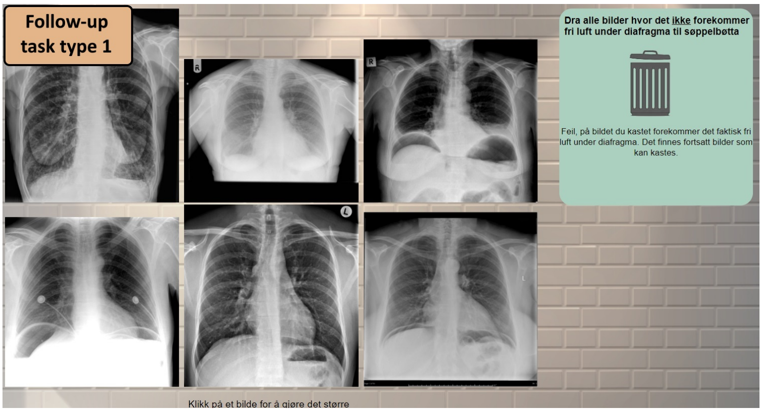
Pneumothorax? Ja [✓ Nei]
Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)? [✓ Ja] Nei
Pleuravæske (ensidig, bilateral)? Ja [✓ Nei]
Hyperinflasjon? Ja [✓ Nei]
Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)? Ja [✓ Nei]
Lungehilii forstørret (bilateralt eller ensidig)? [✓ Ja] Nei
Hjerte forstørret? [✓ Ja] Nei
Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)? [✓ Ja] Nei
Diagframa uskarpt avgrenset eller avflatet? [✓ Ja] Nei
Fri luft under diafragma? Ja [✓ Nei]
Hiatus hernie? Ja [✓ Nei]
Luftveier er forskjøvet (trachea, carina)? Ja [✓ Nei]
Okkluderte hovedbronkier? [✓ Ja] Nei
Normal? Ja [✓ Nei]

  - Lite lærerik
  - Noe lærerik
  - Veldig lærerik

- I systemet hadde du mulighet til å klikke på en knapp for å få opp klinisk kontekst for noen av bildene. Hvor hjelpsomt var dette for å komme frem til riktig svar?

  - Ikke hjelpsomt i det hele tatt
  - Litt hjelpsomt
  - Veldig hjelpsomt
  - Jeg klikket aldri på knappen for klinisk kontekst

- Fikk du en oppgave som ser ut som bildet under, hvis ja, synes du denne typen oppgave var lærerik?



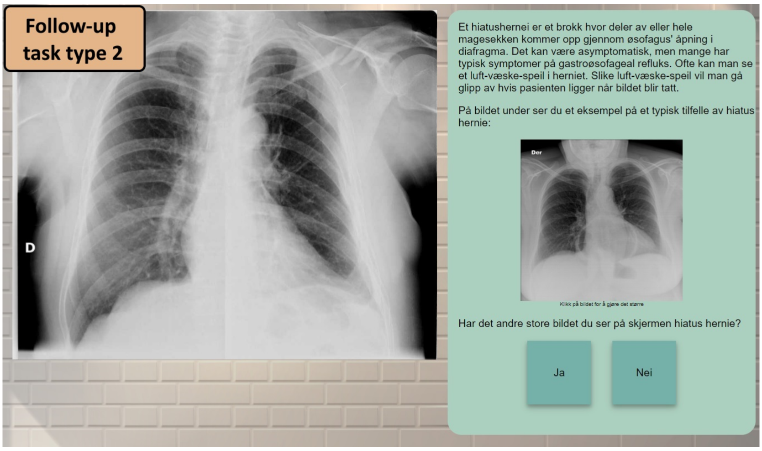
Dra alle bilder hvor det ikke forekommer fri luft under diafragma til søppelbøtta

Feil, på bildet du kastet forekommer det faktisk fri luft under diafragma. Det finnes fortsatt bilder som kan kastes.

Klikk på et bilde for å gjøre det større

  - Lite lærerik
  - Noe lærerik
  - Veldig lærerik
  - Fikk ikke oppgaven

- Fikk du en oppgave som ser ut som bildet under, hvis ja, synes du denne typen oppgave var lærerik?



Et hiatushernei er et brokk hvor deler av eller hele magesekken kommer opp gjennom øsofagus' åpning i diafragma. Det kan være asymptomatisk, men mange har typisk symptomer på gastroøsofageal refluks. Ofte kan man se et luft-væske-speil i herniet. Slike luft-væske-speil vil man gå glipp av hvis pasienten ligger når bildet blir tatt.

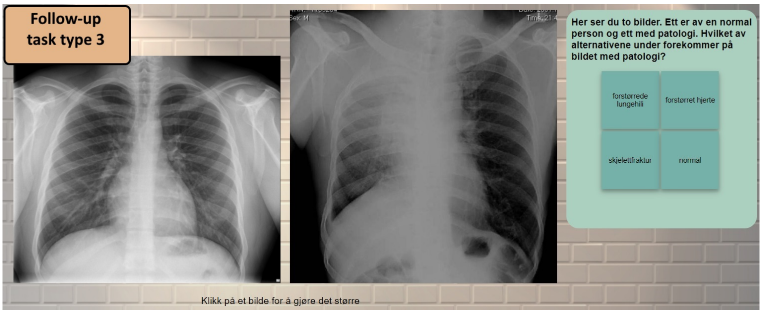
På bildet under ser du et eksempel på et typisk tilfelle av hiatus hernie:

Klikk på bildet for å gjøre det større

Har det andre store bildet du ser på skjermen hiatus hernie?

- Lite lærerik
- Noe lærerik
- Veldig lærerik
- Fikk ikke oppgaven

- Fikk du en oppgave som ser ut som bildet under, hvis ja, synes du denne typen oppgave var lærerik?



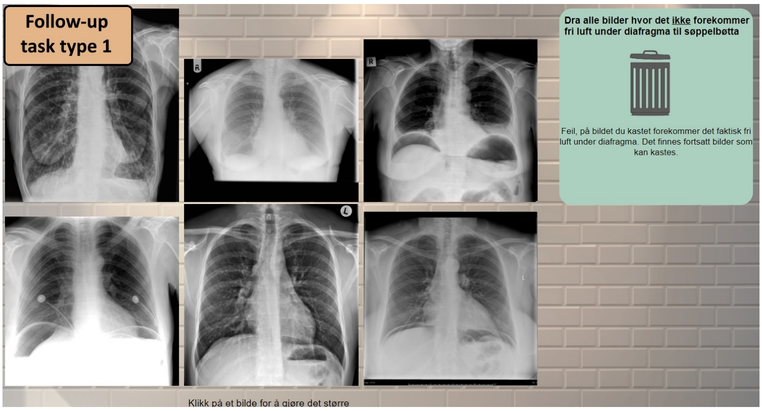
Follow-up task type 3

Her ser du to bilder. Ett er av en normal person og ett med patologi. Hvilket av alternativene under forekommer på bildet med patologi?

Klikk på et bilde for å gjøre det større

  - Lite lærerik
  - Noe lærerik
  - Veldig lærerik
  - Fikk ikke oppgaven

- Hvilken av oppgavetypene under likte du best?
  - Oppgavetype 1



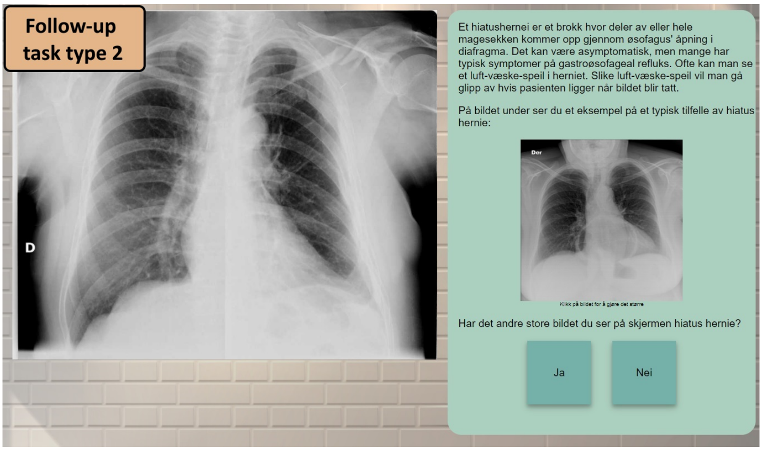
Follow-up task type 1

Dra alle bilder hvor det ikke forekommer fri luft under diafragma til søppelbøtta

Feil, på bildet du kastet forekommer det faktisk fri luft under diafragma. Det finnes fortsatt bilder som kan kastes.

Klikk på et bilde for å gjøre det større

  - Oppgavetype 2



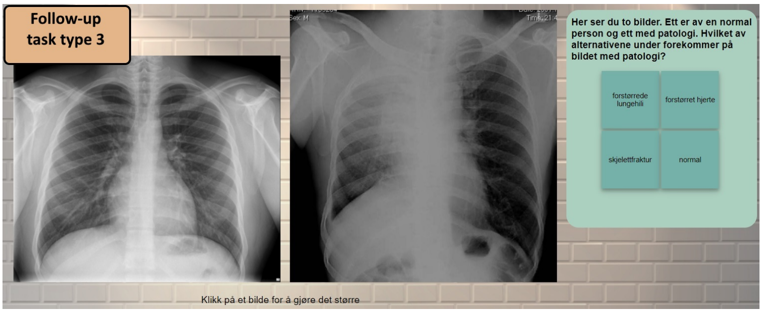
Et hiatushernei er et brokk hvor deler av eller hele magesekken kommer opp gjennom øsofagus' åpning i diafragma. Det kan være asymptomatisk, men mange har typisk symptomer på gastroøsofageal refluks. Ofte kan man se et luft-væske-speil i herniet. Slike luft-væske-speil vil man gå glipp av hvis pasienten ligger når bildet blir tatt.

På bildet under ser du et eksempel på et typisk tilfelle av hiatus hernie:

Klikk på bildet for å gjøre det større

Har det andre store bildet du ser på skjermen hiatus hernie?

- Oppgavetype 3

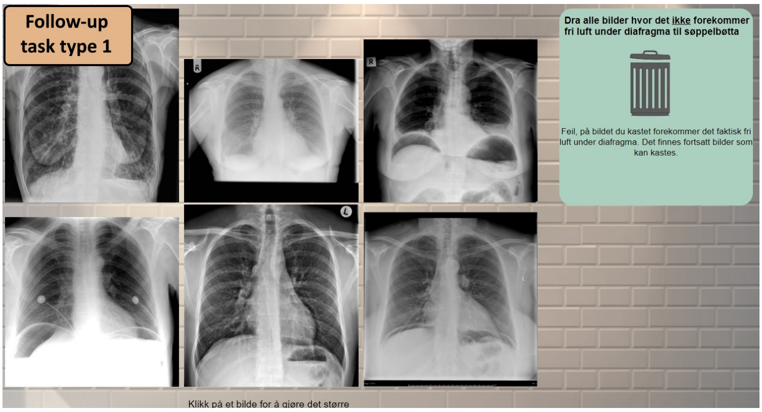


Her ser du to bilder. Ett er av en normal person og ett med patologi. Hvilket av alternativene under forekommer på bildet med patologi?

Klikk på et bilde for å gjøre det større

- Hvilken av oppgavetypene fra forrige spørsmål fikk du oftest da du testet systemet?

  - Oppgavetype 1



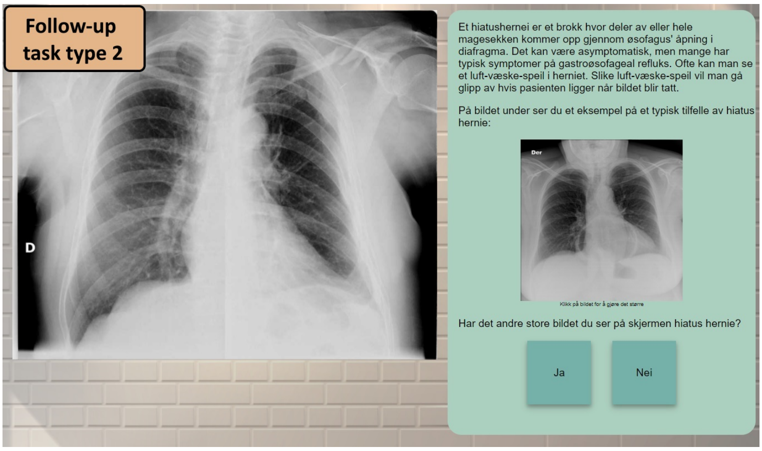
Follow-up task type 1

Dra alle bilder hvor det ikke forekommer fri luft under diafragma til søppelbøtta

Feil, på bildet du kastet forekommer det faktisk fri luft under diafragma. Det finnes fortsatt bilder som kan kastes.

Klikk på et bilde for å gjøre det større

- Oppgavetype 2



Follow-up task type 2

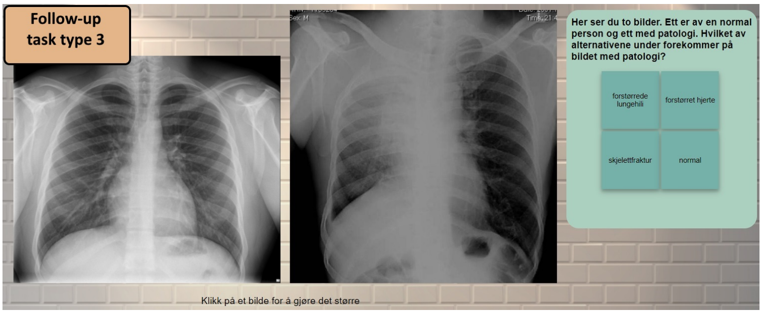
Et hiatushernei er et brokk hvor deler av eller hele magesekken kommer opp gjennom øsofagus' åpning i diafragma. Det kan være asymptomatisk, men mange har typisk symptomer på gastroøsofageal refluks. Ofte kan man se et luft-væske-speil i herniet. Slike luft-væske-speil vil man gå glipp av hvis pasienten ligger når bildet blir tatt.

På bildet under ser du et eksempel på et typisk tilfelle av hiatus hernie:

Klikk på bildet for å gjøre det større

Har det andre store bildet du ser på skjermen hiatus hernie?

- Oppgavetype 3



Follow-up task type 3

Her ser du to bilder. Ett er av en normal person og ett med patologi. Hvilket av alternativene under forekommer på bildet med patologi?

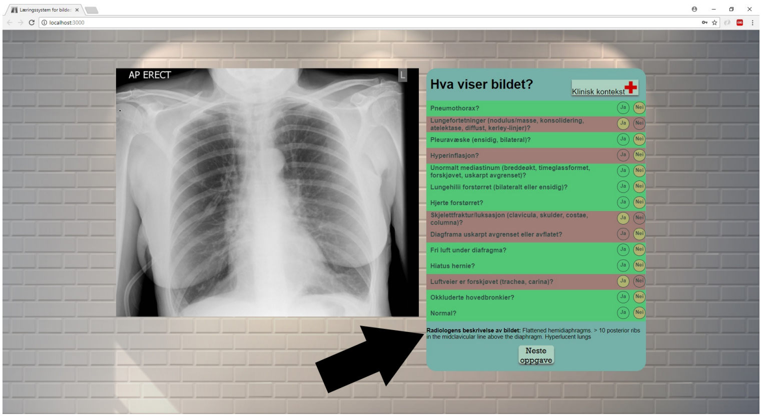
Klikk på et bilde for å gjøre det større

- Har du ellers noe du vil legge til angående oppgavene?

Systemets tilbakemeldinger til deg

- Etter å ha svart på en oppgave fikk du som regel en tilbakemelding fra systemet. Hvor lærerike synes du disse tilbakemeldingene var?
  - Lite lærerik
  - Noe lærerik
  - Veldig lærerik

- På noen av oppgavene kunne det etter at du hadde svart komme opp en melding kalt ”Radiologens beskrivelse av bildet”(som vist på bildet under) Hvor lærerik synes du disse tilbakemeldingene var?



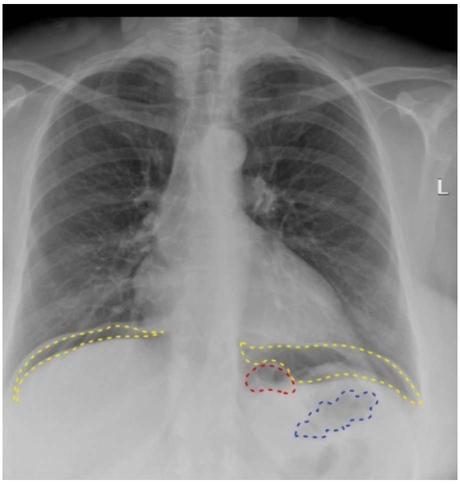
Hva viser bildet?

Pneumothorax? ○ Ja ● Nei
Lungefortetninger (nodulus/masse, konsolidering, atelektase, diffust, kerley-linjer)? ● Ja ○ Nei
Pleuravæske (ensidig, bilateral)? ○ Ja ● Nei
Hyperinflasjon? ○ Ja ● Nei
Unormalt mediastinum (breddeøkt, timeglassformet, forskjøvet, uskarpt avgrenset)? ○ Ja ● Nei
Lungehilii forstørret (bilateralt eller ensidig)? ○ Ja ● Nei
Hjerte forstørret? ○ Ja ● Nei
Skjelettfraktur/luksasjon (clavicula, skulder, costae, columna)? ● Ja ○ Nei
Diagframa uskarpt avgrenset eller avflatet? ○ Ja ● Nei
Fri luft under diafragma? ○ Ja ● Nei
Hiatus hernie? ○ Ja ● Nei
Luftveier er forskjøvet (trachea, carina)? ● Ja ○ Nei
Okkluderte hovedbronkier? ○ Ja ● Nei
Normal? ○ Ja ● Nei

Radiologens beskrivelse av bildet: Flattened hemidiaphragms. > 10 posterior ribs in the midclavicular line above the diaphragm. Hyperlucent lungs

  - Jeg fikk aldri opp slike tilbakemeldinger
  - Lite lærerik
  - Noe lærerik
  - Veldig lærerik

- En potensiell forbedring av dette systemet er at systemet som tilbakemelding markerer på bildet hvor avvikene var. (som vist på bildet under) Hvor stor forbedring tror du dette hadde vært for læringseffekten?



  - Ikke noe bedre enn tilbakemeldingene systemet allerede gir
  - Litt bedre enn tilbakemeldingene systemet allerede gir
  - Noe bedre enn tilbakemeldingene systemet allerede gir
  - Veldig mye bedre enn tilbakemeldingene systemet allerede gir

- Har du ellers noe du vil legge til angående systemets tilbakemeldinger til deg?

Annen informasjon

- Var det mulig å gjette seg frem til riktig svar?
  - Helt umulig
  - Mulig noen ganger
  - Mulig hver eneste gang
- Hvilken av beskrivelsene under synes du passer best til systemet du testet ut?
  - Et system som tester kunnskapene mine
  - Et system som skal forbedre kunnskapene mine
  - Annet...
- Ville du foretrukket at bruk av dette systemet var en del av den obligatoriske undervisningen?
  - Ja
  - Nei
- Ville du brukt dette systemet selv om det ikke var obligatorisk?

  - Ja
  - Nei

- Hvordan anser du dine egne forkunnskaper i diagnostisering av thorax røntgenbilder?
  - Lite forkunnskaper
  - Noen forkunnskaper
  - Mye forkunnskaper

- Totalt sett, hvordan vil du rangere opplevelsen din av systemet på en skala fra 1 til 10, hvor 1 er dårligst og 10 er best?

- Hva var brukernavnet ditt da du testet systemet?

- Er det ellers noe du ønsker å legge til?