RAPPORT

Gruppenavn og kandidater: Gruppe 21

Ahmed Anwar – ahmedan@uia.no

Ayub Omar Aden - ayuboa@uia.no

Hawler Abdullah Mohammed – hawleram@uia.no

Clovis Habonimana Bonja - clovishb@uia.no

Hashmatullah Botoon - hashmatulb@uia.no

Yousef Mouen Al-Saleh – yousefma@uia.no

Emnekode	IS-200-1 24H
Emnenavn	Systemanalyse og systemutvikling
Emneansvarlig	Geir I. Hausvik & Sofie Wass
Innleveringsfrist	20.11.2024 / 14:00
Antall ord (hele dokumentet)	16746
Tittel på prosjekt/produkt	Deliverable 4 : Final report

Vi bekrefter at vi ikke siterer eller på annen måte bruker andres arbeid uten at dette er oppgitt og at alle referanser er oppgitt i litteraturlisten.	**Ja** **X**	**Nei**

Kan besvarelsen brukes til undervisningsformål?	**Ja** **X**	**Nei**

Vi bekrefter at alle i gruppa har bidratt til besvarelsen	**Ja** **X**	**Nei**

Preface

The following report ‘Deliverable 4: Final Report’ explores the highs and lows, the group faced when embarking on the project alongside the organization Kartverket. Over a semester, the group adopted ideologies and techniques from the Scrum Framework and Object-Oriented Analysis & Design to achieve a web application to serve the needs and wants of Kartverket.

Due to a new approach in Scrum being taken by the group, the initial start was slow and ineffective. Aspects like the different Scrum events and roles took time for the group to adopt and fully trust when tackling this project. Over time and the many obstacles faced, the group learnt the values that Scrum provides and how important it is when performed correctly which helped us achieve the group’s end goals and ambitions.

The structure of the report explores the different phases of the project, such as the analysis and design phase, while also reflecting over the use of Scrum. This includes exploring the relevant actors and functions attached to the system while also taking a closer look at prioritized design criterion. In addition, the report expands on the use of Scrum throughout the project, highlighting the advantages and disadvantages the group faced.

We thank our lecturers and teacher assistants for their guidance and support, as well each group member of Group 21 for their efforts and contributions.

Table of Contents

Figures

Tables

1. Introduction

1.1 Overall Project's Goals

Kartverket, also known as the Norwegian Mapping Authority, is the national organization responsible for mapping, property registration and geospatial data. Kartverket was established in 1773 and have over eight hundred employees distributed over four different divisions; geodesy, land, sea, and property. The organization can also be seen as oldest technical agency in Norway that is still running and in operation.

A challenge Kartverket faces today, as an organization, is making sure the map data can be updated and correct at all times. When looking at the current system used by Kartverket, it is eye-opening how difficult it is to send a report on an issue seen on a map, whether that be a wrong street name, changed roads, or a new hiking trail. The system is difficult to navigate through, leading many to either quit while trying to report in an issue found or to not even report issues in the first place which has huge ramifications on the organization's map data and the information they can provide for its nation. This leads to a need for a solution in which reporting an issue found can be done in a user-friendly manner which both improves the map data with updates from its user, but also the inkling to report an issue in the first place knowing that it can be done and handled efficiently.

The project's overall goal is to develop a system where you can report in changes or issues found on the map, get to see the status on the different rapports they have sent in and receive answers and feedback on their reports whether that results into change or not. The goal is for this 'Crowd-Sourcing' solution to be user friendly and accessible for anyone to use, no matter their age or digital competency.

1.2 System Definition using FACTOR

During this introductory period of the project, the creation of a system definition was necessary for the group to get clarity over the overall project and various challenges to be faced. According to Mathiassen, a system definition is "a concise description of a computerized system expressed in natural language." The purpose of defining a system definition is to explore the different interpretations and possibilities, allowing for the opportunity to compare between the different alternatives. A system definition should be short and precise, containing the most vital decisions concerning the system. The system definition that was ultimately landed on, provides the right foundation for further analysis and design later down the line (Mathiassen, 2000, p.23-24).

When defining the system, it is important to listen to the different arguments and perspectives each have over the system and it's context. Through this, the system definition was defined using the FACTOR method which coherently outlines the most fundamental decisions required to develop a solid computerized solution. The FACTOR method was vital in the group defining a definition which highlighted the situations and ideas involved with the system giving a clearer idea of the system at hand, the information it should contain, its functions and contexts its used in. (Mathiassen, 2000, p.24).

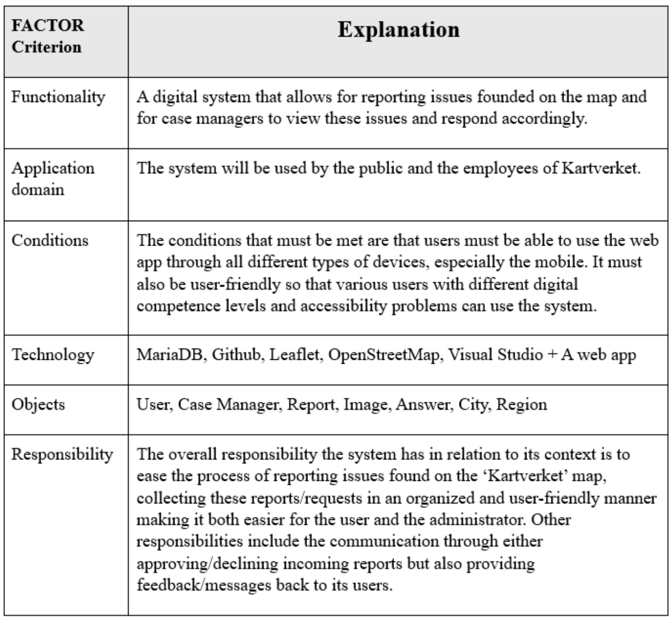
Using the FACTOR method, there are several criterion involved that we must take into account. *Functionality* explores the system's main focus when it comes to supporting the application domain. *Application domain* explores the part of the organization that has control in any capacity over the problem domain. *Conditions* are the set of must haves when developing the system. *Technology* includes the technology used both to develop the system and on which the system will run. *Objects* shares the main objects we see in the problem domain and lastly *Responsibility* sheds light to the overall responsibility the system has in relation to its context.

Our system definition and FACTOR analysis was based on the knowledge and information shared by Kartverket during the start of the project. The main takeaways from our introductory meeting plus the additional documents on the project and its requirements was the need for a system in which both reporting an issue but also answering the issue could be performed in a more efficient and user-friendly manner. By tackling this problem with a system that provides a user-friendly manner for reporting issues, looking at it's status but also for replying to these reports or delegating them and giving feedback will both encourage the public to report an issue they find due to it being simple and easy to do but will also improve efficiency within the organization on how they operate and organize incoming reports from the public leading to Kartverket having the necessary and most relevant map data.

Some of the functional requirements listed by Kartverket at the start of the project was for users to be able to create and log in with an account, report in issues using boxes, lines and points while also including a text box where users could write freely. The user should also be able to follow the status of their rapports, and also that the rapports are assigned to one or multiple municipalities through either the user choosing itself or it being automatically recognized based on the report's geometry. Other requirements, relating to the case managers who are responsible in reviewing the sent in rapports include the ability to accept, deny and give feedback on a rapport, the ability to delegate a rapport to another case manager and to sort out rapports based on municipality, priority or the date it was sent in.

The group first approached creating a system definition by having group members each present their thoughts and ideas on a suitable description for the system. Through listening and comprising on the different criterion mentioned earlier, the group finally landed on a description that felt both suitable and clear.

«A digital system that allows and promotes user-friendly interaction between the user and employees of Kartverket on various functions such as sending reports on changes found geographically, looking at the status on these reports, delegating and distributing different reports but also approving, rejecting, and writing back further comments. The digital system is designed as a web application compatible with working on mobile devices ensuring all levels of digital competence and accessibility can report their concerns and changes found in an easy and organized manner. The system is designed and responsible to be a vessel to ease and organize the process of reporting different matters in a modernized and reliable way.»



FACTOR Criterion	**Explanation**
Functionality	A digital system that allows for reporting issues founded on the map and for case managers to view these issues and respond accordingly.
Application domain	The system will be used by the public and the employees of Kartverket.
Conditions	The conditions that must be met are that users must be able to use the web app through all different types of devices, especially the mobile. It must also be user-friendly so that various users with different digital competence levels and accessibility problems can use the system.
Technology	MariaDB, Github, Leaflet, OpenStreetMap, Visual Studio + A web app
Objects	User, Case Manager, Report, Image, Answer, City, Region
Responsibility	The overall responsibility the system has in relation to its context is to ease the process of reporting issues found on the 'Kartverket' map, collecting these reports/requests in an organized and user-friendly manner making it both easier for the user and the administrator. Other responsibilities include the communication through either approving/declining incoming reports but also providing feedback/messages back to its users.

Tabell 1, System definition using FACTOR

1.3 Description of the developed system

The developed system was designed for the public to report changes or issues found on the map and improve the user's experience of viewing the status of the report and receiving feedback from Kartverket. The developed system also serves and aids the case managers who are employed at Kartverket. Case managers can accept/deny reports, give feedback, delegate reports to other case managers who are assigned a specific municipality, and have a clear overview of the reports sent in.

For the users, the developed system increases the chances of users returning back to the system and reporting issues found which is crucial for Kartverket and their mission of providing the latest and correct geographical data. For case mana gers, the developed system increases the productivity of case managers and their ability to answer users efficiently. Providing a clear overview to the case managers also helps them in structuring the user priority list and workload.

The group have identified four main groups who the system serves, which are the public, case managers, admins and emergency services. As mentioned before, the public have access to reporting issues, viewing the status of these reports, and viewing all reports ever sent in. Case managers have access to also report in an issue but also have access to respond to these reports, give feedback, delegate reports, and have an overview of all reports sent in. Admin have access to all users registered to the system and the ability to assign the role of the case manager to others. On the other hand, emergency services are seen to have the same role and access as the public, with the only difference being that emergency services obtain top priority visible for the case managers to see leading to the reports sent in by the emergency services being handled and answered first. This was an aspect communicated earlier by Kartverket and is further explored within the analysis section.

Something worth mentioning is that the system is flexible for use on both computers and phones making the system more accessible and flexible to use, being another requirement communicated by Kartverket.

2. Analysis

analysis. "the act of studying or examining something in detail, in order to discover or understand more about it, or your opinion and judgment after doing this" (Cambridge Dictionary, n.d.).

When embarking onto different projects, the analysis phase can be seen as a critical cog in the machine informing and creating more clarity on the problem at hand. In our case, it was vital for the group to be well informed and all in agreement when it came to our understanding of 'Kartverket' and what was required in providing a successful solution.

2.1 Object-Oriented Analysis & Design

Object-Oriented Analysis & Design, also known as OOA&D, is an approach and method where key concepts such as objects and classes are used to understand the problem at hand and the system to be designed (G. I. Hausvik & S. Wass, Lecture 3 - Analysis I, 26.08.2024). Within OOA&D, the first goal is to understand what the system is required to do, then identify the relevant objects and afterwards decide how these objects will work and interact with each other (GeeksforGeeks, 2020).

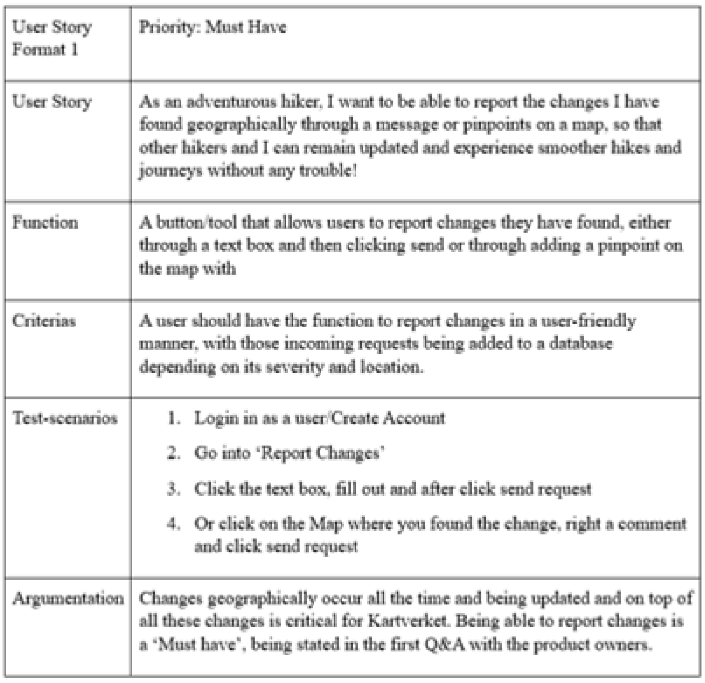
Using the OOA&D approach, we can split the analysis phase into two separate categories: The problem domain and the application domain. The problem domain concerns itself with the part of a context that is administered, monitored, or controlled by the system, whereas the application domain is the organization that administrates, monitors, or controls a problem domain (Mathiassen, 2000, p. 6).

We can also detect that where you start in terms of analysis, has a great deal to say on the approach and result. Beginning by analyzing the application domain "your focus is on the users' work, and you might specify many detailed requirements." Beginning by analyzing the problem domain "your focus will be on what the business is really about, rather than on interfaces and functions" (Mathiassen, 2000, p. 115). Although the easier option to start with can be seen with application domain, starting with problem domain can lead to better results which is ultimately what the group landed on in terms of the approach we took during the analysis phase (Mathiassen, 2000, p. 116).

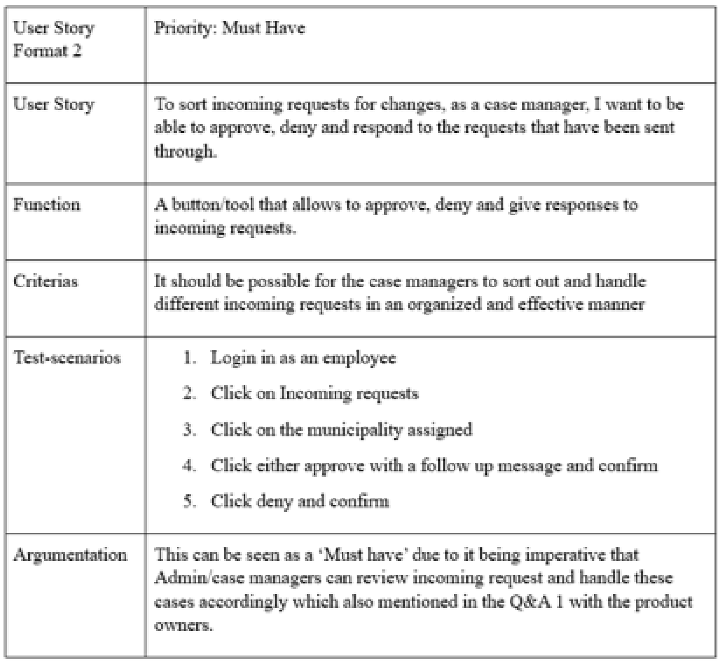
User stories

Before exploring both the problem domain analysis and application domain analysis, we can first introduce some user stories and their associated acceptance criteria related to the system. To clarify, user stories are used to describe, in a solid and compact style, pieces of the functionality from a user's point of view (G. I. Hausvik & S. Wass, Lecture 3 - Analysis I, 26.08.2024).

In creating the user stories, each group member was tasked with creating their own user stories using both formats listed below. This led to the group finding what each other valued when it came to functions and it's priority for the system. (See Appendix A for other examples)



User Story Format 1	Priority: Must Have
User Story	As an adventurous hiker, I want to be able to report the changes I have found geographically through a message or pinpoints on a map, so that other hikers and I can remain updated and experience smoother hikes and journeys without any trouble!
Function	A button/tool that allows users to report changes they have found, either through a text box and then clicking send or through adding a pinpoint on the map with
Criterias	A user should have the function to report changes in a user-friendly manner, with those incoming requests being added to a database depending on its severity and location.
Test-scenarios	1. Login in as a user/Create Account 2. Go into 'Report Changes' 3. Click the text box, fill out and after click send request 4. Or click on the Map where you found the change, right a comment and click send request
Argumentation	Changes geographically occur all the time and being updated and on top of all these changes is critical for Kartverket. Being able to report changes is a 'Must have', being stated in the first Q&A with the product owners.



User Story Format 2	Priority: Must Have
User Story	To sort incoming requests for changes, as a case manager, I want to be able to approve, deny and respond to the requests that have been sent through.
Function	A button/tool that allows to approve, deny and give responses to incoming requests.
Criterias	It should be possible for the case managers to sort out and handle different incoming requests in an organized and effective manner
Test-scenarios	1. Login in as an employee 2. Click on Incoming requests 3. Click on the municipality assigned 4. Click either approve with a follow up message and confirm 5. Click deny and confirm
Argumentation	This can be seen as a 'Must have' due to it being imperative that Admin/case managers can review incoming request and handle these cases accordingly which also mentioned in the Q&A 1 with the product owners.

Figure 1, User stories

In figure 1, we can see an example of user stories using two different formats due to the group deeming them as both necessary, serving different and varied perspectives and directions. *Format one* is structured as: "As a [type of user] I want to [some functionality] so that [some benefit]", whereas *Format 2* is structured as: "To [achieve som value] as a [type of user] I want [some functionality]."

When deciding on the prioritization of the user stories created by the group, it was deemed that the MoSCoW framework was both logical and applicable due to the group's past experiences using the framework. The MoSCoW framework is divided into four different priority levels including: Must Have, Should Have, Could Have and Wont have (G. I. Hausvik & S. Wass, Lecture 3 - Analysis I, 26.08.2024).

2.2 Problem Domain Analysis

We can now introduce problem domain analysis which can be related back to an important question which is "What information should the system deal with?" Within this particular area of analysis, it is important to focus on the different elements and variables at play. Problem domain analysis can be split into three specific activities focusing on the classes, structure, and behavior (Mathiassen, 2000, p. 46).

2.2.1 Classes

The first specific activity within problem domain analysis is the different classes and objects at hand. This requires the selection of the objects, classes, and events that will be the elements of the problem domain model. (Mathiassen, 2000, p. 46-47).

Event table

To begin with, we can state that selecting classes and events helps in characterizing and defining the problem domain. Due to the classes defining the relevant aspects of the problem domain and being seen as dynamic, we can be aware of events and their links with objects. Events are seen as the basic element of object behavior. The classes activity leads us into the exploration of an event table, involving different classes and their related events (Mathiassen, 2000, p. 47).

An Event table is a model to describe events that happen in the system and which objects will be affected. An Event-table is important for our system analysis as it helps the group understand the system on a deeper level. We felt that an event table would be suitable to use and include since it shows the group in an organized and well-structured manner the events and classes that are ever so present alongside their quality which can aid the group in evaluating our chosen events and classes (Mathiassen, 2000, p. 49).

Events	User	Administrator	Report	Image	Answer	City	Region
Sent	**X**	**X**	**X**	**X**	**X**		
Received			**X**		**X**		
Created	**X**	**X**					
Deleted	**X**	**X**					
Edited						**X**	**X**
Added				**X**			

Table 1, Event table

2.2.2 Structure

The second specific activity within problem domain analysis deals with structural relations between the classes and objects. The structure activity can include a class diagram resulting in presenting the chosen classes and highlighting the structural relationships between the classes and objects (Mathiassen, 2000, p. 47).

Class Diagram

When looking at a class diagram at first, it may be hard to decipher what the diagram is trying to show us. However, a class diagram "describes a collection of classes and their structural relationships" (Mathiassen, 2000, p. 336). It is a visual way to analyze the classes in the problem domain and structure them, accordingly, describing the classes presented in the problem domain and the relationship that appear between them.

The diagram below describes eight different classes that are related to each other. At the top there is a region class which represents a region in a country. This region can have one or more cities which are part of it and one or more people. A Person class is a generalization of user and administrator classes which represent the user of the system and an employee in Kartverket. A reports class represents a report which can be sent by one user, belong to one region, include zero or more images, and have one answer. An answer class represents an answer to a report and an Image class representing an image in a report.

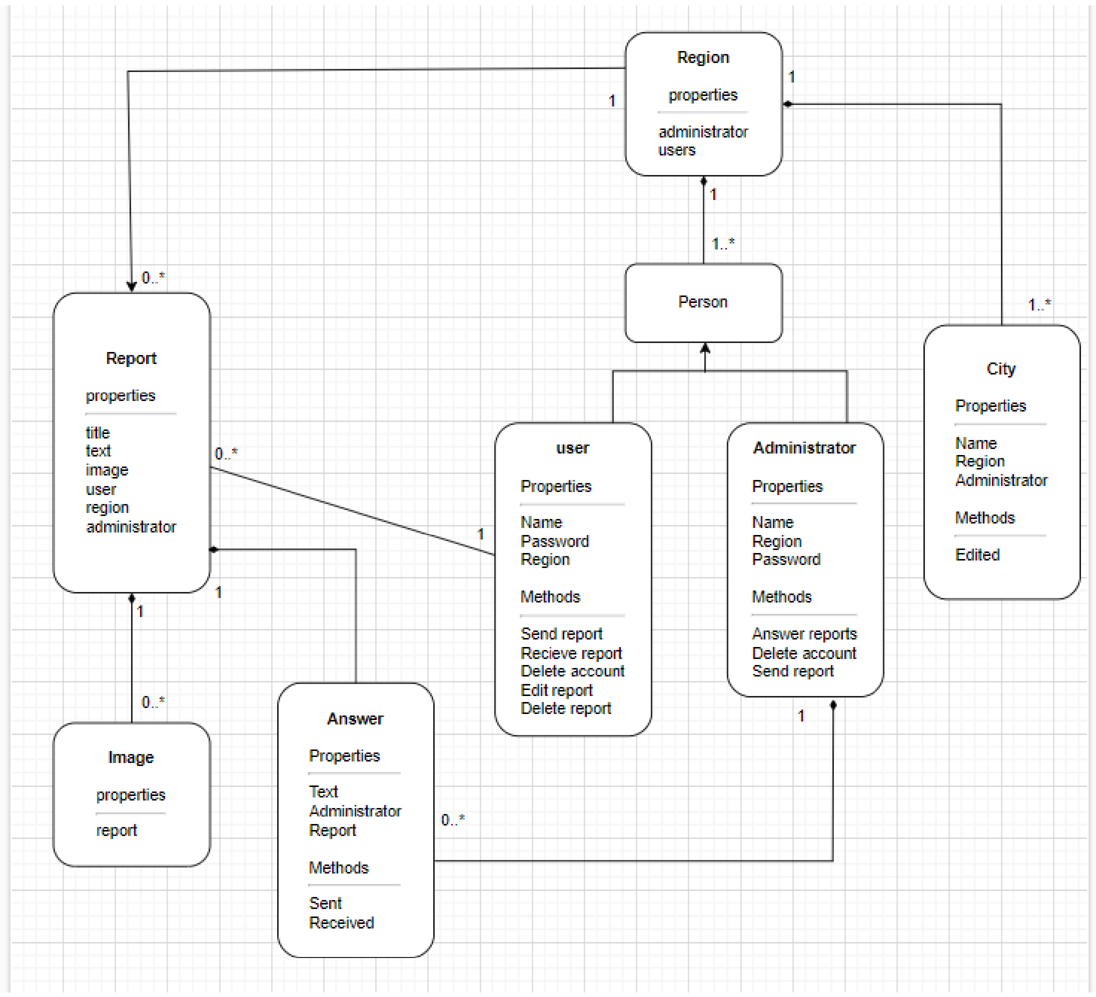


Figure 2, Class Diagram

2.2.3 Behavior

The third and last activity within problem domain analysis involves object behavior and interaction. When looking into behavior, we describe the “dynamic properties and attributes for each of the selected class” creating a more detailed insight into each object of the problem domain (Mathiassen, 2000, p. 47).

Rich Picture

We can first start off with exploring rich pictures, as it was one of the first initial sketches the group created when embarking on visualizing the problem and scenario at hand. Rich pictures can be seen as a broad description of a certain situation including just the key aspects like the objects involved, their relationship with one another, the different processes and structures in play. (Mathiassen, 2000, p. 26-27).

During the exploration of different models to choose from to express the group's thought process and ideas, we found that rich pictures was perfect in assisting as a visual medium for our perspective and understanding behind the situation. In many cases, it can be difficult to get across what we think and how we comprehend the situation at hand through text, so having a visual aid like the figure below helps us also reach a common understanding within the group. It can be seen as a journey you take experiencing the situation visually and gaining a new perspective you might not have got by just reading a text.

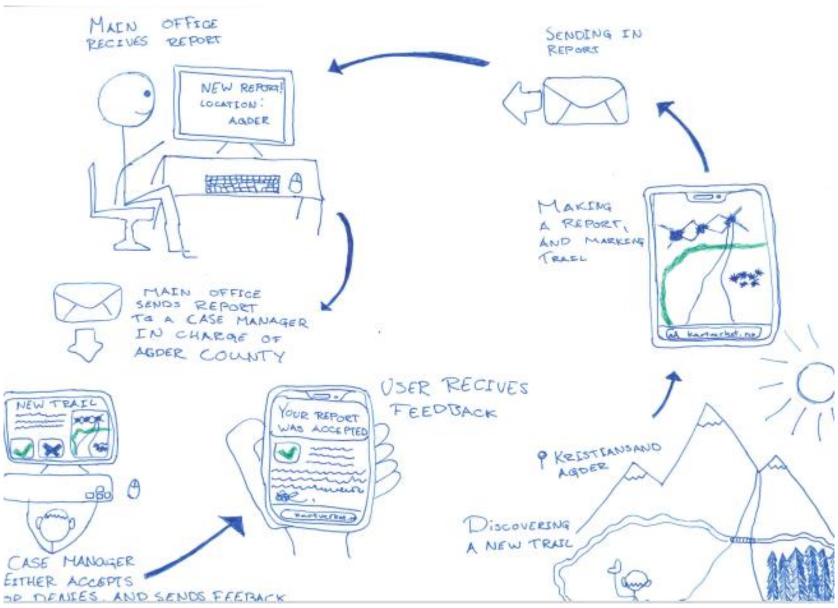

MAIN OFFICE RECIVES REPORT
NEW REPORT! LOCATION: AGDER
SENDING IN REPORT
MAKING A REPORT, AND MARKING TRAIL
kartverket.no
MAIN OFFICE SENDS REPORT TO A CASE MANAGER IN CHARGE OF AGDER COUNTY
NEW TRAIL
YOUR REPORT WAS ACCEPTED
USER RECIVES FEEDBACK
KRISTIANSAND AGDER
DISCOVERING A NEW TRAIL
CASE MANAGER EITHER ACCEPTS OR DENIES, AND SENDS FEEBACK

Figure 3, Rich Picture

The figure above depicts a rich picture which displays a user hiking in the woods who discovers a new trail that is not on the map, they go on kartverket.no and mark the trail on the map on the website. They make a report regarding this and send the report in. An employee from the main office receives this report and sends it to the employee responsible for Agder county/area. This employee reads the report and based on the procedures done checking the context, they accept the report and write a feedback message. The figure visually explains a situation that would likely take place in the problem domain we have defined.

State Chart

The Sate Chart, also known as a State Diagram, is a behavioral diagram used to model the dynamic behavior of a system. It shows the different states a system (or an object) can be in, as well as the transitions between those states. In other words, they help clarify the behavior of objects over time and their response to events (Visual Paradigm Guides, 2023).

The group's state chart models the interaction and behavior between internal (Kartverket employees) and external users of the system.

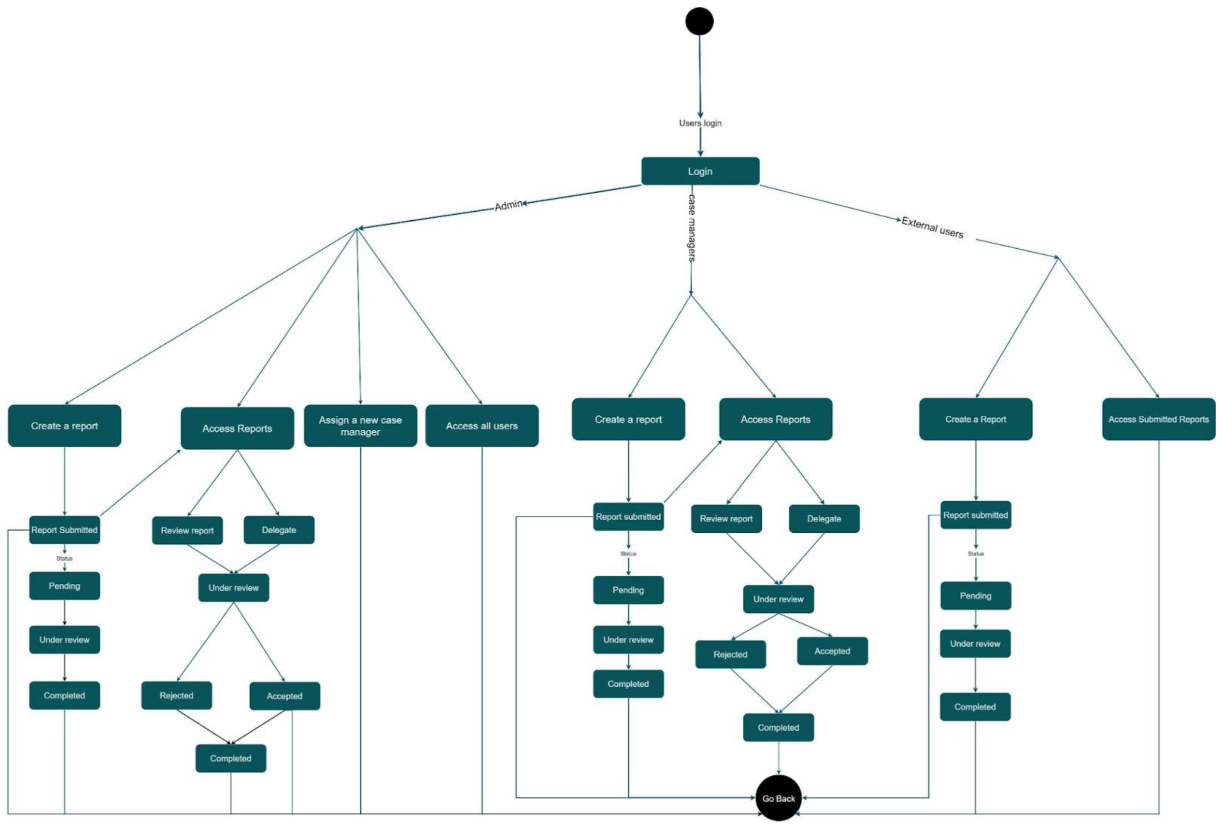


Figure 4, State Chart

All users, no matter their role start by logging in, and based on role being whether Admin, Case Manager, or External User they are directed to specific workflow certain functionalities. Let's have a closer look at these role base workflows:

Admin

- **Create a Report:** Admins can create new reports and submit them. Once a new report is created, and submitted, the report moves to the "Report Submitted" state, followed by "Pending" status. The report stays in "Pending" state until its "Under review" by an internal user before going into "Completed" state.
- **Access Reports:** Admins can access received reports, and then choose to review or delegate them to case managers. Once a report is being reviewed by an internal user (admin/case manager), it moves into "Under Review" state where it can either be "Accepted" if the report is valid, or "Rejected" if it's invalid, before entering "Completed" state. The arrow pointing to the black circle represents the final state of the workflow and that no further actions are needed for that specific report and users can "Go Back" and perhaps start some other task.
- **Assign a New Case Manager:** Admins can appoint new people to the Case manager role.
- **Access All Users:** Admins have full access and overview of the users, both internal and external.

Case Manager

- **Create a Report:** Case managers can create and submit a new report and its process is the same as the one in the Admin workflow
- **Access Reports:** Case Managers gain access to reports received by external users and also reports delegated to them from Admins and other Case Managers. They can either review those reports, or delegate them to other case managers. Reports then enter the "Under review"

state, and get either accepted or rejected before going into “Completed” state and finally end up in the final state

External Users

- **Create a Report**: External users can create and submit new reports if they discover map issues and their workflow for report creation works the same way as Admin and Case Manager workflows.
- **Access Submitted Reports:** When external users login and start submitting reports, they gain full access over their submitted reports, its statuses, whether their reports got accepted or rejected and the feedback they received from Kartverket employees (Admins/Case Managers).

2.3 Application Domain Analysis

After discussing problem domain analysis, we can now move onto application domain analysis which also focuses on an important question: “How will the target system be used?” Identifying how our system will be used is vital for defining the different requirements needed (Mathiassen, 2000, p. 115). Application domain analysis can also be split into activities, with the two being usage and functions.

2.3.1 Usage

The first activity within application domain analysis is Usage, where its main goal is to “determine target-system use in the application domain by identifying and structuring actors and use cases” (Mathiassen, 2000, p. 118). The group found that understanding how the different actors interacted with the system was crucial for our ideas and solutions to match both the user’s needs and also the product owner’s requirements.

Actors

Actors can be seen as an abstraction of people and other systems that interact with the system at hand. To truly understand the actors that were relevant in our case, the group decided in structuring a list including the different actors and how they would interact with the target system (Mathiassen, 2000, p. 120). The actor list created feature four clear actors involved with the system:

The public or regular user would use the system to report in an issue found on the map, to check on the rapport’s status, check on previous reports and also receive feedback on the rapports sent in.

Case Managers working for Kartverket would use the system to receive incoming reports, reply to rapports either accepting or declining while providing feedback and delegate rapports to other Case Managers. It also important to mention that they can also report in an issue they found on the map just like a regular user.

Admin can be seen as Case Managers just with the additional role of having an overview over all users who are registered to the system and the ability to create/assign the role of the Case Manager to others.

Emergency Services would use the system for the same reasons as the public or a regular user but would receive the option to put their incoming rapports onto the top of the priority list due to their occupation and authority within society.¨

To differentiate between the different actors and involvement with the system, the group landed on creating a Use-Case diagram as it demonstrates best the possibilities each actor holds with the system.

Use-Case Diagram

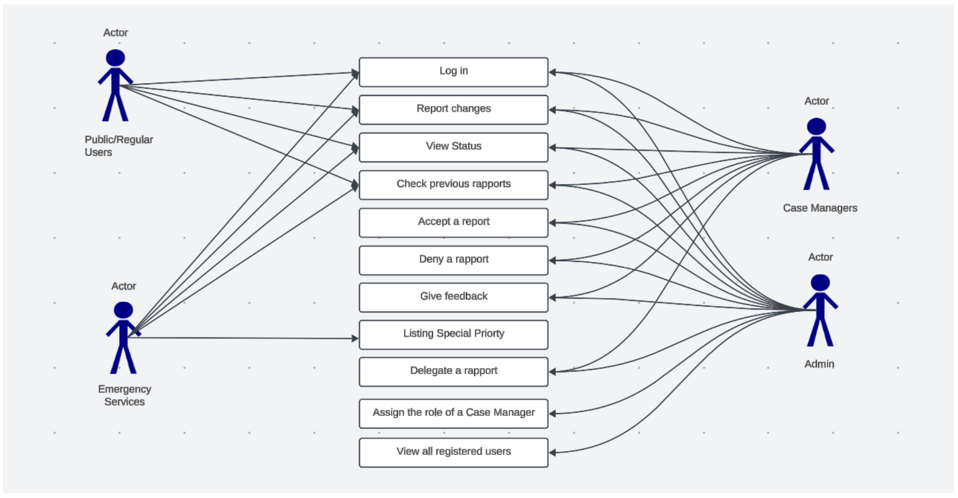


Figure 5, Use-Case Diagram

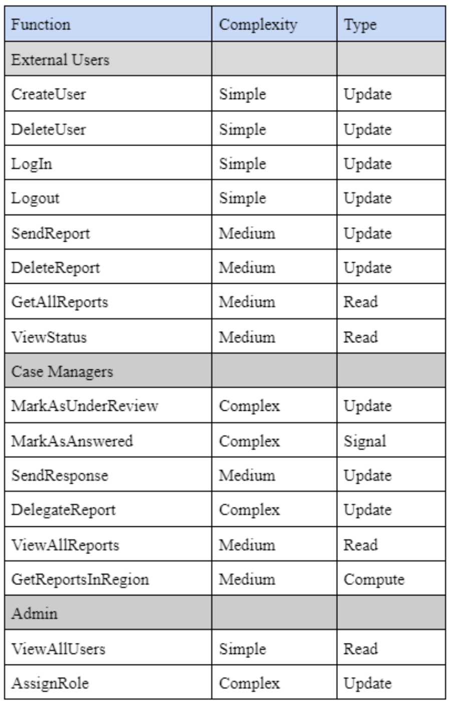
The Use-Case diagram highlights the functions that both the user, admin and emergency services (Actors) have access to when logging in and using the system. In figure 5, everyone has access to logging in, reporting changes found, viewing the status on different reporting sent and checking previous rapports sent. Case Managers possesses additional functions such as approving reports, rejecting reports, sending further comments, and delegating a rapport to another case manager. Admin can be seen as similar to Case Managers as both are internal users, with the only difference being that admin can see all users registered on the system and assign the role of a Case Manager. Emergency Services was also included to shed light on 'special priority' which other users do not have access to. Emergency Services such as the police and/or fire department can notify its position/authority when logging on, report different issues and be granted priority accordingly to receive responses sooner.

2.3.2 Functions

After going through the first activity in Usage, the next step is to define and explore the possible functions relevant to the system for it meet the requirements and goals set both by Kartverket and the group. Functions focuses on what the system can do to aid actors in what they intend to do (Mathiassen, 2000, p. 137).

When embarking on differentiating the different functions to get a clear overview on what is being developed, the group landed on creating a functions list as it is well structured and organized. The functions included by the group were based on the requirements given, the different user stories and the Use-Case diagram.

Before going through our functions list, we can first elaborate on what a function list really is. A functions list is a way to analyze the many functions a system has, their complexity and their type. A basic rule within functions lists is "that you should describe functions briefly and informally in a list" (Mathiassen, 2000, p. 144-145). For complexity, as a group we came to the conclusion that many of our functions listed below were either simple or medium based on how long the group estimated it would take and our experience with implementing similar functions prior to the project. A few of the functions were listed as complex due to the group's lack of knowledge and the estimation attached to the function given by the group.



Function	Complexity	Type
External Users		
CreateUser	Simple	Update
DeleteUser	Simple	Update
LogIn	Simple	Update
Logout	Simple	Update
SendReport	Medium	Update
DeleteReport	Medium	Update
GetAllReports	Medium	Read
ViewStatus	Medium	Read
Case Managers		
MarkAsUnderReview	Complex	Update
MarkAsAnswered	Complex	Signal
SendResponse	Medium	Update
DelegateReport	Complex	Update
ViewAllReports	Medium	Read
GetReportsInRegion	Medium	Compute
Admin		
ViewAllUsers	Simple	Read
AssignRole	Complex	Update

Table 2, Functions List

The list includes functions that are divided into four types: Read, Update, Compute, and Signal. Read functions are functions that only retrieve information from the database and display to the user. Update functions are functions that add new information to the database or update existing information. Compute functions are functions that perform calculations and signal functions are functions that occur as a result of an event in the system (Mathiassen, 2000, p. 140).

3. Design

design. “To make or draw plans for something” (Cambridge Dictionary, n.d.).

After the analysis phase of this project, which included taking a close look at the problem and application domain, it is now time to embark on the design phase. Within the design phase, the main goals for the group to achieve is defining a design criteria for both the system’s user interface and architectural design ensuring the requirements and wishes are met. This includes the exploration and prioritization of various criterion, principles and laws while also figuring out what the group most values when it comes to the system. However, before we get into that, we can first look into identifying the special conditions related to the project and group.

3.1 Identifying special conditions

When identifying specific conditions which are conditions unique to the project and group, we can split them into different categories: Technology, Organization and Human Resources.

The first category, *technology*, relates back to existing hardware, software or systems. This also included the reuse of patterns seen from before. In our case, through the product owners and the parameters that have been set for the project, we have been given quite a lot of independence and freedom in the manner we choose to create things in terms of the hardware, software or even the system. However, it has also been made clear that the product owners and Kartverket would wish for a system that has close similarities with the present system currently to not strand too far from its identity and relevance in other people's mind.

The second category worth exploring is the organization, as it has a big impact within our group and on the overall project. Future plans for further development or possible transferring/merging are unclear and have not been discussed yet in thorough detail by the product owners creating uncertainty with the direction of the structure and delivery of the final product.

The last category worth exploring when it comes to identifying special/specific conditions is human resources which explores the different aspects and variables that comes to play when working as a group. The group acknowledges and are fully transparent towards the little experience we all share when it comes to tackling a relatively long-term project. This affects the way we estimate and delegate tasks which are a huge aspect of working in a project and ultimately succeeding. Our competencies vary within the group, where the same can be said for the workload and expectations we all carry.

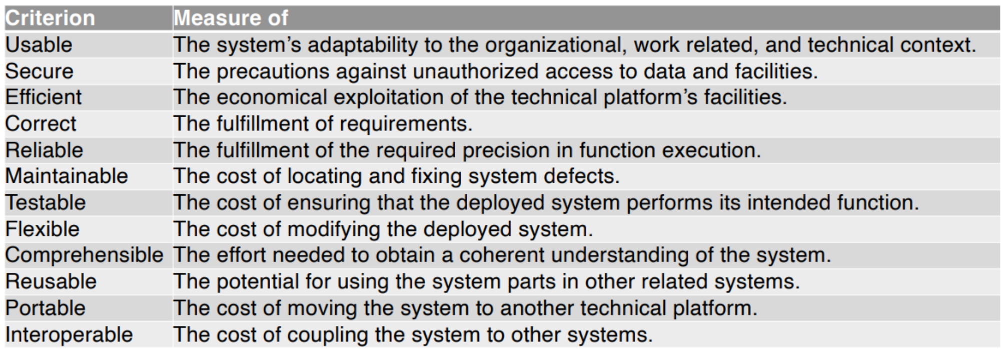
3.2 Design Criteria for the system's user interface

Before delving into the relevant criterion, principles and laws when creating the design criteria for the system's user interface, giving an explanation on what a user interface is gives a better understanding and more context on what the group values and strives for in terms of the system's user interface.

The user interface (UI) is "the point of human-computer interaction and communication in a device" (Churchville & Hashemi-Pour, 2021). Something also worth mentioning is that a good UI is molded into suiting the users' tasks and idea of the system (Mathiassen, 2000, p. 152).

3.2.1 Relevant criterion from the general criteria

Starting off with the design criteria for the system's user interface, we can first introduce the 'General Criteria' seen in figure 6 down below. The 'General Criteria' introduces us to a variety of criterion each showing their way of measuring the success of the criterion. The group felt the inclusion of the general criteria was important as each criterion shed light on the different aspects of the project and their importance.



Criterion	Measure of
Usable	The system's adaptability to the organizational, work related, and technical context.
Secure	The precautions against unauthorized access to data and facilities.
Efficient	The economical exploitation of the technical platform's facilities.
Correct	The fulfillment of requirements.
Reliable	The fulfillment of the required precision in function execution.
Maintainable	The cost of locating and fixing system defects.
Testable	The cost of ensuring that the deployed system performs its intended function.
Flexible	The cost of modifying the deployed system.
Comprehensible	The effort needed to obtain a coherent understanding of the system.
Reusable	The potential for using the system parts in other related systems.
Portable	The cost of moving the system to another technical platform.
Interoperable	The cost of coupling the system to other systems.

Figure 6, General Criteria

Before going into detail on the relevant criterion at hand for the system's user interface, it is important to preface that some of criterion left out will be picked upon again when discussing the architectural design. This creates a nice balance and a well-structured format giving the right amounts of concentration to each criterion being described and argued for. It is also important to note that many of the criterion taken from the 'General Criteria' can overlap being vital for not only the system's user interface but also the architectural design.

‘Usable’ explores on “the system’s adaptability to the organizational, work related, and technical context” (Mathiassen, 2000, p. 178). In other words, this criterion measures if the system can be used or not and if so, how usable? In our case, this can be seen as something that is critical as a huge focus is improving the overall experience both the internal and external user has when performing their desired actions on the system.

‘Secure’ explores the security and safety of the system including “precautions against unauthorized access to data and facilities” (Mathiassen, 2000, p. 178). This is something the group values a big deal as within the system, data and authority can be seen as ever so present with data needing to be filled in but also internal users having other functions the rest of the users do not have. Having a secure system is deemed critical and one of the priorities within the group as it allows users to feel safe giving data but also internals users obtaining the right functions allowed to complete varied tasks.

Moving on to the next criterion chosen to be identified and argued is ‘Correct’, being one of few that is self-explanatory. Fulfilling its requirements and being categorically correct is something that is critical to obtain as the whole idea behind this project is correcting issues and misinformation.

‘Reliable’ is concerned with “the fulfillment of the required precision in function execution” (Mathiassen, 2000, p. 178). This relates back to ensuring that the system executes with the exact same level of accuracy as intended every time. Making sure our system is dependable has significant importance to this project due to the system being used and operated by many. Especially when it comes to the priority list including police and ambulance workers, it has been made clear for a reliable and quick way for reporting various issues.

‘Flexible’ can be seen as “the cost of modifying the deployed system” (Mathiassen, 2000, p. 178). In our case, it is important our deployed system is flexible in the way it operates but also looks. It has been made clear by the product owners that the system should be flexible to use both on the computer and the phone, something the group has looked a great deal into.

An example of ‘Flexible’ being implemented in the developed system can be seen in the figure below. Here we can see two versions of the registration page, the first version being how it looks on a mobile screen while the second version is how it looks on a computer screen.

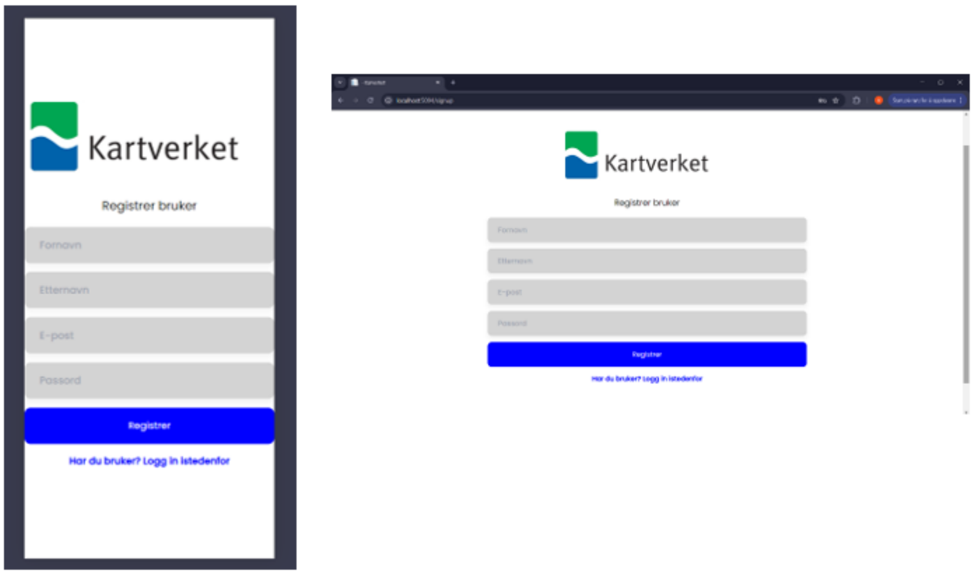
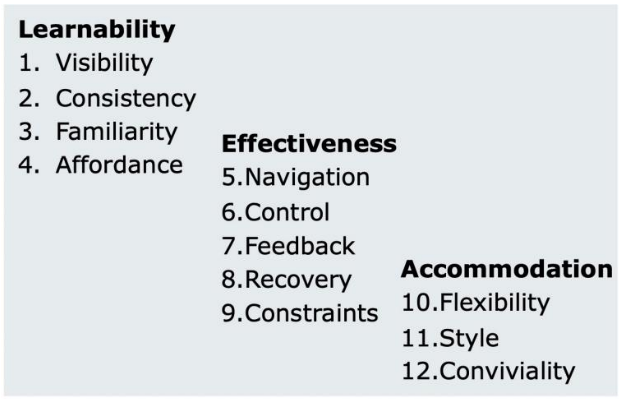


Figure 7, Criterion 'Flexible' implemented in the developed system

The last criterion out of the general criteria chosen to be identified is 'Comprehensible'. This explores the effort it takes to gain a general understanding of the system. In terms of our project, this is something that can be truly relevant in terms of users logging in and knowing the functions and possibilities they have but also how to execute said functions.

3.2.2 Relevant principles from Benyon's 12 design principles

When identifying the design criteria for the system's user interface, the group felt it was fitting and beneficial to also include Benyon's 12 design principles due to past experience and success with the inclusion of the principles. Design principles can actually guide designers throughout the design phase and can also be used to evaluate and analyze different design ideas and choices. According to Benyon, for memorization, the principles have been grouped into three categories – learnability, effectiveness and accommodation – but has been made clear of these groups not being rigid. "Systems should be learnable, effective and accommodating" (Benyon, 2019, p. 116).



Learnability
1. Visibility
2. Consistency
3. Familiarity
4. Affordance

Effectiveness
5. Navigation
6. Control
7. Feedback
8. Recovery
9. Constraints

Accommodation
10. Flexibility
11. Style
12. Conviviality

Figure 8, Benyon's 12 Design Principles

It is important to note that only the most relevant principles for the system's UI will be explored in further detail. The principles and its categories can be seen in figure 8, shown above. "Principles 1–4 are concerned with access, ease of learning and remembering, Principles 5–7 are concerned with ease of use and principles 8 and 9 with safety, and Principles 10–12 are concerned with accommodating differences between people and respecting those differences" (Benyon, 2019, p. 117).

Some of design principles recognized as relevant for the system's UI within this project included principles such as Visibility, Familiarity, Navigation, Feedback, Recovery and Flexibility. The group felt like the remaining principles, although important to acknowledge, were not relevant to the system's UI in terms of priority and applicability.

Starting off with 'Visibility'**,** it was important to acknowledge how important things being visible is for the user. In our case, we paid close attention to the structure of the system's UI making sure the functions and possibilities were made visible for the user to act on. This included experimenting with different colors, sizes and shapes but also the positioning of different content making it easier to see what was going on.

In the figure below, we can see the example of the principle 'Visibility' being implemented within the system through buttons and color. In this instance, we can clearly see the 'send' button on the map page which will help users report any changes seen on the map

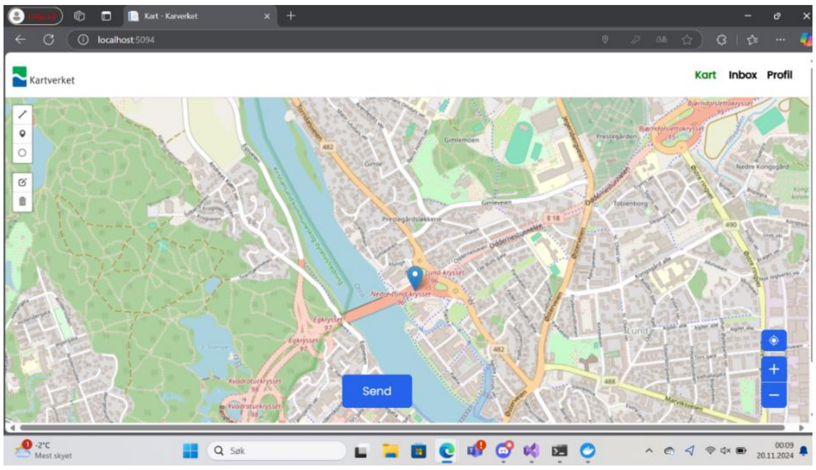


Figure 9, Principle ‘Visibility implemented in the developed system

The next principle worth exploring is ‘Familiarity’. This was something that the group felt was quite important as it was previously communicated through the product owners of having a system quite similar to the one already established. It was something the group kept in mind when producing different sketches of what the system would look like. According to Benyon, familiarity is also about using “language and symbols that the intended audience will be familiar with” (Benyon, 2019, p. 117). In this instance, we explored different symbols, colors and content structure Kartverket have used before. This included the organizations colors and symbols that are currently implemented on the current website and system.

In terms of ‘Navigation’, the principle explores the approach of providing support to help users move around within the system through maps, directions and indicators. This improves the overall user experience and gives the user a sense of control and confidence when on the system (Benyon, 2019, p.117). In our case, the inclusion of the having the breadcrumb stay green within the navbar to indicate where you are on the system and redirecting links highlights the implementation of this principle.

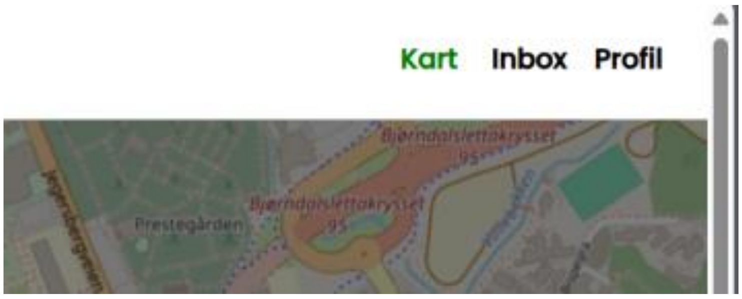
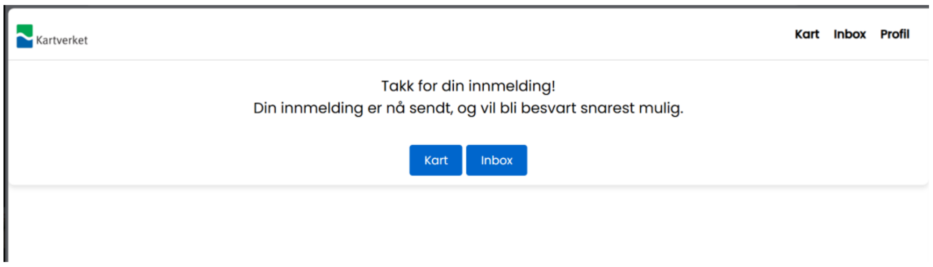


Figure 10, Principle ‘Navigation’ implemented in the developed system

According to Benyon, ‘Feedback’ explores the idea of rapid feedback information from the system to the user making it clear the effects the users actions have had. In our case, when thinking about possible designs for the system, screens with confirmation after confirming an action like reporting an issue was something well received within the group as it gives a clear indication to the user that their action has been received by the system. When looking at the impact the principle Feedback can have, we can see that “constant and consistent feedback will enhance the feeling of control” (Benyon, 2019, p.117).


Kartverket

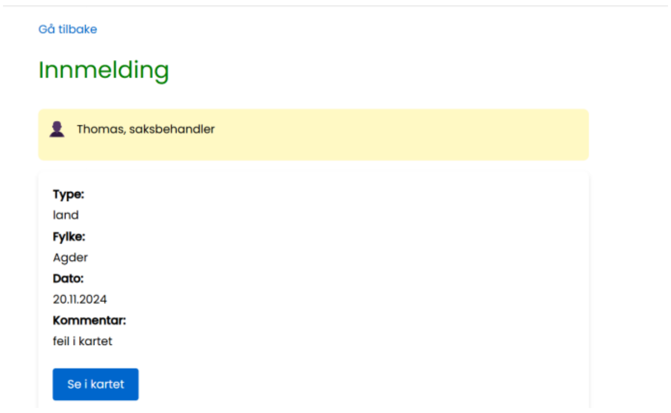
Kart Inbox Profil

Takk for din innmelding!
Din innmelding er nå sendt, og vil bli besvart snarest mulig.

Kart Inbox

Figure 11, Principle 'Feedback' implemented in the developed system

When taking a closer look, we can see 'Recovery' shares the idea of enabling "recovery from actions, particularly mistakes and errors, quickly and effectively" (Benyon, 2019, p.117). An example of this, being implemented in our designs, could be when the user is on inbox page and clicks an unseen message, the user then has the ability after reading the message to go back to the inbox page via a link seen at the top. This marked as blue to indicate to users of it being a redirecting link


Gå tilbake

Innmelding

Thomas, saksbehandler

Type:
land
Fylke:
Agder
Dato:
20.11.2024
Kommentar:
feil i kartet

Se i kartet

Figure 12, Principle 'Recovery' implemented in the developed system

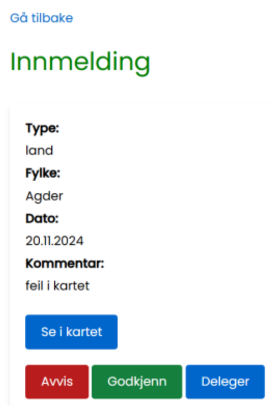
The last principle worth mentioning is 'Flexible'. This principle introduces the idea of letting users do something in a way that suits them. Although the group can see value in allowing multiple ways of doing things to accommodate different levels of expertise and experience, changing the way things look or behave to personalize the system is not something the group values to highly (Benyon, 2019, p.117).

3.2.3 UX laws

Alongside the criterion and principles mention earlier, the group also felt the inclusion of UX laws were perfect to include from past experiences with digital interaction design. The laws we felt suitable and applicable to the system's user interface was Fitt's law and Postel's law.

Starting off with Fitt's law, we can see the main ideas and themes in this case are about human movement. The law states "the time to acquire a target is a function of the distance to and size of the target" (Yablonski, 2023). The main takeaways from this law was touch targets should be big enough for users to accurately select them, there should be enough spacing between touch targets and finally that touch targets should be placed in places that allow for them to be easily used. This is something we felt was especially important, as a big part of the target audience are elderly people who maybe need the extra assistance when it comes to technology use. Having buttons visible and easy to press is something very key as it caters to our audience and improves the overall process in report in an issue founded (Yablonski, 2023).

In the figure below, we can see the clear use of the Fitt's law on how Case Managers handle reports.


Gå tilbake

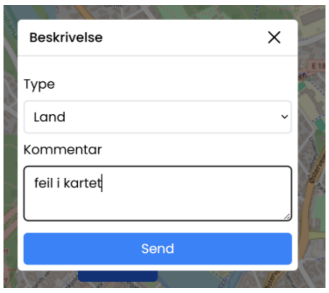
Innmelding

Type:
land
Fylke:
Agder
Dato:
20.11.2024
Kommentar:
feil i kartet

Figure 13, Fitt's law implemented within the developed system

Postel's law introduces us to the idea of being "liberal in what you accept, and conservative in what you send" (Yablonski, n.d.). What we can take away from this law is the ability to be empathetic, flexible and tolerant to the various actions and inputs a user may provide. In a practical sense, the system would have to anticipate pretty much anything in terms of inputs while providing a solid interface for the user to experience. This law especially stood out to us, as we have previously been informed by the product owners of having a system quite flexible and liberal in what it receives when it comes to reporting an issue. This is a main reason into including text boxes for users to fill out as it is much easier getting your point across then filling out rigid one worded answers (Yablonski, n.d.).

The figure below showcases the implementation of Postel's law through the text boxes mentioned above. In this instance, the text boxes allows for users to go more into detail on the issue at hand.


Beskrivelse

Type
Land

Kommentar
feil i kartet

Figure 14, Postel's law implemented within the developed system

3.2.4 Prioritizing table for the system's user interface

When prioritizing for the criterion chosen and what was most valued for the system's user interface while also taking into consideration the structure it would be presented in, we landed on having individual prioritizing tables for the criterion, principles and laws mentioned earlier.

When prioritizing what was important to us and the overall success of this project, we stumbled across three principles for a good design. The first being that it has no major weakness, the second being that it balances several criteria and the last being that it is usable, flexible, and comprehensible. Following the last principle for a good design, we can see in our prioritizing table, the criterion usable, flexible and comprehensible all being listed as 'very important'. According to Mathiassen, OOA&D "emphasizes three general design criteria, as expressed in the following principle: A good design is usable, flexible, and comprehensible" (Mathiassen, 2000, p. 179).

The first prioritizing table, seen underneath, involves the criterion usable, secure, correct, flexible and comprehensible. Criterion such as usable, secure, correct, flexible and comprehensible were all seen as ‘very important’ as having an interface that is both usable, secure, correct and comprehensible is vital considering the purpose of our system which is sending in errors and issues found by the public. Especially when we look at our target audience and regular user, it is imperative we deliver on these criterions alongside the criterion flexible and comprehensible. The other criterion reliable was put under as important as they were still important but just not on the very top of our list of priorities.

Criterion	Very Important	Important	Less Important	Irrelevant	Easily fulfilled
Usable	X				
Secure	X				
Correct	X				
Reliable		X			
Flexible	X				
Comprehensible	X				

Table 3, Prioritizing table using the general criteria

The next prioritizing table, seen underneath, involves Benyon’s 12 design principles with the principles visibility, familiarity, navigation, feedback, recovery and flexibility all being evaluated in terms of their importance. Out of the principles mentioned, Visibility, Navigation, Feedback and Recovery were set at ‘very important’. Beginning with Visibility, as mentioned before, the principle is seen by the group as crucial to achieve as it is critical for our users to see everything whether that is important feedback or buttons that can lead them to perform other actions which can lead us to another principle set in ‘very important’ which is navigation. This principle was also set in as ‘very important’ due to navigating within the system is one of the most important aspect in whether or not it is usable, which is mentioned earlier and also set at ‘very important’. Another principle set in ‘very important’ is Feedback, as this is one of the main point and purpose of the general system. When reporting changes and issues found, it is imperative that users who send their concerns in should receive feedback by the system making it clear the decisions they took in an amicable manner. The last principle set at ‘very important’ is Recovery, when considering our regular users can be seen to be on the elder side. When reporting in or dealing with issue sent in from both sides, it is important to have recovery in place to avoid or undo small mistakes or actions.

The principle which is seen as ‘important’ is Familiarity. The reason behind it being put to important is since the product owners have made it clear that it would be nice for the system to be similar to the existing system in place but is not necessary. We took this as something we find important to do but not at the expense of other principles which are listed in ‘very important’. Finally, can we see the last principle being under ‘less important’ which is Flexibility. Although it can be seen as a plus for users to have a plethora of ways to perform an action, it is not at the top of our priority list and is something we would look at end, considering the time and resources at hand.

Principles	Very Important	Important	Less Important	Irrelevant	Easily fulfilled
Visibility	X				
Familiarity		X			
Navigation	X				
Feedback	X				
Recovery	X				
Flexibility			X		

Table 4, Prioritizing table using Benyon's 12 design principles

Moving onto the last prioritizing table, seen underneath, we can detect that it involves UX laws, with the particular laws being Fitt's law and Postel's law. The first law, which is Fitt's law, was set under 'important' due to it being important for the system's touch target to be big enough for people to see and touch them, with a focus on spacing between all the touch targets as well. The next law, Postel's law, was set under 'very important', due to system's ability in being "liberal in what you accept, and conservative in what you send" something we found extremely important to achieve and something the product owners also wished to see (Yablonski, n.d.).

Laws	Very Important	Important	Less Important	Irrelevant	Easily fulfilled
Fitt's law		X			
Postel's law	X				

Table 5, Prioritizing table using UX laws

3.2.5 Initial Sketches and Diagrams

After dissecting and exploring the different criterion, principles and laws attached to the design criteria for the system's user interface, it can also be beneficial to expand on the initial sketches and diagrams created based on the groups knowledge and understanding of the project at the time but also the design criteria agreed upon.

Navigation Map

The first relevant diagram/model the group landed on creating was a navigation map. Navigation maps focus on how people move through a site or application. The aim of navigation maps is to learn and focus on how users will experience the site or application. Within a navigation map, a single box represents a single page in the site/application that can be accessed including lines and arrow to show the possible actions and directions that can be made. (Benyon, 2019, p. 191)

Below in figure 8, we can see an example of a navigation map that shows the possible 'turns and decisions' a regular user can take within the system. (See Appendix B2 for Internal Users)

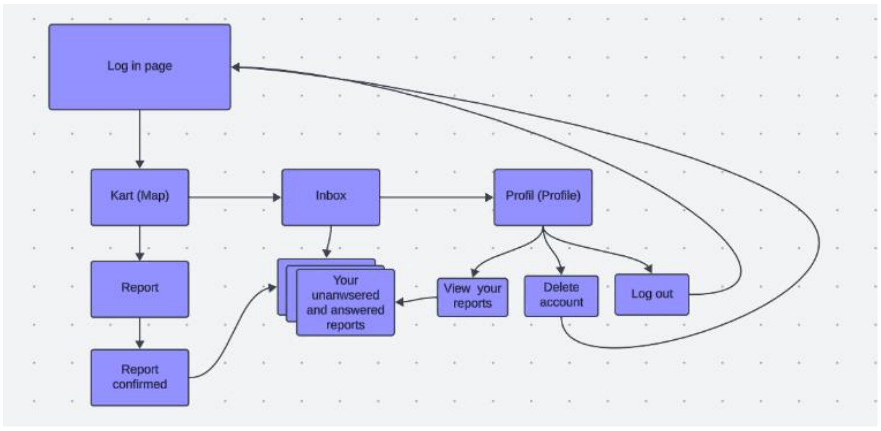


Figure 15 , Navigation Map for External Users

The group deemed navigation maps to be crucial in understanding how the system would be built and the possible turns a certain 'actor' could take. It gave the group a clear overview and functioned as blueprints for the wireframes.

Wireframes

This leads us to the next relevant diagram/model the group created which was the wireframes based on the navigation maps created earlier. Wireframes can be seen as "outlines of the structure of a software system" (Benyon, 2019, p.194). They tend to focus on the structure of particular types of pages and on the navigation between the different pages. The group chose wireframes as it was a great way to have something on paper that represented our ideas and suggestions that were made. Due to it requiring no color and other pieces of content that we were still not sure about yet, wireframes acted as a great solution giving us the freedom and flexibility to evaluate our wireframes in terms of their functionality and how it catered to our users and their experience, also known as usability.

Below in figure 9, can we see an example of a wireframe showcasing the group's idea for what the dashboard would look like for the case managers and how they would delegate and answer incoming reports. (See Appendix B3 for the remaining wireframes)

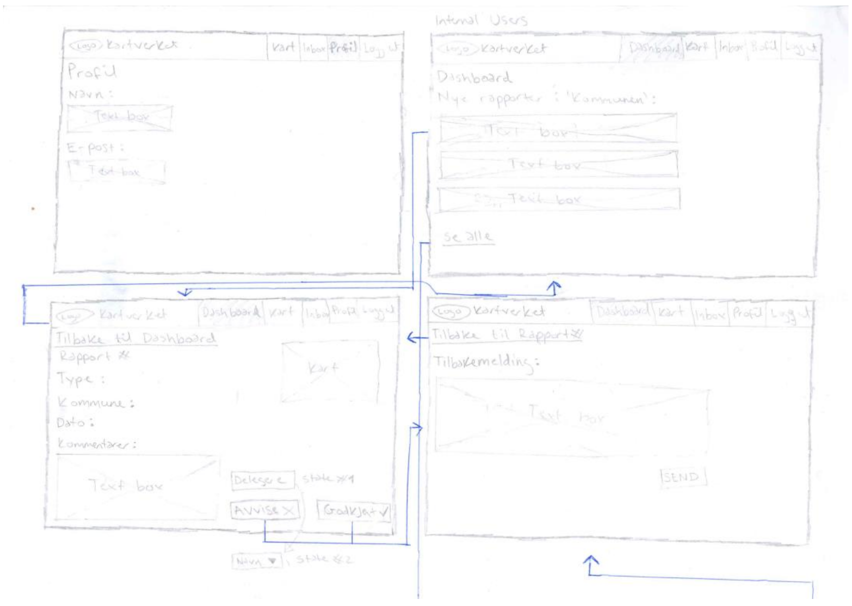


Figure 16, Wireframes for the Dashboard to Internal Users

3.3 Design Criteria for the architectural design

After exploring the different criterion, principles and laws attached to the design criteria for the system's user interface, it is now time to turn our attention to the relevant criterion and principles attached to the group's design criteria for the architectural design.

Before delving in, we can first mention the idea behind an architectural design. Something noticeable between various systems are that successful systems flourish and stand out compared to less successful systems due to strong architectural designs (Mathiassen, 2000, p. 173). An architectural design serves the purpose of structuring a computerized system, with IEEE defining architectural design as "the process of defining a collection of hardware and software components and their interfaces to establish the framework for the development of a computer system" (GeeksforGeeks, 2018).

3.3.1 Relevant criterion from the general criteria

Starting off, can we explore the relevant criterion, deemed by the group, referring back to figure 6 and the general criteria. The criterion that are relevant for the group's design criteria to the architectural design are efficient, maintainable, comprehensible, testable, reusable, portable and interoperable.

Criterion ‘Efficient’ introduces us to think about the “economical exploitation of the technical platform’s facilities” (Mathiassen, 2000, p. 178). When looking at the back-end of the system we are designing, it is important to keep in mind the efficiency it must obtain for it to be a success.

Criterion ‘Maintainable’ explores the idea of “the cost of locating and fixing system defects” (Mathiassen, 2000, p. 178). Having a back-end which is easy-maintenance but also always open to working on and further improving itself is an important aspect the group wishes to achieve.

Moving onto the criterion comprehensible, we can notice that this was mentioned earlier for the system’s user interface and can now be mentioned for the architectural design. Having the back-end of our system coherently understandable to read and go through is something we found as quite important as there are situations like new additions to the team or a takeover in ownership where having a comprehensible architectural structure really pays off.

The next criterion that stood out to the group was ‘Testable’, due to it featuring the idea of testing back our system to see if it performs the way expected. As a group, we came to the consensus that this can be seen as an important criterion as changes along the way are expected to be made and testing back our back-end is a great way to track progress and evaluate new concepts being introduced.

Criterion ‘Reusable’ states “the potential for using the system parts in other related systems” (Mathiassen, 2000, p. 178). This is something the group has considered when thinking about the potential and future of the outcome of this project and the impact the system can have. The system could have the potential to please the product owners in a way where including aspects of system to the already existing system can be, although minute, a possibility.

The criterion ‘Portable’ explores “the cost of moving the system to another technical platform” (Mathiassen, 2000, p. 178). Although not at the top of the priorities we have, it still is something to think about when looking at the architecture and structure of our system’s backend.

Looking at the criterion ‘Interoperable’, similar to the criterion reusable, shares the idea of connecting the system to another system. According to Mathiassen, this criterion measures the aspect of “the cost of coupling the system to other systems” (Mathiassen, 2000, p. 178).

3.3.2 Relevant principles from Benyon’s 12 design principles

Moving on, we can refer back to the Benyon’s 12 design principles and explore a few new principles which are more relevant and suited to the architectural design. Those principles being consistency, being categorized under learnability, and flexibility being categorized under accommodation.

The principle ‘Consistency’ shows us the importance of being consistent in the way of working, language, features, etc. According to Benyon, “Both conceptual and physical consistency are important” (Benyon, 2019, p. 117). In our situation, it is important to hold consistency within our backend to keep the structure organized for the benefits such as other people reviewing, maintaining, improving, connecting and moving the system as mentioned before.

The principle ‘Flexibility’ shares the idea of providing people with the chance of changing the way things look and behave. Within our context, having a backend open to change if a situation ever does occur can be a benefit and open the doors for new possibilities (Benyon, 2019, p. 117).

3.3.3 Prioritizing table for the architectural design

Moving onto prioritizing the architectural design, the first table prioritizes the criterion derived from the general criteria. The following criterion was evaluated and prioritized to what is most important to the project. The criterion being Efficient, Maintainable, Comprehensible, Reusable, Portable, Interoperable. We can first point out that Reusable, Portable and Interoperable has been put on the priority ‘Less important’ as they are things that would be seen as a bonus towards the architectural design. However, criterion like Efficient and Comprehensible are seen as more important as having an architectural design which is readable and understandable but also efficient are aspects the group values a little more. To end off, we can see in the prioritizing table shown below, the criterion Maintainable has been ticked off as ‘very important’ able due to us believing that a system’s backend must be able to maintain but also open to further work and improved being implemented.

Criterion	Very Important	Important	Less Important	Irrelevant	Easily fulfilled
Efficient		X			
Maintainable	X				
Comprehensible		X			
Reusable			X		
Testable	X				
Portable			X		
Interoperable			X		

Table 6, Prioritizing table for architectural design using the general criteria

Now, moving onto the second table prioritizing the architectural design, we can see the prioritized principles derived from Benyon’s 12 design principles. Starting off with the principle Consistency, the group landed on this being ‘very important’. When looking at the back-end side of the project, one thing to note is that it should be consistent with itself, making it easier for either the group to work further on it or somebody else to go through it and be understanding. Consistency is key! This leads us on to our last principle being prioritized which is Flexibility being set under less important. When discussing with our group how flexible we wish our back-end/architecture to be, it was always something that was nice to have and show but set further down on our priority list. Due to lack of clarity on the future and scope of this project, it is something we will look into to see if the right foundations and more prioritized aspects are set in stone.

Principles	Very Important	Important	Less Important	Irrelevant	Easily fulfilled
Consistency	X				
Flexibility			X		

Table 7, Prioritizing table for architectural design using Benyon's 12 design principles

3.3.4 Architectural Design Model

Deployment Diagram

After discussing the certain criterions and principles related to the architectural design, we can present the deployment diagram based on the architectural design which can be seen below in figure. The group found the deployment diagram to be fitting as it highlights the relevant components, dependencies, processors and objects attached to the architecture in a manner that is easy to read and understandable (Mathiassen, 2000, p. 340).

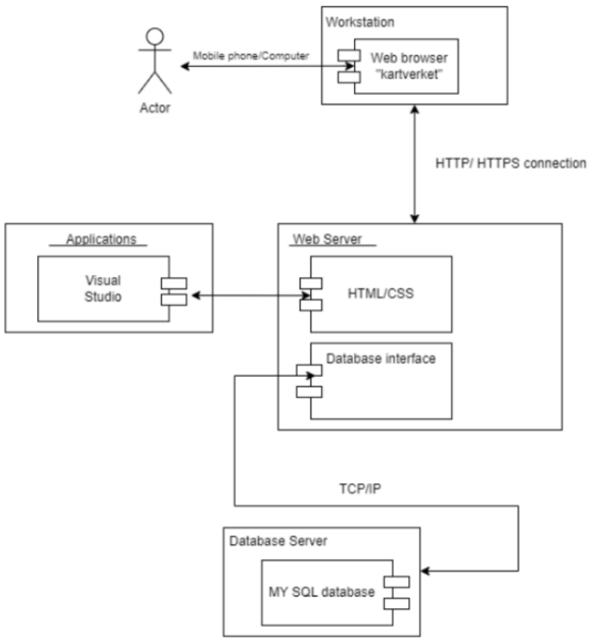


Figure 17, Deployment Diagram

In our program, we have a user interface, system interface, functions and databases. The website is the user interface, where the client interacts with visual elements to complete their tasks. The web server uses html and css as a styling sheet and a database interface in the visual studio application. API is our system interface that allows components to exchange data from the users and administrators.

The database used is the Maria database. The communication protocols are HTTP between the website and the webserver, and TCP/IP between the webserver and the Maria-database server. Their structure and relationships dictate how processes interact within the system. The architecture has a “model view controller” system structure.

MVC Model

Tackling the project at hand, it was vital for the group to adopt an architecture that would both support the project’s needs but also be compatible with the group and how we operate. The group landed on MVC model due to the requirements stated at the start of the project, but later found the advantages that comes with MVC. Reflecting back on this with hindsight, can the group come to the conclusion that MVC was a good fit for both the group and project.

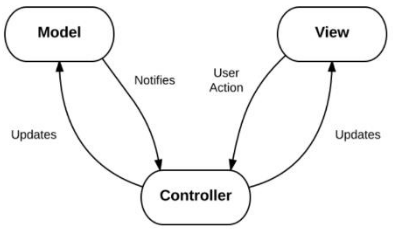


Figure 18, MVC Model (Interserver, 2016*)*

MVC, short for Model, View and Controller, is a framework relating to the structure of how code is written and the organization of program files. This framework organizes a system’s logic into three distinct layers which are Model, View and Controller. These layers carry out a specific set of tasks and communicate with one another to ensure the system's functionality. While the Model layer is responsible for maintaining all aspects of the data and ensuring its integrity and accessibility, the View is responsible for delivering the UI that allows users to interact with the application. Additionally, the Controller layer acts as an interface between the view and model layers by facilitating communications across the application (Sheldon, 2023).

With the introduction of MVC, the group found a plethora of advantages that came such as flexibility to have multiple group members contributing saving time and effort throughout the project. Another advantage found was the ability to divide work in a simple and more organized manner, which improved the efficiency of progress achieved on different components within the. A challenge the group faced when it came to adopting MVC was how easy the complexity could arise to, thus leading to the group using more time to comprehend the back-end than originally planned.

4. Scrum

When figuring of ways to approach a project, many different ideologies, methods, and frameworks pop up all having their own different advantages and disadvantages. The two main approaches can be narrowed down to the waterfall and agile project management. The waterfall project management approach follows a rigid sequence of phases, where a phase must be fully completed to advance onto the next phase. The agile project management approach is iterative focusing on continuous releases incorporating feedback and reflection. In today's climate with the need for quick responses and change required by clients and organizations, the agile approach allows for the flexibility to adapt and iterate within a project, something the waterfall approach struggles to match up with. A framework closely related to the agile approach and ideology is Scrum (Radigan, 2022).

What?

Scrum is an agile project management framework that can help aid groups and teams manage and structure their workload. Scrum can also be seen to encourage learning through experiences, working independently and/or in groups and reflecting over the good and the bad. It is also important to note that while scrum is heavily involved with agile, the two are vastly different where scrum is a framework for getting work done, while agile is just a philosophy (Drumond, 2024).

The name scrum derives from the sport 'rugby'. The term scrum is used within rugby, when a group works together to move the rugby ball forward. In the context of a project, Scrum is used for groups to work together in the aim to move a product forward (Scrum.org, 2020b).

Why?

In this instance, the Scrum framework perfectly serves a suitable and efficient approach to tackling the project at hand. The first point on why the Scrum framework and approach was chosen when embarking on this project was due to the flexibility and adaptability it provides to the group members. Due to the group being relevantly new when it comes to working on projects, its important for us to be able to adapt and make small changes and adjustments when needed without it having major impacts on the project and progress being made. This leads us to another point which is the continuous feedback received throughout the project. Scrum allows for reflection and reviews to take place, which in our case is crucial in figuring out if the group is on the right track or if changes are recommended.

Communication and transparency are enhanced when following Scrum due to the structure in place with the varied Scrum events to be followed but also the roles to be adopted, making sure everyone is aware of the progress being made throughout the project. Overall, Scrum provides a process that pushes group members in the right direction, in terms of how things should be planned, effort and time that is required, the group members involved and the deadlines that need to be met. This leads us to introducing the different roles, events and artefacts related to Scrum and there relevance within the project at hand.

4.1 Roles, Events and Artefacts

4.1.1 Roles

Within the Scrum framework, there are a variety of different roles all carrying different responsibilities and expectations. A Scrum team consists of: Product Owners, a Scrum Master and Team Members.

Roles	Description
Product Owner	The representative of the organization that communicate the needs and wants of the customers and the organization behind the project (West, 2019).
Scrum Master	The person that is responsible for making sure that progress is being made but also that Scrum is being performed well and correctly (West, 2019).
Development Team Members	Involves the personnel that do the work in terms of developing the final product (West, 2019).

Table 8, Scrum roles

Product Owner

The responsibility that lies within the role of being a product owner is ensuring good communication between them and the development team providing value on the wants and needs that are required (West, 2019). In our case, there are multiple product owners who we meet virtually to discuss new ideas and be informed on new updates concerning the system or any other relevant topics. The Q&As are every two weeks and is the perfect chance to gauge what needs to be prioritized and done.

Scrum Master

This role can be seen as the induvial holding it all together. The Scrum Master is constantly educating the rest of the team on Scrum, ensuring it is performed correctly. The role is also concerned with the progress that takes place, acting as a leader figure for the group. The role also helps in maintaining the structure and organization within the group (West, 2019). In our case, a Scrum Master was chosen at the start of the project, however due all group members being new to Scrum, it took some time for the Scrum Master to be confident in leading the group following this approach and teaching the other group members on how Scrum is meant to be performed.

Development Team Members

The development team involves the group members that do the work in terms of developing the wanted final product. The development team can be people with varied skills like writing, programming or even designers (West, 2019). Within the group, we all take the responsibility of being a part of the development of the final product, with all of us each having our own unique skills and experiences which can contribute to different aspects of the development phase and overall project.

4.1.2 Events

Within the Scrum Framework, there are also a variety of meetings and activities set in place to maintain the structure and progress of the project. The range of activities can stretch from planning all the way to reflecting, with these activities and meetings giving the group transparency and time to both be aware of the progress being made but also reflect and regroup when necessary. The events attached to Scrum include sprints, sprint planning, daily scrum, sprint reviews and sprint retrospectives.

Events	Description
Sprint	“short, timed boxed periods when a scrum team works to complete a set amount of work” (Rehkopf, 2019).
Sprint Planning	A meeting at the start of a sprint discussing what is expected to be worked on and completed within the sprint.
Daily Scrum	Daily short meetings performed throughout the sprint to share and be updated on any changes or progress that has been made (Cohn, 2019).
Sprint Review	Performed at the end of each sprint allocating time for the group and product owners to reflect over the progression made and direction the project is heading (Radigan, n.d.)
Sprint Retrospective	A meeting organized by the team to reflect over sprint, discussing the positives and negatives that occurred (Atlassian, n.d.-a).

Table 9, Scrum events

Sprint Planning

We can first begin by stating that as a group, we would meet up physically or digitally to start planning ahead for the sprint at hand, also known as Sprint Planning. This is where you communicate with the group for what is expected to be worked on and completed within the sprint. It is worth keeping in mind that at the end of every sprint, it is ideal to have a product ready to show for the effort and work that was put in, also known as an increment.

Within our sprint planning meetings, would we first discuss the tasks that were expected to be completed or begin with before the end of the sprint. We would then put estimations of how long we thought the task needed for it to be completed, exploring different techniques to help us out which is explored and reflected on later down this document. Having estimations on the different tasks helped us with prioritizing and organizing the workload while also keeping it balanced for everyone, so that there was no member overworking. Afterwards, we began delegating the different tasks based on skill, expertise, experience and also workload / availability.

Daily Scrum

At the start of the project, the ‘daily’ Scrum meetings were performed every other day which later was changed due to reflecting as a group and the feedback received which is explored in detail later on. The group’s daily Scrum meetings would consist of group members, individually or in groups, presenting the progress achieved on the tasks assigned. The meetings would end off with questions and discussions on if any changes are necessary or if priorities have changed.

Sprint Review

Sprint reviews allocates time where questions can be asked, new features and functions can be presented, and feedback can be given. The purpose and goal is to receive feedback and get a sense of clarity on what is expected moving forward in the development phase of the project (Radigan, n.d.). In our cases, the lecturers organized the sprint review meetings, with every sprint review having a different focus in terms of subject. During the sprint reviews, the group would give status updates on the different subjects, the positives and negatives met with that certain sprint and what could have been done differently. In Sprint review #2 and #3, the group was able to have a walk through the current application and show the proposed database structure.

Sprint Retrospective

We then have retrospectives which can be seen as similar and compared to sprint reviews however retrospectives are more concerned about the future scope and reflecting over what went well and not to gather information on what can be improved for next time. Referring back to Atlassian, “A retrospective is anytime your team reflects on the past to improve the future” (Atlassian, n.d.). The Scrum Master would always arrange a sprint retrospective the day before the sprint review to collect the ideas and feelings of the group in order to find out what went well and wrong during that specific sprint.

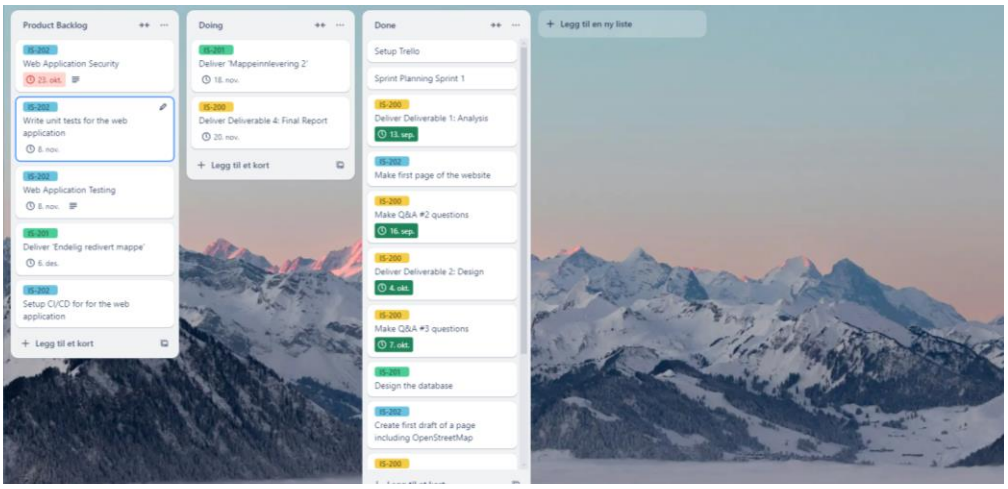
4.1.3 Artefacts

Another area to look into are the artefacts that come into play when performing Scrum. Artefacts are a vital aspect of Scrum as they provide an overview of the progress taking place and showcases to the product owner the current state of where the group lies in terms of development. With that being said, the artefacts relevant in the Scrum framework can be seen in an increment, the product backlog and sprint backlog.

Artefacts	Description
Product Backlog	A list of tasks, including their priority, to be completed during the project.
Sprint Backlog	A list of task, including their priority and estimations given by the group, to be completed during a sprint.
Increment	“a concrete stepping stone toward the Product Goal” (Scrum.org, 2020a)

Table 10, Scrum artefacts

The product backlog is a list of the various features, functions, requirements and changes that are necessary for the project. In addition, to providing a list of the various tasks needed to be completed, the product back log also helps prioritize and structure the various tasks at hand making the tasks more manageable and easier to tackle incrementally (Atlassian, n.d.-b). Below is the product backlog to the group created on Trello. The group found Trello as a great tool to have an overview on the tasks at hand, their status whether in progress or done, but also the group members responsible for each task.



Product Backlog

IS-202
Web Application Security
23. okt.

IS-202
Write unit tests for the web application
8. nov.

IS-202
Web Application Testing
8. nov.

IS-201
Deliver 'Endelig redivert mappe'
6. des.

IS-202
Setup CI/CD for for the web application

Doing

IS-201
Deliver 'Mappeinnlevering 2'
18. nov.

IS-200
Deliver Deliverable 4: Final Report
20. nov.

Done

Setup Trello

Sprint Planning Sprint 1

IS-200
Deliver Deliverable 1: Analysis
13. sep.

IS-202
Make first page of the website

IS-200
Make Q&A #2 questions
16. sep.

IS-200
Deliver Deliverable 2: Design
4. okt.

IS-200
Make Q&A #3 questions
7. okt.

IS-201
Design the database

IS-202
Create first draft of a page including OpenStreetMap

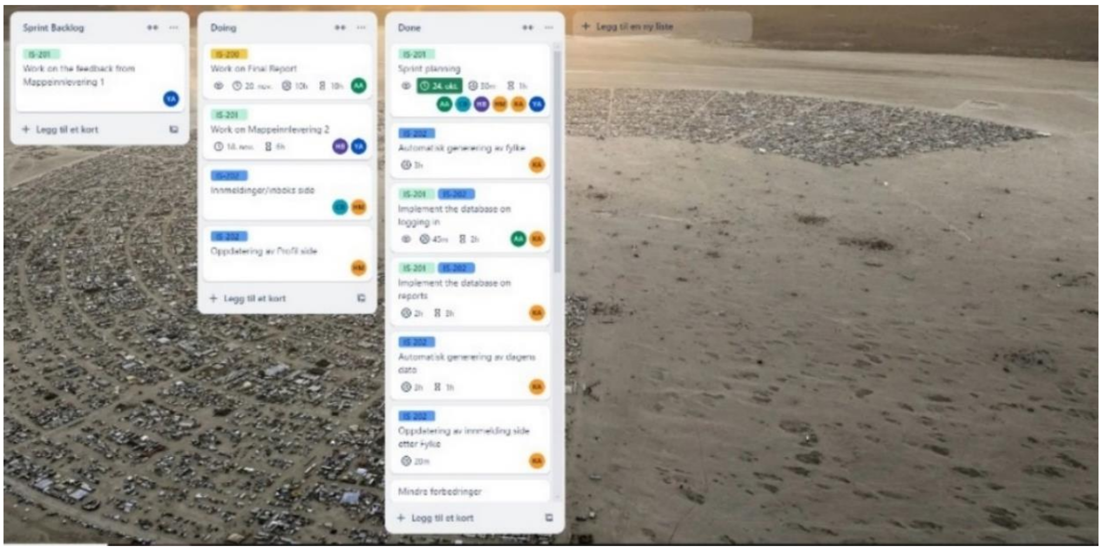
IS-200

Figure 19, Trello Board for the Product Backlog

The purpose of the sprint backlog is to break down the tasks featured in the product backlog to smaller and more manageable tasks to tackle and complete. The sprint backlog get discussed and created at the start of every sprint in the sprint planning meeting. During the daily Scrum meetings, these tasks get mentioned with the main goal of informing the others on the progress made (Atlassian, n.d.-b).

To compare and contrast, the product backlog is dynamic and flexible, ready to adapt and change based on the evolving needs and wishes of the product owners and organization. On the other hand, the sprint backlog can be seen as fixed with the purpose of allowing the group to work which restricts any distractions or focus on other areas (Atlassian, n.d.-b).

Down below, figure 13 is an example of the sprint backlog which was taken from Sprint 4. An important aspect included in the group's Trello boards are the labels attached to different tasks. Yellow represents tasks related to IS-200, green represents tasks related to IS-201 and blue represents tasks related to IS-202.


Sprint Backlog
- IS-201 Work on the feedback from Mappeinnlevering 1 (YA)
- + Legg til et kort

Doing
- IS-200 Work on Final Report — 20. nov. — 10h — 10h (AA)
- IS-201 Work on Mappeinnlevering 2 — 14. nov. — 6h (HB, YA)
- IS-202 Innmeldinger/inboks side (CR, HM)
- IS-202 Oppdatering av Profil side (HM)
- + Legg til et kort

Done
- IS-201 Sprint planning — 24. okt. — 30m — 1h (AA, CR, HB, HM, KA, YA)
- IS-202 Automatisk generering av fylke — 3h (KA)
- IS-201 IS-202 Implement the database on logging in — 45m — 2h (AA, KA)
- IS-201 IS-202 Implement the database on reports — 2h — 2h (KA)
- IS-202 Automatisk generering av dagens dato — 2h — 1h (KA)
- IS-202 Oppdatering av innmelding side etter fylke — 20m (KA)
- Mindre forbedringer
- + Legg til et kort

+ Legg til en ny liste

Figure 20, Trello board of Sprint

The idea of an increment can be related to another idea which focuses on having a product ready to show at the end of every sprint showing the effort and progress made. By adding an increment on top of the already existing increment from the previous sprint ensures progress being made towards the final product or in the case of the group, a digital system.

4.2 Changes in Practice

After going through the relevant roles, events and artefacts related to Scrum, its now time to delve into the different choices the group made in terms of changing the way the group practiced Scrum in hopes for better results throughout the different sprints.

4.2.1 Estimating and Power-Ups

The first change made when practicing Scrum was estimating how long different tasks would take when discussing the sprint backlog. Prior to the change being made, time estimates would be set by the group members who were responsible for that specific task. This would prove to be costly as, the time estimates being set would be much shorter than the task actually required, which led to other tasks not being completed on time, delaying the overall progress of the sprint.

During the first sprint review meeting and sprint retrospective, it was reflected and decided by the group to implement the poker technique when estimating the time for the different tasks to be completed throughout the sprint. The poker technique is performed by meeting up as a group and going through the tasks that need to be completed within the sprint. A task will be presented and then each group member selects a card that represents a time, with each member getting a chance to argue for why they chose that certain card and time. After going through the different estimates and the arguments behind them, the group gets closer to a consensus which better represents what the group thinks of each task, how long it will take and the priority it holds (Obergfell, 2011).

Another point added during the sprint review meeting in sprint 1 was the inclusion of a Trello power-up that acted as a time tracker to properly monitor the amount of time a single task takes while also including the estimated times. This was necessary as the group had no overview on the different estimations assigned to the different tasks featured on the sprint backlog. It can also be used as a learning tool for the group to compare and find out which tasks took longer than others. This also influences how the group prioritize different tasks but also how we give estimations using the experiences we now have.

This led to the group exploring different Trello power ups that could be implemented to serve our need. Trello power ups are applications that add more functionality to Trello and our board, allowing for extra automation. The introduction of third-party tools and also creating custom reports gave us more freedom and creativity in the way we structured and planned our work, optimizing efficient work completed by the group in Trello (Ivory, 2024). The power up the group landed on, to include both the amount of time the group estimated and actually worked, was Time tracking Power-Up by Everhour.

The inclusion of the poker technique and the Trello power-up helped the group better gauge what each sprint required during sprint planning. Although the group did not estimate everything accurately, it was much closer after the implementation of the poker technique. This allowed the group to complete more tasks and have more time to focus on different areas. One example can be in sprint 4 and 5, where all the group members could focus more time and energy into programming due to better time estimations and prioritization of the other tasks.

4.2.2 Daily Scrum

Moving onto the next change the group made when practicing Scrum, which is the daily Scrum meetings. The first change concerning the daily Scrum meetings were made in Sprint 2 after the group reflected that every three to four days was not enough, which led to the daily scrum meetings being performed every other day. However, during sprint review #2 in sprint 3, it was made clear that the group could easily fall behind in terms of progress being made if not careful. This meant focusing on the more important aspects and the bigger picture rather than small details. In the sprint retrospective, the group conveyed the feeling of being lethargic and stuck in terms of motivation. This led to the group agreeing to perform the scrum meetings daily instead of every other day. Gauging from the positive improvements the group felt when changing how 'daily scrum' was performed the first, it was only logical to change it to daily meetings, in hopes for even better improvements.

Performing daily scrum everyday improved the group's communication and maintained the motivation to work and progress. The daily meetings also ensured everyone was on top of their assigned tasks and responsibilities. After the wake-up call that occurred in the sprint review meeting in sprint 3, the daily scrum meeting helped the group get back on track and maintain the progress that was being achieved.

4.2.3 Scrum Master

The last noticeable change implemented by the group was the using the role of the Scrum Master to its fullest potential. Also, during sprint review #2 in sprint 3, it was made clear the Scrum Master role had to be further enforced and taken seriously. Although the Scrum Master would arrange the Scrum events and follow up on the progress being made and how Scrum was being followed, it was easy for the role to be neglected as practically everyone in the group had to act as the development team. This meant Scrum Master had to juggle two roles weakening the significance of the role of Scrum Master.

Due to the anticipated hectic period of the last two sprints, the role of the Scrum Master had to be enforced more, in hopes of pushing the group to newer heights and receiving more updates on how the group is performing. The increase in significance within the role of the Scrum Master coincided with the increase in daily scrum meetings which proved to be enough when it came to rekindling the ambition within the group and improving efficiency with the remaining time of the project.

4.3 Reflection over Scrum

To reflect of the use of Scrum and how it affected the group and overall project, it can be beneficial to explore the positives and challenges surrounding Scrum while also looking at what the group could have done differently.

4.3.1 Benefits using Scrum

The group experienced many positives when implementing Scrum towards this projects, with the first example being communication. The inclusion of the various scrum events helped improved the communication between the group member but also with other individuals such as the product owners, lecturers, and teaching assistants. It also helped the group being on the same wavelength, with the constant updates and planning meetings which if done correctly can lead to harmonious work needed to succeed in any project.

Another positive is the structure and guidance it can give to a group feeling unorganized and chaotic when working. Scrum gave the group, a sort of blueprint structuring the workload and progression that was expected. With Scrum providing sprints and the different meetings, its forced the group to be structured in the way we approached different aspects of the project.

The last positive worth mentioning is the sprint review and retrospective meetings and what they can offer throughout the project. Implementing Scrum to a project allows for time to reflect and reevaluate the progression being made. When looking back at sprint review #2 in sprint three, the group gained valuable and crucial feedback which was much needed to both know what was expected from us, but which also woke us up from staleness and lethargic feeling we had towards the project which can naturally build up after some time. It was the perfect refreshing feeling for the group to be more motivated and gain tunnel vision on the goal set by both the group and the product owners.

4.3.2 Challenges met using Scrum

Some challenges the group faced when implementing Scrum towards this projects was firstly giving tasks an estimation of how long it will take. Especially in the earlier stages of the project, the group struggled in estimating suitable times for tasks to be completed which led to failure of completing other tasks. The second challenge the group faced was the constant need for motivation from the group members. For Scrum to succeed, group members have to have the motivation to be an active member in terms of attending the varied scrum meetings.

Another challenge experienced during this project was the different roles attached to Scrum. What the group experienced was how the Scrum Master was forced to also adopt the role of a development team member due to the workload that featured during this project. It also took a while for the Scrum Master to live up to the role’s responsibility as all the group members was learning about Scrum at the same time with no previous experiences, meaning it was difficult for the Scrum Master to instruct the others early on in how Scrum is meant to be performed.

4.3.3 Aspects that could have been done differently

Aspects the group would have done differently was implementing the poker technique and Trello power-up earlier which would have prevented the group failing to completed tasks in the earlier sprints and also not having a clear overview on the estimations given to each task. Keeping the morale and motivation high throughout the group was another aspect which could have been managed a little differently, such as arranging small group breaks for the group to hang out and not focus on the project, avoiding being drained from the project at hand but also leading to working better as a team.

5. Ethical considerations

During this project, another important aspect the group valued and focused heavily on was the ethical considerations the group must take into account when developing the application. These ethical considerations can be split into two different categories: the front-end and back-end. The front-end relating to the application being user friendly and inclusive for all, while the back-end relates to the data and security that supports the application.

5.1 Universal Design

Universal Design (UD), also known as Design for All, is an approach to designing for accessibility. This means that the application being developed can be used by all, regardless of age, size, digital competency, or disability. The principles of UD includes equitable use, flexibility in use, simple/intuitive use, perceptible information, tolerance for errors, low physical effort and lastly size and space for approach and use (Benyon, 2019, p.105).

The group also took into account the Web Content Accessibility Guidelines (WCAG) which sheds a light on four different areas, ensuring an application is usable for everyone, which includes: Perceivability, Operability, Understandability and Robustness (Wolfe, 2024).

Through following the UD principles and WCAG, the group has gained a better outlook on how to cater for everybody needs and create an application that is usable for all. This led to discussion on the application's target audience and how the group can cater to their needs. The first example was implementing larger buttons to be more accessible and clearer touch targets (See figure 13). This caters to the older users and the users who may have weaker eyesight. Another point was the feedback a user would receive after reporting an issue on the map, helping users acknowledge that their action has been received and performed (See figure 11).

One of the requirements related to WCAG was designing for different viewport sizes (see figure 7). This was especially important due to Kartverket wishing for the application to be accessible and usable on multiple devices like a phone or a computer. This was something the group focused on in terms of how page information would look on a smaller screen.

Due to the application being available for the public to report any concerns or issues met, it was important that the application would be accessible, user-friendly, and easy to learn. The principles of UD and WCAG play a huge role in ensuring no individual is forgotten about and maintains good ethics when it comes to developing applications in today's world for everyone to use and enjoy.

5.2 Data and Security

Another ethical aspect to consider when working on a project, especially in the group's case, is the data that is collected and how secure the system is. The first point to explore is data received by the users registering and logging on to the system. This includes providing personal information such as first and last name, email address and a password. The group decided on not requiring more personal data such as phone numbers or ID, as it would put people off from registering and using the group's system. According to OECD, there is a sense of hesitance from the public when it comes to sharing personal information, with one in five people in Norway not fully trusting authorities with their personal data (OECD, 2022). In the US, around seventy percent of adults say their personal data is less secure than it was five years ago (Auxier et al., 2019). Taking in reports and numbers presented, the group came to the conclusion that the registering on the system should not require any unnecessary personal information for the user to provide.

The other aspect when it comes to security is authorization when it comes to different users and what they can access. Only admins obtain the ability of seeing an overview over all the users who are registered to the system. Admins also obtain the role of assigning the role of the Case Manager, whether that being creating a new account or switching an already existing user's role. This ensures that personal information from our users are seen only by admin out of necessity. It also ensures that our user's reports are overseen by verified users who are assigned to be Case Managers through the admins.

6. Conclusion

6.1 Analysis

The beginning of this project kicked off with the analysis phase which included exploring and understanding the problem and application domain. During this phase, the group started slowly in terms of fully understanding the problem at stake and what a suitable solution would look like. After the first Q&A and more documentation over the requirements and overall context of the situation, the group had a clearer idea on what the intended system would look like and what functions to include.

The first aspect of the analysis phase to be identified was a system definition using FACTOR. After going through various drafts and a variety of perspectives and directions, the group landed on a system "that allows for reporting issues founded on the map and for case managers to view these issues and respond accordingly."

After defining the system definition, the group came together with a joint effort to create a rich picture. This saved time during the analysis phase to focus on other models/diagrams and made the group understand together the context of this project and the unique situation and users who are involved.

After having a better idea on the problem at hand, each member created two different user stories following the two different formats ensuring all perspectives and possible functions were kept in mind. The group quickly found the relevance and value of the different user story formats as they provided the perspective of different users and functions, which was something the group struggled to see during the earlier stages of this project.

The group landed on using the MoSCoW framework in terms of prioritizing the different functions apparent in the various user stories created among the group. The group found this to be positive, as it was a logical way to distinct the various functions and user stories listed and helped in shaping the priority list when developing the system.

Lastly, there were various tables and diagrams adhering both to the problem and application domain analysis that were created in order to better improve the group understanding of the project at hand. This included the likes of an event table, class diagram, functions list, state chart and use-case diagram. These tables and diagrams helped the group get a better understanding of the classes, structure, behavior, usage, and functions attached to the system which created for a nice foundation when beginning the design phase.

6.2 Design

After going through the analysis phase of this project, which was challenging at times, it was now time to begin the design phase. This was divided into two different areas of focus with the first area being the system's user interface and the other area focusing on the architectural design. Before diving into both areas of focus, the group identified the special conditions that were unique to the group and project.

Due to our experience with the previous subject 'IS-104 Digital Interaction Design', creating the different figures and sketches felt smooth and vital in understanding what the group valued and wished to incorporate. Our initial sketches (see in Appendix B1) was crucial for the group to brainstorm ideas and build upon each other. In hindsight, creating the wireframes and navigation maps felt crucial in understanding the depth of our system earlier and understanding how it would looked (see in Appendix B2 & B3). The creation of a high-fidelity prototype also saved time when developing the system, as group members would just refer to the prototype and not be distracted by having to think how the system should look (see in Appendix B4).

During the design phase, the group also focused on the design criteria and what the group valued for the user. When prioritizing for the system's user interface, the group used the general criteria, Benyon's 12 design principles and UX laws due to past experience and how they each cover different areas to consider when creating a usable and enjoyable system. The criterion, principles and laws were all prioritized from 'Very Important to Irrelevant' within their own table.

The group also focused on prioritizing the architectural design, ensuring a certain level of quality the back-end of the system would provide. Due to a lack of experience, when it came to back-end architecture, it was difficult to find what the group valued and what was suitable for the system we were developing. Ultimately, the group found important criterion and principles which were also prioritized from 'Very Important to Irrelevant' within their own table. With the focus being on MVC and its structure, it felt only natural for the group to adopt that approach and structure when it came to programming and developing the system within Visual Studio.

6.3 Use of Scrum

Due to Scrum being new and unknown to most of the group members, the use of scrum to its fullest potential varied from sprint to sprint. During the earlier sprints at the start of the project, the group's approach was less effective due to not maximizing the use of the roles and events that Scrum provides. The Scrum framework was perfect for our situation, as it allowed for the group to reflect and make adjustments along the way from the lessons and experiences being learnt from.

The sprint review and retrospective meetings were crucial for the group in terms of reflecting over how the sprint went and the improvements that could be made for the next sprint. Receiving feedback was another important aspect, as it helped the group be aware on if we were heading towards the right direction. When reflecting over sprint review #2, we can see this as being a particularly crucial time for the group during the project, as it was a major wake up call for the group to not fall behind in terms of the development of the system.

The Scrum framework gave the group a new perspective on group work and how vital communication is when tackling a project. The daily scrum meetings were a particular aspect of Scrum the group found success in, with the daily meetings improving the group's communication leading to more efficient work being produced. It also led to transparency within the group, with all group members knowing each other's responsibilities and tasks, which played a role in the group's improvement in working together effectively.

Implementing the poker technique to our sprint planning meetings and including the time-tracking Trello power-up to our Trello board would have aided the group in giving better time estimations to tasks and had a better overview on the different estimations assigned to the different tasks. Having the estimations available and comparing it with the time it actually took would have given the group the necessary information on the difficulty of the different influencing how they were prioritized.

Although not all of the tasks present in the backlog were finished, the tasks given a 'Must have' priority, derived from the MoSCoW framework, were completed and implemented in the fully functioning application. The application provides the ability for the public to report an issue founded that does not correspond with the map. They are able to follow the status and receive a comment on the report sent in. The case managers working for Kartverket are able to have access to all reports, respond to these reports and give comments, and to delegate the reports to other case managers.

Summary

To summarize, the group explored both the problem and application domain during the analysis phase exploring different diagrams and tables which helped the group gain a better understanding of project and problem at hand. This led to the group then identifying and prioritizing the design criteria for both the system's user interface and architectural design, with the main takeaway being that the system should be inclusive and easy to use for everyone. The report then delves into the group's experience over the use of Scrum, reflecting over the advantages and disadvantages. This also includes changes made during the project and its impact on the group. To end off, the group took into account the different ethical considerations one must take when contributing to a project of this size.

References

Atlassian. (n.d.-a). *Agile retrospectives: Use the past to define the future*. Retrieved 9. October 2024 from: https://www.atlassian.com/agile/scrum/retrospectives

Atlassian. (n.d.-b). *Product Backlog vs. Sprint Backlog: Key Differences*. Retrieved 29. October 2024 from: https://www.atlassian.com/agile/project-management/sprint-backlog-product-backlog

Auxier, B., Rainie, L., Anderson, M., Perrin, A., Kumar, M., & Turner, E. (2019, November 15). *Americans and privacy: Concerned, confused and feeling lack of control over their personal information*. Pew Research Center. Retrieved 12. November 2024 from: https://www.pewresearch.org/internet/2019/11/15/americans-and-privacy-concerned-confused-and-feeling-lack-of-control-over-their-personal-information/

Benyon, D. (2019). *Designing user experience : a guide to HCI, UX and interaction design* (4th ed.). Pearson Education Limited.

Cambridge Dictionary. (n.d.). *ANALYSIS | meaning in the Cambridge English Dictionary*. Retrieved 20. September 2024 from: https://dictionary.cambridge.org/dictionary/english/analysis

Cambridge Dictionary. (n.d.). *DESIGN | meaning in the Cambridge English Dictionary*. Retrieved 27. October 2024 from: https://dictionary.cambridge.org/dictionary/english/design

Churchville, F., & Hashemi-Pour, C. (2021). *What is user interface (UI)?* TechTarget. Retrieved 27. October 2024 from: https://www.techtarget.com/searchapparchitecture/definition/user-interface-UI

Cohn, M. (2019). *The Daily Scrum Meeting*. Mountain Goat Software. Retrieved 8. October 2024 from: https://www.mountaingoatsoftware.com/agile/scrum/meetings/daily-scrum

Drumond, C. (2024). *What is scrum and how to get started*. Atlassian. Retrieved 08. October 2024 from: https://www.atlassian.com/agile/scrum

Interserver. (2016). *What Is MVC? Advantages and Disadvantages of MVC*. Retrieved 19. October 2024 from: https://www.interserver.net/tips/kb/mvc-advantages-disadvantages-mvc/

GeeksforGeeks. (2018, July 23). *Software Engineering | Architectural Design*. Retrieved 28. October 2024 from: https://www.geeksforgeeks.org/software-engineering-architectural-design/

GeeksforGeeks. (2020, April 7). *Object-Oriented Analysis and Design (OOAD)*. Retrieved 26. October from: https://www.geeksforgeeks.org/object-oriented-analysis-and-design/#what-is-objectoriented-analysis-and-designooad

Hausvik, G. I., & Wass, S. (2024, August 26). *Lecture 3 - Analysis I*. Universitetet i Agder.

Ivory, D. (2024, July 3). *What are Trello Power-Ups, and why should you use them?* Kolekti. Retrieved 11. October 2024 from: https://www.kolekti.com/resources/blog/trello-power-ups-and-why-you-should-use-them

Mathiassen, L. (2000). *Object-oriented Analysis & Design* (1. Edition). Marko, 2000.

Obergfell, Y. (2011, December 1). *Scrum Effort Estimations – Planning Poker*. International Scrum Institute^{TM}. Retrieved 10. October 2024 from: https://www.scrum-institute.org/Effort_Estimations_Planning_Poker.php

OECD (2022, March 15). *Drivers of Trust in Public Institutions in Norway*. OECD Publishing, Paris. Retrieved 12. November 2024 from: https://doi.org/10.1787/81b01318-en

Radigan, D. (n.d.). *Three Steps To Better Sprint Reviews*. Atlassian. Retrieved 9. October 2024 from: https://www.atlassian.com/agile/scrum/sprint-reviews

Radigan, D. (2022). *Agile vs. waterfall project management*. Atlassian. Retrieved 29. October 2024 from: https://www.atlassian.com/agile/project-management/project-management-intro

Rehkopf, M. (2019). *Scrum Sprints*. Atlassian. Retrieved 8. October 2024 from: https://www.atlassian.com/agile/scrum/sprints

Scrum.org. (2020a). *What is an Increment in Scrum?* Retrieved 29. October 2024 from: https://www.scrum.org/resources/what-is-an-increment

Scrum.org. (2020b). *What is Scrum?* Retrieved 8. October 2024 from: https://www.scrum.org/resources/what-scrum-module

Sheldon, R. (2023, September). *What Is model-view-controller (MVC)?* Techtarget.com. Retrieved 18. October 2024 from: https://www.techtarget.com/whatis/definition/model-view-controller-MVC

Visual Paradigm Guides. (2023, March 22). *Introduction to State Diagrams: A Comprehensive Guide for Software Engineering*. Retrieved 6. September 2024 from: https://guides.visual-paradigm.com/introduction-to-state-diagrams-a-comprehensive-guide-for-software-engineering/

W3C. (2019, December 6). *Tips for Getting Started Designing for Web Accessibility*. Web Accessibility Initiative (WAI). Retrieved 6. November 2024 from: https://www.w3.org/WAI/tips/designing/

West, D. (2019). *Agile Scrum Roles*. Atlassian. Retrieved 09. October 2024 from: https://www.atlassian.com/agile/scrum/roles

Wolfe, A. (2024, August 8). *LibGuides: Accessibility Toolkit for Open Educational Resources (OER): Accessibility Principles*. Guides.cuny.edu. Retrieved 7. November 2024 from: https://guides.cuny.edu/accessibility/whyitmatters

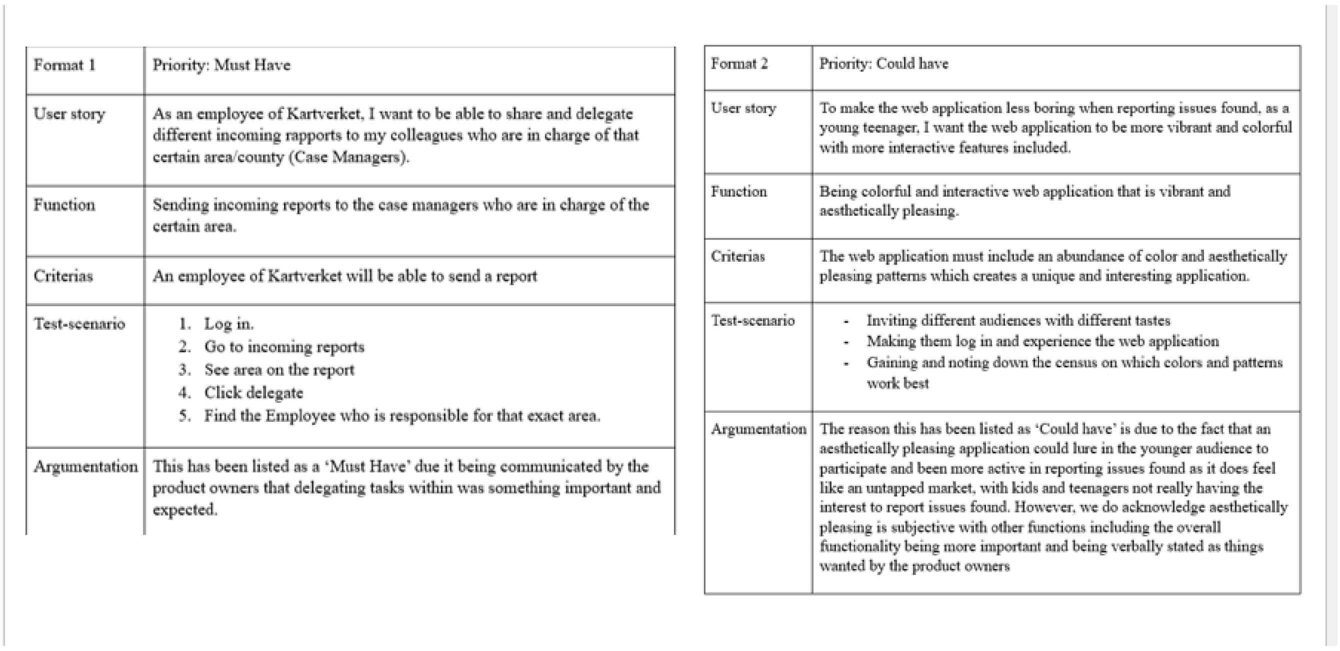
Yablonski, J. (n.d.). *Postel's Law*. Laws of UX. Retrieved 4. October 2024 from: https://lawsofux.com/postels-law/

Yablonski, J. (2023). *Fitts's Law*. Laws of UX. Retrieved 4. October 2024 from: https://lawsofux.com/fittss-law/

Appendix

Appendix A

A1 Examples of other user stories

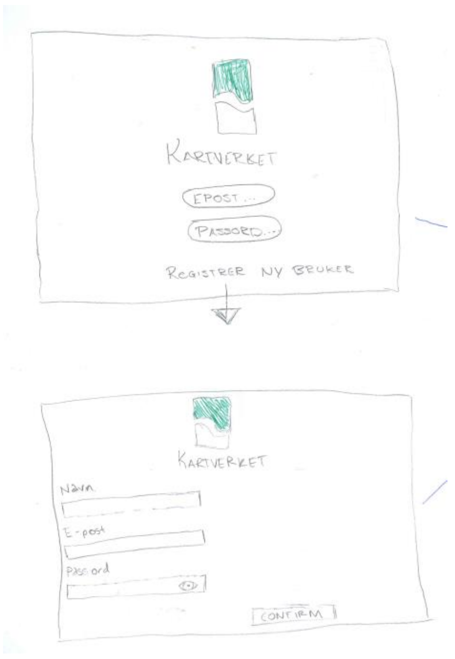


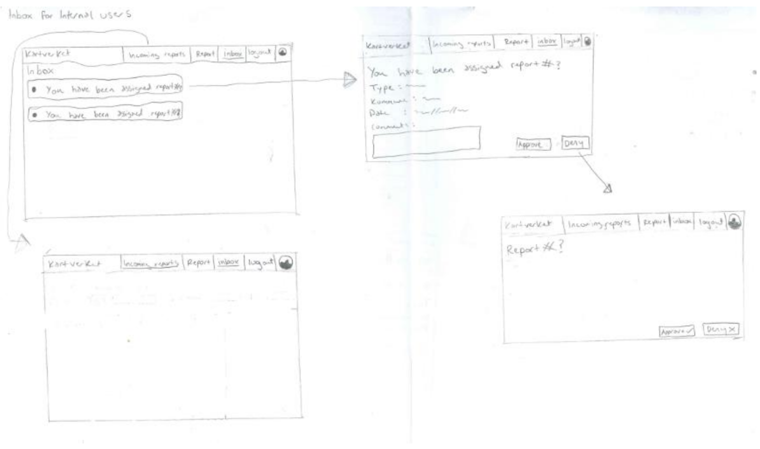
Format 1	Priority: Must Have
User story	As an employee of Kartverket, I want to be able to share and delegate different incoming rapports to my colleagues who are in charge of that certain area/county (Case Managers).
Function	Sending incoming reports to the case managers who are in charge of the certain area.
Criterias	An employee of Kartverket will be able to send a report
Test-scenario	1. Log in. 2. Go to incoming reports 3. See area on the report 4. Click delegate 5. Find the Employee who is responsible for that exact area.
Argumentation	This has been listed as a 'Must Have' due it being communicated by the product owners that delegating tasks within was something important and expected.

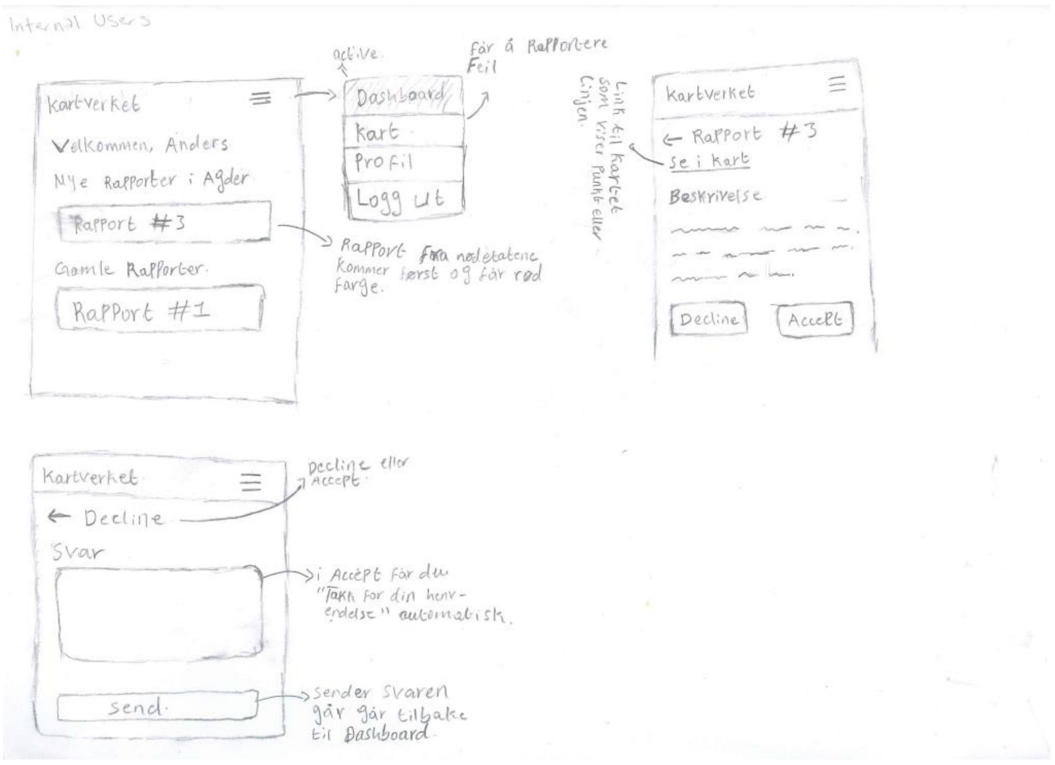
Format 2	Priority: Could have
User story	To make the web application less boring when reporting issues found, as a young teenager, I want the web application to be more vibrant and colorful with more interactive features included.
Function	Being colorful and interactive web application that is vibrant and aesthetically pleasing.
Criterias	The web application must include an abundance of color and aesthetically pleasing patterns which creates a unique and interesting application.
Test-scenario	- Inviting different audiences with different tastes - Making them log in and experience the web application - Gaining and noting down the census on which colors and patterns work best
Argumentation	The reason this has been listed as 'Could have' is due to the fact that an aesthetically pleasing application could lure in the younger audience to participate and been more active in reporting issues found as it does feel like an untapped market, with kids and teenagers not really having the interest to report issues found. However, we do acknowledge aesthetically pleasing is subjective with other functions including the overall functionality being more important and being verbally stated as things wanted by the product owners

Appendix B

B1 Initial Designs/Sketches

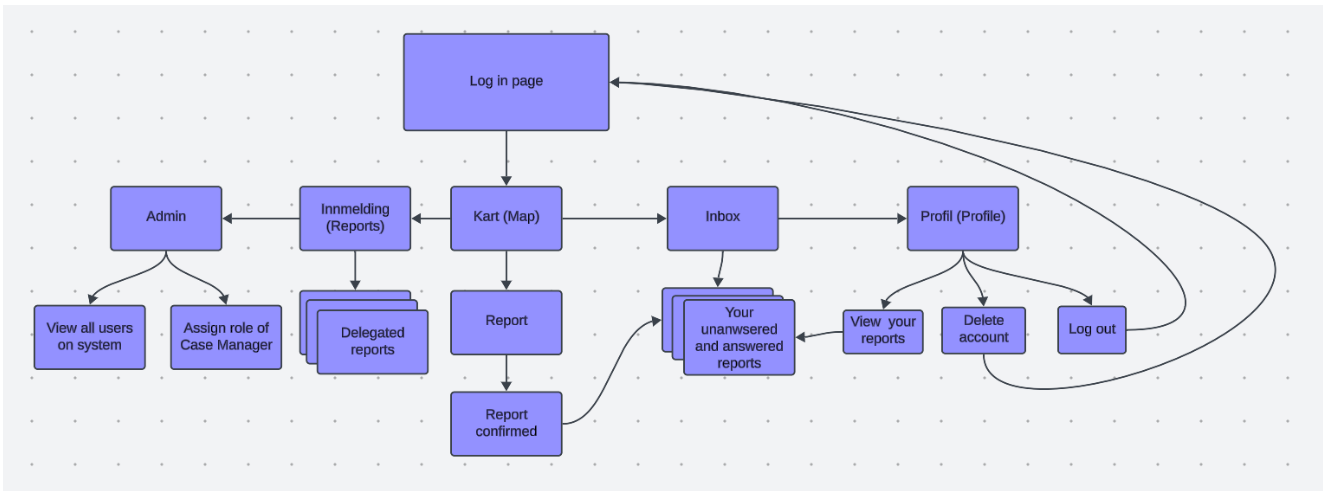




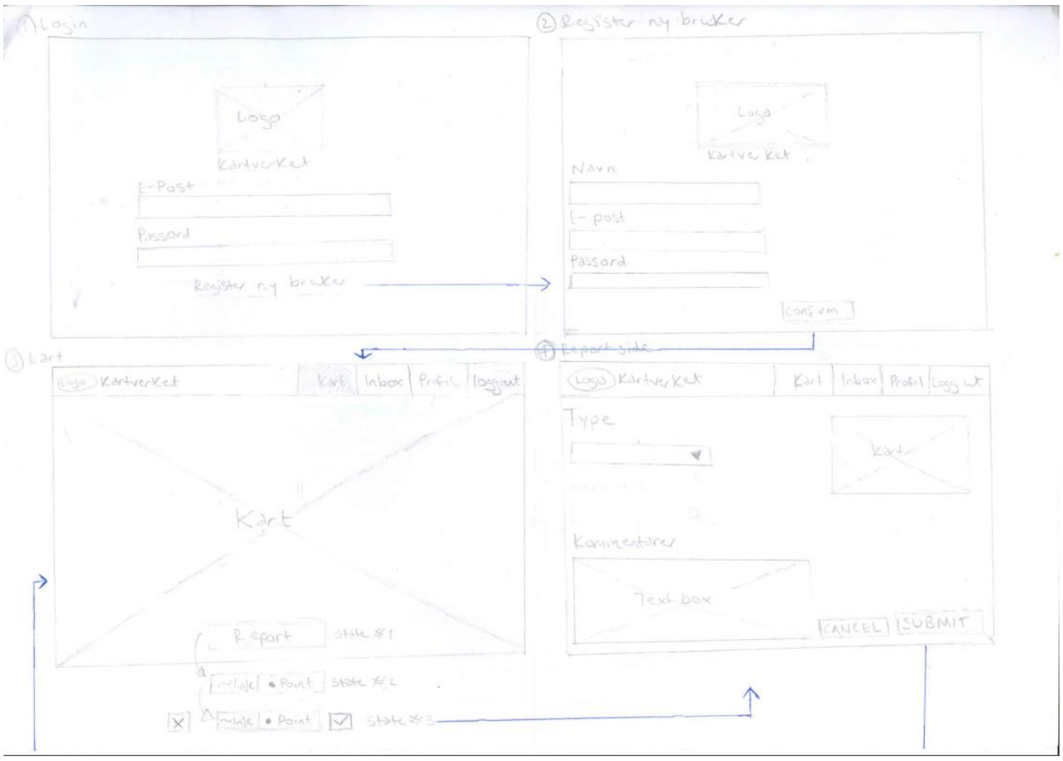


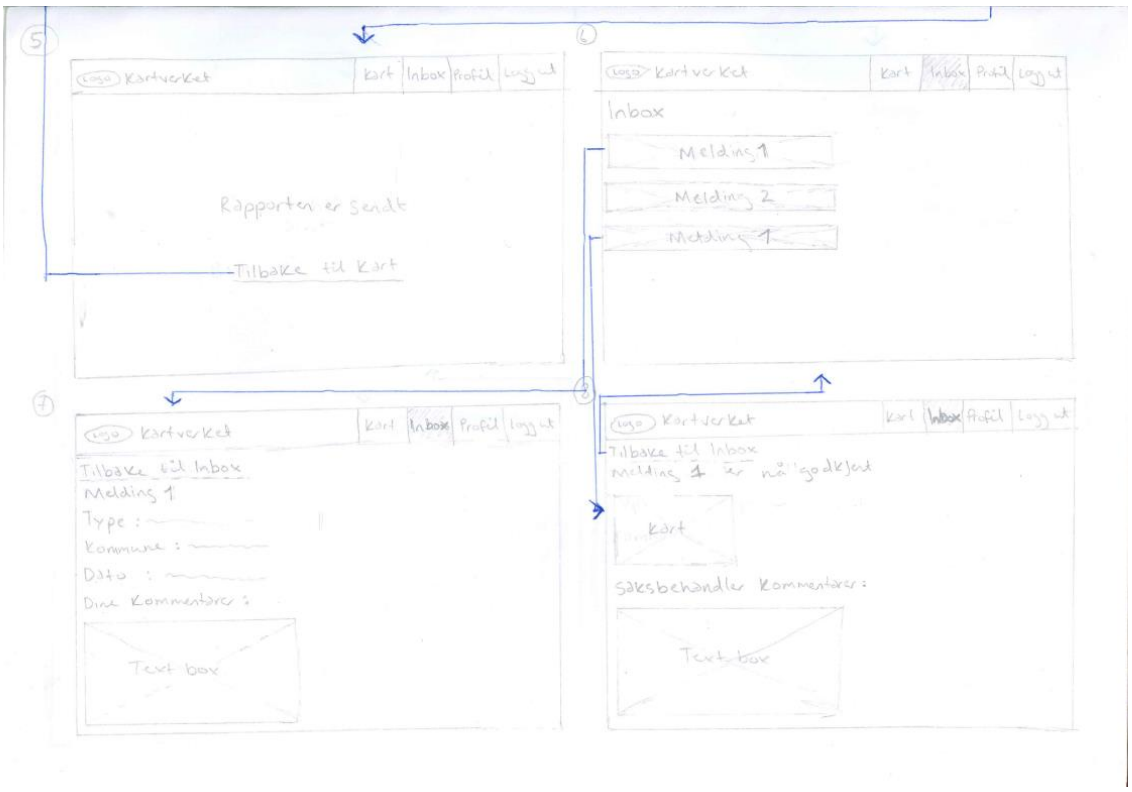
B2 Navigation Map

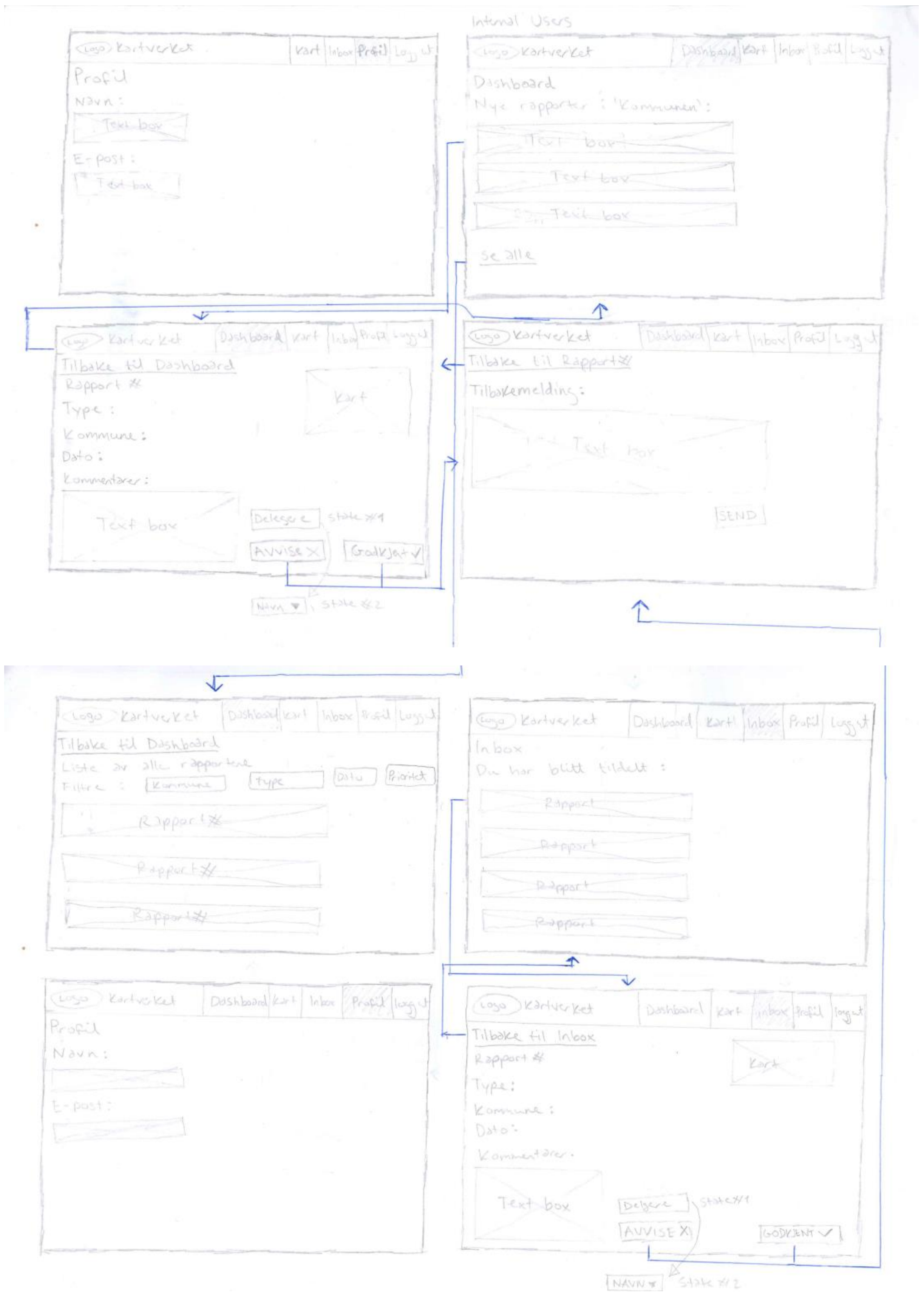
Internal Users



B3 Wireframes



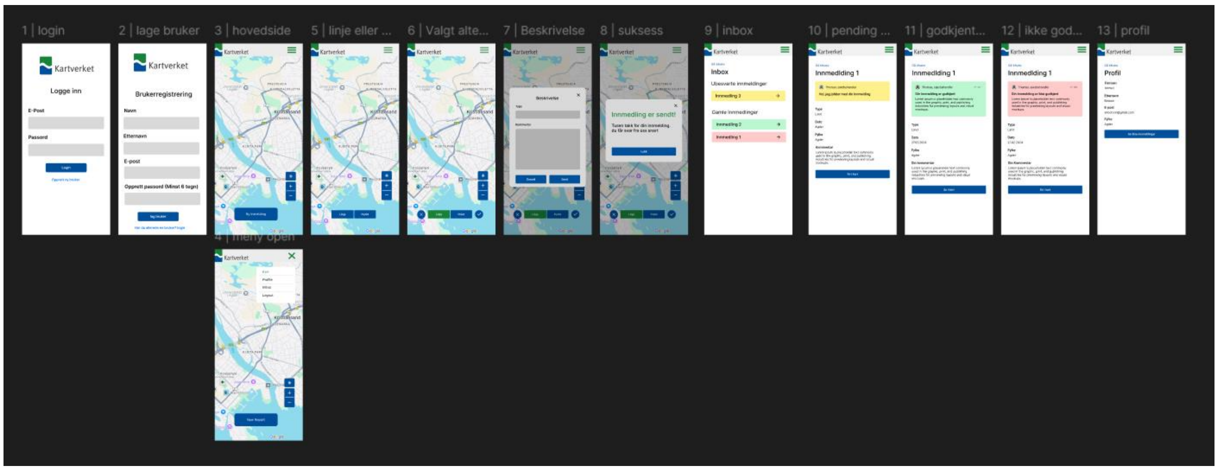




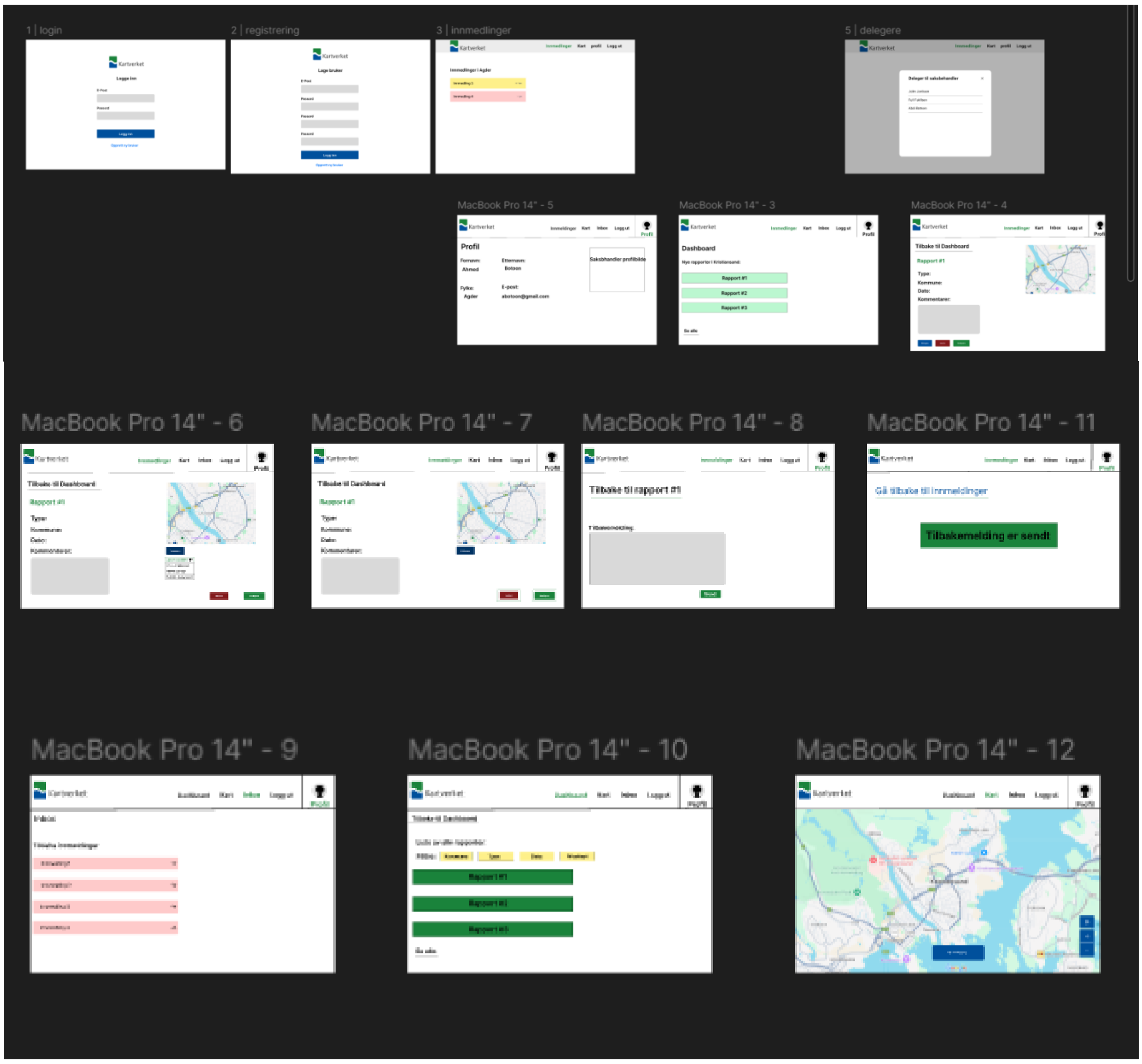
Internal Users
Logo Kartverket
Kart
Inbox
Profil
Logg ut
Profil
Navn:
Text box
E-post:
Text box
Dashboard
Nye rapporter i 'Kommunen':
Text box
Text box
Text box
Se alle
Tilbake til Dashboard
Rapport #
Type:
Kommune:
Dato:
Kommentarer:
Kart
Delegere
State #1
AVVISE X
Godkjent ✓
Navn ▼
State #2
Tilbake til Rapport #
Tilbakemelding:
SEND
Liste av alle rapportene
Filtre:
Kommune
Type
Dato
Prioritet
Rapport #
Inbox
Du har blitt tildelt:
Rapport
Tilbake til Inbox
GODKJENT ✓
NAVN ▼

B4 Interface Design/Prototype I Figma

Internal Users



External Users


1 | login
2 | registrering
3 | innmedlinger
5 | delegere
MacBook Pro 14" - 5
MacBook Pro 14" - 3
MacBook Pro 14" - 4
MacBook Pro 14" - 6
MacBook Pro 14" - 7
MacBook Pro 14" - 8
MacBook Pro 14" - 11
MacBook Pro 14" - 9
MacBook Pro 14" - 10
MacBook Pro 14" - 12
Kartverket
Logge inn
Lage bruker
Deleger til saksbehandler
Profil
Dashboard
Rapport #1
Rapport #2
Rapport #3
Tilbake til Dashboard
Tilbake til rapport #1
Tilbakemelding:
Gå tilbake til innmeldinger
Tilbakemelding er sendt
Inbox