Master's thesis

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

Styve, Svein Olav

Evaluating the Utility of Using a Mobile Application for Data Registration of Grazing Sheep on Open Pastures

Master's thesis in Master of Science in Informatics
Supervisor: Hvasshovd, Svein-Olaf
June 2021

NTNU
Kunnskap for en bedre verden

Styve, Svein Olav

Evaluating the Utility of Using a Mobile Application for Data Registration of Grazing Sheep on Open Pastures

Master’s thesis in Master of Science in Informatics
Supervisor: Hvasshovd, Svein-Olaf
June 2021

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

NTNU
Norwegian University of
Science and Technology

Abstract

Sheep farmers in Norway are legally required to perform weekly supervision trips to follow up and inspect their animals. To ensure animal welfare the government requires a yearly rapport from farmers. Supervision trips provide the information to use in such rapports. There exists no official tool for farmers to use while performing supervision trips for easily storing information in a structured manner and facilitate the rapport generation at the end of the season.

Pen and paper is an existing option for farmers, but with shortcomings both regarding the data input and the rapport generation. This thesis investigates the viability of using a mobile application for data input and data storage, as well as providing aggregate data for the government required rapport. The problem is explored by developing a mobile prototype of the application. To evaluate the prototype and its viability, usability testing was performed on the application, followed by interviews to get feedback on user satisfaction.

The results of this study show that a mobile application is an efficient tool with multiple benefits surpassing the traditional pen and paper approach. It requires less work and less time while also providing good user satisfaction. Some usability challenges were discovered, but they appear manageable for the user. Using a mobile application for data input on supervision trips appear to be a viable option that can benefit the farmer, shepherd, government and animals.

Sammendrag

Sauebønder i Norge er juridisk pålagt å utføre ukentlige oppsynsturer for å følge opp og inspisere dyrene sine. For å sikre dyrevelferd krever myndighetene en årlig rapport fra bønder. Oppsynsturene er metoden som brukes for å innhente informasjon til slike rapporter. Det finnes ikke et offisielt verktøy som bønder kan bruke når de utfører oppsynsturer for å enkelt lagre informasjonen strukturert slik at skriving av rapport til myndighetene ved sesongslutt blir lettvint.

Pen og papir er et eksisterende alternativ for bønder, men har åpenbare mangler både ved dataregistrering og generering av aggregert data for sesongrapport. Denne masteroppgaven undersøker muligheten til å bruke en mobilapplikasjon for data-inntasting og datalagring, samt kalkulering av aggregerte data til bruk i sesongrapporten. Utforskingen av problemet er gjort ved å utvikle en prototype av mobilapplikasjonen. Deretter er brukbarhetstesting utført på applikasjonen, etterfulgt av en intervjurunde for å få tilbakemelding på brukerenes tilfredshet.

Resultatet av denne studien viser at en mobilapplikasjon er et effektivt verktøy med mange fordeler som overgår den tradisjonelle metoden som bruker pen og papir. Applikasjonen krever mindre arbeid og mindre tidsbruk samtidig som den også har høy tilfredshet blant testerne. Noen utfordringer ved brukbarhet ble avdekket, men de fremstår som håndterbare for brukeren. Å benytte en mobil applikasjon for data-inntasting på oppsynsturer fremstår som et levedyktig og anvendbart alternativ med fordeler for både bonden, gjeteren, myndighetene og dyrene.

Preface

This master's thesis was written by Svein Olav Styve as part of his Master of Science in Informatics at NTNU, Department of Computer Science (IDI) in Trondheim. The supervisor for the project was Svein-Olaf Hvasshovd, providing guidance, feedback and valuable insight into the problem domain. The thesis was delivered in spring of 2021.

Contents

Figures

Tables

Code Listings

Acronyms

API Application Programming Interface. 42

DAO Data Access Object. 21

GPS Global Positioning System. 1

IDE Integrated Development Environment. 38

JSON JavaScript Object Notation. 28

MVVM Model-View-ViewModel. 20

RQ Research Question. 2

SDK Software Development Kit. 38

UI User Interface. 19
URI Uniform Resource Identifier. 23
URL Uniform Resource Locator. 42

WMTS Web Map Tile Service. 42

Glossary

dialog A window overlay that displays on top of the current fragment, prompting the user for data input or making a decision . 22, 46

fragment A portion of the UI of the app, generally associated with one screen of the app. 42

mockup A model of a design used for design evaluation and as a guideline for implementation . 19

observation The acquired information regarding grazing animals or relevant environmental aspects . 2

supervision trip A trip performed by the farmer or designated shepherd to follow up and inspect grazing animals. . 1

swiping An input gesture where the user drags a finger swiftly across the screen of the device. 48

usability The extent to which a product can be used by specific users to achieve specified goals with effectiveness, efficiency, and satisfaction in a specified context of use [1]. 2

use case A high-level description of an action the user can perform on the system. 15

Chapter 1

Introduction

This chapter begins by describing the background and motivation for the thesis. Then the problem definition, research goal and research questions are presented, followed by the scope of the project, the *design and creation* research strategy, and the main stakeholders. Finally the structure of the thesis is laid out.

1.1 Background and Motivation

Norwegian sheep farmers let their animals roam freely on open pastures from early spring to late autumn. The last couple of years the reported loss of sheep and lamb has been about 3% and 6,7% respectively, making up a total of 17 200 deaths in 2018 [2]. The government pays compensation for such losses to farmers. Norwegian sheep farmers are required by law to perform weekly supervision trips to inspect their animals, ensure animal welfare and reduce risks of injury or death [3]. While performing a trip the farmer will monitor and note information about the herds he sees, such as how many sheep and lamb are observed at a pasture. The farmers look for herds in the forest, on mountains and on open fields, which can be time consuming. Furthermore they have to document their findings and provide a report for the government at the end of each season.

Currently, a simple approach is to bring pen and paper on supervision trips to document important information. While the simplicity and flexibility of this approach is convenient, there exists potential benefits in using a mobile device for entering and storing observation data. Such a system has the ability to structure the data entered by the farmer, log locations using the Global Positioning System (GPS) and store observations and trips digitally. It might also lead to less manual labor when writing reports for the government as the application could provide aggregated data measures for the trips and observations of a season. A mobile application could prove simpler to use and more efficient than pen and paper, which could save time and effort for the farmer.

1.2 Research Goal and Research Questions

This project aims to design, develop and test a system for storing and managing supervision trips and associated observations of animals. This enables exploration of the utility, usability and usefulness of using a mobile application when performing supervision trips.

Research Goal
The goal of this project is to simplify and streamline the follow-up and data registration process for farmers, when they perform supervision trips to look after sheep and lamb on open pastures.

The main artifact of this project will be a prototype of a mobile application for sheep farmers. The mobile application should give farmers the ability to go for supervision trips while tracking their GPS position, creating a GPS trail. When observing animals, the farmer need the ability to store observation data of herds and animals by entering it into the application. All the entered and generated data must have the ability to be exported. The main purpose of the mobile application is for manually registering observations, providing an easy and comprehensive collection of data and the ability to later use the data, such as in a report to the government. Designing, implementing and testing this system will provide useful insight into the following research questions.

Research Questions

- RQ1: What value and usability does a mobile application provide for a sheep farmer when he is performing supervision trips?
  - RQ1.1: What efficiency and accuracy does the swiping gesture provide for entering herd animal counts while not looking at the phone screen?
  - RQ1.2: How does a mobile application compare to pen and paper for registering observations?

1.2.1 Project Scope

The application needs to provide functionality to match writing down information with pen and paper, such as animal counts of a herd with time and location, as well as information about dead and injured animals with their animal identification. Furthermore this project aims to expand the functionality to surpass the limitations of pen and paper. One such feature is tracking the location of the farmer when he is roaming, creating a trail of each trip. Another is being able to take images of animals to document injury or death. The application should also provide the ability to export data for use in government reports or data analyses.

The project will not look at multi-user support with login and authentication. It will neither provide a designated platform for farmers to share observation data with each other, but rather a simple export functionality such as email or cloud file services. These restrictions are made to keep the focus of this project on the main functionality relevant for sheep farmers and shepherds regarding performing supervision trips.

An interesting use of the data would be to perform analyses on it, such as finding areas where animals often are located, areas where predators often kill animals and areas where animals get injured. A simple presentation of observation data will be part of the mobile application, providing farmers to easily make educated guesses of where to conduct supervision trips or to discover areas with large amounts of injury or death. A more thorough analysis and visualisation tool for the collected data is not prioritized in this project.

1.3 Design and Creation

The Design and Creation [4] research strategy is followed as the research requires a mobile application to evaluate and perform tests with. This strategy makes it possible to elicit requirements for the system based on a preliminary study, subsequently implementing the system based on these requirements. The prototype can then be tested by users to evaluate utility and usability. The data generation method for these tests is observation where the participants are observed while performing common tasks on the system. This results in both qualitative data from observations and participant statements, as well as some quantitative data such as time usage and completion status of each task. The prototype is also an artifact for feedback, and interviews will be performed with each participant after having performed the usability test, leading to more qualitative data regarding the satisfaction of the user. Analyzing the results of the tests and interviews will help determine the viability of using a mobile application on supervision trips. It will also specifically test the utility and usability of the prototype, leading to an evaluation of the user interface and input methods used in the prototype.

1.4 Stakeholders

The project has a number of stakeholders which may benefit from the research and the application prototype. This section describes who the stakeholders are and what they might gain from the results of this thesis.

1.4.1 Farmers

Farmers are the primary stakeholder in this project. The loss of animals means loss of revenue even if some compensation is received from the government. The proposed app could simplify the process of performing supervision trips to easier comply with laws, as well as making supervision trips less time consuming and demanding. The collected data is potentially useful in analyses of animal behaviour to determine where herds could be located and where animals regularly are lost, injured or killed.

1.4.2 Shepherds

The farmer is not necessarily the person performing the supervision trip. A designated shepherd can be hired by the farmer to perform this task. This means that the shepherd is an important stakeholder as he is the main user of the application. The shepherd will have requirements regarding what information is stored, how it is entered and how it is displayed. Note that this project looks at the shepherd as a role that can either be assigned to the farmer himself or to a separate shepherd.

1.4.3 Government

Another important stakeholder is the government. As the government pays compensation for the loss of sheep and lamb, there can be money to save if the farmers have better tools for storing information about their grazing animals and to ensure their well-being and safety. As each farmer is to send a rapport after each season, adopting the application might result in more structured rapports of higher quality with more useful data.

1.4.4 Developers and Designers

Developers trying to implement similar systems, or systems that need reliable interaction methods when entering data on a mobile device in different environmental conditions, could benefit from the results of this study. Particularly the findings on using swipe gestures for registering counts of different animal types without requiring looking at the phone could be of interest.

1.5 Thesis Structure

This thesis is structured as follows. In Chapter 2 theory and results of the preliminary study are presented. The problem domain is explored and explained, paving the way for the requirements of the application. Then a literature study is performed to investigate what research exists in the domain. Finally alternative solutions are considered and evaluated.

Chapter 3 begins by describing the requirement elicitation process, followed by the functional requirements presented as use cases. Finally the non-functional requirements intended to make a good user experience are listed.

In Chapter 4 the system architecture design is presented, followed by the data storage and data model design. Then sketches of the User Interface show the main aspects and functionality of the system.

Chapter 5 starts by describing the development process and the tools used to support it. Subsequently platform and framework decisions are laid out, followed by how local storage is implemented. The map framework and map service is described in more detail. Finally the application prototype is presented with screenshots and explanations of the UI and functionality.

In Chapter 6 usability testing is first described, followed by its purpose in this project. Then the usability test and its tasks are presented. The main findings from the results of the usability testing and interviews are presented.

Chapter 7 evaluates the test results and discuss their implications. The research questions are answered based on the results and evaluation, followed by a presentation of the main contributions of the work. Some limiting factors are presented along with suggestions for future work.

Chapter 8 summarizes the findings and draws a conclusion regarding the viability of using a mobile application on supervision trips.

Chapter 2

Theory and Preliminary Study

This chapter explores and establishes the problem domain for the thesis. Then a literature study is performed, setting the context for the research effort. Subsequently, existing solutions are evaluated to see if they can be used at supervision trips for registering the desired observation data.

2.1 Problem Domain

The project is done with remote access to a domain expert. The domain expert has years of experience and performed many supervision trips looking for sheep and lamb. This provides a valuable resource for acquiring knowledge about the domain and the development of the project idea. His input will contribute in questions such as determining what data a farmer is interested in storing and what the most crucial aspects of the application are.

The most relevant part of the domain is the supervision process where the farmer or a designated shepherd is going for trips to follow up the well-being and safety of animals. On such a trip the farmer documents relevant information, such as the animals in a herd or injuries he detects. To properly understand this activity the observed entities, primarily sheep and lamb, need to be examined further.

Norwegian sheep farmers have a number of herds consisting of sheep and lamb. The composition of each herd does not need to be constant and herds might split and merge while out on the pastures. Each sheep is to walk together with a group of lamb ranging from zero to three. It is recommended to mark the sheep with colored ties, indicating the number of lamb it is walking with [5]. Counting the number of lamb and the number of sheep with different tie colors in a herd can help determine if a lamb is missing. The coloring is described in Table 2.1.

The most important information for a farmer is to know if an animal is dead, missing or injured, as this requires special attention and action. A dead or injured animal needs to be registered with its unique id. The id is normally found on the

Tie Color	Number of Lamb
Red	0
Blue	1
Yellow	2
Green	3

Table 2.1: Sheep Tie Colors

ear tag of the animal. The animal should also be photographed to document the death or injury.

Another useful piece of information is to know where herds are spotted and how many animals have been observed in each herd. This can reveal animals that have migrated from one herd to another and are not actually missing. Farmers can over time develop knowledge about where the herds typically are located, and make educated guesses on where to begin looking for them.

Due to the fact that sheep farmers tend to have a large number of animals, it does not make sense to uniquely identify each animal in the herd when a herd is spotted, but rather count the number of sheep and lambs, as well as their tie colors. Animals can be of different color and registering the amount of white, grey and black animals is useful to distinguish herds quickly. Herds are mostly observed from a distance using binoculars, which further justifies not storing animal ids when observing herds. Useful data to store would thus be the count of each of the mentioned types. See Table 2.2 for the set of data associated with an observation of a herd.

Count Type	Description
Sheep	Number of sheep observed
Lamb	Number of lamb observed
White	Number of white animals observed
Grey	Number of grey animals observed
Black	Number of black animals observed
Red Tie	Number of sheep with red ties observed
Blue Tie	Number of sheep with blue ties observed
Yellow Tie	Number of sheep with yellow ties observed
Green Tie	Number of sheep with green ties observed

Table 2.2: Herd Observation Data Points

While using binoculars to observe herds it is cumbersome and error-prone to constantly put down the binoculars to update the observed animal counts, just to find that the animals have moved when you resume looking in the binoculars again.

For this reason a user interaction method for registering animal counts without needing to look at the phone screen would likely be beneficial. An area of focus in this thesis is to implement and test functionality for this purpose.

As mountains and trees can obscure the view of the farmer, resulting in incomplete observation data, it is interesting to know where the farmer was positioned when making the observation. Both the lookout point location of the observer and the observed entity location need to be stored. As the farmer might discover new information from another location, storing a secondary lookout point in relation to the observed entity could be beneficial. Storing more than two lookout points is not considered useful as it could quickly complicate observation data without much benefit.

One reason for death of grazing animals is predators [6]. Because of this it is valuable for the farmer to be able to mark predators and predator traces on the map. This can help the farmer pinpoint dangerous areas with higher risk so that he can keep his herds away from such areas or plan other appropriate measures.

Farmers can discover valuable concerns in the environment other than those already mentioned. For this reason the opportunity to create a generic environment observations is considered relevant. This enables the farmer to mark areas of special interest with an additional note and image, such as the status of certain fields, traces of animals or a fence that needs repair.

This analysis of the problem domain has revealed five types of observations the farmer should be able to store in the application. A short summary of each type is presented in Table 2.3.

Observation Type	**Description**
Herd	Register the counts of different animal types in the herd and optional text notes
Injured	Register an injured animal with its identification number, images and text notes
Dead	Register a dead animal with identification number, images and text notes
Predator	Register a predator with images and text notes
Environment	Register environmental condition with images and text notes

Table 2.3: The Five Observation Types

The person performing the supervision trip takes the role as a shepherd. This role can either be taken by the farmer himself or a designated shepherd. For simplicity, the terms user, shepherd and farmer are used to mean the user of the application throughout the thesis.

2.2 Related Literature

A literature review was conducted to examine what research exists in the problem domain and related areas. NTNU Oria was used as the search engine, which allows searching in NTNU's University Library. All results were filtered on peer reviewed articles. The following search queries were used:

- application AND pasture AND (animal OR grazer OR grazing)
- (grazers OR grazing) AND (tracking OR track)
- monitor* AND livestock

As the problem domain is narrow it was difficult to find good search queries yielding relevant literature. For this reason multiple search queries were used. The chosen search queries yielded only a couple of relevant results. However, no results were found discussing the use of mobile applications for supervision of grazing animals on open pastures. Some literature was considered relevant for further review. These articles could be divided into two groups, either regarding the use of GPS tracking collars on animals in field or the use of unmanned areal vehicles to detect and count animals. One article from each group is reviewed below.

2.2.1 Perspectives on The Use of Unmanned Aerial Systems (UAS) to Monitor Cattle

According to [7], the use of unmanned aerial systems (UAS) has seen limited use in monitoring cattle. The article points to a number of factors that could be the reason for this. One of the reasons seem to be strict government regulation of these aerial vehicles, where a licence is often required for operation. It is clear that it is a big investment for the farmer in both time and money to get a licence to be able to use an unmanned areal vehicle (UAV). Another factor is the cost of these vehicles themselves, which are too high to be economically beneficial. The farmer cannot be sure if buying an UAS will prevent enough injuries and deaths to pay for itself.

A number of practical challenges such as landing and limited operational time are also pinpointed. Fixed-wing drones need some sort of runway for landing, while rotary drones are more agile. While battery technology has come a long way, the achievable flight duration is still a limiting factor. Additionally technical challenges such as needing small and light sensors with a sufficient resolution are of concern, since the sensors can not be too heavy. Even if all these problems are surpassed, the analyses of images for detection and counting of animals has a margin of error, and can give bad output. It can falsely classify animals and herds or give incorrect counts.

Neither the technical, practical or regulatory challenges have been surpassed yet, and the use of UAS for monitoring cattle currently does not seem like a viable option for farmers. A functioning system would be valuable by giving the farmer information about where animals are located and some estimate of the count in each group spotted. This would enable the farmer to know where to find his animals and if herds might have split up. In spite of the mentioned challenges the article claims that UAS could be a good option for monitoring cattle, but underlines the fact that more work and research is needed.

2.2.2 Use of GPS Tracking Collars and Accelerometers for Rangeland Livestock Production Research

The article [8] takes a look at collars equipped with GPS and accelerometer. The collars are used for tracking the location and movement of livestock. The authors indicate that at some point in the future such a collar might not just tell the farmer where the animal is located, but also rapport on illness and well-being of the animal. This is assumed to be possible using real-time analysis of accelerometer data.

One existing benefit of using GPS collars on livestock is for getting the location of animals in field. The farmer can see where his sheep and lamb currently are and where they have recently been. This helps the farmer in determining where to go look for his animals. There are however some limitations with this approach. The collar runs on battery and will therefore have a limited time of operation. To be able to determine the location of an animal the GPS requires signals from multiple satellites, which can be obscured by trees and mountains in the area. As collars might use a mobile network for transmitting location data to the farmer, the same obstacles could hinder this data transmission. At last, even if the price is dropping, fitting collars on a large number of animals is costly, and might not seem worth the cost for a farmer.

2.2.3 Findings

Options such as Unmanned Aerial Systems and digital collars could provide value for farmers in determining the whereabouts of their animals. Digital collars also have some potential to derive the well-being of an animal by using an accelerometer. It is however apparent that in their current state, these systems provide limited utility in regards to the full set of information a farmer needs to obtain from his grazing animals. The information gained from manual inspection trips is more comprehensive and provides greater value when following up on animal welfare, pasture quality, predators and other environmental aspects. The alternatives investigated in the literature is therefore not able to eradicate the need for manual inspection. Such methods can however be useful in the planning of supervision trips, such as by having some knowledge of what areas animals might reside in, and plan a supervision trip to those areas. They provide a promising complement to the manual animal supervision trip approach.

2.3 Alternative Solutions

While pen and paper is a simple tool for writing down observations, other alternatives might already exist, and should be examined. Mobile applications should be evaluated to see if there exists applications that could support the farmer when performing supervision trips. Functionality of such apps are explored and discussed in this section.

2.3.1 Google Maps

An initial idea for a farmer might be to use a map application to log observations of herds and animals. To investigate the feasibility of using a plain map application I look at the easily available and popular app Google Maps [9]. This app is primarily used for looking up places or navigating from one place to another, and has obvious shortcomings for use in the forest. Firstly it does not have detailed topographic maps with height and terrain markings. The app does however provide the ability to download map areas for offline usage. It also lets the user mark locations on the map, but does not provide any means of storing data on the marked location. The app provides a timeline of positional data showing your movement for each day, but does not have the ability to explicitly create trips.

2.3.2 Norway Topos Maps

A more specialized map application is Norway Topos Maps [10]. This application has detailed topographical maps that provides the user with height details and terrain coloring. It also enables users to log GPS trails of their trips. The application gives the user the option to download map areas so that trips can be performed without a data connection. The app does however lack the ability to store observations as there does not exist any way to mark positions on the map and log observation data.

2.3.3 Telespor

The company Telespor sells a collar that tracks sheep and stores their location regularly to a server [11]. This way a farmer can see the location of animals over a period of time. The GPS trails of sheep can be monitored on web or using an app. This approach bypasses the manual observation trip and directly stores information about animals to a server. A clear benefit is that the farmer can see where animals are located and their movement pattern.

There are some drawbacks to this method as it can not provide as much information as a manual observation trip can. Such collars will additionally not be able to report any predators or environmental conditions. The collars can run out of battery or be in an area where there is no reception and thus the location is not sent to the server. Additionally the cost of fitting these on a large number of sheep would be substantial, and having a collar or two per herd might be a more typical use, but provide less comprehensive information.

2.3.4 BeiteSnap

BeiteSnap is an app meant for notifying farmers when someone has seen their animals [12]. It lets users enter animal observations that are automatically sent to farmers. A farmer is only notified if he is registered in the application and if the observation was made in an area tied to the farmer. The location of the observation is automatically stored using GPS, but there is no differentiation between where the user was position and where the animal was positioned. Furthermore an observation only distinguishes between having seen a living, dead or injured animal with the option to note other details as text. Observations of herds are not supported and it does not provide a structured way to enter the data points described in Table 2.2. Additionally there is no notion of trips in the app, just plain observations.

2.3.5 Comparison and Findings

Analyzing alternative solutions with different levels of relevance show that some solutions have a subset of desired functionality to support the needs of the farmer. No application has all the desired functionality, as illustrated in Table 2.4. Therefore the proposed mobile application will provide a set of features unlike what currently exists, tailored to the farmers requirements for supervision trips. The functionality of registering observations with the set of data proposed in Table 2.2 is not facilitated by any of the reviewed alternatives. Additionally no solution has the option of registering animal counts without looking at the screen.

Application	**Topographical Map**	**Offline Map**	**User GPS Trail**	**Observation**
Google Maps	No	Yes	No	No
Norway Topo Maps	Yes	Yes	Yes	No
Telespor	Yes	No	No	No
BeiteSnap	Yes	Yes	No	Partial

Table 2.4: Alternative Solutions with Features

Chapter 3

Requirements

This chapter begins by describing the requirement elicitation process, followed by use-cases that make up the functional requirements. Finally a list of non-functional requirements are presented.

3.1 Requirement Elicitation

The first meeting with the domain expert was used to get an insight into the domain, a description of the current approach to supervision trips and some general goals of the desired system. Multiple meetings were performed over the course of the project to elicit both functional and non-functional requirements as well as getting feedback on the application as it was developed. From the meetings with the domain expert the following overarching goal for the mobile application emerged:

The application should provide an easy and efficient way to store observations when performing supervision trips.

This goal provides the foundation for the functional and non-functional requirements. Together with the preliminary study and more meetings with the domain expert a set of initial requirements were elicited. Following the agile development approach these requirements were not considered final, and could be changed during development. The requirements presented in this section are the final requirements.

3.2 Functional Requirements

The functional requirements are presented as use cases, defining the actions users can take on the system. Further decomposition from use cases into development tasks were done *just-in-time* as an adaptation of the agile development process.

Offline Map Areas

The app needs to function without a data connection as it will be used in areas without mobile network connectivity. The user will have to download the map area needed before starting the supervision trip. The user should also be able to see details about all the observations registered in the area. The related use cases are:

- Download new Map Area
- View all downloaded Map Areas
- View details of a Map Area
    - View all observations registered in the area
    - See aggregated info about observations, trips, dead and injured animals in the area
- Delete a Map Area

Trips

One of the main concepts in the app is a trip. A user will perform trips at a map area. While walking, the trip trail is logged as a list of GPS positions. The related use cases area:

- Create a Trip
- View details of the ongoing Trip
- View all conducted Trips
- View details and observations for a previously conducted Trip
- View the GPS Trail for the Trip
- Delete a Trip

Observations

The user will make a set of observations for each trip. There are multiple types of observations, as described in section 2.1. Both the lookout point of the observer, as well as the location of the observation, should be stored. A secondary lookout point can be added to an observation since different lookout points can reveal new information that needs to be included in the observation. The related use cases are:

- Add an observation to a Trip
    - Select the observed location
    - Select the type of observation - herd, dead, injured, predator, environment
    - Take/attach images - for observations of dead animal, injured animal, predator and environment
    - Register animal counts without looking at the device - only for herd observations

    - Add a secondary lookout point for an observation - only for herd observations

- View observation details
- Change observation details
- View all observations for a Trip - on the map and in a list
- Delete an observation

Data Export

As the main purpose of the application is data input and storage, the app should provide the user with export functionality for all data stored in the app. The related use cases are:

- Export a single trip and all observation data for that trip
- Export a simple rapport of aggregated and key data measures
- Export the entire collection of data stored in the app

Settings

To facilitate different users the app has settings that can be adapted for different demands. The related use cases are:

- Set GPS time interval - how often to log GPS location
- Set GPS distance interval - how often to log GPS location
- Request OS Permissions required by the Android Application

3.3 Non-functional Requirements

The non-functional requirements describe aspects of the application needed to provide a good experience for the user. Meeting these requirements will impact the usability of the application, and will be important in the design of the application. The application will be regularly tested to meet these requirements.

- The user should be able to go for supervision trips without having an internet connection.
- The user should be asked to confirm critical actions like deleting data.
- The user should be able to go for trips lasting multiple hours and register at least 15 observations for each trip.
- The user should be able to take at least five images per observation.
- The app should have a consistent language and layout for similar UI elements.
- The app should support both English and Norwegian language - based on system language settings.
- The user should be able to download map areas from Norway.

Chapter 4

Design

This chapter shows the technical and graphical design of the application. First the possible mobile platforms are considered and the general architecture of the application is described, followed by a description of the underlying data model. Then mockups of the user interface (UI) are presented and explained.

4.1 Platform

The platform requirement for this project is that the application should run on a mobile device for the farmer to bring on supervision trips. Different mobile platforms are evaluated and discussed in this section.

Cross-Platform Development

In recent years the interest in cross-platform mobile development has increased with frameworks such as React Native and Xamarin Forms. Such frameworks enable developers to write their app once and run it on multiple platforms, such as iOS and Android. React Native enables developers already experienced in web development with React to use their previous knowledge in developing mobile applications. Xamarin Forms enable .NET developers familiar with C# to develop mobile applications without learning a new language and keep using some of the familiar concepts from their .NET experience. These alternatives do however come with restrictions compared to the native way of developing apps for both iOS and Android as they must abstract away some of the particular concepts specific to each platform. Cross-platform frameworks are therefor known to be limiting in certain situations. Additionally the same learning curve exist for developers not already familiar in either React web development or .NET development. To not be kept back by possible cross-platform framework limitations it was decided to focus on a single mobile platform.

Mobile Platforms

There are two large mobile platforms that make up the majority of the smartphone market today: iOS and Android. There has existed and does to some extent still exist other alternatives, but these make up such a small portion of the market share that developing an app for such a platform likely would benefit an insignificant amount of potential farmers. Thus the two stated mobile platforms are considered.

Both iOS and Android are operating systems made for mobile devices and they have similar features. Both platforms provide features such as GPS, map UI views, storage of data, background processing and sound output. Both platforms also have large market shares and SDKs for modern application development. As no platform has obvious drawbacks, the platform to develop for is chosen based on practicality, experience and cost. The phone available for use when testing is running Android. iOS development requires a Mac, but the only available development PCs run Windows. As I have some experience with Android Development this likely would enable quicker and better development, with the potential of more features and a better polished app. These considerations ultimately lead to the choosing of Android as the platform for the mobile application.

4.2 Architecture

The architecture of the app follows the guidelines recommended by Google. This architecture was chosen as it reduced the overhead of developing for Android. Using the recommended frameworks and libraries enable quicker development with less chances of bugs. It provides clear guidance to design and implementation, compared to the traditional approach. The main aspects of this architecture are presented in this section.

Fragment

A Fragment is a portion of the UI of the app. Each fragment has a layout that defines the view presented to the user. A fragment can respond to input events from elements in the layout. Each fragment can be in one of may states throughout its lifecycle, such as whether it is shown to the user or is in the background. When a fragment changes state the appropriate lifecycle method is called. When users navigate between different screens in the app, they will be navigating between different Fragments in Android.

Model-View-ViewModel (MVVM)

The MVVM pattern is followed, separating objects into three groups: Model, View and ViewModel. Each screen of the application follows this pattern. The view contains visual elements and controls. The model hold application data. The ViewModel connects the model and the view by holding the data that the associated view relies on. In Android the view is part of a Fragment. Each Fragment can have an associated ViewModel containing data. The ViewModel has a reference to a Data Access Object (DAO) which contains methods for accessing and updating data, see Figure 4.1.

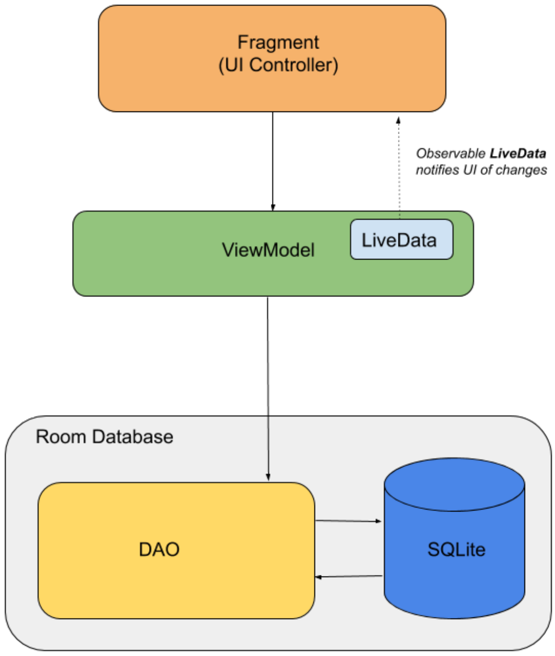

Figure 4.1: Android Application Architecture

Binding and LiveData

In addition to the MVVM pattern, this application uses Data Binding. Data Binding allows for binding between components in the UI with data sources in the app. Combined with LiveData this enables the UI to automatically update when the underlying data changes. This is possible because LiveData notifies views when there are changes in the corresponding data in the database. LiveData objects are defined in the ViewModel and follows the Observable pattern. The dotted line from LiveData to Fragment in Figure 4.1 illustrates how data changes propagate to the UI.

Room Database

The Room Persistence Library is used for data storage, following the best practices of Android. This is a local storage option based on SQLite. It uses Data Access Objects (DAOs) for interacting with the underlying SQLite database. This is illustrated in the bottom part of Figure 4.1. Each ViewModel has a reference to a DAO, enabling it to retrieve and update data. For more details about the storage and data model see section 4.3.

Services

In Android a Service is a component of the application that can perform long-running tasks in the background. To enable the logging of GPS trails even when the app is not in the foreground, a Service can be used. The Service continues to run if the user switches to another application, goes to the Android home screen, receives a call or turns off the phone screen.

Dialog

A dialog in Android is a window overlay that is displayed on top of another fragment. This dialog can either prompt the user for data input or for making a decision.

Intents

An Intent in Android can be used to start an Activity of another application and receive back a result. Intents can be used for capturing new or selecting existing images. For example it can open the Camera app and let the user take an image. A reference to the captured image is given back to the Fragment that started the Intent.

4.3 Data Storage and Data Model

The main usage of the mobile application is for input of data by the user. The data consists of information the farmer wants to store when he is performing supervision trips. In these situations the farmer generates data that needs to be stored using an appropriate data model. This section presents the data storage solution and data model.

4.3.1 Data Storage

As the focus of this thesis revolves around how the user interacts with the application, it was evident that developing a server for data storage provided little value for the research. This would additionally introduce the need for security to protect the data and ensure only authorized users could access the server. As this would introduce work that is not directly related to the problem it was decided to store the data locally on the device with the option to export data manually.

The Room Persistence Library is able to fulfill the storage requirements of the application. The library stores data in an SQLite database on the device. All interactions with the database are done using a Data Access Object (DAO). Each entity is created as a data class, which automatically generates the required tables in the database. Images are stored as files on the device file system with an entity referencing a file using a Uniform Resource Identifier (URI).

4.3.2 Data Model

The data model was created based on the system requirements. Each entity has its own table in the database with foreign keys making up relations between the entities. See Figure 4.2 for the ER diagram of entities and the relationship between them. This diagram shows that each Trip is performed in association with a MapArea. Each Trip has a number of TripMapPoints making up the GPS trail the user walked during the Trip. From any TripMapPoint an Observation can be registered, associating it with the TripMapPoint. This way each Trip indirectly has a number of Observations associated with it. The location of where the user was positioned when registering the Observation is stored as the relation between an Observation and a TripMapPoint. There can be two such relations, supporting a secondary lookout point. As each Observation can have two TripMapPoints, a separate relation between an Observation and a Trip is kept to simplify querying for all Observations of a Trip.

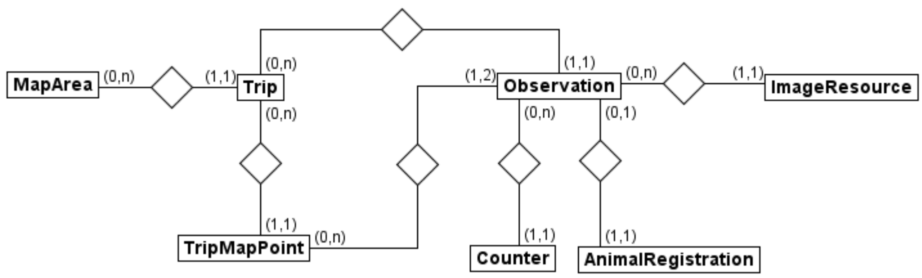

Figure 4.2: ER Diagram of Data Model

Each Observation has an observation type such as *herd* or *injured animal*. The type of the Observation is determined by the observation_type field. If the Observation type is Herd nine Counter entities are created, one for each counter type. Examples of counter types are *Sheep*, *Lamb* and *RedTie*, see Table 2.2 for all possible counter types. Each of the Counter objects tell how many animals have been spotted of that specific type of animal for the associated Observation.

If the Observation type is of Injured Animal or Dead Animal then an AnimalRegistration entity is created and associated with the Observation. The AnimalRegistration entity will contain information about the identification number of the animal and a description of its condition.

Note that the observation type determines if Counter objects or an AnimalRegistration object is created and associated with the Observation object, as these objects store data specific to that Observation type. See Table 4.1 for details about the different Observation types that lead to Counter or AnimalRegistration objects being created and associated with it. Observations where the Observation type is Predator or Environment store all required information in the Observation entity itself.

Each Observation can have any number of ImageResource entities associated with it. Each ImageResource points to an image file on the file system. This indirectly lets each Observation be associated with any number of image files via ImageResource entities.

Observation Type	Counter	AnimalRegistration
Herd	9	-
Injured Animal	-	1
Dead Animal	-	1
Predator	-	-
Environment	-	-

Table 4.1: Entities Created for Different Observation Types

Using the Room Persistence Library, each entity is defined in its own data class. Objects of these data classes are used in the ViewModels, which interacts with the database through methods in the DAO. The DAO enables creating, retrieving, updating and deleting objects. Each table in the database is presented below.

MapArea

A MapArea represents a portion of the map that is stored on the device. It provides the ability of performing trips without a data connection.

Field	Type	Description
map_area_id	Long	Primary Key
map_area_name	String	Name of area
map_area_min_zoom	Double	Minimum zoom level
map_area_max_zoom	Double	Maximum zoom level
map_area_bounding_box	String	The geographical area covered

Table 4.2: MapArea Table Fields

Trip

A Trip represents a supervision trip performed by the user. It is associated with a MapArea, keeping all trips organized and enabling aggregations for different MapAreas.

Field	Type	Description
trip_id	Long	Primary Key
trip_name	String	Name of trip
trip_date	Long	Start date of trip
trip_finished	Boolean	Completion status of trip
trip_finished_date	Long	Completion date of trip
trip_owner_map_area_id	Long	Foreign Key to owner MapArea

Table 4.3: Trip Table Fields

TripMapPoint

A TripMapPoint is a geographical position with a timestamp. A collection of these points, ordered by the timestamp, make up the GPS trail of a Trip.

Field	Type	Description
trip_map_point_id	Long	Primary Key
trip_map_point_lon	Double	Longitude
trip_map_point_lat	Double	Latitude
trip_map_point_date	Long	Date and time
trip_map_point_owner_trip_id	Long	Foreign Key to owner Trip

Table 4.4: TripMapPoint Table Fields

Observation

An Observation contains data entered by the user. It can be of several different types, based on the observation_type field. The types are described in Table 2.3. Observations with the observation type of Herd stores the animal counts in Counter entities. Observations with the observation type of Injured or Dead store information about the animal in an AnimalRegistration entity.

Field	Type	Description
observation_id	Long	Primary Key
observation_note	String	Observation notes
observation_lat	Double	Latitude
observation_lon	Double	Longitude
observation_date_time	Long	Date and time of observation
observation_type	Int	Type of observation, e.g. Herd
observation_owner _trip_map_point_id	Long	Foreign Key to owner TripMap-Point
observation_secondary _trip_map_point_id	Long	Foreign Key to optional secondary TripMapPoint
observation_owner_trip_id	Long	Foreign Key to owner Trip

Table 4.5: Observation Table Fields

Counter

A Counter stores the amount of animals for any of nine different count types, such as Sheep, Lamb and RedTie, see Table 2.2 for a all the possible count types. Each Counter is associated with an Observation where the Observation has its observation_type field set to Herd.

Field	Type	Description
counter_id	Long	Primary Key
counter_value	Int	Count value
counter_type	Int	Type of counter, e.g. Lamb
counter_owner_observation_id	Long	Foreign Key to owner Observation

Table 4.6: Counter Table Fields

AnimalRegistration

An AnimalRegistration is used to store data about an animal when the associated Observation has its observation_type field set to DeadAnimal or InjuredAnimal. It stores the animal identification number and a description about its condition.

Field	Type	Description
animal_registration_id	Long	Primary Key
animal_registration _sheep_number	String	The identification number (ear tag) of the animal
animal_registration_note	String	A note about the animal observation
animal_registration_owner _observation_id	Long	Foreign Key to Observation

Table 4.7: AnimalRegistration Table Fields

ImageResource

An ImageResource is used to store an image associated with an Observation. It contains the URI of the image file stored on the file system of the device.

Field	Type	Description
image_resource_id	Long	Primary Key
image_resource_uri	String	The URI of an image file on the device file system
image_resource_observation_id	Long	Foreign Key to Observation

Table 4.8: ImageResource Table Fields

4.3.3 Exported Data

Data stored in the app is exportable. The architecture of the export functionality is shown in Figure 4.3. Users have multiple options for exporting data in the app:

- Exporting the trip and observation data for a single trip
- Exporting a simple rapport of aggregated and key data measures
- Exporting the full database including images

After the user has performed a trip it would be desired to export the trip and observation data to the farmer. Exporting a single trip would mean exporting a database file only containing rows of data related to the given trip as well as any image files for the observations. This enables the farmer to import the trip and observation data into a separate system to keep an overview of all collected data, potentially from multiple users or multiple devices. Being able to export a single trip at a time allows the farmer to be regularly updated by any shepherd.

The export of rapports enables a user to send a simple rapport of aggregated and important data points such as the number of trips, dead animals and injured animals to himself or to someone else. This can simplify and shorten the process of writing a government rapport, since much of the key information is provided. The rapport can either be exported as a text file or a JSON file.

As the application is primarily a data collection tool it supports exporting the full data set. The data consists of the database file with all its tables as well as any images associated with the observations. This provides advanced users the ability to perform complex analysis on the data. The database is exported as an .db file.

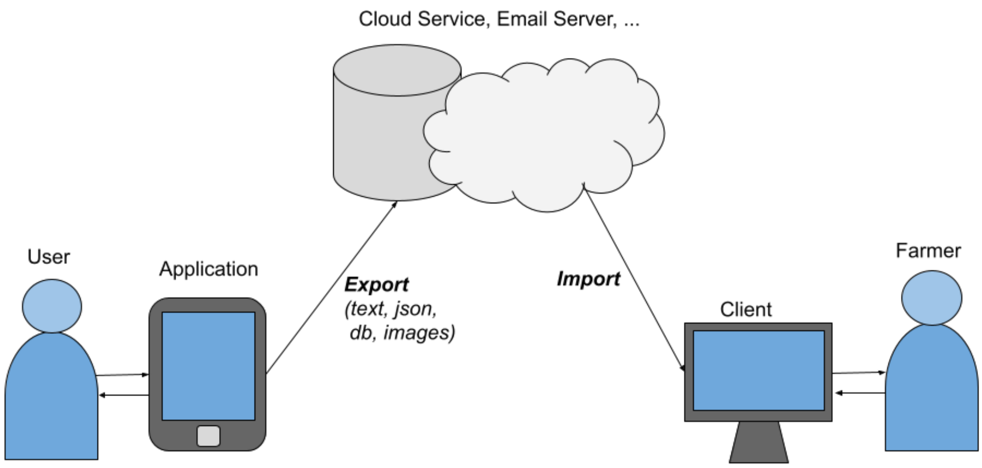

Figure 4.3: Data Export Architecture

4.4 Prototype Design and User Interface

Based on the requirements in chapter 3, rough drafts of the user interface was sketched out on paper. After reaching promising designs, the main sketches were drawn digitally. These mockups provide a way to evaluate the design, consider the navigation between different screens in the app and were used as a reference when implementing the UI. The core parts of the UI is presented and described in this section.

4.4.1 Start Menu

The start menu is the first page the user sees when opening the app. This page shows five clickable buttons. The first button navigates to a list of offline map areas, where the user has the option to download new map areas. The second button navigates to a list of all conducted trips. The third button shows a simple rapport with aggregated data and has the option to export data. The fourth button takes the user to a settings page. The last button creates a new trip if the user is not currently performing a trip. If the user has an active trip this button resumes and displays the active trip. See Figure 4.4 for a sketch of the start menu page.

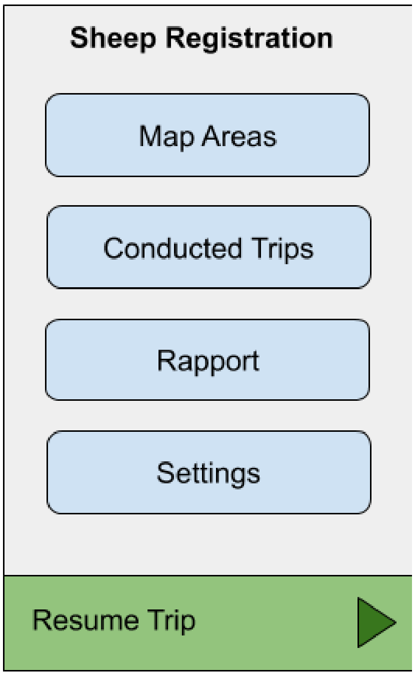

Figure 4.4: Mockup of Start Menu

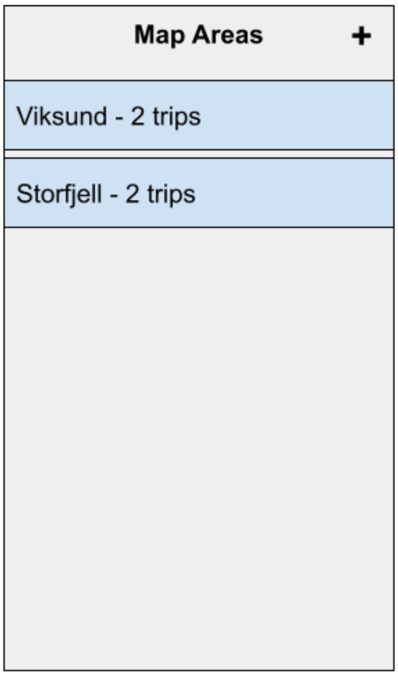

Figure 4.5: Mockup of MapAreas Page

4.4.2 MapAreas

The MapAreas page shows all offline MapAreas the user has downloaded, see Figure 4.5. In the top right the user can click a "+" button to go to the MapAreaDownload page. Clicking any of the list elements takes the user to the corresponding MapAreaDetails page.

4.4.3 MapAreaDownload

Users might not have access to internet when performing supervision trips and should be able to download a map area for offline use, see Figure 4.6. This sketch consists of a map view that is navigable by dragging, and a download button in the corner. The user will navigate to an area on the map he wishes to download. Pressing the download button will download the area currently displayed in the map view and store the area on the device. Before the download starts, the user will be prompted to enter a name for the area. After the download is completed the area can be used when performing supervision trips.

Figure 4.6: Mockup of MapAreaDownload Page

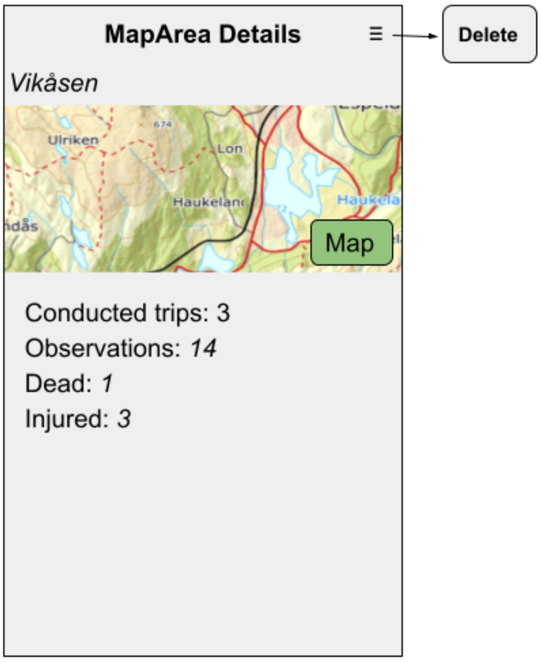

Figure 4.7: Mockup of MapAreaDetails Page

4.4.4 MapAreaDetails

The sketch in Figure 4.7 lets the user see details about each map area such as the amount of trips performed in the area, how many observations have been made or how many dead and injured animals have been spotted. Clicking the map button lets the user navigate in the map, which contain markers of all observations in the area. The menu in the top right lets the user delete the map area with all associated trips and observations.

4.4.5 ConductedTrips

As a shepherd could be interested in the trail of an earlier trip or what observations he has recently made, all conducted trips are stored in the app. These are presented in a list, ordered from most to least recent. The list is filterable on year by clicking on the filter button in the top right. Clicking on a trip will show details about it such as the trip trail and the list of observations. See Figure 4.8 for the mockup.

Figure 4.8: Mockup of ConductedTrips Page

4.4.6 TripDetails

Users should be able to see the details about any selected trip with a quick glance. In Figure 4.9 the name of the trip is at the top, with the associated map area displayed beneath. Details such as date, duration and distance walked are presented, as well as how many observations were done and how many dead and injured animals have been spotted on this trip. Clicking the map button will show an interactable map containing the trail and all the observations of the trip.

For a finished trip the top right menu will appear, with a menu item for exporting the trip data. Clicking this button will let the user export all data in the database related to the trip, such as observations and trip trail including images for each observation. The user gets export options such as sending the files using email or uploading the files to a cloud file service.

4.4.7 TripMap

When performing a trip the users location is logged using GPS, creating a trail. This trail is shown on the trip map, see Figure 4.10. It also provides the ability to create observations by long-pressing an area on the map. This prompts the user to select one of the five different types of observations, which leads the user to a new screen to enter details about the observation. The user is able to drag the map to navigate and see the trail (black line) as well as all observations made on the trip

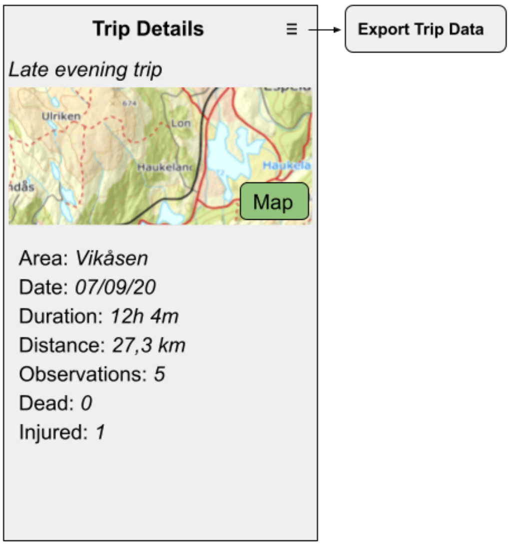

Figure 4.9: Mockup of TripDetails with Export Options

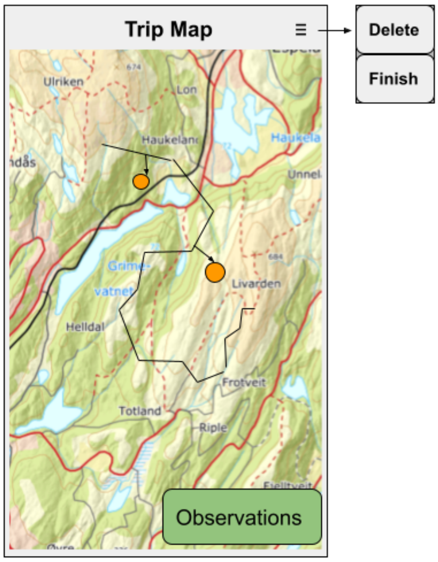

Figure 4.10: Mockup of TripMap Page

(orange circles). Clicking the observations button shows a list of all observations for the trip. The top right menu button lets the user delete a trip as well as finish a trip if it is not already finished. The finish menu item button will only appear if the trip is active.

4.4.8 HerdObservation

When creating a herd observation the user is presented with a screen for entering animal counts in the herd, see Figure 4.11. A list of animal types is shown, corresponding to the types described in Table 2.2. For each type the corresponding count is shown, as well as buttons to increment and decrement the count. Clicking on the number itself allows entering the count value with a numeric keyboard.

In the top right is a menu with the option of adding a secondary lookout point. Clicking this button will create a new TripMapPoint of the current location and associate it with the Observation. This lets the user register a secondary lookout point when observing a herd.

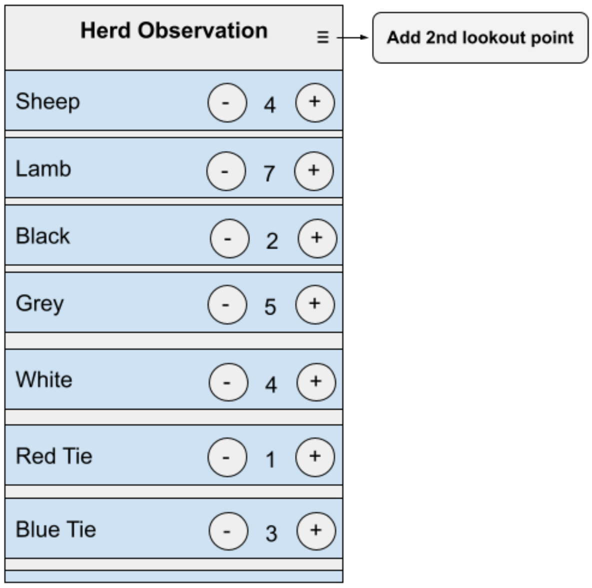

Figure 4.11: Mockup of HerdObservation

4.4.9 Dead and Injured Animal Observation

When storing an observation of a dead or injured animal it should be possible to store the identification number of the animal, notes and images of what is observed, see Figure 4.12. The same general UI is used for these two cases, only altering the observation type displayed at the top.

4.4.10 Predator and Environment Observation

An observation of a predator and of an environmental condition share the same underlying structure, with the top label showing the type, see Figure 4.13. These observations consist of notes about the observation as well as images of what is observed.

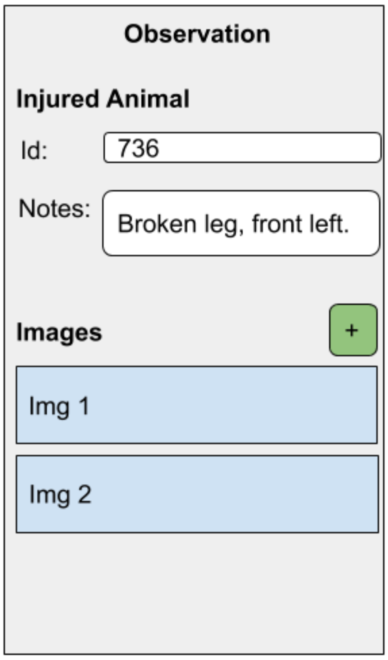

Figure 4.12: Mockup of Dead/Injured Animal Observation

Figure 4.13: Mockup of Predator/Environment Observation

4.4.11 Swiper

To accommodate the need for entering animal count data without having to look at the device, the swipe gesture is used. The swipe gesture was chosen as it requires little precision, but still provides enough functionality for the required actions. The alternative of using multiple big buttons would require more precision. Users would need to click at the correct area of the screen to perform an action, likely leading to errors and less accurate data input.

The user should be able to enter the amount of animals while simultaneously observing a herd. The user is not to enter the total amount of animals, but the amount of each animal type in the herd, as described in subsection 4.4.8. There are nine types of animal counts the user can store, as described in Table 2.2. The four swipe directions up, down, left and right are used. At any point one count type is selected. This can be incremented by swiping up and decremented by swiping down. Swiping left and right changes the count type to the previous and next respectively. Table 4.9 shows the four different swipe gestures and their effect. See Figure 4.14 for the sketch of the swiper page. All labels update based on the swipe, such as when changing the count value or the count type.

Swipe direction	Action
Up	Increment the value of the selected count type
Down	Decrement the value of the selected count type
Left	Change to the previous count type
Right	Change to the next count type

Table 4.9: Swipe Directions and Corresponding Actions

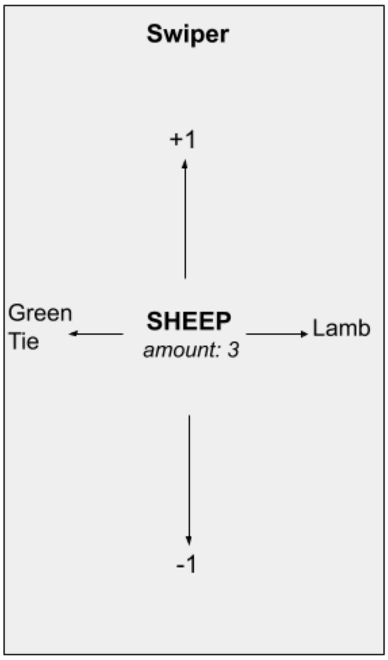

Figure 4.14: Mockup of Swiper

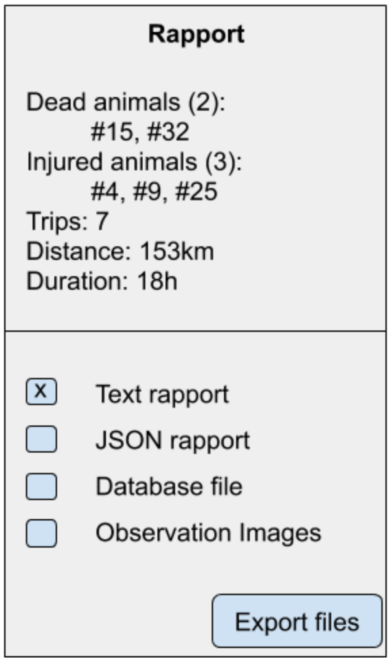

Figure 4.15: Mockup of Rapport and Export

4.4.12 Rapport and Export

All observations and trips are stored on the device, making it possible to calculate aggregated data and display it as a simple rapport. The rapport as well as the full database file and image files can be selected for export by checking the corresponding checkbox. See Figure 4.15 for the mockup. Clicking the export button lets the user select between options such as sending files using email or uploading files to a cloud file service.

Chapter 5

Implementation

This chapter describes how the application is implemented. First the development process is discussed. Then the technical choices are presented and justified. Subsequently the application prototype is laid out and described.

5.1 Development Process

A main focus in selecting a Software Development Process for this project was to provide a structured development process for one person, without incurring a large overhead. It was also desired to follow a process where new and changing requirements could quickly be adapted to and implemented. For this reason the Waterfall process was not considered due to its strict and rigid process of producing all requirements up front. Agile methods that are designed to respond to frequent changes in requirements are considered.

While Scrum provides structure using a backlog and sprints, it would incur too large an overhead for one person to hold daily and weekly meetings and producing sprint backlogs [13]. The meetings would also need to be modified as it is not immediately apparent how a meeting with only one person should be held, or whether such meetings should be held at all.

Another alternative was to follow the general agile approach and use a Kanban board to keep track of development tasks [14]. This keeps development tasks organized and prioritized while providing little overhead. Additionally the progress of the project can be determined with a quick glance at the Kanban board. This approach also supports new and changing requirements as development tasks can easily be added, deleted, edited, moved or reordered. Due to these benefits the agile approach was chosen as the development process for this project.

The Kanban board is a simple yet effective way to organize development tasks. It consists of a number of columns, each containing a set of tasks. An adapted version of this board was chosen to structure and organize the development process.

Columns were divided into Todo, NextUp, Doing and Done. ***Todo*** acts as a backlog containing all planned development tasks, ordered by importance. ***NextUp*** acts as a prioritized list of development tasks for the current iteration of development. When this column is empty a new set of tasks are moved from the Todo column to the NextUp column. ***Doing*** contains the current development tasks, taken from the NextUp column. This column strives to contain a single element at any point in time to keep work effort focused. An important aspect of Kanban is to limit the number of development tasks in progress simultaneously. When the current task is completed, it is moved to the Done column and a new task is selected from the NextUp column. ***Done*** contains completed and tested development tasks. See Figure 5.1 for an example of the workflow using Trello.

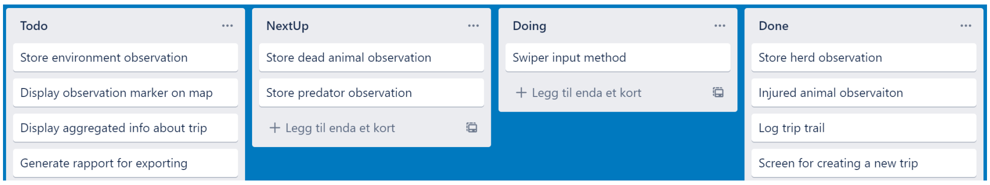

Figure 5.1: Trello Kanban Board

5.2 Android SDK

The Android SDK provided by Google is used for developing applications for the Android platform. This includes software development tools such as an IDE and libraries for Android. Using the Android SDK traditionally meant developers would use the Java programming language, but in recent years Google has introduced and recommended the use of Kotlin to develop Android applications. To keep with recommendations and stay modern, the Kotlin programming language was chosen instead of Java.

To follow best practices Android Jetpack will be used. It is a set of libraries, in addition to the Android SDK, meant to make Android development easier, safer and faster by reducing boilerplate code and simplifies following best practices. Android Jetpack was introduced in 2017 and gathers existing libraries from the Android Support Library as well as adding new libraries. Using Android Jetpack will affect the architecture of the application.

The Android SDK comes with the Android Studio IDE for developing Android apps. This will be used as it is the recommended IDE for Android development. It contains features such as linting to warn for possible bugs and enables easily running the app either using an emulator or a physical device. Additionally it contains a number of templates of Fragments to reduce the time spent writing boilerplate code. A visual layout editor allows for simpler creation of the UI with an additional navigation graph editor simplifying the implementation of navigation between Fragments.

5.3 Storage

As discussed in section 4.3 the Room Persistence Library is used for data storage. Using Room, entities are defined as Kotlin data classes. An example is the TripMapPoint entity, shown in Code listing 5.1. The @Entity annotation specifies TripMapPoint as an entity, sets the table name and describes the foreign key relationship to the Trip entity. The @PrimaryKey annotation marks the id field of the entity and @ColumnInfo enables the specification of the column name in the SQL table.

Code listing 5.1: TripMapPoint Entity

```
@Entity(tableName = "trip_map_point_table",
    foreignKeys = [
        ForeignKey(
            entity = Trip::class,
            parentColumns = arrayOf("trip_id"),
            childColumns = arrayOf("trip_map_point_owner_trip_id"),
            onDelete = ForeignKey.CASCADE
        )
    ])
data class TripMapPoint(
    @PrimaryKey(autoGenerate = true)
    @ColumnInfo(name = "trip_map_point_id")
    var tripMapPointId: Long = 0L,

    @ColumnInfo(name = "trip_map_point_lon")
    var tripMapPointLon: Double,

    @ColumnInfo(name = "trip_map_point_lat")
    var tripMapPointLat: Double,

    @ColumnInfo(name="trip_map_point_date")
    var tripMapPointDate: Date,

    @ColumnInfo(name = "trip_map_point_owner_trip_id")
    var tripMapPointOwnerTripId: Long

    ){}
```

All access to the database is provided by a Data Access Object (DAO). The code in Code listing 5.2 shows an excerpt of the AppDao interface, defining methods that can be called to insert, delete, retrieve and modify data. All methods are annotated. Some annotations have provided a custom SQL statements in the cases where the queries cannot be inferred automatically. Some of the methods return a LiveData object wrapping the data object. This lets changes in the database propagate to the ViewModel and UI automatically. All ViewModels access local storage through the DAO, using the methods defined in the interface.

Code listing 5.2: AppDao Interface Excerpt

```
@Dao
interface AppDao {
    @Insert
    fun insert(mapArea: MapArea): Long

    @Query("SELECT * FROM map_area_table WHERE map_area_id = :key")
    fun getMapArea(key: Long): MapArea?

    @Query("SELECT * FROM map_area_table WHERE map_area_id = :key")
    fun getMapAreaLD(key: Long): LiveData<MapArea?>

    @Query("SELECT * FROM map_area_table ORDER BY map_area_name ASC")
    fun getMapAreasLD(): LiveData<List<MapArea>>

    @Query("DELETE FROM map_area_table WHERE map_area_id = :key")
    fun deleteMapArea(key: Long)
}
```

5.4 Map Framework and Tile Service

The application needs to store and display map areas. For this purpose a map framework and a map tile service is needed. This section describes the map framework and tile service used.

Map Framework - OSMDroid

While Google provides a UI component to display a map, it does not suit the purpose for this application. The Google Maps Android SDK requires API keys and occasional contact with servers to be used, being substantial drawbacks for the purpose of this app [15]. Mapbox is a popular alternative to Google Maps API, but has the same undesired features as described above [16]. It also requires payment plans which make it even less compelling.

Another option is OSMdroid [17]. This library does not require any server interaction and supports setting a custom source for the map tiles. This means that any server delivering tiles can be used. OSMdroid also supports storing map tiles locally on the device, to enable displaying maps offline. The library is open source and available at no cost. As this library meets all requirements without requiring server interaction or payment, it was chosen to be used in the application.

The concept of map tiles is central in map frameworks. A map tile is a small square image of a specific part of a map, specified by an x and y coordinate and a zoom level. Figure 5.2 show map tiles at a given zoom level for different x and y coordinates.

Figure 5.2: Map Tiles

Map Service - Kartverket

The application is designed for farmers in Norway and therefore needs access to a map spanning all of Norway. While Google Maps provides a map service spanning the entire country, it is not a topographic map with terrain details. Kartverket provides topographic maps free of charge [18]. It also provides multiple APIs that can be used by the OSMdroid map library, both for showing a map and for downloading a map area for offline usage. The application interacts with the API through the OSMdroid library as shown in Figure 5.3.

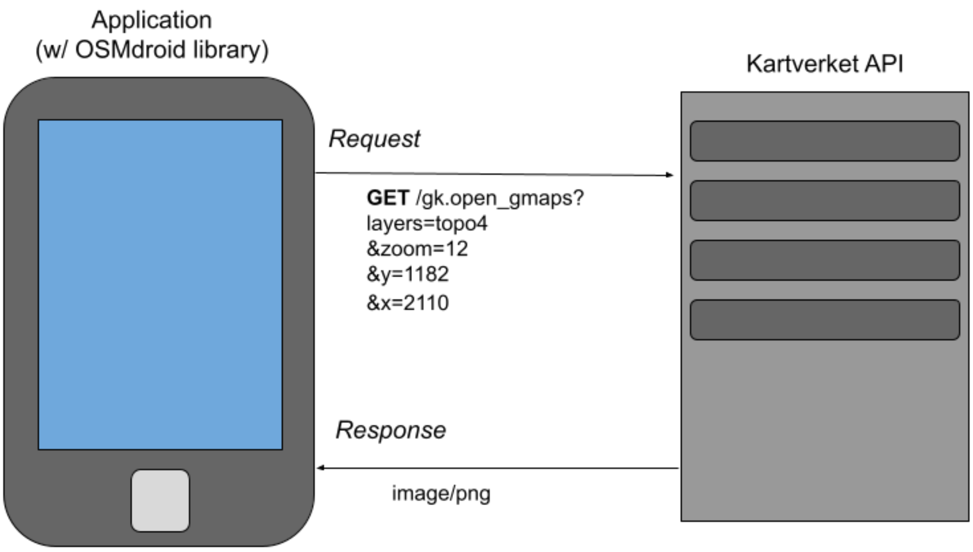

Figure 5.3: OSMdroid and Kartverket API Interaction

The chosen API from Kartverket is a Cache-server that delivers map tiles using Web Map Tile Service (WMTS). The base URL for the API is `https://opencache.statkart.no/gatekeeper/gk/gk.open_gmaps`, requiring four parameters for specifying the desired map tile. The parameters are presented in Table 5.1.

Parameter Name	Description	Example Value
layers	The map type, e.g. topographical	topo4
zoom	The zoom level of the tile	16
x	The x coordinate of the tile	2110
y	The y coordinate of the tile	1182

Table 5.1: Kartverket WMTS API Parameters

5.5 Location Service

To enable the logging of the users trail during a trip, a separate background service had to be made. The Location Service logs the position of the mobile device at the distance and time intervals specified by the user in the settings. Performing the trail logging in the background also enables logging even when the application is in the background or the phone screen is off. The logging is started when the user either creates or resumes a trip and thereby enters the TripFragment. This fragment is responsible for starting and stopping the Location Service. If the fragment is popped from the back stack the trip is paused and logging stops. If the user ends the trip, the logging is stopped permanently. The Location Service stores new TripMapPoint instances to the database using the DAO. These instances make up the trip trail. A notification is displayed when the Location Service runs and tracks the location of the user, see Figure 5.12.

5.6 Application Prototype

This section discusses the implementation of the application prototype. The implemented UI of the application is based on the initial mockups in section 4.4, but has evolved during development. Hence the final UI differ somewhat from the mockups. Each screen from the design sketches is implemented as a Fragment. The main Fragments of the application are presented and the functionality of UI elements are explained.

Application code repository: https://github.com/svein007/SheepLogger.

5.6.1 Start Menu

The final implementation of the start menu Fragment is shown in Figure 5.4. The four buttons for map areas, conducted trips, rapport and settings have gotten icons to supplement the text. This will enable the user to recognize the icon instead of reading the text. The bottom button has information about the currently active trip, if there is one, otherwise it lets the user create a new trip. Clicking any of the buttons navigates to another, corresponding Fragment.

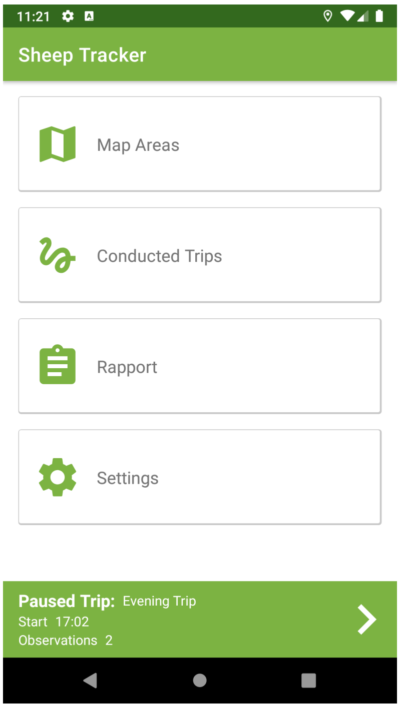

Figure 5.4: Start Menu Fragment

5.6.2 MapAreaDownload

The MapAreaDownload Fragment keeps most of the design as presented in the mockup. See Figure 5.5 for the final design. The lower right button has an icon instead of text. Clicking the download button will show a dialog where the user is asked to enter the name of the map area, see Figure 5.6. Clicking download will start the download of map tiles of the area shown on screen. In the top right another button is added. Clicking this button will center the map on the location of the user and keep it centered as the user moves around.

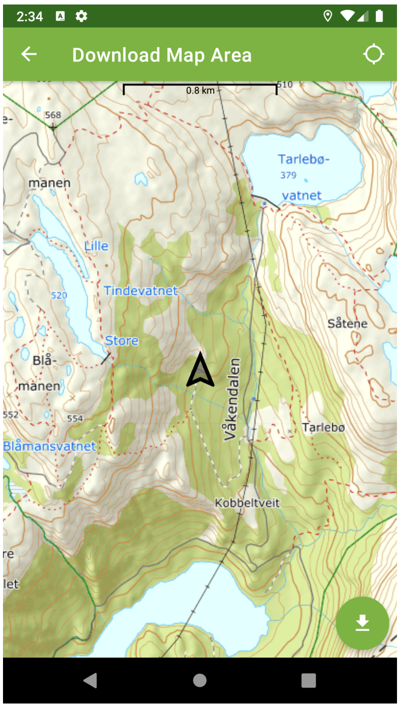

Figure 5.5: MapArea Download Fragment

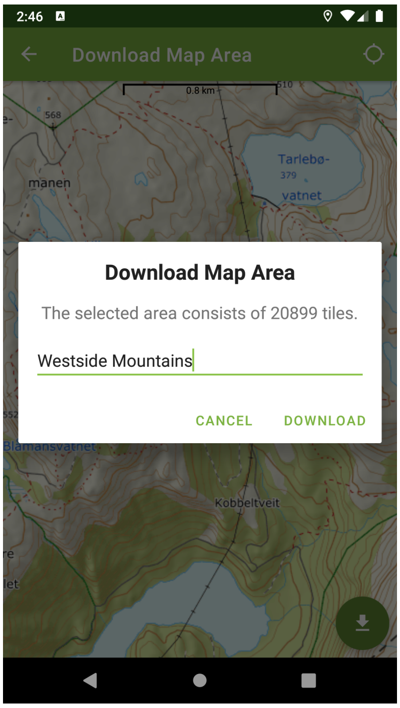

Figure 5.6: MapArea Name Dialog

5.6.3 MapAreas and MapAreaDetails

All downloaded MapAreas are stored on the device. Figure 5.7 shows the MapAreas Fragment which contains a list of all the MapAreas downloaded by the user. Elements in the list show details about zoom levels and how many trips have been performed in the area. Clicking on one of the map areas takes the user to the MapAreaDetails Fragment. The plus button takes the user to the MapAreaDownload Fragment.

Figure 5.8 shows the final UI of the MapAreaDetails Fragment. Most aspects of the UI follows the design of the mockup in section 4.4. This page shows the user more details about each MapArea such as how many trips and observations have been done in the area and how many dead and injured animals have been observed. Clicking the map icon in the top right corner of the map lets the user navigate in the map, which contains markers on the locations of all observations performed in the area. See Figure 5.9 for the implemented UI of the map. This map UI was not part of the original design, but was added as it is a useful visualisation of all observation data in a given map area for the farmer. Clicking the top right menu shows the option to delete the map area.

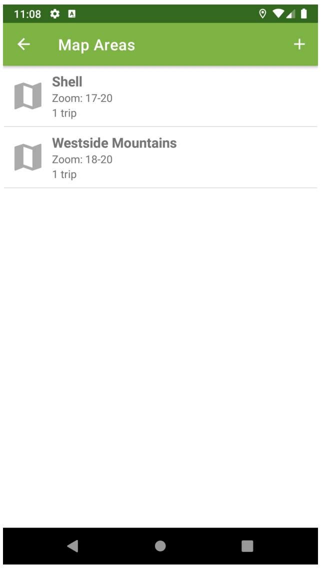

Figure 5.7: MapAreas Fragment

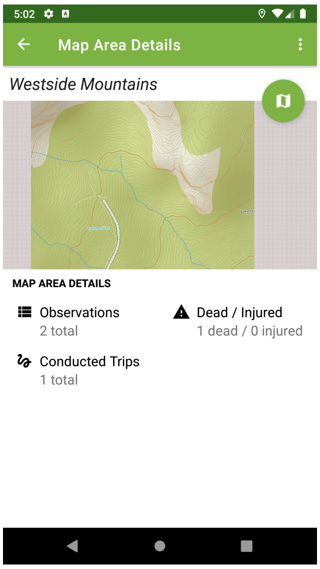

Figure 5.8: MapAreaDetails Fragment

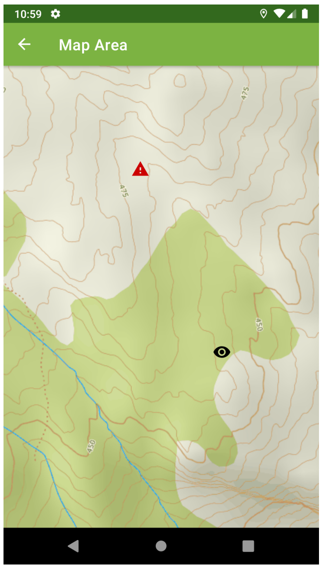

Figure 5.9: MapArea Fragment

5.6.4 NewTrip

If there is no active trip and the user clicks the bottom button in the start menu, the NewTrip Fragment is shown, see Figure 5.10. This UI lets the user enter a title for the trip in the top text field and select the MapArea it will be performed in from the list below. The list contains all offline MapAreas stored on the device. Clicking the Start button in the top right takes the user to the TripMap Fragment.

5.6.5 TripMap

The TripMap Fragment shows the trip trail and marks all observations on the map with icons, see Figure 5.11. Every observation is connected with a line to the trip trail, indicating where the user was positioned when the observation was done. Clicking on an observation marker shows some details about it, see Figure 5.15. The white arrow in the green circle at the top of the map lets the user display and hide a small overlay of the count of herds, as well as dead and injured animals observed on this trip. The upper right icon centers the map on the users position and keeps the map centered on the user while the user is moving. Clicking the lower right icon takes the user to the Observations Fragment, showing a list of observations for the trip to the user.

The TripMap Fragment is also the main UI the user sees when performing a trip. When the trip is active, new lines will be appended to the trip trail as the user moves. As seen in Figure 5.12, a notification is displayed when the trip is active

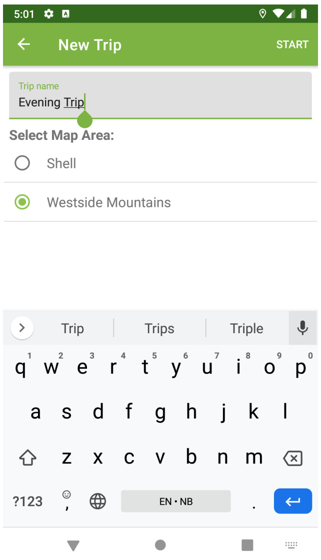

Figure 5.10: NewTrip Fragment

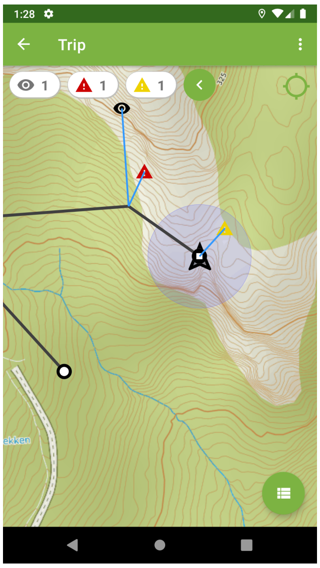

Figure 5.11: TripMap Fragment

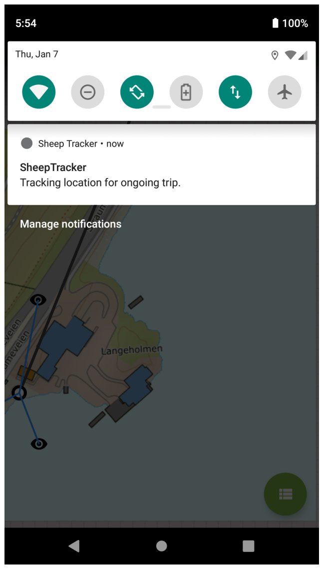

Figure 5.12: Location Tracking Notification

and the location is being tracked. Long-pressing on any area of the map lets the user create a new observation at that location. This opens a Dialog with five options corresponding to the five observation types, see Figure 5.13. Selecting an observation type takes the user to a Fragment where the user will enter details about the observation.

5.6.6 Observations

The user can reach the Observations Fragment from the TripMap. This UI shows the list of observations for the selected trip, as seen in Figure 5.14. The observations are ordered from most to least recent. In the top right a filter button reveals a drop-down menu to filter the observations based on observation type. Each entry in the list of observations has an icon and a label denoting the observation type. The date and time of the observation is shown as well as a short description specific to each type. Clicking on an element in the list will show details about the observation by navigating to another Fragment.

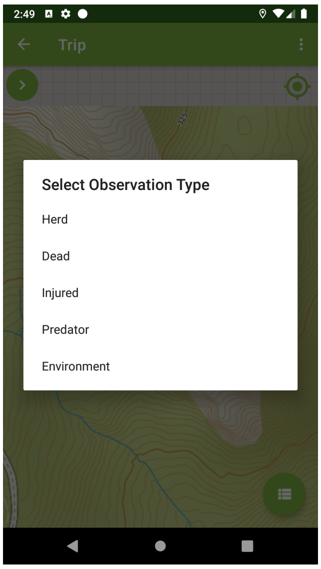

Figure 5.13: New Observation Dialog

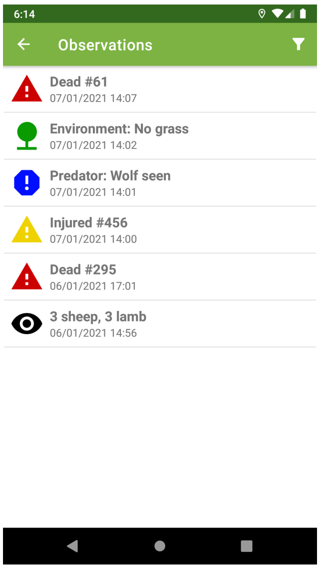

Figure 5.14: Observations Fragment

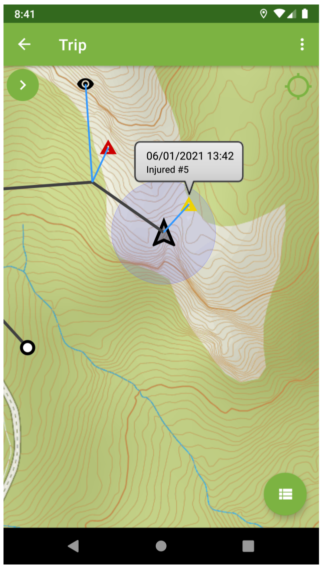

Figure 5.15: Observation Marker on Click

5.6.7 ConductedTrips, TripDetails and Trip Export

As seen in Figure 5.16, all conducted trips are presented in a list with details such as area and start date. As the list of trips will grow long over time, trips can be filtered by the year they were conducted. This simplifies finding trips performed in a single season. Filtering is done by clicking the filter icon in the top right and selecting the desired year. Clicking on one of the trips takes the user to the TripDetails Fragment.

The TripDetails Fragment, see Figure 5.17, shows a preview of the area with a trip trail, including visual markers of all observations. It also shows information such as the area the trip was performed in, the date it was performed, its duration, the distance the user walked, the observation count and the count of injured and dead animals. Clicking the button in the top right of the map takes the user to the TripMap Fragment.

In the top right of the TripDetails fragment a menu will appear if the currently displayed trip is finished. This menu will contain a button that exports all data of the trip, see Figure 5.18. Clicking this button will make a dialog appear where the user selects how the data should be exported. This could be sending the files using email or uploading it to a cloud file service. The functionality lets a user send the collected data for a trip as soon as he has finished it. This way the farmer is always updated on the most recent trips.

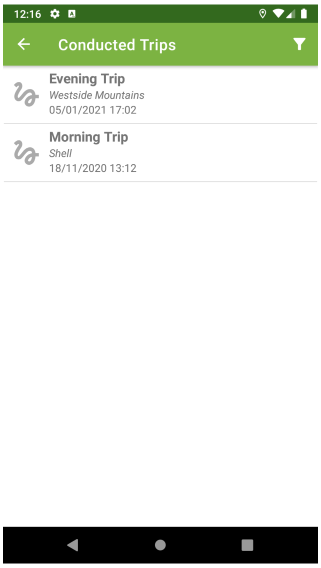

Figure 5.16: ConductedTrips Fragment

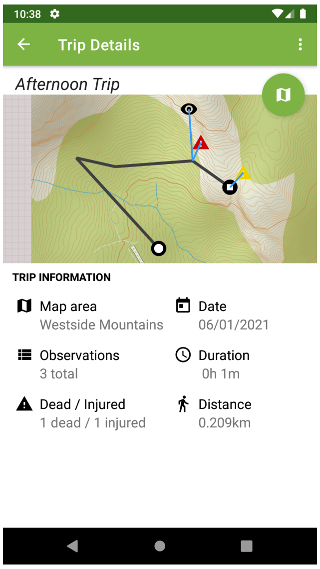

Figure 5.17: Trip-Details Fragment

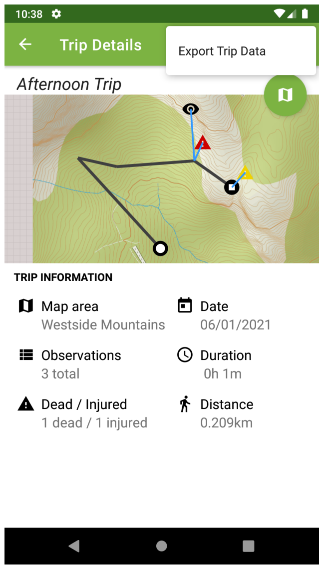

Figure 5.18: Trip Export

5.6.8 Herd Observation

One of the observation types is a herd observation, storing multiple counts, such as one for sheep, one for lamb, and so forth. As seen in Figure 5.19 these counts are presented in a scrollable list with options for incrementing and decrementing the count, as well as using a keyboard by clicking on the number. At the top right of the list the amount of lambs seen as well as the expected lamb count based on different tie colors seen is presented. Additionally each observation supports notes as text input. At the top a map shows where the herd was located as well as the lookout point where the user spotted the herd from. Clicking "Add" in the top bar will store the new herd observation, while clicking the back arrow will not. Clicking the three dots at the top right followed by "Swiper" takes the user to the Swiper Fragment.

5.6.9 Swiper

The Swiper Fragment gives the option to enter counts of animals for a Herd Observation without looking at the device. The UI is shown in Figure 5.20. The center label shows the selected count type and the current count. Swiping up will increment the count, while swiping down will decrement the count. Swiping right selects the next count type, while swiping left will select the previous count type, in the same order as in the list in the HerdObservation Fragment. The arrows in the UI indicate the effect of each swipe direction. Each swipe will update the labels on the screen as well as give verbal output to the user, telling what count

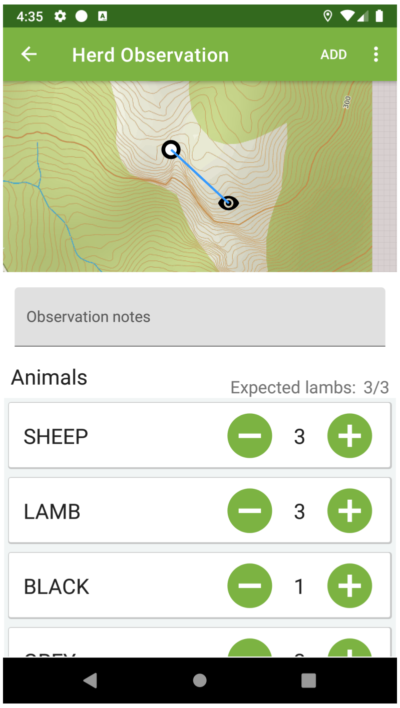

Figure 5.19: HerdObservation Fragment

Figure 5.20: Swiper Fragment

type is selected and what the current count is. This way the user gets feedback from the swipe action and knows the current count, without needing to look at the device. This provides the user the ability to use binoculars without interruption while entering all animal counts of the Herd Observation.

5.6.10 Dead and Injured Animal Observation

Observations of dead and injured animals are similar, and both observation types store the same kind of data. Figure 5.21 show the Fragment when storing data about a dead animal, while Figure 5.22 shows the same Fragment for an injured animal. The label and icon in the top left are different, with the label being either "Dead" or "Injured". In the top right the date and time of the observation is shown. A map shows where the user was located when entering the observation, and where the animal was spotted. The white circle with a black outline shows the location of the user, while the triangle shows the observed animal. The location of the animal corresponds to the location where the user long-pressed on the map to create the observation. The animal identification number input field takes a numeric identification number of the animal, while the text field for notes takes an alphanumeric string. At the bottom a list of images for the observation is presented, with a "+" button for adding a new image either using the camera or by selection one from the phone storage, see Figure 5.23. When a new image is ad-

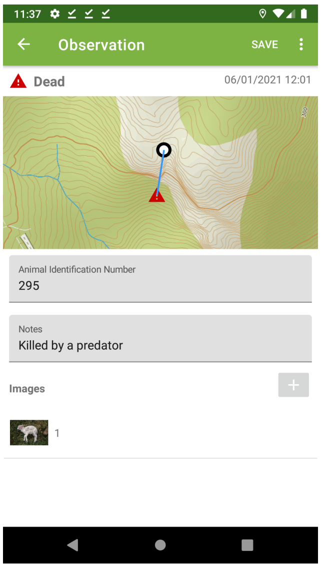

Figure 5.21: DeadAnimal Observation Fragment

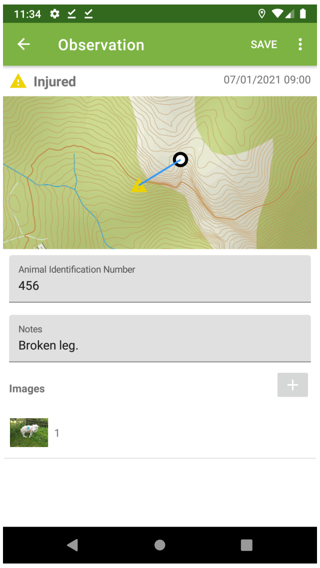

Figure 5.22: InjuredAnimal Observation Fragment

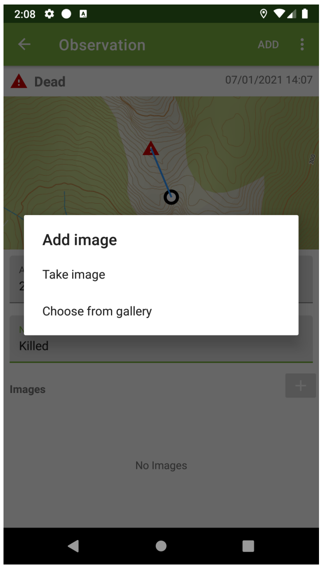

Figure 5.23: Add Image Dialog

ded it is stored on the filesystem of the device and a new instance is created of ImageResource, with a reference to the image file. Clicking the Add button in the top right will store the observation on the device.

5.6.11 Predator and Environment Observation

Observations of predators and the environment are similar to observations of dead and injured animals, but don't need the animal identification field. The UI for storing predator and environment observations show the observation type name in the top left, the date and time of the observation in the top right, a map with the location of the user and observed object, a text field for entering notes and a list of images for the observation. The difference between the Fragment for predator and environment is the top left icon and label, as seen in Figure 5.24 and Figure 5.25.

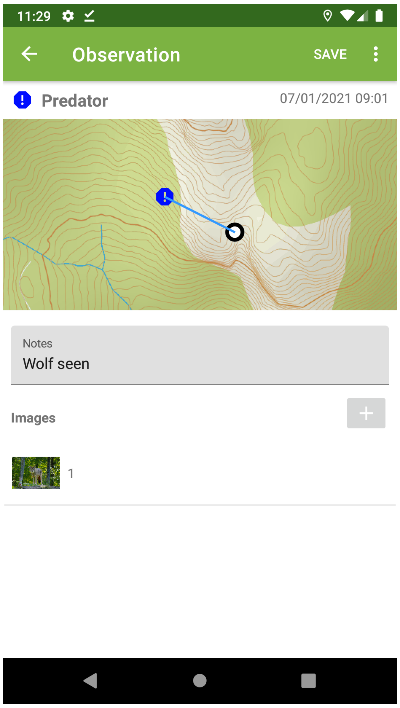

Figure 5.24: Predator Observation Fragment

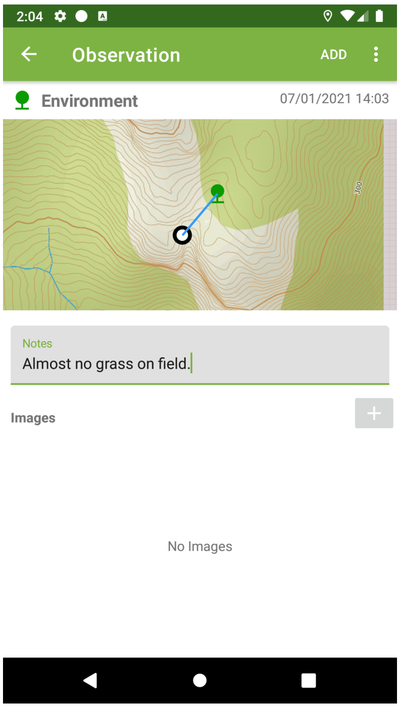

Figure 5.25: Environment Observation Fragment

5.6.12 Rapport and Export

The rapport UI shown in Figure 5.26 presents aggregated data to the user. This presents how many dead and injured animals have been spotted and their identification numbers. Furthermore the number of trips, the total distance traveled and the total duration of all trips is displayed. In the top right the user can click to reveal a drop-down menu for filtering the rapport by year. At the bottom the user can select the format of the exportable rapport and whether to include the full database file and images of observations. The year filtering and file type selection is shown in Figure 5.27. Clicking the Export Files button brings up options of where to export the data, such as Google Drive or Gmail, see Figure 5.28. Other export options may also be presented, based on what apps are installed on the device.

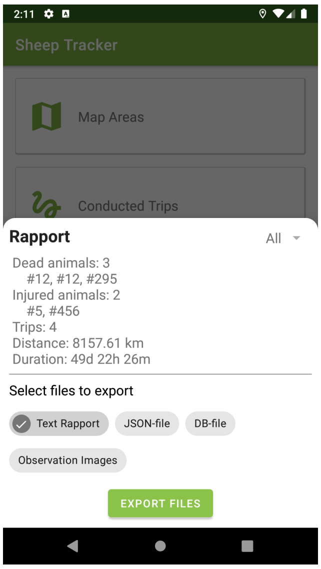

Figure 5.26: Rapport Dialog

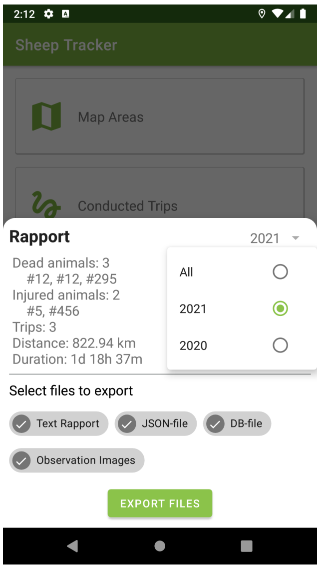

Figure 5.27: Rapport Dialog Filtered on Year, With Export Options Selected

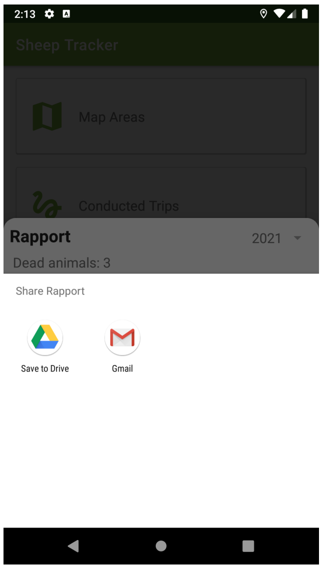

Figure 5.28: Rapport Export Share Menu Dialog

5.6.13 Settings

A separate Settings Fragment lets the user configure the application to best suit his needs, see Figure 5.29. The top two options lets the user change the GPS distance interval and time interval. This gives the opportunity to get trip trails with frequent location logging or rare location logging. The bottom option lets the user request the app to get permissions for Location and Storage if the user has not accepted them already.

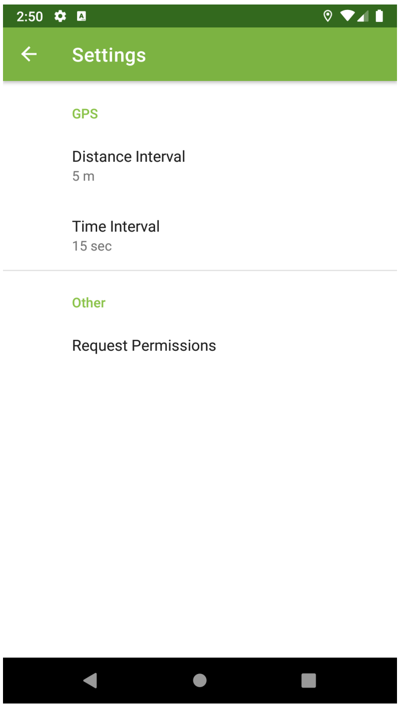

Figure 5.29: Settings Fragment

Chapter 6

Testing and Test Results

This section describes how the application prototype was tested and shows how usability testing was performed to measure the effectiveness, efficiency and user satisfaction of the application. The results presented were obtained through observation and interviews.

6.1 Usability Testing

The usability of the application prototype was tested based on the definition of usability in ISO 9241-11 [1]. This ISO standard defines usability as "*The extent to which a product can be used by specified users to achieve specified goals with effectiveness, efficiency and satisfaction in a specified context of use.*" This reveals three important factors when measuring usability: users, their goals and their context of use.

The users of the application are primarily sheep farmers and designated shepherds, as described in the preliminary study. Their goals are given by the use cases presented in chapter 3. When the users are performing supervision trips they encounter different environments. These environments represent the context of use.

Testers

The sheep farmer is not necessarily the person performing supervision trips. The farmer can have others perform the trip for him. The person performing the trip takes the role of a shepherd and might be an employee, a friend or a family member that does not know much about the domain. Therefore the target group of testers should consist of a variety of people with different levels of knowledge about the domain. Due to the variety of domain knowledge of the testers, a short introduction to the problem domain is performed before the test. It was desired to include sheep farmers as participants of the usability test, but due to a lack of available farmers this was not possible.

Tasks

The use cases presented in section 3.2 constitutes the goals of the user. The tasks of the usability test are based on these use cases, representing real world usage of the application. Participants are presented with a scenario description to help imagine a realistic situation when performing the tasks. The usability test can be found in Appendix A. It contains the scenario description and all tasks.

Context of Use

Outdoor areas such as forests, fields and mountains represent the context of use. The environment and weather conditions that a shepherd faces can vary greatly, but testing in different condition with different participants could skew the results. For this reason it was desired to have all participants perform the usability test in the same environment with similar weather conditions. The usability testing was performed on days with no rain, some clouds and a comfortable outdoor temperature. No real animals or predators were part of the testing. The tester was to move from location to location and be told at certain points to perform specific tasks in response to imagined scenarios.

Measures

The ISO standard states three main measures to use in a usability test: effectiveness, efficiency and subjective user satisfaction. To measure effectiveness each task was evaluated based on its completion. A task was either completed with a satisfactory result or not. The time each participant used to complete a task was also measured, representing efficiency. All participants were continuously observed to pick up on behavioural cues as well as explicit verbal statements during the testing. By analysing the three measures it is possible to make an assessment about the usability of the application and satisfaction of the user.

6.2 The Usability Test

When performing the test, every participant was presented with the purpose of the test, the application prototype and its limitations. The only tool the user had was the mobile device with the application installed. It was made clear that the application was being tested and not the user. Additionally each participant was taught how to "think out loud". This means they were encouraged to say what they were thinking while seeing different pages, performing actions or feeling uncertain about how to solve a task. This was done to get an insight into the users mental model, which could reveal bad design decisions, misleading language or confusing UI elements.

Participant were observed while performing each task. The tasks were performed in sequence and each task was finished before the next task was presented, meaning only one task was performed at a time. Notes were taken by the observer during the test, such as what actions the user took, if the user seemed confused and what the user was saying during the test.

The usability test consisted of 14 tasks, outline below. The complete description of all tasks can be found in Appendix A. Most tasks regarded the data input and storage of different observations as this was the main purpose of the application. A couple tasks tested whether the user was able to retrieve useful data stored in the application. The prototype installed on the test device contained one MapArea with multiple Trips and Observations to support the data retrieval in tasks 10 to 14.

Tasks of The Usability Test:

1. Download a map area
2. Create and start a new trip
3. Create an observation of a dead animal
4. Create an observation of a herd - using swiper
5. Create an observation of a herd - without swiper
6. Create an observation of an injured animal
7. Create an observation of a predator
8. Create an observation of a pasture in bad condition
9. End the trip
10. Find the number of trips performed in an area
11. Find the number of dead animals observed in an area
12. Identify a location where many dead animals have been spotted
13. Display a rapport for the year 2018
14. Export a text rapport, observation images and database file for year 2019 to Google Disk

The usability test focused on use cases related to the thesis goal and research questions. The tasks in the usability test reflect common tasks a sheep farmer would perform on the system. The purpose of performing usability tests was to evaluate the application prototype and consider whether the implemented functionality worked as expected and whether it was clear to the user. Through observation and interview, the tests aimed to determine the usability of the system, subjective satisfaction of the participants and get feedback on the application.

6.3 Results of Usability Testing

This section shows the results of usability testing. The results are mostly qualitative, containing notes from observing and listening to the participants as they performed tasks. The completion status and time usage for each task make up the quantitative data gathered. The full set of results are found in Appendix B while charts, aggregated results and the main findings are presented in this section.

6.3.1 Pen and Paper

To have a baseline for comparison, usability testing was performed using pen and paper, with the same scenario and tasks as for the mobile application. This would make it possible to measure improvements and drawbacks, comparing pen and paper with the application. For these tests the user was given pen, paper, a map, binoculars and a camera.

Not all tasks were applicable, like downloading a map area or finishing a trip. The tasks considered relevant for pen and paper were task 3, 4, 6, 7 and 8. All these tasks regard the act of writing down information about different types of observations.

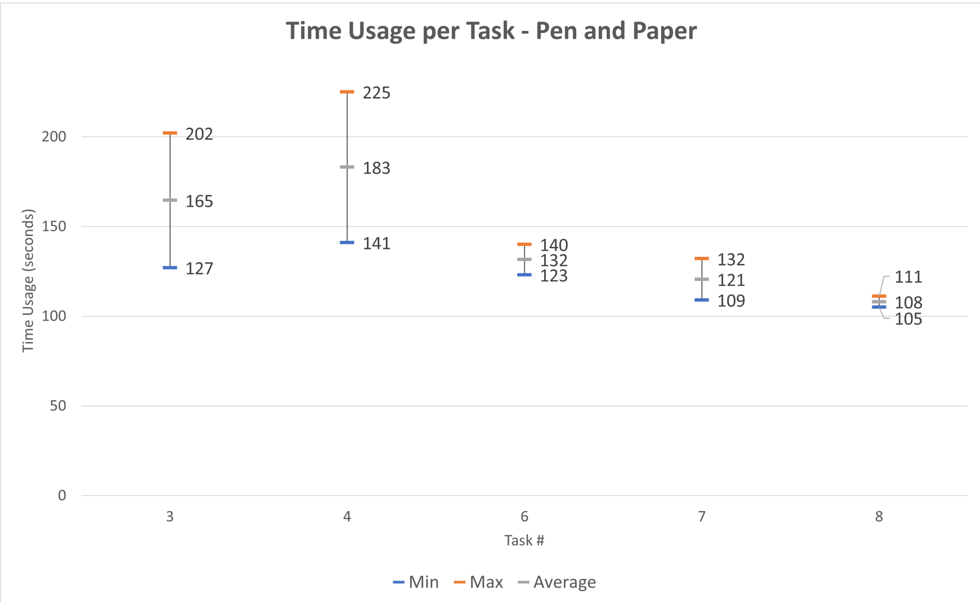

Figure 6.1: Average, Max and Min Time Usage per Task for Pen and Paper.

Figure 6.1 shows time usage for each task done with pen and paper. For task 3 and 4 there are large variations between the min and max time usage, while task 6, 7 and 8 have small variations. The maximum time usage of the tasks seem to decline for tasks performed later in the test. This might indicate that the user has overcome any initial confusions, or might simply be due to differing complexity of different tasks. The minimum time usage is more stable from the first to the last task performed, indicating that if a user has few or no confusions to start with, the limiting factor might be the pen and paper approach itself rather than the experience of the user. Participants use at least 100 seconds to perform each of the tasks in the test.

Participant 1

This participant completed all tasks successfully. It appears that the time usage stays reasonably stable across the tasks, with a range of 30 seconds. Consistently around 50 seconds was spent finding the latitude and longitude of the observation using the map. Performing task 4 the user was not able to enter animal counts of different animal types for the herd while looking in binoculars.

The participant stated that using pen and paper quickly became repetitive, having to write a lot per observation. It was also expressed that finding the coordinates of each observation was tedious and that automating it would enable the user to focus on the actual content of the observation. For the raw test results for participant 1, see Table B.1.

Participant 2

Participant 2 also completed all tasks successfully. Time usage between tasks varied with up to two minutes. This participant spent consistently around 85 seconds to find the latitude and longitude of the observations on the map. The participant did mention that due to possessing limited knowledge about maps it might take some time finding the correct coordinates for each observation. This is supported by the results.

The user had to shift focus between looking in binoculars and writing down herd information for task 4. It was mentioned that having to power on the camera, take an image, note the time of when the image was taken and then powering it off was cumbersome. It would also require additional effort when the image for a specific observation later needs to be retrieved. For the raw test results for participant 2, see Table B.2.

Findings - Pen and Paper

The usability test using pen and paper revealed a number of difficulties with the approach. Firstly it requires a separate camera or phone if the user wants to take pictures of an observation. It requires either a GPS device or a map for registering the location of an observation. The test also gave some measures on how fast the pen and paper method is, with most tasks taking around two to three minutes. The time usages does not seem to indicate a big problem, but finding the coordinates of observations on the map made up a large portion of the time usage for each observation.

While most data for each observation was accurate, the location was just an estimate as it proved hard to get an exact latitude and longitude from the map. This approach for determining the location also seemed dependent on some knowledge about maps and coordinates. It became apparent that it is hard to make notes on paper while using binoculars. Both participants stated aspects of this approach that were not satisfactory, such as repetitiveness and cumbersome tasks. Note that the test was only performed on two participants and that different users might get other results.

6.3.2 Mobile Application Prototype

Four participants performed the usability test on the application prototype. The results of these tests are presented.

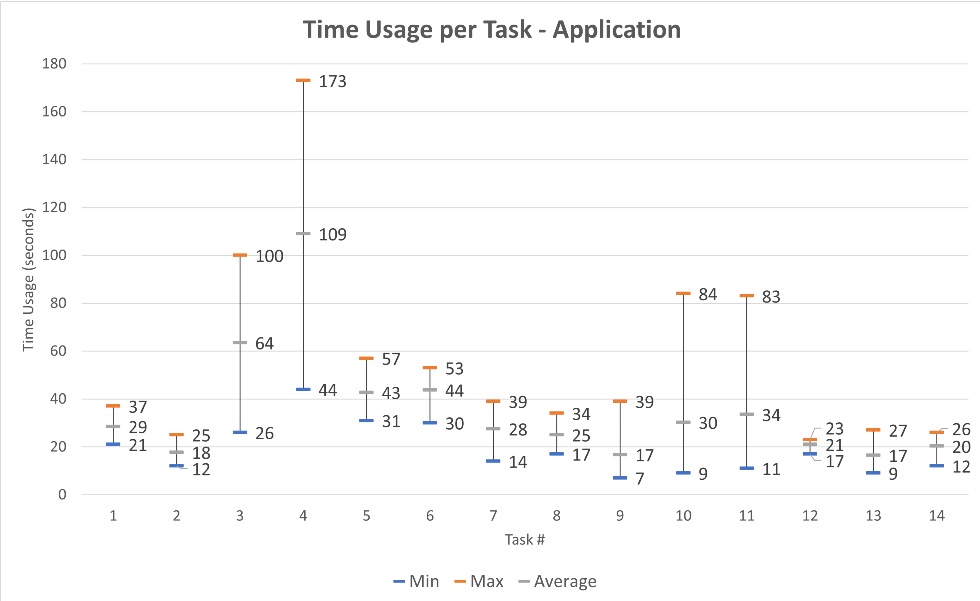

Figure 6.2: Average, Max and Min Time Usage per Task for The App.

Figure 6.2 shows the time usage per task done with the mobile application. It is clear that with even just four participants the time usage can vary greatly between participants when performing the same task, such as for task 3, 4, 10 and 11. Other tasks have a lower variation, such as task 12 with only six seconds separating the slowest and fastest participant. Time usage differs between tasks, with some tasks having a considerably higher average than others. Registering complex observations of herds using binoculars (task 4) takes longer than registering simpler observations such as the state of a pasture (task 8). Looking at the minimum time usage for each task shows that all tasks can be done in less than a minute with the app.

Participant 1

The first participant managed to complete all tasks, but encountered some challenges along the way. When prompted with the task of creating a new observation for a trip, the user spent excessive time discovering that long-pressing an area on the map would create an observation at that area. After discovering the long-press in task 3, creating observations for the other tasks was not a problem. This reflects the tendencies identified in Figure 6.2 regarding a higher time usage for the first observation done by the user.

While performing task 4 the user spent excessive time finding the swiper page and understanding the swiper functionality. This resulted in it being the task with the highest time usage. Once the swipe gestures were understood the user managed to enter the correct data without looking at the device screen.

For task 10 and 11 the user was asked to retrieve information residing in the app. The participant looked at the ConductedTrips page and manually counted the number of trips and the number of dead animals for each trip in the given MapArea. This resulted in the correct information, but the same information could be found at the MapAreaDetails page. It is apparent that the user did not know about this page containing the requested information. This manual approach resulted in a higher than desired time usage for task 11.

For task 12, the visualisation of observations for all trips in a MapArea enabled the user to quickly identify a cluster of dead animals. Both task 13 and 14 were performed quickly and correctly, with little time spent finding the correct pages in the app.

Participant 2

The second participant had some knowledge about the domain and is an experienced mobile user. All tasks were completed correctly and no task took longer than one minute to perform. The user was able to find the correct pages for each task and interact with the correct elements to efficiently perform tasks. Task 4 took longer than task 5 which shows that the user spends more time entering a herd observation using the swiper than using buttons for increment and decrement. The efficiency of this user indicates that when a user is comfortable with and understands the user interface, the tasks can be performed quickly and accurately.

Participant 3

The third participant had little knowledge about the domain and little experience with the Android platform. The user managed to complete all tasks correctly but some tasks had a high time usage due to confusion and uncertainty. When creating a new observation in task 3 the user became unsure of how to save the observation after entering its details.

While performing task 4 the user stated that the word "Swiper" did not indicate the functionality that the participant was looking for, leading to an increased time usage. When using the swipe gestures for entering data the user quickly learnt how it worked and entered the correct data without looking at the screen.

The participant spent time discovering how to end the trip in task 9, leading to a significantly higher time usage than other participants. In task 10 the Rapport page was visited, but the user became confused about whether it showed the requested information or not, increasing time usage. After further exploration of the app the user discovered the MapAreaDetails page containing the correct information. The awareness of the MapAreaDetails page let the user perform task 11 quickly, almost a minute faster than task 10, even though the information is found at the same page for both tasks.

Participant 4

The fourth participant had little knowledge about the domain but much experience with the Android platform. The participant actively tried to understand the application by exploring the user interface and different pages. The time usage show that the user quickly discovered how to solve the tasks correctly.

While performing task 1 the user did not immediately find the correct page for downloading an area of the map, but after some exploration it was found. Some additional time was spent figuring out how to create an observation by long-pressing the map.

In contrast to other participants, the swiper page was quickly found in task 4, and after multiple test swipes the user understood the interaction method, and was able to enter the correct data without looking at the screen. This let the user look in the binoculars uninterrupted while entering animal counts for the herd.

For task 10 and 11 the user found the requested information in the MapAreaDetails page as desired, resulting in a low time usage for these tasks. This indicates that the user was able to find the page displaying the correct information after some exploration of the application.

Findings - Mobile Prototype

It is apparent that a new mobile application requires the user to explore and try out the application before feeling comfortable and reaching high efficiency. The results show that most participants spend time figuring out how to create a new observation by long-pressing a location on the map. In Figure 6.2 this is reflected in the high time usage for task 3. This is only a challenge for the first observation, and for following observations it does not seem to slow down the user.

Participants did not like the word "Swiper" as it was not descriptive enough regarding what functionality it provided, leading to confusion and additional time spent performing task 4. This is the main reason for the big variation between min and max time usage of task 4 as seen in Figure 6.2. It was proposed that the "Swiper" should be changed to indicate its purpose clearly, such as "herd registration using binoculars". The swiper page itself with the swipe gesture input seems to be quick and easy to learn. All participants were able to enter the correct data without looking at the screen.

While some users found the MapAreaDetails page quickly, others either did not find it or spent extra time exploring the application before finding it. This indicates some confusion. It might not be a huge problem however, as users are likely to find the correct page with some exploration of the app. It does still pose the question of whether the navigation structure between pages should be reconsidered to avoid this confusion.

An important takeaway is that even though some users spent time exploring the user interface, they always figured out how to complete the tasks in a reasonable amount of time. After exploration in one task the results show that users often were able to perform other tasks quicker due to a new awareness of functionality and pages in the app. As the participants performed most tasks correctly it shows that the prototype supports users in performing the defined use cases. It is also apparent that experienced users are able to use the application very efficiently, using less than a minute per task, as achieved by participant 2.

6.4 Results of Interviews

In addition to the usability testing, each participant testing the application prototype was interviewed to determine their subjective satisfaction with the prototype. The interview also let participants give general feedback on the application, such as the UI and functionality. The full interview answers are shown in Appendix C while the main findings are presented in this section.

Interview Questions:

1. Was the system easy to use?
2. Do you feel that you managed to perform all of the tasks?
3. What parts of the application was easy to understand and use?
4. What parts of the application was difficult to understand and use?
5. Did you find the system to be too complex?
6. Is there anything you think should be different with the application?
7. Was something missing in the application?

6.4.1 Findings - Interview

All four participants answered that the prototype was easy to use, but some clarified that they needed time to explore the application at the start. Additionally all users felt they managed to perform the tasks correctly, even though some of them felt uncertain at times. Generally participants state that this uncertainty led to spending extra time on the task at hand, such as finding the correct page for solving a specific task.

The interview confirmed that users found the word "Swiper" to not indicate the functionality of entering herd observation data while using binoculars. It was also stated that long-pressing the map could be more obvious for creating a new observation as it was not immediately clear how to create a new observation. All users managed to figure this out, but making the UI clearer could improve the speed and satisfaction the first time the user creates an observation.

One of the participants had difficulties finding the button for ending a trip. The person stated it was hard to find as it was located in a separate menu and not on the main screen. Regarding the structure of navigation it was proposed that the trips for a MapArea should be reached from the MapAreaDetails page rather than the StarMenu page to provide better navigational structure in the app.

Even though users discovered things that can be improved, none of the participants felt that the system was too complex. All users managed to perform the tasks successfully and were satisfied with the system. The problems discovered and changes proposed should be considered implemented in a new version of the application.

Chapter 7

Discussion

This chapter discuss the results of the usability tests and interviews in regard to the application and research goal. Based on this the research questions are answered. Afterwards some limitations and ideas for future work are laid out.

7.1 Discussion

The purpose of the mobile application was to provide a useful and easy system for the farmer to store observations of animals, predators and the environment. Comparing Figure 6.1 with Figure 6.2 we see that all tasks can be performed faster with the application than with pen and paper. This is likely because the app is giving the user a lower workload. The app automates the storage of GPS location and integrates image capturing functionality. As apparent in the results, a large portion of time is spent finding the location on the map manually when not using the app. Furthermore the app is able to store a more accurate location of each observation compared to the manual approach of looking it up on a map.

As the app integrates the image capture capability and image storage into the app the workload is lowered. The user does not need to carry a separate camera or log the time of capture for each image taken, to later match it with the correct observation. Another aspect is that logging a trail can be tedious when done manually. For this reason it is unlikely that a farmer will do more than noting down a couple of locations for the trip trail when having to use pen and paper. The application automatically logs the entire trail of the trip. This frees up capacity for the farmer to focus on the content of each observation while still acquiring detailed trail data.

One important and novel feature of the application was the support for registering animal counts of a herd without needing to look at the device. The test results clearly show that the "Swiper" title did not indicate this functionality, as it lead to confusion for some participants. Some users spent a notably long time exploring the user interface, looking at the "Swiper" menu item, and considering what it meant, often looking for other things before returning and trying to click it. A potential change could be calling it "Observe with Binoculars" as this more clearly states the functionality. After finding the swiper page participants managed to understand the interaction method and enter the correct animal counts efficiently. The verbal feedback of count type and count value seem sufficient to not require the user to look down at the device to confirm the effects of swipe actions. Most of the time usage for this task consisted of finding the swiper page and learning the interaction method. This means that for subsequent registrations users would likely perform the task substantially faster.

In the process of entering and storing observation data, the application seem to be both time saving and require less work. It also gives more accurate location data, in comparison to using pen and paper. Furthermore the data is structured and stored digitally which enable quick aggregation of key data for usage in government rapports. This storage benefit may also prove useful with regards to more advanced analysis of observation data. It is worth noting that all participants were new to the app without prior knowledge, and that time usage for common tasks is likely to drop substantially after having tried and explored the application. This can be seen in the difference in time usage between task 3 and 6, which are very similar to perform. Figure 6.2 shows that the average time usage drops by 20 seconds from task 3 to task 6. This is likely because participants have understood the user interface and know how the task should be performed after recently having performed a similar task.

Another aspect of the application is the visualisation of observations and trips on a map. A farmer using pen and paper could mark the location of different observations on a paper map. The problem with this approach is that it would be tedious and complex as the number of trips grow, with the farmer keeping one paper map for each trip. Having all data stored in the app lets the user get a quick glance of the location of different observations. Furthermore this presentation shows the type for each observation, and enables filtering observations per trip and per area. The results of task 12 indicate that users are able to quickly identify clusters of markers on the map, such as dead animals. This is beneficial for farmers as they can spot areas where herds spend much time or areas with a high amount of injury, death or predator activity.

When asked to find the number of trips performed and dead animals spotted in an area, most participants found the MapAreaDetails page which shows details about MapAreas. One participant opened the ConductedTrips page showing a list of all trips and proceeded by manually aggregating the desired info. Another participant initially looked at the Rapport page. This confusion led to higher time usage, as seen for task 10 and 11 in Figure 6.2. It is likely that users would find the details page displaying this information with sufficient exploration and usage of the app. However this shows that some users do not fully understand all aspects of the app initially. One option would be to add a button on the MapAreaDetails page that navigates to a page containing a list of trips performed in that area only. This would enable the removal of the Conducted Trips button at the start menu, reducing the likelihood of users manually going through all trips, instead of finding the information at the MapAreaDetails page.

The results of task 13 and 14 show that participants easily and quickly are able to display aggregated data for an entire season by visiting the Rapport page. If users were to perform the same task with the pen and paper approach it would likely require a lot of time and effort as well as being error-prone. The user would have to manually go through all paper notes of all previous trips. The mobile application thus yields a large reduction in time usage when the farmers need to write a rapport for the government at the end of the season.

The usability testing and interviews revealed aspects of the UI that need improvements, showing areas where users spend more time than desired due to confusion. The findings show these to be minor problems that are easily changed and that are primarily a challenge the first time a user interacts with the application. The testing and user feedback indicate that the app supports the outlined use cases and that users are able to perform tasks correctly with good efficiency and satisfaction.

7.1.1 Research Questions

Based on the test results and discussion, the research questions are answered.

RQ1.1: What efficiency and accuracy does the swiping gesture provide for entering herd animal counts while not looking at the phone screen?

The swiper input method was used to support the need for entering herd count data without simultaneously looking at the device. Users had to learn this new input method where the different direction of swipes have specific effects. The results indicate that users understood and mastered this custom interaction method. Furthermore it seems to be a reliable and beneficial input method for farmers allowing them to keep the herd in sight at all times using binoculars. The combination of swiping as an interaction method and audio feedback to verify the action and effect of the action enables users to observe a herd without the need to look at their phone. This lets the user be fully focused on looking at an animal herd with binoculars, not having to break visual contact with it. In addition to being both effective and efficient, user feedback indicate a high satisfaction with the implemented swiper functionality.

RQ1.2: How does a mobile application compare to pen and paper for registering observations?

The mobile application appears to be more efficient than the pen and paper approach, as time usage for tasks in the usability tests are lower, especially after users have spent some time getting to know the UI and functionality of the application. Additionally it simplifies the process by automatically storing GPS trails, GPS position of observations and integrating image capturing into the application. The results indicate that unifying all required functionality and automating aspects of the application is well received by users as it lowers the work load. As all data is stored digitally it also simplifies the writing of a government rapport as key data can be extracted rather than having to manually go through text notes for the trips and observations of an entire season.

RQ1: What value and usability does a mobile application provide for a sheep farmer when he is performing supervision trips?

With its simple yet effective input methods, the app manages to provide high usability for farmers and shepherds going for supervision trips. It supports all of the observation types identified in the pre-study and even surpasses the pen and paper approach in both functionality and efficiency. This is among other things due to simplification, automation and integration of useful functionality. User feedback show satisfaction with the design and functionality provided, even though some issues were pinpointed. The results seem to imply that a mobile application is a viable option for farmers to use on supervision trips, supporting the storage of desired information regarding trips and observations. It also greatly simplifies the usage of this information as it is stored digitally.

7.2 Contributions

The application prototype seems to fulfill the research goal of simplifying and streamlining the follow-up and data registration process for sheep farmers. Based on the results and evaluation in this thesis, the viability of using a mobile application for registering observations on supervision trips is considered high. Furthermore the user interface and interaction methods used in the prototype enable effective and efficient data input by the user. Specifically swipe gestures with audio feedback have the appearance of being a simple yet accurate and effective way to support data input without requiring the user to look at the display. The data entered this way is only numeric, so it is limited, but works as desired for storing animal counts.

The findings indicate that a mobile application would provide a good experience for farmers, supporting the required use cases and both simplifying and reducing the work needed. This is seemingly a substantial benefit in itself, but another important aspect is what this approach enables in terms of easily sharing data with other farmers or simple aggregation and analysis. Having a digital tool for data input and data storage paves the way for exploiting the data in new ways. While these questions are not explicitly dealt with in this thesis, the prototype and findings could be a starting point for such future work.

7.3 Limitations and Future Work

One big limitation of this study is the fact that the prototype was not tested on sheep farmers, due to a lack of availability. This means that the utility of the prototype for the target user has not been thoroughly tested. However the app is tested for usability on multiple participants acting as shepherds. A future research effort could perform usability tests on farmers and interview them to get feedback on the prototype regarding design, usability and functionality. A crucial step would be to verify that the application has the functionality that sheep farmers require during supervision trips. One specific aspect to investigate is whether the types of observations fit the needs of the farmer, and if the data fields in the different observation types should be altered.

The visual presentation of trips and observations on maps would also be subject to feedback. It should be tested whether farmers find value in inspecting a map with markers of previous observations in a MapArea in the planning of a new supervision trip in that area. Additionally it should be verified that the generated rapport provides valuable data. Feedback should be acquired on what parts might need change. Furthermore while participants in the usability testing performed many common tasks on the system, the app was not tested over a full season. Performing full season tests with multiple real sheep farmers could reveal more intricate problems with the prototype and reveal new requirements.

As the app focuses on the data input aspects it would be interesting to investigate the creation of a desktop application for managing and inspecting the collected data. While the current application supports the exportation of all collected data, it does not look into how the data can be utilized beyond rapport generation and visualization of trips and observations on a map. Performing statistical analysis on the data might reveal intricate patterns and information useful for the farmer. Furthermore if many farmers submit their data to a separate entity, it might be possible to analyze the data of multiple farmers collectively.

The app was created for offline usage and only designed to support a single farmer in observing his own animals. It is likely that farmers could benefit from seeing observations done by other farmers as well as being able to register herds or animals belonging to other farmers. Adding support for multiple users where users can share their observations and trips with a group of other farmers would be interesting to look into. This might reduce the collective effort of going for supervision trips as other farmers can already have made recent observations of your herds. This would require a separate server with new data models for users and teams of farmers. Such work could build on both the findings and the implemented prototype of this thesis.

Usability testing was performed on a group of participants only once, and performing multiple iterations where the application would be changed based on the feedback of the tests could provide new and more elaborate results. It might also bring the application closer to a releasable product rather than a prototype.

As the usability tests were primarily performed in good weather conditions it is not possible to determine the usability in harsh weather that the farmer might encounter. It is however natural to expect that as the mobile application gives a lower workload compared to pen and paper it will likely still be better off. Having a mobile device that is waterproof and can handle low temperatures might be a benefit. To determine the usability in other weather conditions it would be interesting to perform usability testing in less ideal weather conditions. This could show how a mobile application performs in a more varied and realistic environment.

As the manual supervision trip is only one way to gather information about animals, it would be possible to integrate other systems into the application. One such system could be GPS collars worn by animals, that send location information to a server. The app could fetch this data from the server and present it to the user on a map. This could help the farmer in determining where he might go look for his animals when planning a supervision trip.

Chapter 8

Conclusion

The results of the usability testing and interviews show that the application prototype outperformed pen and paper for storing trip and observation data during supervision trips. The prototype provided better time usage, data accuracy, functionality and user satisfaction. Participants managed to correctly and efficiently perform the most common tasks on the system, both regarding data input and data retrieval. Some challenges were discovered, but they appear manageable for the user and could be fixed in a future version of the application. This indicates that using a mobile application on a smartphone is a viable option for storing observations of animals in field. The user interface and input elements of the prototype made up a simple yet effective way for farmers to register data.

Multiple technical capabilities of the smartphone were able to be utilized. Using the GPS of the phone enabled automatic trail logging of trips and location storage of observations. As smartphones have cameras, the application could integrate image capturing so that images can be stored in association with observations. By creating an app that unifies different tasks the farmer performs on a trip, the app was able to reduce time usage and workload of the farmer. Visualizations of trip trails and observation markers on maps enable quick evaluations by the farmer and provides an overview of the collected data. Having farmers test the prototype over a longer time period and continuing development based on their feedback could likely result in an application that could be distributed to farmers for real world usage. Supervision trips still provide valuable information for the farmer which systems such as UAV's and digital collars are unable to provide. An application such as the one developed for this thesis could simplify supervision trips for the farmers as well as the usage of the gathered data. The results indicate it can reduce time usage when storing observations, enable the entering of observation data while simultaneously looking in binoculars and provide digital and structured data for both government rapports and independent analysis.

Bibliography

[1] ISO 9241-11:2018(en), 'Ergonomics of human-system interaction — Part 11: Usability: Definitions and concepts,' International Organization for Standardization, Geneva, CH, Standard, 2018.

[2] Mattilsynet. (2019). 'Årsrapport 2019,' [Online]. Available: https://www.mattilsynet.no/om_mattilsynet/aarsrapport_2019__mattilsynet.38708. (accessed: 01.09.2020).

[3] Lovdata. (2005). 'Forskrift om velferd for småfe,' [Online]. Available: https://lovdata.no/forskrift/2005-02-18-160/%C2%A719. (accessed: 01.09.2020).

[4] B. J. Oates, *Researching Information Systems and Computing*. SAGE Publications Inc, 2005, ISBN: 9781412902243.

[5] NSG. (2011). 'Bjelleslips - kodemerking for lammetall på beite,' [Online]. Available: https://www.nsg.no/a-a/merking-av-smafe/bjelleslips/. (accessed: 07.11.2020).

[6] Miljødirektoratet. (2020). 'Tap til rovdyr holder seg på lavere nivå enn før,' [Online]. Available: https://www.miljodirektoratet.no/aktuelt/nyheter/2020/januar-2020/tap-til-rovdyr-holder-seg-pa-lavere-niva-enn-for/. (accessed: 15.01.2021).

[7] J. Barbedo and L. Koenigkan, 'Perspectives on the use of unmanned aerial systems to monitor cattle,' *Outlook on agriculture*, vol. 47, pp. 214–222, Jun. 2018. DOI: 10.1177/0030727018781876.

[8] D. W. Bailey, M. G. Trotter, C. W. Knight and M. G. Thomas, 'Use of GPS tracking collars and accelerometers for rangeland livestock production research1,' *Translational Animal Science*, vol. 2, no. 1, pp. 81–88, Jan. 2018, ISSN: 2573-2102. DOI: 10.1093/tas/txx006.

[9] Google. (2020). 'Google maps,' [Online]. Available: https://play.google.com/store/apps/details?id=com.google.android.apps.maps. (accessed: 19.11.2020).

[10] ATLOGIS Geoinformatics GmbH & Co. KG. (2020). 'Norway topo maps,' [Online]. Available: https://play.google.com/store/apps/details?id=com.atlogis.norway. (accessed: 20.11.2020).

[11] Telespor AS. (2020). 'Telespor,' [Online]. Available: https://play.google.com/store/apps/details?id=org.telespor.telesporapp. (accessed: 24.11.2020).

[12] Fant AS. (2020). 'Beitesnap,' [Online]. Available: https://www.beitesnap.no/. (accessed: 24.11.2020).

[13] H. Kniberg, *Scrum and Xp from the Trenches - 2nd Edition*. Lulu.com, 2015, ISBN: 9781329224278.

[14] Kanbanize. (2020). 'What is kanban board,' [Online]. Available: https://kanbanize.com/kanban-resources/getting-started/what-is-kanban-board. (accessed: 03.10.2020).

[15] Google. (2020). 'Google Maps Android SDK,' [Online]. Available: https://developers.google.com/maps/documentation/android-sdk/overview. (accessed: 08.09.2020).

[16] MapBox. (2020). 'MapBox,' [Online]. Available: https://docs.mapbox.com/android/beta/maps/guides/. (accessed: 08.09.2020).

[17] (2019). 'OSMdroid,' [Online]. Available: https://github.com/osmdroid/osmdroid. (accessed: 08.09.2020).

[18] Kartverket. (2020). 'Kartgrunnlag Fastlands-Noreg,' [Online]. Available: https://www.kartverket.no/api-og-data/kartgrunnlag-fastlands-norge. (accessed: 17.09.2020).

Appendix A

Usability Test and Interview

This appendix shows the usability test created and used for testing the prototype as well as the questions for the interview.

Usability Test

System: Sheep Registration Mobile Application – functional prototype
Purpose: test the *usability* of the application by having participants perform typical use cases.
Measures:

Are the tasks performed successfully?
How much time is used to perform each task?
How satisfied is the user with the experience?

Results usage:

Determine if the application fulfills its purpose for the targeted user group (sheep farmers and shepherds).
Determine aspects, interfaces, components, etc. of the application that needs change.
Determine what value the application provides compared to paper-based note taking approach.

Scenario & Tasks

General situation:
As a farmer you are to perform a trip to follow up on the well-being of your sheep and lamb. The animals have been grazing for a week and you know where you left your two herds. These two locations will be the areas you initially plan to visit on this trip.

Task 1 – Download a map area:
You are to perform the first supervision trip this season. To perform the trip, you need to download the map area you are to perform the trip in.

- Download the area where you will perform the trip. Call it "West Hills".

Task 2 – Start a trip:
Having downloaded the map area, you are ready to start your supervision trip.

- Start a new trip in the area "West Hills". Call it "First supervision trip".

Task 3 – Register an observation of a dead animal with an image:
You walk to your first pasture and find a lamb lying on the ground covered in blood. It is dead.

- Create an observation of the dead lamb. Include the ear tag id and an image.

Task 4 – Register an observation of a herd:
You walk further and see a herd in the distance and must use binoculars to see it better.

- Create a new observation of the herd.
- Register 2 sheep, 3 lamb, 4 white, 1 grey, and 2 yellow ties without looking at the screen.
- Verify that the herd has the correct number of lambs w.r.t. tie colors, otherwise make a note about the inconsistency.

Task 5 – Register an observation of a herd:
Walking further you come across your second herd. It is close by, so you will *<u>not</u>* need to use binoculars.

- Create a new observation of the herd.
- Register 2 sheep, 5 lamb, 3 black, 2 white, 2 grey, 1 yellow tie and 1 green tie normally.

- Verify that the herd has the correct number of lambs wrt. tie colors, make a note otherwise.

Task 6 – Register an observation of an injured lamb

You look closer at one of the lambs and see that it has a cut on one of its feet.

- Create a new observation of the injured lamb. Include the ear tag id. Note that the rear left foot has a cut.

Task 7 – Register a predator

As you walk towards home, you suddenly spot a wolf.

- Create a new observation of the predator. Note that it is a wolf. Take an image of it.

Task 8 – Register a bad pasture

Further on your trip home you spot a pasture that is muddy.

- Create a new observation of the muddy pasture.

Task 9 – End trip

You arrive home safely.

- End the trip.

Task 10 – Find the number of trips performed in an area

You want to know how many trips you have performed in the area.

- Find the number of trips performed in the area "Wide Valley"

Task 11 – Find the number of dead animals spotted in an area

You want to know how many dead animals you have spotted in the area.

- Find the number of dead animals spotted in the area "Wide Valley"

Task 12 – Identify a location in the map "Wide Valley" where many dead animals have been spotted

You find it odd that so many animals have died in the area and want to see if some location points out.

- Identify a cluster of dead animals on the map for "Wide Valley".

Task 13 – Show a rapport from 2018

You want to look at how things went in 2018.

- Show a rapport from 2018.

Task 14 – Export a text rapport, images, and database file for 2019 to Google Disk

You need to export a rapport as well as images and the database file to yourself so that you can have the documentation on you PC.

- Export a text rapport, images, and database file for 2019 to Google Disk

Interview

1. Was the system easy to use?
2. Do you feel that you managed to perform all the tasks?
3. What parts of the application was easy to understand and use?
4. What parts of the application was difficult to understand and use?
5. Did you find the system to be too complex?
6. Is there anything you think should be different with the application?
7. Was something missing in the application?

Appendix B

Usability Testing Results

This appendix presents the test results of usability testing for pen and paper as well as the application prototype.

B.1 Pen and Paper Results

Task Number	Completed	Time Usage	Notes
3	Yes	2m 7s	About 50s was used to find the position (lat, lon) on the map.
4	Yes	2m 21s	The user was unable to write while using binoculars.
6	Yes	2m 3s	
7	Yes	2m 12s	
8	Yes	1m 51s	

Table B.1: Pen and Paper Usability Test Results - Participant 1

Task Number	Completed	Time Usage	Notes
3	Yes	3m 22s	About 85s used to find position on the map.
4	Yes	3m 45s	The user had to shift focus between writing and looking in binoculars regularly.
6	Yes	2m 20s	The user mentions the potentially large effort required to find an image for a specific observation later on.
7	Yes	1m 49s	
8	Yes	1m 45s	

Table B.2: Pen and Paper Usability Test Results - Participant 2

B.2 Application Prototype Results

Task Number	Completed	Time Usage	Notes
1	Yes	22s	
2	Yes	17s	
3	Yes	1m 40s	The user is uncertain of how an observation is added, and explores the UI until trying a long press on the map and succeeds.
4	Yes	2m 53s	The user has some trouble finding the swiper functionality. The user also swipes the wrong way a couple of times. Completed successfully without looking at phone screen.
5	Yes	31s	
6	Yes	48s	Task is similar to task 3, but completed much faster now that the user knows about long-pressing the map.
7	Yes	19s	
8	Yes	17s	
9	Yes	12s	User finds the functionality quickly.
10	Yes	9s	The user counted elements in the list of performed trips.
11	Yes	1m 23s	The user looked at the details for each trip in the area and manually summed the count of dead animals.
12	Yes	23s	The user is quickly able to identify a cluster of dead animal markers on the map.
13	Yes	16s	The user spends some time locating the year filter button.
14	Yes	12s	The user exports the correct data.

Table B.3: Application Usability Test Results - Participant 1

Task Number	Completed	Time Usage	Notes
1	Yes	21s	
2	Yes	12s	The user immediately finds the button at the bottom of the start screen.
3	Yes	26s	The user quickly tries long press on the map and is able to create a new observation.
4	Yes	44s	The user finds the swiper screen quickly from the top right menu, and understands the swipe gestures after a couple of test swipes.
5	Yes	36s	The user enters animal counts quicker than with swiping.
6	Yes	30s	
7	Yes	14s	
8	Yes	24s	
9	Yes	7s	
10	Yes	12s	The user found the information at the MapAreaDetails page.
11	Yes	11s	The user found the information at the MapAreaDetails page.
12	Yes	17s	
13	Yes	9s	
14	Yes	18s	

Table B.4: Application Usability Test Results - Participant 2

Task Number	Completed	Time Usage	Notes
1	Yes	37s	
2	Yes	25s	
3	Yes	1m 16s	The user stated uncertainty about the effects of the "Add" button, and would prefer it being labeled "Save" as it is clearer. The button for adding an image took some time to find as it was not adjacent to the "Images" label.
4	Yes	2m 37s	The user is uncertain about what "Swiper" means, and proposes that it should be changed. The swiping functionality is quickly learnt and the user is able to store all animal counts.
5	Yes	57s	
6	Yes	44s	
7	Yes	38s	
8	Yes	25s	
9	Yes	39s	The user had some trouble finding the "End trip" menu item.
10	Yes	1m 24s	The user first looked for the information at the "Rapport" page. After exploring the app further the info was found in the MapAreaDetails page.
11	Yes	27s	
12	Yes	23s	The user identified a cluster of red markers indicating dead animals.
13	Yes	14s	The user quickly found the page as it was discovered in task 10.
14	Yes	26s	

Table B.5: Application Usability Test Results - Participant 3

Task Number	Completed	Time Usage	Notes
1	Yes	34s	The user had to dig around the UI a bit before finding the correct page for downloading a map area.
2	Yes	17s	
3	Yes	52s	The user spends some time figuring out that long-press creates a new observation.
4	Yes	1m 2s	The user performs many swipes to understand the interaction method, but quickly learns it and is able to input all desired data without looking at the device.
5	Yes	47s	The user mentions that grouping the list of animals types would make the UI clearer.
6	Yes	53s	
7	Yes	39s	
8	Yes	34s	
9	Yes	9s	
10	Yes	16s	The user finds the information at the MapAreaDetails page.
11	Yes	13s	The user finds the information at the MapAreaDetails page.
12	Yes	21s	
13	Yes	27s	
14	Yes	25s	

Table B.6: Application Usability Test Results - Participant 4

Appendix C

Interview Results

Participant 1

1. Yes, after some time exploring the app. It took some time getting used to the swiping.
2. Yes.
3. The easy observations was simple to enter.
4. It took some time to figure out how I could add an observation. I also was uncertain of what "Swiper" was before I clicked it, so this name could be improved.
5. No.
6. The "Swiper" button could be simpler to find. The swiper screen should also have a "save" button as I became uncertain if the changes would be stored.
7. No.

Participant 2

1. Yes, I quickly managed to find what I was looking for.
2. Yes.
3. It was easy to understand the start screen and how I could reach different portions of the application.
4. Long-pressing the map to create a new observation was not my initial idea, but worked fine once I discovered it.
5. No, things made sense.
6. From the map area details page it would be useful to reach all trips performed in that area.
7. Maybe smartwatch integration could be added, so that I can see status on my watch while the phone is in my pocket.

Participant 3

1. It was pretty easy to use, although I had to learn a few things, like how to register an observation (long-press) and the registering without looking. I think that once you have learned those things, it's easy to use.

2. Yes. Or, at least after the settings on the phone were set correctly. I actually thought that one of the most challenging things was to end the trip.
3. Creating a new trip was easy, and to register the sheep and lambs. Also, to choose which type of observation was pretty straight-forward.
4. Ending a trip (hard to find it), creating a map (didn't first understand that you had to pre-save a map to be able to go on a trip). But nice that it tells you to create a map when you try to start a trip without it.
5. Not really. Did not have many unnecessary features, but I'm not a sheep farmer, so I can't be too certain about that.
6.
   - Easier to end a trip.
   - Automatic closing of the keyboard during observation.
   - Shorter time to download a map, although I don't know if that's possible.
7. Maybe a prompt asking if you wanted to pause or end the trip when you clicked the back-arrow during a trip? My trip ended up lasting for 10+ days since I didn't know that I had not ended it.

Participant 4

1. Yes, after some usage it felt natural.
2. Yes.
3. I felt that details about map areas was easy to find.
4. It took some time to figure out that long-pressing the screen created a new observation, but I liked the way of doing it once I discovered it.
5. No, but some things took a little time to figure out how to do, but I feel that is the case anyway.
6. I would like the list of animal types for a herd observation to be grouped. I would like sheep and lamb in one group and the colors in another and the tie colors in a third group.
7. Maybe a message saying that long-pressing creates a new observation when using the app for the first time.

Styve, Svein Olav

Mobile App for Data Registration of Grazing Sheep

NTNU

Kunnskap for en bedre verden