Wearable Weather Forecast

Creating a forecast through iOS and Smart Clothes

EMIL JUZOVITSKI
Abstract

Smart Clothes refers to clothes that have been embedded with computer devices. Items of the category can be used to inform an individual about his health. The technology can likewise be used to track an individuals surroundings.

There are many possible uses for Smart Clothes. This thesis is based on the idea of creating a Smart Clothes item that can create weather forecasts independently. In order to create such system unique problems have to be solved. One major problem to overcome is realizing what a mobile application a mobile application has to do in a Smart Clothes system that predicts weather to the user. This thesis tries to answer the problem by examining how an existing weather forecast model can be implemented into an application to later display a forecast to a user. The study is built on an inductive approach and has a purpose to explore the necessary steps in developing a mobile application that predicts the weather.

Presenting an implemented prototype is the purpose of the thesis. As such, the purpose becomes to create a mobile application that implements a weather forecast model and translates the created weather forecast into a UI suited for a mobile application. The weather forecast model should use atmospheric pressure differences to create a weather forecast. The goal of the thesis is to find out how the purpose can be achieved, how a mobile application should behave in order to assure that the Smart Clothes system can create a weather forecast for the user. A partial goal is to understand how atmospheric pressure should be converted into a weather forecast and how the weather forecast should be translated into a UI.

The result is presented in the form of a prototype, an iOS application. The created prototype uses atmospheric pressure differences to create a weather forecast. The weather forecast is shown to the user through a intuitive user interface. The prototype is tested through a user-test and is evaluated through a thorough model with criteria.

The evaluation shows that the prototype is easy to use and understand. There are in the same time improvements to be done for the future. Proposals have been given for improvements. The user-test has shown that the interest in the prototype is higher in situations where the user lacks internet connectivity. There is thus, a link between the prototype's usability and internet connectivity. In other words, people find the prototype to be of a big benefit as it can be used without Internet connectivity.

Keywords: iOS, Weather Prediction, User Interface, NotificationCenter, Prototype Creation, Development of Smart Clothes
Abstrakt

"Smart Clothes" refererar till klädesplagg som har integrerats med dataenheter. Plagg i kategorin kan användas för att informera individen om hans hälsa. Teknologin kan likvä användas för att spåra individens omgivning.


Nyckelord: iOS, Väderprognos, Användargränssnitt, NotificationCenter, Prototypskapande, Utveckling av Smart Kläder
# Table of Contents

1. Introduction ........................................................................................................... 3  
1.1 Background ........................................................................................................... 3  
1.2 Problem and Research Question ......................................................................... 3  
1.3 Purpose ................................................................................................................ 4  
1.4 Goal ....................................................................................................................... 4  
1.5 Benefits ................................................................................................................ 4  
1.6 Ethical Aspects ...................................................................................................... 5  
1.7 Research Method .................................................................................................. 5  
1.8 Target Group ........................................................................................................ 5  
1.9 Delimitations ......................................................................................................... 5  
1.10 Outline ............................................................................................................... 6  

2. Research Methodology .......................................................................................... 7  
2.1 Research Strategy ................................................................................................ 7  
2.2 Research Method ................................................................................................ 7  
2.3 Research Phases .................................................................................................. 8  
2.4 Research Tools ..................................................................................................... 8  
  2.4.1 Literature Review ......................................................................................... 8  
  2.4.2 Interviews ...................................................................................................... 9  
2.5 Sampling ............................................................................................................... 9  
2.6 Validation ............................................................................................................ 9  

3. Theoretic Background ......................................................................................... 11  
3.1 Atmospheric pressure and Barometers ............................................................ 11  
  3.1.1 Weather Forecast Based On Atmospheric Pressure ................................. 11  
3.2 Visualizing Information on Mobile Applications To The User .................... 12  
  3.2.1 Usability ....................................................................................................... 12  
  3.2.3 Affordance .................................................................................................... 13  
3.3 Structure of an iOS Application ....................................................................... 13  
3.4 Related work ....................................................................................................... 15  
  3.4.1 Weather Forecasts ....................................................................................... 15  
  3.4.2 The Current Scene ....................................................................................... 16  

4. Work Structure ................................................................................................... 19  
4.1 Pre-Study ............................................................................................................ 19  
  4.1.1 Idea Generation and Feasibility Study .................................................... 19  
  4.1.2 Concept Creation ....................................................................................... 19  
  4.1.3 Choice of an Applicable Weather Forecast Model .................................. 21  
4.2 Software Development Process ....................................................................... 21  
  4.2.1 Iterations of Implementation Phases ......................................................... 21  
4.3 Evaluation of Prototype ................................................................................... 21  
  4.3.1 Evaluation Criteria ....................................................................................... 22  
  4.3.2 Exploration Criteria .................................................................................. 22  
4.4 Testing ................................................................................................................ 22  

5. Implementation: Design ..................................................................................... 25  
5.1 Atmospheric Pressure to Weather Forecast ...................................................... 25  
5.2 Updating the User Interface ................................................................................ 26  
5.3 User Interface Layout ......................................................................................... 26  
  5.3.1 User Interface Elements ............................................................................ 26  
  5.3.2 Gathering Measurement Sequence ......................................................... 26  
  5.3.3 New Forecast Sequence ........................................................................... 26  
5.4 User Interface Design Choices .......................................................................... 26  
  5.4.1 Colors, Font, Text, Icons and Buttons .................................................... 26
6. Implementation: Programming.................................................................27
   6.1 Design Pattern..................................................................................27
      6.1.1 Controller.................................................................................27
      6.1.2 View.........................................................................................27
      6.1.3 Model.......................................................................................27
   6.2 Broadcasting....................................................................................28
   6.3 Weather Forecast Algorithm.........................................................28
   6.4 newPredictionButton.......................................................................28
   6.5 updateUserDescription....................................................................28
   6.6 timeInfoModifier............................................................................28
7. Testing and User Test...........................................................................31
   7.1 Feature Testing.................................................................................31
   7.2 User-Test.........................................................................................31
8. Results...................................................................................................33
   8.1 Result of Feature Testing.................................................................33
   8.2 User Experience and Interface.......................................................33
   ...............................................................................................................34
   8.3 Results of Evaluation.......................................................................34
      8.3.1 Result of Evaluation Criteria....................................................34
      8.3.2 Result of Exploration Criteria.................................................34
9. Discussion..............................................................................................37
   9.1 Result Discussion.............................................................................37
   9.2 Method discussion...........................................................................37
   9.3 Validity............................................................................................37
   9.4 Discussion of Evaluation...............................................................38
   9.5 Functionality Discussion.................................................................39
10. Conclusion & Future Work.................................................................41
References...............................................................................................43
Appendix A..............................................................................................47
Appendix B..............................................................................................51
1. Introduction

As technology continues to advance, it also becomes increasingly more sophisticated. An expanding set of problems can be solved through the usage of technology. For instance, the development of smaller and more energy efficient computer devices has allowed for the introduction of “Smart Clothes”, that is embedding computer devices onto clothes [1]. Smart Clothes can be viewed as a sub-category of the larger category of “Wearable technology”, which includes products such as fitness trackers. The words “clothes” and “tech” might not seem as a natural combination. But, in fact, they can be used together to significantly increase the usability of one of our everyday items.

1.1 Background

Although there have been articles written about the potential of Smart Clothes, there have not been many products of the category that has reached market. Searching for the term “Smart Clothes” through Google’s search engine returns a “top” search result titled “Smart Clothes Are The Future Of Wearables”[2][3]. The article was published in 2016 and explains how Smart Clothes have the potential to be in the center of wearable technology. However, at the time of writing, there is no vivid traction for the product type.

Items of the Smart Clothes product category have common use-cases and properties. The use-cases and properties are important to note and are mentioned below:

- Smart clothes, currently available on the market, are often built to inform a user of any changes in health or physical activity, but can also be used to inform the user about external environmental or environmental conditions.

- Smart clothes require an underlying system of devices. Mobile phones and consequently mobile applications have been used as a calculation source for converting measurements to valuable information for the user. The mobile application is then also used to display the created information to the user [4].

To show how usability could be increased for a clothing company a traditional clothing item- the raincoat had to be reintroduced with a Smart Clothes system that would increase the general usability of the item. People bring raincoats when they expect it to rain at some point of the, however they might not always know when to expect rain. Will it rain in 20 minutes or 3 hours? To tackle this problem, it was decided to create a Smart Clothes system for the raincoat that produces a local weather forecast that is generated locally, dependent on your current location. The system were to be based on atmospheric pressure and barometers since differences in atmospheric pressure can be used to create a basic weather forecast [5]. Integrating barometers into a clothing item would allow barometers to track atmospheric pressure changes and in turn, the allow for a local weather forecast.

The underlying system of the proposed smart clothing item was to use three devices: barometers (sensors), a mobile device, and, a database. Attached barometers on a clothing item were to track the current atmospheric pressure. Atmospheric pressure data is then sent to a mobile application. The data was then to be also sent through the mobile application to a database. Finally, data given from both the sensors and the database was to be used in the creation of a weather forecast within the mobile application.

1.2 Problem and Research Question

As of now, there is no mobile application that can support a Smart Clothes system that informs a user of a local weather forecast through a mobile phone. Inadvertently, it is not clear how exactly measurements should be converted into a weather forecast. It is also not clear how the mobile application should display the weather forecast in a user-friendly manner.
Therefore the resulting research question becomes:

How can a weather forecast model be implemented and displayed on a mobile application, so that the user intuitively understands the resulting weather forecast?

1.3 Purpose

The purpose of this thesis is to present an implementation of a prototype mobile application that has the ability to create a weather forecast based on atmospheric pressure measurements and to display it in a way that is suitable for a mobile interface.

The prototype implements a model that converts atmospheric pressure differences into a weather forecast. In addition, a user interface is created to display the useful information to the consumer.

1.4 Goal

The goal of the project is to find out how a weather forecast could be modeled and implemented into a smart clothing system, with the soul focus being on the mobile application. A weather forecast model that converts atmospheric pressure differences into weather descriptions has to be implemented to allow people to predict weather through the application. The weather descriptions should be translated into symbols suited to a mobile interface to be able to display the weather forecast to a user.

1.5 Benefits

The created prototype would create a local weather forecast for a user. The user would not be dependent on anyone else to create the forecast. The only thing he needs is the clothing item he is wearing. The local measurements could be used together with weather stations to create a more precise and dense weather prediction map that is useful for both people in less dense areas and researchers.

One benefit of local measurements is that a forecast from the mobile application does not require Internet connectivity. The created system is not reliant on Internet to give the user a forecast. People traveling without Internet could therefore be great beneficiaries of the Smart clothes item.

Weather stations create measurements at a static location. They track a large area. The weather forecast becomes less accurate at areas further away from weather stations. People further away from the weather station could therefore use a forecast that is created closer to their location. A smart clothing system isn’t bounded to a static location. It is instead mobile. Mobility would allow measurements to be created wherever a user is located. These measurements can be weighed against a weather station's measurement to create a better weather forecast for everyone.

A user can create their own weather forecast without the internet. The created system is not reliant on internet to give the user a forecast. People traveling around without internet could therefore use the created smart clothing system.

Improving weather predictions allows people to plan for the weather. Changes in atmospheric pressure can affect a person’s health. Pressure drop can lead to headache or another type of pain, as well as affect blood pressure. [6]. The created mobile application could therefore be used to warn people about possible changes in health before they happen.

People can use the information for planning a sustainable lifestyle. One can expect more traffic due to rain. A person can drive ahead of the traffic caused by the weather if he or she can expect the rain. Farmers want to avoid using fertilizers when rain is due as the fertilizers are washed away by rain [7]. It is therefore important for farmers to get local predictions that allows avoids wasting fertilizers.
1.6 Ethical Aspects

Ethical concerns can arise when devolving a mobile application. A creator’s own preferences is often interwoven into a mobile application. Inadvertently, some design choices suit some people better than others. The color palette of an application could for example hinder people with color vision deficiency to read the layout properly. There is therefore an ethical motivation to test a created application on different people to make sure that the application can be of use to as many people as possible.

1.7 Research Method

A research method had to be chosen for this report. A methodology refers to the process behind the choice and use of a particular method. The research question usually guides the author to which research methodology to use [8]. Research methodologies refer generally, to one of two basic categories: quantitative or qualitative [9]. In quantitative research, data produced is numerical. Qualitative research is instead research that is difficult to represent through numerical data [10].

The research question of this thesis asks how a process of creating a prototype can be made. Since the research question of this thesis asks about the description of a process the methodology used should be of the qualitative form[11]. The methods of qualitative research generate comprehensive description of processes, mechanisms or settings [10]. In other words, methods answer what, why or how a particular outcome is achieved.

1.8 Target Group

A company named Tracy Trackers [12] are the stakeholders of this thesis. Tracy Trackers has a background in creating Artificial intelligence(AI) products. For this project Tracy Trackers has moved outside of the AI scene to see if they can help promote and conceptualize the idea of Smart Clothes to clothing companies. In doing so the target group becomes a targeted clothing company. The customers of a clothing company are people wearing their clothes. They are the ones who will use the Smart Clothes item and are therefore also the target group.

1.9 Delimitations

This thesis has restricted its range onto the mobile application, implementing a model of a weather forecast and displaying it through a user interface. There are no mentions of the construction of the grander project, which is the development, integration and communication with other devices of the system such as barometers and the database. To clarify it further, measuring atmospheric pressure through a barometer and sending it to the mobile application is not a part of this thesis. Furthermore, other topics of the grander project such as security aspects of storing data are also not be mentioned in the report.

The mobile application will be developed for iOS. The application will not be developed for any other platforms (e.g. Android). The thesis will only talk about the creation of the application through the perspective of iOS development.
1.10 Outline

The outline of the thesis is as following:

Chapter 2 describes The Research Strategy used in this thesis, clarifies the components of the research strategy: the research method, the chronological order of the phases, the tools and the sampling. And finally, the Chapter presents validity aspects.

Chapter 3 explains the theoretical background of the project. It mentions how weather forecasts can be made using only atmospheric pressure. Furthermore, it mentions the structure of an iOS application and a brief section about the topic usability. Finally the Chapter describes the software development process that are to be used.

Chapter 4 describes the work structure. It details the different stages of pre-study. It explains the choice of software development process. The planned iterations of the implementation phase is mentioned. Finally, the chapter introduces the evaluation and exploration criteria of the thesis.

Chapter 5 brings up the design that went on to be implemented into the iOS application. The phases of the designed algorithm exemplifies as a table. User Interface Design Choices explains. Chapter 6 describes how the design was implemented to become an iOS application. Each component of the implementation is defined. Chapter explains various methods used to allow for various functionality of the application. Chapter 7 shows how testing and a user test were made for the thesis. Chapter discusses testing functionality of the application.

Chapter 8 describes the result of the thesis, the created prototype and results of testing the prototype. Results of the two sets of evaluation criteria are shown. The initial evaluation criteria is compared with the feedback from the user-test. The results of the exploration criteria are shown as a comparison between the prototype and the criteria created at the concept creation stage.

Chapter 9 discusses the different aspects of the thesis, for example how the result answers to the research question, aspects of research strategy and validity aspects. Chapter 10 holds the thesis conclusion and Chapter 11 mentions the proposed future work by the author.
2. Research Methodology

This chapter describes the methods used in the thesis. Section 2.1 introduces the research strategy of the thesis. Continuing, each component of the research strategy is introduced further at the following sections of the chapter. Section 2.2 explains the choice of research method. Section 2.3 overviews the research phases (The specifics of each research phase is found in chapter 4.). In the next section; 2.4 the tools of the research is described and motivated. Section 2.5 explains the sampling done. Finally, Section 2.6 ends the chapter by explaining how validation is assured in the work.

2.1 Research Strategy

To achieve the purpose of the thesis, a research strategy must be applied. The nature of a research question usually leads to the choice of research a methodology. [8]. This thesis aims at developing a proposal for how a mobile application can be implemented. The thesis can therefore be based on a hypothetical research effort that explores the development of a mobile application. To create a proposal a research strategy, and thereby also research tools are necessary[11].

The research strategy used in this thesis is shown in Figure 1. It includes the following components: (1) Design of Research Phases, (2) Choice Of Research Methods, (3) Method for selecting respondents, (4) Construction of Research Integuments and (5) Management of Validity Threats.

![Research Paradigm Diagram]

Figure 1: Research strategy

2.2 Research Method

This work aims at solving the following research question: How can a weather forecast model be implemented and displayed on a mobile application, so that the users intuitively understand the resulting weather forecast? The focus is therefore on understanding how the current application may look like, consequently a concept or an idea must be created. Thus an inductive approach is needed.

An easy way to explain why an inductive research target is needed is to compare deductive and inductive reasoning. Generally speaking, the difference between deductive and inductive research can be explained by what the research begins with. Inductive research begins with a research question and the gathering of empirical data, used to generate hypothesis and theory, while deductive thinking usually begins with a theory-driven hypothesis that governs data collection and analysis [13].
Research effort is inductive, starting with collecting data relevant to the subject. Finding models for converting atmospheric pressure to a weather forecast is deductive, in other words, an existing solution, a model, is used. It can therefore be found that the method of the report is qualitative with deductive distinctive features in terms of the use of given models [11]. Furthermore, the focus is on understanding how the current application can look so that the user appreciates it. However, to get an idea of the user's perceptions, the application must be tested. Thus we need to gather empirical and interpret the user's perceptions [11]. In other words, an inductive qualitative approach is needed.

Qualitative research revolves around identifying ideas and trends. The research delimits itself onto giving a flow of concepts that the researcher analyses to get an understanding for the product-development. It is not a measurable research, but focus is on nuances. The process design of the prototype creation is flexible and evolving rather than being structured or predetermined. The design characteristics are suited for a qualitative approach is needed[14].

### 2.3 Research Phases

An overview of the research process is found in Figure 2. Shown in Figure 2, are the four main phases and their consequent sub-phases. The main phases as shown are the following: (1) Pre-Study, (2) Planning Implementation and Validation, (3) Prototype implementation and (4) Prototype Validation and Evaluation.

The Pre-study includes all phases (except planning) that were required to be accomplished before starting to develop the mobile application. Planning Implementation and Validation includes all the planning that was necessary before starting the development of the mobile application. Prototype implementation is the development stage. Each phase is explained more in-depth from chapter 4. and forth.

![Figure 2: Research Phases of the thesis](image)

### 2.4 Research Tools

Through this section the used research tools are explained for this thesis (See Figure 1). First, the tool is explained then, the reason behind using the tool is explained. The tools mentioned are (1) Literature Review and (2) Interviews.

#### 2.4.1 Literature Review

A literature review can simply be seen as a summary of sources with one exception. It usually involves re-organizing or reshuffling important information of the sources. The review can allow for new interpretation of old material [15]. By gathering information a greater understanding is found about a topic or field [16]. As such, the literature review becomes an important research tool for the process of understanding how the development of a mobile application can be done. For this thesis the review has been used to create the extended background of the thesis. The extended background includes topics of information that was found through the review that can help a reader understand the work of the thesis better.
2.4.2 Interviews

Interviews can be categorized into three different types. They can be structured, semi-structured and unstructured [17].

Structured interviews are questionnaires where the questions asked are predetermined. The interview type does not allow for any follow-up question. They are easy to hold, but provide limited responses from participants. The type is therefore no ideal if ‘depth’ is required [17].

Unstructured interviews are interviews with no essential structure. Instead the questions are open, where the participant could answer anything. The interviews progress based on the response of the participant. Unstructured interviews unlike structure take a lot of time and can be harder to participate in. The type is used almost only if almost nothing is known about a topic or perspective [17].

Semi-structured interviews are a compromise between structured and unstructured interviews. Predetermined key questions are asked however, the interviewer can diverge from the predetermined questions to dig deeper into a response of a participant. This type of interview compared to structured interviews can allow for discovering information from participants that was not known to the interviewer from before [17].

An unstructured interview was done with the stakeholder during the concept creation to assert what the stakeholder wants from a to be created prototype. From the assertion exploration criteria could be created. A semi-structured interview was used as a user-test to evaluate the created prototype. It gives the opportunity to understand how criteria of the prototype has been met in terms of functionality and user-friendliness.

2.5 Sampling

Projected users of the created smart clothing item are the target group for this thesis. The projected users are seen to be people that are attracted technology. Through convenience sampling engineers of different KTH faculties became the sample group as they suit the projected users and were easy to find for the author. A user-test was used to understand what the projected users of the created prototype thought of the prototype.

2.6 Validation

The concept of validity in a qualitative approach applies to the entirety of a research process. Validity is meant to detect phenomenons, interpret and understand the meaning of subjective answers and to describe perceptions or, and, to understand a certain culture [18]. Qualitative Research looks at four principles of validity in order to assure the soundness of the research. The four principles are: (1) Credibility, (2) Transferability, (3) Dependability and (4) Confirmability [19].

The first aspect is credibility. Credibility can be described as the trustworthiness and believability of the findings, while the second principle, Transferability is explained as the degree of how the research results can be transmitted onto the reader and how well the results can be generalized. Dependability is in qualitative research limited to the reliability of the results. It simply shows how reliable the results of the study are. The fourth principle, Confirmability, refers to the degree that the results of the dissertation can be confirmed and verified by other researchers and results [18] [19].
3. Theoretic Background

This chapter describes the information retrieved that have been deemed to be relevant subjects for the work to implement a prototype. Section 3.1 explains why atmospheric pressure could be used to build a weather forecast and presents a model that converts atmospheric pressure to weather predictions. Section 3.2 mentions visualizing information to a user. Section 3.3 describes the fundamentals of iOS creation. Section 3.4 goes through related work, weather forecasts in different mediums and wearable technology.

3.1 Atmospheric pressure and Barometers

Atmospheric pressure can be defined as the weight of air at ground level. In other words it is the weight of air from the top of the atmosphere down to the surface location of the measurement [20].

A barometer can be used to measure atmospheric pressure [21]. Barometers are tools to predict weather changes. A difference of atmospheric pressure is created by measuring the atmospheric pressure during different occasions. Tendencies in atmospheric pressure can allow for a weather forecast [5].

Weather maps use the symbols “L” and “H” to show that a certain area is predicted to have high or low pressure. That a certain area is predicted to have high or low pressure actually means that the atmospheric pressure is higher or lower at the specific area then it normally is [22].

Traditional barometers have used millibars (100 pascals) to measure atmospheric pressure [23]. Modern digital barometers use the unit of hectopascals (100 pascals) [24]. It is the SI unit for measuring atmospheric pressure.

3.1.1 Weather Forecast Based On Atmospheric Pressure

Changes in atmospheric pressure is used to create weather forecast. Given how the atmospheric pressure changes a prediction can be made. A decrease, rise or an unsettled change of atmospheric pressure relieves different forecasts. Generally speaking, a rise in atmospheric pressure would indicate stable weather while a decrease indicates bad weather [8]. There are various models that create weather forecasts based on changes in atmospheric pressure. Many rely on more variables than just atmospheric pressure[25]. One traditional model uses current weather (Wet, Frosty, Hot, Fair) to predict what changes in atmospheric pressure mean [26]. The resulting forecast can denote that there is a high chance of thunder, rainy weather, stable weather etc.

NXP, a semiconductor company [27], has created a simple model for creating forecasts based on atmospheric pressure [28]. The model was created for their digital barometer MPL115A1 [29]. The model revolves around categorizing atmospheric pressure changes into 5 different weather forecasts; not stable, stable good weather, stable weather, stable rainy weather and unstable rainy weather (thunderstorms etc.) as shown in Table 1. Changes in time and pressure is used for the model to predict the weather. Forecasts are created through measurements taken periodically every 30th minute for three hours. Throughout these measurements the forecast matures. Table 1 shows how differences in pressure (measured in kilo-Pascal) per hour is mapped to a weather prediction.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Weather Prediction</th>
</tr>
</thead>
<tbody>
<tr>
<td>dP/dt &gt; 0.25 kPa/h</td>
<td>Quickly rising High Pressure System, not stable</td>
</tr>
<tr>
<td>0.05 kPa/h &lt; dP/dt &lt; 0.25 kPa/h</td>
<td>Slowly rising High Pressure System, stable good weather</td>
</tr>
<tr>
<td>-0.05 kPa/h &lt; dP/dt &lt; 0.05 kPa/h</td>
<td>Stable weather condition</td>
</tr>
<tr>
<td>-0.25 kPa/h &lt; dP/dt &lt; -0.05 kPa/h</td>
<td>Slowly falling Low Pressure System, stable rainy weather</td>
</tr>
<tr>
<td>dP/dt &lt; -0.25 kPa/h</td>
<td>Quickly falling Low Pressure, Thunderstorm, not stable</td>
</tr>
</tbody>
</table>

Table 1: NXP Weather forecast model
Atmospheric pressure can also predict wind speeds. A model has been created (shown in Table 2.) that looks how atmospheric pressure in the region of the North Sea and Baltic can be used to predict wind speeds in the region [30]. The model measures the pressure difference (measured in hecto-Pascal) per hour. Beaufort scale (Bft) is used for measuring wind speed [31] and is used in the model as the unit for the wind speed prediction.

<table>
<thead>
<tr>
<th>Measurement</th>
<th>Wind Speed Prediction (Bft)</th>
</tr>
</thead>
<tbody>
<tr>
<td>dP/dt &gt; 3.3 hPa/h</td>
<td>10</td>
</tr>
<tr>
<td>dP/dt ≈ 3 hPa/h</td>
<td>8-9</td>
</tr>
<tr>
<td>dP/dt &gt; 1.3 hPa/h</td>
<td>6-7</td>
</tr>
<tr>
<td>dP/dt ≈ -2 hPa/h</td>
<td>6-7</td>
</tr>
<tr>
<td>dP/dt &lt; -2 hPa/h</td>
<td>8-12</td>
</tr>
</tbody>
</table>

*Table 2: Wind speed prediction given from pressure difference*

### 3.2 Visualizing Information on Mobile Applications To The User

Information visualization is the art of representing data in a way that it is easy to manipulate and understand [32]. The science of Human Computer Interaction (HCI) is the science of the “layer” that separates a human to the system. One important aspect of HCI is to ease the interaction between a human and computer.

User interfaces (UI) are systems that allow for interaction between users and a system [33]. In a mobile application a user interface is everything a user can see and interact with [34]. Since the user interface is what a user sees it is also the visualization of information that an application wants to give to the user. The design of the user interface has to therefore be created with usability and user-friendliness in mind.

#### 3.2.1 Usability

Designing a product to be useful is helped by a structured and systematic process, beginning with high-level goals and moving to specific objectives. True usability is invisible, for example if something is going well, we don’t notice it [35]. And, though there are quantitative approaches to testing the usability of products, it is impossible to measure the usability of something. We can only measure how many problems we have using something.

ISO 9241-11 is an international standard for “Guidance on Usability” [36]. It provides a definition for the term usability. “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” is how the standard defines usability [36].

Efficiency is the quickness with which the user’s goal can be accomplished accurately and completely and is usually a measure of time. For example, you might set a usability testing benchmark that says “95 percent of all users will be able to load the software within 10 minutes.” [35].

Effectiveness refers to the extent to which the product behaves in the way that users expect it to and the ease with which users can use it to do what they intend. This is usually measured quantitatively with an error rate. [35]. Effectiveness and Intuitiveness correlate. It means that an effective application is bound to be intuitive.

Learnability is a part of effectiveness and has to do with the user’s ability to operate the system to some defined level of competence after some predetermined amount and period of training (which may be no time at all). It can also refer to the ability of infrequent users to relearn the system after
periods of inactivity. [35]. Satisfaction refers to the user’s perceptions, feelings, and opinions of the product, usually captured through both written and oral questioning

3.2.2 Intuitive Interaction

According to the Oxford English Dictionary ‘Intuition’ is a phenomenon in which a person understands something instinctively without the need of conscious reasoning[37]. In connection with mobile phones, this definition can be interpreted as the ability to understand a user interface without conscious reasoning. Based on this description, intuitive interaction can be described as an experience where a user can directly use a user interface, as the interface works as the user expects. If an application would truly be intuitive then people of different technical background would be able to use the application in the same way, quick and somewhat effortless. This is somewhat hard to achieve. People get used to how different things work within their mobile environment. To achieve better intuitiveness of an application it is important to work within the guidelines of the platform. If a mobile application behaves as others on the same platform, users will instinctively know how to use the application.

3.2.3 Affordance

Understanding if you should pull or push to open a door is not always intuitive. If the door needs pushing and has a button that needs to be pressed by both hands you will be put in a position where you be able to push the door. This is affordance [38].

Affordance is used in user interface design, just like physical objects you have to make objects that work as a user would think they would. Unlike physical objects, digital views have to rely on representing usage clues through visual representation. Objects that are associated with user gestures are for instance buttons and scroll-able views [39].

Types of affordances related to user interface design and user experience are explicit, pattern, hidden, false, metaphorical and negative affordance. Explicit affordance can be seen in a raised button that state click me. The raised level of the button gives a clue that the button can be pressed. The wording of “click me” tells a user explicitly that it is possible to click the button. Pattern affordance is used for the format of pages. Knowing that a link is blue and underlined for a certain interface is due to pattern affordance. Hidden affordances try to remove certain objects so that the user is not overwhelmed with info. Instead objects of lesser importance are put for instance into a drop down menu. False affordance is when intuitive clues for objects don’t lead to the assumed usage. Metaphorical affordance is a way to design digital objects to work as a certain physical object. Negative affordance happens when objects that tend to have a function don’t do anything.

3.3 Structure of an iOS Application

It was concluded by the stakeholders that the mobile application created should be an iOS application. For development purposes it is of reason to understand the structure of an iOS application. This section presents Apple iOS Documentation on topics of interest.

iOS development is done through the Object-oriented design pattern of Model-View-Controller (MVC) [40]. Objects in an application can be assigned into one of three roles model, view, or controller. The Pattern defines how objects communicate with each other. The controller can speak to the Model and Views, Models and views can respectively use notifications and delegates to the controller. Objects of the form of models and views cannot talk with each other.

Model objects can be referred as the “What” of the application. The back-end of the application is situated here, such as the programmed algorithms that are meant to be used. Controller objects are responsible for the “How’s” of the application. The controller decides how the model should be presented to the user. View objects are the dependents of the controller. They are used by the controller to present the model to the user. This includes generic UI items [41]
Creating user interfaces for iOS involves managing controllers and views. View controllers manages a part of a mobile applications user interface. Content view controllers handle a specific page. Container view controllers organize how content views will get displayed. For instance a split view controller allows two view controllers to fit onto the screen simultaneously. Another way to organize pages is through a Navigation controller. An instance of use is when use want to allow navigation back and forth between pages. Every content view controller (which is inferred as just a view controller) can hold multiple view objects. A view object are items that you see on your screen. They can be labels, buttons or a custom object. View objects can call methods upon user interactions.

The creation of a user interface and user experience is in iOS development done in parts through the Storyboard as seen in Figure 3. It is a visual representation of the application as seen in Illustration 1. It contains the different pages of the application and the connections of the pages. The creation of view controller and page connections (segues) can be done in the storyboard through click-and-drag procedures of Xcode’s Interface builder.

The software for the creation of iOS applications is done exclusively through Xcode. It is an IDE that allows creation of applications to any device in the Apple eco-system. The most up-to-date programming language for iOS creation is Swift 3.0, an object-oriented language. The language strives for a simpler syntax.

Figure 3: A storyboard created in Xcode (by author)
3.4 Related work

Related work for the project are other mediums that display weather forecasts. Additionally, the current scene is mentioned through describing the functionality of trackers that have been deemed relateable to the application creation.

3.4.1 Weather Forecasts

Weather forecasts is not a new phenomenon. Existing weather forecasts can give explanations into the essentials of representing a weather forecast. Forecasts look different depending on the media. It is of interest to determine the differences and similarities between forecasts on different media.

Video-Based Weather Forecasts

Video-based forecasts can allow for longer explanations of the weather. Due to the format a consumer will have extended time to grasp the information. The video is narrated by a weatherman, explains the weather with familiar terms. Figure 4. are pictures taken from an NBC New York video forecast (“New York Weather,” n.d.). The pictures show how the video is narrated and how with colors chances of different precipitations types are showed. Continuing on the report switches to a classic 7-day weather forecast. This view shows an image of the weather, temperature at day and morning and the subjected dates. Additionally the picture shows a description of the temperature at certain dates. This could be of the reason that at times temperatures do feel warmer or colder than the temperature indicates.

Website-based Weather Forecast

A popular site for checking the weather is the The Weather Chanel [46]. The analyzed page seen in Figure 5. is the general page for a city. In the top left there is an area that shows the forecast of the current date. It explains the weather mostly through text. There are still graphical representations. There is an icon of a cloud, graphically representing what the weather looks like. There is a background to the top left box that is a picture of the weather. Together they are combined to allow for 3 different descriptions of the weathers excluding the temperature. Similarly to the video forecast, there is a 36 hour forecast that adds wind and rain percentage to temperature and titles of the days.
Mobile application-based Weather Forecast

The studied application is the android version of AccuWeather, it has over 50 million downloads on Google Play [47]. Figure 6. shows different pages of the application. The left-most page in Illustration has a descriptive detail of the current weather through an icon and text descriptions of the weather. It involves a “feels like” temperature. “Looking Ahead” a sub-section of the mobile application that details a prognosis in text of how the weather going forth will look like. More detailed views on the application shows a graphical representation of what areas will have rain.

3.4.2 The Current Scene

Throughout the last couple of years, the clothing industry has produced items that embed technology onto the items we wear. One genre of products is wearable activity trackers. They are electronic devices such as wristbands that are used to monitor fitness and other health related metrics of the user [48]. These devices has seen a surge of popularity during the last couple of years. 69% of the U.S. population keep track of health indicators through various means, an indication that there is a market for trackers.

Instances of wearable activity trackers has seen the use of a mobile application that allow people to track a health indicator. Nike FuelBands, which track workout activities of user, use “fuel points” to
explain for a user the workout quality [49]. The decision to create the unit of “fuel points” could have been to not overwhelm people with information. People have showed a frustration in not knowing what the points mean. This shows the difficulties of displaying measurements without the representation of the measurements becoming too convoluted or vague to understand.

Figure 6: AccuWeather Android Application
4. Work Structure

This chapter mentions how the work was structured. Section 4.1 explains how the pre-study was planned and the reason behind each phase of the stage. In section 4.2 the software development process is introduced and the iterations of implementation are explained. Continuing, Evaluation criteria are set up in section 4.3. The last section 4.4 explains how a testing phase will be planned to proceed and see how well the prototype meets the initial criteria.

4.1 Pre-Study

The pre-study includes five phases. They are, in chronological order (1) Idea Generation, (2) Feasibility Study, (3) Literature Study, (4) Concept Creation and (5) Choice of Weather Forecast Model. The following sub-chapters will explain each phase with the exception of Literature Study as it has already been mentioned in Chapter 2. as a research tool. The main focus of the pre-study was to be asserted that all necessary information had been obtained to start developing the prototype. Everything after the pre-study stage can be seen as prototype development.

4.1.1 Idea Generation and Feasibility Study

Idea Generation was the initial phase of the project. The phase was done to examine what can be measured that is of interest to a company that makes raincoats and other outdoor-wear. Together with the stakeholder and two other members of the smart clothing creation team a whiteboard was used to brainstorm all possible implementations. The brainstormed implementations were then ranked. The first ranked implementation involved using atmospheric pressure to create a weather forecast. The group decided together to proceed with the first ranked implementation.

The feasibility study proceeded the Idea generation phase. It involved searching for weather forecasting using atmospheric pressure through the Google. This phase was a quick test to understand whether the idea of using air pressure to create a weather forecast was in fact feasible.

4.1.2 Concept Creation

Since there were no vivid functionality criteria more than that the mobile application should somehow create a weather forecast for the user, a concept creation was planned. It can be seen as a throwaway prototype with the exception that it is not a functioning prototype. However, it shares a goal: to understand what functional requirements are necessary in order to follow the criteria from the concept creation.

The concept was created from drawing and was done together with the stakeholder. The reason for creating the concept with the stakeholder was to use the concept creation as an unstructured interview for creating criteria on how the mobile application should work and function. It was also done to understand what is needed from a weather forecast model.

The drawings were made without technical restrictions in mind. With created sketches it could be determined what underlying technical functions were needed be implemented and what weather forecast model would need to be able to do.

Figure 7 are two drawings that show what the application should display when starting the application. Furthermore the drawings show an example of what a user can see when a forecast has been made. The left page has a label describing to the user that no measurements have been done at the current location. There is also gesture indicator that should be swiped downwards to get a new measurement. There are labels describing to the user either that no more measurements has been made yet. The right picture has a label describing the prognosis and a label describing when the last measurement was made. The yellow circle represents an icon that represents the weather.
Figure 8 is another example drawing on how a page could look when a weather forecast is given. The drawing adds labels that describe where the user is located through coordinates and city name. It also holds a play-list button for a functionality of the prototype that is not of interest for the thesis. Furthermore, a label was added to describe when weather would change. Further drawings were made to grasp the sequence of the mobile application (See Appendix A, in Swedish).

**Figure 7**: Left page shows how the application before measurements are taken. Second page shows an example of how a stable weather prognosis would look like

**Figure 8**: A page shown when better weather is predicted
4.1.3 Choice of an Applicable Weather Forecast Model

Based on limitations of the given smart clothing item, applicable models for weather forecast creation were restricted to only using atmospheric pressure and other measurements an iPhone is able to provide. Given the exploration criteria (See Table 4., Section 4.3.2), which was created in the concept creation phase, an applicable weather forecast model should be able to map atmospheric pressure differences to different forecasts. Furthermore, it should be able to predict when a weather prediction will happen. The resulting research showed no applicable models that could meet both requirements using atmospheric pressure differences. However, only one model was found that could satisfy the criteria to map atmospheric pressure differences into different forecasts. Although predicting when weather will change within a X amount of minutes was a criteria, the chosen weather forecast model did not have the feature to show in minutes when weather change is expected. And so, the criteria cannot therefore be seen as something that can be met by the resulting prototype.

4.2 Software Development Process

This thesis uses the prototype model since a prototype is created without any initial requirements. An iterative model is used for this thesis as it is well suited for smaller teams and allows for changes throughout the process of creating a prototype.

Throwaway prototyping will be used in the concept creation stage of the project. It is partially done to determine exploration criteria. To understand how measurements would be converted, it had to be understood which weather forecast model suited the concept creation the best. After choosing the model the criteria whom which the model is not suited are removed as they are improbable to achieve.

4.2.1 Iterations of Implementation Phases

The iterations of work were planned the following way: The first phase was planned to create a user experience for the prototype. The purpose of the phase was to understand, design and implement how the pages of the drawn in the concept creation (see Appendix A.) could be translated into pages in an iOS environment. The second phase was planned to design and implement all the elements of the user interface. The phase was planned to allow all the elements of the user interface to work that wasn’t dependent on the weather model. Furthermore, those who were prepared to allow for the implementation of algorithm that was to be based on the weather model.

The third phase was planned to implement the weather forecast as an algorithm, and update the of the view controller that was dependent on getting a update when a new weather measurement or forecast was made. The essential goal with this phase was to find a way to notify the view controller in real time of new measurements and forecasts.

4.3 Evaluation of Prototype

To evaluate the thesis, criteria have been set up. The criteria have been categorized into two sets: Evaluation criteria and Exploration criteria. The initial evaluation criteria were set-up to answer the usability of the prototype. The exploration criteria compare the finished prototype with the concept creation. Features will be evaluated by testing the application. To see how well the evaluation criteria have been met, a brief post-prototype user-test that involves a user-test interview that was planned to answer questions of the prototype such as user-friendliness, value of forecast and use-cases[35]. Exploration criteria was planned to be evaluated by comparing the resulting prototype with the criteria created in the concept creation.
4.3.1 Evaluation Criteria

This thesis has 3 initial criteria. They each try to partially answer the usability of the prototype. Table 3 presents these criteria. They are the following:(1) Effectiveness, (2) Usefulness, and (3) Efficiency. A user-test done after the creation of the prototype helped evaluate how the criteria have been answered. The user-test tries to answer the level of effectiveness by asking what the participant thinks about the user-friendliness of the application and how well they could understand the prognosis. The criteria is assessed by asking people what they think about the value of the prognosis. Did they think it satisfied there needs, what could have been improved?[35]. The subject of the interview question to satisfy the criteria Efficiency therefore revolves on how long a user would wait for a forecast given if either he has internet connection or does no.

<table>
<thead>
<tr>
<th>Criteria</th>
<th>Subject of Interview Question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness</td>
<td>User-friendliness of application</td>
</tr>
<tr>
<td></td>
<td>Intuitiveness of prognosis</td>
</tr>
<tr>
<td>Usefulness</td>
<td>Value of prognosis</td>
</tr>
<tr>
<td></td>
<td>Improvements of prognosis</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Willingness to wait for a forecast without internet connectivity</td>
</tr>
<tr>
<td></td>
<td>Willingness to wait with internet connectivity</td>
</tr>
</tbody>
</table>

Table 3: Initial Evaluation Criteria

4.3.2 Exploration Criteria

Comparing the resulting prototype with the Concept Criteria makes an evaluation. Explorative criteria are shown in Table 4. This is done to see how well the prototype satisfies the stakeholders’ request from the application. It is also made to see how well an idea with limited technical background research can be implemented into a mobile application.

Criteria structure the functionality requirements of the concept into user interface requirements and weather forecast model requirements. The user interface category has four requirements. The prototype has to be able to track the user. Describe the weather to the user through text and image and have the functionality to time the last time a measurement was made. The weather forecast category has two requirements the forecast should be able to map measurements into different forecasts and it should also be able to predict when the weather will change.

Through pre-study was found that different criteria were necessary to complete the entire process of creating the prototype. These are described as prospecting criteria. The phase of creating exploration criteria was the concept creation phase. Prospecting criteria involve the prototype functionality based on concept creation. Table 4 structures the concept’s functionality in user interface requirements and in weather forecast model requirements.

<table>
<thead>
<tr>
<th>Category</th>
<th>Criteria from Concept</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>Location Tracking</td>
</tr>
<tr>
<td></td>
<td>Weather Description</td>
</tr>
<tr>
<td></td>
<td>Updating with the Weather Forecast</td>
</tr>
<tr>
<td></td>
<td>Timer between measurements</td>
</tr>
<tr>
<td>Weather Forecast Model</td>
<td>Mapped Forecasts</td>
</tr>
<tr>
<td></td>
<td>Predict When Weather Will Change</td>
</tr>
</tbody>
</table>

Table 4: Exploration Criteria from Concept Creation

4.4 Testing

Testing of the finished prototype was arranged be understood if the prototype works, as it should. Moreover it was a tool to understand how well the evaluation criteria has been met. It was planned to test the prototype first without using the measurements of a barometer and then afterwards test it with the measurements coming from a connected barometer. An interview was planned to evaluate how easy and understandable the created prototype would be. Additionally, the interview wanted to answer
the value of the created forecast and the use-cases of the prototype. The interview was planned to be semi-structured with predefined questions and follow up questions where the participant is free to answer whatever he wants. Engineers of different faculties were to be interviewed, as they could be seen as a target group for buying new technology. The interview was planned be followed up by case study questions to understand if the participants would find the forecasts useful during decided moments.
5. Implementation: Design

This chapter describes the detailed design of the implemented mobile application prototype. It covers the steps from that a mobile application receives a measurement of atmospheric pressure from barometric sensors until a weather forecast is displayed to the user in 5.1. and 5.2. Section 5.3 explains how the user interface was designed to look during the different phases of the prototype. 5.4 continues the chapter by explaining the graphical choices of the user interface.

5.1 Atmospheric Pressure to Weather Forecast

The NXP model[28] that converts atmospheric pressure to a weather forecast has been followed to create an algorithm that is suited for the prototype. Diagram 2. show the phases of the designed algorithm derived from the NXP model. The following paragraphs will describe the diagram:

The first phase of the algorithm occurs when the first measurement is received. The measurement is stored in an initial measurement variable. The initial value will be used as a comparison measurement for further measurements. The first measurement occurs at the start of the mobile application.

The next phase occurs when a new measurement is created. A new measurement is only taken if 15 minutes have passed since the initial measurement was. It can occur through a user requested measurement or an automatic measurement that happens every 30 minutes. User requested measurements are created when the user asks for a new measurement through a button push. The new measurement is then normalized to allow each measurement to be weighed by the same factor. The formula of \((60/y)\) where \(y\) is the time will be used to normalize each measurement to the time of one hour.

Figure 9: Phases of creating a weather prediction and how the weather forecast changes the user interface
The ensuing phase calculates the difference between the new measurement and the initial measurement. Afterwards, the difference value gets mapped to a forecast, at that point the forecast is done. When a forecast is achieved the user interface gets notified and updates the weather icon and description and sets the last measurement to zero.

5.2 Updating the User Interface

For each new measurement after the initial measurement a prediction can be displayed to the user. The predicted forecast is sent to a view controller that is always waiting for a new forecast. Given that it gets a new forecast the display gets updated with an icon and text that represent that forecast.

5.3 User Interface Layout

The user interface layout tries to follow the concept created in its basics. In terms of iOS development that involves two different view controllers. One view controller shows off the weather forecast, the other a music player playing music for when it rains. The page of importance for this thesis is the one showing the weather forecast. It can be divided into two time phases.

5.3.1 User Interface Elements

The main page has five main elements. An icon representing the weather, a text description of the weather, another text that shows either how long it has been since a measurement or if a measurement is being generated (text time), a button that requests a new measurement, and, location coordinates. The first four elements will look different depending on the sequence.

5.3.2 Gathering Measurement Sequence

The gathering measurement sequence occurs when a user opens the application for the first time (a measurement request will get called immediately) or at any other time a measurement is requested. The weather description and icon is hidden. The button is hidden and replaced by an activity indicator (buffer). Text time will show that measurements are being gathered.

5.3.3 New Forecast Sequence

The second sequence occurs when the algorithm gives out a weather forecast. This sequence is what the user will see most, during his usage of the mobile application. The weather description and the weather icon describe and represent the weather of the algorithms output. The same button as in the first sequence is shown and can be used to ask for a new measurement. The buffer is not shown in this sequence. The time text shows when the last measurement was made.

5.4 User Interface Design Choices

There was an agreement between the stakeholder and author of this thesis to keep the interface minimalist while still being able to provide the essentials the user needed to understand the application. The design ended up being flat.

5.4.1 Colors, Font, Text, Icons and Buttons

A gray-scale palette of colors was chosen as it suited the minimalist approach. The icons and text were a darker gray whilst the background was on the lighter side of gray. To keep it simple Apple’s system font SF Text font was chosen for the descriptions. The weather description text was chosen to explain the weather, as it is about to happen. Furthermore, certain weather forecasts give out motivation to the user. The icons of weather descriptions were chosen to try and explain the weather forecast in the simplest way possible. Quickly high rising results in a wind icon, slowly rising good weather results in a sun icon, stable rainy weather results in a rain icon, and thunderstorm results in a thunder icon. The concept idea was to create a swipe gesture that when swiped would update the weather. Instead the
6. Implementation: Programming

This chapter describes how the design was implemented to become an iOS application. Each component of the implementation is described. 6.1 Describes the implementation through the design pattern of MVC. 6.2 Summarizes how NotificationCenter was used to broadcast and update the user interface. 6.3-6.6 describes various methods used to allow for various functionality of the application.

6.1 Design Pattern

The “Model-View-Controller” pattern was used for this project. Creation of an iOS application is almost always done through this pattern.

The application ended up being fairly simple in its structure. Using two view controller, one of which is the focus in this thesis. Figure 10. shows the class diagram of the project.

6.1.1 Controller

The application uses one view controller for the page that displays the weather forecast. This view controller has several views, which it controls. It also holds several other variables that are used in the view controller.

6.1.2 View

The views can be communicated through code by their respective instance variables of the view controller. The page that displayed the weather forecast had four views.

Descriptions of time, weather, coordinates were implemented into UILabel which is the stock view for holding text. These labels are outlets for this application. Meaning their properties can be updated through other functions in the view controller.

UIButton was used for the button that allows the user to ask for a new forecast. UIButton view is a stock view that allows for basic button interaction. The view was extended to include a border with rounded corners. The button was implemented as an action. This means that an interaction with the button leads to a function being run. This is the only way it can run. This button is therefore not a variable which can have its properties changed from outside the function.

UIImageView was used for representing the weather with an image icon. This is the stock view class for holding images.

BackgroundView is a custom view based on the UIView. The difference between BackgroundView and UIView is the added properties. Corner Radius and Gradients are added properties that can be changed in the storyboard. It was used for the background of the page. The thought was that it would be reused for several views. The custom class ended up being reluctant since it was decided to only use a one colored background.

6.1.3 Model

The application used a single model. The model was used to update the user interface of the main view controller. The model includes the implementation of the model that creates the weather forecast. When an algorithm prediction is made it gets mapped to a certain forecast in the model and the view controller gets notified.

The view-controller holds an instance of the model. It is a variable. The variable has been set to be an instance of the model class. Methods of the model class can be called through the instance variable.
6.2 Broadcasting

In order to make the views of the view controller update when a new forecast was made a NotificationCenter was implemented. It provides a mechanism for broadcasting information within a program[50]. A method can post(send) a notification or observe a notification. When a notification is posted the observer calls a method.

Three different notifications were implemented for the application measurementInProgress, NewWeatherForecast and firstMeasurement. They are all posted(sent) in the model and observed in the viewDidLoad method in the viewcontroller.

measurementInProgress was a notification that notified the view controller that a new measurement has been initiated. The method called would hide the button that allows for a new manual forecast request. To replace it an activity indicator (buffer) would appear. The time label would also be updated to repeatedly show that measurements are being gathered.

NewWeatherForecast was a notification that notified the view controller that a new forecast had been created. The method called in the view controller gets the forecast object that holds an enum of the new forecast. With the help of the enum it updates the icon, weather description and the measurement of the user interface. It also sets the time since last measurement to zero. Continuing, it calls to stop showing an activity indicator and instead show the button that allows for a new forecast. The method in the model that posts the notification is the weather forecast algorithm that runs the post when a forecast has been made.

firstMeasurement was a notification that notifies that first measurement was finished. Like, newWeatherForecast its observing method calls to stop showing an activity indicator and instead show the button that allows for a new forecast.

6.3 Weather Forecast Algorithm

The steps of creating a weather forecast creation shown in the Figure 9. (Chapter 6. Design) were used for the implementation. The implementation of the weather forecast model was simply a method with numerous if statements. The if statements were implemented to determine what the prognosis were and if it was to early to make a new prognosis. The method was also implemented to post notifications (see broadcasting)

6.4 newPredictionButton

The request for a new measurement was as stated done through a click of a button that results in an action method. The action method simply calls a method of the model instance variable to start a new measurement cycle.

6.5 updateUserDescription

The request for updating the user description is done at observation of newWeatherForecast. The method takes the enum that describes the weather given from the notification and finds the right icon and weather descriptions in dictionaries and set the weather description and icon to fit the enum.

6.6 timeInfoModifier

A method was created to allow the functionality of seeing when the last measurement was made. It is initialized when a new Forecast is made and gets called every 60th second until another forecast is made. The initialization happens when newWeatherForecast or firstMeasurement gets called. For every 60th second a time variable gets incremented by one. The time label is then updated to show “Last Measurement Made X minutes ago”. At a new forecast the time gets reset to zero.
Figure 10: Class diagram of prototype
7. Testing and User Test

The following chapter describes how testing was done. Sub-section 7.1 discusses testing if features of the application worked as intended. 7.2 mentions how the user-test interview was done.

7.1 Feature Testing

The last step in each iterative process phase was the testing phase. Xcode development allows several ways of testing the code. During the programming process a developer gets real-time warnings and errors that signals errors in a code line. For testing the application created, a developer can run the application on a simulator. The code can also be run with a debugger that allows for the user to insert breakpoints to test certain parts of the code. The simulator imitates a real Apple phone and can be chosen to be any iPhone or iPad model. By choosing different models one can test how the user interface looks on different devices. The restriction with the simulator is that Bluetooth connectivity can’t be tested. For testing functionality with the need of Bluetooth the created application can be run on actual hardware, an iphone. It allows to test all the functionality.

The functionality was tested first without the connection with the chip that holds the barometer. All the timers in the program were set to allow the prototype to run as fast as possible. A random number generator was used to send in theoretically reasonable values into the weather forecast algorithm. Breakpoints were added at every point a notification gets posted and observed. This allows for the partial test of whether the weather forecast algorithm gives out the correct forecast for the correct value. To see if the weather description and icon got updated correctly, the printed out weather forecast was compared with what the simulator shows. Other functionaries were tested the same way to see if they are called at the right instance.

The functionality was also tested with the barometer and the chip that holds it. The same setup was used. The only difference was measurements were done outdoors and the measurements used were from the barometer it was therefore necessary to wait for a real forecast. The purpose of this test was to see if the created prototype worked with the barometer and if not what the problem was.

7.2 User-Test

To evaluate the prototype further user tests were done that involved a brief interview and case study. The location of the user tests was “Nymble”, it is a place where engineers of all faculties hang out and study. 10 random engineers were chosen. The participants got a functional background of the prototype. Each question has a subject that the question tries to answer.

The following questions (noted Q:) were asked and the following answers (noted A:) were what the participants could answer:

Subject: User-friendliness of application
Q: Was it easy to use the application?
A: Yes, I found it easy
A: I'm impartial
A: No, I found it hard to use
A: No comment

Subject: Intuitiveness of prognosis
Q: Did you understand the prognosis you got?
A: Yes
A: No
A: No comment
Subject: Value of prognosis
Q: Did you find the prognosis valuable?
A: Yes, I found it valuable
A: I found it somewhat valuable
A: No, I did not found it valuable

Subject: Value of prognosis (follow-up)
Q: If the answer was no or somewhat, how could the forecast have been improved? (follow-up)
A: (short answer)

Case Study: Usage Scenarios

After the above questions were answered a participant got asked for how long they would wait for a measurement of the prototype in a certain scenario.

Subject: Willingness to wait for a forecast without internet connectivity
Q: Say you are camping or adventuring at a not densely inhabited area and have not internet. As such it might be hard for you to get a weather forecast. For how long would you be willing to wait to get a forecast from the application?
Note: the application can run and predict a forecast automatically.

A: I wouldn’t wait for a forecast
A: less than 5 minutes
A: 5-10 minutes
A: 10-30 minutes
A: 30-60 minutes
A: 1-3 hours
A: more than 3 hours

Subject: Willingness to wait with internet connectivity
Q: For daily use where you have internet and are able to get weather forecasts from other sources. For how long would you be willing to wait to get a forecast from the application?
Note: the application can run and predict a forecast automatically.

A: I wouldn’t wait for a forecast from the application
A: less than 5 minutes
A: 5-10 minutes
A: 10-30 minutes
A: 30-60 minutes
A: 1-3 hours
A: more than 3 hours
8. Results

This Chapter describes the resulting prototype - the mobile application. Section 8.1 summarizes the results from feature testing. Section 8.2 shows how the resulting prototype’s user interface ended up looking. Section 8.3 mentions the feedback given from the interviews done to evaluate the prototype.

8.1 Result of Feature Testing

Tests without the barometer showed that given that two measurements have been generated to have the difference within a certain forecast prediction, the algorithm notifies the view controller with the correct prediction.

Tests that were done with the barometer connected, showed that only one forecast was showed during the time-frame of 3 hours. The forecast showed on the prototype was thunderstorm. The cause was that all measurements gathered were all around negative thousand (kPa) that is completely unreasonable. The actual weather throughout the day was sunny and stable.

The test shows faulty in an area that is outside the spectrum of the thesis but will be discussed in Chapter 9.

8.2 User Experience and Interface

The resulting interface can show five different forecasts. A user can interact with the application through a push of button to request a new forecast. While a new forecast gathers the prototype is in a Buffer Sequence as seen in Figure 11. When the forecast is finished it updates the interface to one of the following forecasts seen in Figure 12.

![Figure 11: Buffer Sequence](image)
8.3 Results of Evaluation

To evaluate the prototype criteria of evaluation have been established (see section 4.3). The criteria have been categorized into two sets: Evaluation criteria and Exploration criteria (see Tables 3 and 4). Section 8.3.1 shows the result of the evaluation criteria and 8.3.2 shows the result of the exploration criteria.

8.3.1 Result of Evaluation Criteria

The result of Evaluation Criteria are gathered from accomplished user-test and is shown in Table 5. It evaluates the usability of the prototype. The resulting diagrams of the interview can seen in Appendix B. The interview had 10 participants of engineering background.

8.3.2 Result of Exploration Criteria

Comparing the resulting prototype with the exploration criteria created at the concept creation stage is done here to evaluate to what extent the set-up criteria has been met. The criteria involves prototype functionality. The results is shown in Table 6.

Location tracking is found in the prototype. The user interface shows an always-updated user location. It is essentially done in the same way as the idea was in the concept creation. It can therefore be stated that the criteria has been met.

<table>
<thead>
<tr>
<th>Initial Criteria</th>
<th>Subject of question</th>
</tr>
</thead>
<tbody>
<tr>
<td>Effectiveness</td>
<td>User-friendliness</td>
</tr>
<tr>
<td>Usefulness</td>
<td>Value of prognosis 30%</td>
</tr>
<tr>
<td>Efficiency</td>
<td>Willingness to wait for forecast without internet connectivity</td>
</tr>
<tr>
<td></td>
<td>10% can wait for 5-10 minutes.</td>
</tr>
<tr>
<td></td>
<td>60% would wait 10-30 minutes</td>
</tr>
<tr>
<td></td>
<td>30% would wait 30-60 minutes</td>
</tr>
<tr>
<td></td>
<td>Willingness to wait for forecast with internet connectivity</td>
</tr>
<tr>
<td></td>
<td>60% would not wait</td>
</tr>
<tr>
<td></td>
<td>40% would wait for 5 minutes</td>
</tr>
</tbody>
</table>

| Table 5: Result of Evaluation Criteria |
The resulting prototype has weather descriptions that describe the weather through icons. The concept creation didn’t show how the icons would look like. The result has a text label describing the weather.

Updating with the weather forecast was a criteria that came through the assertion that the mobile application has to be able to inform the user instantly and automatically that a new forecast has been created. The resulting prototype gets updated not only when a new forecast has been made but also when a new measurement is being created. This allows the user to understand what is going on with the mobile application at the current moment.

The criteria of mapped forecasts was created to be able to have predetermined interval of what differences in atmospheric pressure mean in terms of a prognosis. The resulting weather forecast model’s properties allows for the creation of an algorithm or rather a filter that determines what a measurement means in terms of a forecast.

Timer between measurements a criteria meant to allow the user to understand when the last forecast was implemented fully in the prototype. The prototype shows when the last measurement was made in minutes. Additionally, the prototype tells the user when measurements are being gathered and when it is too early to tell the forecast.

Determining when weather changes will happen was deemed non-realistic after the choice of using a weather forecast model that didn’t have the properties to predict when weather changes would occur. Therefore that criteria could not be met.

<table>
<thead>
<tr>
<th>Category</th>
<th>Criteria from Concept</th>
<th>Achieved?</th>
</tr>
</thead>
<tbody>
<tr>
<td>User Interface</td>
<td>Location Tracking</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Weather Description</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Updating with the Weather Forecast</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Timer between measurements</td>
<td>YES</td>
</tr>
<tr>
<td>Weather Forecast Model</td>
<td>Mapped Forecasts</td>
<td>YES</td>
</tr>
<tr>
<td></td>
<td>Predict When Weather Will Change</td>
<td>NO*</td>
</tr>
</tbody>
</table>

*The nature of the chosen weather forecast model does not allow to predict in a 10-minute approximity when the weather change will occur

Table 6: Results of Exploration Criteria
9. Discussion

This chapter discusses upon the results obtained in this thesis. Section 9.1 reflects on how the result answers to the research question, purpose, and, goal set up in the introduction chapter. Section 9.2 discusses the methods used. Section 9.3 mentions credibility. 9.4 discusses the two sets of criteria results. 9.5 ends the chapter with general thoughts on the functionality of the prototype.

9.1 Result Discussion

The goal of the project was to present how weather forecast could be modeled and implemented into a smart clothing system. A prototype has been created that converts atmospheric pressure to a weather forecast that is then displayed through a user interface to the user. The conversion is done through an algorithm that is based on a weather forecast model. The user interface presents the forecast to the user through symbols and text that are suited for a mobile interface. The resulting prototype is able to predict the weather given that it gets measurements that are dependable and correct. It can therefore be stated that the goals of the project has been reached.

Section 1.3 states the purpose of this thesis. The purpose was the following: ‘The purpose of this thesis was to present an implementation of a prototype mobile application that has the ability to create a weather forecast given atmospheric pressure measurements and display it in a way that is suitable for a mobile interface. ‘ A functional prototype has been created that both converts atmospheric pressure to a weather forecast and displays it to the user. The purpose of this thesis has been met, as the prototype has the ability to create a weather forecast. It can only be stated that the prototype should be able to predict the weather. Any real-world testing proved impossible due to obtaining incorrect atmospheric pressure measurements, which is outside of the scope of this thesis.

Two main benefits are met by the outcome of the thesis. Firstly, the outcome allows for mobile measurements and forecasts that are not bound to a specific position. In short, the prototype allows for mobility. Secondly, the outcome allows for a weather forecast that does not require internet connectivity. A benefit that cannot be determined if met is the subject of warning people about possible health concerns due to weather changes. At no time does the prototype try to tell the user about health changes. However, users could still use the mobile application as a tool for determining possible health concerns before they happen.

9.2 Method discussion

The thesis presents an implementation of a mobile application that has the ability to create a weather forecast. It explores and explains the process needed to develop a functional prototype. The research approach was inadvertently qualitative. To evaluate the prototype a User-test was conducted. A quantitative approach was for this step since the data of the User-test was in the form of numbers(percentages).

The study is based on an inductive research effort that explores factors surrounding the development of a mobile application. In other words, the study presents an explanation of how a weather forecast prototype can be implemented.

9.3 Validity

As mentioned in Chapter 2, the research was exposed to various validity threats. Such threats affect the credibility, transferability, dependability, and confirmability of the thesis. The research process tried to tackle these threats by assuring that each of the validity factors of qualitative research was addressed.
To be Credible, the choice of method must be highly objective and reliable [11]. Great reliability assures that the research process has significant Dependability. This study uses generally a qualitative method. To adhere to the credibility threat a user-test was done to link the research with reality answering: What does the target group think of the prototype? In order to achieve solid reliability, a semi-structured interview format was used for the user-test. The advantage of semi-structured interviews are that the respondents are given the opportunity to express themselves more freely than a structured interview while also having predetermined questions that makes the interview easier to analyse than an unstructured interview. The interview format was used with the aim of increasing objectivity, giving follow-up questions when more information was of interest (See Chapter 7.).

The user-test could have been improved. While all respondent fit the target group they are all from the same background, being engineering students. What engineering students think and have experience with might not necessary reflect onto the target group as a whole. The objectivity of the results could therefore have been affected. As for the reliability of the study, it is relevant to mention that the user-test was carried on all at once. To assure better reliability, the user-test could have instead been done on several occasions with a mix of people from the target group. Higher reliability would lead to other researchers being able to duplicate and verify the results of the interview. A greater Confirmability would have been assured.

Transferability was assured by explanation, describing how the prototype and user-test was created. The thick description allows the reader to have a proper understanding of the results. With a proper understanding the reader may find areas in which he applies to the gathered knowledge.

9.4 Discussion of Evaluation

The results of the user test show how well the created prototype has achieved the evaluation criteria. The created prototype shows that 4 of the 5 criteria made in the concept creation have been reached partially or fully. Location tracking, tracking the user’s coordinates to allow the user to understand that the forecast is being mapped to the current location was successfully implemented. In the prototype the user interface dynamically shows the coordinates of the phone. Weather Description, the functionality of allowing the user to get an accurate weather description based on current prediction was implemented through describing the weather forecast through icons and text. The icons and text are based on the forecast descriptions of the weather forecast model. The criteria of updating with the weather forecast was reached by using the notificationCenter method that allowed the model of the program to notify the view controller each time a new forecast was made. This allowed the user interface to show the user to show the always up to date forecast to the user. Timer between measurements was like the previous criteria made possible through notifications. Each time a new forecast was made the timer was reset to zero. Reaching the criteria allowed the user to see when the latest forecast was created.

Mapped forecasts, is a functionality that is based on the properties of the weather forecast model. A chosen weather forecast model has to specifically be able to tell what a differentiation in atmospheric pressure means. In other words an atmospheric pressure difference value should always return the same forecast. The chosen weather forecast model was able to do so. The prototype implementation of the model was tested and displayed the right weather prediction given the atmospheric pressure difference measurement. Predicting when weather change due in terms of a 10-minute approximation was deemed improbable as the chosen forecast model did not allow that functionality.

The user-test allows evaluation of the initial evaluation criteria. The effectiveness criteria are shown has been reached as the questions correlated to the criteria (User-friendliness and Intuitiveness). Result shows that a large majority believes the application is easy to use and the prognosis is intuitive to understand. The answers of the user-test show that the usefulness criteria cannot be deemed have been fully achieved. The participants deem that the prototype lacks the functionality of stating when a weather forecast will be due. Everyone did however see the prototype being somewhat to very useful for their needs. The usefulness criteria can therefore be seen as partially achieved.
From the results gathered from the user test it can be concluded that application is easy to use and the weather forecast description was simple to understand. In other words intuitive (to a certain extent) and user-friendly. It can also be derived that the criteria of specifying *when a weather forecast was due* was important for the value of the forecasts made. Indicating in what hour weather prediction would occur might have been a good enough approximation to solve the issue of the forecast being somewhat interesting.

Participants of the user test showed that people were only interested to wait for a prognosis of the prototype if they were in a situation where they don’t have internet giving them no possibility to check other forecasts. It can be interpreted that the interest in using the product is higher in situations where the customer lacks internet connection. Thus, there is a link between product usability and internet connection. It can be also argued that the structure of the question allowed people to think that they actually have to sit and wait for the forecast. In reality the participant could have done whatever he wished during the time frame of the forecast creation.

### 9.5 Functionality Discussion

The prototype is reliant on the functionality of the model used for implementing the weather forecast. However it needs time to create a precise weather forecast. A user won’t get the best prediction until 3 hours of using the application, but on positive side, the preliminary weather forecast can be established within 30 minutes of using the application if requested. A push of a button won’t give the user an instant update of the forecast. The user has to wait for measurements to be gathered. The weather forecast algorithm is restricted in that the barometer takes a minute to create a measurement. This means that the user has to wait a minute after requesting a new forecast.

The necessity of a button for requesting a new forecast can be questioned. Instead the functionality of the button could have been automatic. Given that the minimum time needed for creating a weather forecast is met the forecast is updated every minute. The removal of the button would also remove the need for interactivity with the user. A further question becomes if the mobile application could be replaced by a widget or a background task.

Two ways to remove the uncertainty of whether the application actually can predict the weather would have been to verify the model before the implementation into the system. The second would be to compare it with other forecasts given that the measurements of the barometer are correct.
10. Conclusion & Future Work

Through the used development process a weather forecast prototype has been created. The prototype can convert a measurement of difference in atmospheric pressure into a weather forecast, achieved through the implemented weather forecast model. The forecast is displayed to the user through an UI. An analyze of the User-test shows that the target group – Tech enthusiasts, find the the prototype’s UI intuitive. Furthermore, it shows that one of the major strengths of the prototype was the ability to predict the weather without the need of a internet connection. Additional functionality could have been achieved if the weather forecast model could have estimated at a 10-minute approximation when a weather change was bound to occur.

A created development process was a necessity in creating the prototype. Without it there would not be enough structure for the prototype to be created and evaluated. The development process lead to the finished prototype. More iterations of the process and prototype could have lead to a better prototype. The process still has the ability to be refined. It would have been beneficial to add further interviews with the target group throughout the period of developing the prototype. Revising the criteria with each interview to make the resulting prototype suit the wants of the target group better. With that said, a similar process as to the one used can be used for development of other Smart Clothes. The difference in approach would lay in the research topic and choices of parameters. Parameters that are used to find an applicable model for displaying information to the user.

An interesting way forward would be to see how the prototype could be implemented further. One way would be to allow the prototype to “talk” with other weather stations and share that data between the measurement holders. It would be a way to decentralize weather forecasting. The final result of such implementation could lead to improving weather forecasts globally if a large user base is found. This idea extends weather forecasts. In fact, all Smart Clothes that measure a users surroundings and has internet connectivity at some point of the day can be scaled in the same way.

Additional features could be added to the prototype. When a forecast is made a notification should be sent to the notification center. The feature would allow the user of the prototype to get updates on forecasts even if he’s not currently in the application. An obstacle of adding this feature would be to make sure that the prototype can run as a background task. Another feature would be to fetch forecasts from a weather forecast provider when the user has internet connectivity. Both forecasts could then be displayed or used to create one improved forecast.
References


Stanford CS 193P iPhone Application Development - Lecture 2: MVC; iOS, Xcode and Swift Demonstration.


“View object,” Apple Developer. [Online]. Available:


“NotificationCenter - Foundation | Apple Developer Documentation.” [Online]. Available:
Appendix A

Start a measurement if no measurements of the current location

New location. Want to determine rain?

Start Measurement

No indications of bad weather; last measurement 5 min ago

STUTTERHEIM

Vad när den inte hittar regn progros?

Sannolikhets index för regn
Kanske 3?

Kommander vart du är, så användaren vet att vi mäter just där

Rain Description
It will rain in 3 hours. Embrace it.

När den hittar regn
Album cover
Artist namn + låt
Byta låt?
Få bitar utav låt vid varje regn tillfälle?

När den kommer i kontakt med regn

Pålist

last measurement 5 min ago

N 32
E 122

Clear skies ahead

Stockport Heights

In 30 min expect the weather to change

När prognosen ger ut klart värde
No Indications of Bad Weather
last measurement 5 min ago

STUTTERHEIM

Tillbaka hem - Vad når den inte hittar regn prognos?
Vad når den inte hittar regn prognos?
Appendix B

Did you understand the prognosis you got?
10 svar

- 100% Yes

Did you find the prognosis valuable?
10 svar

- 70% Yes, I found it valuable
- 30% I found it somewhat valuable

If the answer was no or somewhat, how could the forecast have been improved?
7 svar

- 1 (10%) Be more specific
- 1 (10%) No temperature prognosis
- 5 (50%) Show when the prognosis is...
Was it easy to use the application?

10 svar

Say you are camping or adventuring at a not densely inhabited area and have not internet. As such it might be hard for you to get a weather forecast. For how long would you be willing to wait to get a forecast from the application? Note: the application can run and predict a forecast automatically.

10 svar

For daily use where you have internet and are able to get weather forecasts from other sources. For how long would you be willing to wait to get a forecast from the application? Note: the application can run and predict a forecast automatically.

10 svar