



**NTNU – Trondheim**  
Norwegian University of  
Science and Technology

# Application for hydropower on a mobile device with focus on front-end

**Jørgen Aaberg**

Master of Science in Computer Science

Submission date: June 2014

Supervisor: Sobah Petersen, IDI

Norwegian University of Science and Technology  
Department of Computer and Information Science



**Title:** Application for hydropower on a mobile device  
with focus on front-end

**Student:** Jørgen Aaberg

**Problem description:**

Electricity is costly to store, and is therefore produced simultaneously as it is used in Norway. The production is done by hydroelectric power, which can easily be stored in the form of holding back water. Complex, predictive models are needed to utilize these water resources in the best way. Such models and simulation programs are being developed at SINTEF Energy, and consist mostly of Command Line Interfaces (CLIs) on desktop computers. This project should focus on the need for a Graphical User Interface (GUI) for internal visualization of hydropower simulations at SINTEF Energy.

To optimize the power production, SINTEF Energy does complex computations to predict future events. These computations are done in CLIs, and needs expert user experience to operate. This project should address the lack of usability of the models by examining literature, and developing a front-end application for mobile devices. It should also examine the existing Common Information Model (CIM) of the system to make it fit with the application.

In particular the work should:

- Provide a background study on mobile user interfaces, mobile user experiences, CIM, and relevant applications.
- Develop a front-end application to illustrate a potential of visualizing SINTEFs existing applications with a GUI.
- Continuously evaluate the application through several iterations of prototyping, and finally by evaluation methods from the design science framework.

**Supervisor:** Sobah Abbas Petersen, IDI

**External supervisor:** Idar Petersen, SINTEF Energy





## Abstract

SINTEF Energy (hereafter called SINTEF) develops complex computational models for energy production in Norway. These models are controlled by command line interfaces when used for testing, in meetings, or in other contexts. They may, to some extent, be visualized with Python or MATLAB scripts, but lacks a good graphical user interface, as well as including the users in an interactive manner. SINTEF desires to look at the capabilities of moving some of today's CLI-based workflow to a mobile environment, with an interactive user interface. They want to expand their view, exploring new ways of working. They also want to know if their current information models are able to handle these new environments.

We will develop a front-end application for an Android tablet. With an agile prototyping methodology, we identify requirements to make such an application both user-friendly and useful to the end users. The application is primarily a proof-of-concept to illustrate the potential of making a new interface to SINTEFs existing computational models. The development process is also used to see if SINTEFs existing information models are extensive enough to match with the new mobile environments. The application will first and foremost be evaluated by its potential to present a prototype GUI, but also by a set of design principles regarding mobile design, and its strengths and weaknesses.

Our concept illustrates the potential of visualizing parts of SINTEFs existing applications to a mobile device in an interactive manner. The users saw a potential in saving time by using such an application. The application is also used to suggest an extension to SINTEFs existing information model. This proposed CIM extension has been implemented in our application, and is generally viable for other software applications utilizing information model.



## Sammendrag

SINTEF Energi (heretter kalt SINTEF) utvikler komplekse beregningsmodeller for energiproduksjon i Norge. Disse modellene er kontrollert via kommandolinje-baserte grensesnitt (CLI) når de skal testes, vises i møter, eller benyttes i andre sammenhenger. Beregningsmodellene kan, til en viss grad, visualiseres ved hjelp av Python- eller MATLAB-script, men mangler et godt grafisk brukergrensesnitt, i tillegg til å involvere brukerne på en interaktiv måte. SINTEF ønsker å se på mulighetene for å flytte noe av dagens CLI-baserte arbeidsflyt til et mer mobilt miljø, med et interaktivt grafisk brukergrensesnitt. De ønsker også å finne ut om deres nåværende infomasjonsmodeller kan håndtere et nytt mobilt miljø.

Vi vil utvikle en front-end-applikasjon for et nettbrett som kjører Android. Med en agil prototypingsmetode vil vi identifisere krav for å gjøre en slik applikasjon brukervennlig og nyttig for brukerne. Applikasjonen vil først og fremst være en prototype for å illustrere et potensial for å lage nye grensesnitt til SINTEFs beregningsmodeller. Utviklingsprosessen er også brukt til å sjekke om SINTEFs eksisterende informasjonsmodeller er omfattende nok til å håndtere interaktive mobile applikasjoner. Applikasjonen vil først og fremst bli evaluert ut ifra dens potensial til å presentere et grafisk brukergrensesnitt, men også gjennom et sett med designprinsipper for mobilt design, samt dens styrker og svakheter.

Vårt konsept viser et potensiale for å visualisere deler av SINTEFs eksisterende applikasjoner til en mobil enhet på en interaktiv måte. Brukerne så et potensiale for å spare tid ved å bruke en slik applikasjon. Applikasjonen er også benyttet til å foreslå en utvidelse av SINTEFs eksisterende infomasjonsmodell. Den foreslåtte utvidelsen er implementert i vår applikasjon, og vil også passe for annen programvare som benytter denne informasjonsmodellen.



## Preface

This thesis is submitted to the Norwegian University of Science and Technology (NTNU) as the final part of a five-year Msc in Computer Science at the Department of Computer and Information Science (IDI).

The work has been conducted with Sobah Abbas Petersen as supervisor and Idar Petersen as external supervisor. I would like to give a special thanks to Idar for giving me invaluable help the last year in understanding the domain and continuously providing feedback. I would also like to thank my roommate Mr. Walker for giving me much-needed motivation during the final weeks of the project. Finally, thanks to all my friends and family for their support through the last 5 years.



# CONTENTS

<b>List of Figures</b>	<b>xi</b>
<b>List of Tables</b>	<b>xiii</b>
<b>List of Abbreviations</b>	<b>xv</b>
<b>I Project Introduction</b>	<b>1</b>
<b>1 Introduction</b>	<b>3</b>
1.1 Motivation . . . . .	3
1.2 Project domain . . . . .	4
1.3 Project plan . . . . .	5
1.4 Report outline . . . . .	6
1.5 Development tools and technology . . . . .	7
<b>2 Problem Elaboration</b>	<b>11</b>
2.1 User group . . . . .	11
2.2 Scenarios . . . . .	11
2.3 Research questions . . . . .	12
2.4 Contributions . . . . .	13
2.5 Scope and delimitations . . . . .	13
<b>3 Methodology</b>	<b>15</b>
3.1 Research method . . . . .	15
3.1.1 Guidelines . . . . .	16
3.1.2 Evaluation . . . . .	17
3.1.3 Data Acquisition . . . . .	17
3.2 Design and development methodology . . . . .	18

<b>II Background Study and Theory</b>	<b>21</b>
<b>4 Relevant Applications</b>	<b>23</b>
4.1 Oslo Bysykkel . . . . .	24
4.2 Hydrovallées Bigorre . . . . .	25
4.3 Yr.no . . . . .	26
4.4 FINN . . . . .	27
4.5 Kraftsystemet . . . . .	28
4.6 Otra Kraft . . . . .	29
4.7 Summary and discussion . . . . .	30
<b>5 Going mobile</b>	<b>31</b>
5.1 Design principles . . . . .	31
5.2 Android elements . . . . .	32
5.3 Element size . . . . .	33
5.4 Input . . . . .	34
<b>6 Information Modeling</b>	<b>35</b>
6.1 UML . . . . .	35
6.2 CIM UML in hydro power production . . . . .	36
6.3 Existing model made by SINTEF . . . . .	37
6.4 4+1 View Model . . . . .	40
<b>III Application Design and Development</b>	<b>41</b>
<b>7 Application Design</b>	<b>43</b>
7.1 Design risks and challenges . . . . .	43
7.2 Prototyping . . . . .	44
7.2.1 Sketching iteration 1 . . . . .	44
7.2.2 Sketching iteration 2 . . . . .	46
7.2.3 Sketching iteration 3 . . . . .	48
7.2.4 Digital mock-ups iteration 4 . . . . .	50
7.2.5 Digital mock-ups iteration 5 . . . . .	51
7.3 Adapting icons . . . . .	52
<b>8 Application Implementation</b>	<b>55</b>
8.1 Implementational risks and challenges . . . . .	55
8.2 Requirements . . . . .	57
8.3 Implementation . . . . .	61
8.3.1 Implementation iteration 6 . . . . .	61
8.3.2 Implementation iteration 7 . . . . .	62
8.4 4+1 View model of the application . . . . .	63



<b>IV Results</b>	<b>65</b>
<b>9 Results</b>	<b>67</b>
9.1 GUI and functionality . . . . .	67
9.2 Suggested CIM in hydro power production . . . . .	70
<b>10 Evaluation and Conclusion</b>	<b>73</b>
10.1 Methodology . . . . .	73
10.2 Design principles . . . . .	74
10.3 Strengths and weaknesses . . . . .	75
10.4 Answering research questions . . . . .	76
10.5 Reflections . . . . .	77
10.6 Further work . . . . .	78
10.7 Conclusion . . . . .	79
<b>References</b>	<b>81</b>
<b>V Appendices</b>	<b>85</b>
<b>A Summary of the Pre-thesis Project</b>	<b>87</b>
<b>B Hydro API</b>	<b>93</b>
<b>C Installation Guide</b>	<b>95</b>
<b>D Development Environment</b>	<b>97</b>



# LIST OF FIGURES

1.1	Hydropower system . . . . .	4
1.2	Tools used by SINTEF . . . . .	9
3.1	The iterative design and critique process . . . . .	19
4.1	Oslo Bysykkel . . . . .	24
4.2	Hydrovallées Bigorre . . . . .	25
4.3	Yr.no . . . . .	26
4.4	Finn.no . . . . .	27
4.5	Kraftsystemet . . . . .	28
4.6	Otra Kraft . . . . .	29
5.1	Android tab bar . . . . .	32
5.2	Android buttons . . . . .	33
5.3	Android slider . . . . .	33
5.4	Switches . . . . .	33
5.5	Input . . . . .	34
6.1	Association . . . . .	36
6.2	Generalization . . . . .	36
6.3	Aggregation . . . . .	36
6.4	Reservoir Model . . . . .	38
6.5	Aggregated model . . . . .	39
6.6	One-magasin as a diagram . . . . .	39
7.1	Selected sketches from iteration 1 . . . . .	45
7.2	Selected sketches from iteration 2 . . . . .	47
7.3	Selected sketches from iteration 3 . . . . .	49
7.4	Application structure . . . . .	49
7.5	Digital prototype from iteration 4 . . . . .	50
7.6	Digital prototype from iteration 5 . . . . .	52

7.7	Current icons . . . . .	53
7.8	New icons . . . . .	53
7.9	Clustered icons . . . . .	53
8.1	Screenshots from development iteration 1 . . . . .	61
8.2	Screenshots from development iteration 2 . . . . .	62
8.3	Logical view. . . . .	63
8.4	Physical view. . . . .	63
8.5	Development view. . . . .	64
9.1	Resulting GUI 1 . . . . .	68
9.2	Resulting GUI 2 . . . . .	69
9.3	Resulting GUI 3 . . . . .	69
9.4	Resulting CIM . . . . .	70
A.1	Reservoir icons . . . . .	88
A.2	Clustered icons . . . . .	88
A.3	GUI 1 . . . . .	89
A.4	GUI 2 . . . . .	89
A.5	GUI 3 . . . . .	90
A.6	GUI 4 . . . . .	90
A.7	Parallelization result 1 . . . . .	91
A.8	Parallelization result 2 . . . . .	92

# LIST OF TABLES

1.1	Project plan . . . . .	5
3.1	Design science research evaluation methods . . . . .	17
8.1	Design requirements . . . . .	57
8.2	Functional requirements . . . . .	59
8.3	Non-functional requirements . . . . .	60
A.1	Parallelization techniques provided by Android. . . . .	91



# LIST OF ABBREVIATIONS

- ADT** Android Development Tools.
- API** Application Programming Interface.
- CIM** Common Information Model.
- CLI** Command Line Interface.
- EA** Enterprise Architect.
- EMPS** Multi-area Power-market Simulator.
- EMS** Energy Management Systems.
- EOPS** EFI's One-area Power-Market Simulator.
- EPRI** Electric Power Research Institute.
- GUI** Graphical User Interface.
- HTTP** Hyper Text Transfer Protocol.
- IDE** Integrated Development Environment.
- IEC** International Electrotechnical Commission.
- JDK** Java Development Kit.
- JSON** JavaScript Object Notation.
- MIT** Massachusetts Institute of Technology.

**NTNU** Norwegian University of Science and Technology.

**NVE** Norwegian Water Resources and Energy Directorate.

**OMG** Object Management Group.

**PaaS** Platform as a Service.

**REST** Representational state transfer.

**SDK** Software Development Kit.

**SHOP** Short-term Hydro Operating Planning.

**UML** Unified Modeling Language.

**UX** User Experience.



## Part I

# Project Introduction



# INTRODUCTION

This chapter will introduce the motivation of the project and a brief intro to the project domain. This will be followed by a project plan, a report outline, and the tools and technology used in the project.

## 1.1 Motivation

Norway is a major player in hydropower generation and energy transmission in very harsh climate. Norway has 550 power plants, which produce about 110 TWh annually. It is the most important source of energy production in the country.

Hydropower is pollution free, renewable and can be stored easily by holding back water. The hydroelectric systems are ideal to meet variations in power demand by the short response time of hydro turbine regulation. This helps balancing rapid variations in demand and energy prices. Optimizing generation plans and schedules can create a huge gain in handling the changing demand, precipitation, market conditions and energy prices. Although there are lots of benefits with hydropower, there are also lots of challenges. A hydropower system usually consists of several reservoirs, watercourses and power stations that make them very complex. Fortunately, decision support systems help maximizing revenue by calculating inflow, demand and prices on both short and long term. [1]

SINTEF perform modeling and analysis of such energy systems. One of their main areas of focus is hydropower and congestion management in the transmission grid. They offer research based solutions and services. With computer clusters and algorithms they can provide complex computations based on hydropower data. These prediction models are based on CLIs, and need expert user experience to use [2]. They may, to some extent, visualize the results with for example MATLAB or Python scripts, but this is still not regarded as a good solution. A more interactive application with a user-friendly GUI would give a lower threshold of use, a way of

easy demonstrating their models internally, and a way to quickly gain overview over errors and alerts in their models. Also, they are performing a lot of work such as aggregation of reservoirs manually. This is very time consuming.

To conclude, there is a desire for an application to visualize SINTEF's models in a more interactive way, and to automate time-consuming tasks that occurs daily. This may lead to quicker and better user interaction, which in turn would save a lot of time.

## 1.2 Project domain

The domain of the domain is calculations of hydropower systems. As mentioned in the motivation, hydropower systems are reliable, renewable sources of energy. They account for 95.3 % of Norway's total electricity generation, and 16.1 % of the worlds total energy production [3].

The concept is based on simple mechanics. As illustrated in Figure 1.1, water is stored in a reservoir. There is a penstock pipe from the reservoir, guiding the water through a turbine. This turbine makes a generator rotate, transforming the kinetic energy into electricity. The turbine and the generator constitutes to what we know as a hydropower plant in its simplest form.

SINTEFs computational models covers everything from forecasting inflow to the reservoir, water pressure in the penstock, turbine efficiency, to limitations in the long distance power lines.

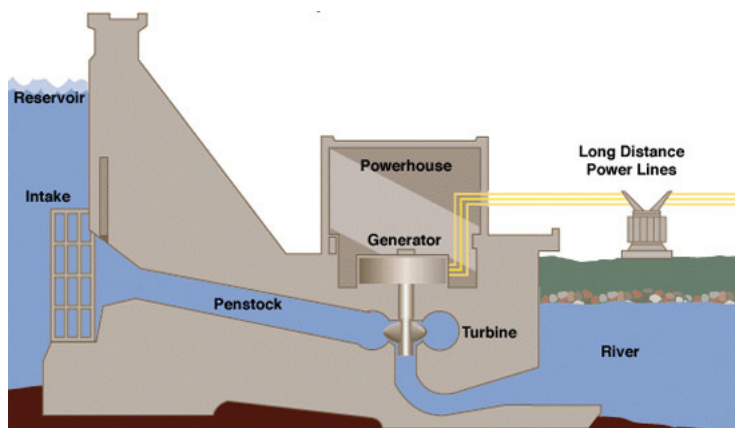


Figure 1.1: Hydropower system<sup>1</sup>

<sup>1</sup>Figure taken from <http://www.alternative-energy-news.info/technology/hydro/>

### 1.3 Project plan

Table 1.1 presents the project plan.

<b>Week</b>	<b>Activity</b>
JANUARY	
4	Introduction
5	Research and problem definition
FEBRUARY	
6	Interview, structure report and project planning
7	Literature review
8	Literature review and requirements
9	Literature review and sketching
MARCH	
10	Literature review and sketching
11	Thesis writing and Sketching
12	Thesis writing and Mock up
13	Thesis writing and Mock up
APRIL	
14	Development
15	Development
16	Thesis writing and Testing
17	Development
MAY	
18	Development
19	Thesis writing
20	Thesis writing
21	Thesis writing
22	Proofreading
JUNE	
23	Delivery

**Table 1.1:** Project plan

## 1.4 Report outline

### Part I: Introduction

- 1- INTRODUCTION presents the motivation for the project, along with a project plan, and the tools and technology used in the project.
- 2- PROBLEM ELABORATION discuss the problem relative to the motivation, and user group. Further, it defines the research questions, contributions, and scope for the project.
- 3- METHODOLOGY introduces the research and development methodology for the project. It also defines guidelines, evaluation methods, and how data will be acquired through the project.

### Part II: Background Study and Theory

- 4- RELEVANT APPLICATIONS describes applications that are relevant in either design or domain. Three of the applications are acquired from a pre-thesis project in which you find a summary of in Appendix A, and three are new ones.
- 5- GOING MOBILE presents theory and literature about moving from desktop applications to mobile applications.
- 6- INFORMATION MODELING describes literature and importance of information models. It presents existing information models that we later will extend.

### Part III: Application Design and Development

- 7- APPLICATION DESIGN defines the risks and challenges of the design. Then it describes and discusses the design process, and feedback acquired from tests and interviews.
- 8- APPLICATION IMPLEMENTATION defines the risks and challenges of the implementation. Then it describes and discuss the implementation process, and feedback acquired from tests and interviews.

### Part IV: Results

- 9- RESULTS present the results of the application, including the application, and the extended information model.
- 10- EVALUATION AND CONCLUSION first evaluates the project based on criteria defined in Part I. It presents suggestions for further work, and ends with a conclusion to the project.

## 1.5 Development tools and technology

This section presents the tools and technology used in development of the application.

### **Android SDK**

Android Software Development Kit (SDK) [4] is the required SDK to develop Android applications. It provides the API libraries a developer need, and is well provided with tutorials to get started. The kit includes an emulator, documentation of the source code, sample code, tutorials, and a debugger.

### **Android ADT**

Google provides Android Development Tools (ADT) [5] bundle for free to developers. The ADT makes it easier to use the functions and tools that the Android SDK provides, by giving access through the user interface of Eclipse

### **Eclipse IDE**

Eclipse is a multi-language Integrated Development Environment (IDE), mostly written in Java [6]. It is open source, and therefore very widespread. It supports Java by default, and may support additional languages by means of various plug-ins. Since Google offers ADT bundle for Eclipse, it is easy to get started.

### **SPARX Enterprise Architect with CIMEA add-in**

Sparx System has developed a comprehensive analysis and design tool called Enterprise Architect (EA), which is used by SINTEF in UML modeling. The CIMEA add-in enables CIM profile creation. The tool enables UML model viewing, the user can browse hundreds of CIM classes, and it is already in use by SINTEF.

### **Adobe Photoshop**

Photoshop is a powerful graphics-editing program developed by Adobe Systems. It is widely used to lots of purposes such as color and exposure adjustments, layers, it also has a suite for video editing. For this project it is used to create mock-ups of an Android application.

### **Git**

The version control system Git were used. Git is a distributed source code management system, and was developed by Linus Torvalds in 2005 [7]. It comes with a lot of nice features like roll backing to previous versions, file history, and possibility to work

on several branches, among others. The project repository was hosted at Github<sup>2</sup>.

## **Java**

Java is a programming language developed by Sun Microsystems in 1995, and is licensed under the GNU General Public License. Java can be run on any Java Virtual Machine (JVM), which means that it can run on any platform. It is an object-oriented language and is based upon classes.

## **Java Development Kit**

Java Development Kit (JDK) is an implementation of Java released by Oracle. It is released under the GNU General Public License (GPL), making it free. It must be used for Android SDK to function as intended.

## **JSON**

JavaScript Object Notation (JSON) is a format used for storing and sending data in a light, human-readable and easily parsable format [8]. It's based on JavaScript notation, but uses only a small subset of the JavaScript syntax; only strings, numbers, lists, objects, booleans and the null value are included. JSON have been used for storing data about each hydropower module.

## **Samsung Galaxy Tab 10.1**

Samsung has released a number of tablets with Android in the Galaxy Tab series. The one used for this project is a Galaxy Tab 10.1, with Android version 4.0.4, Ice Cream Sandwich. A 1GHz dual core application processor powers the device [9].

### **1.5.1 Tools used by SINTEF**

This section will provide information on the tools SINTEF use in their daily work. The tools are modeled in Figure 1.2 and we will briefly explain how they work together. [10, 11]

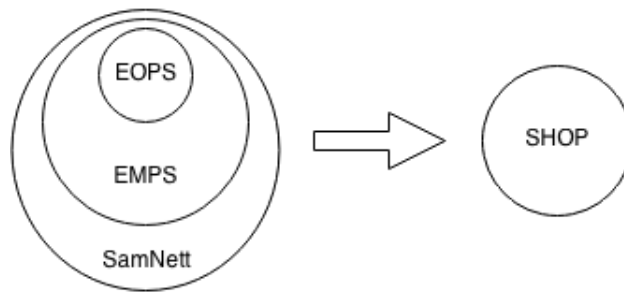
EFI'S ONE-AREA POWER-MARKET SIMULATOR (EOPS) is a program made for long term scheduling and expansion in hydropower data. It is often used by power producers in planning production when there are no major transmission constraints within the specified area.

MULTI-AREA POWER-MARKET SIMULATOR (EMPS) is used for networks of power producers in a larger geographical area. It use instances of EOPS files in addition to files that connect the modules together.

---

<sup>2</sup>See more at: <http://www.github.com>





**Figure 1.2:** Tools used by SINTEF

SAMNETT is a model that integrates energy management and power simulation models with load flow analyses.

SHORT-TERM HYDRO OPERATING PLANNING (SHOP) is a program used by SINTEF to maximize profit, and optimize the use of available resources. It is able to model the most complex configurations of watercourses.



# PROBLEM ELABORATION

This chapter will do a thorough explanation of the problems that will be addressed in this thesis. The motivation for the project is already explained, and we will continue by examining our user group and real scenarios that they encounter in their daily work. Based on these scenarios we present a set of research questions, and define the scope of the thesis.

## 2.1 User group

This project is targeted at SINTEF Energy's researchers. SINTEF is the largest independent research organization in Scandinavia. As they develop technological solutions, they are all pretty technically experienced. The researchers all have a background in science and/or technology, thus making a pretty concrete user group. Nevertheless, they have different interests, works with different domains, and we cannot rely on specific stereotypes other than their background as scientists.

## 2.2 Scenarios

These scenarios are obtained through meetings with both students and researchers working on SINTEF's projects. They explain tasks encountered on a daily basis. They will be used to elaborate the research questions of the project, and as evaluation of the applications necessity.

### S1 – Flooding

A SINTEF researcher wants to get a quick overview over potential flooding risks in a hydropower system i.e. when the predicted water level in a reservoir is higher than the amount of water the dam could hold back. If there are predicted risks of flood, he also wants to know where and when this flooding is predicted. This is necessary to take precautions in order to minimize the costs of flooding. Flooding is often the

highest source of costs in SINTEF's models, and therefore needs to be carefully taken care of.

### **S2 – Water values**

A SINTEF employee is doing calculations on water values. His goal is to optimize the costs and production by finding errors in the existing system. As the value of the water depends on many different factors, this is complicated work. It requires many look-ups in different tables, several steps ahead in time. Since this is done in CLIs, this takes a huge amount of time.

### **S3 – Pressure**

A SINTEF employee does calculations on the efficiency of turbines in a hydropower system. He does calculations on pressure in penstocks to make the turbines as efficient as possible. There are a lot of factors that needs to be taken into consideration, like water drop and amount of water. As in the previous scenario, this requires many table look-ups, which in turn takes a lot of time.

### **S4 – Turbines**

A person taking care of maintenance in hydropower systems want to know which turbines are in movement and which are not. If a turbine is not moving, it may be broken in some way, and need to be fixed. It may also be that someone have miscalculated the amount of running water, which make the turbine run in a sub-optimal manner. A still-standing turbine is a potential loss in production.

### **S5 – Understanding results**

A student is building a model to solve the problem of optimal allocation of a hydropower system in a long term. A lot of results can be obtained from such a model. This can for instance be how much water is in a reservoir at the end of a week. This will be used by another program to find the optimal allocation of water within a week. This will provide a basis for real-time access to these sorts of information to users so that they could conduct real-time decision-making and influencing of the power consumption. The results, though, is only textual and hard to understand for other people than the person building the model. [12]

## **2.3 Research questions**

The motivation and obtained scenarios elaborated in the previous section provide the basis for questions to be addressed in this thesis. The following questions will be the research questions for this project.

1. How can we adapt SINTEFs existing CLI-based systems onto mobile devices?
  - a) How can users get an overview of the entire hydropower system, whilst giving the depth down to each individual reservoir?
  - b) How can we improve the efficiency in finding errors and alerts in SINTEFs models?
  - c) In what way could we adapt today's watercourse maps to mobile devices with touch screens?
  - d) How can we display results from hydropower calculations on mobile devices?
2. How can the CIM be improved to support aggregation of reservoirs and hydropower stations?

## 2.4 Contributions

This section will serve the purpose of clarifying what elements of this project that are considered as contributions.

We will develop an application for an Android tablet that will serve the purpose of solving the problems described in the motivation and the scenarios. The resulting application, with its design, icons and functionality, is considered a contribution supported with theory in this thesis. The application illustrates an existing problem, and a possible solution that may apply to several of the obtained scenarios. Even though the application is considered a proof-of-concept, it is still developed according to SINTEFs existing information model, and connecting it to the back-end is possible.

A second contribution is an extended version of SINTEFs existing CIM. It is considered a necessary extension, not only to fit this application, but also other software utilizing CIM.

The resulting application and CIM are presented in Chapter 9 Results.

## 2.5 Scope and delimitations

There are few people at SINTEF that have expertise of the existing CLI systems. They are very busy with their own projects and deadlines, working for SINTEF and/or Norwegian University of Science and Technology (NTNU). For this reason the project has been limited to getting feedback from three potential users, presented Section 3.1.3.

The application had to be developed on an Android device, provided by NTNU. The device was a Samsung Galaxy Tab 10.1, supporting Android 4.0.4 Ice Cream Sandwich, which is not the newest firmware.

The application is first and foremost a proof-of-concept. It is developed to apply to some of the problems the test-users encounters in their daily work. We have elaborated scenarios based on meetings with the potential users that address certain problems. We will focus more closely on one of the scenarios for the implementation, but make a concept that will be easily transferable to the other scenarios as well.

# METHODOLOGY

This project has been conducted with the design science research methodology, which has contributed great in structuring the work. As for design and development, we have used an agile prototyping method. Both design science research and prototyping have contributed in shaping the final results, and they are further described in this chapter.

## 3.1 Research method

We separate between routine design and innovative design. Routine design is to apply existing knowledge from the knowledge base to organizational problems in order to create artifacts. Innovative design solves new problems in unique ways, or solved problems in more effective and efficient ways [13, 14]. Design science research, our chosen research methodology, is considered innovative design. It identifies a contribution to the existing knowledge base, and attempts to create artifacts to serve human purposes.

It fits with our project, as we will create an application that attempt to improve SINTEFs existing workflow. March and Smith [15] defines progress in design science as “when existing technologies are replaced by more effective ones”, which is exactly what we are trying to do by moving from a CLI environment, to a mobile one. In addition we contribute to the existing knowledge base by working on SINTEFs information models.

There are a lot of interpretations of design science research, and many ways of conducting it have been proposed [13, 15, 16]. We have chosen the framework suggested by Hevner et al. [13] This is a concise conceptual framework, providing clear guidelines to understanding, executing, and evaluating the research. These guidelines and evaluation methods will be related to our project in the two next sub-sections.

### 3.1.1 Guidelines

The design science framework suggested by Hevner et al. offers 7 guidelines. These guidelines will be related to this project, and is used to describe the research process.

1. *Design as an artifact: The research must provide a viable artifact.*
  - The outcome of the project includes an Android application providing SINTEF an interactive user interface for their models.
2. *Problem relevance: The objective is to develop solutions to business relevant problems.*
  - There are problems in understanding computational models and results internally on SINTEF in a fast way. This may be very time consuming, which illustrates the problem relevance.
3. *Design evaluation: The artifact must be rigorously demonstrated via good evaluation methods.*
  - The design will be evaluated by simulations, black box testing, informed argument and scenarios. The application will also continuously be evaluated through design iterations with potential users.
4. *Research contribution: Effective design-science research must provide clear and verifiable contributions in the areas of the designed artifact.*
  - The contribution will be a prototype application with a sensible user interface in which illustrates how an application may be used to communicate computational models and results internally at SINTEF. There will also be a contribution to the existing information model that is useful for any application utilizing this model.
5. *Research rigor: It should rely upon the application of rigorous methods in both the construction and evaluation of the designed artifact.*
  - We will find relevant applications and research made by others to learn and build on their work. The application will also be constructed based on iterative interviews and testing with potential users.
6. *Design as a search process:*
  - An iterative prototyping process will be used to develop requirements and design that will be further described in Section 3.2.
7. *Communication of research: Research must be presented efficiently both to technology-oriented as well as management-oriented audiences.*
  - This thesis, in combination with the application code, is used to communicate the work and thoughts to both technological-oriented as well as management-oriented audiences.



### 3.1.2 Evaluation

Evaluating a software product can be done in a great number of ways, such as performance testing, security testing, usability testing, integration testing and so forth [17]. But when it comes to evaluating a research prototype, one must focus on the forms of evaluation that provides feedback on the questions that the researchers are set out to investigate.

The design science framework suggested by Hevner et al. [13] provides several research evaluation methods in which will be used in Section 10.1 to evaluate the research method. The methods that will be used are listed in Table 3.1.

Descriptive	
Informed argument	Using information from the knowledge base (i.e. relevant research) to argue for the applications usefulness, and justify choices made in design and development.
Scenarios	Constructing real scenarios that the users encounter to demonstrate its utility.
Experimental	
Simulation	Using artificial data in order to make the application ready for testing, even though the real data may be missing. This enables testing in an early phase.
Testing	
Black-box testing	Control if the system holds the required abilities as proposed in the requirements.

**Table 3.1:** Design science research evaluation methods

### 3.1.3 Data Acquisition

This project is based on qualitative data gathering. There are few available resources at SINTEF, which means quantitative feedback wasn't really an option [18].

We have purposefully selected participants and sites that will help us understand the problem domain. The persons that have provided us information both through related documents and interviews are:

1. Idar Petersen – Idar is a research scientist at SINTEF Energy, and also the co-supervisor of this thesis. He works with electric power systems. His re-

search activities focuses on transmission expansion planning for integration of renewables with focus on grid connection of renewable power.

2. Frederic Dorn – Frederic is also a researcher at SINTEF Energy. He works with energy systems. His research activities focus on optimization in production planning for short-term hydro-power scheduling.
3. Hallvard Braaten – Hallvard is a student at Applied Physics and Mathematics, NTNU. He did a project developing a model for parallel execution on a cluster for SINTEF. He does not directly match the targeted user group of SINTEFs researchers, as he is a student. Nevertheless, he is doing research on a project for SINTEF, and is regarded as a potential user as well.

There are not much written material addressing the same problems and domain as we do. Thus, we need to listen to our chosen participants, and build an artifact based on their feedback. By performing semi-structured interviews, we had control over the line of questioning. Creswell and Clark argue that this may lead to biased responses and that there are room for filtering by our views [18], but their information and feedback was still invaluable in shaping the project.

They also provided important documents, which enabled us to obtain domain language that gave us a shared context of understanding.

### 3.2 Design and development methodology

This project has been conducted with no strict development methodology. Though, there has been an agile prototyping pattern through the whole process. As stated in [19], prototyping is “especially useful for resolving unclear objectives; developing and validating user requirements; experimenting with or comparing various design solutions; or investigating both performance and the human computer interface.” It also “provides a quick implementation of an incomplete, but functional, application”. Since we were developing an application in a new domain with few defined requirements, this was a good method to follow.

The prototyping process started with sketching on paper. Several ideas were explored and then presented to Idar Petersen, and he provided feedback. Together, we decided to reject some ideas, and worked further with others. This presentation and feedback were done in several iterations described in more detail in Section 7.2, and illustrated in Figure 3.1. Idar gave feedback once a week for three weeks, until we had decided on the basic functionality and design.

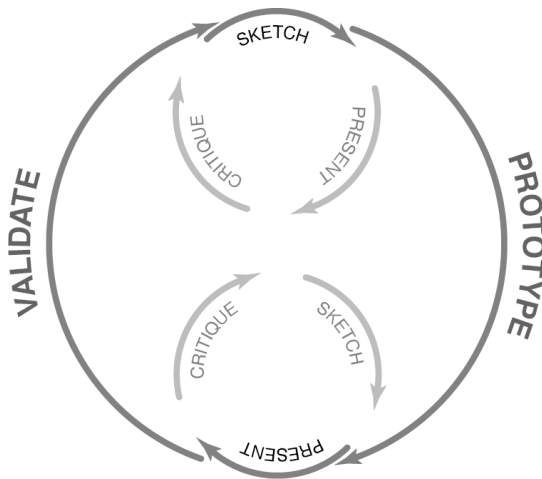
Further, we decided to make a digital mock-up, since this gives a better feel of the design and interaction than paper sketches does. Several prototyping tools were

explored, such as Fluid UI<sup>1</sup>, WireframeSketcher<sup>2</sup>, and Adobe Photoshop<sup>3</sup>. Photoshop was the only tool that had adequate options for customizing to the depth we needed, and was therefore chosen. It had a steeper learning curve than the other alternatives, but was still preferable.

We did two weeks of creating mock-ups, and followed the same cycle as in the sketching. After two meetings of feedback with Idar, we were ready to implement the application.

The implementation was done in two iterations as described in Section 8.3. For the implementation, we got feedback from both Idar Petersen and Frederic Dorn. This was very valuable in getting new perspectives, to learn, and to prioritize what to be implemented next.

This agile prototyping approach to the development directly addressed the problems of rapid change [20], which was needed for this project.



**Figure 3.1:** The iterative design and critique process (Figure taken from [21])

<sup>1</sup>Fluid UI – <https://www.fluidui.com/>

<sup>2</sup>WireframeSketcher – <http://wireframesketcher.com/>

<sup>3</sup>Adobe Photoshop – <http://www.adobe.com/products/photoshop.html>



## Part II

# Background Study and Theory



## RELEVANT APPLICATIONS

This chapter will present mobile applications that are relevant to the project in either design or domain. The first four applications all have in common that they use a map to provide information. Some of them even encounter the same problem we are, displaying lots of information in small geographical areas. The last two applications are related in domain.

Some of the work reviewing applications was done in the pre-thesis project. Since this project deals with similar domain and issues, we chose to include reviews of these applications in addition to the new ones. The applications that were reviewed in the pre-thesis project are Oslo Bysykkel, FINN, and Kraftsystemet. They are slightly expanded, but most of the work are similar.

These analyses are not meant to be grounded in any specific theory or literature. They are used to find common elements in different applications that we may use as inspiration in the design of our application.

## 4.1 Oslo Bysykkel

Oslo, like many other big cities, offers city bikes. These bikes are placed in stands all over the city. Everyone can pay for a yearly membership. You borrow a bike from one stand, and place it in any of the stands when finished. Oslo Bysykkel (Figure 4.1) is an application for iPhone that pinpoints these stands on a map. It is a really simple application with only the basic information presented in a visual way.

The markers on the map point to an exact location, and makes no doubt about where the stand is located. Each marker is labeled with a number that tells how many bikes are available at the stand in real-time. This gives the user a quick overview over available bikes nearby. If a stand is empty, the marker also turns red, leaving no doubt that there is no bikes there at them moment. When a user presses a marker, he is provided with additional information, such as an address, and available locks. Because you need available locks to deliver the bike back, this is an essential feature.

The combination of quick and easy access to just enough information in addition to a simple design made the application popular.

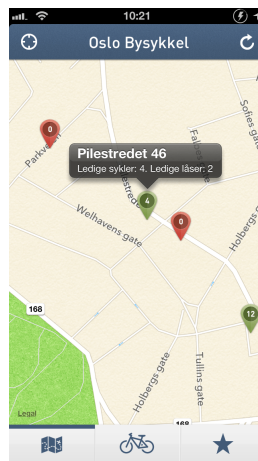


Figure 4.1: Oslo Bysykkel, an application for iPhone

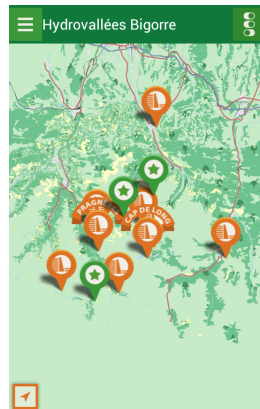


## 4.2 Hydrovallées Bigorre

Hydrovallées Bigorre is an application involving hydroelectric resources and centers of tourist interest in the Pyrenees. They want to target visitors, providing them information about their hydroelectric systems, historical information and heritage [22].

What is interesting about this application is that they are providing a map with markers that mark different reservoirs on a map. This is similar to what we will do in our application. Even though they are aiming at another user group, some of the concepts are still similar, such as pinpointing a reservoir to a map, and providing information about that reservoir.

Similar to Oslo Bysykkel, they use teardrop-shaped markers to point out an exact location, making no doubt were it is located. Each marker has an icon inside, making us easily understand if it is a reservoir, or something else. When the user clicks on a marker, a pop up box is shown with additional information.



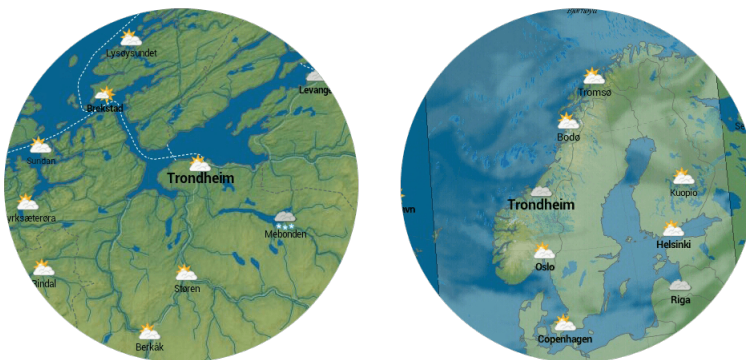
**Figure 4.2:** Hydrovallées Bigorre, an application for Android phones

### 4.3 Yr.no

Yr.no is a weather forecasting application for Android devices with weather information from over 700 000 locations in Norway. It's mainly focused on Norway, but also provides forecasting of locations outside Norway.

Yr.no has a map-view in which the user can view the forecasted weather on a map. Since they have information about many locations, it would be hard to visualize them all at once on a map. What Yr.no does, is to present only the largest cities on the map based on your zoom level. E.g. if you zoom out and have Norway within the device screen, only the largest cities like Oslo and Trondheim will be displayed. If you zoom closer to only Trondheim, the largest urban areas within Trondheim are also displayed. This is illustrated in Figure 4.3.

It relates to our application by having lots of information within small geographical areas. They choose to select the largest locations, since it will relate to more people.



**Figure 4.3:** Cropped screenshots from the Android application yr.no with different zoom levels

## 4.4 FINN

FINN.no is Norway's biggest online market place. They have specialized in advertisement for the private market. On an average day, over 300 000 ads are available at FINN.no. They have made an application for Android tablets called FINN. You can either distribute your own ads, or search the hundreds of thousands of ads available.

What is interesting regarding this application is that you can search, and see where the ads product is located. As you can see in Figure 4.4 they have chosen to split the screen with a search panel on the left side, and a map on the right side. If the user finds an ad that matches his search criteria, he simply presses the marker to see the ad.

A problem occurs when there are multiple ads on the same location. E.g. if there are 3 apartments for sale in the same house. FINN solves this problem by aggregating the closest markers together. When the user presses such an aggregated marker, he gets presented all the apartments from that location.

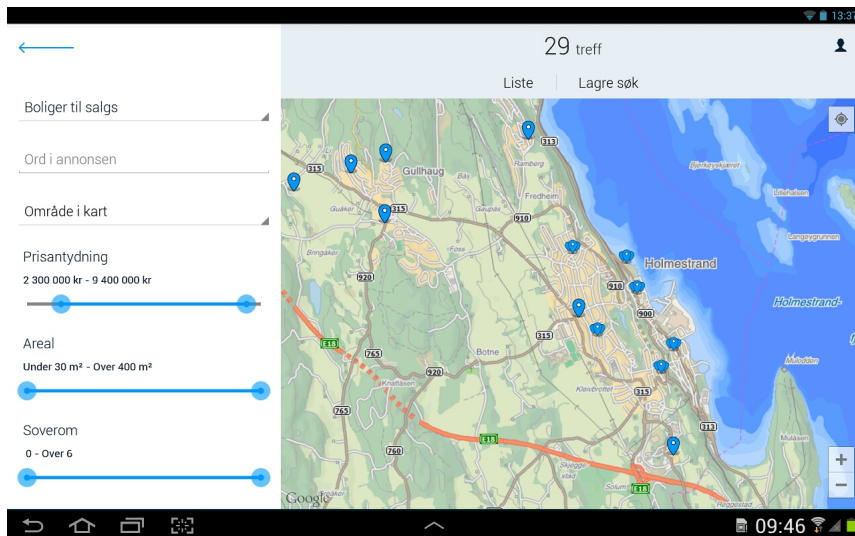


Figure 4.4: FINN, an application for Android tablets

## 4.5 Kraftsystemet

Statnett is the system operator in the Norwegian energy system, and responsible for maintaining balance in the power system. They released an application for Android tablets 30 October 2013 (during the pre-thesis project), to monitor Norway's power system.

The application includes an overview over the Nordic power balance, hydrological data, production and usage, the power situation and import and export. It is related to this project, especially the hydrological data and the map providing information on the power supply in the different zones of Norway. They provide statistical data on the water levels during the year, with data all the way back to 1993.

The big difference from this projects application is the level of detail, and the target audience. Kraftverket tries to provide statistical information about Norway's overall power balance to the Norwegian people. This means that it has to be presented on a high level for more people to understand. Our application is targeted against SINTEF internally, and is on a much deeper level. Our application will provide information all the way down to a single reservoir, to be able to conduct real-time decision-making or future estimates.



**Figure 4.5:** Kraftsystemet, an application to monitor Norway's power system. This screenshot is a chart over the water level over the course of the year

## 4.6 Otra Kraft

Otra Kraft is a Norwegian energy production company that mainly produces electricity. The purpose of the company is to exploit hydropower resources in Otra and Valle in the best way possible. They also have the responsibility for emergency response to accidents and flooding. [23]

Otra kraft has developed an application with the fitting name, Otra Kraft. Its purpose is to provide information about their watercourses. The user can read about different locations, get news from NVE, and see the latest measured water levels from their reservoirs.

As illustrated in Figure 4.6, they have chosen to present water levels as a list. They also list the minimum required water level for each location. This gives the user a quick overview over all the water levels to compare, and immediately notice if there is less water than required at a location. But this does not give any additional information, such as location, inflow, and how these water sources are connected.

Vassføring m3/s	Krav. til min. v.føring m3/s	Avlest 09.09.2013 kl:
<b>Børtemannsbekken</b>		
4,4	4,0	16:00
<b>Heisel</b>		
101,4	0,0	17:00
<b>Hoslemo</b>		
5,0	4,0	16:00
<b>Lislevatn</b>		
3,4	2,0	16:00
<b>Syrtveit</b>		
94,3	15,0	16:00
<b>Valle</b>		
6,5	3,0	16:00

Figure 4.6: Otra Kraft, an application for Android phones

## 4.7 Summary and discussion

As mentioned early in the chapter, we will use these applications as an inspiration in designing our application. We will now draw some conclusions based on our reviews.

To pinpoint something on a map, applications usually use markers that clearly points out the exact location of the marker. Further, the markers provide some basic information just by its looks, such as colors, numbers, or icons inside the marker. This confirms our conclusions from the pre-thesis project. The markers also provide several levels of information. A click on the marker will provide an information window with more information, and a click on the information window will in many cases provide even more information, e.g. in a pop-up window.

FINN and Yr.no encounters the same problem as we do, namely to present a large amount of information on a small geographical area. FINN solves it by aggregating markers close to each other, and change icon to a multiple-marker-icon. Yr.no solves it by only presenting the most significant locations based on the zoom level. These are two good solutions in solving a similar problem. The FINN-way probably fits our application best, as we cannot afford to lose any information in the aggregation as Yr.no does when it just skips the smaller places.

Kraftsystemet and Otra Kraft are both in the same domain as our application, but targeted at another user group. They are presenting information on a much higher level, but it is useful to see how they choose to present data as charts, lists, and maps.

## GOING MOBILE

Porting what worked for your desktop application to a mobile surface does not always make sense. We have constraints and capabilities on mobile devices that are completely different from desktop computers. Smaller screens force us to prioritize what really matters. Poor connections and performance force us to optimize the loading and processing of data. And new possibilities such as touch screens, GPS, and camera force us to think different. We need to align these constraints and capabilities with our users needs. [24]

To do so, we will in this chapter present a set of mobile design principles. We will further present some standard Android elements, made especially for interactive interfaces. Finally, we summarize literature regarding icon sizes and input methods in which differs a lot from desktop environments.

### 5.1 Design principles

As developers often want to distribute their applications on multiple platforms, it often leads to a “design once, ship anywhere”-approach [25]. But design decisions that makes totally sense on one device, does not necessarily make sense on another. In order to help programmers design better applications for the Android platform, Google have listed a set of design principles. As they state [26]:

“These design principles were developed by and for the Android User Experience Team to keep users’ best interests in mind. Consider them as you apply your own creativity and design thinking. Deviate with purpose.”

Many of these principles are guidelines to help the developers make more attractive and better applications, and some of them are rules, meant to follow without deviation. The most central ones to be followed in this project are:

**KEEP IT BRIEF.** The application must use short and useful information. People are likely to skip sections if they are too long, or if the information seems irrelevant.

**ONLY SHOW WHAT I NEED WHEN I NEED IT.** It is important to present information in digestible chunks. Therefore the application will present the most important information first. The user should be able to find more extensive information if needed.

**I SHOULD ALWAYS KNOW WHERE I AM.** The menus shall leave the user no doubt where he is, at any time. The user shall gain confidence in using the application, not feel insecure.

**MAKE IMPORTANT THINGS FAST.** The most important information and functions should be easy to find.

Furthermore, it will be strived to follow Jacob Niensens 10 Usability Heuristics for User Interface Design [27] with the most central one being:

**MATCH BETWEEN SYSTEM AND THE REAL WORLD.** The application should use the same language as the users. That is, words, phrases, and concepts should follow the existing conventions from [28] to make it familiar to the users.

## 5.2 Android elements

We will follow the Pure Android pattern [25], which means that we will use standard Android elements and only deviate if we find it necessary. This section introduces relevant Android elements<sup>1</sup> that we will face in developing our application.

**TABS** – Tabs in the action bar in an easy way to switch between different views and functionality. The user navigates between tabs by touching the preferred tab. The tab bar is placed at the top of the screen by default in Android, and the selected tab contains a thick line at the bottom.



**Figure 5.1:** Android tab bar

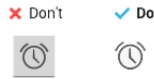
**BUTTONS** – Buttons come in many variations, and they may consist of both text and/or an icon. It is usually considered unnecessary to use a background on buttons with icon. An icon alone works if the action can be well understood using only the

---

<sup>1</sup>All the element figures in this section are taken from <http://developer.android.com/design/building-blocks/>



icon. Text alone is appropriate if the action is hard to visualize by an icon. Using both text and icon is considered appropriate in dubious cases.



**Figure 5.2:** Android buttons

SLIDERS – Sliders make it possible to select a value from continuous discrete ranges. The smallest value is placed to the left, and the largest to the right.



**Figure 5.3:** Android slider

SWITCHES – We have three kinds of switches; checkboxes that allows the user to select multiple items from a set; radiobuttons that let the user choose one item from a set; and on/off switches representing states.



**Figure 5.4:** From the left: Checkboxes, Radio buttons, On/Off switches

### 5.3 Element size

It is not uncommon to see web designers respond to smaller screens by shrinking the interface in order to make it fit. Many mobile applications are designed as if they were just shrunken desktop applications, but we are not just shrinking sizes, we create a totally new experience [29]. When we are using a touch screen with our fingers, we lack the pixel-level accuracy of a mouse pointer. Our fingers are imprecise pointing instruments. They slip around and come in different sizes and shapes. An empirical study made by Kaplan found that nearly half of mobile application users are more likely to tap an advertisement by mistake than intentionally [30]. To prevent touch by mistake, there have been made guidelines recommending the size of objects on touch screens. [24, pp.67–84]

Massachusetts Institute of Technology (MIT)’s Touch Lab determined that average needed size of objects touched by fingertips should be 8-10mm [31].

Microsoft recommends 9mm touch targets with a minimum size of 7mm, and a minimum spacing of 2mm in their Windows 7 Guidelines [32].

Apple’s iOS Human Interface Guidelines recommends 44\*44 points<sup>2</sup>, which was about 6.8mm on the original iPhone [33]. Though, this has been concluded slightly too small with field experiments presented by Schildbach and Rukzio [34]. It showed that this size had an error rate of up to 23% (targets of 9.5 mm had a much lower error rate).

Also, Parhi et al. [35] concludes that a target size of 9.2mm for discrete tasks and 9.6mm for serial tasks should be large enough for a thumb without degrading performance and preference.

The guidelines and studies all conclude slightly differently, but there seems to be a consensus around 8-10mm depending on which finger you are using, and what state you are in. This also includes the spacing between the targets.

## 5.4 Input

Some may argue that the user input should be kept minimal on mobile devices, because writing on small screens is a hassle. But there are reasons to let people contribute through mobile devices as well. After all, over four billion text messages were sent per day in the US alone in 2010 [24, p.86]. We just have to make the input forms easy to use.

Most forms made for desktop computers adapt the real world paper forms with labels on the left side of the input field. This works well on big screens. But in most cases, there is not room for left-aligned labels on a mobile display. Also, when an input field is focused on mobile devices, they are usually zoomed in so that the input field covers the whole width of the screen. This means that there is not room for a column with labels on the left side. To handle this, mobile forms often use top-aligned labels. This ensures that the user will see the label while typing in the text field. Both left aligned and top aligned labels are illustrated in Figure 5.5. [24, pp.85–107]



**Figure 5.5:** The two phones to the left illustrate input fields with left aligned labels. The two phones to the right illustrate input fields with top aligned labels.

<sup>2</sup>Apple points, not to be confused with the typographers’ 1/72-inch point.

# INFORMATION MODELING

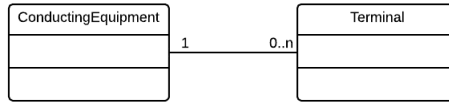
Information models are used in software engineering as a representation of concepts, rules and relationships among others. They are used to create a common shared context, and to prevent ambiguity. [36, 37] In this project, we make a front-end application for SINTEF. Although we will not connect with the back-end in this project, we will explore if their existing information models may comply with our interactive application. In this chapter we first describe Unified Modeling Language (UML), followed by CIM, an information model based on UML made for the energy industry. Finally, we present the 4+1 view model in which will be used to model the implementation from different perspectives.

## 6.1 UML

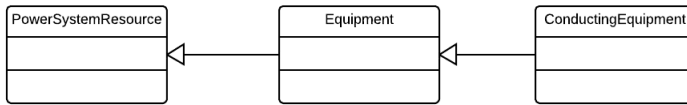
UML is a standardized modeling language, consisting of various diagrams, elements and notations. It was originally developed to design object-oriented software by Object Management Group (OMG), but has greatly expanded since then. Since UML is object-oriented, it enables reusability among components. The components of UML are seen as classes, and the classes may have different relations between them. The classes can also be grouped in different packages by functionality, similarity or such. In order to get an understanding of UML, we will present some relations that exists between classes. [36, 37]

### **Association**

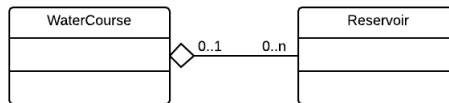
An association is a conceptual connection between classes. It is used to illustrate relationship between classes on a very basic level. It may also have multiplicity in both directions.

**Figure 6.1:** Association**Generalization**

Generalization defines a hierarchy of classes. A sub-class inherits all the attributes contained in its parent, and can also have additional attributes on its own.

**Figure 6.2:** Generalization**Aggregation**

The aggregation relationship indicates that one class is a container for the other.

**Figure 6.3:** Aggregation**6.2 CIM UML in hydro power production**

There has been an increasing need for the exchange of information between stakeholders in the energy industry lately. Since companies have been using several different formats of data storage, there has been lots of extra work in translating information on exchange between different companies. This is why the energy industry has created their own template for information modeling, CIM. [10, 38]

CIM defines an industry standard database and interface so that developers can develop CIM-compliant applications without the hassle of translating into proprietary interfaces [39]. CIM is built upon UML to take advantage of its reusability and object-oriented notions. Real components in the power system are represented by classes. Every class contains attributes, which describe parameters related to the

component. It was first developed by Electric Power Research Institute (EPRI), and is managed by Energy Management Systems (EMS).

CIM consists of two standards defined by the International Electrotechnical Commission (IEC). The first one, IEC 61970, describes the components of a power system and the relationships between the components. The other standard, IEC 61968, extends the first model by taking other aspects of power software data exchange into consideration, such as customer billing and work scheduling. [36, 37, 40]

Statnett, a major player in Norwegian power production, is moving their systems towards the CIM. They held a presentation in 2012 mentioning some of the benefits they identified in doing so. These benefits are quoted below:

- *Improved data consistency and quality for operation, planning and protection.*
- *A common topology enables efficient exchange of model case data between different tools.*
- *A common power system modeling practice within the company.*
- *Make it possible for regional utilities to use different analysis software than Statnett.*

[41]

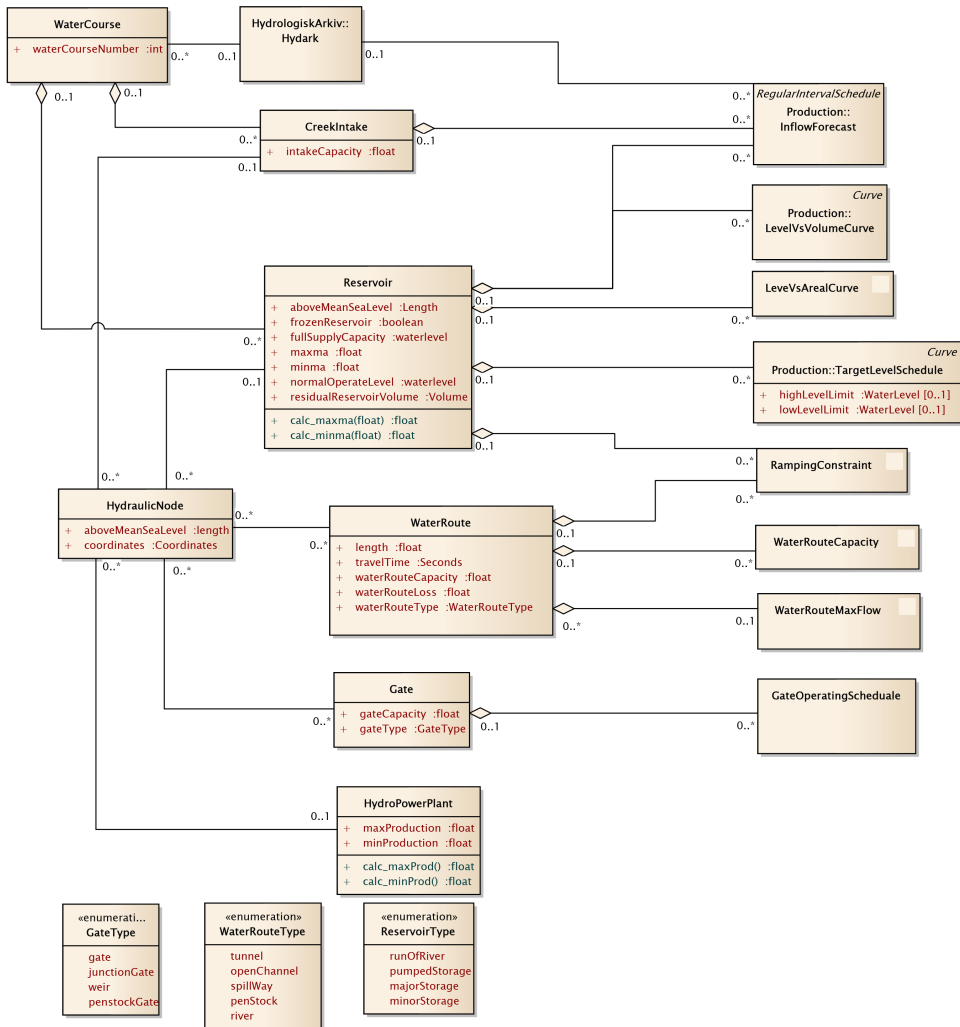
### 6.3 Existing model made by SINTEF

SINTEF is now trying to take advantage of the benefits a common information model might provide. But CIM is still young with respect to modeling capability and documentation [41]. Researchers at SINTEF also argue that the parts of CIM including hydropower is underdeveloped, and that there is a need for further improvements [10].

The following two subsections present subsets of SINTEF's CIM that is relevant for this project.

#### 6.3.1 The Reservoir Model

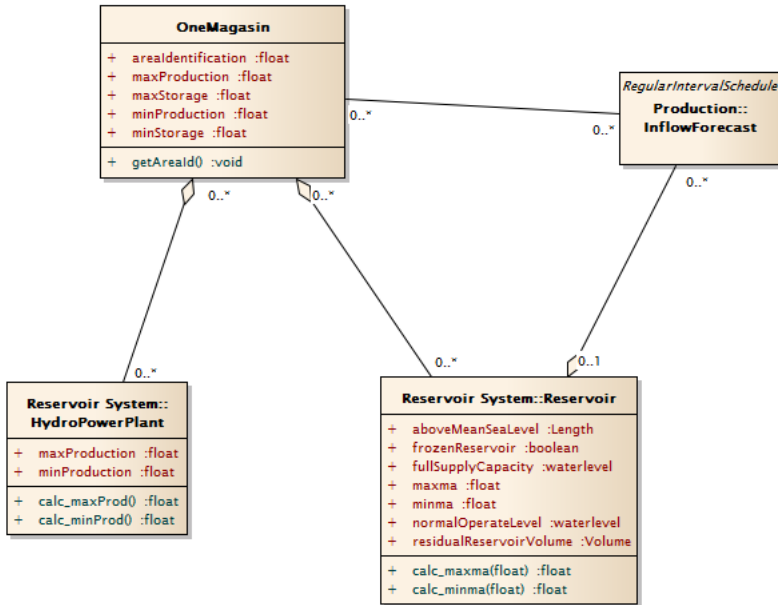
Figure 6.4 illustrates a model for a reservoir system. It shows an overview of how a reservoir system could be modeled with CIM.



**Figure 6.4:** SINTEFs existing Reservoir model  
(Figure in courtesy SINTEF Energy As, Norway)

### 6.3.2 The Aggregation Model

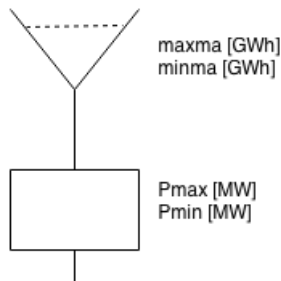
Figure 6.5 is a model of how reservoirs and hydro power plants could be aggregated to a one-magasin. Appropriate methods for calculating production from the hydro power plants and storage from the reservoirs are included. It is a simplified model made by SINTEF to check whether an aggregation is feasible or not.



**Figure 6.5:** Sintef’s existing aggregated model  
 (Figure in courtesy SINTEF Energy As, Norway)

Figure 6.6 is a representation of the same one-magasin, but now as a diagram. It is considered as one system, with the triangle illustrating a reservoir, and the square representing a hydro power plant. The parameters *maxma* and *minma* represents the maximum and minimum energy based on the amount of water. *Pmax* and *Pmin* represent the maximum and minimum production that a plant can produce.

Further work has been carried out on the existing one-magasin model, and the corresponding calculations. The results are presented in Section 9.2.



**Figure 6.6:** One-magasin as a diagram

## 6.4 4+1 View Model

Kruchten has defined something called the 4+1 view model [42, 43]. It is a model to describe a software architecture based on the use of multiple concurrent views, to address different stakeholders concerns. It consists of four views, the logical view, the development view, the process view, and the physical view. In addition, it has scenarios or use-cases that are regarded as the *plus-on view*, used to consolidate the other views. They are all based on UML notation.

**LOGICAL VIEW** is often represented by one or more UML sequence diagrams presenting internal logic.

**DEVELOPMENT VIEW** illustrates the application from a programmer's perspective. It is commonly modeled as a UML package diagram with its appropriate classes.

**PROCESS VIEW** lets us describe and study how the system's processes communicate. It may be useful in multiple, simultaneous processes, and it may be used in several different levels of abstraction.

**PHYSICAL VIEW** is used to illustrate how the application executes in a network of computers.



## Part III

# Application Design and Development



# APPLICATION DESIGN

This chapter will introduce the risks and challenges of designing the application. Then it presents the five iterations of making sketches and digital mock-ups for the application, with discussion and feedback for every iteration.

## 7.1 Design risks and challenges

This section will address the most important risks and challenges in the design process.

### **User experience**

Complicated user interfaces may lead to frustration and confusion, which in turn is time consuming [44]. We want to make the users more effective in their daily tasks. Should the application fail to engage the users, the possibilities of them turning back to the old systems are high. This is a challenge that may face any application, but it is exceedingly important for our application.

### **Communication**

The users have greater domain knowledge than us. This may make the scope wider than necessary. Time will be spent on getting a common shared context, which in turn may slow down prototyping process.

### **Selection of test users**

SINTEF has a strict time accounting system. Everyone is assigned to projects, which makes it hard to reach out to potential users. As stated in Section 3.1.3, we have purposefully selected our participants, but other participants may still have influenced the design differently.

## Iterations

How many iterations of sketching are appropriate? When should we start making a digital mock-up? When is the design ready for implementation?

It is a challenge to decide how many iterations we need to make satisfying sketches and mock-ups. We do not have the time to make pixel-perfect mock-ups, nor is it necessary. But when do we move on to the next design phase? Changing design is more time-consuming in digital design than paper sketching. Likewise, changing design is more time-consuming in implementation than in digital design. To choose exactly when to move from sketching to digital design, and when to move from digital design to implementation may be a challenge.

## 7.2 Prototyping

This section will present the prototyping process, as described in theory in Section 3.2. This includes sketching prototypes, and making a digital mock-up. We did three iterations of sketching, and two iterations of digital mock-ups. We will describe the work done during each iteration, the given feedback from the users, and supply with selected figures from the process.

### 7.2.1 Sketching iteration 1

After meetings with Idar Petersen and Frederic Dorn, we made different scenarios presented in Section 2.2. The first iteration was all about exploring different functionality based on these scenarios. Figure 7.1 illustrates some of the ideas, such as pressure in creek intakes, water level information, and flooding alerts.

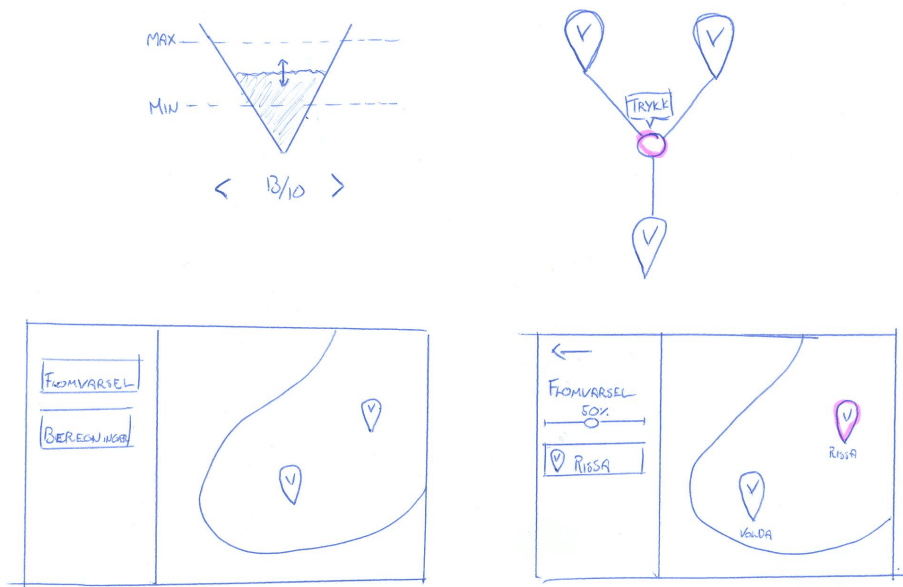
We did most work on the flooding scenario (S1), and the two drawings at the bottom are attempts to cover that. We did it because flooding is often considered the highest cost factor in SINTEF's computational models. Our thoughts were that the users could set a certain water-level threshold, and that they should be notified about all the reservoirs that have a water level higher than this threshold. We used a slider for the user to set this threshold. Every reservoir that is crossing it will turn red, and they would also appear in a list beneath the slider.

With this concept, we did not only address the flooding scenario. This concept could also be transferable to multiple other scenarios, as the issue is similar; getting alerted when something is wrong.

**Feedback:**

Idar thought flooding alerts were a good scenario to cover. He also agreed that this was a concept that later could apply to the other scenarios as well.

What was missing was a perspective of time. The researchers deal with either long-term or short-term calculations, but they are always working with time steps back and forth in time. Another issue was that the reservoirs are tied together in a system. The water that goes out from one reservoir will often float down a river into the next one. It would be valuable to see how flooding in a reservoir would affect other reservoirs further downstream. This is one of the things that – in today’s workflow – may need lots of table look-ups, and therefore take much time.



**Figure 7.1:** Selected sketches from iteration 1

### 7.2.2 Sketching iteration 2

The flooding scenario was now in focus, and we needed to explore different alternatives of presenting data with a perspective of time. As illustrated in Figure 7.2, one of our solutions was a chart displaying the water volume as a function of time. We added a line indicating the maximum water level of the reservoir, and since reservoirs also have a restriction of a minimum water level (due to environmental issues), we included that as well. The user is now able to see when the flooding occurs by the graph crossing the max line.

To address the problem of one flooding reservoir affecting others, we included a sheet that is similar to what SINTEF is already using (illustrated in the bottom right of the figure) [28]. What is separating this from the existing watercourse maps, is that we included a slider in order for the user to move ahead in time, and continuously see how the water levels affect each other.

#### **Feedback:**

The chart were received with thumbs-up, and several interactional features were suggested; the user must be able to click on the graph to be presented the exact values; the user should be able to zoom in on the graph; and the amount of water above the max line should be presented with an exact number. This would help estimating the severity of the flood.

Further, the application should address the problem of aggregating modules. It should be possible to aggregate different modules together, based on criteria such as area or ownership. There was done some work on aggregation in the pre-thesis project, but the modules were only clustered based on distance between the modules. Aggregation criteria will help the researchers limit the range of hydropower modules, making it easier to gain a quick overview over different sections of the hydropower system.

Finally, we should address the complex computations needed to be done in SINTEF's computational models. As concluded in the pre-thesis project, the tablets are still somewhat too weak to handle the more complex calculations [12]. We should therefore look at the possibilities sending values to a server and getting results back.

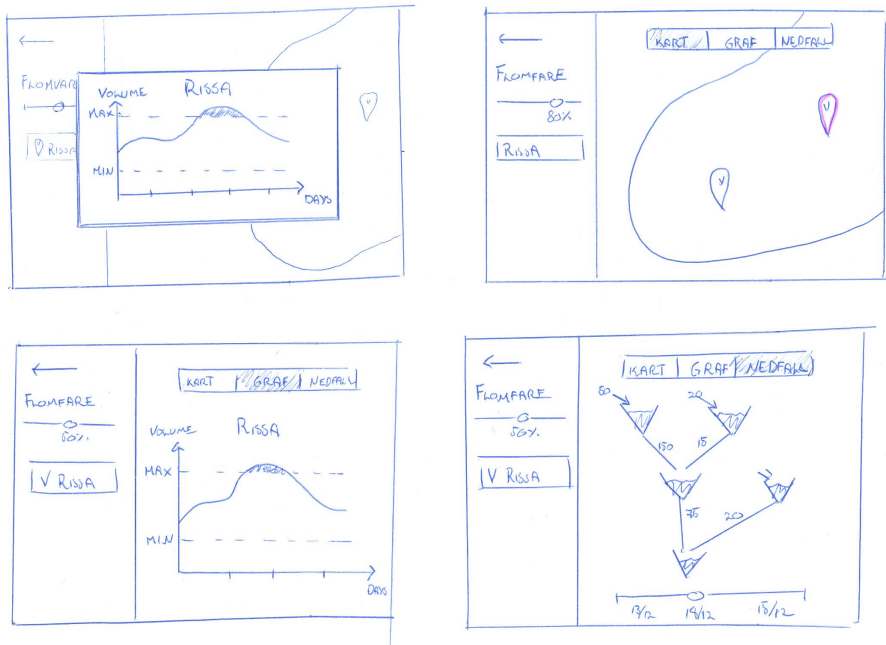


Figure 7.2: Selected sketches from iteration 2

### 7.2.3 Sketching iteration 3

More functionality was desired in the previous feedback session, and we had to take a step back and think thoroughly about the applications structure. We wanted to include calculations and aggregation among others, and we wanted to do that without drowning the user in confusing menus.

We came up with three different alternatives, all presented in Figure 7.3. The first one being a left-aligned menu bar as made popular by an application Twitter<sup>1</sup> released for Android tablets. The second one is a Google Chrome-like tab bar on the top, and the final one is a standard Android tab bar on the top.

#### **Feedback:**

Idar agreed that all the three structures would seem logical to the user. Though, there was one alternative that stood out from the others, and that was the left-aligned menu. The application will be solely designed for a tablet in landscape mode. There is plenty of more space in width, than in height. We therefore chose to go for the left aligned menu, as it was important to maintain the height, and it would feel confusing to have two similar menus above each other.

By now, we agreed in starting with a digital prototype to get a better feel of the design, as we pretty much had agreed on functionality and structure. Figure 7.4 names the different views in the application in which will make it easier to refer to specific sections of the GUI later in the report.

---

<sup>1</sup>Twitter on tablets – <https://blog.twitter.com/2013/twitter-for-android-now-on-tablets>





Figure 7.3: Selected sketches from iteration 3

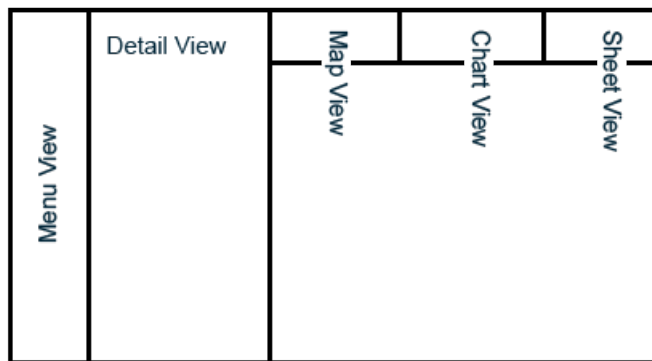


Figure 7.4: Application structure

### 7.2.4 Digital mock-ups iteration 4

Figure 7.5 presents the first iteration of the sketch made digital. There has not been much change in the design, but we now get a feel of the actual colors, sizes, and such. This makes us more aware of available (or not available) space on the screen. There were only time to create the initial screen in the first iteration, but it was enough to get another round of feedback.

#### Feedback:

All the modules should be listed beneath the slider, and not just the ones crossing the threshold set by the slider. The module list should also follow the same color conventions as the markers. If a marker turns red the module should also turn red in the list beneath the slider.

Considering the map, they thought the reservoir icons were pretty misleading. *Is this reservoir half full?* The reservoir icon should be consistent through the application, and using the V-shaped flask icon was more familiar.

The icons should be more similar to the standard icons that SINTEF defines in [28] to prevent confusion. Also, the tab bar currently used is not the standard tab bar used in Android.

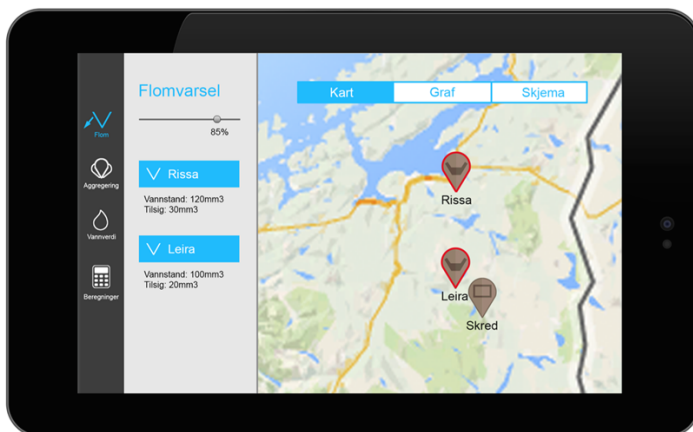


Figure 7.5: Digital prototype from iteration 4

### 7.2.5 Digital mock-ups iteration 5

Figure 7.6 presents the second and final iteration of the digital mock-ups. A lot of effort was done to finalize the mock-ups according to earlier feedback.

Several changes were done mid-section of the application. We now list all the visible modules in the list beneath the slider, and the modules are color coded like the markers. This means that if a reservoir is crossing the threshold, both the button in the list beneath the slider and the marker on the map will turn red.

The aggregation functionality has also been digitalized. It is made simple, with checkboxes representing different clustering-criteria. Only the modules that match the selected criteria will now be clustered together, and the other modules will partially fade as illustrated in the bottom left tablet in Figure 7.6.

The button-list and the map view now act together. When the user selects a module from the list, its appropriate marker will also be selected, and vice versa. This is illustrated in the top-left tablet in Figure 7.6. The module buttons now holds the same functionality as the radio buttons presented in Section 5.2, and is thus violating the standard appearance of such elements. But as the standard radio buttons only appears to be 4 mm in diameter on the screen, we wanted to change them. Studies reviewed in Section 5.3 clearly conclude that a diameter of 8-10 mm or bigger is appropriate for touchable surfaces, supporting our choice. We also did minor changes to the map icons according to the given feedback.

#### **Feedback:**

There was not much ambiguity in the design, not even the radio buttons. We therefore decided to move on to the implementation phase. Things may still feel different on an actual tablet, but we thought we had prevented any big structural changes by this point.



Figure 7.6: Digital prototype from iteration 5

### 7.3 Adapting icons

The icons currently used by SINTEF are illustrated in Figure 7.7. We wanted to adapt these icons to interactive maps on small screen sizes. We wanted to do as little changes as possible, but at the same time use the same style as other mobile applications. They also need to be comfortable touching, even with clumsy hands. We argued in Section 5.3 that elements should be at least 8-10 mm for maximal accuracy, so we therefore made the icons appear slightly bigger than this on the screen.

There was made upgrades to the icons in the pre-thesis project (presented in Appendix A. Though, the tests in iteration 5 (Section 7.2.5) showed that they needed another re-design to prevent ambiguity.

The resulting icons are presented in Figure 7.8. They are formed as droplets, which we concluded in Section 4.7 is usual in interactive maps. We made icons to match the clustering functionality as well. We borrowed some design ideas from the FINN application reviewed Section 4.4, and they are illustrated in Figure 7.9. The number tells how many entities that are clustered together.



**Figure 7.7:** The currently used icons in watercourse maps



**Figure 7.8:** New icons made for interactive maps



**Figure 7.9:** Clustered icons, the number (2) states the number of modules in the cluster



# APPLICATION IMPLEMENTATION

This chapter will introduce the risks and challenges of implementing this application. Further, it contains the requirements of the application, and the development iterations. Finally, it presents the 4+1 view model, with different views of the application, concerning different stakeholders.

## 8.1 Implementational risks and challenges

There are numerous risks and challenges involved in developing a software application, related both to technical and business aspects. Identifying risks is important in software development, as it makes us aware of potential flaws and weaknesses of our software. Every application has weak spots, and identifying them make us aware of what problems may occur. This section will address the risks and challenges regarding the development of the application.

### **Technical expertise**

A limited experience in developing Android applications may cause a slower development. The risk is high that lots of time will be spent on getting to know Android development and finding the appropriate libraries to make the implementation go smooth.

### **Data**

Our intention is to make the front-end of an application that will save users time in their daily work. We will be using SINTEFs existing CIM as our information model as far as it goes. Though our intention is also to suggest an extension to the existing CIM if needed. There is therefore a risk that existing CIM-based systems may not directly comply with ours.

**Firmware**

The Android tablet used for development, does not support the newest firmware, thus neither the newest Application Programming Interface (API). There is a medium risk that this may increase the development time, since much of the documentation and guides online are directed against the newest API.

**Changing design**

We will do two iterations of development, and there *will* be design changes. Changing design during the implementation phase will often take much time. Though, it is necessary, as the interaction may feel completely different from what we thought in the mock-up phase. This is a challenge that will appear, but we have hopefully prevented any big structural changes by being thoroughly with the mock-ups.



## 8.2 Requirements

This section will present the requirements of the application. These requirements are made based on interviews, tests, and the given feedback. Requirements were continuously added along the design and development process due to the chosen prototyping methodology. We always made more requirements than we thought there was time to implement, in case the development went faster than expected. This is why some of the requirements are not completed.

All of the requirements are evaluated by black-box testing, i.e. each requirement was manually tested against the final application and labeled completed or not.

### 8.2.1 Design requirements

Table 8.1 presents the design requirements of the application.

**Table 8.1:** Design requirements

<b>ID</b>	<b>Description</b>	<b>Priority</b>	<b>Completed</b>
D1	ICONS		
D1.1	Anything clickable should be larger than 9x9mm for maximal accuracy.	High	Yes
D1.2	The markers should relate to SINTEFs existing icons.	High	Yes
D1.3	The icons should look and feel like standard Google Maps markers.	High	Yes
D1.4	The markers should provide an icon illustrating what kind of module it is.	High	Yes
D1.5	The icons should give visual feedback when a set limit is crossed. This feedback should be a red border around the icon.	High	Yes
D1.6	The clustered icons should reveal the number of modules in the cluster.	Medium	Yes
D1.7	The icons should turn transparent if they do not match the set aggregation criteria.	Medium	Yes
D2	MAP		
D2.1	It should be possible to pinch-to-zoom.	Medium	Yes
D2.3	The marker of a cluster should be placed in the center of the clustered markers.	Medium	Yes

Continued on next page

Continued from previous page

<b>ID</b>	<b>Description</b>	<b>Priority</b>	<b>Completed</b>
D2.4	The info window should present the modules name, ID(s), and max estimated value.	Low	Yes
D3	CHART		
D3.1	The chart should show an estimated series of water levels for the next X time steps for each reservoir or cluster.	High	Yes
D3.1	The chart should visualize the set limit that should not be crossed.	Medium	Yes
D3.2	The chart should visualize the reservoirs maximum water level.	Medium	Yes
D3.3	The chart should visualize the reservoirs minimum water level.	Low	Yes
D3.4	The chart should have labels on the axis.	Low	Yes
D4	MENU		
D4.1	The selected element in the menu should have another color than the other elements in the menu.	Medium	Yes
D5	MID-SECTION		
D5.1	The selected button should have another fill color than the other buttons.	High	Yes
D5.2	The module buttons in the list beneath the slider should turn red if the estimated values exceed the set limit.	Medium	Yes
D5.3	There should be a label that indicates the value of the progress bar.	Medium	Yes

### 8.2.2 Functional requirements

Table 8.2 presents the functional requirements of the application. F3 is only partially completed, as the clustering does not happen when the markers are exactly 20px apart. It happens when the marker icons are starting to overlap, which is a fair trade-off. F7 is partially completed as the user may estimate the volume of water lost during a flood by the chart, but is not presented the exact value. F11 was not implemented, because the estimated time it would take to implement greatly exceeds what we would gain by implementing it. The same goes for F14 and F15.

**Table 8.2:** Functional requirements

<b>ID</b>	<b>Description</b>	<b>Priority</b>	<b>Completed</b>
F1	There should be a menu bar to switch between different functions.	High	Yes
F2	The user should be able to aggregate modules based different criteria.	High	Yes
F3	The clustering of modules should happen when the modules are less than 20 px apart, and according the clustering criteria selected in the aggregation-view.	High	Partially
F4	The limit should be set according to the progress bar.	High	Yes
F5	The chart should always present the values of the selected module.	High	Yes
F6	The user should be able to select the time steps of the chart.	Medium	Yes
F7	The user should be able to know exactly how much water is lost during an estimated flood.	Medium	Partially
F8	The charts limit line should increase/decrease according to the given value in the progress bar.	Medium	Yes
F9	When a module's marker is selected, its corresponding button in the module-list should also be set selected.	Medium	Yes
F10	When a module's button is selected, its corresponding marker should also be set selected.	Medium	Yes
F11	The application should be able to send values to a server, run a script, then get the results back.	Medium	No

Continued on next page

Continued from previous page

<b>ID</b>	<b>Description</b>	<b>Priority</b>	<b>Completed</b>
F12	The application should get hydro values from an online API if it has internet connection, and use the most recent information if it has not.	Low	Yes
F13	It should be possible to select a point in the chart.	Low	Yes
F14	The user should be able to refresh the hydro-power values at any time.	Low	No
F15	The user should be able to sort the module list by name, status or cluster/not cluster.	Low	No

### 8.2.3 Non-functional requirements

Table 8.3 presents the non-functional requirements of the application. Since we did not have many architectural tactics for our application such as modifiability and performance, we do not have many non-functional requirements. We also consider potential non-functional requirements regarding usability covered by our design requirements. The reason NF3 is only partial completed, is that we have not considered caching of the maps. But if the map is not cached, there is still possible to use the rest of the application.

<b>ID</b>	<b>Description</b>	<b>Priority</b>	<b>Completed</b>
NF1	The application should use phrases related to the domain.	High	Yes
NF2	It should be explored if the existing CIM need to be changed or extended.	Medium	Yes
NF3	The application should still work if the tablet loses Wi-Fi connection.	Medium	Partially

**Table 8.3:** Non-functional requirements

## 8.3 Implementation

We followed the suggested development methodology from Section 3.2 with small iterations. As soon as something new was developed, it was presented to Idar, and he gave feedback. Thus, we will in this section present two main iterations. These are the iterations were both Idar and Frederic could attend the feedback session.

### 8.3.1 Implementation iteration 6

In the first iteration of development, the design from the digital mock-up was implemented. Design tends to feel a little different once you can interact with it, which led to minor adjustments in consultation with Idar.

As we implemented the whole structure from scratch, it was not time to implement everything that we had sketched, but we still had enough to get feedback. We invited both Idar and Frederic for a meeting, to share thoughts and feedback. The intention of this meeting was gather new ideas as they now could feel the design on a tablet.

#### Feedback:

They would like to see the chart labeled with appropriate time steps. The user should be able to change time interval for the chart in an easy way. This is currently not possible. Other than that, they wanted to be transferred to the chart tab when they press the information window above a marker. This is a good example of functionality that is hard to uncover on mock-ups, but easily shines when the user get to interact with the implemented solution.

Idar points out that aggregation should gain focus in further development. As pointed out in the design phase, this is a key feature in gaining a quick overview, and will certainly be useful.

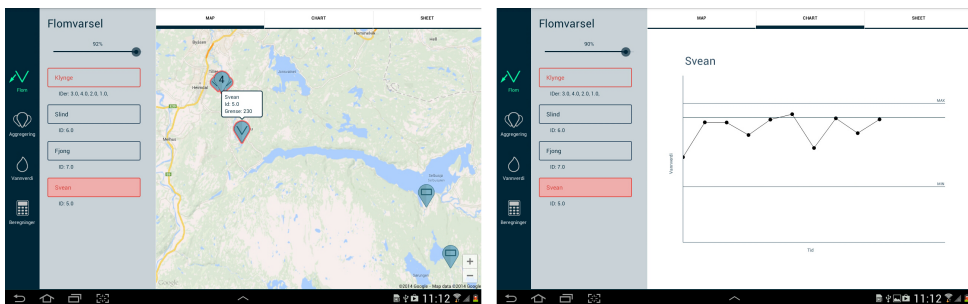


Figure 8.1: Screenshots from development iteration 1

### 8.3.2 Implementation iteration 7

Iteration 7 was our final iteration due to time constraints. We had, in consultation with Idar, decided to focus on the feedback from the previous iteration, including developing the aggregation functionality.

First, we adjusted the chart as requested. We labeled the time steps, and we made a selector to change the time interval of the chart. We also made clicking a markers information window lead the user directly into the chart tab.

Further, we developed the aggregation view between the menu and the map. Aggregation of modules will now happen based on the selected criteria. What is happening, is that we are assigning the modules to different cluster groups, based on the checked criteria. If a module does not match the selected criteria, they will not aggregate together with the modules that match. They will also fade partially. This is all illustrated in Figure 8.2. There are a variety of possibilities in what you could use as clustering criteria based on your interests. We chose ownership and area.

#### Feedback:

There are still things to be implemented for the application to be fully completed, but we have reached far enough to answer the questions we were out to investigate. As Frederic said: “I can see this as a useful tool when we test our models internally to quickly gain overview of errors in our models”. Some key points for further development are presented in Section 10.6.

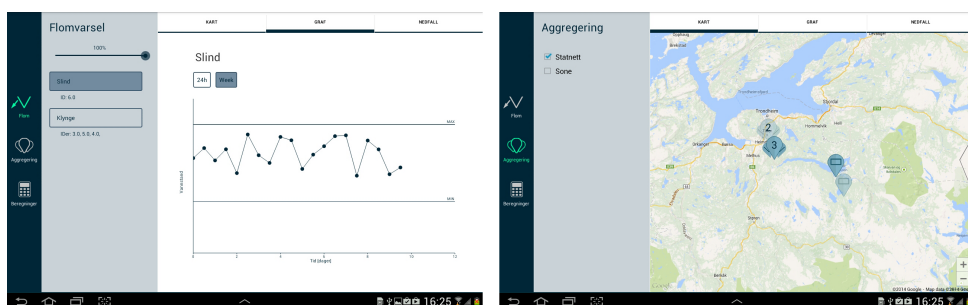


Figure 8.2: Screenshots from development iteration 2

## 8.4 4+1 View model of the application

These sections present the logical view, the development view, and the physical view of the application. The process view is omitted, as there is no special pattern in the usage. The plus-one view is already presented as scenarios in Section 2.2.

### 8.4.1 Logical view

In Figure 8.3 we present a sequence diagram of what happens when the user changes the slider on the application. We are using fragments to separate the different views in the application. When we move the slider in the detail fragment, we need a way of sending that change to the map and the chart fragments. Communication between fragments must be done through the activity. Via an interface, we send the changed value to the activity. The activity then pass the value to the tab host, which distributes it further to its fragment tabs.

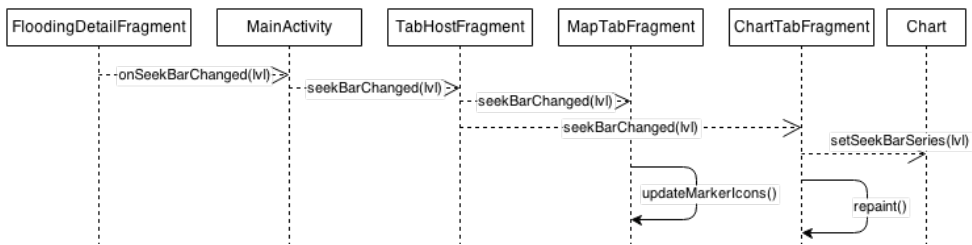


Figure 8.3: Logical view.

### 8.4.2 Physical view

The physical view is used to illustrate how the application executes in a network of computers [43]. Figure 8.4 illustrates this for our application. We have deployed an API hosted on Heroku (see Appendix B), and when the application needs hydropower values, it retrieves it with a HTTP GET request.



Figure 8.4: Physical view.

### 8.4.3 Development view

The development view is presented as a package diagram in Figure 8.5. The arrows between the packages are drawn to illustrate the initialization order. It starts with the activity that further initializes the fragments. The fragments puts the parser in action, which parse data retrieved from an online API as described in Appendix B, and put this data into the models. The models of *reservoirs*, *hydro-power-plants*, and *one-magasins* are expanded to show that they comply with SINTEF's existing CIM presented in Section 6.3, and the extended CIM suggested in Section 9.2.

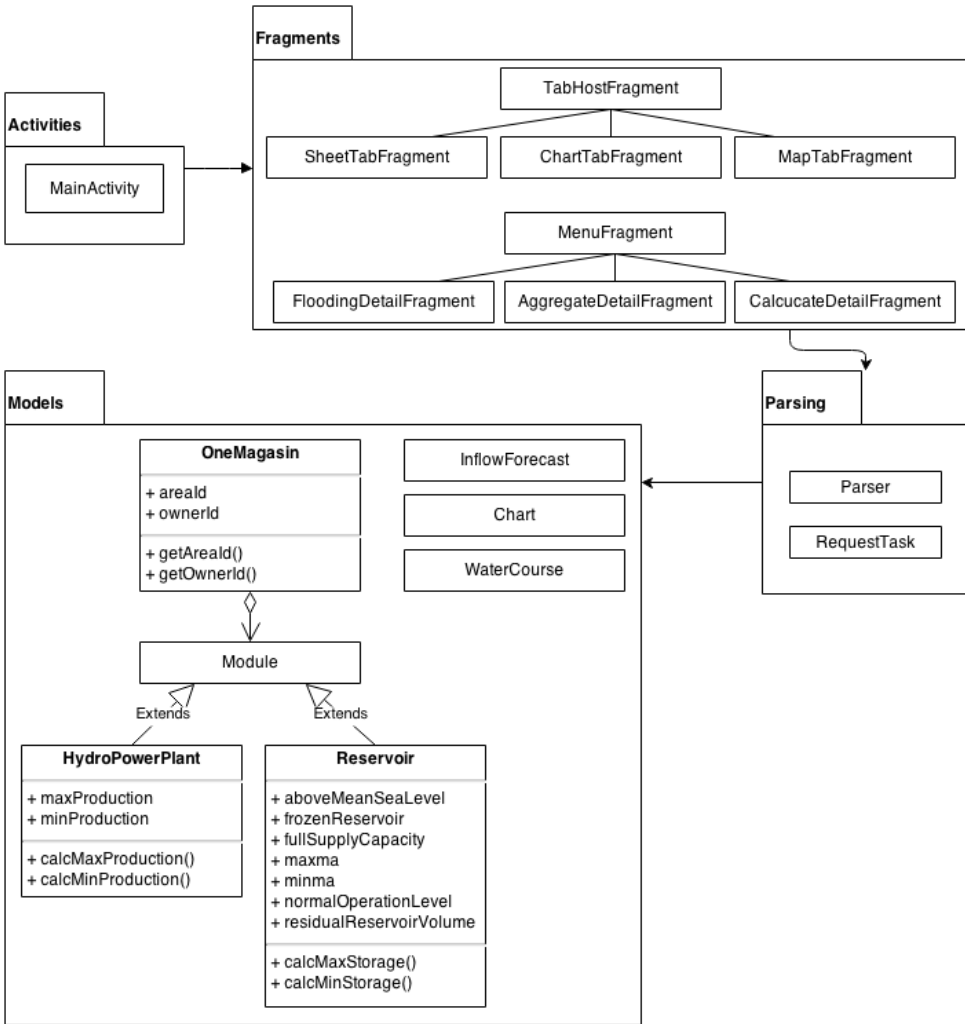


Figure 8.5: Development view.



**Part IV**

**Results**



## RESULTS

SINTEF wanted to explore the mobile domain, looking for opportunities of moving software from CLI environments to mobile ones. Based on interviews, we made scenarios covering concrete problems experienced in the researchers' work. These scenarios have resulted in the final application in which we will describe here. The main result is the design of the application, both visually and functional. In addition, we suggest an extension to SINTEFs existing CIM that will be useful in aggregating reservoirs and hydropower plants to a one-magasin.

### 9.1 GUI and functionality

We chose to explore scenario S1 (Flooding) in depth, because floods are often the highest cost factor in SINTEFs calculations on hydropower. The main purpose is to give the user a quick overview over the potential risks of floods, but at the same time provide detailed information about every single unit in the hydropower system.

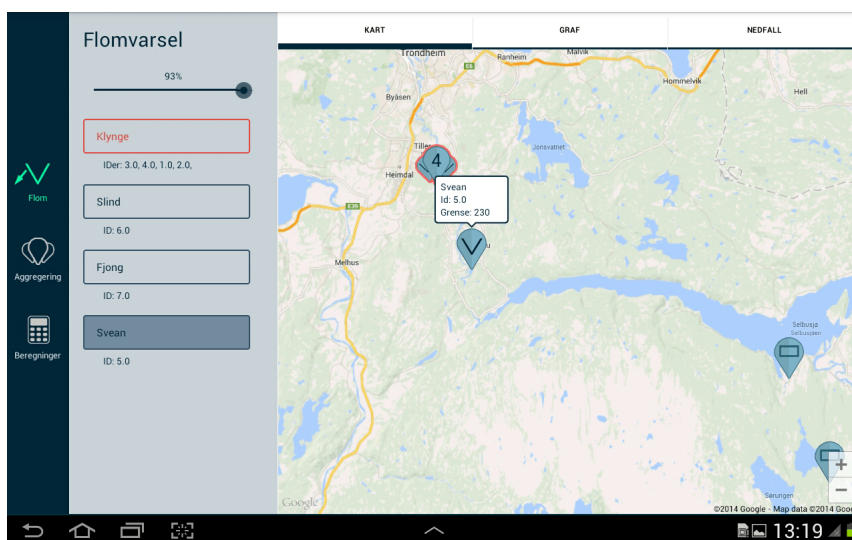
Figure 9.1 provides the resulting GUI for the flooding functionality. The user may set a threshold with a slider. The threshold is initially set to 100 %, meaning we want to be alerted of any reservoir that has an estimated water level higher than the reservoir can handle (100 % of the maximum water level). To alert the users, we make a red border around the marker on the map. Beneath the slider, there is a list of all the visible modules on the map, and this list follow the same coloring as on the map, meaning that if a marker turns red, the corresponding button in the list turns red.

To cover the requirement of giving the user a quick overview, we aggregate modules together when the user zooms out. We pull the markers together into a cluster, labeling the cluster with a number telling how many modules are clustered together. If one of the modules inside a cluster is in danger of flooding, the whole cluster will go red. This makes it easy to localize potential dangers, even when zoomed out on a

large geographical area. When the user zooms in on a small geographical area, the clusters split into single modules again.

The user may switch to the chart tab, and see estimated water values for each module. Figure 9.2 illustrates this. From the list beneath the slider, the user see that the module “Svean” is still selected. The chart contains four graph-lines. Two of them marks the maximum and minimum required water levels. The third line represents the limit set by the slider, and the fourth is the actual estimated water values. This makes it easy to get an impression of the severity of the estimated flooding, making it easier to take appropriate precautions. The user may also switch between different time intervals. We have used 24 hours and a week as examples, but this may be set to anything the user need.

Another important functionality is the users ability to decide what modules that should be clustered together. Different stakeholders have different interests. Some may be interested in specific areas, some may be interested in the modules owned by themselves. We have made one criteria, but this part could be further explored, as there are probably many different interesting criteria. Figure 9.3 illustrates the clustering functionality. Our criterion, “Statnett”, is based on ownership. By selecting the checkbox, we select only the modules owned by Statnett. All the others will be partially faded out. Also, there will be no clustering between the modules owned by Statnett and the rest.



**Figure 9.1:** Flooding GUI with a marker selected, Map-view

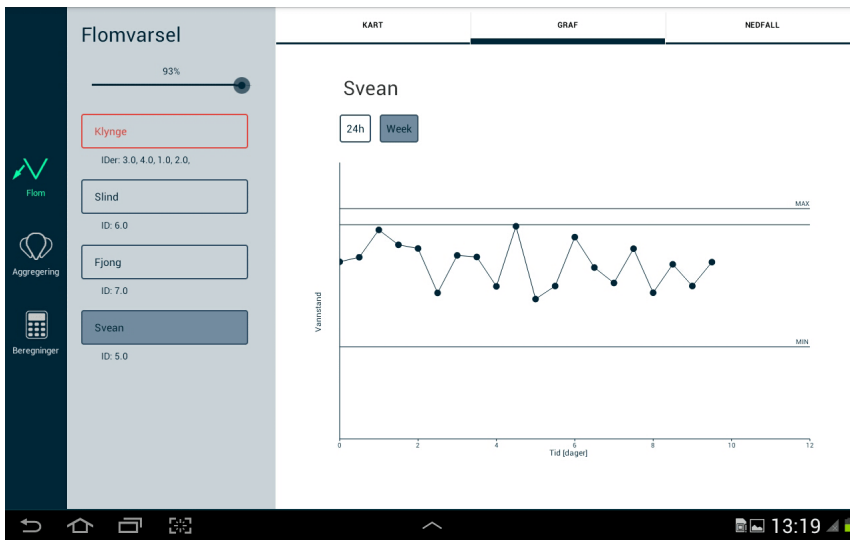


Figure 9.2: Flooding GUI with a marker selected, Chart-view

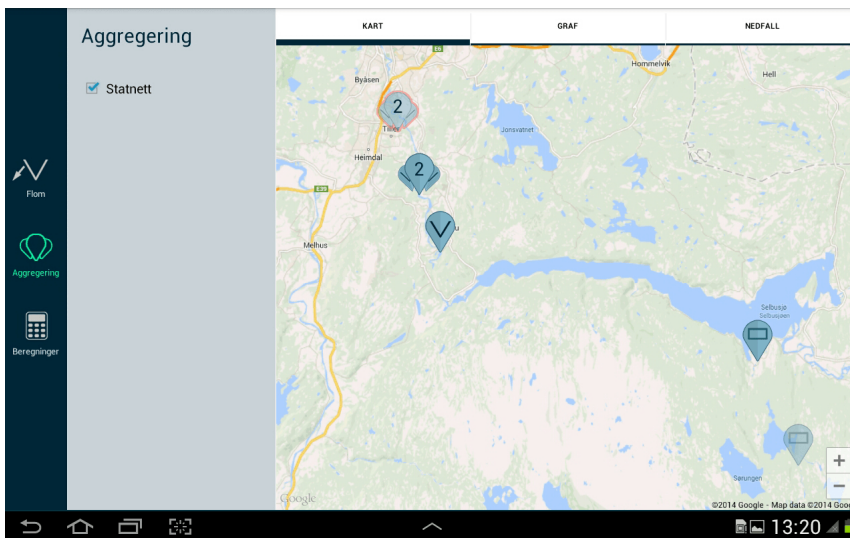
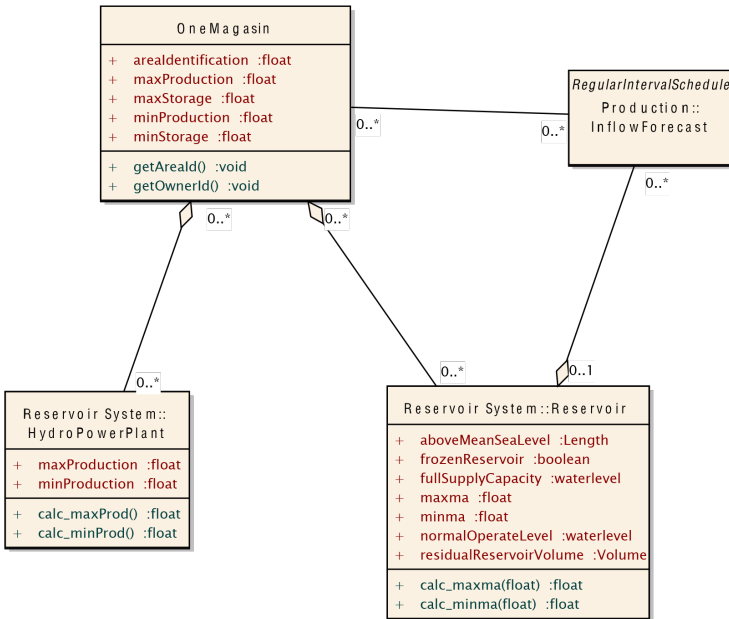


Figure 9.3: Aggregation GUI with a criterion selected

## 9.2 Suggested CIM in hydro power production

To have a fully developed, semantic information model, it is important to define various applications and building sharable databases. And to operate with interacting interfaces is getting more important as information systems are getting more complicated.

The existing aggregated model SINTEF has made, is simplified. Most of the constraints are removed, and it's objective has only been to see if an aggregation is possible. We now suggest an extension to this model, as we have explored the domain by developing an application. In the application we do an aggregation of the modules' ownership, as it will be a valuable aggregation criteria. We therefore suggest to add this to the existing model. The new model is presented in Figure 9.4.



**Figure 9.4:** Suggestion to new aggregated model

The implementation of the models will be pretty straightforward. We only need the right formulas to calculate the parameters maxma, minma, Pmax and Pmin (as explained in Section 6.3). SINTEF's existing formulas consider only time-steps (week)

as input. We suggest the following formulas, extended with ownership as well.

$$maxma[GWh] = \sum_{week,owner,i} maxma(i) \sum_{j=1}^{sea-level} E(j) \quad (9.1)$$

$$minma[GWh] = \sum_{week,owner,i} minma(i) \sum_{j=1}^{sea-level} E(j) \quad (9.2)$$

$$P_{max}(week, owner)[MW] = \sum_{stations} P(Q_{max}(week, owner)) \quad (9.3)$$

$$P_{min}[MW] = \sum P(g_{min}(week, owner)) \quad (9.4)$$





## EVALUATION AND CONCLUSION

This chapter will evaluate and conclude the project. We start by evaluating the research and development methodology based on criteria from the introduction. Then we consider some of the design principles that we wanted to abide, followed by the strengths and weaknesses of the project. We will also answer the research questions and reflect upon the project. Finally we provide key points for further work, and conclude the project.

### 10.1 Methodology

As presented in Section 3.1 we chose to use design science as research method, which fitted the project well. The research were conducted in the spirit of the suggested guidelines, and evaluated with the methods presented in Section 3.1.2.

#### **Descriptive – Informed argument**

Section 1.1 gives the motivation for such an application, and Chapter 4 analyze applications, similar in either domain or functionality. Chapter 7, Application Design, is dedicated to argue for the taken design choices. The arguments are both gathered from relevant research, and interviews with potential users at SINTEF. Together, this gives us strong informed arguments for our choices.

#### **Descriptive – Scenarios**

There have been created scenarios in Section 2.2 to demonstrate the applications utility. We chose to cover S1 in depth, but in such a way that we can easily transfer the same concepts to cover some of the other scenarios as well.

#### **Experimental – Simulation**

The application is using geographical data found on Norwegian Water Resources and

Energy Directorate (NVE) Atlas<sup>1</sup>, which is the leading provider of these kinds of data in Norway. The application also simulates randomly estimated water levels for the modules, to present functionality in interviews and tests.

### Testing – Black Box

The requirements listed in Section 8.2 have been manually tested against the final application, to verify whether they are accomplished or not. These methods have been helping a lot in understanding the domain, and to perform test with potential users.

The prototyping development methodology was without doubt a great match for the development. As this was a new kind of application, with unknown end-results, it was necessary with feedback along with the development. This continuous evaluation with the end-users contributed greatly in increasing the applications utility.

## 10.2 Design principles

There was pointed out some design principles in Section 5.1 to make the application align with both mobile, and more specific, Android standards. This section will briefly sum up how these principles were used.

**KEEP IT BRIEF.** We only serve small portions of information, but still enough to inform the user.

**ONLY SHOW WHAT I NEED WHEN I NEED IT.** We provide several layers of information, and the user may do actions if more information is needed. E.g. we provide information just by the looks of the markers. Further, the user may click on the markers for an information window, and then switch to the chart tab to go even deeper in the levels of information.

**I SHOULD ALWAYS KNOW WHERE I AM.** The menu, the tab bar, and the module buttons in the list beneath the slider are all separated from the other elements by color if they are selected. The user should at all time feel comfortable with the menus. Also, when the user selects a button in the button list, the corresponding marker on the map will be centered and an information window pops up.

**MAKE IMPORTANT THINGS FAST.** The main functionality, namely to see if any modules are crossing a flooding limit is easily accessible from the start screen.

**MATCH BETWEEN SYSTEM AND THE REAL WORLD.** We strived to make the lingo in the application match the correct domain language.

---

<sup>1</sup>NVE Atlas – <http://atlas.nve.no/>

## 10.3 Strengths and weaknesses

### Strengths

The main strength of this project is the application illustrating the potential of making common tasks at SINTEF faster. We could not find any mobile application that covers our problems in such a depth. It highlights that it is possible to move from the CLI environment, to a user-friendlier mobile environment.

Another strength is the scenarios, and the prototyping development method as described in Section 7.2. Many requirements were unclear in the beginning of the project, and an iterative process with continuous feedback was necessary. It contributed greatly in understanding the problems, and it really shaped the application.

The last big strength of this project is the thesis, providing supporting theory on the choices taken. It also contributes to the existing CIM in which will be useful for SINTEF in further development projects.

### Weaknesses

The greatest weakness of the project has been the difference in domain knowledge by the users and the developer. It has probably been used more time than necessary to get a common shared context.

Another weakness was is that the project had three evaluators. User experience is not a “one size fits all” [45], and more potential users may have given different solutions.

A technical weakness is a lack of experience in developing Android applications. Although we are very well known with Java, it has taken some time getting to know Android. Fortunately, there are unlimited of information available, but it did certainly slow down the development.

A final weakness that we also consider a strength, is that the application is only a proof-of-concept. There was more focus on quick development to prove concepts, rather than writing sexy code. This made us able to keep up with short iterations, with lots of great feedback. It made us take a few shortcuts in the implementation that may need refactoring for further use. It is still following SINTEFs CIM, and may be further developed if desired.

## 10.4 Answering research questions

Q1a: How can users get an overview of the entire hydropower system, whilst giving the depth down to each individual reservoir?

- We have taken inspiration by the reviewed applications, and made modules close to each other cluster together. These clusters provide information such as how many modules are clustered, the IDs of the modules, and if one or more modules are in danger of flooding. In this way, the user will gain a quick overview of the modules when zoomed out. If the user finds a cluster interesting due to e.g. that it contains modules with alerts, he may zoom in on that particular cluster to see each individual modules. This will save time in excluding all the other clusters. This is an updated continuation from the pre-thesis project.

The user may also choose criteria deciding what modules should aggregate together. This gives further possibilities for the user to get overview over, e.g. geographical zones and ownership. We may simply exclude all the modules not owned by “Statnett” by checking the criteria of “Statnett” in the aggregation view.

Q1b: How can we improve the efficiency in finding errors and alerts in SINTEFs models?

- As of now, the user may set a limit and all the modules exceeding that limit will turn red. Together with the clustering presented in the previous answer, the user will quickly localize the module crossing the limit. This concept may be transferred in every scenario that needs to illustrate alerts or errors.

Q1c: In what way could we adapt today’s watercourse maps to mobile devices with touch screens?

- We concluded in the pre-thesis project that Google Maps would be the best fit for our application, due to its compatibility with Android, and the broad documentation and functionality. Also, by using geographical maps, the user may more easily observe the surrounding topography. We have developed the existing icons further, and made them comply with other Android applications that use map functionality to increase affordance. We have also added a historical perspective to the existing watercourse maps that makes it more interactive, and the user may easily see changes over time.

Q1d: How can we display results from hydropower calculations on mobile devices?

- We have chosen to present results as a chart, which may be modified easily to fit different scenarios. The chart is useful in presenting hydropower values with a perspective of time. The color-coding of the modules in the map is also a way of quickly presenting information. Finally, we have presented an interactive version

of the existing static map, but we did not have time to implement it. Thus, it did receive positive feedback from the end users, and should be considered for further development.

Q2: How can the CIM be improved to support aggregation of reservoirs and hydropower stations?

— The existing CIM related to aggregation was a simplified model to see if aggregation of reservoirs and hydropower plants was feasible. As we have explored the domain further by developing an application, we suggested an extension to this CIM in Section 9.2. In the application we do an aggregation on ownership. This is not supported in the existing information model. We suggest including that in the model, as it would benefit the value of aggregation.

## 10.5 Reflections

This section will reflect upon elements we encountered during the project.

GOING MOBILE – Moving from a CLI environment to a mobile device isn't necessarily an easy task. As we discussed earlier, there are lots of new opportunities, but also lots of new constraints. The biggest opportunities lie in a new way of interacting with the devices. By using touches and gestures instead of keyboard and mouse, designers have to think differently. To think different may also be the biggest constraint, as it may be hard to get out of the old paradigms. As for SINTEF going mobile, we see it as possible. It would still be more credible when tablets gain enough power to run their complex calculations without relying on other servers. They will also need to go through a great amount of work on the design, as we have only touched very few aspects of their application stack.

NUMBER OF ITERATIONS – We tried to conduct as many iterations as necessary in each of the phases, sketching, prototyping, and implementation. The sketching and digital design was done in 3+2 iterations. We decided to move from sketching to digital prototypes when most of the functionality was in place. We decided to move from digital prototypes to implementation when we had the structure in place. A change of structure in the application could have turned into a time bomb, as we probably would have to re-write big parts of the application. We managed to implement much in the two iterations we did of development, and we implemented enough to answer our research questions, although we did want to make the application more complete.

FUTURE POTENTIAL OF CONTRIBUTION – The source code may not be the most interesting. There are taken a few shortcuts in order to make the final tests, and the code need some refactoring (as explained in Appendix D). We think that the biggest

contribution of the application would be that we proved a potential of moving to a mobile environment. It was interesting working with new concepts of visualizing data. The most tangible contribution is probably the suggested extension of the CIM, as this would be useful in a variety of applications.

KNOWLEDGE – As a lessons learned, we will highlight the importance of domain knowledge. We targeted a similar user group in both the pre-thesis project, and the main project. Nevertheless we ended up with quite different applications. Of course, we targeted another scenario, but this has much to do with the domain knowledge, as it has increased drastic in the course of the project.

## 10.6 Further work

This section includes recommendations for further research and development.

### Explore functionality for other scenarios

This thesis covers the flooding scenario in depth, with the argument that the concept is transferable to several of the other scenarios. It should be done further research on the other scenarios, and implemented solutions for them as well.

### Interactive topographical map

The third tab labeled “Nedfall” was not implemented. We did make digital mock-ups presented in Section 7.2.4, but did not prioritize time to implement it. Though, it received positive feedback from the test users, and should be further explored.

### Calculations with actual models

In the pre-thesis project, we concluded that Android tablets are still not powerful enough for the more complex calculations SINTEF do. But the application would not be very useful without these calculations. As for now, these calculations are made up by random generators. We would suggest the following:

Sending parameters over Internet to SINTEF’s servers were the calculations would be performed. Then the results must be sent back, and visualized in the application. This is the solution that we would try if we had more time for implementation, and the solution we currently would recommend until mobile devices becomes more powerful.

### Aggregation criteria

We have currently one aggregation criterion, ownership. But more potential criteria should be further explored. There are probably lots of different criteria that may be

useful to the SINTEF researchers, and it would expand the work on their information models.

### **Offline mode**

We use an API to fetch values to the application over Wi-Fi. When Wi-Fi is not available, we get the values from a file on the tablet. The application is still functional, but the map may not be working if it is not cached on the tablet. This caching has not been addressed in this project, but must be further investigated for the application to work fully offline.

## **10.7 Conclusion**

SINTEF wanted to explore the mobile domain, looking for opportunities of moving software from desktop environments to mobile ones. Based on interviews, we made scenarios covering concrete problems experienced in the researchers' work. These scenarios made the foundation for our application. Even though we only cover one of the scenarios in detail, we have presented a solution that may apply to several of the scenarios.

By designing and developing an application for an Android tablet, we have illustrated a potential in making the current CLIs mobile and interactive. We have further developed SINTEFs design and icons, and adapted them to a touchable environment.

One of our test users said: "I can see this as a useful tool when we test our models internally to quickly gain overview of errors in our models". This confirms that our application has contributed in revealing opportunities in converting SINTEFs models from a CLI to an interactive GUI.

We are still at an early stage of developing a mobile application. We have developed a new GUI front-end of a prototype application, and we have followed SINTEFs information model, making it possible to later tie front and back-end together. In addition, we have suggested a small, but still valuable contribution to the existing CIM. This proposed CIM extension has been implemented in our application, and is generally viable for other software applications based on this information model.





## REFERENCES

- [1] Powel, “Powel smart generation.” [http://powel.no/Global/DOCUMENTS/Worldwide/Products/White\\_Paper\\_Powel\\_Smart\\_Generation.pdf](http://powel.no/Global/DOCUMENTS/Worldwide/Products/White_Paper_Powel_Smart_Generation.pdf). [Online; accessed 10-February-2014].
- [2] SINTEF, “Sintef energy research.” [http://www.sintef.no/upload/ENERGI/%C3%85rsberetning%20og%20brosjyrer/SINTEF%20Energy%20Research%20may%202013\\_web.pdf](http://www.sintef.no/upload/ENERGI/%C3%85rsberetning%20og%20brosjyrer/SINTEF%20Energy%20Research%20may%202013_web.pdf). [Online; accessed 18-February-2014].
- [3] I. E. A. (EIA), “Key world energy statistics,” 2013. <http://www.iea.org/publications/freepublications/publication/KeyWorld2013.pdf> [Online; accessed 14-March-2014].
- [4] “About android software development kit.” <http://developer.android.com/>. [Online; accessed 03-March-2014].
- [5] “About android developer tools.” <http://developer.android.com/>. [Online; accessed 03-March-2014].
- [6] “About eclipse ide.” <http://www.eclipse.org/>. [Online; accessed 03-March-2014].
- [7] “About git.” <http://git-scm.com/about>. [Online; accessed 03-March-2014].
- [8] “About json.” <http://www.json.org/>. [Online; accessed 03-March-2014].
- [9] “About samsung galaxy tab.” <http://www.samsung.com/global/microsite/galaxytab/10.1/index.html>. [Online; accessed 03-March-2014].
- [10] I. Petersen, “Datastruktur i energisystemer of vannkraftproduksjon,” 2012. SINTEF.
- [11] S. Shrestha, “Study of cim and its use in information modeling of a power system-focusing on hydropower,” 2012.
- [12] J. Aaberg, “Front-end application for hydropower on a mobile device.” <http://folk.ntnu.no/jorgeaab/sp2013.pdf>, 2013. [Online; accessed 08-May-2014].

- [13] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” *MIS quarterly*, vol. 28, no. 1, pp. 75–105, 2004.
- [14] V. Vaishnavi and W. Kuechler, “Design research in information systems,” 2004.
- [15] S. T. March and G. F. Smith, “Design and natural science research on information technology,” *Decision support systems*, vol. 15, no. 4, pp. 251–266, 1995.
- [16] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, “A design science research methodology for information systems research,” *Journal of management information systems*, vol. 24, no. 3, pp. 45–77, 2007.
- [17] E. Van Veenendaal, “Standard glossary of terms used in software testing,” *International Software Testing Qualifications Board*, pp. 1–51, 2010.
- [18] J. W. Creswell and V. L. P. Clark, *Designing and conducting mixed methods research*, pp. 173–204. Wiley Online Library, 2007.
- [19] CMS, “Selecting a development approach.” <http://www.cms.gov/Research-Statistics-Data-and-Systems/CMS-Information-Technology/XLC/Downloads/SelectingDevelopmentApproach.pdf>. [Online; accessed 10-February-2014].
- [20] A. Cockburn and J. Highsmith, “Agile software development, the people factor,” *Computer*, vol. 34, no. 11, pp. 131–133, 2001.
- [21] T. Z. Warfel, *Prototyping: a practitioner’s guide*, p. 30. Rosenfeld Media, 2009.
- [22] “Hydrovallées bigorre.” <https://play.google.com/store/apps/details?id=fr.edf.hydrovallees.bigorre>. [Online; accessed 23-March-2014].
- [23] “Otra kraft.” <https://play.google.com/store/apps/details?id=no.appotech.otrakraft>. [Online; accessed 24-March-2014].
- [24] L. Wroblewski, *Mobile first*. Editions Eyrolles, 2012.
- [25] “Pure android.” <http://developer.android.com/design/patterns/pure-android.html>. [Online; accessed 23-February-2014].
- [26] “Design principles.” <http://developer.android.com/design/get-started/principles.html>. [Online; accessed 13-March-2014].
- [27] “Design principles.” <http://www.nngroup.com/articles/ten-usability-heuristics/>. [Online; accessed 13-March-2014].
- [28] SINTEF, “Vansimtap – dokumentasjon og brukerveiledning,” pp. 1–10.
- [29] N. Savio and J. Braiterman, “Design sketch: The context of mobile interaction,” in *Mobile HCI*, pp. 284–286, 2007.

- [30] D. Kaplan, “Pontiflex: About half of mobile app clicks are accidental.” <http://gigaom.com/2011/01/27/419-pontiflex-about-half-of-mobile-app-clicks-are-accidental/>. [Online; accessed 03-March-2014].
- [31] K. Dandekar, B. I. Raju, and M. A. Srinivasan, “3-d finite-element models of human and monkey fingertips to investigate the mechanics of tactile sense,” *Journal of biomechanical engineering*, vol. 125, no. 5, pp. 682–691, 2003.
- [32] Microsoft, “Design library for windows phone.” [http://msdn.microsoft.com/library/windowsphone/develop/fa00461b-abe1-41d1-be87-0b0fe3d3389d\(v=vs.105\).aspx](http://msdn.microsoft.com/library/windowsphone/develop/fa00461b-abe1-41d1-be87-0b0fe3d3389d(v=vs.105).aspx). [Online; accessed 14-March-2014].
- [33] Apple, “ios human interface guidelines, layout.” <https://developer.apple.com/library/ios/documentation/userexperience/conceptual/mobilehig/LayoutandAppearance.html>. [Online; accessed 14-March-2014].
- [34] B. Schildbach and E. Rukzio, “Investigating selection and reading performance on a mobile phone while walking,” in *Proceedings of the 12th international conference on Human computer interaction with mobile devices and services*, pp. 93–102, ACM, 2010.
- [35] P. Parhi, A. K. Karlson, and B. B. Bederson, “Target size study for one-handed thumb use on small touchscreen devices,” in *Proceedings of the 8th conference on Human-computer interaction with mobile devices and services*, pp. 203–210, ACM, 2006.
- [36] A. W. McMorran, “An introduction to iec 61970-301 & 61968-11: The common information model,” *University of Strathclyde*, 2007.
- [37] I. E. Commission *et al.*, “Iec 61970-301 energy management system application program interface part 301: Common information model (cim) base,” *IEC Reference number IEC*, pp. 61970–301, 2003.
- [38] E. P. R. Institute, “Common information model primar,” 2011.
- [39] R. Podmore, D. Becker, R. Fairchild, and M. Robinson, “Common information model-a developer’s perspective,” in *System Sciences, 1999. HICSS-32. Proceedings of the 32nd Annual Hawaii International Conference on*, pp. 6–pp, IEEE, 1999.
- [40] A. Holmlund and R. Sjöberg, “Information exchange with cim for engergy industry,” 2011.
- [41] C. u. g. Statnett, “A tso perspective on cim.” [http://cimug.ucaiug.org/Meetings/London2012/CIMug%20Presentations%20%20London%202012/Day%20%20-%20Wednesday,%20May%2016/AM3\\_A%20TSO%20perspective%20on%20CIM\\_Sorbotten\\_Statnett.pdf](http://cimug.ucaiug.org/Meetings/London2012/CIMug%20Presentations%20%20London%202012/Day%20%20-%20Wednesday,%20May%2016/AM3_A%20TSO%20perspective%20on%20CIM_Sorbotten_Statnett.pdf), 2012. [Online; accessed 15-April-2014].

- [42] P. B. Kruchten, “The 4+ 1 view model of architecture,” *Software, IEEE*, vol. 12, no. 6, pp. 42–50, 1995.
- [43] M. Kontio *et al.*, “Architectural manifesto: Designing software architectures, part 5,” *IBM developerworks, Feb*, vol. 2, pp. 1–7, 2005.
- [44] H. Obendorf, *Minimalism: Designing Simplicity*. Springer, 2009.
- [45] M. Hassenzahl, “User experience (ux): towards an experiential perspective on product quality,” in *Proceedings of the 20th International Conference of the Association Francophone d’Interaction Homme-Machine*, pp. 11–15, ACM, 2008.
- [46] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, 2000.

Part V

Appendices



# SUMMARY OF THE PRE-THESIS PROJECT

This appendix contains a short summary of the specialization project, conducted from August 2013 to December 2013. It is used as a pre-project to this master thesis.

The project was supervised by Sobah Abbas Petersen (IDI, NTNU). Idar Petersen (SINTEF Energy) was external supervisor. The full project thesis is to be found at [12].

## A.1 Problem description

SINTEF Energy makes complex models to calculate on energy results for hydropower. These models outputs textual results, which gives little meaning to others than expert users. There is currently no user-friendly graphical user interface for mobile devices to visualize these models. There is a desire to simplify the interaction with these models, and make a mobile application with a graphical user interface understandable to other people than the model developer itself. It is also a desire to eventually run these complex models directly on mobile devices, and not only on desktops.

## A.2 Objective

The objective of the project is split in two parts. The first is to adapt today's static maps to a touchable, interactive screen. The second is to explore the possibilities to execute complex computations directly on a tablet, and not through external servers. The overall research questions we asked was:

1. How can we adapt today's watercourse maps on a touchable, interactive screen?
2. What can we do to run SINTEF's models in a most effective way on mobile devices?

### A.3 Methodology

The project was conducted with Design science research method, and an iterative development methodology. This meant weekly meetings with Idar Petersen to identify requirements, and to get feedback on the application. We started with defining some basic functionality. Then we worked out paper prototypes, which have low cost, and are easy to modify. We further made prototypes in Photoshop, and finally developed the application in Java for an Android tablet. We continuously added requirements along the iterations.

Beside prototyping the design we ran heavy computations with artificial data to test the possibilities to split the workload on the tablets two cores.

### A.4 Results

#### A.4.1 GUI

The resulting GUI, with icons, functionality, and maps are one of the main outcomes of this project. The resulting icons are presented in Figure A.1 and Figure A.2. The new icons are upgraded and adapted a touchable interface, from existing watercourse maps [28].



**Figure A.1:** New reservoir icons made for interactive maps.

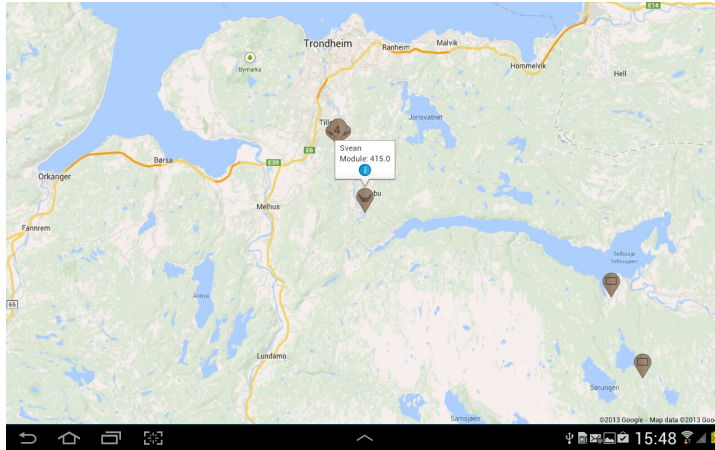


**Figure A.2:** Clustered icons. The number (2) states the number of modules in the cluster.

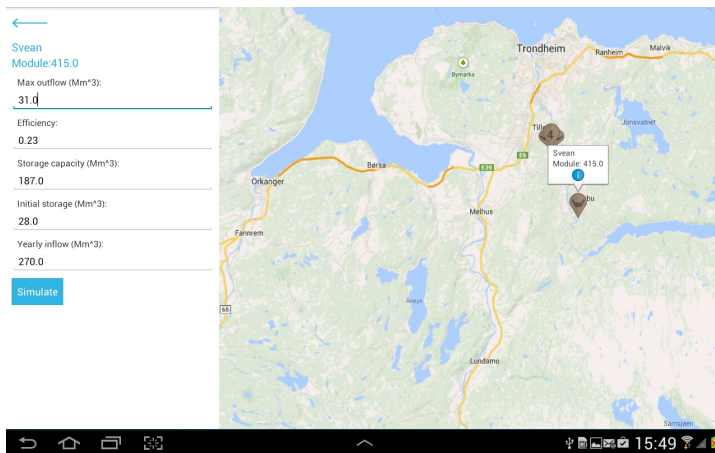
The application is presented in figures A3–A6. Figure A.3 illustrates the map with a selected module. We can see that the pins by itself are giving basic information such as location and module type. When a module is selected you get its key information presented in the information window. It also includes an info-icon to hint that a press on the info window will provide more information. If the info window is pressed, the user get the screen in Figure A.4, were the user might change parameters or run a simulation. Figure A.5 presents the same, but with a cluster selected. A cluster



cannot be modified, because that has to be done on an individual level. Figure A.6 illustrates the results as a chart.



**Figure A.3:** GUI, a module is selected.



**Figure A.4:** GUI, the info window is pressed.

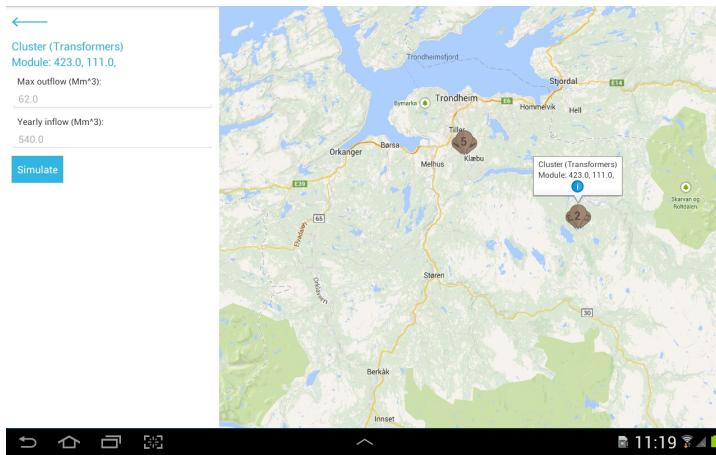


Figure A.5: GUI, a cluster is selected.

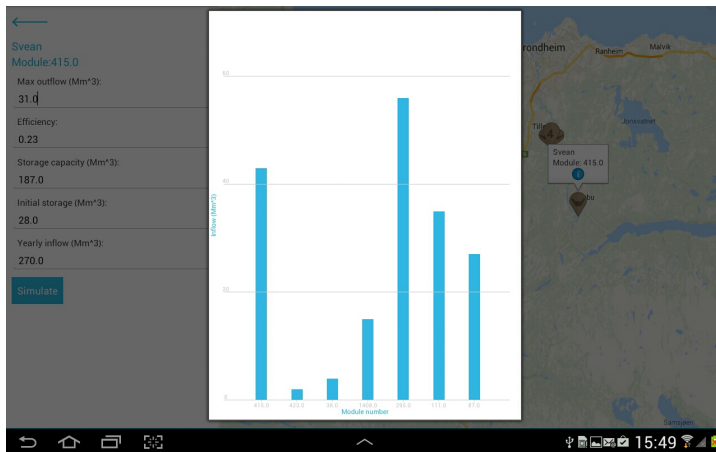


Figure A.6: GUI, resulting chart.

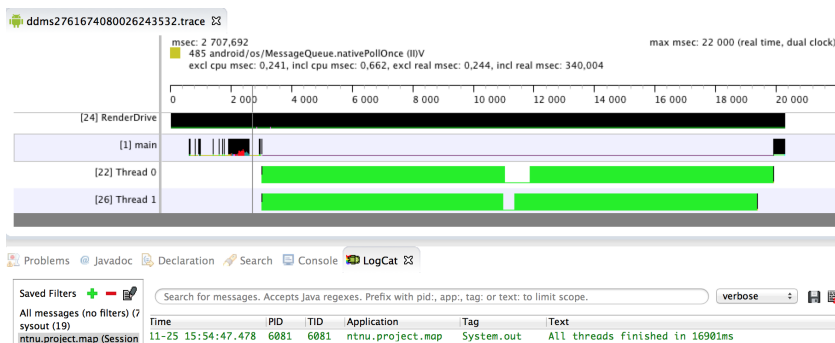
### A.4.2 Parallelization

There are three types of parallelization techniques for Android. For calculations complex enough to last more than a few seconds, it is most efficient to use Java threads.

Parallelization technique	Task
Java threads	For longer tasks in the UI thread, more than a few seconds.
Async tasks	For shorter tasks in the UI thread, shorter than a few seconds.
Intent service	For longer tasks that needs only one worker thread.

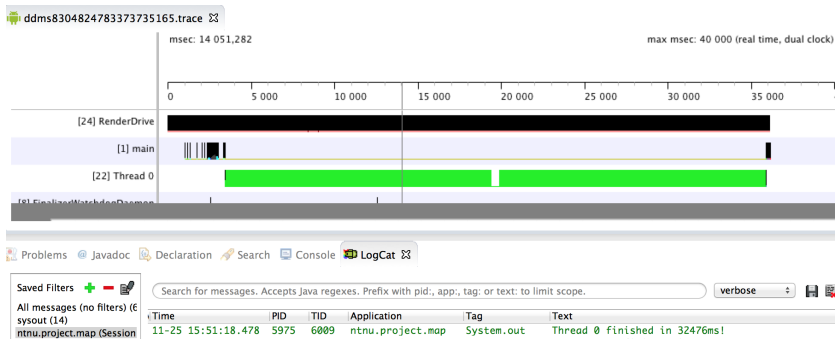
**Table A.1:** Parallelization techniques provided by Android.

Figure A.7 presents tests where two threads are run in parallel. Each thread is doing about 225 million additions and multiplications. We see that the calculations are finished in about 16.9 seconds<sup>1</sup>. Figure A.8 presents a test where a single thread is performing the same amount of additions and multiplications alone. We see that this takes approximately twice as long (32.5 seconds). We may conclude that there is possible to utilize Android's multi-core processors. They are becoming strong enough to handle a large amount of parallel computations in which are promising for moving some of SINTEF's models onto the device, but they are not yet strong enough for the most complex calculations.



**Figure A.7:** A test run of two Java threads run in parallel.

<sup>1</sup>In reality it only takes about 2.7 seconds, but the tracing creates additional workload on the device. It is still valid to compare with other traces.



**Figure A.8:** A test run of a single Java thread.

## A.5 Discussion and conclusion

By creating a mobile application for touchable devices, we have illustrated a great potential in making the current CLI interfaces, mobile and interactive. It is important to adapt design and icons to a touchable interface to increase the affordance.

We have only scratched the surface of potential for such an application. It is early to say when mobile devices will be able to handle the more complex models SINTEF are developing, but we have pointed out an efficient way to utilize the increasing computational power on mobile devices.

# HYDRO API

We wanted to be able to get values from the application over Wi-Fi if it had Internet connection. If it did not have connection, it should use the latest retrieved values. As we did not have access to SINTEFs back-end data, we made a simple API, which we called Hydro API. The API is adapted to fit with SINTEFs existing CIM, so it should not provide problems to change data source from this API and to SINTEFs back-end. This appendix will in brief explain how the API works.

## B.1 Hosting

The API is currently hosted on Heroku<sup>1</sup>. Heroku is a cloud Platform as a Service (PaaS), which support a lot of programming languages. It is an easy-to-use deployment tool where we deployed our API.

## B.2 API

The API is developed with Node.js and Express. Node.js is a scalable server-side application written in JavaScript, and Express is just a simple framework put atop on Node.js.

We use a RESTful API design. Representational state transfer (REST) is an architectural style introduced by Fielding [46] in his doctoral thesis. A RESTfull API just means that we use standard HTTP methods to retrieve resources, which are structured in a certain way.

By performing a HTTP-GET request as presented in Listing B.1, you will retrieve the response on the format listed in Listing B.2. The response is formatted in JSON.

---

<sup>1</sup>Heroku – <https://www.heroku.com/>

---

GET <http://intense-chamber-2865.herokuapp.com/modules>

---

**Listing B.1:** HTTP GET request to the Hydro API.

---

```
{
  "1": {
    "type": "reservoir",
    "latitude": 63.382602,
    "longitude": 10.394046,
    "module_name": "Nedre Leirfoss",
    "ownerId": 101,
    "full_supply_capacity": 187,
    "above_mean_sea_level": 300,
    "residual_reservoir_volume": 10,
    "maxma": 50,
    "minma": 30,
    "normal_operation_level": 5,
    "frozen_reservoir": false
  },
  "6": {
    "type": "hydro_power_plant",
    "latitude": 63.203771,
    "longitude": 10.957562,
    "module_name": "Slind",
    "ownerId": 100,
    "max_production": 31,
    "min_production": 10
  },
  ...
}
```

---

**Listing B.2:** HTTP GET response from the Hydro API.

## INSTALLATION GUIDE

This appendix explains how to install the application on a device.

### C.1 How to install the application on your device

The installation requires:

- An Android tablet with Android 4.0 or higher
- An USB-cable to connect the device to your computer
- An Android file browser application (e.g. “My Files”)
- For mac: Android File Transfer

Following is the process for installing a filename.apk-file on your device:

1. Connect your device to your machine.
  - a) On Windows:  
Copy your filename.apk onto your device (e.g. /sdcard/download).
  - b) On Mac:  
Open Android File Transfer and move your filename.apk onto your device.
2. Make sure that “Unknown sources” is checked under Settings > Applications (or Settings > Security).
3. Open a file browser (e.g. “MyFiles”) on your device.
4. Locate filename.apk, and click it.
5. Choose “Verify and install”
6. The application should now be installed





## DEVELOPMENT ENVIRONMENT

The following sections are presenting how to prepare the development environment, and import the code from this project.

### D.1 Setting up device for development

Since the application is using Google Maps, which is a part of Google Play Services, you need to test the application on an actual device. This requires some setup both on the device, and in your developer environment. This section will cover what is done to set up the environment for this project.

#### D.1.1 Preparing your device for development

You have to enable USB debugging on your device to run your own applications on your device. You can turn this on by navigating to Settings > Developer options. If you have a version that is older than Android 4.0, you find it under Settings > Applications > Development. You can verify that your device is connected to the computer by executing `adb devices` from your SDK `platform-tools/` directory when the device is connected over USB. If it is recognized, it will be listed as a “device”.

#### D.1.2 Preparing development environment

To develop an application on the Android platform, the easiest starting point is to use the Eclipse IDE [6]. To get you started you need to download the Android ADK and ADT-bundle from Androids developer-pages<sup>1</sup>.

When this is done you need to download the appropriate packages to run your projects on an actual device. When you are in Eclipse, go to Window > Android

---

<sup>1</sup><http://developer.android.com/sdk/installing/installing-adt.html>

SDK Manager. Here you have to download the right API. I'm using a Samsung Galaxy Tab 10.1 running Android 4.0.4, which is suited for API level 15<sup>2</sup>.

Note that this application is only tested on a Samsung Galaxy Tab 10.1 running Android 4.0.4 as it is a proof-of concept. The code may run into problems on other devices with other firmware.

### D.1.3 Importing the code into Eclipse

It will not be any magic going on importing the code into Eclipse. You simply press File > Import > Existing Android Code Into Workspace. Then you browse to the project folder, and import it. Be aware that there are two projects that needs to be imported (SintefInternal and android-maps-extensions). android-maps-extensions is a library for the Google Maps to make the GoogleMaps work with clustering markers.

Now you should be all set. When you now press run in Eclipse, your device should be present. Choose it and run.

## D.2 Continue coding

There are some information the developer need if he is going to further work on the existing code. First of all, we recommend taking a look at the 4+1 view model in Section 8.4 to get an overview of the code. Also, as mentioned several times in the thesis, there were taken some shortcuts to be able do the testing in the speed necessary for the project progression. This section will further provide you what need to know.

RESOURCE VALUES – The values are mostly coded directly in the classes, and not through a values.xml-file. The only exception is the colors in which you find all in colors.xml.

CHART – Most of the values in the chart (except the graph itself) are hard coded in the chart model, or the ChartTabFragment (e.g. axis intervals and max and min line). We receive all the necessary values from the API, but did not find it valuable enough to use time to change it. All the values for the graph are randomly generated (within a specified interval) in the InflowForecast class.

GOOGLE MAPS API KEY – You want to change the Google Maps API key to your own in the AndroidManifest file.

MENU ICON – The menu icon does not turn green until you press it once. You need to apply a green filter on the icon when starting the application.

---

<sup>2</sup><http://developer.android.com/guide/topics/manifest/uses-sdk-element.html>