



Western Norway
University of
Applied Sciences

A Mobile Application for Fire Risk Notification based on Edge Computing

Requirement documentation

Authors:

Emilie Hinna Fisketjøn, Abu Tallaha Hussain, Thorbjørn Svendal

Supervisors:

Prof. Lars Michael Kristensen, Ph.D. candidate Ruben Dobler Strand

Department of Computer science, Electrical engineering, and Mathematical sciences

Western Norway University of Applied Sciences

May 23, 2022

REVISION HISTORY

Date	Version	Description	Author
27.02.2022	1.0	First iteration	Emilie, Tallah and Thorbjørn
05.05.2022	2.0	Second iteration	Emilie, Tallah and Thorbjørn
12.05.2022	3.0	Third iteration	Emilie, Tallah and Thorbjørn
18.05.2022	4.0	Fourth iteration	Emilie, Tallah and Thorbjørn
21.05.2022	5.0	Final iteration	Emilie, Tallah and Thorbjørn

Table of Contents

- INTRODUCTION..... 1**
- FUNCTIONALITY 2**
 - 2.1 USE CASE DIAGRAM 2
 - 2.2 USE CASE DESCRIPTION 3
 - 2.3 SEQUENCE DIAGRAMS 7
- XAMARIN 9**
- DOMAIN MODEL..... 10**
- PROTOTYPES..... 11**
 - 5.1 GRAPHICAL USER INTERFACE DESIGN 11
 - 5.1.1 Wireframe..... 11
 - 5.1.2 First iteration 12
 - 5.1.3 Final design iteration..... 13
- BIBLIOGRAPHY 16**
- TABLE OF FIGURES 17**

Introduction

The purpose of the requirement documentation is to function as an overview regarding the requirements of the project and application the project owner set. This document contains the prototypes regarding the graphical user interface, a presentation of the use cases, sequence diagrams associated with the use cases and the domain model.

The documentation contains use cases and sequence diagrams which explain in detail how the user will interact with the system, which functions are performed in the background and the control and data flow of the mobile application. This document also includes a domain model which is used to function as an overview of the project. Lastly, this document has a set of prototypes, many of these prototypes are graphical user interface drawings made by the project owner which functioned as starting points for the development of the application.

Functionality

2.1 Use case diagram

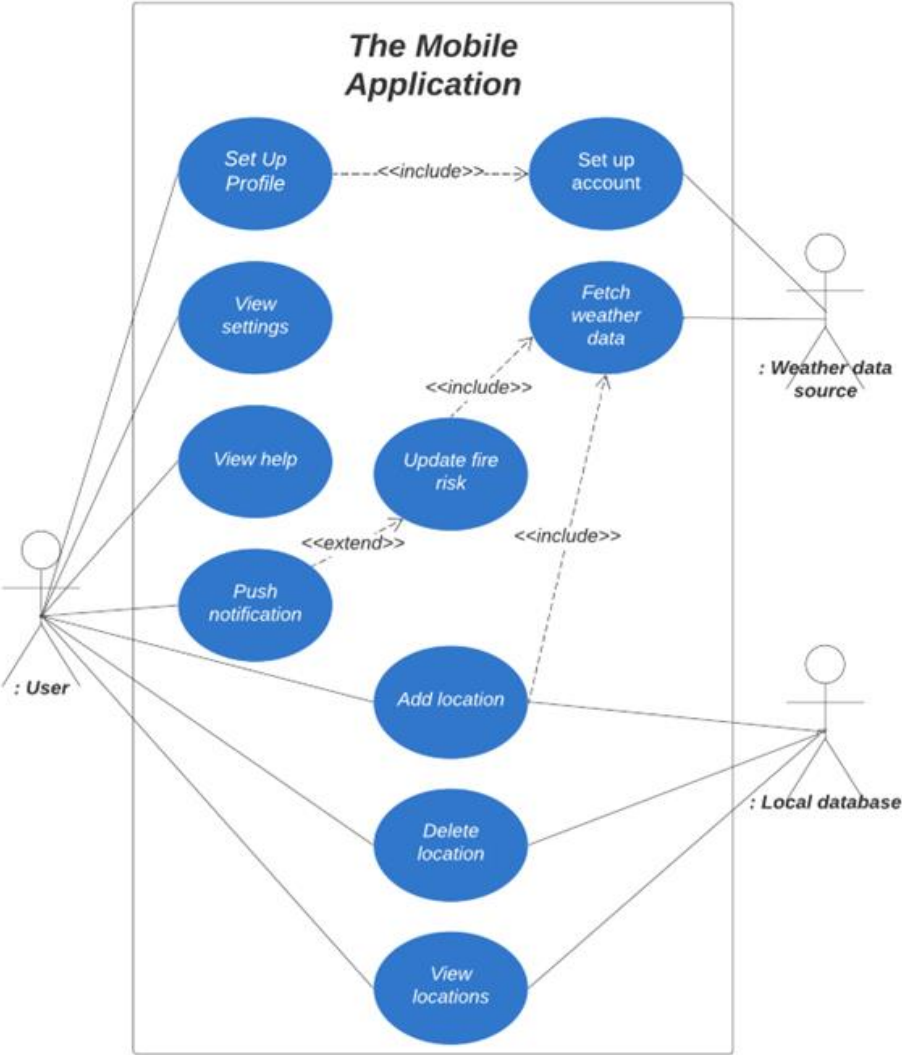


Figure 1. Use case of the application

2.2 Use case description

Name:	<i>Set Up Profile</i>
Actors:	<i>User</i>
Purpose/Goal:	<i>This use case aims to set up a MET profile and receive the MET ID to be able to use the application.</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user wishes to add a new location, but does not have a MET account</i> 2. <i>The user goes to “Help” page, and read the instructions under “Get started”.</i> 3. <i>The user will trigger the Set up account use case</i> 4. <i>The user goes to “Settings” in the application and paste the MET ID and presses “Submit”.</i>
Alternative flow:	<ol style="list-style-type: none"> 1. <i>The user already has a MET account</i> 2. <i>The user can skip the Set up account use case and go directly to “Settings” and paste in his/hers ID.</i>
Exception flow:	<ol style="list-style-type: none"> 1. <i>The user exit the application</i>

Name:	<i>Set up account</i>
Actors:	<i>User</i>
Purpose/Goal:	<i>This use case <<includes>> the Set Up Profile use case, and is triggered when the user sets up a met account.</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user goes to the frost.met.no website</i> 2. <i>The user type in his/hers email and presses “submit”</i> 3. <i>The user receive a MET ID</i> 4. <i>The user copy the generated MET ID</i>
Exception flow [#1]:	<ol style="list-style-type: none"> 1. <i>The user goes to frost.met.no</i> 2. <i>The website was not able to provide the user with an ID.</i>
Exception flow [#2]:	<ol style="list-style-type: none"> 1. <i>The user goes to frost.met.no</i> 2. <i>The website is under maintenance</i>

Name:	<i>View settings</i>
Actors:	<i>User</i>
Purpose/Goal:	<i>This use case aims to show all available options/views in “Settings” page</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user goes to Settings</i> 2. <i>The user can submit a MET ID.</i>
Exception flow:	<ol style="list-style-type: none"> 1. <i>The user quits the application</i>

Name:	<i>View Help</i>
Actors:	<i>User</i>
Purpose/Goal:	<i>This use case aims to show all available options/views in “Help” page</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user goes to “Help” page</i> 2. <i>The user can choose between different drop down boxes “Get started”, “Add a new location”, and “What is FireGuard?”.</i>
Alternative flow:	<ol style="list-style-type: none"> 1. <i>The user gets redirected to “Help” page from Add location use case, and “Get started” tab is already open.</i>
Exception flow:	<ol style="list-style-type: none"> 1. <i>The user quits the application</i>

Name:	<i>Push notification (See Figure 2)</i>
Actors:	<i>The application, User</i>
Purpose/Goal:	<i>The purpose of this use case is to further explain the push notifications and all actions that follows. The user is not responsible for triggering this use case, but only receive a notification.</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The Update fire risk use case is triggered</i> 2. <i>If changes in fire risk is detected, the user will receive a push notification from the application</i>
Exception flow [#1]:	<ol style="list-style-type: none"> 1. <i>The user has disabled all notifications from the application.</i>

Name:	<i>Update fire risk</i>
Actors:	<i>Local timer within the application</i>
Purpose/Goal:	<i>The purpose of this use case is to further explain the use case that automatically will be triggered by a local timer within the application</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The application will be triggered every sixth hour</i> 2. <i>All locations will be retrieved from the database</i> 3. <i>Fire risk for each location will be updated, this includes the use case Fetch weather data use case</i> 4. <i>All locations will be updated in the database</i> 5. <i>If changes occur, the Push notification use case will be triggered.</i>
Exception flow [#1]:	<ol style="list-style-type: none"> 1. <i>The application fails to retrieve all location from the database</i> 2. <i>Fire risk for locations will not be updated</i>
Exception flow [#2]:	<ol style="list-style-type: none"> 1. <i>The Fetch weather data use case fails</i>

	2. <i>Fire Risk for locations will not be updated</i>
--	---

Name:	<i>Fetch weather data</i>
Actors:	<i>Weather data source</i>
Purpose/Goal:	<i>The purpose of this use case is to further explain how the application fetch weather data from the weather data source</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The application construct a request to the weather API's source</i> 2. <i>The API respond with a JSON object</i> 3. <i>The application will convert the JSON object to .NET weather data object</i> 4. <i>The application will use the weather data object</i>
Exception flow [#1]:	<ol style="list-style-type: none"> 1. <i>Throws the exception "Unauthorized", which means that the MET DI is invalid</i>
Exception flow [#2]:	<ol style="list-style-type: none"> 1. <i>Throws the exception "Bad request", which means that the coordinates are invalid.</i>

Name:	<i>Add location (see Figure 3)</i>
Actors:	<i>User, local database, weather data source</i>
Purpose/Goal:	<i>The purpose of this user case is to add a location. Through this user case, the user will create a location, give it a name and specify its geographic coordinates (longitude and latitude).</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user wishes to add a new location, and presses the button.</i> 2. <i>The user gives the location a name, typically a city name.</i> 3. <i>The user specifies the longitude and latitude or uses the "get my location" which fetches the user's location based on his position.</i> 4. <i>The application will trigger the Fetch weather data use case</i> 5. <i>The application creates the location and stores the location in the local database.</i> 6. <i>The user is redirected to the dashboard which displays all the locations</i>
Exception flow [#1]:	<ol style="list-style-type: none"> 1. <i>The user wishes to create a new location and therefore presses the "add location" button.</i> 2. <i>The user receives an alert which notifies him that he needs to create an MET account, which include the Set Up Profile use case</i> 3. <i>The user submit the MET ID</i>
Exception flow [#2]:	<ol style="list-style-type: none"> 1. <i>The user wishes to create a new location and therefore presses the "add location" button.</i>

	<ol style="list-style-type: none"> 2. <i>The user fills the form</i> 3. <i>The Fetch weather data use case will be triggered</i> 4. <i>The application will show the exception message on the mobile screen</i> 5. <i>The user will press “ok”</i>
--	---

Name:	<i>Delete location</i>
Actors:	<i>User</i>
Purpose/Goal:	<i>The purpose of this use case is to make it possible for the user to delete a location.</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user wish to delete a location</i> 2. <i>The user is in the View locations use case</i> 3. <i>The user press the location</i>
Exception flow [#1]:	<ol style="list-style-type: none"> 1. <i>The user wishes to edit an existing location</i> 2. <i>The user makes the changes</i> 3. <i>The user is alerted, a location which covers this area exists already and is asked to provide another address.</i>

Name:	<i>View locations</i>
Actors:	<i>User</i>
Purpose/Goal:	<i>The purpose of this use case is to get more information, regarding the fire risk for a specific location</i>
Normal flow:	<ol style="list-style-type: none"> 1. <i>The user will go to the “My location” page</i> 2. <i>The user will choose a location and press the drop down option</i> 3. <i>The application will display the fire risk and wind data for each day</i>
Alternate flow [#1]:	<ol style="list-style-type: none"> 1. <i>The user presses the “Delete” button and the Delete use case is triggered</i>

2.3 Sequence diagrams

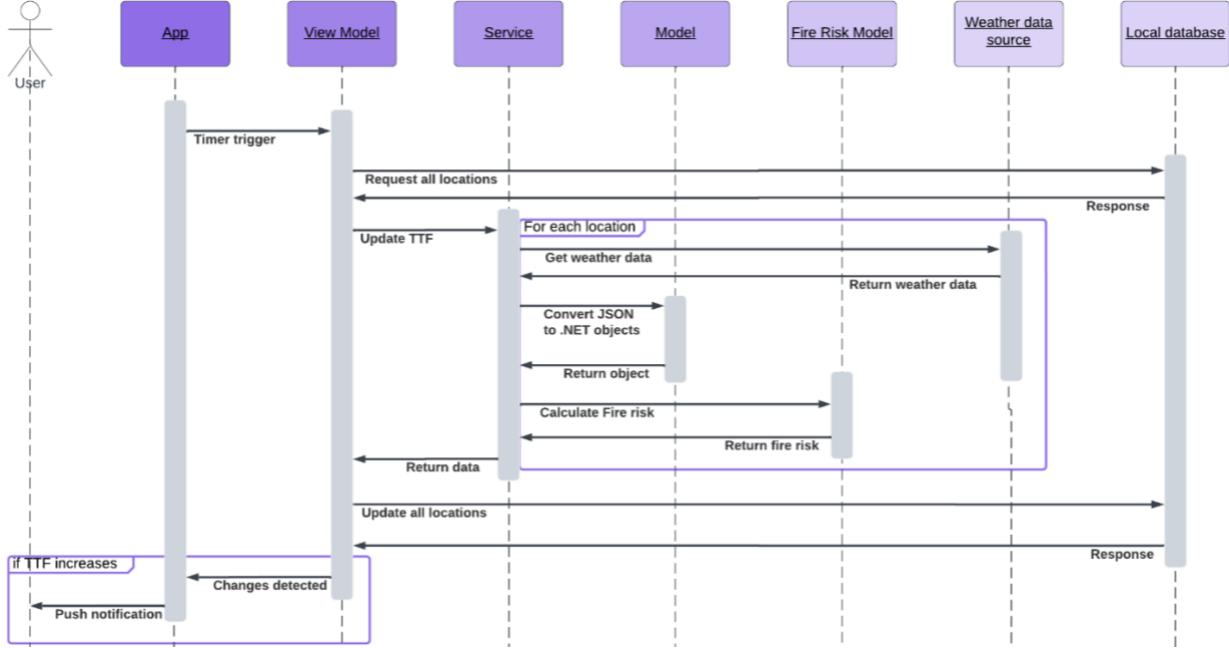


Figure 2. Sequence diagram for Push notification

The *Add location* sequence diagram illustrates how all components interact when a new location is added. The View Model component triggers the Service component and is responsible for updating the newly created object to the database before notifying the View. The Service component is responsible for retrieving and converting the response from JSON to .NET objects and in addition, calculating fire risks for the new location.

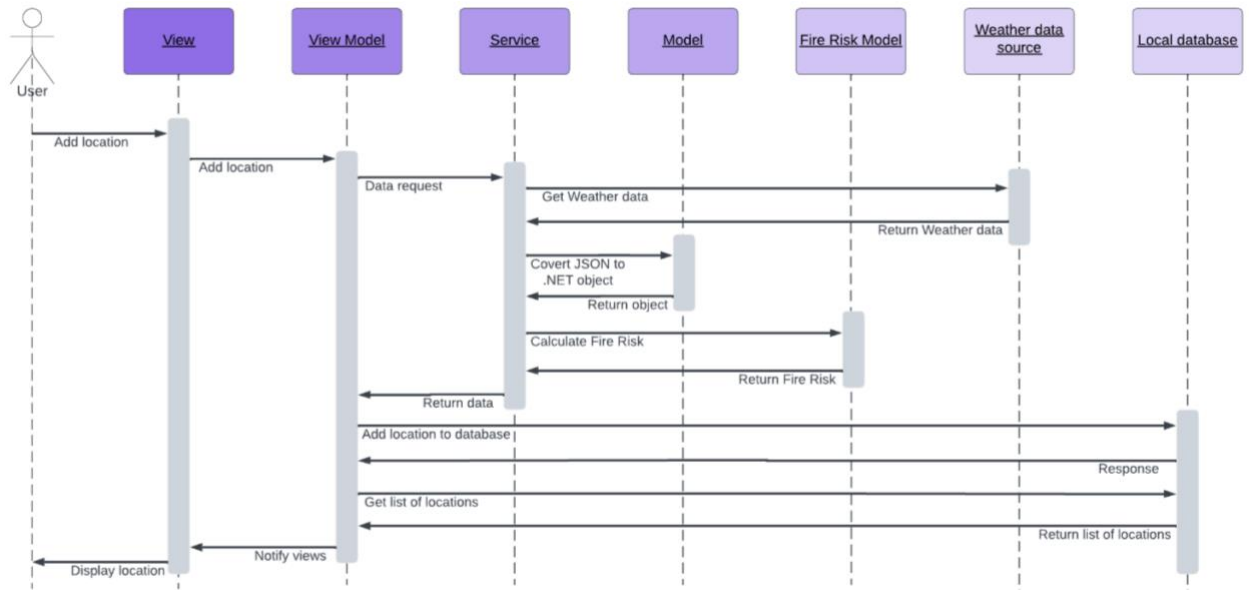


Figure 3. Sequence diagram for Add location use case

It is thought that the *Push notification* gets triggered by a timer, which will utilize the View Model component to retrieve all locations from the database. The View Model uses an instance from the Service component and updates the fire risks. Eventually, the mobile application will notify the user if the fire risk is increased.

Xamarin

An initial requirement the project owner had set was that the application must be accessible and available from either of the two major mobile operating systems: Android and iOS. Xamarin was therefore chosen.

Xamarin is an open-source platform for developing modern and responsive cross-platform mobile applications. It is built on top of .NET, which automatically manages tasks such as memory allocation, garbage collection and interoperability with underlying platforms [1].

Xamarin uses C# as its programming language and provides facilities for direct invoking of Objective-C, Java, C and C++ libraries giving the developer the power to use a wide array of third party code [1].

A transition from Xamarin to MAUI must eventually be made, because Microsoft is planning to end updates to the Xamarin mobile app development platform in November 2022 [2]. Concerning the fact that MAUI will over time replace Xamarin, the probability of having to rewrite the Xamarin app to meet MAUI standards is seen as most improbable. The process of migrating the application over is expected to be small amounts of code changes [3].

Domain Model

A Domain model is a type of flowchart that aims to show the real-world flow and relationships between entities and concepts. The model helps various domain experts get a solid understanding and agree upon the system [4]. Figure 4 illustrates the different concepts that exist within the mobile application.

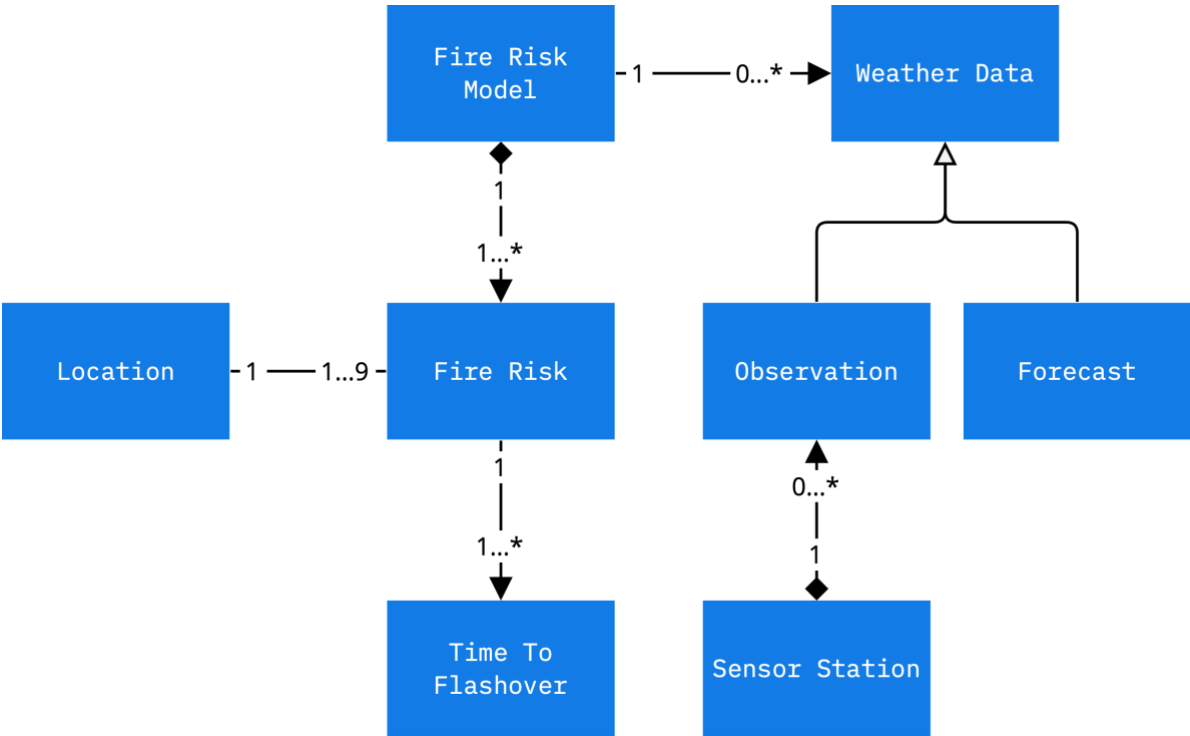


Figure 4. Domain model for the project

Two compositions [5] exist in the model, one from Fire Risk Model to Fire Risk and one from Sensor Station to Observation. This injection means that the entities Fire Risk Model and Sensor Station determine Fire Risk and Observation's lifetime. For instance, a Fire Risk cannot exist without a Fire Risk Model. Weather data can either be an Observation or a Forecast, depending on what type of weather data is requested.

Relations between all entities are defined, and in most cases, they follow “one to many principle”.

Prototypes

5.1 Graphical User Interface Design

The project owner was heavily involved in the design and planning of the Graphical User Interface (GUI). The GUI was in constant development and was designed to be adaptable based on the project owner's wishes and feedback from the evaluation.

The project owner visualized his ideas of the GUI by drawing prototypes of the different pages, the interpretation of these prototypes was later programmed.

5.1.1 Wireframe

After the first meeting, some wireframes were made to function as bases of how the group visualizes the design of the application. These were sent and presented to the project owner who provided feedback accordingly and at a later point sent new prototypes (see Figure 5).

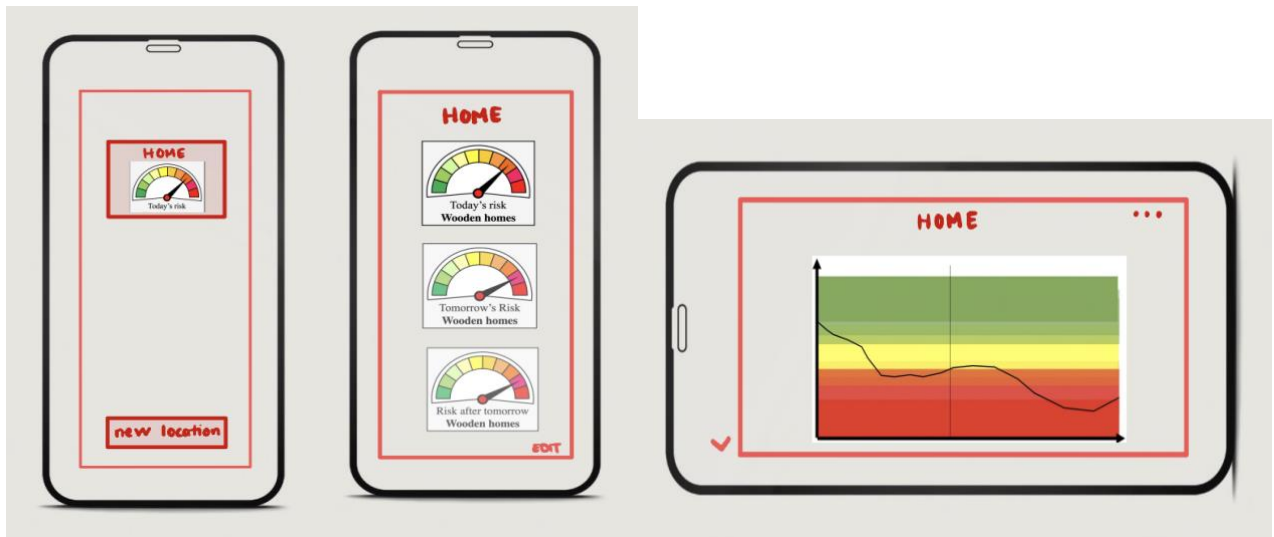


Figure 5. WireFrame prototype

5.1.2 First iteration

After having presented the wireframes and programmed a prototype of the wireframe, the project owner sent a new prototype of the application (see figure 6). This was again programmed (see figure 7). The difference between this prototype and the prototype from the wireframe was that this design also highlighted the risk for tomorrow, the risk in two days and finally the risk in three days. It was decided that the speed-o-meter will no longer be used to display the concepts of TTF, this was decided after some discussions with the project owner and the project supervisor.

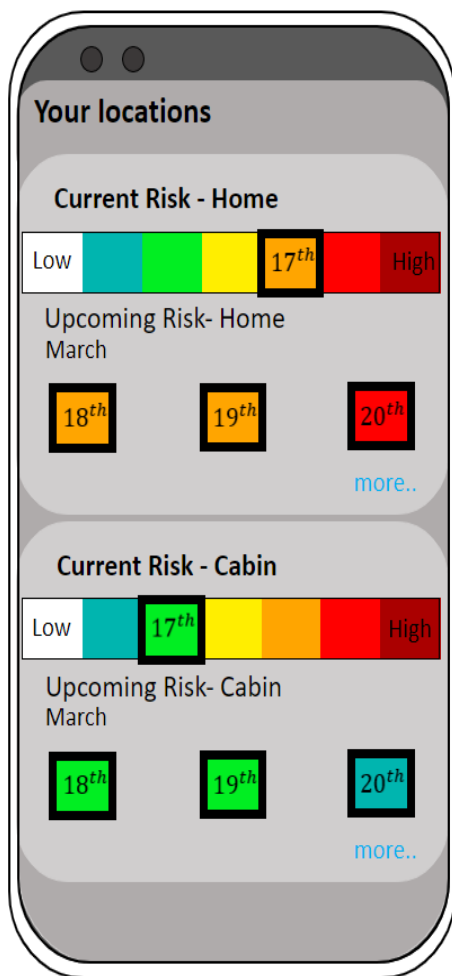


Figure 7. Prototype suggestion from the project owner

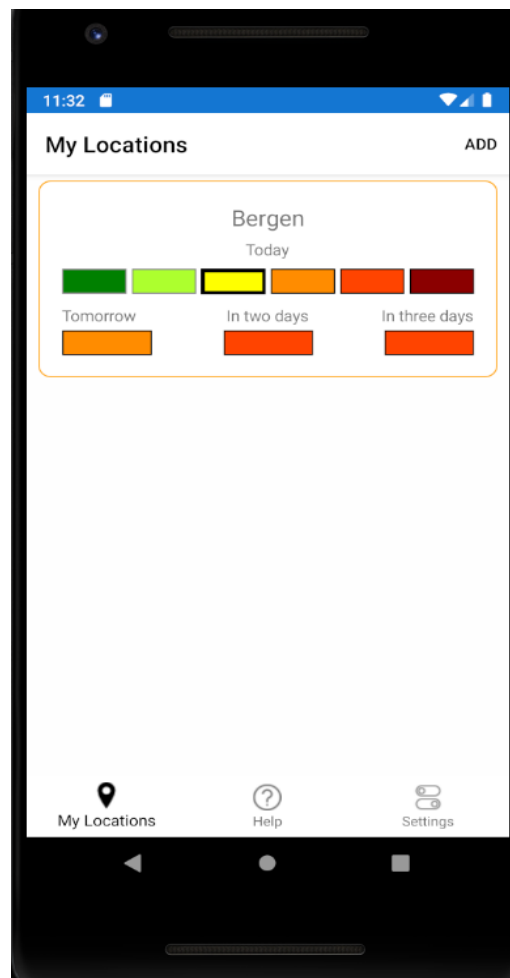


Figure 6. The group's response to the first prototype suggestion

5.1.3 Final design iteration

After the second iteration was programmed, the project owner sent a new prototype. It was decided that this prototype will be the base of the final version of the graphical user interface. This decision was made along with the project owner and the supervisor, the decision was mostly lingering on time. The project owner visualized the dashboard for all the locations by displaying a list of the locations (see figure 8). The in-depth view of the location (see figure 9) included the same color scheme which is used by MET to display risk for forest fires. The idea is that the user will enter this page when pressing on the desired location.

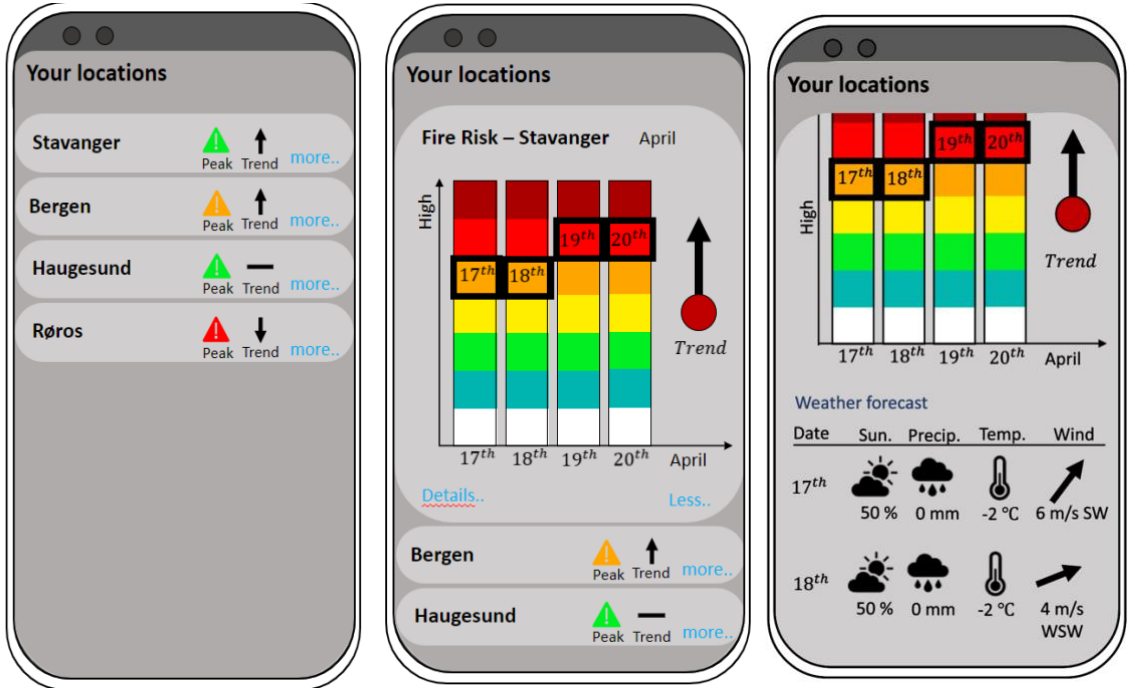


Figure 8 Final suggestions for the dashboard and location view

After having received the final prototype of the desired graphical user interface, the prototype was programmed. There were some adjustments in the design that was programmed, these were made in accordance with the project owner. The Graphical User Interface programmed by the group included the parameters wind in addition to a link to YR.no which displays the weather of the same location. Figure 9 displays the dashboard and the presentation of fire risk for a specific location.

Figure 10 shows the updated GUI, where the term IFD is used.

Figure 11 shows the evolution of the mobile application.



Figure 9. The group's response to the project owner's suggestion for GUI

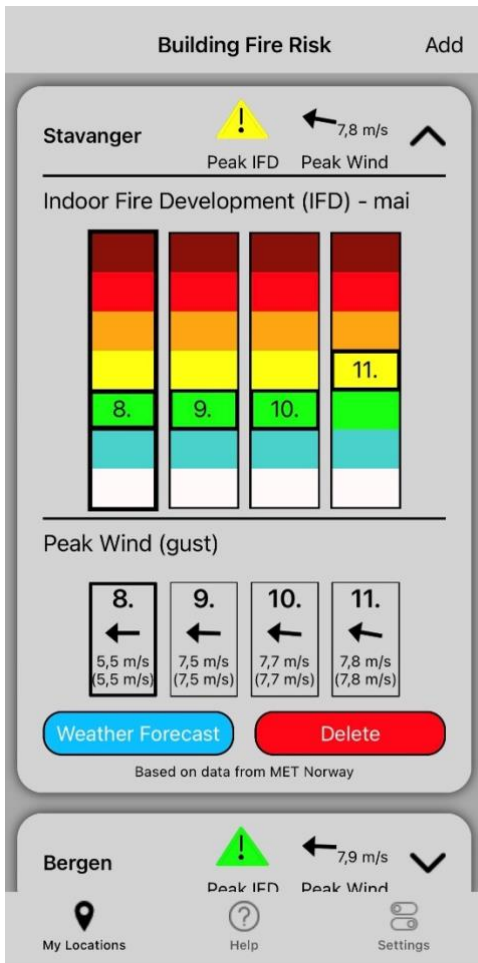


Figure 10. The last iteration, with the use of IFD



Figure 11. The evolution of the GUI

Bibliography

1. *What is Xamarin?*, in *Microsoft Documentation*. 2021, Microsoft.
2. Ewbank, K. *Microsoft Will Replace Xamarin Forms With MAUI .NET* 2021 [cited 2022 08.03.2022]; Available from: <https://www.i-programmer.info/news/89-net/14762-microsoft-will-replace-xamarin-forms-with-maui-net.html>.
3. Microsoft. *Migrate your app from Xamarin-Forms*. 2021; Available from: <https://docs.microsoft.com/en-us/dotnet/maui/get-started/migrate>.
4. Gibbs, M., *Domain Modeling: Definition & Examples* 2017.
5. Bandarupalli, K. *Domain Model Using UML*. 2009; Available from: <https://www.techbubbles.com/softwarearchitecture/domain-model-using-uml/>.

Table of Figures

Figure 1. Use case of the application 2

Figure 2. Sequence diagram for Push notification 7

Figure 3. Sequence diagram for Add location use case 8

Figure 4. Domain model for the project 10

Figure 5. WireFrame prototype 11

Figure 7. The group’s response to the first prototype suggestion 12

Figure 6. Prototype suggestion from the project owner 12

Figure 8 Final suggestions for the dashboard and location view 13

Figure 9. The group’s response to the project owner’s suggestion for GUI 14

Figure 10. The last iteration, with the use of IFD 15

Figure 11. The evolution of the GUI..... 15