

Final assignment

62550 User Experience And Mobile Application Development

Group 8

Veteranlegend



Ahmad
El-Hag
S210030

msbeih



Muhammed
Sbeihi
S210085

Moe241



Mohamad
El-Asadi
S224301

radimib



Mahdi
Ibrahimi
S210077

xMire0



Faruk Emir
Degirmenci
S222359

Ahsunalla



Ahsunalla
Wahidi
S184858

Contents

Contents	1
Introduction	2
Requirements	3
Analysis	4
Iterative design	4
Usability	4
Design	6
Navigation Diagram	9
Implementation	13
BPMN Diagram	14
Code Snippets	14
Appendix	15
Link to github repository	16

Introduction

In a world full of surprises, the weather is no exception. Constantly being surprised by the weather has become a normal state for the billions of people who share this blue planet of ours. But why has being surprised by the weather become a priority, rather than making use of a simple program on a device, such as a phone or computer? In this report, we discuss exactly that, how we have come to create such an app, the obstacles we have faced, and the methods and designs taken into consideration.

This report captures our journey of designing and implementing an app that is user friendly and stands out from other weather apps while still implementing the given requirements set for the project.

Noteworthy features that also fall under the category of the requirements for the project include showing today's forecast, hourly forecast, weekly forecast, forecast for at least 10 different cities, adding cities as favorites, and caching. These are just some of the features, the report will further discuss the features and design decisions of the weather app as we dive deeper into the report.

These features might seem like features you may find in every ordinary weather app laying around out there, but it is not that our features are different that makes our app stand out from the rest, it is our choices when the topic of interaction design comes up. We have found it very common in weather apps to not be very user friendly and messy when it comes to how and where the data is shown. For that exact reason, we decided that our app was going to be different. We wanted an app where the information was at the tip of the fingers and no scrolling down was needed to access information.

Weather uncertainties strive in our day to day lives, but with an app where access to information is no further than looking outside the window, we have provided a look into our determination and strive for this project. As we move on and unfold the layers of this project, we ask you to navigate through our decisions when it comes to design and user friendliness, and last but not least, our decisions when it comes to interaction design.

Requirements

Through the process of making the weather app we had to take into account what requirements we had to fulfill, not only that but also to what extent we were able to fulfill the requirements and what requirements we weren't able to succeed with.

Firstly we had to implement the forecast for the day, meaning we had to implement or display the current weather with a minimum and a maximum expected temperature, name of the city, conditions for the day, like cloudy, rainy etc. and day of the week.

The Second requirement that we had to implement was the hourly forecast, meaning displaying forecasts for all the hours in the current day with temperature, humidity and condition.

Thirdly we had to implement the weekly forecast, meaning we had to be able to display a list with the expected weather for the next 7 days, including minimum & maximum temperature, conditions and the humidity.

The fourth requirement was allowing the selection of at least 10 cities, and showing today's and weekly weather forecast.

Fifthly we had to, on the basis of the earlier requirement, be able to mark some cities as favorites, and allow a quick way of switching the forecast between them.

The sixth and final requirement was being able to cache the forecasts that were opened on the current day and allow them to be accessed when the device is disconnected from the internet.

From the first requirement we managed to implement all of the features. In the requirement we had to implement the forecast of the day along with the minimum and maximum temperature, which we have done, but not in the good old fashioned way with the API. Since our API didn't have the data for minimum and maximum temperature, we chose to instead just take the lowest temperature of the day along with the highest temperature and set the data as the minimum and maximum temperature.

The rest of the requirements are all successfully implemented, and the features are all visible and available on the weather app.

The way we could fix the problem we stumbled upon this far into the development phase would be incredibly difficult. We would have to swap our API with another API that includes the data for minimum and maximum temperature along with the rest of the data that we need and the API has to be free.

Analysis

When it comes to user experience, we have taken several things into consideration. such as an iterative design process, usability and visual design.

Iterative design

In our iterative design phase, we created several interactive prototypes of our weather app using the Figma program. Each time we created a prototype we got a response from our professors or teaching assistants on how we could improve our design and app. The entire process laid the foundation of everything we have today.

We didn't quite have enough time to test our app on people not related to the project, so we decided that we would move on and judge the weather app based on our own instincts and what we have been told by our professors and teaching assistants.

Usability

Usability is a major factor when developing an app, for that reason, we decided that our app was going to be a single page where everything would be accessible and there would be no need for scrolling or page swiping, even though we were told otherwise by one of our teaching assistants. Being users of weather apps ourselves, we knew what our weather should look like design wise, what features we were unhappy with in other weather apps and we were pleased with our own thoughts on how the final version of our weather app came to look like.

We made everything reachable by a single gesture of the finger, because of the need to quickly access data and not always wanting to go through a whole process of swiping, scrolling and clicking. We also came to the conclusion that we are going to skip all the fancy boxes and drawings of suns and clouds, and instead have it as two simple columns one above the other, with all the data the user needs, so they can go on with the rest of day knowing, what the weather would look like and what there is to be expected. Although it is nice and pretty with all the fancy pictures of the sun and clouds and the rest of the condition states, it's quite time consuming to look around at all the things there is happening at the same time on the screen, so we chose to skip it entirely, even though they were part of our design in the early stages of the project. Decisions come with sacrifices, and so did ours. Because we decided to have only two columns, with one containing the hourly forecasts and the other, the weekly forecasts where we had to place the humidity, day or time, conditions and degrees all in the columns as well, and for that reason it may look a bit squished together and might even look clumsy, but that is a sacrifice we felt was necessary, in order to get as much information in the shortest time possible. The navigation flow in our weather app also primarily follows the usual standard navigation flow of the other big tech companies. We made it so that the daily forecast, along with the hourly and weekly forecast would be the first thing that popped up on the screen when opened, and the menu button would contain the rest of the cities and the favorite cities. By doing it in that formation, we have instilled our own touch upon the design and yet not diverged too far from the usual designs of weather apps, and we have assured ourselves of a great design by doing that. We decided that for a weather app some type of weather condition was going to be in the background. We wanted a moving and changing condition depending on the weather, but we quickly realized that time was of the essence and we weren't going to make it, if we focused on the smaller design components instead of the actual logic and data of the app. Therefore we chose one weather condition which we thought was going to suit the app and also give an uplifting and hopeful feeling to the user. Taking our teaching assistants' feedback into consideration, we decided to make the text for our entire app, the color white, which we also think suits our app the most.

Design

We decided early on that we would have a weather condition as the background for our app, we wanted the condition to be ever changing depending on the weather, but unfortunately we didn't have enough time and we had to choose a stand still picture of a weather we found uplifting and good looking.

Down below is how we imagined it would look like in Figma:

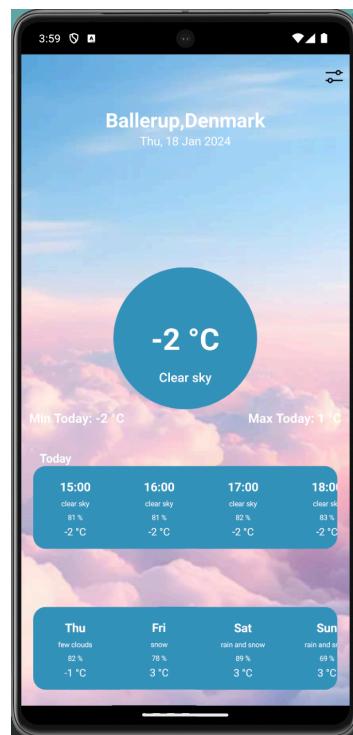


We drifted off from the idea of having a bar that would be long and the user would have to scroll down, the reason being that it would be troublesome for the user to look around for the data, and it would be easier to just have simple bars containing all the data and information the user needs.

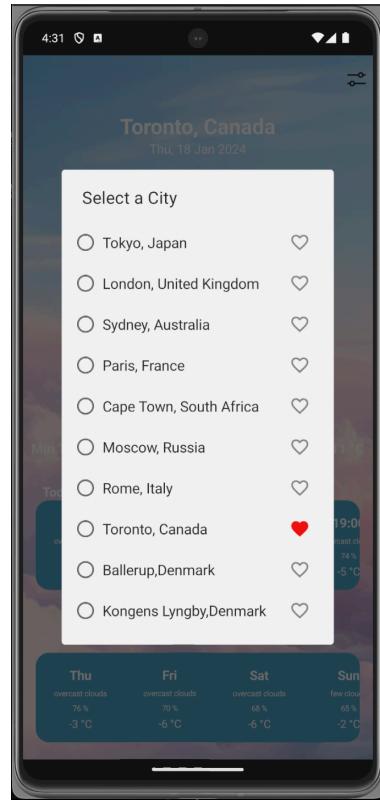
Down below is how we imagined the design for the weekly forecast bar:



And here is how it ended up looking in the final stage of the project:



We also decided that due to not having enough time, we would not be making a search bar, and instead have a bar with all the cities where you would also be able to then chose the city as favorite:

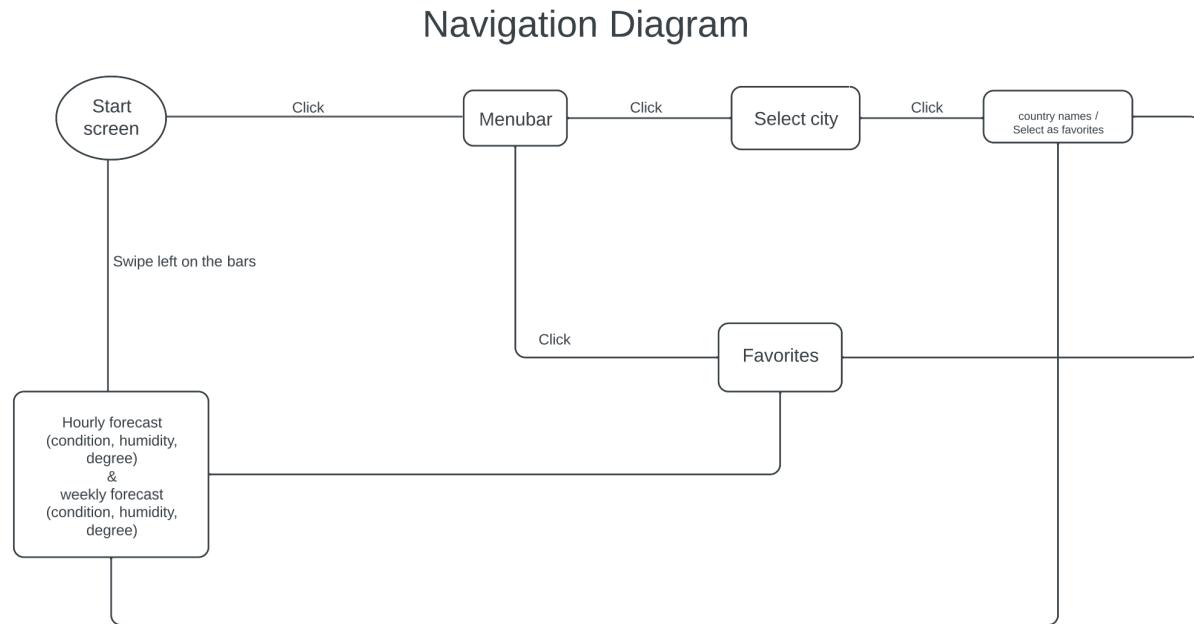


This is how the design of the favorite button and the search button looked in the design phase in Figma:

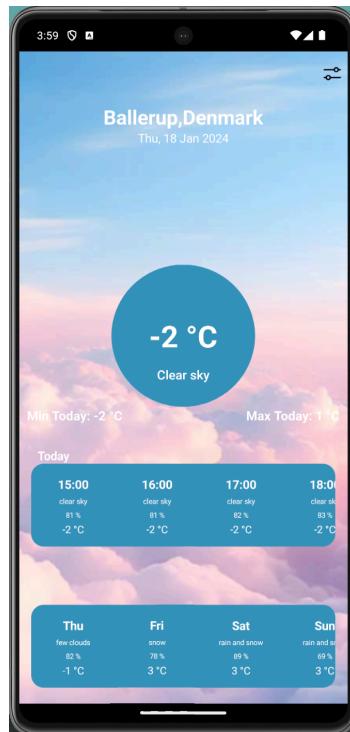


Navigation Diagram

Down below is the navigation diagram for our weather app showing the flow and choices that can be made throughout our app.



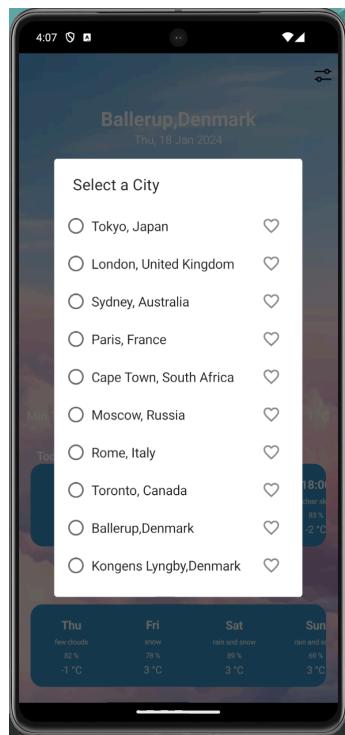
This is the start screen as seen on the navigation diagram and the Hourly/Weekly forecast satisfying the two boxes on the left in the navigation diagram:



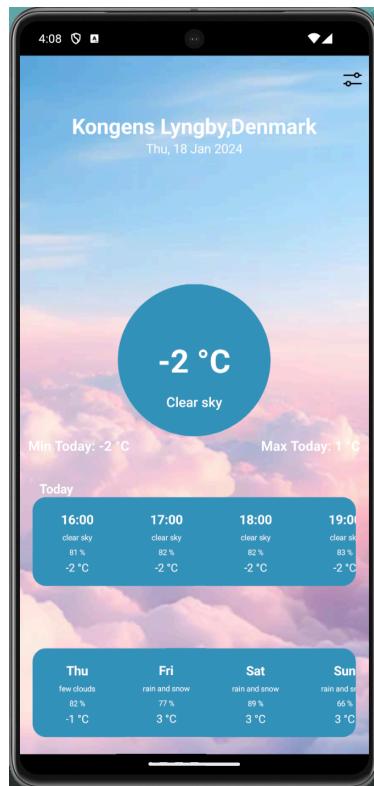
Down below is the option to click on the menu bar as seen on the navigation diagram where you will then be shown two options to either chose select city or favorites:



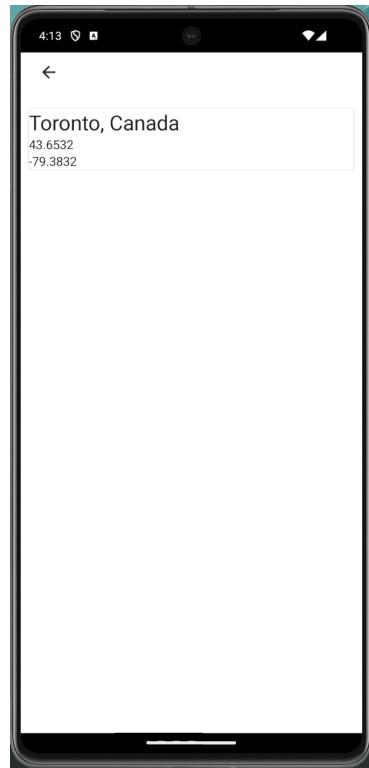
Down below is what is shown when select city is pressed, where you can chose a city as favorite and the ten cities are shown, and when a city is clicked on as seen in the navigation diagram, you are the shown the homescreen with the hourly and weekly forecast along with today's forecast of that city:



Kongens Lyngby is chosen and then displayed:



If favorites are chosen instead of select city, you will then be shown a list of your favorite cities, which you can chose and see the data for that city as seen down below and in the navigation diagram:



Down below is the favorite city that is clicked on:



Implementation

For the weather apps, we used MVVM, which is a design pattern architecture and is often used in software development. MVVM stands for “Model-View-ViewModel”.

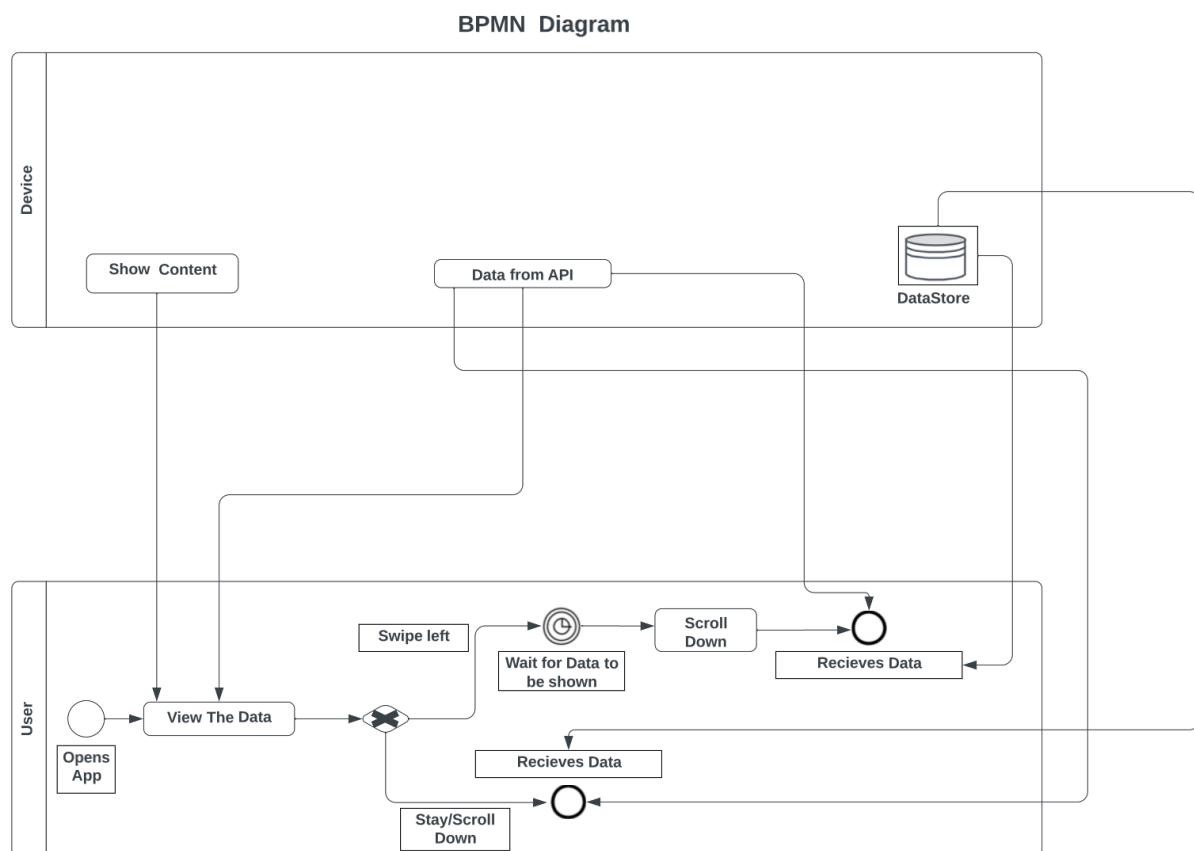
Model is responsible for managing data and notifying the ViewModel of any kinds of changes in the data and thus allowing for real time updates in the user interface.

View represents the user interface, and displays the data that is provided to it by the ViewModel. View is responsible for showing or presenting the information to the user of the app and also capturing the input from the user.

ViewModel is like a middleman between Model and View. It exposes data and what commands to be displayed and to be executed in the view or user interface.

We have in our project as stated earlier, made use of the MVVM. We have two specific folders only for that reason. In our “app” folder we have the View, being the user interface and everything the user sees and interacts with. In our app, we also have the file called “MainViewModel” which is our ViewModel.

BPMN Diagram



Code Snippets

There are many important pieces of code in our project, but a few could be the function “fetchOneCall”:

```

88     fun fetchOneCall(lat: Double, lon: Double){
89         viewModelScope.launch {
90             repository.fetchOneCall(lat, lon).collect { result ->
91                 _oneCallWeather.value = result
92             }
93         }
94     }
    
```

What this function is responsible for is making an asynchronous call, it does so to fetch data with the use of the “repository.fetchOneCall” method and updating “_oneCallWeather” stateFlow with the result.

```
96     fun saveWeatherData(context: Context, data: OneCall) {
97         val sharedPref = context.getSharedPreferences("WeatherApp", Context.MODE_PRIVATE)
98         with(sharedPref.edit()) {
99             putString("weatherData", Gson().toJson(data))
100            apply()
101        }
102    }
```

The function above has the responsibility of saving weather data locally, it does so for offline functionality, and the saved data can then be retrieved with the “getSavedWeatherData” function.

```
51     fun saveCityFavoriteState(cityName: String, isFavorite: Boolean) {
52         viewModelScope.launch {
53             favoriteRepository.saveFavoriteState(cityName, isFavorite)
54             // Update the state
55             val currentFavorites = _favoriteCitiesState.value.toMutableMap()
56             currentFavorites[cityName] = isFavorite
57             _favoriteCitiesState.value = currentFavorites
58         }
59     }
```

The function above is managing the favorite state of cities, storing that information and keeping the ViewModel’s state updated for UI observability.

Appendix

Due to some problems with GIT we had to make a new repository. For transparency reasons this appendix will detail each group member's contribution in regards to the code:

Today's forecast:

This was primarily Faruk Emir Degirmenci.

Favorites:

This was primarily Faruk Emir Degirmenci.

Hourly forecast:

This was primarily Ahmad El-Hag

Weekly forecast:

This was primarily Mohamad El-Asadi

Forecast of different cities:

This was primarily Ahsunalla Wahidi.

Caching:

This was primarily Mahdi Ibrahimi & and Muhamed Sbeihi

Min/Max:

We pair-programmed for this part.

API

We pair-programmed, so we all contributed more or less equally to this section of the code.

Miscellaneous

We pair-programmed for this part.

Link to github repository

<https://github.com/Ahsunalla/WeatherFinal>