EXWeather Android Application

Candidate Number: 109527

March 15, 2022

1 Introduction

The application that this writing will take a further reflection of is the "EXWeather" application for android devices. This application is intended as a multi functional weather application which uses the most up-to-date API technology, aiming to incorporate both a seamless user-experience(UX), and display informative data about desired weather and climate information.

The main reason for this software's initial development and further production is because I have found it difficult to find a weather application on both desktop and mobile devices that fits the needs that I had wanted it to. To quickly and clearly display weather information from a public API, and also have the ability to display a country's weather information with a few simple clicks. To be able to have an application that does not include so-called "bloat features" but that also has a clear application for a few simple purposes. A paper by Chong Law, Roto and colleagues through a survey on UX found that "Respondents tend to agree on a concept of UX as dynamic, context-dependant and subjective, which stems from a broad range of potential benefits a user may derive from a product" [1] This ignited the initial idea of the project, as to how I could create a weather application. An application that grants positive potential benefits to the user, mainly being its simple straightforward purpose of presenting weather information clearly and quickly.

2 Description and Design Rationale of app components

The EXWeather application has implemented a plethora of both android development principles and fundamentals which work cohesively together to produce a functioning application. Some key features that had to be met were that multiple screens allowed for the user to navigate between screens, using both explicit and implicit intents, having menus, utilisation of shared preferences and recycler views, use of the internet being an API call as well as the application functioning whether the device has an internet connection or not. Each of these were met and designed with intention.

2.1 Front-end Elements

The first design element that had to be considered was the use of multiple screens, giving the user the ability to navigate and switch between them. The method of which the user would navigate through screens were through button implements which would call and create new activity intents, and have the "settings" and "multiple country view" pages be a sub-child of the main application screen. The intention behind this was so that the user always had a main landing page where they could return back to by either pressing the back button on their device or the in app back button that was implemented.

A unique design choice that was made for EXWeather was that an initial splash screen was created. This screen showed off the title of the application and a small mission statement of what the application aims to do. The reason for doing this was that in terms of the users UX it allowed them to see what the application was about and it gives them more interaction with the application as they can see the first screen allowing them to proceed into the actual weather application. In terms of how the layout elements were placed on the activities screens, it was entirely dependent on the content that was being shown. For example a linear layout was used for the recycler view as it was the most effective way to present the data. Whereas the initial splash screen home page utilised a relative layout.

Working in tandem with the multiple screens that the user could navigate through, a menu at the top right hand side of the screen was implemented. The menu had two options that would create intents and help the user nav-

igate to either the “Multiple Countries View” or the “Settings” screen. The design decision of keeping the menu as a simple drop down menu opposed to floating buttons is because it helps keep the user’s UX, allowing them to easily navigate without hassle, as further optional pages are kept off screen rather than having large space being taken up on the main screen.

Furthermore the recycler view screen which can be accessed through the menu at the top of the screen which displays a list of countries which are put into a linear layout. The choice for a recycler layout in this activity was because it was the easiest and most effective way to present a long list of data that could be scrolled through by the user. And in contrast to a table of data, the recycler view allows the user to tap on one of the countries to save that location which can then be used later to request that country’s weather and climate.

2.2 Back-end Elements

The design approach that was used for allowing the users to navigate between pages was done through the use of intents. When a button or menu option was clicked an intent was created to help navigate the user to the desired page. Both cases of explicit and implicit intents were used when intents extra string data was passed through to one of the pages.

However, data that needed to be accessed across multiple screens was done through user shared preferences. The idea to use shared preferences over storing the data locally to the device in a text file was that the data was user specific and saved its state across different sessions, as well as allowing for default data to be present. The user shared preferences manager was also used multiple times to access and set the countries default location, both manually on application startup as well as by the user depending on which recycler view option was chosen, or through the settings page where the user could type in the desired country the API called from.

In order to show weather information of a country the app would require both internet access and for the app to see if the internet was connected. If the two were met then the application would be able to call weather data from a desired location. And due to good programming practices whether or not the internet is connected or the application has access, the ExWeather application would still function, just not allowing the API calls to go through. This is checked by try catch statements and exception han-

dling.

The api that was chosen was the “https://www.weatherapi.com/login.aspx”. This was due to the fact that the API call would return an easily readable JSON file, which could be converted to a json object then was read into the application. The API chosen would also allow for a large amount of API call requests per hour which would mean the user is not limited to how many times they could use the application.

3 Development Process Reflection

Looking back at the planning stage of the EXWeather android application, I had a clear idea of what the application was going to do and its intended purpose. This was more clearly defined as I continued along with the project. A small mind map was created which showed the direction I wanted to take the application. Hence this helped guide me as to what the app would look like and the features and design choices I would implement. However, when planning I did not have a clear direction on the api that I was going to use which should have been chosen in the early planning stages to save time on development.

Before starting the product development process I had sketched some light drawings of how the different screens and activities would look. Having a general plan of what the buttons elements, menus and text views would look like. This would then make it easier to design the code layouts of the activities during the next implementation stage. However I was limited during the design process as I was new to the android studio environment and frameworks. This meant that the large ideas of animations I had within the application had to be scrapped and my scope became more narrow due to my programming skill set.

Finally when development started I had a clear programming procedure that I had put in place. Being the proper implementation of github and coding practices. By having a clear timestamp of my work being completed I could look at my github commits, in addition to clear and concise coding comments to where I could easily pick off where I had started the day before. Although an issue that had come up because of me not planning the API I was going to use in the earlier, I had wasted many hours trying to get different weather API’s to work when I should have found one during the planning stage. If I were to continue to develop another app or extend the

life cycle of EXWeather there are a few key components which I would work upon further.

4 Future of EXWeather App

Looking into the future of the EXWeather application, if I had more time to develop the application I would make the UI of the application look more aesthetic in addition to making the flow of the screen activities more coherent. Such as the ability to swipe left and right to change screens rather than having to use a menu option. Another key feature that could be implemented is to allow the user to select what they want the api to display to them, rather than just showing them the temperature and humidity. This would be easy to implement and give an increased user UX. Lastly, another quality of life feature for the user would be to enhance the settings options. A large feature that I would implement first is to allow the user to change the API that the weather app calls from. This would mean if the API goes down the user can take it in their own hands to make another API work, this would not only allow the EXWeather application to be more functional under dynamic conditions but also give the user more transparency about how the application they are using is working.

Bibliography

References

[1] Effie Lai-Chong Law et al. "Understanding, Scoping and Defining User Experience: A Survey Approach". In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. CHI '09. Boston, MA, USA: Association for Computing Machinery, 2009, pp. 719–728. ISBN: 9781605582467. DOI: 10.1145/1518701.1518813. URL: https://doi.org/10.1145/1518701.1518813.