Mobile Computing

CSCI 5708
Fall 2018

ASSIGNMENT 3

Gurjot Singh (B00811724)- gurjot@dal.ca

Project Overview

The task was to develop a weather application with the help of JSON and Android Weather API and perform the application of Neilson's usability heuristics. The application required a network connectivity and the weather details were fetched of a default city set as "Halifax" on its initial start. Users can then provide a city in the space provided and press the button "Get Weather Details". Weather is the state of the atmosphere, describing, for example the degree to which it is hot or cold, wet or dry, calm or stormy, clear or cloudy[1]. The android application will then provide all the weather details of the provided city such as Temperature, Minimum Temperature, Maximum Temperature, Humidity, Weather Description and many more. The Android Weather API is an "open weather map" API which can be called by city name. API responds with a list of results that match a searching word[2]. The results would be then imported using JSON Objects and displayed in the application.

Software Design and Architecture

The first step in the development of the application was its wireframe. The wireframe provides a potential visual guide to an application. It is also known as a page schematic or screen blueprint created for the purpose of arranging elements to best accomplish a particular purpose[3]. The wireframe used in this project is shown in Figure 1.

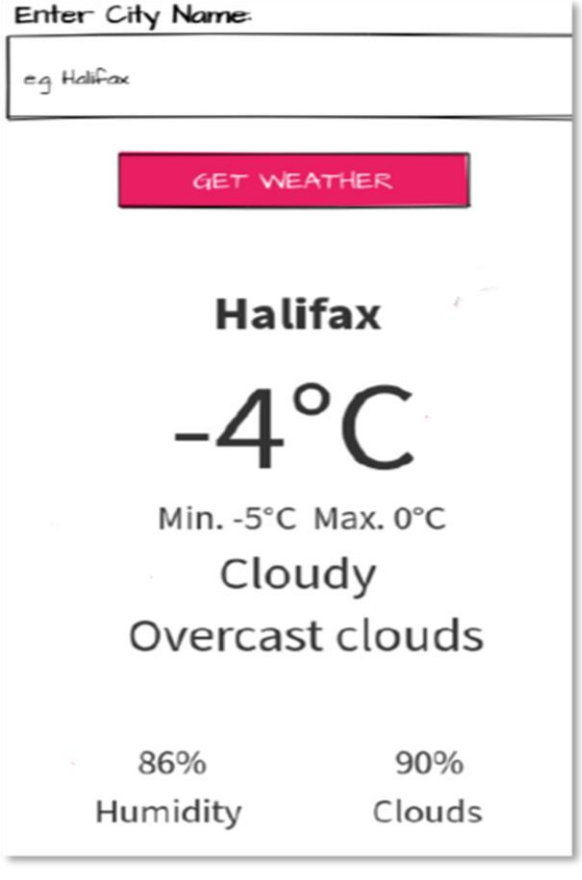

Figure 1 : Wireframe

Various “Textview” and a “Editview” were used to develop the application. The button “Get Weather” would help the user to fetch the details of the city provided. The design was kept simple and minimalist so that users do not have any issues with understanding the weather details. The details have been fetched from the API and converted to a readable format. The complete data is then displayed in the application developed according to the wireframe.

Implementation Issues

The application was developed by fetching the weather details of a provided city from the “Open Weather” API. The results fetched from the API were in a JSON format. While fetching the details from the JSON file, extracting the data was a bit difficult since all the data was not being fetched from array in JSON.

Solution:

A JSON array object was then created which made it simple to fetch the details from the JSON file[4]. There were no more major issues encountered.

Testing

Methodology

The application is always successful only if it has undergone testing. The Weather application had been tested with various aspects to minimize the possibility of errors and maximize efficiency. The Neilson’s Heuristics were applied to all the aspects of the project wherever they were required.

Test Plan Use Cases

The Weather application was tested with various test cases and it provided the outcome as expected. The test cases are provided in Table 1.

Test Cases	Possible Outcome	Actual Outcome
Button “Get Weather” Functionality	The button should fetch the city details provided by the user and provide the required action.	The button fetches the city details provided by the user and provides the required action.
Default City Details	The application should provide the city details of a default city.	The application provides the city details of a default city as shown in Figure 2.
Error Handling for Incorrect Cities	The application should provide a specific error message for entering a city that is invalid.	The application provides a specific error message to the user to provide a correct city name when the entered city is invalid as shown in Figure 3.
Error Handling for Connectivity Issues	The weather application works only if the device has a working internet connection. A proper error message should be provided if the device has connectivity issues.	The application provides a proper error message stating that there is no internet connectivity available as shown in Figure 4.
Accuracy in details	The application should provide accurate weather results for all the cities.	This application provides accurate results for all the cities as shown in Figure 5.
Efficiency in Handling in Multiple Cities	The application should provide accurate results for multiple cities entered by the user one after the another and remove the details of the previous cities also.	The application provides the weather details of the current city entered by the user and removes the details of the previously entered city.

Table 1: Test Cases

Screenshots from the Application while Performing Testing

Figure 2: Default City "Halifax" Results Displayed

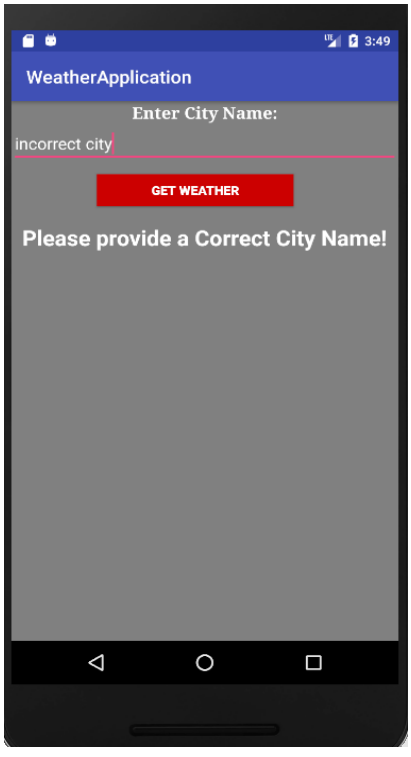

Figure 3: Incorrect City Error Handling

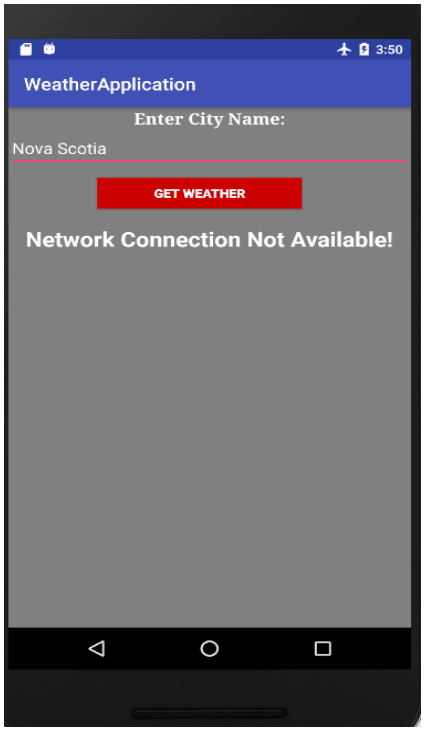

Figure 4: Network Connectivity Error Handling

Figure 5: Accurate Results

Nielsen's Usability Heuristics

This application was designed in such a way that the Nielsen's Heuristics were completely satisfied. The Weather application provided the "visibility of system status" by either providing the weather details or the error messages to the user. The weather details are converted and have been displayed in a manner that there is a proper match between system and the real world. The temperatures were displayed in "Celsius" with a proper readable text. Error prevention was done to all aspects. If there would be "Internet connectivity" issues in the device, error will be displayed both at the start up or in between the application. "Aesthetic and Minimal Design" was followed in the development process. No irrelevant information would be displayed in the description for the weather in the application. Proper error messages have been provided so that the users can easily diagnose and recover from the errors. Some of the heuristics applied are shown in Table 2.

Nielsen's Usability Heuristics	Possible Outcome	Actual Outcome
Minimalist Design	No irrelevant information is displayed in the application when fetching the details of a city.	No irrelevant information is displayed in the application when fetching the details of a city.
Help users recognize, diagnose, and recover from errors	The application should provide proper error messages for the application if any component of the application fails.	The application provides proper error messages for the application if any component of the application fails.
Match between system and the real world	The application should display the weather details in a natural and logical order.	The weather details are converted in Celsius and displayed.
Visibility of system status	The application should keep users informed about the status of the application.	The application displays the proper weather details when user enters a city and also provide the error messages in case of invalid cities.

Table 2: Nielsen's Usability Heuristics

References

[1]"Weather", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Weather. [Accessed: 03- Nov- 2018].

[2]"Current weather data - OpenWeatherMap", *Openweathermap.org*, 2018. [Online]. Available: https://openweathermap.org/current. [Accessed: 05- Nov- 2018].

[3]"Website wireframe", *En.wikipedia.org*, 2018. [Online]. Available: https://en.wikipedia.org/wiki/Website_wireframe. [Accessed: 05- Nov- 2018].

[4]M. Bilal and M. Bilal, "JSON Parsing in Android using Android Studio | MobileSiri", *MobileSiri*, 2018. [Online]. Available: https://mobilesiri.com/json-parsing-in-android-using-android-studio/. [Accessed: 05- Nov- 2018].