CSCI 5708

Weather Application

MOBILE COMPUTING ASSIGNMENT 2

MEGHNA RAMACHANDRA HOLLA | B00812604

Overview

I created a weather application using the OpenWeatherMap API. I used Volley library to handle the HTTP request and response. The weather application is easy to understand and implement. I have followed Neilson's Usability Heuristics to make the application user-friendly.

Application Design

My application has a welcome screen that displays a message – 'Good day!'.

The application runs on full screen like a conventional weather application does. The keyboard that appears while typing the city name disappears on clicking the 'What's the weather?' button and enables the user to see the content clearly.

It has a textbox that allows a user to type in a valid city name. The text box has clear instructions with example to make the user understand what to type.

Image that represents the current weather is displayed using OpenWeatherMap API. Using the image icon ID that is in the OpenWeatherMap end-point, I have displayed images of the current weather using Drawable resource.

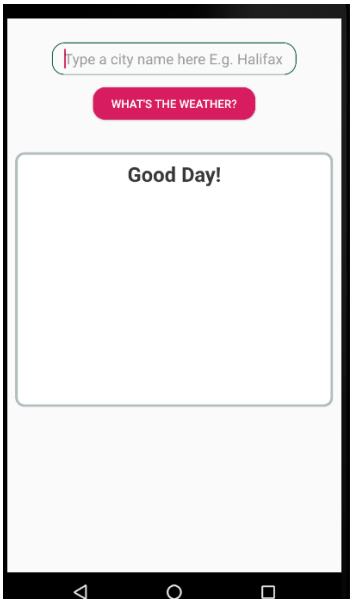

Figure 1) Welcome Screen

Figure 2) Application Output

Implementation Issue and Solution

I encountered a problem when the internet connection was lost (Mobile data or WiFi). On clicking the 'What's the Weather?' button, the weather application, initially, did not display anything when there was no internet connectivity. Only the last state of the application was displayed on the screen. After figuring out the error, I handled this issue. Now, when there is no internet connection, it displays an appropriate error message and clears the previous state of the application.

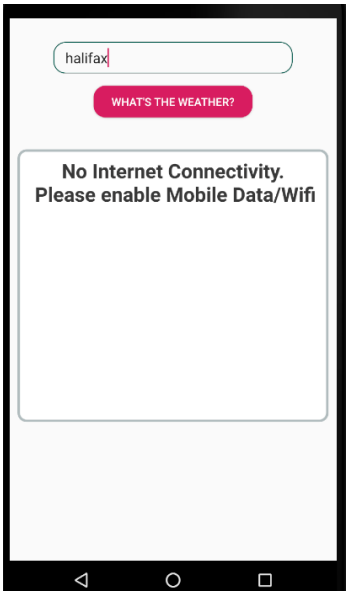

Figure 3) No Internet Connection

Neilson's Heuristics Implementation

1. **Heuristics 1: Visibility of System Status**
   The application I have developed gives appropriate feedback. The user is made aware of what is happening with the application with suitable response to one's actions. For instance, if a user enters invalid input, one is informed about error in an understandable way.

2. **Heuristics 2: Match between the System and the Real World**
   The language used in the application is straight-forward and is understandable to the user. The phrases in the application is clear.

3. **Heuristics 6: Help users to recognize, diagnose, and recover errors**
   With the appropriate and comprehensible error messages, the user can recover from the errors made. The error messages displayed are courteous and do not offend the user.

4. **Heuristics 9: Focus on Aesthetic and Minimalist Design**
   The application uses minimalistic design and simple language. The design consists of only three main components – edit-text, button and linear layout.

Testing Methodology

I tested the Application on Nexus 6 Android emulator and OnePlus5T phone. While testing the application, I noticed that the API would return values even when digits were entered. I think this is due to some ID associated to a particular city. As this weather application needs to support only text input, I handled this issue by not making a HTTP request for numeric value and displaying appropriate error message.

I tested the application for invalid city name and displayed an understandable error message. Also, the application displays appropriate message when a user clicks the button without typing any city name.

Figure 4) Digit Input

Figure 5) No Input

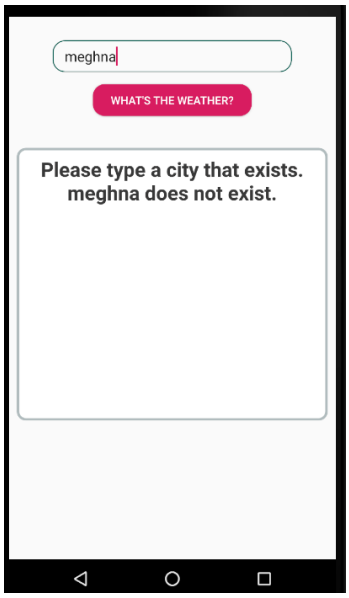

Figure 6) Wrong City Name

References

[1] Connect to the network Android Developers. (n.d.). Retrieved from https://developer.android.com/training/basics/network-ops/connecting

[2] Welcome to Brightspace at Dalhousie University. (n.d.). Retrieved from https://dal.brightspace.com/d2l/lms/dropbox/user/folder_submit_files.d2l?db=48909&grpid=0&isprv=0&bp=0&ou=86633

[3] Drawable resources Android Developers. (n.d.). Retrieved from https://developer.android.com/guide/topics/resources/drawable-resource

[4] OpenWeatherMap.org. (n.d.). Current weather data. Retrieved from https://openweathermap.org/current