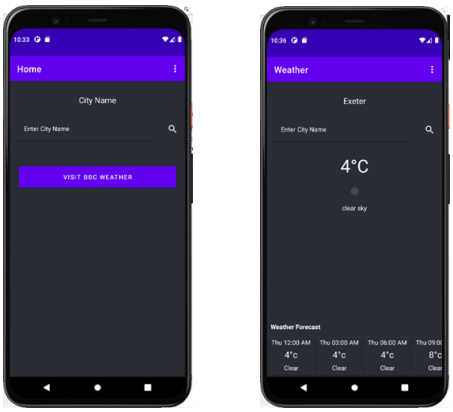
ECM2425 - Weather App

Candidate Number: 132717

1 Introduction

This is a weather app - which is a service that tells the users what weather they should expect in a chosen location along with information about the weather in the coming hours and days. This will help users to make informed decisions as weather conditions and temperatures have a huge impact on our daily lives.

2 Description of key app components



(a) MainActivity (b) ChildActivity

Figure 1: Activities - Dark Mode

There are two activities, MainActivity and ChildActivity. An explicit intent is used to navigate from the MainActivity to the ChildActivity, passing through the entered city as extra data so that when the ChildActivity starts, the weather information for the entered city can be displayed.

The MainActivity displays the home page, allowing the user to enter the desired city to get the weather information. There is also a button with a link to BBC Weather to give the user an option to get more weather details for their chosen location. This activity also includes a menu that allows the user to change between a dark and light mode – changing the colours of the screen accordingly.

The ChildActivity displays the weather information of the chosen city. In the centre of the screen, the temperature is displayed in degrees Celsius. Below this is an icon depicting the weather condition and below the icon is the description of the weather condition. At the bottom of the screen, there is a recycler view of the weather details given at three-hour intervals.

The internet is used to get the weather details of the chosen city. It fetches the data from a weather API called OpenWeatherMap in a JSON format. Specific weather details are then extrapolated to be displayed on the screen. The internet is also used when the 'Visit BBC Weather' button is clicked to direct the user to their web page using implicit intent.

3 Design rationale

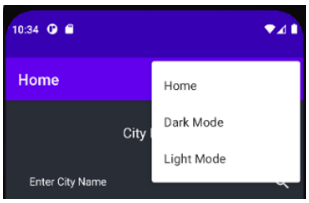


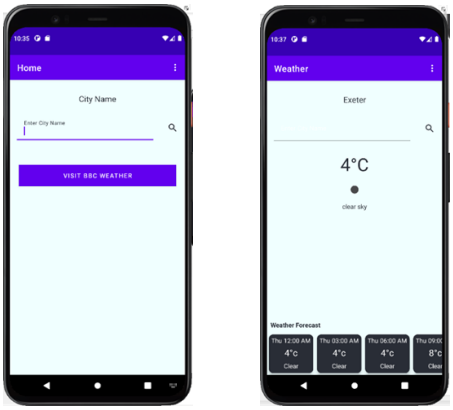
Figure 2: Menu

A menu is used to store the options of the light and dark mode along with a home button item, which when clicked, navigates the user back to the home screen. A menu is used as the items stored in it are constantly an option for the user to choose so this design is more functional and cleaner than having physical buttons on the screen.

Figure 3: Recycler View

A recycler view is used to display the weather forecast at three-hour

intervals. Instead of a list view, a recycler view was used as it efficiently displays large sets of data by dynamically creating the elements when they're needed, reusing cells.



(a) MainActivity-Light (b) ChildActivity-Light

Figure 4: Activities - Light Mode

This app includes a dark and light mode which changes the background colour and text accordingly. This design choice was made so that the user can adjust the settings of this app to match the themes of the rest of the user's phone – they might prefer light or dark mode on their device's settings. This setting is saved using shared preferences so when the app is closed and started back up again, their settings will be remembered.

4 Reflection on the development process

In the planning stage, I decided that there needed to be at least two activities as intents needed to be used. I first designed the layout of the home page with a sketch and then designed the page that would lead on from the home page – the weather information. Looking through the task specification, I added the details that were needed such as the menu and recycler view. This is a good practice as it makes sure that all the requirements are met beforehand so that there are no difficulties trying to add in the additional feature at the end which may result in rewriting code which is inefficient.

When choosing which API I should use, I looked for a free weather API

that not only gives the current data on the weather conditions of a chosen location but also the future weather condition – such as an hourly forecast. This planning ahead is good practice as I am checking beforehand if the API I am using is going to fully meet my requirements. This eliminates the risk of having to rewrite code due to a change in API and therefore JSON format.

I started the project by creating the XML files as the code is based on the UI. The next step was to get the basic functionality working such as when text is inputted into the input box and the search image is clicked, the ChildActivity starts. The next step is the get the weather details from the internet and have them stored in the views. Then more java classes needed to be made to handle to recycler view. At this point, the basic functionality of the app is complete. The addition of the menu, the shared preferences of the light and dark mode, the navigation to the home page, the use of the implicit intent to go to a website, allowing screen rotation with the data being retained and other details were added after.

I could have done better with my time management to allow time to fix problems that could occur and leave you stuck for a day. However, to prevent this from happening in the first place, I could have looked more closely at the code and read it line by line to make sure that a line isn't missing to would prevent part of my code from not working.

5 How to improve the app in future

To improve the app, I could implement more features such as creating a fragment from when an item from the recycler view is clicked which would give more details about that item. I could also have bookmarks of the user's favourite places so instead of typing the city's name again, the user could click on a recently searched city or a city that has been saved – this would give a more personalised experience on the app for the user.