Coursework Report

Erik Sanne
40276539@napier.ac.uk
Edinburgh Napier University - Module Title (SET08114)

1 Introduction

The Idea Originally I intended to do create a live bus time tracker such as the Lothian Buses Mobile App: [2]. However as this is a little too big of a project, I decided to scale it down to one component.

Scope and content My app is only implements a Bus Stop Finer rather than a full Bus Trcking App. It is works for Edinburgh and specifically all Lothian Buses Bus Stops.

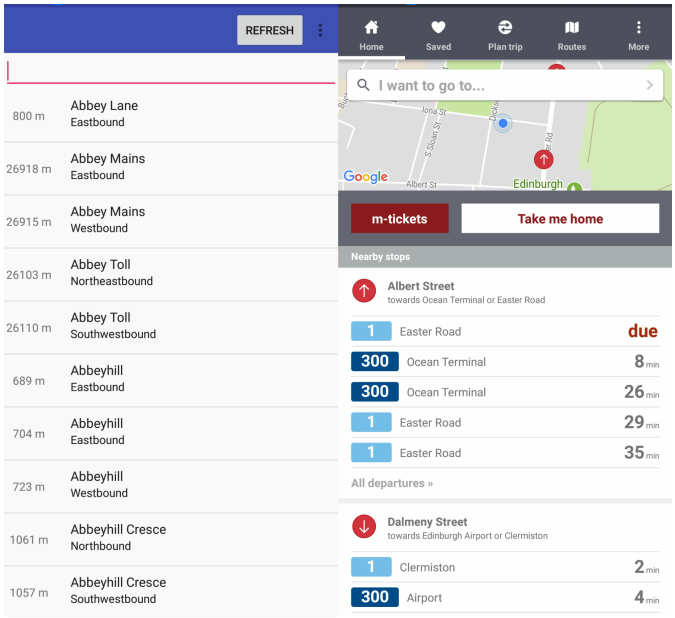

Figure 1: **Bus Stop Finder - Lothian Buses App**

2 Software design

To design the software I used a Use and a Class diagram, as this is a one person project there was no further planning required and much was decided while developing the app. If working with a team I believe it would be necessary to define the app better during the design process.

2.1 Use Case diagram

Very small but useful for future features.

2.2 Class diagram

I used the class diagram to visualize how to set up my project. It doesn't have much details but provides a good overiew over the structure of the project.

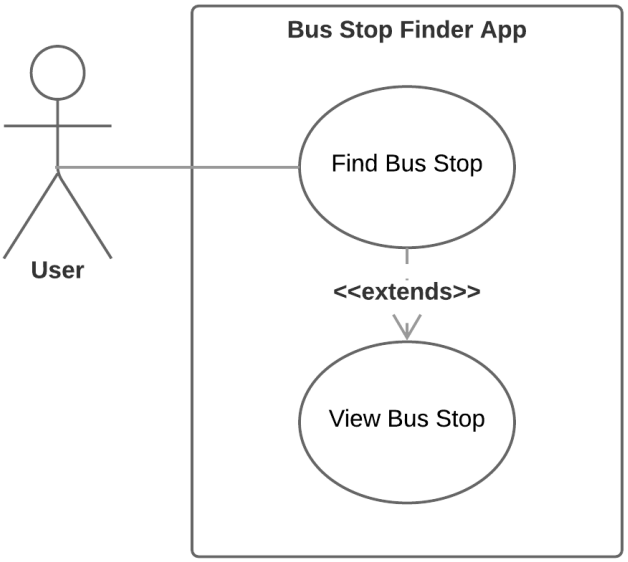

Figure 2: **Use case diagram**

3 Implementation

Firstly I implemented the back end, the API[1] access as well as the SQLite database. Then starting to build the main app I developed the front end step by step as it was needed.

4 Evaluation

Compared to other similar application there is a strong lack of features available and the design is not as good. Both the Lothian buses app as well as the My Bus Edinburgh apps provide live tracking, a map of stops and many other features. However my app is very simple and allows the user to quickly find a bus stop by its name.

It is easily possible to add features in the future as the underlying Bus Stop framework can be used for things like Favourites, Routes, Displaying more information as well being the center piece for a live times implementation.

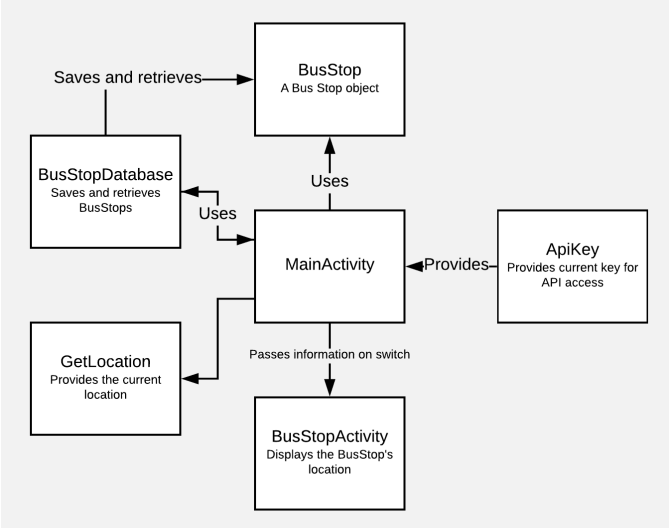

Figure 3: **Class diagram**

```
@Override
protected String doInBackground(URL... urls){
    int count = urls.length;
    long totalSize = 0;
    StringBuilder resultBuilder = new StringBuilder();
    for (int i = 0; i < count; i++) {
        try {
            // Read all the text returned by the server
            InputStreamReader reader = new InputStreamReader(urls[i].openStream());
            BufferedReader in = new BufferedReader(reader);
            String resultPiece;
            while ((resultPiece = in.readLine()) != null) {
                resultBuilder.append(resultPiece);
            }
            in.close();
        } catch (MalformedURLException e) {
            e.printStackTrace();
        } catch (IOException e) {
            e.printStackTrace();
        }
        // if cancel() is called, leave the loop early
        if (isCancelled()) {
            break;
        }
    }
    // save the result
    this.result = resultBuilder.toString();

    return result;
}
```

Figure 4: **Update databse: Background task**

5 Personal evaluation

Starting out, my knowledge of Java was already a bit rusty and I had very little knowledge of Android development. Although frustrating at times, it was a great pleasure to learn. Android can be very different from Java at times, for instance: to test my API implementation I built an extra Java project but upon porting it to the App, I found out that android requires an Async task to make an API call. Luckily there are many resources available online that can help fix these knowledge issues.

References

[1] MyBus Edinburgh API. http://www.mybustracker.co.uk/?page=API%20Key.

[2] Lothian buses app. https://lothianbuses.co.uk/apps.