Major parts of the application

The application is built up by four main parts:

- The main activity *MainActivity* which is the base for all the fragments and the navigation drawer. This activity handles switching between fragments and navigation. The fragments are the different sections of the application that is visible to the user:
    - *TravelFragment* - this is the first screen that the user encounters when the application has started. This fragment uses the GPS in the background and tries to determine the user's location. When an estimated location has been found a call to Västtrafik will be made to acquire nearby stops using the coordinates found by the GPS. A spinner will be filled with those stops, if any are found, in order of distance. The selected stop will be at the top of the spinner's list and is the one that is closest to the user.

      When a stop has been selected in the spinner a list will be filled with departures from that stop. This information is also acquired from Västtrafik using its *DepartureBoard* feature. When a user clicks on such a departure in the list, she will be directed to the next fragment - *DestinationFragment*.
    - *DestinationFragment* shows the possible destinations from a chosen departure. This information is acquired from Västtrafik using a reference URL from the departure that was selected in the previous fragment.

      When a destination has been selected the user will be directed to the next fragment - *InfoFragment*.
    - *InfoFragment* shows some basic information about the trip, source and destination of the trip, and a counter. If the user is waiting at the station the counter will show how many minutes there are until the vehicle arrives. If the user is already on board the counter will show the time it takes until the destination has been reached.

      The counter is updated by a timer which requests new journey information every 20th second. When the information is returned, which usually takes about one second, the counter is updated with new values. The counter time is based upon real time data if available, otherwise it uses the timetable information which is also provided from this information.
    - *PaymentFragment* is a prototype of a payment system, it shows as an example how easy it could be to pay for your trip inside an app.
    - *SettingsFragment* allows the user to turn on/off push notifications that alerts the user when the destination is at the next stop. However, this setting is currently not used by the application.

- The REST-API classes. The main classes are *ECClient* which handles communication with ElectriCity and *VTClient* which communicates with Västtrafik. To ease communication with the servers a library called Retrofit [1] is used. This library translates HTTP API queries to a Java interface and the returned data (such as JSON or XML) will be converted to Java classes.

  ECClient connects to ElectriCity's servers and sends queries to download resources or sensor data from buses. The server returns JSON data which will be converted to Java classes that can easily be used by the applications fragments.

VTClient connects to Västtrafik’s REST server and downloads information regarding stops and lines which will be converted from XML to corresponding Java classes.

- *WifiFinder* is a helper class which uses the device's built in WiFi and checks if any routers are nearby which has the name “ElectriCity”. If such a router is found the class tries to match the router's MAC address with a predefined list with mappings from MAC to ElectriCity bus ID’s.
- *AlarmService* is started from the InfoFragment when the WifiFinder has successfully found a vehicle and is handed an ElectriCity bus ID and a desired destination. It repeatedly asks the ElectriCity API if that vehicle’s next stop corresponds to the destination. If it does, it sends a push notification to the device, telling the user that it’s time to get off.

Design decisions

The application is built to support SDK version 15 and up. This decision was made to support as many devices as possibly except for the really old ones. The application itself does not require anything specific that is only available for newer devices.

External dependencies

The application requires the Retrofit libraries which requires a converter called simpleXML since Västtrafiks data is returned in XML format, as well as a client called okHTTP which is used by this library to handle HTTP communication.
The application requires the Android support library since it supports SDK version 15 and higher.

[1] Retrofit, A type safe HTTP client for Android and Java - http://square.github.io/retrofit/