HelloMap - Outdoor/Indoor Positioning App

GPS is not normally available inside buildings, so maps need to also integrate indoor positioning to accurately find locations of their users wherever they are. HelloMap is an indoor/outdoor positioning map designed to tackle this problem. It uses Google Maps API to display the outdoor map and a custom built-indoor positioning app for the University of Edinburgh's Fleeming Jenkin Building. This document provides a user guide that describes the features of the system and a programmer's guide which describes the implementation of the app.

1. User Guide

This section will describe all the features of this app and how to use them.

Outdoor

The outdoor map uses shows a hybrid map, which overlays map information such as street names and building names on top of satellite/aerial imagery.

The main features of the outdoor maps are:

- Compass on the top left which shows which direction North is
- Locator Button automatically zooms into your current location
- Blue dot marker and a Red Marker, which approximately shows where you currently on the map
- A Floor-Plan which has been overlayed on top of the Fleeming Jenkin Building, overlay is oriented against the Earth's surface rather than the screen, so rotating, tilting or zooming the map will change the orientation of the image.
- Touching the floorplan automatically takes you to the indoor positioning mode
- A shortcut on the bottom right which takes you to the indoor positioning mode
- Scrolling, tilting and rotating options have been enabled so user can easily manipulate the map view to suit their needs.
- Indoor Level Picker has been enabled so that, if google already has floorplans of building we can navigate through different levels of the floorplan within the map itself. Try visiting the University of Edinburgh's Informatics forum on HelloMap, to see this feature in use.

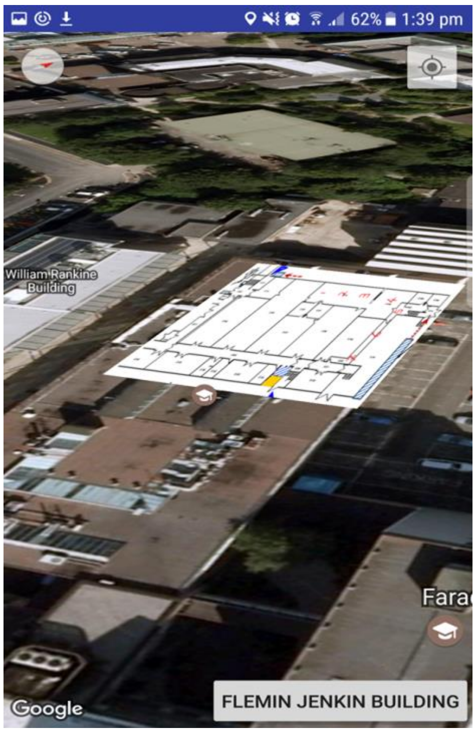

Figure 1: The HelloMap in Outdoor Mode

Indoor

Indoor Mode represents the indoor positioning functionality. There are two possible ways to switch over to the Indoor Mode. Either by tapping the map overlay on top of Fleeming Jenkin Building or by tapping the shortcut on the bottom right corner. When the app enters indoor mode, there is an alert message which describes the main features of this mode and how to use them.

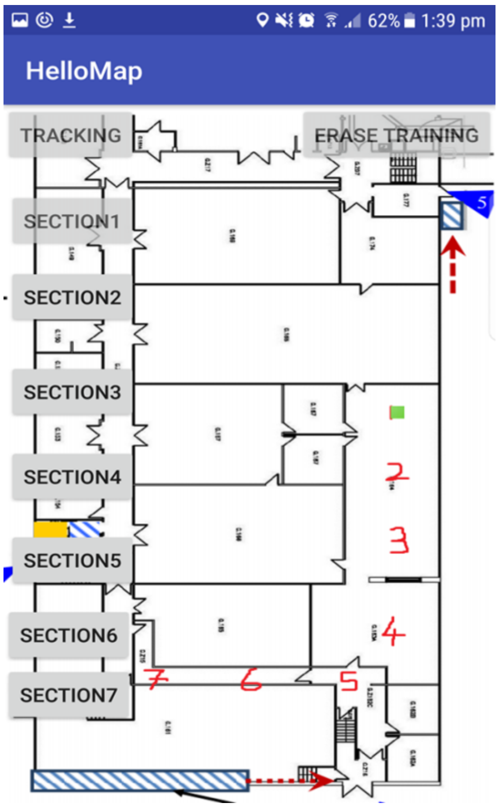

Figure 2:Indoor Mode

The Indoor mode consists of a background of the Fleeming Jenkin Building floorplan.

There are seven section buttons which correspond to the red numbers on the on the map. We can see numbers 2-7 in Figure 2. Number 1 is currently being covered by the green tracking marker. When you press these buttons then a Wi-Fi Scan is done to sense the signal strengths of two particular Wireless Access Points (WAP) (see Programmer Guide for more details) at these coordinates, then the signal strength levels are stored along with the corresponding section in a database.

We will need to initially populate the database with Wi-Fi signal strength values for each section. For example, we can tap Section 1 if we are in section 1 and the signal strength levels will be stored in the database. The user can keep on adding as many values as they wish for each section, and adding more values will only increase the accuracy. Obviously this app will only be as accurate as the values provided by the user, so they should ensure that they are in the correct section before choosing which section button to tap.

The Tracking button, scans the current signal strengths of the WAP's and then finds a matching row in the database with the same signal values, then uses the section number column to determine where you currently are. The tracking button places a green marker on top of the section you are currently in.

The Erase Training button on the top right, clears the database of all entries. This way the user can start fresh if they have made a mistake or if they are getting inaccurate results they can clear the database so inaccurate rows are deleted and then they can add new database entries.

2. Programmer's Guide

This section will describe how all the features described in the user guide above were implemented. The app has been split into 5 java classes for modularity. The MapsActivity is the initial activity which starts when we start the app. The Indoor Mode class implements the indoor mode functionality. DBAdpater implements the database, and provides an interface in which to interact with the database. The pointer class contains the code to draw the green tacking marker and move it around the screen. GPSTracker is used to get the latitude and longitude, which we use to place a red marker of our current position on google maps. All these classes will be described in more detail below.

MapsActivity

This class uses Google map service to build a map-based application by using Google Maps Android API v2. The MapsActivity extends the Fragment activity and implements the onMapReadyCallback and Google.mapOnGroundOverlayClickListerner.

The indoor button is implemented in the function init(). The button has a setOnClickListener(), so that when it detects the user tapping it then the button starts an Intent to start the IndoorMode activity.

The onCreate() method initialises the map layout using setContentView() and initialises the gpsTracker and my locations. The method also obtains the SupportMapFragment and gets notified when the map is ready to be used.

The onMapReady() method implements all the options for google maps, these include:

- Adding a Adding a marker for current location
- Marker for Edinburgh
- Moving the camera to the current location
- Enabling map options such as compass, rotation, scrolling, tilting and indoor level picker.
- Adding an overlay of the the Fleeming Jenkin floorplan to the coordinates of the Fleeming Jenkin building on google maps, and adjusting ti so that it fits on top of the building.

Finally the ground overlay is connected to a OnGroundOverlayListener in the onMapReady method. This starts an intent to switch to the Indoor mode activity when tapped on.

IndoorMode

In the OnCreate() methods, a AlertDialog.Builder is implemented to display a message showing how to use the indoor mode. Then the database and all the buttons are initialised throught the openDB(), init() , display() and delete() methods.

The init() method implements the onCLickListeners for the tracking button . The button starts a scan of the WAP's. Then stores their details of them in a List of scan results called wifiList. Then stores the the signal strength levels of **"58:97:1e:14:75:51"** and **"6c:99:89:0d:91:41"** into variables one and two. These two WAP's were chosen as they are accessible from all seven sections, and have different signal strength for each section. These two values are then compared to the rows in the database which have the same signal strength values for one and two, and then if there is a match then we use that row to get the sector number and move our pointer to that position.

The display() methods implement the onClickListeners for the seven section buttons, which when tapped, scan the WAP's then store the signal strength values of the two WAP's, **"58:97:1e:14:75:51"** and **"6c:99:89:0d:91:41",** and then stores them in the database along with the corresponding section number.

The delete() method, implements the onClickListener for the button that deletes all entries in the database. It uses the myDB.deleteALL() method to clear the database and a Toast to tell the user that this action has been done.

The onDestroy() method closes the database using closeDB(), when the activity is destroyed.

The openDB() method creates a new instantiation of the DBAdpater and stores it in myDB, then opens the database using myDB.open():

DBAdapter

This creates an interface from which we can interface with the database. It implements the database using SQLite, DatabaseHelper and implements the Cursor functions from which we can navigate the records of a database.

It creates a table called MyDB, with four columns. One for the unique key for each row, two for the signal strength values and one for the sector value.

It implements all the classes from which we can interface with the database, such as open(), close(), insertRow() and deleteAll(). There is also a custom interface : getMatch(), which match two signal strength values with the rest of the database. This is used when we are looking for matching signal strength rows, in the Tracking button in Indoor Mode.

Pointer

This class draws the green tracking marker onto the screen. Uses the bitmap to mark the image file onto the screen.

This also maps the bitmap to a matrix, this is needed if the bitmap needs to rotate, this hasn't been used but can be used in future updates, for example if we want to display the orientation of the user and make the marker rotate accordingly.

This bitmap, can then be moved around in the Indoor Mode activity by calling, pointer.setX() and pointer.setY() methods to change its coordinates on screen.

GPSTracker

This class implements the LocationListener, which is used to tag the marker to your current location in the MapsActivity. It uses LocationManager, and scans to check if GPS is available. If available itgets latitude and longitude form the GPS and if not available it uses the Network provider to get the latitude and longitude values.

The class returns location, which contains the latitude and longitude.