Mobile App Engineering – 3rd Assignment: Localization and the Daily Path

Kristen Wong
RUID: 159000497
netID: kew132

Part 1 (45 Points Max)

For this section, I created an activity that opens when the application is initially booted up. The app displays the user's location in terms of the user's latitude, longitude, and address. Android's location services are used to retrieve latitude and longitude coordinate data, and Android's Geocoder class is used to retrieve the user's address from the coordinates. All of these values are displayed as a TextView on the screen. There is a "Check In" button that is also located on this screen that allows the user to check into their current location. When the user presses this button, they are prompted to add a name for the check in. Once the user clicks "Add" after entering the check in name, the check in data for that location is stored in the SQLite database. At the bottom of the screen, a list of all previous check ins that were made either by the user, or in the background by the system, are displayed on the screen. The list shows each check in name (if any), address, time, latitude, and longitude. If the check in was associated with a previous check in, then the name and address of the previous check in will be displayed on the list.

The application also checks for when the user changes locations and updates the UI accordingly. In order to eliminate race conditions, the callback to check for changed locations is called on the UI thread so that the UI does not update at the same time that the location is being retrieved in another thread.

I was able to use the app to check into various locations around the Rutgers campus, as depicted on the map from the assignment document. However, since this data is being stored on the phone's individual SQLite database, this data will not be transferred when the app and assignment are submitted.

The application will also associate a check in with a previous check in, if that check in is within 30 meters of the previous check in. The new check in will take the name and postal address of the previous check in. Different check ins that are associated with the same previous check ins can be differentiated by their time and exact latitude and longitude values. All of this data is stored in the SQLite database in 2 tables. One table represents "locations", which hold the names and addresses that previous check ins can be associated with, and the other table represents "check ins" which hold all of the check in data, including latitude, longitude, and time, as well as the associated location. When a check in is being added to the list, it also queries the location table and checks if any of the stored locations are within 30 meters of the check in being added. If the check in is outside of all 30 meter ranges, then a new location is added to the locations list as well.

Part 2 (75 Points Max)

On the first activity screen there is a button that says "View Map". When the user clicks this button, another activity is launched displaying a map centered on the user's current location. There is also a blue marker indicating the user's current location. The map has Zoom

In/Zoom Out buttons as well as a UI compass enabled. Below the map, there is a button labeled “My location”, when the user clicks this button, the map resets the camera to center around the user’s current location.

All of the previous check in locations (from the locations list) are displayed on the map as red dots marking the location.

When the user clicks on the map, a dialog prompting the user to enter a name for a new marker appears. Once the user presses “Add”, a new red marker appears on the map. The users is able to drag this marker, and the location of the marker will be stored in the “locations” list in the database.

When the user is in this map activity, and they are within 30 meters of a previously named location (from the “locations” list), a dialog with appear notifying them of the location and the time of last check in. (The SQL query for this is “SELECT * FROM “check ins table” ORDER BY “time column” DESC LIMIT 1”). When they leave the 30 meter radius, this dialog will disappear.

Part 3 (90 Points Max)

The application uses a service that runs threads in the background to automatically create check ins every 5 minutes, or when the user has moved more than 100 meters. This is done by using a thread handler with the postDelayed method. When a check in is created with these methods, it automatically is saved under the name “Auto Check In”. The service also broadcasts whenever a check in is created so that the ListView in the first activity can be updated immediately. The threads also run when the application is in the background.

Part 4 (100 Points Max)

I created a neat and clean UI with custom buttons and color scheme. I also worked to make the report as informative and understandable as possible.