Sahil Lamba - 13103649

Ajayveer Singh - 13103418

Web technology lab minor

AUTOHUSH
location based android application

Introduction

Students/Working class/Corporate people in today's world generally get disturbed by their phone while they are busy working. What if there were some app which could change the way their phone works when they reach different locations? *AutoHush* will be an android utility application that will provide the user the advantage of getting tasks done automatically when the device arrives at a particular location.
'Location' will be the **theme**. Using intelligent UI design, the app will save a new 'Location' card (Latitude & Longitude coordinates) along with the required location based feature.

Objectives

It will provide the following location based features :

- **Location based vibration (and also power on/off)** - Put your phone on vibration whenever you reach a particular location. So whenever a student enters his class his phone will switch to silent mode (or switch it off). Neither the teacher nor the class will get disturbed anymore!
- **Location based Wifi and Mobile data** - Switch on Wifi and automatically connect to the (remembered) Wifi router. So when one exits office, phone's wifi will be switched off, mobile data will be switched on, and then when he reaches home, wifi will be switched on again.

- **Location based reminders** - Reminding yourself about some work you were supposed to do upon reaching a particular place. *"Buy 1 kg apples, 2 litre milk, chocolates for son"* so that you don't forget to buy these when you reach the market.

Division of work

The group members did all the work collectively. The team is planning to write a research paper under the members' and mentor's name at the end of the project.

Background study and findings

We had to go through a lot of research related to a device's location. As soon as we implemented the location coordinates class, we were able to retrieve the latitudes and longitudes of the device. We tested the accuracy across various locations in Delhi (Connaught Place, Karol Bagh, Model town) and Noida(Sector 62, Sector 18). This was done to check the device's gps provider's efficiency in areas where there is good network coverage and where there is not.

We analyzed the data that we got, and realized that the latitudes and longitudes are accurate till the 3rd significant digit for radius of about 50metres (tested on Nexus 5).

For example, if the previous value of latitude at current location is 28.8917658 then the current value of latitude at the same location could be 28.8916871.

- **Google’s API**

The Google Location Services API, part of Google Play Services, provides a more powerful, high-level framework that automatically handles location providers, user movement, and location accuracy. It also handles location update scheduling based on power consumption parameters you provide. In most cases, you'll get better battery performance, as well as more appropriate accuracy, by using the Location Services API.

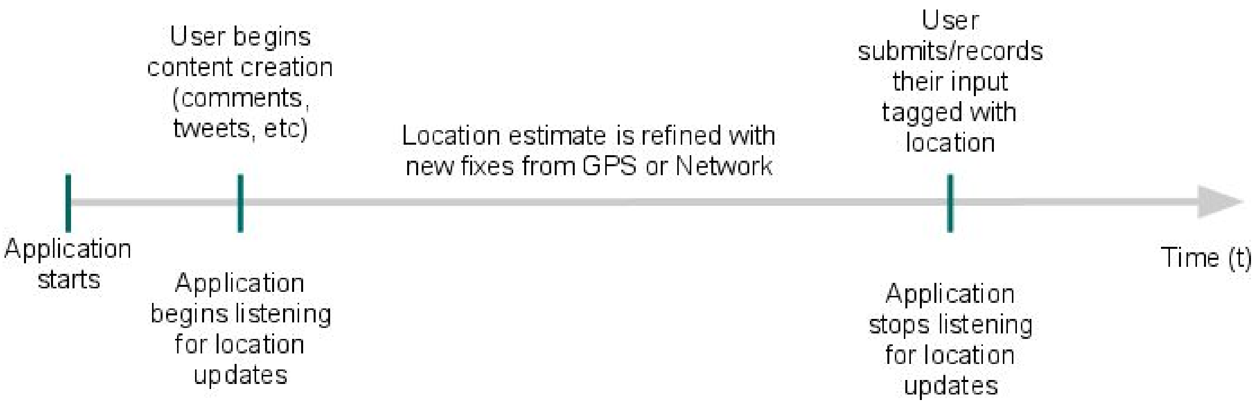
When developing a location-aware application for Android, you can utilize GPS and Android's Network Location Provider to acquire the user location. Although GPS is most accurate, it only works outdoors, it quickly consumes battery power, and doesn't return the location as quickly as users want. Android's Network Location Provider determines user location using cell tower and Wi-Fi signals, providing location information in a way that works indoors and outdoors, responds faster, and uses less battery power.

Challenges in Determining User Location

- **Multitude of location sources -** GPS, Cell-ID, and Wi-Fi can each provide a clue to users location. Determining which to use and trust is a matter of trade-offs in accuracy, speed, and battery-efficiency.

- **User movement -** Because the user location changes, one must account for movement by re-estimating user location every so often.

- **Varying accuracy -** Location estimates coming from each location source are not consistent in their accuracy. A location obtained 10 seconds ago from one source might be more accurate than the newest location from another or same source.


User begins content creation (comments, tweets, etc)
Location estimate is refined with new fixes from GPS or Network
User submits/records their input tagged with location
Application starts
Application begins listening for location updates
Application stops listening for location updates
Time (t)

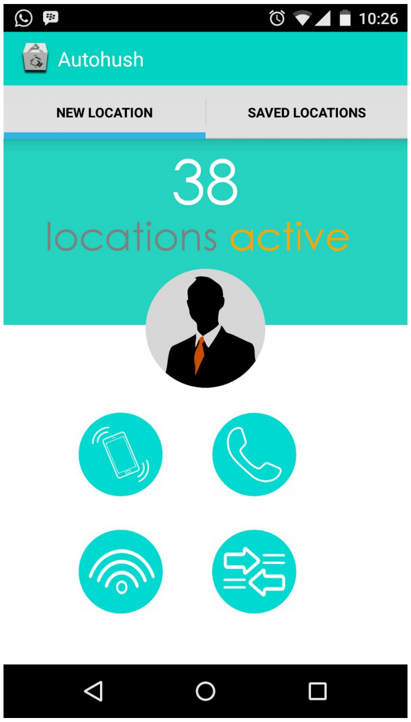
Design

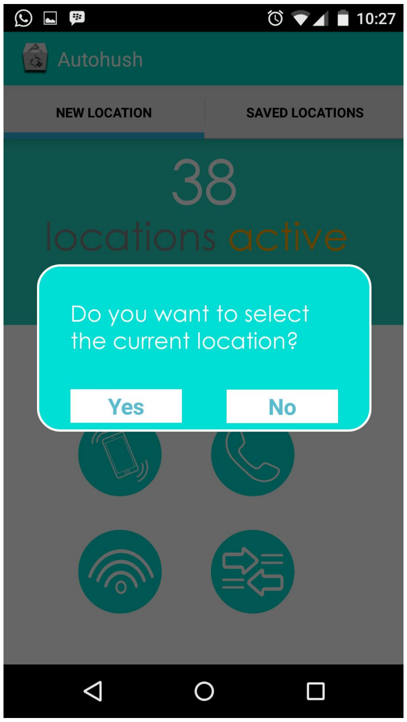
The software design consists of a few activities (directly displayed) and other classes who are used for creating the database etc.

Class name	Purpose
MainActivity.java	Displaying the home screen and giving the user the location based options available
CustomDialogClass.java	Displays a dialog box for letting the user choose between current location and searching some other location
Activity2Activity.java	Displays the current live location of the user on a map.
GPSTracker.java	Enables the gps to fetch the current location coordinates
Confirm_Feature_DialogBox.java	Confirms the user's choice if he wants to save the current location or go back
DatabaseActivity.java	Handles the communication between android and the SQLite database
LocationDataSource.java	Manages the database
LocationData.java	Serves a model for the type of data in one

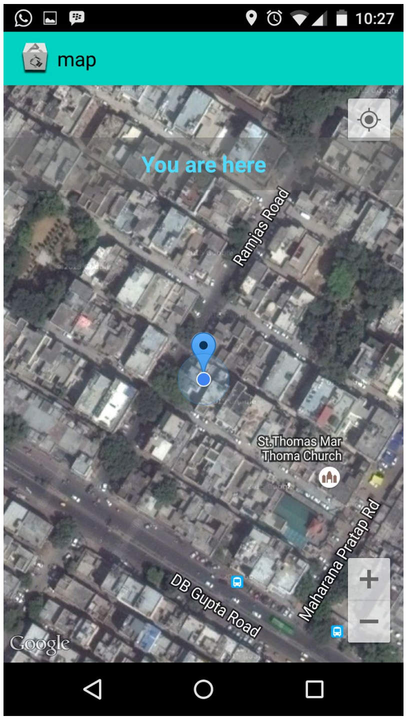
	row of the table
SQLiteHelper.java	Creates the design for the database using SQLite commands
NewLoc_Fragment.java	Displays the home screen in a tab
SavedLoc_Fragment.java	Displays all the locations saved in the database before through android cards
TabsPagerAdapter.java	Fetches the fragments (NewLoc_Fragment and SavedLoc_Fragment)
LocationsService.java	Creates a background service to check if the user has reached a saved location and put the device on vibration mode
ListLocations.java	Creates a model for the location cards to be displayed in SavedLoc_Fragment
MyAdapter.java	Fetched the locations from the database in an adapter to be displayed in cards

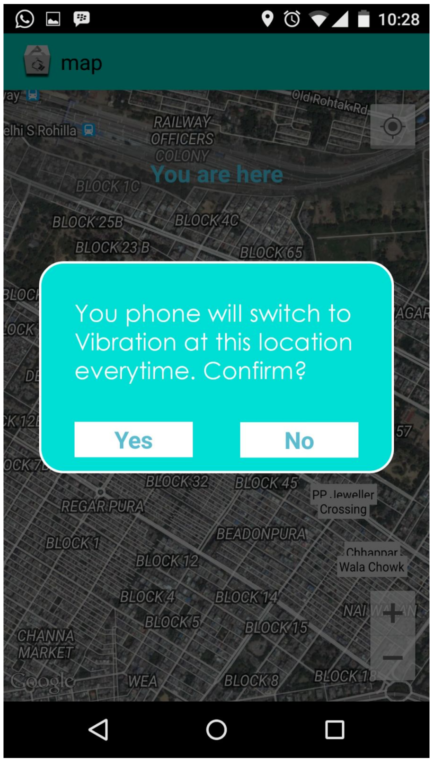
Screenshots



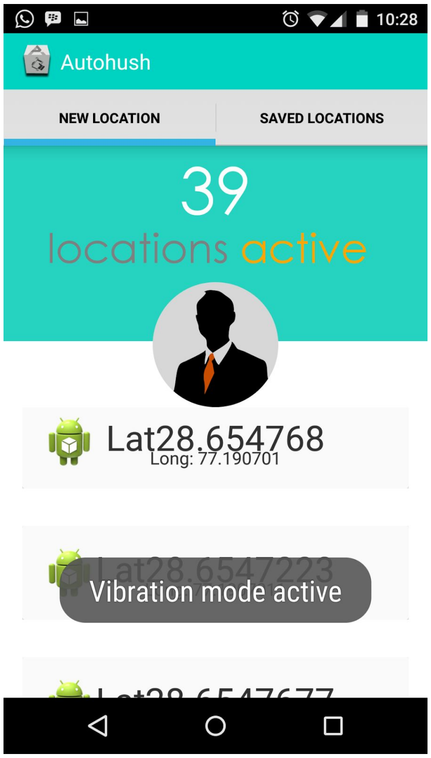


Do you want to select the current location?

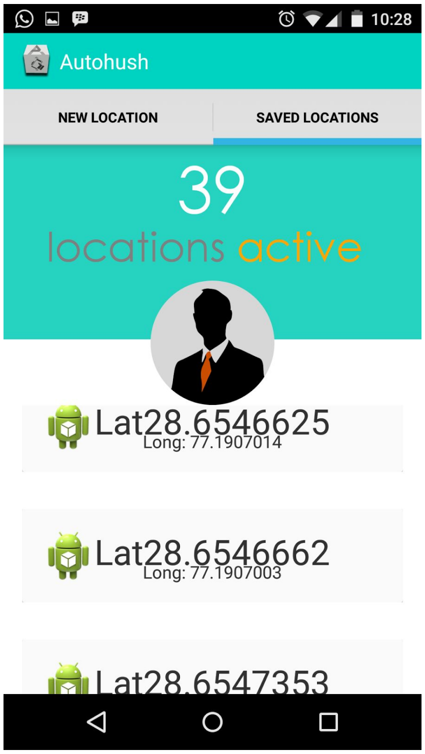




You phone will switch to Vibration at this location everytime. Confirm?



Vibration mode active



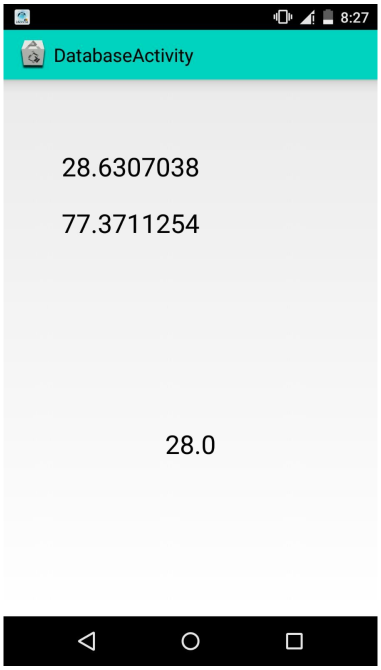
Code

We have created about 15 classes right now along with the XML layouts of the required activities. Some elements of the UI have been made using illustrator and photoshop. As obvious it is, we will be creating more classes depending upon how the project proceeds and what it requires.

Testing

The kind of app we're building, it requires intense testing. Testing the device at different locations especially for accuracy. As the service (of fetching coordinates) keeps running in the background, it will consume a lot of battery. We have tried to reach an optimum condition between the accuracy and battery so that the app gets the maximum accuracy possible by least battery usage.

Coordinates of LRC



8:27

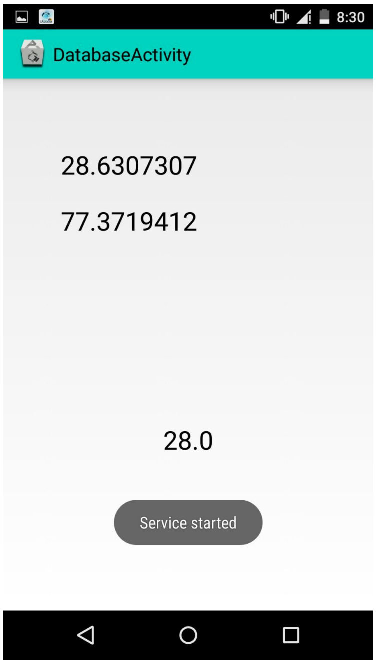
DatabaseActivity

28.6307038

77.3711254

28.0

Coordinates of Annapurna



8:30

DatabaseActivity

28.6307307

77.3719412

28.0

Service started

For that, we decided not to fetch the location coordinates continuously. Instead, we will set intervals of 2 minutes to 5 minutes for fetching the location data. Through this way, we won't have to compromise on the accuracy and in addition, it will consume less battery comparatively.

References

- http://developer.android.com/index.html
- Uber taxi android app
- http://www.vogella.com/
- https://**www.zomato**.com/
- http://www.acma.gov.au/webwr/_assets/main/lib310665/location_services_research.pdf
- http://stackoverflow.com/
- https://**maps.google**.com/