

Project 3 Report
Rong Zhang / rz214

In this project, I implemented an Android application which meets all the undergraduate requirements and 1) of the graduate requirements and part of 2) (not completed).

NOTE: This application is tested on an Android phone rather than an emulator. There may be some problems when running on the emulator.

There are two activities and one service in my app. A database is created in this project to store the data.

DATABASE DESIGN

There are two tables in my database named "locations" and "checkins". Table locations used to store the information (name, longitude, latitude, address, id) of different locations. Table checkins used to store the checkin time of the locations (time, locationid).

TABLE location

id	long	lati	name	address
1			ass	
2			ass	
3			dff	

TABLE checkins

Id	Time	locationid
1		1
2		2
3		1

When the user check-in a location, there will be some queries to complete this action.

1) SELECT id FROM locations WHERE long="+currentLongitude+" AND lati="+currentLatitude+"

This query is to check whether this location had been checked in before. If there exists, get the id of the location.

If not:

2) SELECT long, lati, name FROM locations

This query is to load all the locations to find if there is a location is within a radius of 30m of current one. Find this location's name and change current location's name.

3) INSERT INTO locations (long, lati, address, name) VALUES (?, ?, ?, ?)

If there is not the same location which had been checked-in before, insert this location.

4) SELECT id FROM locations ORDER BY id DESC

Find the id of current location.

5) INSERT INTO checkins (time, locationid) VALUES (?, id)

Insert time information to table checkins.

In this way, there is no redundant information to store a same location for many times in table "location". We use id to match the time and location, and shows them on the listview.

'MainActivity' is an activity which shows information (longitude, latitude and address) of current location. User can add a custom name for current location and check in this location. Any name of the location within a radius of 30m of an existing check-in will be automatically changed to the existing one. All checked-in locations will be showed in the listview below, which contains longitude, latitude, address, custom name, and check-in time. This list will also show the auto check-ins by the service in background. Update of the location information comes from UI thread using the service. No other thread is used in my application. Because location change is an event existing only for a short time, it is safe for the activity running. Based on test on the smartphone, the race condition can be handled in this application.

'Main2Activity' is an activity which implemented the part of Google Maps. When user clicks the button 'MAP' in MainActivity, it will jump to this activity. You should use my location button to focus your current location. Your current location will be updated in time. Initially, the map will show all the stored locations as markers. Touching the marker will show the information of this location. You can click a new place to add a custom name and store this location in the database. There will be also a new marker which can be dragged on the map. Every time you approach a place within a radius of 30m of a marker, the marker will show the information as a window.

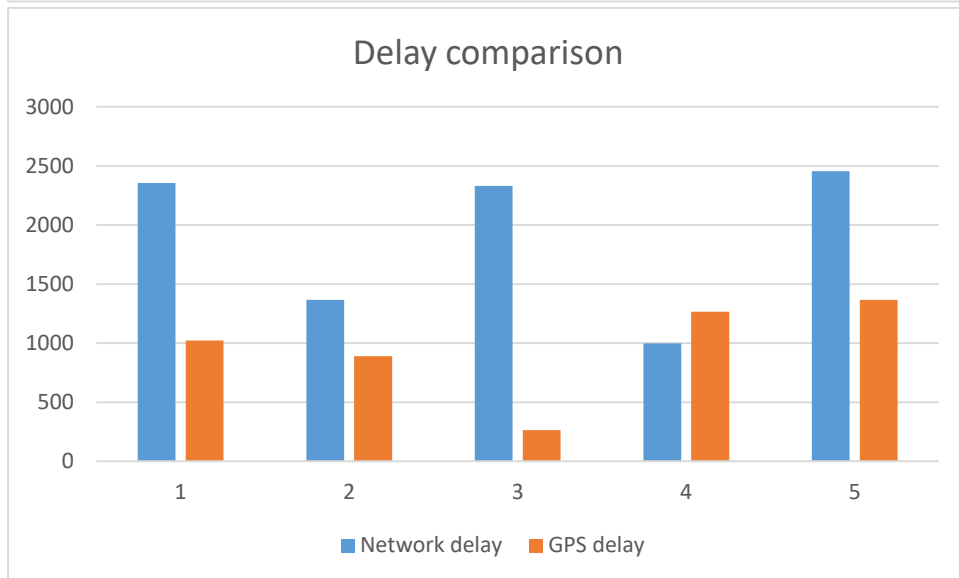
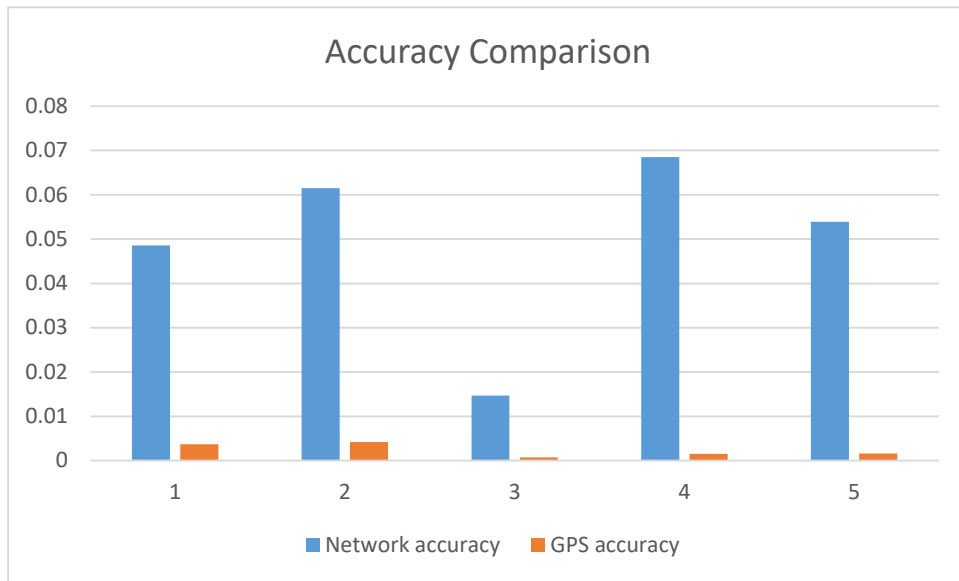
'UpdateService' is a background service used to update the location information for two activities. At same time, it will automatically check-in current location every 5 minutes or the when the user walks 100m far.

GRADUATE PART

1) I test my application when choosing different providers. Code for this part is at the start in MainActivity. You need to modify the code using different providers in both MainActivity and service and check the results in logcat. Several points indoors and outdoors are chosen to collect data. Here are some results of data analysis. (NOTE: Accuracy is represented as the distance between the location collected and the accurate location.)

INDOORS

	1	2	3	4	5
Accurate location	40.514703,-74.466998	40.524785,-74.466557	40.525035,-74.467455	40.521643,-74.460023	40.524512,-74.467992
Network location	40.514730,-74.467275	40.525393,-74.466898	40.524734,-74.467922	40.521188,-74.458786	40.524664, -74.468471
Network accuracy	0.0308	0.0421	0.0528	0.1384	0.0535
Network delay	398	693	466	887	1023
GPS location	40.514698,-74.467102	40.524781,-74.466456	40.525508,-74.467664	40.521442,-74.460079	40.524535, -74.467923
GPS accuracy	0.0116	0.0112	0.0072	0.0086	0.0077
GPS delay	5527	13213	12866	6562	8897

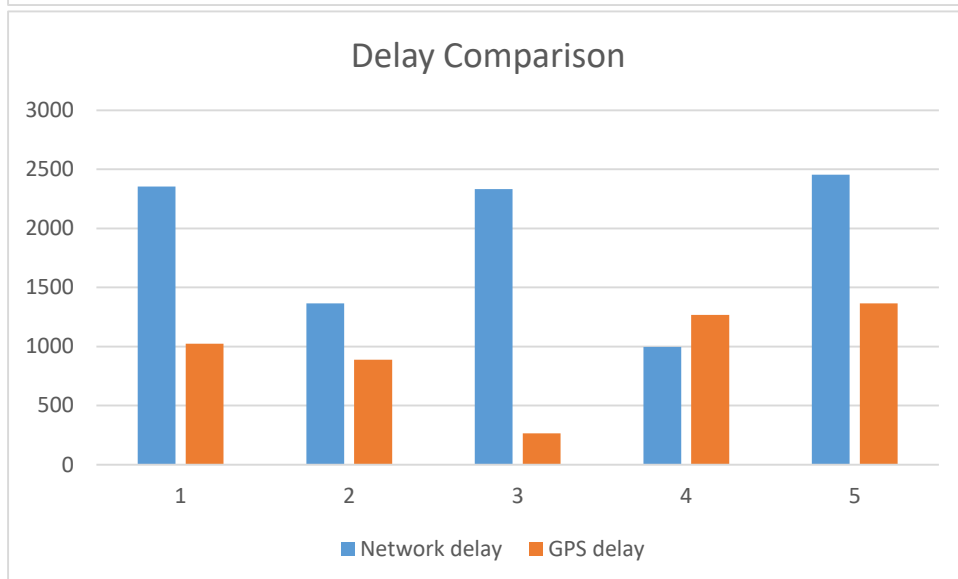
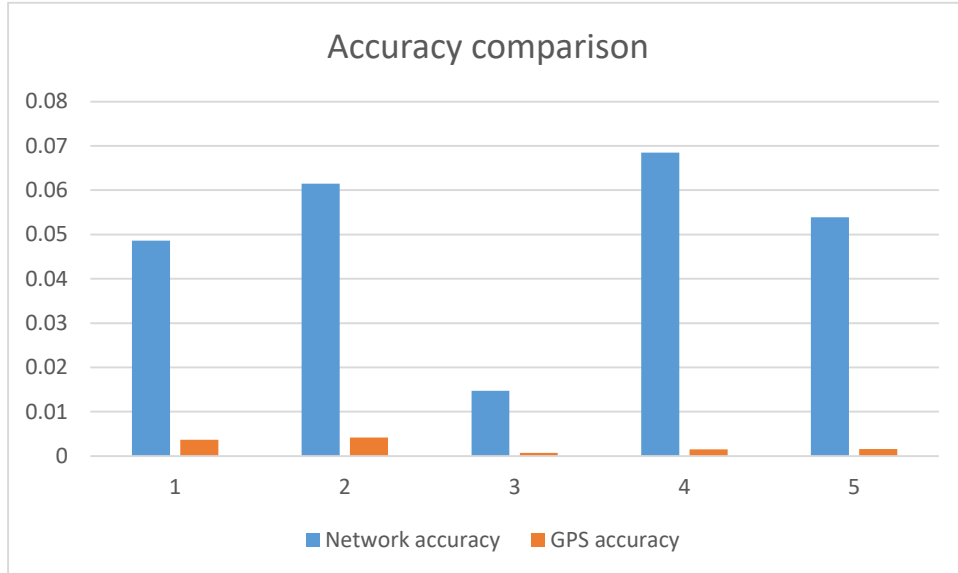


From the figures and graphs, we can find that GPS is more accurate than network, but GPS cost much more time indoors. This is because the signal from GPS is weak indoors, at the same time it is easier to locate using network provider.

OUTDOORS

	1	2	3	4	5
Accurate location	40.522467,- 74.465922	40.525153,- 74.456632	40.521188,- 74.464562	40.529886,- 74.462776	40.516512,- 74.467996
Network location	40.522572,- 74.465486	40.525342,- 74.457182	40.521635,- 74.464506	40.529032,- 74.462205	40.516864, - 74.468471
Network accuracy	0.0486	0.0615	0.0147	0.0685	0.0539
Network delay	2354	1366	2332	997	2455

GPS location	40.522472,- 74.465889	40.525143,- 74.456594	40.521167,- 74.464564	40.52989,- 74.462745	40.516552, - 74.467986
GPS accuracy	0.0037	0.0042	0.0007	0.0015	0.0016
GPS delay	1023	889	264	1267	1365



We can find that GPS is more accurate than network outdoors. And because there is no wifi signal always, only cellular is used in network, GPS's delay is even smaller than network. GPS is the best way for positioning outdoors.

2) In this part, I added a button to let the user switch the system. However, the details is not completed, this part is also not working.