CSCI 585- Database Systems
Spring 2010
Homework Assignment 3
Due: 05/02/2010 @ 8:00pm
Description

GeekDroid Application

In this project, *GeekDroid*, the goal is to design and implement a spatial microblogging system using mobile phones. With this system, users utilize their mobile phones to post microblog entries, geo-tag them, and insert them into a database. Once the entries are in the database, the users can use this database of geo-tagged microblogs for querying. Users can see the results of each query either textually on a list or spatially on phone's map.

In this assignment, you will get familiar with the design and implementation of a real-world application (of spatial database) using Droid phones and sqlLite database.

You are supposed to implement the following phases:

1- Create local database

In this phase, you need to create your local database. You can get familiar with SQLite Database on Android from this link. When the GeekDroid application is run on your phone, three buttons appear on the screen as in Figure 1. By clicking on the "Create DB", your db table(s) should be created. Note that this snapshot (as well as other snapshots in this document) is provided to give you a better understanding of what should be done. You are not restricted to have the exact GUI as displayed here.

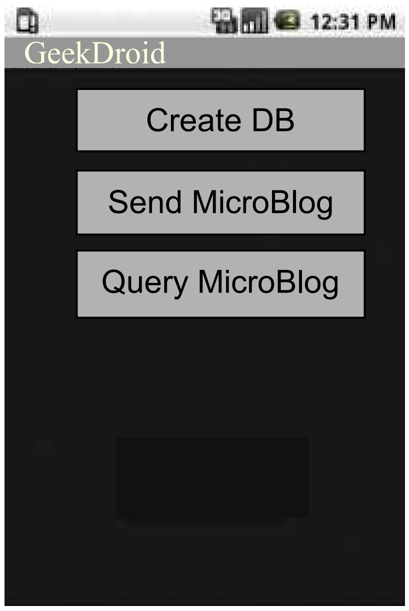

Figure1.

2- Create/Send Microblogs

In this phase, you should use your mobile phone to create and send microblogs. To do this, the user will click on the "Send MicroBlog" button in Figure 1, and that will take him/her to another interface (Figure 2). In this interface, there is a textbox/textfield in which user can write his/her microblog entry. There is also a "send" button on the interface. By pressing the “send” button, the content of the textfield (microblog entry) in addition to its location (phone’s location) will be sent to your local (sqlLite) database on the phone. Note that, in your program, you have to extract the current location from the phone and send it along the blog entry to the database. As the result, this data (geo-tagged blog entry) will be stored in the phone’s database. You also need to assign a unique ID to each entry you send to the database.

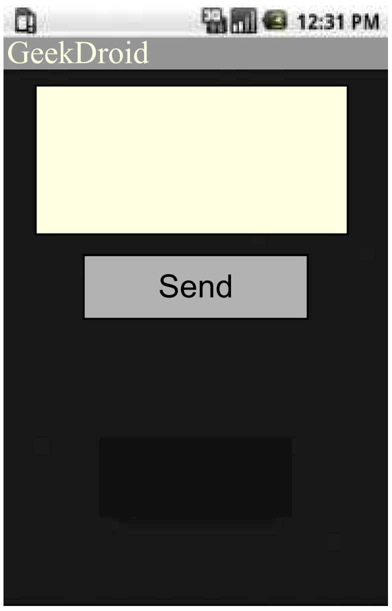

Figure 2.

3.1- Querying Microblogs

Suppose the user is in a location, and wants to see the (relevant) blog entries close to his/her current location. To do so, you need to do as follows. You need to create an interface similar to Figure 3. The user will get to this interface after clicking on “Query MicroBlog” in Figure 1. In this interface, there are five components. A textbox for querying keywords, a textbox to specify the number of results (*k*) and a “search” and “map” buttons to submit the query to the database (note that you will not use “map” button” in this phase). Finally, there is a list which displays the results. In the first text field, user can write 0, 1 or more keywords (separated by space) he/she is interested in. In the second textbox, he/she inputs a number (from 1 to 5) specifying how many results he/she is looking for. By pressing the “search” button, your program should query the database based on the query keywords and query location (which is user’s current location) and returns back top-k results satisfying query parameters. Top-k results are k-closest results (blogs) to the current location containing (at least one of the) the query keywords. All the satisfying results should have at least one of the query keywords. If no keyword is inputted, then the results are only based on the (current) location. You can calculate the distance (closeness) using simple Euclidian distance.

In the output list, you need to show the ranked results (based on the distance to the query location). Each item in the list corresponds to one blog entry and displays its ID, location and first few words of its content (you do not need to display the entire microblog’s content but enough that we can distinguish different entries from each other).

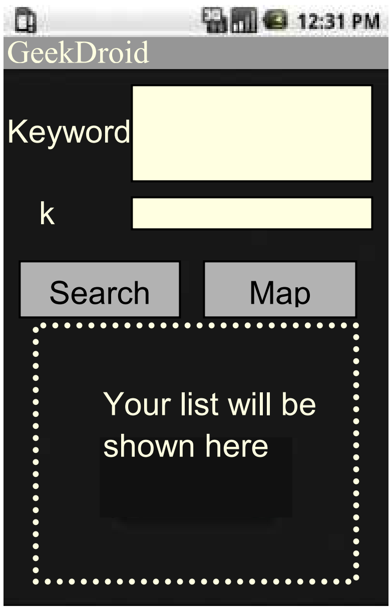

Figure 3.

3.2- Displaying the results on the map

To complete your application, you need to implement one last feature. Now, we want to do the same procedure as step 3.1 but show the final results on the map (Figure 4). You are supposed to do the following. Add another button called “Map” button to the interface in the previous phase. Now, anytime user presses the “map” button, instead of showing the results on a list, you should display them on the phone’s (Google) Map (Note that users can press “map” button instead, before or after pressing “search” button. These two buttons are independent). In other words, after pressing the “map” button, map application of the phone should open and display the results on their corresponding location on the map (results, query and input parameters are the same as the previous step). You have to put one icon for each of the results (blogs) on the map. When user clicks on an icon (blog entry), its content (text) should be displayed to the user.

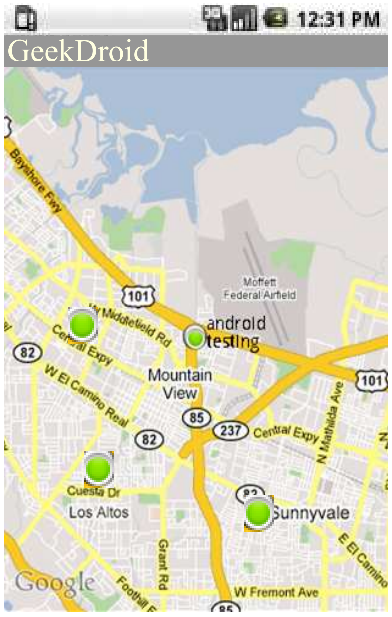

Figure 4

Submission Guideline

1. Make a tar/zip file to include all of your files in one file. You must only send one file (either tar or zip) per group, which includes all your files.

2. Each group needs to submit their assignment electronically using the digital drop box on the DEN website. To do so, you login to DEN, and go to the course page. Click on the Tools option on the left panel and then click on the Digital Dropbox. There you can attach/send your file with the name "HW3-GroupXX". Make sure you see the confirmation that your homework has been successfully sent.

3. You need to have a readme.txt file that should include your group name, student id/name of the students in the group, and any extra information the grader should know about your application. There is 15 points penalty if this file or some of the required information is missing from your submission.
4. Since this is a group project, the group should collaborate, and each of them should know about the details of the code. During the grading, all the members of the group should be present, and be expecting questions from the grader. The grader will email all groups, and assign time slots to the groups to demo their work.
5. DEN students that are working individually, will do the given project on the emulator only.

Distribution of points:

As this table shows, the total points assigned to this homework is 120, which has a 20 point extra credit.

Points	Contents
30	Creating database/table(s)
30	Inserting Microblogs
30	Querying Microblogs
30	Showing on the map
120	