# A Mobile Phone App for Location Based Reminder

Heena Kausar, Asst. Professor, Dept. of Computer Science, S.S. Tegnoor PU & Degree College, Gulbarga, India, heena.mca123@gmail.com

Mazhar Ahmed, Programmer, Dept. of Computer Science, R. M. Polytechnic Diploma College, Gulbarga, India, raza.software.solution@gmail.com

Abstract - **Reminders are important part of our life. They help us in doing day today activities even we get busy with some work. The current mobile applications which set the reminders are based on the time and date. Even though they are good for normal usage sometimes we are in a situation where we don’t know or don’t need any time and date but require to be reminded when we are nearer to any particular location. The main problem of the current reminder application is that the reminders doesn’t have any specific time or date. In this paper we introduce a smart phone Android application which is designed for setting the reminders based on the location. The application creates a geofence by taking the latitude, longitude and the radius around which the reminder needs to be triggered. The User can set the reminders based on specific location. The reminder then goes on when the user enters the radius set for the location. This radius is custom and hence while setting the reminder, user can define when he/she should be reminded. The application also has the ability to set locations as favorite which helps in setting reminders much faster as the location co-ordinates for these favorite locations are pre-defines. This application works on smart phone, tablets that uses android platform**

***Keywords — Geofence, Fused Location Provider, API V2, Google Play Services***

## I. Introduction

Every day we use special messages on order to help us remember future tasks, known as reminders. Reminders may take many forms, such as post-it notes, emails, manual list or an electronic calendar alerts.
For example, taking medicine at 9 am, or student has to return the library books next day etc. The reminders may be time-based reminders or location-based reminders. Time-based reminders, reminds the task should be carried out at specific time. On the other hand, if a task is location-based, we mean that the task should be performed at a specific location, such as to buy notebook while nearer to college. Or in the case when we don't know what time we will be crossing the computer science department but needs to remember to drop off that assignment whenever you do. Or want to buy vegetables when crossing the market today. On the other hand, if a task is location-based, we mean that the task should be performed at a specific location, such as to buy a bottle of milk at the convenient store nearest to your home while returning from the workplace. Or in the case when we don't know what time we will be crossing the science department but needs to remember to drop off that assignment whenever you do. Or want to reminder to pick up medicines when crossing the pharmacy today. In such case, setting an alarm triggered by time may not be appropriate if you are not sure when you will be passing by the convenient store.

These technical and social limitations motivate the focus on location as a context cue. Our aim is to find how location-based reminders are used when available throughout a person’s day. Of course there are types of reminders in which location is not useful, but our focus is on those that could benefit from the additional location information.

The application lets setting reminders by creating a geofence based on the Google Maps API V2. A geofence is a collection of three variables – latitude of the location, longitude of the location and the radius for a particular location. The latitude and longitude helps in pinpointing the location while the radius around this point helps determine when the reminder should be set off. The reminder then goes on when the user enters the radius set for the location. This radius is custom and hence while setting the reminder, user can define when he/she should be reminded.

The other features include defining a final time and date so that if the user doesn’t go to a particular location set for the reminder until this set time and date then the reminder will go on telling the user about the reminder and also giving the option to reset the final time and date to keep the reminder active. While setting the reminder, user also sets the location of the reminder i.e. the place where this reminder needs to be triggered. When the reminder is set, the geofence is formed. This geofence gets activated when the user starts moving and if the user enters any of the geofence on the map, the reminder gets triggered reminding the user about the work that needs to be done. The map is shown in two formats – satellite view and normal view.

The application also has the ability to set locations as favorite which helps in setting reminders much faster as the location co-ordinates for these favorite locations are pre-defines.

DOI : 10.18231/2454-9150.2018.0479

## II. SYSYEM ANALYSIS

### A. Existing System

The existing application has the ability to set reminders based on date and time. The problem with the system was that the user crosses the specified location but they was unable to get the task reminder about the location and if they are at another location then task reminder alert is displayed in this way there was no proper synchronization between task reminder and specified location. For example when you have to be reminded about getting something from friend's home. At this time, user cannot be sure when he/she will be near friend home. So a time and date reminder won't be helpful over here.

Some drawbacks of existing system are as follows

- Searching for a location is difficult.
- Location cannot be set accurately.
- Only one task reminder can be set on one location.
- Date and time range is limited.
- Task cancellation and rescheduling the task is difficult.

To overcome these drawbacks we proposed the application entitled "A Mobile App for Location Based Reminder".

### B. Proposed System

The proposed application has the ability to set reminders based on location. It works by creating a geofence around some location. A geofence consists of three things – the latitude of the location, the longitude of the location and a radius that needs to be formed around the location. While setting the reminder, user also sets the location of the reminder i.e. the place where this reminder needs to be triggered. When the reminder is set, the geofence is formed. This geofence gets activated when the user starts moving and if the user enters any of the geofence on the map, the reminder gets triggered reminding the user about the work that needs to be done. The map is shown in two formats – satellite view and normal view.

The satellite view shows the map in real time such as roads, trees and buildings while the normal view shows the map as in a map book.

This application also allows the user to set multiple task reminders at the same location. And when the user reaches the specified location all the tasks that have to be performed are notified to the user so that no jobs are missed or left incomplete.

The main feature of the proposed system is that the user can search the location by entering the location name when the user is do not know where the location is exactly situated.

Advantages of Proposed System

- User friendly interface and flexible to use.
- Location is selected using map that is shown to user.
- The map is shown in satellite view and normal view which helps cater to different types of users.
- Compact in design.
- Allows the user to set multiple tasks in the same location.

## III. SYSTEM REQUIREMENT

### A. Hardware Requirements

Hardware used:

- Core i3/i5/i7 processor.
- 50GB HDD.
- 2GB RAM.

Minimum requirements:

- Android phone with OS 4.0 and above.
- GPS enabled.

### B. Software Requirement

The software required for the development as listed below:

- Windows OS.
- Eclipse ADT bundle.
- JDK 1.7.x_x
- Android Studio Toolkit
- Eclipse ADT IDE

Eclipse ADT IDE is used to develop the application. This IDE comes with the ADT (Android Developer Tools) which are required to compile and run the application. It contains an SDK and AVD manager. The SDK manager is used to download Android Software Development Kit (SDK) which contains the platform image files of the different versions of android which need to be specified for development. The Android Virtual Device (AVD) manager helps in creating a virtual android device as the name suggests. This virtual device can be used to test the application.

## IV. MODULES DESCRIPTION

After careful analysis the proposed application is divided into following modules.

- Database
- Map
- New reminder
- Settings
- Favorites

### A. Database

This is the first module of the project. It contains the list of the entire reminders listed one below the other. After clicking on any of the reminder in the list, user is given the option to edit or delete the reminder.

### B. Map

This module contains Google Map which is displayed for the sake of selecting the location by clicking on the page. After clicking on map, a marker appears to indicate the place that has been selected with a red circle around it that indicates the geofence.

### C. Set reminder

This page contains fields for title, description and location name which is entered by the user. All these fields are

DOI : 10.18231/2454-9150.2018.0479

necessary for user understanding as the marker is hardcoded containing the latitude and longitude of the marker point which is not in user understandable format.

### D. *Settings*

This page contains settings related to the notification of the reminder such as notification tone, vibration preferences, etc.

### E. *Favorites*

This is another page that contains the list of locations that the user has selected as favorite. The favorite locations are those for which a user will be setting reminders regularly. This helps the user in setting reminders much faster.

## V. Literature Review

### A. *About Google API Map*

The Google Maps API is responsible for accessing the Maps from the Maps database from Google and showing it on the user's device. This is done by integration of the Google Play Services library which needs to be downloaded from the SDK manager. Apart from providing the Maps integration, it also provides the different classes and interfaces which are necessary for correctly using and displaying the maps.

Google Maps API comes with various features such as getting user location, showing different types of maps (hybrid, satellite, etc.), creating geofence, etc.

### B. *Google Play Services*

This is a library that needs to be imported into the application which contains various classes and interfaces like map view, maker, etc.

### C. *About Java*

Java is a computer programming language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "write once, run anywhere" (WORA), meaning that code that runs on one platform does not need to be recompiled to run on another. Java applications are typically compiled to byte code (class file) that can run on any Java virtual machine (JVM) regardless of computer architecture. Java is, as of 2014, one of the most popular programming languages in use, particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by James Gosling at Sun Microsystems (which has since merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

One characteristic of Java is portability, which means that computer programs written in the Java language must run similarly on any hardware/operating-system platform. This is achieved by compiling the Java language code to an intermediate representation called Java byte code, instead of directly to platform-specific machine code. Java byte code instructions are analogous to machine code, but they are intended to be interpreted by a virtual machine (VM) written specifically for the host hardware. End-users commonly use a Java Runtime Environment (JRE) installed on their own machine for standalone Java applications, or in a Web browser for Java applets.

Standardized libraries provide a generic way to access host-specific features such as graphics, threading, and networking. A major benefit of using byte code is porting. However, the overhead of interpretation means that interpreted programs almost always run more slowly than programs compiled to native executable would. Just-in-Time (JIT) compilers were introduced from an early stage that compile byte code to machine code during runtime.

### D. *About Android*

Android is an operating system based on the Linux kernel, and designed primarily for touchscreen mobile devices such as smartphones and tablet computers. Initially developed by Android, Inc., which Google backed financially and later bought in 2005, Android was unveiled in 2007 along with the founding of the Open Handset Alliance—a consortium of hardware, software, and telecommunication companies devoted to advancing open standards for mobile devices. The first publicly available smartphone running Android, the HTC Dream, was released on October 22, 2008.

The user interface of Android is based on direct manipulation, using touch inputs that loosely correspond to real-world actions, like swiping, tapping, pinching, and reverse pinching to manipulate on-screen objects. Internal hardware—such as accelerometers, gyroscopes, and proximity sensors—is used by some applications to respond to additional user actions, for example adjusting the screen from portrait to landscape depending on how the device is oriented. Android allows users to customize their home screens with shortcuts to applications and widgets, which allow users to display live content, such as emails and weather information, directly on the home screen. Applications can further send notifications to the user to inform them of relevant information, such as new emails and text messages.

### E. *About SQLite*

SQLite is a relational database management system contained in a C programming library. In contrast to other database management systems, SQLite is not a separate process that is accessed from the client application, but an integral part of it. SQLite is ACID-compliant and implements most of the SQL standard, using a dynamically and weakly typed SQL syntax that does not guarantee the domain integrity. SQLite is a popular choice as embedded database for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems, among others. SQLite has many bindings to programming

DOI : 10.18231/2454-9150.2018.0479

languages. The source code for SQLite is in the public domain.

## VI. System Design

Software design sits at the technical kernel of the software engineering process and is applied regardless of the development paradigm and area of application. Design is the first step in the development phase for any engineered product or system. The designer’s goal is to produce a model or representation of an entity that will later be built. Beginning, once system requirement have been specified and analyzed, system design is the first of the three technical activities -design, code and test that is required to build and verify software.

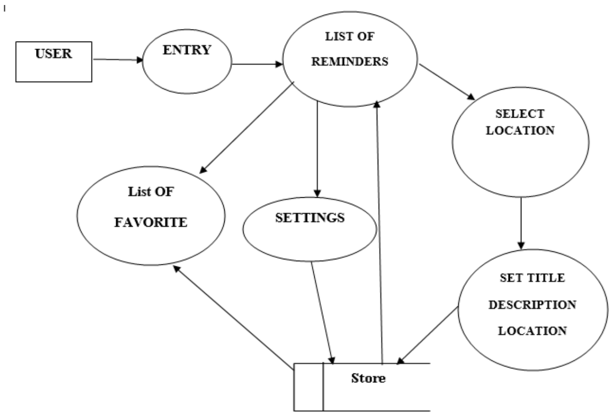
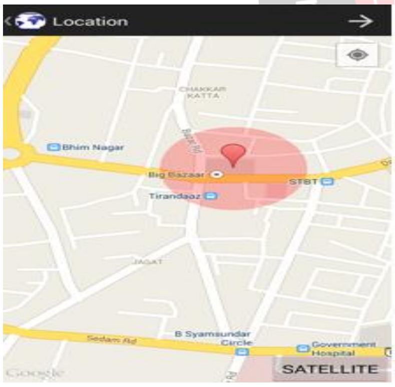


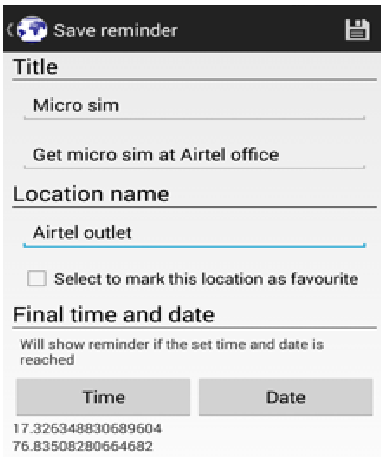
Fig: 1 First Level Data Flow Diagram

## VII. Work Flow

### 1. Reminder Activity



### 2. Set Reminders



Save reminder

Title

Micro sim

Get micro sim at Airtel office

Location name

Airtel outlet

□ Select to mark this location as favourite

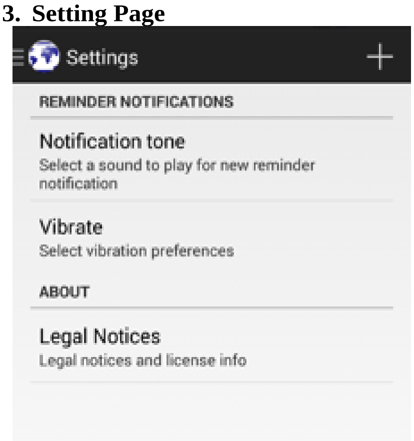
Final time and date

Will show reminder if the set time and date is reached

Time | Date

17.326348830689604
76.83508280664682

### 3. Setting Page



Settings

REMINDER NOTIFICATIONS

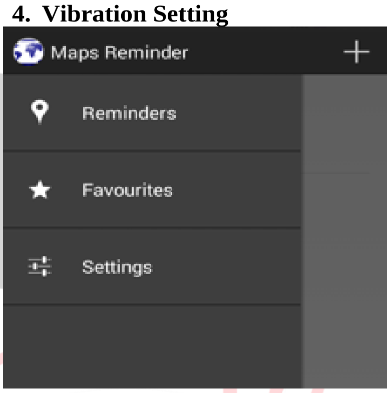
Notification tone
Select a sound to play for new reminder notification

Vibrate
Select vibration preferences

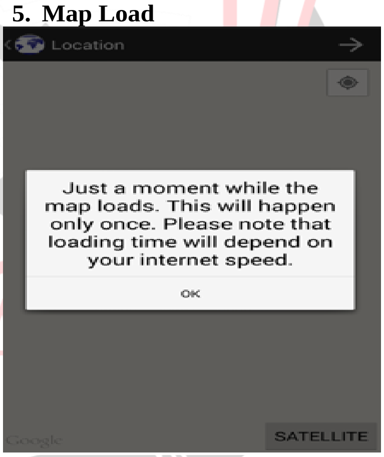
ABOUT

Legal Notices
Legal notices and license info

### 4. Vibration Setting



### 5. Map Load

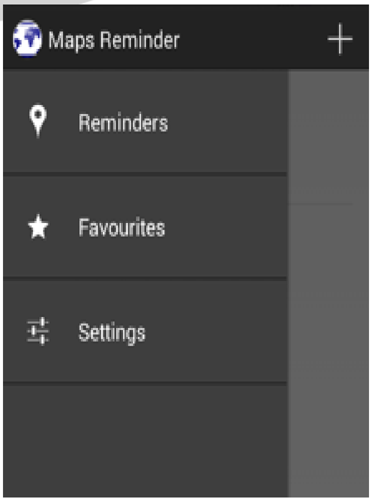


Location

Just a moment while the map loads. This will happen only once. Please note that loading time will depend on your internet speed.

OK

### 6. Application Option



DOI : 10.18231/2454-9150.2018.0479

## VIII. Implementation

The proposed approach was implemented in Google Android platform. The android platform was chosen to exploit the user friendly features of android and this work used the Google search engine as the backend search engine. In Android, every application runs in its own process, which gives better performance in security, protected memory and other benefits. Therefore, Android is responsible to run and shut down correctly these processes when it is needed. It is important that application developers understand how different application components (in particular Activity, Service, and Broadcast Receiver) impact the lifetime of the application's process. Not using these components correctly can result in the system killing the application's process while it is doing important work. To determine which processes should be killed when low on memory, Android places each process into an "importance hierarchy" based on the components running in them and the state of those components.

## IX. Conclusion

The app Location Based Reminder is designed to set reminders based on locality. The app has been developed on Android platform using Java and XML for the frontend and SQLite as the backend. It has been designed in such a way that the app provides info (it reminds the user) at the proper time needed. It also gives the ability to set an end time and date for the reminder so that if the user doesn’t go to particular locality before the set date and time then the reminder goes off giving the user to reset the reminder or delete it if it’s not valid anymore.

The application works by creating a geofence around the point that is selected in the map. This geofence is a special class provided by Google which using which code can be written to calculate if the user is inside any geofence. The Fused Location Provider is another class that provides updates to location continuously based on GPS, Network Provider and Wi-Fi. All these combined together helps in calculating user location and setting the reminder.

In this paper we have put forth the idea of a modern reminder application which makes it more useful to potential users. The application which is android based uses technologies such as GPS and internet connectivity to make reminders more context aware by adding the dimension of location to traditional time based reminders. We also enlisted the considerations to be taken care of while developing a location based application and described the proposed features.

## References

[1] Babic, J., & Podlubný, I. (2011, May). Software application for GPS devices using Google Maps. In Carpathian Control Conference (ICCC), 2011 12th International (pp. 1-5). IEEE.

[2] Dey, Anind K., and Gregory D. Abowd. "Cybreminder: A context-aware system for supporting reminders." International Symposium on Handheld and Ubiquitous Computing. Springer, Berlin, Heidelberg, 2000.

[3] Battin, Pradnya, and S. D. Markande. "Location based reminder Android application using Google Maps API." Automatic Control and Dynamic Optimization Techniques (ICACDOT), International Conference on. IEEE, 2016.

[4] Sierra, Kathy, and Bert Bates. Head First Java: A Brain-Friendly Guide. “O’Reilly Media, Inc.", 2005.Java Complete Reference by Herb Schildt.

[5] Categorical Index of SQLite Documents, [online] Available: https://www.sqlite.org/docs.html.

[6] Eclipse documentation, http://www.eclipse.org/documentation/

[7] Google Maps Android API v2. https://developers.google.com/maps/documentation/android/

DOI : 10.18231/2454-9150.2018.0479