

# Location Based Chat Android Application

Vivek Bhavsar

Department: Information Technology  
College: PCET NMIET, Talegaon  
Pune, Maharashtra, India  
vivek131299@gmail.com

Yash Bhakre

Department: Information Technology  
College: PCET NMIET, Talegaon  
Pune, Maharashtra, India  
bhakreyash18@gmail.com

Rohit Pradhan

Department: Information Technology  
College: PCET NMIET, Talegaon  
Pune, Maharashtra, India  
rohit.pradhan.3192@gmail.com

Desmond Rodrigues

Department: Information Technology  
College: PCET NMIET, Talegaon  
Pune, Maharashtra, India  
dezu252000@gmail.com

Prof. Sushma Bhosle

M.E. E&TC  
College: PCET NMIET, Talegaon  
Pune, Maharashtra, India  
sushma.bhosle28@gmail.com

**Abstract**—As we know, to communicate with a person we may need at least some information like a person's contact/mobile number in order to chat with him/her. There may be instances where we want to contact someone nearby and we do not have their contact details, in such a situation our app plays a crucial role in the communication. To overcome this, we came up with an idea of making an app which will use the location/GPS of a user's device to locate nearby users. We are also aware that the privacy policy of most of the popular apps state that they would sync our data with their parent company for advertisement purposes. Old chatting applications lack security, and the addition of more security layers is our major priority. As the old Chinese apps are banned by the Indian government we felt the need to develop an Indian app which can contribute towards the 'Make in India' movement.

**Keywords**— GPS, Chat, Sharing, GPS Parental Control.

## I. INTRODUCTION

At the advent of the 19th century no one knew that a device would be created that would change the face of the world, the way that we would communicate with each other would no longer be the same. It is the invention of the telephone. Neither did Alexander Graham Bell know that he was about to create history with his invention. To communicate over long distances we had to make trunk calls which took an hour to connect to the end user. These trunk calls were expensive to make.

As humans underwent evolution our communication devices also evolved to become our personal smartphone that we know today. Our smartphones have replaced a number of things that we used in the past for e.g radios, calendars, cameras, pager, letters, alarm clocks etc. Due to replacement of these gadgets by a smartphone, it has become more popular and the demand for high end applications has arisen.

From 1875 we have come a long way, our ways of communication have changed, we no longer depend only on telephonic calls, we now are mostly dependent on SMS, text, images to communicate. More than 2.5 million android devices are activated each day and more than 70 billion app downloads from Google play in each month. This clearly shows that the usage of mobile devices is widespread. In this widespread network of mobiles, Android operating system is the main environment because it has many features like:

1. It is Open source and based on Linux.

2. Many built-in features for developers to use in their apps.
3. Management of the lifecycle of an app is automatically done.
4. Portable for a wide range of hardware which makes it future proof.

Android Applications are developed by using SDK (Software Development Kit). It supports mainly Java and Kotlin languages.

Basic steps for Android App development:

1. Setting up the environment for development which includes Android SDK, IDE, database.
2. Actual Android app development.
3. Debugging the app by running the developed application and checking for further improvements.
4. Preparing to release the application for public use and improve time-to-time by user feedback.

Existing chat apps require the user's personal contact number to open an account and to communicate with one another. Sharing personal information like one's phone number may prove to be dangerous. There are chances that the data that we possess may be used by these firms in an unethical manner. Also, extra permissions may be requested from users which may not be required for an app to function. Many of them also do not provide an option for logging out users.

While developing our application, we have overcome the disadvantages of existing applications like contact information security, location related security, communication with people in vicinity and parental control related features.

## II. SYSTEM

### A. Design(GUI):

Location based chat app, an app designed while keeping user's needs into consideration. The User Interface is modern yet simple to use made by following a minimalistic design approach.

First time users will interact with the registration page, on which he/she will need to fill in details such as Username, email id and also set a strong password, on doing so one's account will get created successfully. On completion of this step one may need to login using the same credentials from the previous step.

After successful logging in of the user, he/she will be displayed 3 sections namely Chats, Map (nearby users), Profile. The chats section displays all the individuals/groups that the user has had a conversation with. The Maps section shows all the nearby users in the selected radius. And the profile section shows the user's profile in which his/hers profile picture, username and email id will be displayed.

In the Users section, users will be able to select a specific radius for searching nearby users from the given options which are {1km, 5kms, 10kms, 25kms, 50kms}. After selecting the specific radius, the users in that radius will be displayed on the map with the pin(marker). If one wants to chat or share some file to any user, then one can click on any of the pins(markers) and then he/she can chat and share files with each other and that person's chat will be automatically added into the Chats section once they start communicating.

After the user starts to chat, he/she can also share files within the chat itself. To share files, one needs to click on the attachment icon. Files such as images, videos, documents, etc. can be shared securely.

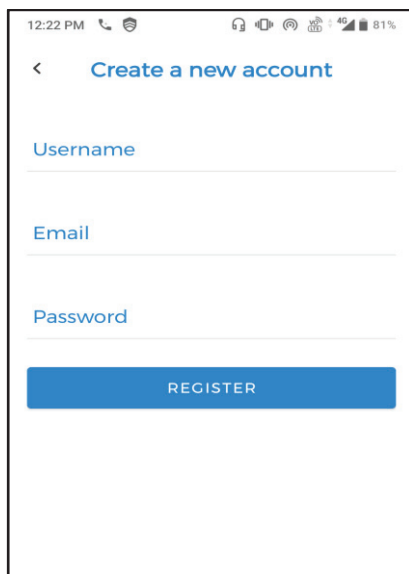


Fig. 1. User Registration.

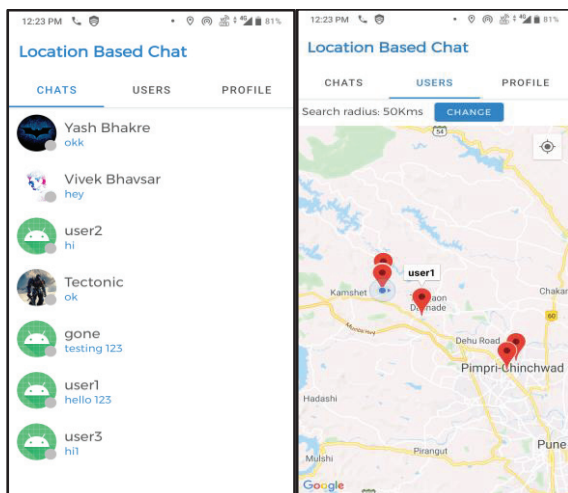


Fig. 2. Chats section.

Fig 3: Map section.

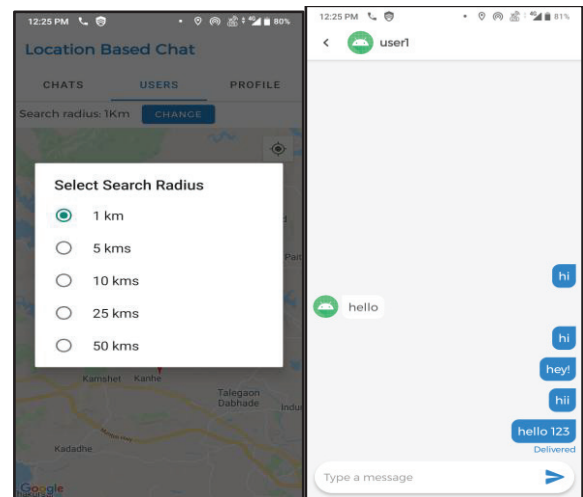


Fig. 4: Radius selection.

Fig. 5. Personal chat.

## B. Working

After creation of the user's account the account gets added on the Firebase which is a Google owned cloud database. While the user tries to login, authentication of the user's email-id and password is itself done by Firebase authentication service. After logging-in the user the 'status' field of the user in the database changes from 'offline' to 'online'. Once the user is online, the app requests permission for accessing the user's location by turning ON GPS. Once a user permits necessary permissions, an app starts to fetch the user's current location and stores it into a database in the form of latitude and longitude until the user turns OFF the location access.

While the location sharing switch is turned ON by the user, the location of the user will be continuously updated into the database with the time interval of 1000 milliseconds(1 second). Once a user's location is updated in the database, nearby users can see him/her on the map and are ready to communicate, if he/she is in a selected radius. After the user changes the selected radius, the newly selected radius will be passed to the query of GeoFire to search available users in the new radius.

After the nearby available users are displayed on the map, the user can click on one of the users for starting to chat with them. When the user clicks on any specific pin(marker) on the map, the UID(unique ID) of that user will be passed into a new activity for starting chat and sharing files. Once the user starts the chat with anyone by sending a message or a file, then that particular chat will be automatically added into the 'Chats' section of the app.

Users will also be able to share files once he starts to chat with any user. After selecting a file which needs to be shared, the file gets uploaded into the database and the receiver's device fetches and downloads that file from the database and gets stored into the local storage of the device.

We also have given a feature for creation of a room/group, this feature is valuable when one wants to create a group for the people to contact him/her in nearby vicinity. This group will be visible only within the radius which the group admin has set while creating it. Once any user comes into that radius, he will be able to see the group in the map and he can join it and start communicating with the

participants of that group until he/she exits from the permissible radius.

A small but necessary feature of parental control is provided in this application, the main use of this is for parents to track their child's activity while they are travelling. In this feature, the parent first sends a request to the child's mobile device to allow tracking, then the child is required to accept the request. Once the request is accepted, the parent can set a radius for the child on the map. Parents will be alerted through a notification if the child exits the selected radius, also if the child enters into the selected radius(Refer Fig: 10).

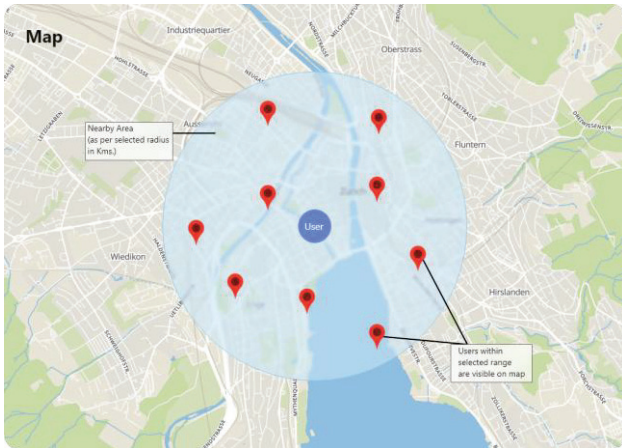


Fig. 6. Concept Diagram.

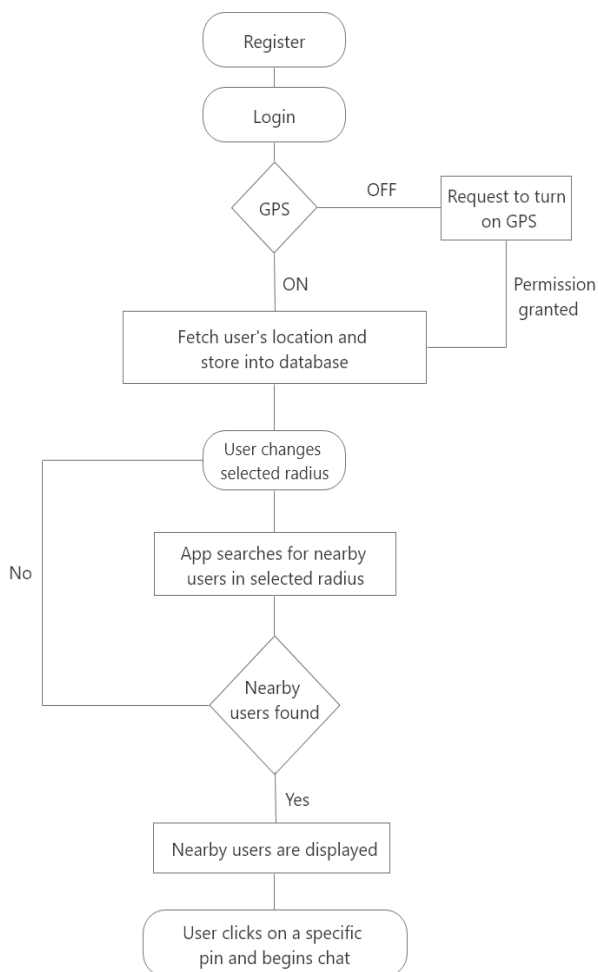


Fig. 7. Flow Chart

### III. IMPLEMENTATION

As a software developer, for making a successful bug free software we have to integrate different tools for different purposes like database handling, User Interface (UI) designing, middleware. Apart from these tools we may have to use APIs (Application Programming Interface) for some additional features which may be open source or paid depending upon API developer's policy.

Initial idea of GUI was expressed using Adobe XD software. We created a modern and simplistic UI which will be easy for any type of user to use.

For developing an actual android app, the most popular IDE available today is Google's and JetBrains Android Studio. The two main languages supported by Android Studio are Java and Kotlin for backend and XML for frontend development.

For connecting Firebase Database, GeoFire, Google Maps to the app, we have added necessary dependencies into the build.gradle file of the project.

#### *build.gradle file*

```

implementation 'com.google.android.gms:play-services-maps:17.0.0'

implementation 'com.google.firebase:firebase-auth:20.0.2'

implementation 'com.google.firebase:firebase-database:19.6.0'

implementation 'com.google.firebase:firebase-core:18.0.2'

implementation 'com.google.firebase:firebase-storage:19.2.1'

implementation 'com.google.firebase:firebase-messaging:21.0.1'

implementation 'com.firebase:geofire-android:2.1.1'

implementation 'com.google.android.gms:play-services-location:17.1.0'
  
```

For accessing various services of a user's device, we need to request necessary permissions from the user. We have to specify these permissions in the AndroidManifest.xml file.

#### *AndroidManifest.xml*

```

<uses-permission
    android:name="android.permission.READ_EXTERNAL_STORAGE" />

<uses-permission
    android:name="android.permission.INTERNET" />

<uses-permission
    android:name="android.permission.ACCESS_FINE_LOCATION" />

<uses-permission
  
```



```

android:name="android.permission.ACCESS_COARSE_
LOCATION" />
<uses-permission
android:name="android.permission.WRITE_EXTERNAL_
STORAGE" />

```

For finding nearby users in a specified radius, we have used GeoQuery in the `getUsersAround()` function in `UsersFragment.java` file.

```

DatabaseReference usersLocation =
FirebaseDatabase.getInstance().getReference().child("UsersLocation");

GeoFire geoFire = new GeoFire(usersLocation);

GeoQuery geoQuery = geoFire.queryAtLocation(new
GeoLocation(mLastLocation.getLatitude(),
mLastLocation.getLongitude()), search_radius);

```

For displaying those fetched users onto the map, we have used the `addMarker` function of Google Maps API.

```

LatLng userLocation = new LatLng(location.latitude,
location.longitude);
Marker mUserMarker = mMap.addMarker(new
MarkerOptions().position(userLocation).title(user_name))
;
mUserMarker.setTag(key);
mUserMarker.showInfoWindow();
markerList.add(mUserMarker);

```

For storing user's data, we have used Google's Firebase Realtime Database. It can handle unstructured data and stores it in collections.

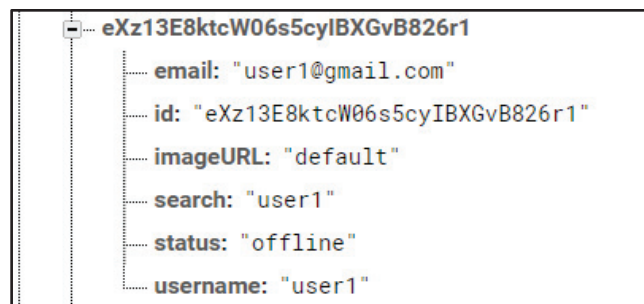


Fig. 3. User's information in Firebase

The user's location is stored in the form of latitude and longitude in the Firebase database which is stored and fetched by using GeoFire's GeoQuery.

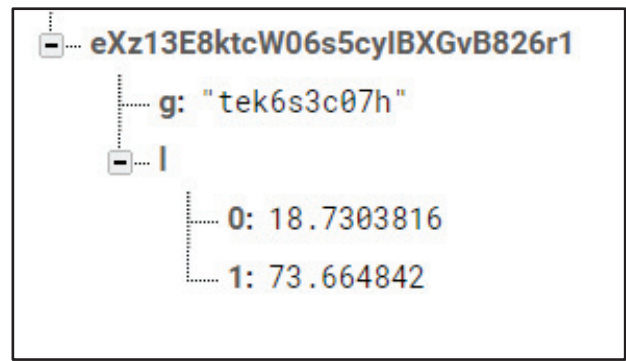


Fig. 8. Location of the user in Firebase.

#### IV. SECURITY:

Over the past years technology has given rise to a great number of inventions but on the other hand it has also produced some ill effects on society when fallen into the wrong hands. There are people who use this technology in an unethical manner to harm another user's privacy. This is the reason why large IT companies spend a large sum of their income on privacy and security. In our application we have given high priority to the users privacy, and have considered their data to be a valuable commodity.

After installing an application from Play Store, these apps ask for unnecessary permissions from the user. Our application on the other hand does not ask for these permissions. We have added more security by providing a feature to turn OFF your location when the app is not being used.

The user's chats are securely stored on the Google Firebase server. We have also provided a chat disappearing feature in our application, by turning ON this feature, after every 100 messages, the messages disappear and are sent to the users email id for privacy purposes, after the chats are sent to the user these chats are deleted from our firebase server.

The additional feature of parental control can be considered a great security feature. Parents can track their children's activity when they are travelling.

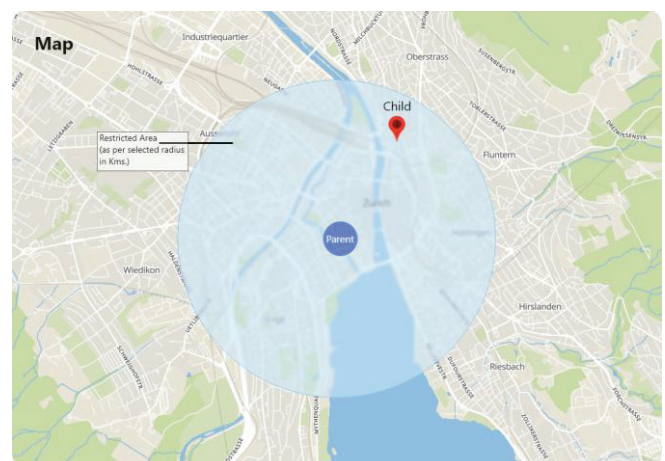


Fig. 9. Concept diagram for parental control.

## V. APPLICATIONS:

1. No more dependency on a person's phone number for starting a conversation.
2. Finding nearby users and stores for communication : If one is in need of a commodity which can be bought in a store he/she can search for nearby stores in the maps section and can chat with the shopkeeper if he has stock of the required commodity.
3. As the location gets updated per second it can be used for tracking a child's activity in parental control.
4. One can enquire for any information at any office, store, mall, college without going on their website or calling them.
5. Groups can be formed at certain locations so that users nearby that location can join that group for communication.

## VI. CONCLUSION:

With the rapid growth in android devices and computer networks, we have seen a massive outburst of data produced. The data produced is rising day by day. The necessary information that we want to send to another user has to be sent in a secure manner. Our application that we developed

will serve as a very useful application to its users upon release to the market. It will enhance the customer's experience and privacy. Hence, we have designed and initiated the implementation of an android application which will allow users to chat and share files with the nearby users within the selected radius (in kms) displayed on the map with the help of GPS, and the user's data being securely stored in our databases. Some of the important features of our app is one can start communicating with another person without having his/her personal contact information, one can track their child's activity using the parental control feature, location can be turned ON/OFF by the user whenever required for privacy concerns.

## REFERENCES

- [1] Location Based Mobile Apps Development on Android Platform Bhuvana Sekar and Jiang B.Liu Computer Science & Information Systems Department Bradley University Peoria, Illinois, U.S.A. [jiangbo@bradley.edu](mailto:jiangbo@bradley.edu)
- [2] Where2Buy: A Location-based Shopping App with Products-wise Searching Kin Chi Chan\*, Tak Leung Cheung\*, Siu Hong Lai\*, Kin Chung Kwan\*, Hoyin Yue† and Wai-Man Pang\* \*The School of Computer and Information Sciences, Caritas Institute of Higher Education, Hong Kong †Department of Computing, Hang Seng Management College, Hong Kong
- [3] <https://developer.android.com/studio/intro>