Implementation and Analysis of a Location Based Community Organizer

Institute of Computer Science
Georg-August-Universität Göttingen

Advisor: Dr.David Koll
Lecturer: Prof. Dr. Xiaoming Fu
Author: Hari Raghavendar Rao Bandari
11334055
h.bandari@stud.uni-goettingen.de
Seshagiri Prabhu
21410690
seshagiri.prabhu@stud.uni-goettingen.de

Date: April 22, 2015

Abstract

The aim of this project is to develop a mobile social networking application, eventually using android OS platform with a dedicated web server for the community, therefore the concept is to share your location and activities within your closed social network. For example If two members of a community are near to each other in regards to their location immediately our android web application server will trigger events, therefore our android application allows community members to organize a meeting point. Although events can be created right away based on the member's location or they can create an event activity offline also.

Contents

List of Diagrams

1 Introduction

The online social network application plays a major role in our daily life, it has changed our way of interacting with friends in the computer world. These activities, manage contacts with aged friends or to gain new friends. Online social networking applications such as Facebook places have millions of users, therefore users will share location details, merely it is centralized social network and eventually it causes few problems, every social network application stores information of the user, later user doesn't have any control over their own personal data. The user is now treated as product for the company immediately all social based companies sell user data to other third party e-commerce companies to promote their products to customers, and respective individual data supports them to analyze around customer behavior.

In conclusion, we suggest a unique solution to the above problems to have dedicated decentralized server rather depending on a centralized server. With our approach users data is secure and safe. Our android application provides the same level of user interaction as compared to many social networking applications. It also allows users to have more control over their own personal data. As our android application web server will be administrated by community organizer member only.

1.1 Motivation

Our day-to-day relaxation activities are occupied with the exiting social network application, but every social activity services are centralized, with this user doesn't have any control of their individual data. Therefore, we developed an android application with a dedicated web server for the community and it will be administrated within the community group by the community member. In conclusion, our decentralized approach provides higher data security along with complete control of their individual data.

Therefore project has two main perspectives. They are User's perspective: sharing location details to closed community and gaining higher data security with decentralized approach. Research perspective: Periodically, our web server will collect android application usage data, along with other social activities.

1.2 Usage of Smart phones

The utilization of smart phone devices is immensely growing every year. Therefore emarketer Germany gives information about the smart phone usage is about 44.5% millions in 2015. According to Nielsen 28% of user spending their leisure time on a social networking application. The need of smart phone is required for young generation to stay in contact with family and friends along with this usage of smart phone is increased due to all services are mobile.

According to the mobile device OS market, Android OS leads to be a wiser choice over other mobile OS. By the end of 2015, it is expected that the Android OS will enjoy a 48.8 percent portion of the global market. This reveals the custom of the Android OS.

1.3 Web Services

A web service is a way of connecting two computerized devices over the network. It also provides a facility to switch data between particular applications and particular platforms. Therefore, in our instance, the android device is connected to a Django REST Framework through retrofit REST Client Android API. This technique will help to link our android application data easily in a secure way with our dedicated server.

1.4 Contribution of project

We propose a design of decentralized social community, with our proposed approach, members in the community organizer can safely store all personal data and can easily share among friends in a closed community.

For example below figure :1 explains, if two members are closed to each other our server request locations information of each member. Once server receives the location information of every individual member in community, the server will send the results back to the android device to display other members location details to each other

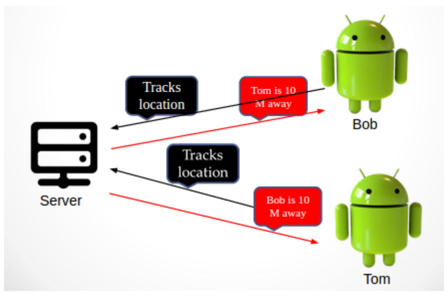


Figure 1: Diagram showing two user interaction with web server[22]

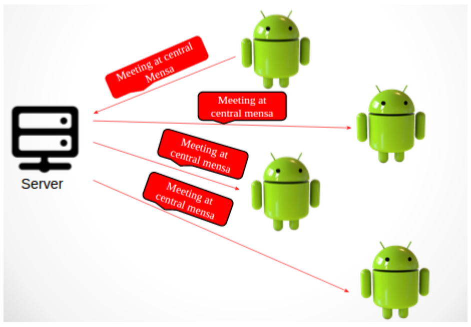


Figure 2: Diagram describe how events are created and shared[23]

Above figure:2 explains about how members in the community can create a meeting activity instantly and share particular event details with other members in the community group regards to location distance. Any member in the group can create a event activity then our web server will share that information to other members in the community.

2 Underlying Technologies

2.1 Client Technologies

Geo-fencing

Geo-fencing is a technology which uses cellular phone tower and WiFi network to provide the user location to the service provider. It also uses up very less battery of the smart phone device. Therefore, it provides valuable information about the user latitude and longitude without tracking continuously user location, but it will create a virtual boundary around a real-world geographical area for each Geo-fence we can ask location service about entry and exits. It automatically alerts the users based on delay time to notify whenever the user goes into or exits the Geo-fence radius.

For instance, It generates virtual boundaries when the user enters into Geo-fence area the user receives the notification about the Geo-fence and even when a user exits it will notify. Here Geo-fence doesn't provide exact user location data, but it notifies whether the user is in the radius of the particular Geo-fence or not.

In conclusion, we have created and monitored few Geo-fences to know the user location without draining mobile battery life.

Drawbacks

Geo-fence has limited accuracy and highly depends on the density of transmitter.

Google Play Service Location API

One of the most significant feature in the smart phone is Location awareness. Therefore, users carry mobile always with them it will be advantageous to recognize the user information and it will be helpful to tag photo's location easily. Furthermore, we can achieve automated location tracking, Geofencing, Activity recognition.

Google Play Service Location API provides 1. User last Known location: Once user connected to google play store it uses fused location provider to retrieve user device current location. 2. Receiving Location Update: To receive periodic updates of the user location again we use fused location provider and in response it will update the most recent location of the mobile device based on WiFi or GPS 3. Displaying Location Address: It helps to convert latitude and longitude to the corresponding meaning full address to the user 4. Creating and Monitoring Geofences: It creates and monitors the Geo-fences based on the latitude and longitude with the radius limit and it will report the user enters/exits into virtual boundaries.

Active Android

It is an active record style ORM (object relational mapper). It gives flexibility to save and retrieve SQLite database records in the android mobile phone internally without writing any SQL statements. Therefore, each record in the database is wrapped into a class with easy understandable methods like save () and delete (). In android, accessing the database is a hassle, active android plays a major part which handles all the set-up and messy stuff, and with only few understandable steps of configuration.

ReactiveX

It is a library which handles to build asynchronous and event based programs by using observable sequence and query operators.

It extends the observer patterns which maintains a list of its dependents, which will notify automatically of any state change to support a sequence of data and events, adds respective operators that allows to compose sequences jointly and concern about few things like low level threading, concurrent data structure and non blocking I/O.

Retrofit

Retrofit is third party free Android library that will allow our API in simple Java interface and automatically converts into a REST Client.Its an good solution for organizing API calls in a project. The request method and relative URL are added with an annotation, which makes code clean and simple. It uses Gson by default, so there is no need for custom parsing.

2.2 Server Technologies

RESTful-Django REST Framework

The Django REST framework is a powerful tool and make it easy to build a web API it returns a raw JSON by default. The Django Rest Framework provides powerful model serialization that supports object relational mapper and non object relational mapper data sources, display data using standard function based views, or get granular with a powerful class based views for more complex functionality. All in a one complete, fully REST compliant wrapper. The most used and trusted by large companies like Mozilla and Evenbrite.

Drawbacks

Django REST Framework can used for production purpose. But the performance is comparatively less when compared to other Web Frameworks.

3 Architecture

There is an Android application, the web server, and the database. The Android application behaves as a client. The user interacts with the client in order to use our decentralized approach of the community group to share location and generate event activities. The client interacts with the web server through retrofit REST Client.

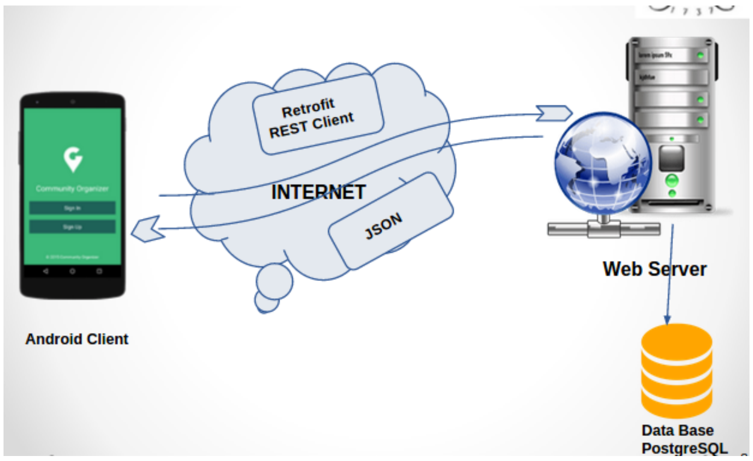


Figure 3: Diagram Communication between Android Application and web server 25

The web server on the other hand plays as the connecting link between the client and the database. It can be imagined as the middleman of the system. It exchanges messages with the client, the database. The web service should offer all the necessary information to the client (and indirectly to the user), while at the same time it ensures the security and the well-being of the system.

The user data will be stored in the database. As soon as data is stored in the database, it will be portioned out among other users in the community group. On the other hand, Android Application will interact with the web service through retrofit REST Client connection and format of communication is "JSON" format.

Ultimately, a database is used to store the received location information as well as created event activity details. Apart from this our database keeps track of newly added friends list and usage of our android application.

Our android application uses a local database in android mobile to store all necessary details, later it will periodically send all information to the web server.

4 Requirements

Client Requirements

4.1 Integrated Development Environments (IDE)

An Integrated Development Environment is a PC application software that provides the developer with all necessary tools for a successful software development. Usually it consists of three main parts:

1. Source Code Editor: It is a window where code will be written.
2. Build Automation Tool: It will build all respective tools for coding.
3. Debugger: It will help to debug the code.

4.2 Android Studio

Android Studio is the authoritative Integrated Development Environment (IDE) from the Android google team. It is built on the popular IntelliJ IDEA, Java IDE and with the release of android studio it is easy for android developer to develop Android applications in a well structured way, we can access a new set of features to enable your development workflow. Android Studio offers few extraordinary valuable and less time consuming for developing Android applications they are: 1. Gradle-based build system offers new way of adding dependencies and third party libraries. 2. Every android application will have its own window to work generates multiple apk files. 3. It is now easier and less time consuming to build common application. 4. Instead of coding just drag and drop items will create a view interface where user will interact. 5. It explains about the application consuming time to load and performance etc. 6. It is easier to add an our own application to google play store. 7. It offers Built-in support for Google Cloud Platform, making it easy to integrate Google Cloud Messaging and App Engine.

4.3 Software Development Kit (SDK)

While developing an Android application the programmer might need a set of tools and libraries in order to be able to use a certain updated resource of the system, or just to invoke a function easier using code that already exists. Therefore, set of tools is called SDK. Usually SDK is simply an application programming interface (API) contains set of les that are used in order to get access to already implemented code. For example, when developing an android app a set of methods may have to be called or implemented (interface) in order for the application to work on the specic platform. The SDK also includes extra set of tools required for the IDE to produce a more appropriate coding experience which is suitable for the corresponding platform. It usually, also includes documentation les that provides information about specic functions and sample code for the developer to decrease its learning curve.
In Conclusion, the SDK contains all prevalent and useful package with updated android version.

Server Requirements

4.4 Django

Django is a free and open source web application framework with high-level Python Web framework that encourages fast development and clean, pragmatic design. Built by experienced python developers, it takes care of much of the difficulty of Web development, it provides a more flexibility to focus on writing application without the need of reinventing the wheel. It gives ready made components to develop new site which handles (signing up, signing in, signing out) Advantages: 1. It provides developers to build a new website in no time due to its already components in it.2. It handles security seriously to avoid common mistakes by developers. It has an excellent feature is to scale rapidly when web application requires it.

4.5 PostgreSQL

PostgreSQL is a powerful web hosting, database that stores all our android application information. Therefore, it is an open source object-relational database system. It is capable of storing any kind of data and it is easy to store and retrieve information through simple SQL statements. It runs on all well known operating systems, such as Linux, UNIX, and Windows. It is used by different web programming languages such as python, PHP etc.

5 Design and Implementation

5.1 Server Application

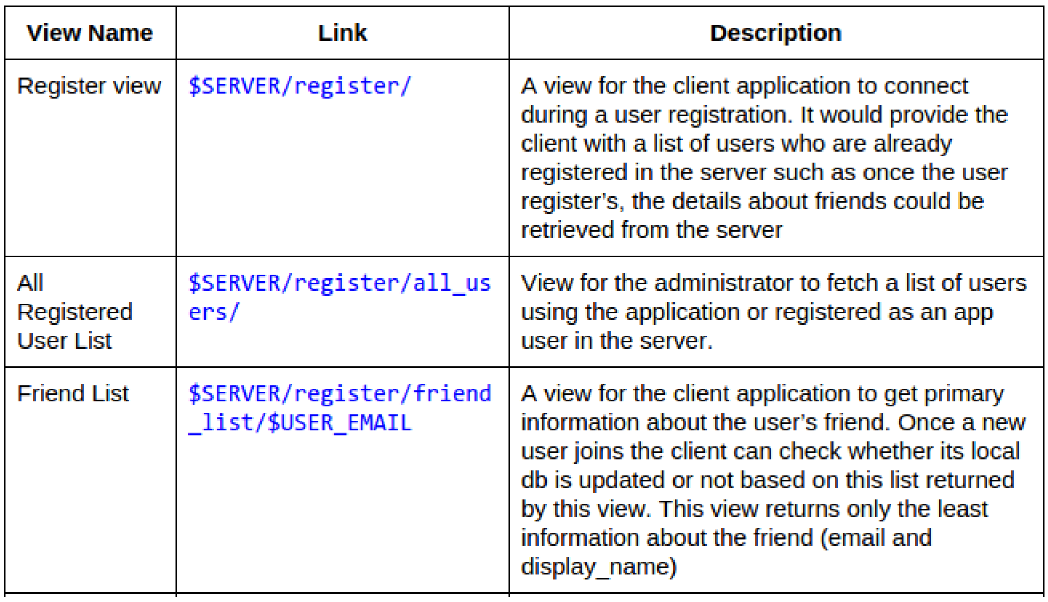
Python is a dynamic, OOP language that can be applied to various software development. The huge collection and support of standard libraries makes python used even for production designs. For the rapid web development, Django is most usually used as it a high-level python web framework. As its been extensively built up by experienced open source community members and as it follows model view controller, all it required for the project is to only develop the apps. The server side of the project consists of 6 custom apps. They are:
Registration: Handles user registration, friend list etc. 1. User Location: Collects location information from users for analysis 2. Defense: Stores the list of grievances and provides the Geofence details to the clients upon request. 3. Event: Handles event creation, attendance etc. 4. Data Analysis: Does the analysis of the data accumulated by the other apps and analyses it to give a graphical representation on a map. 5. User Phone Data: Collects the phone data from the user for analysis.

Representational State Transfer (REST) software architecture is also used as the project demands scalability. The server side is configured only with HTTP and enabled with communication over RESTful API with HTTP verbs (GET, PUT, POST, DELETE etc.). The client side (handled by type-safe REST client - Retrofit which turns the server side RESTful API into Java interface) communicates with the server through the RESTful API.

The server applications are equipped with a user interface designed with the help of Bootstrap templates for the ease of development and quick data upload. The data analysis application uses Open Street Maps for displaying the user activities on the screen.

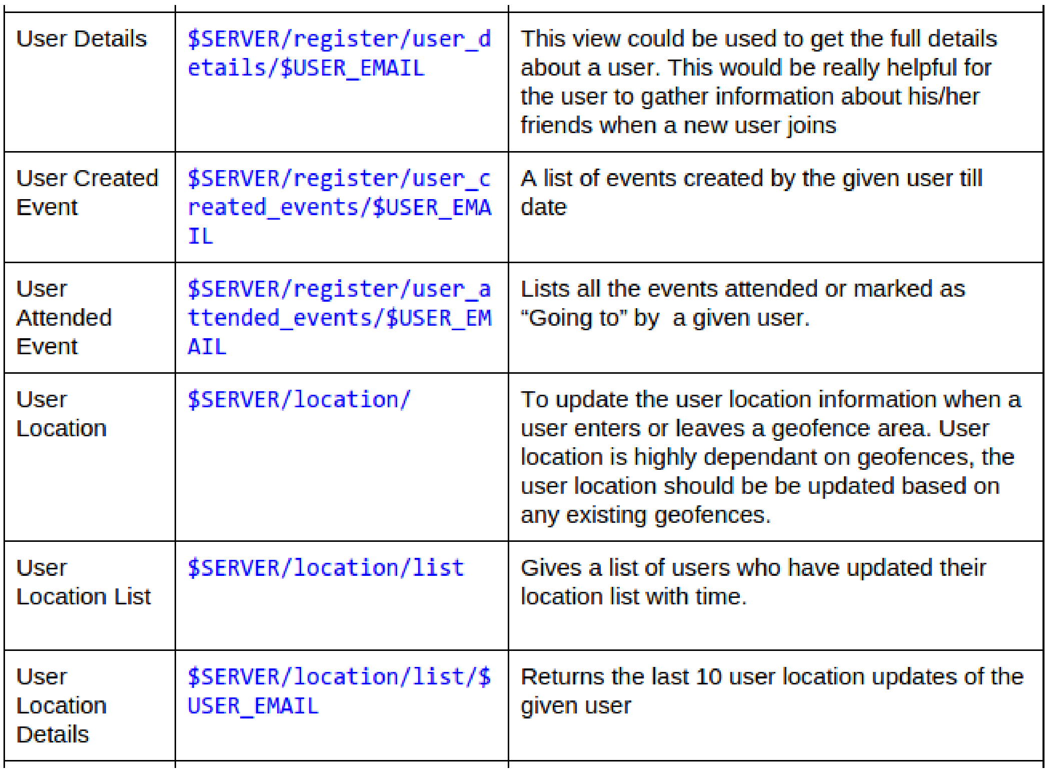
5.2 API Views

API Views are the only possible ways for the client to interact with the server app. The server can accept/supply data in html format also also in JSON format.



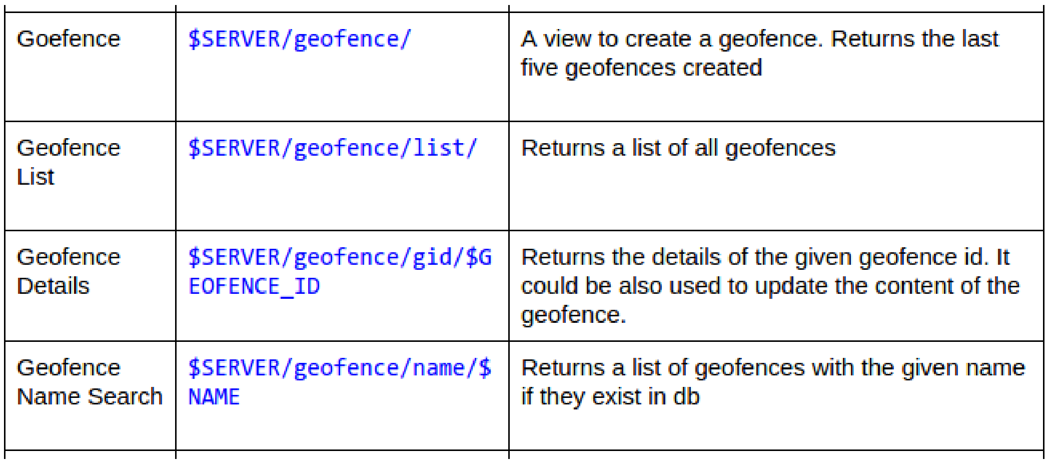
View Name	Link	Description
Register view	`$SERVER/register/`	A view for the client application to connect during a user registration. It would provide the client with a list of users who are already registered in the server such as once the user register's, the details about friends could be retrieved from the server
All Registered User List	`$SERVER/register/all_users/`	View for the administrator to fetch a list of users using the application or registered as an app user in the server.
Friend List	`$SERVER/register/friend_list/$USER_EMAIL`	A view for the client application to get primary information about the user's friend. Once a new user joins the client can check whether its local db is updated or not based on this list returned by this view. This view returns only the least information about the friend (email and display_name)

Figure 4: API View for User Register and Friend List[26]



User Details	`$SERVER/register/user_details/$USER_EMAIL`	This view could be used to get the full details about a user. This would be really helpful for the user to gather information about his/her friends when a new user joins
User Created Event	`$SERVER/register/user_created_events/$USER_EMAIL`	A list of events created by the given user till date
User Attended Event	`$SERVER/register/user_attended_events/$USER_EMAIL`	Lists all the events attended or marked as "Going to" by a given user.
User Location	`$SERVER/location/`	To update the user location information when a user enters or leaves a geofence area. User location is highly dependant on geofences, the user location should be be updated based on any existing geofences.
User Location List	`$SERVER/location/list`	Gives a list of users who have updated their location list with time.
User Location Details	`$SERVER/location/list/$USER_EMAIL`	Returns the last 10 user location updates of the given user

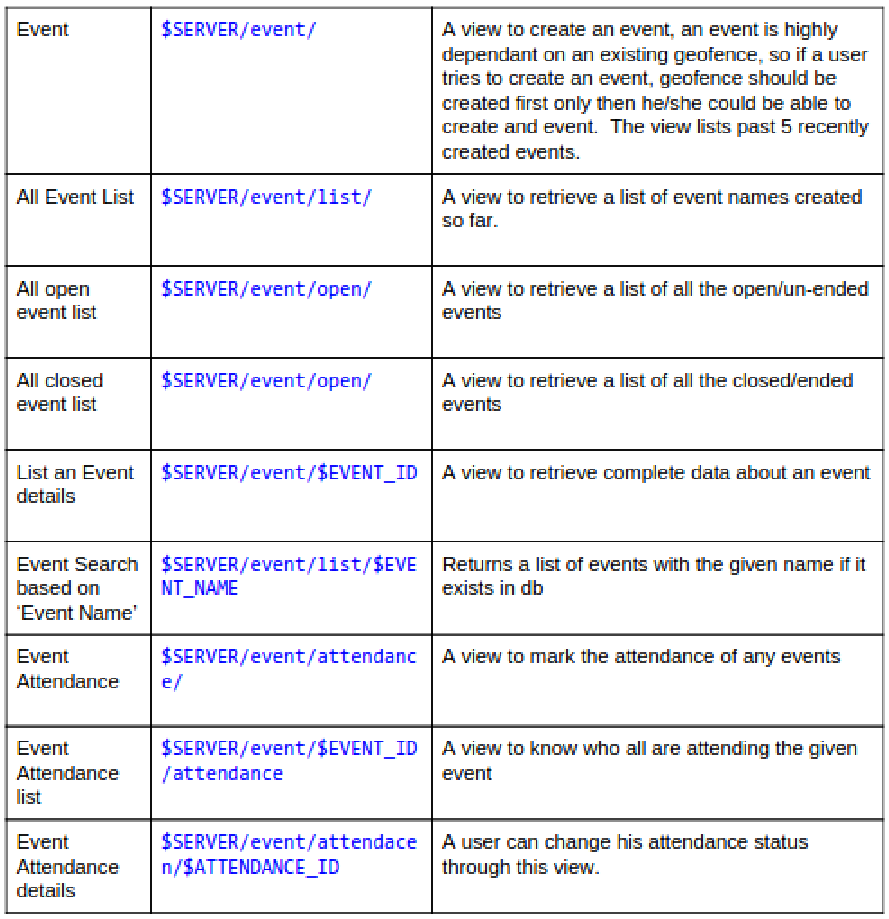
Figure 5: API View for User Details[27]



Goefence	`$SERVER/geofence/`	A view to create a geofence. Returns the last five geofences created
Geofence List	`$SERVER/geofence/list/`	Returns a list of all geofences
Geofence Details	`$SERVER/geofence/gid/$GEOFENCE_ID`	Returns the details of the given geofence id. It could be also used to update the content of the geofence.
Geofence Name Search	`$SERVER/geofence/name/$NAME`	Returns a list of geofences with the given name if they exist in db

Figure 6: API View for Geofence details[28]

Below tables gives the API view of the event activities.



Event	`$SERVER/event/`	A view to create an event, an event is highly dependant on an existing geofence, so if a user tries to create an event, geofence should be created first only then he/she could be able to create and event. The view lists past 5 recently created events.
All Event List	`$SERVER/event/list/`	A view to retrieve a list of event names created so far.
All open event list	`$SERVER/event/open/`	A view to retrieve a list of all the open/un-ended events
All closed event list	`$SERVER/event/open/`	A view to retrieve a list of all the closed/ended events
List an Event details	`$SERVER/event/$EVENT_ID`	A view to retrieve complete data about an event
Event Search based on 'Event Name'	`$SERVER/event/list/$EVENT_NAME`	Returns a list of events with the given name if it exists in db
Event Attendance	`$SERVER/event/attendance/`	A view to mark the attendance of any events
Event Attendance list	`$SERVER/event/$EVENT_ID/attendance`	A view to know who all are attending the given event
Event Attendance details	`$SERVER/event/attendacen/$ATTENDANCE_ID`	A user can change his attendance status through this view.

Figure 7: API View for Event creation[29]

5.3 Client Application

Android Development

Android uses Activities to show the UI of the applications. It is a single focused screen the user can interact with. The programmer has in his disposal some standard functions to manipulate the behaviour of this screen during the various stages of the application. These various stages of an application are known as the Activitys Life cycle. Activities are managed using an activity stack. When a new activity is started, it is placed on the top of the stack, which contains other running activities. An activity remains below another newly added Activity until the later one exits. At that time the older Activity returns to the foreground.

Activity Life cycle 1.Resumed 2.paused 3.stopped

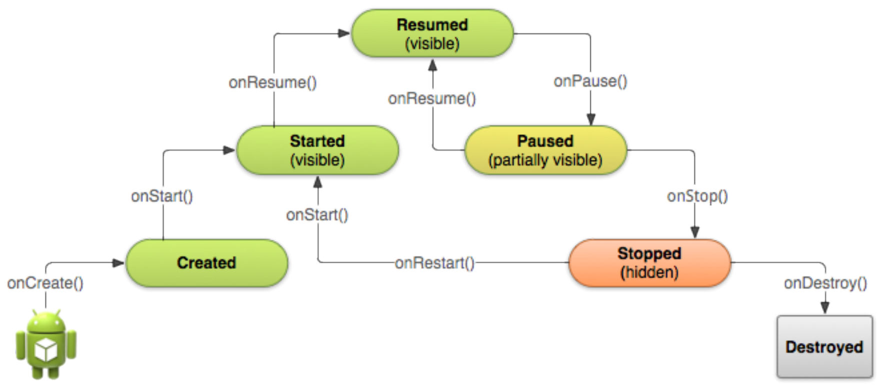


Figure 8: A simplified illustration of the Activity lifecycle[30]

Screen Sizes in Android

Special directories are provided for the necessary resources, which have to be properly scaled to the various density buckets. The appropriate scale for each density bucket is:
xhdpi: 2.0
hdpi: 1.5
mdpi: 1.0 (baseline)
ldpi: 0.75

Different Platform Versions

Android is constantly under development. This results in a broad range of changes on the various methods and functionality from one version to the other. While from one perspective this is a positive feature, as it means that bugs and security matters are resolved, the changes made from one Android version to the other, may require a specic code for a functionality to run properly, that depends on the version of Android currently 5.0 Lollipop. We are using Android version 5.0.2 Lollipop. The developers need to focus on the current version of the android

[30]http://developer.android.com/training/basics/activity-lifecycle/starting.html

because it might have many bugs which leads to unstable to debug. To Resolve all the issues must wait until the next update is available from google developer website.

5.4 Application Framework fundamentals

Most important components in android. They are

Java

Java code is all going to be in the src/main/Java directory. Here project main activity class will be created.

Resources

Initially, the app/res (Resources) folder contains:
1.”Drawable” folders that hold images just the default launch icon for now. And with the android studio, we can create images using its library containing images in its default folder.
2. The ”layout” folder with XML that represents the screen designs. Here in this folder we can create respective UI interface design for every activity where the user will be interacting with the screen.
SampleCode : Creating Linear Layout

```
res/layout/SignIn.xml
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:orientation="horizontal" >
Add a text Field
<EditText android:id="@+id/Signin"
    android:layout_width="wrap_content"
    android:layout_height="wrap_content"
    android:hint="@string/enter_Email" />
</LinearLayout>
```

3. The ”menu” folder with XML of the items that will appear on the Action Bar. Here we can determine which option users must can able to view on the Action Bar when our application runs and performs respective activities to launch.
4. The ”values” folder with XML containing dimensions, strings, and styles. Here folder contains ”strings.xml” it helps to define or label the buttons and screen messages, where the user will read it and perform the respective tasks.

```
res/values/strings.xml
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Community_Organizer</string>
    <string name="enter_Email">Enter a EmailID</string>
    <string name="enter_pass">Enter a Password</string>
    <string name ="action_send">Signin</string>
</resources>
```

Android Manifest.XML

This XML file informs your system of the apps hardware and software requirements and contains your apps name, icon, and version. The manifest also filters the Intents.

Build.Gradle

Android Studio uses Gradle to build and compile the android application. In ”build.gradle” folder builds dependencies are set along with it defaultconfig setting in it.
CompileSdkVersion: This represents the platform version of our android application will be compiled. By default it will set to the updated version of android. Here we can also use an older version of android, but new features cannot be implemented.
Application ID: This represents our project name with ID which we have assigned a name while creating a New Project Workflow.
minSdkversion: This represents the minimum SDK version which we have specified at the start of a New Project Workflow.

TargetSdkversion: This represents the target SDK version in which android application will run on android mobile phones.

Intent

An Intent is a message object which helps to interact with other components in our android application. Therefore, it decides to launch one application activity after one activity is completed. It will manage to run other activity when the user selects on particular options from the Action Bar.

Start an activity:
An Activity performs a single screen in an application. Therefore, To start a new instance of an activity by passing and Intent to startActivity(). Intent will send a message to an activity to launch or start and handle out any significant data which is required.

Start a service:
An Activity which takes place without any user interaction and it will run in the background, Therefore, we utilize this component to perform a one-time operation by passing information an Intent to startService(). This illustrates the service to start and handle out any significant data which is required.

Deliver a broadcast:
A broadcast is a message that any application can receive which are running on Android OS and shared among all applications. The system sends various broadcasts information to all android applications such as device is in charge state or when the device boots up. You can deliver a broadcast to another application by passing an Intent to sendBroadcast(), sendOrderedBroadcast(), or sendStickyBroadcast().

Use Case

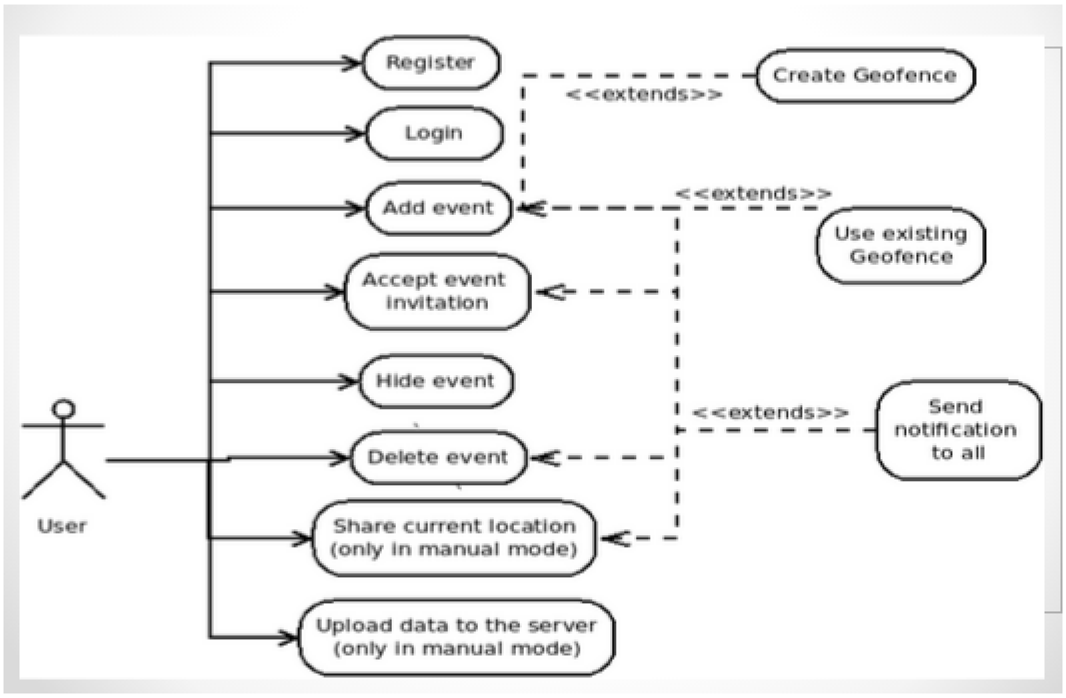


Figure 9: Use case diagram of user activity[31]

Use case "figure 9" explains about the user actions. User first register with our android application, then user registration details will send to our server through retrofit REST client in JSON format later user will login with "Email Id" and "password" and user can create can an event, activity based on the user location. Once the event is created it will be shared among other members in the community and other members need to accept the invitation. The accepted information will be shared among remaining members and that they can participate in the event activity. All this information will be stored in android database and periodically all data will be sent to the server.

In this android application we are using Geo-fencing technology to get user location information, and communicate with the server we are using retrofit REST client API.

MVC(MODEL VIEW CONTROLLER) Architecture

In our android development, we have followed an MVC architecture.

Models: Content Providers

The models consist of business logic and data in its logical form. Therefore, data will be shared among inter-application. Although it often uses the observer pattern to allow its listeners, such as the view, to get updates from it, and often, the model class is a plain old Java object.

Views: Activities

The view is responsible for representing data and state that defined in the model, to the Android user. Therefore, by separating the view from the model allows android application to display the same data in different ways. They are containers for Views (android.view.View).

Controllers: Services

The Controller takes user inputs provided by the view and converts it to model-changing operations. Therefore, These are components which run in the background without any user interaction. They run invisibly and perform ongoing unattended processing.

The below diagram explains about the MVC architecture in detail way.

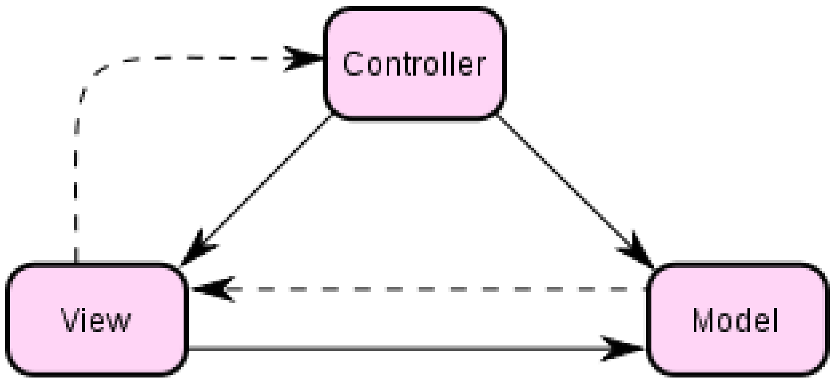


Figure 10: MVC Architecture[32]

5.5 Design

Android Application Screen design:

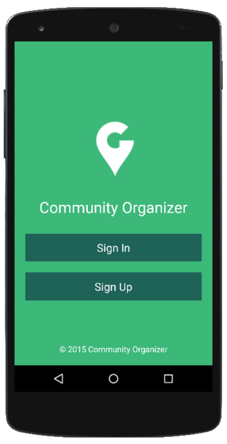


Figure 11: Welcome Screen[33]

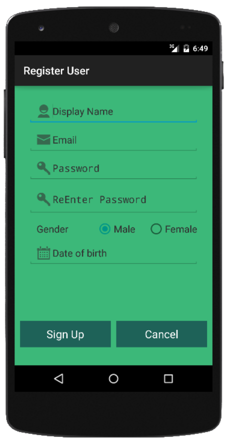


Figure 12: SignUp Screen[34]

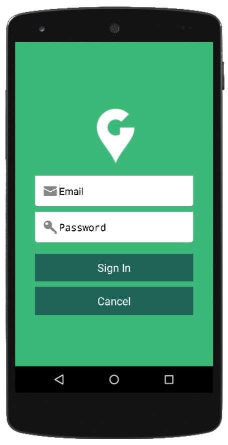


Figure 13: Signin Screen[35]

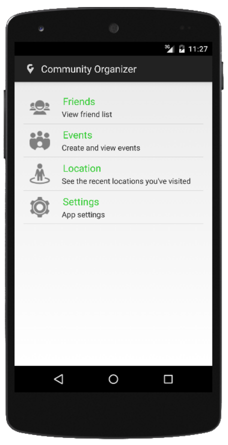
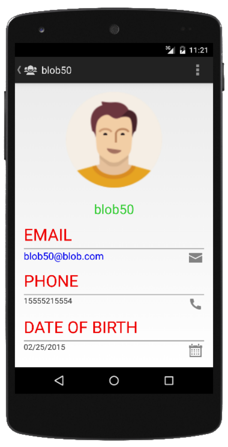


Figure 14: Home Screen[36]



11:21

blob50

blob50

EMAIL

blob50@blob.com

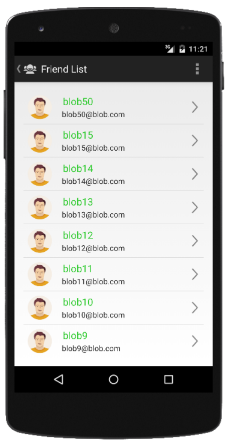
PHONE

15555215554

DATE OF BIRTH

02/25/2015

Figure 15: Friend Details screen[37]



11:21

Friend List

blob50
blob50@blob.com

blob15
blob15@blob.com

blob14
blob14@blob.com

blob13
blob13@blob.com

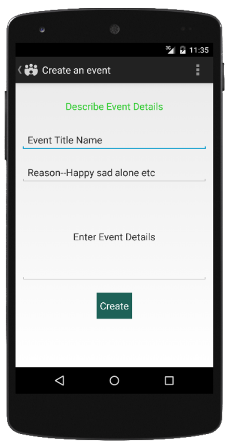
blob12
blob12@blob.com

blob11
blob11@blob.com

blob10
blob10@blob.com

blob9
blob9@blob.com

Figure 16: FriendList Screen[38]



11:35

Create an event

Describe Event Details

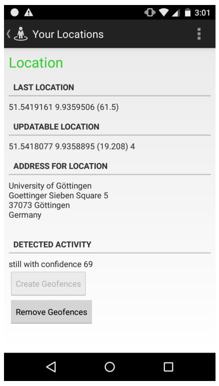
Event Title Name

Reason--Happy sad alone etc

Enter Event Details

Create

Figure 17: Create Event[39]


3:01

Your Locations

Location

LAST LOCATION

51.5419161 9.9359506 (61.5)

UPDATABLE LOCATION

51.5418077 9.9358895 (19.208) 4

ADDRESS FOR LOCATION

University of Göttingen
Goettinger Sieben Square 5
37073 Göttingen
Germany

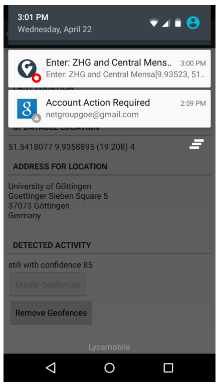
DETECTED ACTIVITY

still with confidence 69

Create Geofences

Remove Geofences

Figure 18: Create and Remove Geo-fence[41]


3:01 PM
Wednesday, April 22

Enter: ZHG and Central Mens.. 3:00 PM
Enter: ZHG and Central Mensa[9.93523, 51..

Account Action Required 2:59 PM
netgroupgoe@gmail.com

51.5418077 9.9358895 (19.208) 4

ADDRESS FOR LOCATION

University of Göttingen
Goettinger Sieben Square 5
37073 Göttingen
Germany

DETECTED ACTIVITY

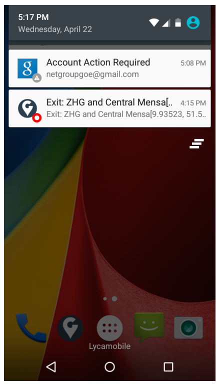
still with confidence 85

Create Geofences

Remove Geofences

Lycamobile

Figure 19: Geo-fence Entry(Central Campus)[43]


5:17 PM
Wednesday, April 22

Account Action Required 5:08 PM
netgroupgoe@gmail.com

Exit: ZHG and Central Mensa[.. 4:15 PM
Exit: ZHG and Central Mensa[9.93523, 51.5..

Lycamobile

Figure 20: Geo-fence Exit(Central Campus)[45]

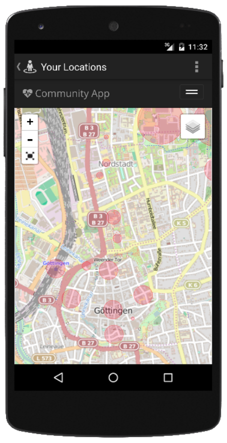


Figure 21: Geofence Map Screen[46]

6 Conclusion

We built up a dedicated decentralized server for the community organizer, With our approach users' data is secure and safe. Our android application offers the same level of user interaction as compared to other social networking applications, It also allows users to have full control over their own personal data. As our android application web server will be administrated by community organizer member only. In conclusion, we implemented all features in our android application, including Geo-fencing details where to track the user entry and exit locations of the Geo-fence without draining the battery life of smart phone. Therefore, User location information is stored locally in the Android device.

7 Future Work

Our future aim is to transmit the necessary data of our android application to our web server for data analysis such as location information of the user, battery state, memory usage, last online time & duration and through which connectivity members of the community are frequently using our application. Along with this implementation, we also want to develop a tracking system to track the members' waypoints by granting specific permission to the community members, once they have created an event and shared among others while they are inside Geo-fencing radius. In addition, we will offer community members to create their own Geo-fences.

References

[1] emarketer, *2 Billion Consumers Worldwide to Get Smart(phones) by 2016* url = http://www.emarketer.com/Article/2-Billion-Consumers-Worldwide-Smartphones-by-2016/1011694

[2] Nielsen, *HOW SMARTPHONES ARE CHANGING CONSUMERS DAILY ROUTINES AROUND THE GLOBE* url = http://www.nielsen.com/us/en/insights/news/2014/how-smartphones-are-changing-consumers-daily-routines-around-the-globe.html

[3] TechEmpower, *Web Framework Benchmarks* url = https://www.techempower.com/benchmarks/# section=data-r9& hw=peak& test=json

[4] MOBISOFT, *3 ways to implement efficient location tracking in Android applications* url = http://mobisoftinfotech.com/resources/blog/android/3-ways-to-implement-efficient-location-tracking-in-android-applications/

[5] Android Developer website, url = http://developer.android.com/training/location/geofencing.ht

[6] Reactivex, *An API for asynchronous programming with observable streams* url = http://reactivex.io/

[7] Django REST framework, url = http://www.django-rest-framework.org/

[8] Retrofit, *A type-safe REST client for Android and Java* url = http://square.github.io/retrofit/

[9] Activeandroid, url = http://www.activeandroid.com/