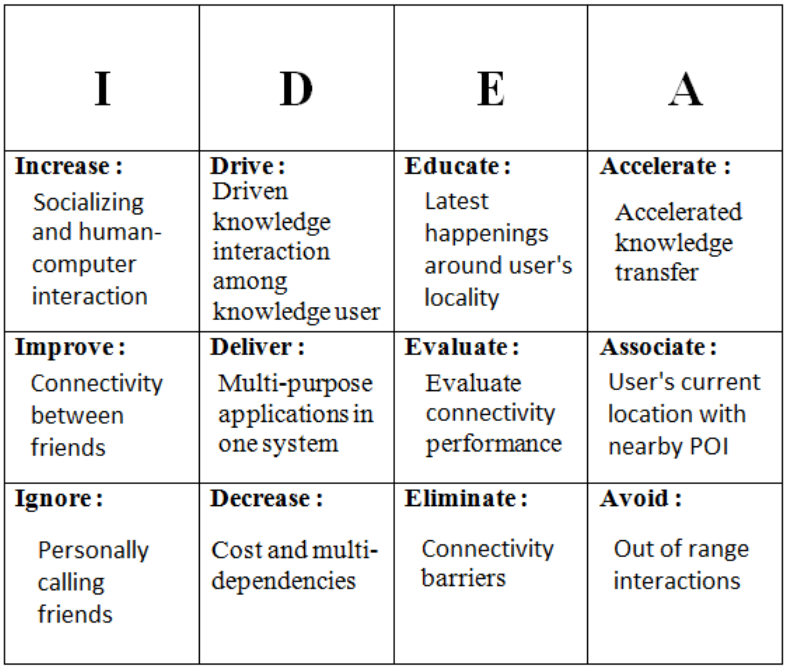
Assignment 1

Knowledge canvas and IDEA Matrix

Knowledge canvas is one that depicts the knowledge forces and knowledge flow across organization and extended organizations. It captures the current knowledge state and knowledge forces in the environment. It tries to build bigger and broader knowledge scenario for user and system environment.

Principle component for knowledge canvas for hand gesture recognition system are:

1. Knowledge force for cost saving
2. Knowledge about precision
3. Knowledge about nearby interaction
4. External knowledge forces
5. Location based event awareness



I	D	E	A
Increase : Socializing and human-computer interaction	**Drive :** Driven knowledge interaction among knowledge user	**Educate :** Latest happenings around user's locality	**Accelerate :** Accelerated knowledge transfer
Improve : Connectivity between friends	**Deliver :** Multi-purpose applications in one system	**Evaluate :** Evaluate connectivity performance	**Associate :** User's current location with nearby POI
Ignore : Personally calling friends	**Decrease :** Cost and multi-dependencies	**Eliminate :** Connectivity barriers	**Avoid :** Out of range interactions

Figure 1: IDEA Matrix Framework

Software and Hardware Specification:

1. **Hardware Specification:**

   (a) Android Smartphone
   (b) Memory of 1 GB RAM (or more) in android phone

2. **Software Interface:**
   (a) Front end : Android
   (b) Back end : Parse Cloud
   (c) Android studio
   (d) Android sdk tools
   (e) Java SE JDK 1.7

3. **Technologies Used:**

   (a) **Android**

   Android is an operating system for mobile devices. It is mostly used for cell phones, like Google's own Galaxy Nexus, as well as by other phone manufacturers like HTC and Samsung. It has also been used for tablets such as the Motorola Xoom and Amazon Kindle Fire. Android's kernel is based on Linux. Programs for Android, also called "apps", come from the Google Play store. The android programs have an extention of .apk. Android programs are built in C, C++, or Java programming languages but the UI is always made using Java. There are over 900,000 apps available for Android. Each version of Android has both a number and a name based on a dessert. The version numbers and names are:

   i. Beta versions: Astro and Bender
   ii. 1.5: Cupcake
   iii. 1.6: Donut
   iv. 2.0 and 2.1: Eclair
   v. 2.2: Froyo (FROzen YOgurt)
   vi. 2.3: Gingerbread
   vii. 3.x: Honeycomb (a tablet-only version)
   viii. 4.0: Ice Cream Sandwich
   ix. 4.1, 4.2 and 4.3: Jelly Bean

x. 4.4: KitKat

xi. 5.0: Lollipop

xii. 6.0: Marshmallow

Features in the Android operating system :

1. Messaging

2.Web browser

3. Voice-based features

4. Multitasking

5. Connectivity

6. Media support

7. Location based service

(b) **Parse Cloud**

Mobile Backend as a Service(MBaaS), also known as “backend as a service” (BaaS), is a model for providing web and mobile app developers with a way to link their applications to backend cloud storage. The Parse Android SDK allows you to store data, manage users, send push notifications, track analytics, and more in just a few lines of code. Keep your users engaged and coming back for more with Parse Push. Create highly effective push campaigns with just a few clicks and measure their success with Parse Analytics. With a single line of code, track any data point in your app using Parse Analytics. View app usage, push campaigns, and custom analytics in the user-friendly Parse dashboard. Overlay graphs, filter by date, and more to gain insight into the effectiveness of your app.

(c) **Location based service**

A location-based service (LBS) is a software application for a IP-capable mobile device that requires knowledge about where the mobile device is located. Location-based services can be query-based and provide the end user with useful information such as ”Where is the nearest ATM?” or they can be push-based and deliver coupons or other marketing information to customers who are in a specific geographical area.

Assignment 2

Project problem statement

1. **Feasibility assessment using NP-Hard, NP-Complete or satisfiability issues using modern algebra**

   Problem Definition

   Recommending friends instantly based on current location of users in the real world has become increasingly popular in Location-based mobile social network (LMSN). However, the existing friend recommendation methods based on topological structures of a social network or non-topological information such as similar user profiles cannot well address the instant making friends in the real world.

   (a) **Problem :** Calculate distance between any two friends.

   (b) **Input :** Latitude and Longitude of the all friends in favourites list.

   (c) **Output :** Distance is calculated between user and all friends, then if the distance is less than range provided by user then it shows nearest friends on map.

   So that problem definition is **NP-Complete**.

2. **Mathematical model**

   S is a system for implementing a location based android application.

   S = s, e, x, y, DD, NDD, success, failure, CPU core count

   s = Start state

   e = End state

x = input : Current location

y = output : List of nearby friends

DD = Deterministic Data : Current location.

NDD = Non-Deterministic Data : User interest based suggestions, range, nearby friends.

Success case : List of all nearby friends within the range.

Failure case : GPS is disabled

CPU core count = No. of cores required to execute the application : 1

A representation in mathematical terms of the behaviour of real devices and objects. A mathematical model is a description of a system using mathematical concepts and language. The process of developing a mathematical model is termed mathematical modelling. Method of simulating real life situations with mathematical equations to forecast their future behaviour. Mathematical modelling uses tools such as decision theory, queuing theory, and linear programming, and requires large amounts of number crunching.

3. **Need Of Mathematical Model**

   Since the modelling of devices and phenomena is essential to both engineering and science, engineers and scientists have very practical reasons for doing mathematical modelling. In addition, engineers, scientists, and mathematicians want to experience the sheer joy of formulating and solving mathematical problems.

   (a) Enables a thorough understanding of the system modelled.

   (b) Prepares the way for better design or control of the system.

   (c) Allows the efficient use of modern computing capabilities.

4. **Objective**

   (a) Locate your friends on Map

   (b) Chat with friends

(c) Create events and suggest users on the basis of their interest

(d) Locate interested members for an event on map to properly schedule event

5. **Outcomes**

   Success: Locate the near-by users within a given range.

   Failure: GPS is not enabled, so unable to locate users current location.

Assignment 3

Distributed/parallel/concurrent processing

1. **Domian Information :**

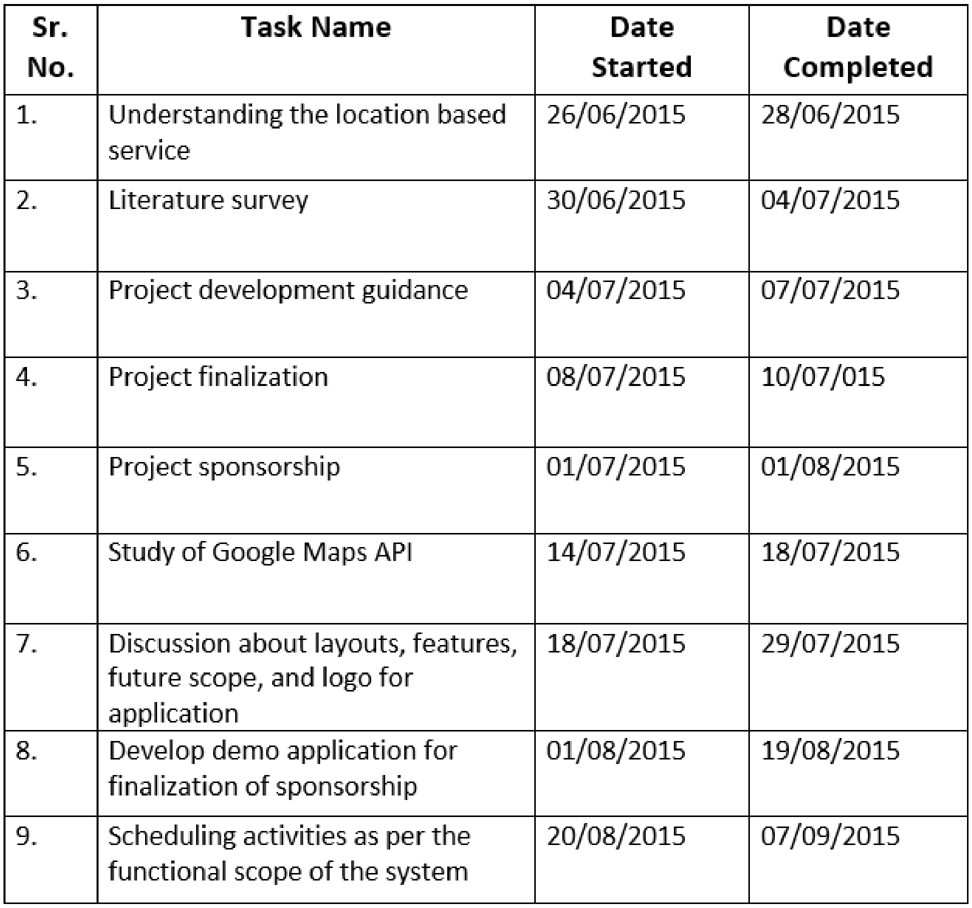
   (a) **Location Based Service (LBS):** Location-based services (LBS) are a general class of computer program-level services that use location data to control features. As such LBS is an information service and has a number of uses in social networking today as an entertainment service, which is accessible with mobile devices through the mobile network and which uses information on the geographical position of the mobile device.

   A location-based service (LBS) is a software application for a IP-capable mobile device that requires knowledge about where the mobile device is located. Location-based services can be query-based and provide the end user with useful information such as "Where is the nearest ATM?" or they can be push-based and deliver coupons or other marketing information to customers who are in a specific geographical area. An LBS requires five basic components: the service provider's software application, a mobile network to transmit data and requests for service, a content provider to supply the end user with geo-specific information, a positioning component (see GPS) and the end user's mobile device. By law, location-based services must be permission-based. That means that the end user must opt-in to the service in order to use it. In most cases, this means installing the LBS application and accepting a request to allow the service to know the device's location.

   Although location-based services have been around since 2000, they have mostly been used in commerce with a subscription-based business model. The release of Apple's 3G iPhone and Google's LBS-enabled Android operating system, however, has allowed developers to introduce millions of consumers to LBS. According to the 2008 fourth-quarter report from Nielsen Mobile, a division of The Nielsen Company, location-based services account for 58 percent of the total downloaded application revenue for mobile phones in North America.

2. **Implementation Plan**

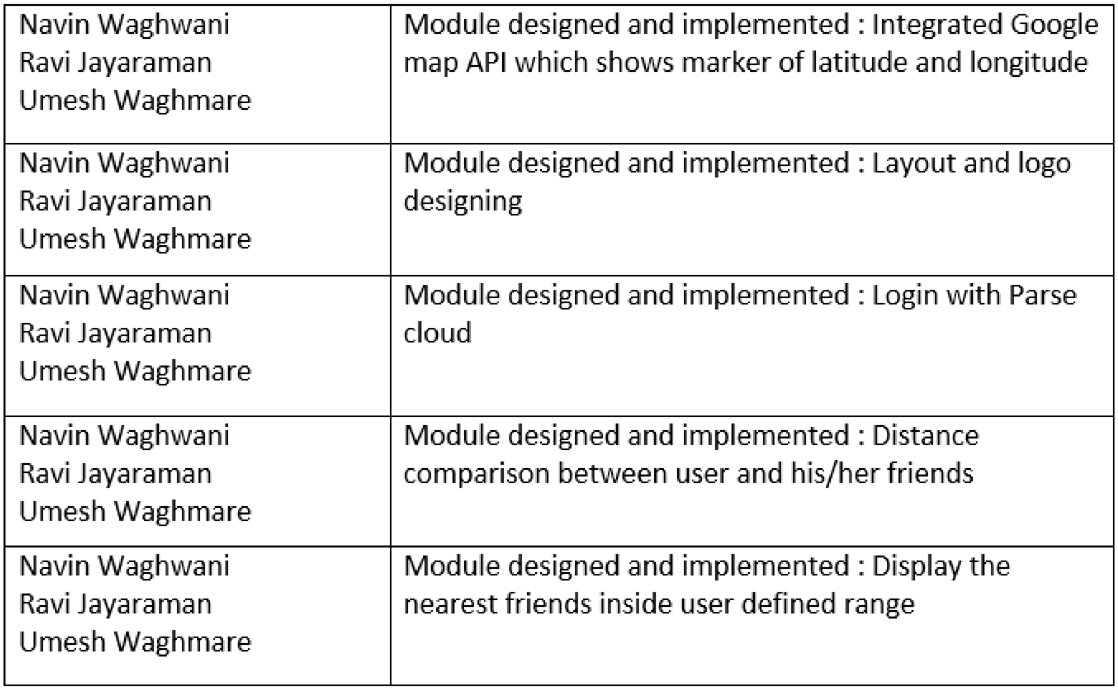
   Phase I



Sr. No.	Task Name	Date Started	Date Completed
1.	Understanding the location based service	26/06/2015	28/06/2015
2.	Literature survey	30/06/2015	04/07/2015
3.	Project development guidance	04/07/2015	07/07/2015
4.	Project finalization	08/07/2015	10/07/015
5.	Project sponsorship	01/07/2015	01/08/2015
6.	Study of Google Maps API	14/07/2015	18/07/2015
7.	Discussion about layouts, features, future scope, and logo for application	18/07/2015	29/07/2015
8.	Develop demo application for finalization of sponsorship	01/08/2015	19/08/2015
9.	Scheduling activities as per the functional scope of the system	20/08/2015	07/09/2015

Figure 2: System implementation plan phase 1

3. **Team Structure**



Navin Waghwani Ravi Jayaraman Umesh Waghmare	Module designed and implemented : Integrated Google map API which shows marker of latitude and longitude
Navin Waghwani Ravi Jayaraman Umesh Waghmare	Module designed and implemented : Layout and logo designing
Navin Waghwani Ravi Jayaraman Umesh Waghmare	Module designed and implemented : Login with Parse cloud
Navin Waghwani Ravi Jayaraman Umesh Waghmare	Module designed and implemented : Distance comparison between user and his/her friends
Navin Waghwani Ravi Jayaraman Umesh Waghmare	Module designed and implemented : Display the nearest friends inside user defined range

Figure 3: Team structure

Assignment 4

Software modelling methods

(a) DFD

(b) To draw functional dependency graphs and relevant, techniques including UML diagrams or other necessities using appropriate tools.

i. **Data Flow Diagrams**

Level 0 DFD

It is common practice to draw the context level DFD first, which shows the interaction between the system and user. On the context diagram the systems interactions with the outside world are modeled purely in terms of data flows across the system boundary. The diagram shows the entire system as a single process. The Level 0 DFD shows how the system is divided into sub-systems (processes), each of which deals with one or more of the data flow to or from an user and data owner, and which together provide all of the functionality of the system as a whole.

The DFD level 0 shows a simple implementation for the system in which system takes input and pre process the image and simply convert into clear output.

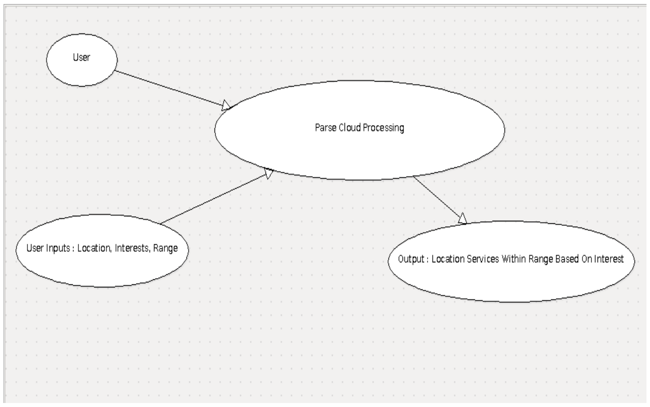

User
Parse Cloud Processing
User Inputs : Location, Interests, Range
Output : Location Services Within Range Based On Interest

Figure 4: level 0 DFD

Level 1 DFD

The Level 1 DFD shows various steps for generating a clear document image from the degraded image.The image is first converted into gray scale.then the image segmentation algorithm is applied and the post processing is done to convert the image into clear document image.

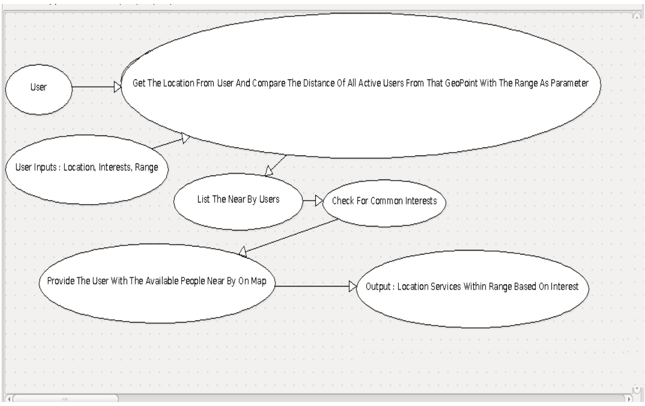

User
Get The Location From User And Compare The Distance Of All Active Users From That GeoPoint With The Range As Parameter
User Inputs : Location, Interests, Range
List The Near By Users
Check For Common Interests
Provide The User With The Available People Near By On Map
Output : Location Services Within Range Based On Interest

Figure 5: level 1 DFD

ii. Class Diagram

Class Diagram is a type of static structure diagram that describes the structure of a system by showing the systems classes, their attributes, and the relationships between the classes. Class diagrams are the backbone of almost every object oriented methods, including UML. They describe the static structure of a system. An object Class describes a group of objects with similar properties (attributes), common behaviour (operations),common relationships to the other objects, and common semantics.

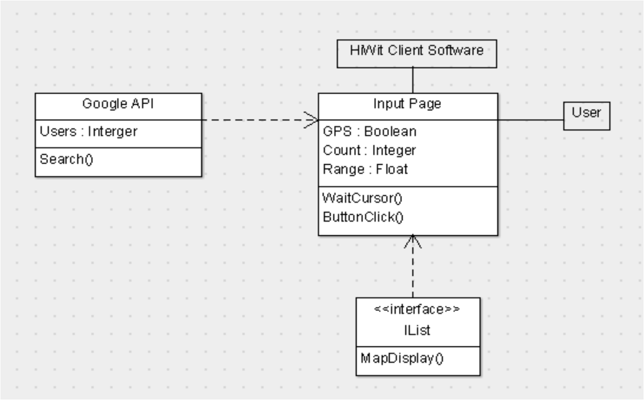


Figure 6: Class Diagram

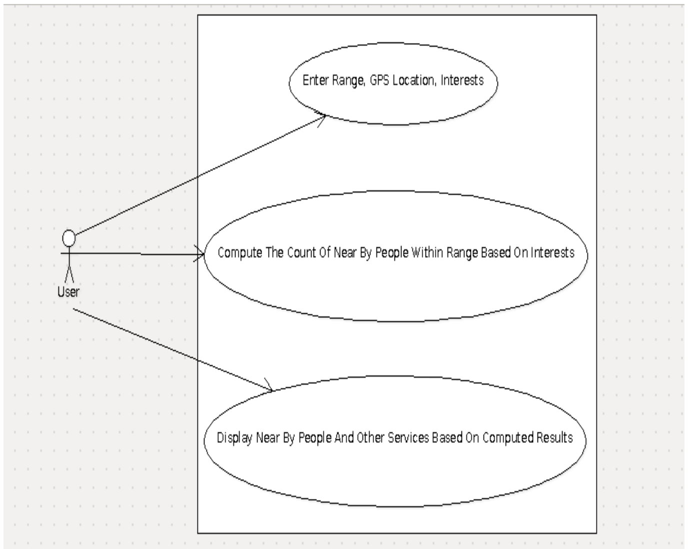

Enter Range, GPS Location, Interests
User
Compute The Count Of Near By People Within Range Based On Interests
Display Near By People And Other Services Based On Computed Results

Figure 7: UML Diagram

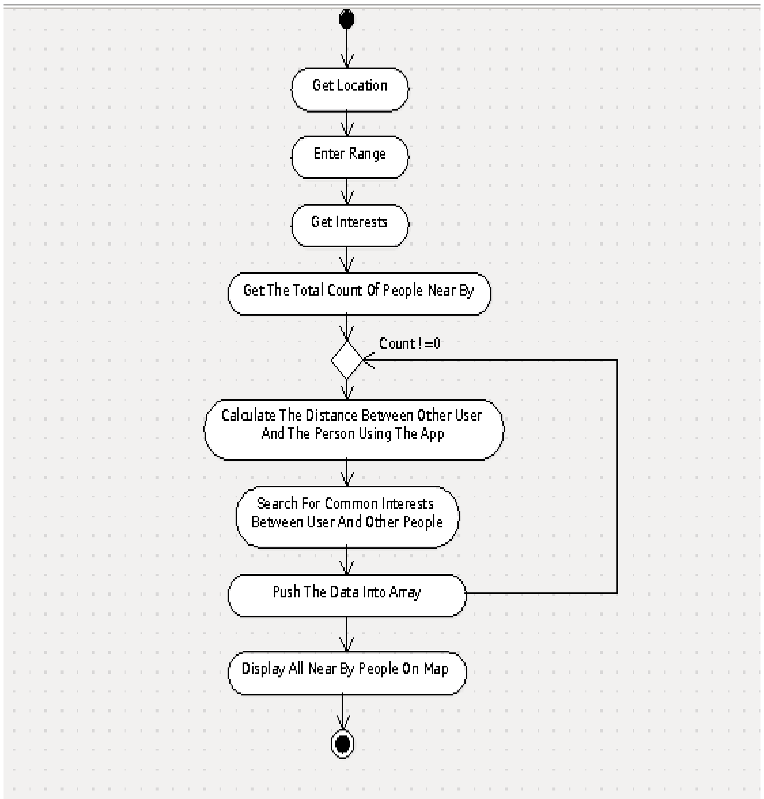

Get Location
Enter Range
Get Interests
Get The Total Count Of People Near By
Count !=0
Calculate The Distance Between Other User And The Person Using The App
Search For Common Interests Between User And Other People
Push The Data Into Array
Display All Near By People On Map

Figure 8: Activity Diagram

Assignment 5

Testing of project problem statement

(a) Test data (using mathematical models, GUI, Function testing principles, if any)

(b) Selection and appropriate use of testing tools, testing of UML diagram's reliability.

Introduction: Software testing is the process of executing a program or system with the intent of finding errors. Or it involves any activity aimed at evaluating an attribute or capability of the system and determining that it meets its required results. Software is unlike other physical processes where inputs are received and outputs are produced. Where software differs is in that manner in which it fails. Most physical systems fail in a fixed set of ways. By contrast, software can fail in bizarre ways. Detecting all of the different failure modes for software is generally infeasible. Testing is performed for following purposes: 1. To improve quality 2. For verification and validation Basic software testing.

Phases of software development:
-Analysis
-Design
-Coding
-Testing

(a) **Test Case**

i. **Black Box Testing** Black Box Testing is testing without knowledge if internal working of the item being tested. For Example when black box testing is applied to software engineering, the tester would only know the "legal" input and what the excepted output should be, but not how the program actually arrive at those output. It is because of this that

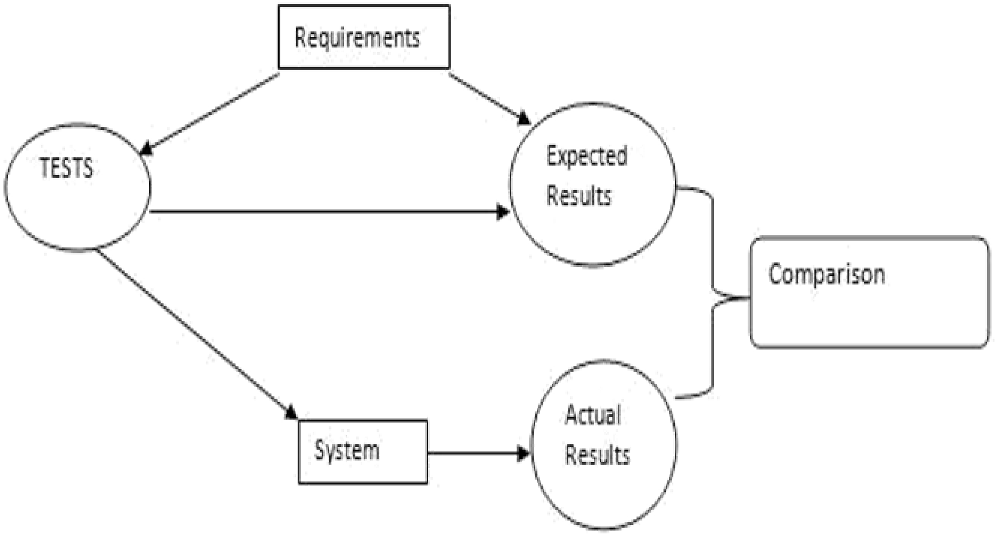


Figure 9: Framework Of Test Cases

black box testing can be considered testing with respect to the specifications, no other knowledge of the program is necessary. For this reason, the tester and the programmer can be independent of the one another, avoiding programmer bias toward his own work. Foe this testing, test group are often used. For glass box testing, the test cases cannot be determined until the code has actually been written. Both of these testing techniques have advantages and disadvantages, but when combine, they help to ensure through testing of the product.
Testing methods where user is not required:
-Functional: In this type of testing, the software is tested for the functional requirements. The tests are written in outer to check if the application behaves as expected.
-Stress: The application is tested against heavy load such as complex numerical values, lager number inputs, large number of queries etc. which check for the stress/load the application can withstand.

ii. **White Box Testing** White box testing strategy deals with the internal logic and structure of the code. White box testing is also called as glass, structure, open box or clean box testing. The tests written based on the white box testing strategy incorporate coverage of the code written. Order to implement white box testing, the tester has to deal with the code and hence is needed knowledge of coding and logic i.e. internal working of the code.

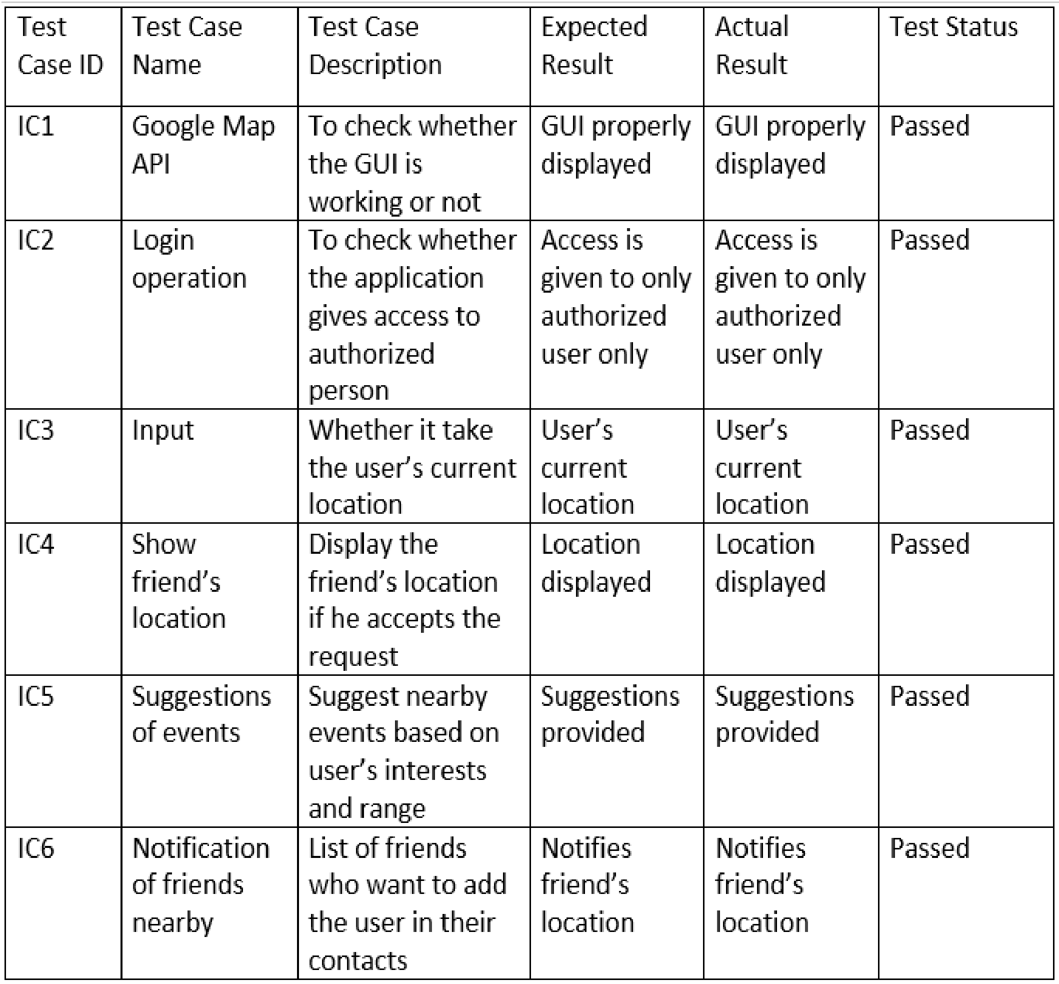
Types of testing white/glass box testing strategy:

-Unit Testing: The developer carries out unit testing in order to check if the particular module or unit of code is working fine. The unit testing comes at the very basic level as it is carried out as and when the code is developed or a particular functionality is built.

-Static and Dynamic Analysis: Static analysis involves going through the code in order to find out any possible defect in the code. Dynamic analysis involves executing the code and analysing the output.

-Security Testing: Security Testing is carried out to find out how well the system can protect itself from unauthorized access, hacking cracking, any code damage etc. which deals with the code of application. This type of testing needs sophisticated testing techniques.

iii. Test Plans



Test Case ID	Test Case Name	Test Case Description	Expected Result	Actual Result	Test Status
IC1	Google Map API	To check whether the GUI is working or not	GUI properly displayed	GUI properly displayed	Passed
IC2	Login operation	To check whether the application gives access to authorized person	Access is given to only authorized user only	Access is given to only authorized user only	Passed
IC3	Input	Whether it take the user's current location	User's current location	User's current location	Passed
IC4	Show friend's location	Display the friend's location if he accepts the request	Location displayed	Location displayed	Passed
IC5	Suggestions of events	Suggest nearby events based on user's interests and range	Suggestions provided	Suggestions provided	Passed
IC6	Notification of friends nearby	List of friends who want to add the user in their contacts	Notifies friend's location	Notifies friend's location	Passed