# CHAPTER 1

# INTRODUCTION

## 1.1 AIM

To develop a mobile application which would help the users to host events or matches, this can be viewed and joined by other users with similar interests.

## 1.2 PROBLEM STATEMENT

Physical Exercise is of utmost importance in today's urban environment, but people face several constraints like lack of knowledge about grounds in the locality and also it is tough to maintain physical fitness when there is no company or friends with similar interests.

## 1.3 SCOPE OF THE PROJECT

Researchers have found more than 2 billion people worldwide are now overweight or obese. If that is shocking, over 70% of India's urban population is found to be obese. These are people who work five days a week and hardly have time even for their near and dear ones. India has the world's largest share of diabetic patients and it is mostly attributed to lack of physical exercise. Especially, the women of our country have the lowest physical training among developed and developing nations. But in a country like ours you will be shocked to see the number of college students, young corporate people and even a high share of people who are over 30 years of age playing games like cricket, badminton and football during weekends in the various city playgrounds and parks. They wait for an entire week so that they can catch up with their old friends and have a competitive day of sports on weekends. But after a while it gets boring to play with the same 6-7 people and gradually they stop playing any game and when this happens they have virtually no exercise.

## 1.4 DESCRIPTION

Our application has a simple aim of providing people with a chance to meet like-minded people who share their interests and provide a competitive environment which would help them de-stress and also provide them the much needed physical activity which would keep them healthy in the longer run. We aim to achieve this by providing a very simple app which would help the users to create an individual or team account and create an event or match with specifics such as sport, time, and venue. Anyone who has a team which matches the requirements can fix a match with the opponent using the application.

## 1.5 BENEFITS

- During summer holidays, students are free throughout the day. The app aims to capture this audience by providing them with a platform to organize matches and spend their time in a useful and productive manner.
- The benefits of physical exercise, in general is well documented and is one of the most important needs for today’s generation.
- The application helps to save valuable time by providing opportunity to interact with people who have similar interests and passions

# CHAPTER 2
# LITERATURE SURVEY

## 2.1 EXISTING SYSTEM

- There is no such existing application which helps people with similar sporting interests to meet and play with each other. The most common thing which happens is people meet random people in the ground, get their contact and setup matches with them.

- Another possible idea is to create events in social media like facebook, Google+ and other social networking sites and hoping someone would see our event. But the possibilities of someone stumbling over it are remote.

## DISADVANTAGES:

- The major disadvantage of contacting random people and setting up matches with them is that there is no guarantee they will turn up for the match, and for normal working people wasting an entire weekend waiting for someone to turn can be taxing and they could end up being disinterested.

- The problem with creating events in social media is that it is usually hidden behind volumes of other unnecessary data and safety settings. It ends up 1being difficult for even people who are searching for a similar event to find them. This also faces the problem of accountability. There is no guarantee the opponent would turn up.

## 2.2 PROPOSED SYSTEM

Our application tries to overcome these difficulties by providing a dedicated platform for people to create and join matches, meet people with similar sporting interests, forging new friendships and providing a peaceful atmosphere for people who are looking to unwind their weekend in a fun but meaningful way.

- The most important feature of this app is to provide a rating system for each team. The rating gives information about how often a team turns up for a match and keeps its word. In this way, a team looking for an opponent would know whether they can trust the opponent.
- The other feature of the application is to provide opportunity for people to find teams and people from the same locality. Our application will provide them an opportunity to meet these people and continue their sporting interests.

**ADVANTAGES:**

- The main advantage of the application is to provide a feature, i.e. rating system which tries to provide reliable information whether the opponent is more likely to turn up or not.
- Another advantage is for women. Even best of friends don’t necessarily like to play outdoor games, but our application enables them to find women with similar interests and this would serve them well in the longer run to keep them fit.

# CHAPTER 3

# SYSTEM ANALYSIS

## 3.1 FEASIBILITY STUDY

The main objective of feasibility study is to test the Technical, operational and economical for adding new modules and debugging old running system. All system is feasible for adding new modules and debugging old running system.

There are aspects in the feasibility study portion of the preliminary investigation:

- Technical feasibility
- Economic feasibility
- Operational feasibility

### 3.1.1 TECHNICAL FEASIBILITY

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. To develop this system, we first worked with web applications and then found that Android would be technically feasible. The requirements here are very modest because the system here supports very basic android version which is 4.1 and above. So, this system is technically feasible because it needs only android with a basic version.

### 3.1.2 ECONOMICAL FEASIBILITY

Economic feasibility is the most frequently used method for evaluating the effectiveness of the proposed system. The System is cost effective because it is freely available as android application and can be downloaded at free of cost from the internet and installs the app on their phones and can run it.

### 3.1.3 OPERATIONAL FEASIBILITY

The aspect of the study is to check the level of acceptance of the system by the user. This includes the process of training the user to use the system efficiently. This system will not threaten the user instead it is friendly in its operation. All the user needs to have is an Internet connection provided to his mobile.

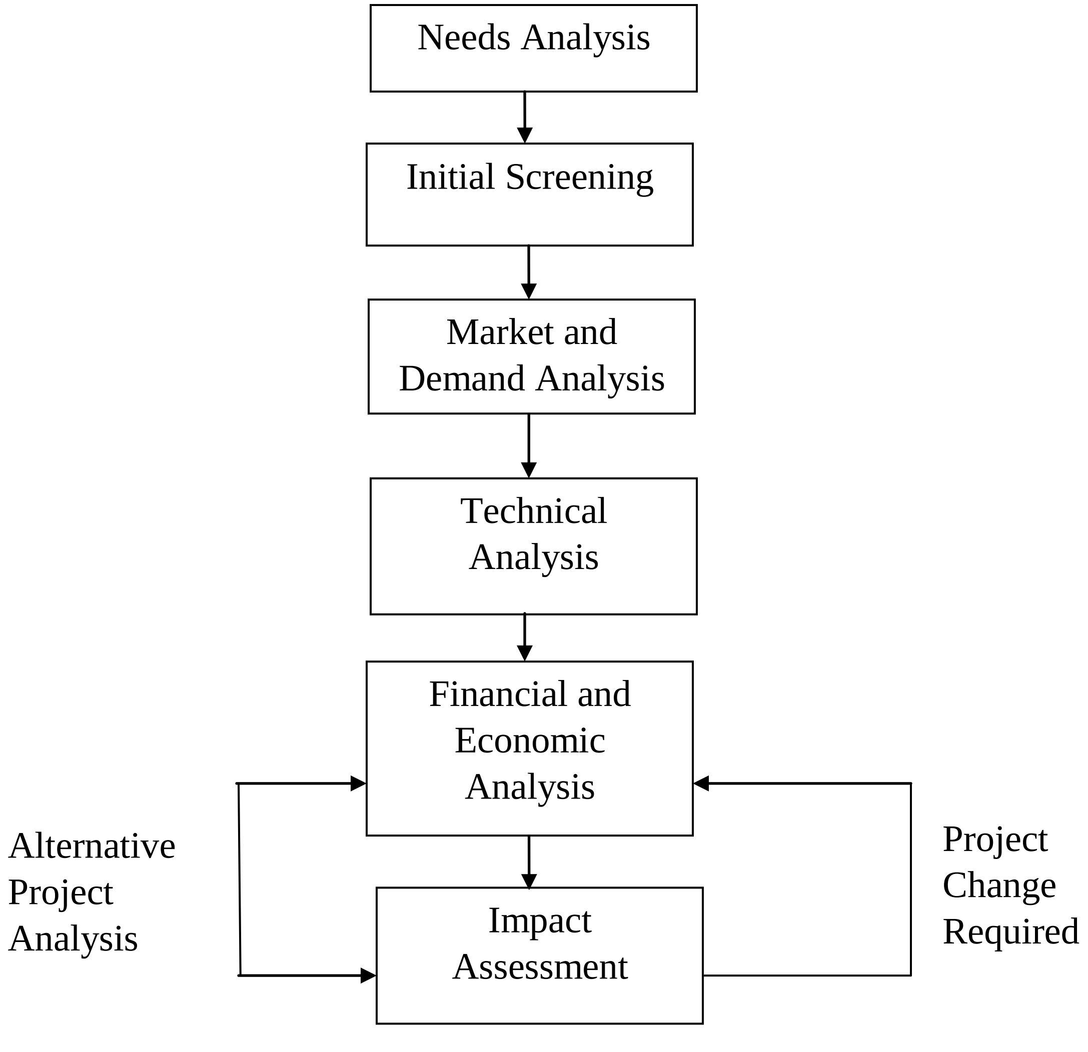


**Fig. 3.1: Feasibility Analysis**

## 3.3 HARDWARE USED

| | | |
|---|---|---|
| System | : | Core I3 processor |
| Hard Disk | : | 100 GB |
| Ram | : | 2 GB |
| Graphics | : | 1 GB Radeon |
| Device | : | Android phone |

## 3.4 SOFTWARE USED

| | | |
|---|---|---|
| Operating system | : | Android 8 pro 64-bit |
| SDK | : | Android studio |
| Coding Language | : | JAVA |
| Front End Tool | : | XML |
| Emulator | : | Android phone SDK |

# CHAPTER 4

# DETAILED DESIGN

## 4.1 SYSTEM ARCHITECTURE

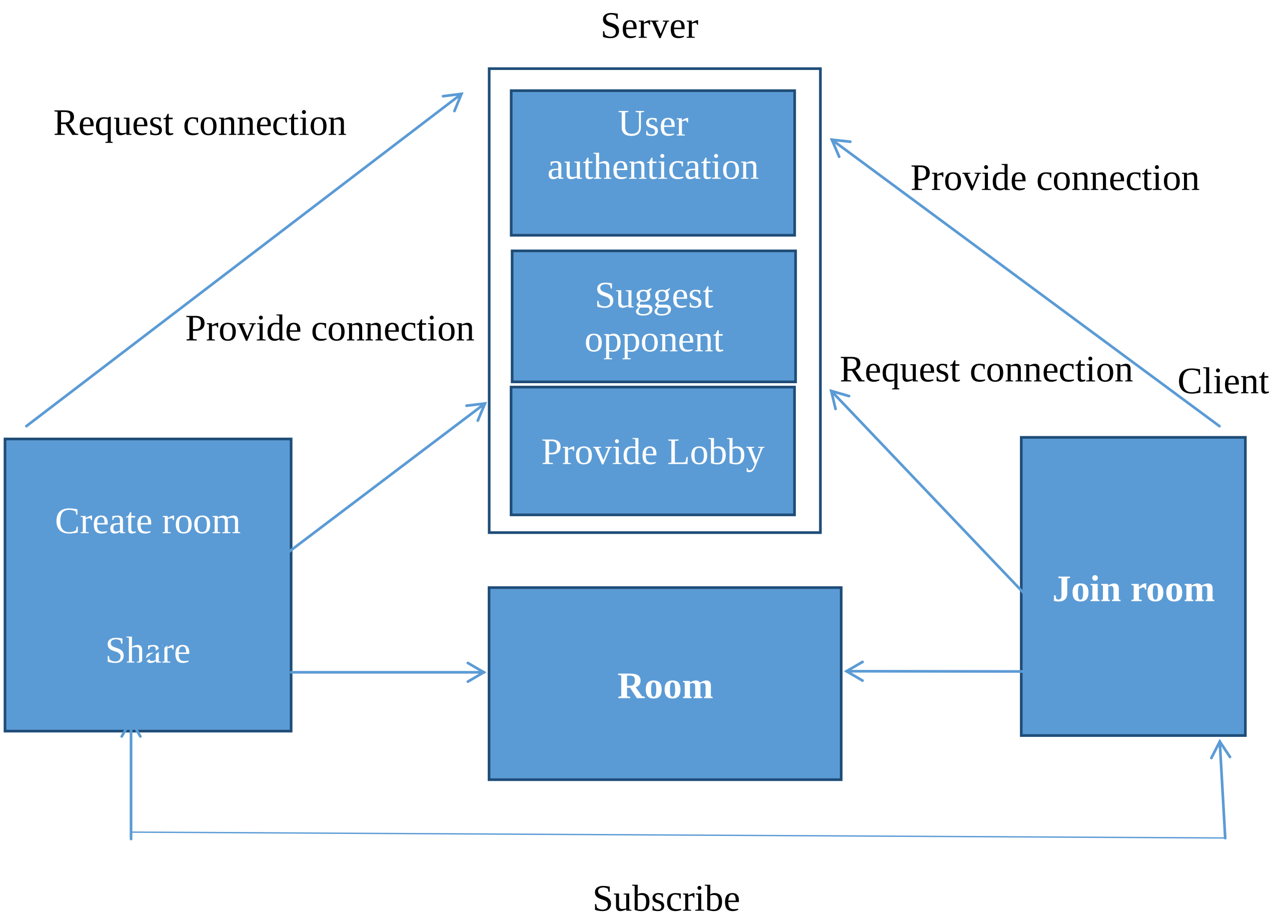


**Figure 4.1: System Architecture**

### 4.1.1 MODULES

#### Signing up (or) creating a Team

Creating a team account is the first step for anyone to access the application. New users should choose the signup option in the launcher page and provide the necessary details and register their team. The Sign Up page consists of three important options, Team name, team ID and Password.

#### Login Screen

Once the team is registered the users can use the Team ID and Password to log into the team account to setup their matches, view notifications and so on.

#### Change Password

In case the user forgets the password or for security reasons, the team members can change their password after logging into their account.

#### Team Profile

On Logging into the account the screen displays the team profile, which is the actual screen which would be visible for opponents when they click the team profile.

The Team Name for several teams maybe is the same. Each locality might have a team with the same name. In order to properly find the teams without any ambiguities each team is provided with a unique Team ID.

The Team ID is a team’s basic identity within the application and shall be used for all the important purposes within the application.

Team Rating denotes the rating of the particular team which would be visible to anyone who visits the team profile. The ratings are provided based on the number of matches the particular team has scheduled and the number of times they have turned up for the game.

### Create Match

On clicking Create Match button, a new window opens in which the details of the particular match is to be filled by the team which is creating the particular event. Once the details are filled by the user, Create button is clicked. The event is linked to an online database.

### Database

The Database would contain the list of events or matches which have already been created by various teams. The database would be an online entity like cloud storage, from which the data can be shared to other people through the internet. The details stored in the event are Team Name, Team Description, Sport and Date & Time. The details stored in the database are retrieved by the opponents when they want to join an existing match.

### Join Match

On Clicking the Join Match button, the list of matches which have already been created by other users is displayed. The users can select any one of the existing events if it matches with their preferences. The joining team can view the profile of the team which created the match and can decide whether they would be suitable opponents.

## 4.2 UML DIAGRAMS

The Unified Modeling Language (UML) is a general purpose modeling language in the field of software engineering. The basic level provides a set of graphic notation techniques to create visual methods of object-oriented software-intensive systems. Object-oriented analysis and design (OOAD) is a software engineering approach that models a system as a group of interacting objects.

### 4.2.1 USE CASE DIAGRAM

Use case describes the interaction between one or more actors and the system itself, represented as a sequence of simple steps that take part in a sequence of activities in a dialog with the system to achieve goal.

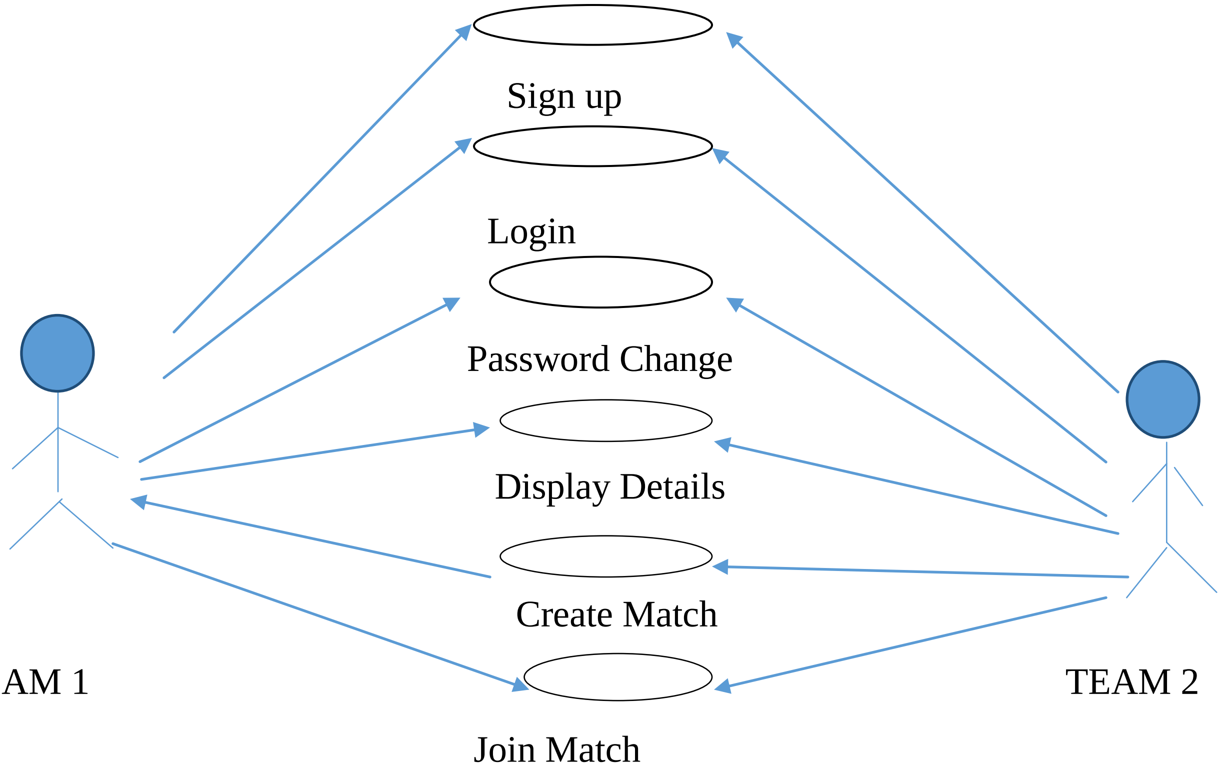


**Figure 4.2: Use case diagram**

### 4.2.2 ACTIVITY DIAGRAM

Activity diagram are graphical representation of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In the Unified Modeling Language, activity diagrams can be used to describe the business and operational step-by-step workflows of components in a system.

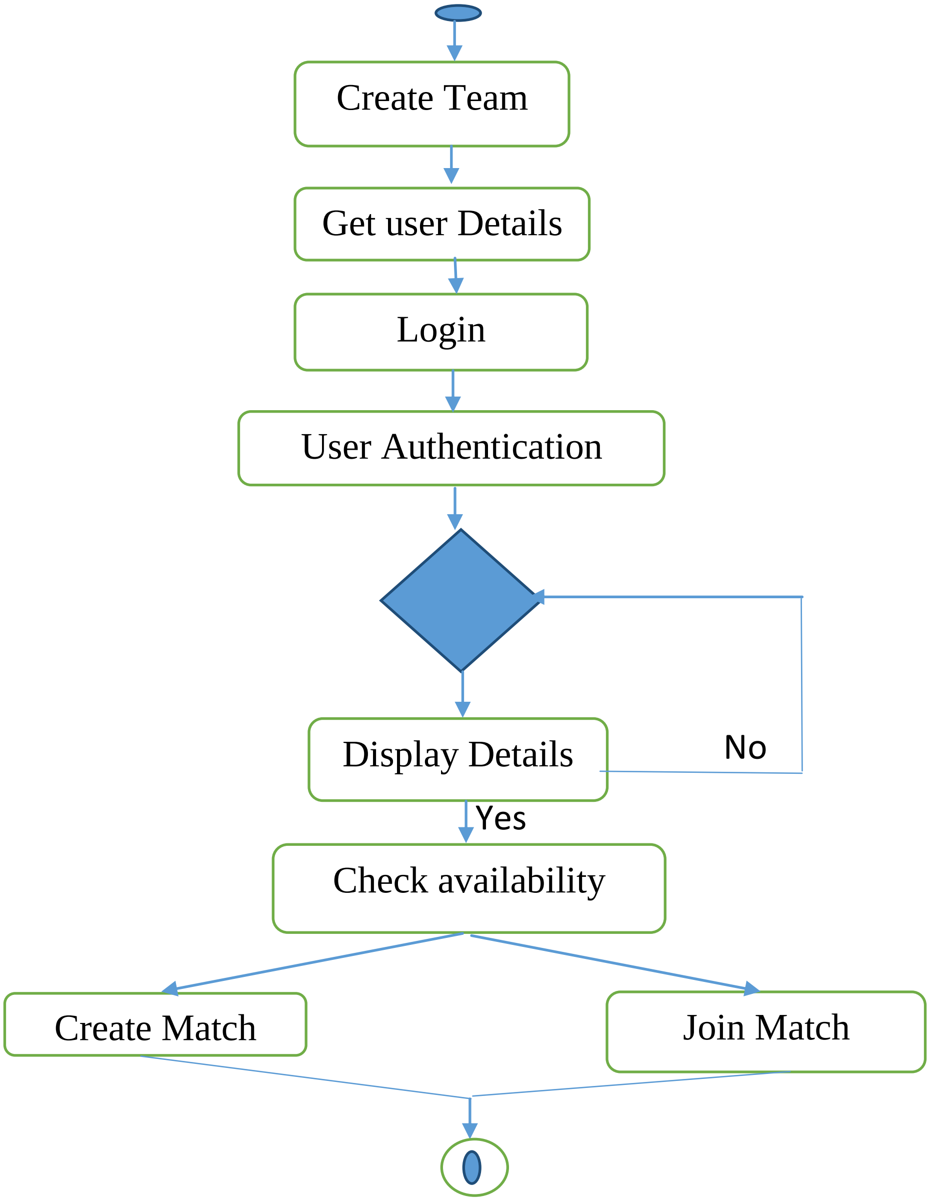


**Figure 4.3: Activity Diagram**

### 4.2.2 SEQUENCE DIAGRAM

A Sequence diagram shows, as parallel vertical lines different processes or objects that live simultaneously and as horizontal arrows, the messages exchanged between them, in the order in which they occur. This allows the specification of simple runtime scenarios in a graphical manner.

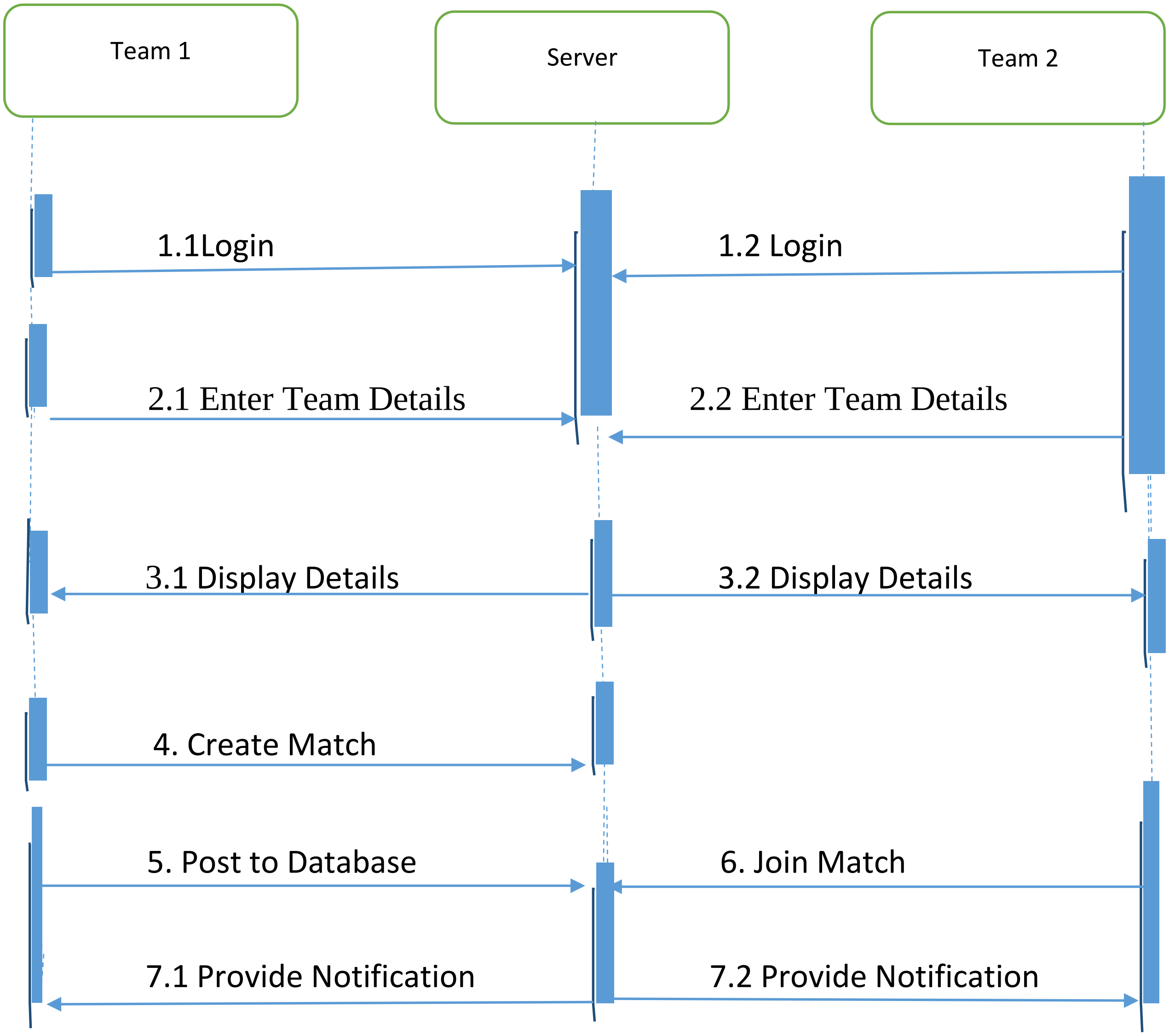


**Figure 4.4: Sequence Diagram**

### 4.2.4 DATA FLOW DIAGRAM

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system, modeling its process aspects. DFDs can also be used for the visualization of data processing.

A DFD shows what kind of information will be input to and output from the system, where the data will come from and go to, and where the data will be stored. It does not show information about the timing of process or information about whether processes will operate in sequence or in parallel.

**LEVEL 0 DFD**:

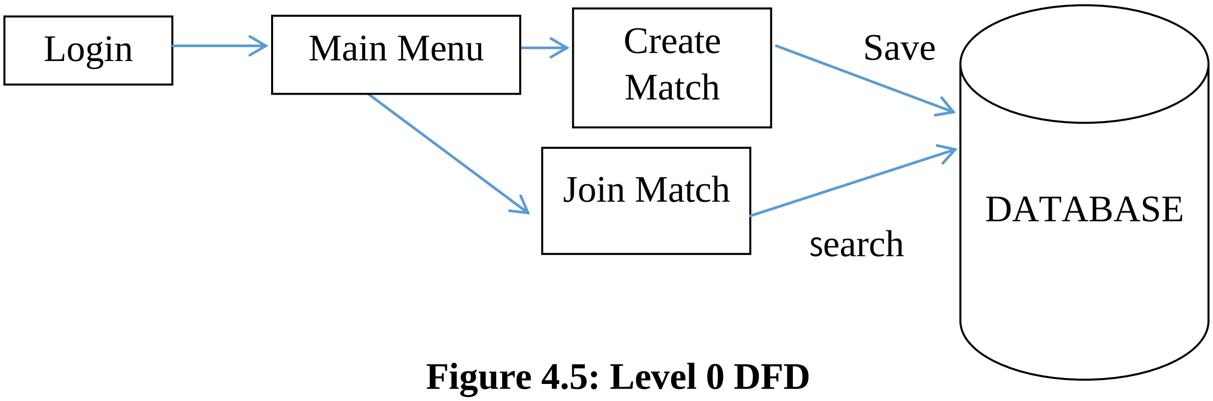


**Figure 4.5: Level 0 DFD**

**LEVEL 1 DFD**

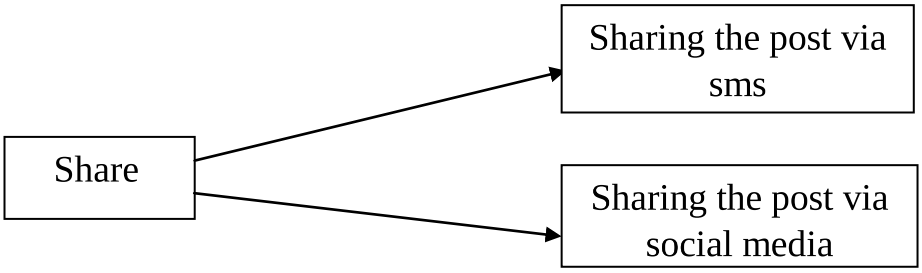


**Figure 4.6 Level 1 DFD**

**LEVEL 2 DFD**

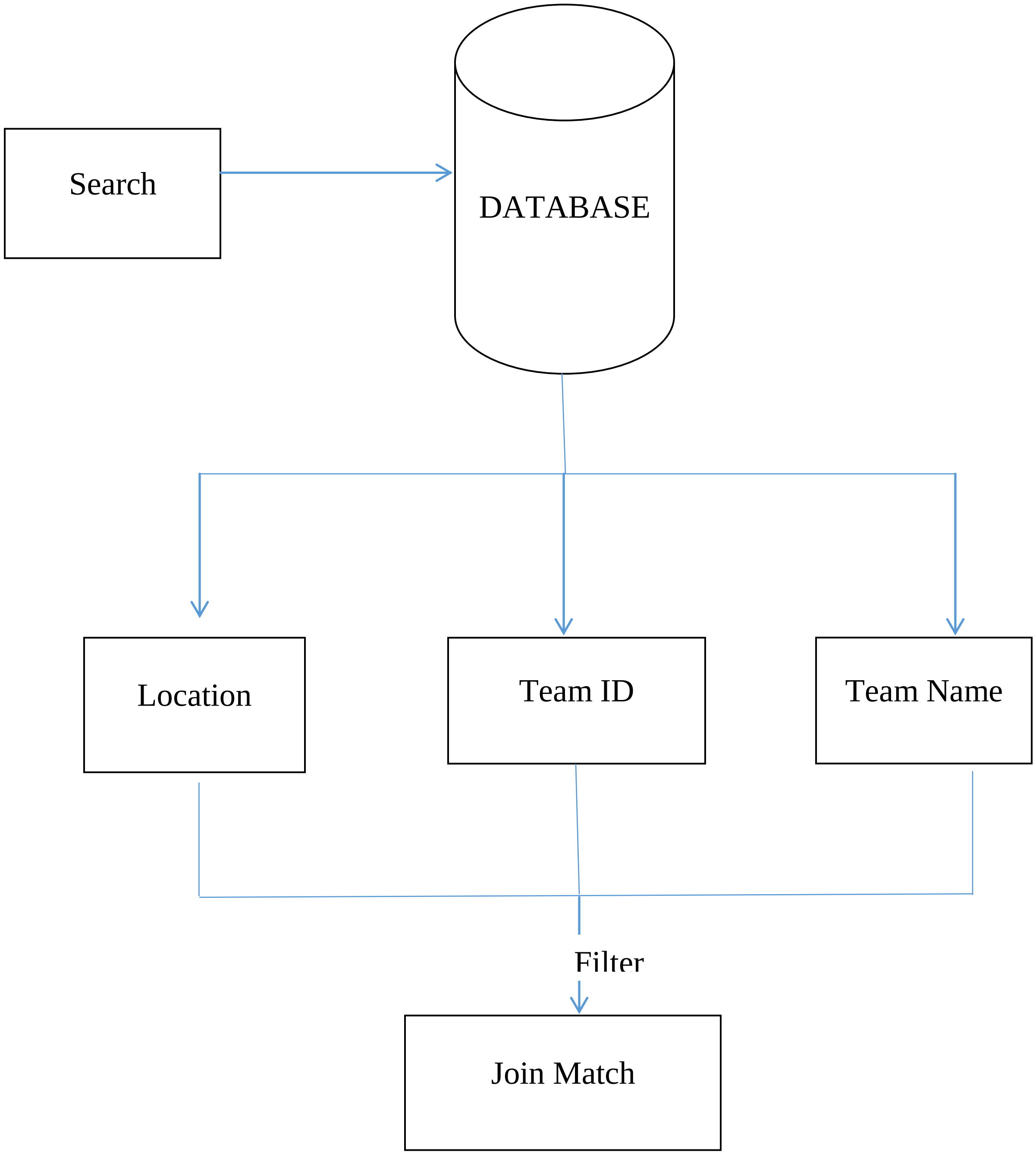


**Figure 4.7: Level 2 DFD**

## LEVEL 3 DFD:

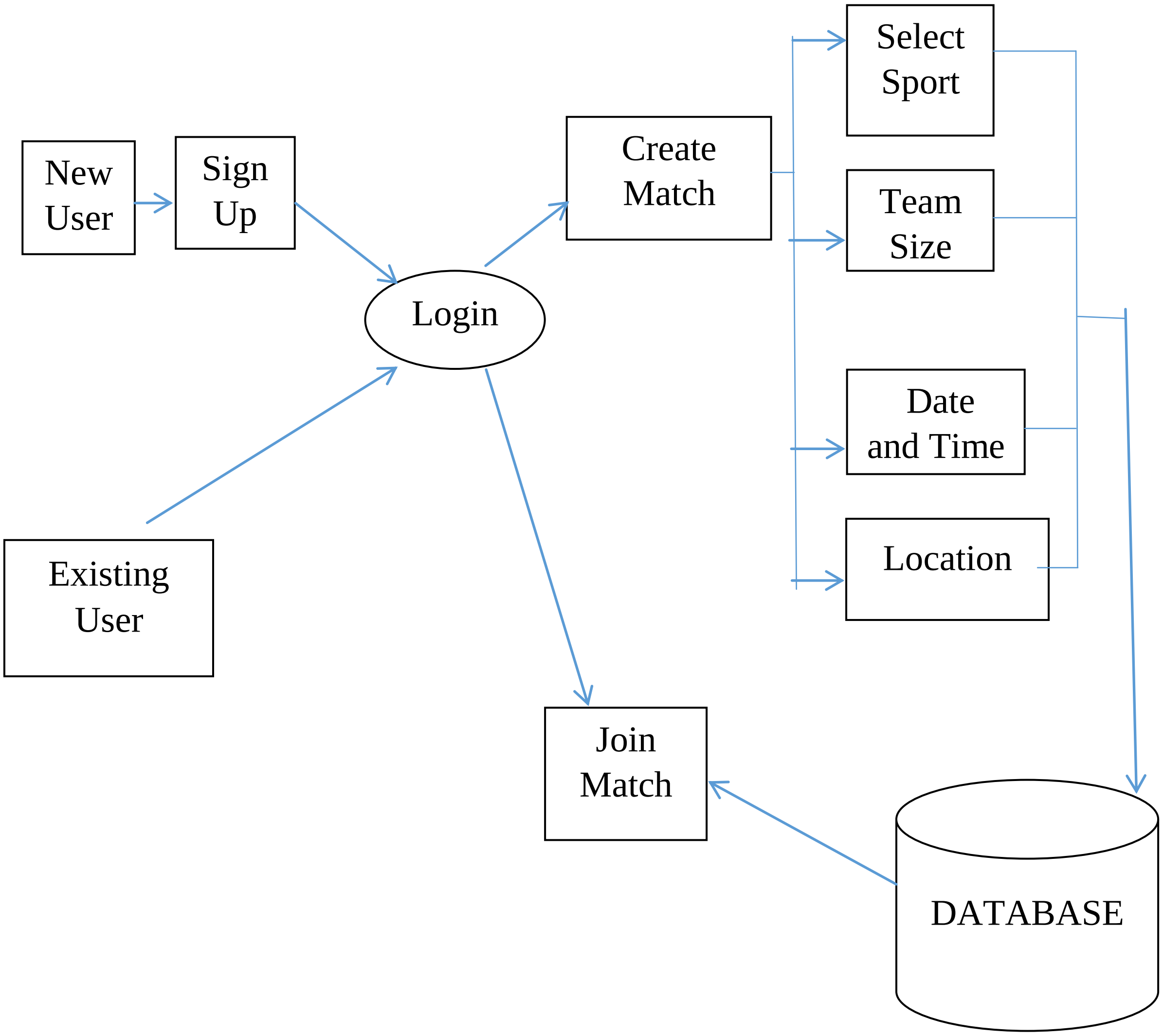


**Figure 4.8: Level 3 DFD**

# CHAPTER 5

# IMPLEMENTATION AND TESTING

## 5.1 IMPLEMENTATION

This project deals with interaction between the users and the system through an android based application. The very beginning step of our project is to register the team with username, password and team name by the user. The data entered by the user is stored in the database for future processing.

The user is provided with an option to either create a new match or joining an existing. When the user clicks the create match button it opens a new activity displaying the list of sports from which the user can select his preferred sport. On selection of sport an activity displaying four fields is displayed. They are Team Name, Location, Date and Time. The user specifies his preferred option and creates the match. On providing the required details and submitting, the data is mapped onto a database table in SQLite.

When a second user logs in he uses the join match button in the homepage to view the list of existing matches. The page displays only the location and on clicking the location a new page displaying all the details as specified by the creator is seen. If the second user is interested in that particular match he can fix-it by clicking the confirm button.

On confirmation a notification is sent to the one who created the match stating that your match has been fixed by the particular opponent. The notification maybe sent by email or through sms.

### Registration

Creating a team account is the first step for anyone to access the application. New users should choose the signup option in the launcher page and provide the necessary details and register their team. The Sign Up page consists of three important options, Team name, team ID and Password.

### Login

Once the team is registered the users can use the Team ID and Password to log into the team account to setup their matches, view notifications and so on.

### Change Password

In case the user forgets the password or for security reasons, the team members can change their password after logging into their account.

### Profile

On Logging into the account the screen displays the team profile, which is the actual screen which would be visible for opponents when they click the team profile.

The Team Name for several teams maybe is the same. Each locality might have a team with the same name. In order to properly find the teams without any ambiguities each team is provided with a unique Team ID.

The Team ID is a team's basic identity within the application and shall be used for all the important purposes within the application.

Team Rating denotes the rating of the particular team which would be visible to anyone who visits the team profile. The ratings are provided based on the number of matches the particular team has scheduled and the number of times they have turned up for the game.

### Create Match

On clicking Create Match button, a new window opens in which the details of the particular match is to be filled by the team which is creating the particular event. Once the details are filled by the user, Create button is clicked. The event is linked to an online database.

### Database

The Database would contain the list of events or matches which have already been created by various teams. The database would be an online entity like cloud storage, from which the data can be shared to other people through the internet. The details stored in the event are Team Name, Team Description, Sport and Date & Time. The details stored in the database are retrieved by the opponents when they want to join an existing match.

### Join Match

On Clicking the Join Match button, the list of matches which have already been created by other users is displayed. The users can select any one of the existing events if it matches with their preferences. The joining team can view the profile of the team which created the match and can decide whether they would be suitable opponents.

## 5.2 TESTING

Testing is an important phase that focuses on an empirical investigation in which the results describe the quality of the system. It cannot confirm system functions properly under all conditions but can establish that it fails under specific conditions. The prime purpose of testing is to guarantee that system successfully built and tested in the development phase meets all the requirements and design parameters.

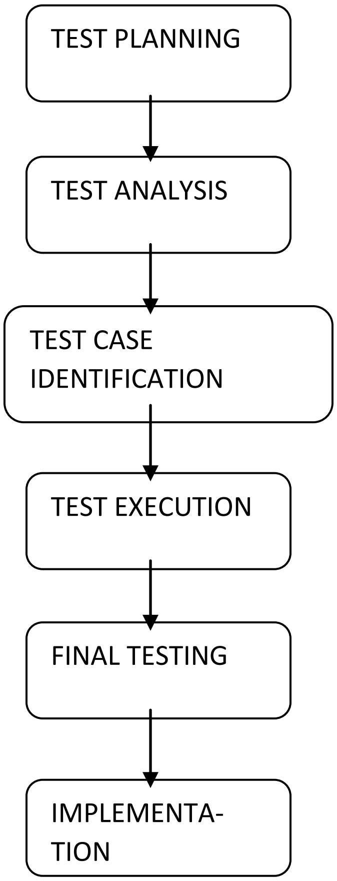


**Figure 5.1 Process of Testing**

### 5.2.1 UNIT TESTING

### SIGN UP

Table 5.1: Sign Up

| No | Test Case | Expected Output | Observed Output | Result |
|---|---|---|---|---|
| 1 | Entering Team Name, Team ID, Password. | Create profile. | The profile is displayed | Pass |
| 2 | Missing data fields. | Error message should be displayed. | Please enter all the details. | Pass |

### LOGIN

Table 5.2: Login

| No | Test Case | Expected Output | Observed Output | Result |
|---|---|---|---|---|
| 1 | Entering the username and password | Display the profile. | The profile is displayed | Pass |
| 2 | Entering the wrong user name. | Display the error message. | The error message is displayed. | Pass |

## CREATE MATCH

**Table 5.3: Create Match**

| No | Test Case | Expected Output | Observed Output | Result |
|---|---|---|---|---|
| 1 | Create a Match. | Display the Match details. | The scheduled matches are displayed to all the users. | Pass |

## SEARCH

**Table 5.4: Search**

| No | Test Case | Expected Output | Observed Output | Result |
|---|---|---|---|---|
| 1 | Type team name or team id in search engine. | Display the respected profiles of the users. | The user’s profiles are displayed. | Pass |
| 2 | Type wrong team name in search engine. | Display error message | Error message is displayed | Pass |

## CONNECT

**Table 5.5: Connect**

| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
|---|---|---|---|---|
| 1 | Click on the facebook logo. | Connect to facebook page. | The facebook page is connected. | Pass |
| 2 | Click on the twitter logo. | Connect to the twitter. | The twitter is connected. | Pass |

## JOIN MATCH

**Table 5.6: Join Match**

| **No** | **Test Case** | **Expected Output** | **Observed Output** | **Result** |
|---|---|---|---|---|
| 1 | Click on the Join Match Button | Display the scheduled matches. | The scheduled matches are displayed. | Pass |
| **2** | On joining match. | Send notification to both users. | Notification is sent successfully. | Pass |

### 5.2.2 INTEGRATION TESTING

**Table 5.7 Integration testing**

| No | Test Case | Expected Output | Observed Output | Result |
|---|---|---|---|---|
| 1 | Input the username and password. | Display the user data | User data is displayed. | Pass |
| 2 | Input the username in search box. | Display the profile of the user searched. | The searched user profile is displayed. | Pass |

### 5.2.3 ACCEPTANCE TESTING

**Table 5.8 Acceptance Testing**

| No | Test Case | Expected Output | Observed Output | Result |
|---|---|---|---|---|
| 1 | Username entered are available. | Display the user’s profile. | The users profile are displayed | Pass |
| 2 | Username entered are not available. | Do not display any profile of the user. | No profile of the user is displayed. | Pass |

## 5.3 TEST PLAN

The project is tested to verify its correctness and identify the bugs. The test plan includes the various test cases that acts as the set of conditions or variables that determine whether the corresponding feature in the system is working as it originally established to do so. When this test plan is executed, the errors spotted are rectified and the final testing yields following result.

## 5.4 TEST ANALYSIS

In this phase of testing, the requirements for software testing are analyzed and later its feasibility is determined. In the feasibility study the possibility of project development is found through suitable test cases.

## 5.5 TEST RESULT

All the test cases mentioned above passed successfully. No defects encountered.

## 5.6 RESULT

The application is tested and found to function as expected with no errors. This application provides an interface for the users to login and search for other matches and also connects to social networks and can inform about the matches created to all the other users using the application.

# CHAPTER 6

# CONCLUSION AND FUTURE ENHANCEMENT

## 6.1 CONCLUSON

The mobile users are increasing at a steady rate and so is the number of obese people in the country. The application would be helpful for people to catch up with people who have similar sporting interests and spend their time in a useful manner. The additional facility of providing a rating system for each team based on the number of matches turned up to the number of matches committed provides the much needed trust factor for the users. The search option provides an easier way for the teams to lookup for existing matches and chooses the appropriate one according to their requirements. The database is dynamic and the events scheduled are removed from the database once the time is elapsed.

## 6.2 FUTURE ENHANCEMENT

In future we would like to add certain enhancements to our application to make it more users friendly and attractive to the users. Also we would be adding provisions for creating local tournaments and also add features to increase the trust factor by providing a more comprehensive rating system. The constantly increasing obese population accompanied by a need for physical activity poses a serious threat as well as opportunity for us to constantly innovate and improve our application to suit our user needs based on further research and feedback.

# APPENDIX-A

# SAMPLE SOURCE CODE

## Main Activity.java

```
package com.example.varun.fix_it;
import android.util.Log;
import android.view.View;
import android.view.Window;
import android.view.View.OnClickListener;
import android.widget.Button;
import android.widget.EditText;
import android.widget.TextView;
import android.widget.Toast;
import android.app.Activity;
import android.app.Dialog;
import android.content.Intent;
import android.os.Bundle;

public class MainActivity extends Activity {

    LoginDataBaseAdapter loginDataBaseAdapter;
    Button login;
    Button registerr;
    EditText enterpassword;
    TextView forgetpass;

    @Override
        });

        login.setOnClickListener(new OnClickListener() {

            @Override
            public void onClick(View v) {
// TODO Auto-generated method stub
                String
Password=enterpassword.getText().toString();

                String
storedPassword=loginDataBaseAdapter.getSinlgeEntry(P
assword);

                if(Password.equals(storedPassword))
                {
                    Toast.makeText(MainActivity.this,
"Congrats:            Login            Successfully",
Toast.LENGTH_LONG).show();
                    Intent                              ii=new
Intent(MainActivity.this,Home.class);
                    startActivity(ii);
                }
                else
                if(Password.equals("")){
                    Toast.makeText(MainActivity.this, "Please
Enter Your Password", Toast.LENGTH_LONG).show();
                }
                else
                {
                    Toast.makeText(MainActivity.this,
"Password Incorrect", Toast.LENGTH_LONG).show();
                }
            }
        });

        forgetpass.setOnClickListener(new
OnClickListener() {

            @Override
            public void onClick(View v) {
// TODO Auto-generated method stub

                final Dialog dialog = new
Dialog(MainActivity.this);
                dialog.getWindow();
dialog.requestWindowFeature(Window.FEATURE_NO_
TITLE);

dialog.setContentView(R.layout.activity_forget_search);
        dialog.show();

        final EditText
security=(EditText)dialog.findViewById(R.id.securityhin
t_edt);
        final TextView
getpass=(TextView)dialog.findViewById(R.id.textView3
);

        Button
ok=(Button)dialog.findViewById(R.id.getpassword_btn)
;
        Button
cancel=(Button)dialog.findViewById(R.id.cancel_btn);

        ok.setOnClickListener(new
View.OnClickListener() {

          public void onClick(View v) {
```

```
                    String
userName=security.getText().toString();
                    if(userName.equals(""))
                    {

Toast.makeText(getApplicationContext(),  "Please  enter
your security hint", Toast.LENGTH_SHORT).show();
                    }
                    else
                    {
                       String
storedPassword=loginDataBaseAdapter.getAllTags(user
Name);
                       if(storedPassword==null)
                       {

Toast.makeText(getApplicationContext(),  "Please  enter
correct security hint", Toast.LENGTH_SHORT).show();
                       }else{
                          Log.d("GET
PASSWORD",storedPassword);
                          getpass.setText(storedPassword);
                       }

                    };
```

**Layout.xml**

```
<RelativeLayout
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:background="#FFFFFF"
    tools:context=".MainActivity" >

    <RelativeLayout
        android:id="@+id/rel1"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:layout_marginTop="60dp" >

    </RelativeLayout>

    <RelativeLayout
        android:id="@+id/rel2"
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/rel1"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:layout_marginTop="15dp" >

    </RelativeLayout>

    <RelativeLayout
        android:layout_width="match_parent"
        android:layout_height="wrap_content"
        android:layout_below="@+id/rel2"
        android:layout_marginLeft="30dp"
        android:layout_marginRight="30dp"
        android:layout_marginTop="15dp"
        android:id="@+id/relativeLayout2">

    </RelativeLayout>

    <TextView
        android:id="@+id/textView2"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
        android:text="Forgot password"
        android:layout_marginTop="78dp"
        android:layout_below="@+id/login_btn"
        android:layout_alignLeft="@+id/login_btn"
        android:layout_alignStart="@+id/login_btn" />

      <Button
        android:id="@+id/login_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:paddingLeft="20dp"
        android:paddingRight="20dp"
        android:text="Login"
        android:layout_centerVertical="true"
        android:layout_alignLeft="@+id/relativeLayout2"
        android:layout_alignStart="@+id/relativeLayout2"
    />

      <Button
        android:id="@+id/register_btn"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"

        android:paddingLeft="20dp"
        android:paddingRight="20dp"
```

```
        android:text="register"
        android:layout_alignTop="@+id/login_btn"
        android:layout_alignRight="@+id/relativeLayout2"
        android:layout_alignEnd="@+id/relativeLayout2"
/>

    <EditText
        android:id="@+id/password_edt"
        android:layout_width="match_parent"
        android:layout_height="40dp"

        android:ems="10"
        android:inputType="textPassword"
        android:padding="5dp"
        android:layout_above="@+id/login_btn"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true">

        <requestFocus />
    </EditText>
    <TextView
        android:id="@+id/textView1"
        android:layout_width="wrap_content"
        android:layout_height="wrap_content"
```

```
                android:text="password"
                android:layout_above="@+id/password_edt" />

</RelativeLayout>
```

**CreateMatch.java**

```
package com.example.varun.fix_it;


import android.content.Intent;
import android.support.v7.app.ActionBarActivity;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.AdapterView;
import android.widget.ArrayAdapter;
import android.widget.ListView;
import android.widget.Toast;
public class creatematch extends ActionBarActivity {
    private static ListView list_view;
    private static String[]
Sports={"Cricket","Football","Basketball","Badminton",
"Gaming"};
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_creatematch);
        listVIew();
    }

    public void listVIew(){
        list_view=(ListView)findViewById(R.id.listView);
        ArrayAdapter<String> adapter=new
ArrayAdapter<String>(this,R.layout.sport_list,Sports);
        list_view.setAdapter(adapter);
        list_view.setOnItemClickListener(
                new AdapterView.OnItemClickListener() {
                    @Override
                    public void onItemClick(AdapterView<?>
parent, View view, int position, long id) {
                        String
value=(String)list_view.getItemAtPosition(position);

                      Intent i=new
Intent("com.example.varun.create.sport");
                        startActivity(i);

                    }
                }
        );
```

```
          }
          @Override
          public boolean onCreateOptionsMenu(Menu menu) {
        .

        getMenuInflater().inflate(R.menu.menu_creatematch,
        menu);
            return true;
          }

          @Override
          public boolean onOptionsItemSelected(MenuItem
        item) {
int id = item.getItemId()
            if (id == R.id.action_settings) {
              return true;
            }

            return super.onOptionsItemSelected(item);
```

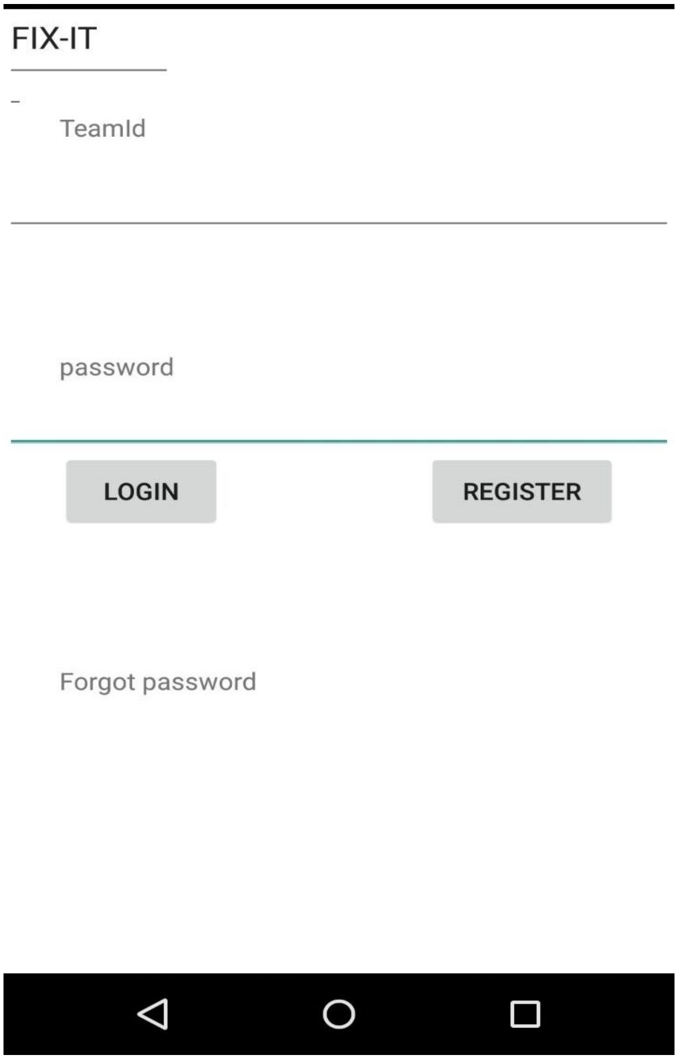
**Creatematch.xml**

```
<RelativeLayout
xmlns:android="http://schemas.android.com/apk/res/android"
android:paddingLeft="@dimen/activity_horizontal_margin"

android:paddingRight="@dimen/activity_horizontal_margin"
android:paddingTop="@dimen/activity_vertical_margin"

android:paddingBottom="@dimen/activity_vertical_margin"
android:background="#FFFFFF"

tools:context="com.example.varun.create.creatematch">
    <ListView
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:id="@+id/listView"
        android:layout_alignParentLeft="true"
        android:layout_alignParentStart="true"
        android:layout_alignParentRight="true"
        android:layout_alignParentEnd="true" />
</RelativLayout>
```

# APPENDIX-B

## SCREEN SHOTS

### LOGIN - PAGE



FIX-IT

TeamId

password

**Figure b.1: Login Page**

When the user enters into the application, the first page that appears is the login page.

## REGISTER PAGE

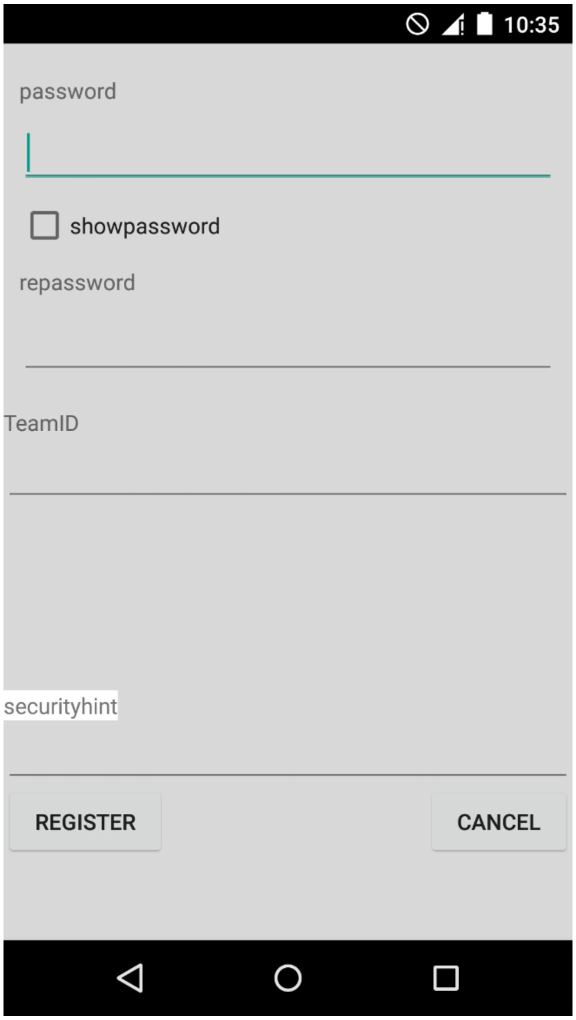
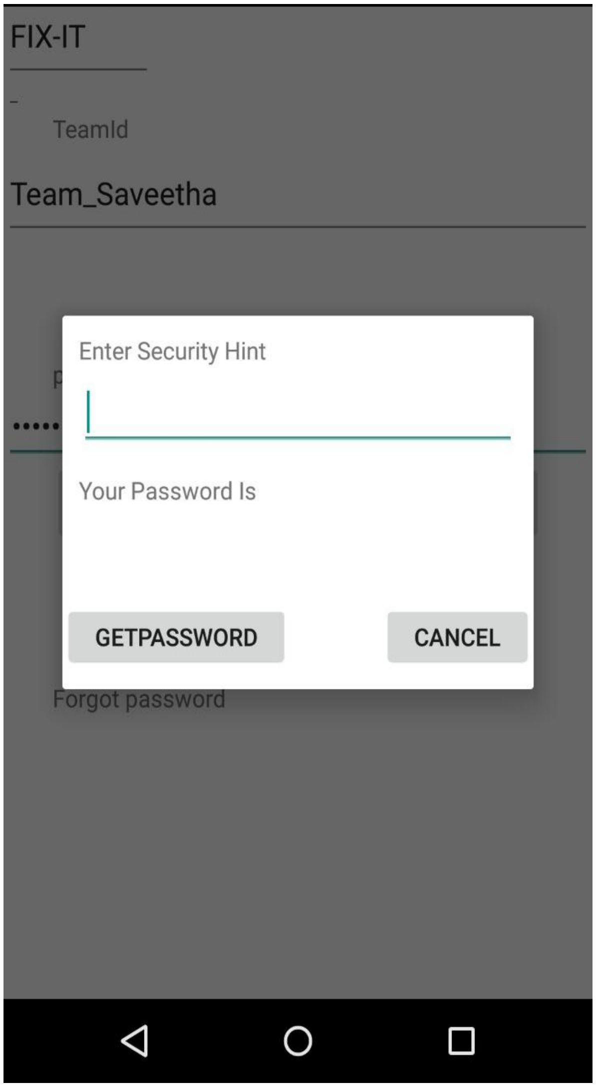


**Figure b.2 Registration Page**

The registration page consists of  team id, password an security hint.

## FORGOT PASSWORD



FIX-IT

TeamId

Team_Saveetha

Enter Security Hint

Your Password Is

**Figure b.3: Forgot Password**

If the user forgets his password he uses the above page to enter the security hint and get the password.

## TEAM PROFILE

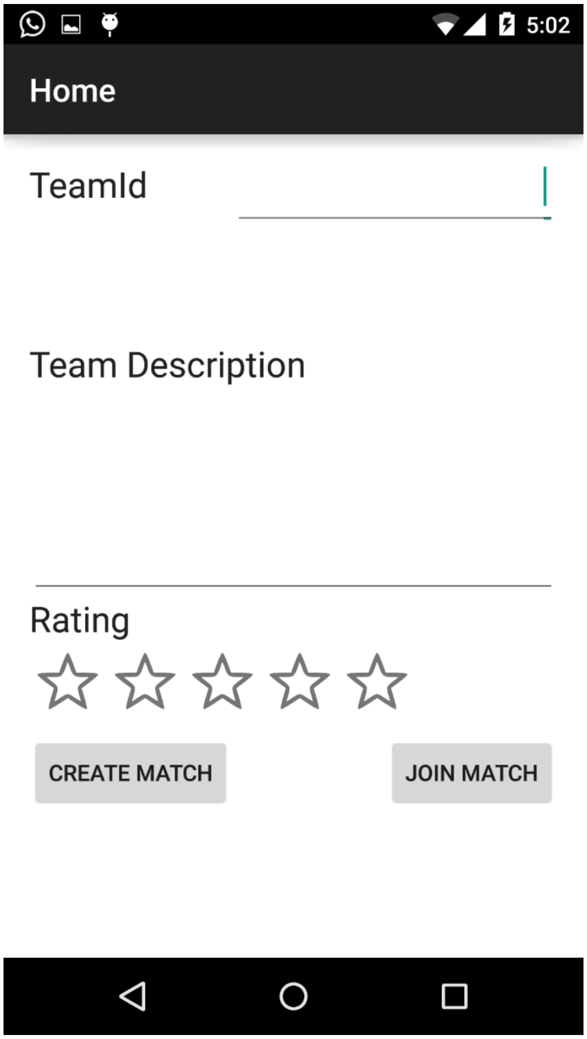
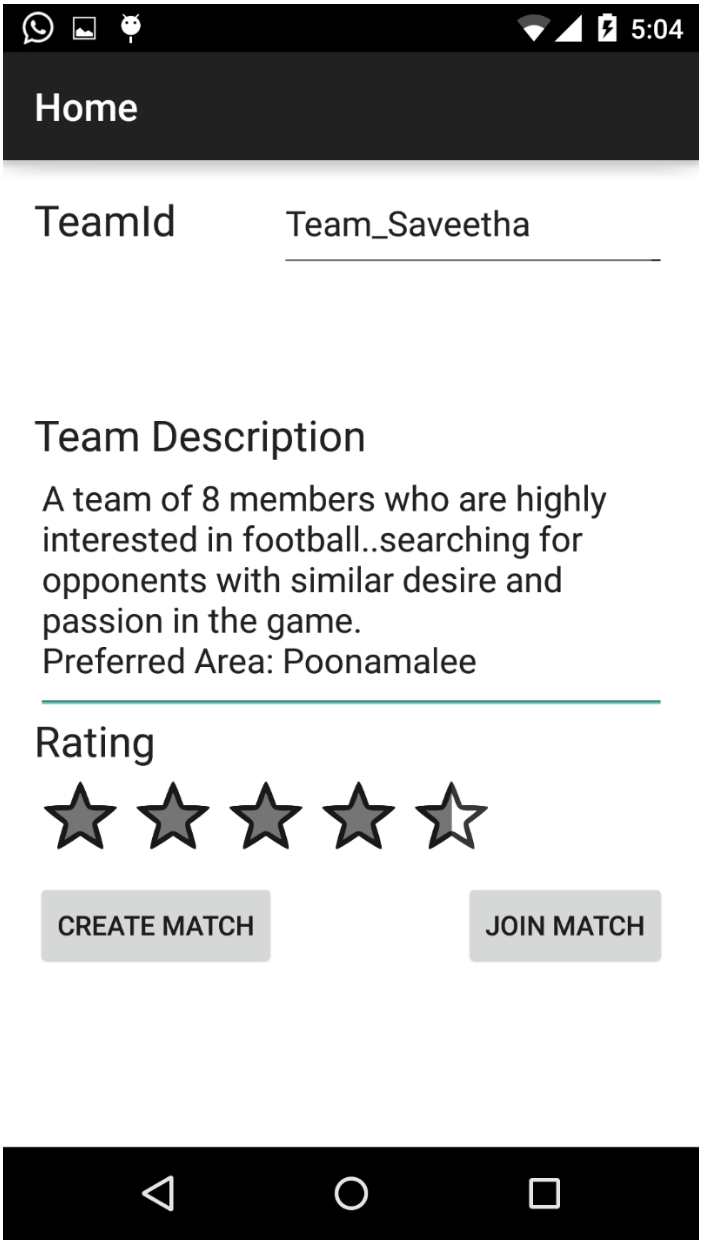


**Figure b.4: Team Profile**

The team Profile is entered in this page. This consists of Team Name, Team Description and Rating.

## TEAM PROFILE AFTER DATA ENTRY



TeamId Team_Saveetha

Team Description

A team of 8 members who are highly interested in football..searching for opponents with similar desire and passion in the game.
Preferred Area: Poonamalee

**Figure b.5 Team Profile After Data Entry**

## CREATE MATCH

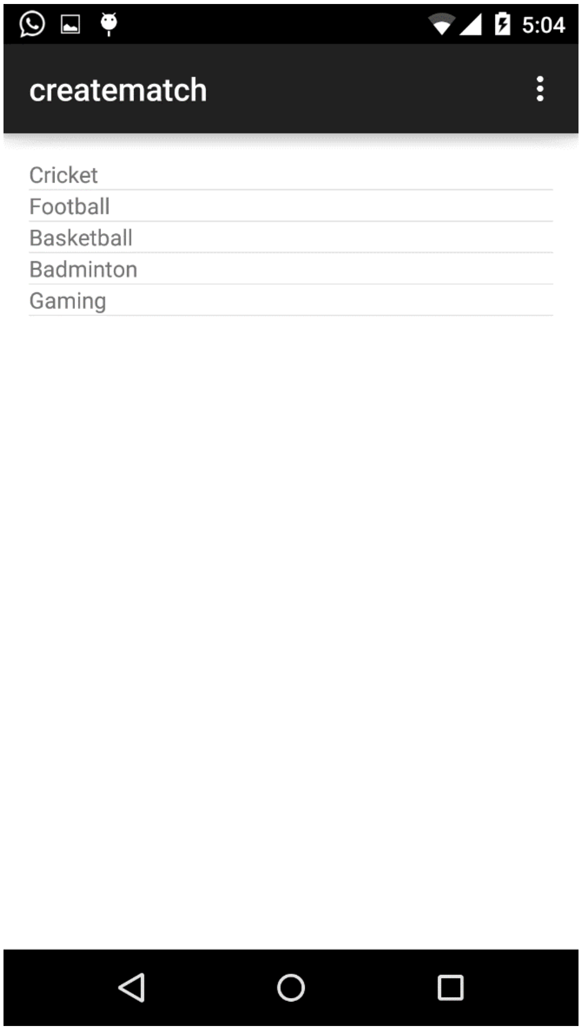


**Figure b.6: Create Match**

The Create Match page consists of the list of sports or games from which the user can choose his preferred option.

## SPORT PAGE -BEFORE DATA ENTRY

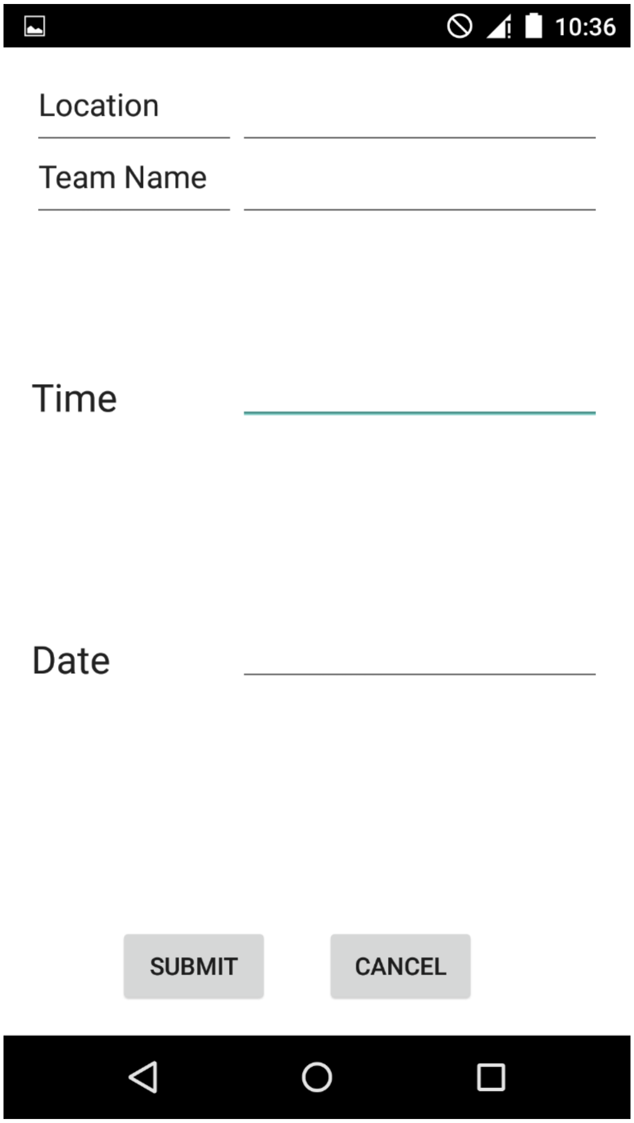
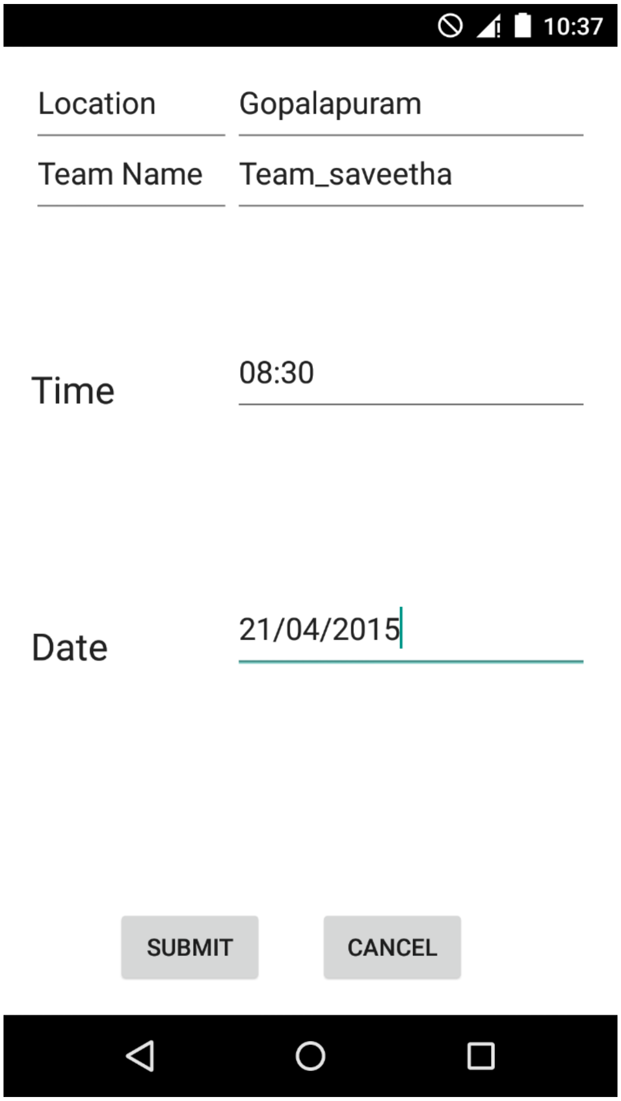


**Figure b.7: Before Data Entry**

When the user clicks on the sport in “creatematch” page the above page opens where the data is entered.

## SPORT PAGE



Location Gopalapuram

Team Name Team_saveetha

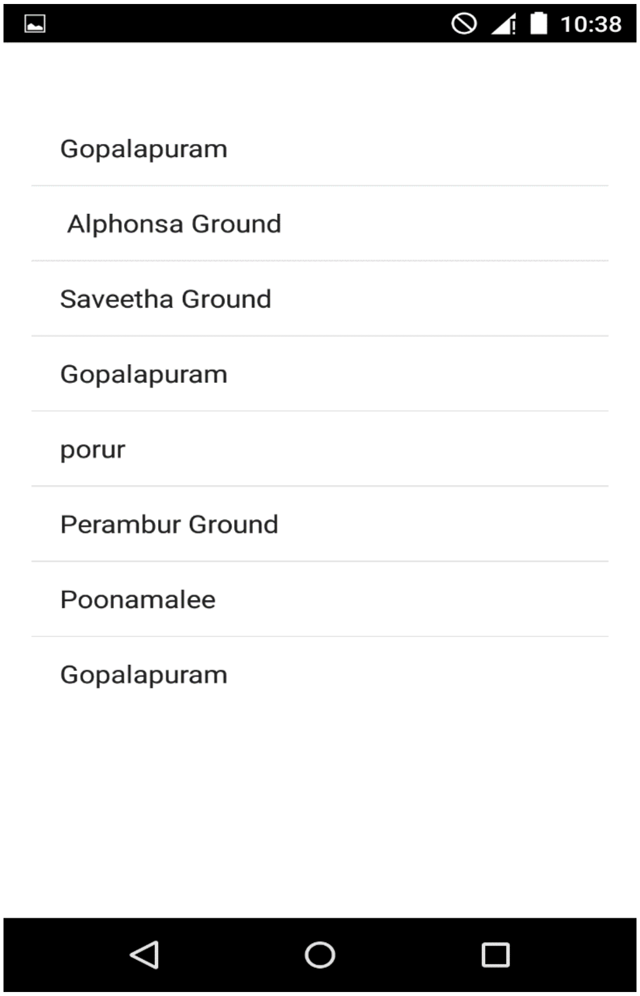
Time 08:30

Date 21/04/2015

**Figure b.8: Sport Page**

The above page is meant for setting up the match and submitting onto the database.

## JOIN MATCH PAGE



10:38

Gopalapuram

Alphonsa Ground

Saveetha Ground

Gopalapuram

porur

Perambur Ground

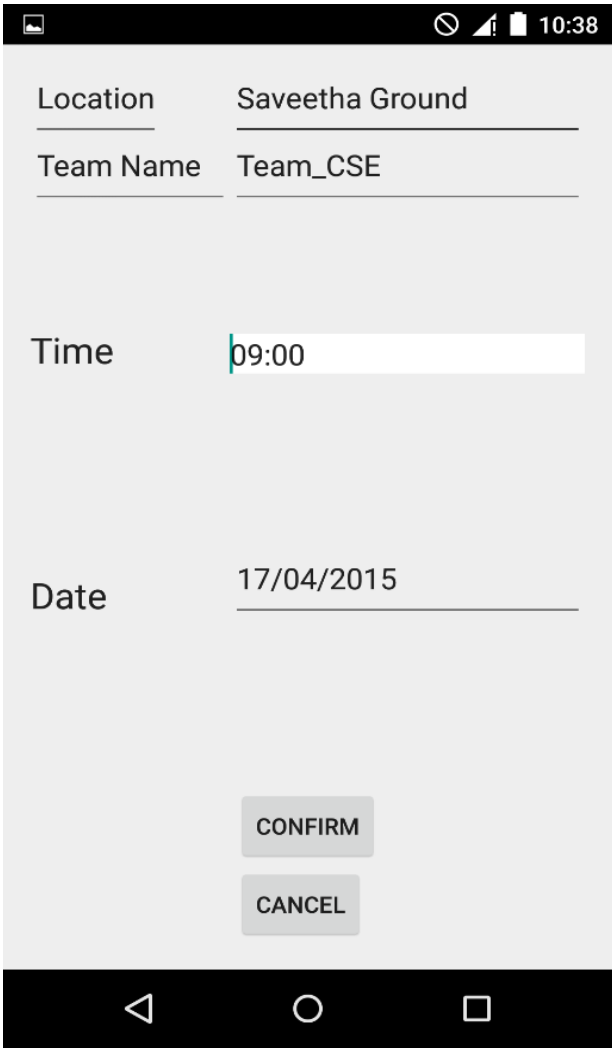
Poonamalee

Gopalapuram

**Figure b.9 Join match**

The matches created in the create match page is passed into SQLite database and displayed in the Join Match Page.

## CONFIRMATION PAGE



Location Saveetha Ground

Team Name Team_CSE

Time 09:00

Date 17/04/2015

**Figure b.10: Confirmation Page**

The user views the match created and confirms the match if it satisfies his requirements.

# REFERENCES

[1] “Professional Android 4 Application Development” by Reto Meier –

[2] “Programming Android Java Programming for the New Generation of Mobile Devices” by Zigurd Mennieks, Laird Dornin , G. Blake Meike ,& Mausmi Nakamura

[3] “Android Cookbook” by Ian F Darwin

[4] “Hello, Android Introducing Google’s Mobile Development Platform” by EdBurnette

[5] “Learning Android Building Applications for the Android Market” by Marko Gargenta

[6] T. Joachims, Text categorization with support vector machines. Technical report, LS VIII Number 23, University of Dortmund, 1997.

[7] C.-W. Hsu and C.-J. Lin, A comparison of methods for multiclass support vector machines, IEEE Transactions on Neural Networks, Vol. 13, pp. 415– 425, March 2002.

[8] Giles M. Foody, and Ajay Mathur, A Relative Evaluation of Multiclass Image Classification by Support Vector Machines, IEEE Transactions on Geoscience and Remote Sensing, Vol. 42, No. 6, June 2004.

[9] J. Cervantes, X. Li and W. Yu, Support Vector Classification for Large Data Sets by Reducing Training Data with Change of Classes, IEEE International Conference on Systems, Man and Cybernetics, 2008.

[10] Jiawei Han and Micheline Kamber, Datamining Concepts and Techniques, Second Edition, Morgan Kaufmann Publishers, November 2005.