CRICKET SCORE CARD WITH MAP (ONLINE)

By

S. M. Ali Fraz Bukhari
(2015-ARID-2401)

Bachelor of Computer Science
(BS-CS)

Barani Institute of Information Technology
PMAS Arid Agriculture University, Rawalpindi

August 2019

CRICKET SCORE CARD WITH MAP (ONLINE)

A report submitted in partial fulfillment of the
Requirements for the degree of
Bachelors of Computer Science

Submitted By
S. M. Ali Fraz Bukhari
(2015-ARID-2401)

Supervised By
Mr. Zahid Ahmed

Barani Institute of Information Technology
PMAS Arid Agriculture University, Rawalpindi

August 2019

CERTIFICATE

It is certified that the contents and form of thesis entitled **Cricket Score Card With Map (Online)** submitted by **S. M. Ali Fraz Bukhari** have been found satisfactory for the requirement of the degree.

SUPERVISORY COMMITTEE

PROJECT SUPERVISOR: _____

Mr. Zahid Ahmed

REPORT COORDINATOR: _____

Mr. Adeel Sohail

WRITEUP COMMITTEE HEAD: _____

Ms. Noor-Ul-Ain

ACKNOWLEDGEMENT

Final semester project is largely successful due to the effort from a number of wonderful people, who always gave their valuable advice and Lend helping hand. First of all, I thank **ALLAH**, then my **Parents** for supporting me all the time, whenever I found any kind of difficulty, or needed motivation. I sincerely appreciate the inspiration, support and guidance of all those people who have been instrumental in making this project a success. After that I am very thankful to my supervisor **Mr. Zahid Ahmed** for the confidence and for providing special knowledge relative to this project. I would also like to thank to all faculty members of Barani Institute of Information Technology for their critical advice and guidance without which this project would not have been possible.

Last, but not the least, I place deep sense of gratitude towards my friends who have been constant source of inspiration during the preparation of this project.

ABSTRACT

This application is made for personnel's relative to cricket. While playing cricket, it's hard to store record of all teams and players on paper or on a computer, as papers are difficult to maintain and computer is not mobile, therefore this application is on android phone and its data will be stored on an online server. So it is major help in record keeping of cricket data. It is easy to understand and can be enhanced in quick manner.

Cricket Score Card with Map (Online) is an android application. Considering the gist of cricket record keeping, the person using this application can now easily record and maintain batsman, bowler and cricket match record. Furthermore, this application can also record batsman shots, over cricket group map, and shots are colored, to distinguish shot type that is Six, Four or other runs, i.e. singles, doubles or triples. Its interface is easy; anyone can work with it. Even if the application is somehow uninstalled or android phone is busted, one can reinstall app, as data comes from remote server using WCF REST API.

Developers should consult this document and its revision as the only source of requirements for the project they should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

`

Table of Contents

Contents **Page No.**

LIST OF FIGURES

Figure **Page No.**

LIST OF ANNEXURE

Annexure **Page No.**

CHAPTER 1
INTRODUCTION

1.1 Introduction

The project titled “Cricket Score Card With Map (Online)” is basically an Android Application. The main purpose to build this project is for those who are relative to cricket. The application provides interface for adding teams, players, playing a match and also display a score card at Non Managerial Users end. The front end of system is made in Android software development kit manager. The Managerial User will have to login and then they can create, add players, add teams create and play a match.

It is Server base application, all the data/record is being sent and received to and from WCF Web service i.e. REST API.

1.2 Problem Statement

Previous versions stored its data on the device itself, so if that device is lost, whole data is lost. Previously all systems were mostly installed on desktop or laptop computer thus decreasing mobility. User could not store batsman shots. Interfaces were confusing, a lot of bugs

1.3 Proposed Solution

Now a day all systems are being transformed towards small devices, people want to carry heavy laptops and computers to access their data, and perform cricket relative operations on them, and also if that computer/laptop is somehow mal-functioned, all the work/data would be lost. The proposed solution is a Cricket Score Card with map (Online) application. This application runs on Android Phone, stores and retrieves data from remote site. So people don’t have to carry heavy/large laptops computers, and also even if android phone is lost or malfunctioned, one can reinstall this application and retrieve data from server again.

1.4 Scope

This app will be used by people who relative to cricket, all over the world. The purpose of this documentation is to provide a platform for recording cricket data, while playing cricket and also show scores to non-managerial users. This project covers all aspects relative to cricket, i.e., adding teams, adding players, creating match, toss, and then scoring, this application furthermore records batsman shots, over cricket ground maps, the shots distinguished by sixes, fours and other using colors. Developers should consult this document and its revision as the only source of requirements for the

project they should not consider any requirements statements, written or verbal as valid until they appear in this document or its revision.

1.5 Objectives

- Especially designed for the people performing operations for match scoring.
- Create teams, add players and matches.
- Provide interface for Match scoring and also save batsman shots over map.
- To get rid of primitive record keeping that on paper of record books.
- To prevent data loss, by saving data on secure, remote server.
- Show batsman shots relative to bowler and bowler type over cricket ground map.
- Provide Non-managerial user to see match records and also see live matches details.
- Provide interface for users to see details of players and teams.

This application is better solution used to Record cricket detail. In this all functional requirements are specified in order to get a clear cut idea of project.

CHAPTER 2
PROJECT BACKGROUND

This chapter contains discussion about the related to Cricket Scoring. App which was designed in the past. Also this chapter contains discussion about related Apps in the Market. With ever changing and fast moving world it is the need of the time to develop efficient and helpful platform for cricket scoring.

2.1 Related Projects, Screen Shots & Research Articles

There are many applications are related to this. You can find few Cricket Scoring apps from the app store but none of them provide server database, record and display batsman shots over cricket ground map, and also they have a lot of bugs, errors, confusing interface. This application has Better and innovative GUI. Another major problem is that one app does not provides match scoring and displaying a score card. They have different application for displaying score card, and different application for match scoring.

Previous applications did not provide an interface for displaying Teams, and players, and their details. this application provides these functionalities. previous apps had a made the devices, on which that application was installed, caused huge lags, cause of so much amount of data, but this app stores all its data over power full servers, that is why it's also light weight. previous applications had a problem of bad interface designs, users, managerial and non-managerial, would get fed up and get stressed out by their interface designs and color scheme, this app uses pleasant color scheme, which also puts much load of GPUs of android phone. Aim of improving the user interface, both in terms of functionality and performance. Some of screen shots are as follows,

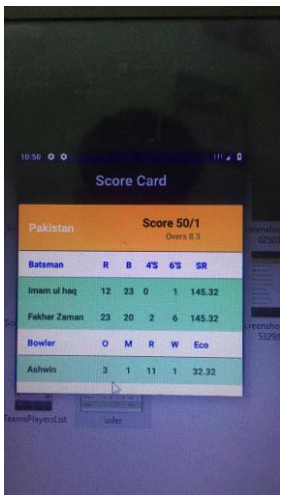

Figure 2.1 Score Card

In Figure 2.1 score card is implemented, this design does not differentiate between bowlers and batsman section, and description is also poorly displayed.

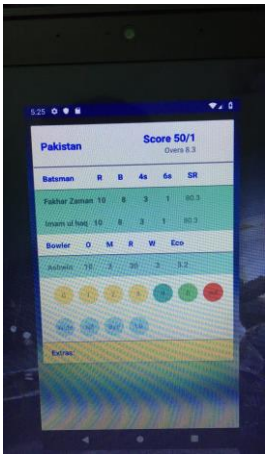

Figure 2.2 Match Scoring

Figure 2.2 is page where match scoring is conducted, as clearly shown, scoring buttons are merely looking like button, they look like labels, details for over is also not

Figure 2.3 Matches

Figure 2.3 shows discrete matches detail, but it does not show main information like match type, whether its T-20 or Test Match, Also it does not show which side was won.

The source code for Android is available under free and open source software licenses. Google publishes most of the code under the Apache License version 2.0 and the rest, Linux kernel changes, under the GNU General Public License version 2 and after it many other versions are available. Now the latest is 9.0. JSON is the backbone of every dynamic android application because using JSON data we can send and receive data from online SQLite database server using PHP language. An all apps screen lists all installed applications, with the ability for users to drag an app from the list onto the home screen. A Recent screen lets users switch between recently used apps Application those are already exits having same functionality but in this application some extra features have been added like “Cricket Score card”. Some application those work same like this application but in our app we add some extra feature.

CHAPTER 3
CONCEPTUAL DESIGN

In the previous chapter the application already designed had been expressed whereas this chapter contains the requirements briefly which are necessary to utilize and applied for designing the project. These requirements include Android Studio Software used to build android applications. This software use java language to build applications on android operating system. SQLite is the database software used to make database of different software through database queries. This is every platform and connects to every application that we want to connect. This chapter includes the discussion about the sources of requirements, main functional and non-functional requirements also about how the requirements will model in the form of DFD and how the requirements will be validated. Also it includes discussion about database design of this project and how it will be shown through ERD's and tables and how logical design will show through conceptual diagram.

3.1 Requirements Elicitation (Source of Requirement)

The question is that from where the requirements are gathered? So the requirements are gathered from director of BIIT and from my supervisor. The director has to make meet ups with me and my supervisor. In each meeting there was discussion of assigning the objectives/requirements. I had to follow each instruction and implement on it. At first they gave me a task that you have to make GUI after that they asked me to do main work that is the creating match scoring and recording batsman shots over Cricket Ground Map.

3.2 Requirement Specification

According to nature of attaining specific behavior of the application following specification were made mandatory by the supervisor and the director.

3.2.1 Functional Requirements

- Main Menu
  In this section this application will allow user to select the Option.
- Create Teams
  In this section user has to Create Teams.
- Add Players
  Here user can add players & assign team at the moment or later.
- Create Match
  Here user can create match.
- Squad Selection
  Here, after creating match, user can select players for both teams.

- Match Scoring
  Here use can record score and also this section display score at the moment.

- View Player/Team Detail
  In this Section user can view player and team details.
- Match Detail
  In this section, user can view match details

3.2.2 Non Functional Requirements

Following are the non-functional requirements which are mandatory for this application according to the international policies.

- Splash Screen

3.3 Data Flow Diagram

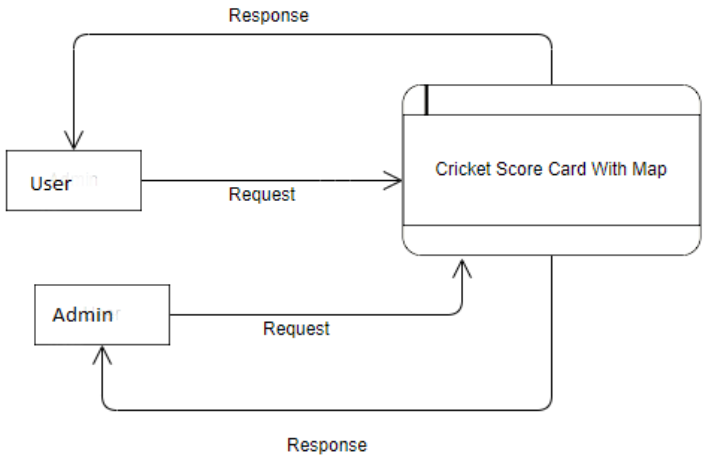

Figure 3.1: Representation of Data Flow Diagram

Approved By: Sir Zahid Ahmed

3.4 Database Design

Every application requires a database design because we have to store the data for every entity. All the data that are used in project is save in the database. Database Design is a collection of processes that facilitate the designing, development, implementation and maintenance of enterprise data management systems. Following is the design of the database of my application.

3.4.1 Entity Relationship Diagram

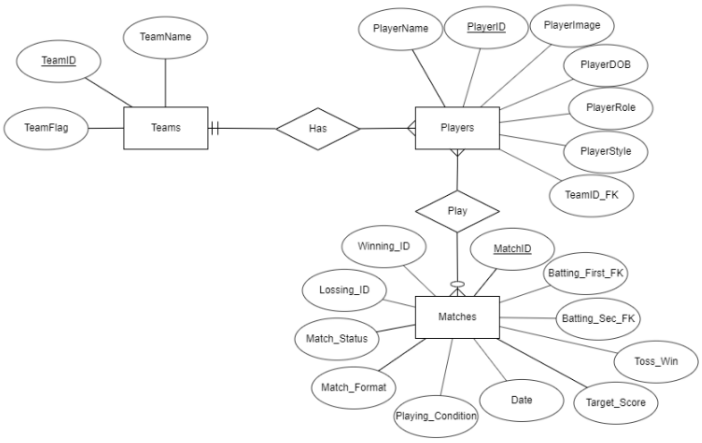

Figure 3.2: Representation of Entity Relationship Diagram

Approved By: Sir Zahid Ahmed

3.5 Logical Design

Below is the conceptual diagram of the system which represents the basic concept of who the system will work.

- **Conceptual Diagram**

A Conceptual Diagram is essentially an illustration depicting the arrangement and relationships of key attributes with in a system by using a variety of appropriate symbols that can be easily understood. Put simply, **Conceptual diagrams** are " Thought Drawings"

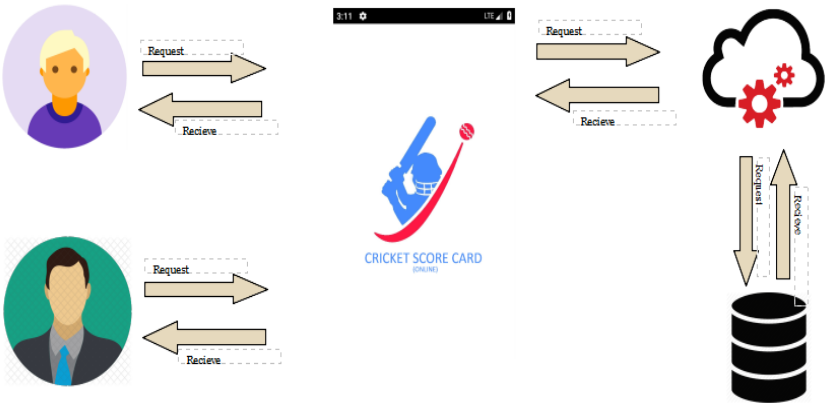

Figure 3.3 Conceptual Diagram

Approved By: Sir Zahid Ahmed

CHAPTER 4
IMPLEMENTATION

This chapter includes the discussion about the Tools and Technologies which have been used for designing of this application. It also includes how the application has been designed through coding and discussion about graphical user interface that can be shown through project screen shots.

4.1 TOOLS & TECHNOLOGIES

Following is the list of Tools and Technologies which is used to develop this application. It includes development tool, language, and database information.

- **Android Studio**

Android Studio is Application Software used to build android applications. This software use java language to build applications on android operating system.

- **SQLite Database**

SQLite is the database software used to make database of different software through database queries. This is every platform and connects to every application that we want to connect.

4.2 Pseudo code

It is an informal way to express the design of a computer program or an algorithm. The aim is to get the idea quickly and also easy to read without details. It uses the structural conventions of a normal programming language, but is intended for human reading rather than machine reading. Pseudo code typically omits details that are essential for machine understanding of the algorithm, such as variable declarations, system specific code and some subroutines. It is like a young child putting sentences together without any grammar. There are several ways of writing pseudo-code. But to reduce ambiguity between what you are required to do and what you express let’s base the pseudo code on the few defined conventions and carry out the exercise.

4.2.1 Create Match

Pseudo Code

BEGIN
If btn_createMatch_click== true
Intent (Create Match)
Dismiss
END

Code Screenshot

```
setSupportActionBar(toolbar);

adddua.setVisibility(View.INVISIBLE);
//getting database
 mydatabase=new DBHandler(MainList.this,"",null,1);
try {
    mydatabase.createDatabase();
} catch (IOException e) {
    throw new Error("unable to create database");
}

try {
    mydatabase.openDataBase();
}catch (SQLException sqle)
{
    throw new Error("unable to open database");
}

//getting database Activity table data into arrays
String[] aids;
c=mydatabase.query("Dua",null,null,null,null,null,null);
static_class.Did = c.getCount();
aids = new String[c.getCount()];
int ac=0;
if (c.moveToFirst())
{
    do {
       aids[ac] = c.getString(1);
        ac = ac + 1;
    }while (c.moveToNext());
}
for (int k = 0;k<aids.length;k++)
{
    Log.i("Aids",String.valueOf(aids[k]));
}
```

Figure 4.1 Create Match

Output

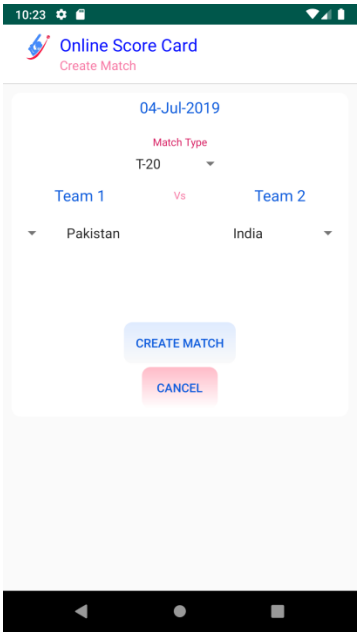

Figure 4.2 Create Match

- **Create Team**

Pseudo Code

BEGIN
If btn_createTeam_click== true
Intent (Create Team)
Dismiss
END

Code Screen Shot

```
audio.setEnabled(false);
explanation.setEnabled(false);
audio.setAlpha((float) 0.5);
explanation.setAlpha((float) 0.5);
stop.setEnabled(false);
pausebtn.setVisibility(View.INVISIBLE);
scrubber.setVisibility(View.INVISIBLE);
mydatabases = new DBHandler(duapage.this, "", null, 1);
try {
    mydatabases.createDatabase();
} catch (IOException e) {
    throw new Error("unable to create database");
}
try {
    mydatabases.openDataBase();
} catch (SQLException sqle) {
    throw new Error("unabletoopen database");
}
duaname = (TextView) findViewById(R.id.duanametxt);
duatxt = (TextView) findViewById(R.id.duatxt);
Intent d = getIntent();
duaname.setTypeface(Typeface.createFromAsset(getApplicationContext().getAssets(), "Jameel_Noori_Nastaleeq.ttf"));
duaname.setText(d.getStringExtra("duaname"));
duatxt.setTypeface(Typeface.createFromAsset(getApplicationContext().getAssets(), "Jameel_Noori_Nastaleeq.ttf"));
duatxt.setText(static_class.shared_variable);
 opendua= (ListView) findViewById(R.id.opendua);
CustomAdapter customAdapter=new CustomAdapter();
opendua.setAdapter(customAdapter);
check_fvrt();

opendua.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
```

Figure 4.3 Create Team

Output

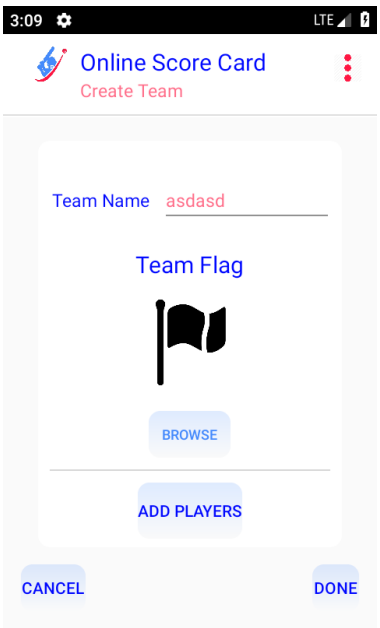

Figure 4.4 Create Team

- **Add Player**

Pseudo Code

BEGIN
If btn_addPlayer_click== true
Intent (Add Player)
Dismiss
END

Code Screen Shot

```
audio.setEnabled(false);
explanation.setEnabled(false);
audio.setAlpha((float) 0.5);
explanation.setAlpha((float) 0.5);
stop.setEnabled(false);
pausebtn.setVisibility(View.INVISIBLE);
scrubber.setVisibility(View.INVISIBLE);
mydatabases = new DBHandler(duapage.this, "", null, 1);
try {
    mydatabases.createDatabase();
} catch (IOException e) {
    throw new Error("unable to create database");
}
try {
    mydatabases.openDataBase();
} catch (SQLException sqle) {
    throw new Error("unabletoopen database");
}
duaname = (TextView) findViewById(R.id.duanametxt);
duatxt = (TextView) findViewById(R.id.duatxt);
Intent d = getIntent();
duaname.setTypeface(Typeface.createFromAsset(getApplicationContext().getAssets(), "Jameel_Noori_Nastaleeq.ttf"));
duaname.setText(d.getStringExtra("duaname"));
duatxt.setTypeface(Typeface.createFromAsset(getApplicationContext().getAssets(), "Jameel_Noori_Nastaleeq.ttf"));
duatxt.setText(static_class.shared_variable);
 opendua= (ListView) findViewById(R.id.opendua);
CustomAdapter customAdapter=new CustomAdapter();
opendua.setAdapter(customAdapter);
check_fvrt();

opendua.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
```

Figure 4.5 Add Player

Output

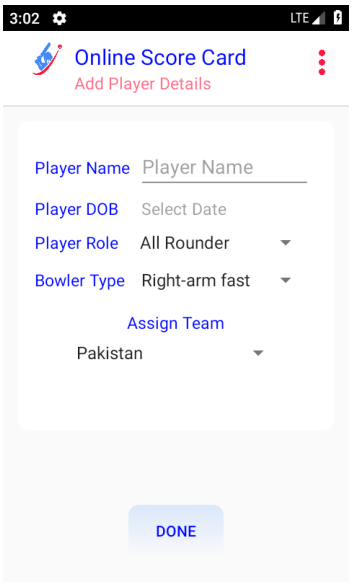

Figure 4.6 Add Player

- **Team Detail**

Pseudo Code

BEGIN
If btn_teamDetail_click== true
Intent (Team Detail)
Dismiss
END

Code Screen Shot

```
TeamsBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
        Intent i = new
Intent(getApplicationContext(),Teams.class);
        startActivity(i);
    }
});
```

Figure 4.7 Team Detail

Output

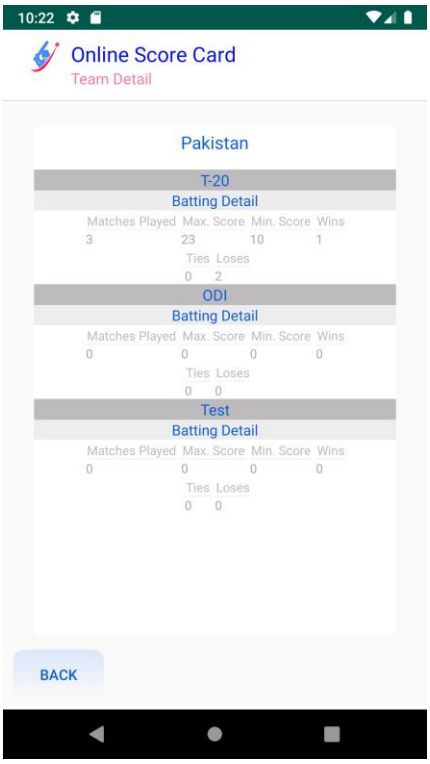

Figure 4.8 Team Detail

- **Player Detail**

Pseudo Code

BEGIN
If btn_playerDetail_click== true
Intent (Player Detail)
Dismiss
END

Code Screen Shot

```
PlayersBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View v) {
        finish();
        Intent i = new
Intent(getApplicationContext(),Teams.class);
        startActivity(i);
    }
});
```

Figure 4.9 Player Detail

Output

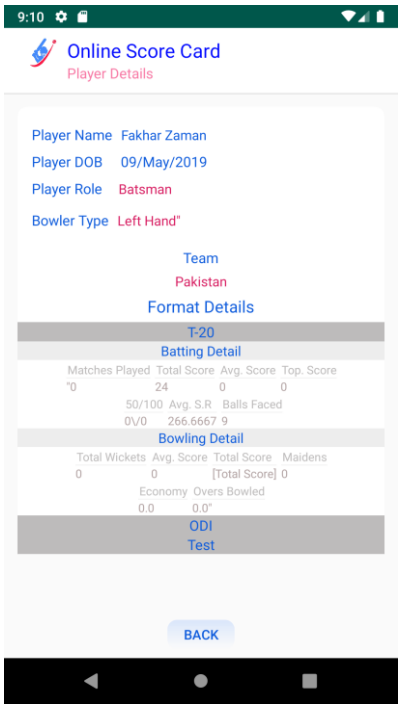

Figure 4.10 Player Detail

- **Match Scoring**

Pseudo Code

BEGIN
If matchid!=null
Intent (Match Scoring)
Dismiss
END

Code Screen Shot

```
    private void onclick(View v) {
        PopupMenu popup= new PopupMenu(getBaseContext(),v);

popup.getMenuInflater().inflate(R.menu.adminmenu,popup.getMenu());
        popup.setOnMenuItemClickListener(new
PopupMenu.OnMenuItemClickListener() {
            @Override
            public boolean onMenuItemClick(MenuItem item) {
                int itemid=item.getItemId();
                if(itemid==R.id.exititem)
                {
                    finish();
                    System.exit(0);
                }
```

Figure 4.11 Match Scoring

Output

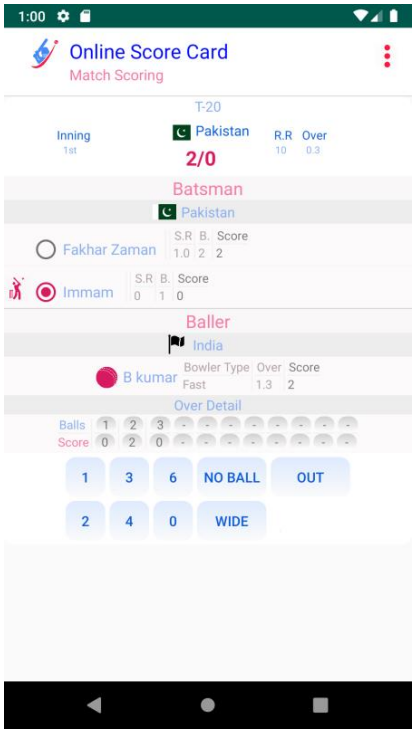

Figure 4.12 Match Scoring

- **Record Batsman Shots**

Pseudo Code

BEGIN
If btn_score1_click== true
Intent (Map Work)
Else If btn_score2_click== true
Intent (Map Work)
Else If btn_score3_click== true
Intent (Map Work)
Else If btn_score4_click== true
Intent (Map Work)
Else If btn_score6_click== true
Intent (Map Work)
Dismiss
END

Code Screen Shot

```
public class BatsmanshotsActivity extends Activity {

    TextView DateTv;
    Button CreateBtn,CancelBtn;
    Spinner MatchTypeSpinner,Team1NameSpinner, Team2NameSpinner;
    final TeamDetails team1= new TeamDetails();
    final TeamDetails team2= new TeamDetails();
    List<String> mydata = new ArrayList<String>();
     ArrayAdapter<String> CountryNameadapter;
     ArrayAdapter<String> CountryNameadapter1;

    @RequiresApi(api = Build.VERSION_CODES.O)
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_creatematch);
        CreateBtn= findViewById(R.id.CreateBtn);
        CancelBtn= findViewById(R.id.CancelBtn);
        MatchTypeSpinner = findViewById(R.id.MatchTypeSpinner);
        Team1NameSpinner=findViewById(R.id.Team1Spinner);
        Team2NameSpinner=findViewById(R.id.Team2Spinner);
        JSONTask t1 = new
JSONTask(creatematchActivity.this,"getteamgroup1",0);
        t1.execute();
```

Figure 4.13 Record Batsman Shots

Output

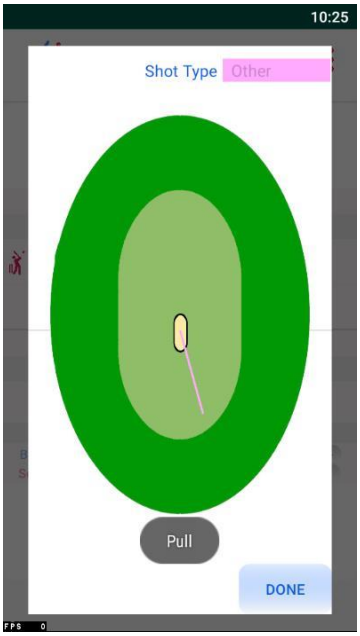

Figure 4.14 Batsman Shots

4.3 Graphical User Interface

Following are the screen shots of this project, these Screen shots provide the actual graphical user interface of the project and these are also called the user end interface.

Splash Screen

The page of screen that user sees first before begin given the option to continue to the main content of the App.

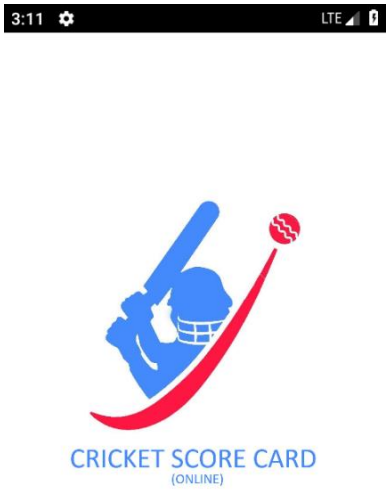

Figure 4.15 Splash Screen

Matches Page

This is the landing page after splash screen, this is 1st tab page to show discrete details of match on a list view.

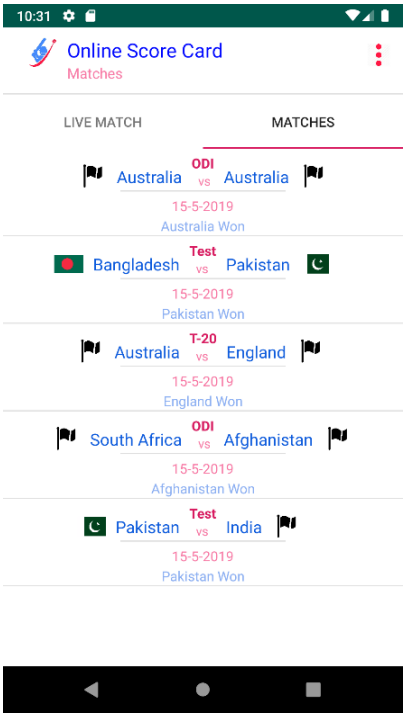

Figure 4.16 Matches Page

Matches Detail Page

This page shows match details upon selecting a certain match from precious matches list view.

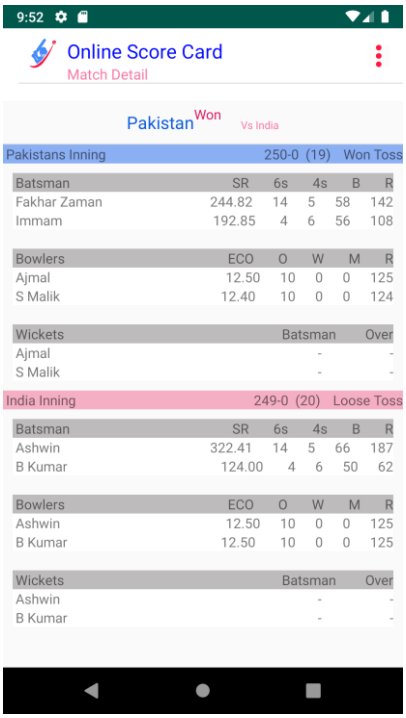

Figure 4.17 Matches Detail Page

Live Matches Page

This is the landing page after splash screen, this is 2nd tab page to show discrete details of live match on a list view.

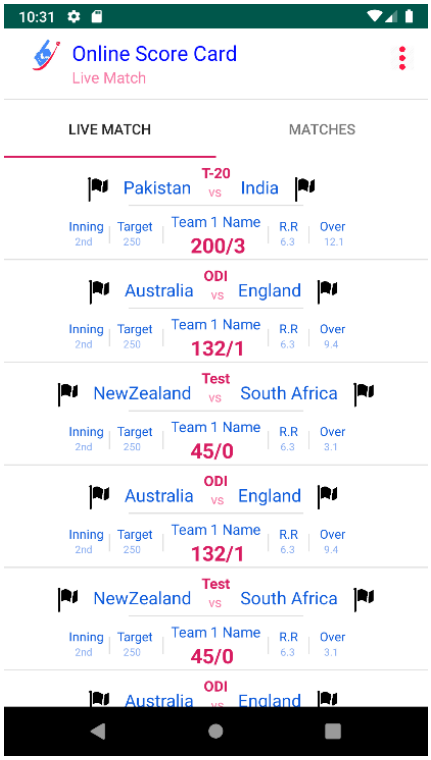

Figure 4.18 Live Matches Page

Live Matches Detail Page

This page shows live match details upon selecting a certain match from precious live matches list view.

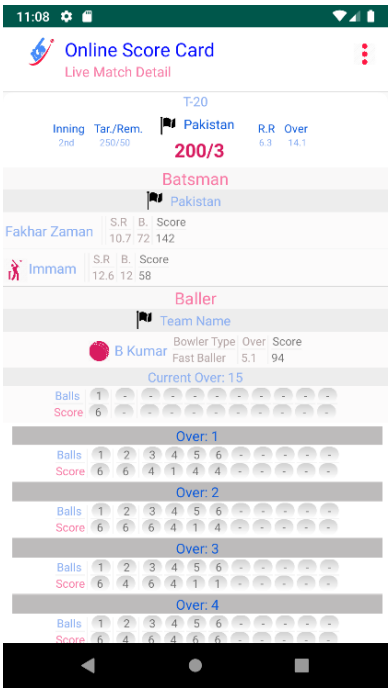

Figure 4.19 Live Matches Detail Page

Admin Menu Page

This page allows user to select option for creating match, add new teams, add new player and also see their details.

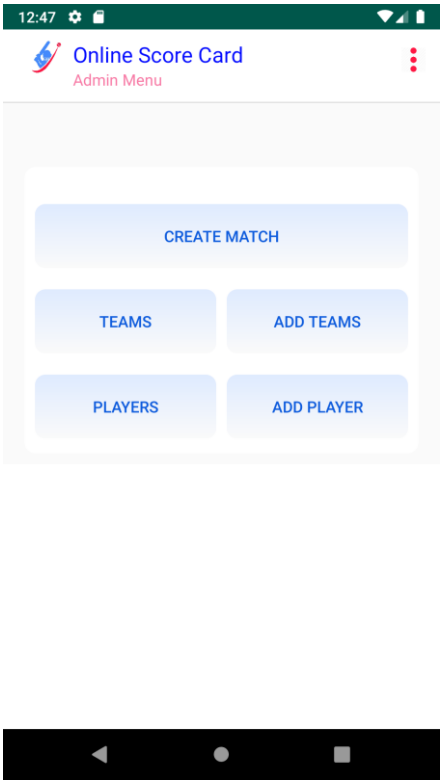

Figure 4.20 Admin Menu Page

Create Match Page

This page allows user to create a match.

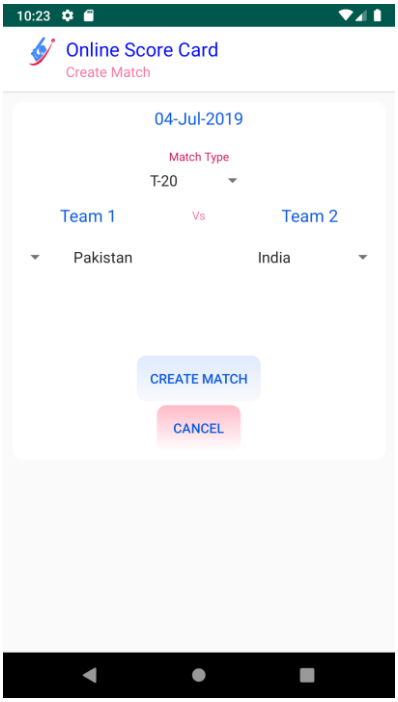

Figure 4.21 Create Match Page

Select squads Page

This Page Allows to select players to create a squad from team existing players from list view.

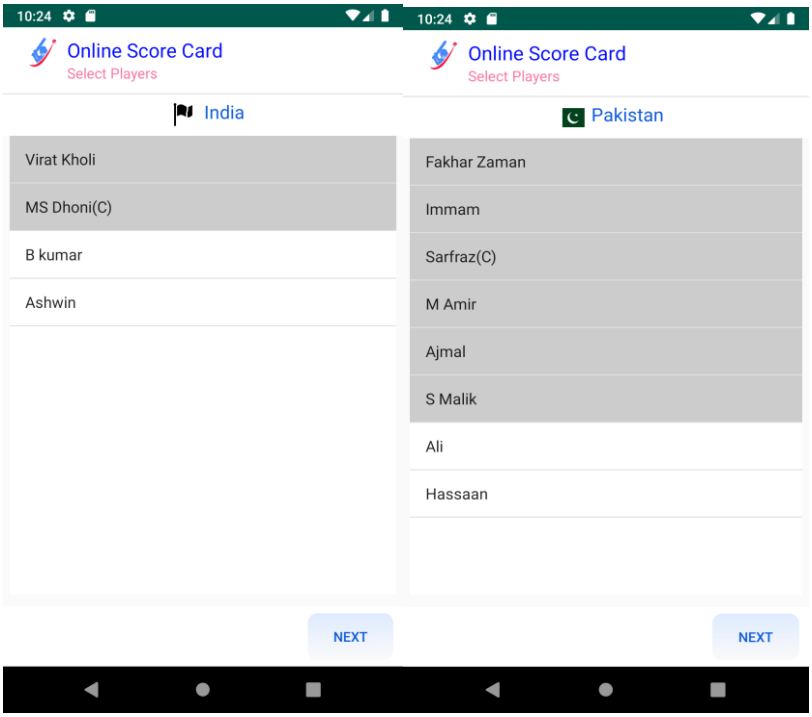

Figure 4.22, A & B, Select Squads Page for 1st & 2nd Team

Toss page

This is page for toss after selecting squads.

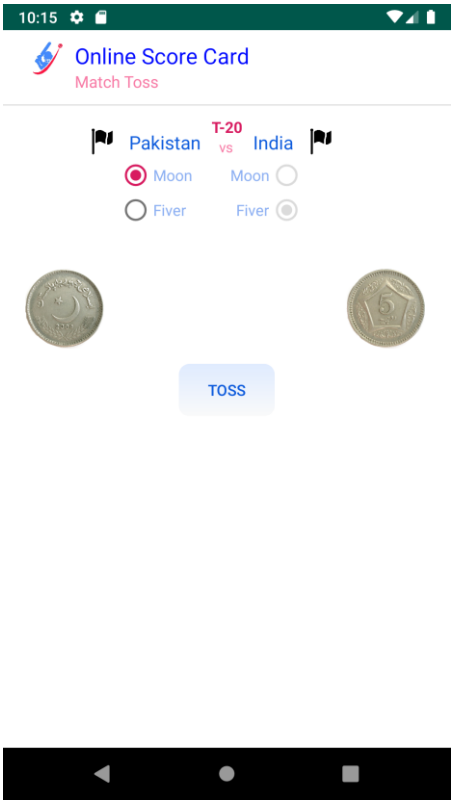

Figure 4.23 Toss Page

Match Scoring

This page allows user to record match score after toss.

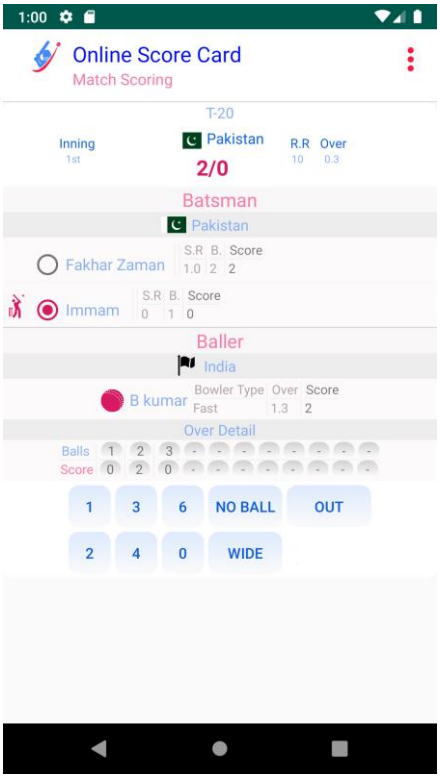

Figure 4.24 Match Scoring Page

Recording Batsman Shots Page

This page allows user to record batsman shot over cricket map ground.

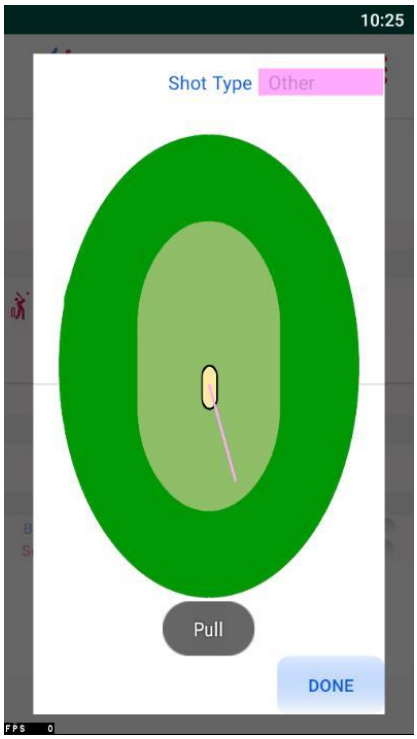

Figure 4.25 Recording Batsman Shots Page

Add New Team Page

This page allows user to add new team.

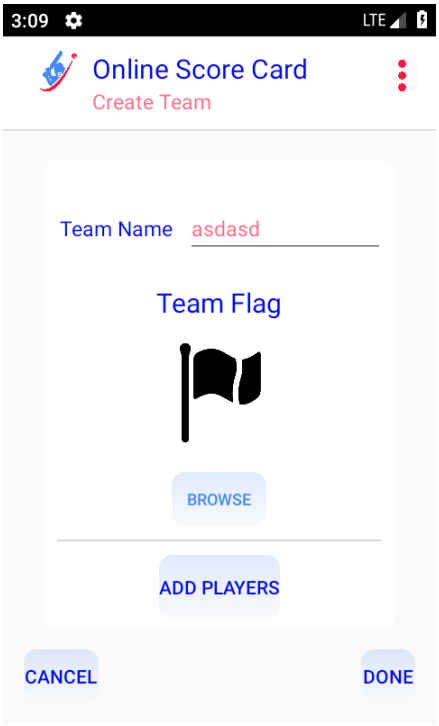

Figure 4.26 Add New Team Page

Add New Player Page

This page allows user to add new player and also assign team.

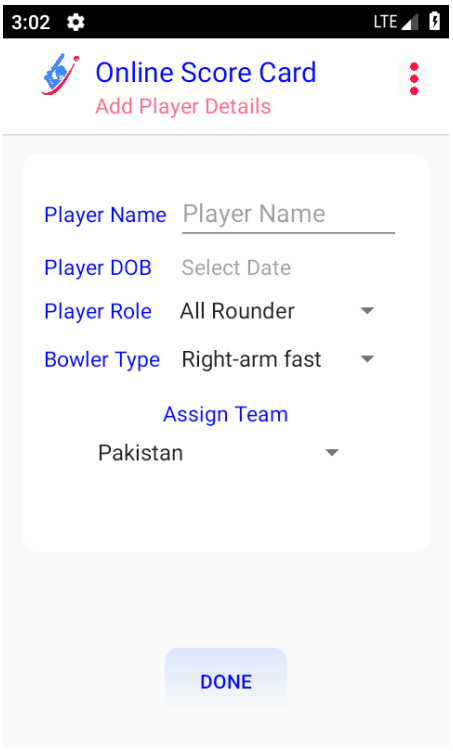

Figure 4.27 Add New Player Page

Team Detail Page

This page allows user to display Team Details.

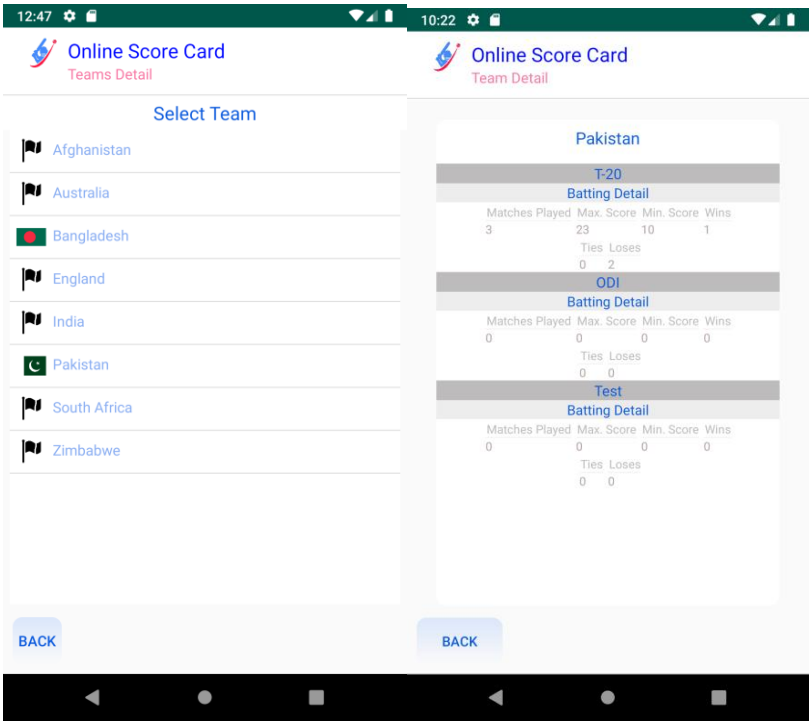

Figure 4.28, A & B, Select Team to Show Detail

Player Detail Page

This page allows user to display Player Details.

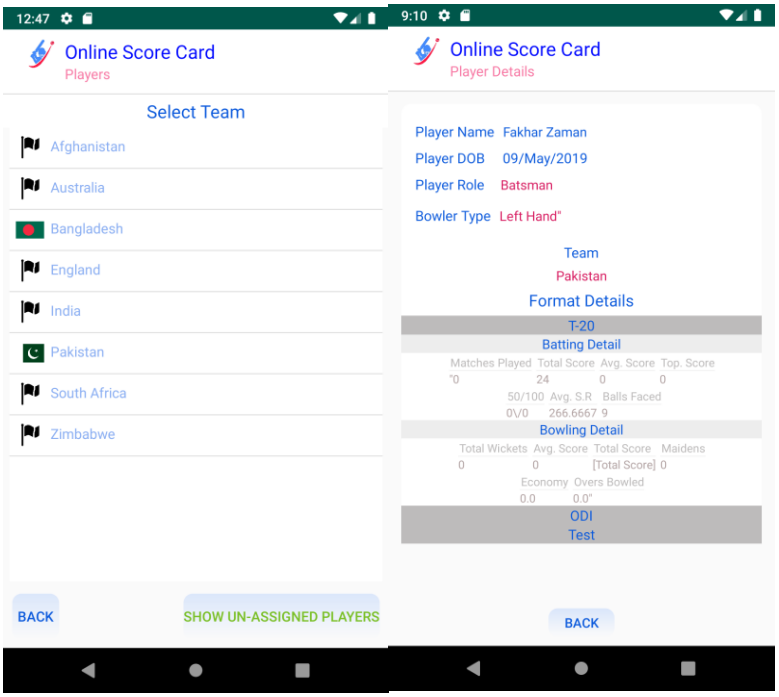

Figure 4.29, A & B Select Player to Show Detail

About Page

This page information just shows details about the project.

Figure 4.30 About Page

CHAPTER 5
CONCULSION

5.1 Concluding Remarks

Cricket Score Card With Map (Online) is successfully completed and meets the requirements and objectives. The application covers all the major modules which is use to fulfil cricket scoring requirements and also provide effective and efficient system for match scoring.

This application can be easily used and handle by any cricket literate person, its interface is carefully planned and design considering the interface design rules.

To conclude the research, it is enough that my application is intelligent and fulfills the basic requirements and modern requirements too. It is useful for cricket match scoring and its record keeping as well, as its database is on remote server.

5.2 Future Directions

In future I can update this application add some extra feature like notifications to users if a new match has started. User can also get match schedule detail on the application. This application can upload later on Play store app store so that other people or user use it.

5.3 Limitations

This software application is easy to use and can install in Android supported mobile/devices. The application is dependent on Android devices only. Users have to just install it and use it without any limitations. Cricket literate user would know how to work with this application, therefore also have knowledge of interface usage that how to use android applications.

REFERENCES

[1] Umair tasawar (2018) Cricket Score Card BSCS thesis, Barani Institute of Information Technology.

[2]https://stackoverflow.com/questions/loading-images-from-gallery-into-the-app-in-android

[3] https://stackoverflow.com/questions/how-to-get-the-date- taken- and display-it (2019) getting metadata of images (Accessed 2018, April 25).

[4] https://www.dropbox.com/developers-v1/core/start/android (2018) date-time picker. (Accessed 2019, June 15).

[5] www.stackoverflow.com (2019) Solutions of specific problems (Accessed 2019, Jul 15).

[6]https://stackoverflow.com/questions//how-to-get-the-date-taken

ANNEXURE

Annexure A
Splash Screen

```
import android.content.Intent;
import android.graphics.Typeface;
import android.os.Bundle;
import android.os.Handler;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.widget.TextView;
* Created by Faizan Ali
        */
public class splashActivity extends AppCompatActivity {
Override
protected void onCreate( Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
 TextView appname=(TextView)findViewById(R.id.name);
appname.setTypeface(Typeface.createFromAsset(getApplicationContext().getAssets(),"Jameel_Noo
ri_Nastaleeq.ttf"));
appname.setText("زاراہ");
setContentView(R.layout.splashlayout)
New Handler().postDelayed(new Runnable() {
@Override
public void run()
 Intent i=new Intent(getApplicationContext(),MainList.class);
 startActivity(i);
finish();
 },2000);
```

Annexure B
Matches

```
Intent h=getIntent();
check_lang=h.getIntExtra("selecttedlang",static_class.check);
static_class.check_lang=check_lang;
/Log.i("oncreatecheck",String.valueOf(static_class.check));
matchlist=(ListView)findViewById(R.id.matchlist);
 CustomAdapter customAdapter=new CustomAdapter();
```

```
matchlist.setAdapter(customAdapter);
matchlist.setOnItemClickListener(new AdapterView.OnItemClickListener() {
@Override
public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {
static_class.currentmatch=i+a1;
Intent d=new Intent(getApplicationContext(),matchpage.class);
if (check_lang==1) {
 static_class.matchtitle = i;
d.putExtra("matchname",static_class.matchlisteng[i] )
static_class.currentmatch=i+1
static_class.Aid = i+1;
c_sec=mydatabase.getmatch(i+1);
/jitni matche return ho rhe utne size ki array bna d
pressmatch=new String[c_sec.getCount()];
presstrans=new String[c_sec.getCount()]
pressayatno=new String[c_sec.getCount()];
pressmatchcount=new String[c_sec.getCount()]
static_class.clickmatch=new String[c_sec.getCount()]
static_class.audiopath=new String[c_sec.getCount
if(c_sec.moveToFirst(){
int d_count=0
 if (c_sec.moveToFirst()

 do {

pressmatch[d_count]=c_sec.getString(4)
presstrans[d_count]=c_sec.getString(5);
pressayatno[d_count]=c_sec.getString(7)
pressmatchcount[d_count]=c_sec.getString(10);
static_class.audiopath[d_count]=c_sec.getString(8)
static_class.shared_variable= "\n\n"+
pressmatch[d_count]+ "\n\n" + presstrans[d_count]+ "\n\n" + pressayatno[d_count];
static_class.clickmatch[d_count]=static_class.shared_variable;
d_count=d_count+1;
}while (c_sec.moveToNext())
 Log.i("numberofrecords",pressmatcheng[a1]);
 }else
{
d.putExtra("matchname",static_class.matchlisturdu[i])
c_sec=mydatabase.getmatch(i+1);
static_class.Aid = i+
```

```
//jitni matchereturnho rhe utne size ki array bna d
pressmatch=new String[c_sec.getCount()];
presstrans=new String[c_sec.getCount()];
pressayatno=new String[c_sec.getCoun
pressmatchcount=new String[c_sec.getCount
static_class.clickmatch=new String[c_sec.getCount()]
 static_class.audiopath=new String[c_sec.getCount()];
 int d_count=0
if (c_sec.moveToFirst())
{
do {
pressmatch[d_count]=c_sec.getString(4)
presstrans[d_count]=c_sec.getString(6)
pressayatno[d_count]=c_sec.getString(9)
pressmatchcount[d_count]=c_sec.getString(10);
static_class.audiopath[d_count]=c_sec.getString(8);
static_class.shared_variable= "\n\n"+
pressmatch[d_count]+ "\n\n" + presstrans[d_count]+ "\n\n" + pressayatno[d_count]
static_class.clickmatch[d_count]=static_class.shared_variable;
d_count=d_count+1;
}while (c_sec.moveToNext());
}}
startActivity(d);

                    }
                });
```

Annexure C
Create Match

```
import android.content.Context;
import android.content.Intent;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.view.View;
import android.widget.Button;
/*
* Created by Faizan Ali.
```

```
 */
public class settingsmain extends AppCompatActivity {
Button eng,urdu
int check;
Intent h;
 @Override
protected void onCreate( Bundle savedInstanceState)
super.onCreate(savedInstanceState);
setContentView(R.layout.settingslayout);
Toolbar toolbar=(Toolbar)findViewById(R.id.toolbarfosettings); setSupportActionBar(toolbar);
eng=(Button)findViewById(R.id.engsbtn)
 urdu=(Button)findViewById(R.id.urdusbtn);
 }
 public void selectedlang(View v)
{
 switch (v.getId()
{
case R.id.engsbtn
static_class.check=1;
 //static_class.sharedPreferences = this.getSharedPreferences("com.example.kashaf.zaderah",
Context.MODE_PRIVATE)
//static_class.sharedPreferences.edit().putInt("savelang",static_class.check)
 h=new Intent(getApplicationContext(),MainList.class);
 h.putExtra("selec ttedlang",static_class.check);
startActivity(h);
break
 case R.id.urdusbtn
static_class.check
 //static_class.sharedPreferences = this.getSharedPreferences("com.example.kashaf.zaderah",
Context.MODE_PRIV
  //static_class.sharedPreferences.edit().putInt("savelang",static_class.check);
 h=new Intent(getApplicationContext(),MainList.class)
 h.putExtra("selecttedlang",static_class.check)
startActivity(h)
 break;
case R.id.homesbtn
 h=new Intent(getApplicationContext(),MainList.class);
sstartActivity(h);
```

Annexure D

Toss

```
public void play_exp(View v)
 {
 if (check==1)
 {
    mediaPlayer.stop();
  }
String audiopath = "a" + static_class.audiopath[audionum].toLowerCase();
 // int i1=R.raw.a1;
int resId = getResources().getIdentifier(audiopath , "raw" , getPackageName()); if (resId != 0)
 {
 mediaPlayer = MediaPlayer.create(matchpage.this, resI
scrubber.setMax(mediaPlayer.getDuration());
new Timer().scheduleAtFixedRate(new TimerTask() {
 @Override
   public void run() {
               scrubber.setProgress(mediaPlayer.getCurrentPosition());
            }
         },0,100);
scrubber.setOnSeekBarChangeListener(new SeekBar.OnSeekBarChangeListener() {
            @Override
            public void onProgressChanged(SeekBar seekBar, int i, boolean b) {
               //  mediaPlayer.seekTo(i);

        @Override
            public void onStartTrackingTouch(SeekBar seekBar) {

            }

            @Override
            public void onStopTrackingTouch(SeekBar seekBar) {

            }
         });
         pausebtn.setBackgroundResource(R.drawable.pausez);
         pausebtn.setVisibility(View.VISIBLE);
         scrubber.setVisibility(View.VISIBLE);
         mediaPlayer.start();
         check=1;
```

```
        }
        else
        {
            Toast t=Toast.makeText(getApplicationContext(),"audio not
available",Toast.LENGTH_LONG);
            t.show();
        }

        chk_popup = 0;

    }
    public void play_audio(View v)
    {
        if (check==1)
        {
            mediaPlayer.stop();
        }
    {

        if (chk_ply_ps == 0)
        {
            mediaPlayer.pause();
            pausebtn.setBackgroundResource(R.drawable.playz);
            chk_ply_ps = 1;
        }
        else
        {
            mediaPlayer.start();
            pausebtn.setBackgroundResource(R.drawable.pausez);
            chk_ply_ps = 0;
        }
```

Annexure E
Add New Match

```
public void open_matchmenu(View v)
    {
 previousbtn.setVisibility(View.INVISIBLE);
  nextbtn.setVisibility(View.INVISIBLE);
  pausebtn.setVisibility(View.INVISIBLE);
```

```
  popup.setVisibility(View.INVISIBLE);
   addmenu.setVisibility(View.VISIBLE);
    }

    public void cancelbtn(View v)
    {
       addmenu.setVisibility(View.INVISIBLE);
       previousbtn.setVisibility(View.VISIBLE);
       nextbtn.setVisibility(View.VISIBLE);
       pausebtn.setVisibility(View.INVISIBLE);
    }
    public void add_match(View v)
    {
       if (  .getText().toString().equals("") || engtrans.getText().toString().equals("") ||
urdutrans.getText().toString().equals("")) {

           Toast.makeText(this, "All fields required", Toast.LENGTH_SHORT).show();

       }
       else
       {
         // SQLiteDatabase myDatabase =
this.openOrCreateDatabase("Zaderah2",MODE_PRIVATE,null);
           // myDatabase.execSQL("INSERT INTO Match
(Aid,titleEng,titleurdu,  ,TransEng,TransUrdu," +
                  // "Count) VALUES (12,'dfdh','fhdh','dfh','ghg','dghdgh','dghh',5)");

mydatabases.pastematch(  .getText().toString(),engtrans.getText().toString(),urdutrans.getText().toSt
ring(),recording_path.toString());

           addmenu.setVisibility(View.INVISIBLE);
           previousbtn.setVisibility(View.VISIBLE);
           nextbtn.setVisibility(View.VISIBLE);
           pausebtn.setVisibility(View.INVISIBLE);

       }
    }
       public void gobacktohome(View v) {
           static_class.shared_variable = "";
           Intent h = new Intent(getApplicationContext(), MainList.class);
           startActivity(h);
```

```
    }

    public void changematch(View v) {
       switch (v.getId()) {
          case R.id.previousbtn:
             audio.setEnabled(false);
             explanation.setEnabled(false);
             audio.setAlpha((float) 0.5);
             explanation.setAlpha((float) 0.5);
             audio.setTextColor(Color.WHITE);
             explanation.setTextColor(Color.WHITE);
             pausebtn.setVisibility(View.INVISIBLE);
             scrubber.setVisibility(View.INVISIBLE);
             if (check==1)
             {
                mediaPlayer.stop();
             }
             if (static_class.check_lang == 1 && static_class.currentmatch > 1) {
                static_class.matchtitle = static_class.matchtitle-1;
```

Annexure F

Cricket SCoring

```
matchname.setText(static_class.matchlisteng[static_class.matchtitle]);
 //ccc=mydatabases.getmatchnmae(static_class.currentmatch-a1);
c_sec = mydatabases.getmatch(static_class.currentmatch - 1);
 static_class.Aid = static_class.currentmatch - 1;
 // c_sec = mydatabases.getmatch(12);
  static_class.shared_variable = "";

                //jitni matchereturnho rhe utne size ki array bna di
                pressmatch = new String[c_sec.getCount()];
                presstrans = new String[c_sec.getCount()];
                pressayatno = new String[c_sec.getCount()];
                pressmatchcount = new String[c_sec.getCount()];
                static_class.clickmatch=new String[c_sec.getCount()];
                static_class.audiopath=new String[c_sec.getCount()];
                int d_count = 0;
                if (c_sec.moveToFirst()) {
                   do {
```

```
                    //matchname.setText(ccc.getString(2));
                    pressmatch[d_count] = c_sec.getString(4);
                    presstrans[d_count] = c_sec.getString(5);
                    pressayatno[d_count] = c_sec.getString(7);
                    pressmatchcount[d_count] = c_sec.getString(10);
                    static_class.audiopath[d_count] = c_sec.getString(8);
                    static_class.shared_variable= "\n\n"+
                          pressmatch[d_count]+ "\n\n" + presstrans[d_count]+ "\n\n" +
pressayatno[d_count];
                    static_class.clickmatch[d_count]=static_class.shared_variable;
                    d_count = d_count + 1;

                  } while (c_sec.moveToNext());
                }


                CustomAdapter customAdapter=new CustomAdapter();
                openmatch.setAdapter(customAdapter);
                static_class.currentmatch -= 1;
                check_fvrt();

              } else if (static_class.check_lang != 1 && static_class.currentmatch > 1) {
                static_class.matchtitle = static_class.matchtitle-1;
                matchname.setText(static_class.matchlisturdu[static_class.matchtitle]);
                // ccc=mydatabases.getmatchnmae(static_class.currentmatch-a1);
                c_sec = mydatabases.getmatch(static_class.currentmatch - 1);
                static_class.Aid = static_class.currentmatch - 1;
                static_class.shared_variable = "";

                //jitni matchereturnho rhe utne size ki array bna di
                pressmatch = new String[c_sec.getCount()];
                presstrans = new String[c_sec.getCount()];
                pressayatno = new String[c_sec.getCount()];
                pressmatchcount = new String[c_sec.getCount()];
                static_class.clickmatch=new String[c_sec.getCount()];
                static_class.audiopath=new String[c_sec.getCount()];
                int d_count = 0;
                if (c_sec.moveToFirst()) {
                  do {

                    // matchname.setText(ccc.getString(3));
```

```
                    pressmatch[d_count] = c_sec.getString(4);
                    presstrans[d_count] = c_sec.getString(6);
                    pressayatno[d_count] = c_sec.getString(9);
                    pressmatchcount[d_count] = c_sec.getString(10);
                    static_class.audiopath[d_count] = c_sec.getString(8);
                    static_class.shared_variable= "\n\n"+
                          pressmatch[d_count]+ "\n\n" + presstrans[d_count]+ "\n\n" +
pressayatno[d_count];
                    static_class.clickmatch[d_count]=static_class.shared_variable;
                    d_count = d_count + 1;

                } while (c_sec.moveToNext());
            }
```

Annexure G
Add player

```
import android.content.Intent;
import android.database.Cursor;
import android.database.SQLException;
import android.graphics.Typeface;
import android.os.Bundle;
import android.support.annotation.Nullable;
import android.support.v7.app.AppCompatActivity;
import android.support.v7.widget.Toolbar;
import android.util.Log;
import android.view.View;
import android.view.ViewGroup;
import android.widget.AdapterView;
import android.widget.BaseAdapter;
import android.widget.ImageView;
import android.widget.ListView;
import android.widget.TextView;

import java.io.IOException;

/**
 * Created by S. M. Ali Fraz Bukhari.
 */

public class fvrt extends AppCompatActivity {
```

```
Cursor c_sec;
ListView fvrts;
String[] defaultfvrt = {"no favourite"};
DBHandler mydatabase;
int id;
String[] pressmatch;
String[] presstrans;
String[] pressayatno;
String[] pressmatchcount;

@Override
protected void onCreate(@Nullable Bundle savedInstanceState) {
   super.onCreate(savedInstanceState);
   setContentView(R.layout.fvrt_layout);

   Toolbar toolbar=(Toolbar)findViewById(R.id.toolbarforfvrt);
   setSupportActionBar(toolbar);

   mydatabase=new DBHandler(fvrt.this,"",null,1);
   try {
      mydatabase.createDatabase();
   } catch (IOException e) {
      throw new Error("unable to create database");
   }

   try {
      mydatabase.openDataBase();
   }catch (SQLException sqle)
   {
      throw new Error("unable to open database");
   }

   fvrts = (ListView)findViewById(R.id.fvrtmatchlist);

   Cursor c = mydatabase.getfvrtid();
   int count=c.getCount();
   static_class.fvrtids = new int[count];

   int d_count=0;
   if (c.moveToFirst())
```

```
{
    do {
        static_class.fvrtids[d_count] = c.getInt(1);
        d_count = d_count + 1;
    }while (c.moveToNext());
}

static_class.fvrtlist = new String[static_class.fvrtids.length];
static_class.urdufvrtlist = new String[static_class.fvrtids.length];
Log.i("fvrt ids","------------------kashaf--------------");
// Log.i("fvrt ids",String.valueOf(static_class.fvrtids[1]));


for (int j=0;j<static_class.fvrtids.length;j++)
{
    c=mydatabase.getfvrttitle(static_class.fvrtids[j]);
    //int d_cunt=0;
    if (c.moveToFirst())
    {
        do {
            static_class.fvrtlist[j] = c.getString(1);
            static_class.urdufvrtlist[j] = c.getString(2);
            //d_count = d_count + 1;
        }while (c.moveToNext());
    }
}

CustomAdapter customAdapter = new CustomAdapter();
fvrts.setAdapter(customAdapter);

fvrts.setOnItemClickListener(new AdapterView.OnItemClickListener() {
    @Override
    public void onItemClick(AdapterView<?> adapterView, View view, int i, long l) {

        Intent d=new Intent(getApplicationContext(),fvrtmatchpage.class);
        if (static_class.check_lang==1) {
            static_class.fvrtmatchtitle = i;
            d.putExtra("matchname",static_class.fvrtlist[i] );
            static_class.currentmatch=i;
            c_sec=mydatabase.getmatch(static_class.fvrtids[i]);
```

```
            //jitni matche return ho rhe utne size ki array bna di
            pressmatch=new String[c_sec.getCount()];
            presstrans=new String[c_sec.getCount()];
            pressayatno=new String[c_sec.getCount()];
            pressmatchcount=new String[c_sec.getCount()];
            static_class.clickmatch=new String[c_sec.getCount()];
            static_class.audiopath=new String[c_sec.getCount()];
            int d_count=0;
            if (c_sec.moveToFirst())
            {
               do {

                  pressmatch[d_count]=c_sec.getString(4);
                  presstrans[d_count]=c_sec.getString(5);
                  pressayatno[d_count]=c_sec.getString(7);
                  pressmatchcount[d_count]=c_sec.getString(10);
                  static_class.audiopath[d_count]=c_sec.getString(8);
                  static_class.shared_variable= "\n\n"+
                        pressmatch[d_count]+ "\n\n" + presstrans[d_count]+ "\n\n" +
pressayatno[d_count];
                  static_class.clickmatch[d_count]=static_class.shared_variable;

                  d_count=d_count+1;

               }while (c_sec.moveToNext());
            }

            // Log.i("numberofrecords",pressmatcheng[a1]);


         }else
         {
            d.putExtra("matchname",static_class.urdufvrtlist[i]);
            c_sec=mydatabase.getmatch(static_class.fvrtids[i]);

            //jitni matchereturnho rhe utne size ki array bna di
            pressmatch=new String[c_sec.getCount()];
            presstrans=new String[c_sec.getCount()];
            pressayatno=new String[c_sec.getCount()];
            pressmatchcount=new String[c_sec.getCount()];
```

```
            static_class.clickmatch=new String[c_sec.getCount()];
            static_class.audiopath=new String[c_sec.getCount()];
            int d_count=0;
            if (c_sec.moveToFirst())
            {
               do {

                  pressmatch[d_count]=c_sec.getString(4);
                  presstrans[d_count]=c_sec.getString(6);
                  pressayatno[d_count]=c_sec.getString(9);
                  pressmatchcount[d_count]=c_sec.getString(10);
                  static_class.audiopath[d_count]=c_sec.getString(8);
                  static_class.shared_variable= "\n\n"+
                        pressmatch[d_count]+ "\n\n" + presstrans[d_count]+ "\n\n" +
pressayatno[d_count];
                  static_class.clickmatch[d_count]=static_class.shared_variable;

                  d_count=d_count+1;

               }while (c_sec.moveToNext());
            }}
         startActivity(d);

      }
   });

}



public void deletefvrt(View v)
{
   Log.i("idm pressed",String.valueOf(id));

}

public void goback_home(View v)
{
   Intent i=new Intent(getApplicationContext(),MainList.class);
   startActivity(i);
}
```

```
class CustomAdapter extends BaseAdapter {
    @Override
    public int getCount()
    {
        if (static_class.fvrtlist.length > 0) {
            return static_class.fvrtlist.length;
        }
        else
        {
            return defaultfvrt.length;
        }

    }

    @Override
    public Object getItem(int i) {
        return null;
    }
```

Annexure H
Add Team

```
public void reorder(int xaid,int yaid,String xeng,String yeng,String xurdu,String yurdu)
{
SQLiteDatabase db = this.getReadableDatabase();
db.execSQL("update Activity set EnglishTitle = '"+yeng+"',UrduTitle = '"+yurdu+"' where Aid =
'"+xaid+"' ");
db.execSQL("update Activity set EnglishTitle = '"+xeng+"',UrduTitle = '"+xurdu+"' where Aid =
'"+yaid+"' ");
db.execSQL("update Match set Aid = 000 where Aid = '"+xaid+"'");
db.execSQL("update Match set Aid = '"+xaid+"' where Aid = '"+yaid+"'");
db.execSQL("update Match set Aid = '"+yaid+"' where Aid = 000");
}
```