PROJECT REPORT

On

“SCORZZ”

Department of Computer Engineering & Applications

GLA UNIVERSITY

GLA University
Mathura- 281406, INDIA
2022-2023

SUBMITTED BY:-

Aman Kumar Singh (201500075)
Avinash Tiwari (201500186)
Prince Kumar Sharma (201500521)

SUBMITTED TO:-

Mr. FarmanUl Haque
(Technical Trainer)

Declaration

We hereby declare that the work which is being presented in the Android Cricket Score Checker Project “**SCORZZ”,** in partial fulfillment of the requirements for Scorzz Project viva voce, is an authentic record of our ownwork carried by the team members under the supervision of our mentor Mr. FarmanUl Haque.

Group Members:

Aman Kumar Singh (201500075)
Avinash Tiwari (201500186)
Prince Kumar Sharma (201500205)

Course: B.Tech (Computer Science and Engineering)

Year: 3rd

Semester: 5th

Supervised By
Mr. FarmanUl Haque, Technical Trainer,

GLA University, Department of Computer Engineering & Application

Department of computer Engineering and Applications

GLA University, Mathura

17 km. Stone NH#2, Mathura-Delhi Road, P.O. – Chaumuha,
Mathura – 281406

Certificate

This is to certify that the above statements made by the candidates are correct to the best of my/our knowledge and belief.

____________________ Supervisor

Mr. FarmanUl Haque

Technical Trainer

Dept of CEA, GLA University

____

Project Mentor

(Mr. FarmanUl Haque)

____

Program Coordinator

(Mr. Shashi Sekhar)

About the Project

Cricket Score Checker App "Scorzz" is an Android based Application that works over a stable Internet Connection. This app is basically for the purpose of checking scores of any cricket match happening around the globe at a competitive level. The main aim of this project is to provide users a purely authentic application to look for cricket scores.

The idea behind this application was to make it user friendly to the users of all ages, even a small kid. Unlike most of the sports application that provide users with fantasy cricket (which has its drawbacks), this app is purely dedicated to check the scores and details of matches only.

API key is used to get the live data of matches and the data is updated time and again. A stable Internet connection is must to use this application.

The main objectives of the Scorzz include:

- Checking the summary of Cricket matches.
- Checking the player list.
- Checking the batting card of both innings of a match.
- Checking the bowling card of both innings of a match.
- Checking the list of upcoming Cricket Series.

Motivation

The “SCORZZ” app has been developed to make the users able to check the score of cricket matches and to make this app suitable for people of all ages. The most of the android applications these days that are used for checking the scores of Sports Matches are including Fantasy games and betting and thus as a result making the app inappropriate for minors. So, we decided to make a mobile application fully dedicated to check the score of Cricket matches and to get the detail information about the matches happening across the globe.

Requirements

a). Software Requirements:

- Technology Implemented: Android App Development
- Languages/Technologies Used: JAVA, XML,JSON
- IDE Used: Android Studio
- Android Virtual Device

- GitHub: GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub Repository: A GitHub repository can be used to store a development project. It can contain folders and any type of files (HTML, CSS, JavaScript, Documents, Data, Images). A GitHub repository should also include a license file and a README file about the project. A GitHub repository can also be used to store ideas, or any resources that you want to share.

- Android Studio : Android Studio provides extensive tools to help you test your Android apps with JUnit 4 and functional UI test frameworks. With Espresso Test Recorder, you can generate UI test code by recording your interactions with the app on a device or emulator. You can run your tests on a device, an emulator, a continuous integration environment, or in Firebase Test Lab.

b). Hardware Requirements:

- Processor Required: Intel i3
- Operating System: Windows 10
- RAM: 4GB
- Hardware Devices: Computer System
- Hard Disk: 256GB

Acknowledgement

The success and final outcome of this project required a lot of guidance and assistance from many people and we are extremely privileged to have got this all along the completion of my project. All that we have done is only due to such supervision and assistance and we would not forget to thank them.

We respect and thank **Mr. FarmanUl Haque** for providing us the opportunity to do the project work andgiving us all support and guidance, which made us complete the project duly. We are extremely thankful to him for providing such a nice support and guidance, although he had a busy schedule managing the corporate affairs. We owe our deep gratitude to our project guide **Mr. FarmanUl Haque** who took keen interest in our project work and guided us all along, till the completion of our projectwork by providing all the necessary information for developing a good project. After doing this project we can confidently say that this experience has not only enriched me with technical knowledge but also has unparsed the maturity of thought and vision. The attributes required in being a successful professional.

Aman Kumar Singh (201500075)

Avinash Tiwari (201500173)

Prince Kumar (201500521)

Abstract

The "Scorzz" is developed to provide users with a fully authentic application dedicated to check the score of Cricket Matches only. This application is developed with keeping in mind the users of small age also. This application helps you to provide as much facility as it can. No formal knowledge is needed for the user to use this application. The user only needs a device and a stable internet connection.

Contents

Introduction

1.1 Overview

This system was intended to develop an Android application to perform functionalities like checking the scores of cricket matches happening across the globe, getting the batting card and bowling card of both the innings of a match.

API key provides the live online data which is used in form of JSON array and data. A stable Internet Connection is a must to use this application.

1.2 Objective

Scorzz is an android application for cricket lovers to get the score of any cricket match happening around the globe. Scorzz helps user to see the detailed batting card and bowling card of both the innings of a match and to check the list of players of both the teams for that particular match.

1.3 Organization of the Project

- In the next chapter the technology review is defined which includes terms like Android Studio, Java, JSON.
- In the next chapter we deal with the software design. It includes the various data flowdiagrams, use case diagrams etc.
- The following chapter implementation which includes all the code snippets.
- The next chapter deals with the output screens.
- Final chapter has the conclusion part.

TECHNOLOGIES USED

Android Studio

Android software development is the process by which applications are created for devices running the Android operating system. Google states that "Android apps can be written using Kotlin, Java, and C++ languages" using the Android software development kit (SDK), while using other languages is also possible. All non-Java virtual machine (JVM) languages, such as Go, JavaScript, C, C++ or assembly, need the help of JVM language code, that may be supplied by tools, likely with restricted API support. Some programming languages and tools allow cross-platform app support (i.e. for both Android and iOS). Third party tools, development environments, and language support have also continued to evolve and expand since the initial SDK was released in 2008. The official Android app distribution mechanism to end users is Google Play; it also allows staged gradual app release, as well as distribution of pre-release app versions to testers.

Features of Android Studio:

- Gradle-based build support.
- Android-specific refactoring and quick fixes.
- Built-in support for Google Cloud Platform, enabling integration with Firebase Cloud Messaging and Google App Engine.
- Android Virtual Device (Emulator) to run and debug apps in the Android studio.

Introduction to Java

Java is a widely used object-oriented programming language and software platform that runs on billions of devices, including notebook computers, mobile devices, gaming consoles, medical devices and many others. The rules and syntax of Java are based on the C and C++ languages. One major advantage of developing software with Java is its portability. Once you have written code for a Java program on a notebook computer, it is very easy to move the code to a mobile device. When the language was invented in 1991 by James Gosling of Sun Microsystems (later acquired by Oracle), the primary goal was to be able to "write once, run anywhere." It's also important to understand that Java is much different from JavaScript. Javascript does not need to be compiled, while Java code does need to be compiled. Also, Javascript only runs on web browsers while Java can be run anywhere.

Introduction to API keys

An API key or application programming interface key is a code that gets passed in by computer applications. The program or application then calls the API or application programming interface to identify its user, developer or calling program to a website.

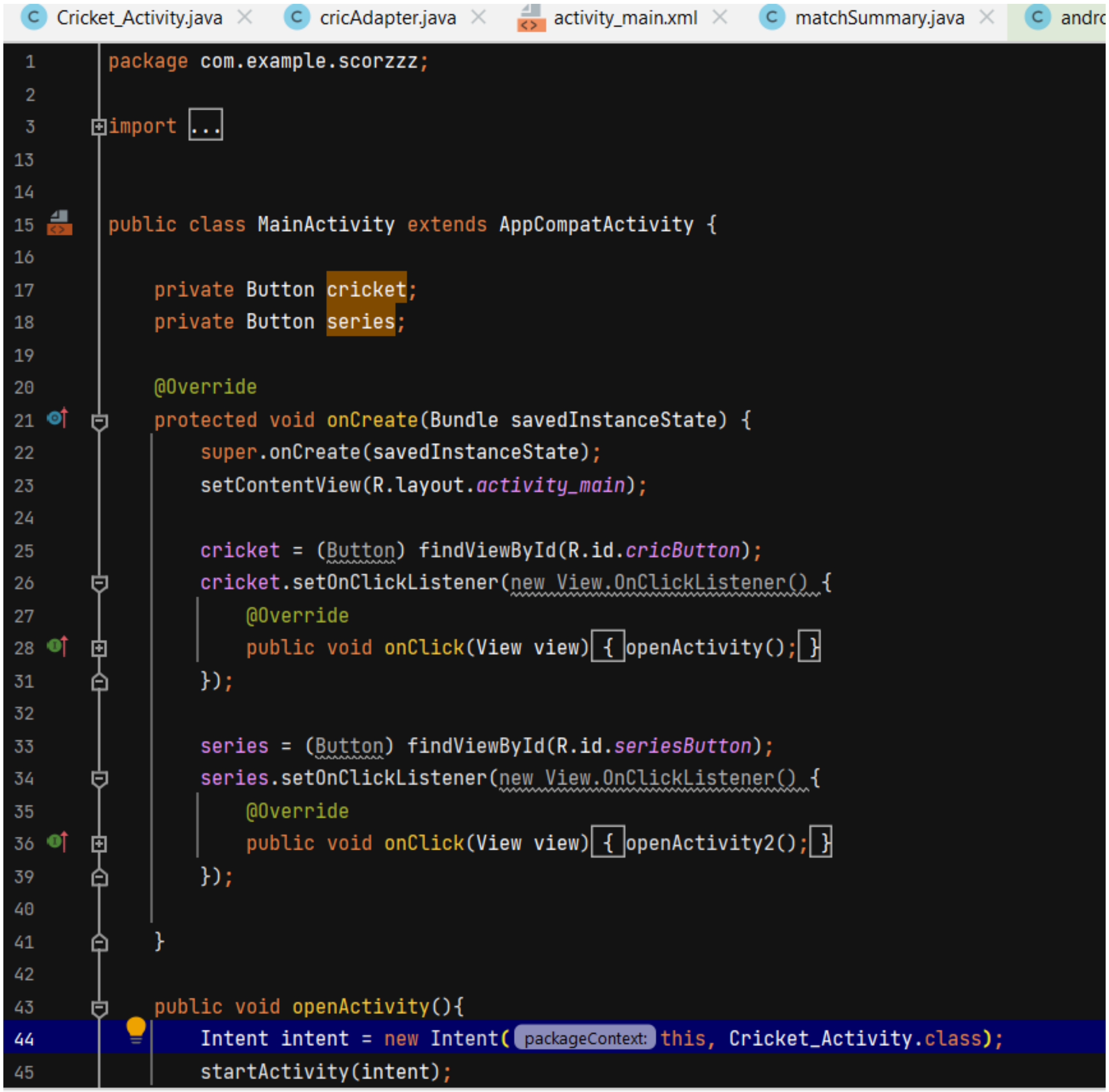
An API key can acts as a secret authentication token as well as a unique identifier. Typically, the key will come with a set of access rights for the api that is associated with it.

API keys are used with projects, while authentication is designed for the users. Cloud Endpoints will, in many cases, handle both the authentication procedures as well as the API keys. The differentiating factor between the two is:

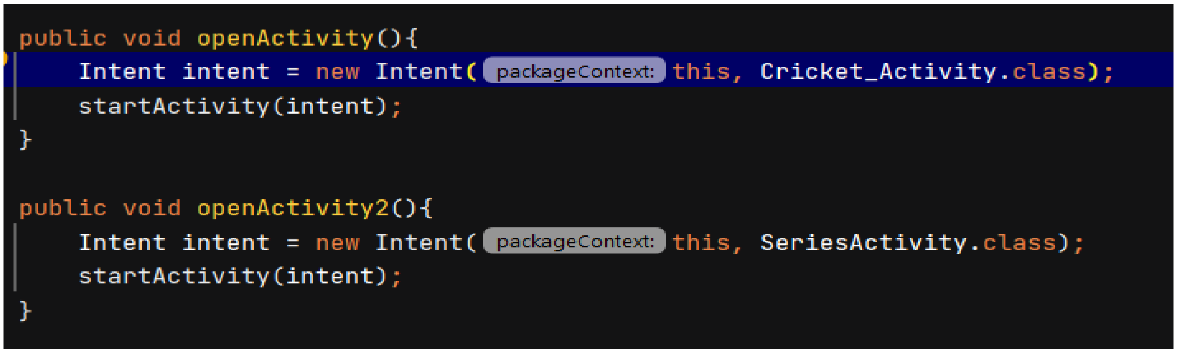
- Authentication tokens are used to identify the user.
- API keys are used to identifying the project making the call. This can either be the website or the application that is making the call to application programming interface.

PROJECT CODE SNIPPETS

❖ Main Activity

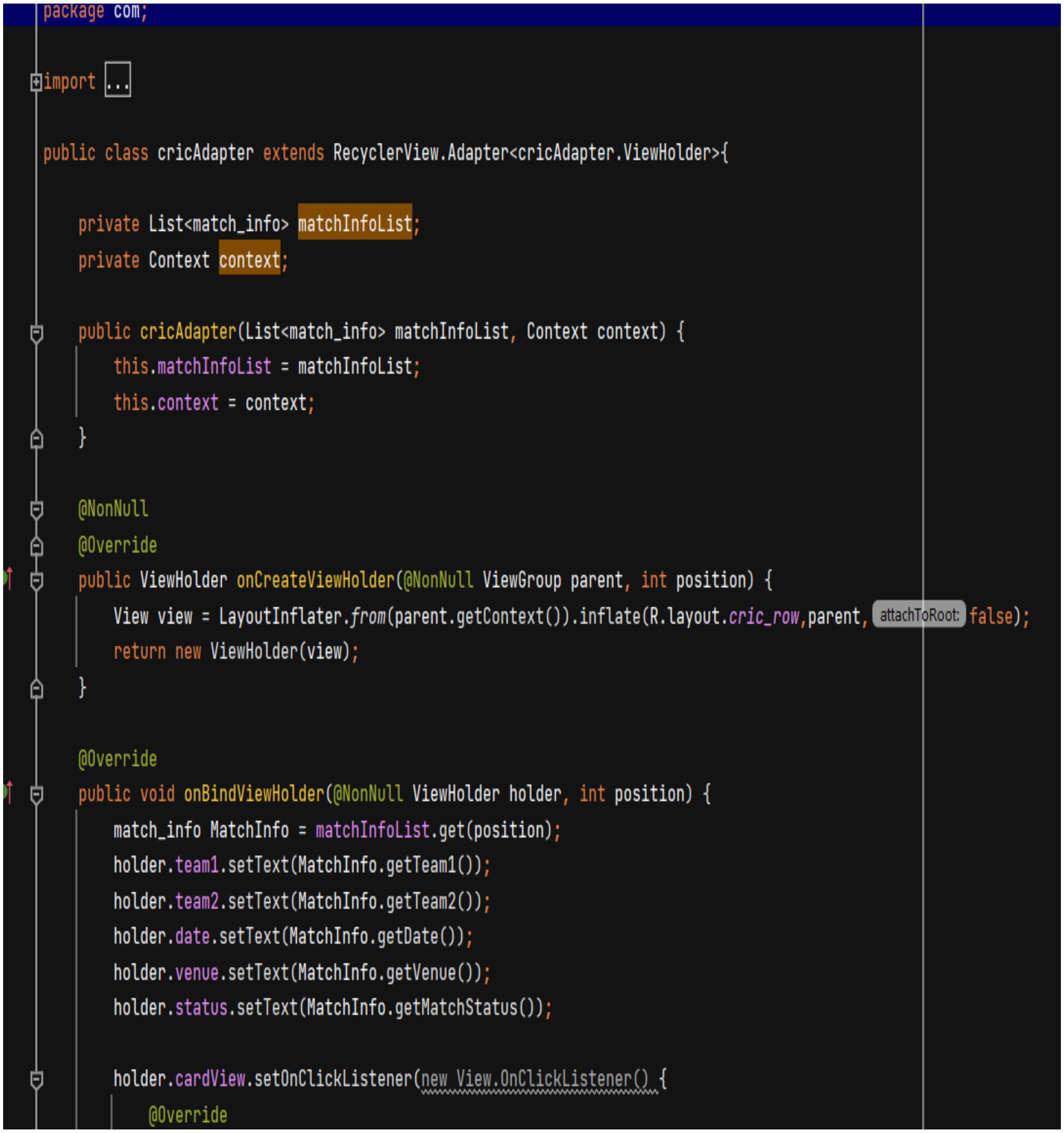


```
package com.example.scorzzz;

import ...


public class MainActivity extends AppCompatActivity {

    private Button cricket;
    private Button series;

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

        cricket = (Button) findViewById(R.id.cricButton);
        cricket.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { openActivity(); }
        });

        series = (Button) findViewById(R.id.seriesButton);
        series.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { openActivity2(); }
        });

    }

    public void openActivity(){
        Intent intent = new Intent( packageContext: this, Cricket_Activity.class);
        startActivity(intent);
```

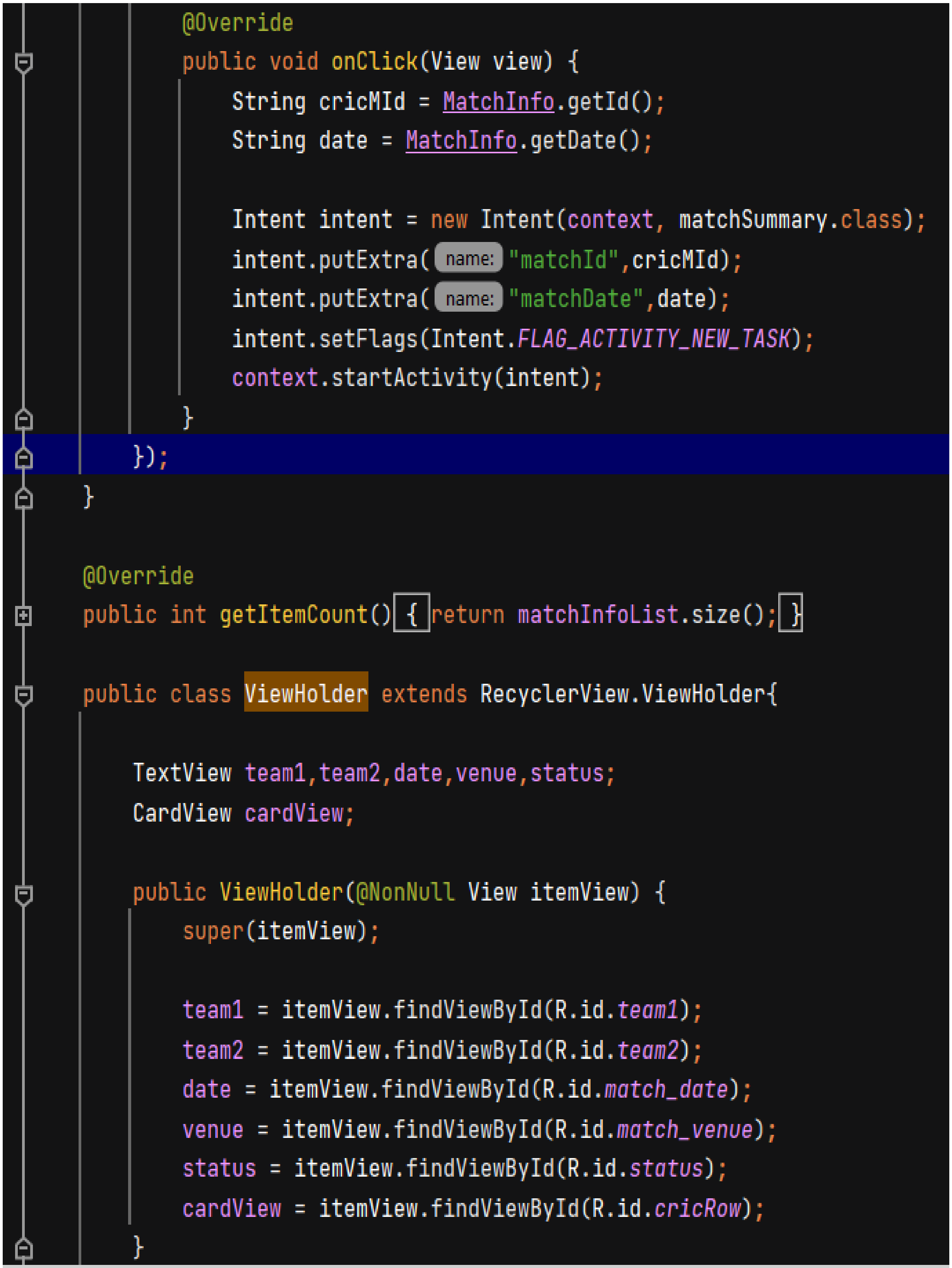


```
public void openActivity(){
    Intent intent = new Intent( packageContext: this, Cricket_Activity.class);
    startActivity(intent);
}

public void openActivity2(){
    Intent intent = new Intent( packageContext: this, SeriesActivity.class);
    startActivity(intent);
}
```

❖ cricAdapter

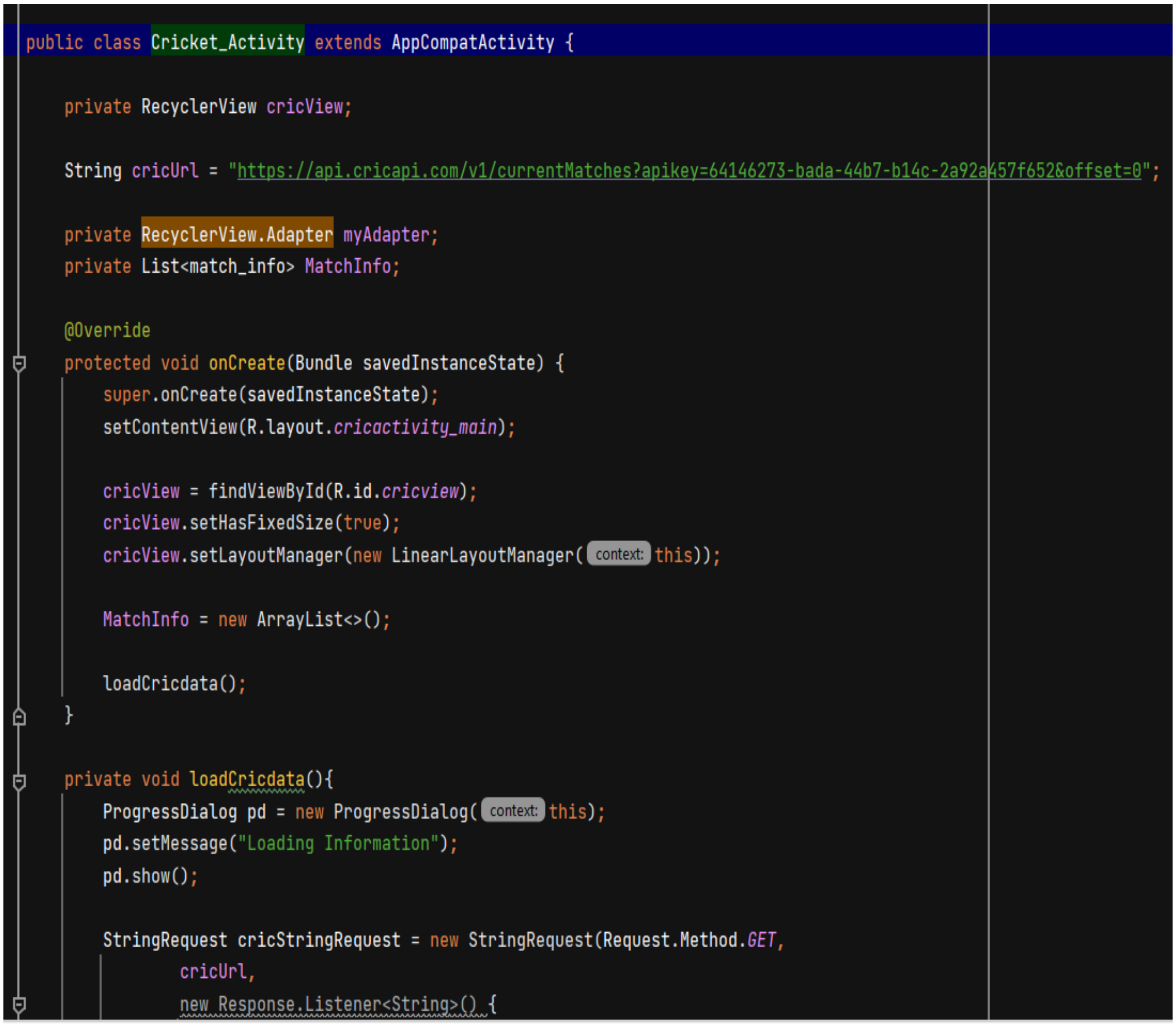


```
package com;

import ...

public class cricAdapter extends RecyclerView.Adapter<cricAdapter.ViewHolder>{

    private List<match_info> matchInfoList;
    private Context context;

    public cricAdapter(List<match_info> matchInfoList, Context context) {
        this.matchInfoList = matchInfoList;
        this.context = context;
    }

    @NonNull
    @Override
    public ViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int position) {
        View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.cric_row,parent, attachToRoot: false);
        return new ViewHolder(view);
    }

    @Override
    public void onBindViewHolder(@NonNull ViewHolder holder, int position) {
        match_info MatchInfo = matchInfoList.get(position);
        holder.team1.setText(MatchInfo.getTeam1());
        holder.team2.setText(MatchInfo.getTeam2());
        holder.date.setText(MatchInfo.getDate());
        holder.venue.setText(MatchInfo.getVenue());
        holder.status.setText(MatchInfo.getMatchStatus());

        holder.cardView.setOnClickListener(new View.OnClickListener() {
            @Override
```

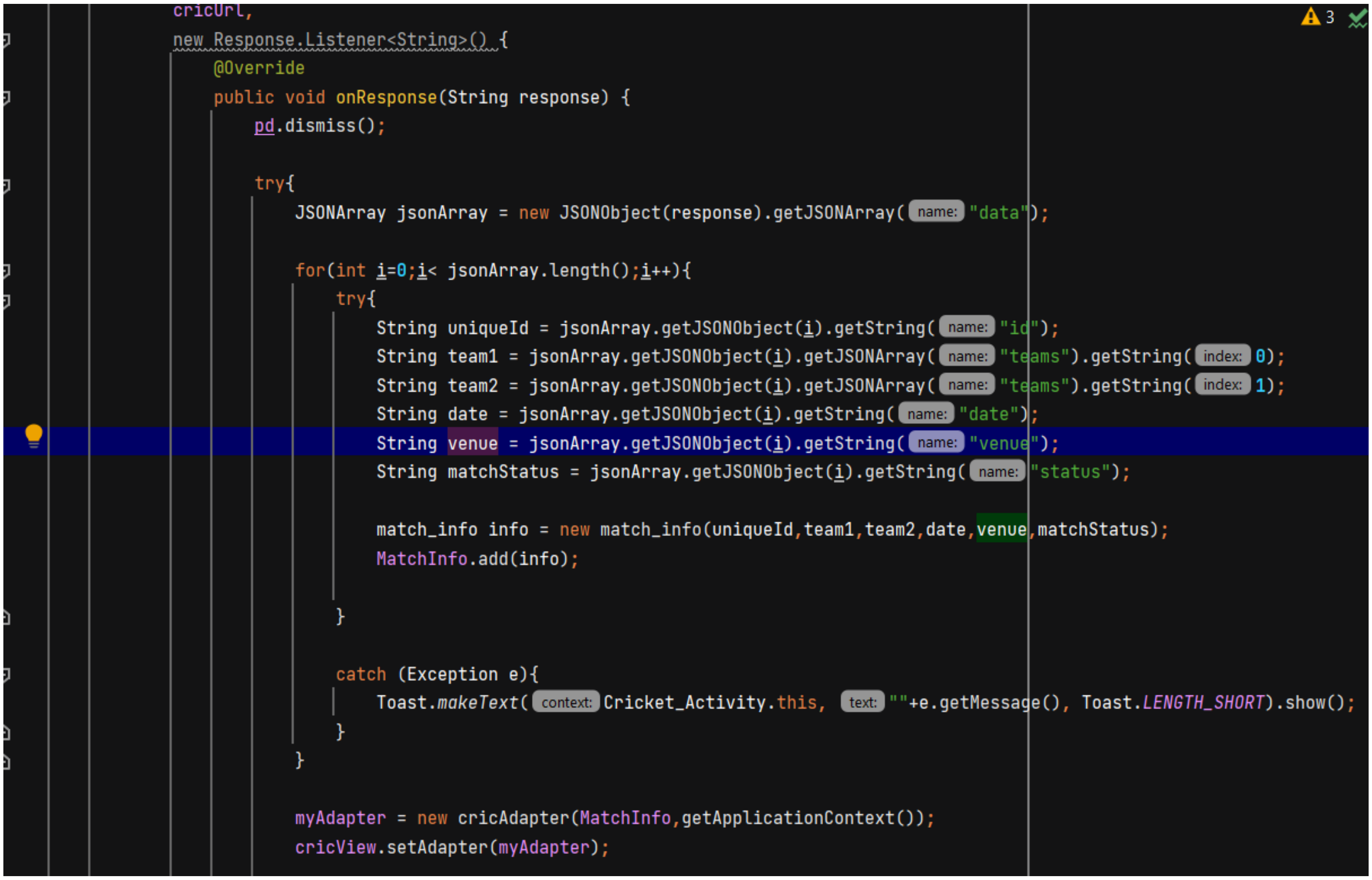


```
            @Override
            public void onClick(View view) {
                String cricMId = MatchInfo.getId();
                String date = MatchInfo.getDate();

                Intent intent = new Intent(context, matchSummary.class);
                intent.putExtra( name: "matchId",cricMId);
                intent.putExtra( name: "matchDate",date);
                intent.setFlags(Intent.FLAG_ACTIVITY_NEW_TASK);
                context.startActivity(intent);
            }
        });
    }

    @Override
    public int getItemCount() { return matchInfoList.size(); }

    public class ViewHolder extends RecyclerView.ViewHolder{

        TextView team1,team2,date,venue,status;
        CardView cardView;

        public ViewHolder(@NonNull View itemView) {
            super(itemView);

            team1 = itemView.findViewById(R.id.team1);
            team2 = itemView.findViewById(R.id.team2);
            date = itemView.findViewById(R.id.match_date);
            venue = itemView.findViewById(R.id.match_venue);
            status = itemView.findViewById(R.id.status);
            cardView = itemView.findViewById(R.id.cricRow);
        }
```

❖ Cricket Activity

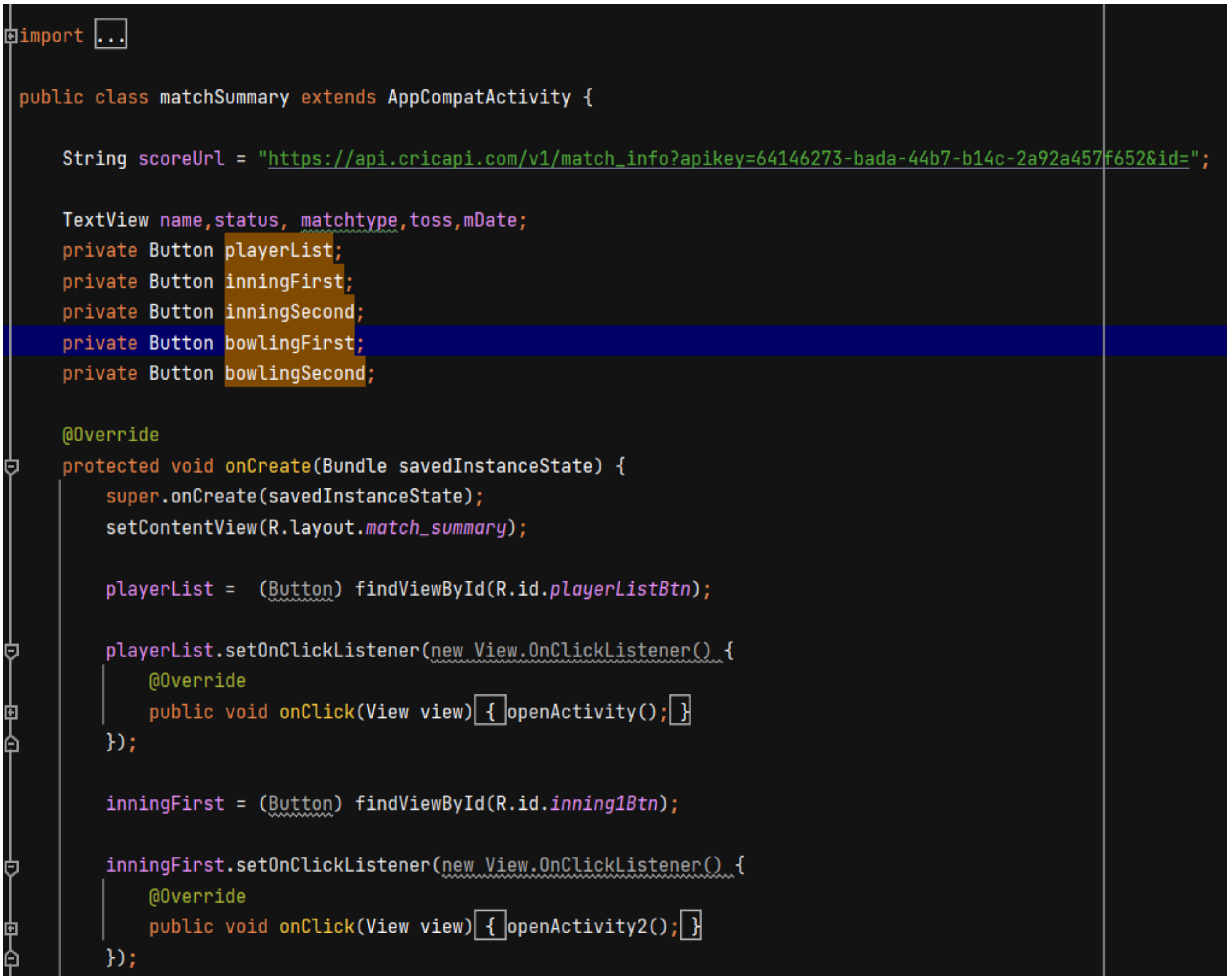


```java
public class Cricket_Activity extends AppCompatActivity {

    private RecyclerView cricView;

    String cricUrl = "https://api.cricapi.com/v1/currentMatches?apikey=64146273-bada-44b7-b14c-2a92a457f652&offset=0";

    private RecyclerView.Adapter myAdapter;
    private List<match_info> MatchInfo;

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

        cricView = findViewById(R.id.cricview);
        cricView.setHasFixedSize(true);
        cricView.setLayoutManager(new LinearLayoutManager( context: this));

        MatchInfo = new ArrayList<>();

        loadCricdata();
    }

    private void loadCricdata(){
        ProgressDialog pd = new ProgressDialog( context: this);
        pd.setMessage("Loading Information");
        pd.show();

        StringRequest cricStringRequest = new StringRequest(Request.Method.GET,
                cricUrl,
                new Response.Listener<String>() {
```

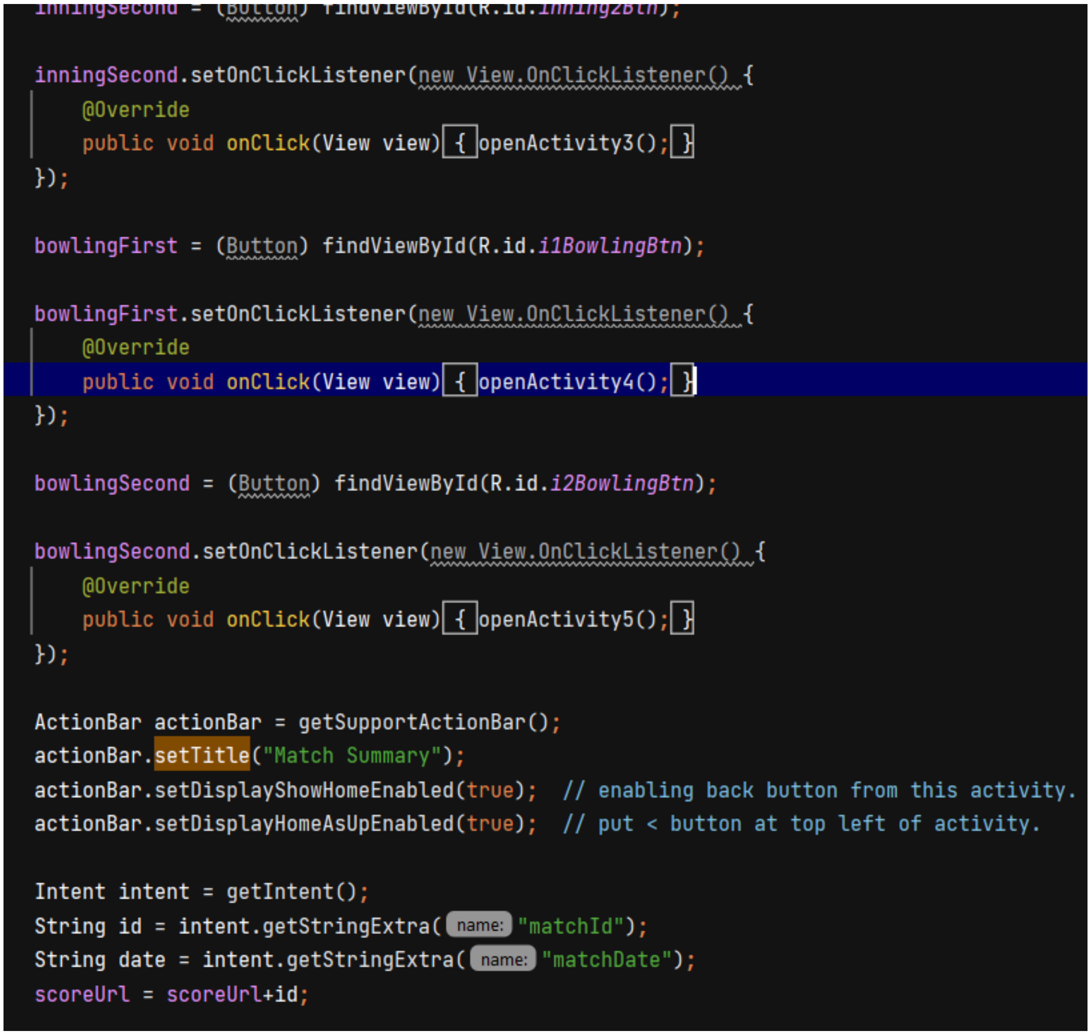


```
        cricUrl,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                pd.dismiss();

                try{
                    JSONArray jsonArray = new JSONObject(response).getJSONArray( name: "data");

                    for(int i=0;i< jsonArray.length();i++){
                        try{
                            String uniqueId = jsonArray.getJSONObject(i).getString( name: "id");
                            String team1 = jsonArray.getJSONObject(i).getJSONArray( name: "teams").getString( index: 0);
                            String team2 = jsonArray.getJSONObject(i).getJSONArray( name: "teams").getString( index: 1);
                            String date = jsonArray.getJSONObject(i).getString( name: "date");
                            String venue = jsonArray.getJSONObject(i).getString( name: "venue");
                            String matchStatus = jsonArray.getJSONObject(i).getString( name: "status");

                            match_info info = new match_info(uniqueId,team1,team2,date,venue,matchStatus);
                            MatchInfo.add(info);

                        }

                        catch (Exception e){
                            Toast.makeText( context: Cricket_Activity.this,  text: ""+e.getMessage(), Toast.LENGTH_SHORT).show();
                        }
                    }

                    myAdapter = new cricAdapter(MatchInfo,getApplicationContext());
                    cricView.setAdapter(myAdapter);
```

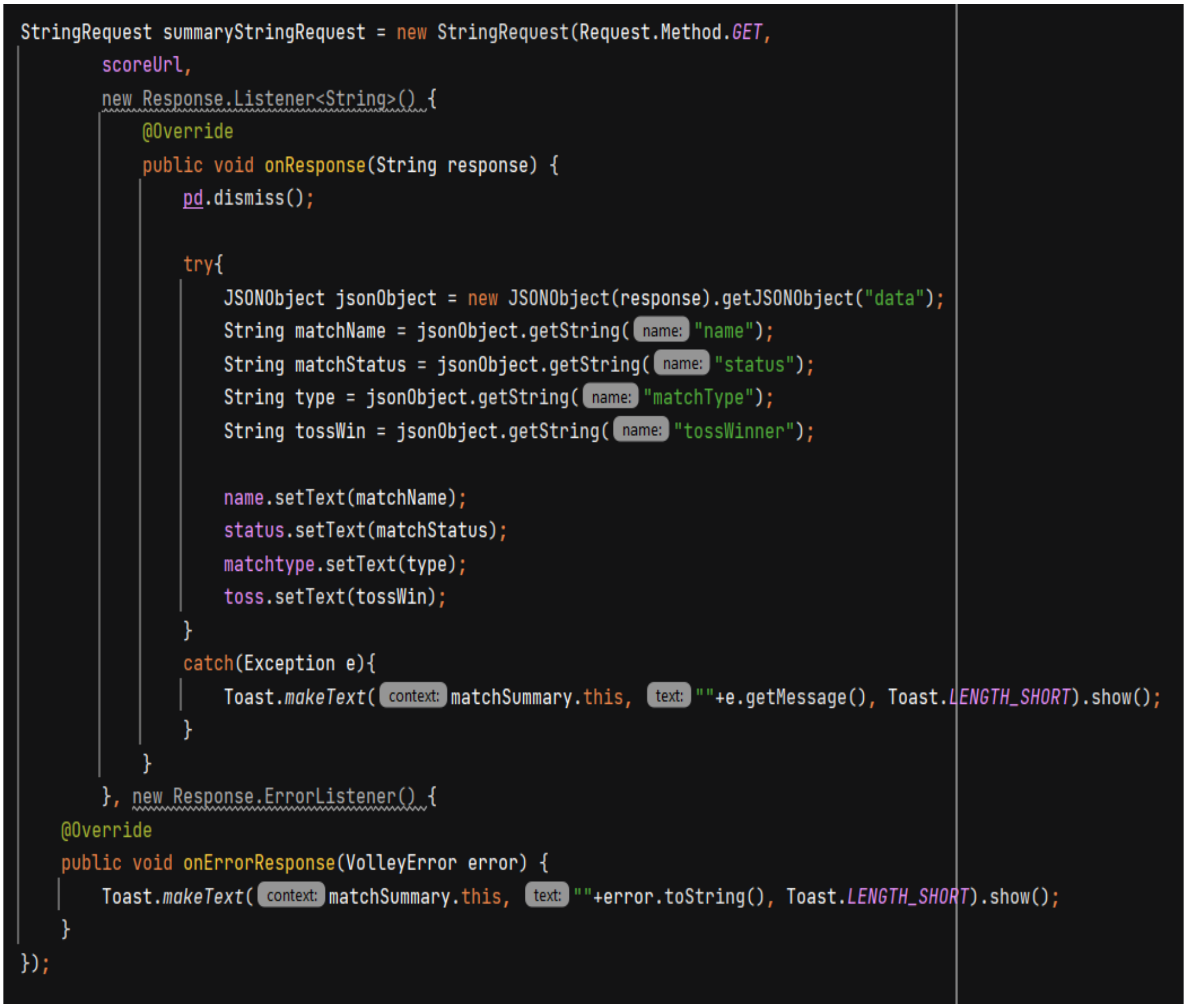
❖ Match Summary



```
import ...

public class matchSummary extends AppCompatActivity {

    String scoreUrl = "https://api.cricapi.com/v1/match_info?apikey=64146273-bada-44b7-b14c-2a92a457f652&id=";

    TextView name,status, matchtype,toss,mDate;
    private Button playerList;
    private Button inningFirst;
    private Button inningSecond;
    private Button bowlingFirst;
    private Button bowlingSecond;

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

        playerList =  (Button) findViewById(R.id.playerListBtn);

        playerList.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { openActivity(); }
        });

        inningFirst = (Button) findViewById(R.id.inning1Btn);

        inningFirst.setOnClickListener(new View.OnClickListener() {
            @Override
            public void onClick(View view) { openActivity2(); }
        });
```

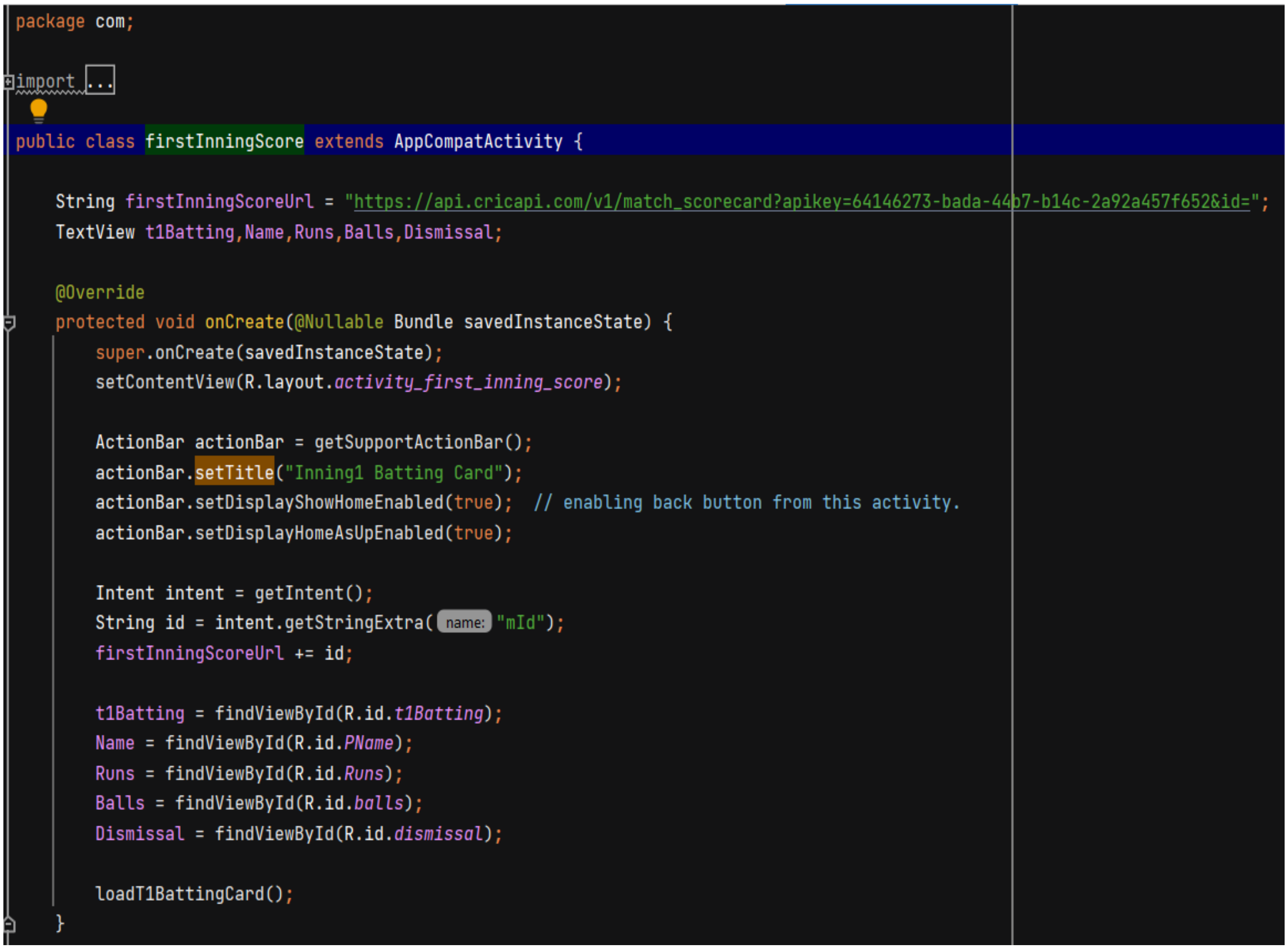


```
inningSecond = (Button) findViewById(R.id.inning2Btn);

inningSecond.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { openActivity3(); }
});

bowlingFirst = (Button) findViewById(R.id.i1BowlingBtn);

bowlingFirst.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { openActivity4(); }
});

bowlingSecond = (Button) findViewById(R.id.i2BowlingBtn);

bowlingSecond.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) { openActivity5(); }
});

ActionBar actionBar = getSupportActionBar();
actionBar.setTitle("Match Summary");
actionBar.setDisplayShowHomeEnabled(true);  // enabling back button from this activity.
actionBar.setDisplayHomeAsUpEnabled(true);  // put < button at top left of activity.

Intent intent = getIntent();
String id = intent.getStringExtra( name: "matchId");
String date = intent.getStringExtra( name: "matchDate");
scoreUrl = scoreUrl+id;
```

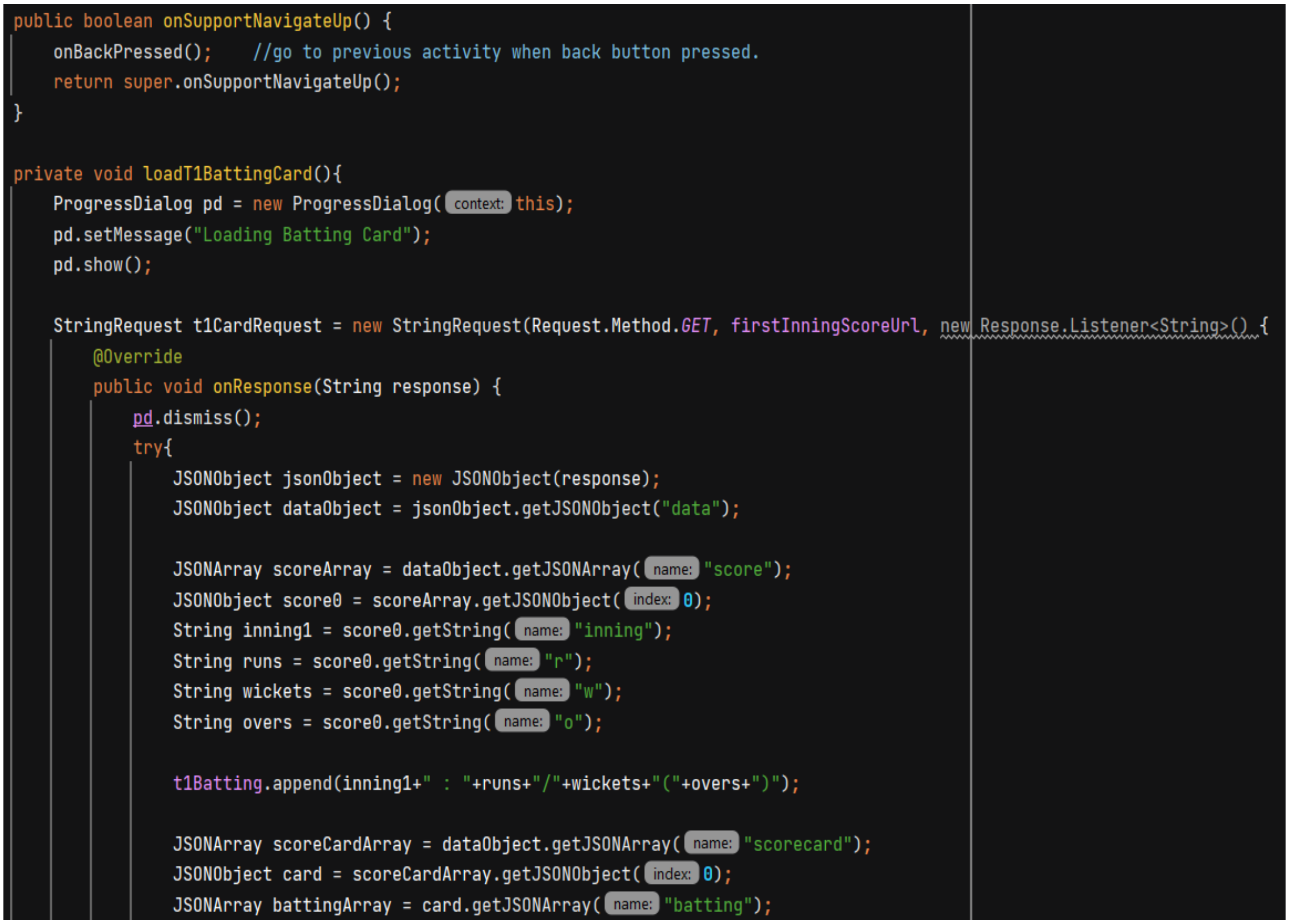


```
StringRequest summaryStringRequest = new StringRequest(Request.Method.GET,
        scoreUrl,
        new Response.Listener<String>() {
            @Override
            public void onResponse(String response) {
                pd.dismiss();

                try{
                    JSONObject jsonObject = new JSONObject(response).getJSONObject("data");
                    String matchName = jsonObject.getString( name: "name");
                    String matchStatus = jsonObject.getString( name: "status");
                    String type = jsonObject.getString( name: "matchType");
                    String tossWin = jsonObject.getString( name: "tossWinner");

                    name.setText(matchName);
                    status.setText(matchStatus);
                    matchtype.setText(type);
                    toss.setText(tossWin);
                }
                catch(Exception e){
                    Toast.makeText( context: matchSummary.this,  text: ""+e.getMessage(), Toast.LENGTH_SHORT).show();
                }
            }
        }, new Response.ErrorListener() {
    @Override
    public void onErrorResponse(VolleyError error) {
        Toast.makeText( context: matchSummary.this,  text: ""+error.toString(), Toast.LENGTH_SHORT).show();
    }
});
```

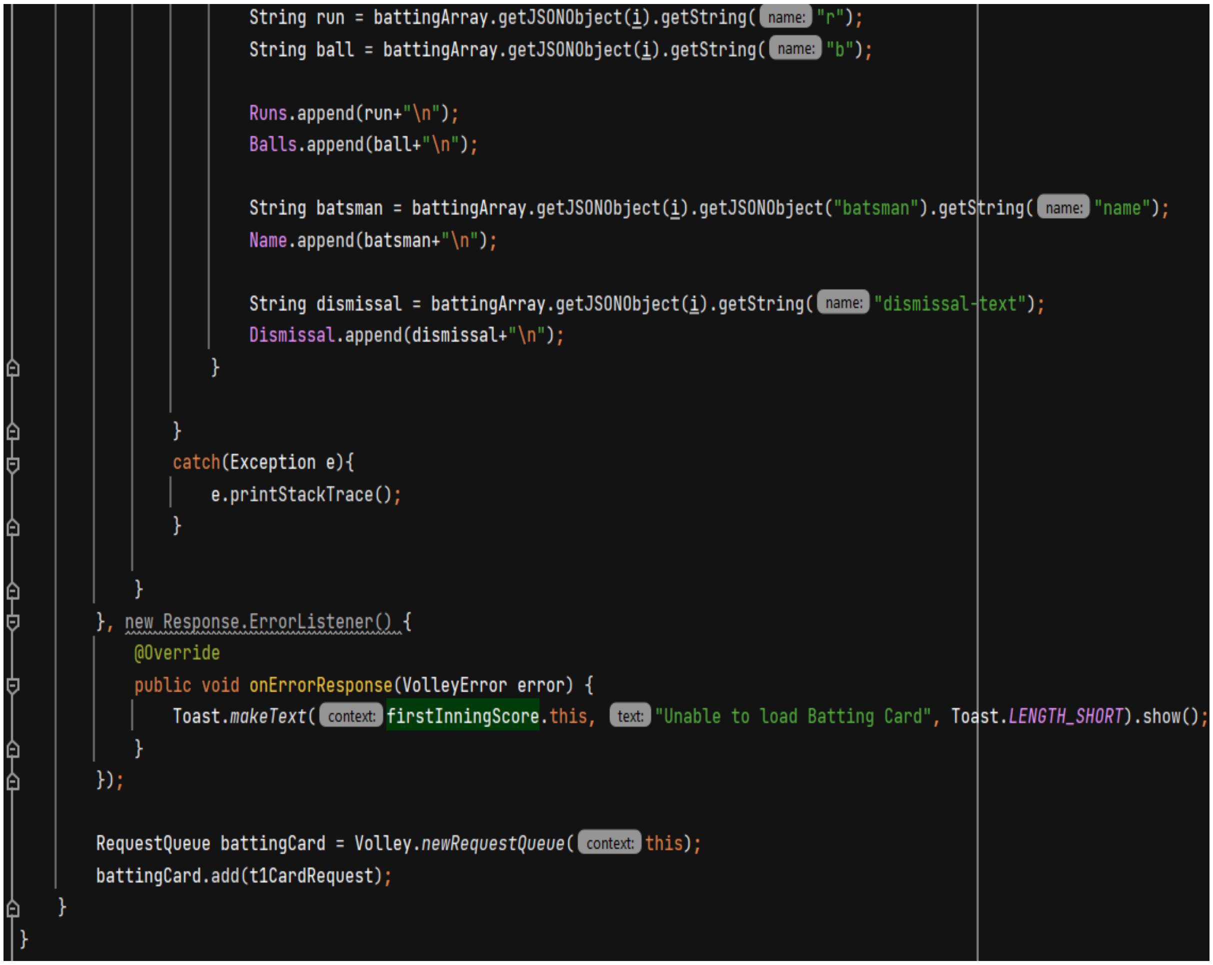
❖ Batting Card Activity



```java
package com;

import ...

public class firstInningScore extends AppCompatActivity {

    String firstInningScoreUrl = "https://api.cricapi.com/v1/match_scorecard?apikey=64146273-bada-44b7-b14c-2a92a457f652&id=";
    TextView t1Batting,Name,Runs,Balls,Dismissal;

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

        ActionBar actionBar = getSupportActionBar();
        actionBar.setTitle("Inning1 Batting Card");
        actionBar.setDisplayShowHomeEnabled(true);  // enabling back button from this activity.
        actionBar.setDisplayHomeAsUpEnabled(true);

        Intent intent = getIntent();
        String id = intent.getStringExtra( name: "mId");
        firstInningScoreUrl += id;

        t1Batting = findViewById(R.id.t1Batting);
        Name = findViewById(R.id.PName);
        Runs = findViewById(R.id.Runs);
        Balls = findViewById(R.id.balls);
        Dismissal = findViewById(R.id.dismissal);

        loadT1BattingCard();
    }
```

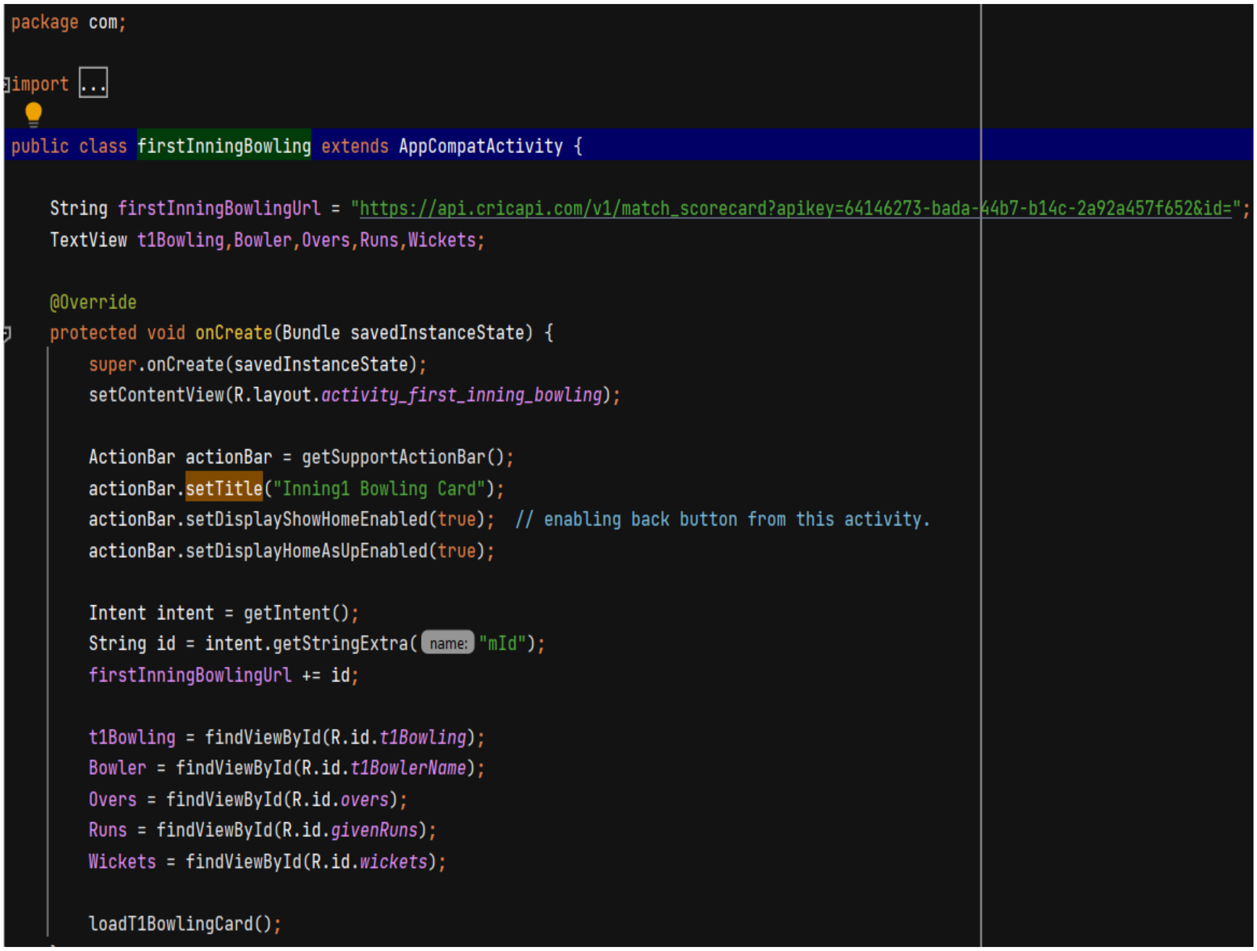


```
public boolean onSupportNavigateUp() {
    onBackPressed();    //go to previous activity when back button pressed.
    return super.onSupportNavigateUp();
}

private void loadT1BattingCard(){
    ProgressDialog pd = new ProgressDialog( context: this);
    pd.setMessage("Loading Batting Card");
    pd.show();

    StringRequest t1CardRequest = new StringRequest(Request.Method.GET, firstInningScoreUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            pd.dismiss();
            try{
                JSONObject jsonObject = new JSONObject(response);
                JSONObject dataObject = jsonObject.getJSONObject("data");

                JSONArray scoreArray = dataObject.getJSONArray( name: "score");
                JSONObject score0 = scoreArray.getJSONObject( index: 0);
                String inning1 = score0.getString( name: "inning");
                String runs = score0.getString( name: "r");
                String wickets = score0.getString( name: "w");
                String overs = score0.getString( name: "o");

                t1Batting.append(inning1+" : "+runs+"/"+wickets+"("+overs+")");

                JSONArray scoreCardArray = dataObject.getJSONArray( name: "scorecard");
                JSONObject card = scoreCardArray.getJSONObject( index: 0);
                JSONArray battingArray = card.getJSONArray( name: "batting");
```

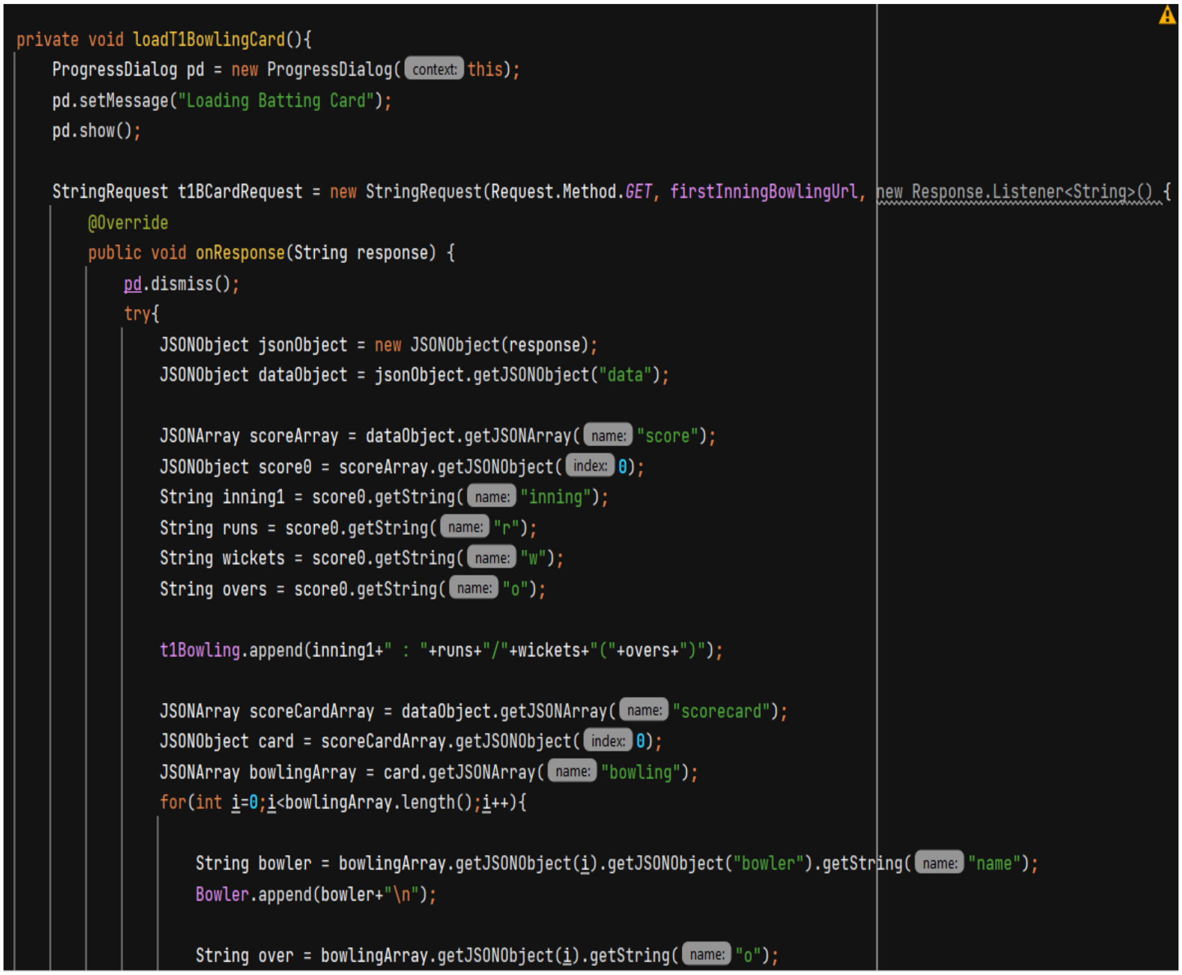


```
                        String run = battingArray.getJSONObject(i).getString( name: "r");
                        String ball = battingArray.getJSONObject(i).getString( name: "b");

                        Runs.append(run+"\n");
                        Balls.append(ball+"\n");

                        String batsman = battingArray.getJSONObject(i).getJSONObject("batsman").getString( name: "name");
                        Name.append(batsman+"\n");

                        String dismissal = battingArray.getJSONObject(i).getString( name: "dismissal-text");
                        Dismissal.append(dismissal+"\n");
                    }

                }
                catch(Exception e){
                    e.printStackTrace();
                }

            }
        }, new Response.ErrorListener() {
            @Override
            public void onErrorResponse(VolleyError error) {
                Toast.makeText( context: firstInningScore.this,  text: "Unable to load Batting Card", Toast.LENGTH_SHORT).show();
            }
        });

        RequestQueue battingCard = Volley.newRequestQueue( context: this);
        battingCard.add(t1CardRequest);
    }
}
```

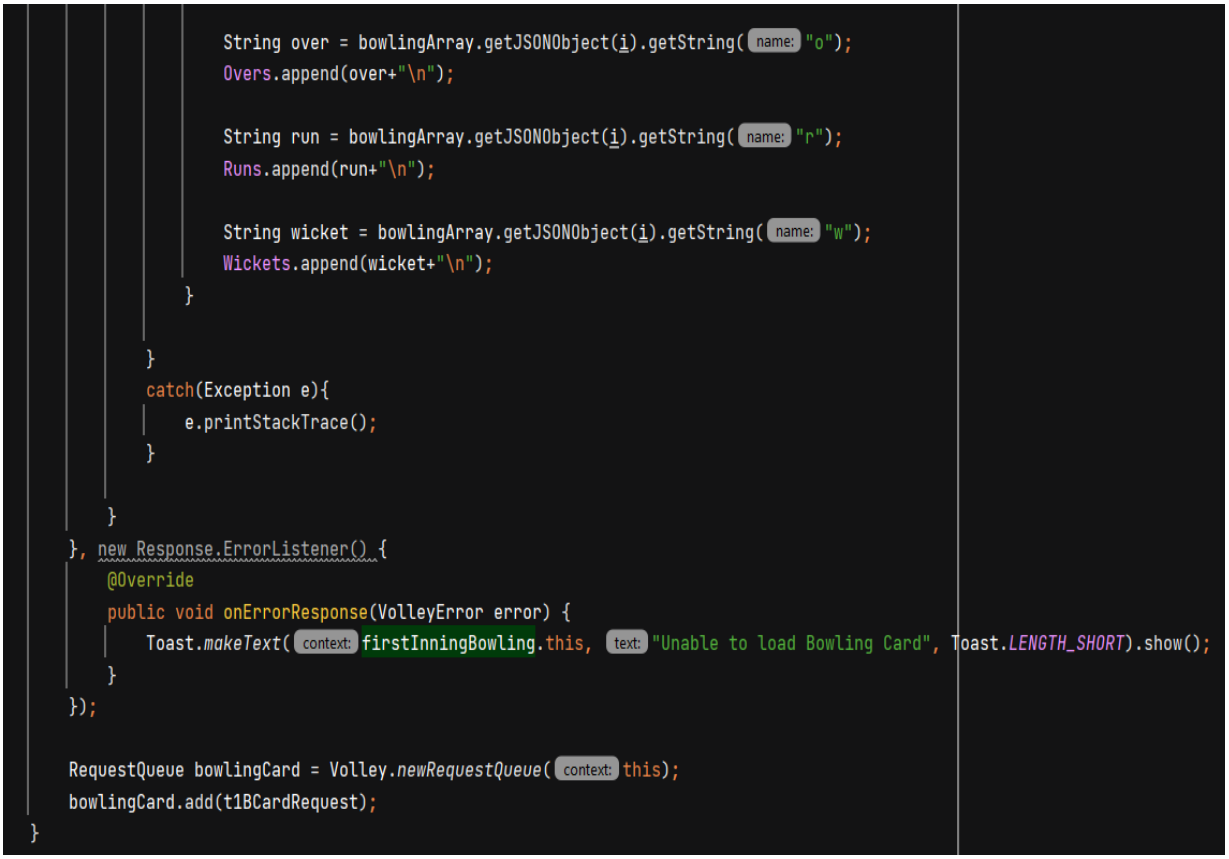
❖ Bowling Card



```
package com;

import ...

public class firstInningBowling extends AppCompatActivity {

    String firstInningBowlingUrl = "https://api.cricapi.com/v1/match_scorecard?apikey=64146273-bada-44b7-b14c-2a92a457f652&id=";
    TextView t1Bowling,Bowler,Overs,Runs,Wickets;

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

        ActionBar actionBar = getSupportActionBar();
        actionBar.setTitle("Inning1 Bowling Card");
        actionBar.setDisplayShowHomeEnabled(true);  // enabling back button from this activity.
        actionBar.setDisplayHomeAsUpEnabled(true);

        Intent intent = getIntent();
        String id = intent.getStringExtra( name: "mId");
        firstInningBowlingUrl += id;

        t1Bowling = findViewById(R.id.t1Bowling);
        Bowler = findViewById(R.id.t1BowlerName);
        Overs = findViewById(R.id.overs);
        Runs = findViewById(R.id.givenRuns);
        Wickets = findViewById(R.id.wickets);

        loadT1BowlingCard();
```

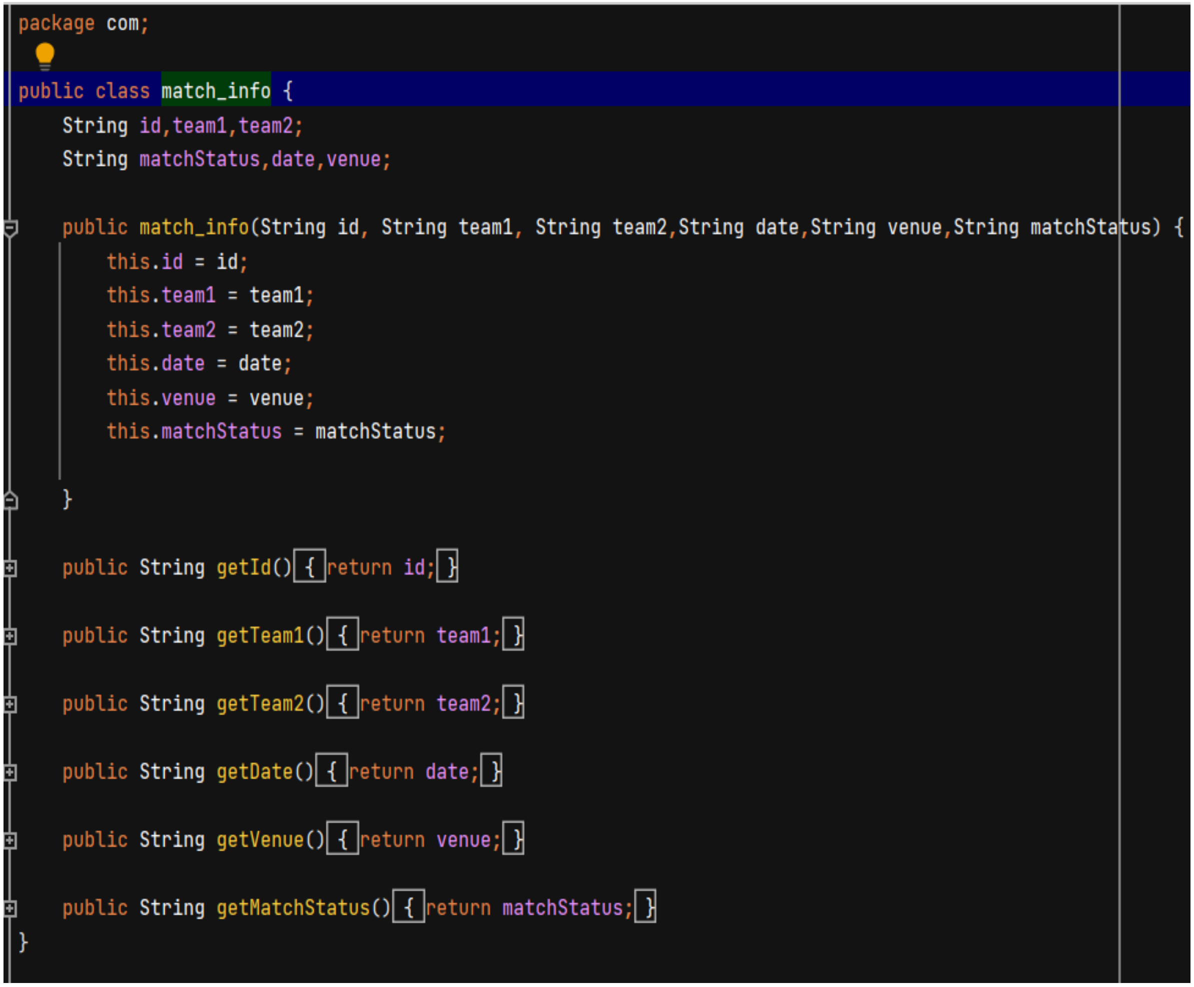


```
private void loadT1BowlingCard(){
    ProgressDialog pd = new ProgressDialog( context: this);
    pd.setMessage("Loading Batting Card");
    pd.show();

    StringRequest t1BCardRequest = new StringRequest(Request.Method.GET, firstInningBowlingUrl, new Response.Listener<String>() {
        @Override
        public void onResponse(String response) {
            pd.dismiss();
            try{
                JSONObject jsonObject = new JSONObject(response);
                JSONObject dataObject = jsonObject.getJSONObject("data");

                JSONArray scoreArray = dataObject.getJSONArray( name: "score");
                JSONObject score0 = scoreArray.getJSONObject( index: 0);
                String inning1 = score0.getString( name: "inning");
                String runs = score0.getString( name: "r");
                String wickets = score0.getString( name: "w");
                String overs = score0.getString( name: "o");

                t1Bowling.append(inning1+" : "+runs+"/"+wickets+"("+overs+")");

                JSONArray scoreCardArray = dataObject.getJSONArray( name: "scorecard");
                JSONObject card = scoreCardArray.getJSONObject( index: 0);
                JSONArray bowlingArray = card.getJSONArray( name: "bowling");
                for(int i=0;i<bowlingArray.length();i++){

                    String bowler = bowlingArray.getJSONObject(i).getJSONObject("bowler").getString( name: "name");
                    Bowler.append(bowler+"\n");

                    String over = bowlingArray.getJSONObject(i).getString( name: "o");
```



```
                    String over = bowlingArray.getJSONObject(i).getString( name: "o");
                    Overs.append(over+"\n");

                    String run = bowlingArray.getJSONObject(i).getString( name: "r");
                    Runs.append(run+"\n");

                    String wicket = bowlingArray.getJSONObject(i).getString( name: "w");
                    Wickets.append(wicket+"\n");
                }

            }
            catch(Exception e){
                e.printStackTrace();
            }

        }
    }, new Response.ErrorListener() {
        @Override
        public void onErrorResponse(VolleyError error) {
            Toast.makeText( context: firstInningBowling.this,  text: "Unable to load Bowling Card", Toast.LENGTH_SHORT).show();
        }
    });

    RequestQueue bowlingCard = Volley.newRequestQueue( context: this);
    bowlingCard.add(t1BCardRequest);
}
```

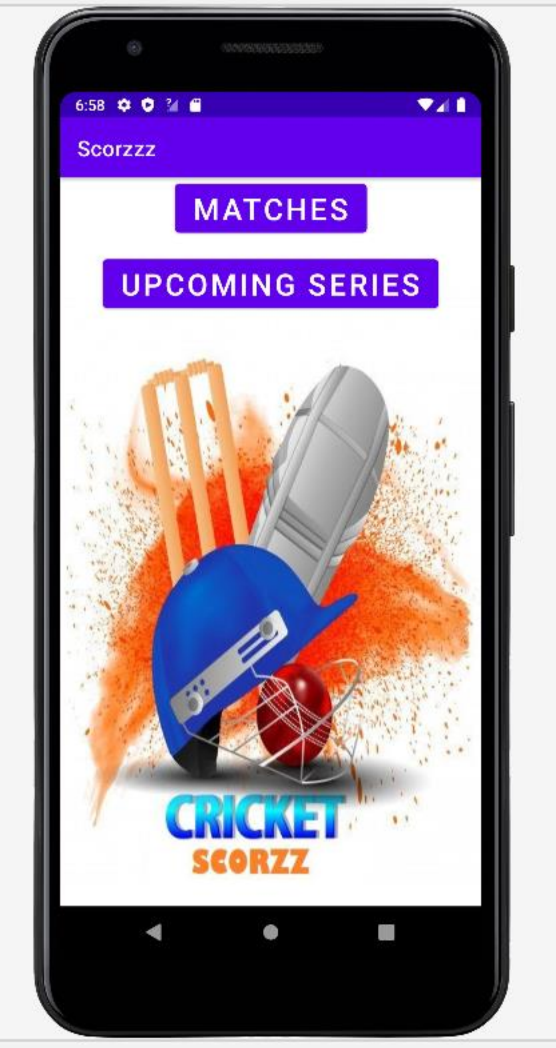
❖ Match Info



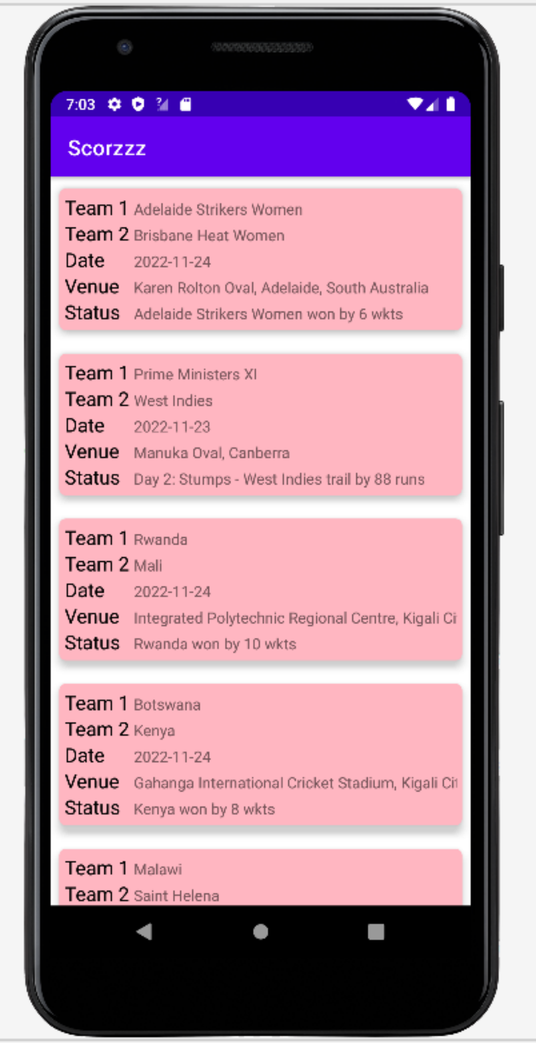
```java
package com;

public class match_info {
    String id,team1,team2;
    String matchStatus,date,venue;

    public match_info(String id, String team1, String team2,String date,String venue,String matchStatus) {
        this.id = id;
        this.team1 = team1;
        this.team2 = team2;
        this.date = date;
        this.venue = venue;
        this.matchStatus = matchStatus;

    }

    public String getId() { return id; }

    public String getTeam1() { return team1; }

    public String getTeam2() { return team2; }

    public String getDate() { return date; }

    public String getVenue() { return venue; }

    public String getMatchStatus() { return matchStatus; }
}
```

OUTPUT SNIPPETS

❖ HOME SCREEN



❖ Match List Page



7:03

Scorzzz

Team 1	Adelaide Strikers Women
Team 2	Brisbane Heat Women
Date	2022-11-24
Venue	Karen Rolton Oval, Adelaide, South Australia
Status	Adelaide Strikers Women won by 6 wkts

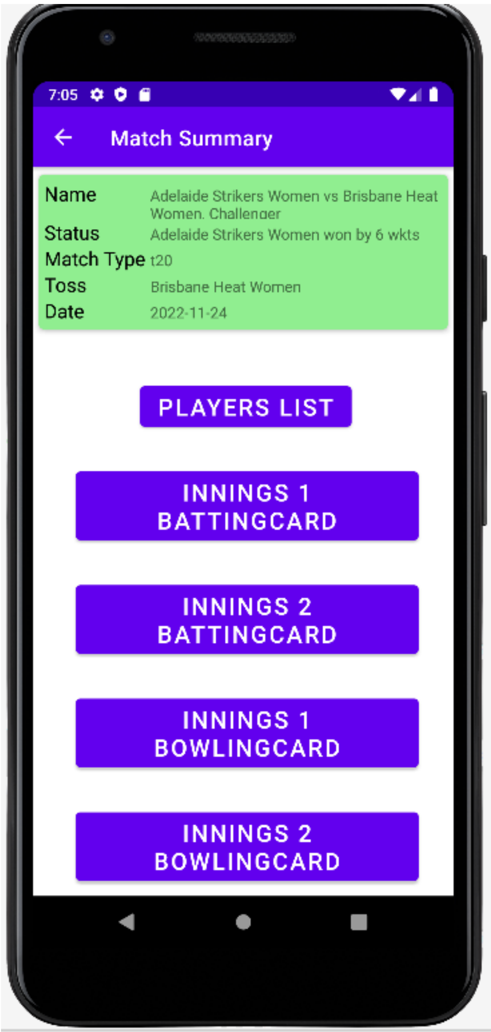
Team 1	Prime Ministers XI
Team 2	West Indies
Date	2022-11-23
Venue	Manuka Oval, Canberra
Status	Day 2: Stumps - West Indies trail by 88 runs

Team 1	Rwanda
Team 2	Mali
Date	2022-11-24
Venue	Integrated Polytechnic Regional Centre, Kigali Ci
Status	Rwanda won by 10 wkts

Team 1	Botswana
Team 2	Kenya
Date	2022-11-24
Venue	Gahanga International Cricket Stadium, Kigali Cit
Status	Kenya won by 8 wkts

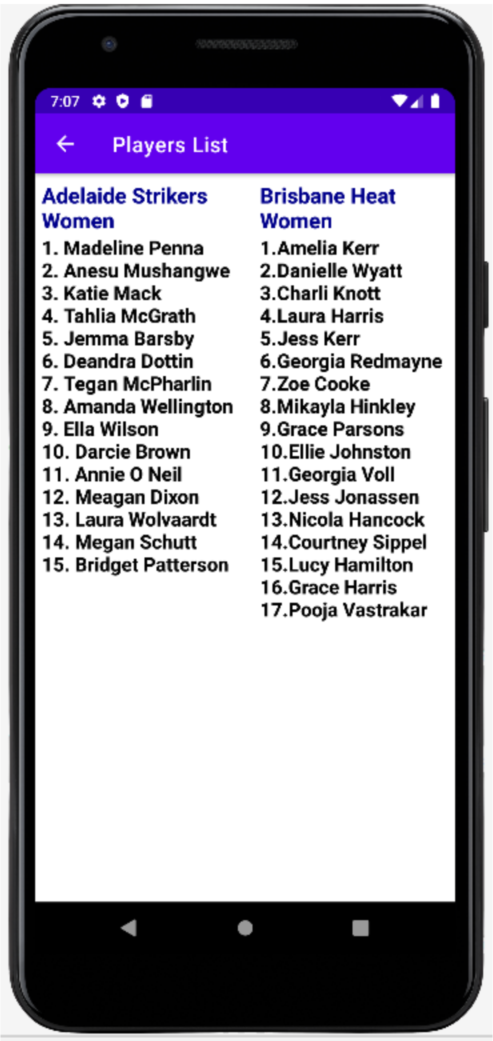
Team 1	Malawi
Team 2	Saint Helena

❖ Match Summary Page



Name	Adelaide Strikers Women vs Brisbane Heat Women. Challenger
Status	Adelaide Strikers Women won by 6 wkts
Match Type	t20
Toss	Brisbane Heat Women
Date	2022-11-24

❖ PLAYER LIST PAGE



Players List

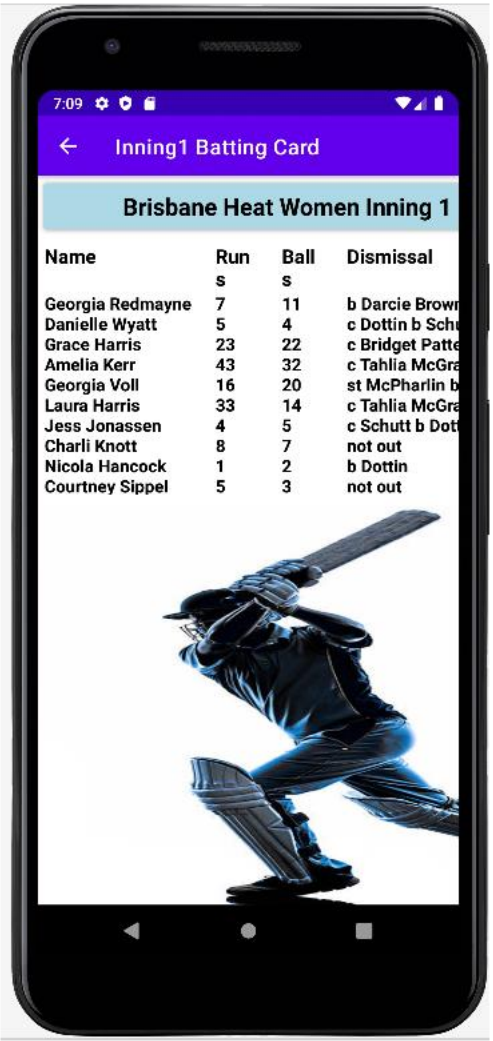
Adelaide Strikers Women

1. Madeline Penna
2. Anesu Mushangwe
3. Katie Mack
4. Tahlia McGrath
5. Jemma Barsby
6. Deandra Dottin
7. Tegan McPharlin
8. Amanda Wellington
9. Ella Wilson
10. Darcie Brown
11. Annie O Neil
12. Meagan Dixon
13. Laura Wolvaardt
14. Megan Schutt
15. Bridget Patterson

Brisbane Heat Women

1.Amelia Kerr
2.Danielle Wyatt
3.Charli Knott
4.Laura Harris
5.Jess Kerr
6.Georgia Redmayne
7.Zoe Cooke
8.Mikayla Hinkley
9.Grace Parsons
10.Ellie Johnston
11.Georgia Voll
12.Jess Jonassen
13.Nicola Hancock
14.Courtney Sippel
15.Lucy Hamilton
16.Grace Harris
17.Pooja Vastrakar

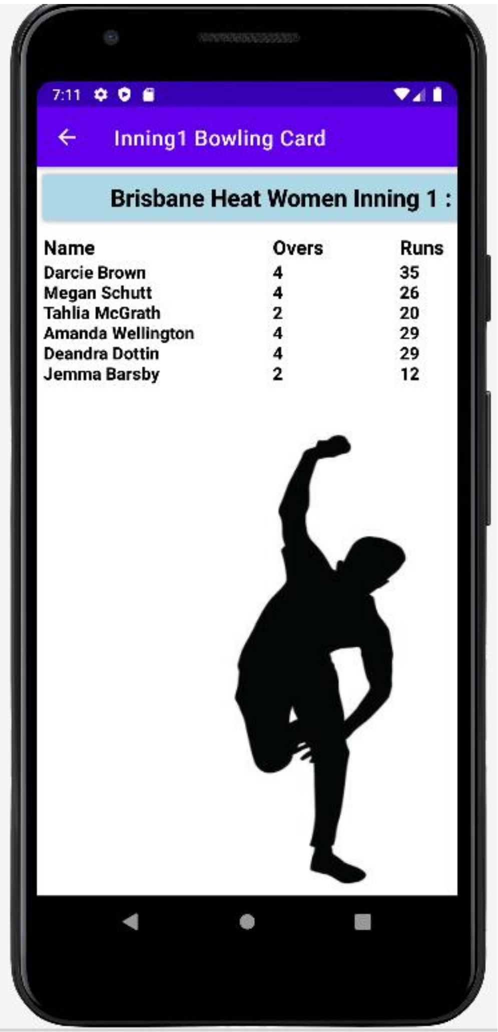
❖ BATTING CARD PAGE



Brisbane Heat Women Inning 1

Name	Runs	Balls	Dismissal
Georgia Redmayne	7	11	b Darcie Brown
Danielle Wyatt	5	4	c Dottin b Schu
Grace Harris	23	22	c Bridget Patte
Amelia Kerr	43	32	c Tahlia McGra
Georgia Voll	16	20	st McPharlin b
Laura Harris	33	14	c Tahlia McGra
Jess Jonassen	4	5	c Schutt b Dott
Charli Knott	8	7	not out
Nicola Hancock	1	2	b Dottin
Courtney Sippel	5	3	not out

❖ BOWLING CARD PAGE



Inning1 Bowling Card

Brisbane Heat Women Inning 1 :

Name	Overs	Runs
Darcie Brown	4	35
Megan Schutt	4	26
Tahlia McGrath	2	20
Amanda Wellington	4	29
Deandra Dottin	4	29
Jemma Barsby	2	12

CONCLUSION

We have completed our project within time limit with the coordination of our team members under the supervision of our mentor Mr. FarmanUl Haque.

Our project repository is available at
https://github.com/AmanSingh3175/Scorzz