

Code Testing Document

Software Engineering Lab

Group no. 16

Team members

Annapurne Krishna - 180101009

Anshul Mittal - 180101010

Munindra Naik - 180101045

Sanket Pandhare - 180101069

1. Introduction

This document contains the complete software review of our app Inter-IIT Sports Tournament. Code review for a model is carried out after the module is successfully compiled and all the syntax errors have been eliminated. Code reviews are extremely cost-effective strategies for reduction in coding errors and to produce high quality code. Normally, two types of reviews are carried out on the code of a module. These two types of code review techniques are code inspection and code walk through. The code testing team in isolation tests different units and modules of the system. Different members of the code testing team have submitted their reports. Although, we are only performing the Code Inspection part wherein each member goes through code to discover some common types of errors caused due to oversight and improper programming. The main objectives of the walk through was to discover the algorithmic and logical errors in the code.

The members noted down their findings, which were discussed in a walk through meeting where the coder of the module was also present.

2. Code

There is 1 Module LeaderBoard containing the following files: MainActivity.java, Homescreen.java, Leaderboard.java, MyAdapter.java, standingEntry.java.

2.1. MainActivity:

```
C: > Users > sanket pandhare > Downloads > MainActivity.java
1  package com.example.interiitapp;
2
3
4
5  import androidx.appcompat.app.AppCompatActivity;
6
7
8  import android.content.Intent;
9  import android.os.Bundle;
10 import android.view.View;
11 import android.widget.Button;
12
13 public class MainActivity extends AppCompatActivity {
14     Button signparticipant,signadmin,continueguest;
15     @Override
16     protected void onCreate(Bundle savedInstanceState) {
17         super.onCreate(savedInstanceState);
18         setContentView(R.layout.activity_main);
19
20         signparticipant = findViewById(R.id.button2);
21         signadmin = findViewById(R.id.button3);
22         continueguest = findViewById(R.id.button4);
23
24         signparticipant.setOnClickListener(new View.OnClickListener() {
25             @Override
26             public void onClick(View v) {
27                 Intent intent = new Intent(MainActivity.this, Homescreen.class);
28                 startActivity(intent);
29             }
30         });
31     }
32 }
```

C: > Users > sanket pandhare > Downloads > MainActivity.java

```
19
20     signparticipant = findViewById(R.id.button2);
21     signadmin = findViewById(R.id.button3);
22     continueguest = findViewById(R.id.button4);
23
24     signparticipant.setOnClickListener(new View.OnClickListener() {
25         @Override
26         public void onClick(View v) {
27             Intent intent = new Intent(MainActivity.this, Homescree.class);
28             startActivity(intent);
29         }
30     });
31
32     signadmin.setOnClickListener(new View.OnClickListener() {
33         @Override
34         public void onClick(View v) {
35             Intent intent = new Intent(MainActivity.this, Homescree.class);
36             startActivity(intent);
37         }
38     });
39
40     continueguest.setOnClickListener(new View.OnClickListener() {
41         @Override
42         public void onClick(View v) {
43             Intent intent = new Intent(MainActivity.this, Homescree.class);
44             startActivity(intent);
45         }
46     });
47 }
48 }
```

2.2. Homescreen

```
C: > Users > sanket pandhare > Downloads > Homescreen.java
1  package com.example.interiitapp;
2
3  import androidx.appcompat.app.AppCompatActivity;
4
5  import android.content.Intent;
6  import android.os.Bundle;
7  import android.view.View;
8  import android.widget.ImageView;
9
10 public class Homescreen extends AppCompatActivity {
11     ImageView leaderboard,matchresult,schedule;
12     ImageView menu;
13     @Override
14     protected void onCreate(Bundle savedInstanceState) {
15         super.onCreate(savedInstanceState);
16         setContentView(R.layout.activity_homescreen);
17
18         //binding the images with the variables
19         leaderboard = findViewById(R.id.leaderboard);
20         matchresult = findViewById(R.id.matchresult);
21         schedule = findViewById(R.id.schedule);
22
23         menu = findViewById(R.id.menu);
24
25         leaderboard.setOnClickListener(new View.OnClickListener() {
26             @Override
27             public void onClick(View v) {
28                 Intent intent = new Intent(Homescreen.this,Leaderboard.class);
29                 startActivity(intent);
30             }
31         });
32     }
33 }
```

2.3. Leaderboard

```
1 package com.example.interitapp;
2
3 import androidx.appcompat.app.AppCompatActivity;
4 import androidx.recyclerview.widget.LinearLayoutManager;
5 import androidx.recyclerview.widget.RecyclerView;
6
7 import android.os.Bundle;
8
9 import com.firebase.ui.database.FirebaseRecyclerOptions;
10 import com.google.firebase.database.DatabaseReference;
11 import com.google.firebase.database.FirebaseDatabase;
12
13 import java.util.Objects;
14
15 public class Leaderboard extends AppCompatActivity {
16
17     RecyclerView recyclerView;
18     MyAdapter adapter;
19     DatabaseReference mbase;
20
21     @Override
22     protected void onCreate(Bundle savedInstanceState) {
23         super.onCreate(savedInstanceState);
24         setContentView(R.layout.activity_leaderboard);
25         Objects.requireNonNull(getSupportActionBar()).hide();
26
27         mbase = FirebaseDatabase.getInstance().getReference();
28         recyclerView = findViewById(R.id.leaderboardlist);
29         recyclerView.setLayoutManager(new LinearLayoutManager(this));
30         FirebaseRecyclerOptions<standingEntry> options
31             = new FirebaseRecyclerOptions.Builder<standingEntry>()
32                 .setQuery(mbase, standingEntry.class)
33                 .build();
34         recyclerView.setAdapter(adapter);
35
36     }
37
38
39     @Override
40     protected void onStart() {
41         super.onStart();
42         adapter.startListening();
43     }
44
45     @Override
46     protected void onStop() {
47         super.onStop();
48         adapter.stopListening();
49     }
50 }
```

2.4. MyAdapter

C: > Users > sanket pandhare > Downloads > MyAdapter.java

```
1  package com.example.interiitapp;
2
3
4  import android.view.LayoutInflater;
5  import android.view.View;
6  import android.view.ViewGroup;
7  import android.widget.TextView;
8
9
10 import androidx.annotation.NonNull;
11 import androidx.recyclerview.widget.RecyclerView;
12
13 import com.firebase.ui.database.FirebaseRecyclerAdapter;
14 import com.firebase.ui.database.FirebaseRecyclerOptions;
15
16
17 public class MyAdapter extends FirebaseRecyclerAdapter<standingEntry, MyAdapter.entryViewHolder> {
18
19     public MyAdapter(@NonNull FirebaseRecyclerOptions<standingEntry> options) {
20         super(options);
21     }
22
23     @Override
24     protected void onBindViewHolder(@NonNull entryViewHolder holder, int position, @NonNull standingEntry model){
25         holder.rank.setText(String.valueOf(position));
26         holder.name.setText(model.getName());
27         holder.points.setText(String.valueOf(model.getPoints()));
28         //System.out.println("GIVEN HERE" + model.getName());
29     }
30 }
```

C: > Users > sanket pandhare > Downloads > MyAdapter.java

```
19     public MyAdapter(@NonNull FirebaseRecyclerOptions<standingEntry> options) {
20         super(options);
21     }
22
23     @Override
24     protected void onBindViewHolder(@NonNull entryViewHolder holder, int position, @NonNull standingEntry model){
25         holder.rank.setText(String.valueOf(position));
26         holder.name.setText(model.getName());
27         holder.points.setText(String.valueOf(model.getPoints()));
28         //System.out.println("GIVEN HERE" + model.getName());
29     }
30
31     @NonNull
32     @Override
33     public entryViewHolder onCreateViewHolder(@NonNull ViewGroup parent, int viewType){
34         View view = LayoutInflater.from(parent.getContext()).inflate(R.layout.cardview,parent,false);
35         return new entryViewHolder(view);
36     }
37     static class entryViewHolder extends RecyclerView.ViewHolder{
38         TextView rank,name,points;
39
40         public entryViewHolder(@NonNull View itemView){
41             super(itemView);
42
43             rank = itemView.findViewById(R.id.rank);
44             name = itemView.findViewById(R.id.name);
45             points = itemView.findViewById(R.id.points);
46         }
47     }
48 }
```

2.5. standingEntry

```
C: > Users > sanket pandhare > Downloads > J standingEntry.java
1  package com.example.interiitapp;
2
3  public class standingEntry {
4      String name;
5      Integer points;
6
7      public standingEntry(String name, Integer points) {
8          this.name = name;
9          this.points = points;
10     }
11
12     public String getName() {
13
14         return name;
15     }
16
17     public Integer getPoints() {
18
19         return points;
20     }
21 }
22
```

3. Code Testing Team

3.1. Team Profile

The code testing team comprises the following members, all of whom are Undergraduates currently pursuing Bachelor of Technology at Indian Institute of Technology Guwahati, India in the Department of Computer Science & Engineering. All of the members are currently in their final year.

I. Tanmay Sandbhor

II. Samish Gholap

All members of the team are proficient in Java and have past experience in developing android applications.

4. Code Inspection Reports

4.1. Modules Tested by the team

The different modules tested by the team includes-

- LeaderBoard
 - MainActivity.java
 - Homescreen.java
 - Leaderboard.java
 - MyAdapter.java
 - standingEntry.java

We start from MainActivity.java and after login as a participant or admin, or continuing as a guest, onClickListner is triggered and we move on to Homescreen.java.

Whenever we press the leaderboard tab in homescreen, leaderboard.setonClickListener is triggered and we move onto Leaderboard.java.

In Leaderboard.java

- At line no. 27, we connect to the database.
- At line no. 30, we fetch leaderboard data from the database.
- At line no. 34, we try to output the leaderboard into the screen by moving into MyAdapter.java.

MyAdapter.java processes the obtained data from the database and passes it to the screen UI to print it.

standingEntry.java is used to store the data coming from the database in a structured way.

4.2. Report by Tanmay Sandbhor

After going through the code, there were some logical errors found in the algorithm used in the code. All these logical errors are reported in this report

- The headers of each module does not have all the details that are required from a good header, like - Name of the module, Date on which the module was created, Author's name, Modification history, Synopsis of the module, Different functions supported, along with their input/output parameters.
- Proper indentation is followed while writing code.
- OnCreate function in the leaderboard.java takes about 10s to load, it should have been minimized.
- mbase variable in Leaderboard.java is not checked for NULL, which may lead to error.
- Member variables of class standingEntry should have been set as private.
- Comments are required in the code as the functionality of some functions are not understandable and hence the complete functionality of the whole code could not be understood properly.
- Meaningful naming conventions are not practiced here.
- Import statements have not been justified in the code.
- No explanation of dependencies used is found.
- Proper folder structure is not followed, every file is put in a single folder.
- There were none uninitialized variables found in any module.
- standingEntry class variables should have been set as private.

4.3. Report by Samish Gholap

After going through the code, there were some logical errors found in the algorithm used in the code. All these logical errors are reported in this report.

- JUMP (go to) statements were not found in the code.
- Some of the functions are not properly commented because of which it is hard to understand their functionality. Comments in these functions should be more explanatory.
- All the loops will terminate according to their terminating conditions respectively after some finite iterations.
- No index of array is out of bound.
- Module loading time is approx 8 to 10 s which is more than expected.
- standingEntry class does not have any setter functions.
- Some variables in MyAdapter.java are not tested for NULL, which is not a good practice.
- Firebase output may result in NULL, which has not been checked in Leaderboard.java.
- No comments are found in the code, which makes it difficult to understand the code.
- Proper debugging statements should have been written and commented out.

5. Conclusion

The members of the code review team submitted the reports during their final meeting with the development team. From these submitted reports, we get to know about a few logical errors that were encountered during the execution of the code inspection and were listed down.

These errors will affect the code to run. The key errors found from the code review are as follows :-

- Execution time of the application is more than it should have been.
- Key variables should have been set as private for security purposes.
- Many variables are not checked for NULL, which may crash the execution.
- No function to set class variables (setters).
- No comments and function descriptions given.
- Debugging statements are not made.

Fixing these bugs can improve our application, Inter-IIT Sports Tournament.