Android Application Project

Basketball Team Application

By Louis Devèze & Victor Grégoire

Introduction:

For this project, we choose to make an application to help basketball coaches of the NBA to see the statistics of their teams and the matches played by interacting with a database. In this report, we introduce the whole architecture of the app and the design choices. We also describe the functionalities we implemented and provide screenshots of what we've done.

List of functions implemented:

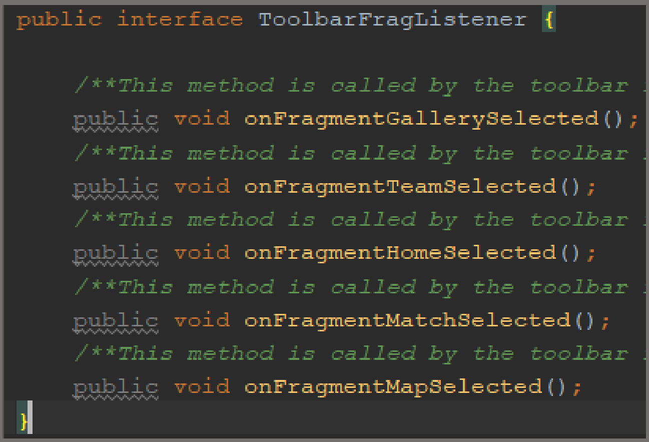
- Nice Design with Fragments and landscape layout for each view. Each Fragment communicates with the main Activity by interfaces and the toolbar fragment allows smooth navigation between content views.
- Retrieve External database information within JDBC Connection to a local WampServer
- Push some information about teams in the External Database
- Save External Data into a Local SQLite Database server and retrieve information locally saved
- Use of Async Tasks to do each information load into the application
- Geolocation of a match using the Google Cloud API Key to do reverse Geolocation and see the address on the map
- Take Photos with the default Photo application of the system and visualize them into a Gallery
- Shared Preferences to select the language of the app
- Intuitive and Ergonomic Design 😉

Possible improvements:

- Add a possibility to update or delete information of the database using the application
- Stock videos on a Server and retrieve them to look them
- Optimize the way we use Fragments inside main Application, to let the Fragment Manager recycle them if possible

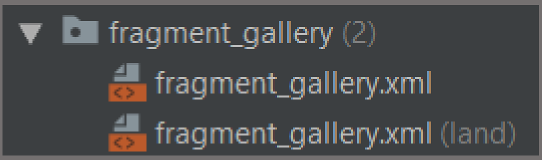
Code Snippets:

Interface Listener for Toolbar



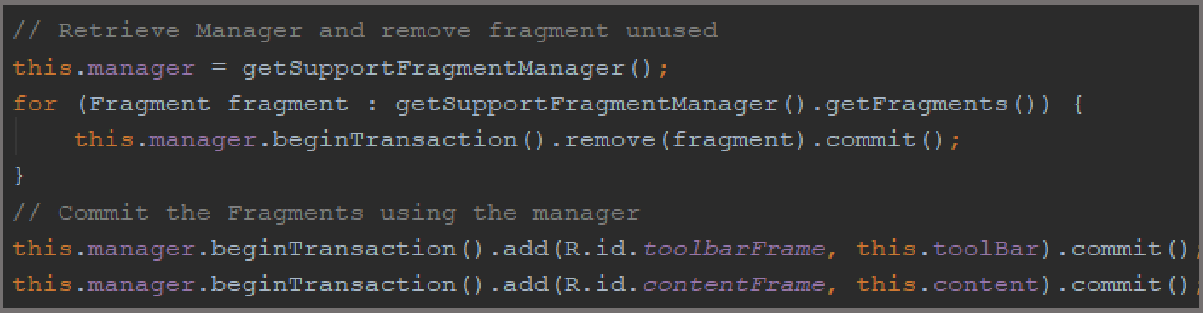
```
public interface ToolbarFragListener {

    /**This method is called by the toolbar
    public void onFragmentGallerySelected();
    /**This method is called by the toolbar
    public void onFragmentTeamSelected();
    /**This method is called by the toolbar
    public void onFragmentHomeSelected();
    /**This method is called by the toolbar
    public void onFragmentMatchSelected();
    /**This method is called by the toolbar
    public void onFragmentMapSelected();
}
```

Landscape – Portrait



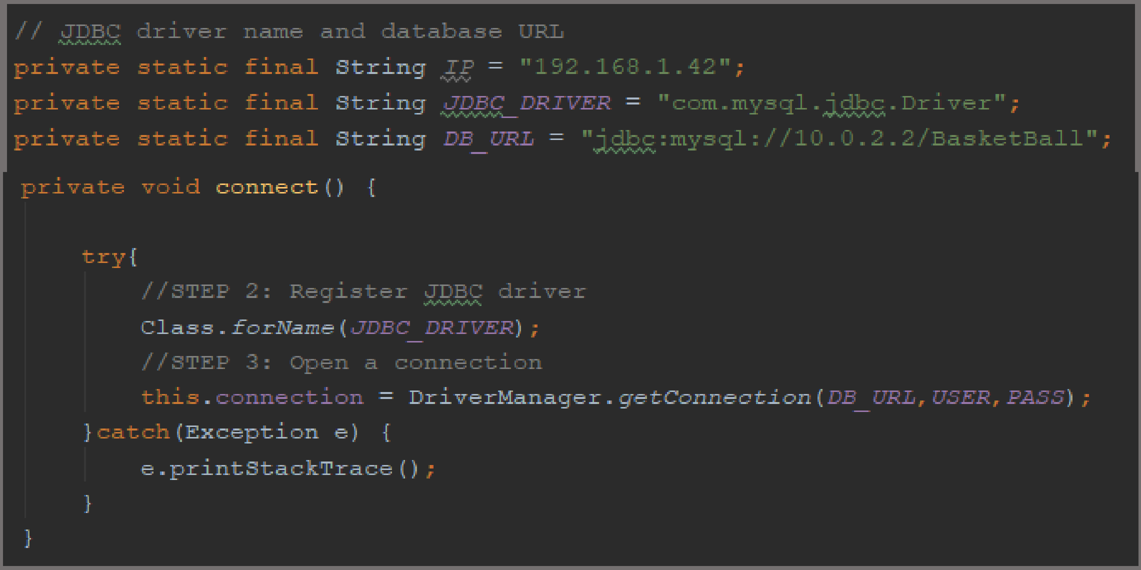
```
▼ fragment_gallery (2)
      fragment_gallery.xml
      fragment_gallery.xml (land)
```

Fragment Manager – Recreate Fragment and add it with Transactions



```
// Retrieve Manager and remove fragment unused
this.manager = getSupportFragmentManager();
for (Fragment fragment : getSupportFragmentManager().getFragments()) {
    this.manager.beginTransaction().remove(fragment).commit();
}
// Commit the Fragments using the manager
this.manager.beginTransaction().add(R.id.toolbarFrame, this.toolBar).commit()
this.manager.beginTransaction().add(R.id.contentFrame, this.content).commit()
```

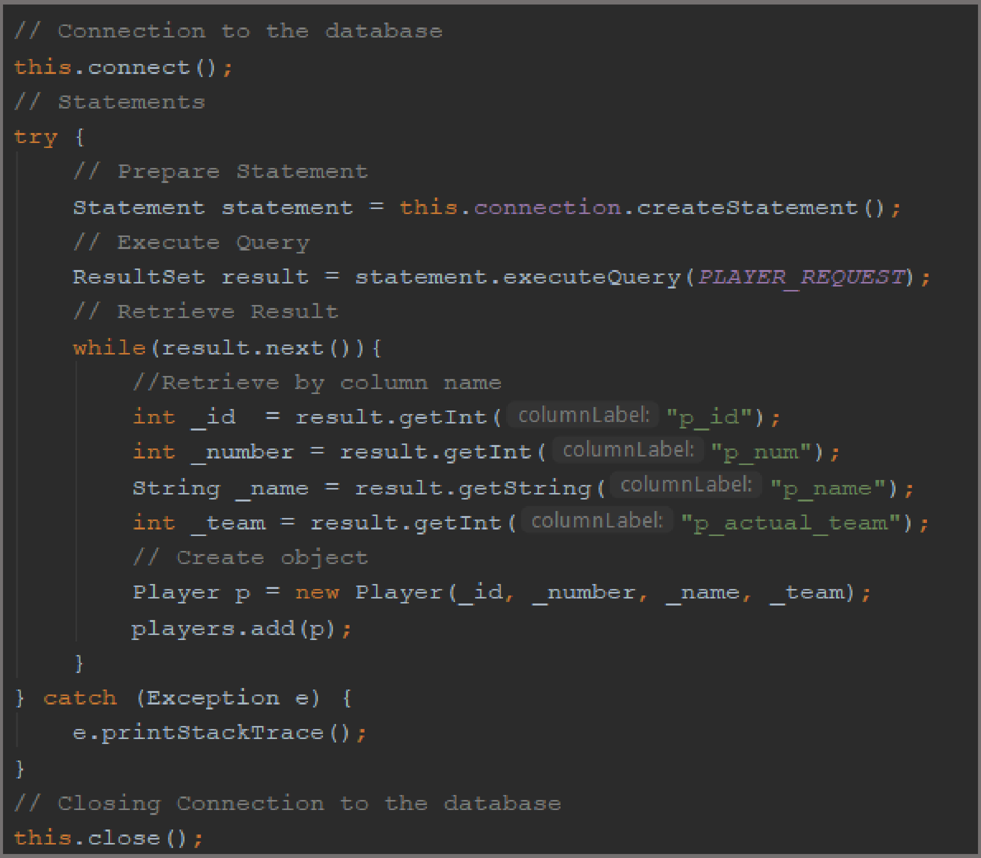
Database External Manager (Local Host)



```java
// JDBC driver name and database URL
private static final String IP = "192.168.1.42";
private static final String JDBC_DRIVER = "com.mysql.jdbc.Driver";
private static final String DB_URL = "jdbc:mysql://10.0.2.2/BasketBall";
```

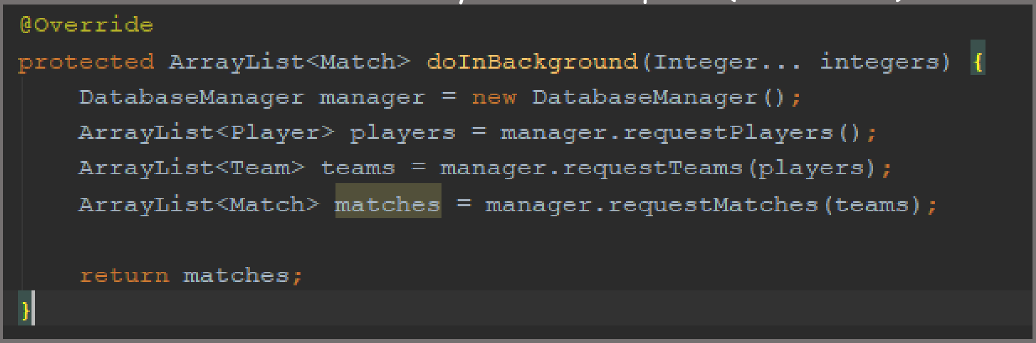
```java
private void connect() {

    try{
        //STEP 2: Register JDBC driver
        Class.forName(JDBC_DRIVER);
        //STEP 3: Open a connection
        this.connection = DriverManager.getConnection(DB_URL,USER,PASS);
    }catch(Exception e) {
        e.printStackTrace();
    }
}
```

Database External Manager Request (Local Host)



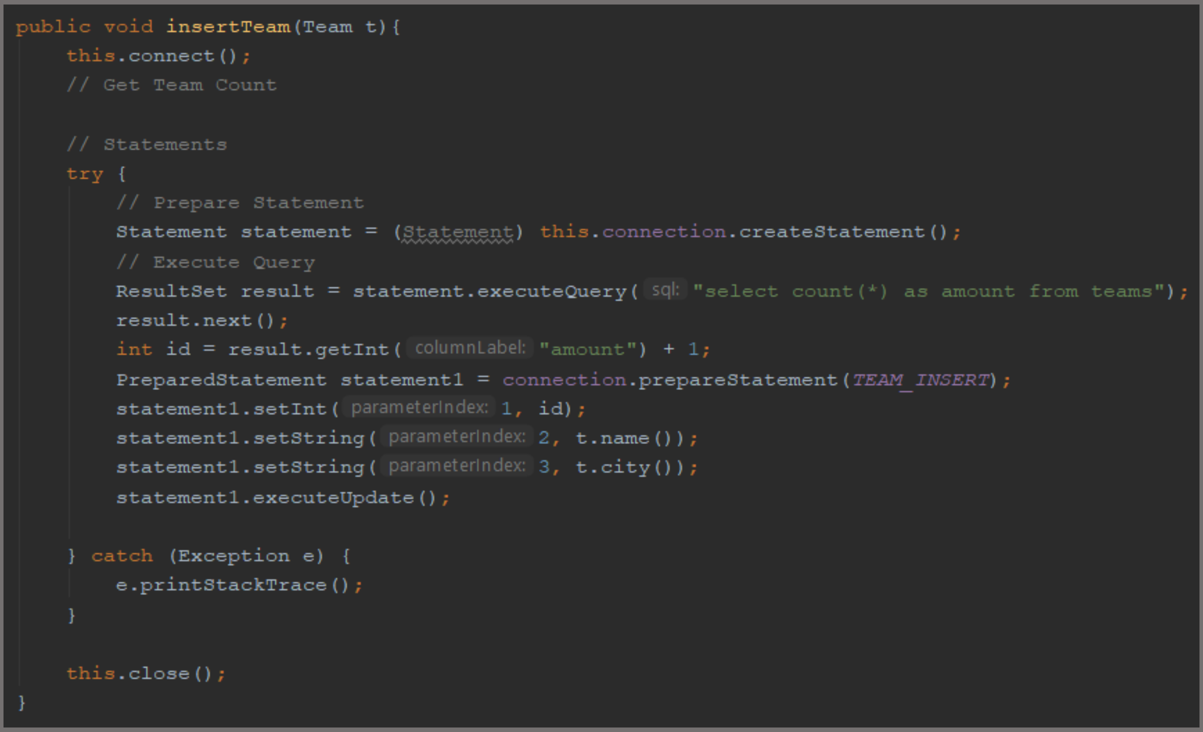
```java
// Connection to the database
this.connect();
// Statements
try {
    // Prepare Statement
    Statement statement = this.connection.createStatement();
    // Execute Query
    ResultSet result = statement.executeQuery(PLAYER_REQUEST);
    // Retrieve Result
    while(result.next()){
        //Retrieve by column name
        int _id  = result.getInt( columnLabel: "p_id");
        int _number = result.getInt( columnLabel: "p_num");
        String _name = result.getString( columnLabel: "p_name");
        int _team = result.getInt( columnLabel: "p_actual_team");
        // Create object
        Player p = new Player(_id, _number, _name, _team);
        players.add(p);
    }
} catch (Exception e) {
    e.printStackTrace();
}
// Closing Connection to the database
this.close();
```

Database External Async Task Request (Local Host)



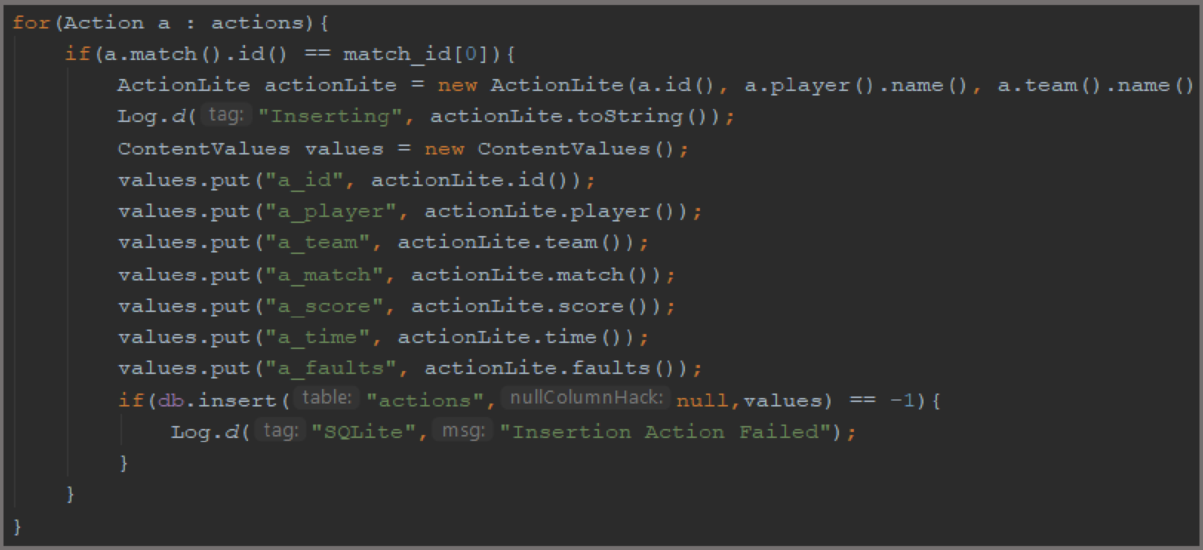
```java
@Override
protected ArrayList<Match> doInBackground(Integer... integers) {
    DatabaseManager manager = new DatabaseManager();
    ArrayList<Player> players = manager.requestPlayers();
    ArrayList<Team> teams = manager.requestTeams(players);
    ArrayList<Match> matches = manager.requestMatches(teams);

    return matches;
}
```

Database External Insertion JDBC



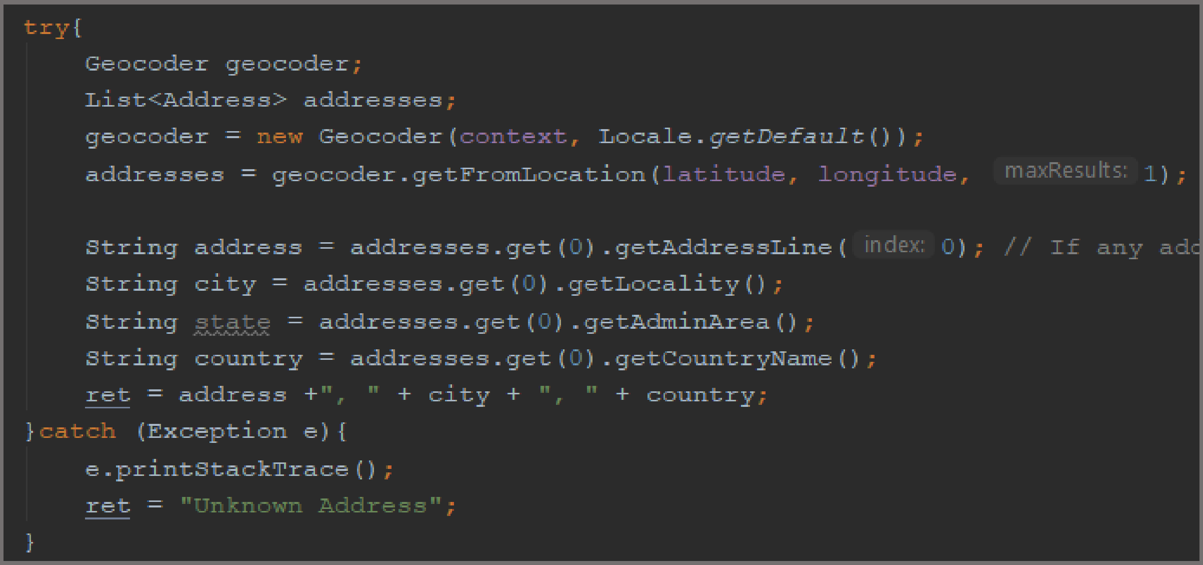
```java
public void insertTeam(Team t){
    this.connect();
    // Get Team Count

    // Statements
    try {
        // Prepare Statement
        Statement statement = (Statement) this.connection.createStatement();
        // Execute Query
        ResultSet result = statement.executeQuery( sql: "select count(*) as amount from teams");
        result.next();
        int id = result.getInt( columnLabel: "amount") + 1;
        PreparedStatement statement1 = connection.prepareStatement(TEAM_INSERT);
        statement1.setInt( parameterIndex: 1, id);
        statement1.setString( parameterIndex: 2, t.name());
        statement1.setString( parameterIndex: 3, t.city());
        statement1.executeUpdate();

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

    this.close();
}
```

Database Local Insertion



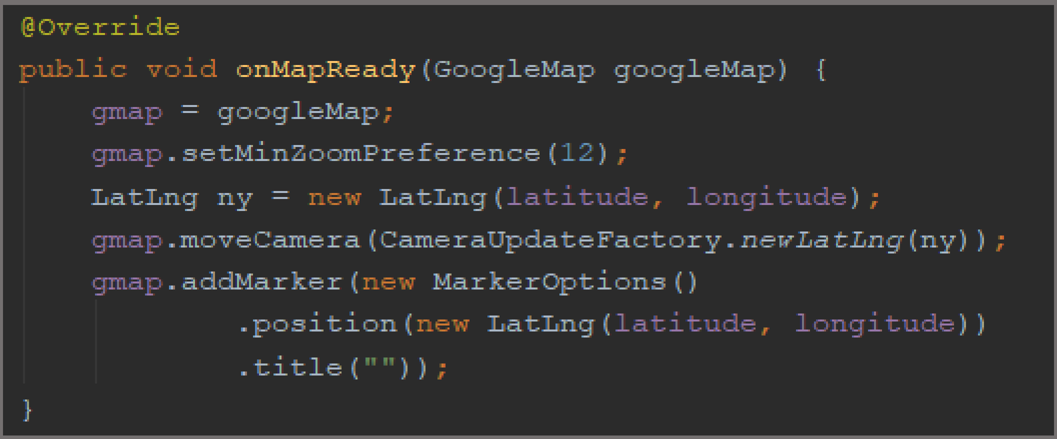
```java
for(Action a : actions){
    if(a.match().id() == match_id[0]){
        ActionLite actionLite = new ActionLite(a.id(), a.player().name(), a.team().name()
        Log.d( tag: "Inserting", actionLite.toString());
        ContentValues values = new ContentValues();
        values.put("a_id", actionLite.id());
        values.put("a_player", actionLite.player());
        values.put("a_team", actionLite.team());
        values.put("a_match", actionLite.match());
        values.put("a_score", actionLite.score());
        values.put("a_time", actionLite.time());
        values.put("a_faults", actionLite.faults());
        if(db.insert( table: "actions", nullColumnHack: null,values) == -1){
            Log.d( tag: "SQLite", msg: "Insertion Action Failed");
        }
    }
}
```

Geolocation to find address



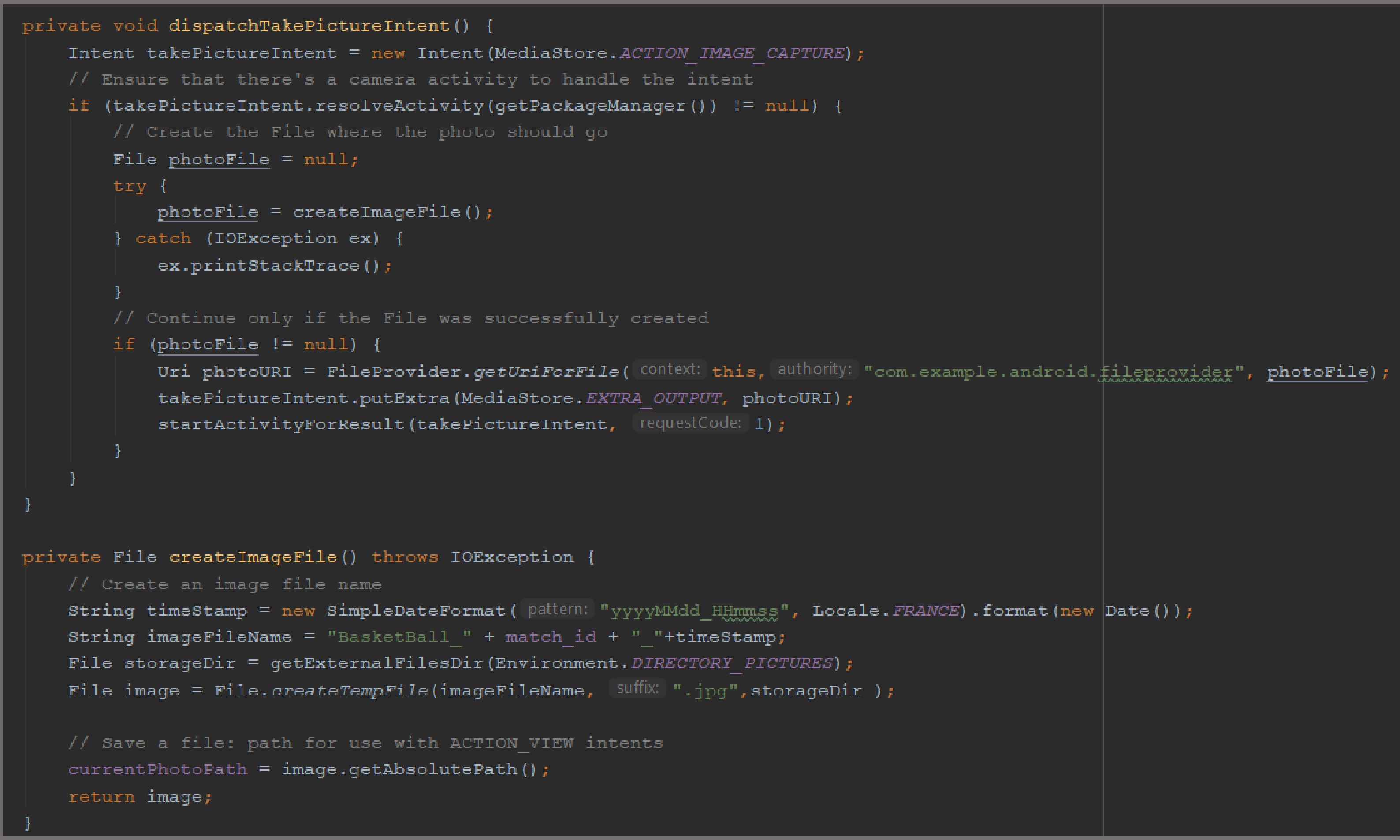
```java
try{
    Geocoder geocoder;
    List<Address> addresses;
    geocoder = new Geocoder(context, Locale.getDefault());
    addresses = geocoder.getFromLocation(latitude, longitude,  maxResults: 1);

    String address = addresses.get(0).getAddressLine( index: 0); // If any add
    String city = addresses.get(0).getLocality();
    String state = addresses.get(0).getAdminArea();
    String country = addresses.get(0).getCountryName();
    ret = address +", " + city + ", " + country;
}catch (Exception e){
    e.printStackTrace();
    ret = "Unknown Address";
}
```

Geolocation Map



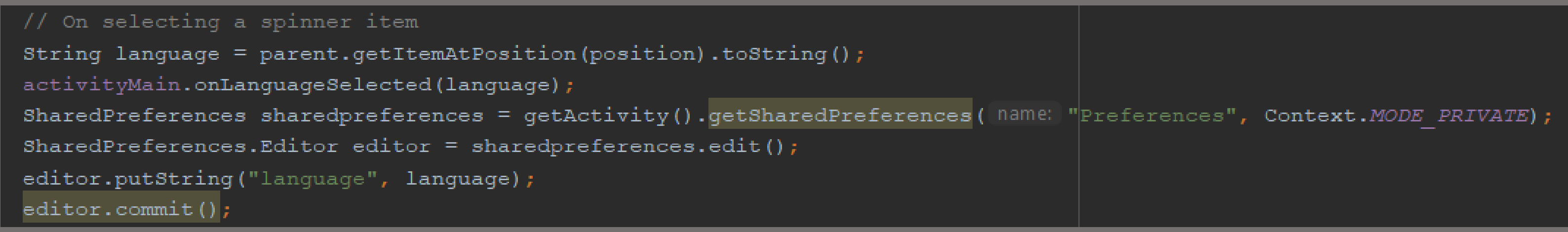
```java
@Override
public void onMapReady(GoogleMap googleMap) {
    gmap = googleMap;
    gmap.setMinZoomPreference(12);
    LatLng ny = new LatLng(latitude, longitude);
    gmap.moveCamera(CameraUpdateFactory.newLatLng(ny));
    gmap.addMarker(new MarkerOptions()
            .position(new LatLng(latitude, longitude))
            .title(""));
}
```

Take Photos



```
private void dispatchTakePictureIntent() {
    Intent takePictureIntent = new Intent(MediaStore.ACTION_IMAGE_CAPTURE);
    // Ensure that there's a camera activity to handle the intent
    if (takePictureIntent.resolveActivity(getPackageManager()) != null) {
        // Create the File where the photo should go
        File photoFile = null;
        try {
            photoFile = createImageFile();
        } catch (IOException ex) {
            ex.printStackTrace();
        }
        // Continue only if the File was successfully created
        if (photoFile != null) {
            Uri photoURI = FileProvider.getUriForFile( context: this, authority: "com.example.android.fileprovider", photoFile);
            takePictureIntent.putExtra(MediaStore.EXTRA_OUTPUT, photoURI);
            startActivityForResult(takePictureIntent,  requestCode: 1);
        }
    }
}

private File createImageFile() throws IOException {
    // Create an image file name
    String timeStamp = new SimpleDateFormat( pattern: "yyyyMMdd_HHmmss", Locale.FRANCE).format(new Date());
    String imageFileName = "BasketBall_" + match_id + "_"+timeStamp;
    File storageDir = getExternalFilesDir(Environment.DIRECTORY_PICTURES);
    File image = File.createTempFile(imageFileName,  suffix: ".jpg",storageDir );

    // Save a file: path for use with ACTION_VIEW intents
    currentPhotoPath = image.getAbsolutePath();
    return image;
}
```

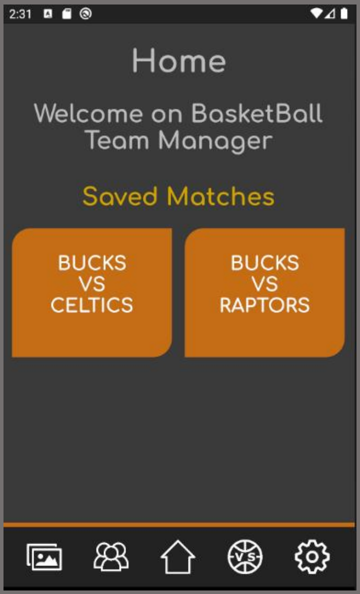
Shared Preferences Languages



```
// On selecting a spinner item
String language = parent.getItemAtPosition(position).toString();
activityMain.onLanguageSelected(language);
SharedPreferences sharedpreferences = getActivity().getSharedPreferences( name: "Preferences", Context.MODE_PRIVATE);
SharedPreferences.Editor editor = sharedpreferences.edit();
editor.putString("language", language);
editor.commit();
```

Screenshots:

Home

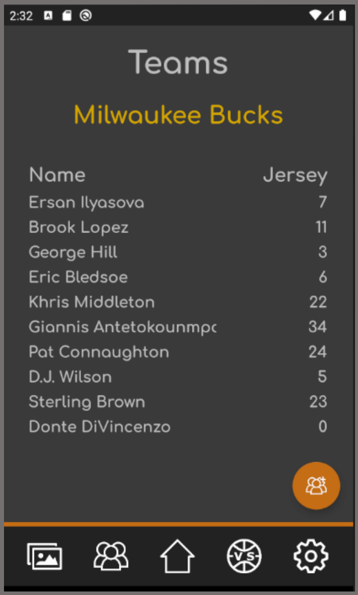


Home

Welcome on BasketBall Team Manager

Saved Matches

Teams

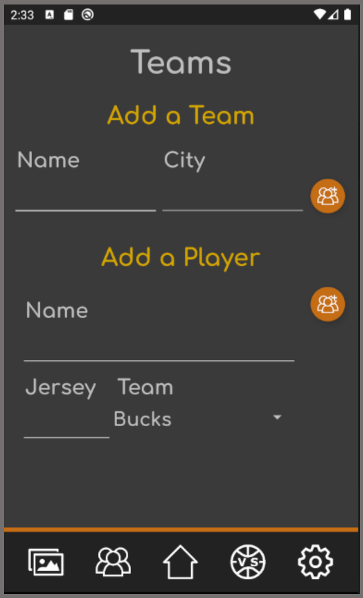


Teams

Milwaukee Bucks

Name	Jersey
Ersan Ilyasova	7
Brook Lopez	11
George Hill	3
Eric Bledsoe	6
Khris Middleton	22
Giannis Antetokounmpo	34
Pat Connaughton	24
D.J. Wilson	5
Sterling Brown	23
Donte DiVincenzo	0

Team Adder



Teams

Add a Team

Name City

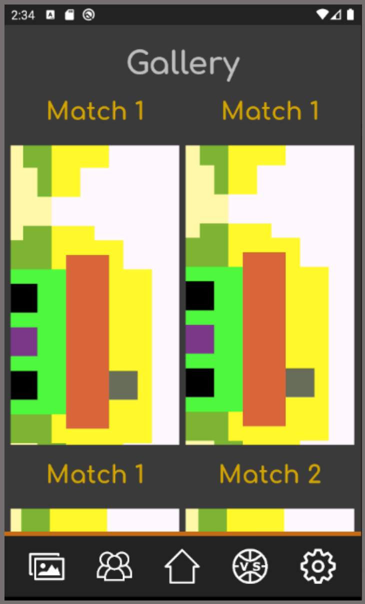
Add a Player

Name

Jersey Team

Bucks

Gallery

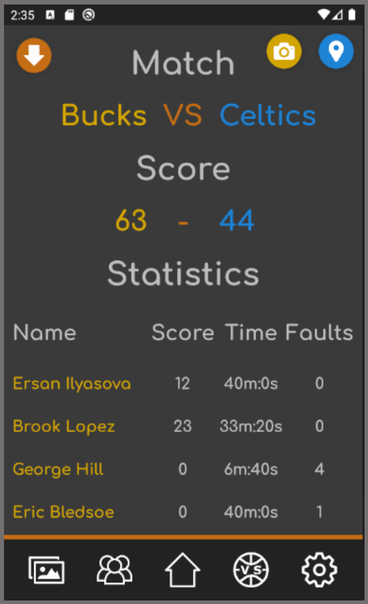


Gallery

Match 1 Match 1

Match 1 Match 2

Match Page



Match

Bucks VS Celtics

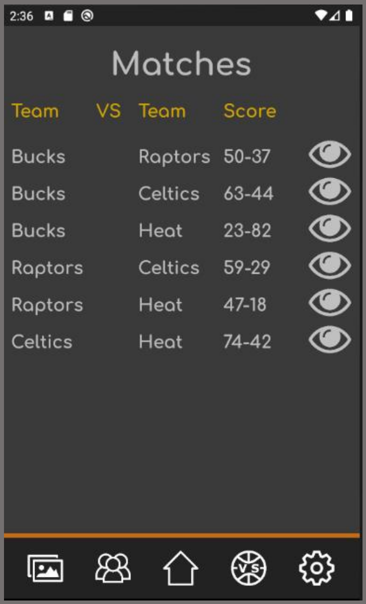
Score

63 - 44

Statistics

Name	Score	Time	Faults
Ersan Ilyasova	12	40m:0s	0
Brook Lopez	23	33m:20s	0
George Hill	0	6m:40s	4
Eric Bledsoe	0	40m:0s	1

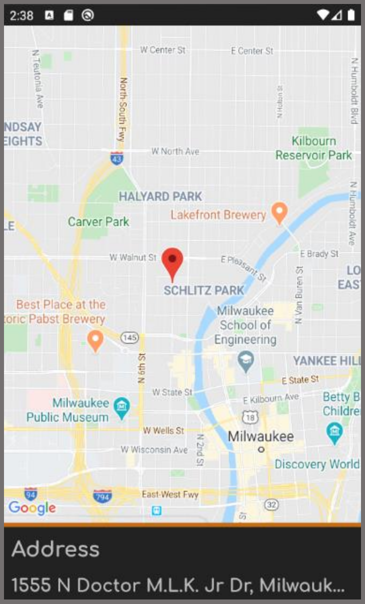
Match Menu



Matches

Team	VS	Team	Score
Bucks		Raptors	50-37
Bucks		Celtics	63-44
Bucks		Heat	23-82
Raptors		Celtics	59-29
Raptors		Heat	47-18
Celtics		Heat	74-42

Match Map



Address

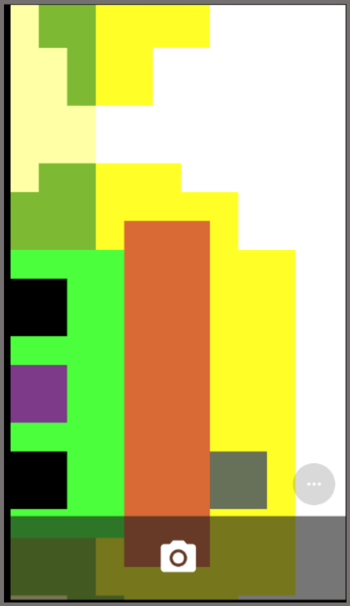
1555 N Doctor M.L.K. Jr Dr, Milwauk...

Options

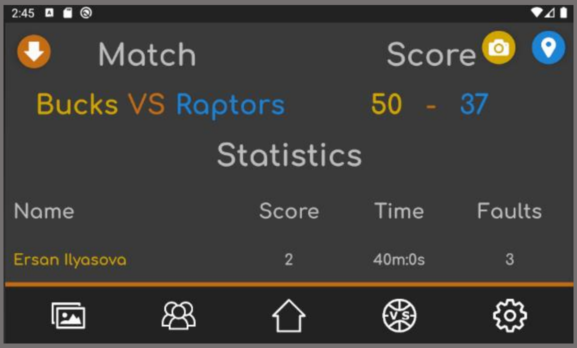
Options

English

Taking Photo



Landscape Example



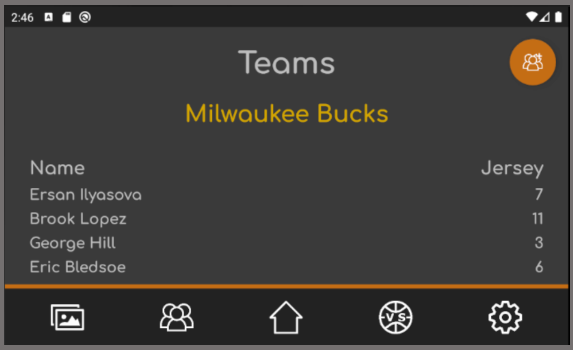
Match Score

Bucks VS Raptors 50 - 37

Statistics

Name	Score	Time	Faults
Ersan Ilyasova	2	40m:0s	3

Landscape Example



Teams

Milwaukee Bucks

Name	Jersey
Ersan Ilyasova	7
Brook Lopez	11
George Hill	3
Eric Bledsoe	6