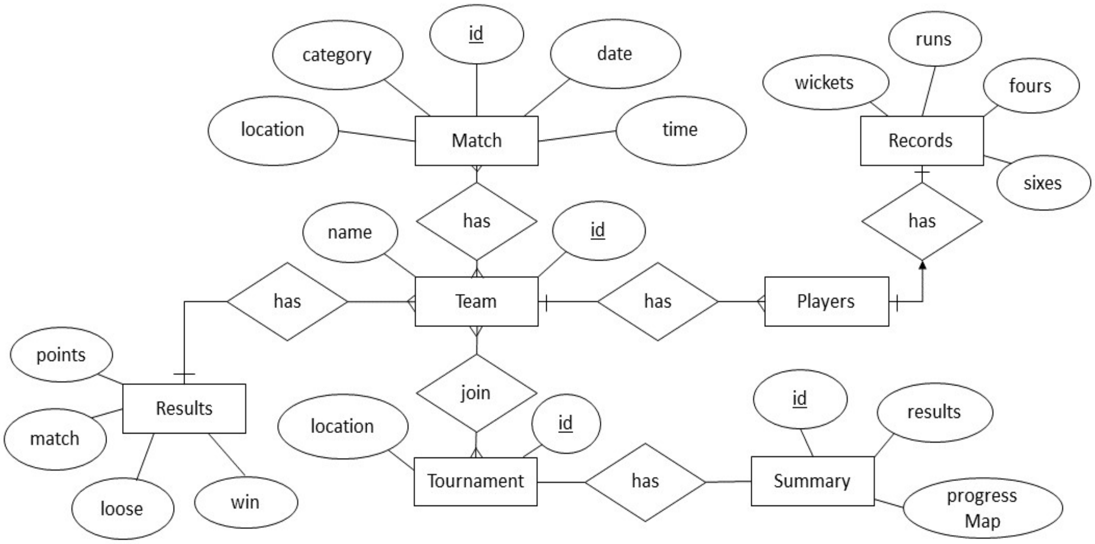
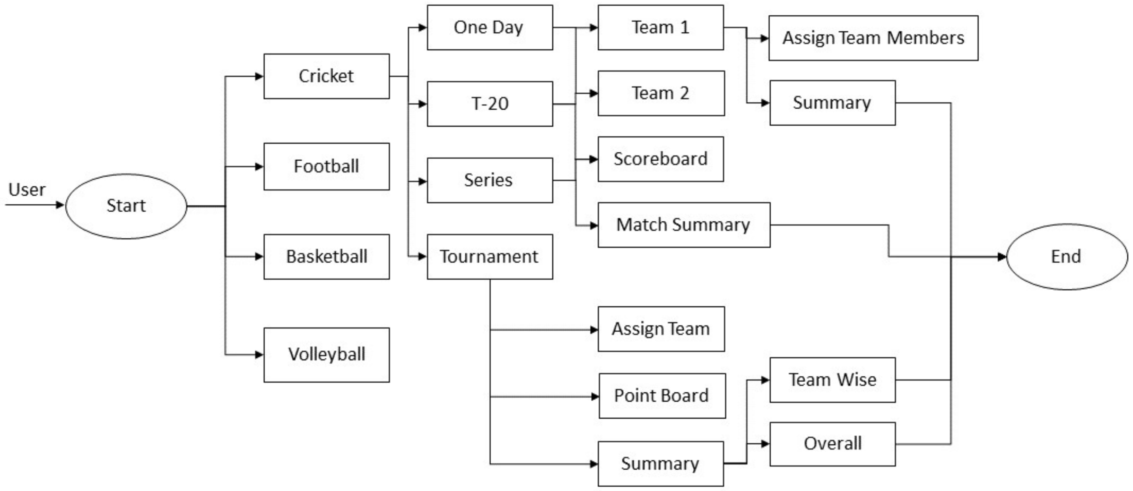
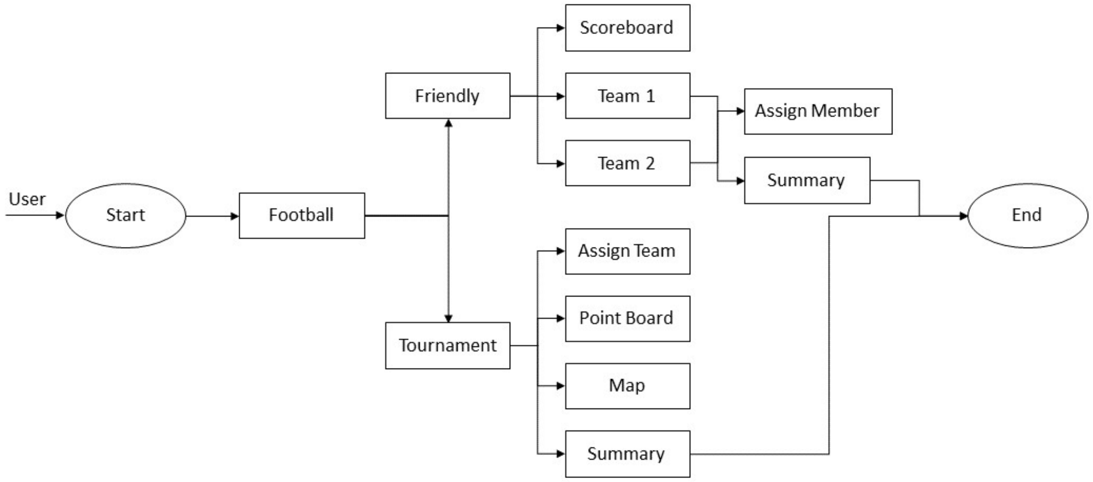
```
package com.hrtechnology.smartsports;

import android.content.ContentValues;
import android.content.Context;
import android.database.Cursor;
import android.database.sqlite.SQLiteDatabase;
import android.database.sqlite.SQLiteOpenHelper;
import android.widget.Toast;

import java.util.ArrayList;
import java.util.List;

public class DatabaseHandler extends SQLiteOpenHelper {
    private static final String DATABASE_NAME = "smartSports";
    private static final int DATABASE_VERSION = 1;
    private static final String TABLE_MATCH = "matches";
    private static final String TABLE_PLAYER= "players";
    private static final String KEY_ID = "id";
    private static final String KEY_MATCHNAME = "matches_name";
    private static final String KEY_MATCHDATE = "matchs_date";
    private static final String KEY_TEAM1RUN = "team1s_run";
    private static final String KEY_TEAM2RUN = "team2s_run";
    private static final String KEY_TOTALOVER = "total_overs";
    private static final String KEY_TEAM1NAME = "team1s_name";
    private static final String KEY_TEAM2NAME= "team2s_name";
    private static final String KEY_WINER = "winers";
    private static final String KEY_LOOSER = "loosers";
    private static final String KEY_OVER = "overs";
    private static final String KEY_TEAM1WICKET = "team1wickets";
    private static final String KEY_TEAM2WICKET = "team2wickets";
    private static final String KEY_PLAYERNAME = "players_name";
    private static final String KEY_PLAYERRUN = "players_run";
    private static final String KEY_PLAYERBALL = "players_ball";
    private static final String KEY_BATTINGOVER = "batting_overs";
    private static final String KEY_BOWLINGOVER = "bowling_overs";
    private static final String KEY_TEAMNAME = "team_name";

    private static final String CREATE_TABLE_MATCHES = "CREATE TABLE " +
TABLE_MATCH +
            "(" + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
              + KEY_MATCHNAME + " TEXT,"
              + KEY_MATCHDATE + " TEXT,"
              + KEY_TEAM1RUN  + " INTEGER,"
              + KEY_TEAM2RUN  + " INTEGER,"
              + KEY_TOTALOVER + " INTEGER,"
              + KEY_TEAM1NAME + " TEXT,"
              + KEY_TEAM2NAME + " TEXT,"
              + KEY_WINER     + " TEXT,"
              + KEY_LOOSER    + " TEXT,"
              + KEY_OVER    + " INTEGER,"
              + KEY_TEAM1WICKET + " INTEGER,"
              + KEY_TEAM2WICKET + " INTEGER );";
    private static final String CREATE_TABLE_PLAYERS = "CREATE TABLE " +
TABLE_PLAYER+
            "(" + KEY_ID + " INTEGER PRIMARY KEY AUTOINCREMENT,"
              + KEY_PLAYERNAME + " TEXT,"
              + KEY_MATCHNAME  + " TEXT,"
              + KEY_MATCHDATE  + " TEXT,"
              + KEY_TEAMNAME   + " TEXT,"
              + KEY_PLAYERRUN  + " INTEGER,"
              + KEY_PLAYERBALL + " INTEGER,"
              + KEY_BATTINGOVER+ " REAL,"
              + KEY_BOWLINGOVER+ " REAL );";

    public DatabaseHandler(Context context) {
        super(context, DATABASE_NAME, null, DATABASE_VERSION);

    }
    @Override
    public void onCreate(SQLiteDatabase db) {
        db.execSQL(CREATE_TABLE_MATCHES);
        db.execSQL(CREATE_TABLE_PLAYERS);
    }

    @Override
    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {
        db.execSQL("DROP TABLE IF EXISTS '" + TABLE_MATCH + "'");
        db.execSQL("DROP TABLE IF EXISTS '" + TABLE_PLAYER + "'");
        onCreate(db);
    }

    void addNewMatche(Match match) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_MATCHNAME, match.getMatch_Name());
        values.put(KEY_MATCHDATE, match.getMatch_Date());
        values.put(KEY_TEAM1RUN, match.getTeam1_Run());
        values.put(KEY_TEAM2RUN, match.getTeam2_Run());
        values.put(KEY_TOTALOVER, match.getTotal_Over());
        values.put(KEY_TEAM1NAME, match.getTeam1_name());
        values.put(KEY_TEAM2NAME, match.getTeam2_name());
        values.put(KEY_WINER, match.getWinner());
        values.put(KEY_LOOSER, match.getLooser());
        values.put(KEY_OVER, match.getOver());
        values.put(KEY_TEAM1WICKET, match.getTeam1wicket());
        values.put(KEY_TEAM2WICKET, match.getTeam2wicketl());

        db.insert(TABLE_MATCH, null, values);
        db.close();
    }
    void addNewPlayer(Player player) {
        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues values = new ContentValues();
        values.put(KEY_PLAYERNAME, player.getPlayer_Name());
        values.put(KEY_MATCHNAME, player.getMatch_Name());
        values.put(KEY_MATCHDATE, player.getMatch_Date());
        values.put(KEY_TEAMNAME, player.getTeam_name());
        values.put(KEY_PLAYERRUN, player.getPlayer_run());
        values.put(KEY_PLAYERBALL, player.getPlayer_Ball());
        values.put(KEY_BATTINGOVER, player.getBatting_over());
        values.put(KEY_BOWLINGOVER, player.getBowling_over());

        db.insert(TABLE_PLAYER, null, values);
        db.close();
    }
    //Get all Match
    public ArrayList<Match> getAllMatches() {
        ArrayList<Match> matches = new ArrayList<Match>();

        String selectQuery = "SELECT  * FROM " + TABLE_MATCH;
        SQLiteDatabase db = this.getWritableDatabase();
        Cursor cursor = db.rawQuery(selectQuery, null);

        if (cursor.moveToFirst()) {
            do {

              String match_name=cursor.getString(1);
              String match_date=cursor.getString(2);
              matches.add(new Match(match_name,match_date));
            } while (cursor.moveToNext());
        }
        return matches;
```

```
  }

  //Get all Players
  public ArrayList<Player> getAllPlayers(String match_name_,String match_date_,String
team_name) {
    ArrayList<Player> players = new ArrayList<Player>();


    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.query(TABLE_PLAYER,
        new String[] { KEY_PLAYERNAME,
            KEY_PLAYERRUN,
            }, KEY_MATCHNAME + "=? AND " + KEY_MATCHDATE + "=? AND " +
KEY_TEAMNAME + "=?"  ,
        new String[] {
String.valueOf(match_name_),String.valueOf(match_date_),String.valueOf(team_name) }, null,
null, null, null);

    if (cursor.moveToFirst()) {
      do {
        String player_name=cursor.getString(0);
        int player_run=Integer.parseInt(cursor.getString(1).toString());
        players.add(new Player(player_name,player_run));
      } while (cursor.moveToNext());
    }
    return players;
  }

  //Get Single Match
  public Match getMatch(String match_name, String match_Date){
    SQLiteDatabase db = this.getReadableDatabase();

    Cursor cursor = db.query(TABLE_MATCH,
        new String[] { KEY_MATCHNAME,
                KEY_MATCHDATE,
                KEY_TEAM1RUN,
                KEY_TEAM2RUN,
                KEY_TOTALOVER,
                KEY_TEAM1NAME,
                KEY_TEAM2NAME,
                KEY_WINER,
                KEY_LOOSER,
                KEY_OVER,
                KEY_TEAM1WICKET,
                KEY_TEAM2WICKET}, KEY_MATCHNAME + "=? AND " +
KEY_MATCHDATE + "=?" ,
                new String[] { String.valueOf(match_name),String.valueOf(match_Date) }, null, null,
null, null);
        if (cursor != null)
            cursor.moveToFirst();
        Match match = new Match(cursor.getString(0),
                                cursor.getString(1),
                                Integer.parseInt(cursor.getString(2)),
                                Integer.parseInt(cursor.getString(3)),
                                Integer.parseInt(cursor.getString(4)),
                                cursor.getString(5),
                                cursor.getString(6),
                                cursor.getString(7),
                                cursor.getString(8),
                                Integer.parseInt(cursor.getString(9)),
                                Integer.parseInt(cursor.getString(10)),
                                Integer.parseInt(cursor.getString(11)));

        return match;
    }

    //Get Single pLAYER
    public Player getPlayer(String match_name, String match_Date, String team_name, String
player_name){
        SQLiteDatabase db = this.getReadableDatabase();

        Cursor cursor = db.query(TABLE_PLAYER,
                new String[] { KEY_PLAYERNAME,
                        KEY_MATCHNAME,
                        KEY_MATCHDATE,
                        KEY_TEAMNAME,
                        KEY_PLAYERRUN,
                        KEY_PLAYERBALL,
                        KEY_BATTINGOVER,
                        KEY_BOWLINGOVER}, KEY_MATCHNAME + "=? AND " +
KEY_MATCHDATE + "=? AND " + KEY_TEAMNAME + "=? AND " +
KEY_PLAYERNAME + "=?" ,
                new String[] {
String.valueOf(match_name),String.valueOf(match_Date),String.valueOf(team_name),String.val
ueOf(player_name) }, null, null, null, null);
        if (cursor != null)
            cursor.moveToFirst();
        Player player = new Player(cursor.getString(0),
                cursor.getString(1),
                cursor.getString(2),
                cursor.getString(3),
        Integer.parseInt(cursor.getString(4)),
        Integer.parseInt(cursor.getString(5)),
        Float.valueOf((cursor.getString(6)).toString()),
        Float.valueOf((cursor.getString(7)).toString()));
    return player;
  }
  //Updatea score
  public int updateMatchScore(Match match) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_TEAM1RUN, match.getTeam1_Run());
    values.put(KEY_TEAM2RUN, match.getTeam2_Run());


    // updating row
    return db.update(TABLE_MATCH, values,  KEY_MATCHNAME+ " = ? AND " +
KEY_MATCHDATE + "=?",
        new String[] { match.getMatch_Name(),match.getMatch_Date() });
  }

  //Updatea Over
  public int updateMatchOver(Match match) {
    SQLiteDatabase db = this.getWritableDatabase();

    ContentValues values = new ContentValues();
    values.put(KEY_OVER, match.getOver());



    // updating row
    return db.update(TABLE_MATCH, values,  KEY_MATCHNAME+ " = ? AND " +
KEY_MATCHDATE + "=?",
        new String[] { match.getMatch_Name(),match.getMatch_Date() });
  }


}
```


category
id
date
location
Match
time
wickets
runs
fours
Records
sixes
has
has
name
id
has
Team
has
Players
points
join
Results
location
id
id
results
match
Tournament
has
Summary
progress Map
loose
win


User
Start
Cricket
Football
Basketball
Volleyball
One Day
T-20
Series
Tournament
Team 1
Team 2
Scoreboard
Match Summary
Assign Team Members
Summary
End
Assign Team
Point Board
Summary
Team Wise
Overall


User
Start
Football
Friendly
Tournament
Scoreboard
Team 1
Team 2
Assign Member
Summary
End
Assign Team
Point Board
Map
Summary