
STATS IN GRASS
FINAL PRESENTATION
YUSUFBEK KARAMATOV . HUSNU ARKCA . MUSTAFA AKSELI . YUNUS EMRE ASLAN

STATS IN GRASS

MAIN OBJECTIVE:

THE PRIMARY OBJECTIVE OF THE STATS IN GRASS PROJECT IS TO DEVELOP A DESKTOP APPLICATION DESIGNED TO MANAGE AND DISPLAY FOOTBALL MATCH STATISTICS. THIS APPLICATION AIMS TO FACILITATE THE REAL-TIME RECORDING AND VIEWING OF MATCH DATA, ENSURING ACCURACY AND EFFICIENCY FOR USERS.

YUSUFBEK KARAMATOV . HUSNU ARKCA . MUSTAFA AKSELI . YUNUS EMRE ASLAN

STATS IN GRASS

KEY FEATURES:

- USER FRIENDLY INTERFACES
- REAL-TIME DATA HANDLING
- DATABASE INTEGRATION
- ERROR HANDLING AND DATA CORRECTION

YUSUFBEK KARAMATOV . HUSNU ARKCA . MUSTAFA AKSELI . YUNUS EMRE ASLAN

STATS GRASS IN

WHAT WE USED?

YUSUFBEK KARAMATOV . HUSNU ARKCA . MUSTAFA AKSELI . YUNUS EMRE ASLAN

STATS IN GRASS

WHO WORKED ON WHAT?

BACKEND

HUSNU ARKCA

YUSUFBEK KARAMATOV

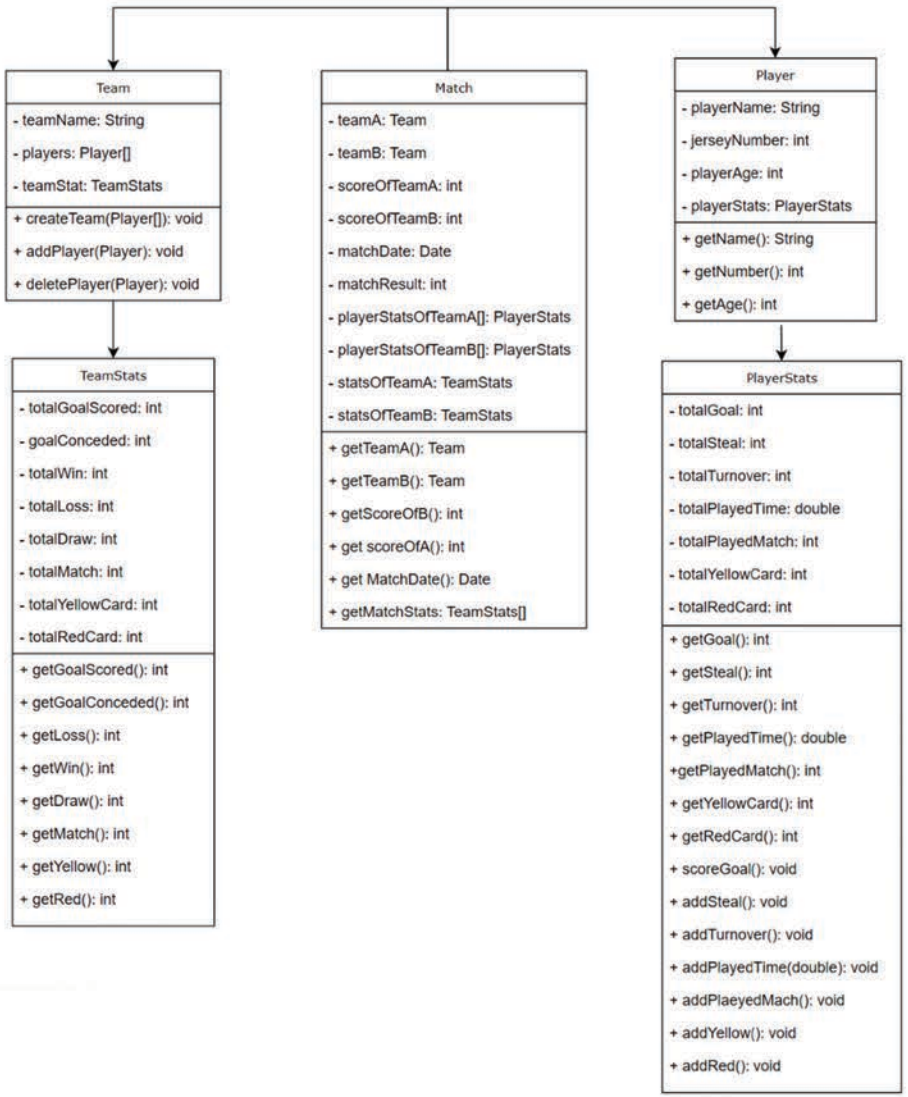
FRONTEND

MUSTAFA AKSELI

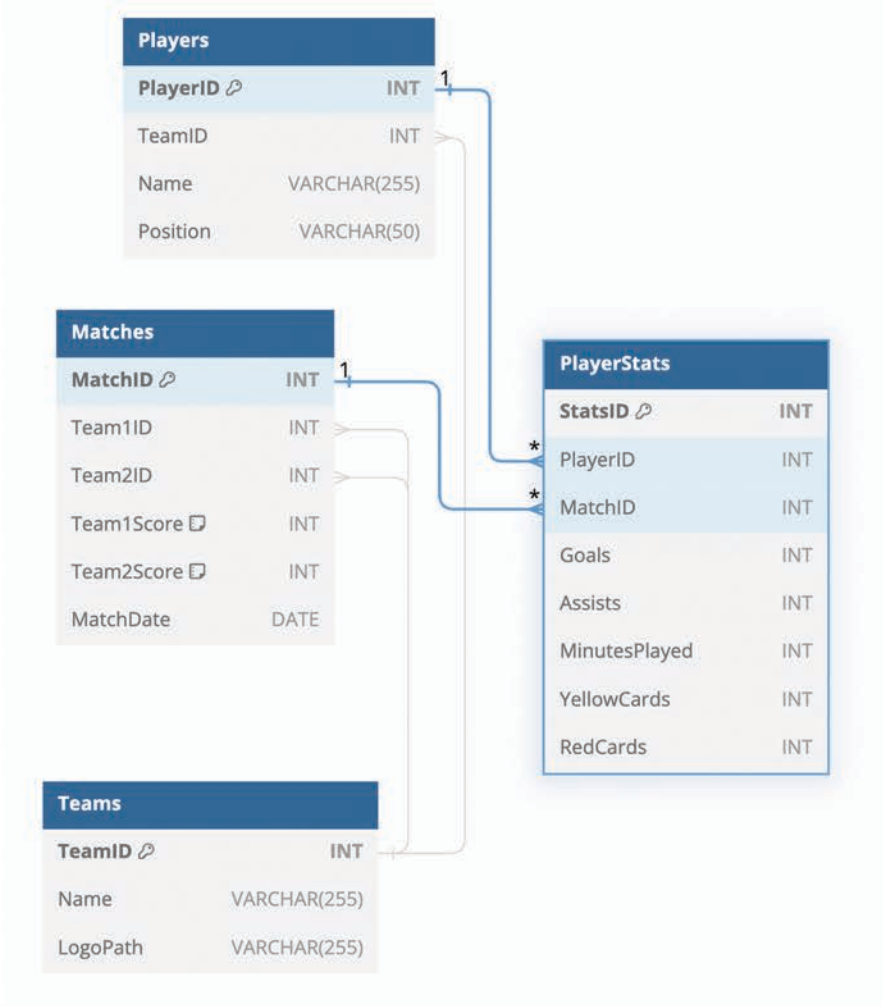
YUNUS EMRE

YUSUFBEK KARAMATOV . HUSNU ARKCA . MUSTAFA AKSELI . YUNUS EMRE ASLAN

UML DIAGRAM



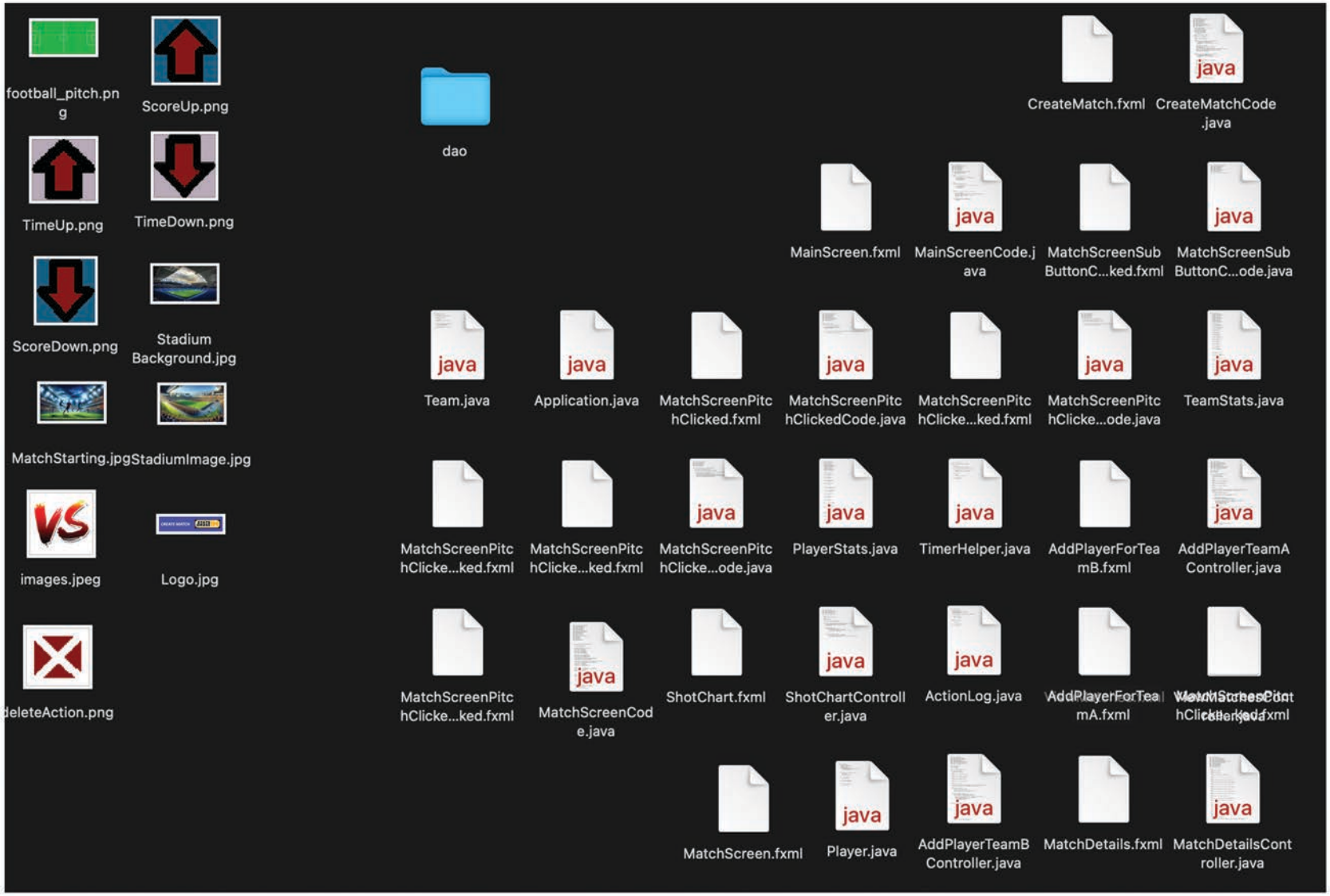
DB DIAGRAM



YUSUFBEK KARAMATOV . HUSNU ARKCA . MUSTAFA AKSELI . YUNUS EMRE ASLAN

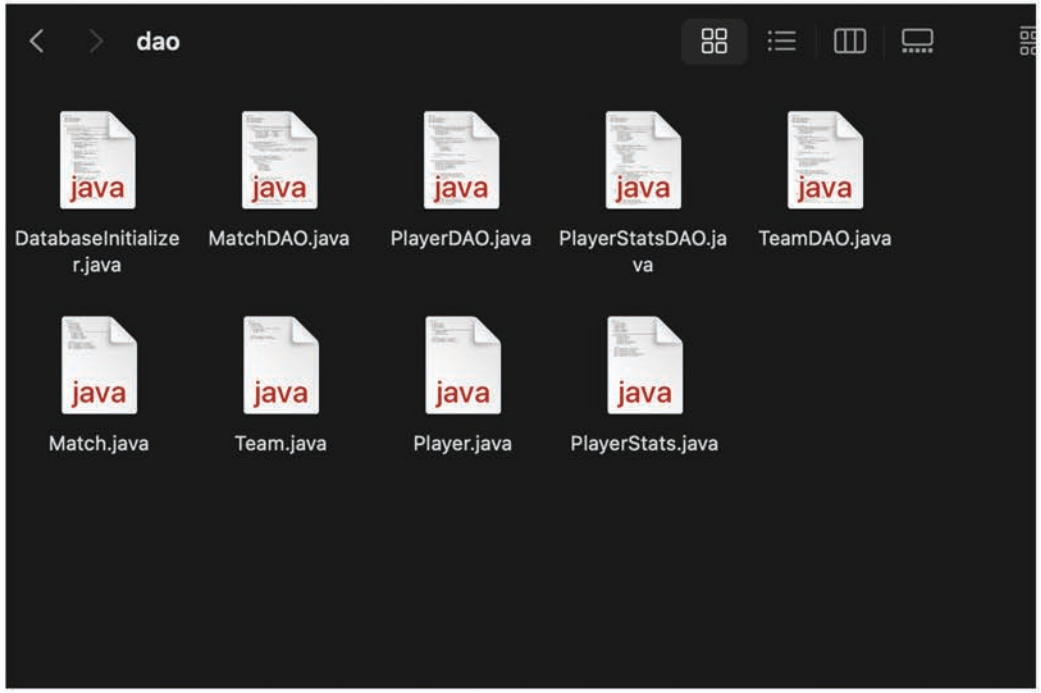
WHAT IS SPECIAL ABOUT STATS_IN_GRASS DB

DAO - DATA ACCESS OBJECT. IT IS A DESIGN PATTERN USED IN SOFTWAR ENGINEERING TO ABSTRACT AND ENCAPSULATE ALL ACCESS TO A DATA SOURCE



WHAT IS SPECIAL ABOUT STATS_IN_GRASS DB

DAO - DATA ACCESS OBJECT. IT IS A DESIGN PATTERN USED IN SOFTWAR ENGINEERING TO ABSTRACT AND ENCAPSULATE ALL ACCESS TO A DATA SOURCE



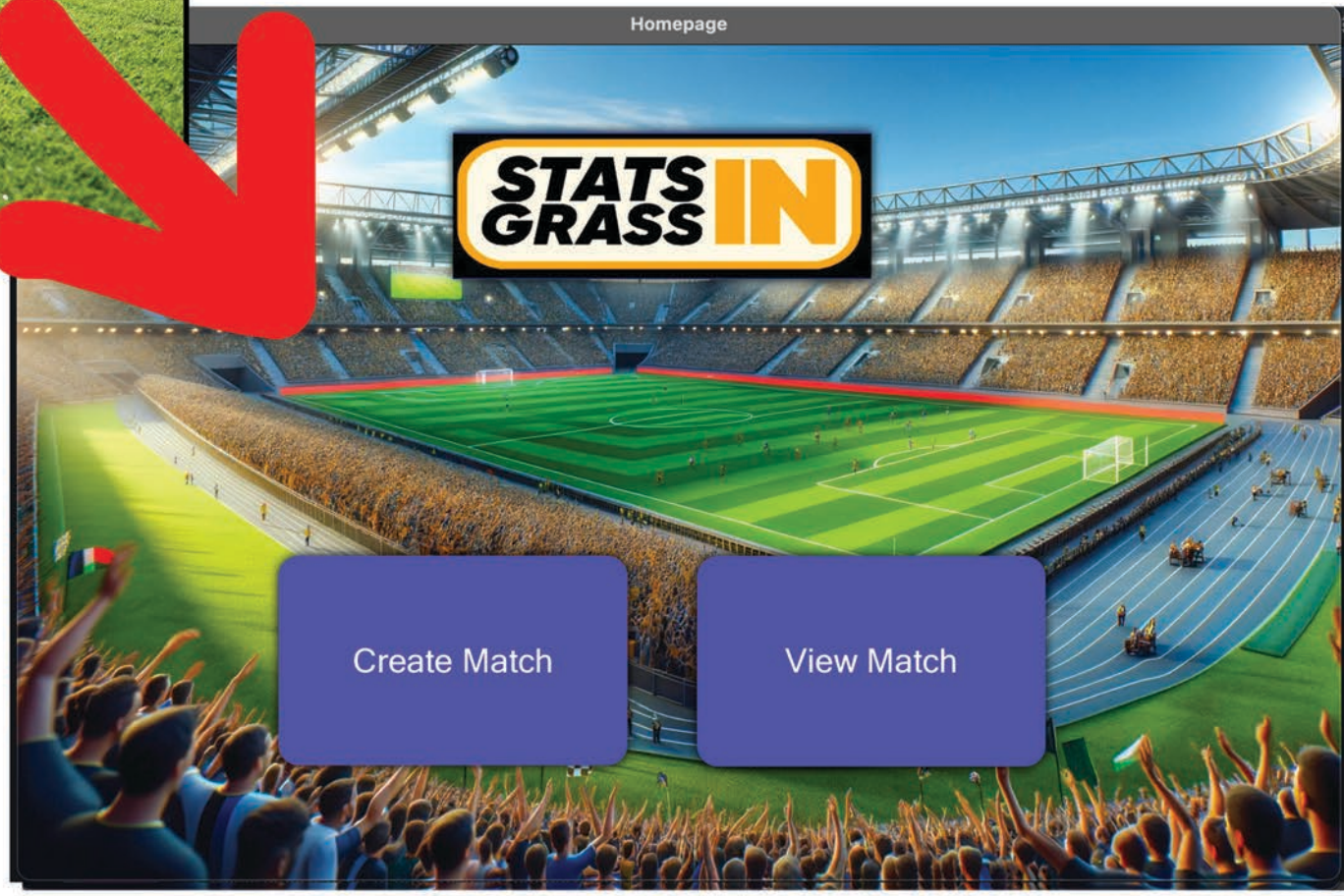

```java
package dao;

import java.sql.Connection;
import java.sql.PreparedStatement;
import java.sql.ResultSet;
import java.sql.SQLException;
import java.util.ArrayList;


public class TeamDAO {
    public void addTeam(String name, String logoPath) throws SQLException {
        String sql = "INSERT INTO Teams (Name, LogoPath) VALUES (?, ?)";
        try (Connection conn = DatabaseInitializer.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setString(parameterIndex:1, name);
            stmt.setString(parameterIndex:2, logoPath);
            stmt.executeUpdate();
        }
    }
    public Team getTeam(int teamId) {
        String sql = "SELECT * FROM Teams WHERE TeamID = ?";
        Team team = null;
        try (Connection conn = DatabaseInitializer.getConnection();
             PreparedStatement stmt = conn.prepareStatement(sql)) {
            stmt.setInt(parameterIndex:1, teamId);
            ResultSet rs = stmt.executeQuery();
            if (rs.next()) {
                team = new Team(
                    rs.getInt(columnLabel:"TeamID"),
                    rs.getString(columnLabel:"Name"),
                    rs.getString(columnLabel:"LogoPath")
                );
            }
        } catch (SQLException e) {
            System.out.println("Database error: " + e.getMessage());
        }
        return team;
    }
```

WHAT WE ACHIEVED GOOD?

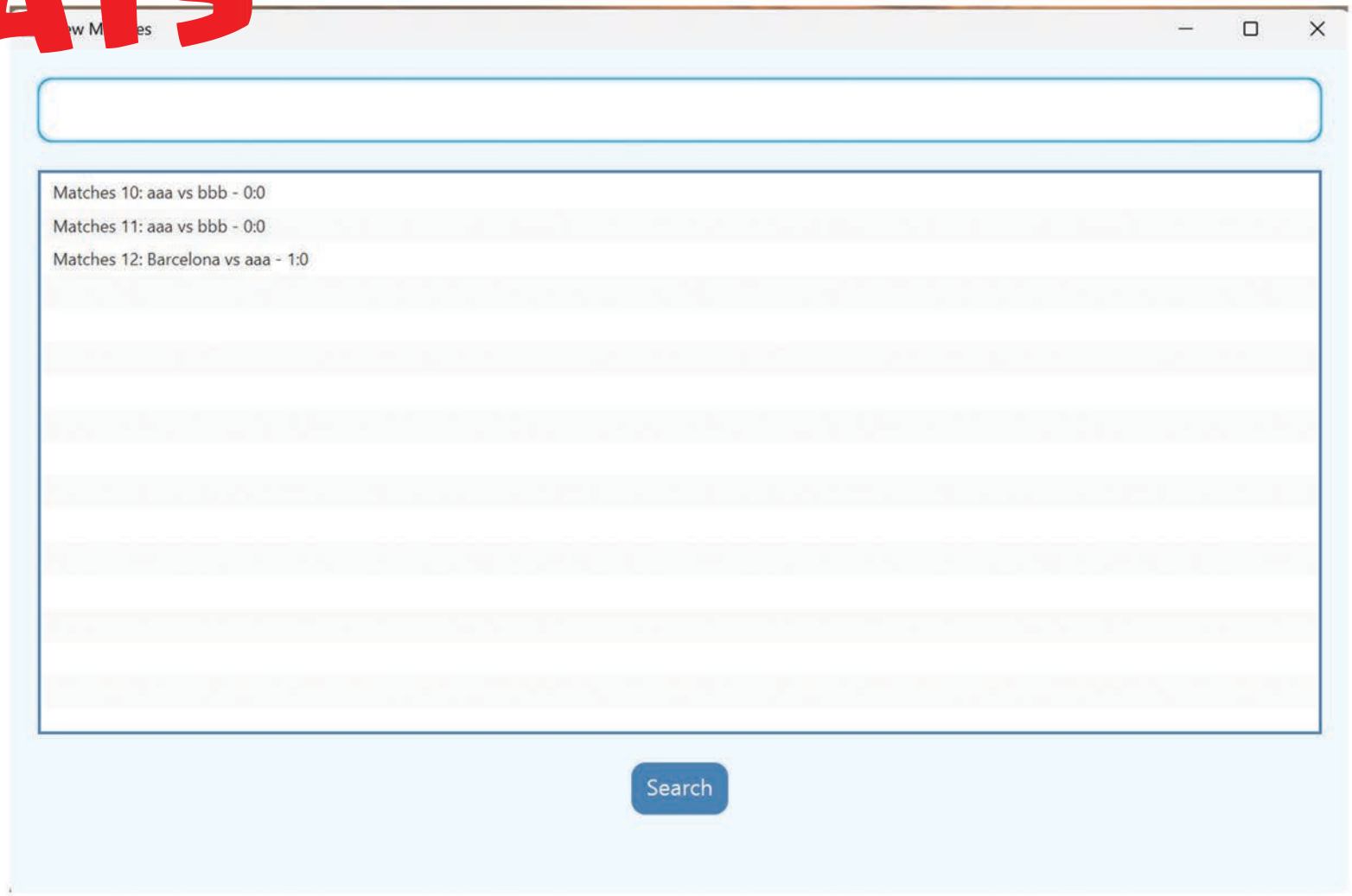
DESIGN IS NEARLY THE SAME



UML & DB IS BETTER

WHAT WE COULD NOT ACHIEVE SO GOOD?

PAST STATS




STATS
GRASS
IN