<Goal Score>

Analysis and Design Document

Student: Ciuca Daniel Claudiu

**Group: 30238**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| <dd/mmm/yy> | <x.x> | <details> | <name> |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

Table of Contents

I. Project Specification 4

II. Elaboration – Iteration 1.1 4

1. Domain Model 4

2. Architectural Design 4

2.1 Conceptual Architecture 4

2.2 Package Design 4

2.3 Component and Deployment Diagrams 4

III. Elaboration – Iteration 1.2 4

1. Design Model 4

1.1 Dynamic Behavior 4

1.2 Class Design 4

2. Data Model 4

3. Unit Testing 4

IV. Elaboration – Iteration 2 4

1. Architectural Design Refinement 4

2. Design Model Refinement 4

V. Construction and Transition 5

1. System Testing 5

2. Future improvements 5

VI. Bibliography 5

# Project Specification

# Project name is GoalScore, the main idea of this project is to create a football score application for more users. An user will be able to create an account, view scores of him favorite team or other teams, view rankings of different leagues or view upcoming matches, top scorers and teams form.

# Elaboration – Iteration 1.1

# Domain Model

An user will be able to select many favorite teams, not only one.

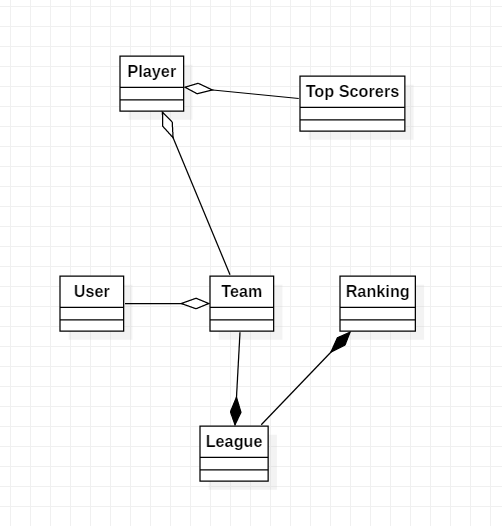
A league will be created from 1 or more teams.

A team is made of 15 or more players.

A player will be part of a single team.

Rank represent the standing of the teams in a league.

Top Scorers represent the standing of the players in a league by the number of goals.



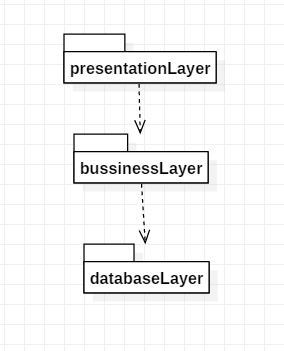
# Architectural Design

## Conceptual Architecture

The architecture I choose to use is a Layered Architecture. The reasons why I choose this architecture is because of advantages:

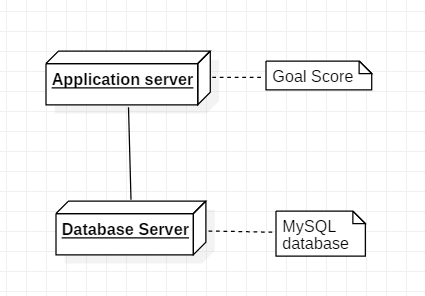
* **Simplicity**: The concept of layered architecture is easy to learn and implement.
* **Consistency**: The layers along with the overall code organization is consistent across all the layered projects.
* **Browsability**: All the objects are kept together. So, when you need to change something in some or all the objects of a particular type, it is easier to quickly find an object.

## Package Design

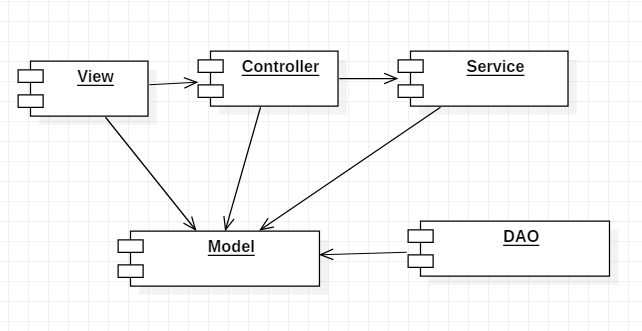


## Component and Deployment Diagrams

Deployment



Component



# Elaboration – Iteration 1.2

# Design Model

## Dynamic Behavior

*[Create the interaction diagrams (1 sequence, 1 communication diagrams) for 2 relevant scenarios]*

## Class Design

*[Create the UML class diagram; apply GoF patterns and motivate your choice]*

# Data Model

*[Create the data model for the system.]*

# Unit Testing

*[Present the used testing methods and the associated test case scenarios.]*

# Elaboration – Iteration 2

# Architectural Design Refinement

*[Refine the architectural design: conceptual architecture, package design (consider package design principles), component and deployment diagrams. Motivate the changes that have been made.]*

# Design Model Refinement

## *[Refine the UML class diagram by applying class design principles and GRASP; motivate your choices. Deliver the updated class diagrams.]*

# Construction and Transition

# System Testing

The system will be tested using unit tests and integration test, I will use spring boot test framework to load data in an in-memory database and test all possible scenarios when user try to add a team in favorites that already is in him favorite teams. I will also use mocks for tests to see if expected methods are called.

# Future improvements

A future improvement will be to add more sports than football.

# Bibliography