Volleyball Manager

Analysis and Design Document

Student: Coca Sergiu

**Group: 30233**

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 06/04/2017 | 1.0 | <details> | Coca Sergiu |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

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

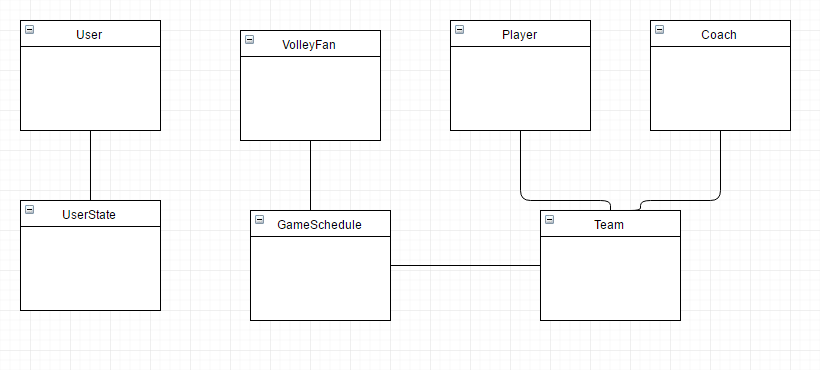
# Project Specification

This application it’s a perfect one for volleyball fans but also it’s useful for coaches. This application provide informations about a volleyball player like height, wheight, play team, etc. On the other hand if a coach want to transfer a player he can search him and if it’s available he can make an offer. When a player change his status to avaible, interested coaches are notified.

# Elaboration – Iteration 1.1

# Domain Model

The domain model should contain the following classes:



# Architectural Design

## Conceptual Architecture

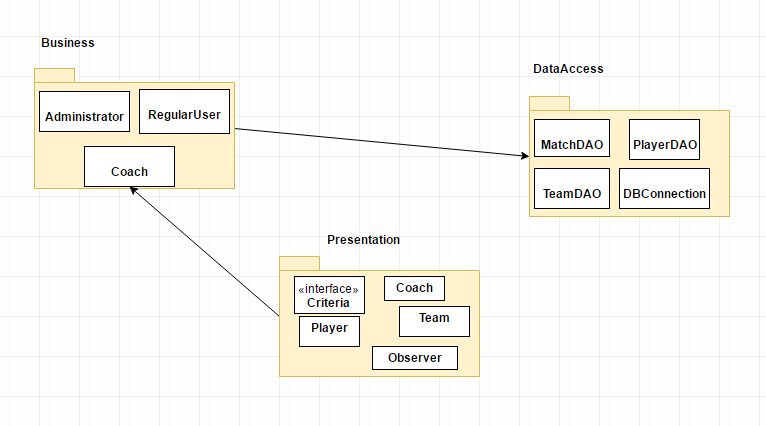
Three-tier applications have been common since the growth of the database. A three-tier system satisfies the need for implementation isolation. Most frequently, this is desirable in any system where the storage/database layer of an application may need to be changed. However, this technological isolation is not restricted to just databases. It can, and should, be used whenever it is valuable to share code without requiring the application developer, or more importantly, the application maintainer, to have a detailed understanding of the implementation details of the lowest layer.

The application will be structured using the Layers architectural pattern and will also include MVC pattern

The Layers pattern will structure the application in four major modules:

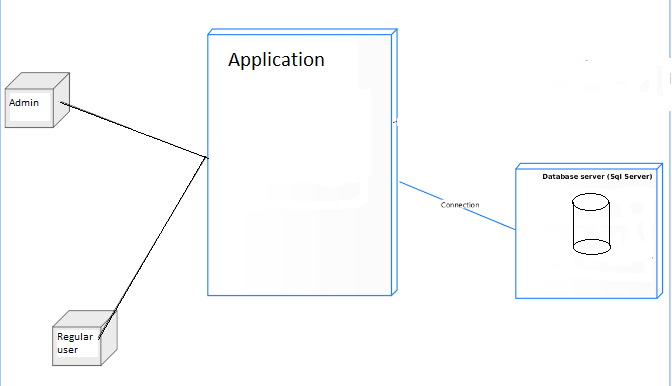
* Business Layer – it will include all the functionality the application needs
* Presentation Layer – this layer will include everything that can be seen in a interface by the user
* DAO Layer – this layer will be responsible with the database access and CRUD operations requested by the business layer

## Package Design

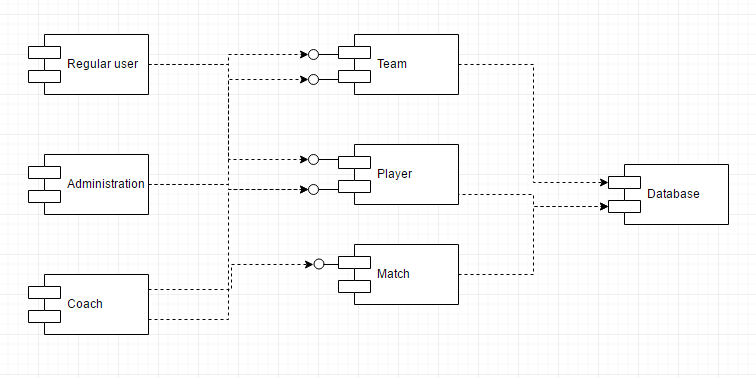


## Component and Deployment Diagrams

**Deployment**



**Component diagram**

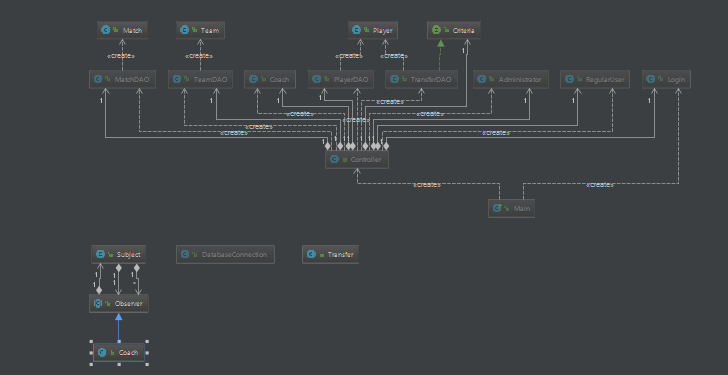
****

# Elaboration – Iteration 1.2

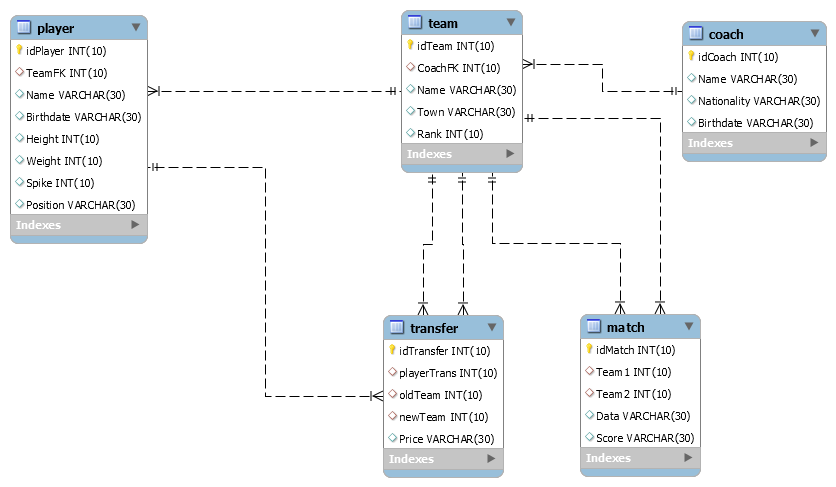
# Design Model

## Dynamic Behavior

## Class Design



# Data Model



A database model is a specification describing how a database is structured and used. Several such models have been suggested. Common models include:

-[Flat model](https://en.wikipedia.org/wiki/Flat_file_database)

-[Hierarchical model](https://en.wikipedia.org/wiki/Hierarchical_model)

-[Network model](https://en.wikipedia.org/wiki/Network_model)

-[Relational model](https://en.wikipedia.org/wiki/Relational_model)

-[Object-relational model](https://en.wikipedia.org/wiki/Object-relational_model)

-[Star schema](https://en.wikipedia.org/wiki/Star_schema):

The simplest style of data warehouse schema. The star schema consists of a few "fact tables" (possibly only one, justifying the name) referencing any number of "dimension tables". The star schema is considered an important special case of the [snowflake schema](https://en.wikipedia.org/wiki/Snowflake_schema).

# Unit Testing

# Elaboration – Iteration 2

# Architectural Design Refinement

# Design Model Refinement

# Construction and Transition

# System Testing

For login operation I checked if the username and password exist in database. Otherwise an error message occurred: “Invalid username or password, please try again”.

Another test was if a searched player or team does not exist in database a message appears in interface.

# Future improvements

A improvement can be to add more operations.

# Bibliography

<https://www.tutorialspoint.com/design_pattern/observer_pattern.htm>

<https://stackoverflow.com/>

<https://www.tutorialspoint.com/design_pattern/filter_pattern.htm>