###### **Sports Franchise**

Software Requirements Document

Version: 1.0

Date: 11/20

###### Prepared by Joseph DeMario, Frank Palaia, Joseph Tooker

**Table of Contents**

[1. Preface 4](#_heading=h.1ksv4uv)

[1.1. Modification History 4](#_heading=h.44sinio)

[1.2. Intended Audience 4](#_heading=h.2jxsxqh)

[2. Introduction 4](#_heading=h.z337ya)

2.1 Purpose of the requirements document 4

2.2 Scope of the product 5

2.3 Reference 5

2.4 Overview of the remainder of the document 5

[3. Glossary](#_heading=h.3j2qqm3) 5

[4. User Requirement Definition](#_heading=h.tyjcwt) 6

[5. System Requirements Specification](#_heading=h.3dy6vkm) 6

[6. System Architecture](#_heading=h.1y810tw) 7

[7. System Models](#_heading=h.4i7ojhp) 8

[7.1. Use Case Diagram](#_heading=h.2xcytpi) 8

[7.2. Sequence diagram](#_heading=h.17dp8vu) 8

[7.3. Class diagram](#_heading=h.3rdcrjn) 9

[8. System Evolution](#_heading=h.1ci93xb) 9

[9. Appendices](#_heading=h.3whwml4) 10

[10. Index](#_heading=h.2bn6wsx) 10

**Table of Figures**

[Figure 7.1 Use Case Model 8](#_heading=h.qsh70q)

[Figure 7.2 Sequence Model 8](#_heading=h.3as4poj)

[Figure 7.3 Class Diagram](#_heading=h.1pxezwc) 9

# **Preface**

This is version 1.0 of the software requirements document. Teams, players, managers and agents are able to easily view a web based application that displays the appropriate information pulled from our sports franchise database. From this application, managers can organize and set up meetings with agents, add players to a team after meeting with said player, agents can view their assigned players and vice versa, and teams can pay players the appropriate amount of money based on their payroll.

## **Modification History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Author** | **Summary of changes** |
| 11/18/2020 | V1.0 | Joseph DeMario, Joseph Tooker, and Frank Palaia | Started document |
|  |  |  |  |
|  |  |  |  |

## **Intended Audience**

This document is intended for players, managers, teams and agents to better a sports franchise.

# **Introduction**

Utilizing Microsoft Visual Studio IDE incorporates class libraries into web applications, where the DLL component fetches data from our Microsoft Access database. The class library will define the function to be performed by the system and will be referenced in the web application as an outside third party source, created separately from the system itself. The object data source created by the class library will be displayed via the ASP.NET framework gridview design. To fulfill business objectives, the system will also handle additional functions to add to list objects as the organization wishes to build a roster for the team. Dynamic paging will ensure organization to keep track of finances to ensure affordability and structure of the team.

## **Purpose of the requirements document**

Software requirement document will log key and vital steps towards the completion of the project. Each version update will help identify major changes done to the project within the time constraint as well as set new goals or change them entirely to adhere to project principles and functionality.

## **Scope of the product**

Product will be an improvement of midterm in which the personnel to be added to the organization’s team was displayed via the web application. Users will now have the ability to manage a team and simulate managerial tasks. An organization will start off with a predefined base payroll and will hire personnel accordingly with regards to remaining payroll after each transaction is complete. Product will effectively manage money and communication amongst agents to ensure success of sports organization.

## **Reference**

System will reference the DLL created by the class library as well as the Access Database to create a data object source to be displayed.

## **Overview of the** remainder **of the document**

Remainder of the document will describe system requirements and specifications. System architecture will explain how different system components will interact and work together. System models will exemplify theoretical scenarios and practices that the system will be expected to handle. Additionally, technicalities and user activities to be performed with the system are predefined. System Evolution will explain future aspirations and goals set for the project to be the best version released to users.

# **Glossary**

*Define technical terms used in the document.*

|  |  |
| --- | --- |
| **Term** | **Meaning** |
|  |  |
| Product users | These are users who interact with the system directly. For instance, employees- supervisors, project managers, project engineers and admin. |

# **User Requirement Definition**

***User Requirements***

*● Allow managers to contact agents and schedule meeting*

*● Allow managers to add players after they meet with agents*

*● Allow teams to manage their payroll by paying players the appropriate amount of money*

*● Allow agents to view their assigned players*

*● Allow players to view their assigned agent*

# **System Requirements Specification**

***System Requirements***

*● Available personnel are referenced from Access database*

*● Implementation of reservation system*

*● Implementation of shopping cart system*

*● Management of payroll*

*● Must include one business object (DLL component)*

*● Must include one web service component that allows end users to interact with the system for certain business functions*

***Functional Requirements:***

*● Only managers can schedule meeting with agents*

*● Can only meet agents upon approval*

*● Can only hire personnel if salary is within remaining/available payroll*

*● Can only hire personnel if they are available*

*● Can only hire/pay personnel once*

***Un-functional Requirements:***

*● Each personnel has their own agent*

*● Personnel, agents, amd department have unique IDs ,*

*● Agents may represent one or more personnel*

*● Should hire at least one coach to be considered team*

*● Managers can renegotiate personnel salaries*

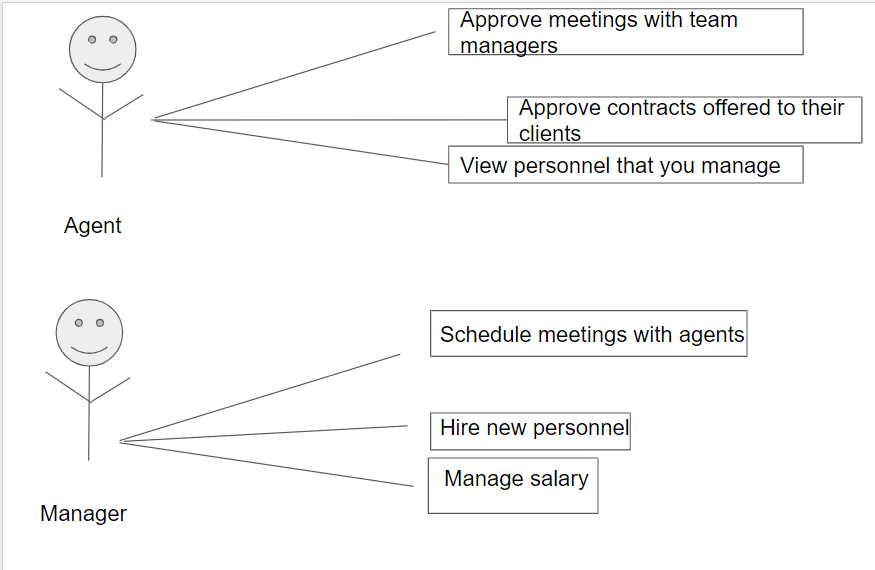
# **System Architecture**

The anticipated architecture system of our project is one that is multi-layered. We believe this architecture model best fits our sports franchise project as it is the required model needed to complete this project. In other words, a multilayered architecture model is a client server application in which visual display, application processing and data management are separated. In our case, the visual display is the web application that appears when running our code. The application processing is the chunk of the project that makes the functionality possible, or the code needed to connect the database and the functions needed to add components to the web application. The data management is the sports franchise database and how that database changes over the course of the project.

*Present the high-level overview of the anticipated system architecture. Architectural component that are reused should be highlighted.*

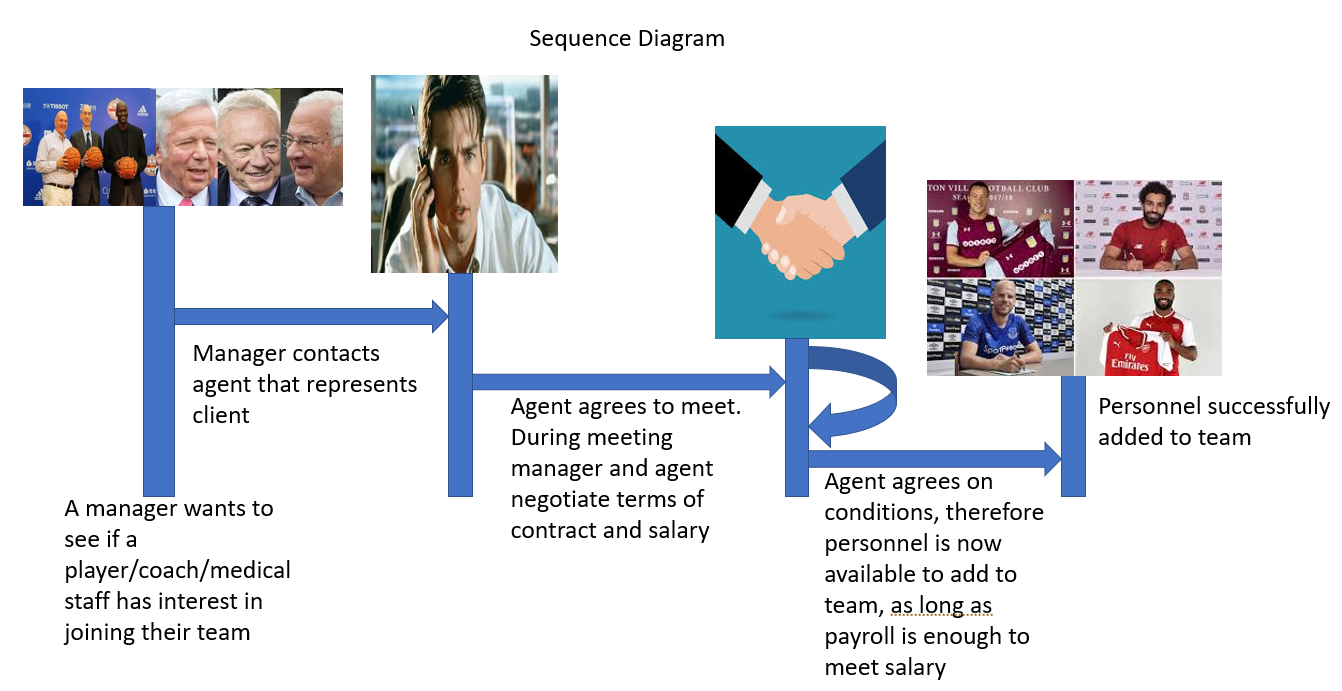
# **System Models**

## **Use Case Diagram**

**

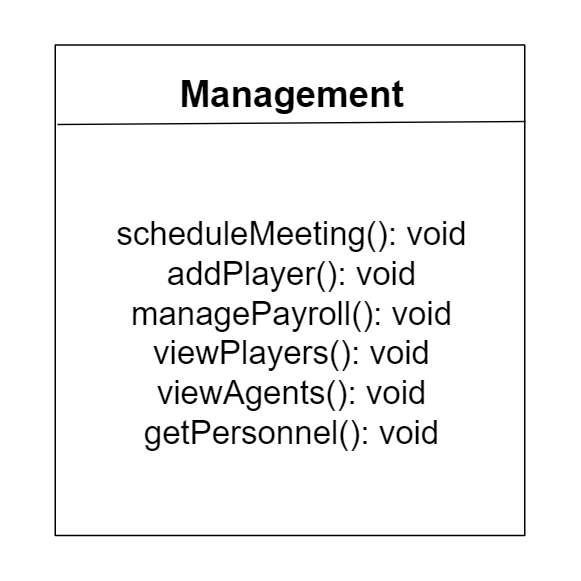
This diagram shows the different functionalities of the application for agents and for managers.

## **Sequence diagram**

**

This diagram displays the fundamental concept of our project based on the sports franchise database

## **Class diagram**

**

* scheduleMeeting() - managers can schedule a meeting with the player’s agents
* addPlayer() - managers can add players to their team if and only if the manager met with the player’s agents
* managePayroll() - teams can manage their payroll by paying players based on the amount of money they have; teams cannot pay players if they don’t have enough money
* viewPlayers() - agents can view the players they are assigned to
* viewAgents() - players can view the agent they were assigned to
* getPersonnel() - returns the personnel table from the sports franchise database

# **System Evolution**

The program can display information from the personnel table in our sports franchise database as of right now. We plan on adding more user interaction to our web application in the future. These user interactions include the ability for managers to schedule a meeting, teams managing their payroll in order to pay players and players viewing their assigned agent.

# **Appendices**

We used Microsoft Visual Studio to connect our sports franchise database to the web application.

# **Index**

1. [Preface](#_heading=h.1ksv4uv) Page 4

[1.1. Modification History](#_heading=h.44sinio) Page 4

[1.2. Intended Audience](#_heading=h.2jxsxqh) Page 4

[2. Introduction](#_heading=h.z337ya) Pages 4-5

2.1 Purpose of the requirements document Pages 4-5

2.2 Scope of the product Page 5

2.3 Reference Page 5

2.4 Overview of the remainder of the document Page 5

[3. Glossary](#_heading=h.3j2qqm3) Page 5

[4. User Requirement Definition](#_heading=h.tyjcwt) Page 6

[5. System Requirements Specification](#_heading=h.3dy6vkm) Pages 6-7

[6. System Architecture](#_heading=h.1y810tw) Page 7

[7. System Models](#_heading=h.4i7ojhp) Pages 8-9

[7.1. Use Case Diagram](#_heading=h.2xcytpi) Page 8

[7.2. Sequence diagram](#_heading=h.17dp8vu) Pages 8-9

[7.3. Class diagram](#_heading=h.3rdcrjn) Page 9

[8. System Evolution](#_heading=h.1ci93xb) Pages 9-10

[9. Appendices](#_heading=h.3whwml4) Page 10

[10. Index](#_heading=h.2bn6wsx) Page 10