Software Engineering
Software Requirements Specification (SRS) Document

Personal Sports Tracker

2/20/23

V1

By: John Abdellmelk, Jonathan Gallimore, Lennon Burchette

[Honor Code]

Table of Contents

1. Introduction 3
   1.1. Purpose 3
   1.2. Document Conventions 3
   1.3. Definitions, Acronyms, and Abbreviations 3
   1.4. Intended Audience 4
   1.5. Project Scope 4
   1.6. Technology Challenges 4
   1.7. References 4
2. General Description 4
   2.1. Product Perspective 4
   2.2. Product Features 4
   2.3. User Class and Characteristics 5
   2.4. Operating Environment 5
   2.5. Constraints 5
   2.6. Assumptions and Dependencies 5
3. Functional Requirements 5
   3.1. Primary 5
   3.2. Secondary 5
4. Technical Requirements 6
   4.1. Operating System and Compatibility 6
   4.2. Interface Requirements 6
      4.2.1. User Interfaces 6
      4.2.2. Hardware Interfaces 6
      4.2.3. Communications Interfaces 6
      4.2.4. Software Interfaces 6
5. Non-Functional Requirements 6
   5.1. Performance Requirements 6
   5.2. Safety Requirements 7
   5.3. Security Requirements 7
   5.4. Software Quality Attributes 7
      5.4.1. Availability 7
      5.4.2. Correctness 7
      5.4.3. Maintainability 7

5.4.4. Reusability 7
5.4.5. Portability 7
5.5. Process Requirements 7
5.5.1. Development Process Used 7
5.5.2. Time Constraints 7
5.5.3. Cost and Delivery Date 7
5.6. Other Requirements 7
5.7. Use-Case Model Diagram 8
5.8. Use-Case Model Descriptions 8
5.8.1. Actor: Actor Name (Responsible Team Member) 8
5.8.2. Actor: Actor Name (Responsible Team Member) 8
5.8.3. Actor: Actor Name (Responsible Team Member) 8
5.9. Use-Case Model Scenarios 8
5.9.1. Actor: Actor Name (Responsible Team Member) 8
5.9.2. Actor: Actor Name (Responsible Team Member) 9
5.9.3. Actor: Actor Name (Responsible Team Member) 9
6. Design Documents 9
6.1. Software Architecture 9
6.2. High-Level Database Schema 9
6.3. Software Design 9
6.3.1. State Machine Diagram: Actor Name (Responsible Team Member) 9
6.3.2. State Machine Diagram: Actor Name (Responsible Team Member) 9
6.3.3. State Machine Diagram: Actor Name (Responsible Team Member) 9
6.4. UML Class Diagram 9
7. Scenario 10
7.1. Brief Written Scenario with Screenshots 10

1. Introduction

1.1. Purpose

The goal of the Personal Sports Tracker is to allow users to keep track of all their favorite sports teams in one area, this will allow them to have the convenience of all sports that they like in one area.

1.2. Document Conventions

The purpose of this Software Requirements Document (SRD) is to describe the client-view and developer-view requirements for the Personal Sports Tracker (PST). Client-oriented requirements describe the system from the client's perspective. These requirements include a description of the different types of users served by the system. Developer-oriented requirements describe the system from a software developer's perspective. These requirements include a detailed description of functional, data, performance, and other important requirements.

1.3. Definitions, Acronyms, and Abbreviations

Java	A programming language originally developed by James Gosling at Sun Microsystems. We will be using this language to build the Personal Sports Tracker
MySQL	Open-source relational database management system.
.HTML	Hypertext Markup Language. This is the code that will be used to structure and design the web application and its content.
SpringBoot	An open-source Java-based framework used to create a micro Service. This will be used to create and run our application.
MVC	Model-View-Controller. This is the architectural pattern that will be used to implement our system.
Spring Web	Will be used to build our web application by using Spring MVC. This is one of the dependencies of our system.
Thymeleaf	A modern server-side Java template engine for our web environment. This is one of the dependencies of our system.
NetBeans	An integrated development environment (IDE) for Java. This is where our system will be created.
API	Application Programming Interface. This will be used to implement a function within the software where the current date and time is displayed on the homepage.
VS Code	Visual Studio Code, also commonly referred to as VS Code, is a source-code editor made by Microsoft with the Electron Framework, for Windows, Linux and macOS.
Javascript	JavaScript, often abbreviated as JS, is a programming language that is one of the core technologies of the World Wide Web, alongside HTML and CSS.
CSS	Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML or XML.

1.4. Intended Audience

1.5.

Client: User
John Abdellmelk: Project Managers, Developer
Jonathan Gallimore: Developer
Lennon Burchette: Developer

1.6. Project Scope

[Specify how the software goals align with the overall business goals and outline the benefits of the project to business.]

The goal of the software is to provide an easy-to-use interface for all users, moderators, and developers, as well as provide user with flexibility to meet their needs. This aligns with the overall business goals of a Personal Sports Tracker as a tracking service requires flexibility and adaptability this will be a helpful tool for users convenience.

The benefits of the project to business include:

- Keeping track of favorite teams all in one area.
- A place where people can talk about sports and review them.
- Convenience to the user of a centralized location for all sports teams.

1.7. Technology Challenges

Where reviews are stored and how the website or application will be constantly updated and running.
API troubleshooting and finding.

1.8. References

ESPN Internet Ventures. (n.d.). *Serving sports fans. anytime. anywhere.* ESPN. Retrieved February 20, 2023, from https://www.espn.com/

262588213843476. (n.d.). *ESPN Hidden Api Docs*. Gist. Retrieved February 20, 2023, from https://gist.github.com/akeaswaran/b48b02f1c94f873c6655e7129910fc3b

2. General Description

2.1. Product Perspective

Trouble keeping track of favorite teams in one location so this idea stems from the convenience of centralized sports tracking.

2.2. Product Features

The product features include the ability for individual users and sports fans to create accounts and the ability for administrators to manipulate those accounts. Users can also add favorite teams to their profiles, where they can track their teams winnings and record among other stats. For moderators, the functionality also includes the ability to create reviews and keep track of favorite teams and team stats.. For administrators, the functionality also includes the possibility to view and delete accounts. Admins can also accept or decline reviews based on appropriate reviews for the site.

2.3. User Class and Characteristics

This website will be fairly accessible to anyone who has an idea of what sports are and is able to use a website other than that there will be no pre-knowledge necessary because this is a tool for the user. The only knowledge that is necessary is that of which what sports you like and what sports you don't like and if you can type and review other players in sports and general ideology of sports in a review system.

2.4. Operating Environment

Currently the idea is only being built on the premise of a website but in the future will be looked into for a app.

2.5. Constraints

Design and implementation constraints would be on the factor that the website is being heavily reliant on the use of apis and a constraint is also the use of a server and or storage factor that will be in place to store reviews and constantly run.

2.6. Assumptions and Dependencies

The software will be dependent on Spring Web and Thymeleaf in order to create and execute the MVC architecture that will be developed within NetBeans. The application will also use the ESPN API https://www.espn.com/ that will display the current teams and stats on the home dashboard for everyone to see.

3. Functional Requirements

3.1. Primary

[All the requirements within the system or sub-system in order to determine the output that the software is expected to give in relation to the given input. These consist of the design requirements, graphics requirements, operating system requirements and constraints if any.]

- FR0: The website will allow the user to sign in and sign out while keeping all information intact and the way it was when they left.
- FR1: The system will allow the user to add and remove their favorite teams.
- FR2: The system will allow the user to add a reviews in a designated area

3.2. Secondary

- Password protection for information only accessible to developers, moderators, and each individual table.
- Authorization scheme so that users can edit and see personal information that others can not.
- Allowing moderators to remove reviews that are harmful.

4. Technical Requirements

4.1. Operating System and Compatibility

The application will be compatible with any operating system that is able to view and to interact with traditional web pages.

4.2. Interface Requirements

4.2.1. User Interfaces

Home tab, reviews tab, favorites tab, search bar, profile icon.

4.2.2. Hardware Interfaces

4.2.3.

The web application will run on any hardware device that has access to the internet, the ability to display webpages, and the ability to interact with web pages. This includes, but is not limited to, smartphones, tablets, desktop computers, and laptops.

4.2.4. Communications Interfaces

It must be able to connect to the internet as well as the local database on phpMyAdmin. The communication protocol, HTTP, must be able to connect to the World Time API and return the current date and time.

4.2.5. Software Interfaces

We will use VS code and Spring Boot ThymeLeaf to help build the frontend, as well as JPA for the backend database functionality. We will also use Spring Boot with Java to connect the frontend to the backend.

5. Non-Functional Requirements

[Constraints on the services or functions offered by the system (e.g., timing constraints, constraints on the development process, standards, etc.). Often apply to the system as a whole rather than individual features or services.]

5.1. Performance Requirements

[The performance requirements need to be specified for all the functional requirements.]

- **NFR0(R):**

  N/A yet

5.2. Safety Requirements

Moderators to constantly check for malicious or false information.

5.3. Security Requirements

- NFR4(R): The personalied system will only be usable by authorized users, this includes the review system and favorite system.
- NFR5(R): All website visitors have access to the home page.

5.4. Software Quality Attributes

5.4.1. Availability

Self sustaining 24/7

5.4.2. Maintainability

Apis constantly update

5.4.3. Reusability

Self sustaining

5.4.4. Portability

Will be available on all platforms that can access websites.

5.5. Process Requirements

5.5.1. Development Process Used

Agile Process

Software Process Model

5.5.2. Time Constraints

The end of the semester of spring 2023 at UNCG

5.5.3. Cost and Delivery Date

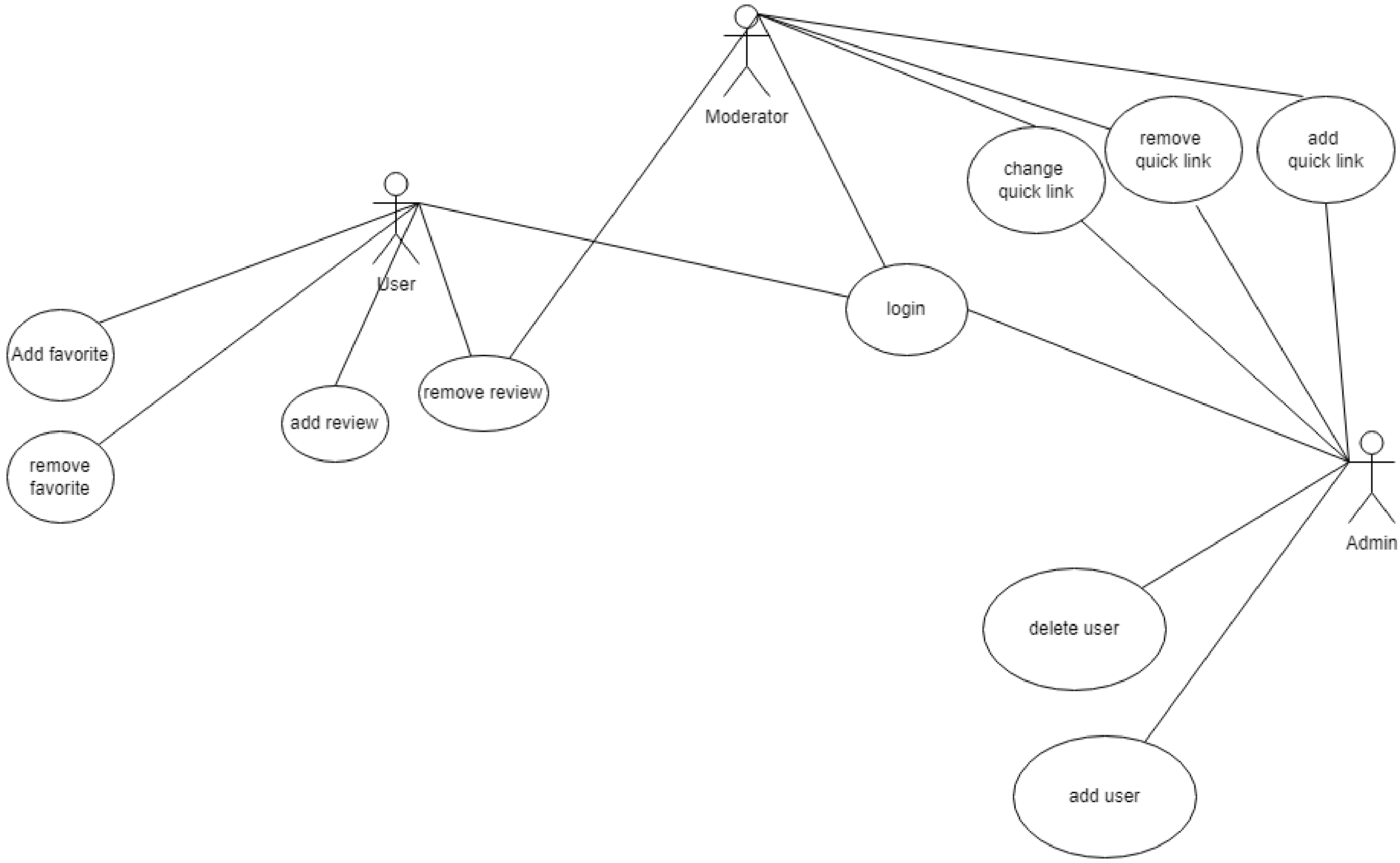
The end of the semester of spring 2023 at UNCG

Free

5.6. Other Requirements

TBD

5.7. Use-Case Model Diagram



5.8. Use-Case Model Descriptions

5.8.1. Actor: Moderator (John Abdellmelk)

- **Use-Case Name**: Moderating the website through removing reviews and updating quicklinks

5.8.2. Actor: User (Johnathan Gallimore)

- **Use-Case Name**: Adding and removing reviews, teams, and being able to make a profile to keep track of everything

5.8.3. Actor: Admin (Lennon Burchette)

- **Use-Case Name**: Add and remove users and all things that the moderator can do except remove reviews.

5.9. Use-Case Model Scenarios

5.9.1. Actor: Moderator (John Abdellmelk)

- **Use-Case Name**: Remove Reviews
  - **Initial Assumption**: malicious reviews, inappropriate, incorrect reviews are in place.
  - **Normal**: The removal of reviews
  - **What Can Go Wrong:** Spamming and the uncontrolablity of reviews.
  - **Other Activities**: none
  - **System State on Completion**: note started
- **Use-Case Name**: Login
  - **Initial Assumption**: the need to get in and out of a website
  - **Normal**: going in and out of a website with ease
  - **What Can Go Wrong**: failure in storage or server issues with communication

- **Other Activities**: none
- **System State on Completion**: not started

- **Use-Case Name**: Add quicklink
    - **Initial Assumption**: bring in more people to website
    - **Normal**: adding a quicklink with images
    - **What Can Go Wrong**: img isnt shown correctly
    - **Other Activities**: none
    - **System State on Completion**: not started

- **Use-Case Name**: Change quicklink
    - **Initial Assumption**: keep people engaged and more traffic
    - **Normal**: changing a quick link with images
    - **What Can Go Wrong**: cant remove the old to put the new
    - **Other Activities**: none
    - **System State on Completion**: not started
    -
- **Use-Case Name**: remove quicklink
    - **Initial Assumption**: no need for current or outdated link
    - **Normal**: changing a quick link with images
    - **What Can Go Wrong**: cant remove because option isnt working
    - **Other Activities**: none
    - **System State on Completion**: not started

-

5.9.2. Actor: User (Johnathan Gallimore)

- **Use-Case Name**: Add/remove favorite teams
    - **Initial Assumption**: Able to add/remove teams of choosing
    - **Normal**: Adding or removing of teams on a profile
    - **What Can Go Wrong**:
    - **Other Activities**: none
    - **System State on Completion**: not started
- **Use-Case Name**: Add reviews
    - **Initial Assumption**: Being able to add reviews to favorite teams recent performance
    - **Normal**: Adding reviews to post
    - **What Can Go Wrong**: Harmful post
    - **Other Activities**: none
    - **System State on Completion**: not started
- **Use-Case Name**: Remove/Edit reviews
    - **Initial Assumption**: Change reviews about their favorite team
    - **Normal**: Removing or editing post
    - **What Can Go Wrong**: Harmful, or changing of words in post
    - **Other Activities**:none
    - **System State on Completion**: not started
- **Use-Case Name**: Make profile
    - **Initial Assumption**: Make profile to be able to keep and store info

- **Normal**: Make a profile that keeps track of favorites and recent post
- **What Can Go Wrong**: False info about personal info
- **Other Activities**: being able to add the favorites to profile
- **System State on Completion**: not started

5.9.3. Actor: Admin (Lennon Burchette)

- **Use-Case Name**: Add users
  - **Initial Assumption**: need to add a user manually for necessary reasons
  - **Normal**: having an override to add a user to the system
  - **What Can Go Wrong**: not being able to add them to the database
  - **Other Activities**: none
  - **System State on Completion**: not started
- **Use-Case Name**: delete users
  - **Initial Assumption**: need to remove a user manually for necessary reasons
  - **Normal**: having an override to remove a user to the system
  - **What Can Go Wrong**: not being able to remove them from the database
  - **Other Activities**: none
  - **System State on Completion**: not started
- **Use-Case Name**: login
  - **Initial Assumption**: no need for current or outdated link
  - **Normal**: changing a quick link with images
  - **What Can Go Wrong**: cant remove because option isnt working
  - **Other Activities**: none
  - **System State on Completion**: not started
- **Use-Case Name**: Add quicklink
  - **Initial Assumption**: bring in more people to website
  - **Normal**: adding a quicklink with images
  - **What Can Go Wrong**: img isnt shown correctly
  - **Other Activities**: none
  - **System State on Completion**: not started
- **Use-Case Name**: Change quicklink
  - **Initial Assumption**: keep people engaged and more traffic
  - **Normal**: changing a quick link with images
  - **What Can Go Wrong**: cant remove the old to put the new
  - **Other Activities**: none
  - **System State on Completion**: not started
  -
- **Use-Case Name**: remove quicklink
  - **Initial Assumption**: no need for current or outdated link
  - **Normal**: changing a quick link with images
  - **What Can Go Wrong**: cant remove because option isnt working
  - **Other Activities**: none
  - **System State on Completion**: not started
-

6. Design Documents

6.1. Software Architecture

6.2. High-Level Database Schema

6.3. Software Design

6.3.1. State Machine Diagram: Actor Name (Responsible Team Member)

6.3.2. State Machine Diagram: Actor Name (Responsible Team Member)

6.3.3. State Machine Diagram: Actor Name (Responsible Team Member)

6.4. UML Class Diagram

7. Scenario

7.1. Brief Written Scenario with Screenshots