Football Season Organizer

Author(s) Joshua Ruiz

CST-452 Final Architecture plan

Grand Canyon University

Instructor: Brandon Bass

Revision: 04/07/2024

Date:04/07/2024

**ABSTRACT**

The Football Season Organizer project is about putting football schedules in one convenient to use place that allows my wife and I to pull up the schedules for our favorite teams without having to google search for games each week. Our current setup is my wife manually searches for the teams we want to watch each week and then prints out the schedules from google and puts them up on our fridge for reference. I want to take the work out of the process for her and mass-upload everything before football season begins.

The application is going to take the football schedule data and maintain it and the application in Amazon AWS. The UI will then pull the data down, allow my wife and I to add our favorite teams to our profiles and output a list of all of their games with the date/times and networks they are being broadcast on in a list or calendar format depending on the view we need for the task we are planning for. The application will also send us notifications each time we login with so that we do not miss any games that may be coming up in the next 7 days.

The project is going to help us to better manage our football season as we enjoy not only watching the games with our family, but we love to have our Christian family over to watch games, eat, and fellowship with each other. The application, once completed, will help us to setup our schedules so that we can have after-church watch parties and can use the time to spread the gospel and build relationships that are integral in making disciples. Overall I am hoping this project saves us time and allows us to be better organized as our lives become more hectic during the school year.

|  |
| --- |
| History and Signoff Sheet |

**Change Record**

|  |  |  |
| --- | --- | --- |
| **Date** | **Author** | **Revision Notes** |
| 09/03/2023 | Joshua Ruiz | Initial draft for review/discussion |
| 03/31/2024 | Joshua Ruiz | Updated for current class and iteration |
| 04/07/2024 | Joshua Ruiz | Updated for CSV Import class and functions |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

|  |
| --- |
| **Overall Instructor Feedback/Comments** |

**Integrated Instructor Feedback into Project Documentation**

Yes  No

**Project Approval**

Professor Mark Reha

**TABLE OF CONTENTS**

Project Overview and Project Objectives 4

Project Scope 5

Project Success Measures 6

Project High-Level Solution 7

Project Controls 8

Project Cost and Schedule 10

Functional Requirements 13

Non-Functional Requirements 15

Technical Requirements 17

Logical System Design 18

User Interface Design 19

Reports Design 21

Appendix A – References 22

Appendix B – Copyright Compliance 23

Project Overview and Project Objectives

**State the Problem and Background**

The purpose of this project is to allow a simple way to gather data for both college and NFL games during football season. The idea came about as my wife and I have trouble always remembering or figuring out which teams are playing on what day and on which network we can find them. This app will allow me to input the data as soon as the NCAA and NFL schedules are released and then utilize the application to get a calendar output that shows each game for the teams that we select. Due to where my wife and I grew up our NFL teams are the Raiders and Cowboys and our NCAA teams are Cornhuskers, UCLA, University of Texas, and potentially UT San Antonion if our adopted son starts to play there next year. It takes along time to compile all of these schedules manually (which my wife did this year because she is amazing) so it gave me the idea to put together something that could potentially automate this process for us.

**Christian Worldview**

The football organizer application would be more of a secular item; however, it is possible to utilize it as a way to organize Christians together to watch games, fellowship and even potentially create relationships between people Christians and non-Christians where the Gospel can be shared. As I am writing this I believe that there may be a way to integrate this type of application into something like Evite to make it easier to create football watch parties for the games giving us more opportunities to fellowship with other Christians and witness to those who are non-Christians.

**Project Objectives**

The objectives of this will be simple:

1. Are we able to integrate the football data into a database and get it to feed into the application.
2. Does the application allow for a user to sign-in and select their favorite teams and are those favorites saved correctly for the user.
3. Does the application allow for a UI interface that renders the applicable team data for the end user so they can track the games they would like to watch or record.
4. Do the email reminders work correctly so that the end-user is notified about upcoming games in advance so they can plan accordingly. (May need to have a notification feature that allows for different timing on notifications.
5. Does the application save time when trying to compile all of the teams that the end user wants to watch during a football season.

**Challenges**

1. Obtaining football data for NFL and NCAA teams in a timely manner. APIs may or may not be available to pull the data down so manual inputs into the database may be required. If this is the case then only a handful of teams will be entered to make sure the project can be completed on time.
2. Security – is any end user personal data and passwords secured during the applications use.
3. UI usability – User acceptance testing will be required to make sure that the UI is usable and helpful.

**Benefits and Opportunities**

The benefits for this application are mainly personal. This application will be for my wife and I to better organize our football seasons each year. It will also give me practice with creating, maintaining and updating an application. I will also be able to use this as a sample for future job postings and projects. I have already integrated a lot of principles learned through programming into my reporting designs and have improved my skill set tremendously.

Project Scope

1. The scope of the project will include the following:
   1. SQL database that can store NFL game schedule data, NCAA game schedule data, HS game schedule data, favorites, and login credentials
   2. User Interface that can display game data in both list and calendar formats
   3. Create profile & Login functionality
   4. Ability to save NFL and NCAA teams into profile to pull down a specific schedule football schedule. Game data will include, Date, Time, Network its playing on, Home or Away game.
   5. Application notifications about upcoming games each week.
   6. Input form to allow for High School game schedules to be input. This form will allow for an entire season of games to be input. (Ex: my adopted son plays highschool football so I will be able to put his teams data into the app.)
2. Use the template to list all known stakeholders and contacts, if applicable, including self (for some projects self may be the only name listed)

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
| Joshua Ruiz | Developer | Entire development process and application maintenance |
| Rachael Ruiz | Spouse | End-user of app so she no longer has to manually compile our football season schedules |

1. List the work breakdown required to satisfy the project objectives. Identify teams and other resources that may be required to successfully complete the project.

|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Work Breakdown Structure | | | | | | | | | | |
| ID | Task | Dependencies | Status | Effort Hours | Cost | Start Date | Planned Completion | Estimate to Completion | Actual Completion | Resource |
| 1 | Create/Design SQL database for Storage | All Data components | Not Started | 2 | 0 | 03/25/2024 | 04/14/2024 | 0% | 100% | PostgreSQL |
| 2 | Create Classes for Application | Business Logic and UI | Not Started | 2 | 0 | 03/25/2024 | 04/14/2024 | 0% | 50% | Spring Boot |
| 3 | Create Business Logic for Application | UI and Basic Functionality | Not Started | 3 | 0 | 03/25/2024 | 04/14/2024 | 0% | 50% | Spring Boot |
| 4 | Create Security Functions | All Application components | Not Started | 4 | 0 | 03/25/2024 | 04/14/2024 | 0% | 40% | Spring Boot |
| 5 | Create UI for Application | UI | Not Started | 8 | 0 | 03/25/2024 | 04/14/2024 | 0% | 40% | Angular or React |
| 6 | UAT | All | Not Started | 8 | 0 | 03/25/2024 | 04/14/2024 | 0% | 0% | Wife will be my tester |

Project Success Measures

|  |
| --- |
| Project Completion Criteria |
| 1 – Able to import game schedule data and manually input games into the application utilizing an import tool/form |
| 2 - UI is able to pull game data into a calendar and list format |
| 3 – Users are able to create login credentials and sign in to create favorites |
| 4 – Users are able to select favorite teams and save them to their profile to view schedules |
| 5- Users will be able to receive notifications of upcoming games for that week when logged into the application |

1. Use the template to list the project assumptions and constraints, if applicable. An assumption is an educated guess that a likely condition or circumstance is presumed to be true. A constraint is a limiting condition or circumstance that defines the project boundaries. Assumptions allow the project to succeed. Constraints restrict or limit the project execution.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 | Game Schedule data will be readily available to collect and input into the database | APIs or manually searching will allow for games schedules and data to be input into the database | Assumption | Valid | 08/20/2023 |
| 2 | Limited ability to input data into database | Will only be able to enter a few NFL and NCAA teams schedules into the database due to time constraints for project. | Constraint | Valid | 8/20/2023 |

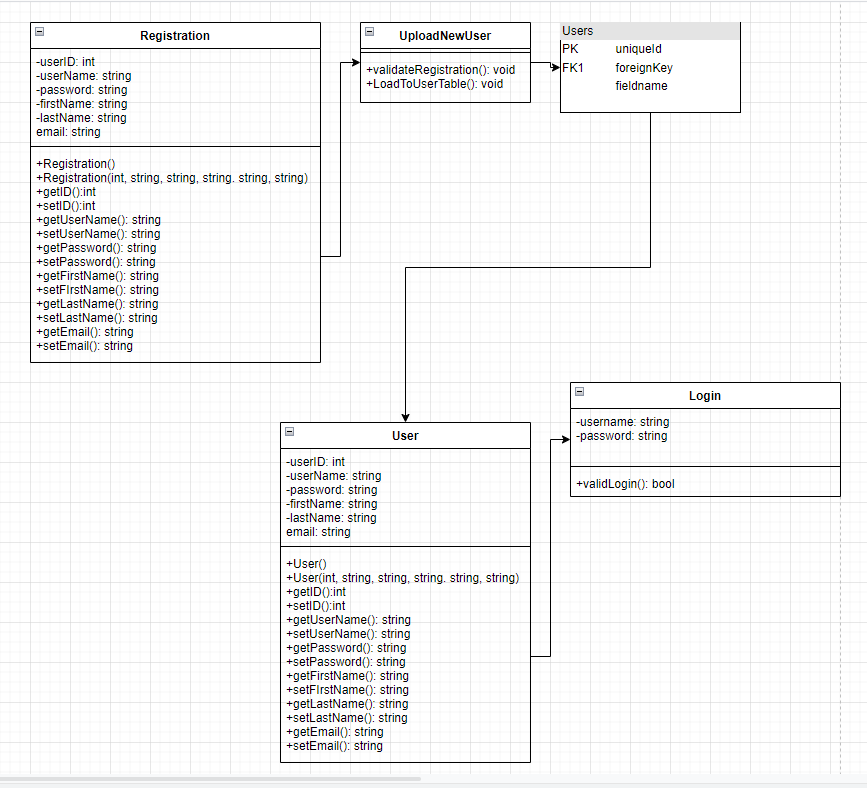
Project High-Level Solution

**Introduction**

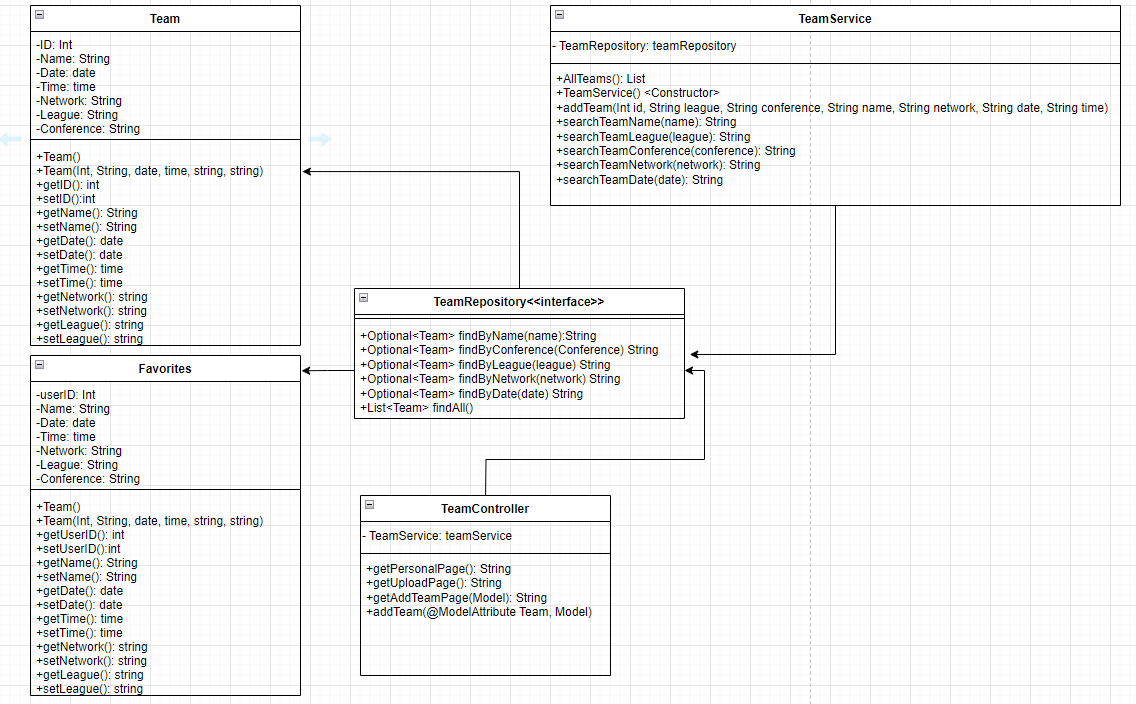
The high-level solution is to create an application that will allow my wife and I to input football game schedule data for Professional and Collegiate level teams into a database that will then output the results in a calendar/list format to make keeping track of our football watch schedules easier. It will require football schedule data to be readily available, but this iteration of the project will only contain a few teams from each level in order to give a proof of concept rather than a fully integrated application.

**Solution**

The solution will be made up of a simple SQL database with tables to store game data, login credentials, and user profiles, a basic UI that will allow the user to login/create an account, save teams to their favorites, and then pull the schedules for those teams into a Calendar or list format for the season. The application will utilize reminders to help the user keep track of the date, time, and broadcasting network for each game through the use of notifications each week. The application will be simple, but useful for our private utilization. The main idea will be to make an application that works, then each season add more functionality to it like integration into an Evite or email system. Basic wireframes and draft UML for login and Teams class attached for reference. Will continue to update as the project develops and new classes are created.

UML Classes – User registration & Login

UML Team Class:

****

Project Controls

1. Use the template to define the risk and list the steps to prevent the risk from occurring or the steps to minimize the chances of it happening. The contingency plan describes alternative solutions to reduce the impact of the risk. An example of a contingency plan is to provide the customer a temporary web server if there are delays in delivery/completion. If the risk has already happened then provide an entry in the issue log.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(high, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| What is the risk? | What is the probability? | What is the impact if the risk occurs? | What can be done to minimize the risk? | What can be done to minimize the impact of the risk? |
| Developer Inexperience | 50% | Delays on project completion | Engage professor and other sources with help in development | Ask for assistance as soon as hit an impasse while programming |
| Time constraints | 100% | Delays due to professional work schedule and family needs | Will need to integrate the development into part of the work day to make sure time is set aside each day to complete tasks | Will utilize Vacation time as project gets closer to due date to ensure time is available to complete tasks. |
|  |  |  |  |  |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | What is the issue? | How will this impact scope, schedule & cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| 2 | Running into problems attaching the uploaded team data to the calendar function | High impact – need for teams and date to populate for application to work | I will research and develop the best way to connect the data in the teams table to the calendar UI. | Developer | High | 04/07/2024 | 04/14/2024 |  |
| 3 | Questions on how best to implement favorites functionality | Need to be able to assign favorites by userID in order for the calendar to populate for individuals. | Will need to look into tutorials for the best way to create a favorites/wishlist to implement | Developer | High | 04/07/2024 | 04/14/2024 |  |
| 4 | Mass Upload for CSV file is accepting the file, but won’t transfer to the db. | The ability to add individual teams is working correctly, however, there are issues with the mass upload. | I will need to research functionality that allows for CSV uploads and the best way to connect the ui to the functionality. | Developer | Medium | 04/07/2024 | 04/14/2024 |  |
| 5 | Need to develop duplication handling functions to avoid duplicate submissions in individual and mass uploads | Avoiding duplicate data will help the application run better and allow for a cleaner look and feel to the application. | I will need to look up the best practices for how to validate the data so that duplicate teams are not allowed to be uploaded | Developer | Medium | 04/07/2024 | 04/14/2024 |  |

1. All projects have either anticipated and planned or unexpected changes. Describe any issues in management or change management due to the anticipated and planned or unexpected changes. Use the template to list anticipated and planned or unexpected changes.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 | May need to utilize non NFL/NCAA team names | High | Josh | 8/20/2023 | 8/20/2023 |  |  | Unkown |  |
| 2 |  |  |  |  |  |  |  |  |  |

1. Use the template to describe how the end user is involved in the software development, if applicable. Include relevant information about meetings, reviews, presentations, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Roles and Responsibilities | | | |
| Name | Team | Project Role | Responsibility |
| Joshua Ruiz | Sole Developer | Lead Developer | All development responsibilities |
|  |  |  |  |

Project Cost and Schedule

1. Project will not require any costs

Project schedule (subject to revision) is found in the work breakdown section. As are the work breakdown and time estimates for the project. All estimates are subject to revision depending needs/ future constraints

Functional Requirements

**Use Cases**

Describe the sequence of functional actions a project performs with either textual Use Cases, UML Use Case diagrams, or if using Scrum, provide a link to the User Stories (see template included in course materials).

NOTE: Once the functional requirements have been completed, there may be situations where Use Cases or User Stories may need to be taken out of scope, possibly due to technical challenges or timeline challenges. Any Use Cases or User Stories that are taken out of scope once the project development has started must be approved by the mentor and instructor with justification as to why the functionality is being removed from the project. The following must be updated if any Use Cases or User Stories are taken out of scope:

|  |  |  |
| --- | --- | --- |
| **Use Case or User Story** | **Approval Date** | **Justification** |
| The application will utilize an import process that allows for data to be moved into Amazon AWS for data storage and hosting. These imports will either be mass uploads or through individual team schedule updates for high school football schedules. NFL APIs exist but will not be able to be used due to licensing costs/requirements |  |  |
| The application will have a form that will allow the user to input game information manually for high school teams. |  |  |
| The application will allow a user to create/edit/delete their account from the application |  |  |
| The application will allow each user login to add favorite teams from NCAA, NFL, and manually input high school teams into the organizer as well as remove favorites from their profiles |  |  |
| The application will display the game schedules for all favorited teams in both calendar and list formats, will display teams for only a selected team, or for a selected league |  |  |
| The application will be a web based application and the UI will utilize React.js to create a UI that will give the user a easy to use streamlined look. |  |  |
| The application will display multiple pages like outlook calendar functionality where the games can be viewed in month, week, & day views. It will also allow for everything to be listed in a table format with excel export functionality. |  |  |
| The application will send out email reminders to the user for upcoming games along with calendar reminders while utilizing the application. |  |  |

***NOTE: If necessary, you may add subsections to those listed in order to match the requirements in the assignment description. Do not remove any top-level sections (those that are listed in the Table of Contents). As required by your project, you may add additional top-level sections (please update the Table of Contents).***

Non-Functional Requirements

**Use Cases**

Describe the sequence of non-functional actions a project performs with either textual Use Cases, UML Use Case diagrams, or if using Scrum, provide a link to the User Stories (see template included in course materials).

NOTE: Once the non-functional requirements have been completed, there may be situations where Use Cases or User Stories may need to be taken out of scope, possibly due to technical challenges or timeline challenges. Any Use Cases or User Stories that are taken out of scope once the project development has started must be approved by the mentor and instructor with justification as to why the functionality is being removed from the project. The following must be updated if any Use Cases or User Stories are taken out of scope:

|  |  |  |
| --- | --- | --- |
| **Use Case or User Story** | **Approval Date** | **Justification** |
| The application will encrypt passwords for storage in the users table: Password requirements will be min 10 characters with lowercase letters, uppercase letters, numbers, and Special characters. |  |  |
| The application will utilize Amazon AWS in order to scale based on data usage. |  |  |
| The application will require the use of Chrome, Edge, or Firefox to operate correctly. |  |  |
| The application will have downtime once a year while the new rosters and schedules are being updated. The downtime will be a few hours as the data is updated. Users only need to access the application a few times a week. Downtime should be minimal. |  |  |
| Maintainability and Manageability will have minimal impact as the application will not have a wide user base nor will it be a larger enterprise level application. This will change as functionality is added to the application or the user base starts to increase. |  |  |
| The application will have high usability as the UI will have easy navigation buttons along with different data views configurable for what the user likes best. |  |  |
| Each page must load within 10 seconds |  |  |

Technical Requirements

**Use Cases**

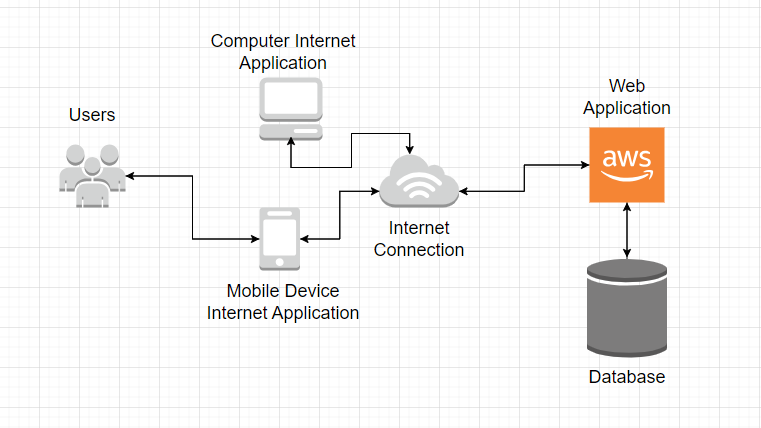
Describe the tools and technologies used in the project.

NOTE: Once the technical requirements have been completed, there may be situations where technologies or tools may need to be taken out of scope or changed, possibly due to technical challenges or timeline challenges. Any technologies or tools that are taken out of scope or changed once the project development has started must be approved by the mentor and instructor with justification as to why the functionality is being removed from the project. The following must be updated if any technologies or tools are taken out of scope or changed:

|  |  |  |
| --- | --- | --- |
| **Technology or Tool** | **Approval Date** | **Justification** |
| Programming Language: Java |  |  |
| Framework: Spring |  |  |
| IDE: Spring Boot |  |  |
| REST: Spring Data REST |  |  |
| UI: React.JS |  |  |
| Git: Github w/ SourceTree to help version the application |  |  |
| Data (1 of 2)  Storage/Access PostgreSQL |  |  |
| Data (2 of 2) Storage/Access/Application hosting: Amazon AWS (Free Tier) (Future State) |  |  |
| Postman: Test application API functionality |  |  |
| Chrome, Edge, Mozilla: Application function testing and UAT |  |  |
| Draw.io: UML, Mapping, Storyboarding, Flowcharting, etc. |  |  |
| CodeWhisperer: simplify integration of application into AWS |  |  |

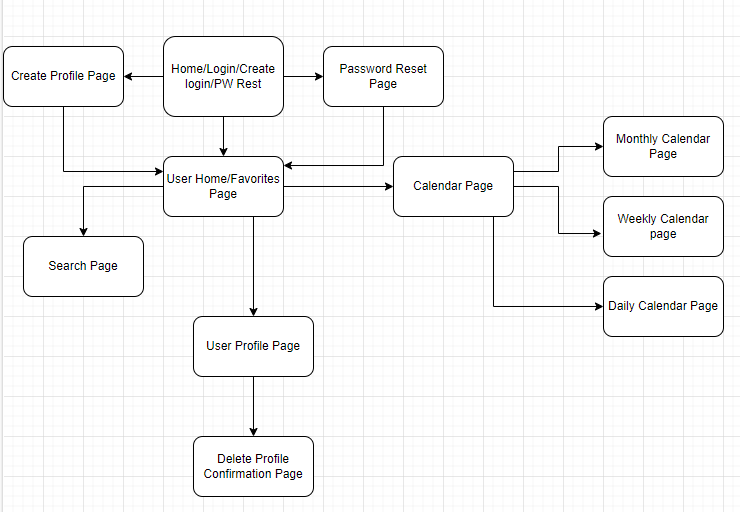
Logical System Design

Provide a diagram of the logical architecture of the system.

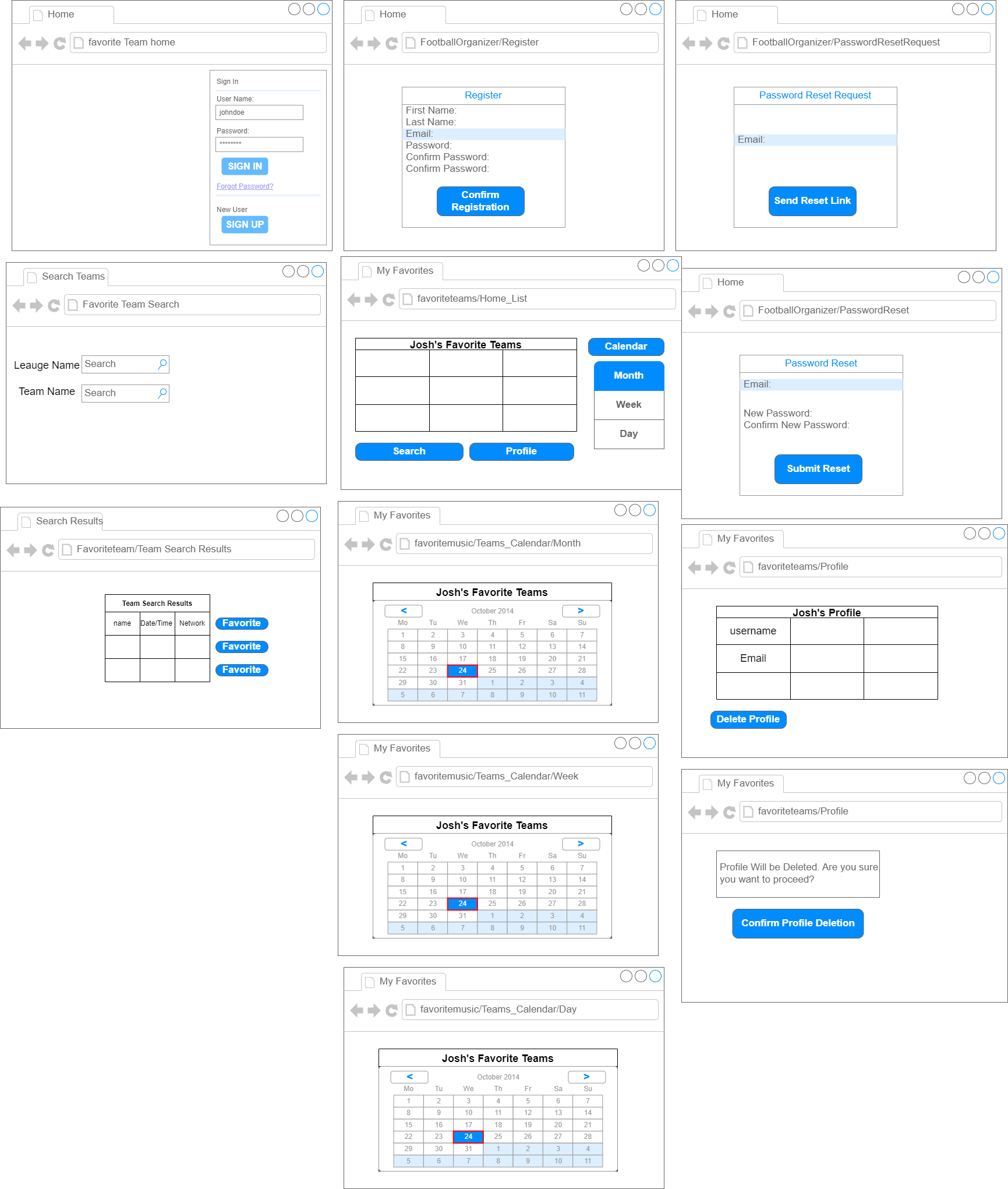
****

User Interface Design

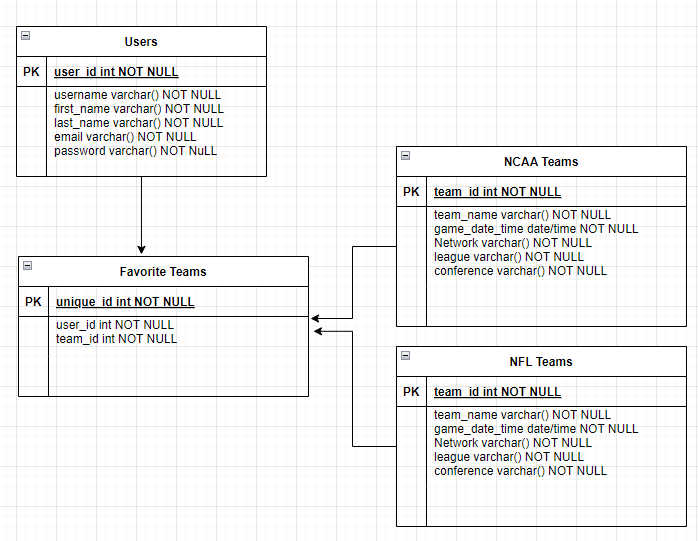
Provide a sitemap and user interface design diagram for each user interface screen in the application, if not applicable, define the components of the project as described in the handbook.

Site Map

Wireframe



ERM Diagram



Reports Design

Provide a listing of the reports that the system will provide, if applicable. If not, state that the system does not produce any reports and provide additional documentation as described in the handbook.

1. Error Logs
2. The Favorite Teams game schedules will be able to be exported to Excel to allow for print functionality.
3. Admins will be able to pull AWS included BI reports through the Free tier access level.

Appendix A – References

N/A

Appendix B – Copyright Compliance

Calendar Template utilized from:

2024. DayPilot Lite, Monthly Calendar in Spring Boot/Java (Open-Source)

r https://code.daypilot.org/58614/using-javascript-html5-monthly-calendar-in-spring-boot-java

**ScreenCast:**

[**https://www.loom.com/share/5d77a038a30e4e68a1f60d521a4309a5?sid=71e68249-2c46-4658-8b42-a4bd392f80a3**](https://www.loom.com/share/5d77a038a30e4e68a1f60d521a4309a5?sid=71e68249-2c46-4658-8b42-a4bd392f80a3)