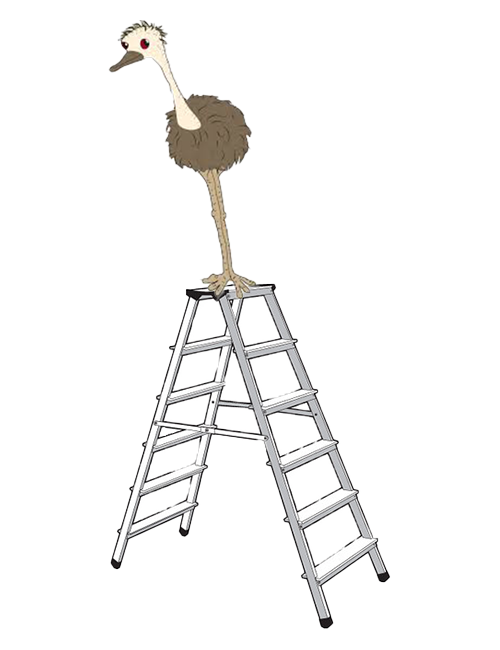
emuLadder

Daniel Frye, Gabriel Wilmoth, Mckenzie Moize, Rinty Chowdhury, Robert Meyer

Table of Contents

**Project Definition 2**

**Functional Requirements 2**

**Usability Requirements 2 System Requirements 3**

**Security Requirements 3**

**Project Specifications 3**

Focus Domain **3**

Libraries / Frameworks **3**

Platform **4**

Genre **4**

Project Definition

The application, emuLadder, is a Fantasy eSports game similar to ESPN’s Fantasy Football, Hockey, etc. Users draft real-life eSports players from games such as League of Legends and compete against other users with their drafted players. Users earn points based on their drafted players stats per game. Fantasy games are very popular in sports such as football, soccer, hockey and many more, however, there is not a strong presence of Fantasy eSports. eSports is an ever-growing scene, and emuLadder seeks to provide a Fantasy experience to a community that not only enjoys eSports but also to those that enjoy video games in general. To achieve this, emuLadder looks to provide a competitive environment that keeps the user involved in both their Fantasy game and the eSports games their players are competing in.

Functional Requirements

1. The application should be able to track eSports data in real-time.
2. The user should be able to join game rooms by invitations.
3. The user should be able to create an account using his/her information.
4. Users should be able to send invitation codes to projected friends. The application should be able to see the user’s contact information and social media in order to create new friends.
5. The application should be able to track player progress and update it in real-time.
6. The application will be able to handle pseudo-money transactions between users. The user can use that virtual money to add or exchange a good player to his/her team.
7. Users should be able to host a game room.

Usability Requirements

1. Game status will be displayed to keep the user updated about the game.
2. The application will match with the real world so that the user can connect with it easily. It will make things easy for the user to play. All the game symbols and characters will be similar to real-world icons.
3. Learning and performing the game should be easy for the new user. The game buttons and icons should be self-explanatory. The game will provide help to the new user when extra help is needed.
4. Users will have some freedom to change the game settings so that the system will be more flexible and user-friendly but it will be limited. Users will be able to undo the job if they don’t want to continue with it.
5. The application will have maintenance options to prevent the system from error. It will also help the user to recognize, diagnose, and recover from bugs.
6. The application will maintain the basic game standard and feature format.
7. Users will be able to save and load their data.
8. The user can customize and personalize the game based on their choice to make the game more fun and engaging.

System Requirements

1. The system will be web-based, thus the application will be available on any machine with a web browser. Supporting the latest versions of Chrome, Firefox, Safari, Safari for iOS and Internet Explorer 9-11.
2. Basic internet connectivity.

Security Requirements

1. The system must maintain the confidentiality of all data that is classified as confidential.
   1. Passwords
   2. Emails
   3. Credit Card Information
2. The systems libraries/plugins must follow and ensure the confidentiality of login information as well as payment information.
   1. Sign-up / Sign-in plugin
   2. 3rd party payment plugin
3. Two-factor authentication to meet standard security requirements.

Project Specifications

Focus Domain

1. Our domain is online betting/drafting applications. Our focus is specifically to make an application for users interested in eSports just like the already booming Fantasy Sports applications. A specific target crowd being the exponentially growing crowd that watches eSports.

Libraries / Frameworks

1. Java backend for connecting to database and auxiliary services.
   1. SpringBoot Framework
   2. IntelliJ Development Environment
2. JavaScript auxiliary service for web scraping and API calls
   1. Node.js
   2. NodeEclipse Development Environment
3. MySQL for database services
   1. AWS Relational Database Management
   2. MySQL Workbench Development Environment
4. JavaScript for functionality of web pages
   1. Angular 8 / Typescript
   2. Visual Studio Code Development Environment
5. HTML5 / CSS3 for front end visualization and web design
   1. Visual Studio Code Development Environment
6. Amazon Web Services for website hosting
   1. AWS Console Development Environment

Platform

A web-based application for laptop, desktop, and mobile machines.

Genre

1. Web application within the genre of Strategic Simulation/Sports Management