Software Requirements Specification

for

DuoLegend Project

**Version 1.0 approved**

**Prepared by Group 3**

**SWR 302 - SU21**

**June 04, 2021**

**Table of Contents**

**Table of Contents ii**

**Revision History ii**

**1.** **Introduction 1**

1.1 Purpose 1

1.2 Document Conventions 1

1.3 Project Scope 1

1.4 References 1

**2.** **Overall Description 1**

2.1 Product Perspective 1

2.2 User Classes and Characteristics 2

2.3 Operating Environment 3

2.4 Design and Implementation Constraints 3

2.5 Assumptions and Dependencies 3

**3.** **System Features 4**

3.1 System Feature 1 5

3.2 System Feature 2 (and so on) 6

**4.** **Data Requirements 2**

4.1 Logical Data Model 2

4.2 Data Dictionary 2

4.3 Reports 2

4.4 Data Acquisition, Integrity, Retention, and Disposal 2

**5.** **External Interface Requirements 2**

5.1 User Interfaces 2

5.2 Software Interfaces 2

5.3 Hardware Interfaces 2

5.4 Communications Interfaces 2

**6.** **Quality Attributes 2**

6.1 Usability 2

6.2 Performance 2

6.3 Security 2

6.4 Safety 2

6.5 [Others as relevant] 2

**7.** **Internationalization and Localization Requirements 2**

**8.** **Other Requirements 2**

**Appendix A: Glossary 2**

**Appendix B: Analysis Models 2**

**Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Date** | **Reason For Changes** | **Version** |
| Group 1 | 21/5/2021 | initial draft | 1.0 |
| Group 1 | 23-5-2021 | approved | 1.0 approved |

# Introduction

## Purpose

## Document Conventions

|  |  |
| --- | --- |
| *DB* | *Database* |
| *DL* | *DuoLegend* |

## Project Scope

Refers to the Vision and Scope1 document, attached in the reference

## References

1. [Vision and Scope document](https://drive.google.com/file/d/1M_KgPYzJRsQ9Nmk3bq5kWzcerzCo5CYz/view?usp=sharing)
2. [Microsoft’s C# Coding convention](https://docs.microsoft.com/en-us/dotnet/csharp/fundamentals/coding-style/coding-conventions)

# Overall Description

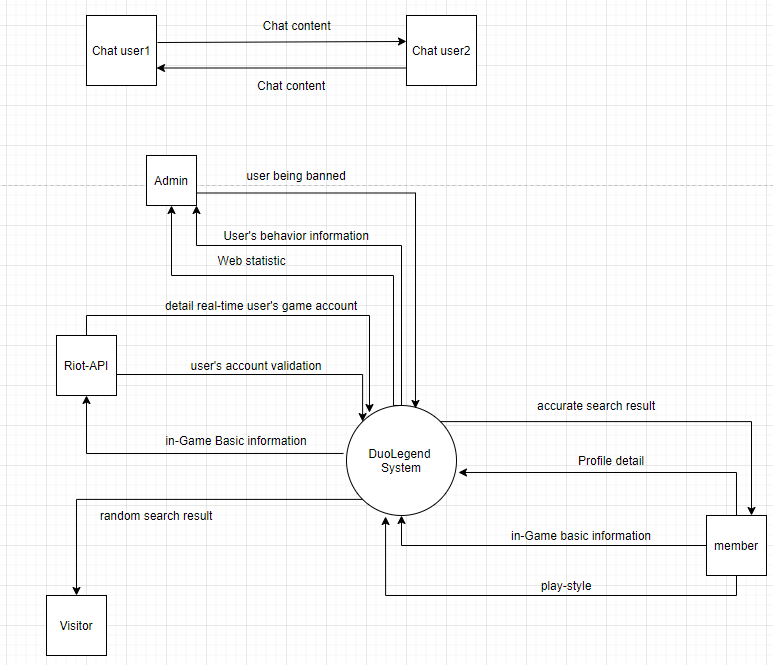
## Product Perspective

The DuoLegend project is a standalone system, the aim of the project is to help players of the game League of Legend find one or more players to play with. Once registered to the system, a user can create a profile with an in-game name. Users will be able to find other users by rank, skill and playstyle.

Users used to have a hard time finding an online-friend to play with, if they use the auto-matching of the game, there was a chance they found a toxic player. Our system helps players to avoid playing with those toxic players and make their game become more fun.

DL has features such as instant messaging, where members can chat with each other and they can connect to each other on Facebook.

The Context Diagram below lists all of the user features that will be implemented in the system.

**

## User Classes and Characteristics

1. Visitors: are the people who just access the DuoLegend website and don’t log in this web

The visitor should be able to do the following function:

* Create an account.
* Login to the DuoLegend website .
* View profile of members.
* View list of random users.

1. Members (favored user class): are the people who have an account of the DuoLegend website, logged into the web and are able to use more basic functions of the web.

The members should be able to do the following function:

* Search and view players based on ranks, servers, roles in League of Legend.
* Chat with other members.
* Follow other members.
* View and edit their own profile and account information.
* Rate other members.
* View profile of other members.
* Verify their own ingame accounts.

1. Admins: are the people who manage the website

The admins should be able to do the following function:

* View user statistics.
* Ban users.
* Delete users.
* View list of users.
* Search users.
* Update champions.

## Operating Environment

* Centralized database.
* Client/server system.
* Operating system: Windows.
* Database: SQL database.
* Platform: ASP.NET.

## Design and Implementation Constraints

1. Uses SQL database.
2. The back-end of the project must be built using C# programming language.
3. The project’s design must conform to the MVC model of web development.
4. The project must use ADO.NET and .NET5.
5. Code written must conform to Microsoft’s C# coding conventions2

## Assumptions and Dependencies

## Because the project is dependent on the API from Riot Games and they limit the request of the website using their API to 20 requests every second so there is one assumption that the API from Riot Games works well for the whole life of a project and the number of requests from users are always under the limitation.

# System Features

## Create account

### Description

Visitors are able to create an account to use the full functionality of the web.

Priority: High.

### Stimulus/Response Sequences

* Visitors access DuoLegend web page and home page appears
* After the visitors click on the “Register” button, the web page will redirect to the registration page.
* DuoLegend asks visitors to fill in the requested information input, including ingame name, password, password confirmation, email.
* After registering successfully, a message will inform visitors that they have created an account and be able to redirect to the homepage.

### Functional Requirements

* The website should allow users to register, including input right email, password, game server.
* If user input is not valid, retyping is required.

## View a list of random users

### Description

Visitors and members are able to view a list of random users when accessing the website.

Priority: High.

### Stimulus/Response Sequences

* Users visit DuoLegend’s main page.
* List of random users is displayed for users.

### Functional Requirements

* The DuoLegend website will display a list of users when visitors or members have just accessed the home page.
* List of users include ingame name, rank, lane, lasted play, win rate, KDA, their description and mic option.

## Search players by style

### Description

Members can view a list of users who are matched with the playstyle they want to search.

Priority: High.

### Stimulus/Response Sequences

* User clicks on the icon represents the position they want to search.
* User clicks on the drop-down button to drop-down the list of servers.
* User clicks on the server they want to search.
* User clicks on the icon represents the purpose of playing a game.
* User clicks on the “Find your duo” button.
* Show a list of results, based on the following priority: server, purpose, position, rank.

### Functional Requirements

* List of users include ingame name, rank, lane, lasted play, win rate, KDA, their description and mic option.

## Update profile

**3.4.1** **Description**

Members are able to update their account information.

Priority: High.

**3.4.2** **Stimulus/Response Sequences**

* Users login.
* Users click on the view profile button.
* Users click on the edit profile button.
* Users fill in all the fields they want to update.
* Users click on the submit button.
* Redirect to profile page.

**3.4.3** **Functional Requirements**

**-**List of update field include: in-game name, server, password, description, mic status.

**3.5 View user statistic**

**3.5.1 Description**

Admin are able to view statistics about users.

**3.5.2** **Stimulus/Response Sequences**

* Admin login.
* Admin clicks on the statistic table at the main page or clicks on the statistic icon on the navigation bar.
* Display user statistics for admin.

**3.5.3** **Functional Requirements**

* Calculate user statistics.
* Display user statistics as graphs.

**3.6 User Login**

**3.6.1 Description**

Members are able to login to use full functionality of the system.

Priority: High.

**3.6.2 Stimulus/Response Sequences**

* User click on the Login button on mainPage.
* User fill in email and password.
* User can choose “Remember Me” to keep login on the next visit.
* User click submit button.
* Redirect to the main page if Login is successful.

-Show warning if email or password are not matched.

**3.6.3 Functional Requirements**

* Login form must be checked on the client side (email type).