**Software Requirements**

**Specification**

Students: Popescu Sabin-Theodor

Patru Maria Manuela

Sectia: Calculatoare Romana

Grupa:H1

## Table of Contents1.Introduction

## 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “League of Challengers” page which will consists in guides and features related to the game League of Legends. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to a customer for its approval and a reference for developing the first version of the system for the development team.

## 1.2 Scope

“League of Challengers” is a web application which helps players improve themselves as well as their gaming experience with League of Legends. The application is based on the game’s public Application Programming Interface which will allow access to game data like users, game history and other game content.

Users can also create their own combinations of champions, runes and items and calculate a rate of succes upon them.

Most of the information will be fetched from the game’s database through the API, but another database located on a web-server will be needed to store users and their particular setups for the game.

## 1.3 Definitions, acronyms, and abbreviations

|  |  |
| --- | --- |
| **Term** | **Definition** |
| Guest | Someone who interacts with the web application |
| Admin/Administrator | Administratorul de system este persona care are permisiunea specifică pentru gestionarea și controlul sistemului |
| User | Someone with a registered account |
| HTTP | Hypertext Transfer Protocol |
| Champion | In-game character |
| Lore | Story of a champion |
| Summoner | The player which controls a champion |
| Summoner Name | The player’s in-game nickname |
| Setup | Combination of champions, runes and items |
| Difficulty | Champion’s difficulity in a range from 1 to 10 |
| Role | A Champion’s role |
| Type | A Champion’s type |
| Abilities | Champions use abilities |
| Passive | Constantly active ability of a champion |
| Q | The ability under the Q key |
| W | The ability under the W key |
| E | The ability under the E key |
| R | The ability under the R key |
| Ultimate | The R ability |
| Lane | The paths that allied and enemy minions follow. |
| AoE | Area of effect |
| Active | Abilities possessed by champions and certain items that require activating (via clicking and/or hitting a key) to function, as opposed to "passive". |
| AP | Ability Power |
| AD | Attack Damage |
| Assassin | A champion type whose forte is rapidly ambushing and slaying enemy champions. They have abilities that may include stealthing themselves or having a strong dash skill. |
| B | Back: to retreat in the general direction of your base or away from the enemy.  Recall: To return to the base by pressing the "B" key by default. |
| Carry | A champion that generally starts off weaker than other champions, but becomes more powerful as the game progresses.  The action of being ultimately responsible for winning a game. |
| Caster | A champion whose main source of damage is their abilities. The damage can be either magic or physical damage, and can scale with either AP (ability power) or AD (attack damage). |
| CD | Cooldown  Waiting for an ability's cooldown to finish before commencing an action. |
| Elo | A mathematical rating system for a player's relative skill level. |
| Gold | The in-game currency used to buy items. |
| Health/HP | Health, also known as "Hit Points" |
| Mana/MP | Resource used by most champions to cast abilities. |
| Mid | Middle lane |
| Minion | The computer-controlled unit spawned from the allied structure (nexus or capture point) to march to the opposing structure along the designated lane. |
| Marksman | A champion that deals high amounts of attack damage as the match progresses, sacrificing it's defensive power and utility. An official term substitutes "ADC" (attack damage carry). |
| Nexus | The primary structure of the match where minions are spawned from. Victory is achieved when the opposing nexus is destroyed. |
| Meta | The game's current play style, consisting of aspects such as lane setup, jungling, and team composition. |
| CDR | Cooldown Reduction |
| Item | An object carried or used by a champion to enhance their performance in the match. |
| HTML | HyperText Markup Language |
| CSS | *Cascading Style Sheets* |

## 1.4 References

[1] <http://www.mobafire.com/>

[2] <https://www.lolskill.net/>

[3] <http://www.lolnexus.com/>

[4] IEEE Software Engineering Standards Committee, “IEEE Std 830-1998, IEEE Recommended Practice for Software Requirements Specifications”, October 20, 1998.

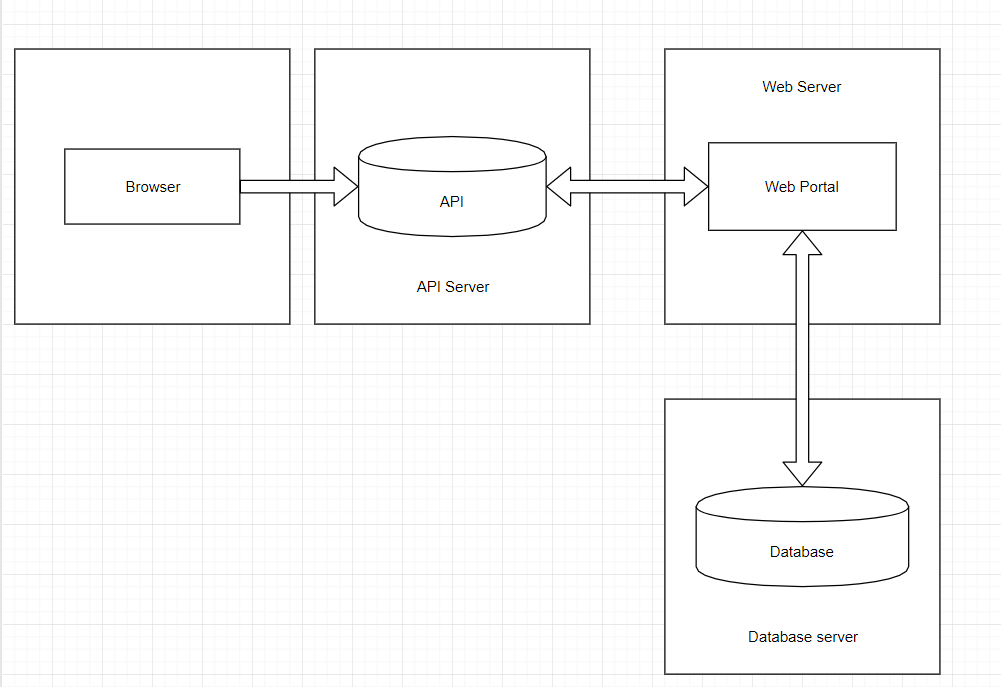
## 1.5 Overview

The remainder of this document includes three chapters and appendixes. The second one provides an overview of the system functionality and system interaction with other systems. This chapter also introduces different types of stakeholders and their interaction with the system. Further, the chapter also mentions the system constraints and assumptions about the product. The third chapter provides the requirements specification in detailed terms and a description of the different system interfaces. Different specification techniques are used in order to specify the requirements more precisely for different audiences. The fourth chapter deals with the prioritization of the requirements. It includes a motivation for the chosen prioritization methods and discusses why other alternatives were not chosen. The Appendixes in the end of the document include the all results of the requirement prioritization and a release plan based on them.

## 2. Overall description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## 2.1 Product perspective



## 2.2 Product functions

-log in

-register

-View all champions

-View all runes

-View all items

-View all skins

-Create mastery 5/6 list

-Add champion in mastery 5/6 list

-Delete account

-Summoner’s stats

-Live game

## 2.3 User characteristics

There are two types of users that interact with the application : guests and registered users; each of them has their own requirements.

The guests can interact with the web portal and view its content, content which will consist in information regarding the game’s data. Guests will also be able to register accounts and become registered users.

Users will be able to create setups consisting in combinations of champions, items and runes. They can also keep track of their Mastery level by populating two lists with champions of level 5 and 6, so they can later decide their picks based on these lists.

Users will also be able to check the details of a Live Game and the statistics of a Summoner by searching it’s name.

## 2.4 Constraints

The application will be constrained by the capacity of the database.

## 2.5 Assumptions and dependencies

The programmin language that we use is C#. The editor (us) should have installed Visual Studio on PC. Otherwise the application doesn’t work. Also, the database will be created using MySQL, and we also use APIes.

## 

## 2.6 Apportioning of requirements

All champions will be stored in a database wich keeps a permanently comunication relationship with the web portal.

English is the main language

Users can acces web portal if they have a wi-fi connection

System Characteristics:

-processor: 1GHz or faster

-memory: at least 128Mb

-disk space: minimum 10 Mb

The application will be implemented in C#

## 3. Specific requirements

This section contains all of the functional and quality requirements of the system. It gives a detailed description of the system and all its features.

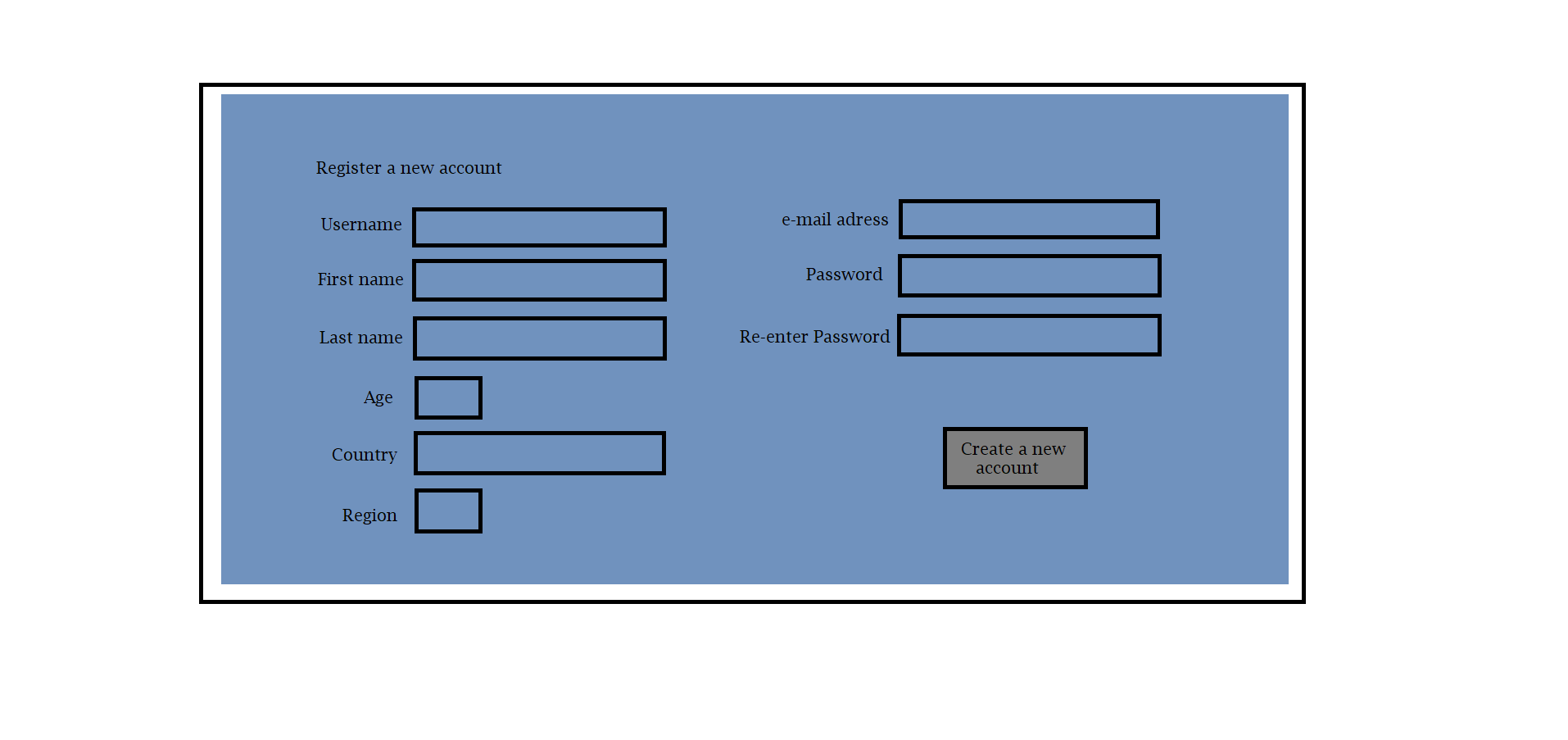
## 3.1 External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the user interface.

## 3.1.1 User interfaces

Guests can acces application too. Guest are users who doesn’t have an account, but they can create one and join the party.

 Our application interface will be sugestive and easy to use.



## 

## 

## 3.1.2 Hardware interfaces

It does not have any direct hardware interfaces.

## 3.1.3 Software interfaces

The application communicates with the database of the game through an API, and this communication consists only in reading data from the database. Some particular data regarding features like Live Game or Summoner Stats will require an API key which will be provided by each user. The application will store all the static data from the game in its own database and the communication between the application and its database will consist of operation concerning both reading and modifying the data.

## 3.1.4 Communications interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems.

## 3.2 Functional requirements

This section includes the requirements that specify all the fundamental actions of the web application.

## 3.2.1 User Class 1 - The Guest

## 3.2.1.1 Functional requirement 1.1

ID: FR1

TITLE: View all Champions

DESC: A guest should be able to access the Champions tab

RAT: In order for a guest to view a complete list of the Champions

DEP: None

## 3.2.1.2 Functional requirement 1.2

ID: FR2

TITLE: View all Skins

DESC: A guest should be able to access the Skins tab

RAT: In order for a guest to view a complete list of the Skins

DEP: None

## 3.2.1.3 Functional requirement 1.3

ID: FR3

TITLE: View all items

DESC: A guest should be able to access the Items tab

RAT: In order for a guest to view a complete list of the Items

DEP: None

## 3.2.1.4 Functional requirement 1.4

ID: FR4

TITLE: View Runes

DESC: A guest should be able to access the Runes tab

RAT: In order for a guest to view a complete list of the Runes

DEP: None

## 3.2.1.5 Functional requirement 1.5

ID: FR5

TITLE: User Registration

DESC: A guest should be able to create an account

RAT: In order for a guest to register an account

DEP: None

## 3.2.1.6 Functional requirement 1.6

ID: FR6

TITLE: User log in

DESC: A guest should be able to log into his registered account

RAT: In order for a guest to register into his account

DEP: FR5

## 3.2.2 User Class 2 –User

## 3.2.2.1 Functional requirement 2.1

ID: FR7

TITLE: Delete account

DESC: A user should be able to delete his registered account

RAT: In order for a user to delete his registered account

DEP: FR6

## 3.2.2.2 Functional requirement 2.2

ID: FR8

TITLE: Create Mastery 5 List

DESC: A user should be able to create a list of champions with mastery 5

RAT: In order for a user to create a list of champions with mastery 5

DEP: None

## 3.2.2.3 Functional requirement 2.3

ID: FR9

TITLE: Create Mastery 6 List

DESC: A user should be able to create a list of champions with mastery 6

RAT: In order for a user to create a list of champions with mastery 6

DEP: None

## 3.2.2.4 Functional requirement 2.4

ID: FR10

TITLE: Add champions to the mastery 5 list

DESC: A user should be able to add champions to the mastery 5 list

RAT: In order for a user to add champions to the mastery 5 list

DEP: FR8

## 3.2.2.5 Functional requirement 2.5

ID: FR11

TITLE: Add champions to the mastery 6 list

DESC: A user should be able to add champions to the mastery 6 list

RAT: In order for a user to add champions to the mastery 6 list

DEP: FR9

## 3.2.2.6 Functional requirement 2.6

ID: FR12

TITLE: Assign a number of tokens to a champion

DESC: A user should be able to assign a number of tokens to a champion

RAT: In order for a user assign a number of tokens to a champion

DEP: FR10, FR11

## 3.2.2.7 Functional requirement 2.7

ID: FR13

TITLE: Delete a champion from a list

DESC: A user should be able to delete a champion from a list

RAT: In order for a user to delete a champion from a list

DEP: FR10, FR11

## 3.2.2.8 Functional requirement 2.8

ID: FR14

TITLE: Create a setup

DESC: A user should be able to create a setup consisting in a champion, a combination of runes and a combination of items.

RAT: In order for a user to create a setup

DEP: None

## 3.2.2.9 Functional requirement 2.9

ID: FR15

TITLE: Update a setup

DESC: A user should be able to update a setup by changing some of its content

RAT: In order for a user to update a setup

DEP: FR14

## 3.2.2.10 Functional requirement 2.10

ID: FR16

TITLE: Delete a setup

DESC: A user should be able to delete a setup by changing some of its content

RAT: In order for a user to delete a setup

DEP: FR14

## 3.2.2.11 Functional requirement 2.11

ID: FR17

TITLE: Live Game

DESC: A user should be able to view a live game’s players by entering his Summoner Name

RAT: In order for a user to view a live game’s stats

DEP: None

## 3.2.2.12 Functional requirement 2.12

ID: FR18

TITLE: Summoner’s Stats

DESC: A user should be able to view a summoner’s stats by entering his Summoner Name

RAT: In order for a user to view a summoner’s stats

DEP: None

## 3.3 Performance requirements

## 3.4 Design constraints

This section includes the design constraints on the software caused by the hardware.

## 3.5 Software system attributes

The requirements in this section specify the required reliability, availability, security and maintainability of the software system.

## 3.5.1 Reliability

**ID: QR9** TAG: SystemReliability 24

GIST: The reliability of the system.

SCALE: The reliability that the system gives the right result on a search.

METER: Measurements obtained from 1000 searches during testing.

MUST: More than 98% of the searches.

PLAN: More than 99% of the searches. WISH: 100% of the searches.

## 3.5.2 Availability

TAG: SystemAvailability

GIST: The availability of the system when it is used.

SCALE: The average system availability (not considering network failing).

METER: Measurements obtained from 1000 hours of usage during testing.

MUST: More than 98% of the time.

PLAN: More than 99% of the time.

WISH: 100% of the time.

ID: QR?

## 3.5.3 Security

TAG: AdminLoginAccountSecurity

GIST: Security of accounts.

SCALE: If an admin tries to log in to the web portal with a non-existing account then the admin should not be logged in. The admin should be notified about log-in failure.

METER: 1000 attempts to log-in with a non-existing user account during testing.

MUST: 100% of the time.

## 3.5.4 Maintainability

ID: QR19

TITLE: Application extendibility

DESC: The application should be easy to extend. The code should be written in a way that it favors

implementation of new functions.

RAT: In order for future functions to be implemented easily to the application.

DEP: none

## 3.5.5 Portability

Doesn’t exist.

## 4. Prioritization and Release Plan

## 4.1 Choice of prioritization method

## 4.2 Release Plan