Software Requirements Specifications

COS 301: REDIRECTION

2018

Team Members

Stephen Teichert - u16254661

Kyle Wood - u16087993

Russell Dutton - u16016612

Jeffrey Russell - u16010648

Justin Grenfell - u16028440

Byron Antak - u16039689

Contents

1 Introduction

The aim of the project is to design a user-friendly and intuitive system that gives users the ability to configure bot scripts for the video game DOTA 2, according to a set of predefined characteristics (desire to push, defend, roam, defeat Roshan etc.). This is in order to allow a non-technical user the ability to control how the bots behave in-game. The different options will allow the user to influence the way in which the game is played by the bot team, from a macro level such as the expected length of the game, to a micro level, such as the choices the bots make when purchasing items and upgrading their character's abilities. The configuration options selected by the user will be converted to Lua scripts and the user will be able to download their newly generated scripts.

2 Purpose

This document is intended to highlight key requirements of the software that is in development, to aid in the design of such software and to serve as a reference for what needs to be accomplished.

3 Scope

The system is still under development and no name has been decided upon for the product. The system will provide a simple interface to customize bot behaviour according to a predefined set of behaviours and actions. The system, however, will **NOT** allow the user to specify exact behaviour (such as always moving in zigzags, or always killing Roshan in the first 2 minutes). Rather, it will allow the user to specify generic behaviour. We are not sure exactly how far we will get with this system and as a result we have decided that we will have the following as non-negotiable.

For team level options, the following will be possibilities:

- Whether the bots want to push early for a win, or delay until late-game, which will be controlled by when the bots want to farm, push and delay lanes. E.g. Bots wanting to delay would likely want to defend lanes more than push them out, and farm a lot
- When to attempt to kill Roshan to pick up the Aegis
- When to buy back to defend a push
- If and when to roam to gank other heroes.

Should we complete the above well before the end of the project, we will attempt to include the following:

For each hero the user will be able to specify:

- A custom Ability progression
- A custom Item progression
- A custom Talent progression
- Lane Assignment (Top, Bottom or Middle)

4 Overview

5 Overall Description

The system that we are developing will consist of the following aspects:

- A web-based interface where the users are able to log in as well as specify and manage bot configuration options for the custom scripts they wish to create. Users are also able to download the generated script from the configuration options that they specified.
- A backend server in which the scripts are generated for each user that requests it.
- A database to store multiple configuration collections for each user.

6 Product Perspective

6.1 User Interfaces

The system consists of a web application interface which can be accessed through a web browser from a user device. The system accommodates for all forms of devices through a responsive interface.

6.2 Software Interfaces

The Angular application represents the client application which handles the front-end interaction by the user. The Express server represents the controlling and information processing of the system, and handles all requests and responses between the Angular application and the database.

6.3 Communications Interfaces

The client application communicates with the system server when making resource requests. The Express server is responsible for database communication and handles data insertion, modification, and retrieval.

6.4 Operations

This section is in progress.

6.5 Product Functions

With the web application, users will be able to create new bot configurations which can then be saved and downloaded as a LUA script to their machine. The user will also be able to view a list of bot configurations which they previously created and saved, as well as modify, delete, and download the configurations. Users can create bot teams, where previously created bots can be added to a specified team. These teams can then be modified or removed.

6.6 User Characteristics

We assume that the users have a novice to intermediate understanding of computers and computing systems, because the users play DOTA 2 and hence are comfortable with basic usage and operation of a computer. We also assume that the user understands DOTA and its terminology (gank, push, mid, etc.)

6.7 Constraints

1. Due to a limited (or non-existent) budget, the product must make use of free plug-ins, frameworks and other necessary components
2. Our team members do have experience with similar technologies however the project is of a larger scale than what we have done in the past

6.8 Assumptions and Dependencies

1. We assume that the users have a novice to intermediate understanding of how to operate a computer
2. We are dependent on the Node.js framework and many packages provided by the framework such as expressjs and angularjs
3. We are using the MVC architecture for this project

7 Specific Requirements

7.1 Functional Requirements

1. A user shall be able to login with existing Facebook or Google accounts.
2. A user shall be able to register with the product (via Auth0) by providing specific information (such as First name, last name, user name, password etc.)
3. A user shall be required to verify their email after signing up by following a link sent to them via their email if they chose not to use Facebook or Google accounts

4. A user shall be able to logout of the system
5. A user should be automatically logged back into their account if they were logged in previously within a specified time period
6. A user shall be able to view all of the bots that they have created and saved thus far
7. A user shall be able to delete a bot configuration which they have made
8. A user shall be able to modify the configuration of an already existing bot
9. A user shall be able to assemble bots into a team
10. A user shall be able to view all teams that they have assembled
11. A user shall be able to delete any team which they have created
12. A user shall be able to choose whether any of their bots are public or private
13. A user shall be able to view another users bot if it is public and they know the URL for that bot
14. A user shall be able to interact with a configuration form to determine what generic behaviour the bots should have
15. A user shall be able to save a specific bot configuration to their account
16. A user shall be able to download the configuration as a LUA script or scripts with instructions regarding the proper installation of the bot script and how to run it
17. A user shall be provided with a video tutorial on the site to indicate how the product can be used and to guide them through the initial steps to ensure that their DOTA 2 environment is correctly set-up (the installation of DOTA 2 Workshop Tools)

7.2 Performance Requirements

1. The system needs to be able to handle multiple users logging in and making requests at the same time.
2. Generation of LUA scripts should not take longer than 10 seconds (self-imposed)

7.3 Software System Attributes

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

7.3.1 Reliability

1. The system should show the correct list of bot configurations and teams created by the user.
2. The system should generate the LUA scripts correctly based on the given bot configuration input.
3. The system should handle errors by displaying appropriate messages without compromising the system's integrity.

7.3.2 Availability

1. The system should be made available at least 99% of the time, not considering network failures.
2. The application should be able to connect to the internet in order to communicate with the database.

7.3.3 Security

1. The system should facilitate secure login and authentication using Auth0 and JSON Web Tokens (JWT).

7.3.4 Maintainability

1. The system should be able to scale well to the increasing number of users on the system.
2. The system structure should be modular to adhere to the concept of low coupling.
3. Coding standards should be adhered to throughout the entire system in order to enhance clarity and readability.

7.4 Design Constraints

1. The product must be implemented as a website
2. The system should use token based authentication
3. The system should be designed to be able to scale up if the need arises
4. The system should make use of a front-end and a back-end system which can be separated and still function together

7.5 Other Requirements

This section is in progress.