Functional and Architectural Requirements Design Document

COS 301: REDIRECTION

2018

Team Members

Stephen Teichert - u16254661

Kyle Wood - u16087993

Russell Dutton - u16016612

Jeffrey Russell - u16010648

Justin Grenfell - u16028440

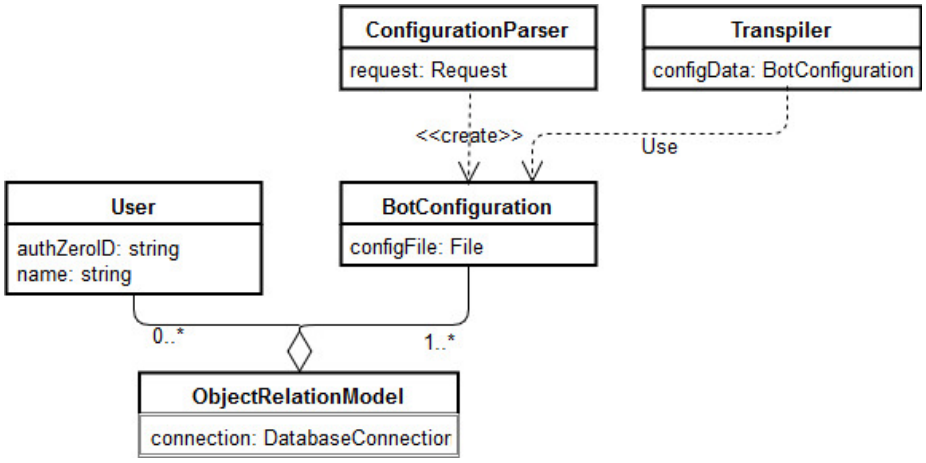
Byron Antak - u16039689

Contents

1 Functional Requirements

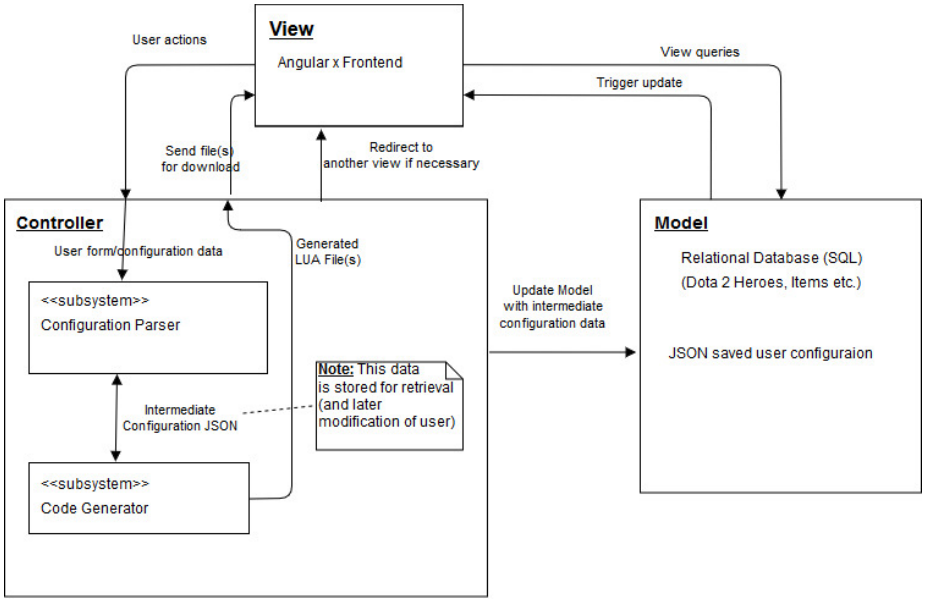
1. A user shall be able to login with existing Facebook or Google accounts.
2. A user shall be able to register with the product (via Auth0) by providing specific information (such as First name, last name, user name, password etc.)
3. A user shall be required to verify their email after signing up by 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(s) 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)

2 Domain Model



Within the Domain Model there are 3 major sections and 2 minor sections. The major sections are User, BotConfiguration and ObjectRelationModel. The minor sections are the ConfigurationParser and the Transpiler. The User Interacts with the front-end and manipulates a BotConfiguration. The BotConfiguration transpiles this into Lua scripts using the ConfigurationParser and the Transpiler. The User can save and retrieve saved configurations to and from the Database, which is represented by the ObjectRelationModel.

3 Architectural Design Structure



The architectural structure design of the system implements both the MVC and Micro services styles.

The MVC is broken down into the View, which consists of the Angular Front-end that uses actions from the controller, sends view queries and receives updates to and from the Model, and receives lua files for downloading from the Controller. The Model consists of the MySQL relational database, which contains JSON objects to store the bot configuration. Finally, the Controller consists of the Express server, which controls the ConfigurationParser and CodeGenerator, which interprets a JSON object and converts the object to Lua code.

The system is also a Micro Service architecture because of the way that the front-end (Angular) and back-end (Express) can be run in entirely separate environments as two different systems.