Software Requirement Specification

By: Corbin Robinson, Ethan Smith, Zaniken Gurule, Parker Rosenberg

10/16/20

Table of contents

Revision History

Version	Editors	Date
v1.0	Corbin Robinson, Ethan Smith, Parker Rosenberg	10/16/20

1. Introduction:
    1.1. Purpose: The purpose of this document is to outline the board game recommender,which provides a way for users to get board game recommendations, via a web server.
    1.2. Conventions:

    Uwsgi: a web hosting service.

    Django: a python web framework

    Python: the primary programming language being used for the product

    1.3. Intended audience: The intended audience for this project is anyone who enjoys board games, and has played more than 3 different board games, and would like to find more, similar board games.
2. Overall Description
    2.1. Product description: The application will be built on Nginx, django, and uwsgi, using python. It will be installable on any unix based server.
    2.2. Product features: It will essentially take one or more board games as input and give back games with similar mechanics, difficulty, or theme.
    2.3. Operating environment:
    Django Web application

    Sqlite database hosting
    Nginx forward proxy
    Linux host machine

    2.4. Constraints: Requires internet connection. Requires database connection.
    2.5. User documentation: Documentation will be provided on a page in the website.
    2.6. Assumptions and dependencies: n/a
3. System Features
    3.1. Functional requirement 1
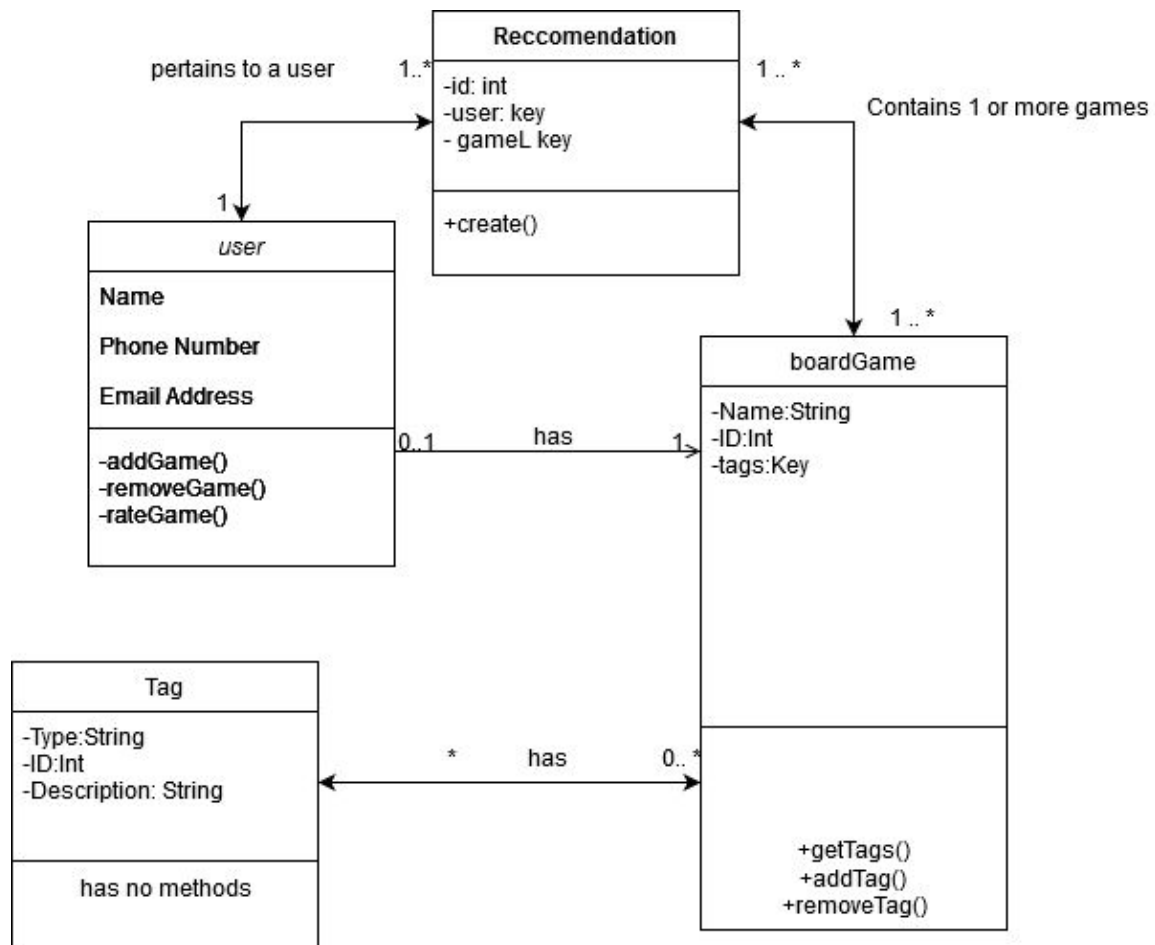   3.1.1. Must accept 1 or more board games as input and recommend 1 or more board games based on mechanics, theme, or difficulty.
        3.1.2. Stimulus: user inputs 1 or more board games by selecting games from a drop down menu or by searching them with a text box
        Response: system displays 1 or more board games that are related in a given way.
        3.1.3. Functional requirements

     3.1.3.1. If a user selects to have recommendations based on mechanics, the system will return based on mechanics.

     3.1.3.2. If a user selects to have recommendations based on theme, the system will return based on theme.

     3.1.3.3. If a user selects to have recommendations based on difficulty, the system will return based on difficulty.

     3.1.3.4. If a user selects to have recommendations based multiple attributes, the system will return based on those attributes.

  3.2. Functional requirement 2 *(this is a secondary goal requirement)*

    3.2.1. A user should be able to log into an account if they enter the username and password correctly

    3.2.2. Stimulus: user inputs username/password correctly into corresponding text boxes
Response: systems gives user access to account matching the username/password given

    3.2.3. Functional requirements

     3.2.3.1. Regular user logon

     3.2.3.2. Admin logon

  3.3. Functional requirement 3

    3.3.1. The system should display an error message when a user enters an incorrect username/password

    3.3.2. Stimulus: user inputs username/password incorrectly into corresponding text boxes
Response: system displays error message in the login box

  3.4. Functional requirement 4

    3.4.1. Users will make their selections by choosing from a drop down list or by searching for a game they like.

     3.4.1.1. Drop downs or search bars for all the attributes listed above:
- Mechanics
- Theme
- Difficulty
- Multiple attributes
- (maybe a star-based system if our stretch goals are reached.)

4. Nonfunctional Requirements

  4.1. Board game database contains at least 20 board games

  4.2. WebApp works on Google Chrome and FireFox

Appendix A

Reccomendation
-id: int
-user: key
- gameL key

+create()

pertains to a user   1..*

1 .. *   Contains 1 or more games

1

user
Name

Phone Number

Email Address

-addGame()
-removeGame()
-rateGame()

0..1   has   1

1 .. *

boardGame
-Name:String
-ID:Int
-tags:Key

+getTags()
+addTag()
+removeTag()

Tag
-Type:String
-ID:Int
-Description: String

has no methods

*   has   0.. *

Appendix B

Game Recommender Use Case Diagram

- log in
- authenticate
- <<extend>>
- <<includes>>
- failed login
- add a game
- <<extend>>
- rate a game
- <<extend>>
- check owned games
- User
- get a reccomendataion
- <<extend>>