

**Project Abstract**

Our app will be used as a tool that gamers can use to connect with each other, share their personal collections, recommend fun games to each other, and help organize the process that board game enthusiasts go through when deciding on a board game to play. The goal of this app is to expand the horizon of gamers who are looking for new, interesting and exciting games they haven't heard of. This software will keep track of board game statistics and libraries, allow for accurate game recommendations, be a platform on which board games and topics related to board games may be discussed.

**Customer**

A person might want to play a board game but might be unsure of which one to choose within his/her collection. A person might want to find a new board game to play but doesn't know what the popular games currently on the market are. The task of keeping track of an active record of a library of board games becomes very mundane and repetitive when doing it on pen and paper. There are many aspects of choosing, playing, and sharing games that people overlook while they go about their gaming lives. Our app will make the system of playing board games a lot simpler and efficient but automating game libraries and storing data about gamers and gamer preferences. The users of this app will most probably be more intense gamers rather than casual gamers however gamers of all levels will be welcome as the concept of the app is relatively simple. Avid gamers will find our app more useful because of their generally large collections of games and the frequency of their gaming.

**Competitive Analysis**

There are some applications on the web with similar functionality as our app however they have problems with their recommendation strategies. They consider only either games that are similar to a particular game viewed, make broad generalizations in game genres, disregard the user's personal collection in the recommendations, or don't allow choosing games via specific attributes. Our app will have a better, all-encompassing system for recommending board games that will solve all these problems that the other applications have. Applications such as Board Game Geek and Board Games for Me are examples of competitors in this niche.

**User Stories:****Actors**

- Regular User:
  - The customer who is looking for board game recommendations
- Developer:
  - The members of group 17. Will produce the code.

## Stories

<b>Iteration 1</b> -User Account Creation	
<b>Actors:</b>	Regular User
<b>Trigger/Pre-condition:</b>	User clicks "Create new Account" button. User is taken to account creation site, whereupon a username and password can be entered. The password is entered twice to confirm the password is entered correctly. Error messages will be given if data is entered incorrectly.
<b>Action/Post-condition:</b>	A new account will be stored in the database, the customer will be taken to their account page with their personal info
<b>Tests:</b>	<ul style="list-style-type: none"> <li>-Assert Username exists</li> <li>-Assert Username is unique</li> <li>-Assert Password exists</li> <li>-Assert Password length &gt;= 6</li> <li>-Assert Password confirmation = Password</li> </ul>

<b>Iteration 2</b> -User Account Creation	
<b>Actors:</b>	Regular User - Brandon
<b>Trigger/Pre-condition:</b>	Brandon wants to create a new account. He clicks the "Create New Account" button. He enters his desired username once in the username field, once in the password field, and the password once again in the password confirmation field.
<b>Action/Post-condition:</b>	A new user account is created in the database. Brandon is taken to his account page.
<b>Tests:</b>	<ul style="list-style-type: none"> <li>-Assert "Create New Account" redirects webpage to the correct page</li> <li>-Assert Username exists</li> <li>-Assert Username is unique</li> <li>-Assert Password exists</li> <li>-Assert Password length &gt;=6</li> <li>-Assert Password confirmation = Password</li> </ul>

<b>Iteration 1</b> -User Login	
<b>Actors:</b>	Regular User
<b>Trigger/Pre-condition:</b>	User visits the homepage clicks the login link and is taken to a login page
<b>Action/Post-condition:</b>	The system will validate the username and password. Login will be forbidden if credentials are invalid and an error presented to user.
<b>Tests:</b>	Assert User exists in database Assert Password = Password associated with User

<b>Iteration 2</b> -User Login	
<b>Actors:</b>	Regular User – Brandon
<b>Trigger/Pre-condition:</b>	Brandon wants to log into his account. He enters his username and password into their respective fields and clicks the submit arrow button.
<b>Action/Post-condition:</b>	The system validates the username and password combination. Brandon is redirected to the homepage of the website if the combination is valid, else an error is presented to him.
<b>Tests:</b>	-Assert Username exists in database -Assert Password entered is the same as the password associated with the username -Assert the redirection goes to the homepage

**Bored?Game!**

## Requirements and Specification Document

November 10<sup>th</sup>, 2015

(Iteration 2)

Git Repo: [https://github.com/JLiman16/boardgame\\_g17](https://github.com/JLiman16/boardgame_g17)App Prototype: [boardgame-g17-epithemeus.c9.io](http://boardgame-g17-epithemeus.c9.io)

<b>Iteration 1</b> -Session Protection	
<b>Actors:</b>	Regular User
<b>Trigger/Pre-condition:</b>	A user attempts to access a portion of the site for which they may not have access
<b>Action/Post-condition:</b>	The user will be denied access to the site if credentials are not confirmed and redirected to an appropriate site
<b>Tests:</b>	Assert User is logged in Assert User = User associated with site visited

<b>Iteration 2</b> -Session Protection	
<b>Actors:</b>	Regular User – Brandon, Regular User - Adam
<b>Trigger/Pre-condition:</b>	Brandon wants to edit another user's profile. He searches Adam's for Adam's profile in the search field. Brandon clicks on Adam's Name and is redirected to Adam's profile. Brandon presses the "Edit Profile" button found on Adam's profile.
<b>Action/Post-condition:</b>	The system recognizes that Brandon is not Adam and denies Brandon access into the editing page for Adam's profile. Error message is presented to Brandon.
<b>Tests:</b>	-Assert Username = Adam OR that current user is an administrator

<b>Iteration 2</b> -BGG Account Syncing	
<b>Actors:</b>	Regular User – Brandon
<b>Trigger/Pre-condition:</b>	Brandon has an account with Bored?Game! and would like to link his account with an account on BGG. Brandon enters the account name of his BGG account in the correct field and presses the “Add Account” button.
<b>Action/Post-condition:</b>	The Bored?Game! system recognizes games from the associate BGG account and displays them in Brandon’s collection on Bored?Game!
<b>Tests:</b>	<ul style="list-style-type: none"><li>-Assert that the BGG account name exists</li><li>-Assert that Brandon’s collection of games on Bored?Game! is updated with games from the associated BGG account</li><li>-Assert that there are no duplicate games resulting from the account sync</li></ul>
<b>Multithread:</b>	Since the API for BoardGameGeek (BGG) limits the number of queries you can make in a period of time, if you query too much at one time, it will give an error. So the code for building a collection has been moved to a separate thread that slowly runs in the background because it takes some time to query BGG.

<b>Iteration 2</b> -Edit User Profile	
<b>Actors:</b>	Regular User – Brandon
<b>Trigger/Pre-condition:</b>	Brandon wants to edit his user profile. He clicks on the “Edit Profile” button. Brandon changes his username and password by entering what he wants into the username, password, and password confirmation fields. He enters Brandon2 as a username and Nguyen2 into password and password confirmation.
<b>Action/Post-condition:</b>	The system takes these new entered strings and updates Brandon’s user account with them. Brandon’s new log in username is Brandon2 and new password is Nguyen2
<b>Tests:</b>	<ul style="list-style-type: none"><li>-Assert Username exists</li><li>-Assert Username is unique</li><li>-Assert Password exists</li><li>-Assert Password length &gt;=6</li><li>-Assert Password confirmation = Password</li></ul>

**Bored?Game!**

## Requirements and Specification Document

November 10<sup>th</sup>, 2015

(Iteration 2)

Git Repo: [https://github.com/JLiman16/boardgame\\_g17](https://github.com/JLiman16/boardgame_g17)App Prototype: [boardgame-g17-epithemeus.c9.io](http://boardgame-g17-epithemeus.c9.io)

<b>Iteration 2</b> -Game Filter/Search	
<b>Actors:</b>	Regular User – Brandon
<b>Trigger/Pre-condition:</b>	Brandon wants to look for an animal game to play with his 8 year old son. He enters 8 into the "Maximum Age" field, 2 into the "Number of Players" field, and animals into the "Category" field. Brandon presses submit.
<b>Action/Post-condition:</b>	The system performs a query on the database of games from all the BGG accounts which have been synced to Brandon. The system returns a list of games that matches the search criteria along with information such as Year, Min. Age, Min. Players, Max Players, Game Time, different categories the game can also be found in, and the mechanics (ie. Dice, cards, etc.) of each game.
<b>Tests:</b>	-Assert that each result of the query meets the filter requirements -Assert that the results of the query belong to one of the associated BGG accounts

<b>Iteration 2</b> -View Collection Images	
<b>Actors:</b>	Regular User – Brandon
<b>Trigger/Pre-condition:</b>	Brandon wants to find out what the game "Catan" looks like so he can physically find it to play. He presses the "View Collection Images" button. He presses ctrl+f and types "Catan" into the find field.
<b>Action/Post-condition:</b>	The system displays grid games in the collection from synced BGG accounts and the associated images with the games.
<b>Tests:</b>	-Assert each image matches the game

**Future Plans for Iteration 3**

- Querying entire BGG database
  - User enters filter criteria into the proper fields
  - System performs query on the very large BGG database
- View recent activity feed for another user
  - User goes to another user's profile by searching for their username and clicking on their profile
  - Recent activity feed for each user should be displayed on each user's profile
    - System pushes recent activity of a user onto a list of activities
- Manually add games to collection
  - User clicks the "Add Game" button and enters in desired attributes into the game attribute fields. User saves game to their collection
  - System creates a new game tuple that's added to the other games in their collection
- Add games from collection into a "Favourites" list
  - User clicks an "Add to Favourites" button that is next to each game in their collection
  - System adds the selected game into a list of "Favourites" within the database
- Uploading a profile picture for their account
  - User presses the "Edit Profile" button, presses the "Change Profile Picture" button, and chooses a picture from their computer.
  - System updates user account with new profile picture

**User Interface Requirements**

View our invisionapp UI Mockup at this link:

<https://projects.invisionapp.com/m/share/QN4TVUXG6?standalone=false#/113482735>

(copy and paste link or it might not work)