

# Final Project for SW Engineering Class CSC 648-848 Fall 2019

Team 203

Cards Against Society

Team Lead: Jose Castanon – jcastan6@mail.sfsu.edu

Back-End Lead: Leslie Zhou – zzhou2@mail.sfsu.edu

Database Master: Shota Ebikawa – sebikawa@mail.sfsu.edu

Git Master: Daryl Ortiz – dortiz4@mail.sfsu.edu

Front-End Lead: Brian Le – ble2@mail.sfsu.edu

<http://54.183.228.36:4000/>

December 19, 2019

## **Table of Contents**

1. Product Summary	2
2. Milestone Documents	
a. Milestone 1v2	4
b. Milestone 2v2	17
c. Milestone 3	57
d. Milestone 4v2	60
3. Screenshots of Final Product	73
4. Screenshots of Database Tables	77
5. Screenshots of Task Management System	79
6. Team Member Contributions	81
7. Post Analysis	83

## **1. Product Summary**

Have you ever come home from a long day out, taken off your Heelys and your bejeweled, boot cut jeans, turned on some Big Time Rush, and sat in your lonely room in nothing but your socks staring at your cardboard cutout of Dolly Parton, contemplating whether to go back into the cold, miserable world in order to meet up with your fake friends for card game night? If so, then Cards Against Society is the game for you!

Cards Against Society is a highly distasteful, strategic game of ad libs, where you pick a card based on it's offensiveness to the ad lib. Players will turns drawing an ad lib card and select the most messed-up/funniest statement to complete that ad lib.

Cards Against Society combines the millennial trend of table-top games with the convenience that technology offers in the modern age. Now you can enjoy the company of your friends from the comfort of your own home. No more feeling guilty for licking your Cheeto crusted finger tips when hanging out with your friends, or feeling that you're missing out on quality time and memories while sitting at home staring at your computer screen in the dark!

Compared to other online Cards Against Humanity clones, Cards Against Society allows you to play multiple matches concurrently to play with two different friend groups. Also, if you wish to create your own drama and make your friends question your friendship, you have the ability to create custom cards.

Functions :

- Unregistered Users
  - Shall be able to create a new account
  - Shall be able to view Terms of Service
- Registered Users
  - Shall be able to log into their account
  - Shall be able to log out of their account
  - Shall be able to view all active games
  - Shall be able to join games
  - Shall be able to create new games
  - Shall be able to come back to a game at any time
  - Shall be able to create custom cards
- Hosts
  - Shall be able to set game rules on room creation
  - Shall be able to start a new round
- Lobby
  - Shall display all open games and its Room Name and the number of players in the room
  - Shall display public chat room
- “My Games” Page
  - Shall display all current active games for the logged in user

- Game Rooms
  - Shall display user's cards
  - Shall display a private room chat
  - Shall display timer for player to submit a play
  - Shall show relevant game information to the player
- Admin
  - Shall be able to modify or remove data entries
  - Shall be able to stop on-going games
  - Shall be able to view all pages
  - Shall be able to delete users
  - Shall be able to see a list of all registered users
  - Shall be able to see a list of all on-going games
  - Shall be able to have access to admin navigation bar

## **2. Milestone documents – M1-M4**

# Cards Against Society

## Milestone 1

SW Engineering CSC648/848 Spring 2019

Section 4

Team 203

03 October 2019

Team Lead: Jose Castanon

Back-End Lead: Leslie Zhou

Database Master: Shota Ebikawa

Git Master: Daryl Ortiz

Front-End Lead: Brian Le

<b><u>Revision #</u></b>	<b><u>Version #</u></b>	<b><u>Date</u></b>
Milestone 1	Version 2	10/10/19
Milestone 1	Version 1	10/3/19

## **Table of Contents**

1. Executive Summary	7
2. Main Use Cases	8
3. List of Main Data Items and Entities	12
4. Initial List of Functional Requirements	12
5. List of non-Functional Requirements	12
6. Competitive Analysis	14
7. High-Level System Architecture and Technologies used	15
8. Team	15
9. Checklist	15

## **1. Executive Summary**

Cards Against Society is an online real time game based on the card game Cards Against Humanity – a game which is most fun when played with friends. In a typical game of CAH, the goal is to reach a certain amount of points to win, one player will choose a black card, which contains a question or blank space. The other players will draw white cards to their hand and then select one of them which contains a response that they think is the funniest response to the black card. After all the players have picked a response, the black card holder will read the prompt on the black card along with each white card to fill the blank space. The black card holder will then pick the response they think is the funniest, and the player who picked that response will gain a point. The game continues like so, and the winner is the person who reaches a set amount of points – usually around five points.

The purpose of this project is to provide the user the same fun experience that they would get if they were playing the game in real life, while allowing them to interact with other people online. Cards Against Society will aim to replicate the experience of playing Cards Against Humanity in real life virtually, while also allowing the users to play the game as normal. They will also be able to communicate in real time with other players and create public or private games. Cards Against Society, unlike other online clones, will focus on simplicity and the user experience which will allow newcomers and returning players to jump into the game with ease.

## **2. Main Use Cases**

### Actors

#### John - Host / User

- College CS student
- Loves playing Cards Against Humanity
- Likes playing with his friends
- Recently moved off to college, misses playing CAH with his friends

#### Paul – Unregistered User

- Has not played Cards Against Humanity before
- Wants to play with his friend who lives far away

#### Karen – Registered User

- Plays Cards Against Humanity online for fun
- Registered user, has played Cards Against Society before

### Use Cases

#### John:

John, a college student, loves playing Cards Against Humanity with his friends. Limited by the distance between him and his friends back home, he turns to Cards Against Society to be able to play with them. Upon entering the site, he is prompted to register or log in before continuing. John creates an account and logs on, and is now able to see the main lobby, along with all the ongoing public games and the public chat. He creates a new lobby titled “John’s room” so that his friend can locate his game later on. Inside the game, John sees his friends join. He sends a message inside the rooms private chat saying, “What’s up!”. During the first round of the game, John is the black card holder. He presents the topic with the black card and then he waits for the other people in the lobby to pick a card that they think best responds to the sentence the black card presents. Once the white card holders choose a card, John can see when they picked. He then proceeds to pick his favorite white card and gives a black card to the funniest player and also awards them with points.

During the game, John’s laptop dies. He connects his charger and logs back into the site. In John’s “My Games” page, he can see all the ongoing games that he is in. He finds his last game, joins it, and can continue playing as before. The game continues until someone reaches the needed amount of points to win the game.

Paul:

Paul's friend group is too tired to go outside tonight and all decide to stay home and play computer games instead. His friends decide to play Cards Against Humanity, but Paul has never played it before. He is sent a link to play Cards Against Humanity from his friend and he is prompted to either log in into an existing account, or create a new one. Paul then creates an account by adding an email, password, and selecting a username which all players can see. He is then prompted if he has ever played Cards Against Humanity before, and he selects no. He is given a picture of basic rules and instructions on how to play the game. Upon completion, he is redirected to the locked (private) room which was created and hosted by his friend that sent the link. Once connected into the room, he sees the usernames of his four friends displayed on top of the chat box with the scoreboard values set to zero next to them. He then uses the chat box to announce he is connected into the room, and the host commences the game by giving him five randomly selected cards.

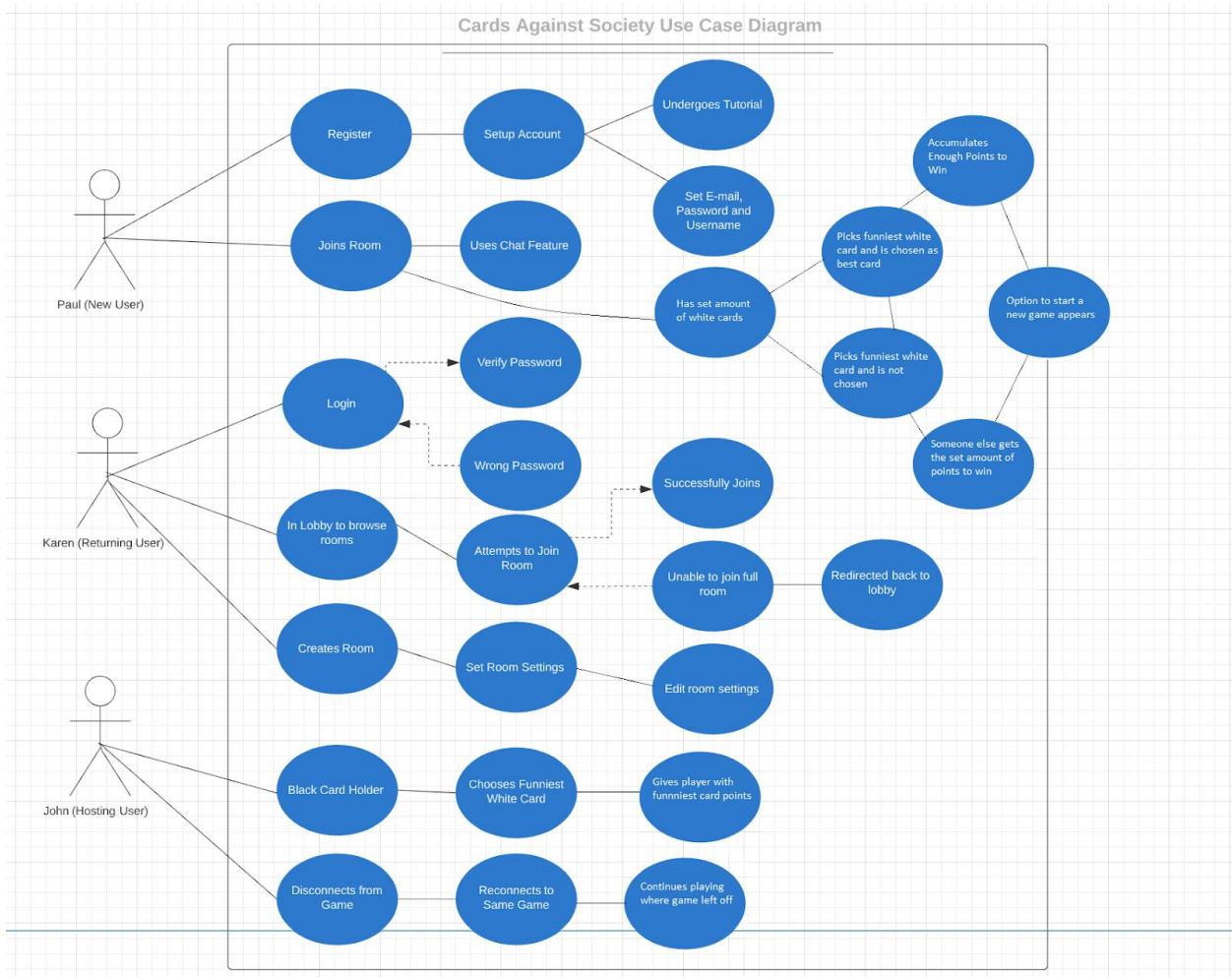
The game begins and the Black Card Holder reveals the black card which prompts the other players to choose a white card which responds best to the contents of the black card. Paul picks a card that the black card holder likes and is therefore given points. If he has enough points then he will win the game. If he wins he will be met with a message signifying that he is the winner. He would then be in the room and have the option to start a new game.

Paul picks a card but the black card holder chooses someone else's card. Someone else will receive the black card and the process repeats itself until someone reaches the set amount of points to win the match.

Karen:

Karen is bored and wants to play Cards Against Humanity online. She logs in into her existing account and browses the homepage for open and non-filled room with a funny name but all the rooms with funny names are filled with the maximum amount of players. Karen decides to host a new room with the name "Funky Monkey" with the settings set to have eight maximum players in the lobby, requires five points to win, and room unlocked to allow random users to join at any time. Once created, a few players instantly join in and Karen has the option to kick a player via button next to their usernames. Next to the chat button, she can access the room settings to edit player limit, point requirement, and whether the room is accessible to the public.

Karen returns to play the app again. When she logs on, the password she entered is incorrect. She is then met with a message that tells her the password is wrong and asks to input the correct password. She then enters the correct and verified password. Once verified she will be able to log in to the game and enter the lobby.



### **3. List of Main Data Items and Entities**

- **User:** Any user must log in/register in order to access the site
- **Registered user:** User who has a registered Cards Against Society account. They are able to join any private or public lobby and participate in games.
- **Unregistered User:** A new user who does not have a Cards Against Society account. They must register in order to continue into the site.
- **Host:** A registered user who creates a public or private room, sets game rules before the start of the game.
- **Lobby:** The homepage for all users, displays all current open games and public chat.
- **“My Games” Page:** The page that displays a user’s active games.
- **Game Rooms:** The page where users can interact with other players and the game. Many users can be in one room.

### **4. Initial List of Functional Requirements**

- New Users
  - Shall be able to create a new account
  - Shall be able to view the lobby
  - Shall not be able to join a game
  - Shall not be able to add another user as a friend
  - Shall be able to spectate an on-going match
- Registered User
  - Shall be able to log into their account
  - Shall be able to log out of their account
  - Shall be able to view all active games in the lobby
  - Shall be able to join games
  - Shall be able to create new games
  - Shall be able to come back to a game at any time
  - Shall be able to send messages through the chat
  - Shall be able to participate in any amount of games concurrently
  - Shall be able to go back and forth between each game
  - Shall be able to add another user as a friend
  - Shall be able to delete friend
  - Shall be able to report users
  - Shall be able to create a profile
  - Shall be able to view friends' profiles
- Host
  - Shall be able to set game rules on room creation
  - Shall be able to kick out users from a game room

- Shall be able to edit game rules in game for next round
- Shall be able to start a new round
- 
- Lobby
  - Shall display all open games and its Room Name and the number of players in the room
  - Shall display a public chat
  - Users can access their account settings
- “My Games” page
  - Shall display all current active games for the logged in user
  - Shall allow the user to leave a game room
  - Shall display rooms that friends are in
  - Shall display friends currently online
- Game Rooms
  - Shall display user’s cards
  - Shall display a private room chat
  - Shall show relevant game information to the player
  - Shall have a AFK option for player who needs a timeout
  - Users can invite friends to the current Game Room

## **5. List of non-Functional Requirements**

- Compatibility
  - System shall be compatible with at least two of the major browsers of Mozilla Firefox, Google Chrome, Safari, and Edge.
  - System shall be compatible with version 67, 68, and 69 of Mozilla Firefox.
  - System shall be compatible with version 75, 76, and 77 of Google Chrome
  - System shall be compatible with version 10, 11, and 12 of Safari.
  - System shall be compatible with version 41, 42, and 44 of Edge.
- System
  - Arrow functions shall be implemented for the on-event handlers.
  - JSX/HTML tags shall have no whitespace between its element name and its angle brackets.
  - New lines shall be started after the opening curly brackets when objects and functions are being initialized.
  - Each objects shall be in a separate line when more than 4 objects from a given module are imported
  - Inline-styles shall be encapsulated in a single variable.

- Application shall be developed using the software stack declared in M0
  - Application shall be deployed using an AWS EC2 instance
  - UI shall be implemented using React
  - Application shall run off of the Master branch of the team's git repo
  - Game animations shall be implemented using CSS
- 
- Content
    - Images shall be in the format of either jpg, png, or jpeg.
    - Query results shall be displayed in alphabetical ascending order.
  
  - Storage, security, environmental requirements
    - Data shall be stored in a MySQL database
    - Game states shall be persistent, saved by the system and loaded when a player joins
    - Application shall allow up to 5 users in one game room
    - Application shall keep public and private chat in their respective places
    - Application shall load a default game state when a room is created
    - Database shall store all of the game cards to be used during the game
  
  - Marketing, legal requirements
    - An individual shall receive permission from the copyright holders to use Cards Against Society's SW code.
    - An individual shall not use Cards Against Society's Logo as one's trademark
    - An individual shall not disclose Cards Against Society's API as it is part of the company's trade secrets.
  
  - Privacy
    - Only relevant data shall be collected, such as user's win rate
    - Password collected shall be encrypted in the database
  
  - Performance
    - System shall update game data in real time
    - System shall update chat message data in real time using sockets
    - Application shall have a responsive game UI
    - Application shall display game without the use of images

## 6. Competitive Analysis

	Pretend Xyzy	You're	Pictures Humanity	Against	Cards Against Society
<b>User Interface</b>	+		+		++
<b>Custom Cards</b>	-		+		+
<b>In-game Chat</b>	+		+		++
<b>Private Room</b>	+		+		+
<b>User Auth</b>	+		+		+
<b>Tutorial</b>	-		-		+

In comparison to our competitors, we plan to implement a more welcoming and exciting user interface. We found that our competitors were set up to be played by people who are already familiar with the game. To improve this, we will add a tutorial for those who are not familiar with Cards Against Humanity, which will make it easy for newcomers to start playing. Our competitors offer similar features to each other, and we plan to implement those features as well. Unlike our competitors, we will keep chat public or private depending on whether the user is in a game room or the lobby.

## **7. High-Level System Architecture and Technologies used**

### Web Server

- Host: Amazon Web Services 1vCPU 1GB RAM
- OS: Ubuntu Server 16.04

### Technologies

- Server side language: Javascript
- Front end: React
- Backend: Node.js, Express.js, Socket.io
- Database: MySQL 8.0.17

### Supported Browsers:

- Mozilla Firefox: (ver. 67, ver. 68, ver. 69)
- Google Chrome: (ver. 75, ver. 76, ver. 77)
- Safari: (ver. 10, ver. 11, ver. 12)
- Edge: (ver. 41, ver. 42, ver. 44)

## **8. Team**

- Team Lead: Jose Castanon
- Back-End Lead: Leslie Zhou
- Database Master: Shota Ebikawa
- Git Master: Daryl Ortiz
- Front-End Lead: Brian Le

## **9. Checklist**

- Team found a time slot to meet outside of class - **DONE**
- Github master chosen - **DONE**
- Team decided and agreed together on using the listed SW tools and deployment server - **DONE**
- Team ready and able to use the chosen back and front end frameworks and those who need to learn are working on learning and practicing - **DONE**
- Team lead ensured that all team members read the final M1 and agree/ understand it before submission - **DONE**

- Github organized as discussed in class (e.g. master branch, development branch, folder for milestone documents etc.) - **DONE**

# Cards Against Society

## Milestone 2

SW Engineering CSC648/848 Fall 2019

Section 4

Team 203

7 November 2019

Team Lead: Jose Castanon – jcastan6@mail.sfsu.edu

Back-End Lead: Leslie Zhou – zzhou2@mail.sfsu.edu

Database Master: Shota Ebikawa – sebikawa@mail.sfsu.edu

Git Master: Daryl Ortiz – dortiz4@mail.sfsu.edu

Front-End Lead: Brian Le – ble2@mail.sfsu.edu

<u>Revision #</u>	<u>Version #</u>	<u>Date</u>
Milestone 2	Version 2	11/7/19
Milestone 2	Version 1	10/24/19
Milestone 1	Version 2	10/9/19
Milestone 1	Version 1	10/3/19

## **Table of contents**

1. Data Definitions V2	19
2. Functional Requirements V2	20
3. UI Mockups and Storyboards	22
4. High Level Database Architecture and Organization	46
5. High Level APIs and Algorithms	53
6. High Level UML Diagram	54
7. High Level Application Network and Deployment Diagrams	54
8. Identify <i>actual</i> Key Risks for Your Project at this time	56
9. Project Management	56

## **1. Data Definitions V2**

Unregistered User : A user who does not have an account or is not currently logged in. They are limited to only view on-going Game Rooms and the public chat on the application.

Registered User : A user who has an account and is currently logged in. These users have full access to all the functions the application will have. They can participate in public or private games, add friends, chat in public chat, view their “My Games” page, and host games.

Host : A user who creates and has control of a game room. The host initializes a room based on their choice of points to win, public or private match (whether the room will be listed as “open” on the lobby screen”, and the amount of players that may join the room. The host will also have access to and ability to change the current game’s setting: points to win, room size limit, private or public, and remove players from the room.

Lobby : The homepage for all registered users. The Lobby page will display all “Open” listed Game Rooms, along with the amount of current players in the room, points needed to win, and Game Room name. Next to the Game Room list, a Public Chat will be displayed and show all on going communication from other users on the Lobby screen. The Lobby will also display online friends on a separate tab of the Public Chat.

Game Room : A Game Room is a room created by a Registered User that becomes the Host of the current Game Room. The Game Room exists as a play space environment for the Registered Users, where the Host can start a game and a set number of cards are distributed to each player in the room. Inside exists a Game Room chat that only players in the Game Room can see, the player’s cards displayed for them – yet hidden from the other players, and the scoreboard and current Black Card holder.

“My Games” Page : Each user gets a specific “My Games” Page which will display all current on-going games the user is participating in. The user can view and rejoin any of the games displayed as long as the Game Room has not ended.

Chat : A chatting mechanism for all users to send messages in either a global or private room chat. This chat has little filter on what users may say.

Admin : A user that is preselected to moderate and have all power to modify all database entries, as well as to stop any on-going Game.

## 2. Functional Requirements V2

### ★ Priority 1

- Unregistered Users
  - Shall be able to create a new account
  - Shall be able to view Terms of Service
  - Shall be able to view the lobby
- Registered Users
  - Shall be able to log into their account
  - Shall be able to log out of their account
  - Shall be able to view all active games
  - Shall be able to join games
  - Shall be able to create new games
  - Shall be able to come back to a game at any time
  - Shall be able to create custom cards
- Hosts
  - Shall be able to set game rules on room creation
  - Shall be able to start a new round
- Lobby
  - Shall display all open games and its Room Name and the number of players in the room
  - Shall display public chat room
- “My Games” Page
  - Shall display all current active games for the logged in user
- Game Rooms
  - Shall display user’s cards
  - Shall display a private room chat
  - Shall display timer for player to submit a play
  - Shall show relevant game information to the player
- Admin
  - Shall be able to modify or remove data entries
  - Shall be able to stop on-going games
  - Shall be able to view all pages
  - Shall be able to see a list of all registered users
  - Shall be able to see a list of all on-going games
  - Shall be able to have access to admin navigation bar

### ★ Priority 2

- Unregistered Users
  - Shall not be able to add another user as a friend
  - Shall be able to spectate an on-going match
- Registered Users
  - Shall be able to send messages through the chat
  - Shall be able to participate in any amount of games concurrently

- Shall be able to go back and forth between each game
- Shall be able to add another user as a friend
- Shall be able to delete friend
- Shall be able to see profile settings
- Hosts
  - Shall be able to kick out users from a game room
  - Shall be able to edit game rules in game for next round
- Lobby
  - Users can access their account settings
- “My Games” Page
  - Shall allow the user to leave a game room
- Game Room
  - Shall have a AFK option for player who needs a timeout
  - Users can invite friends to the current Game Room
- Chat
  - Users can chat globally with random users
- Custom Cards
  - Users shall be able to vote for card decks
  - Users shall be able to create new decks
  - Users shall be able to edit decks
- Admin
  - Shall be able to delete users

### ★ Priority 3

- Registered User
  - Shall be able to report users
  - Shall be able to create a profile
  - Shall be able to view friends' profiles
- “My Games” page
  - Shall display rooms that friends are in
  - Shall display friends currently online
  - Shall display number of players in rooms
- Custom Cards
  - Users shall be able to use set popular decks

### 3. UI Mockups and Storyboards

Logged In Lobby

Cards Against Society	My Games	Help																
			<span style="font-size: small;">\$USERNAME</span>															
<span style="border: 1px solid black; padding: 5px; display: inline-block;">Create Room</span>			<span style="font-size: small;">Profile</span> <span style="font-size: small;">Custom Cards</span> <span style="font-size: small;">Log Out</span>															
<p style="text-align: center;"><b>Game Rooms Active</b></p> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="width: 30%;">Room Name</th> <th style="width: 30%;">Host:</th> <th style="width: 40%;">Join</th> </tr> </thead> <tbody> <tr> <td>NewRoom</td> <td>Admin</td> <td>Join</td> </tr> <tr> <td>Room2</td> <td>User1</td> <td>Join</td> </tr> <tr> <td>Room3</td> <td>User2</td> <td>Join</td> </tr> <tr> <td colspan="3" style="height: 100px;"></td> </tr> </tbody> </table>				Room Name	Host:	Join	NewRoom	Admin	Join	Room2	User1	Join	Room3	User2	Join			
Room Name	Host:	Join																
NewRoom	Admin	Join																
Room2	User1	Join																
Room3	User2	Join																
<span style="font-size: small;">Type your message here...</span> <span style="border: 1px solid black; padding: 2px 5px; margin-left: 10px;">Send</span>																		

Users can view all active rooms and join them by pressing "Join" button

Drop down menu to navigate to user controls

Log in / Sign up

Cards Against Society	Help	Login!
<p>Login Enter your information</p> <p>Username : [input field]</p> <p>Password : [input field]</p> <p><input type="checkbox"/> Remember me    <a href="#">Forgot Password / Username</a></p> <p>Log In [button]</p> <p>No Account? <a href="#">[Sign up here]</a></p>		

Log in / Sign up

Cards Against Society	Help	Login!
<p>Login Enter your information</p> <p>Username : [input field]</p> <p>Password : [input field]</p> <p><input type="checkbox"/> Remember me    <a href="#">Forgot Password / Username</a></p> <p>Log In [button]</p> <p>No Account? <a href="#">[Sign up here]</a></p>		

## Help/FAQ Page

Cards Against Society	My Games	Help		\$USERNAME
-----------------------	----------	------	--	------------

Help Against Society

What is Cards Against Society?

How does one play Cards Against Humanities?

Example Part 1

Example Part 2

Upon click, more information will be revealed to the user



Create Room

Cards Against Society	My Games	Help	\$USERNAME
-----------------------	----------	------	------------

Setup Room

Room Name:

Private

Game Rules

Points to win:

Room Size:

Custom Card Deck:

## Profile

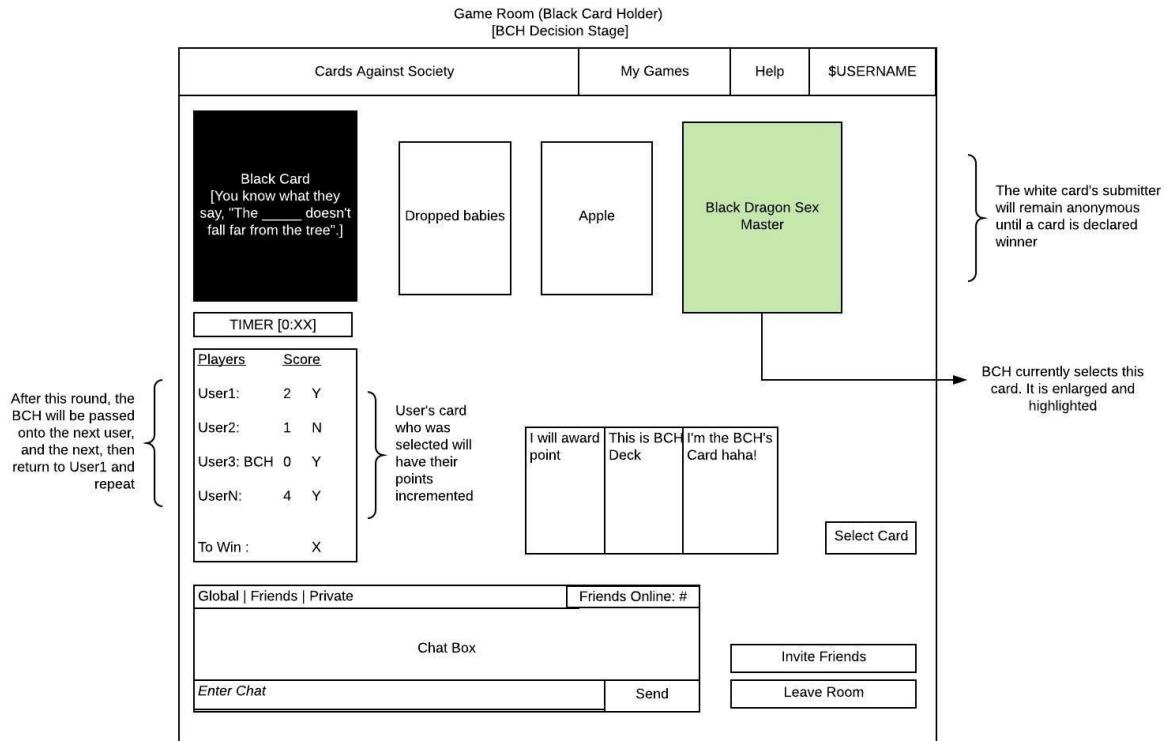
Cards Against Society	My Games	Help	\$USERNAME
<b>Profile</b>			
First Name	<input type="text"/>		
Last Name	<input type="text"/>		
Username	<input type="text"/>		
Email	<input type="text"/>		
Bio	<input type="text"/>		

Game Room (Host)  
[Selection Stage]

Cards Against Society	My Games	Help	\$USERNAME												
<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <p><b>Black Card</b> [You know what they say, "The ____ doesn't fall far from the tree".]</p> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; width: 150px; height: 100px; margin: 5px;"></div> <div style="border: 1px solid black; width: 150px; height: 100px; margin: 5px;"></div> <div style="border: 1px solid black; width: 150px; height: 100px; margin: 5px;"></div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>TIMER [0:XX]</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Submit Card</span> </div> <div style="margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Players</th> <th>Score</th> </tr> </thead> <tbody> <tr> <td>User1:</td> <td>2 Y</td> </tr> <tr> <td>User2:</td> <td>1 N</td> </tr> <tr> <td>User3:</td> <td>0 Y</td> </tr> <tr> <td>UserN:</td> <td>4 Y</td> </tr> <tr> <td colspan="2">To Win : X</td> </tr> </tbody> </table> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <span>Global   Friends   Private</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Edit Game Rules</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span>Chat Box</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Invite Friends</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span>Enter Chat</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Kick Player</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span></span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Leave Room</span> </div>				Players	Score	User1:	2 Y	User2:	1 N	User3:	0 Y	UserN:	4 Y	To Win : X	
Players	Score														
User1:	2 Y														
User2:	1 N														
User3:	0 Y														
UserN:	4 Y														
To Win : X															

Game Room (User)  
[Selection Stage]

Cards Against Society	My Games	Help	\$USERNAME												
<div style="border: 1px solid black; padding: 10px; margin-bottom: 10px;"> <p><b>Black Card</b> [You know what they say, "The ____ doesn't fall far from the tree".]</p> </div> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; width: 150px; height: 100px; margin: 5px;"></div> <div style="border: 1px solid black; width: 150px; height: 100px; margin: 5px;"></div> <div style="border: 1px solid black; width: 150px; height: 100px; margin: 5px;"></div> </div> <div style="display: flex; justify-content: space-between; align-items: center;"> <span>TIMER [0:XX]</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Submit Card</span> </div> <div style="margin-top: 10px;"> <table border="1" style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th>Players</th> <th>Score</th> </tr> </thead> <tbody> <tr> <td>User1:</td> <td>2 Y</td> </tr> <tr> <td>User2:</td> <td>1 N</td> </tr> <tr> <td>User3:</td> <td>0 Y</td> </tr> <tr> <td>UserN:</td> <td>4 Y</td> </tr> <tr> <td colspan="2">To Win : X</td> </tr> </tbody> </table> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 10px;"> <span>Global   Friends   Private</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Invite Friends</span> </div> <div style="display: flex; justify-content: space-between; align-items: center; margin-top: 5px;"> <span>Chat Box</span> <span style="border: 1px solid black; padding: 2px 10px; margin-left: 10px;">Leave Room</span> </div>				Players	Score	User1:	2 Y	User2:	1 N	User3:	0 Y	UserN:	4 Y	To Win : X	
Players	Score														
User1:	2 Y														
User2:	1 N														
User3:	0 Y														
UserN:	4 Y														
To Win : X															



After this round, the BCH will be passed onto the next user, and the next, then return to User1 and repeat

User's card who was selected will have their points incremented

The white card's submitter will remain anonymous until a card is declared winner

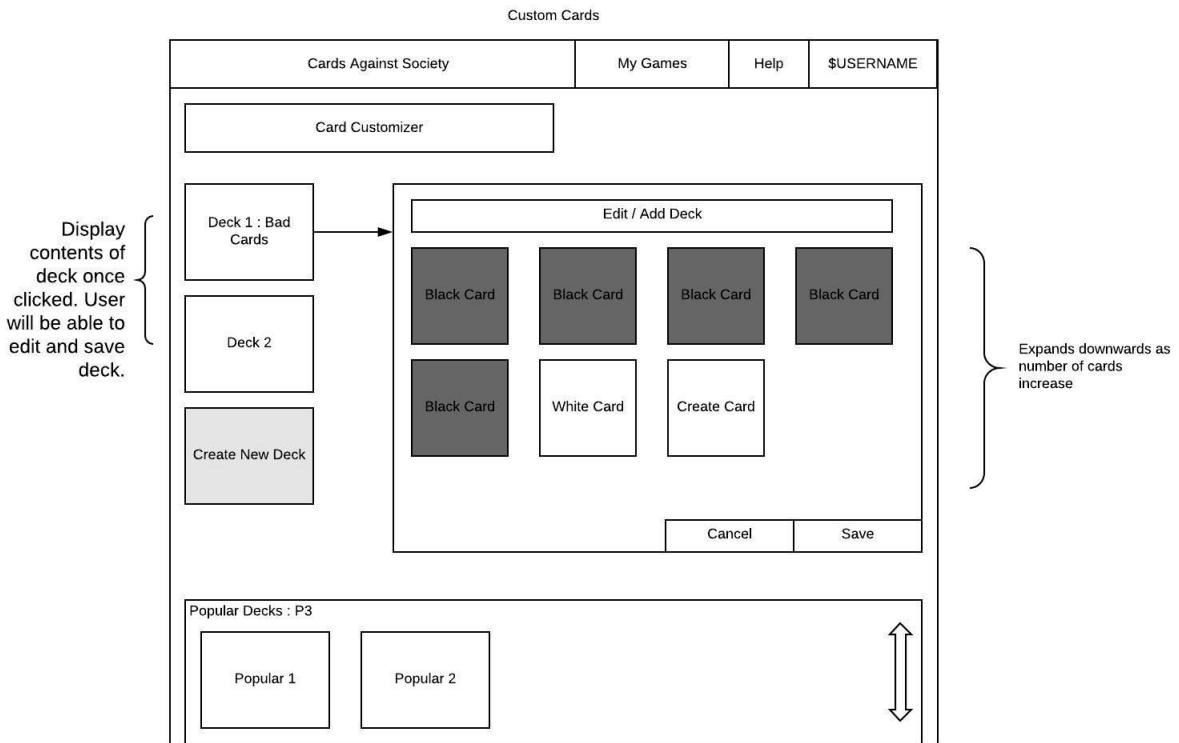
BCH currently selects this card. It is enlarged and highlighted

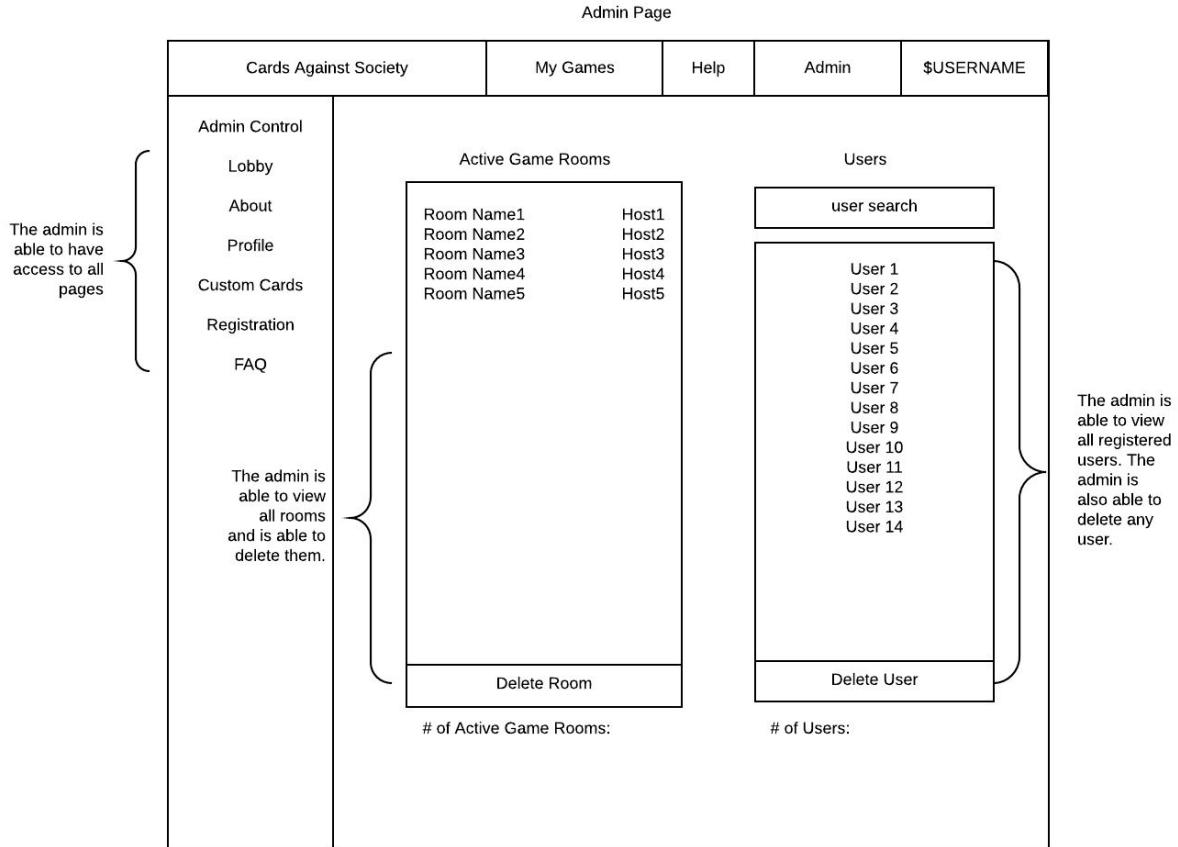
My Games Page

Cards Against Society	My Games	Help		\$USERNAME									
<table border="1"><thead><tr><th>Room Name</th><th>Host:</th><th>Join</th></tr></thead><tbody><tr><td>Room1</td><td>Host1</td><td>Join</td></tr><tr><td colspan="3"></td></tr></tbody></table>					Room Name	Host:	Join	Room1	Host1	Join			
Room Name	Host:	Join											
Room1	Host1	Join											

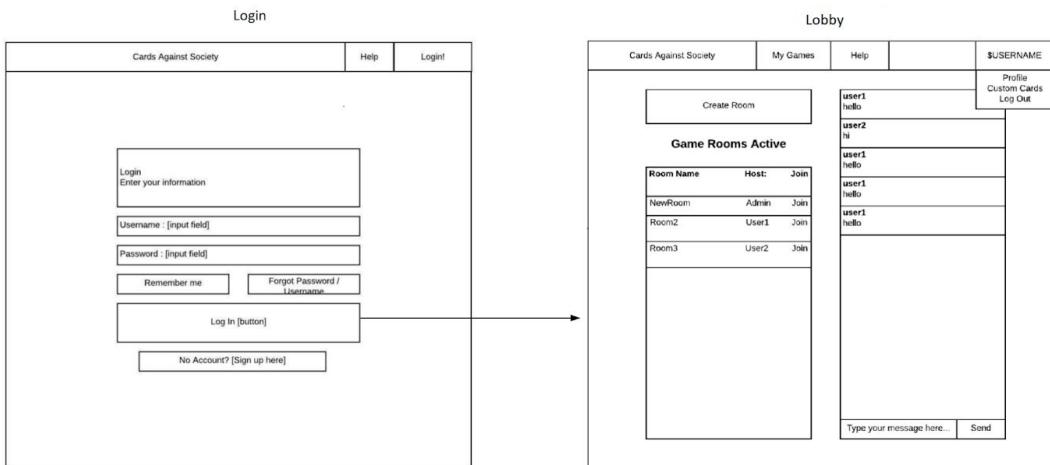
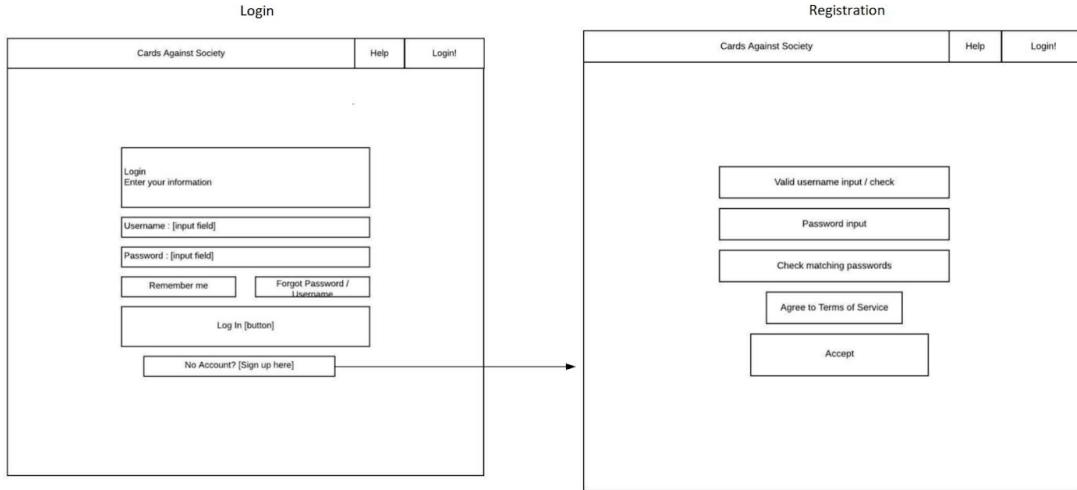
All active Game Rooms and Respective Hosts are displayed

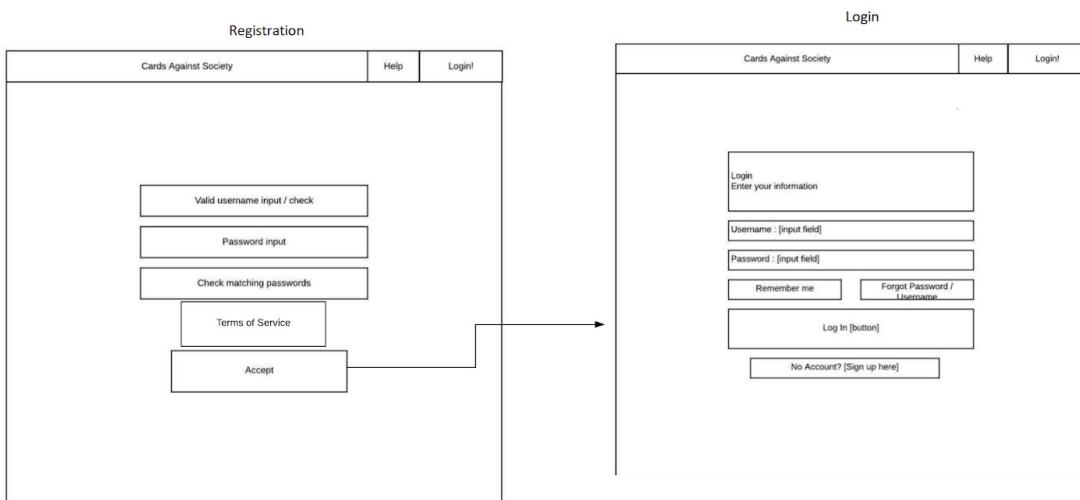
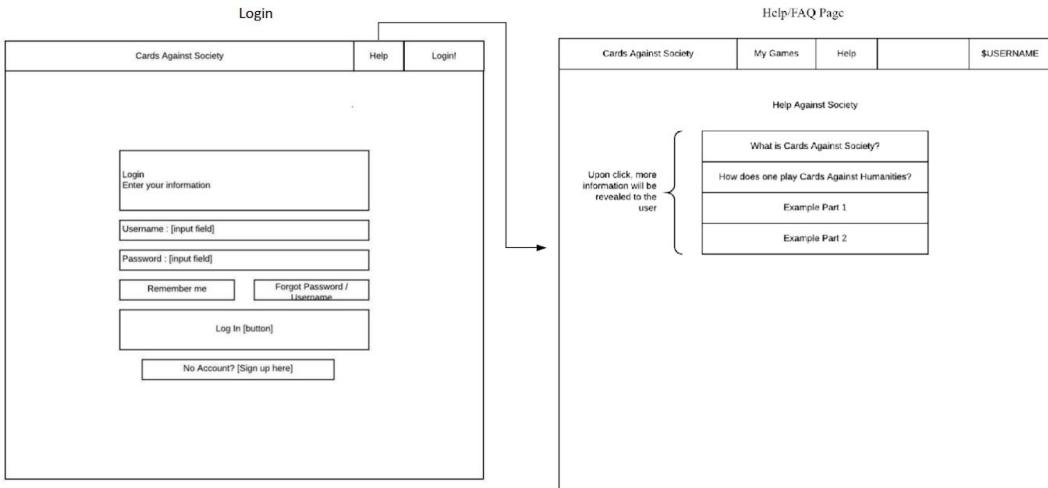
Clicking the "Join" button will allow the user to enter the game room

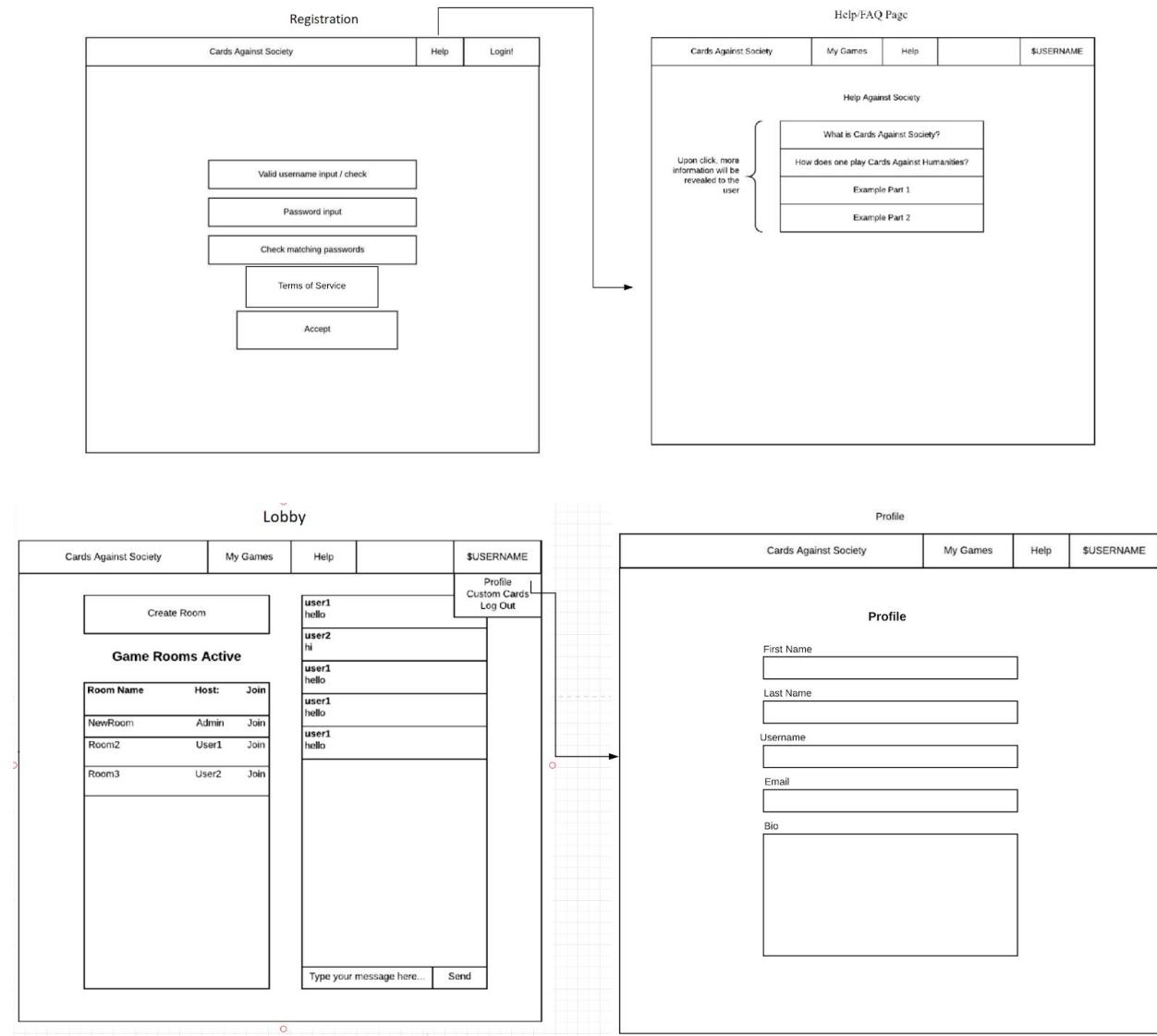


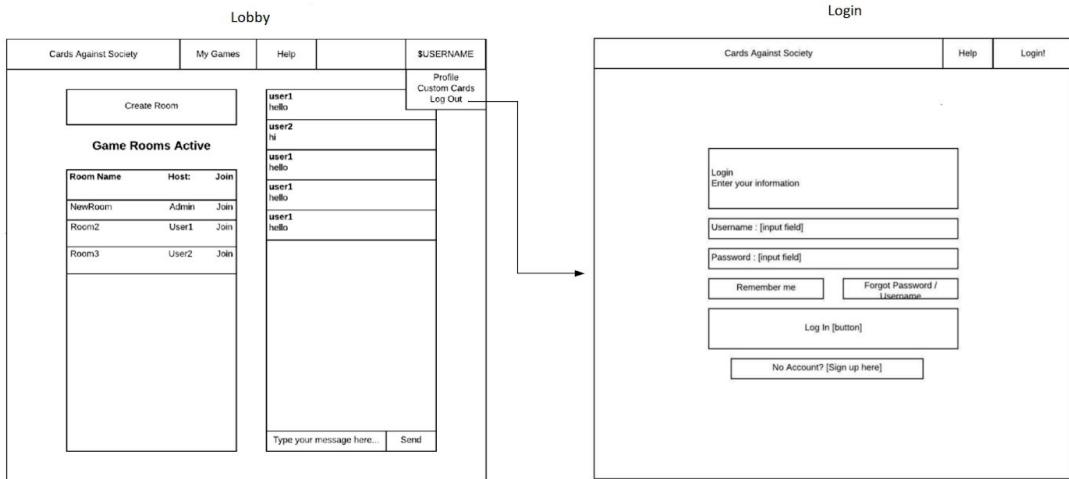
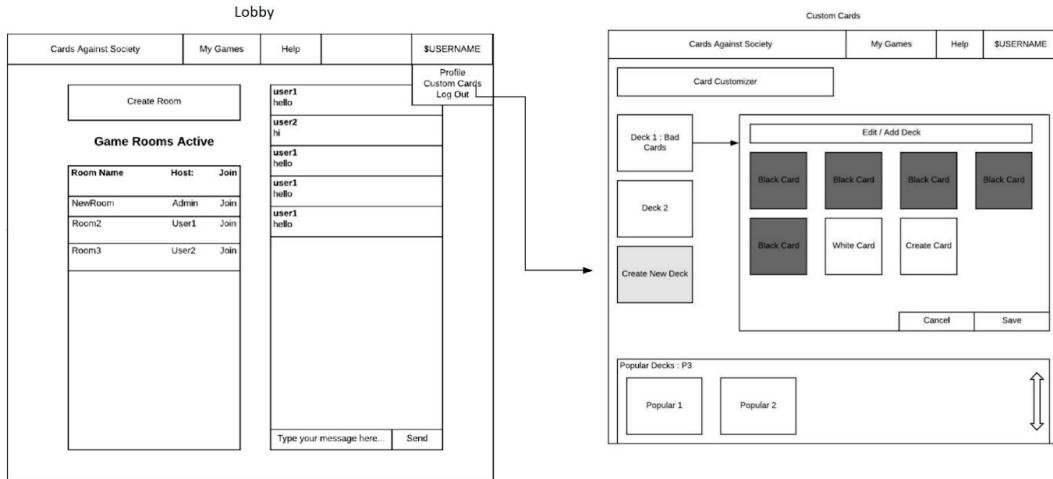


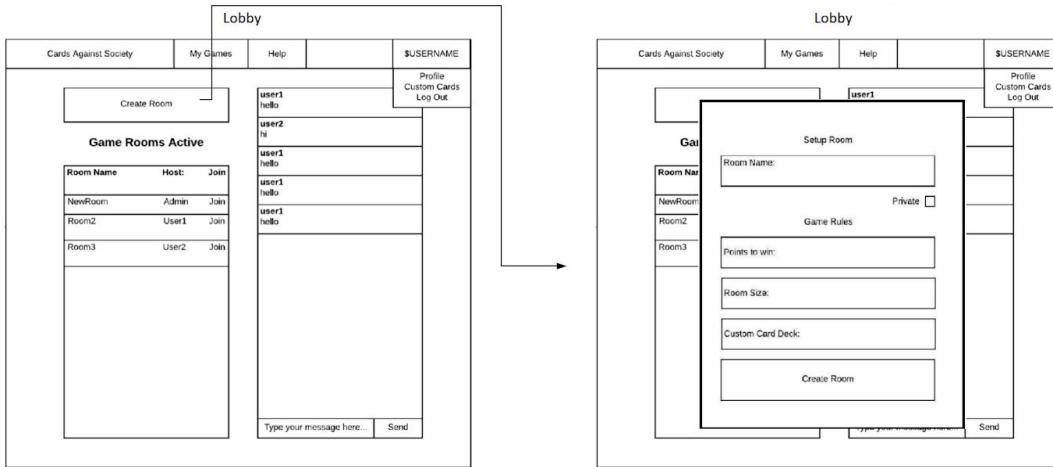
## STORYBOARDS



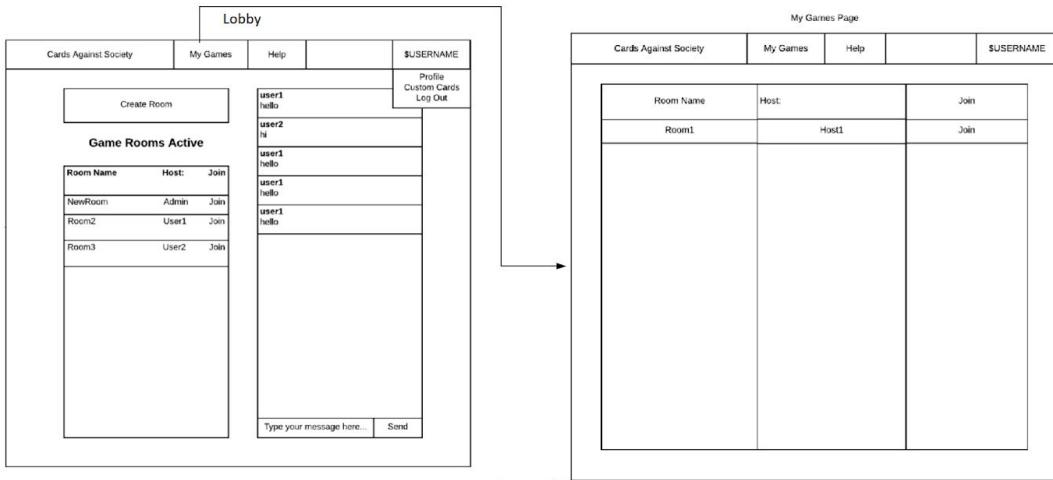


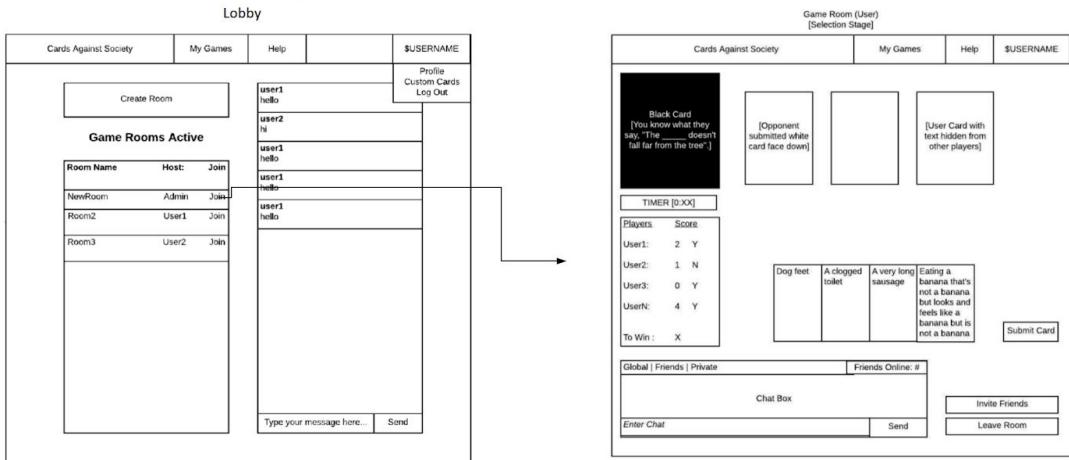
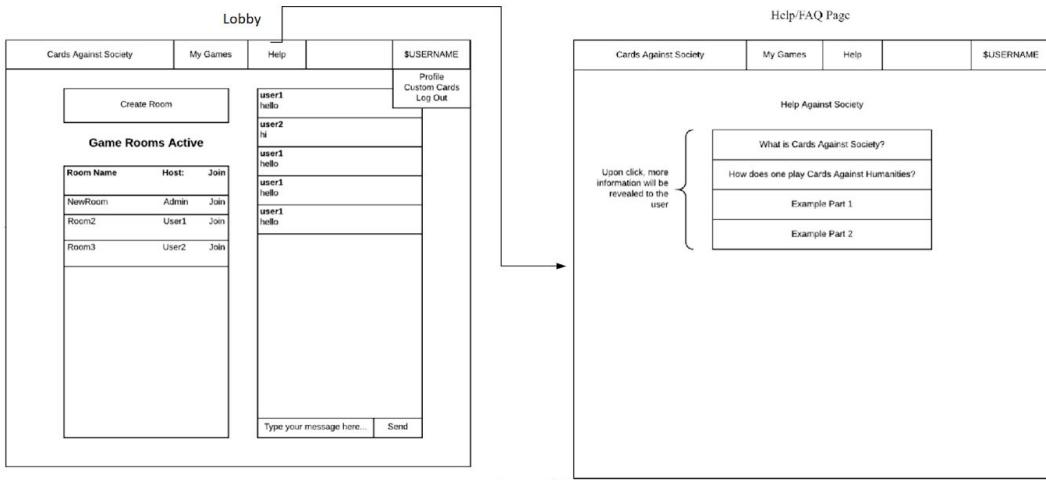


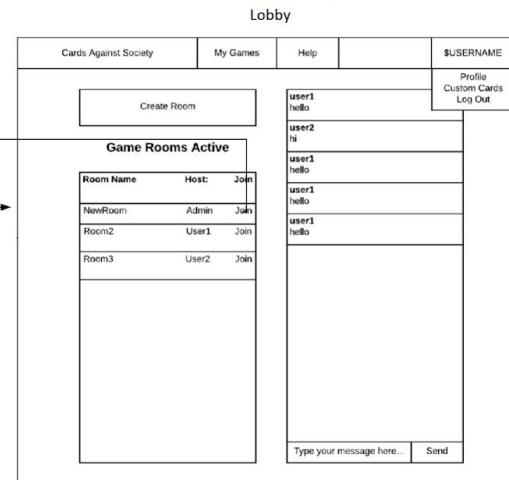
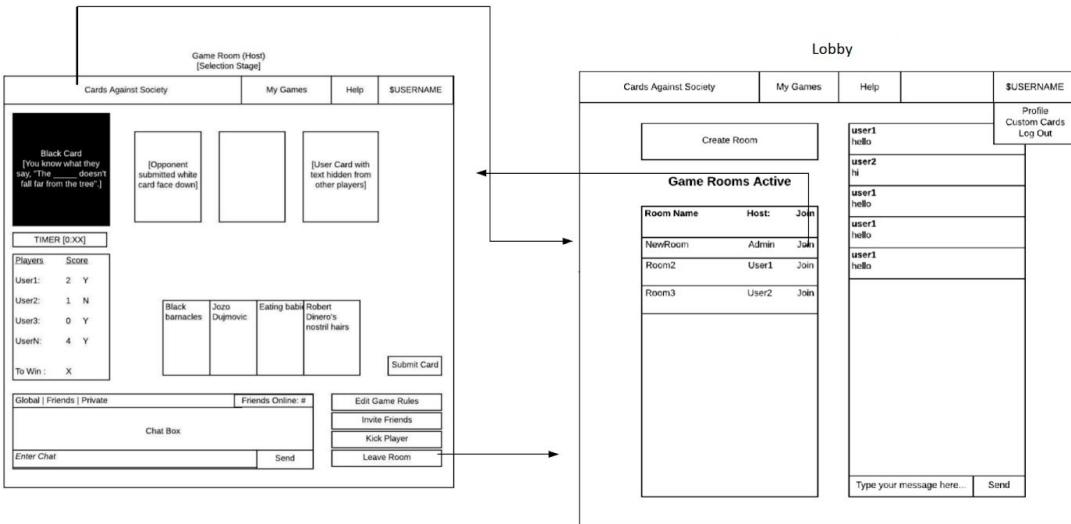
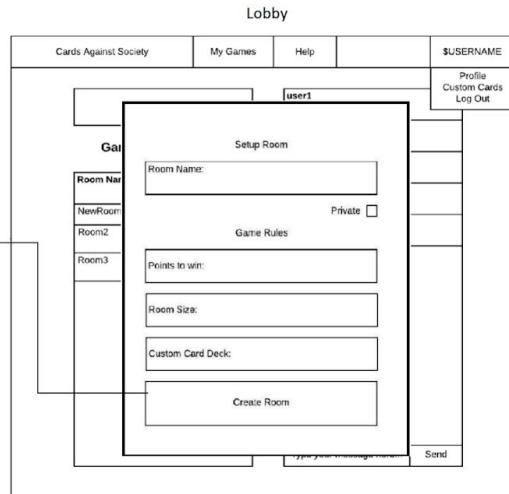
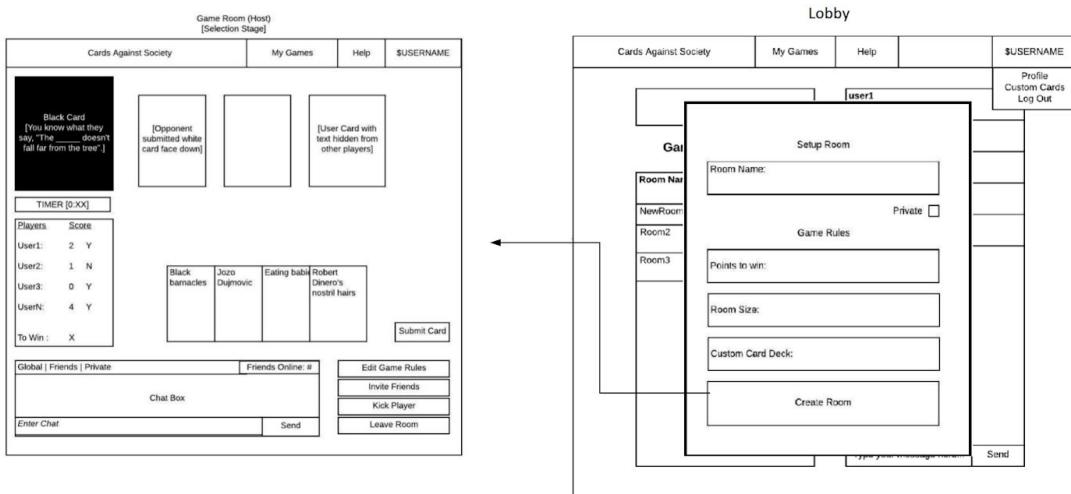


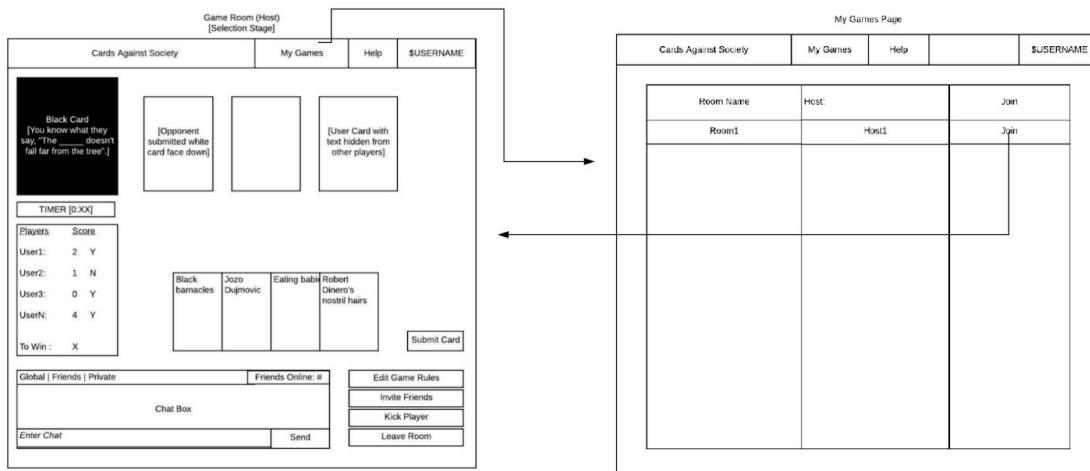
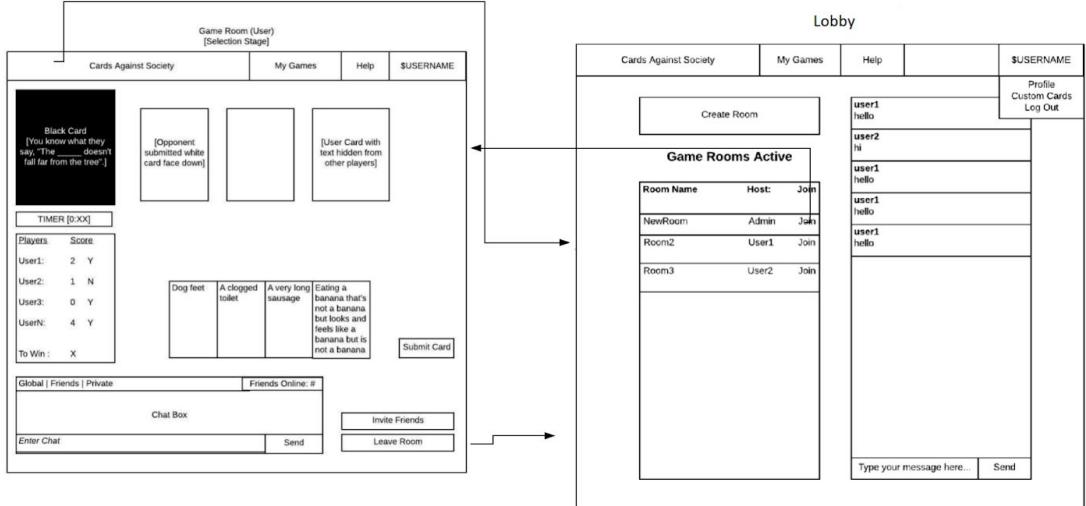


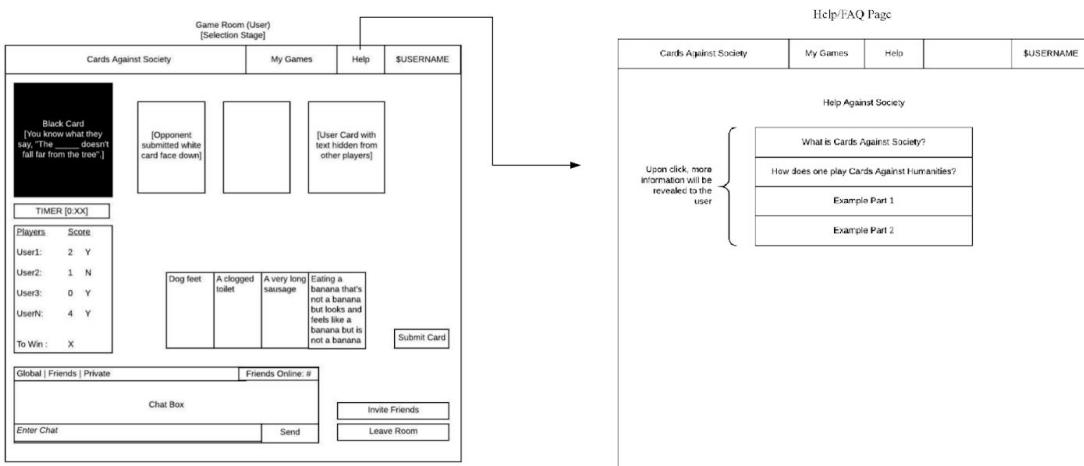
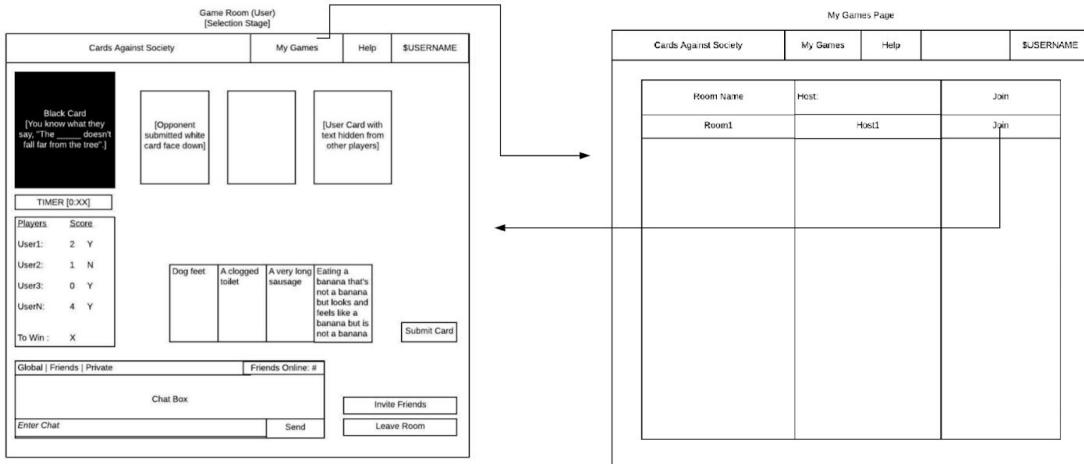
9

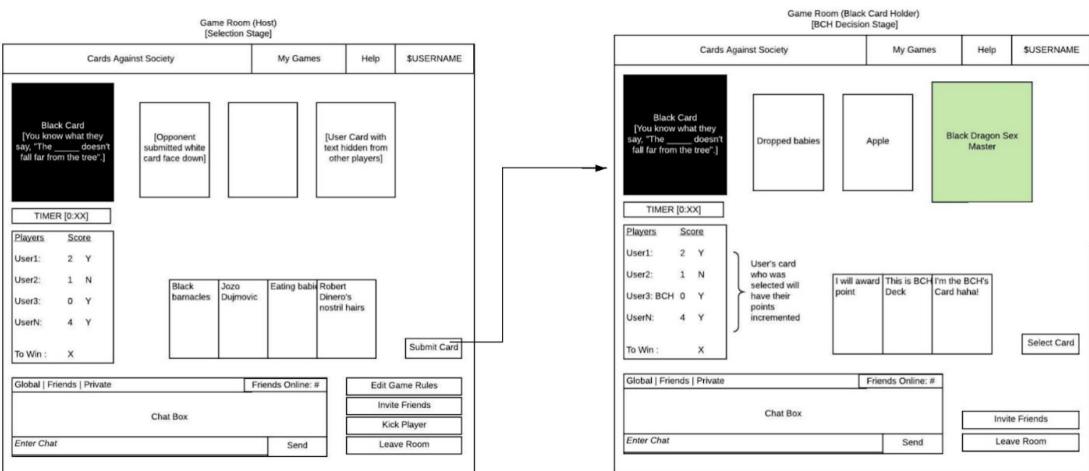
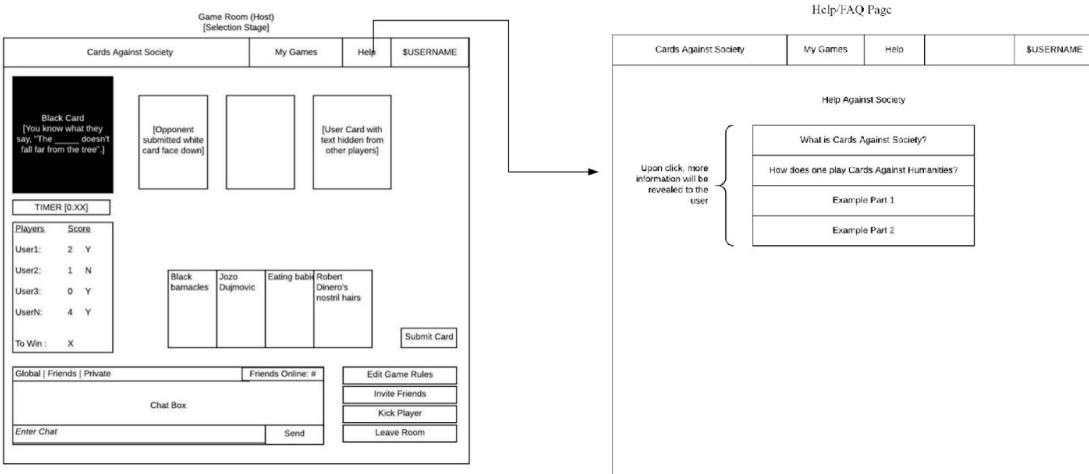




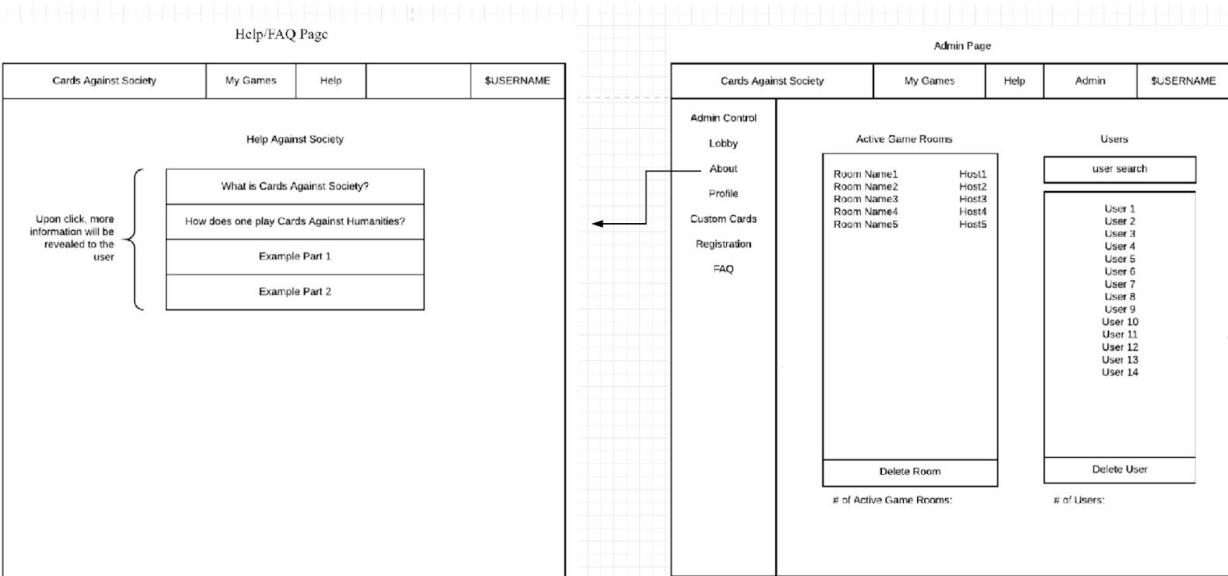
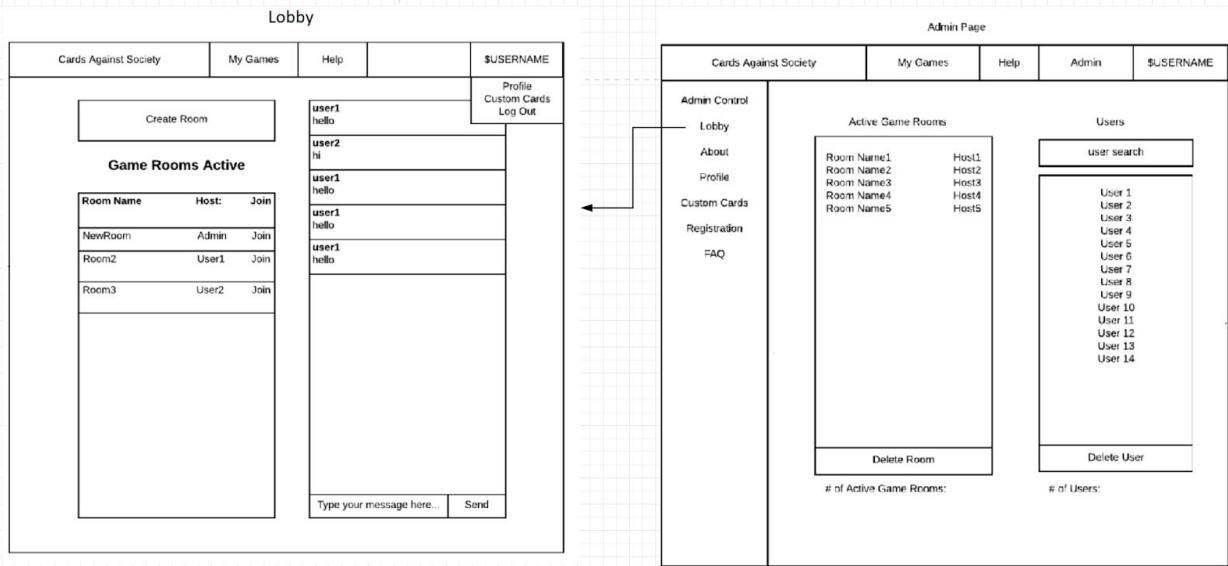


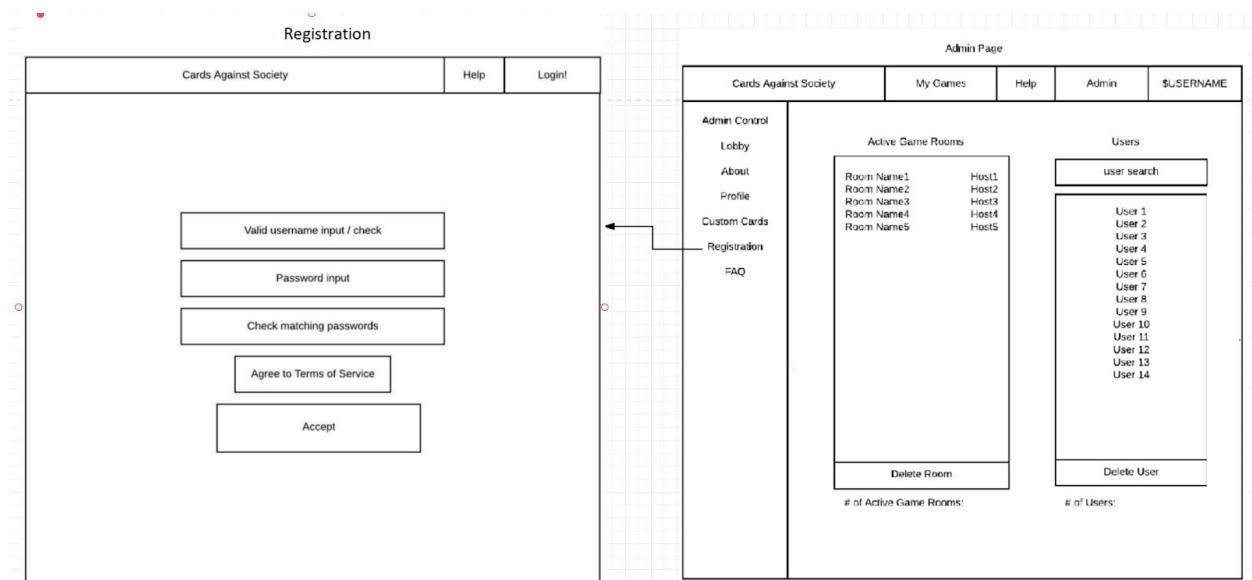
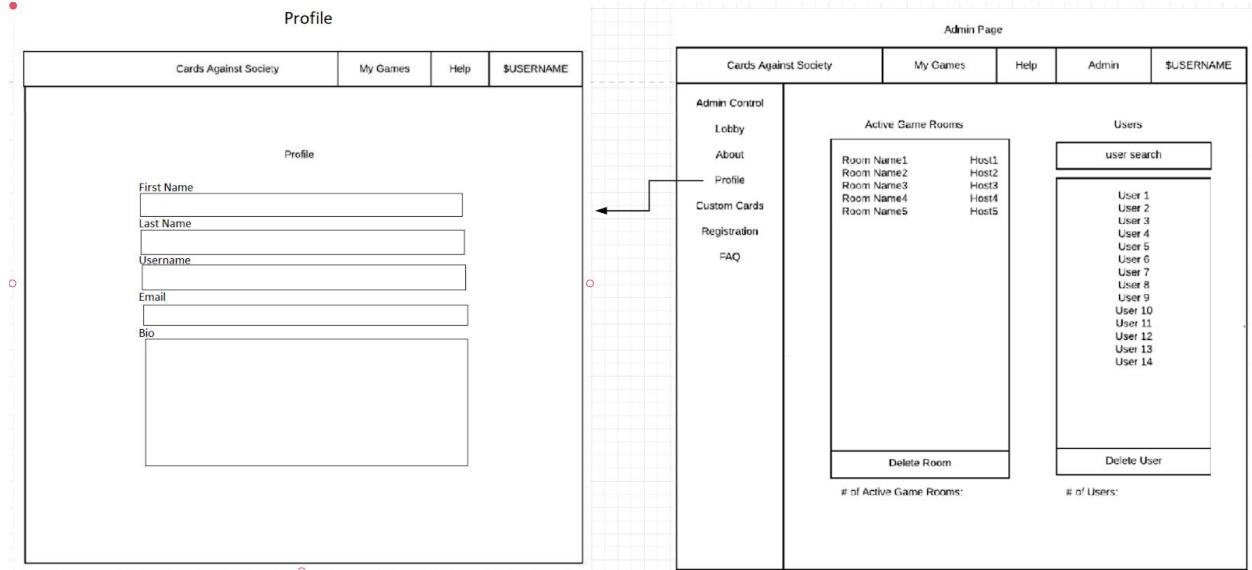


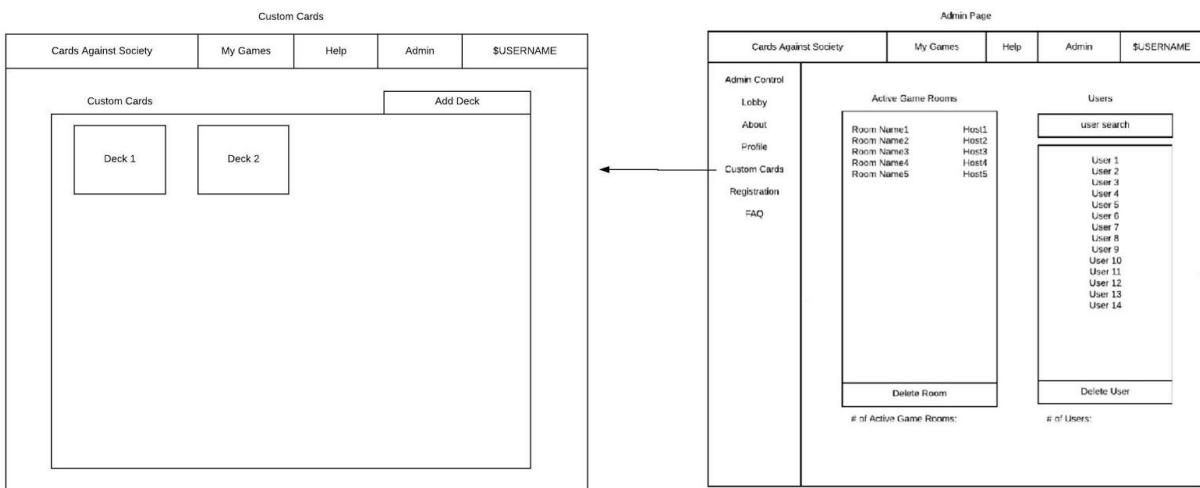
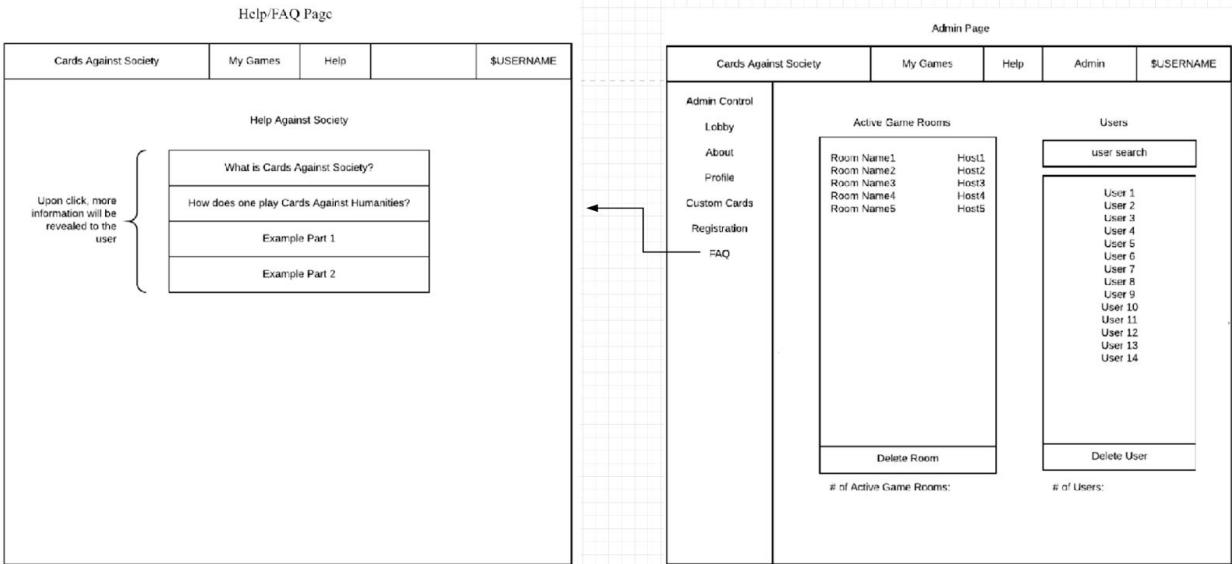


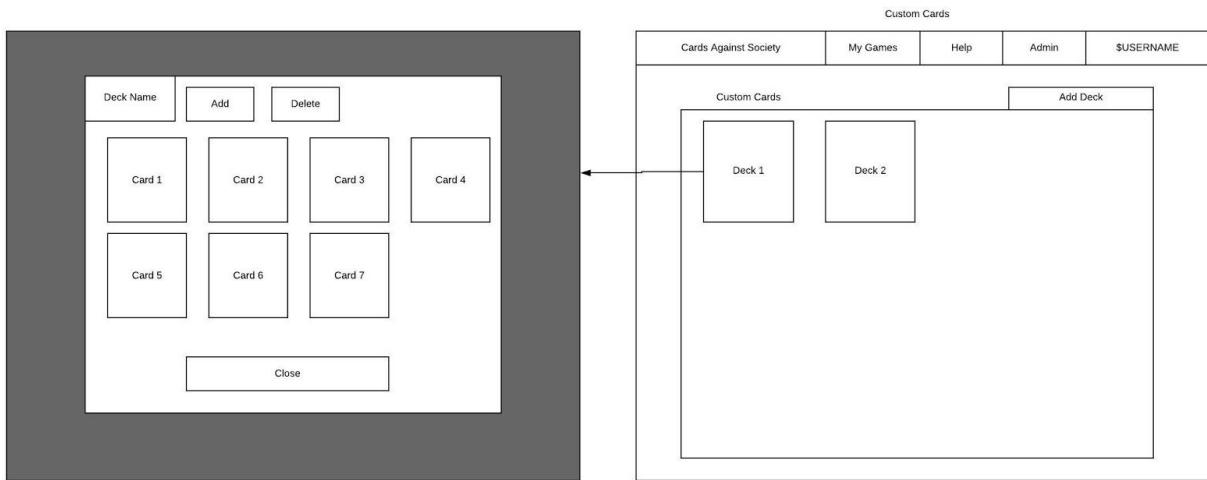
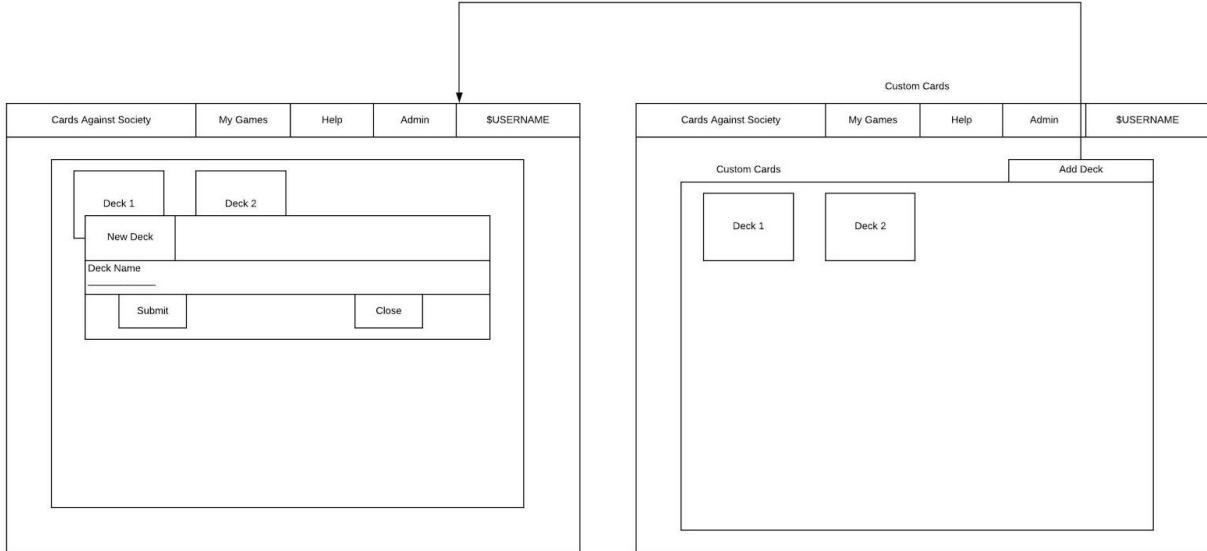


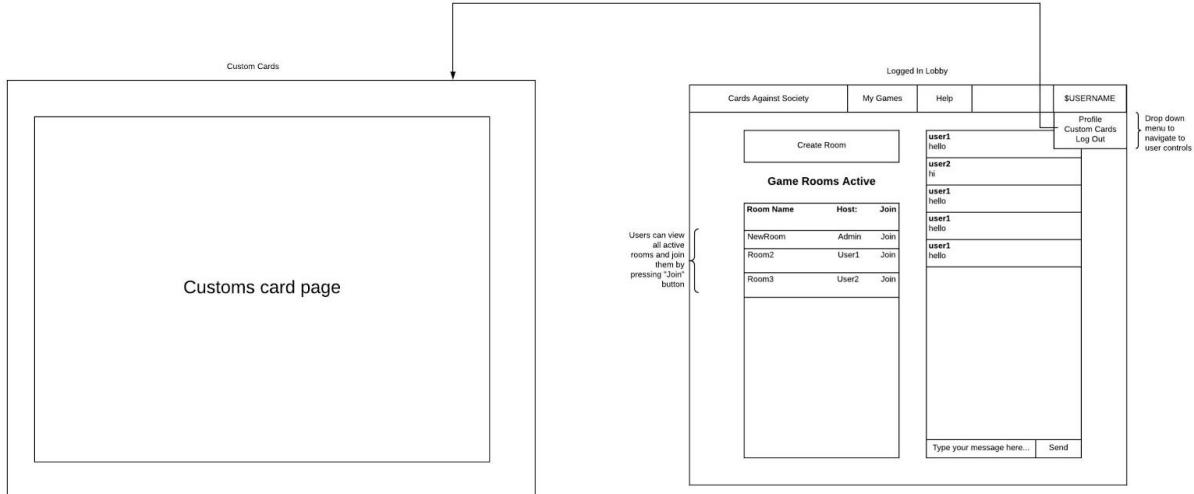
## Logged in as Admin











## **4. High Level Database Architecture and Organization**

### **Business Rules**

1. A User can have zero or many PlayerHands
2. A User can exist in zero or many GameSessions
3. A User can join zero or many PlayerTables
4. A PlayerTable belongs to one GameSession
5. A GameSession belongs to many PlayerTables
6. A GameSession contains one BlackCards
7. BlackCards are in many GameSessions
8. PlayerHands is one Hands
9. Hands contain one or many PlayerHands
10. Hands consist of one or many GamesHands
11. Many GamesHands belong to one GameSession
12. Hands belong to one or many HandsTables
13. HandsTable contains one WhiteCards
14. WhiteCards are in many HandsTables

### **Entities**

#### User

- Relationships :
  - has (PlayerHands)
  - joins (PlayerTable)
  - exists in (GameSession)
- Attributes :
  - User ID (PK, varchar, NN)
  - email (varchar, NN)
  - password (varchar, NN)

#### PlayerHands

- Relationships :
  - has (User)
  - Is A (Hands)
- Attributes :
  - Hand ID (PK, tinyint, NN)
  - User ID (FK, varchar, NN)

## PlayerTable

- Relationships :
  - joins (User)
  - belongs to (GameSession)
- Attributes :
  - GameSession ID (tinyint, NN)
  - Date (datetime, NN)
  - User ID (FK, varchar, NN)

## GameSession

- Relationships :
  - exists in (User)
  - belongs to (PlayerTable)
  - belongs to (GameHand)
  - contains (BlackCards)
- Attributes :
  - Game ID (PK, int, NN)
  - Room Name (varchar, NN)
  - Date (datetime, NN)
  - HostUser ID (varchar, NN)
  - BCHUser ID (varchar, NN)
  - CurrentBlackCard (int, NN)

## Hands

- Relationships :
  - Is A (PlayerHand)
  - consists (GameHand)
  - belongs to (HandsTable)
- Attributes :
  - Hand ID (PK, int, NN)
  - Date (datetime, NN)

## HandsTable

- Relationships :
  - belongs to (Hands)
  - contains (WhiteCards)
- Attributes :
  - Hand ID (FK, int, NN)
  - Date (datetime, NN)
  - White Card ID (FK, int, NN)

## GameHand

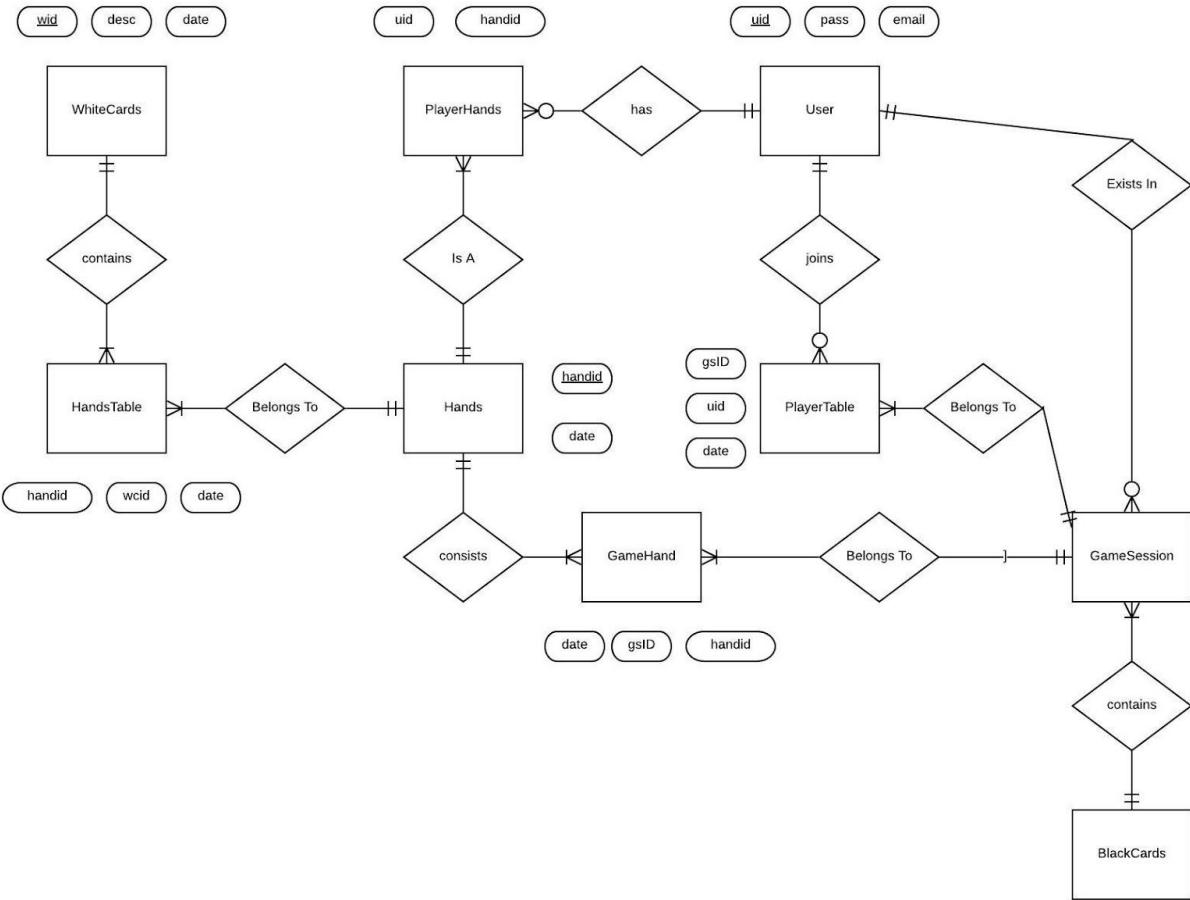
- Relationships :
  - consists (Hands)
  - belongs to (GameSession)
- Attributes :
  - GameSession ID (FK, tinyint, NN)
  - Date (datetime, NN)
  - Hand ID (FK, int, NN)

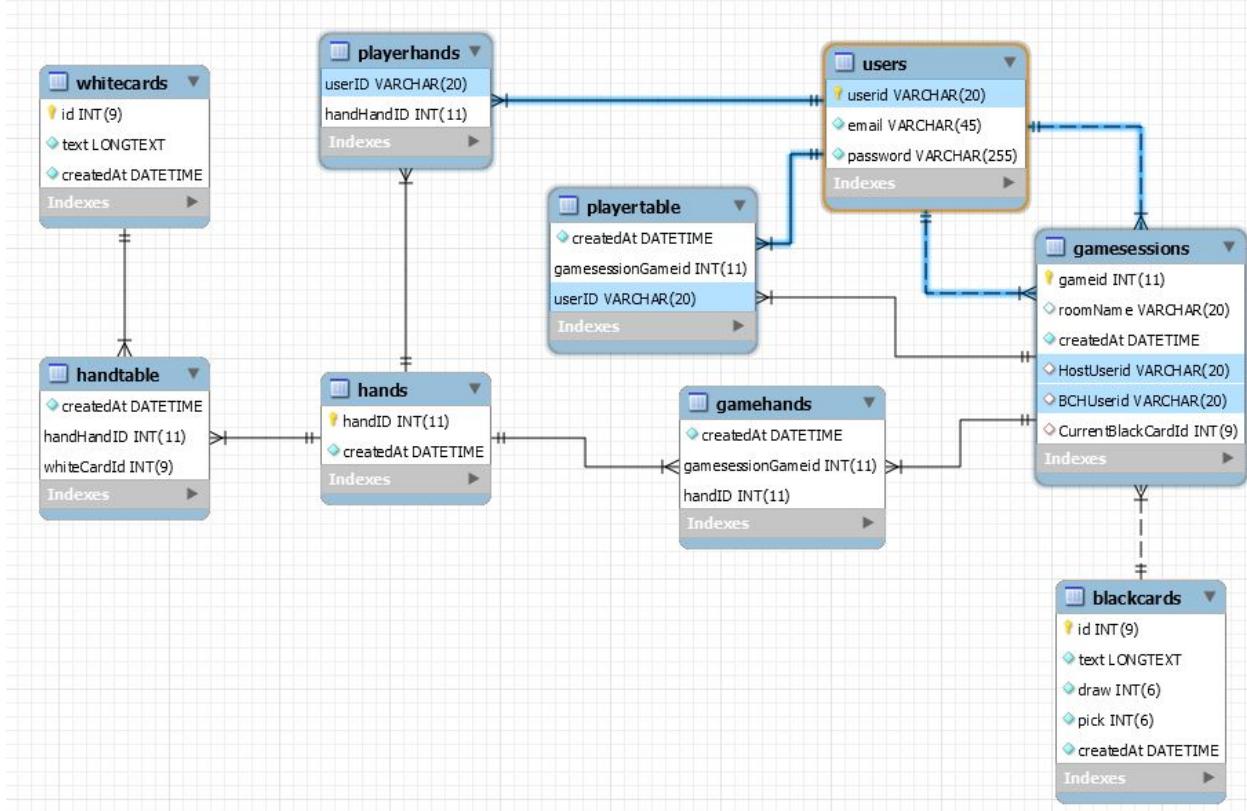
## WhiteCards

- Relationships :
  - contains (HandsTable)
- Attributes :
  - WhiteCard ID (PK, tinyint, NN)
  - Date (datetime, NN)
  - Description (longtext, NN)

## BlackCards

- Relationships :
  - contains (GameSession)
- Attributes :
  - BlackCard ID (PK, tinyint, NN)
  - Date (datetime, NN)
  - Description (longtext, NN)





We will be using MySQL as our database as it will be very convenient and easy to use for our chosen software stack

Search algorithms apply to login and game room listings (all/for specific user). For the login, we use a sequelize find query where the user is exactly the user trying to log in. If the user does not exist, an error is thrown back. The password is then decrypted and checked against the input password. If both fields match, the user is granted access to the account. For all game room listings, we find all game rooms that have open spots for players, and return the set of those rooms to the client. For a specific user's games, we find all games where the user is either a player or a host, and return that set to the client.

## **5. High Level APIs and Algorithms**

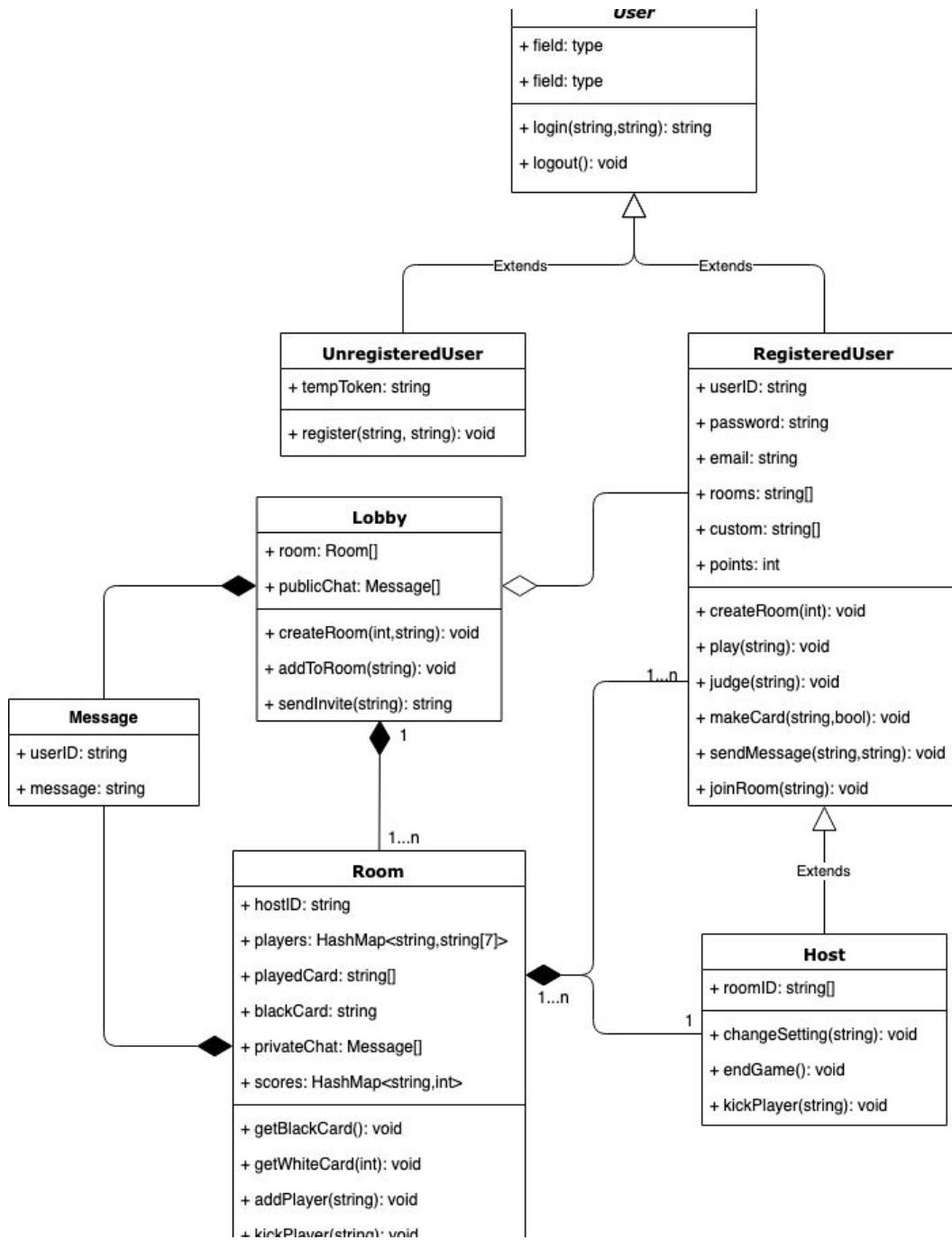
### **Algorithms**

- Randomizing card algorithm - a random card is pulled from its respective database
- Game Room ID's will be automatically incremented when a room is created.
- Card ID's will be automatically incremented when inserted into the database

### **New Frameworks**

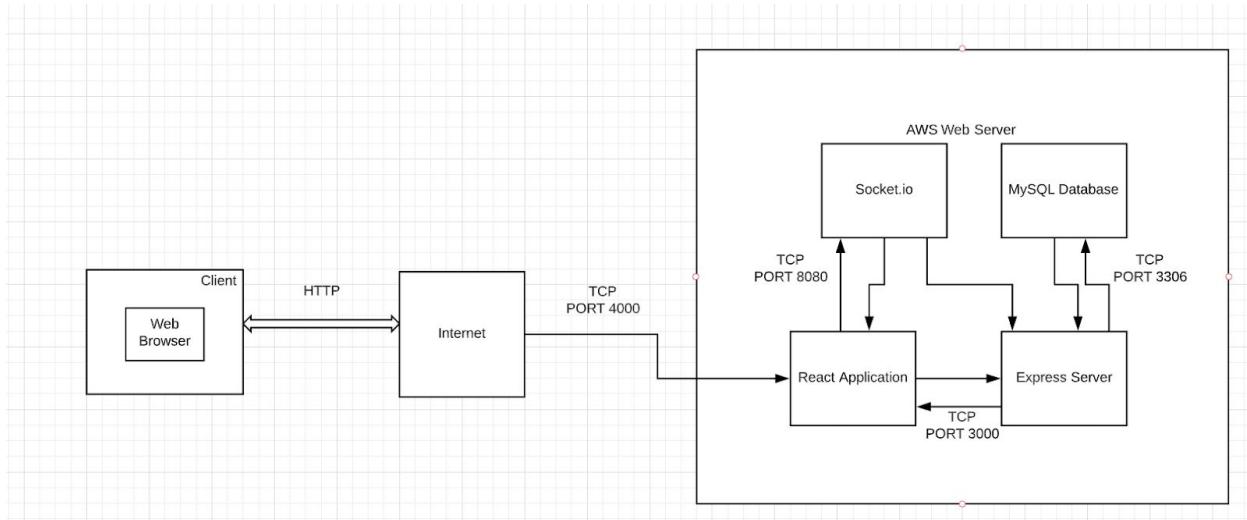
- We are now using Sequelize to manage our database.

## 6. High Level UML Diagram

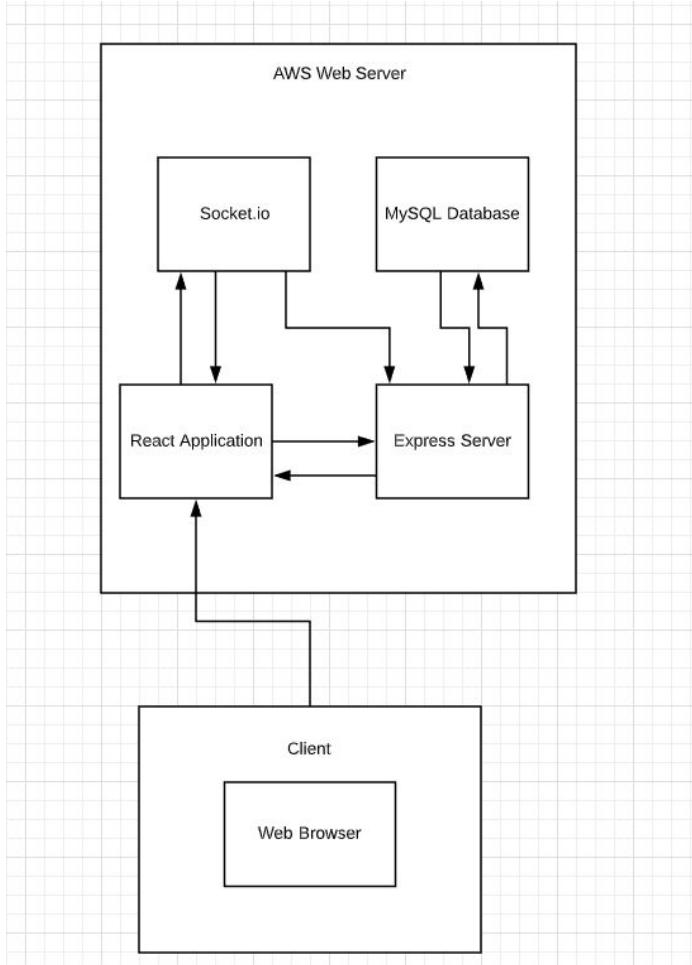


## 7. High Level Application Network and Deployment Diagrams

### Network Diagram



## Deployment Diagram



## **8. Identify *actual* Key Risks for Your Project at this time**

1. Most of our team has not worked with back end development or databases. While this may slow down back end development, this should not hurt the development process. Our application will be very front end heavy, so we will not need everyone to work on the back end to implement our requirements.
2. Our meeting times have been consistent and group communication has been very active. Given what we know and what we have so far, we don't see any big setbacks.
3. The technical risk we see is managing game states and game info. For example, which player's turn it is or which cards every player has. We have ideas to solve this problem, but we are looking for the most efficient way possible, as we plan to update this in real time for the user.
4. Our team has been working very closely together, and there are no teamwork risks.
5. We see no legal/content risks.

## **9. Project Management**

We managed tasks by picking one task each and working on those individually while receiving feedback from the rest of the group. For other parts, we worked collaboratively. In the future, we will need to assign tasks a bit more clearly as it might help our development process. However, everyone in our team is very involved, so task management will be very easy to track. We are using Trello for task management: <https://trello.com/b/2B8ZJ5C5/csc648>

# Cards Against Society

## Milestone 3

SW Engineering CSC648/848 Fall 2019

Section 4

Team 203

21 November 2019

Team Lead: Jose Castanon – jcastan6@mail.sfsu.edu

Back-End Lead: Leslie Zhou – zzhou2@mail.sfsu.edu

Database Master: Shota Ebikawa – sebikawa@mail.sfsu.edu

Git Master: Daryl Ortiz – dortiz4@mail.sfsu.edu

Front-End Lead: Brian Le – ble2@mail.sfsu.edu

<u>Revision #</u>	<u>Version #</u>	<u>Date</u>
Milestone 3	Version 1	11/21/19
Milestone 2	Version 2	11/7/19
Milestone 2	Version 1	10/24/19
Milestone 1	Version 2	10/9/19
Milestone 1	Version 1	10/3/19

## **Table of contents**

1. Instructor Feedback	58
------------------------	----

## **1. Instructor Feedback**

### Home Page

- For the home page, the chatbox should have a private and public section where the private chat shall only be displayed in the lobby.
- The navbar drop down tab needs to be fixed. The text is being cut off on the right.
- Expand the chat box in the lobby to fill in white space.
- In the list room, the buttons need to be fixed so they don't stretch out.
- Fill up the wasted space.

### Login

- Login page is too simple. Needs to have simplicity.
- More info is needed for the user on the login page.
- Add decorations that tells the user what the website is about.
- Register account needs to be bigger.

### Registration

- The terms of service needs to be bigger

### Game Room

- Space needs to be organized

### Custom Cards

- The text is too small and needs to be enlarged for readability.
- Spacing needs to be addressed between cards.



Hi all,

I finished the offline check for M3. There are some minor problems that need to be solved in your repositories, but in general everything seems in order. I was going to send individual reports to the each team lead, but I decided to write a general email, instead, because most of those problems are generalized across all the teams.

**Some common problems in your repositories that need your attention (if applicable)**

1. Some teams haven't updated the main README file yet. That was a M0 task. The main README file should contain your names, roles, name of your project, a small description of the services and functionalities provided by your project (1 or 2 sentences max), and the url to your site. Fix this ASAP before M4 (if needed)
2. I took a quick look into your V2 documents of each milestones, and I saw that a few teams are not addressing the problems stated in V1 of those milestones. I must remind you that V2 of each milestone is the one that will be graded,
3. Contributions to the repo are not balanced. I see students with a lot of commits done, and I see other students from the same team with little commits. Again, number, content, and quality of work in commits is also graded. In addition, quality of the commits comments are graded as well. In the software industry, an excellent comment in a commit can save a lot of headaches and money when a bug needs to be fixed.

**General comments about your horizontal prototype:**

1. I told you this many times in class: simplicity is not equal to simple design. Simplicity in SW engineering is related to the concepts of a product being easy to use, and usable. Take that into consideration when you are doing your UI refactoring for M4 and M5. This is not a SW development class where you can do a simple project and you don't look back to it anymore. This is a SW engineering class and the quality of your project is part of your grade. So, keep that in mind. Also, your home page should state clearly what services your product is going to offer in a few words.
2. Yesterday, many teams did not implement all the storyboards. In addition, there were teams which horizontal prototype was a totally different version than the storyboards they brought to me. This is not acceptable in this class. This prototype was not graded, but make sure that your final UI is similar to your storyboards. Storyboards and mockups are there for a reason, and they represent the vision from the client about their needs in the product. If you change this, then you are putting your needs before the needs of the client.
3. I got many reports from team leads that students are committing to the repository or sending work at the last minute, and in some specific cases, just not sending any work at all. This is not acceptable in this class. Every time you commit or submit work to your team leads, they need time to review the work to make sure that everything is in order, and the quality of the work the one expected.

Please if some of the above apply to your team fix them ASAP.

# Cards Against Society

## Milestone 4

SW Engineering CSC648/848 Fall 2019

Section 4

Team 203

05 December 2019

Team Lead: Jose Castanon – jcastan6@mail.sfsu.edu

Back-End Lead: Leslie Zhou – zzhou2@mail.sfsu.edu

Database Master: Shota Ebikawa – sebikawa@mail.sfsu.edu

Git Master: Daryl Ortiz – dortiz4@mail.sfsu.edu

Front-End Lead: Brian Le – ble2@mail.sfsu.edu

<u>Revision #</u>	<u>Version #</u>	<u>Date</u>
Milestone 4	Version 2	12/10/19
Milestone 4	Version 1	12/05/19
Milestone 3	Version 1	11/21/19
Milestone 2	Version 2	11/07/19
Milestone 2	Version 1	10/24/19
Milestone 1	Version 2	10/09/19
Milestone 1	Version 1	10/03/19

## **Table of contents**

1. Product Summary	62
2. Usability Test Plan	64
3. QA Test Plan	66
4. Code Review	68
5. Self-Check: Best Practices for Security	71
6. Self-Check: Adherence to Original Non-Functional Specs	71

## 1. Product Summary

Have you ever come home from a long day out, taken off your Heelys and your bejeweled, boot cut jeans, turned on some Big Time Rush, and sat in your lonely room in nothing but your socks staring at your cardboard cutout of Dolly Parton, contemplating whether to go back into the cold, miserable world in order to meet up with your fake friends for card game night? If so, then Cards Against Society is the game for you!

Cards Against Society is a highly distasteful, strategic game of ad libs, where you pick a card based on its offensiveness to the ad lib. Players will take turns drawing an ad lib card and select the most messed-up/humorous statement to complete that ad lib.

Cards Against Society combines the millennial trend of table-top games with the convenience that technology offers in the modern age. Now you can enjoy the company of your friends from the comfort of your own home. No more feeling guilty for licking your Cheeto crusted finger tips when hanging out with your friends, or feeling that you're missing out on quality time and memories while sitting at home staring at your computer screen in the dark!

Compared to other online Cards Against Humanity clones, Cards Against Society allows you to play multiple matches concurrently to play with two different friend groups. Also, if you wish to create your own drama and make your friends question your friendship, you have the ability to create custom cards.

Functions :

- Unregistered Users
  - Shall be able to create a new account
  - Shall be able to view Terms of Service
- Registered Users
  - Shall be able to log into their account
  - Shall be able to log out of their account
  - Shall be able to view all active games
  - Shall be able to join games
  - Shall be able to create new games
  - Shall be able to come back to a game at any time
  - Shall be able to create custom cards
- Hosts
  - Shall be able to set game rules on room creation
  - Shall be able to start a new round
- Lobby
  - Shall display all open games and its Room Name and the number of players in the room
  - Shall display public chat room
- “My Games” Page
  - Shall display all current active games for the logged in user

- Game Rooms
  - Shall display user's cards
  - Shall display a private room chat
  - Shall display timer for player to submit a play
  - Shall show relevant game information to the player
- Admin
  - Shall be able to modify or remove data entries
  - Shall be able to stop on-going games
  - Shall be able to view all pages
  - Shall be able to delete users
  - Shall be able to see a list of all registered users
  - Shall be able to see a list of all on-going games
  - Shall be able to have access to admin navigation bar

URL = <http://54.183.228.36:4000/>

## **2. Usability Test Plan**

User satisfaction is the goal for the application and the game room is where it is expected to deliver that. It is important that the user is satisfied with all components of the game room and is having fun. The Game Room is the most important page in the whole application, reason being is the whole purpose of Cards Against Society is to be enjoyable for the user to satisfy user needs. This section is critical and has the most impact on the whole site as it contains usability from other components such as the chat feature and the game itself. To measure user satisfaction is to look into usability metrics. The usability metric will measure effectiveness, efficiency, and user satisfaction.

The first usability metric is Effectiveness where we want to see if users believe the game room system is useful, easy to learn, and that it does what the user wants. The goal for the user is to play the game, use the components fluidly, completely and ultimately find enjoyment. Using the chat feature and other components while playing the game will contribute towards user goals and user satisfaction. We want to see how complete the game room is and if the user interface is visually appealing. From here we can test other usability metrics, efficiency and satisfaction.

To test for efficiency, we can look into efficiency in time, effort and design. How long it takes for the components of the game to proceed will impact how long the user has to wait. The chatbox should instantly receive and send messages. The UI of the application should be easy to use and visually appealing.

Finally we can test the most important goal on how satisfied the user is. For all the components of the game room, we want to measure how comfortable the user is engaging with the different functions in the room. The usability test plan wants to find out if all the components are simple, easy to use and are understandable for the user. The results of the effectiveness, efficiency and user satisfaction will allow for usability to be measured.

To access the application, the users can use various operating systems and browsers. The application can be run on Google Chrome version 77.0.3865.90 on Windows 10 version 10.0.15063 Home Build 15063 or on macOS 10.12 Sierra and above.

To get to the Game Room, the user must first create an account. They will then log in to the lobby using the information they used to register. From the lobby they can either join a game or create a room themselves.

Cards Against Society is for ages 17 and above. With cards that touch on sexual conduct, racial issues, humanitarian crises, inappropriate statements surrounding society and a database full of other taboo topics, this game is best played by people young adults and above.

The URL of the system to be tested is <http://54.183.228.36:4000/>. What will be tested are the components of the game page. User goals in which user goals are being met. How long will the user wait between input and output of the chatbox. How appealing is the UI and if it was easy to understand, use and if the user had fun using the application.

Users will be tasked to test out different functions of the game room such as the chat box and gameplay. Users will join a game room and play the game in order to enter the game room. For the chatroom, the user will type and send messages in the chat box. They will also switch between tabs to send messages privately or publicly. For the gameplay, the user will play the game by reading the black card prompt and submitting a whitecard(s). Users will be asked about the chat box, how easy it was to navigate and if the user had fun.

## Questionnaire

### Cards Against Society - User Satisfaction

I was able to use the chat feature with no issues.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Select	<input type="radio"/>				

The game was easy to play and understand.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Select	<input type="radio"/>				

I felt the game was satisfying to play.

	Strongly Disagree	Disagree	Neutral	Agree	Strongly Agree
Select	<input type="radio"/>				

### **3. QA Test Plan**

There are many different kinds of bugs and they can occur for a number of reasons. Some bugs are easy to diagnose and fix while others are hard to detect. The Game Room will be tested for Quality Assurance as it is the most important function in the application. Most if not all of the potential bugs will come from the Game Room. It is the area where the most components and implementation reside.

There will be three tests for Software Quality Assurance, OS compatibility, pulling information from the database and game functionality. The tests will have specific input and a specific result that will dictate whether or not the test will pass or fail.

To test whether the application will work on different operating systems, the user will be using specific versions of Windows and Mac. For the Hardware, the user will be using a Macbook running macOS 10.15 and another laptop using Windows 10 10.0.15063. This will test whether the application will be consistent and usable on different platforms.

Another test is to test whether or not the database, MySQL was used to store data. To see if the data is successfully being retrieved from the database will signify whether or not the database is working correctly. For the test, the user will enter the game room to conduct observations on whether data from the MySQL server is being displayed on the game cards. Upon success, the cards will display data that was retrieved.

The last test will test functionality with gameplay. The user will test whether or not there are complications within the game room. Some examples of this are components not loading, some aspects of the pages are freezing, or functionality is not working as expected. When the user enters the game room, the user should be able to select a card and submit it.

Being the page with the most work done on, there is a more likely chance to discover a bug. Bugs contribute to user satisfaction on whether they enjoy the feature of an application or not. If there are any bugs that can be found anywhere, then it is best to notified of their existence so they can be addressed as soon as possible. The Game Room contains the chatbox, player list, the navbar and the game itself, all of which can contain bugs. If a bug is found in the game room, then it is likely to be a component found on another part of the application. Finding and correcting bugs will open an opportunity to correct not only the game page itself, but other pages as well.

## Quality Assurance Test Plan

Test #	Test Description	Test Input	Expected	Results
1	User tests out operating system compatibility.	The user uses operating system Windows 10 10.0.15063 and MacOS 10.15 to run the application in the Game Room	The application is the same on different operating systems.	Pass
2	Checking if all cards are being displayed to the user	User entering game room	The user is able to view both white and black cards in the game	Pass
3	Testing whether the user can select and submit a card	The user chooses a card and clicks on the submit button	Card is consumed and submitted	Pass
4	Testing if new cards are displayed after submission.	The user enters the game and submits a card.	A new set of cards will be displayed.	Fail
5	Checking if the user will be redirected to lobby page if not logged in.	The user enters anywhere in the site while not logged in.	The user will be redirected to the lobby.	Fail

## 4. Code Review

### a) Coding Style

- Encapsulate inline-styles in a single variable
- Start a new line after the opening curly brackets when initializing objects and functions
- No whitespace between JSX/HTML tag's angle brackets and its element name
- Use arrow function for the on-event handlers
- Have each objects in a separate line when importing more than 4 objects from a given module

Naming principles:

- Pascal case for Classes (React Component) and JSX file
- Camel case for variables and functions
- Lowercase for json file and image file

### b) Code Reviews

Jose's Code

```
/*
 * Daryl Ortiz
 * - Commits are perfectly placed and simply and clearly explains the implementation.
 * - Everything is structured well. much thought was put into the code.
 * - The structure of the game container follows exactly like the mockups.
 * - Overall everything about the page contains no issues to me and is very well done.

 Leslie Zhou Team 3
 Comment:
 Great code in general
 |the styling of declearing function is different from other files

 Brian Le
 - Holy long batman!
 - Great variable + function naming
 - imports could be categorized / grouped
 - Good comments on functions
 - Easy to follow indentation

 Shota Ebikawa Team 3
 Comment:
 The imports are indented in organized fashion, especially the one with the
 react-bootstrap. I also like how you added few snippets of comments to the
 functions that deals with a complicated logic, such as submitSelection.
 One thing I would do is to add a bit more comment to the gameLayout
 function to help team members understand its logic more easily.
 */

import React, { Component } from "react";
import io from "socket.io-client";
import {
  Button,
  ButtonGroup,
  Row,
  Col,
  Container,
```

## Daryl's code:

```

/*
Shota Ebikawa Team 3
Comment:
The JSX tags are indented in a consistent manner. Overall, I had an easy
time understanding each blocks of codes. One thing I would do is to delete
unused components from react-bootstrap, such as the Button component. Other
than that, there are not much to say from my end.

Jose Castanon
Indentation is not correctly formatted. Page could be better organized. Consider using react Row and Col to better
structure the page.

Leslie Zhou Team 3
Comment:
Great page that does the job

Brian Le
- Personally I would reorder the imports to categorize
- Overall, everything is nicely segmented and categorized for easy reading
*/

```

```

import React, { Component } from 'react';
import Header from './Components/Header';
import { retrieveCookie } from './Components/Cookies';
import './app.css';
import ListRooms from './Components/ListRooms';
import UserSearch from './Components/UserSearch';
import { Container, Row, Col, Button } from 'react-bootstrap';

export default class Admin extends Component {
  state = {
    userid: retrieveCookie('userid')
  };

  render() {
    return (
      <body>

```

## Leslie's code:

```

/*
Brian Le
- Nicely categorized, but the 'react-bootstrap' imports could be one lined rather than 3
- Unused mount 'componentDidMount()'
- All functions in same area :ok_hand:
- HTML is nicely indented (:)

Daryl Ortiz
- Everything is very well structured
- Although it is easy to understand by looking at the code itself, it would also help to have comments.
- Examples of the Custom Decks are all helpful to new users trying to figure out what kinds of custom cards to
make.

Jose Castanon
- Code is very well laid out
- Using a loop to find which deck was clicked is a bit inefficient, although there's not much happening in the
page so it's probably not going affect performance.

Shota Ebikawa
The code is formatted consistently, such as the JSX tags
and I was able to understand every blocks of code. One
thing I would do is to add few lines of comment for the
handleClick. By doing so, team members will have an easier
time understanding the given logic.

*/

```

```

import React, { Component } from 'react';
import { Redirect } from 'react-router-dom';

```

## Shota's Code:

```

Leslie Zhou Team 3
Comment:
marginStyle is empty
not resetting the form after submit
 */

/*
Daryl Ortiz
- Although everything is understand, comments would make everything more clear.
- Everything is indented nicely.
- My only suggestion is to use containers so that the game room doesn't overlap between other modules.
 */

/*
Brian Le
- First glance, everything nice and uniform
- Imports can be categorized by 'from's and 'css'
- Nice and neat coding, easy to read
- Is there anyway to not start with a fragmentation?

Jose Castanon
- I would suggest using a class to better manage states.
- Good formatting, very organized
- Maybe use a css file for styling
 */

import React from "react";
import "../About.css";
import {
  Button,
  FormGroup,
  FormControl,
  FormLabel,
  FormCheck,
  Dropdown,
  Modal
} from "react-bootstrap";

```

## External Review from Team 5

 **Amir A** 4:25 PM  
jose.js ▾

```

1 const express = require('express');
2 const Sequelize = require('sequelize');
3 const bodyParser = require('body-parser');
4
5 const router = express.Router();
6 const models = require('../models');
7
8 const app = express();
9 const op = Sequelize.Op;
10
11 /**
12 * --Amir Team 5
13 * Overall Comment:
14 * I like the how the routes are organized by their usage and easy to find
15 * I would love to see some more comments inside the code to understand better what every route is for
16 * And lastly I see most of the routes do a lot of work as they being called. Wouldn't be better to split the work in different
functions and just export
17 * the functions ? It would help a lot with the organization
18 *
19 */
20 ..

```

## 5. Self-Check: Best Practices for Security

Major assets and validations:

- Usernames
  - Input field requires usernames to be less than forty characters
- User emails
  - Emails are required to have an ‘at’ symbol (@)
- User passwords
  - Passwords are required to be of eight characters minimum
  - Passwords must be repeated for the user to acknowledge and remember their password
  - Once the form is submitted, the passwords are encrypted

## 6. Self-Check: Adherence to Original Non-Functional Specs

- Compatibility
  - System shall be compatible with at least two of the major browsers of Mozilla Firefox, Google Chrome, Safari, and Edge. **-DONE**
  - System shall be compatible with version 67, 68, and 69 of Mozilla Firefox. **-DONE**
  - System shall be compatible with version 75, 76, and 77 of Google Chrome **-DONE**
  - System shall be compatible with version 10, 11, and 12 of Safari. **-DONE**
  - System shall be compatible with version 41, 42, and 44 of Edge. **-DONE**
- System
  - Arrow functions shall be implemented for the on-event handlers. **-DONE**
  - JSX/HTML tags shall have no whitespace between its element name and its angle brackets. **-DONE**
  - New lines shall be started after the opening curly brackets when objects and functions are being initialized. **-DONE**
  - Each objects shall be in a separate line when more than 4 objects from a given module are imported **-DONE**
  - Inline-styles shall be encapsulated in a single variable. **-DONE**
  - Application shall be developed using the software stack declared in M0 **-DONE**
  - Application shall be deployed using an AWS EC2 instance **-DONE**
  - UI shall be implemented using React **-DONE**
  - Application shall run off of the Master branch of the team’s git repo **-DONE**
  - Game animations shall be implemented using CSS **-DONE**

- Content
  - Images shall be in the format of either jpg, png, or jpeg. **-DONE**
  - Query results shall be displayed in alphabetical ascending order. **-DONE**
- Storage, security, environmental requirements
  - Data shall be stored in a MySQL database **-DONE**
  - Game states shall be persistent, saved by the system and loaded when a player joins **-DONE**
  - Application shall allow up to 5 users in one game room **-DONE**
  - Application shall keep public and private chat in their respective places **-DONE**
  - Application shall load a default game state when a room is created **-DONE**
  - Database shall store all of the game cards to be used during the game **-DONE**
- Marketing, legal requirements
  - An individual shall receive permission from the copyright holders to use Cards Against Society's SW code. **-DONE**
  - An individual shall not use Cards Against Society's Logo as one's trademark **-DONE**
  - An individual shall not disclose Cards Against Society's API as it is part of the company's trade secrets. **-DONE**
- Privacy
  - Only relevant data shall be collected, such as user's win rate **-DONE**
  - Password collected shall be encrypted in the database **-DONE**
- Performance
  - System shall update game data in real time **-DONE**
  - System shall update chat message data in real time using sockets **-DONE**
  - Application shall have a responsive game UI **-DONE**
  - Application shall display game without the use of images **-DONE**

### 3. Screenshots of Final Product as shown in the demo

The image displays three screenshots of the Cards Against Society application:

- Screenshot 1 (Top Left):** Shows the "Create Room" interface. It includes a search bar ("Search for a room") and a table titled "Game Rooms Active" with columns: Room Name, Host, Room Capacity, and Join. Three rooms are listed: "asd" (Host: admin, Capacity: 1/1), "xdfcv" (Host: bussy, Capacity: 2/5), and "Jose's corner!" (Host: briante, Capacity: 1/5). Each row has a "Join" button.
- Screenshot 2 (Top Right):** Shows the "Public Chat" interface. It lists messages from users "briante" and "bussy".
- Screenshot 3 (Bottom Center):** Shows the "Create an Account" interface. It includes fields for Username (admin), Email, Password, and Confirm Password. A note states: "By creating an Account, you have read and agreed to our [Terms of Service](#)". A "Create Account" button is at the bottom.

**Basically, we will not store any unnecessary data.**

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Curabitur felis diam, imperdiet vel viverra et dictum eu ex. Nullam tincidunt accumsan lectus, non porta ante venenatis tristique. Sed ultrices, odio pretium pulvinar tempus, urna risus sagittis risus, ac rhoncus nisi odio vel metus. Sed nec dapibus erat. Mauris consectetur semper loren sit amet placidat. Aenean id consectetur orci. Pellentesque molestie arcu a tellis voluptat, cursus egestas felis pulvinar. Curabitur nec dui erat. Nam eget odio ut mauris eleifend ultrices. Praesent malesuada mollis tristique. Ut gravida congue dui. Sed egestas at neque ac mollis. In erat consequat, commodo justo ut, sollicitudin urna. Ut metus nisi, commodo eget consectetur at, maximus et sapien. Praesent convallis quam vel ex pulvinar, facilisis hendrerit elit ullamcorper. Donec justo est, scelerisque sed hendrerit at, porta et ex. In congue placerat ornare. Curabitur tempus eros et augue eleifend gravida. Duis vestibulum nisi id sollicitudin viverra. Sed eros lorem, sollicitudin imperdiet efficitur non, viverra pulvinar ligula. Class aptent taciti sociosqu ad litora torquent per conubia nostra, per inceptos himenaeos. Aliquam rhoncus cursus massa vitae rhoncus. Fusce maximus, sem sit amet imperdiet placenter, mi augue suscipit nisl, at luctus nisi ligula id urna. Mauris pellentesque eu elit nec trident. Phasellus nec nisi at mi tristique ultricies. Etiam imperdiet, nisi eu dapibus fringilla, elit lectus luctus turpis, et eleifend erat nisl at mi. Fusce tempus rhoncus velit quis consequat.

Sed eleifend sapien ac enim scelerisque volutpat. Morbi a convallis leo. Nullam a consectetur quam. Nunc purus quam, feugiat ut nisi molestie, commodo dignissim mauris. Mauris eget elit sit amet nulla consectetur interdum non ac mi. Morbi tristique turpis eget maximus aliquam. Sed blandit ullamcorper ligula nec imperdiet. Curabitur sed pulvinar felis. In a massa imperdiet, tempus est a efficitur neque. Suspendisse a leo vel mauris iaculis sodales. Pellentesque id tellus nec urna lobortis hendrerit ut efficitur urna. Suspendisse pharetra risus dui, vitae ultricies massa condimentum egestas.

Suspendisse potenti. In hac habitasse platea dictumst. Ut id vulputate risus, in exceptus leo. Vivamus vel concomitio nisi. In finibus quam ligula, sed fermentum ligula aliquip et. Nullam dolor nisi, blandit sed placerat id, pretium non purus. Vivamus vitae augue blandit, accumsan turpis at, molestie turpis. Sed non dictum turpis. Quisque facilisis nisi urna, et patentesque magna consequat sit amet. Vestibulum sem nunc.

Cards Against Society   My Games   Help   Admin

Signed in as: admin

### Custom Decks

Add Deck

New Deck

Deck Name: Jozo's Class Reader

Submit Close

Decks

Harry Potter Erotica   Religions

Cards Against Society My Games Help Admin Signed in as: admin

### Create Room

Game Rooms Active			
Search for a room			
Room Name	Host	Room Capacity	Join
asd	admin	1/1	<button>Join</button>
xdfcv	bussy	3/5	<button>Join</button>
Jose's corner!	brianle	2/5	<button>Join</button>
Test room	sebikawa	1/4	<button>Join</button>

10 ▾ 1

### Public Chat

Type your message here... Send

Cards Against Society My Games Help Admin Signed in as: admin

MTV's new reality show features eight washed-up celebrities living with \_\_\_\_\_.  
\*pick 1 card(s)\*

time left in turn: 0:50

Players	Score
sebikawa	0
admin	0



admin



sebikawa

Submit Reset Selection

Waking up half-naked in a Denny's parking lot.

The Thong Song.

Expecting a burp and vomiting on the floor.

Exactly what you'd expect.

That ass.

Public Private

Invite Friends
Leave Room

Cards Against Society My Games Help Admin Signed in as: admin

### Game Rooms Active

Room Name	Host	Room Capacity	Join
asd	admin	1/1	<a href="#">Join</a>
xdfcv	bussy	3/5	<a href="#">Join</a>
Jose's corner!	brianle	3/5	<a href="#">Join</a>
Test room	sebikawa	2/4	<a href="#">Join</a>

10 ▾ 1

Cards Against Society My Games Help Admin Signed in as: admin

### Help Against Society

▼ What is Cards Against Society?

Cards Against Society is a party game for horrible people based on the real card game, Cards Against Humanity. However, unlike most of the party games you've played before, Cards Against Humanity is as despicable and awkward as you and your friends. The game is simple. Each round, one player asks a question from a Black Card, and everyone else answers with their funniest White Card.

▼ How does one play Cards Against Humanities?

One randomly chosen player begins as the Black Card Holder (BCH) and a randomly chosen Black Card is displayed. The Black Card Holder reads the question or fill-in-the-blank phrase on the Black Card out loud. Every player besides the BCH will answer the question or fills in the blank by submitting one White Card from their hand. The submitted White Cards are shuffled and displayed, and then read by the current BCH. For full effect, the BCH should read the Black Card before presenting each White Card if all players are in a voice chat. The BCH then picks their favorite White Card response, and whoever played that answer gains a point.

➤ As I handed my Dad his 50th birthday card, he looked at me with tears in his eyes and said...

➤ Today, my son asked me "Can I have a book mark?" and I burst into tears.

## 4. Screenshots of Key Database Tables

### User Table

	userid	email	password
▶	admin	admin@administrator.com	\$2b\$10\$0N2mgTJCfbYULcM7LzIIWe9AP62p7hK...
	test2	test2@gmail.com	\$2b\$10\$FWIZlI0SoYE6Vc9f64W96edOuZ6LYaS...
	testing2	testing3@gmail.com	\$2b\$10\$71s7bwKFseInG.tBAYcSo.EbmbKQLW...
*	testingout	testingout@ass	\$2b\$10\$biU.1RGUku8pfSiADB9W..YmwvfTTzkA...
	NULL	NULL	NULL

### Gamesessions Table

	gameid	roomName	pick	playersPicked	playerCount	capacity	gameState	createdAt	HostUserId	BCHUserId	CurrentBlackCardId
▶	38	rgf	1	1	2	4	{"state":[{"userid":"admin","cards":["Friction."]}],"points":[{"testingout...":0,"admin":0}]} {"state":[],"points":[]}	2019-12-04 05:24:06	testingout	testingout	73
	39	asd	1	0	2	4	{"state":[],"points":[]}	2019-12-08 22:42:51	testingout	testingout	43
*	40	asdd	3	0	2	2	{"state":[],"points":[]}	2019-12-08 22:44:19	testingout	testingout	86
	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL	NULL

### Black Cards

	id	text	draw	pick	createdAt
1	1	Why can't I sleep at night?	0	1	2019-12-02 22:09:33
2	2	I got 99 problems but ____ ain't one.	0	1	2019-12-02 22:09:33
3	3	What's a girl's best friend?	0	1	2019-12-02 22:09:33
4	4	What's that smell?	0	1	2019-12-02 22:09:33
5	5	_____? There's an app for that.	0	1	2019-12-02 22:09:33
6	6	This is the way the world ends This is...	0	1	2019-12-02 22:09:33
7	7	What is Batman's guilty pleasure?	0	1	2019-12-02 22:09:33
8	8	TSA guidelines now prohibit _____ on ...	0	1	2019-12-02 22:09:33
9	9	What ended my last relationship?	0	1	2019-12-02 22:09:33
10	10	MTV's new reality show features eight...	0	1	2019-12-02 22:09:33
11	11	I drink to forget _____.	0	1	2019-12-02 22:09:33
12	12	I'm sorry, Professor, but I couldn't co...	0	1	2019-12-02 22:09:33

### White Cards

Result Grid		
	id	text
		createdAt
▶	1	Coat hanger abortions.
	2	Man meat.
	3	Autocannibalism.
	4	Vigorous jazz hands.
	5	Flightless birds.
	6	Pictures of boobs.
	7	Doing the right thing.
	8	Hunting accidents.
	9	A cartoon camel enjoying the smooth, refreshin...
	10	The violation of our most basic human rights.
	11	Viagra◆.
	12	SelfLoathing

## 5. Screenshots of Task Management System

# to-do

**crayhoseay** 12/01/2019  
back end todo

```
*player enters game*
if player is bch
    cards are greyed out
    once players have all selected their card(s), cards are displayed to bch
    bch selects best card, submits
    player gets a point if they selected the card
else
    player selects best card
    player submits best card
    if player submits the wrong amount of cards
        cards are deselected, player is alerted to choose the right amount
```

---

**Brian Le** 12/01/2019

```
login page - too simple?
- move down, add decorations / about what this website is / description
- bigger (REGISTER ACCOUNT)
- title + slogan(?)
```

Registration - BIGGER TERMS OF SERVICE

Lobby - Button size
- Wasting space

Game Room - Organize space

Customize cards - organize space / bigger text

Bigger text overall

basically for front end^

---

**crayhoseay** 12/01/2019

```
***backend to-do (P1) ***
- keep track of bch -- DONE
- keep track of each player's points
- keep track of each player's status (cards selected or not)
- keep track of each player's submitted card(s)
- keep track of bch's best card selection, find the owner of the card and give him a point
```

# to-do

- changing chatbox to have two different chatboxes that can be switched back and forth for private and public chats (private only displayed in lobby)
- getting m4 document started
- fixing navbar dropdown tab
- adding info to login page
- fixing chatbox to drop down in the lobby page
- fixing buttons in listroom components so they aren't stretched out
- login page - too simple?
  - move down, add decorations / about what this website is / description
  - bigger (REGISTER ACCOUNT)
  - title + slogan(?)

Registration - BIGGER TERMS OF SERVICE

Lobby - Button size
 

- Wasting space

Game Room - Organize space

Customize cards - organize space / bigger text

Bigger text overall

*player enters game*

```

if player is bch
  cards are greyed out
  once players have all selected their card(s), cards are displayed to bch
  bch selects best card, submits
  player gets a point if they selected the card
else
  player selects best card
  player submits best card
  if player submits the wrong amount of cards
    cards are deselected, player is alerted to choose the right amount
  
```

(edited)

## 6. Team Member Contributions



Daryl Guevara Ortiz  
Thu 12/19/2019 2:58 PM

Shota Sean Ebikawa; Brian Timothy Le; Jose Miguel Castanon; Zun Zhou ▾

a) Here are the things that I have work on for our "Cards Against Society" project:

- Created (Front-end) Lobby, Profile and Admin pages.
- Styled many aspect of the application. (Color Scheme, Borders, Admin tools etc.)
- Worked on the all Milestone documents.
- Worked with Front-end lead develop front-end UI mock-ups for Milestone 2 -3.
- Worked on UI storyboards for Milestone 2.
- Created Usability and QA test plans for Milestone 3.
- Collaborated with whole team on functional requirements in Milestone 1.
- Created Use Case diagram in Milestone 1.
- Reviewed all code from fellow team members.
- Met with group numerous times throughout the course for weekly check-ins.

b) I made a total of 11 commits to the development branch.



Shota Sean Ebikawa  
Thu 12/19/2019 2:19 PM  
Jose Miguel Castanon; Brian Timothy Le; Zun Zhou; Daryl Guevara Ortiz ▾



a) During the development process of Cards Against Society, I have:

- developed GameForm component and integrated it in the Lobby component
- collaborated with the Frontend lead to help make the ListForm component and MyGame component adhere to the responsive UI.
- worked with all of the team members to come up with non-functional requirements in Milestone 1
- worked with all of the team members to a decision regarding priorities of functional requirements in M2
- collaborated with the Frontend lead in the presentation layer of the app in M3
- worked with all of the team members for the code reviews and in iterating coding standards in M4

b) made a total of 10 commits to the development branch.

Shota Ebikawa



Zun Zhou

Brian Timothy Le; Daryl Guevara Ortiz; + 2 ▾

**Contributions****1) Contributions this semester:**

- Competitive analysis in milestone 1
- Database diagram in milestone 2
- Created custom card and card decks page for milestone 3 / 4
- Edited webpacks, fixed slide show issues

**2) I made a total of 20 commits to the development branch.**

Brian Timothy Le

Jose Miguel Castanon; Daryl Guevara Ortiz; + 2 ▾

5:39 F

**Re: My Contributions - Brian**

(i) You replied to this message on 12/19/2019 5:51 PM.

Contributions over the semester as follows:

- M1
  - o Documentation
    - Executive summary
    - Use cases (+ Jose)
    - Entities list
    - Helped with non-functional requirements
- M2
  - o Documentation
    - Functional requirements v2 priority list (+ Jose)
    - UI mockups (+ Daryl)
    - Business Rules (+ Jose)
    - Entities
    - ERD
- M3
  - o Application
    - Front end and UI design
    - Cleaning and styling
    - Registration, Login, About, FAQ
    - Direction and advising
- M4
  - o Application
    - Continuation of previous
- M5
  - o Documentation

Onto the Dev branch, I have committed 6-12+ (GitHub only shows commits max 1 month back)

JM

Jose Miguel Castanon

Daryl Guevara Ortiz; Shota Sean Ebikawa; + 2 ▾

## My Contributions

a) What I worked on:

- Managed team tasks
- Created Game page
- Implemented game page functionality
- Implemented backend API routes
- Designed app database
- Designed database models
- Implemented game page communication to backend

b) I made a total of 90 commits to the development branch.

## 7. Post Analysis

JM

Jose Miguel Castanon

Brian Timothy Le; Daryl Guevara Ortiz; + 2 ▾

5:51 PM

### Post Analysis

During the course of the semester, we faced various challenges that affected the progress of our application. Technically, we faced challenges designing the back end of our application. None of us had much knowledge about how to implement the features we wanted to implement. While we all had a solid front end understanding, the backend required much more planning and learning. In order to solve this, we were all required to try to learn as much as we could and listening to an instructor's feedback during our design stages. We also had some issues understanding the requirements of each milestone document. For every milestone, we had either a part missing, or some kind of mistake on the document layout. Had we checked in with the instructor more often and asked more questions, we could have avoided these problems. However, towards the end of the semester, we improved on this by asking more questions about the requirements.