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

Revision #	**Version #**	**Date**
Milestone 1	Version 2	10/10/19
Milestone 1	Version 1	10/3/19

Table of Contents

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.

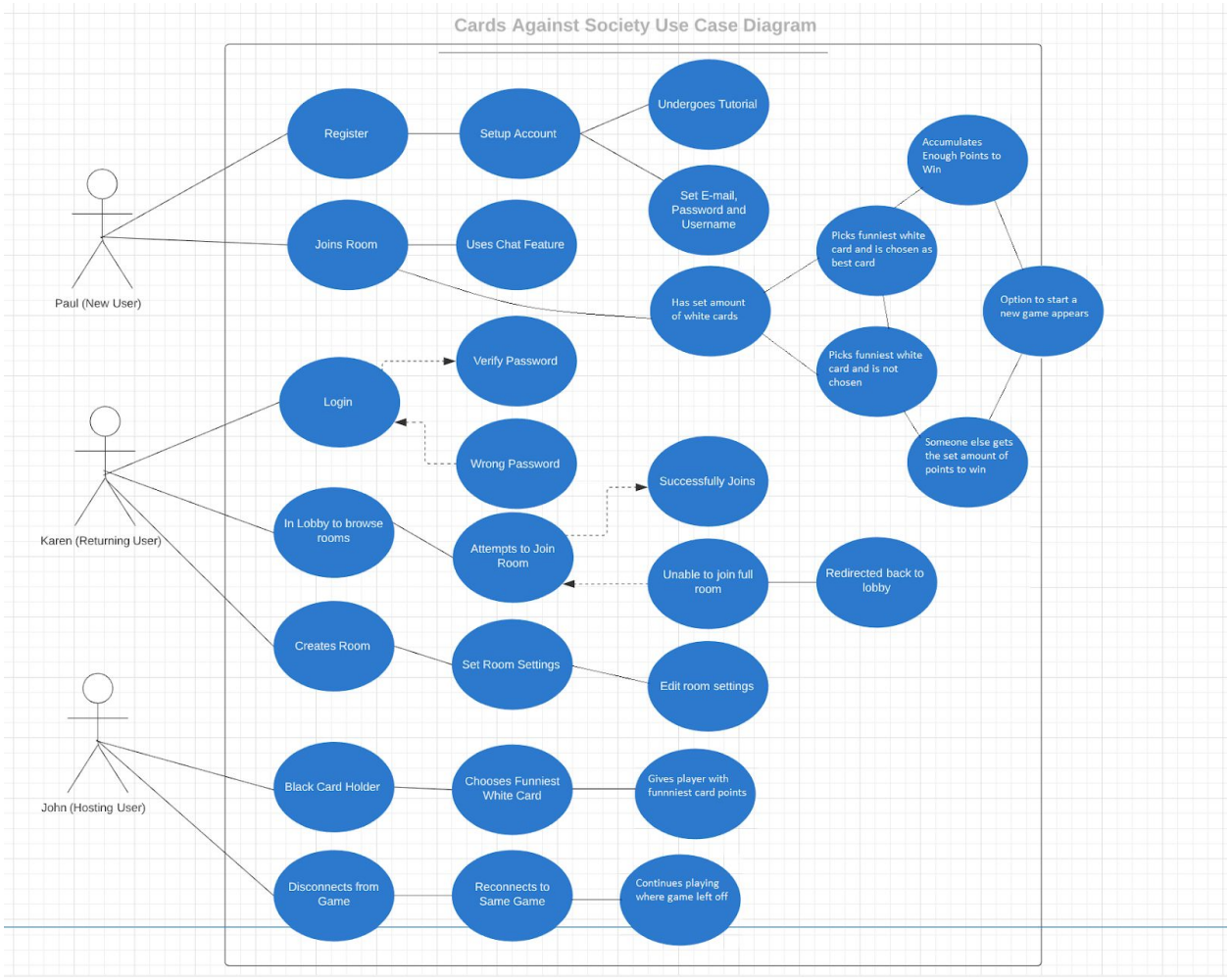
The game begins and the Black Card Holder reveals the black card which prompts the other players to choose a white card which responds best to the contents of the black card. Paul picks a card that the black card holder likes and is therefore given points. If he has enough points then he will win the 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 to 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.


Cards Against Society Use Case Diagram
Paul (New User)
Karen (Returning User)
John (Hosting User)
Register
Setup Account
Undergoes Tutorial
Set E-mail, Password and Username
Joins Room
Uses Chat Feature
Has set amount of white cards
Picks funniest white card and is chosen as best card
Picks funniest white card and is not chosen
Accumulates Enough Points to Win
Option to start a new game appears
Someone else gets the set amount of points to win
Login
Verify Password
Wrong Password
In Lobby to browse rooms
Attempts to Join Room
Successfully Joins
Unable to join full room
Redirected back to lobby
Creates Room
Set Room Settings
Edit room settings
Black Card Holder
Chooses Funniest White Card
Gives player with funnniest card points
Disconnects from Game
Reconnects to Same Game
Continues playing where game left off

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

- System shall update game data in real time
- System shall update chat message data in real time using sockets
- Game states shall be persistent, saved by the system and loaded when a player joins
- Application shall be developed using the software stack declared in M0
- Application shall be deployed using an AWS EC2 instance
- Data shall be stored in a MySQL database
- Application shall display game without the use of images
- Application shall allow up to 5 users in one game room
- Only relevant data shall be collected, such as user's win rate
- System shall be compatible with at least two of the major browsers, versions listed below
- UI shall be implemented using React
- Game animations shall be implemented using React-Animations
- Application shall load a default game state when a room is created
- Application shall run off of the Master branch of the team's git repo
- Database shall store all of the game cards to be used during the game
- Application shall have a responsive game UI
- Application shall keep public and private chat in their respective places

6. Competitive Analysis

	Pretend You're Xyzzy	Pictures Against Humanity	Cards Against Society
User Interface	+	+	++
Custom Cards	-	+	+
In-game Chat	+	+	++
Private Room	+	+	+
User Auth	+	+	+
Tutorial	-	-	+

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**