Product Backlog

Problem Statement

As computers and the internet have become more commonplace in everyday life, the desire to play web-based versions of traditional card games has increased significantly. However, most web-based card games lack a user friendly interface and/or multiplayer version that effectively reduces the hassles of physical aspects of traditional card games such as shuffling, dealing, and managing the game.

Background Information

Over the last 30 years, computers have become increasingly more powerful and easy to use. This change has made computers more predominant in everyday and has shifted how most people consume entertainment. Before this change, one of the primary modes of entertainment was card games. Card games provide a way to have social interactions with family as well as friends. Unfortunately, however, not all card games have made the transition from a the physical form to an online version. Spit, our favorite card game, is one of those games and our project hopes to bring this popular card game into the 21st century.

Environment

We will be coding our project in Javascript on both the front and back end. Our backend will be running a Node.js/Express.js server. This server will communicate to all users and store information about games and users. Communication between users and server will be through a framework such as Socket.io. React will store the game state internally using redux, updating it from the server asynchronously using sockets, and will render the GameView, passing in the current game state as a parameter. The GameView will call a function upon the player playing a card, which will send a request to the server, which will mutate the state if the play is valid, and the cause the GameView to be passed a new game state. The GameView will then animate the change in state using the canvas, i.e. moving a card from one stack to another. All data will be stored in a MongoDB database.

Functional Requirements

Id	Functional Requirement	Hours	Status
1	As a user, I would like cards to animate during play	7	Sprint 1
2	As a user, I would like to be able to access Spit on a website	4	Spring 1
3	As a user, I would like to play Spit with two or more people	6	Sprint 1
4	As a user, I would like to chat with other players	8	

5	As a user, I would like to be able to play with random people	6	Sprint 1
6	As a user, I would like to be able to play with friends	6	
7	As a user, I would like to be able to view my rank against other players	5	
8	As a user, I would like to be able to play using keys	5	Sprint 1
9	As a user, I would like to have a username	5	
10	As a user, I would like a select a ‘hand’ using keys	4	Sprint 1
11	As a user, I would like to select a ‘pile’ using mouse click	4	Sprint 1
12	As a user, I would like a tutorial so I can learn how to play	8	
13	As a user, I would like a timer to know how long a game lasted	4	Sprint 1
14	As a user, I would like to report inactive users	4	
15	As a user, I would like to report users who use vulgar or obscene language in the chat	4	
16	As a user, I would like ‘rounds’ to be triggered by the system when no one can play a card	6	Sprint 1
17	As a user, I would like a help page explaining the game	4	
18	As a user, I would like to combine hands if they are of the same value using a key command	4	Sprint 1
19	As a user, I would like to be place cards from my ‘deck’ to my empty hands using the space bar	4	Sprint 1
	Total		

Non-Functional Requirements

1) As a user, I would like my account information to be secure
2) As a user, I would like to play on different types of browsers
3) As a user, I would like seamless experience without any lag
4) As a user, I would like an intuitive interface
5) As a user, I would like to be matched into games quickly
6) As a user, I would like games to be fast paced
7) As a developer, I would like our servers to be scalable and able to handle traffic at peak hours

8) As a developer, I would like to track the traffic on the website
9) As a developer, I would like the ability to automatically push to staging and production servers
10) As a developer, I would like the website to stay functional even if one server crashes
11) As a developer, I would like a way to seamlessly interact with the database

Use Cases

Action	**System Response**
Case: **Animate cards during play**	
User selects card	Card expands to show selection
User places card	Card animates the move to the pile
Case: **Access Spit on a website**	
User searches Spit or our URL	
DNS directs the user to our homepage	Our homepage is sent to the user via http
Case: **Play Spit with two or more people**	
User selects create game button	Create game appears
User selects random game or game with friends	Players are matched to each other
	Game begins
Case: **Chat with other players**	
Goes to our homepage	
Selects play now button	Player is matched into a game
	Chat dialogue appears in game with which users can send or receive messages.
Case: **Play with Random People**	
User selects play random game	Creates a lobby
	Find other players looking for a random game
	Populates the lobby with those players
	Starts game

Plays game	
Case: **Play with friends**	
User selects play with friends	Creates lobby
	Finds friends who also want to play
	Populates lobby with those players
	Starts Game
Plays with friends	
Case: **View my rank against other players**	
User plays games	System creates rank for user based on win/loss ratio.
User selects leaderboard	Leaderboard window pops up displaying user ranks
Case: **Select a hand**	
User is playing game	
Hits keys 1, 2, 3, or 4	System selects a hand
Case: **Place on a pile**	
User is playing game	
Selects a pile by clicking on it with his mouse	System tries to place card from selected hand onto that pile
Case: **Combine hands**	
User is playing game	
User selects one hand	Animate selected hand
User selects second hand	Combine the two hands if they are of the same type
Case: **Automatically triggered rounds**	
Users are playing a game	
Neither player can play a game	System triggers a countdown from three to zero.

	System automatically takes card from each player's deck (if they have cards left) and places them on each person's respective pile
Case: **Place cards from deck to empty hands**	
User is playing game	
Presses space-bar	System places a new card into an empty hand if one exists
Case: **Have username**	
User selects create account	Brings up new view to ask for email, username, and password
User provides email, username, and password	Creates account for the user and stores credentials in database
Case: **Clean Design**	
User goes to website and plays game	Display shown will have clean, consistent, and flat design
Case: **Help page explaining the Game**	
User selects help button	Displays the rules of the game and a basic overview on how to play
Case: **Tutorial Page**	
Selects tutorial button	Starts a game against itself
	Walks the player through the rules using dialog boxes
	Highlights the different actions users can make
	Waits for the user to make those actions
Users make those actions	
Case: **Have timer showing game length**	
Selects start game	Starts the game and timer
	Displays the time elapsed since game start at the top

Player ends game	Timer stops and sends summary to the user describing who won and thee elapsed game time
Case: **Report inactive or vulgar language users**	
User starts game	Starts the game
Notices a user not making moves or is making vulgar comments in chat	
Hits the report user button	Terminates that user and adds a ‘strike’ to their account. If they get three strikes they are banned from the site.
Case: **System detect inactive users**	