

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 2
2	As a user, I would like to be able to access Spit on a website	4	Sprint 2
3	As a user, I would like to play Spit with two or more people	6	Sprint 2
4	As a user, I would like to chat with other players	8	Removed

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	Sprint 1
7	As a user, I would like to be able to view my rank against other players	5	Sprint 1
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	Sprint 1
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	Sprint 2
13	As a user, I would like a timer to know how long a game lasted	4	Sprint 2
14	As a user, I would like to report inactive users	4	Removed
15	As a user, I would like to report users who use vulgar or obscene language in the chat	4	Removed
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 to be ranked when the game ends.	5	Sprint 2
18	As a user, I would like to combine hands if they are of the same value using a key and mouse command	4	Sprint 1
19	As a user, I would like to place cards from my 'deck' to my empty hands using the space bar	4	Sprint 1
20	As a user, I would like the game to end when nobody has any valid moves left.	4	Sprint 1
21	As a user, I would like the ability to see a replay of a completed game	10	Sprint 2
	Total	113	

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 - DONE

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 with Random People	
User selects play now	Places player into queue
	Find other players looking for a random game
	Populates the lobby with those players
	Starts game when 4 players are found
Plays game	
Case: Play with friends	

User selects new lobby or manually creates their own lobby by visiting spiton.us/game/{gameid}	Creates lobby
User sends invite links to friends	
Friends click invite link	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. 1st place gets 15 points, 2nd 10 points, 3rd 5 points, and 4th 0 points
User selects leaderboard	Leaderboard window pops up displaying top ranked users from highest to lowest. Users with 0 points will not be shown.
Case: Select a hand	
User is playing game	
Hits keys 1, 2, 3, or 4	System selects a hand, from left to right
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 value
Case: Automatically triggered rounds	
Users are playing a game	

Neither player can play a card	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 first empty hand if one exists from left to right
Case: Have username	
User selects create account	Brings up new view to ask for username
User provides username	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: Have timer showing game length	
Selects start game	Starts the game and timer
	Displays the time elapsed since game start at the top
Case: Player ends game	
All players have no more possible moves	Result page appears describing who won and the ranking of the players
Case: Replays	
Finish a game	Confirm game is over when game result screen pops up
Visit spiton.us/replays/{gameid} Ex: spiton.us/game/4392 finishes Visit spiton.us/replays/4392	Replay will automatically play.
Case: In-game leaderboard	
Start a game	In-game leaderboard will display the number of cards each player has left. Leaderboard is organized by who is currently winning (ie: players with less cards will be on top)

