CSC 667 Internet Application Design and Development
John Roberts

Pirate Party

Team: Sushi is the Best

Team Members:
Logan Figgins
Theofanis Koutoulas
Zack Watkins
Derek Florimonte

GitHub Master Repository: https://github.com/SFSU-CSC-667/term-project-spring-2017-sushiisthebest

Contents

Pirate Party

“Yarg, ye land lubbers want a new challenge do ye!?! Aye, I’ve got a challenge for yur, you seem them other pirates… I want their ships, their loot, and heck I could use a good pair o’trousers! Attack em with what ever ye can find! ATTACK!”

Our group decided to embark on an ambitious project to successfully design a new game inspired by the popular drinking game King’s Cup! In this game, you start by choosing a username, an Avatar (all pirate themed characters), a Sushi Card (variety of fun/different abilities that takes place when the player pulls a joker card) and a rule card (another fun customizable abilities that takes place when the user pulls a jack card). The user is pit up against 2 to 4 other players in a battle to the (possibly less foreseeable*) death!

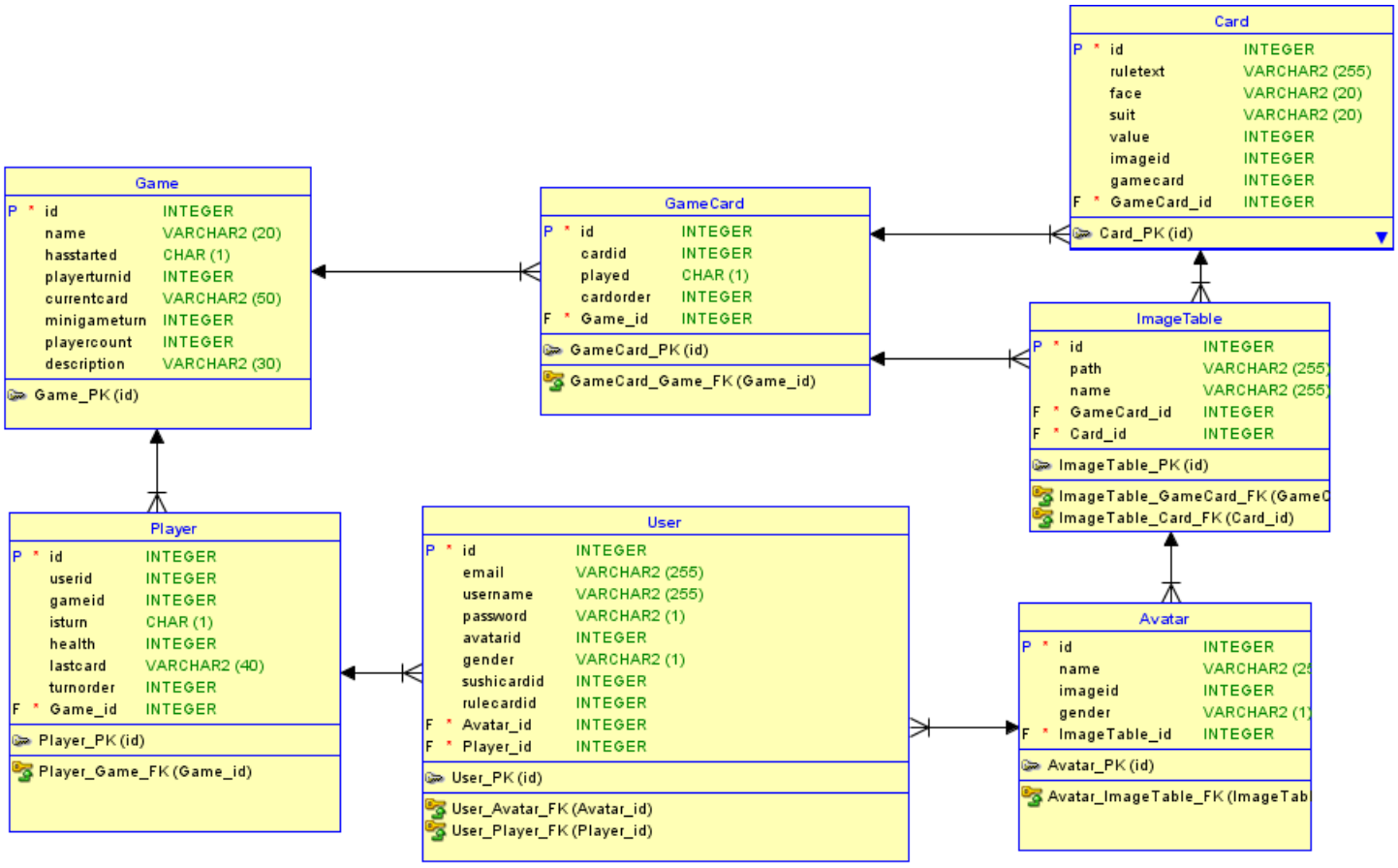
Each games starts with a new deck, based on the classic playing card deck, with new, custom rules and game play. Much like our inspiration, our card game fixes some of the classic drinking rules with our newer health based version. This allows for a more “PG” feel, while also maintaining the influences from the classic version. The health system allows for dynamic in game play, and challenges the user to test their mixture of skill and luck!

The end of the game is signified by either the death’s of all players (except one, ideally) or if the last king is pulled. In the next iteration of Pirate Party, the players who survive under certain conditions, will be rewarded with big experience gains, while dead players will still be rewarded for their feats in life!

Needless to say, the project was quite a large endeavour. With the adjusting for new web technologies, learning a new language and a collection of new frameworks, we needed to communicate, and work together. As a group we decided to tackle such a big project in many parts, so we didn’t get overwhelmed with the size of the project. Splitting up the back-end and the front-end down the middle between four of us. Back-end was Logan and Derek while the front-end was Zack and Frank. Yet with game mechanics and certain designs we made decisions as a group.

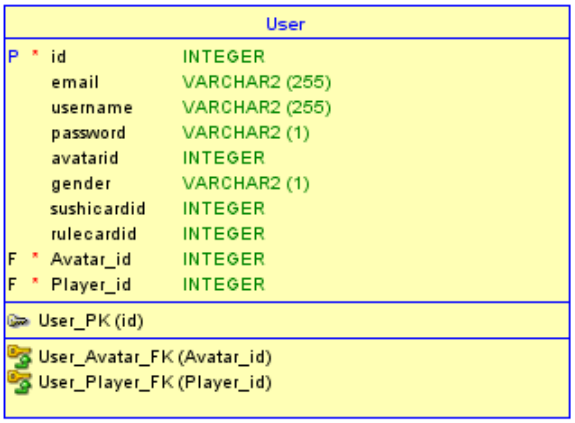
Final Entity Design Revisions

After much trial and error, going through many brainstorming sessions and iterations of our ideas, and coming to a realistic conclusion for the scope of the project we came many conclusions. Having to do with the structure of the database and its relations between tables, while also considering the frequently called and required information, we structured our data in an easy to use scheme. The following is our final revisions on our overall scheme of data;

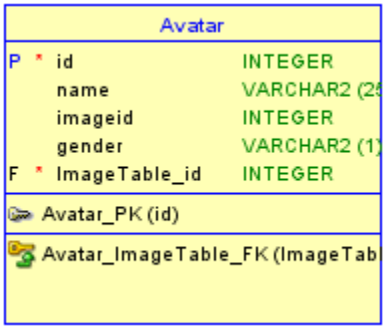


(Figure A) Revised Database Model

Many of our tables may contain extra non-used data, as we, as a group, decided to continue work on this project and is the result of preparing for future iterations. Our “User” table contains many id (INT) columns for use of tracking user choices on extra game functionalities. We also intend to continue with experience and certain benefits for this. Figue a) shows a demonstration of the current and up to date scheme of our data, and shows the relations between the tables decently accurately. We did omit certain columns that we not relevant after many discussions or game functionality. Figure b) and Figure c) continues to show a more explicit view of our “User” table, and also demonstrates the importance of the Avatar relation.

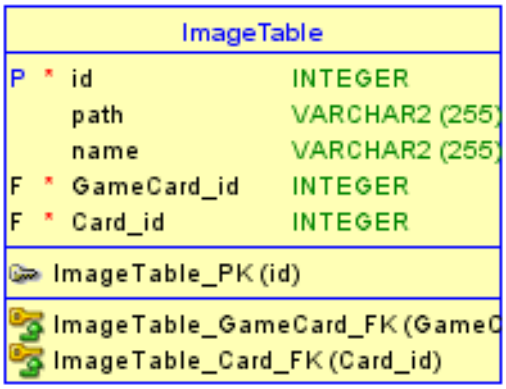


(Figure B) “User” Table

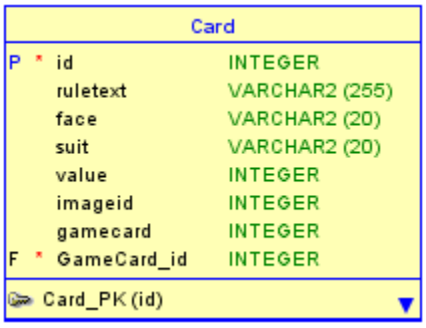


(Figure C) “Avatar” Table

Many of our table connect to the “ImageTable” Table (Figure D), either directly, or indirectly. “ImageTable” Table contains all the main image paths, which leads to the directly that allow the server to “serve” the image files. Many background images, and other extraneous image files are provided with initial GET requests. The “ImageTable is referenced by multiple other tables including “Card” (Figure E) and “Avatar” (Figure C), which provide the images for the player profile character, and profile card choices.

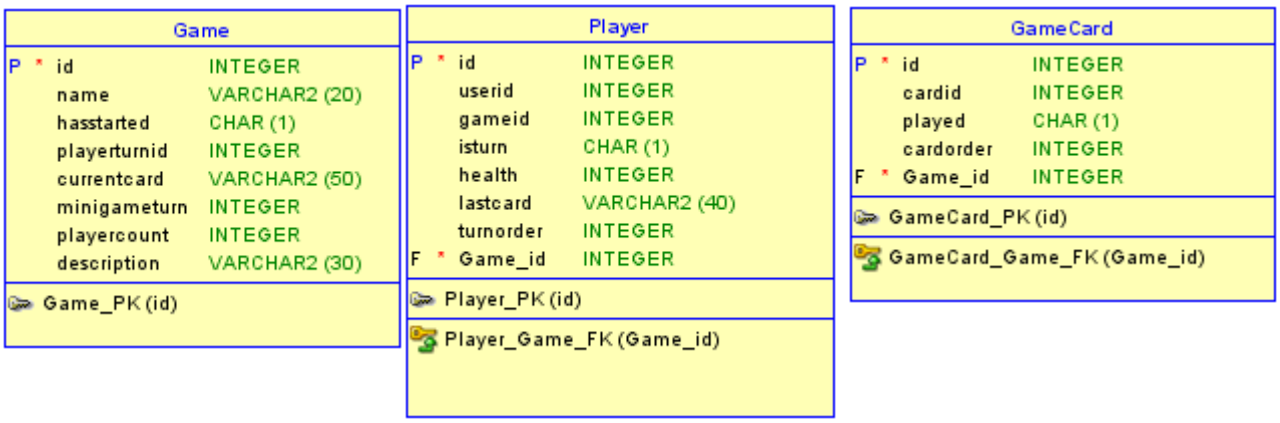


(Figure D) “ImageTable” Table



(Figure E) “Card” Table

During game initialization, where players have successfully joined a player created game, the tables game objects are initialized and the players' information in the current game is recorded. "Player" Table (Figure G) related the "User" Table to the in-game "Player" Table in order to record various things, such are turn order, and last cards played. "GameCard" Table (Figure H) refers to the deck created for this current game and relates the "Card" Table (Figure E) to the game deck that is used by the players. "Game" Table (Figure F) then relates both of these other tables and records whose turn it is, whether the game has started successfully, and other in game information.



(Figure F) "Game" Table (Figure G) "Player" Table (Figure H) "GameCard" Table

Find API Revisions

/user.js

a. GET ‘/:username’
    i. Renders the /username ‘s profile and default, or customized settings
b. GET ‘/login’
    i. Authenticates the user against the username and hashed password
c. GET ‘/register’
    i. Renders the registration page
    ii. Allows user input and stores as a new user
    iii. Enters email, username and password to be hashed
d. POST ‘/:username/changeAvatar’
    i. Authenticates and changes the Avatar in the database
    ii. User’s choice is recorded and the correct table is updated
e. POST ‘/:username/changeSushiCard’
    i. Authenticates and changes the Sushi Card in the database
    ii. User’s choice is recorded and the correct table is updated
f. POST ‘/:username/changeRuleCard’
    i. Authenticates and changes the Rule Card in the database
    ii. User’s choice is recorded and the correct table is updated

/games.js

a. GET ‘/create’
    i. Creates a new game and renders the intended game page
b. POST ‘/create’
    i. Authenticates the user and allows for creation of a new game
    ii. Adds necessary values from player incoming and for starting a new game
    iii. Values include, gameid, and the path for the newly created game
c. POST ‘/join’
    i. Allows other users to join listed games and renders the correct environment
    ii. This is also updates for new entering players
    iii. Also allows for a message to be sent to the user if the game is full
d. POST ‘/start’
    i. Allows the user to start the game, which then broadcasts a start to the other players

ii. Will then initialized the rendering of the correct in-game page

e. POST '/leave'
   i. Allows users to leave a game
   ii. Rendering the changes in the page, as users leave
f. GET '/:gameid'
   i. Allows the population of the games table to show waiting games
   ii. This also updates the pages with the correct player numbers and

/Pirate-Party

Main Routes for most in-game functionality

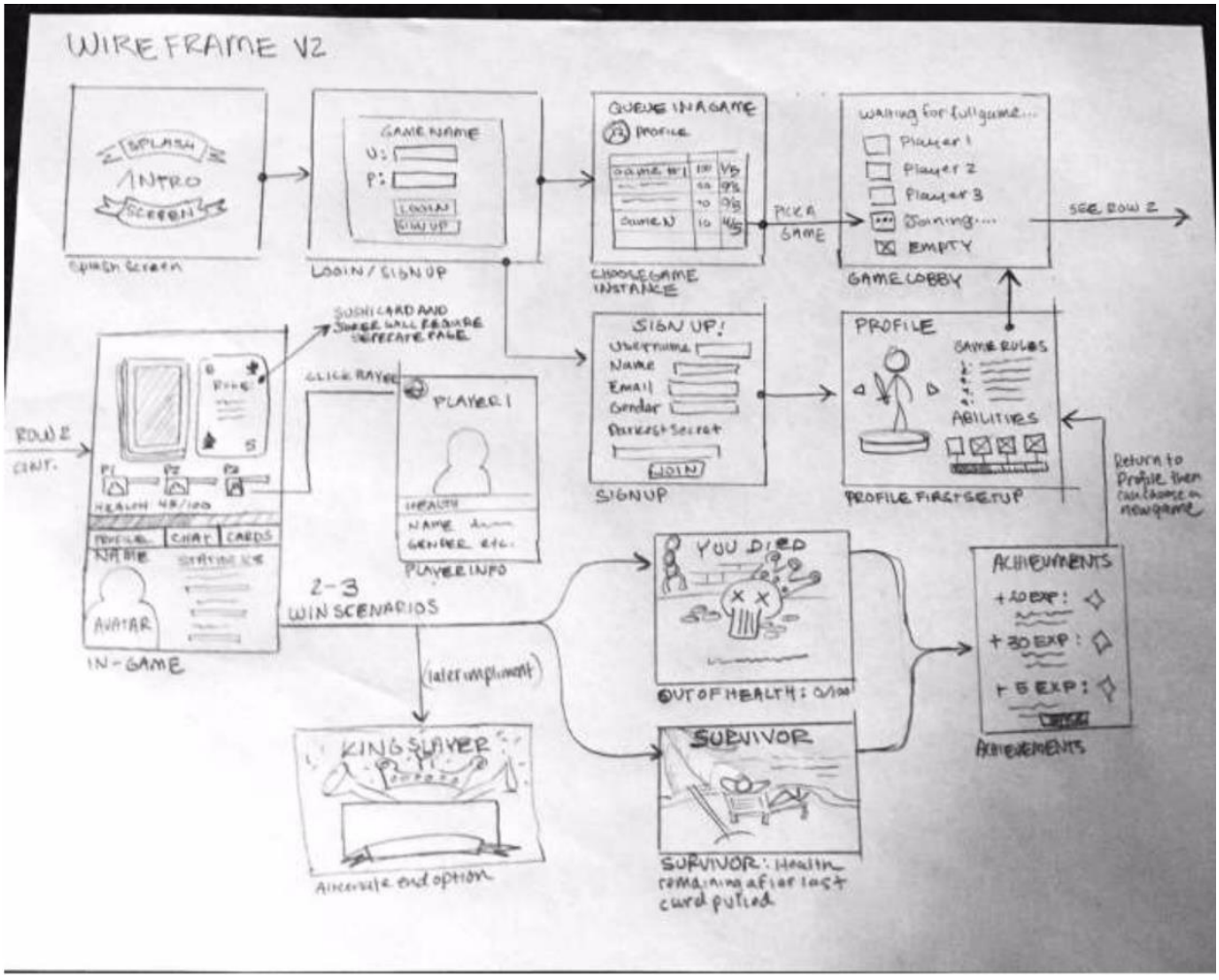
a. GET '/load-view/:gameID'
   i. Loads the updated game view, while in game
   ii. Receives the current card number and receives subsequent data of said card
   iii. Renders the "game-table" accordingly
b. GET '/draw/:gameID'
   i. Allows the user to draw from the intended game deck
   ii. This is also told to other players, and the game deck intended for this game is updated
c. POST '/:gameId/next-turn'
   i. Updates the current player turn, and cascades this change to the turn order
   ii. Records whose turn it is, and informs the users accordingly
d. POST '/:gameID/target/:playerID'
   i. Allows the current turn's user to target another player
   ii. This is in response to in-game functions, this happens because of a card pull
   iii. The damage is recorded from the rules of the played card
   iv. It is then applied to the currently targeted player's health total
   v. This also addressed the case where a user is targeted for a healing card
   vi. Updates the player's health accordingly
e. POST '/:gameID/bard'
   i. Allows a particular case when a bard card is pulled
   ii. Specific context, where player health is updated accordingly, and death is checked
   iii. Diversifies card rules and game functionality
f. POST '/:gameID/mayhem'
   i. Another rule card functionality allowing for a randomly positive or negative action from happening to a targeted individual
   ii. Affect the user when a specific card is pulled

    iii. User’s Health is updated accordingly, and death is checked
    iv. Diversifies card rules and game functionality
g. POST ‘/gameID/me’
    i. Another rule card functionality
    ii. Affects all users with specified gender when a specific card is pulled
    iii. Card rule is read and damage is applied
    iv. User’s health is updated accordingly
    v. Diversifies card rules and game functionality
h. POST ‘/:gameID/wenches’
    i. Another rule card functionality
    ii. Affects all users with specified gender when a specific card is pulled
    iii. Card rule is read and damage is applied
    iv. User’s health is updated accordingly
    v. Diversifies card rules and game functionality
i. POST ‘/:gameID/dudes’
    i. Another rule card functionality
    ii. Affects all users with specified gender when a specific card is pulled
    iii. Card rule is read and damage is applied
    iv. User’s health is updated accordingly
    v. Diversifies card rules and game functionality
j. POST ‘/:gameID/bomb’
    i. Another rule card functionality
    ii. Affects all user’s when a specific card is pulled
    iii. Random player is chosen and damage is recorded
    iv. User’s health is updated accordingly
    v. Diversifies card rules and game functionality
k. POST ‘/:gameID/king’
    i. An important rule card functionality
    ii. Symbolizes the end of the game if it is the last king to be pulled
    iii. Affects every user playing
    iv. Updates all user’s health accordinly
l. POST ‘/:gameID/king’
    i. LAST KING Symbolizes end game
    ii. Also does much more damage
    iii. Allows for extra experience earned if certain conditions are met and user’s health is still above 0%

Front-end Design Principles

One of the main aspects of the game we really wanted to develop was having a fun design that all sorts of people can enjoy. So the actual design and wireframing of the project was a huge part for our team.

As you can be seen in the picture our wireframes were a huge part in designing our project. Not only can you see the time and effort put into the design of the project, but you will also see a lot of changes being made as the semester went on.



Our design changed in drastic ways from the start of this project all the way to the end of it. We still used the same theme and stayed true to what we wanted to accomplish, but we found better ways to do certain aspects of the game.

As you can see the design from the wireframe to the final product really changed and evolved into something that's easy on the eyes. Not to mention we wanted to make it really easy for the user to follow.

The main screen that people will be interacting with is the main battle screen we have right here. As you can see the design and the avatars are very playful and easy for people to enjoy. The chat integration was a very important part also, because it's what

people always migrate to. So we made it a big part of the design and easy for people to access.

When we first started this project we were really focused on the back-end of the project and how much work it would have been. But we ended up dedicating a lot of time into the front-end aspect of the project. The design really tied into the back-end and was a very important part into making this project something we could be proud of.

Problems And Challenges Encountered

There was multiple problems we ran into with this project. Scheduling with member's wasn't that big a problem since we all wanted to make this project something we can all be proud of. Everyone was at the meetings and all members contributed to the final product.

The main problems that we encountered was organizing our JavaScript code so it can be readable between each other. Having the code be organized in a professional way and being accountable with each other.

Authentication was a very tough situation because none of us ever tackled something like that. So we had trouble understand the logic behind that, which took more time than we wanted to spend on it. Logan finally found the solution to the problems we encountered with authentication and he was able to work through it.

One of the most annoying problems we faced was with the requests we dealt with. Most of the requests weren't Ajax which we had to struggle with a lot. Since that was happening we had to figure out a different way to handle them. After a long while we figured out that we can do them with the Cookies. It was forced but it was the best solution we came up with. Which seems to have handled the problem for us.

As mentioned before animations and the UI was something that took more time than we imagined. Not only because we wanted to make something very satisfying but also because we ran into many problems. Design space, having people be targets in the game which took a long time. There were a lot of aspects with the UI that we didn't anticipate which was a great learning experience for all of us.

Another problem that was something different that we never encountered before was dealing with different term branches. Then having to harness all the term branches so we can get to a final product.

Like the first project we researched before we started coding, we wrote on a white board and lined up everything needed. This helped us avoid possible problems that might have happened if we didn't do that.

Conclusion Of The Project

During the weeks of working on this project we changed our game in so many ways, for better and for worse. Some of the better ideas we were able to implement into our final product. Other's we had to leave behind for a later date because we just didn't

have enough time to put certain aspects we wanted into the game. Working together and helping each other was a fun time and we enjoyed the game we built.

Designing this from top to bottom was something we never have done before, so it was a great experience. It took us out of our comfort zone and made us solve problems that took a lot of time. This is a great project for anyone to do because it helps you learn certain things that you wouldn't have learned before.
We plan to keep working on this project after the semester is over because we enjoyed making this game. More importantly enjoyed working with each other.