CSCI 3308 Fall 2022
Instructors: Prof. Nath

Group Project (Trash Royale)

Sean Vestecka, Nic Perrault, Niko Johns, Daniel Hatakeyama, Derek Norman, Tobey Switzer

December 2nd, 2022

Due Date ..... **Friday** December 2, 2022
Team Members ..... **Sean Vestecka, Nic Perrault, Niko Johns, Daniel Hatakeyama, Derek Norman, Tobey Switzer**
Student IDs **109753539, 108458614, 109685832, 109863366, 109755954, 110175207**

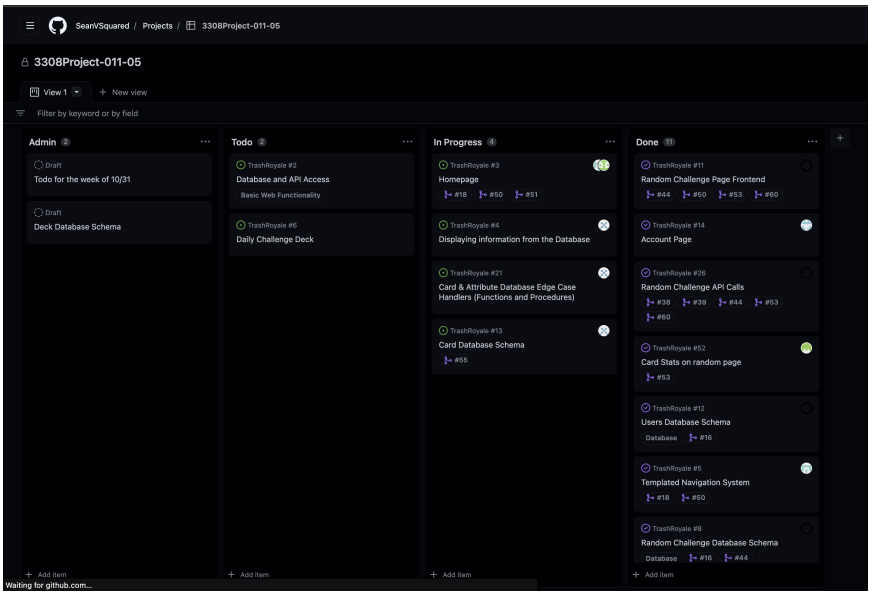
Contents

1 Project Description

Trash Royale is a multipurpose tool that is used in conjunction with the mobile game, Clash Royale. Our application allows users to have a refreshing experience in the game by completing custom card deck challenges and tracking stats with other users. Our tool is aimed at the more experienced users that are looking to test their skills. Specifically, users will be able to generate decks of randomly selected cards or hand-picked bad cards. The users can play with these decks in the game. After that, they can return to the website to verify their completion of a challenge and view their most recent matches. Users will have their challenge completion data tracked and displayed on a global leaderboard to compare with other Clash Royale players. The project will be developed as a simple web app that interfaces with an external API, specifically the API for Clash Royale. The stack will be the same stack used to create previous projects for CSCI3308, primarily using a NodeJS backend and an EJS HTML frontend, all being contained in a Docker.

2 Project Tracker

Project Tracker.



3 Version Control System

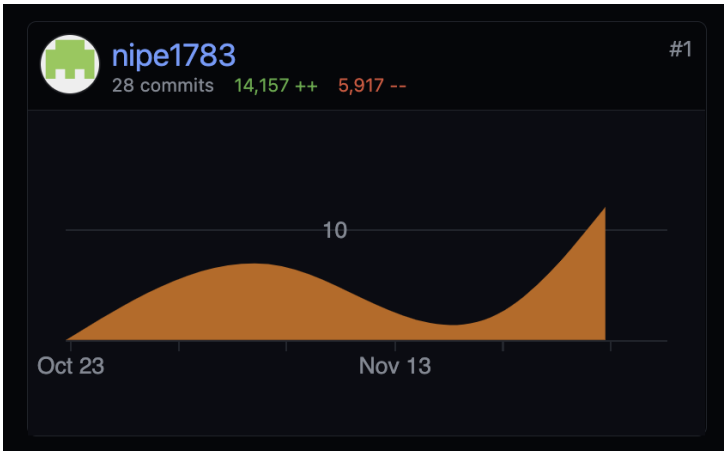
https://github.com/SeanVSquared/TrashRoyale

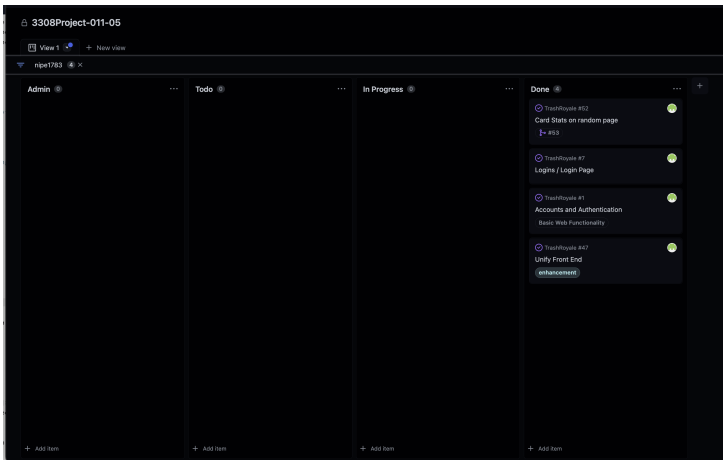
4 Video (Project Demo)

5 Contributions

5.1 Nic Perrault

Throughout this project, I primarily focused my efforts on the front end of the application. I contributed significantly to all styling and layout of the login, register, home and deck pages. The technologies that I used were: Github, VSCode, HTML, CSS, EJS, Node.JS and Docker. Below are images of my GitHub contributions.

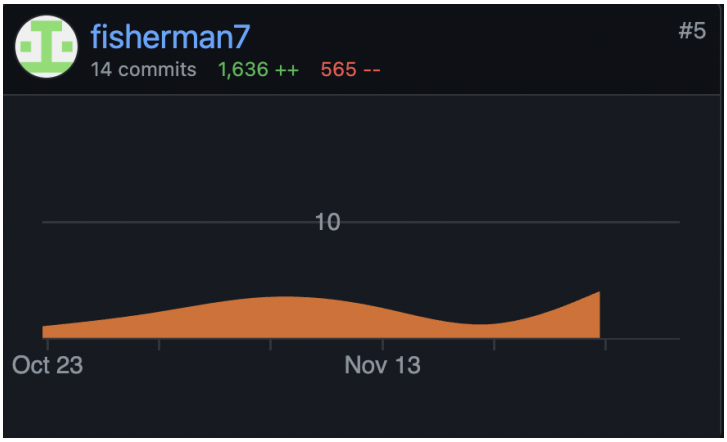


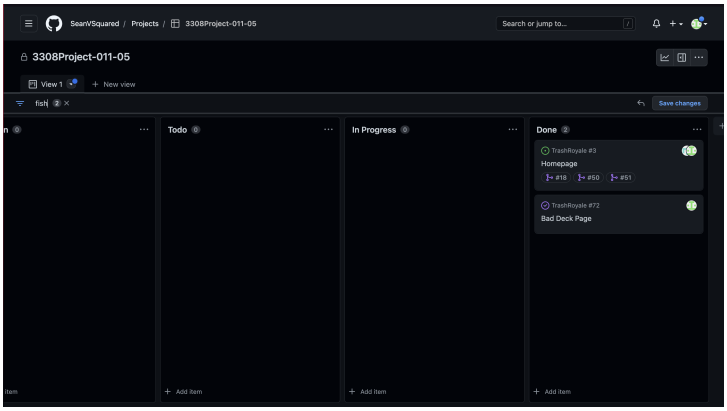


5.2 Tobey Switzer

During our time working on the project, I contributed a little bit everywhere and didn't focus too much any one feature in specific. My biggest contribution was focusing on the logic behind the "bad deck" generator page, which is essentially a modified version of Sean's Random Deck Logic. I also completed many niche parts of our app such as recording a backup demo of our project (**Link to Project Demo**), making the presentation we showed in class, and creating a logo that appears multiple times on the site:

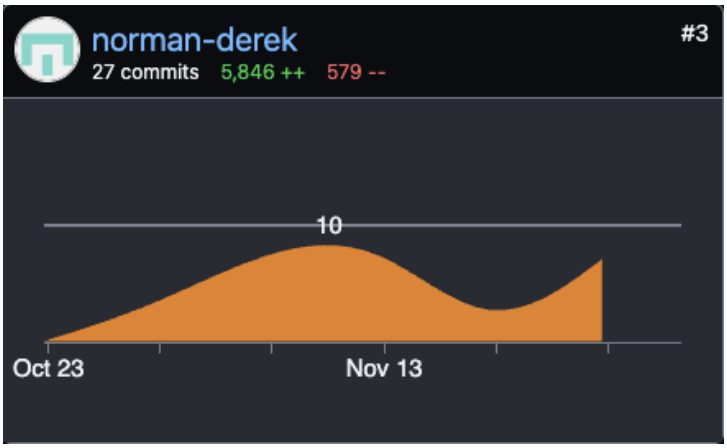
The technologies I used were Adobe Illustrator, GitHub, VSCode, HTML, CSS, EJS, NodeJS, and Docker. Here are the pictures of my GitHub contributions:

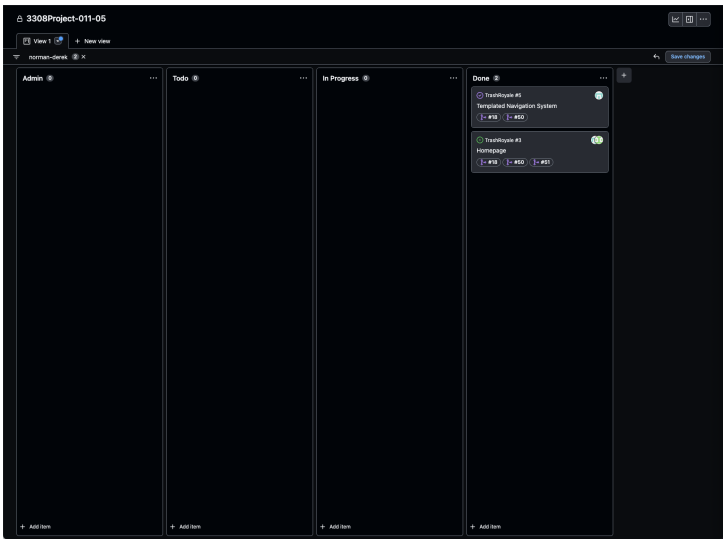




5.3 Derek Norman

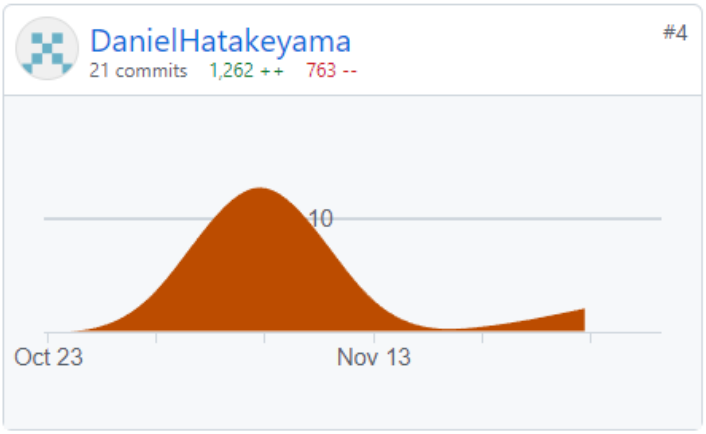
For the project, I primarily focused on the home page. I worked on the front-end and back-end of the home page and did a lot of work to bring functionality to the home page for the profile card, leaderboard card, and recent battles card. I also worked on the navbar and made it responsive depending on if the user is logged in. The technologies I used were: Github, VSCode, HTML, EJS, NodeJS, Docker, and Bootstrap.

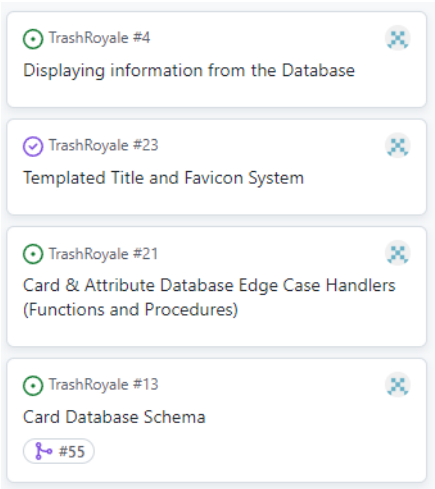




5.4 Daniel Hatakeyama

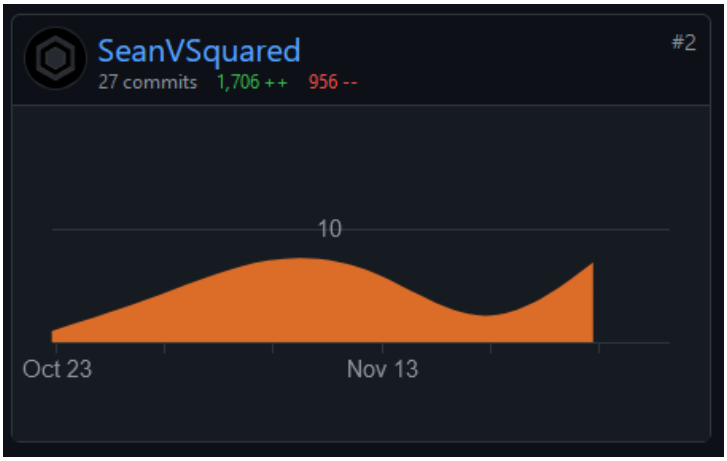
Throughout this project, I primarily focused my efforts on the back-end and database portions, mainly to handle Clash Royale cards, images, and attributes. I was also responsible for the cards display page and contributed to dynamic styling choices such as the generated deck links and custom titling. I used Github, VSCode, HTML, CSS, EJS, Node.JS, postgresql, and Docker. Below are my GitHub contributions.

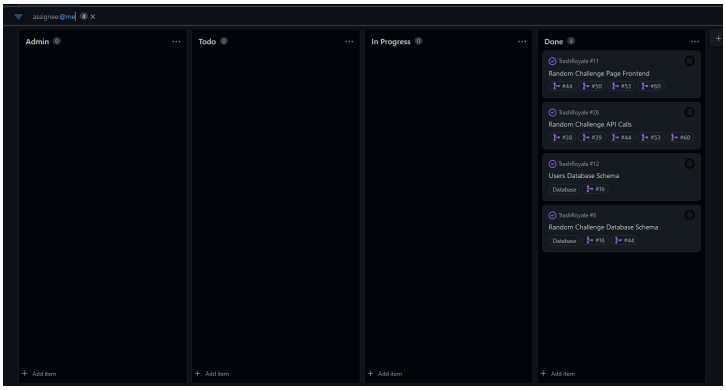




5.5 Sean Vestecka

During the project, I primarily worked on the back-end of the website. I worked to create the algorithms used to verify challenges along with the local databases for keeping track of challenges. I worked closely with Tobey and Daniel to ensure that the website's different challenges and card information worked well together. Additionally, I worked to maintain the project board, track issues on the GitHub repository, and generally manage the team's meetings and our pull requests to main. The tools I used were: paint.net, GitHub, VSCode, HTML, EJS, NodeJS, and Docker. Below are graphs of my contributions from the GitHub repository:

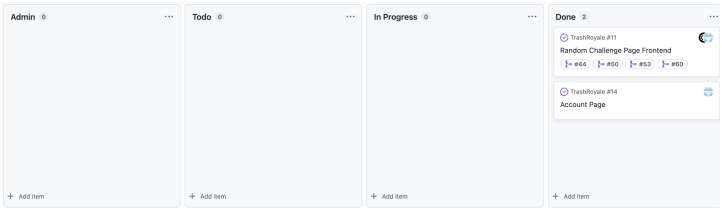




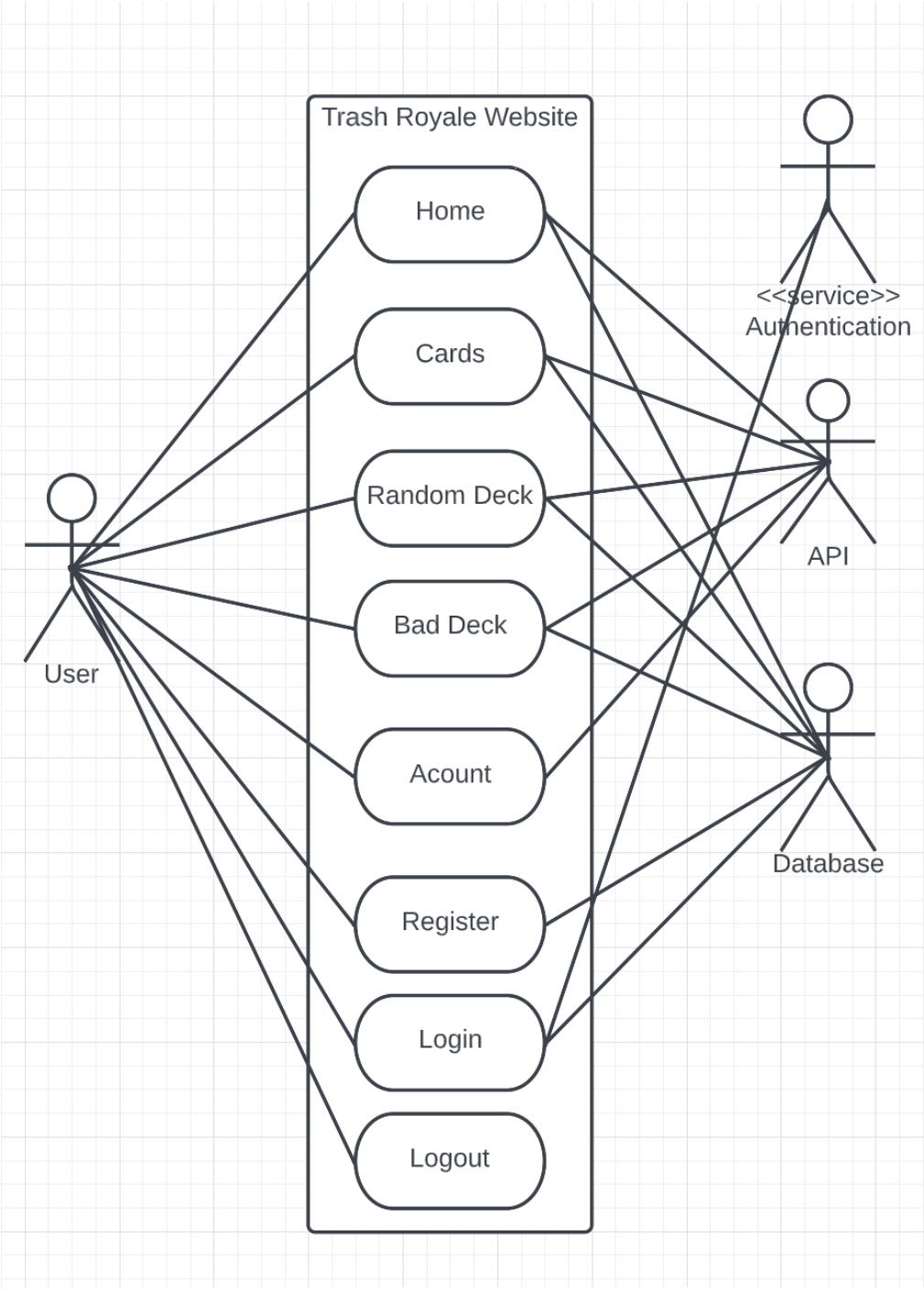
5.6 Niko Johns

Throughout the project, I worked primarily on styling the Account Page and some aspects on the Random Page. Primarily, I worked on creating the spacing for the cards, the card images, and making them clickable to display information when connected to Sean's backend. These took way longer than expected but it became a really good way of taking all the fragments of information I learned in the lab and putting them together. Unfortunately, my contribution from Thanksgiving break on this project leaves a bad taste in my mouth. I made minor contributions, but my Docker would shut my program off upon login so I was left spinning my wheels and not able to take on any more meaningful work to help my team.


NickJ-GH
14 commits 1,295 ++ 466 --
#6
10
Oct 23
Nov 13


Admin 0
Add item
Todo 0
Add item
In Progress 0
Add item
Done 2
TrashRoyale #11
Random Challenge Page Frontend
#44
#60
#53
#60
TrashRoyale #14
Account Page
Add item

6 Use Case Diagram



7 Test Results

For lab 11 we created several test cases that were checked throughout the development of our application. Our first test was verifying the user's login information via unit tests. Our team verified this data was being stored correctly by looking through updates in the database and comparing it to some expected login data. Users were able to create accounts with ease and link their Clash Royale accounts for use in the rest of the app. Our next test was verifying that a users mobile app ID was being tied correctly to an account in our application. Our group verified that this function was working correctly by user acceptance testing. More specifically, by looking at recent match history and comparing the shown results to the known match results. For this test our group

observed a slight delay with the Clash Royale API and updates would populate after about 30 seconds. Lastly, our group tested the functionality of our random deck generation. Our group checked this functionality via user acceptance testing. A dummy user would click the generate random deck button and observe that the cards were being generated correctly and this card data was being stored. Throughout this check we did observe some visual obstacles as the card images used by Clash Royale were various sizes and each had a unique amount of background space. This obstacle made it difficult to neatly organize the cards in a row without looking off-centered.

8 Deployment

Currently, our app is not deployed anywhere accessible to the public. We have future plans to polish up our project and host it somewhere semi-permanently. We want to look at and try some different free hosting options. And also I know a few people in our group are paying for cheap hosting and may be willing to throw it up on there temporarily. So currently our app isn't live, but we have solid future plans. In order to deploy it currently, it can be run with docker on a local machine and accessed from the machine's local IP address.