PlayPal Project Report

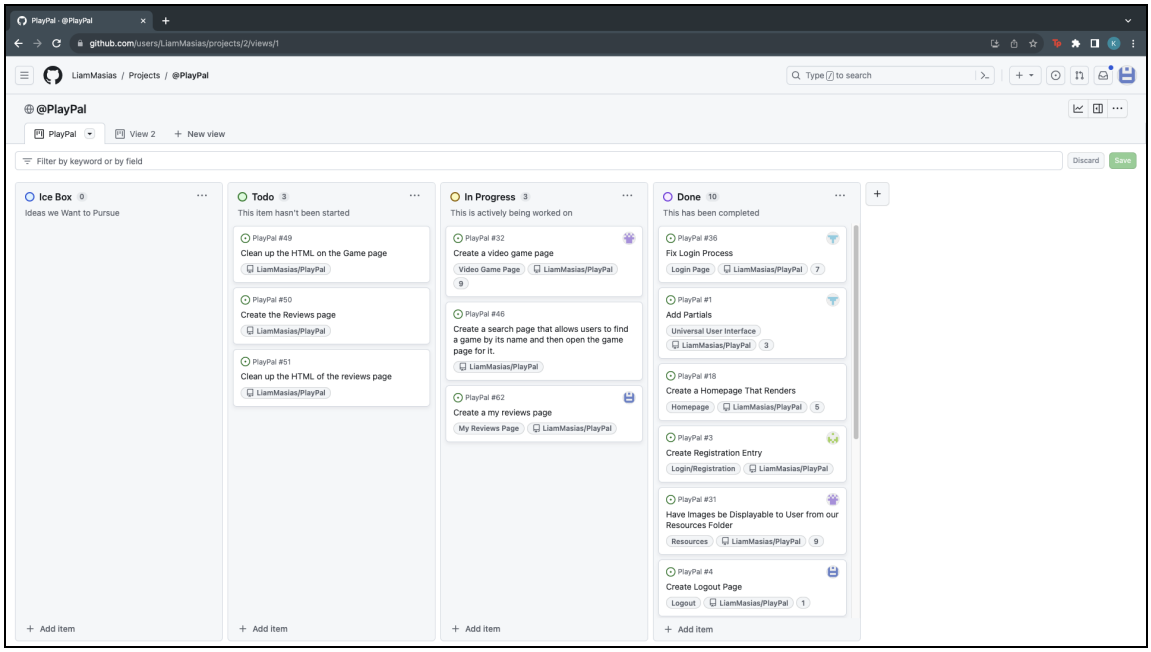
Team #4: Daniel Desmond, Gray Groves, Kevin Nguyen, Liam Masias, William Balliette

I. Project Description

PlayPal is a global social network for video game discussion and discovery, inspired by GoodReads, IMDb, and Letterboxd. The goal of this project is to create a community-driven environment where users can interact with each other centered around video games. Much like GoodReads for books and IMDb for movies, PlayPal strives to connect users by allowing them to give their reviews of video games, see what games their friends are playing, and discover new games to play. The Discover page displays a variety of games while also categorizing them by genres such as indie, shooters, and platformers. If you find a new game that you like, you can use our embedded links to buy the game across several different sellers such as Target, Amazon, and GameStop. Once you've found a game, you are able to share your thoughts on the game and rate it on a scale of 1-5 stars. After posting a review, you'll be able to see a list of all the reviews anyone else has created for that specific game. Furthermore, you can see a list of all the reviews that you have posted across all games on the My Reviews page. In summary, PlayPal creates a platform for users to join a community with a core aspect of video game discussion and discovery.

II. Project Tracker

Here is the link to the Project Tracker: https://github.com/users/LiamMasias/projects/2



III. Video

Here is the link to the video demo

IV. VCS

Here is the link to the GitHub Repository: https://github.com/LiamMasias/PlayPal

V. Contributions

Kevin

- My main task of this project was to implement the My Reviews page which displays a list of all your current reviews. I created a reviews table to link to our users table which enabled me to store all the review information for each user including, review description, rating, and timestamp. I then made a call to the IGDB API to retrieve the game name and game picture cover to display alongside

each users' review. Furthermore, I implemented an add review feature and the ability to delete your reviews on the same page. I routinely checked the database and ensured that everything was being stored correctly across Reviews and My Reviews page. Another page I did but isn't currently displayed is an All Reviews page that lists all the reviews in the database. That code is currently commented out because I did not see an immediate use for it. Lastly, I created the slideshow presentation.

Will

- I was responsible for several of Playpal's pages: the Discover page, the Search page, the Reviews page, and the Game page. For the Discover page, I used several calls to the IGDB API to get and display the information from hundreds of games filtered by genre and popularity. For the Search page, I did much the same thing for a larger collection of games. For the Reviews page, I took a raw framework from Daniel and made it more aesthetically pleasing. For the Game page, I used a framework designed by Liam and built a clean landing page inspired by the landing pages for similar sites such as IMDb, Letterboxd, and Goodreads.

Daniel

- My role for this project was to get our APIs started and decipher calls that could be then used later. The first bit of work I did was to get the IGDB API working for our website and figure out the best ways to make calls to it to pull the data that we needed. This was later further perfected by myself and Will for the areas we needed it in. I was also responsible for the API design of the game page and choosing the right calls and structure to get the right game from the IGDB

database. I also worked on the API calls for the Reviews page including the calls to our PostgreSQL database to get the user reviews by game ID and getting the average score of these reviews. The aesthetics of this page were then further refined by Will as that was more his specialty than mine.

Liam

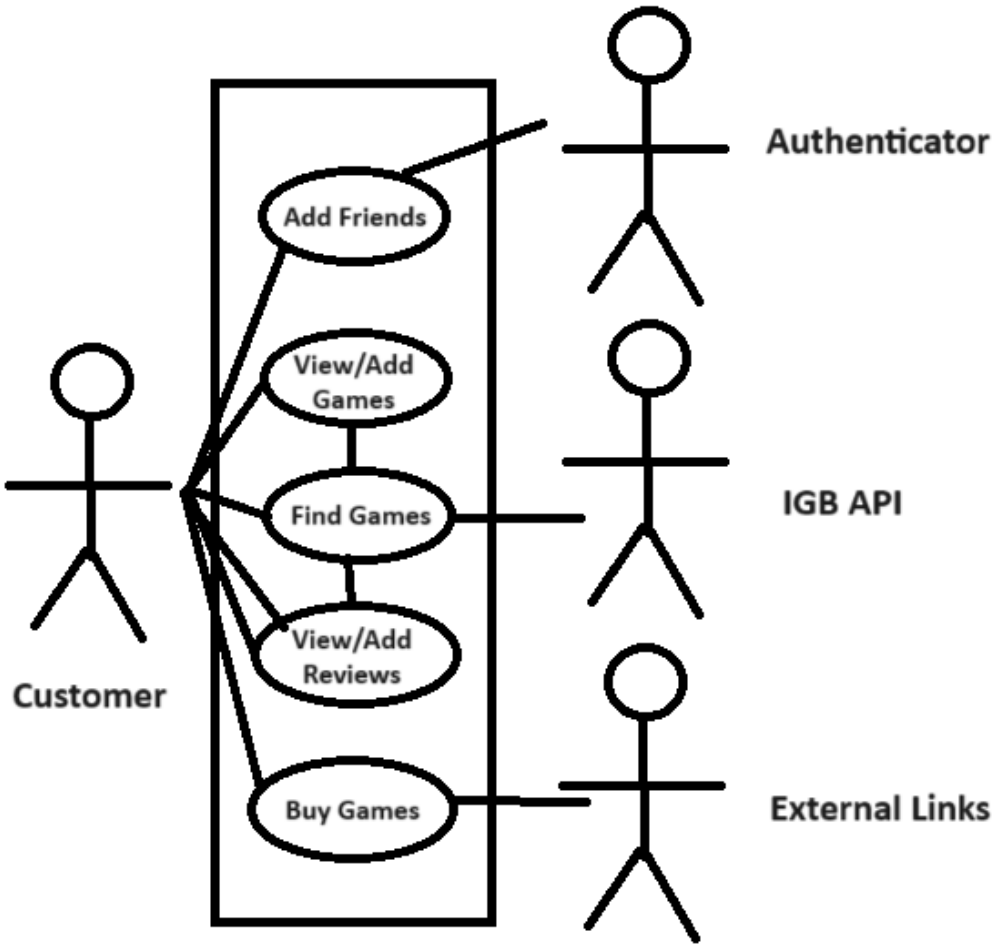
- I had the primary roll to work on the game page, logout page, and login/registration page. I used Daniel's work to integrate the IGDB API calls into the page's UI, HTML and CSS of the website as a whole, as well as creating the tables for the PostgreSQL database. I also had some roles in fixing bugs and HTML needs for the pages that needed attention. Furthermore, I had an ongoing task of creating a saved games table for users to save their favorite games into played, playing, and want to play categories. The implementation was not able to be implemented due to my sickness on the week of the project, so it was pulled from the presentation. Instead, I rebooted the website through Azure cloud services the day of the presentations so people may be able to use the online website for up to 30 days.

Gray

- I had the primary responsibility of handling the work on the login, logout, register, profile, and friends functionalities. The profile page would display the information of the user, the friends of the user, and the saved, favorite, and followed games of the user. I used a bootstrap library to display all of the cards for this. The friends functionality ended up being a later implemented function, as I was having trouble linking the user and friends database. I also maintained the

databases for users and friends, and created API calls for most of the original pages, including discover, profile, login, logout, and register. The saved games and favorite games functionality were not implemented due to extenuating circumstances listed above, but were something that we had fully planned out. I also spent a great deal of time ensuring that we followed directory structure guidelines and fixing varying issues from merges that deleted most of my initial codebase two weeks in.

VI. Use Case Diagram



VII. Test Results

We used Mocha and Chai to ensure that our login and register API routes were always working before the local host server was fully up and running so that at the least we knew the basics of our website were operating properly.

VIII. Deployment

~ Azure Cloudapp

The website was deployed using a Iaas protocol for the application through Azure cloud services under a student account that gives 750 hours, about 31 days, of free virtual machine usage. It is running through a Linux Virtual Machine which downloaded all the needed dependencies for the Docker image to be successfully ran through. We made a test run for the 13th lab using the instructions provided, then, on the day of presentations, it was set up again using the same configurations under http://playpal.eastus.cloudapp.azure.com:3000 .