

Traveling Merchant

Sprint One Planning Document

Team 10

Victoire Beaufils, Nicholas Dullam, Sabrina Eichenberger, Drew Sittley

Sprint Overview

This sprint is primarily focused on creating a workable environment for both the frontend, and backend development. This includes setting up a React.js client, including pages for login and signup, general components like the navbar and game gallery for the home pages, buyer listings, settings, and reviews. Alongside the UI/UX, we will create basic integration with Stripe for payment processing and saving payment methods, and incorporate our own API through our backend server. For the backend server, our intention is to implement the class design, including schemas/models for MongoDB, functions serving as controllers, and routes to access the controllers through our API using Express.js. This will all be hosted on a staging server through Heroku Web Services. And throughout both the frontend and the backend, we will document our components/API through our GitHub repository.

SCRUM Master: Sabrina Eichenberger

Meeting Times:

4:30pm, Wednesday (TA)

8:00pm, Thursday

10:00am, Saturday

Include risks and challenges:

Efficiently delegating tasks and user stories between the frontend and backend development.

Learning how to integrate MongoDB, Mongoose, Express JS, and Node JS in the backend for the members that are new to each of the things listed.

Current Sprint Detail

1. As a user, I would like to access a website to trade my game items and advance my gaming experience.

#	Description	Estimated time	Owner
1	Responsive Figma mockup for <i>Home</i> , <i>Login</i> , <i>Sign up</i> , and <i>my Account Information</i> page	1 Hr	Victoire
2	Navigation bar (Navbar)	1 Hrs	Victoire
3	Search bar with filtering	2 Hrs	Victoire
3	Log in and 3-step Sign up pages with game & game asset preference selection	4 Hrs	Victoire
4	Game gallery in <i>Home</i> page	1 Hr	Victoire
5	Footer	1 Hr	Victoire / Drew
6	<i>Listing</i> page	1 Hr	Nich
7	<i>Checkout</i> page	1 Hr	Victoire / Drew / Nich

- Acceptance Criteria:
 - Given that the Figma mockup is implemented correctly, the *Home*, *Login*, *Sign up* and *Account Information* pages should be responsive, and the pages and their components should render properly on all devices, without needing to scroll horizontally to access content
 - Given that the navigation bar (Navbar) is implemented correctly, it should host links to the *Login*, *Sign up*, **or** a user's profile picture followed by a down arrow dropdown which will have a link to the *My Account Information* page (among other links)
Note: Log in and Sign up will only be displayed if a user hasn't logged in.

- Given that the search bar works correctly, any user should be able to formulate queries for a game item or filter their query by games.
- Given that the listing page works properly, an item page should contain an item image, category (most likely in the form of an icon), price, quantity dropdown, purchase button.
- Given that the checkout page is functional, a user will be redirected to the checkout page upon clicking on "Purchase". Their Stripe and account information will be shown on the page, and the user will be prompted to click on the "Place your order", upon which the payment will be processed.

2. As a user, I would like to easily be able to create an account, provide my information and have it be secure.

#	Description	Estimated time	Owner
1	Server & Database setup	10 Hr	Sabrina
2	Server & Database setup	6 Hr	Nich
3	Integration of hashes and salts for secure password storage	6 Hr	Sabrina

- Acceptance Criteria
 - Given that the database is correctly implemented, a function should be able to access the database to add, edit, and create tables.
 - Given that the server is correctly implemented, a user should be able to access the server locally through HTTPS
 - Given that the passwords are correctly hashed and salted, when a function accesses the passwords in the database, the password should not be plain text.
 - Given that the user schema is properly implemented, a user should be able to make a request to create an account and have it be stored

3. As a user, I would like to modify my account information to easily add game credentials and payment information. (Stripe)

#	Description	Estimated time	Owner
1	Figma Mockup of settings, user information	1 Hr	Victoire
2	Function to store and retrieve view history based on Token	2 Hr	Sabrina
3	Settings page with Account info, favorites items, viewing history	3 Hr	Drew/Victoire
4	MongoDB/Mongoose additional routes set up for user information modification	4 Hr	Sabrina
5	MongoDB/Mongoose additional routes set up for user information modification	2 Hr	Nich
6	Stripe API Implementation	2 Hr	Nich

- Acceptance Criteria
 - Given that the settings page is correctly implemented, a user should be able to easily access and manage changes to their game credentials and payment information.
 - Given that the server routes are correctly implemented, a user's request to make changes to their game credentials should update their documents in the database.
 - Given that the Stripe API is correctly implemented, a user's request to change their payment method should update their information through Stripe's services and webhooks.

4. As a user, I would like to be able to see and leave reviews

#	Description	Estimated time	Owner
1	Figma Mockup of Seller Info page (with current listings and reviews)	1 Hr	Victoire
2	Seller review section & buyer-seller satisfaction rate calculation	1 Hr	Drew
3	Settings page	1 Hr	Drew

4	List of all <u>reviews</u> a <u>user</u> has published	1 Hr	Nich
5	MongoDB/Mongoose additional routes and model/schema set up for <u>reviews</u> , function to verify purchase history before approving <u>reviews</u>	4 Hr	Sabrina
5	MongoDB/Mongoose additional routes and model/schema set up for <u>reviews</u> , function to verify purchase history before approving <u>reviews</u>	2 Hr	Nich

- Acceptance Criteria:
 - Given that the seller info page is implemented correctly, a user should be able to see reviews that are left for a given product
 - Given that the routes are correctly implemented, user requests to add reviews should successfully create a review document in the database
 - Given that the routes are correctly implemented, user requests for reviews of a given product will be sourced from our database and returned to the client.

5. As a user, I would like to see my viewed history

#	Description	Estimated time	Owner
1	Addition of JSON Web Tokens	2 Hr	Sabrina
2	MongoDB/Mongoose additional routes and model/schema set up for <u>views</u>	2 Hr	Nich

- Acceptance criteria:
 - Given that JSON Web Tokens are properly implemented, a user should be able to see their viewed history based on history associated with the user's token
 - Given that Tokens are properly implemented, a user who is not logged-in and does not have cookies enabled should not see a viewing history
 - Given that routes are correctly implemented, a user viewing a product page should generate a view within our database for a given product.

6. As a user, I would like to customize my website appearance

#	Description	Estimated time	Owner
1	Dark Mode	1 Hr	Drew
2	Filter for game preference	1 Hr	Drew

- Acceptance criteria:
 - Given that dark mode is properly implemented, a user should be able to switch between modes
 - Given that dark mode is correctly implemented, a user's changes to their viewing mode should last between sessions
 - Given that filtering is properly implemented, a user's recommended should reflect user preference in the database

7. As a user, I would like to see previous transactions

#	Description	Estimated time	Owner
1	Function to look up user purchase history	2 Hr	Sabrina
2	MongoDB/Mongoose additional routes and model/schema set up for <u>orders</u>	3 Hr	Nich

- Acceptance criteria:
 - Given that the route is properly implemented, the front end should be able to request for a particular user's previous transactions
 - Given that the routes are properly implemented, users' conducted orders should successfully be reflected in the database.
 - Given that the routes are properly implemented, requests for orders should successfully be facilitated through the Stripe API for transactions.

8. As a user, I would like to be able to browse through games.

#	Description	Estimated time	Owner
---	-------------	----------------	-------

1	Filter for search	1 hr	Drew
2	MongoDB/Mongoose additional routes and model/schema set up for <u>games</u>	2 hr	Drew

- Acceptance criteria:
 - Given that the filter is properly implemented, a user should be able to select filters to narrow their search
 - Given that the routes are correctly implemented, a user should be able to request the game categories and reflect the database
 - Given that the schemas are correctly implemented, a user should be able to see specific game attributes, including product types.

9. As a user, I would like to favorite items to come back to later

#	Description	Estimated time	Owner
1	MongoDB/Mongoose additional routes and model/schema set up for <u>favorites</u>	2 Hr	Sabrina

- Acceptance criteria:
 - Given that the favorite schema is properly implemented, a user should be able to favorite an item and that favorite be added to their account information
 - Given that the user's profile page is properly implemented, a user should be able to accessed their favorited products in their client.
 - Given that the seller's profile page is properly implemented, a seller should be able to see the number of favorites on their given product.

10. As an administrator, I would like to be able to ban accounts.

#	Description	Estimated time	Owner
1	Adding ability to modify database from website	2 Hr	Nich/Drew

- Acceptance criteria:
 - Given that the route is properly implemented, an administrator should be able to view the list of users
 - Given that the route is properly implemented, an administrator should be able to ban a particular member
 - Given that the server is correctly implemented, banning a user should successfully disable their ability to use specific services within our web app.

Backlog

Functional

- As a user I would like a platform that is stable without glitches or bugs.
- As a user, I would like for background operations to not limit my experience and slow down my transactions and game asset browsing.
- As a buyer, I would like to be protected from fraudulent transactions when buying services or in-game items.
- As a user, I would like to access the platform on all devices and to browse a responsive website.
- As a user, I would like the option to follow a seller to be notified of their new postings
- As a seller, I would like to receive notifications when customers message me about a product so that I can proceed with transactions as quickly and effectively as possible.
- As a seller, I would like to be rewarded for my continuous use of the platform through having my products be showcased more often than the ones of new sellers.
- As an administrator, I would like to be able to ban accounts.
- As an administrator, I would like to be able to view transactions and nullify scams.
- As a user, I would like to be able to find services for games easily.
- As a user, I would like recommendations based on viewed history and transactions
- As a user, I would like to see what others are selling/buying
- As a user, I would like to see similar items to what I'm viewing
- As a user, I would like to see a progress graph over time of a particular item/service's price
- As an administrator, I would like to be able to remove reviews
- As an administrator, I would like a language filter
- As an administrator, I would like to be able to verify reviews
- As a seller, I would like to be able to have a bid or static price (if time allows)
- As a user, I would like to be able to change my cookie settings (if time allows)

Non-functional

- As a developer, I would like to be able to handle 10,000 requests simultaneously to access the site.
- As a developer, I would like third-party access times to be less than 500 milliseconds.

- As a user, I would like site load time to be less than 1000 milliseconds.
- As a developer, I would like to be able to store 50,000 user accounts.
- As a user, I would like the site to be running 24 hours a day.
- As a developer, I would like database access to be less than 500ms.