

# Traveling Merchant

---

## Sprint Two Planning Document

Team 10

Victoire Beaufils, Nicholas Dullam, Sabrina Eichenberger, Drew Sittley

# Sprint Overview

---

The primary focus of this sprint is the addition of features to make our app safer and more appealing by the addition of features. We plan to give administrators more control over reviews and transactions to prevent scammers and fraudulent transactions. This will be done by the addition of review verification and deletion by administrators and the nullification of transactions by administrators. We also plan to make our app more appealing by adding a language filter and user oriented features like recommendations and user levels. We also plan to include real-time notifications by the use of sockets in the server. This will allow for faster communication between sellers and their customers, and it will allow users to be notified of new postings by sellers they follow. By the end of this sprint, we plan to have a website that caters to both buyers and sellers while also being securely monitored by administrators.

**SCRUM Master:** Sabrina Eichenberger

**Meeting Times:**

4:30pm, Wednesday (TA)

8:00pm, Thursday

10:00am, Saturday

**Include risks and challenges:**

Sockets are a bit tricky to understand, so the implementation of real-time messages might take longer than expected. Another problem arises with filtering and searches taking time, both to write the code and the actual loading time, which will impact user experience on the site. We also wish to include an amazon-like recommendation carousel, but having the recommended be accurate may prove to be difficult.

## Current Sprint Detail

---

1. As a user, I would like the option to follow a seller to be notified of their new postings.

#	Description	Estimated Time	Owner
1	Addition to the create product route to create notifications for users following the seller	1 hr	Sabrina
2	Addition of middleware to get notifications for users who just logged in	2 hr	Sabrina
3	Add a socket emit to notify users in real time who are currently browsing the site (handler on front-end and emit on back-end)	3 hr	Sabrina
4	Create a follow button, a tab with a list of following and one with a list of followers for every profile	2 hr	Victoire
5	Create notification bell icon in Navbar with dropdown and a summary of notifications and activity	2 hr	Victoire
6	Send emails to users when there are new postings of people they follow	2 hr	Victoire
7	Test notifications on front-end	2 hr	Victoire

- **Acceptance Criteria**

- Given that socketIO is implemented correctly on the back-end, the server should be able to handle messages emitted from the front-end and emit the message to the destination socket.
- Given that socketIO is implemented correctly on the front-end, users should be able to receive real-time updates when a seller creates a new product

- Given that socketIO is implemented correctly on the front-end, sellers should be able to receive real-time updates when an user follows them
- Given that email notifications are implemented correctly, a user should receive emails notifying them of a new listing from a seller they follow

**2. 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.**

#	Description	Estimated Time	Owner
1	Addition of socketIO on the back-end	1 hr	Sabrina
2	Addition of emit events to notify users in real-time (back-end)	1 hr	Sabrina
3	Addition of emit events to notify users in real-time (front-end)	2 hr	Sabrina/Nich
4	Notify users in real time who are currently browsing the site (handler on front-end for socket events)	3 hr	Sabrina/Nich
5	Set up email notifications for customer messages	2 hr	Victoire
6	Create messaging frontend	6 hr	Victoire/Drew
7	Test chatting on front-end	4 hr	Victoire

● **Acceptance Criteria**

- Given that socketIO is implemented correctly on the back-end, the server should be able to get unread notifications and notify the user when they revisit the site.
- Given that socketIO is implemented correctly on the back-end, the server should be able to authenticate users before they are allowed to message other users.
- Given that socketIO is implemented correctly on the front-end, users should be able to receive real-time updates when another user messages them

- Given that socketIO is implemented correctly on the front-end, the browser should be able to handle messages from the socket without errors being thrown.
- Given that a message was sent to a seller, said seller should receive email notifications every time a message is sent to them.
- Given that messaging is implemented correctly, a seller and buyer should be able to correspond in real time.

**3. As an administrator, I would like to be able to view transactions and nullify scams.**

#	Description	Estimated Time	Owner
1	Add client admin panel page to search for transactions with pagination	30 min	Sabrina
2	Add additional parameters to order controller and frontend to account for pagination	30 min	Nich
3	Create models for refresh tokens, alongside corresponding controllers and auth routes	2 hr	Nich
4	Adjust auth middleware to create refresh tokens on access token expiry	1 hr	Nich
5	Create models and aggregate controllers for logins to record IP addresses	2 hr	Sabrina
6	Add additional user controllers to ban by IP address	2 hr	Sabrina

7	Add page to client admin panel to view/search transactions	2 hr	Drew
8	Test the deletion of orders/transactions	1 hr	Drew

- **Acceptance Criteria**

- Given that the client admin panel order page is implemented correctly, an administrator should be able to visit a page to see transactions based on a search filter
- Given that the client admin panel order page is implemented correctly, results should be limited by the variable amount with pagination
- Given that the login controllers are implemented correctly, a user should not be able to create an account from an IP-address of a banned account.
- Given that the order controllers are implemented correctly, an admin should be able to delete/nullify orders

#### 4. As a user, I would like to be able to find services for games easily.

#	Description	Estimated Time	Owner
1	Add product sorting by the unit_price within the product search client page	1 hr	Victoire
2	Add search indices for both game and product models with search controllers and routes	2 hr	Nich
3	Add more parameters for the search controller of product, and have that reflected in client query string	1 hr	Sabrina
4	Add pagination to the client product search,	1 hr	Nich

	limiting the number of products on the page with adjustable amounts		
5	Testing search routes & client product and game searches respectively	2 hr	Drew

- **Acceptance Criteria**

- Given that our sort button works correctly, sorting by unit\_price should sort items by ascending and descending order
- Given that the search indices are properly implemented, searching for games and products will not require an exact input of the result's attributes
- Given that the pagination is implemented correctly, users should be able to move between pages of product listings with an adjustable amount per page
- Given that the search parameters are implemented correctly, refreshing the client search should lead to the same search results, page, and sorting

**5. As a user, I would like recommendations based on viewed history and transactions.**

#	Description	Estimated Time	Owner
1	Keep track of games/product-types viewed/bought by a user	2 hr	Sabrina
2	Route and controller to get most common games/product-types viewed/bought by a user	2 hr	Sabrina
3	Route and controller to get products by most common games/product-types	1 hr	Sabrina
4	Create a recommended product gallery within the client	2 hr	Sabrina

	homepage for a given user		
5	Test by visiting product pages and seeing if recommendations match what I was viewing	1 hr	Sabrina

- **Acceptance Criteria**

- Given that the client page is implemented correctly, a user should be able to see a bar of recommended products
- Given that the backend is implemented correctly, a user's view history and transactions should be reflected in the database.
- Given that the routes and controllers implemented correctly, a user should be able to get a list of products that are similar to what a user has viewed/purchases

## 6. As a user, I would like to see similar items to what I'm viewing

#	Description	Estimated Time	Owner
1	Calculate an index of similarity between two objects	1 hr	Drew
2	Make client product bar listing similar items to that of the current product page	1 hr	Victoire/Drew
3	Create a route and controller to get products by similar game/product-type	30 min	Sabrina
4	Test routes by viewing various products and gauging similarity	1 hr	Sabrina

- **Acceptance Criteria**

- Given that the client product page is implemented correctly, a user should be able to see a "similar items" bar
- Given that the route and controller are implemented correctly, a user should be able to get a list of items similar to the current product page.
- Given that the routes are set up correctly, the recommended products should be displayed dynamically and visible to the user.



**7. As a buyer, I would like to be protected from fraudulent transactions when buying services or in-game items.**

#	Description	Estimated Time	Owner
1	Create cron jobs for stripe transactions on backend server, fulfilling orders at auto-complete date	2 hr	Nich/Drew
2	Setup a failsafe function on server mount to ensure auto-complete orders are fulfilled	2 hr	Nich
3	Create client page for viewing orders, customize for buyer and sellers with deliver/cancel functions alongside confirm/deny functions	4 hr	Nich/Drew
4	Setup stripe transfers between the payment request for the fulfilled order and the requesting account	2 hr	Nich
5	Add "Become a Seller" function to the profile page, alongside a "Continue Onboarding" function if their details have not been entered, and have that reflected in the database	2 hr	Nich
6	Add payment method saving opt-in during the checkout. List existing payment methods and allow them to be applied to the order	2 hr	Nich
7	Add stripe webhooks to the backend server, with routes for onboarding status updates to the user profile	2 hr	Nich
8	Test stripe integrations to ensure order transfers are fulfilled,	2 hr	Drew

	accounts and onboardings are completed through the client profile page		
--	--	--	--

- **Acceptance Criteria**

- Given that the chron jobs are implemented correctly, a seller's order should be marked as complete and initiate a transfer
- Given that the failsafe function is implemented correctly, a seller's order that is past the auto-complete date should be handled on server-mount, marking it as complete and initiating a transfer
- Given that the client page for orders is implemented correctly, sellers and buyers should be able to mark the order as delivered and confirm or deny the delivery, viewing the order information
- Given that the stripe transfers are implemented correctly, a seller's portion of a payment request should be transferred to their stripe account
- Given that the stripe webhooks are implemented correctly, a seller's onboarding status should be reflected in the database (for accounts on the live server)
- Given that the seller creation and onboarding functions of the client page are implemented correctly, users should be able to create their stripe account and fulfill their onboarding through their profile page

## 8. As an administrator, I would like to be able to verify reviews.

#	Description	Estimated Time	Owner
1	Create a client admin panel reviews page, listing reviews with the ability to verify reviews	3 hr	Drew/Victoire
2	Add additional parameters to review controller and frontend to account for pagination	30 min	Nich
3	Route and controller to verify the given review	30 min	Sabrina
4	Test the verification of reviews through the client admin panel	30 min	Sabrina

- **Acceptance Criteria**

- Given that the client admin panel reviews page is implemented correctly, reviews should be listed, limited by the variable amount with pagination, with the ability to remove select reviews.
- Given that the route to verify reviews is implemented correctly, the administrator should be able to access and view the reviews.
- Given that the routes are implemented correctly and the admin panel is fully functioning, the administrator should be able to verify a review.

**9. As an administrator, I would like to be able to remove reviews.**

#	Description	Estimated Time	Owner
1	Add removal function to client admin panel reviews page	1 hr	Drew/Victoire
2	Middleware to verify admin status before proceeding to the auth-required route	1 hr	Sabrina
3	Test the removal of reviews through the client admin panel	1 hr	Drew

- **Acceptance Criteria**

- Given that the client admin panel reviews page is implemented correctly, reviews should be listed, limited by the variable amount with pagination.
- Given that the middleware to verify an administrator's status is implemented correctly, the administrator should be able to access admin-auth required routes.
- Given that the admin panel is implemented correctly, the administrator should be able to remove select reviews if necessary.

**10. As an administrator, I would like a language filter**

#	Description	Estimated Time	Owner
1	Create an input form within the client admin panel for a list of	3 hr	Drew/Victoire

	bad keywords to hide on the platform		
2	Creation of the models, routes, and controllers for bad words within the server	2 hr	Nich
3	Apply this validation filter (scanning for bad words) to reviews and products created, and remove reviews if keyword matched	3 hr	Victoire
4	Test the language filter for adding new bad keywords and removal of products/reviews for matched keywords	1 hr	Drew

- **Acceptance Criteria**

- Given that the input form within the client admin panel is implemented correctly, the form should list current keywords.
- Given that the client panel is implemented correctly, a user should be able to remove listed keywords, as well as add additional keywords.
- Given that the bad words controllers, model and routes are implemented correctly, any adjustments to bad words through the frontend client should be reflected in the database.

# Backlog

---

## Functional

1. As a user I would like a platform that is stable without glitches or bugs.
2. As a user, I would like for background operations to not limit my experience and slow down my transactions and game asset browsing.
3. As a user, I would like to access the platform on all devices and to browse a responsive website.
4. 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.
5. As a user, I would like to see what others are selling/buying
6. As a user, I would like to see a progress graph over time of a particular item/service's price
7. As a seller, I would like to be able to have a bid or static price (if time allows)
8. As a user, I would like to be able to change my cookie settings (if time allows)

## Non-functional

1. As a developer, I would like to be able to handle 10,000 requests simultaneously to access the site.
2. As a developer, I would like third-party access times to be less than 500 milliseconds.
3. As a user, I would like site load time to be less than 1000 milliseconds.
4. As a developer, I would like to be able to store 50,000 user accounts.
5. As a user, I would like the site to be running 24 hours a day.
6. As a developer, I would like database access to be less than 500ms.