

SW Engineering CSC648/848 Summer 2021

Jose's Angels (Angelapp.gq)

Team 03

Name	Email	Roles
Mitchel Baker	mbaker3@mail.sfsu.edu	Team lead
Krina Bharatbhai Patel	kpatel11@sfsu.edu	Frontend lead
Charmaine Eusebio	ceusebio1@mail.sfsu.edu	Frontend engineer
Rowena Elaine Echevarria	rechevarria@mail.sfsu.edu	Frontend engineer
Michael Schroeder	mschroeder@mail.sfsu.edu	Backend lead, Github master
Kenneth N Chuson	kchuson@mail.sfsu.edu	Backend engineer
Jamie Dominic Walker	jwalker5@mail.sfsu.edu	Backend engineer

Milestone 1

June 22, 2021

History Table

Version	Date	Notes
M1V1	06/22/2021	

Table of Contents

1. Executive Summary.....	3
2. Main Use Cases	4
3. List of main data items and entities.....	9
4. Initial list of functional requirements	14
5. List of non-functional requirements	23
6. Competitive Analysis.....	31
7. High-level System Architecture and Technologies Used	35
8. Checklist	36
9. Team Contributions	37

1. Executive summary

Last year the world was shocked as we transitioned from the idea of a global pandemic being an outdated science fiction trope, to full blown reality. The various nations of the world urged social distancing and staying at home till the crisis could pass. The challenge is that as great as it is to stay home from work or school all day long, eventually you get bored and want to do some shopping.

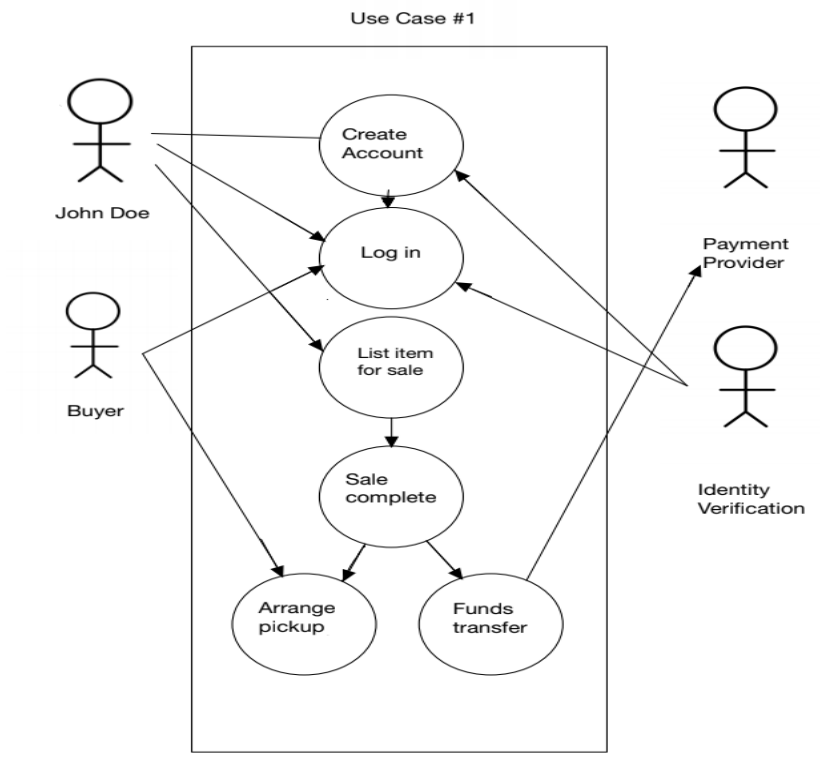
Gone are the days of waking up early on the weekend and driving around town and looking for the best bargains at the local yard sales. These days it is all about finding what you want when you want it. The issue is that the market is flooded with half-baked ideas that have evolved to make ends meet. Facebook was initially intended as a way for college students to keep in touch. eBay was designed as an auction site 25 years ago, and Amazon started as a bookstore. There is not a major digital marketplace that was designed for the current state of the world.

Fortunately for you (and introverts all over the world), we at Jose's Angels are developing the next generation digital marketplace that is designed for the world you are living in now. It is this innovative mindset that sets our team apart from the competition. We focus on maximizing the user experience with industry leading best practices so customers can shop safely and securely. We will offer a simple clean digital storefront that makes the focus all about buying and selling exactly what you want or need. We at Jose's Angels understand that buying or selling for the first time from another person can create unneeded anxieties and we want to ensure sellers have all the information they need to make safe and informed sales. For example, maybe you have a brand-new PlayStation 5 you want to sell but are not sure how much it goes for. No need to price check on Amazon or Walmart, we include a price checker for you when listing an item to cut down on the amount of research you must do.

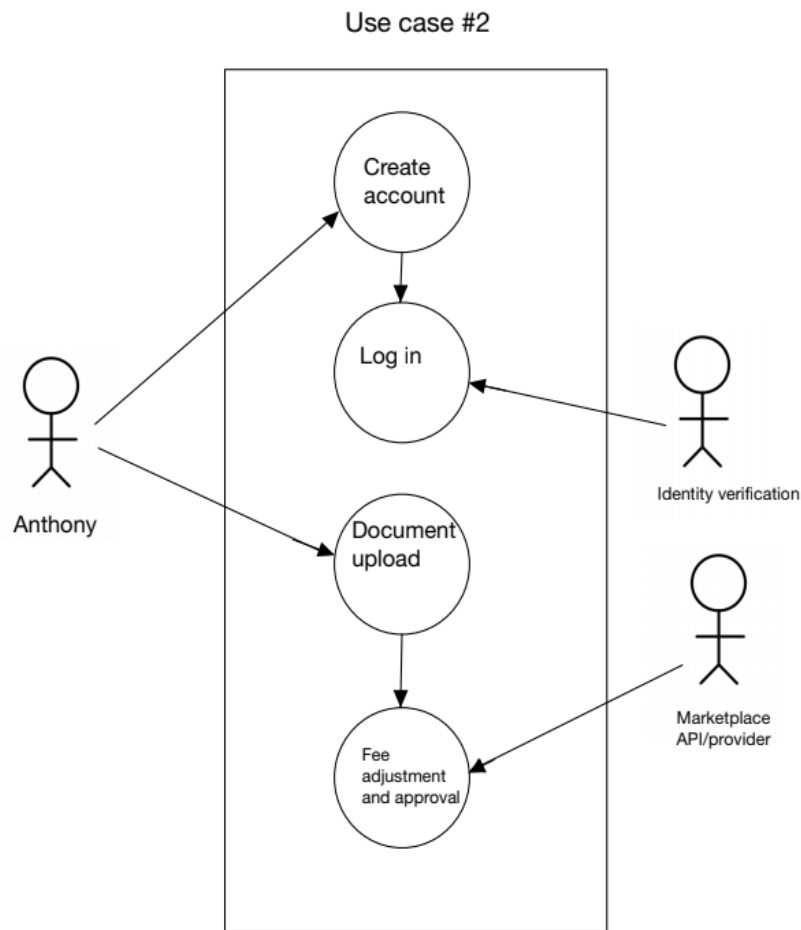
Here at Jose's Angels we have an experienced group of buyers and sellers all working hard to bring you the best possible experience. So, if you want to learn how to upgrade your yard sale to a streamlined marketplace, let us show you how.

2. Main use cases

Title	Easy Onboarding Experience
Actor	John
Description	<p>John is moving into San Francisco. He is in the process of selling some of his old belongings which range from computers, gaming devices, to couches and other kitchen appliances. John hopes to find buyers who offer a reasonable price. As a result, he will be better positioned to finance his upcoming transition to a new city. John has been active with eBay and Facebook marketplace for a while, but he has not had any success with potential buyers. John is moving in the next 3 days. It is no big deal if he cannot sell the small items such as the computers, gaming devices, kitchen appliances since he can always include them in his moving truck to sell in San Francisco. On the other hand, John must sell the couch because his other options are taking it to the dump or donating to goodwill which will not turn him a profit. John is looking to find a buyer so he can make a quick buck, and since he is leaving in 3 days, as fast as possible.</p>

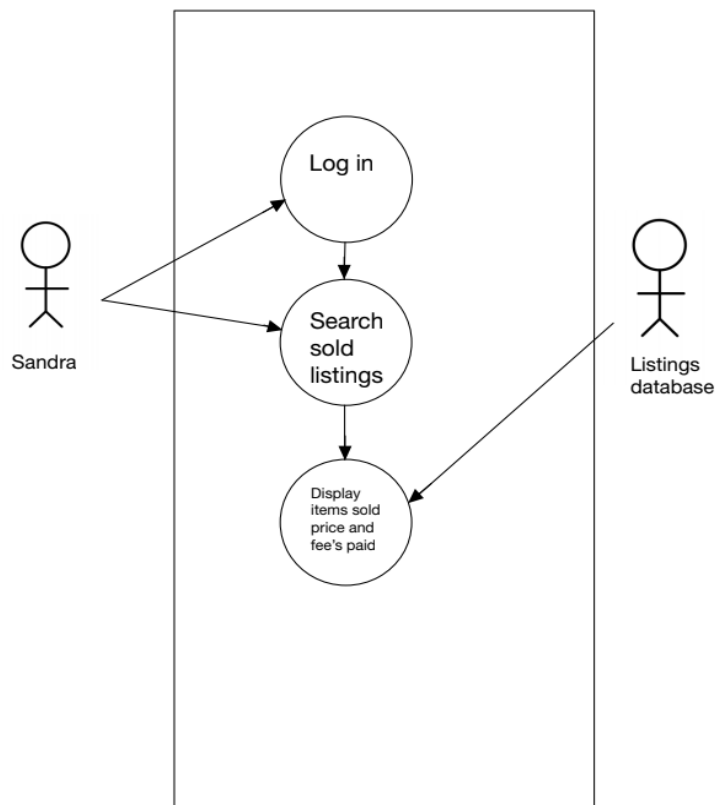


Title	Storefront Features with less fees
Actor	Anthony
Description	Anthony wishes to sell items for the long term. He is looking to provide reasonable deals through his brick-and-mortar location, but the storefront subscription prices are too much for him to start out. Anthony wants to prove he is a small business owner so he can access storefront features and lower fees on our website, instead of a competitor.



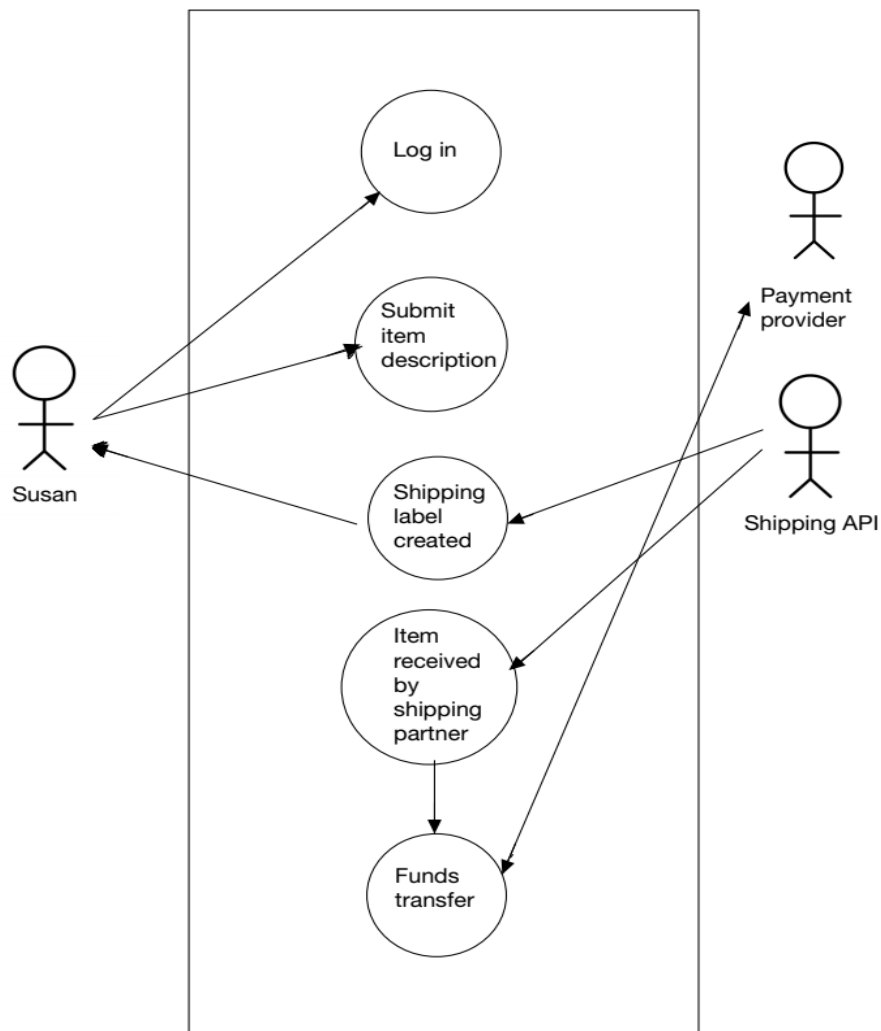
Title	Price Matching Feature
Actor	Sandra
Description	Sandra, a buyer, has the old iPhone 10 model, but they are busy with work and do not have time to go to a brick-and-mortar store. As a result, Sandra needs an easy-to-use website which allows her to get the new iPhone 12 and avoid scalper prices. She checks the price of a new iPhone 12 on Facebook Marketplace, Amazon, and eBay, but all these prices are out of their price range. Sandra can check our website to compare recently sold listings and fees for a smart buying decision.

Use case #3

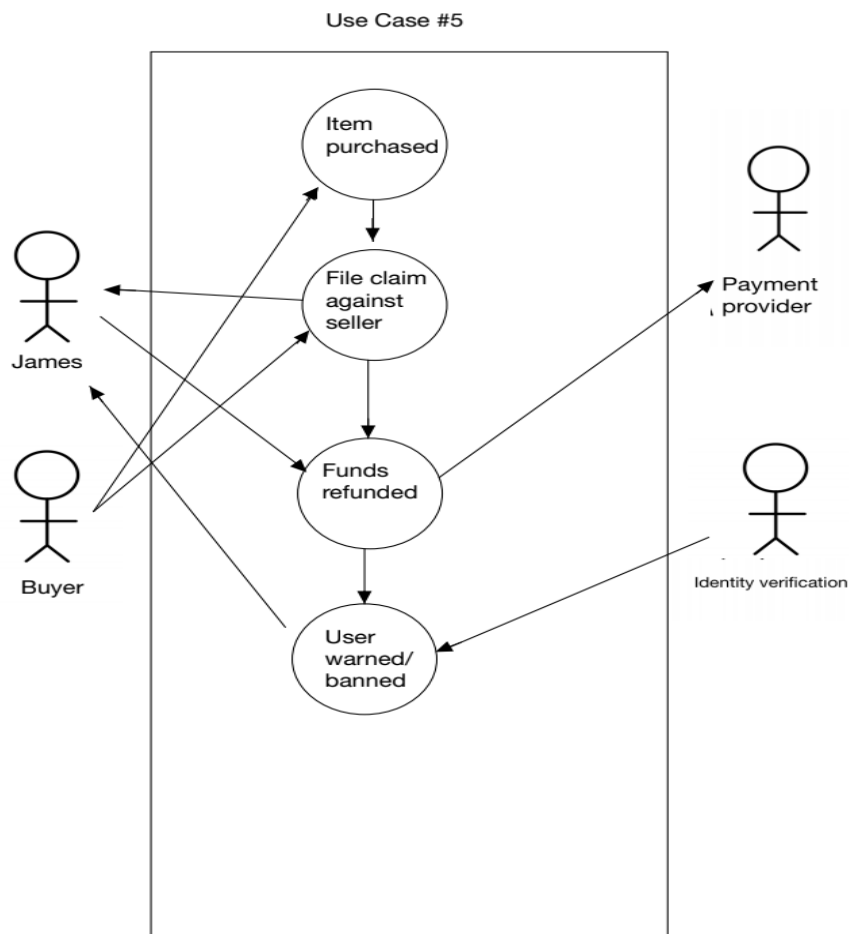


Title	Shipping available with extra benefits
Actor	Susan
Description	Susan is doing some spring cleaning in their home and is too busy with work to be able to spend time devoted to creating individual listings for each of these items. This allows the consignment service to provide the seller with the option of passing over the creation of listings and prices to us.

Use Case #4



Title	Flagging system to avoid fraud deals/scams
Actor	James
Description	<p>James is in the business of fabricating fake PS4 controllers, which he then markets to people as genuine. However, his fabricated replicas are only plastic shells which means they have no functionality whatsoever. James creates an account on our site and begins creating listings for these fake PS4 controllers.</p> <p>Solution: If buyers begin to leave bad reviews for these fake PS4 controllers, they can report James (seller) to us, and we can flag the item as a scam so the listing can be removed.</p>



3. List of main data items and entities

➤ Data categories

1. API Data

- 1) google map locations and places.
- 2) items from e-commerce websites (Amazon, eBay, etc.) for item comparisons
- 3) ads that match customer wanted items.
- 4) UPS developer kit API
- 5) Our own Node API server which will return requested data in JSON format to the react client.
- 6) The Node API server will load user data, post data, messaging data, customer reviews and ratings, seller reviews, and for storing shopping cart data securely.

2. Client

- 1) List of users
- 2) User account information (both customers and sellers)
- 3) user login (active session)
- 4) user logout (inactive session)
- 5) user first name
- 6) user last name
- 7) user username
- 8) user email
- 9) user phone number
- 10) user age
- 11) user password
- 12) user messages
- 13) user street, city, state, zip
- 14) user voice/video calls
- 15) User ID of other party
- 16) Item ID of item requested.
- 17) user meetups
- 18) user schedule

- 19) user payment information
- 20) Cc number, expiration date, 3dig code, zip

3. Customers (buyers)

- 1) customer purchase items
- 2) customer rate items
- 3) customer review items
- 4) customer refund items
- 5) customer report items
- 6) customer return items (not refund)
- 7) customer shopping cart items (private)

4. Sellers

- 1) sellers sell items.
- 2) user driver's license picture/ID or other form of identification to be verified as a seller.
- 3) location of seller
- 4) seller posts (public)
- 5) Post information: name, description, image, pickup/delivery option
- 6) If post is an auction, buy it now, or best offer.
- 7) seller out of stock items
- 8) seller illegal items

5. Sidenav

- 1) hamburger component, set to true/false to display/not display.
- 2) Navigation for Home, Contact, About, Login, Logout
- 3) Shopping Cart:
- 4) Item
 - Item name
 - Item price
 - Item quantity
 - Item size
 - Item selection type (depends on the item)

- Item brand name
- Item year
- Item used/new.

6. Filtering for item searching

- 1) Search bar for querying the database for posts with requested terms.
- 2) Min/Max prices for a search
- 3) Location filtering
- 4) Pickup or delivery
- 5) A through Z alphabetical order for items feature
- 6) Items which belong to the category's menu
- 7) Item's size

7. Menu

- 1) List of items
- 2) Categorical items
 - Categorical items are from customers' selections and this mostly works with API Data of the items from other e-commerce websites.
- 3) Item ID
- 4) Item seller's ID, to connect Item back to its Seller.
- 5) From seller's ID, can get seller name and other account info.
- 6) consignment data
 - Bar codes
 - Shipment information:
 - departing location
 - Destination
 - Time that shipment was sent out.
 - Estimated arrival time.
- 7) Item rating (5 star system)
- 8) Item reviews/comments/discussions
- 9) purchase items

→ topmost purchased items (10 max items)

10) shipping items

11) offer items

→ top offer items (10 max items)

12) similar items

→ top similar items (10 max items)

13) for sale items

→ top sale items (10 max items)

14) buy it now items

→ top buy it now items (10 max items)

15) top daily deals items

→ perhaps time data type since deals exist for 24 hours.

➤ **oop design / functional / components (Entities)**

- 1) React component for Store (Browse, Shop, Explore?)
- 2) component for User login/logout
- 3) component for User account details
- 4) component for Checkout of items in shopping cart
- 5) component for Checkout receipt information
 - By default, buyers will not have their shipment information added. If they choose delivery, then they must be prompted for their address information.
 - When a user signs up to be a seller, they must add their location as a requirement to verify that it is legitimate.
- 6) component for checkout final summary, payment, and invoice
 - payment option, (if user is buying for the first time, then have them update their payment option, if they have opted to remember this information, then payment will already be there)
 - invoice of pickup time/delivery time
 - expected time of arrival.
 - name of seller, location to meet them at, and their contact info.
 - final total to be paid.
- 7) component for which items belong to the categories.

4. Initial list of functional requirements

Auction

- 1) Create an auction form where sellers can set up the settings for their auction.
 - a. They should be able to set the average starting bid, the total duration of the auction (1, 3, 5, 7 days), and the time when the auction begins.
- 2) Sellers can bid on the items as many times as possible.
- 3) Create auction, buy it now, or best offer options for listing.
 - a. More options for sellers to earn money.
- 4) When an auction is posted for bidding, users should be able to see the current price it is set at, the amount of time remaining in the auction, and any bids from others.
 - a. To make auctions real time, we will have to set up some sort of WebSocket lobby where everyone partaking in the auction has access to the auction's stats.
- 5) If a user wins an auction, they should be notified immediately, either by text or by email. (email seems more doable at this point, texts with Twilio are added cost)
- 6) Also, after a user wins an auction, if their browser is open, they should then be notified that they have won and then have the option of redirecting to a new page where they can finalize their payment with the seller.

Marketplace

- 7) Category menu (beauty/cosmetics, home appliances, tech gadgets/computers, athletic tools, clothing. This will allow searching items faster, and to keep all the products organized.
- 8) Search bar for looking up items. A basic search bar that can search when the enter key is pressed, alongside a search button.
- 9) Advanced search bar for more detailed searches. A drop-down menu of more details including size, price range, color, etc.
- 10) Filtering for item searching, min/max price, location filtering (within 50 miles), type of shipping (pickup, delivery. This will help buyers and sellers to choose the type of delivery options.
- 11) Photo slider which has left/right arrows allowing the buyer to navigate through the pictures of an item posted by a seller. This will allow the buyer to see the details on the design of the product.

- 12) Have an option to slide through the posts or go to the next page.
- 13) Most recent
 - a. 50 most recently created posts to show up on the Home page. This will highlight the new products available at any given time.
- 14) Share link.
- 15) Display seller's contact information
- 16) Location of seller (google maps api)
- 17) Item selection style.
 - a. Some items have different styles, for example, a watch has a different selection of colors and they have the same price.
- 18) Allow users to edit schedule meetup times if a buyer and seller agree with each other.
- 19) Offer base price for automatic reject or accept.
- 20) Ads of similar items on product page
 - a. To attract more customers to buy products.
- 21) Shipping fees table
 - a. Breakdown of the shipping fees for time efficiency
- 22) Track package/delivery. This will allow users to confirm that they have received their order.
- 23) Display "out of stock" of empty selling items. This will let users know that the item is no longer available and have the option to be contacted by email when the item comes back in stock.
- 24) Keeping track of the same purchase item multiple times.
 - a. This can help prevent sold out items to avoid if one of the customers purchase multiple same items.
- 25) Flagging system for reporting suspicious items. User can either flag the item as inappropriate, a scam, or they can continue the process further by submitting a ticket to us for further review.
- 26) Share listing feature to be able to share the product with friends, through email, social media, and copied link.
- 27) Wedding registry for those who wish to create a Wishlist for their special day. This will allow users to add products from the app into a list that they want to share with others.
- 28) Recently viewed items bar with product listing icons on the homepage. Users can go back to items that they have recently viewed.
- 29) Order status: confirmed, processing, shipped, returned. This will allow the user to track the status of their order/s.

- a. The user should be able to get immediate updates on the status of their order. After they submit an order, they will be redirected to a page which notifies them of their order being confirmed. They should also be able to check their order history, and then click on specific orders to see its processing, shipping, or return details.
- 30) Must show the maximum time length shipping delivery item.
- 31) Must show the minimum time length shipping delivery item.
- 32) User monitoring so that we can display to the user the posts they recently interacted with. I.e., keep track of the IDs of the posts they click on, and before redirecting to another portion of the app, send a post request to the server which will add these items to the database for interaction later.
- 33) Show people who have the same purchase as you.
 - a. When a customer purchases an item, the feature will pop up the list of customers that have the same purchase and a customer will be able to message or call one of the customers about relating to the similar purchase item.
- 34) If the buyer's and seller's home location are closer around 1 or 2 miles, then the algorithm will hide the time length of the shipping delivery item.

Messaging

- 35) Contact seller directly in website or with email.

Buyers

- 36) Message seller button/form for inquiry of interest or any questions. This will allow seamless customer service between buyers and sellers.
- 37) Buyer agreement to be signed by every user intending to make a purchase on the app. The buyer will agree to conduct business according to laid out rules.
- 38) Purchase/order history for buyers to keep track of their past purchases.
- 39) When a buyer is interested in an item, add it to their shopping cart.
- 40) Shopping cart with number of items icon. The buyer will be able to view the total items in their cart and the total cost of the items.

- 41) Checkout interface. (receipt information first --> then subtotal information, display of estimated fees/tax, total → then input credit card payment --> then final invoice information about the order)
- 42) Users should be able to remove an item from their shopping cart, and the total list of items should be updated. User should also be able to return to the main shopping menu to look for other items if they are not done.
- 43) Cancel/modify order option. This will allow users to edit their orders in case they would like to add more items in their cart or cancel the order.
- 44) If a user is a buyer, they should be able to rate/star products to keep track of them.
- 45) Can choose delivery options.
- 46) Seller review page
- 47) When a user clicks on an item they would like to see more of, display not only the photo slider but also the ability of messaging/sending an inquiry to the seller.
- 48) Product review page
 - a. To provide insight into the functionality or state of the products bought.
- 49) Seller review page/ star rating
 - a. Proving seller's credibility
- 50) Buyers can also see the posts that they interacted with most recently. The post/product listing link will be highlighted a different color than those that have not been visited.
- 51) Product reviews on product listing page. The buyer will be able to submit a review for purchased items.
- 52) Favorite's list - the buyer will be able to add what they are interested into an ongoing list.
- 53) Wishlist. This will allow the buyer to save the items they would like to buy in the future.
- 54) Discount codes for frequent buyers. The codes will have expiration dates and will attract business by providing special discounts for potential buyers.
- 55) Coupon/promo code box for the buyer to enter the code and have the discount applied to their shopping cart.
- 56) Watch item feature/list. If a product is in an auction, the buyer will be able to keep track of the most recent updates to the item.
- 57) Buyers can have an award item.
 - a. If a buyer bought some lucky items, then the buyer will have one of the free award items.
- 58) Birthday promo code / random award item.

- a. if a buyer's today birthday, then the buyer will choose between a Promo code or an award item from a random item generated algorithm.
- 59) Generate a referral URL which existing users can send to their friends.
- 60) Users should be able to subscribe to a specific seller so that they can be notified if the seller adds new items, adds updates, or makes any changes to the current price of an item.
- 61) When a buyer purchased an item, it will show some top-rated items that are related to that buyer's purchase item.
- 62) Email order confirmation to both buyer and seller. The email will contain information of the product sold, time it was sold, and who bought it.

Sellers

- 63) Form for sellers who wish to create a new item.
 - a. Include item name, description, image, etc. Each item also needs their own respective ID which will be used everywhere in the project.
- 64) Sellers should be able to edit the description, title, and price of their items. This will allow the seller to advertise their products better.
- 65) Seller contract to be signed by all users who wish to sell. This will provide sellers with an outline of how to sell on the app and rules they must follow.
 - a. Seller listing fees for each product that is sold. This will include a percentage of each sale being taken as profit for the app.
 - b. Seller background checks for all users intending to sell on the app. This will help to fight against scammers potentially.
- 66) Sellers should be able to adjust the quantity of an item that is available for sale.
- 67) Sellers should be able to delete an item. This will allow sellers to delete any items they think that will not be able to be sold.
- 68) Ability to have multiple select delete items.
- 69) Ability to have multiple select delete posts.
- 70) Under the user's profile information, if they are a seller, they should be able to keep track of all the items they have posted.

- 71) For each item a seller posts, they should also be able to track how many interactions it's had, if there are any people who have reached out for the item, in addition to how well rated the actual item is.
- 72) Storefront profile page for sellers. This will showcase some products that the seller has up for sale, along with some details about who the seller is or what their shop is about.
- 73) Seller tool - relist item. This will provide sellers with the option to easily relist their product being up for sale.
- 74) Seller tool - send offer to specific buyer/buyers. This will allow sellers to gain potential business from buyers who are interested in their products.
- 75) Keeping track of an item if it is the fair price. (this is for sellers)
 - a. For example, if a seller sells a table and he/she sets up to \$20,000. Then this would not accept to sell this item unless a seller changes the price.
- 76) If one of the items has very good ratings, then the feature can hide ratings.
- 77) If the seller's item(s) is/are illegal, then it is the seller's responsibility to display "illegal item", otherwise the seller can have a chance to get into trouble.
- 78) While displaying "illegal item", then the seller must explain the illegal item.
- 79) Striking system for users for wrongdoing. This will allow the app to prevent users from continuing misconduct or violations of the terms of service. Perhaps 3 strikes on a user account will lead to a ban.
 - a. User ban/blacklist

React Components

- 80) The About page should be featured on the home page. This will highlight the group who made the app, as well as the mission we aspire to carry out.
- 81) Create a React Component which has the job of loading profile data for each user. Users should have a profile page/ "My Account". This will allow users to view and update their personal information settings.
 - a. Users can update their account information, change their full name, update their email, change their password. Update their payment information or see their complete payment history.

- 82) Use of Redux `mapStateToProps()` for updating user's shopping cart information, loading user account details,
- a. The `mapStateToProps()` function will be used to load state data, such as a user's shopping cart items as properties passed into React components so that the data can be displayed to the user.
- 83) Function attached to each listing which sends a post request to node API to keep track of how many interactions the post has.

Website Features

- 84) Consignment operations, either for having items delivered or for meetup with the seller, or from a designated pickup location.
- a. Freedom to choose buyer's delivery options.
- 85) Sourcing form (daily item forecast), profitability test, how much it costs to ship the product as opposed to being shipped on other services.
- 86) A "See More" at the end of the top 50 posts, which when clicked will load another 50 posts to view on the Home page. Clients will be able to view more products by just scrolling up and down, instead of going back and forth to different pages.
- 87) Price matching, for items either already posted on our app or compared to other items on other websites (amazon, craigslist, eBay)
- 88) Item hashtags to help with categorization. Sellers will be able to use hashtags to categorize their products. All users will be able to search products using hashtags.
- a. To see the best deals.
- 89) Daily deals on the home page. This will highlight the sales going on and attract potential buyers to listings.
- 90) Show items which are similar in price or category to the item being viewed. A comparison between the item specs could be displayed as well.
- a. As an example, let us say that you are looking to buy an USB-C adaptor for connecting a monitor to your computer. When clicking on one USB-C item, users should have the option to compare the price, item features/details, and specs in depth so that they can make the most accurate decision for what they are looking for.
- 91) Show the statistic of numbers of purchase of seller's items.

UI/UX

- 92) When a user creates an account on the Angelapp platform, they can choose to be a customer, seller, or both.
 - a. After a user's account is created, they then have access to the ecommerce solutions we have to offer. They can view all features related to searching for items on sale or the option of setting up their own digital marketplace and selling items.
- 93) Navigation bar with routes to Home, Notifications, About, Contact, Login, Logout. This would be opened and closed when the hamburger menu is clicked. It will provide easy navigation on the website / app.
- 94) Use of sessions on the backend to keep track of the user's shopping cart information once they redirect away from the main shopping site.
- 95) Use of cookies for keeping track of user login information.
- 96) When users click the hamburger menu, there should be a navigation bar component which is triggered in addition to the sidenav which should pop out from the side.
- 97) Hamburger menu that changes from three bars to x when clicked. The menu will provide an up-to-date inventory of the products.
- 98) Navigation bar should have an animation triggered when it is opened or closed.
 - a. This task should be implemented with some sort of CSS transform or the use of a React component to represent the navigation bar
- 99) When a user clicks the Notifications tab in the navigation bar, a dropdown menu should appear.
 - a. A list of the user's notifications ordered by most recent to least recent is a requirement.
- 100) When a user clicks on a notification from another user, they should be redirected to a page displaying their conversation history.
- 101) If a user is not logged in, views an item, and then clicks the button to message the seller, the user should be prompted to either login with their username/password or to create an account with a button.
- 102) Lock password input in login form from being updated or clicked on after the user clicks login.
- 103) If a user attempts to login and they use the wrong credentials, they should be redirected back to /login route.
- 104) Once a user logs in with the correct username/password, redirect them to the home page at route '/' to start interacting with our app's features.

105) Contact page with forms to fill out.

- a. To store information for future references

106) Highlighted links of pages viewed/visited to let user know where they left off or what they have already viewed.

107) Magnifying glass feature.

- a. This can help a buyer be able to zoom in or zoom out of an item.

108) Switch to dark or white background mode for the app. This feature will help those who want to view the app in a different light or contrast.

109) If a user does not do anything while logging in for like 15 minutes or more. Then it automatically logs out for the user.

110) Option to drag item(s) to the shopping cart button. This will provide shopping ease for buyers.

111) Tax calculated automatically based on buyer origin. This tax will be reflected at the checkout process page.

112) Terms of Service page to be featured on the home page. This will include all the fine details of privacy agreements and the like.

5. List of non-functional requirements

Networking

- 1) There should be WebSocket functionality for displaying the most recent recently created posts, the hottest daily pics, and other displays of item data.
 - a. WebSocket functionality should also be used for displaying user's notifications, messages, and for auctions.

Security

- 2) SSL certificate, which enables websites to move from HTTP to HTTPS, for better website security.
- 3) Warning/caution list page for what sellers/buyers should look out for. A standard list providing details and warnings for potential scammers or bots.
- 4) Submit a ticket. After flagging an item since it seems sketchy, or an item might be fake, user could continue process by filing a ticket and sending a message to us to follow through further with reporting suspicious behavior.
- 5) Limited login attempts for security purposes. For example, if the user enters the wrong password 3 times, they will be locked out of trying to log in to their account for 30 minutes.
- 6) Captcha to fight against bots. Captcha provides an extra level of security for users.
- 7) Email verification for both seller and buyer. Verification ensures that there is a valid user behind each seller and buyer account.
- 8) Two factor authentications for added security measures. The user will be required to authenticate their account on two different devices to help prevent a breach in security.
- 9) Password recovery for all users when they need it. The recovery process will involve having an email sent to the user to the email address associated with their account, where they would have the opportunity to change their password.
- 10) Username recovery for all users when they need it. Like the password recovery process, where the user will get an email to recover their username.
- 11) Load balancers or distributed microservices on the backend so that we do not have just only servers doing all the backend work? We do not want the backend to go down and have the whole server unavailable to all users.

- 12) *Keeping track of unaccepted/accept usernames and write reviews.*
- *They must be appropriate and cannot contain duplicate usernames.*

Server-side validation

- 13) Correct login information entered. Username and password matching correctly.
14) Valid post information when a seller creates a new post.
15) The user's shopping cart should be updated with the total amount of the user's cart and their selected items FROM THE SERVER. User shopping cart data cannot be handled by the client, the client should only be used to display the data sent from the server.

Client-side validation

- 16) Username should be at least 8 characters in length. The username shall have a minimum length for security reasons.
17) User password should contain uppercase/lowercase and numbers with a minimum length of 8 characters. This password requirement ensures an extra layer of security.
18) When a user inputs their email into the registration form, a valid email should have @ along with a valid ending. This requirement checks if the email is in the right format.
19) New post title should not exceed 80 characters in length.
20) New post description should not exceed 500 characters in length.

Password encryption

- 21) A minimum of 10 salt should be used when hashing a user's password.
22) Perform password hashing when a user creates a new account. This will guarantee us that passwords saved in the database are secure.
23) Bcrypt library should be used for hashing a user's newly created password.

Authentication

- 24) A user's session should be serialized after logging in, and deserialized when a user leaves the site.

25) Stripe.js or PayPal payment SDK should be used for creating and validating payments from users.

Database

26) Seller analytics page (demographics of customers, what is sold more frequently, etc.) Analytics will provide the seller with information to help them better their product listings.

27) Passport library should be used to authenticate an existing user's requests.

28) Encapsulate any SQL statements so that SQL injections are not possible.

29) The table for seller posts should have the following constraints for its ID column: integer type, auto increment, and public key.

30) Each post should have a foreign key tied back to the user's id who created the post.

31) Each post should have columns for the posts' title, description, price, photo, the time when the post was created, if the post is still active on the app or whether it has expired

32) Post comments should be represented in a comments table. Each comment's id should be of integer type, auto incremented, and the public key.

33) Each comment should also have a foreign key tied back to the creator of the comment, and a foreign key tied back to the post the comment was for. Each comment should also have the time that it was created.

34) Session data for each user should be stored in the database. If their session is still valid, allow them to log in directly into the app.

Usability

35) Redux should be used to manage the state for the react client. Redux will be essential for keeping track of user's shopping cart information.

36) Multer library should be used for uploading images to the server/database.

37) Browser support for IE, Chrome, Firefox, Brave, etc. Cross-browser support for all users to be able to access.

38) By default, users who are not logged in should still be able to access the marketplace. It will show up on the top right corner that they are not logged in/login is shown instead of logout.

39) When fetching data from API routes, there should be a maximum response time of 1s. Anything more than this will impede upon user experience.

- 40) When a user sends an axios request, the server should be able to parse through urlencoded or json data.

Coding Best Practices

- 41) React components should be kept simple. Design a React component so that it accomplishes one job for the user. If a component is more complex, then use multiple react components with parent/child relationships.
- 42) An object titled INITIAL_STATE should be used in each Redux reducer to hold each reducer's state data.
- 43) Redux actions should start with the name of the reducer in capitals, an underscore, plus whatever the action is doing, an underscore, plus whatever the action is changing. Example: USER_UPDATE_POST, USER_SEND_MESSAGE, etc.
- 44) If a team member must create a new React route, then do so in App.js. This is where all the Routing takes place.
- 45) Variable names should not be redundant, but instead describe what role the variable has. Thinking of good naming conventions is important so everyone else on the team knows why and how the variable was used.
- a. By default, use camel case for variable names.
- 46) Commenting is essential. Before jumping into code, every team member should write some form of pseudocode describing what it is they are going to code before actually doing so.
- a. Outside of pseudocode, commenting should be used everywhere to describe how features or architecture works.

Product Constraints

- 47) At least 3 images of products required on listing. Having more than 1 image of the product provides customers with more information of how the product will look.
- 48) Minimum/maximum size of images on product page. An appropriate size of the images (perhaps 600x600 or 2x2) will help users with product visibility.
- 49) At least a 10-word description of the product being sold. The seller will provide adequate details on the product listing.

- 50) Buyer protections to provide buyers with peace of mind when using the app. (PayPal has 180 days for user to be refunded if there is an issue with the purchased item)
- 51) At least a 5-word title of the product being sold. The seller will provide an adequate title on the product listing.
- 52) Once an item is sold, the listing is automatically updated to reflect as such. This will notify users that the listing is no longer available/has been sold, or that the number of products available is now different.
- 53) Products must be concrete, not abstract (no services, only physical items.) Restrictions for what can be bought and sold on the app should be made clear.
- 54) Maximum number of listings per seller- 100? 1000? To keep balance of how we are trying to cater to the smaller businesses.
- 55) Offer commitment on each listing that has the offer option. Once the buyer makes an offer or bid and wins, a transaction automatically occurs, and the buyer gets their account charged (automatic withdrawal.)
- 56) Product page must include how much of each item is available. The number of products available lets the buyer know how much is in stock.
- 57) Product page has an expiration of 60 days. The expiration ensures that the product listings are current and that sellers are active.
- 58) Sellers should not be able to reject buyer's meetup time within 2 days unless it is an emergency case.
- 59) Product page must include a place of origin. This requirement will let the buyer know where the product is coming from, and perhaps what to expect when it comes to shipping time.
- 60) Product page must include if the item is new or used. This detail is essential to let the buyer know the condition of the product being sold.
- 61) 24-hour window to modify order; after the 24 hours, buyers will be committed to their order.
- 62) Must show the seller's name of a selling item.
- 63) Must show the seller's rating selling items.

Marketing and Revenue

- 64) Stay connected- follow us on social media. This will provide the app with cross platform exposure and potential business.

- 65) App news/announcements. Users will have the option to subscribe to the app to receive newsletters through email.
- 66) Paid ads throughout the site. Advertisements will be another source of revenue for the app. Perhaps the end of the page will include a link titled “advertise with us” to attract potential business.
- 67) FAQ page that highlights all the questions and answers that users generally have when using the app.
- 68) Coupon/promo code expiration. Every code will have an expiration date to keep discounts as special.
- 69) Recently sold items on the home page. This will show users what has recently sold, and for how much it has sold for.

Configuration

- 70) Team members should use pm2 as their process manager when running the project locally. ‘npm start’ is NOT necessary for running the project. Only ‘pm2 start process.config.js’ should be used
- 71) Freenom website should be used to configure a free domain name for our app
- 72) If the AWS EC2 instance is stopped and then reset, the public ipv4 address will have changed. This means the app’s public IP address must be changed in /credentials, and it must be updated in freenom’s domain name configuration.
- 73) Team members should use the ‘systemctl status nginx’ command to test the status of the nginx web server
- 74) If necessary, team members should only update the nginx server blocks contained in /etc/nginx/sites-available/default. Nowhere else.
- 75) In order to verify any errors related to nginx, always use the command ‘sudo nginx -t’
- 76) If team members need to reload the nginx server, then run ‘sudo systemctl reload nginx’

Git

- 77) If team members pull changes to a remote branch, make sure to ‘npm install’ so you have any new packages/libraries installed from other people’s commits.
- 78) Team members should always create new branches from the development branch.

- 79) After you are done with your work, create a merge request from your branch back into development branch so we can compare the differences between the two.
- 80) If you ever have a question about any new code pushed to GitHub, then team members must ask about what they are unsure on in the discord #help channel.
- 81) When team members encounter a bug in the code, team members should write down the steps to duplicate the bug, then create a pull request labeled as 'bug' so that the rest of the time can replicate the bug.
- 82) If team members need to add additional documentation for a feature they have created, they should add the documentation label onto their pull request.
- 83) If a team member is creating a new feature, then they should mark their pull request with the enhancement label.
- 84) If a team member has a question about a portion of code they are writing, then they should create a pull request from their branch back into development with the question label. This way, the rest of the team can review what the question is to resolve it.
- 85) Code should be merged from the development branch into master branch only when all merge requests for the current build have been agreed upon and added to the current git history.
- 86) Whenever a milestone is close to submission, a thorough check of the /credentials folder should be done to ensure everyone has the most recent updates.

Remote EC2 Instance

- 87) SSH should always be used when logging into the remote EC2 instance
- 88) Whenever logging into the EC2 instance, make sure to check for updates: use the following commands: 'sudo apt update', 'sudo apt upgrade' or 'sudo apt full-upgrade', and 'sudo apt autoremove'. If you are prompted to reboot, then reboot the instance.
- 89) Create a CI/CD pipeline for deploying code to the remote ec2 instance
- 90) The Node api server should have access to react build files so it works correctly on the remote ec2 instance
- 91) In order to login to the EC2 instance correctly, team members should ensure that the csc648.cer file in their local github repository's /credential folder has file permission 600.

UI/UX Constraints

- 92) CSS styling to create a box-shadow around an item when the user hovers over it. This will allow the user to view the item clearly.
- 93) High contrast between text and background for ease of use. This will make the overall appearance of the app easy to view.
- 94) The sidenav should be implemented with the react transition group library
- 95) All Contact, About, Careers, company information, etc. should be stored in the footer of the app
- 96) All Login/Logout, Shopping Cart, Notification, Auctions pages should be stored in the sidenav menu
- 97) @media queries should be used for making the app usable on mobile devices.
- 98) In order to satisfy requirements for screen sizes \leq iPhone 5, the minimum width for @media should be 300px
- 99) In order to satisfy requirements for screen sizes \leq iPhone 6, the minimum width for @media should be 360px
- 100) In order to satisfy requirements for screen sizes \leq Ipad, the minimum width for @media should be 720px
- 101) In order to satisfy requirements for screen sizes \leq Desktop, the minimum width for @media should be 1280px

6. Competitive analysis

Company	Facebook Marketplace	Craigslist	LetGo	OfferUp	Ebay	Mercari
URL	https://www.facebook.com/marketplace	https://www.craigslist.org/	https://we.letgo.com/	https://offerup.com/	https://www.ebay.com/	https://www.mercari.com/
Strengths	best digital marketplace in current market	oldest digital marketplace, simple business model, easy to use app	free-to-sell model, image recognition and AI capabilities	fun to use app (designed like social media platforms)	most trusted, Auction	Strong Customer support team
Weaknesses	no price-bidding/auction, long chat with buyers to close the deal	scam/fraud deals, outdated platform	requires in-app purchases for boosting listed item and other advanced features	promote and bump paid features for product boost	not preferred for local sales	high fees leave seller with no profit, sellers don't get money until buyer give reviews
Social Media	Facebook, instagram, Twitter, LinkedIn, Blogs	Blogs	Facebook, instagram, twitter, youtube, Blogs	Pinterest, facebook, instagram, youtube, linkedin, Blogs	Facebook, twitter, linkedin, Blogs	Facebook, Twitter, Youtube, Instagram, LinkedIn, Blogs

Release Year	2016	1995	2015	2011	1995	2013
Pricing	FREE	FREE excluding some services like job, apartment, gifts, furniture, services	FREE (raise funds from investors)	12.9%	12%	10%
Onboarding Experience	Guided step-by-step instructions	Not much support	Functions oriented Onboarding	Seamless, really smooth process	Comfortable buying and selling guide	Good onboarding video in the app
Shipping	5% extra checkout charge	not available	not available	9.9% service fee + shipping fee	10% on final price	based on package weight
User Interface	Similar to facebook	outdated	good	simple mobile-friendly UI	Last update was not up to the marks	good

Features Implemented

Not implemented: -

Implemented: +

Superior feature: ++

Feature	Facebook Marketplace	Craigslist	LetGo	OfferUp	Ebay	Mercari	Jose's Angels
Auction	-	-	-	-	++	-	+
In-App Chat	+	-	+	+	+	+	+
Delivery Options	-	-	-	+	+	+	+
Price Matching	-	-	-	-	-	-	++
Daily Deals	+	-	-	-	+	+	+
Advanced Search Bar	+	-	-	-	+	-	+
Discount Coupons	++	+	++	+	+	+	+
Seller Profile Rating	+	-	+	++	+	++	++

Summary of Competitive Analysis

Jose's Angels is a digital marketplace where a person can create an account as a buyer, seller, or both. Sellers will be able to sell unused or used items locally or globally. Unlike the main competitors in the current market, Jose's Angels supports local pickups as well as shipping of the products to expand the buyer base.

Our competitors are Facebook Marketplace, Craigslist, OfferUp, Mercari, LetGo and eBay. All of these marketplaces are successful in their business strategies and leading the market in one way or another. Our app's priority is to expand the buyer base by giving delivery options, increasing buyer and seller security via secure payment gateway and in-app chat, and build trust by making seller profiles which store the history of all the listings made by that seller. The app will be beneficial for both buyers and sellers. For buyers, it will provide "Price Matching" through comparison algorithms which will compare products with similar products on Jose's Angels as well as other competitors like Facebook Marketplace, Craigslist, Offerup products.

For sellers, Jose's Angels will provide the "Auction" feature which will enable them to set an initial bidding price for the product and the time duration of auction. This feature will add benefits on the seller's side by helping them make more money through sale. The sellers will also have their separate profiles which will contain ratings, reviews and history of listed items.

Jose's Angels app will solve the current problem of choosing whether to sell products locally or globally. The app will also calculate estimated shipping charges for products. The status regarding in stock items, already sold items and Out of Stock items will get updated runtime. Users will be able to modify their order in a 24-hour window. Today's main threat, which is fake products, will be removed through our app, as it will implement a flagging system which will give tickets to bogus products. Angel app will be cost efficient and will have easy to use and attractive UI which will take least time in searching for best deals online.

7. High-level system architecture and technologies used

API Services

- 1) UPS developer kit API (subject to change if we go with another delivery service)
- 2) Stripe.js or Paypal Developer SDK for satisfying checkout experience

List of Backend technologies/frameworks/tools

- 1) Node.js, v14.17.0
- 2) Express.js, v4.17.1, for routing only
- 3) AWS RDS mySQL database, v8.0.23
- 4) MySQL workbench, v6.3.8
- 5) AWS EC2 instance 1vCPU with 1 GB RAM
- 6) Ubuntu 18.04 Server
- 7) Nginx web server, v1.21.0
- 8) NPM, v6.14.13
- 9) Mysql2, v2.2.5
- 10) Multer for handling images, v1.4.2
- 11) Websocket, v7.4.6

List of Frontend technologies/frameworks/tools

- 1) React.js, v17.0.2
- 2) Axios for server requests, v0.21.1
- 3) React-Redux for state management, v7.2.4
- 4) Redux-Thunk library for making redux updates to state easier, v2.3.0
- 5) Browser support for Chrome/Firefox/Safari/IE
- 6) CSS

8. Checklist

Task	Status
Team found a time slot to meet outside of class	<p>ON TRACK</p> <ul style="list-style-type: none"> → Meeting 06/04/2021, 4:00pm PST 7:00pm EST → Meeting 06/08/2021, 2:30pm PST 5:30pm EST → Meeting 06/11/2021, 7:00pm PST 10:00pm EST → Meeting 06/14/2021, 6:30pm PST 9:30pm EST → Meeting 06/15/2021, 7:00pm PST 10:00pm EST → Meeting 06/17/2021, 7:00pm PST 10:00pm EST → Meeting 06/18/2021, 7:00pm PST 10:00pm EST
GitHub master chosen	DONE
Team decided and agreed together on using the listed SW tools and deployment server	DONE
Team ready and able to use the chosen back and frontend frameworks and those who need to learn are working on learning and practicing	ON TRACK
Team lead ensured that all team members read the final M1 and agree/understand it before submission	DONE
Github organized as discussed in class (e.g. master branch, development branch, milestone documents, credentials folders)	DONE

9. List of team contributions

<u>Student Name</u>	<u>Contributions</u>
Mitchel Baker	Functional and non-functional requirements, high-level system architecture and technologies used, checklist, and helped to better organize our list of main data items and entities
Charmaine Eusebio	Functional and non-functional requirements, main use cases, general organization
Kenneth N Chuson	List of main data items and entities, functional and nonfunctional requirements
Krina Bharatbhai Patel	Competitive Analysis, M-1 Editor, Functional requirements
Michael Schroeder	Main use cases, use case diagrams, nonfunctional requirements
Rowena Elaine Echevarria	Functional and non-functional requirements, organization, Competitive Analysis
Jamie Dominic Walker	Executive summary, Functional requirements