"SW Engineering CSC648/848 Section 4"

DrillShare

Rent tools owned by anyone

Section 4 Team 1:

Tyler Heslop (Team Lead) - theslop@mail.sfsu.edu
Trevor Henrich (Front-End Lead)
Saksham Manandhar (Back-End Lead)
Kevin Karmacharya (Scrum Master)
Arsany Attalla (Github Master)

"Milestone 2"

Date of Submission: March 18, 2022

History Table

Revision ID	Revision Date	Revised By
1.0	05/10/2022	Kevin Karmacharya
2.0	05/17/2022	Kevin Karmacharya

1. **Data Definitions V2**
    a. tool (alt names: tool, item, rentable, shareable)
        i. Name
        ii. Category
        iii. model_num
    b. Listing(alt names: listing)
        i. Title
        ii. Description
        iii. Photos
        iv. model_num
        v. ratehourly
        vi. rateDaily
        vii. postOwner
        viii. currentRenter (can use this for isAvailable)
        ix. rentalStatus
    c. Contract (Alt names: agreement)
        i. Rentee
        ii. Renter
        iii. dateStart
        iv. dateEnd
        v. priceRent
        vi. priceDelivery
        vii. isPaid
        viii. isActive
        ix. isComplete
        x. deliveryID
    d. Delivery
        i. Renter
        ii. Rentee
        iii. dateDelivery
        iv. dateDelivered
        v. isActive
        vi. isDelivered
        vii. Address
    e. Profile
        i. NameLast
        ii. NameFirst
        iii. Email
        iv. Phone
        v. Drivers license

   vi. Address
   vii. DOB
f. User
   i. nameLast
   ii. nameFirst
   iii. Email
   iv. password
   v. phone
   vi. driversLicence
   vii. Address
   viii. DOB
   ix. userType (Renter, Owner, or both)
g. Payment
   i. cardNumber
   ii. expDate
   iii. authcode
   iv. creditcardType
   v. User
h. Transaction
   i. List of contracts
   ii. Subtotal
   iii. Tax
   iv. Total
   v. transactionStatus (in progress/complete)
i. Receipt
   i. PayTo
   ii. PayFrom
   iii. Subtotal
   iv. Tax
   v. Total
   vi. Amount paid

2. Functional Requirements V2

1. **Login and Registration (Priority 1)** - A very important feature we want is the ability for the user to register and login into Drillshare such that they can rent out their items or rent an item from another user. We plan to use lazy registration, which will allow for the user to browse our product without having to log in, until they want to post or rent out an item.
   a. R.1.1(P1) - Users will be able to create one account that will allow them to rent a tool or post their tool for rental services.
   b. R.1.2 (P1) - Users will be able to browse the site without having to log in until they start the rental process.

2. **List an item (Priority 1) -** Once users are registered and logged in, they have the option to list an item for sale and other users can buy them or checkout their listings.
   a. R.1.1(P1) - If users are logged, they will be able to list an item for sale by filling out a form.
   b. R.1.2 (P1) - Users will have to fill out a form that allows them to add a title, custom sku, category, condition, description, and up to six images for their item they wish to put up for rent.
   c. R.3.1(P3) - Users will be able to post more than one item after another so they don't have to keep going back to the home screen.

3. **Rent an item (Priority 1) -** Users also have the option to rent an item of theirs for rent.
   a. R.1.1 (P1) - If users are logged, they will be allowed to rent an item.
   b. R.1.2 (P1) - Anyone who isn't logged in will be allowed to rent an item, after they log in or create a new account.

4. **Return an item (Priority 1) -** Users will be allowed to return items they have purchased if the item they received doesn't match the owner's description or if the time period for the item is reached.
   a. R.1.1 (P1)- Once a rentee finishes using their item during the rental time frame, they must return the item, either via pickup or delivery within a postmark time of 24 hours to not be charged a late fee.
   b. R.3.1 (P3) - If an item doesn't match what a rentee wanted, they can return the item for no cost.

5. **Delete an item (Priority 1) -** Renters can delete their listing if the equipment is not available, damaged, or simply doesn't want to rent it out to others.

    a. R.1.1 (P1) - Renters who wish to not rent out their item can delete their listing from the site for any reason.

6. **Searching for an item (Priority 1) -** Users will be able to search for a certain item or search for a certain category of products.
    a. R.1.1 (P1) - Anyone will be allowed to search for items they wish to rent without needing an account
    b. R.1.3 (P1) - Anyone will be able to see the item details and images that the owner of the item has put up for rent.
    c. R.2.1 (P2) - Anyone can search for a certain group of categories such that relevant products show up.

7. **Order History (Priority 2) -** Users will be able to view their order history and all the items they have purchased as well as items they have listed before and were sold or rented.
    a. R.2.1 (P2) - Anyone, with an account, will be able to check their current rentals and items that they have put up on the site for rent.
    b. R.3.1 (P3) - Renter accounts will be able to check their order history of products they have rented in the past.
    c. R.3.2 (P3) - Owner accounts will be able to see their order history for each tool they have posted.

8. **Filtering item (Priority 2) -** There will be an option for filters on a certain page for a certain item to make search easier on users and only return the items they are looking for.
    a. R.2.1 (P2) - Anyone will be able to filter by price, current rent status, and category of the item.
    b. R.3.1 (P3) - Anyone will be able to filter their search results such that they get the results they want for the product they wish to rent.

9. **Messaging (Priority 3) -** Owners and buyers/renters will be able to chat with each other to ask questions or discuss any inquiries regarding a certain listing.
    a. R.3.1 (P3) - Owners will be allowed to chat with the rentee, during the duration of the rental period, to discuss any problems the rentee may have with their tools.
    b. R.3.2 (P3) - Renters will be allowed to chat with the owner of a product, during the duration of the rental period, to discuss any issues the rentee may have with the tool.

10. **Notifications (Priority 3) -** Users will get notifications for new updates and for new listings for a certain item they were looking for or if someone buys/wants to rent their item, or if they get a message from buyer/seller/renter.
    a. R.3.1 (P3) - Owners will get notifications if someone has a question or wants to rent out their product.
    b. R.3.2 (P3) - Renters will get notifications if an item they have recently searched is available to use.
    c. R.3.3 (P3) - Renters will get notifications if an item is about to ship/be delivered.
    d. R.3.4 (P3) - Renters will get notifications if an item's rental due date is close to help prevent them from taking a late fee.

11. **Reviews (Priority 3) -** Users will be able to give reviews for each other. Buyers and sellers will be able to give ratings and reviews based on their purchase experience.
    a. R.3.1 (P3) - Anyone, with an account, can write a review for a product to tell other users of their experience with the owner or product.
    b. R.3.2 (P3) - Renters who have completed a rental transaction, from start to finish, will be able to rate their experience with the owner of the tool as well as the tool.
    c. R.3.3 (P3) - Owners who have completed a rental transaction, from lending their tool to receiving it back will have the opportunity to rate the renter of their tool.

12. **Recommending Items (Priority 3) -** Users will have items recommended for them based on their frequent searches and interests.
    a. R.3.1 (P3) - Based on what users have searched, they will be recommended, along the side bar or bottom bar, similar items that may interest the user.
    b. R.3.2 (P3) - Based on what users have rented, they will be recommended, similar items that may go along with the item they have rented or similar items that they have just rented.

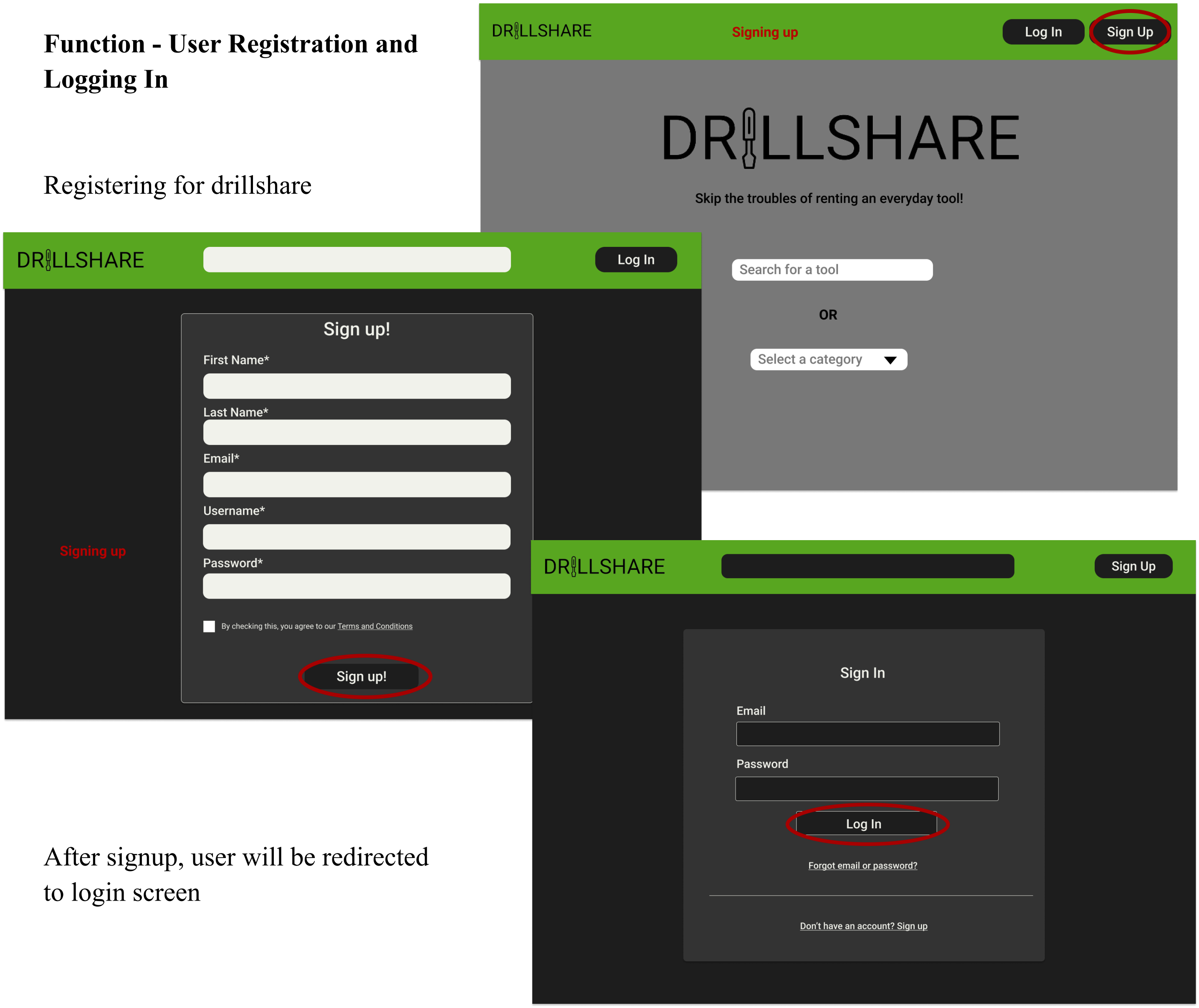
3. UI Mockups and Storyboards (high level only)

UX Validation Meeting: As a team, we came together to talk about the UX principles of our wireframe before creating our GUI design for our storyboard. We talked detailly of each of the 7 UX principles. Everyone gave their feedback about Drillshare's initial design and we changed some of our designs according to each principle. We wanted to make sure that our site followed each of the 7 UX principles so that anyone with basic internet knowledge could use our site at ease. Over spring break, we plan to also ask our friends and family about our UX designs to gather feedback so that we can create a better product for launch. We want to make sure we gain as much feedback as possible so that Drillshare is useful, usable, findable, credible, accessible, valuable, and desirable for everyone!

Function - User Registration and Logging In

Registering for drillshare

After signup, user will be redirected to login screen



DRILLSHARE

Signing up

DRILLSHARE

Skip the troubles of renting an everyday tool!

Search for a tool

OR

Select a category

Sign up!

First Name*

Last Name*

Email*

Username*

Password*

□ By checking this, you agree to our Terms and Conditions

Signing up

Sign In

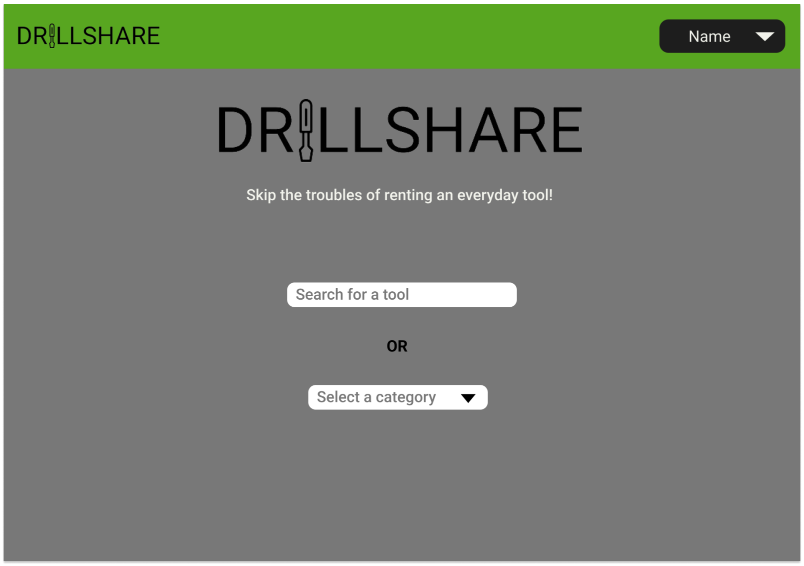
Email

Password

Forgot email or password?

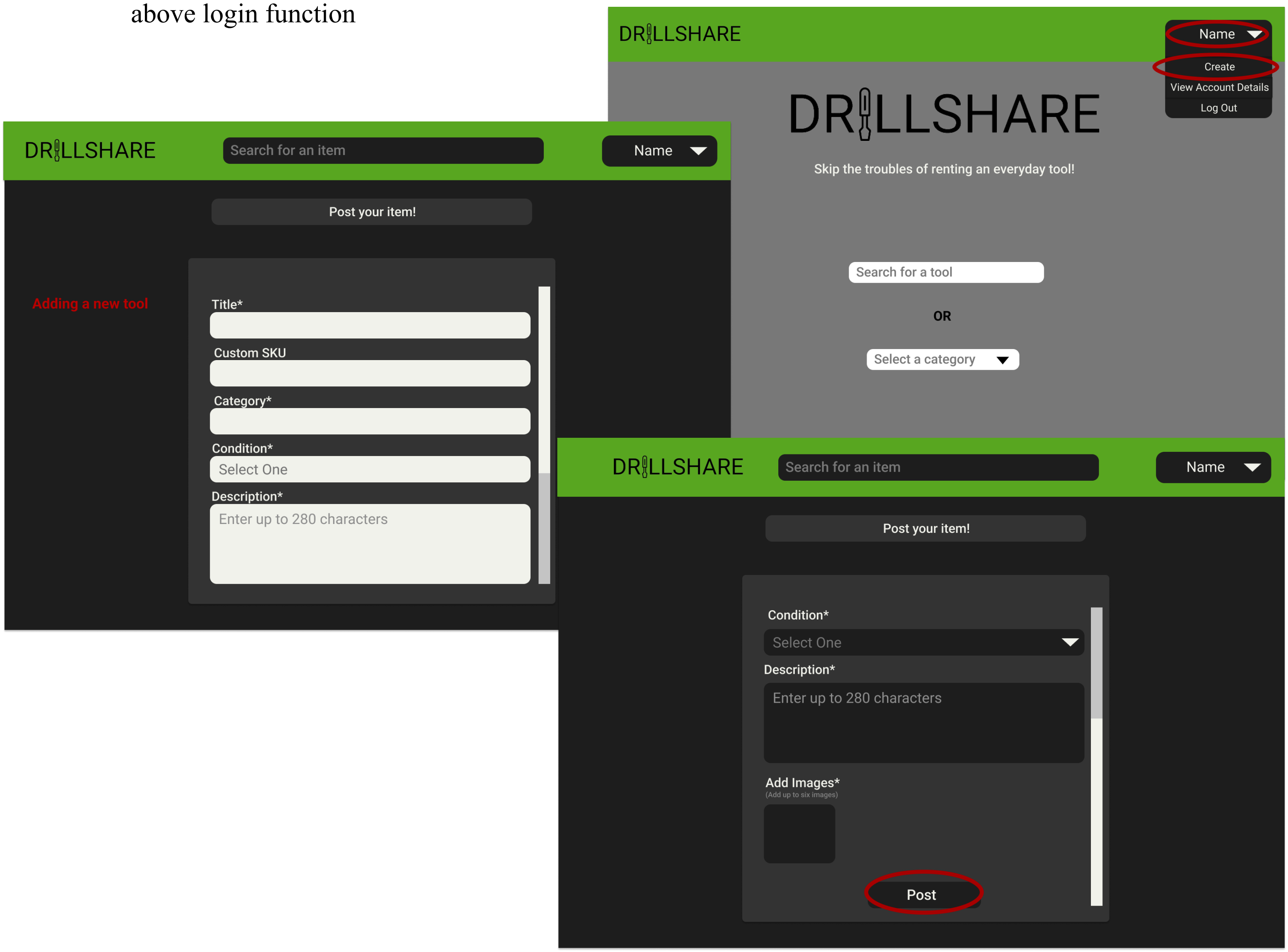
Don't have an account? Sign up

Homepage for all users.

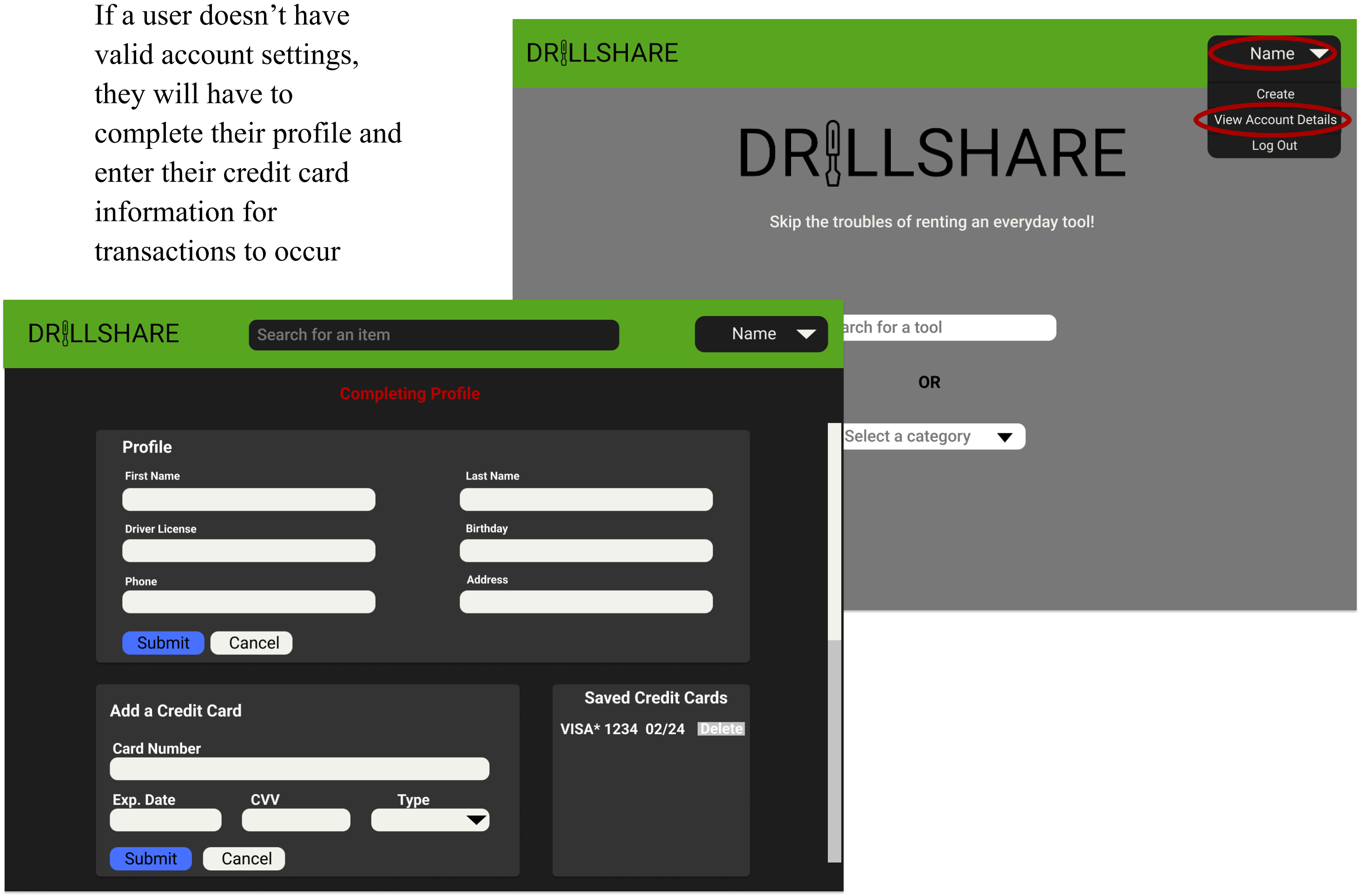


Function - List an item

After logging in, from the above login function

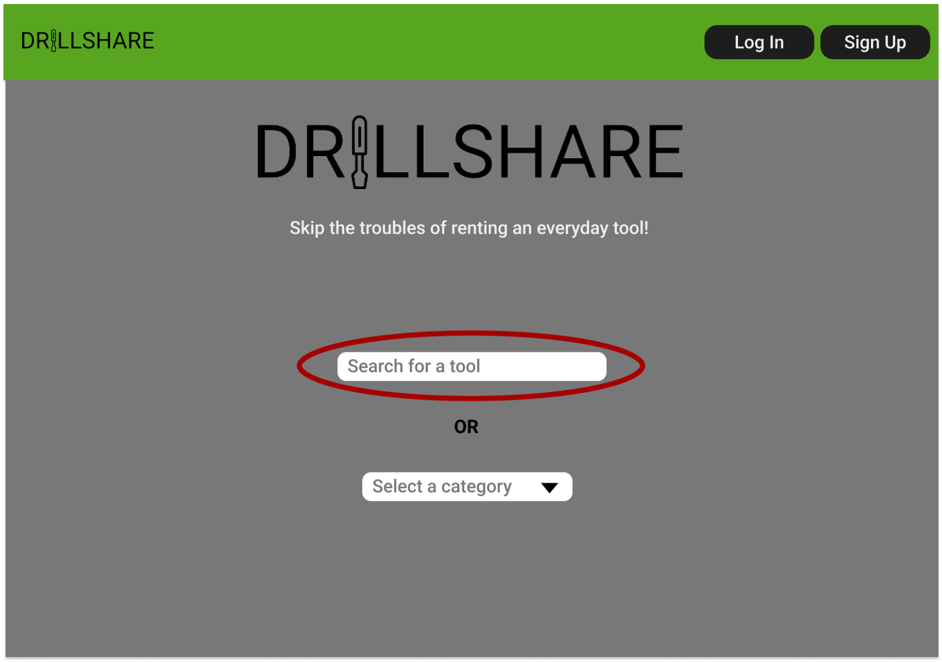


If a user doesn't have valid account settings, they will have to complete their profile and enter their credit card information for transactions to occur

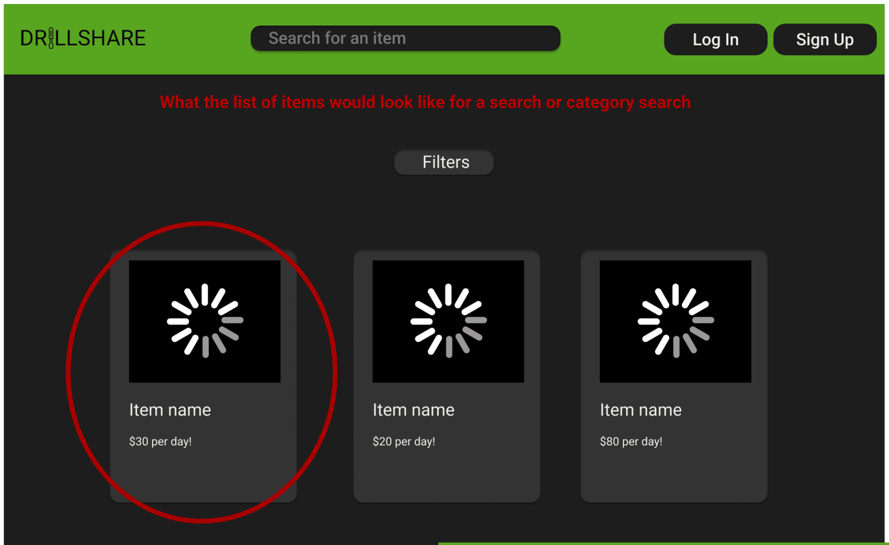


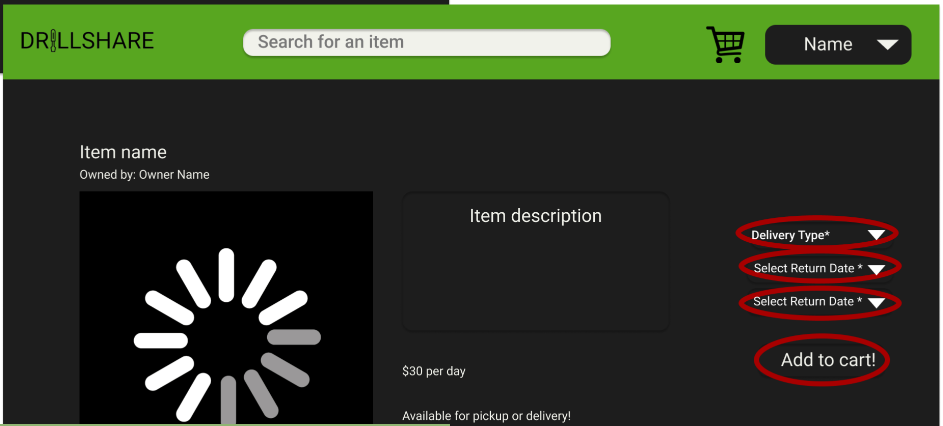
Function - Search an Item and Rent an item

The user doesn't have to login until they want to rent an item.

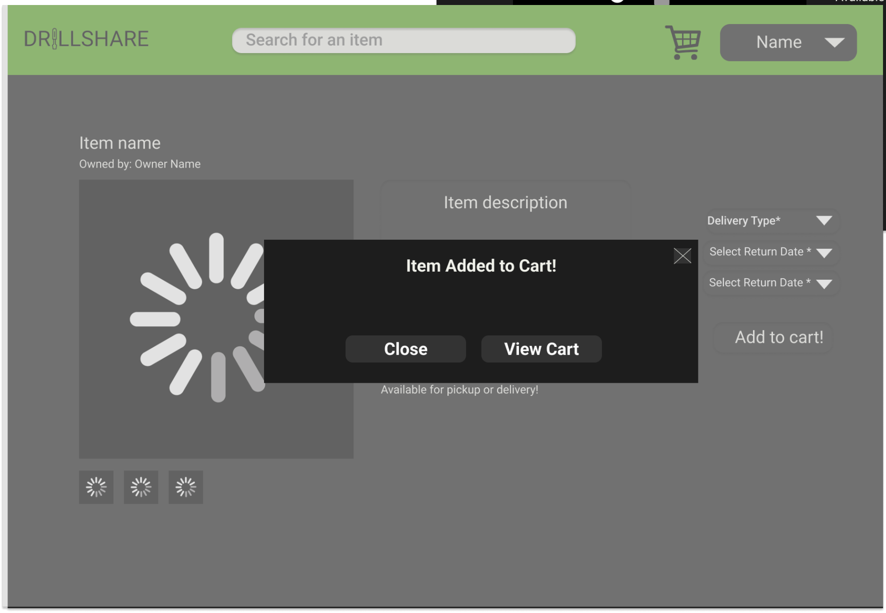


After a user searches for an item

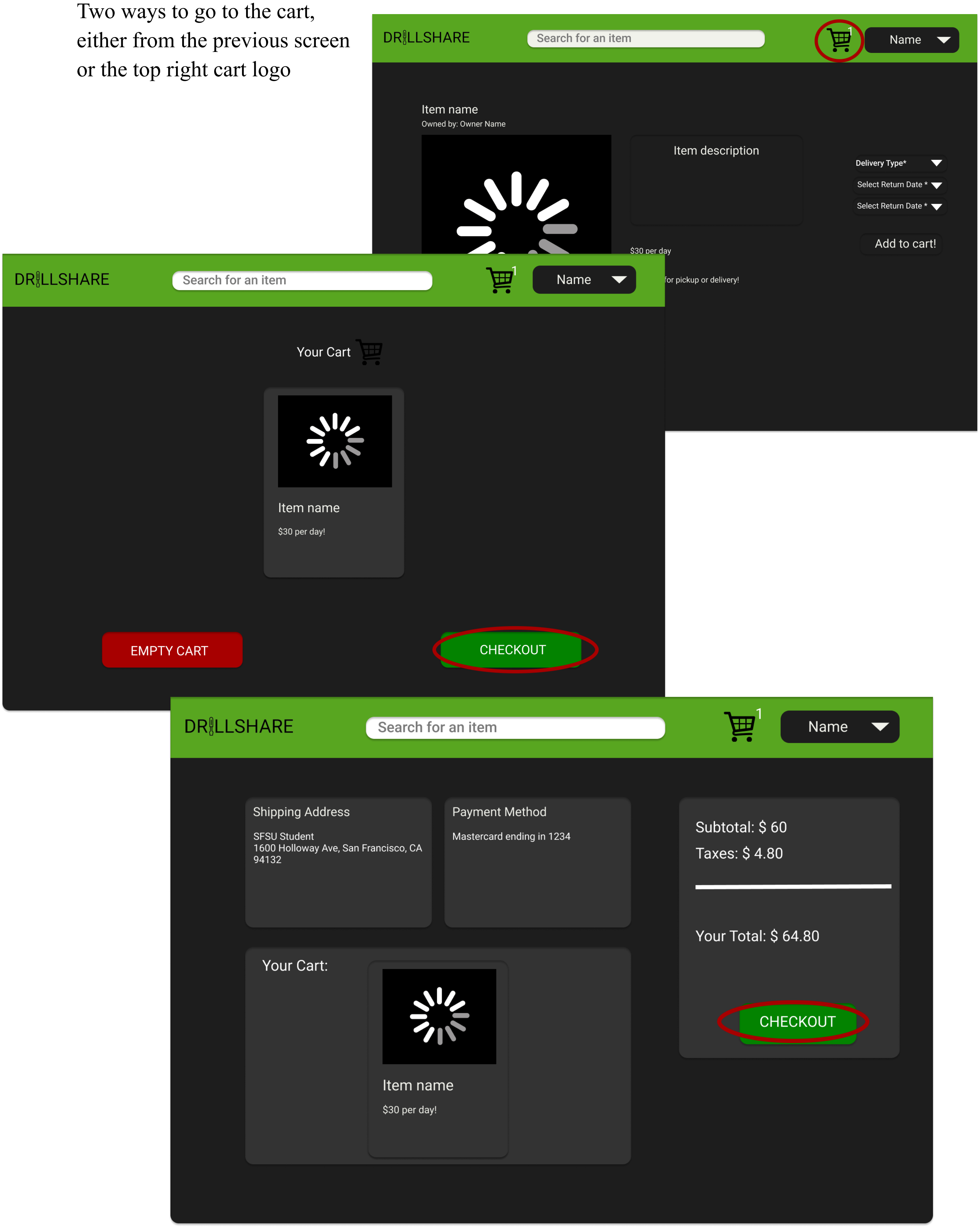

What the list of items would look like for a search or category search



The user must login before they can rent out an item.


Item Added to Cart!

Two ways to go to the cart,
either from the previous screen
or the top right cart logo



Shipping Address

SFSU Student
1600 Holloway Ave, San Francisco, CA
94132

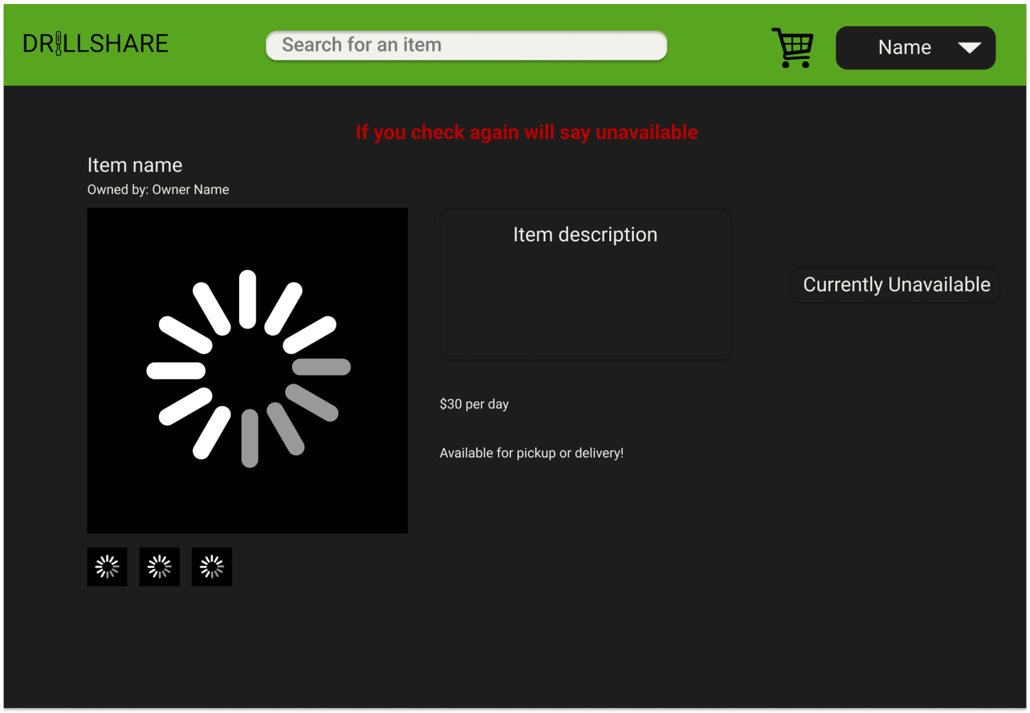
Payment Method

Mastercard ending in 1234

Subtotal: $ 60

Taxes: $ 4.80

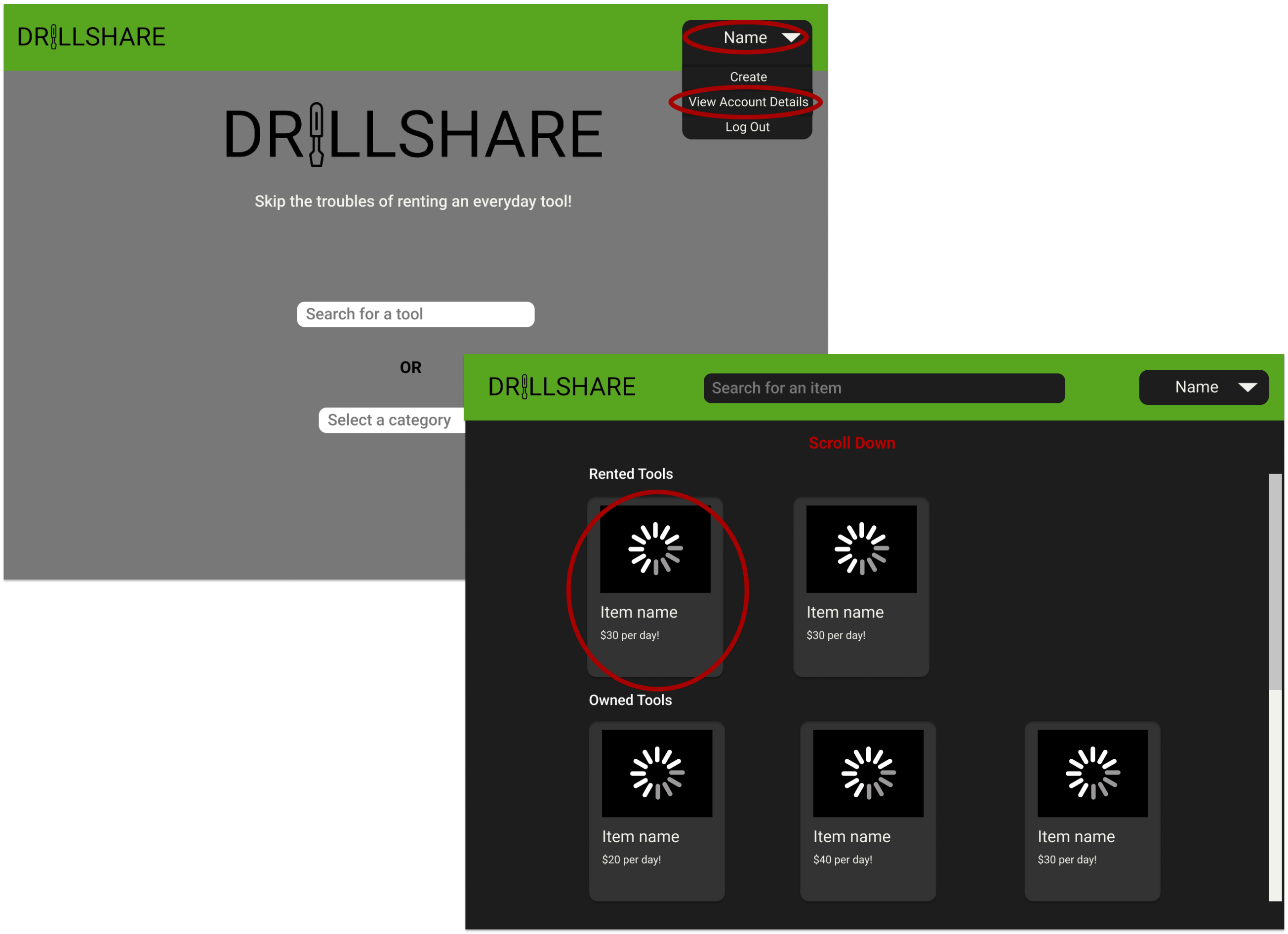
Your Total: $ 64.80



If you check again will say unavailable

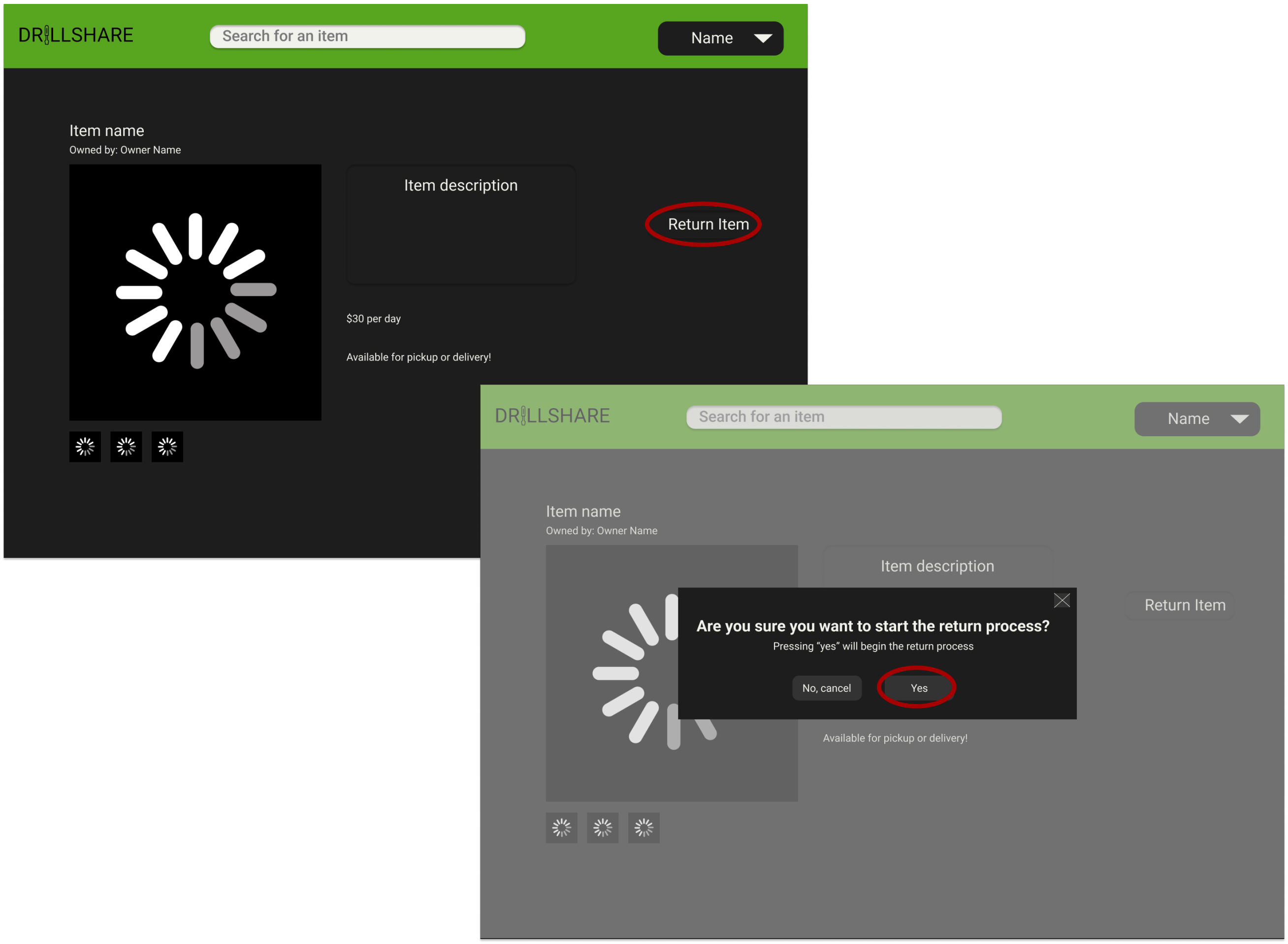
Function - Return an item

After the user has logged in, they can view their own tools and return their tools they are renting.



Skip the troubles of renting an everyday tool!

Scroll Down

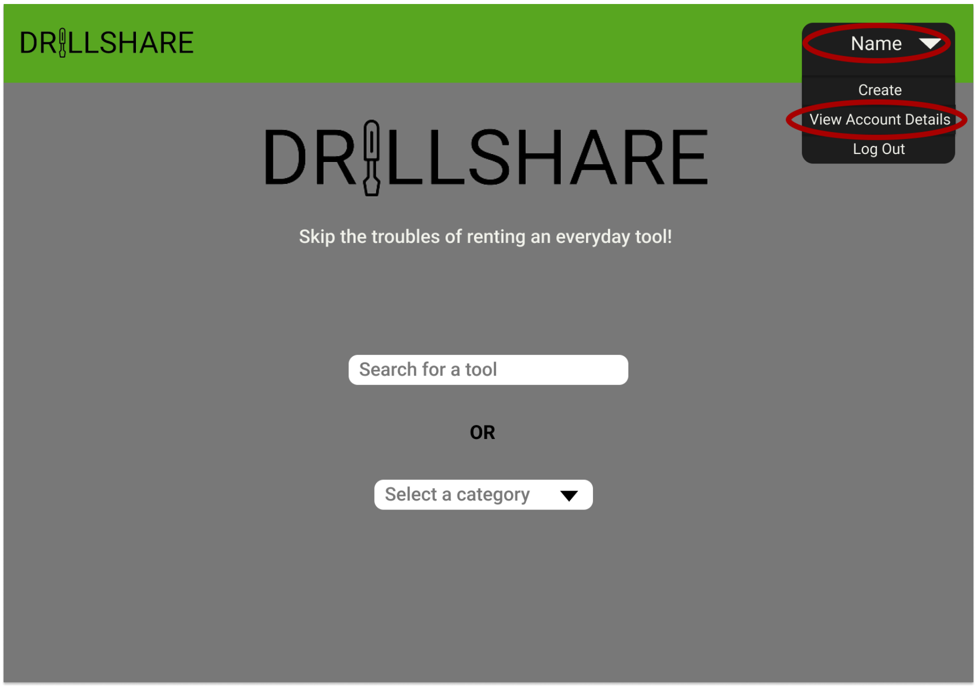


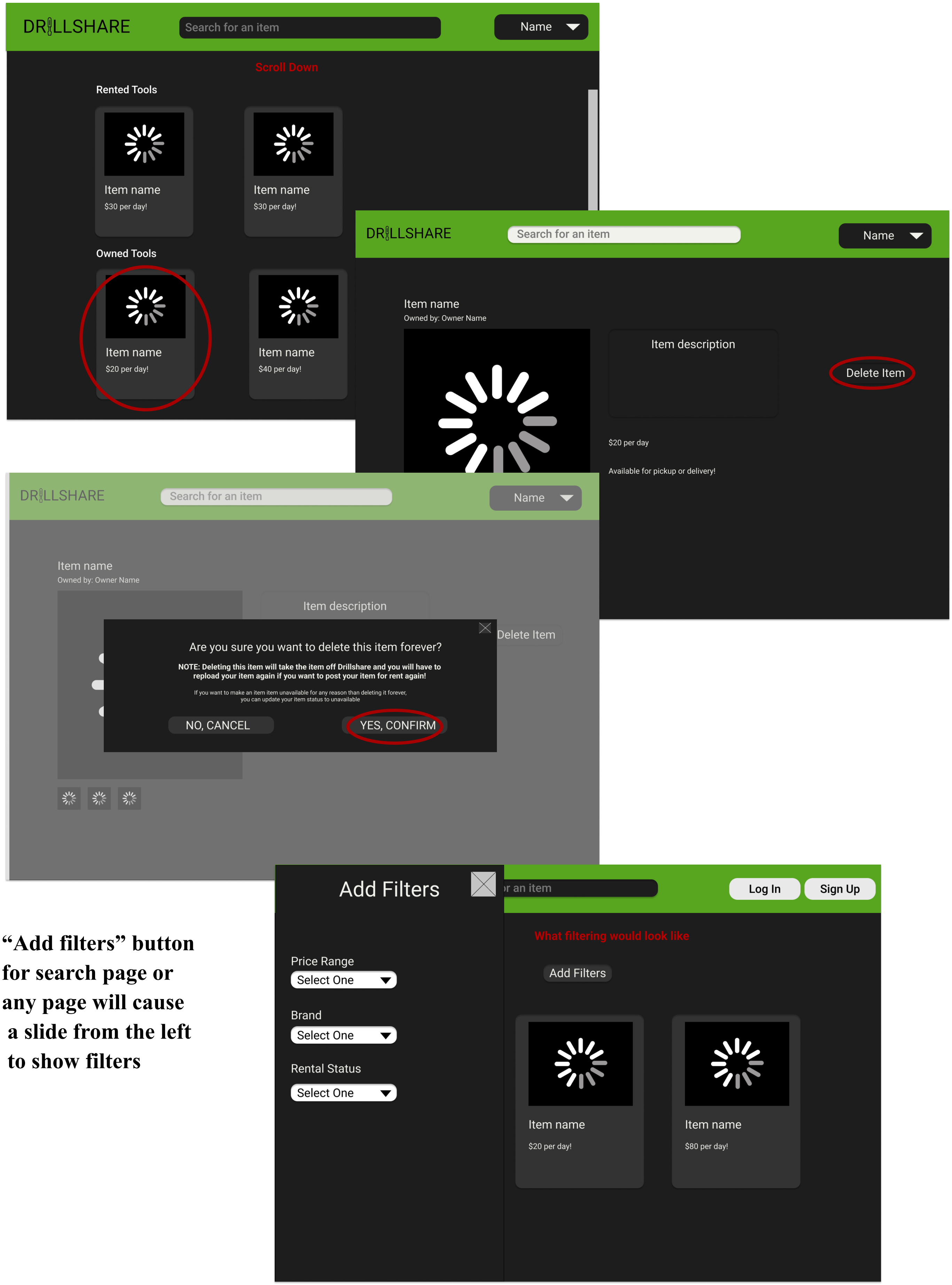
Are you sure you want to start the return process?

Pressing "yes" will begin the return process

Function - Delete an item

After the user has logged in, they can go into account settings to delete their listings from the site.





DR!LLSHARE

Search for an item

Name ▼

Scroll Down

Rented Tools

Item name
$30 per day!

Item name
$30 per day!

Owned Tools

Item name
$20 per day!

Item name
$40 per day!

DR!LLSHARE

Search for an item

Name ▼

Item name
Owned by: Owner Name

Item description

Delete Item

$20 per day

Available for pickup or delivery!

DR!LLSHARE

Search for an item

Name ▼

Item name
Owned by: Owner Name

Item description

Delete Item

Are you sure you want to delete this item forever?

NOTE: Deleting this item will take the item off Drillshare and you will have to reload your item again if you want to post your item for rent again!

If you want to make an item item unavailable for any reason than deleting it forever, you can update your item status to unavailable

NO, CANCEL

YES, CONFIRM

Add Filters

Price Range
Select One ▼

Brand
Select One ▼

Rental Status
Select One ▼

r an item

Log In

Sign Up

What filtering would look like

Add Filters

Item name
$20 per day!

Item name
$80 per day!

“Add filters” button for search page or any page will cause a slide from the left to show filters

4. High Level Architecture, Database Organization

DB Organization:

1. tool (alt names: tool, item, rentable, shareable)
    a. Name
    b. Category
    c. model_num
2. Listing(alt names: listing)
    a. Title
    b. Description
    c. Photos
    d. model_num
    e. ratehourly
    f. rateDaily
    g. postOwner
    h. currentRenter (can use this for isAvailable)
    i. rentalStatus
3. Contract (Alt names: agreement)
    a. Owner
    b. Renter
    c. dateStart
    d. dateEnd
    e. priceRent
    f. priceDelivery
    g. isPaid
    h. isActive
    i. isComplete
    j. deliveryID
4. Delivery
    a. Owner
    b. Renter
    c. dateDelivery
    d. dateDelivered
    e. isActive
    f. isDelivered
    g. Address
5. Profile
    a. NameLast
    b. NameFirst
    c. Email
    d. Phone

    e. Drivers license
    f. Address
    g. DOB

6. User
    a. nameLast
    b. nameFirst
    c. Email
    d. password
    e. phone
    f. driversLicence
    g. Address
    h. DOB
    i. userType (Renter, Owner, or both)
7. Payment
    a. cardNumber
    b. expDate
    c. authcode
    d. creditcardType
    e. User
8. Transaction
    a. contract
    b. Subtotal
    c. Tax
    d. Total
    e. transactionStatus (in progress/complete)
9. Receipt
    a. PayTo
    b. PayFrom
    c. Subtotal
    d. Tax
    e. Total
    f. Amount paid
    g. Total

Add/Delete/Search architecture:

- When inputting the information for listings, the title, description, rates, image, owner of the post will all be added into our database. This information should show up in the listing collection which is then displayed on the website.
- The listings can then be searched using a search bar or the filters. The entries that will be searchable in our database will be the title, and description. In future updates, we hope to expand the search to also include the categories and the

product owner. Regarding the information displayed on the home page, we will have an image with the title and category. When a listing is clicked, it will display a larger image of the product with additional information such as the description, the owner, the rates, and availability on a separate page.

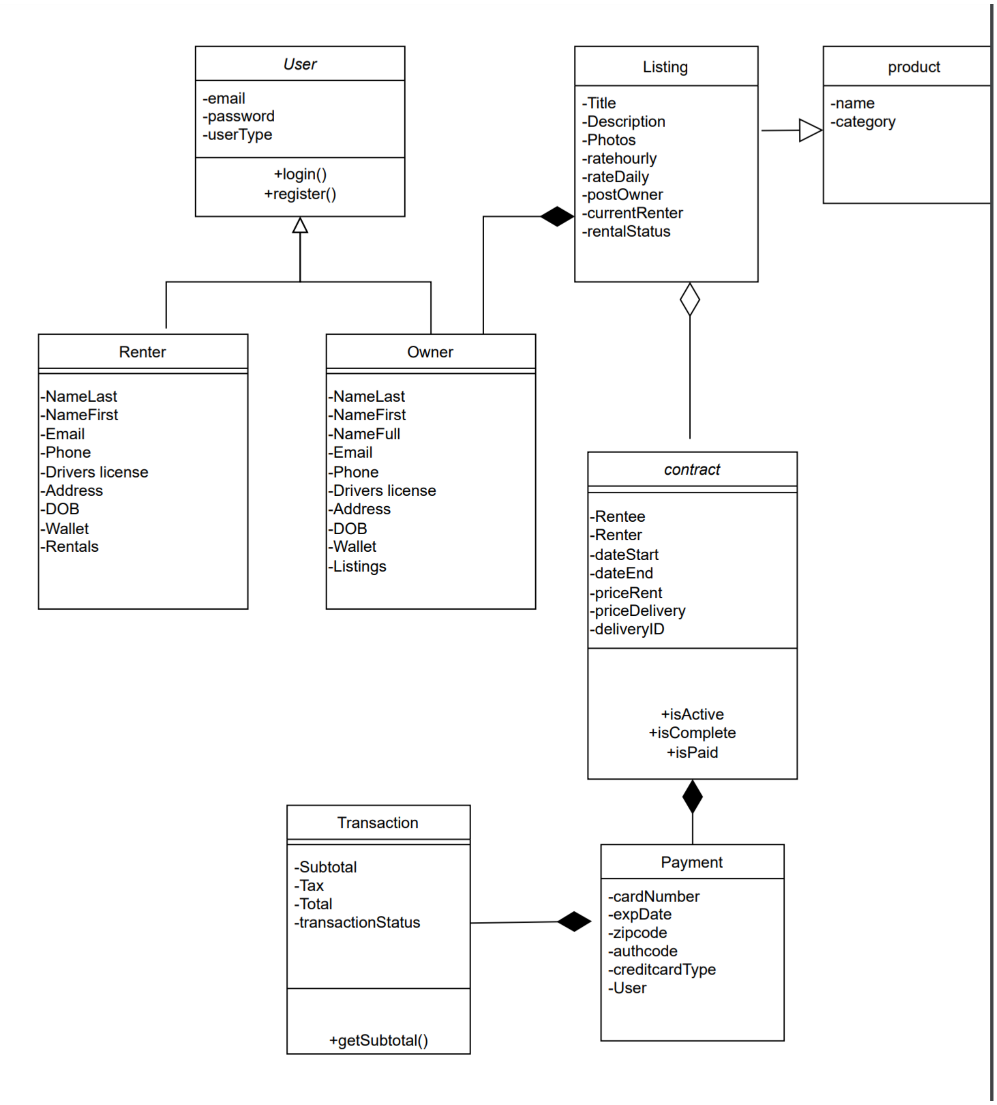
- These same listings can also be deleted by the owners. In the case that the owner wants to remove the listing due to damages or whatever reason, the owner can delete it and it will be removed from the database.
- When users who want to rent create an account and choose a product to rent, they will be required to have a payment method on record. They will fill out the form inserting the proper data fields in which the data will be stored in the database. It will not be displayed as it is confidential information to the user.

API:

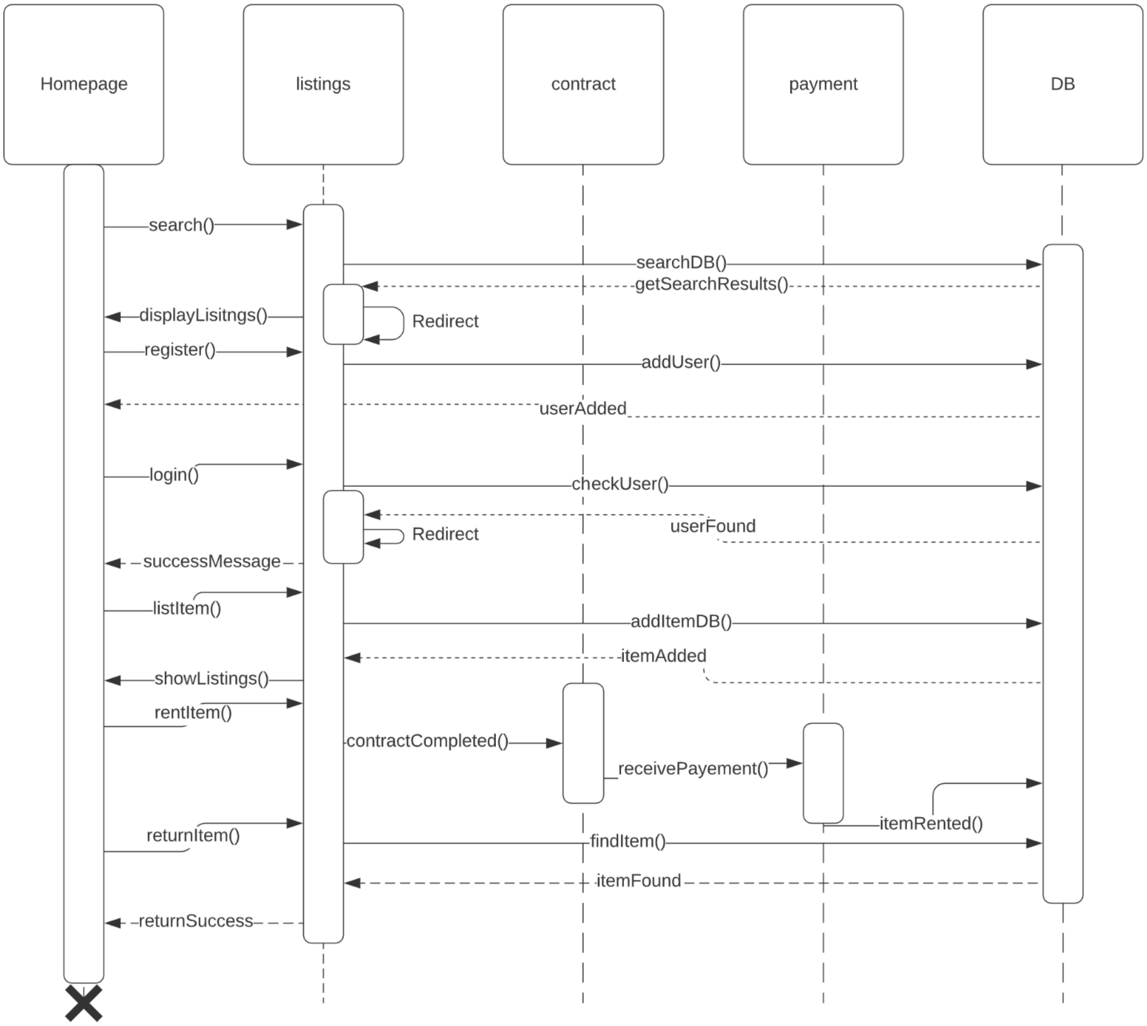
- Tool
    - createTool
- Listing
    - getListings
    - createListing
    - editListing
    - deleteListing
    - searchListing
- User
    - createUser
    - loginUser
    - deleteUser

5. High Level UML Diagrams

- UML Class Diagram



- Sequence Diagram:


Homepage
listings
contract
payment
DB
search()
searchDB()
getSearchResults()
displayLisitngs()
Redirect
register()
addUser()
userAdded
login()
checkUser()
userFound
Redirect
successMessage
listItem()
addItemDB()
itemAdded
showListings()
rentItem()
contractCompleted()
receivePayement()
itemRented()
returnItem()
findItem()
itemFound
returnSuccess

6. Identify actual key risks for project at this time

Skills Risks and Mitigation Plan

Possible Risks	Possible Solutions
Unfamiliar with a needed tool or technology.	- Have open discussions with one another - Encourage posting resources in #framework-review channel on Discord - Schedule team meeting or review days - Schedule quick call
Incompatible versions of tools with one another	- Perform code review as a team - Making sure Github is clean and easy to use

Schedule Risks

Possible Risks	Possible Solutions
Team work is being handled on Discord, Trello and Google Docs which may cause confusion on what one is supposed to do	- Use Trello to make it clear on what needs to be done - Use Discord to show who needs to do what task
Team member cannot make team meeting	- Share meeting minutes - Schedule consistent meetings based on availability
Team member not responding to discussions on Discord	- Maintain open discussions during team meeting - Check in with team member individually
No changes on update status for at least 2 days	- Check to see if anyone has any problems - Ask if anyone has finished a task

Teamwork Risks

Possible Risks	Possible Solutions
Unable to make our second meeting on Friday @1:00pm.	- We have a backup day, the following Saturday @1:00pm. - Post meeting minutes - Communicate on Discord
Team members need help or are falling behind on the task they are assigned.	- Ask if they need help on what is needed to complete the task so we don't fall behind as a group. - Make the task more flexible by dividing work
A team member needs help, but doesn't ask or notify anyone.	- Check in with team member individually - If they continue to not respond, contact to higher authority
Team member doesn't respond for sometime / uncooperative	- Check in with team member individually - If they continue to not respond, contact to higher authority
Deadlines are not met	- Divide task into smaller parts - Remain flexible on tasks - Increase collaboration to work on the tasks
Github has merge conflicts and is not consistent	- Have Github master contact team members where issues arise - Have clear documentation in code - Communicate what needs to be fixed and how to fix it

Legal/Content Risks

Possible Risks	Possible Solutions
A user may post a copyrighted image from another user with having the correct rights to the image.	- Enforce some sort of uniqueness of image - Have active admins check over images

A user may post stolen items to rent out to other users	- Ban user account selling stolen goods - Have active admins check over items - Contact law enforcement

7. Project Management

As a team, we managed our Milestone 2 tasks and Vertical SW Prototype tasks by our scrum meetings, Discord server, Trello server, and Google Docs. We used our scrum meetings to update one another on the progress we were making between our scrum meetings, as that was when everyone was available to communicate at the same time. We also used scrum meetings to decide upon new tasks or to divide tasks with each other. This allows us to complete tasks much faster so that the backend team and frontend team could also work on the Vertical SW Prototype. Everyone in the group transparently shared their progress by showing what they had done in between scrum meetings and talking about the issues they were facing so that another team member could offer help when needed. Outside of the scrum meetings, we communicated using our Discord server to update each other on the task we were working on in between scrum meetings. We also used Discord to have a standardized “To-do” list to work on in-between scrum meetings to see what each group member’s task looked like for the week. To allow flexibility for adding, changing, or checking off tasks for all group members, we used Trello and Google Docs. We used Trello for setting up the dates and times for when each task should be completed by. We used Google Docs for our milestone 2 document, which we are currently on, to set up a checklist to help team members easily check what they had to do when they opened the Milestone 2 document.