The Underbots

RentO

Software Design Document

Version 1.0

Date: 5/5/2023

Computer Science / CSUN
Section #: 21063

Professor Edmund Dantes

TABLE OF CONTENTS

Revisions

Version	Description	Date Completed
1.0	Primary Version	11/21/2022
1.1	Removal of React as our project has no React coding involved.	5/5/2023

1.0 INTRODUCTION

1.1 Purpose

The purpose of this document is to describe the architecture of our e-commerce web application, showing how the front-end and the back-end work together to produce a fantastic website our users can explore and around. This document is intended for the Underbots team and any project managers, software engineers, and anyone else outside of the main team that is interested in reading it.

1.2 Scope

The scope of our software is to allow users to rent and lease certain items that they are interested in, instead of buying it outright, to either test it out before buying it or use it as a one-time use for a project or event. The modules we have include having an account for the user to save their information, transaction history, and if they are leasing, save their items. Another module is having the listing page to show details of the item that is available for rent, and depending on who is seeing the item, can make the option to rent it or edit the details if they are the owner. Another module includes a cart and checkout page where you can see what items you have chosen to rent out and the final step before finalizing the rent contract. Next is having a search list page where the user is shown various items, based on their choice of categories or what they have typed in the search bar. We will also include a home page where users are first introduced to users at our web when they enter the URL of our website. And last two modules, will have a profile page where they will show the users, what they have to offer (if they are leasing), their current review and any contact information, as well as an About Us page, where we will include the team's information, contact us details and a FAQ section. These modules will be explained later in the documents. All the modules will be hosted at AWS, including front-end, back-end, and the database.

1.3 Overview

The overview of this document discusses, in detail, the architecture and diagrams we used to flesh out our project. The document will start off by talking about the system overview and how the web site will operate in a very simple manner. Then the architectural design will follow and will go into more details on how the system will be managed by several components of the website. This will also include the interface design, a decomposition description of what the client and server will do during a user's time interacting with the website, and an explanation on why the group chose these methods. After that. The document will go into greater detail, via class and sequence diagrams and talk about the variables and functions that will be used to manage and process the website. The second to last section is about human interaction and this will explain what the user will see and interact with, including screenshot of our front-end part of certain web pages, and explains the actions and objects used. Lastly is the requirement matrix and this will show the traceability between the functional requirements from the Software Requirements Specification and the information explained in this document.

1.4 Reference Material

N/A

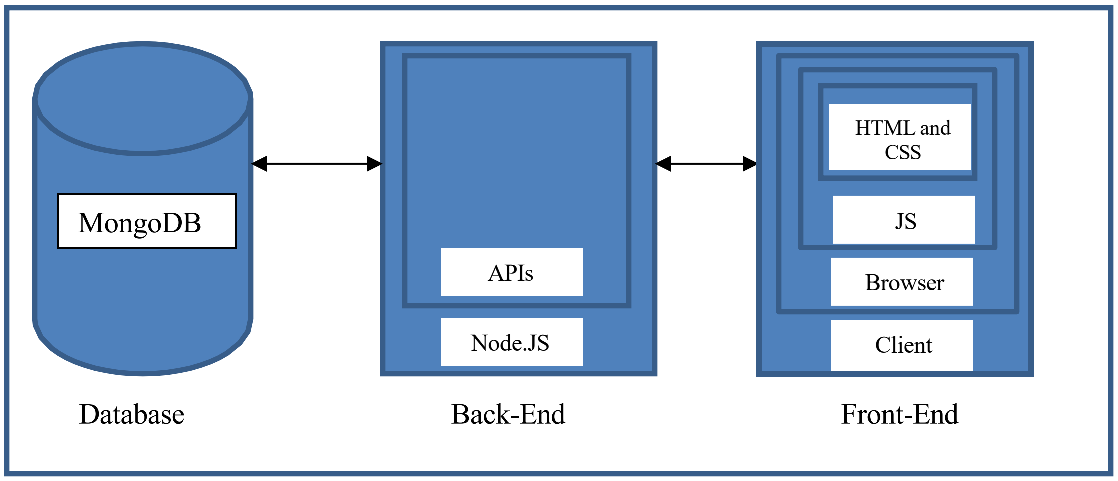
1.5 Definitions and Acronyms

AWS: Amazon Web Services

FAQ: Frequently Asked Questions

URL: Uniform Resource Locators

2.0 SYSTEM OVERVIEW



AWS

FIGURE 1

The above diagram represents the architectural structure that is suitable for RentO. Before we go into details, the reason we chose MongoDB, Node.JS and JavaScript as our main tools is because these were taught in the Odin Project, the website that teaches people how to become a web developer, and since these are all compatible with the Amazon Web Services, it will be easier to focus on these three. And we will have the web hosted with AWS so that we don't need to worry about having a server to run or deal with any additional maintenance and costs.

RentO will have various data that needs to be stored, including accounts, items that are being leased, images, transactions, payment information, and many more. So that is why we need a database that will be able to store that kind of data and not create any hassle for the users to reupload the same information repeatedly. This will also help maintain proof, if an error has occurred, and we are able to locate the data in the database such as missing rental item, fraud, order history, and reviews.

Next is the back end of the website and this will be the unit that holds together the database and front-end. The back end can pull data from the database and display it to the front end, and make any necessary calculation such as sales tax, total price of the items being rented, and the actual review of the user based on the feedback they received from other users. It can also use a Sort method to find the items the user requested to search and display it to the front end.

Finally, the front end will be accessed by the users so they can use our website and shop around to find the item they want to rent or make money and lease their items. This will have the user experience, including the layout and UI design, easy and simple access to the website's features and display the necessary information from the database.

3.0 SYSTEM ARCHITECTURE

3.1 Architectural Design

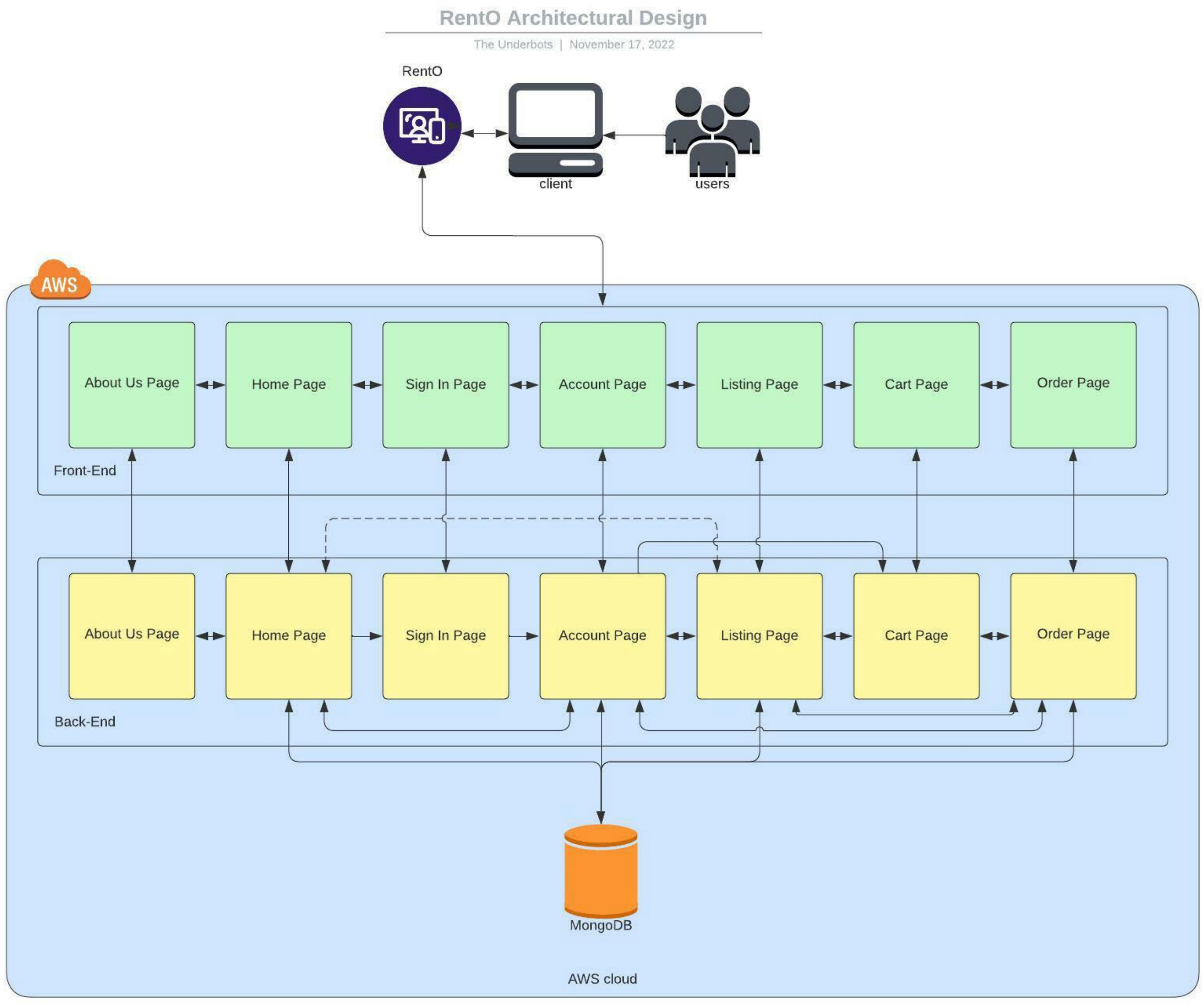


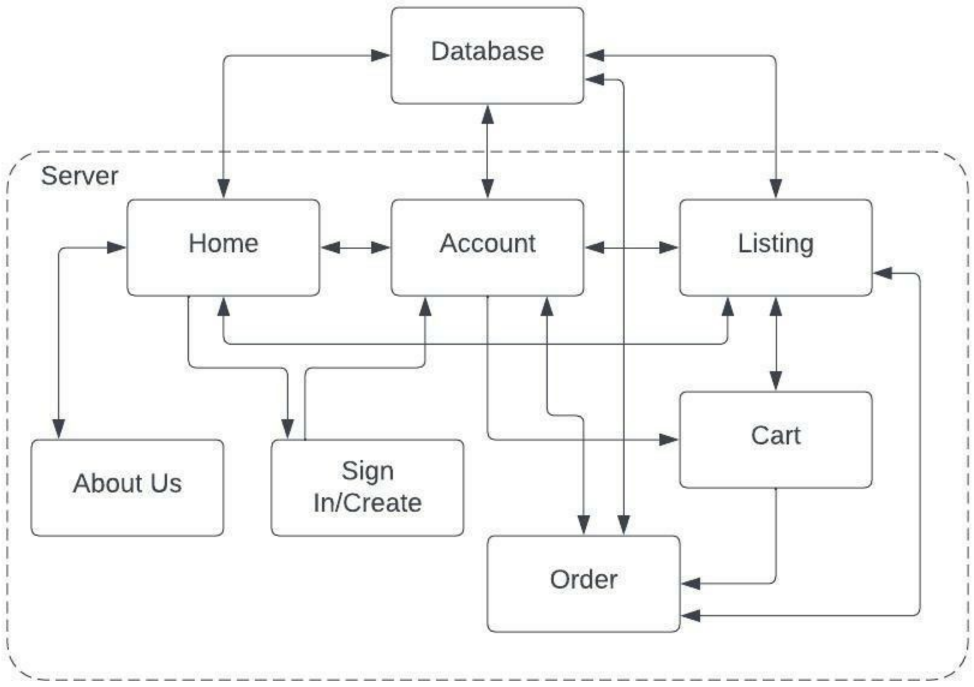
Figure 1. RentO Top Level Architecture

Figure 2 above displays a top-down description of how our ecommerce is expected to work and how the web will interact with each other and the user. Amazon Web Services will host both the front end and the back end, as well as the database. While the user can enter any location of the website, using the correct URL, the main component would be the account page. Without the User having a proper account page, they are blocked off from using the full features of the website. They can look around and see what item is up for rent, they can't rent it, message the user, or put one of their own up for lease. Sign in Page will be the component used to create or sign into an account and be able to use the website to its full abilities. Home page is the default site what the users sees when they enter the website, it will have certain items listed and gives the user options to do different tasks. The Listing Page will be used to display a specific item, including the price to rent, the location, who is renting, and many more, giving details to the user if they decide to rent the item or not. It is also where leasers can edit or delete their listings if they choose to. Cart Page is used to hold the items that the user wants to rent out and is able to make the transaction using one of their saved payment methods or add a new one. Order Page is used for transaction history, including what you have rented in the past and any current rentals. For the leaser, it will show the items they own that is being rented now and the history of rentals their items has amounted. Each page handles the data via the database, meaning each function within the page will have access to the data, make any changes, if necessary and either update it or leave it as it is. Users will interact with the front end of each page while the back end will do the rest.

3.2 Interface Design

- External Interfaces
    - Proper handling of sensitive information
    - Easy use of input, such as mouse and keyboard and touchscreen
    - Easy way to change website from Desktop to Smartphone mode
    - Communication of MongoDB for data storage and accessing
    - Sales and Tax system
    - Email Communication System
    - Communication of NodeJS for data management
- Internal Interfaces
    - Client-Server
        - The Client sends:
            - Users will access the homepage, create an account via the sign in page and begin listing the items they want to lease, in the listing page, and create a payment method in the Account Page, to receive payment.
        - The Server will:
            - Direct the user to the homepage and then to the sign in page for account creation.
            - Once Account is created, Access to features available with an active account.
                - Create listing
                - Create Method of Payment
                - Enter address and contact information

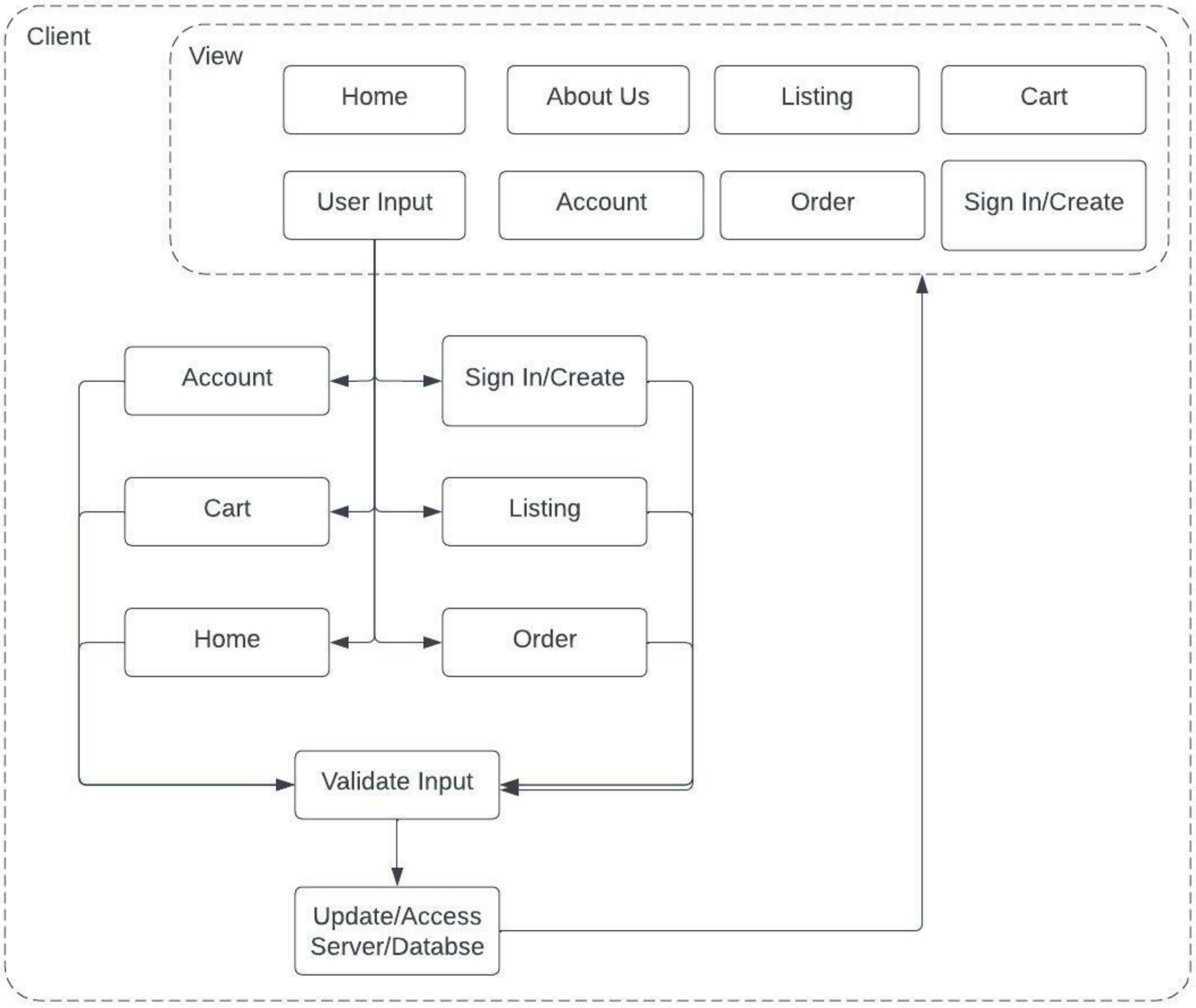
3.3 Decomposition Description



- A client will use the sign in/create component to either log in or create a new account.
  - If an account was created, data such as email and password, as well as a unique ID is stored in the database.
  - Otherwise, the client will have access to their information, which is pulled from the database and displayed in the appropriate web page.
- A client may view or update their Account in the Account page, which will be stored in database.
  - Server will store any information such as email, phone number, address, and payment method to the database.
- Client will use the listing to view an item or create their own
  - Stored listings are pulled from the database
  - Any newly created listing will be stored into the database
    - If unique ID of item matches to Unique ID of account, Account is able to

edit or delete listing.

- A client can view items they added to the cart and decide to make the final transaction.
    - Server will pull stored payment option from database and other information
    - Once an item is rented, the server will mark the listing as unavailable and hide it so it can't be viewed from a client, unless it is the owner or the client renting it.
- A client can access the order component and see information such as transaction history, and what payment options were used
    - Details is pulled from the database
- A client can access the Home and About Us component to see other listings or contact customer support



- When a client enters the website, the client displays the appropriate data from the database, based on the web page.
- Validate Input makes sure that the data entered is valid, such as credit card number or name of user with no number.
    - Once it is approved, it will update the database or use the data to search and access listings or other data.
- User Input includes mouse and keyboard, as well as touchscreen, data entry, and buttons
- Update/Access Server/Database will either update the database or server with new data or pull the

information and update the webpages for the user to view.

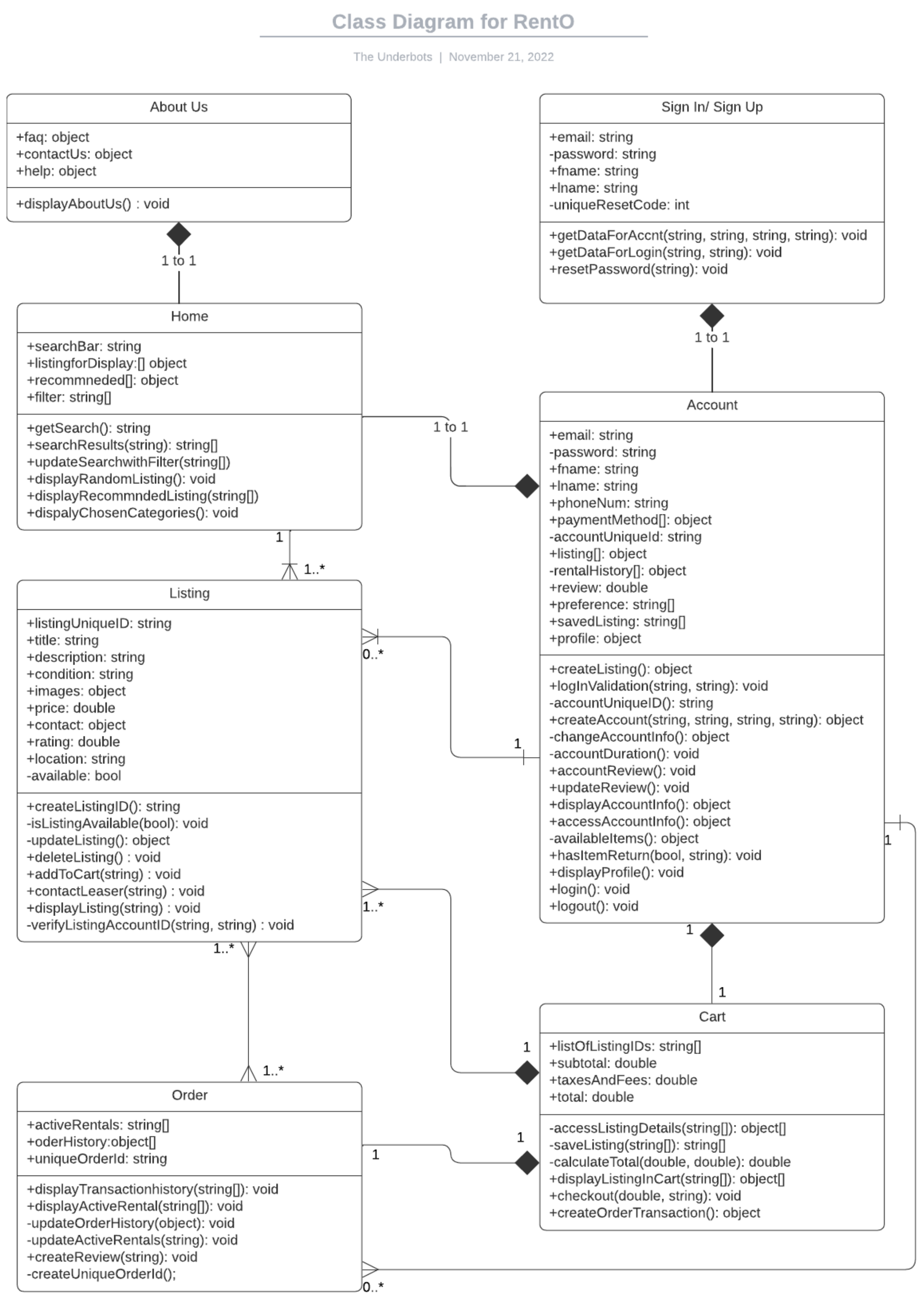
3.4 Design Rationale

The reason we chose this architecture is because we wanted to find one that was ideal for our ecommerce. While the examples in Canvas proved useful, we didn't think it would be suitable for our project. So far examples include a web game and a travel planner. The architecture didn't fit our online commerce style. Thus, we went with the one used in 3.1 as it summarizes how ecommerce would work based on the web pages, and the methods each will contain, to make the web pages perform their purposes. It is possible that we might be missing some useful features because of the architecture, but we are able to step back and reevaluate which architecture might be suitable. We believe our ecommerce is simpler compared to the examples we were provided with and thus could use a simpler architecture.

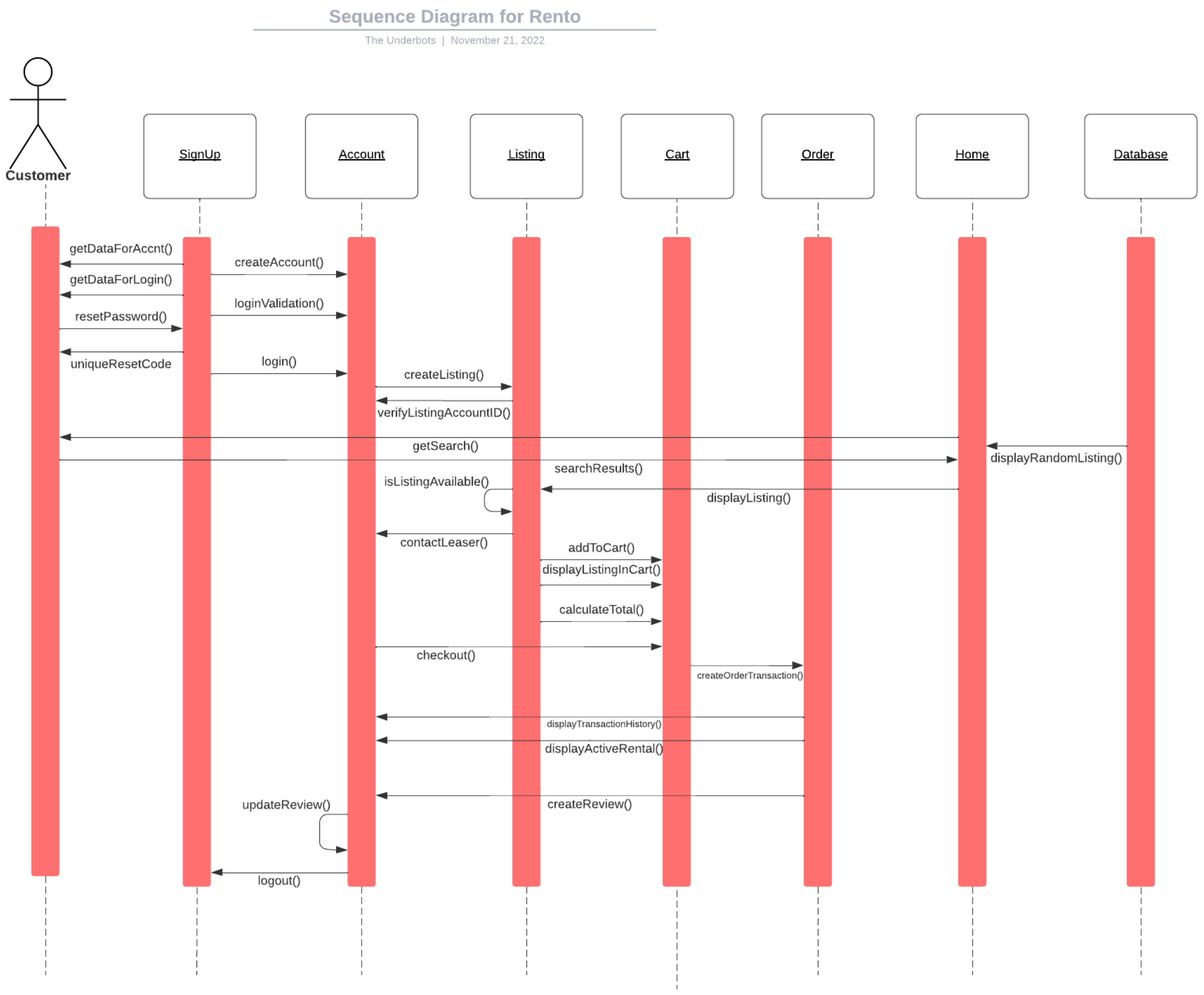
4.0 COMPONENT DESIGN/DETAILED DESIGN

The purpose of each software component, including its development status and planned utilization of computer hardware resources is discussed in this part of the SDD report. RentO is a website that is handled by a front end and back-end component, both which are hosted on the Amazon Web Host site. While the front-end deals with the display of the data and receives the input from the user, the back end is where the main operation occurs, storing, updating, and accessing data and using functions to calculate mathematics situation or making sure the necessary data is accessed and properly displayed for the user. Users will interact with the front-end while the back end does everything else and displays it to the user. Depending on the web pages, some data will be transferred using the necessary method and access the data from the database. The class diagram will have both the front end and back end, for easier understanding.

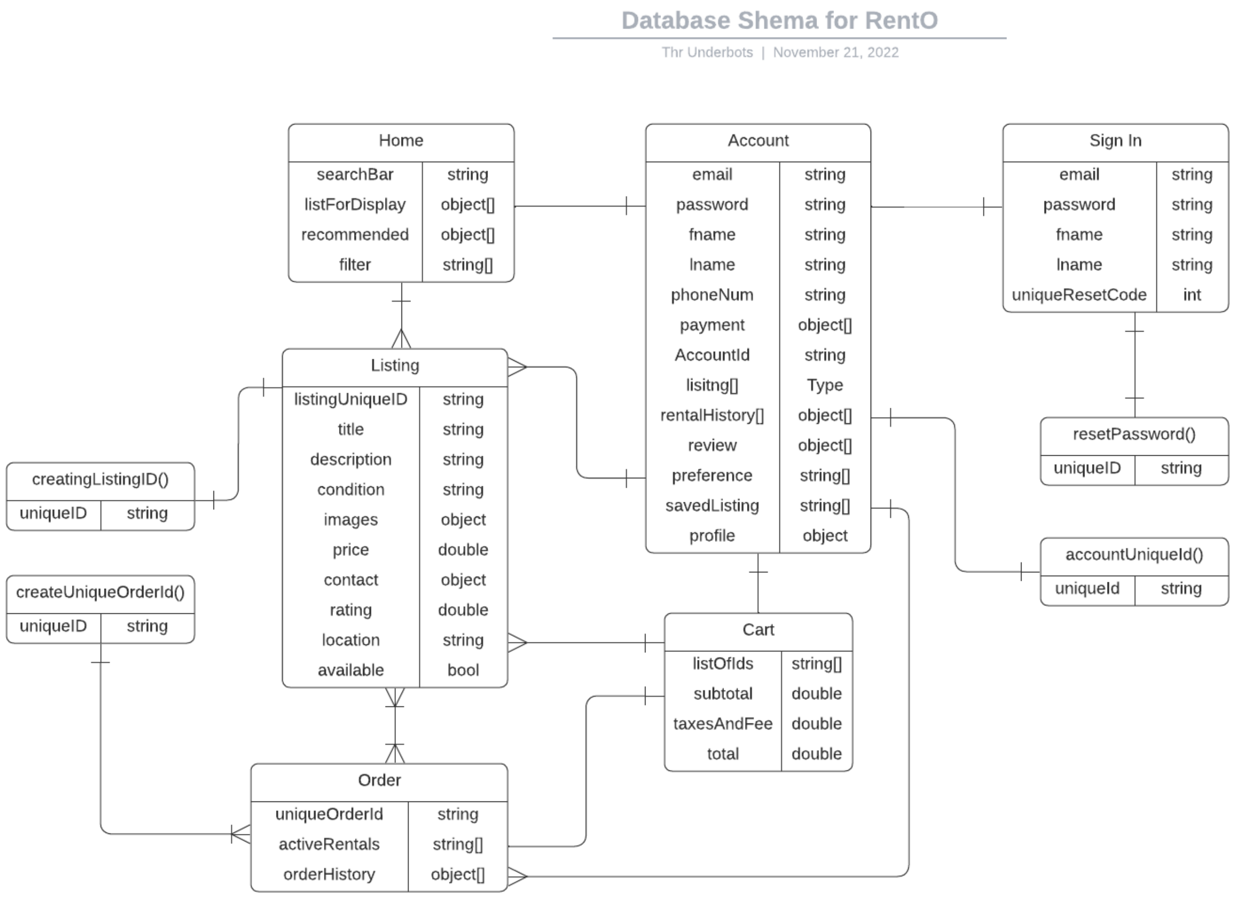
4.1 Class Diagrams


Class Diagram for RentO
The Underbots | November 21, 2022
About Us
+faq: object
+contactUs: object
+help: object
+displayAboutUs() : void
1 to 1
Sign In/ Sign Up
+email: string
-password: string
+fname: string
+lname: string
-uniqueResetCode: int
+getDataForAccnt(string, string, string, string): void
+getDataForLogin(string, string): void
+resetPassword(string): void
1 to 1
Home
+searchBar: string
+listingforDisplay:[] object
+recommneded[]: object
+filter: string[]
+getSearch(): string
+searchResults(string): string[]
+updateSearchwithFilter(string[])
+displayRandomListing(): void
+displayRecommndedListing(string[])
+dispalyChosenCategories(): void
1 to 1
Account
+email: string
-password: string
+fname: string
+lname: string
+phoneNum: string
+paymentMethod[]: object
-accountUniqueId: string
+listing[]: object
-rentalHistory[]: object
+review: double
+preference: string[]
+savedListing: string[]
+profile: object
+createListing(): object
+logInValidation(string, string): void
-accountUniqueID(): string
+createAccount(string, string, string, string): object
-changeAccountInfo(): object
-accountDuration(): void
+accountReview(): void
+updateReview(): void
+displayAccountInfo(): object
+accessAccountInfo(): object
-availableItems(): object
+hasItemReturn(bool, string): void
+displayProfile(): void
+login(): void
+logout(): void
1
1..*
Listing
+listingUniqueID: string
+title: string
+description: string
+condition: string
+images: object
+price: double
+contact: object
+rating: double
+location: string
-available: bool
+createListingID(): string
-isListingAvailable(bool): void
-updateListing(): object
+deleteListing() : void
+addToCart(string) : void
+contactLeaser(string) : void
+displayListing(string) : void
-verifyListingAccountID(string, string) : void
0..*
1
1..*
1
1
1
1..*
1..*
Cart
+listOfListingIDs: string[]
+subtotal: double
+taxesAndFees: double
+total: double
-accessListingDetails(string[]): object[]
-saveListing(string[]): string[]
-calculateTotal(double, double): double
+displayListingInCart(string[]): object[]
+checkout(double, string): void
+createOrderTransaction(): object
1
Order
+activeRentals: string[]
+oderHistory:object[]
+uniqueOrderId: string
+displayTransactionhistory(string[]): void
+displayActiveRental(string[]): void
-updateOrderHistory(object): void
-updateActiveRentals(string): void
+createReview(string): void
-createUniqueOrderId();
1
1
0..*

4.1.1 Sequence Diagram of the applicable Base Class (4.1, 4.2, 4.3, etc)



4.1.2 Database Schema of the applicable Base Class (4.1, 4.2, 4.3, etc)



5.0 HUMAN INTERFACE DESIGN

5.1 Overview of User Interface

When users enter the website using the URL, they will be shown the RentO homepage, where several random items, by default or based on preference, are shown, including images, and depending on if it’s a desktop or mobile app, the price and title of the item. From there they have several options, they can sign into their account, unless they are already signed in, prior to entering the site, they can click on one the items to see the listing or go to their account settings. There are a few more options, but these are just examples. In the sign in page, they can login into their account, using their email address and password they used when they created the account. Or if they are brand new users, they can sign up using the sign-up page and add the details to create and account and use all our features. Section 6.2 shows details of how it looks like.

5.2 Screen Images

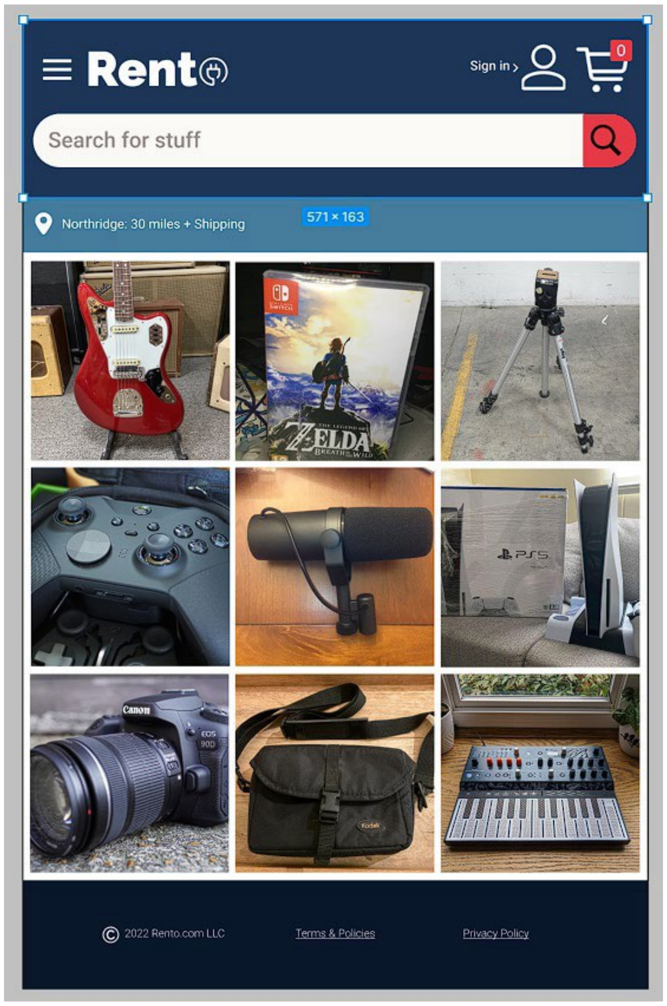
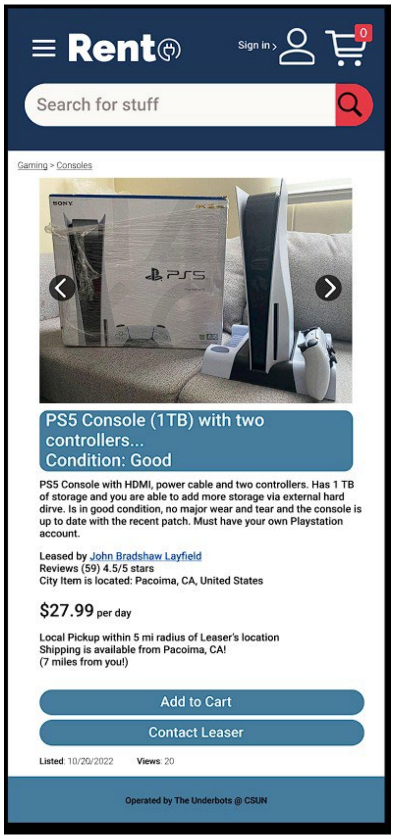


Figure 5.1 shows the mobile version of our homepage, showcasing different items for rent, ranging from a console to cameras and music equipment. You can click any of them or choose to search for a specific item or category, using the search bar or tapping on the three-hamburger bar and choosing what you want to see. Users can sign in or check their cart and see what they have, ready to be rented.



PS5 Console (1TB) with two controllers...
Condition: Good

PS5 Console with HDMI, power cable and two controllers. Has 1 TB of storage and you are able to add more storage via external hard dirve. Is in good condition, no major wear and tear and the console is up to date with the recent patch. Must have your own Playstation account.

Leased by John Bradshaw Layfield
Reviews (59) 4.5/5 stars
City Item is located: Pacoima, CA, United States

$27.99 per day

Local Pickup within 5 mi radius of Leaser's location
Shipping is available from Pacoima, CA!
(7 miles from you!)

Listed: 10/20/2022 Views: 20

Operated by The Underbots @ CSUN

Figure 5.2 displays an item that is being leased, in this case a PS5 console. The user is shown details, including the price, condition, description, leaser’s details, and various options from adding it to the cart or contacting the person to ask more questions. Users are shown different ways to obtain the item, such as local pickup and shipping, with a fee included from the leaser.

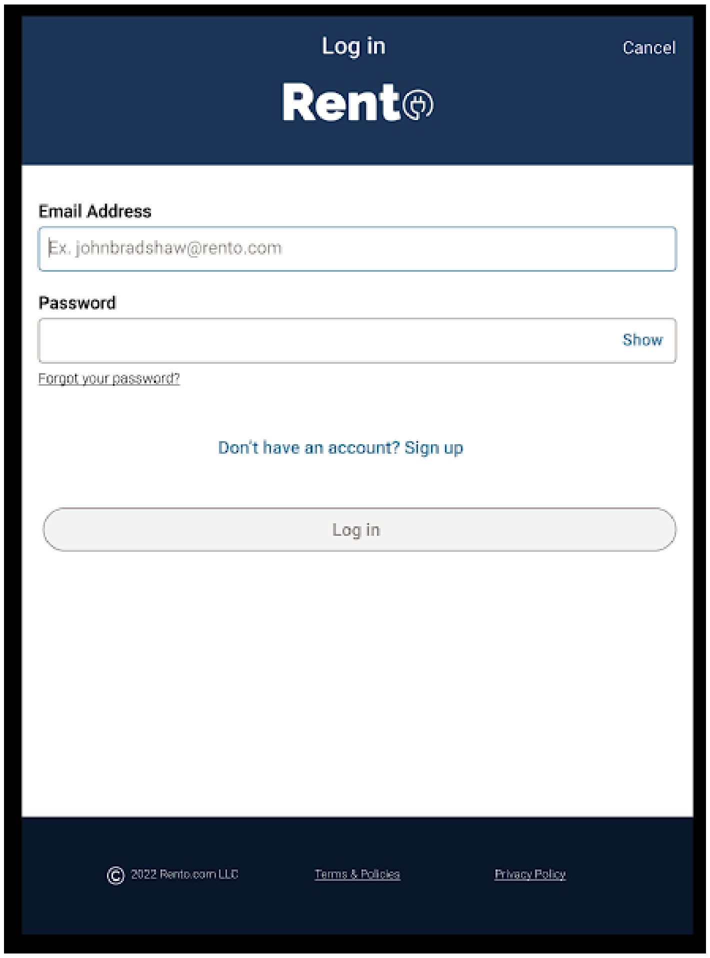
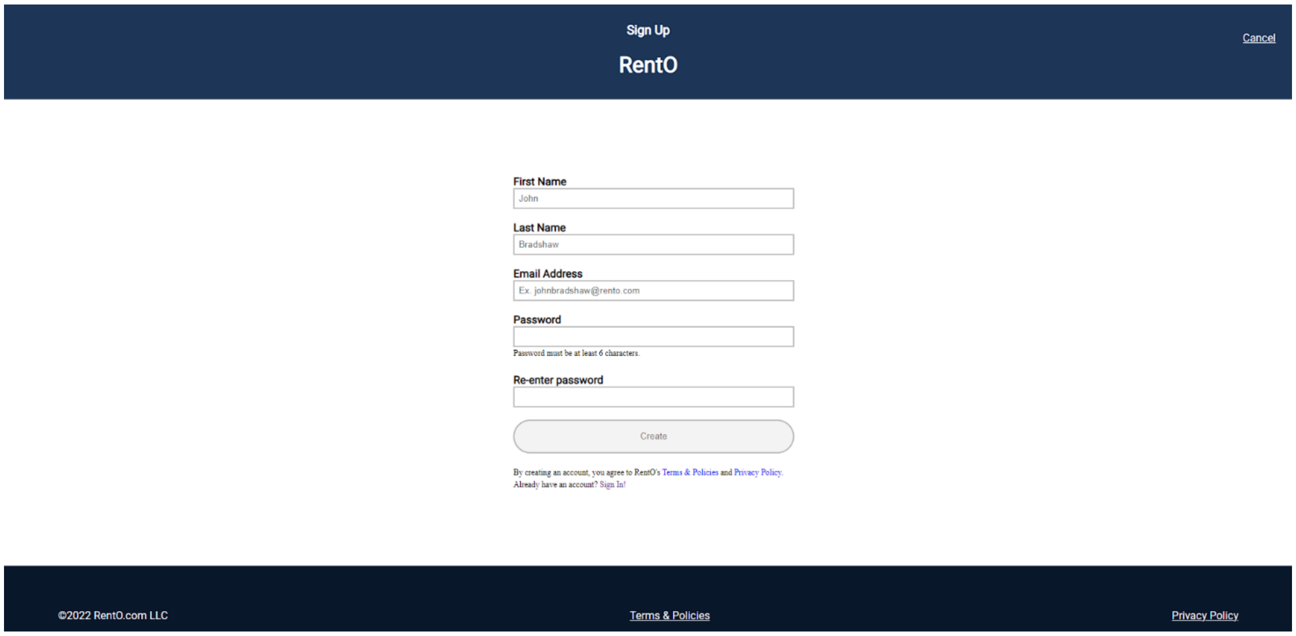


Figure 5.3 shows Users being able to login, with their email and password. They are offered the option to reset their password via the "Forgot your password" link and also to create an account if they don't have one yet. If they decide to change their mind, they can go back to the previous page, using the cancel link in the top right corner of the page.



Sign Up

RentO

Cancel

First Name

John

Last Name

Bradshaw

Email Address

Ex. johnbradshaw@rento.com

Password

Password must be at least 6 characters.

Re-enter password

Create

By creating an account, you agree to RentO's Terms & Policies and Privacy Policy.
Already have an account? Sign In!

©2022 RentO.com LLC

Terms & Policies

Privacy Policy

Figure 5.4 displays the creation of an account page. In the event the users want to create an account, they will be directed here to enter basic information, such as their full name, email address, and a password and once these are validated and approved, will create an account for them to use.

5.3 Screen Objects and Actions

Figure 5.1:

Search bar: Enter any phrase or string of letters and the server will find items based on the string of characters. The more specific, the accurate the search result will be. Also, it is possible to broaden the search by typing in the category or manufacturer of items. If the value of string does not match any of the items, will let user know, no item is displayed based on what the user typed. Object will be *searchItem.*

Images: Tapping on the images will make the server pull up the necessary data from the database to make the appropriate listing for that item.

Sign In link: Clicking or tapping on this link will take the user to the sign in page, where they need to input the correct values, to access their account, or to create a new one.

Cart button: This will direct the user to the cart page, where they can see their items and verify if they want to rent them or not.

Figure 5.2:

Search bar: Enter any phrase or string of letters and the server will find items based on the string of characters. The more specific, the accurate the search result will be. Also, it is possible to broaden the search by typing in the category or manufacturer of items. If the value of string does not match any of the items, will let user know, no item is displayed based on what the user typed. Object will be *searchItem.*

Images: Tapping on the left or right side of the image will make the server to switch to a different image of that item, if the leaser added them to the database, via their account.

Sign In link: Clicking or tapping on this link will take the user to the sign in page, where they need to input the correct values, to access their account, or to create a new one.

Cart button: This will direct the user to the cart page, where they can see their items and verify if they want to rent them or not.

User link: This will direct the user to the leaser's profile page, where they can see their reviews and any items they have up for rent. Also displays their contact information from the database.

Add to Cart button: Adds the unique ID and basic information of the item to the cart so that it is ready to be rented and has the item pulled out from the website (if they finalized the rental), until it is ready to be leased again.

Contact Leaser button: Pulls up the email system from the backend and creates an email ready to be sent to the leaser's email, using the user's email address.

Figure 5.3:

Cancel bar: Directs user to the previous page if they decided to not make an account.

Email input: Prompts user to enter email address, which can be as low as 7 characters, including the "@", "period", and the domain type, and whatever they have in their email address. Value will be validated to see if it's a legit email address or they need to type a different input. If false, then will be prompted to type a different email, and if true, along with the password, will be checked to see if it matches the data in the database. Object will be *email.*

Password input: Prompts user to enter password, which can be as low as 6 characters, and may include whatever the user prefers to be and follows industry standards. Input will be checked to see if it matches the one stored in the database, along with the email address. If it is true, the user has access to an account, if not, then user needs to type the correct email and password again. Object will be *psswrd.*

Forgot Password link: Directs user to forgot password page, where they need to enter their email address and get a one-time code sent to their email and must enter the code in that webpage to make sure it is them that owns the email account. Soon after, they can reset their password and update it in the database.

Sign Up link: Directs the user to the create account page, where they need to enter several strings to create an account, such as first and last name, email, and password.

Log In button: Will transfer the data entered in the email and password form box to the backend and compare it to the one stored in the database. If both the email and password match, the user will access the account. If not, it will prompt the user that either one is incorrect or needs to try again.

Figure 5.4:

Cancel bar: Directs user to the previous page if they decided to not make an account.

First Name: Prompts the user to type in their first name, which can range from 1 character to 20 if they have a long or short name. No numeric or special characters is allowed and once it is validated from a special function, will be stored in an object named, *fname*.

Last Name: Prompts the user to type in their last name, which can range from 1 character to 20 if they have a long or short name. No numeric or special characters is allowed and once it is validated from a special function, will be stored in an object named, *lname*.

Email input: Prompts user to enter email address, which can be as low as 7 characters, including the "@", "period", and the domain type, and whatever they have in their email address. Value will be validated to see if it's a legit email address or they need to type a different input. If false, then will be prompted to type a different email, and if true, along with the password, will be checked to see if it matches the data in the database. Will be stored in the object called, *email*. Will be checked to see if email is valid by following email standards.

Password input: Prompts user to enter password, which can be as low as 6 characters, and may include whatever the user prefers to be and follows industry standards. Input will be checked to see if it follows proper security standards, If it is acceptable, the password will be stored in an object called *psswrd*.

Re-enter Password field box: Prompts user to re type the same password to make sure it is the same one they know and must remember. This will be stored in *rstpsswrd* and will be compared with *psswrd* to see if they are the same. If not, they need to retype either the password value or the reset password value to fix the issue.

Create button: Will transfer the data entered in the previous text boxes to the backend and store in the database. An account object will be created, filled with the new data, and will be given a unique ID identifier, in the event the user changes emails.

6.0 REQUIREMENTS MATRIX

Requirements	Description	Location in SDD	Details
FUNC_WEB_001	Shall have a button that contains our logo and when clicked, takes the user to the home page.	Section 5.2	Figure 5.1
FUNC_WEB_002	Shall have a search function to find the specific item or a type of item the user is interested in.	Section 5.3	Search Bar description
FUNC_WEB_003	Shall have a drop-down menu that will show the assorted options to choose from.	Section 4.1	Class Diagram (Home)
FUNC_WEB_004	Shall have a filter system that will narrow the search results based on the user's choice of preference.	Section 4.1	Class Diagram (Home)
FUNC_WEB_005	Shall have a function that will be used to access and acquire information from the database of RentO	Section 3.3 & 2.0	Decomposition Description and System Overview
FUNC_WEB_006	Shall have a mock system to showcase how users will pay for the rental and how leasers will get paid.	Section 4.1	Class Diagram (Cart)
FUNC_WEB_007	Shall have an email system, where the users are able to communicate with each other.	Section 4.1 & 5.2	Class Diagram (Listing) & Figure 5.2
FUNC_WEB_008	Shall have a two-way review system, where users are able to review the products, they rented and each other.	Section 4.1	Class Diagram (Account & Order) & Sequence
FUNC_WEB_009	Shall have a cart button that would take the user to the page where it will display the items they added.	Section 4.1 & 5.2	Class Diagram (Cart) & Figure 5.2

FUNC_WEB_010	Shall have a checkout page that will show the total costs of the rentals and ask for the payment method. Order will be added to the Order History page.	Section 4.1	Class Diagram (Cart) & Sequence Diagram
FUNC_WEB_011	Shall have an order button, where it will display the user the items they had rented or is currently being rented.	Section 4.1	Class Diagram (Order)
FUNC_WEB_012	Shall have an account button, where it will display the Account page, in which the user can access, edit and perform various tasks in regard to their account.	Section 4.1	Class Diagram (Account)
FUNC_WEB_013	Shall have the footer section that will have useful information, including Contact Us, FAQ, and About us.	Figure 5.2	Figure 5.1 to 5.4
FUNC_WEB_014	Shall have a Back to the Top button for easy access or the users.	N/A	N/A
FUNC_ACC_001	Shall have an account system where the user's data is stored and can be accessed via email and password.	Section 4.1	Class Diagram (Account & Sign in)
FUNC_ACC_002	Shall have a login and password system to maintain proper security and privacy for the user.	Section 4.1	Class Diagram (Account & Sign in)
FUNC_ACC_003	Shall have a unique identifier for the account system so no two accounts are the same.	Section 4.1	Class Diagram (Account)
FUNC_ACC_004	Shall allow the user to create an account by filling out information, such as email, password, and name.	Section 4.1	Class Diagram (Account & Sign In
FUNC_ACC_005	Shall have a function for the user to properly login and logout of their account.	Section 4.1	Class Diagram (Account)
FUNC_ACC_006	Shall have a review system in the Account page, where the users see how many reviews they received and what is their current star rating.	Section 4.1	Class Diagram (Account & Order)
FUNC_ACC_007	Shall have a settings function to allow user change details to their account, including email address, name, contact information, and address.	Section 4.1	Class Diagram (Account)
FUNC_ACC_008	Shall have a page where the user is able to setup a listing for their items for rent, including adding	Section 4.1	Class Diagram (Account)

	information and images.		
FUNC_ACC_009	Shall display a page of the user's inventory, including what they have available and what is still being rented.	Section 4.1	Class Diagram (Account)
FUNC_ACC_010	Shall have a function that allows the user to notify the website that they have received their items and is either ready to relist it or make some changes to the details or delete the listed item.	Section 4.1	Class Diagram (Listing & Order)
FUNC_HOME_001	Shall have a listing of the items that can be rented on the homepage. This will show assorted items, either the ones that the user prefers or just random items to see if the user is interested.	Section 4.1 & 5.2	Class Diagram (Home) & Figure 5.1
FUNC_HOME_002	Shall have a section for recommended items based on what the user has searched or rented in the past. This will be a separate listing compared to FUNC_HOME_001.	Section 4.1	Class Diagram (Home)
FUNC_CART_001	Shall have a cart page where it will show the items that the user wishes to rent out before making the final transactions.	Section 4.1	Class Diagram (Account & Cart)
FUNC_CART_002	Will have a function where the user is able to save the items listed in the cart for later use, so they do not have to worry about finding the item again.	Section 4.1	Class Diagram (Account & Cart)
FUNC_CART_003	Will have a checkout button that will link the user to the checkout page, which is FUNC_WEB_010.	Section 4.1	Class Diagram (Cart) and Sequence
FUNC_ORDER_001	Shall have an Order History page where the user can see the transactions they have made and what item they are currently renting.	Section 4.1	Class Diagram (Order)
FUNC_LIST_001	Shall have a Listing Page where more details are being displayed of an item the user has clicked, including the photos, condition, description, and star rating.	Section 4.1 & 5.2	Class Diagram (Listing) & Figure 5.2
FUNC_LIST_002	Shall have an Add to Cart button, to allow the user to place the item they wish to rent into a cart for storage until they go to the checkout page.	Section 4.1 & 5.2	Class Diagram (Listing) & Figure 5.2

FUNC_LIST_003	Shall have a Contact Seller section for the user to have access in case they need to communicate with the leaser and ask questions or comments.	Section 4.1 & 5.2	Class Diagram (Listing) & Figure 5.2
FUNC_LIST_004	Shall have a function in the Listing Page where it will display if the item is available for rent or is out for rental.	Section 4.1 & 5.2	Class Diagram (Listing & Order) & Figure 5.2
FUNC_LIST_005	Shall have a function where if the unique ID of the item matches with the unique ID of the leaser, they are able to see a button to edit the listing, such as images, details, delete the listing. If it does not match, the button will be hidden.	Section 4.1	Class Diagram (Listing)
FUNC_SIGNIN_001	Shall have a Sign In/Create page where FUNC_ACC_004 will be used to allow a user to create an account for the RentO site. This will also use FUNC_ACC_002 to have the user login using an email address and password.	Section 4.1 & 5.2	Class Diagram (Sign In) & Figure 5.3 and 5.4
FUNC_SIGNIN_002	Will have a function that allows the user to reset their password in case they have forgotten via email.	Section 4.1	Class Diagram (Sign In / Up)
FUNC_SIGNIN_003	Will have an email verification system where the user will receive an email containing a unique code that must be entered in our website to verify their email address.	Section 4.1	Class Diagram (Sign in / Up)
FUNC_CONTACT_001	Shall have a page dedicated to Contact Us, Help and FAQ sections for the user to have in order to contact us or seek assistance.	Section 4.1	Class Diagram (About Us)