Rent-It
System Requirements Specifications

CSE686- Object Oriented Design

Professor Edmund Yu

October 17, 2020

Team Members:
Sahil Sehgal
Aditi Mallesh
Tejas Mehta
Nikhil Gohil
Akshay Mahajan

Table of Contents

1. INTRODUCTION

Purpose:

Rental applications have emerged in the past and are now spreading across all businesses. Rental items help users to save money by eliminating services and maintenance cost. Reading about the piece of information about the product does not always provide you the actual picture of what it will be like to use it. Renting gives access to try new items/products before making a buying decision. Rental cars, books, mountain bikes, furniture, electronics and construction equipment are some of the common items which can be rented. There are multiple sites available for each of these items.

As a user, to search for the specific items with given requirements will take a long time and sometimes the user doesn't get the expected results. Our aim is to provide a single platform where they search for the items like cars, books, furniture, electronics and construction equipment etc. on rent on a daily, weekly or monthly basis. A platform like this will benefit a wide range of customers in saving money by eliminating service and maintenance costs and earning money by renting out from items which are not in daily use.

Product Scope:

Our main idea is to connect people through an application where they can post their items on rent and can also search for other items which they might want to take on a daily, weekly or monthly rental basis. Therefore, we are proposing to create a web application which will cover end to end transaction flow of how customers can publish and rent items.

Any user can search for items and can look for different prices and options available. In order to take that item on rent, he or she needs to create an account using a mobile number and an email ID. Once account creation is completed, users can put items to cart and can place an order.

The second flow is where users can upload their items. In order to upload, a user first needs to create an account. There will be options to attach pictures and specify daily, weekly and monthly rental price of the item.

There are many websites available for specific products or items. But there is not a single platform which can give any product with just a single search. Our goal is to integrate these ideas and come up with a solution which acts as a single platform that can be used to either sell or rent any of these items. The is an application created to provide a single bridge between users and the product they want. Since this solution is accommodating all types of items be it cars, books, dresses or furniture, the trend of renting or lending is likely to continue for a very long time. Hence our application is flexible in terms of items included and can be useful for a broad range of customers.

Definitions, Acronyms and Abbreviations:

1. Account : Every User has to create an Account in-order to access the web-application
2. Consumer : User who want to rent a product of interest based on its availability
3. Notification : Once the product is rented, and order is placed, both Publisher and Consumer will receive an email notification

4. Order : An order consist of Product detail, Publisher and Consumer info including Start an End date for an ongoing or previous transaction
5. Publisher : User who can add/modify/delete a product which a Consumer can rent.
6. Product : An Item which is owned by a Publisher that can be rented by a Consumer

2. OVERALL DESCRIPTION

Product Perspective and Functions:

As the advancement in technology continues, the dependency on it is equally proportioning. Number of people depending on online shopping is efficiently increasing. Our application will provide an efficient solution by combining multiple products into a single platform. The users can assess the items by simply searching for them. The website will display a list of all the items that are currently available for rent. Users don't need to login or register to look for an item. However, in order to rent an item, they need to register with the application. The registration requires email id, password and contact details (includes home address, mobile number). The items details are available based on their rent policy(hourly, weekly or monthly basis).

The application is also giving a platform for users who want to lend the items to make money. There is an option available for lending the items. This feature is available once the user is registered with an application. Registered users can upload their items. The users need to provide the pictures of the items that they want to lend and also specify daily, weekly and monthly rental prices of the item. If the user is not previously registered, then he/she would be directed to the registration page.

Another application feature includes a recommendation system. The application will recommend the items based on the user's previous search history and other parameters like zip code, location etc. Each registered user will get recommended products based on their history.

User Characteristics:

The web application will consist of two main user flows which includes Publisher flow and Consumer flow. The high-level overview of these flows provided by the system are:

Publisher flow:

1. Publishers can lend an item. First, they will have to create an account, if they are not previously registered.

2. Publishers can add new products with all the details. They can also manage the product by specifying all the details including products availability and prices.

3. Publishers can add, modify and delete the post at any time.

4. Publishers can check the transaction details, which display whether the product has been rented out or not along with the availability.

5. Publishers will get the notification once the consumer ordered the product.

Consumer flow:

1. Consumers who are not previously registered with the application, they can sign-up on the application by providing the email address, password and contact details.
2. Consumers who want to place an order need to register. Unregistered users will be first redirected to the registration page, if they want to rent out something.
3. Consumers can visit the application and see the list of all the items currently available for renting. They can also apply the filters while searching for an item.
4. Consumers can see the details of the items which includes the items availability and renting policy (hourly, weekly or monthly prices), if they click on the add to cart or like button, then they will be redirected to the order confirmation page.
5. Consumers can access their profile and can check their order history, upcoming order and liked items. They can also edit their profile and change the information.

Design and Implementation Constraints:

1. Key constraints of the system are that the system is inherently reliable on the availability of an active internet connection.
2. Since the app runs on a client-server architecture, it is of utmost importance to have a good internet connection
3. Another main constraint is regarding the dataset, for the recommendation system to work its best, we rely heavily on the closely linked database set, so user can have the more similar items in their recommendation

Assumption and Dependencies:

1. The system development depends on open-source control tools and it is assumed that the tools and framework used is always available.
2. The data provided by the publisher or consumer about the product is assumed to be true or valid. The application is not using any third-party app to validate the data.

3. INTERFACE REQUIREMENTS

User Interfaces

HTML, CSS, JavaScript are used by the web-application for user interface. The users of this web application can be anyone who wants to rent or buy the products. To accommodate users who want to upload items to sell/rent or users who are logging in as a customer, we will have two user flows.
Our aim is to deliver the web application which is feasible in such a way it demonstrates

flexibility with respect to User Interface, like images/resizable texts. This web application can be launched from web browser on mobile or tablet, PC, and we are following best practices with respect to web design and development. It is important to give best user experience with respect to maintaining the web application simple and consistent, including portability.

We have different use cases with respect to owner and renter and the flow varies respectively based on permissions with respect to each role. The user shall then select the respective roles and fill out the details according to the flow.
Basic flow for a publisher and consumer includes:

1. User creates an account where they can Log In/Signup
2. User profile page where users can see following things.
    a. Interested items
    b. Order history
    c. Upcoming or ongoing order details
3. Ability to edit user profiles and can change basic information
4. User will be treated as a Publisher, if he/she select Add Product
5. Publisher can add/update/delete a product
6. User will be treated as a Consumer if he/she want to rent a product
7. Homepage has listing of all the available products based on the zip code entered by the user
8. Search bar where consumer can search different products of interest
9. Product details page where users can view details of the product, availability of the same and can also add to cart.
10. Cart and checkout page where Consumer can place an order.

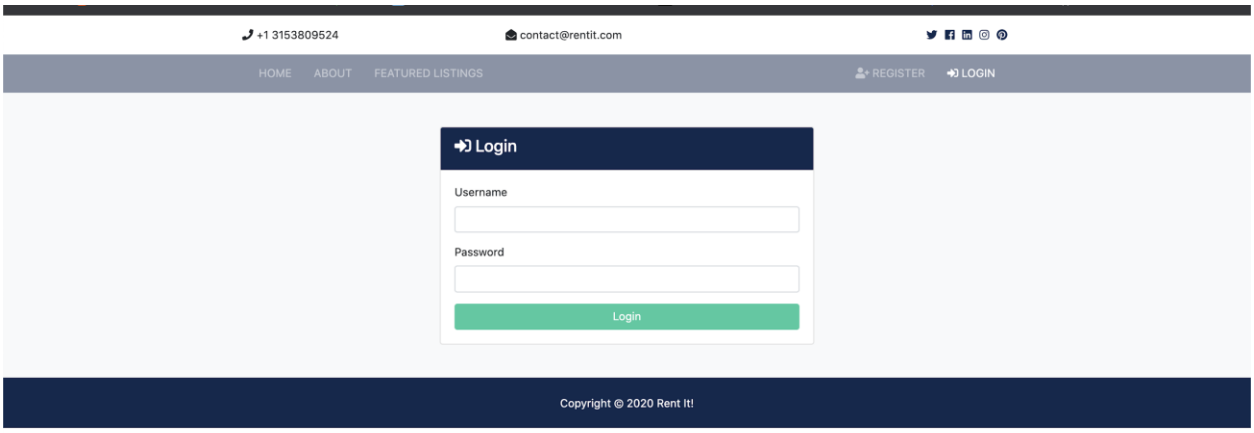

Figure 1. Login Page

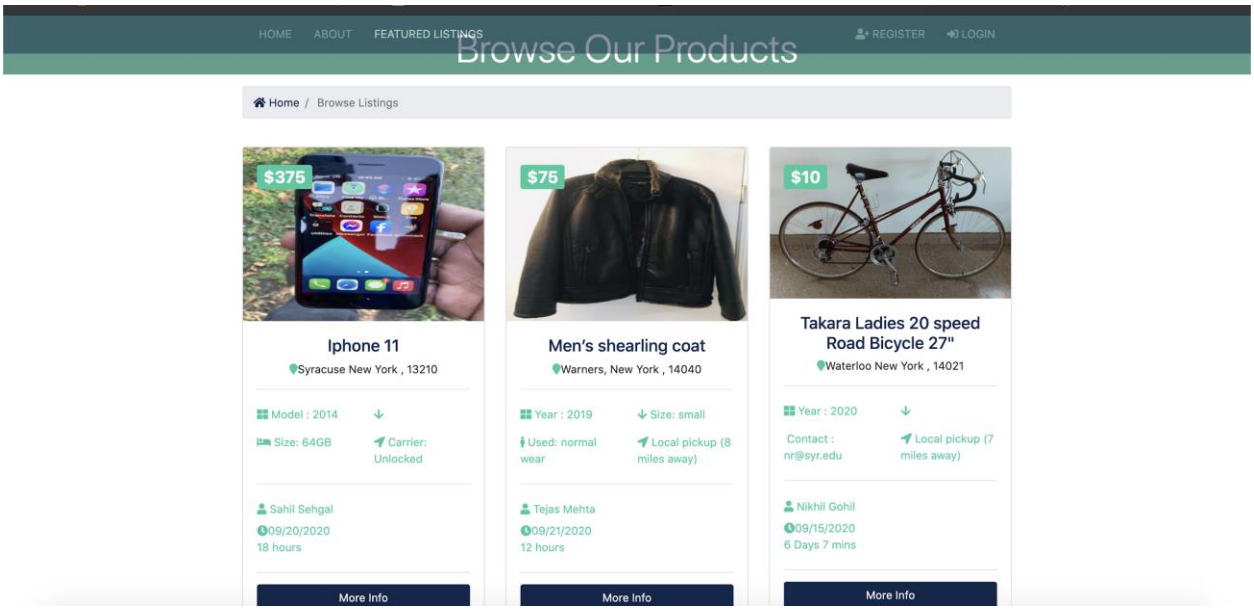

Figure 2. HomePage

Hardware Interfaces

The users will need a simple modern web browser to access the web application in computer, phone or tablet.

Software Interfaces

In order to create a highly scalable app, we will be using the following technologies throughout the Project:

1. Backend: Python Django Framework with object-oriented classes implemented
2. Database: PostgreSQL is used as it supports JSON data type for column values. Also, it is fast in elastic search implementation.
3. Frontend: Django template with server-side rendering for a faster performance.
4. Storage: Files will be stored on a third-party cloud like AWS S3 storage.
5. Access to users' location is enabled based on the zip code entered.

Communication Interfaces

1. Encrypted messages will be sent to ensure safe data transmission while communicating between the client and the server.
2. We have used HTTPS for all the communication.

4. Requirements

Functional User Requirements:

UR-1 Each User shall have a private account. If a User forgets his/her account or password, then he/she can easily retrieve them as well as be able to change the password.

UR-2 Users shall complete or update their basic personal information in their own private account page

UR-3 Ability for the Users to view the available products based on the location

UR-4 Ability for the Users to search the products they are interested in

UR-5 Ability for the Users to sort the data based on their criteria

UR-6 Ability for the Users to rent the selected items

UR-7 Ability for the Users to upload and manage the posts

UR-8 Ability for the Users to view the order history

Functional System Requirements:

F-1.1 System should be able to display the list of items based on category selected by the users
F-1.2 System should be able to provide list of products with search filters
F-1.3 System should be able to recommend the products based on the zip code

F-2.1 The database which stores the information of the user's account is retrievable whenever the User wants to.
F-2.2 The database storing the User information, is updated as and when the User profile is edited or password is Updated.
F-2.3 System shall have access to the user's current location by checking the user's Location Service on the Devices or through the zip code entered.

F-3.1 System shall display the data and give respective authorities to users based on their role(publisher/consumer)
F-3.2 System shall allow the users to edit the profile or edit the posts made by them

F-4.1 The app shall support communication between users. When someone posted a message or comments on the discussion forum, other users can check it immediately.
F-4.2 The system shall allow users to post and receive the real-time messages

F-5.1 The app shall extract the keywords searched by users in the search menu using natural language processing and returning the related/matched information in detail.
F-5.2 The database shall be updated immediately whenever User write a new post or leave a comment on a post

F-6.1 The system shall support the communication between the users, when any User show interest in a post or a product, it is made visible to the owner in real-time

F-7.1 The User shall receive a notification after registration
F-7.2 System should be able to send email notifications to both customers and the owners when an order is placed

Non-Functional User Requirements:

UR-1 The application shall be available to anyone with the access to internet at any time

UR-2 Privacy of the Users shall be maintained through separate login credentials and stored securely in the system

UR-3 System shall get the appropriate location information for each user based on the zip code entered by the Users

UR-4 The Application shall have two main roles for Users, as someone who owns the items that are posted, and someone who wants to rent the item

UR-5 The system shall be used by anyone without any formal training

UR-6 Users shall be able to restart the system if the application does not work properly or fails unexpectedly

Non-Functional System Requirements:

NF-1.1 The Users shall have access to the internet connection to launch this application
NF-1.2 Uses can launch the web application through a browser like Google Chrome or Firefox
NF-1.3 The Users can run this application on mobile or tablet or computer

NF-2.1 System should be scalable, resilient and fault tolerant
NF-2.2 System shall maintain the personal information of the users securely and not share it with any third party

NF-3.1 System shall match the keyword input by the User and return the correct results

NF-4.1 System shall offer the User with simple and user-friendly UI
NF-4.2 System shall offer Users with clear navigation buttons and best user experience with respect to User Interface

NF-5.1 System shall grant access to specific functions and features based on user roles.

NF-6.1 System shall be able to retrieve the user's data if they are lost

5. MODELS

Use Case Diagrams

Figure 3. Use Case Diagram

The above use-case diagram is a representation of the associations between application users (publisher and consumer) and the use cases:

Create Account : The user can create a profile on web application using registration form
Select User Role : The user can have more than one role, publisher/consumer
Add New Product : Publisher has ability to add new products that he/she wish to put on rent
Manage Products : Publisher has the ability to add/update/delete the post
View Transaction : Publisher can view the order details for respective products and update the availability dates on each item accordingly
Search Products : Consumer can search for the products based on the interest
View Products : The products will be displayed on the homepage based on the availability within the location, according to zip code entered by the user
Rent Products : Ability for the consumer to rent the product of the interest that is available

Activity Diagrams

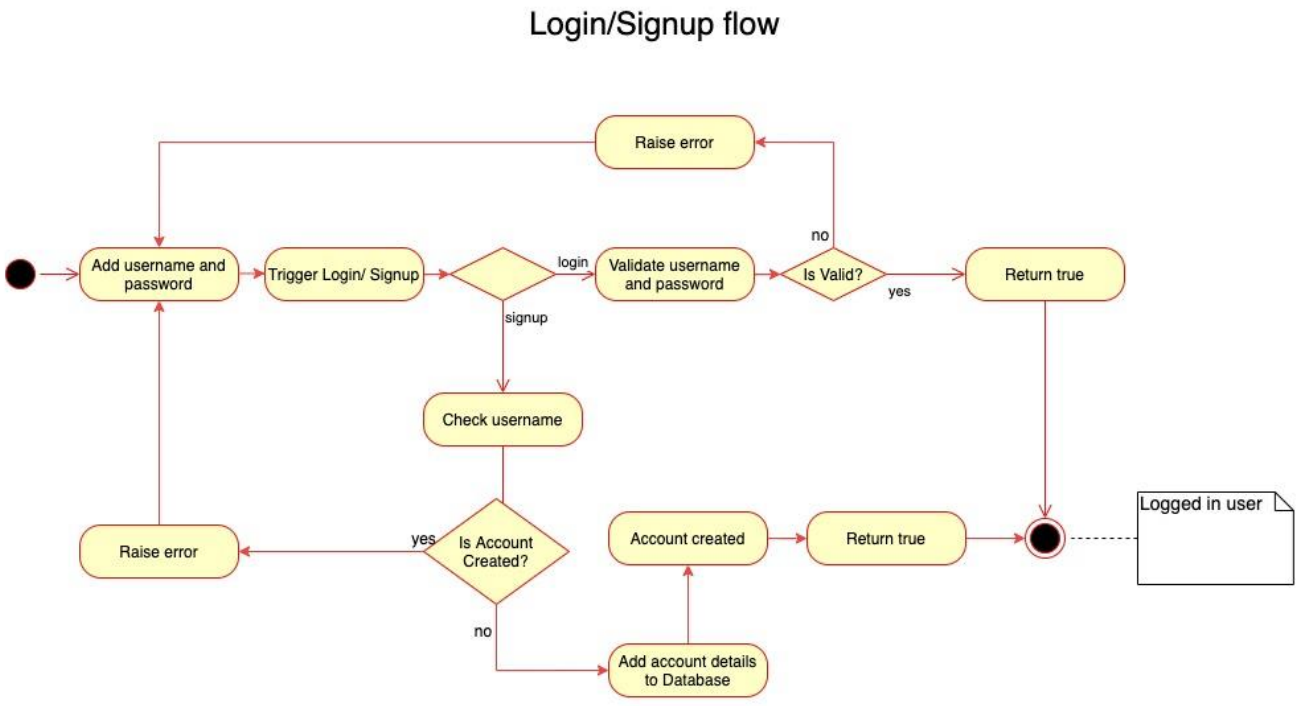

Figure 4. Activity Diagram for Login/Signup flow

- Here, (Figure 2) the activity diagram shows the scenario for login/signup of users. The user can either sign up for a new account or login to an existing one. The user also has the ability to reset the password by selecting Forgot password.
- The users can either take the role of: Publisher and consumer, once logged in.
- Next, (Figure 3) diagram shows the scenario for adding a new product. The Publisher has the ability to add new products which a consumer can rent based on availability. Publishers can add the product details in the Add new product page, once the user is verified, the details are updated on the database. Once updated the user will be notified on the same.

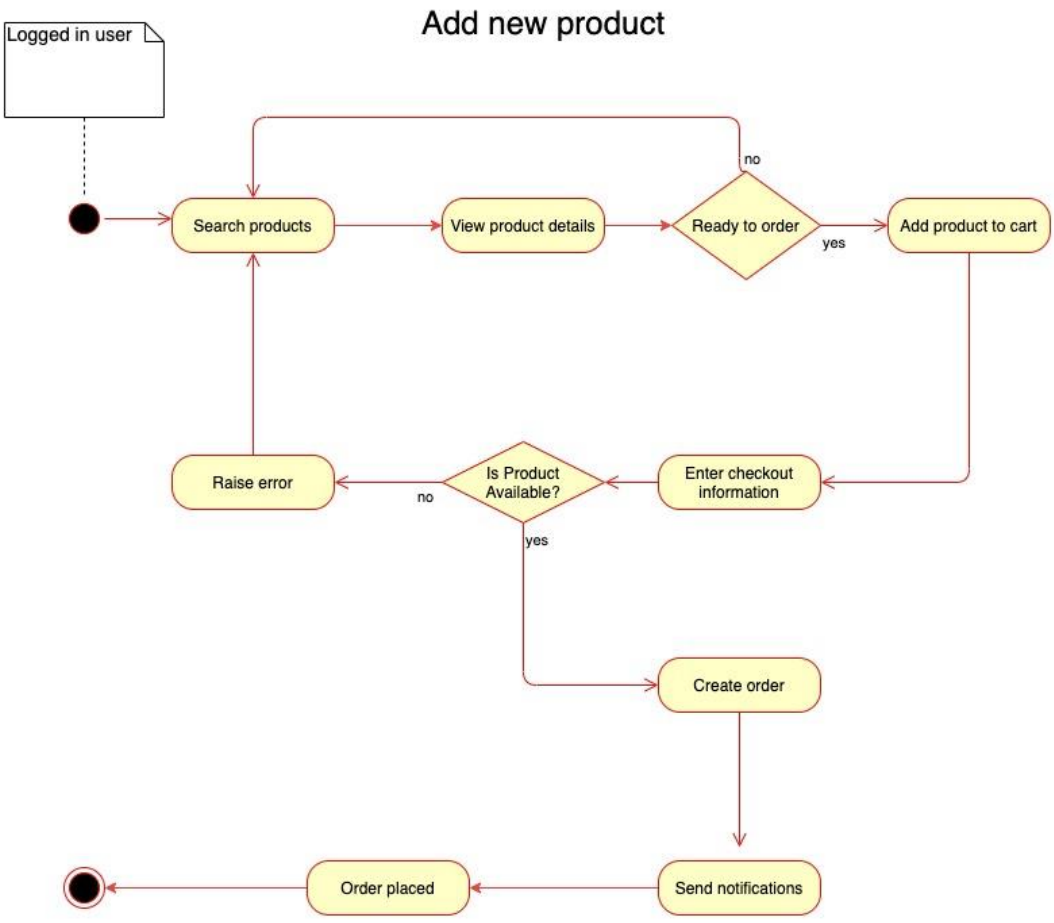

Figure 5. Activity Diagram for adding new product

Sequence Diagrams

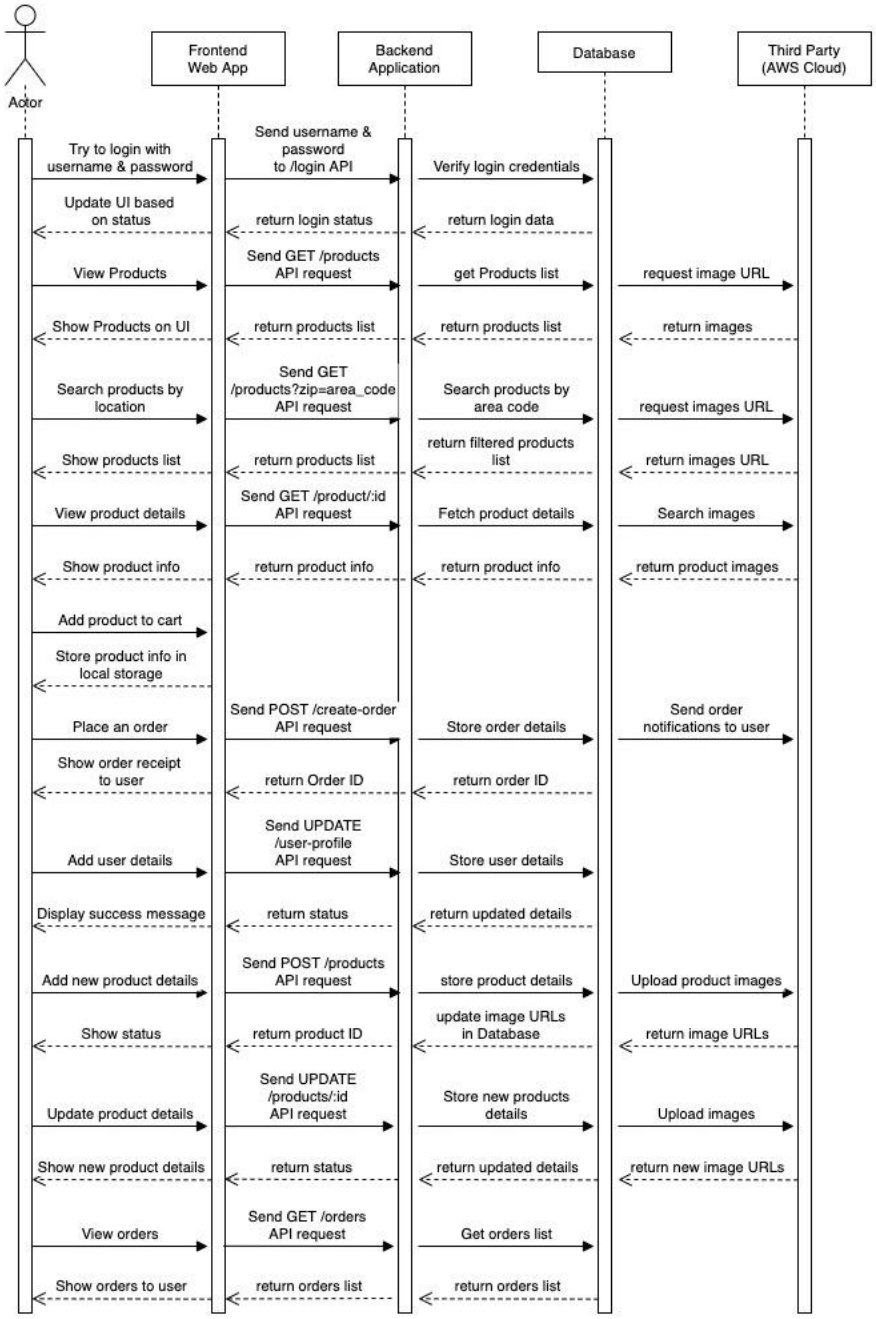

Figure 6. Sequence diagram

The sequence diagram is a representation of the message transfers between the Publisher and the Consumer across RentIt platform. It describes the actions that are performed by the Publisher and Consumer with respect to the permissions they have.
It shows how the data flow occurs across the system, starting from publisher adding new product to consumer searching for the product of interest, retrieving the details of the product, finally being able to rent the product based on its availability.

Class Diagrams

All user: Includes all methods and details related to registration of the user and the roles

- **Publisher** - Includes all functions of users who can add/update/delete products they wish to give for rent
- **Consumer** - Includes all functions of users who can search and place order for the products they wish to rent based on its availability

Products: Includes all functions related to the products like name, description, availability, owner details and zip code to help for better recommendation

Orders: Includes all details related to orders placed, like owner and renter details, the price paid and also, the information related to the date (start and end of the rental)

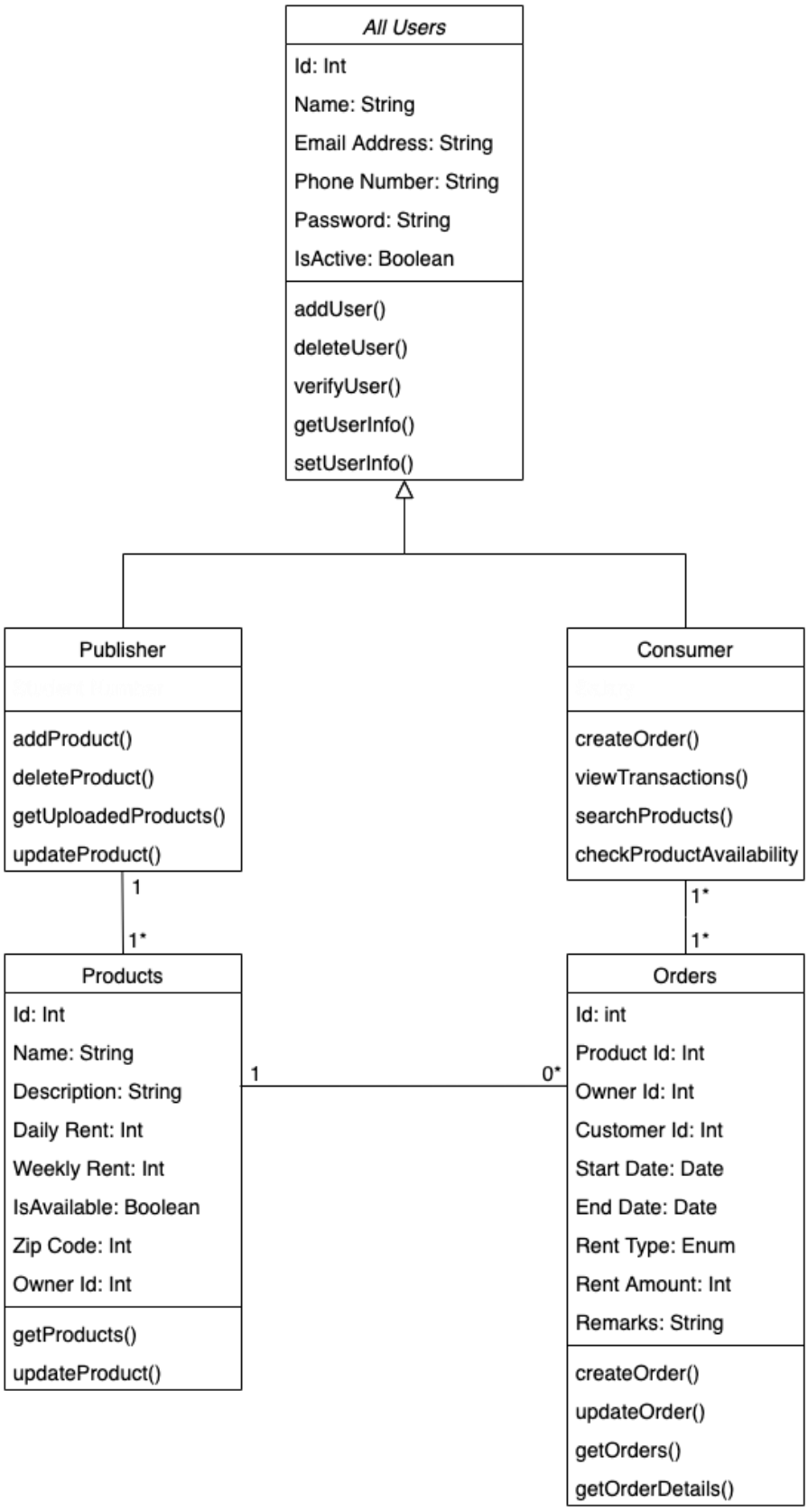

Figure 7. Class diagram

State Diagrams

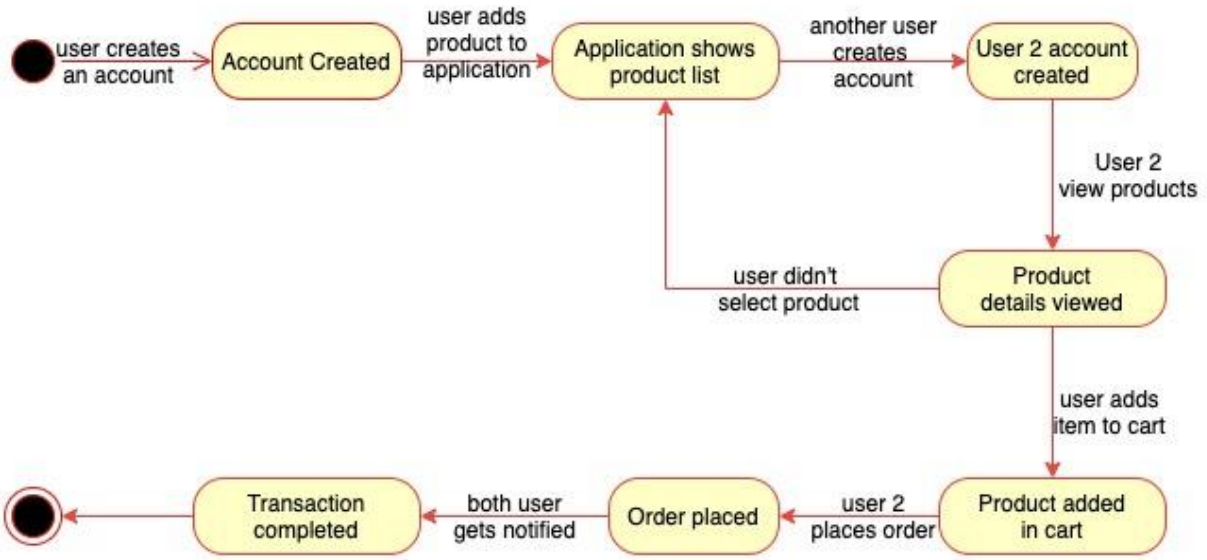

Figure 8. State Diagram

- Once account is created, the application displays the list of the items and their availability based on the information provided by the Publisher
- Products and the details of the product are displayed to the Consumer based on the location through zip code
- Display the availability of the product, verify the same before placing order
- If product is available, allow user to rent it and update the database
- Send email notification to both Publisher and Consumer of the product

6. SYSTEM ARCHITECTURE

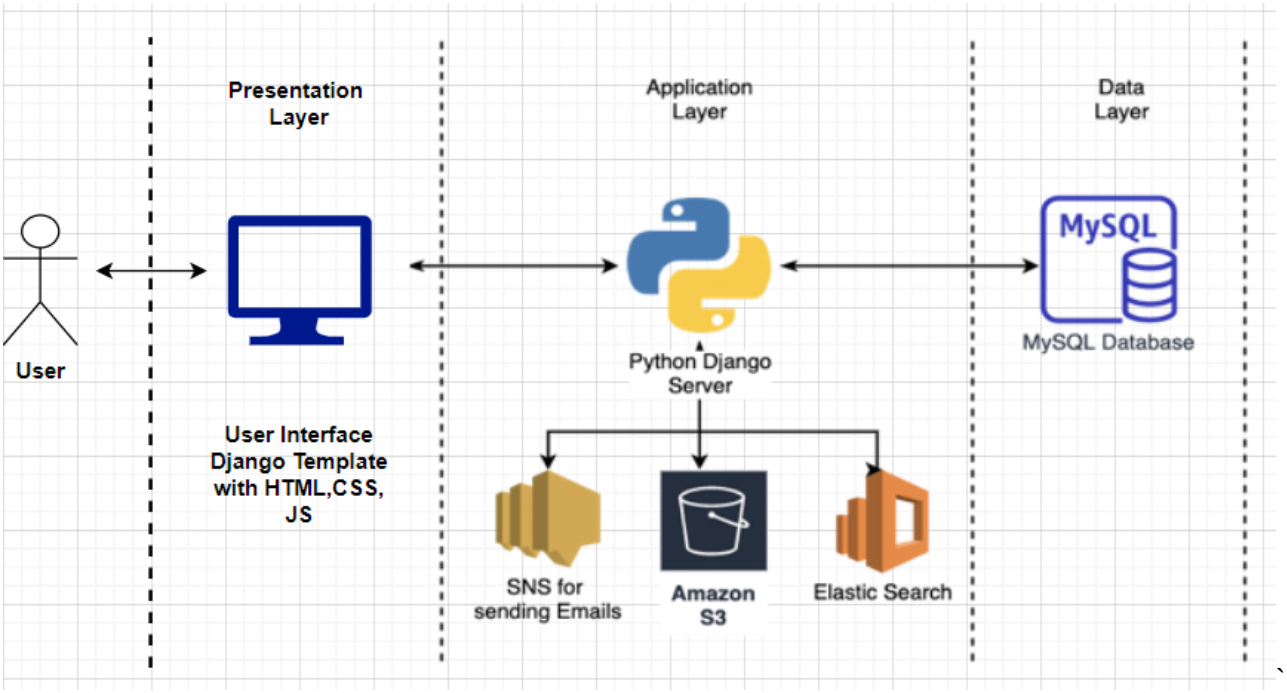

Figure 9. System Architecture

1. Backend: Django follows a Model-View-Controller (MVC) architecture, which is split up into three different parts:
    - The Model is the logical data structure behind the entire application and is represented by a database (generally relational databases such as MySQL, Postgres).
    - The View is the user interface — what you see in your browser when you visit a website. These are represented by HTML/CSS/JavaScript files.
    - The Controller is the middleman that connects the view and model together, meaning that it is the one passing data from the model to the view.

   With MVC, your application will revolve around the model—either displaying it or manipulating it. We are going to implement object-oriented programming concepts through different class based architectures.

2. Database: MySQL is used because it supports JSON data type for column values. Also, it is quite fast in elastic search implementation. It will communicate using pyscopg2 drivers to communicate with Django.
3. Frontend: We have used the server-side template binding for SEO rendering. The template allows us to write flexible, reusable webpages. The template provides us all of the HTML, CSS and JavaScript that we need to create the webpages. It will communicate using restful API's.
4. Storage: Images of the products will be stored on a third-party cloud either on AWS S3 storage.

5. Nginx : To host the website and serve static files for caching.
6. Elastic search: To improve the search time by adding indexes to the frequently searched items.

7. REFERENCES

1. https://yourstory.com/mystory/top-4-online-rental-marketplace-platform-201920-a9anc20jwv
2. https://docs.djangoproject.com/en/3.10/
3. https://developer.mozilla.org/en-US/docs/Learn/Server-side/Django/Introduction
4. https://aws.amazon.com
5. https://en.wikipedia.org/wiki/Object-oriented_programming