Software Design Document

HoneyComb
Online Resource Sharing Platform

Group - P008

Rahul Mahbubani (190001050)
Rishabh Soni (190001052)
Saket Kumar (190001054)

May 5, 2021

Table of Contents

Purpose

This design document is aimed to specify the implementation of the HoneyComb software in continuation with the Software Requirement Specification document of the same.
The software is designed to ease the process of buying, selling and renting of commodities within a college community.

Design Overview

Description of Problem

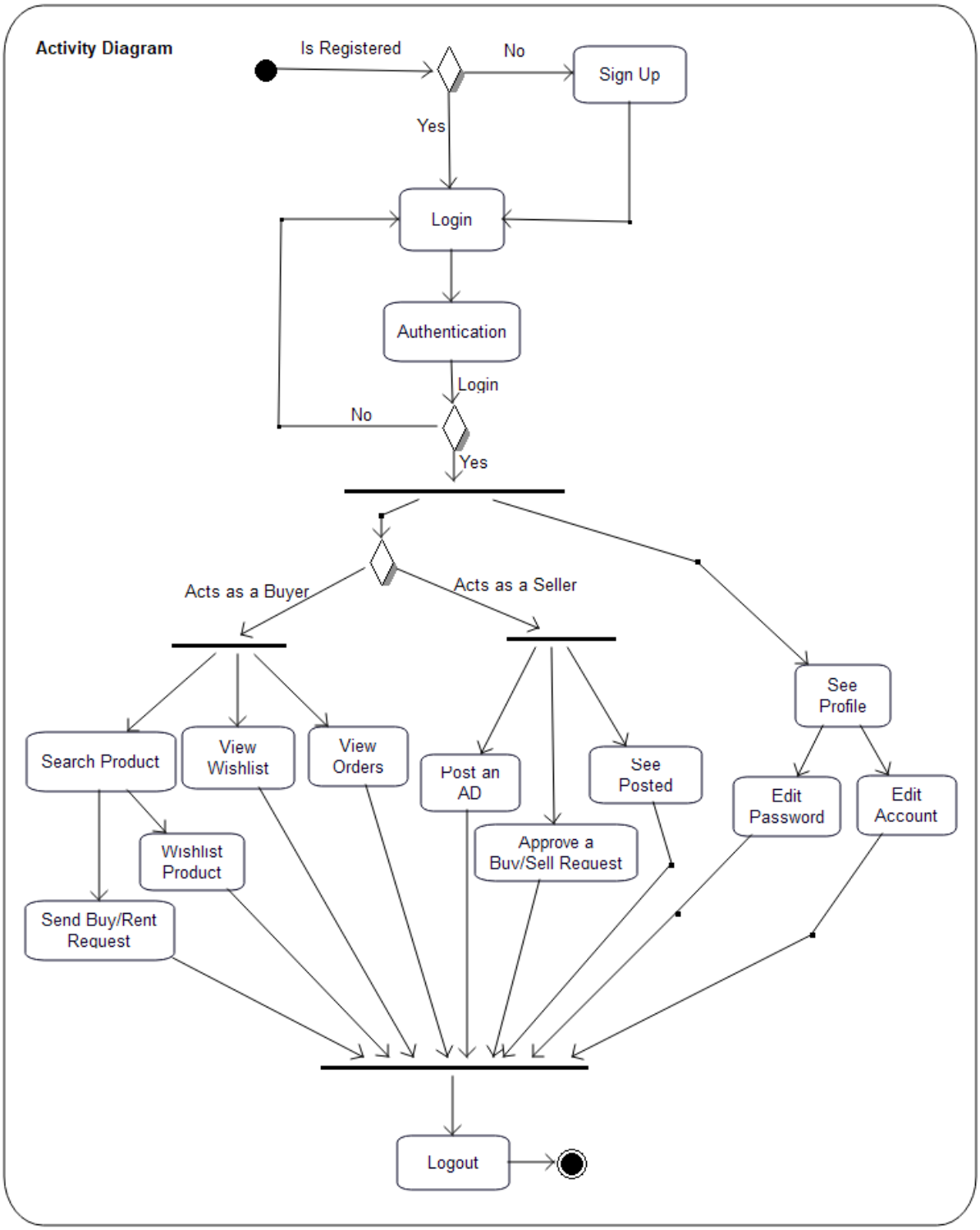
College students mostly find it difficult to search for the commonly used products like drafters, lab coats, books, subject notes etc., while others find it difficult to dump these products which are no longer in use for them.
This software aims to bridge the gap between commodity seekers and owners by providing an interface where the owners can put up the products for selling them.

Tech-Stack

Frontend - HTML, CSS, Bootstrap
Backend - Django(Python), MySQL

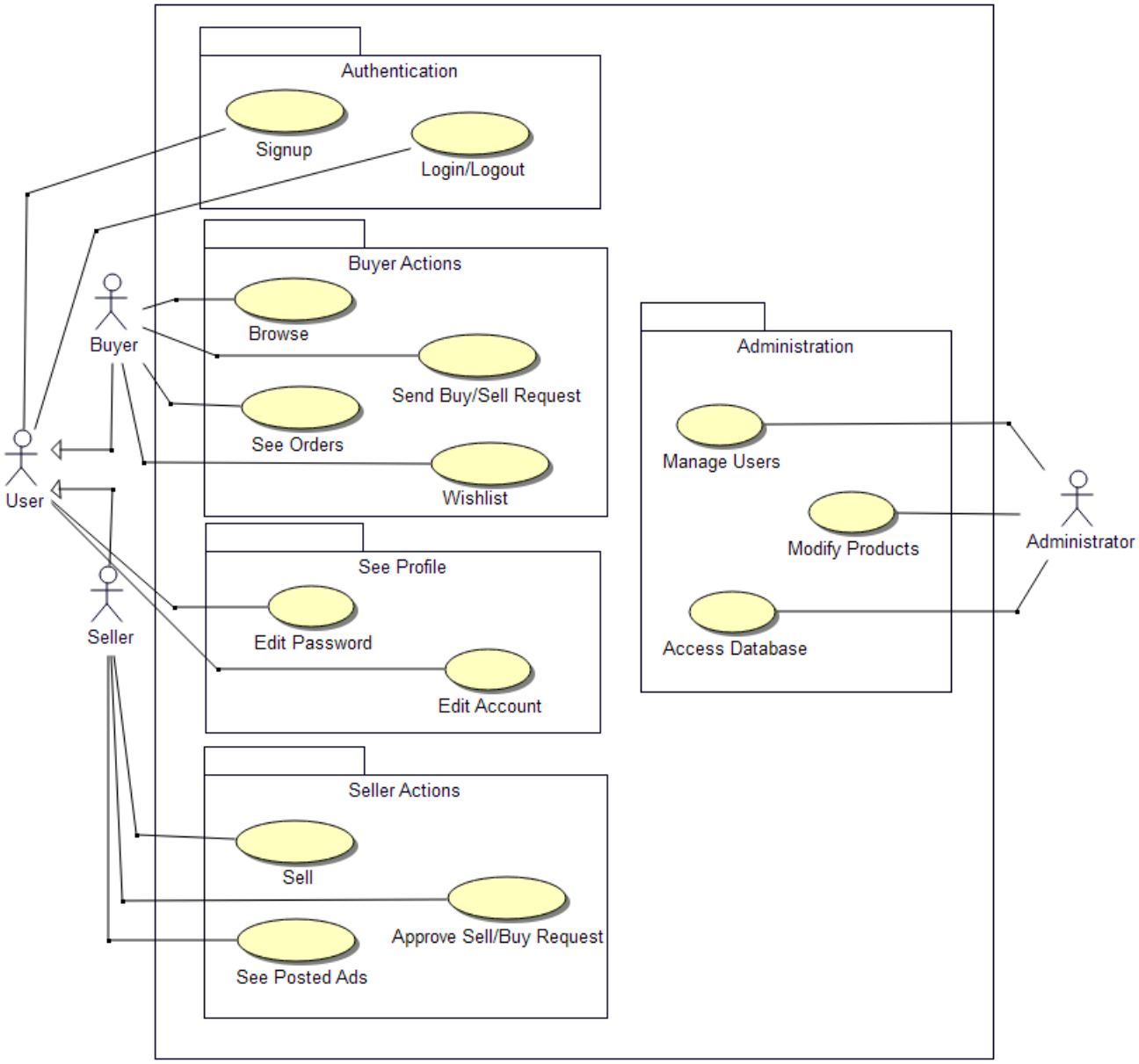
UML diagrams and descriptions

Activity Diagram



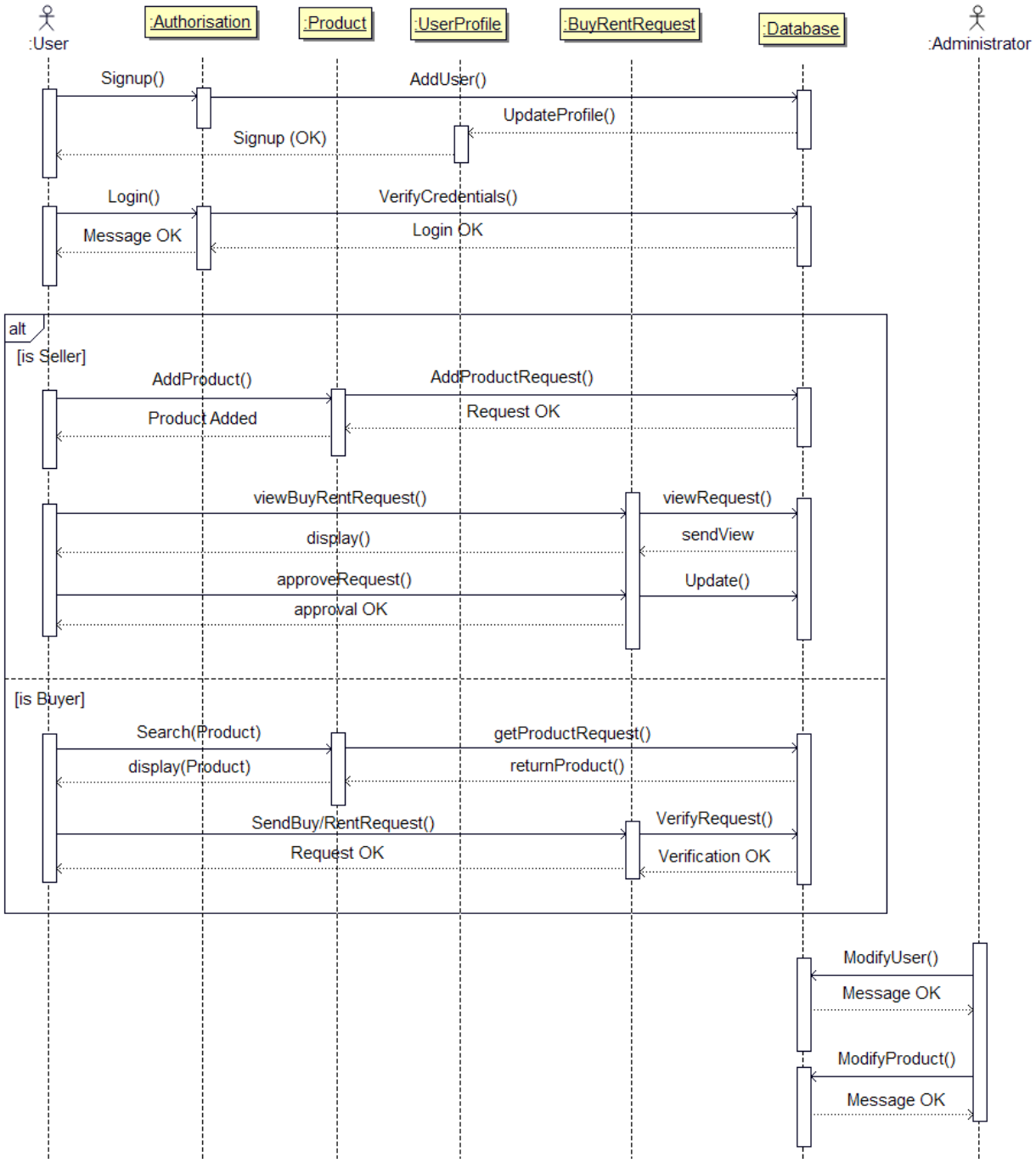
This diagram is used to model the workflow depicting conditions, constraints, sequential and concurrent activities.
We can easily understand the flow of activities on a high level. It enables us to figure out constraints and conditions that cause particular events in the web application.
We can easily briefly understand the activities in a process and the flow of control from one activity to another using this diagram.

Use Case Diagram



Buy/Sell/Rent system: Use case tabular description	
Actors	● User ● User: Buyer ● User: Seller ● Administrator
Description	A user can ● Signup into the website when he visits the website for first time ● Log into the system ● Edit his account details such as phone number and email ● Change his login password A buyer can ● Browse the website and search for the desired product using its name or category tags ● Send buy/rent request to the seller of the product ● View all the completed transactions ● Wishlist a product if he wants to buy/rent the product at some later time. He can also view all the products wishlisted by him A seller can ● Put up an AD for a product ● See all the product ADs posted by him ● Approve buy/rent requests of buyers Administrator can ● Manage and modify users. For example, he can delete a user, modify his account details. ● Modify and delete products. ● Access the complete database, for example, all the orders and transactions, all the buy/rent requests
Data	User's personal information, product summary
Stimulus	Buy/Rent request issued by a buyer
Response	Confirmation of buyer's buy/rent request by a seller
Stimulus	Request to put up ADs of products on website
Response	AD is now added to the database and shall be visible to the buyer
Comments	User must be authenticated before buying/renting/selling a product

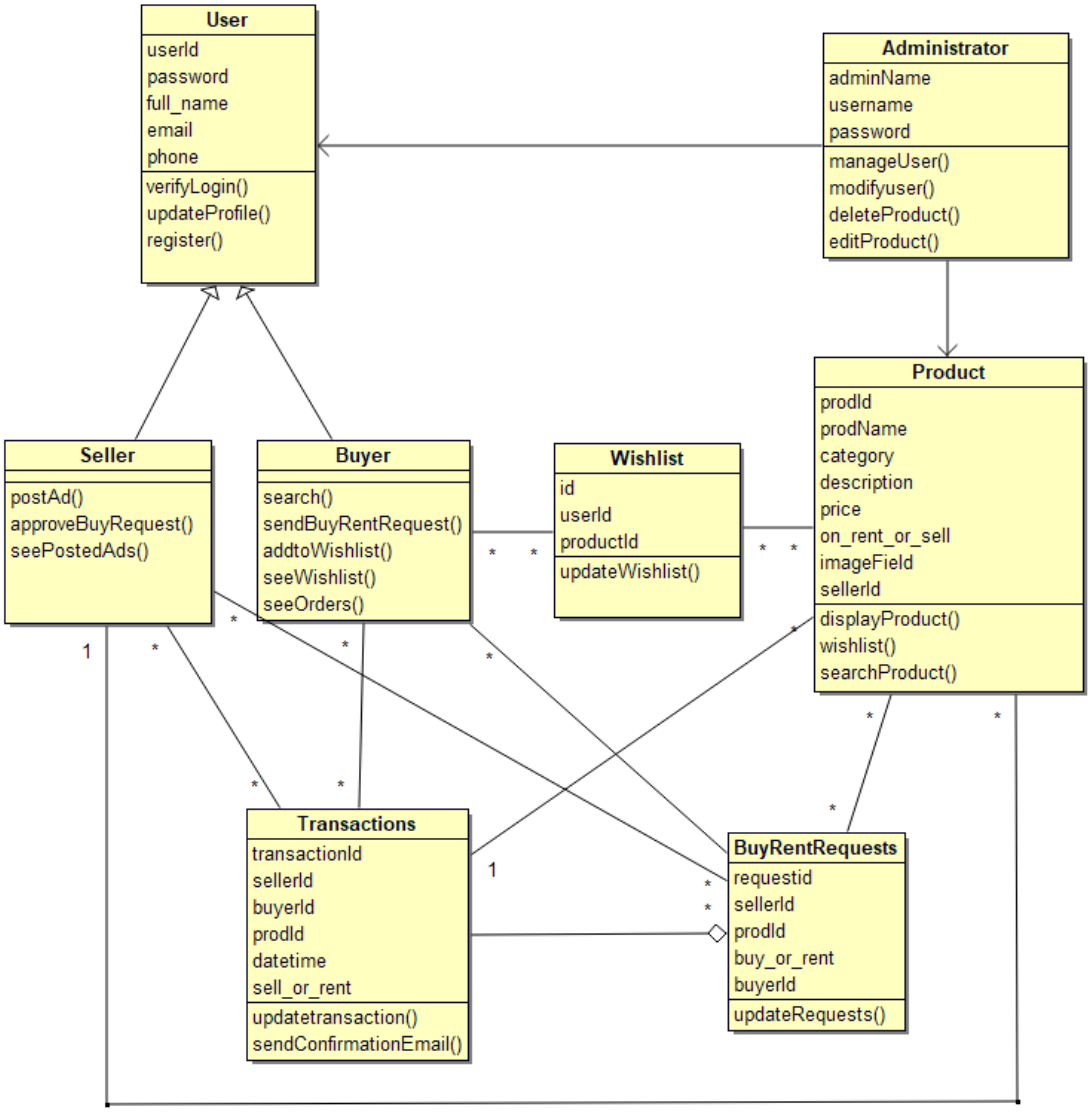
Sequence Diagram



The modeled interactions between the actors and the objects within the system are depicted in the above sequence diagram. Various use-case instances are shown in a sequential manner.

Buy/Sell/Rent system: Sequence diagram description	
Actors	• User • Administrator
Objects	• Authorisation • Product • UserProfile • BuyRentRequest • Database

Class Diagram



User: Class Description	
Attributes	
Attribute name	**Datatype**
userId	int
password	string
full_name	string
email	string
phone	int
Methods	
Method Name	**Return type**
verify_login()	void
updateProfile()	void
register()	void

Product: Class Description	
Attributes	
Attribute name	**Datatype**
prodId	int
prodName	string
category	string
description	string
price	int
on_rent_or_sell	string
imageField	string
sellerId	int
Methods	

Method Name	**Return type**
displayProduct()	void
wishlist()	void
searchProduct()	list of products

Administrator: Class Description	
Attributes	
Attribute name	**Datatype**
adminName	string
username	string
password	string
Methods	
Method Name	**Return type**
manageUser()	void
modifyUser()	void
deleteProduct()	void
editProduct()	void

Seller: Class Description	
Methods	
Method Name	**Return type**
postAd()	void
approveBuyRequest()	void
seeBuyRequests()	List of users

Buyer: Class Description	
Methods	
Method Name	**Return type**
search()	List of products
sendBuyRentRequest()	void
seeWishlist()	List of products
addToWishlist()	void
seeOrder()	List of products

Wishlist: Class Description	
Attributes	
Attribute name	**Datatype**
Id	int
userId	int
productId	int
Methods	
Method Name	**Return type**
updateWishlist()	void

Transactions: Class Description	
Attributes	
Attribute name	**Datatype**
transactionId	int
sellerId	int
buyerId	int

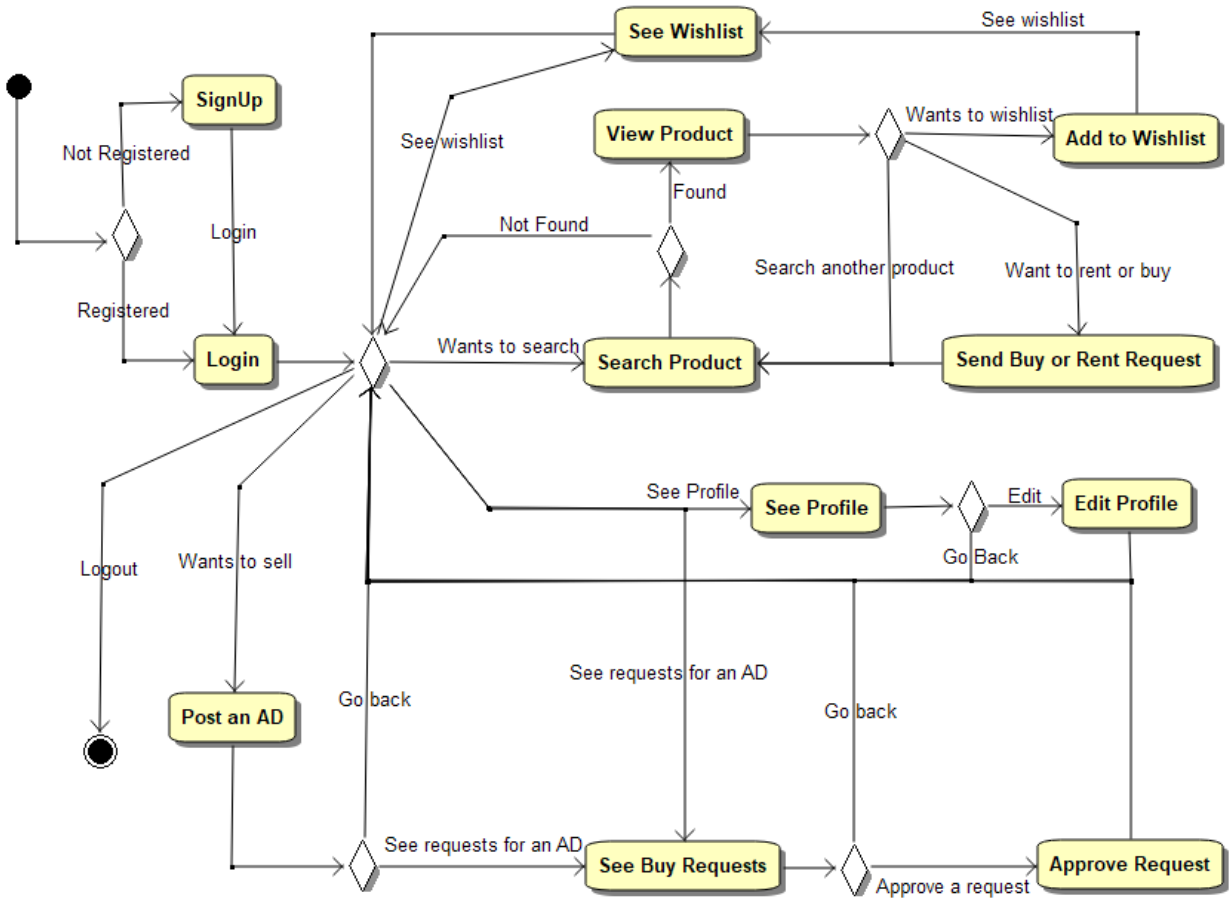
productId	int
datetime	datetime
sell_or_rent	string
Methods	
Method Name	**Return type**
updatetransaction()	void
sendConfirmationEmail()	void

BuyRentRequests: Class Description	
Attributes	
Attribute name	**Datatype**
requestId	int
sellerId	int
prodId	int
buy_or_rent	string
buyerId	int
Methods	
Method Name	**Return type**
updateRequests()	void

Note:

- The Buyer and Seller class are inherited from the User class.
- Transactions class is aggregated from the BuyRentRequests class.
- Administrator has a directed association with User and Product class.

State Diagram



Buy/Sell/Rent system: State and Stimuli descriptions	
Signup	Users need to sign up the first time they visit the website. After this step they need to login. If a user is already registered, then he can skip this signup step.
Login	Users can log into the website using the credentials filled in the signup form. After a successful login, users land up on the home page which bifurcates into several states depending on the choice made by the user.
See Profile	Users can view their profile which displays their details such as name, username, address, email etc. From here users can either go back to home page, or can choose to modify their details.
Edit Profile	Here, users can update any credential like address or email. After updating, users are redirected to home state, where they can make their next decision.

Search Product	Users who want to buy/rent any commodity can opt to search for the desired product. They shall be directed to view product if that product is found in the database, else they are redirected to home state.
View Product	Users can view additional details of a product such as its seller, description, price etc. From here users can opt to wishlist the product or send a buy/rent request to the seller
Add to Wishlist	Users can wishlist a product if they wish to buy/rent the product later. From here, they are directed to their wishlist (see wishlist)
See Wishlist	Users can see all the products wishlisted by them. After this they are directed to home state.
Send Buy or Rent Request	Users can send buy/rent request to the seller of the product, which the seller shall be able to view in his See Buy Requests state. From here, they will be redirected to search product state.
Post an AD	Users can wish to proceed to this state from home state if they wish to post an AD for a product. From here, they can opt to see all the buy/rent requests or go back to home state.
See Buy Requests	Here, users shall be able to see all the buy/rent requests for any product posted by them as an AD. Now, users can opt to approve the request or go back to home state.
Approve Request	Sellers can approve the buy/rent requests of the potential buyers after which they are directed to home state again.
Logout	Once the users have completed the desired operations on the platform they can logout from the system

Additional Comments

Proper reformatting of code is required since this project is aimed to be an open source project.