Software Requirements Specification

for

eCommerce System

Version 1.0

Prepared by Nanthakarn Limkool

Kasetsart University

9 May 14, 2021

Table of Contents

1. Introduction

1.1 Purpose

Defining and describing the functions and specifications of the E-Commerce System (ECS) is the primary goal of this Software Requirements Specification (SRS). This Software Requirements Specification illustrates, in clear terms, the system's primary uses and required functionality as specified by our customer.

1.2 Document Conventions

This document uses the following conventions.

ECS	E-Commerce System - this project
Visitor	The people who get on the website without register or login
User	The people who registered and already login to the website
Admin	A role of user [user role: Admin, Customer(default)]
Client	The people who intend to use this website [Visitor & User]
CRUD	Create, Read, Update, Delete
Heroku	A platform as a service (PaaS) that use to build, run, and operate applications entirely in the cloud for this ECS project
RoR	Ruby on Rails

1.3 Intended Audience and Reading Suggestions

This ECS is a prototype system. This project is useful for software engineering students as well as for beginners in web development.

1.4 Product Scope

The software system being produced is called E-Commerce System or ECS. It is being produced for a customer interested in selling their own product via the Internet. This system is largely cross-platform and is available to anyone. The system will be run on a Heroku web server with each user having a remote user interface through a web browser to interact with it.

The E-Commerce System will allow any visitor to create an account to become a user. The user, through the process of account creation, will have the option to become a member of the site. The system will allow users to view, create, update, delete (CRUD) their products and buy other products. The ECS also allows an admin to manage the inventory with full create, retrieve, update and delete (CRUD) functionality with regards to users, products, orders, comments, etc. in the system.

1.5 References

1. Mr. Borzoo Bonakdarpour, Elicitation Meeting, September 25th 2007.
2. IEEE-SA Standards Board, “IEEE Recommended Practice for Software Requirements Specifications”, Software Engineering Standards Committee of the IEEE Computer Society, June 25th, 1998.
3. Prof. Betty H.C. Cheng, “Intro to Specifications”, CSE 435, East Lansing, MI, September 2007.

2. Overall Description

2.1 Product Perspective

ECS is an online shopping website that supports a number of functions for both the consumer and store management.

The website must be available to anyone and as such must work correctly in any browser. As stated by the customer, there are no hardware or software requirements beyond these including, but not limited to, memory or specific software packages that need to be utilized nor software packages that need not be utilized.

2.2 Product Functions

BECS will provide a number of functions; each is listed below.

- Allow visitor to view all published products
- Allow visitor to sign up as a user
    - Information need for registration are username, email, and password
- After signup, allow users to create their products
    - Information need for creating a product are title, description, number of stock, and price
- Allow users to view all items [status: draft, published, archived], edit, and delete them
- Allow users to view all categories, add, edit, and delete them
- Allow users to comments/reviews product and view them
- Allow users to view all orders, add, edit them
    - Order available on only published and not out of stock products

2.3 User Classes and Characteristics

The typical ECS user is simply anyone that has access to the Internet and a web browser. It is assumed that the user is familiar enough with a computer to operate the browser, keyboard, and mouse and is capable of browsing.

2.4 Operating Environment

The operating environment for the ECS is as follow:

- Heroku Web Service
- Operating System: Windows, macOS, Android, etc.
- Database: Postgresql
- Platform:
    - Framework: Ruby on Rails
    - Storage service: Amazon Aurora / Amazon RDS
    - UX/UI design: Adobe XD
    - UX/UI: HTML, CSS, slim, JavaScript

2.5 Design and Implementation Constraints

As stated by the user, security is a concern for this system. The database stores password and password recovery features after numerous invalid login attempts. A strong password is a password that meets a number of conditions that are set in place so that the user's passwords cannot be easily guessed by an attacker. Generally, these rules include ensuring that the password contains a sufficient number of characters and contains not only lowercase letters but also capitals, numbers, and in some cases, symbols.

2.6 User Documentation

- Use Cases: https://docs.google.com/document/d/1s6eQ5tuxSrDUpTsanqusEJuv6K8a3gkn10iSfn_6EaY/edit?usp=sharing

2.7 Assumptions and Dependencies

Client:
We have assumed that clients' computers are in proper working condition and that the user is capable of operating these system's basic functions including. And any browser could navigate to the address of this ECS website.

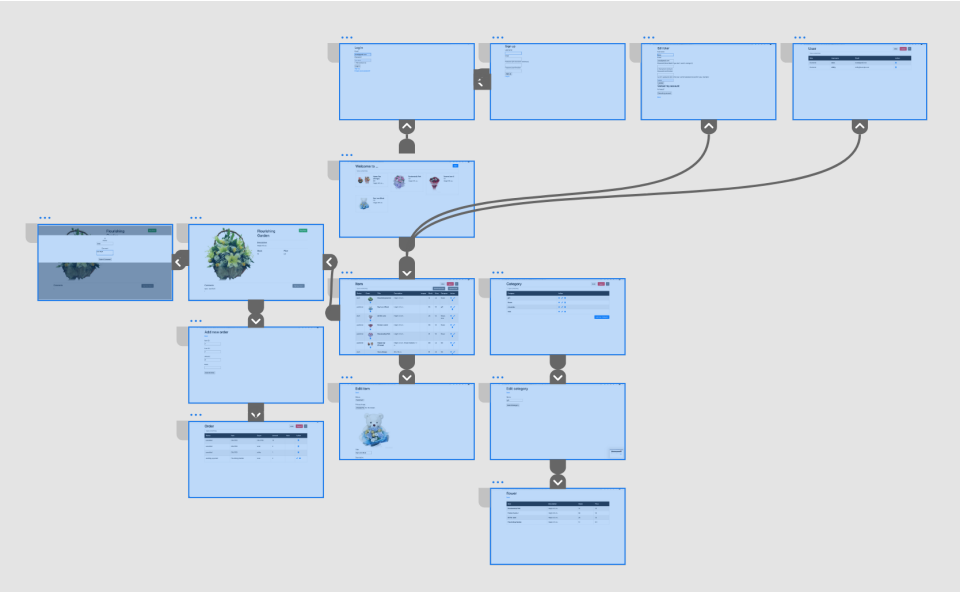
Provider:
We have assumed that the ECS will be running on a properly working web server and database system with an Internet connection that allows this system to perform all communications with clients.

Assumptions:

- Users should be able to order more than a single product in a transaction.
- Only the current admin is able to assign the admin role to other users.
- The admin cannot have the customer role at the same time.

3. External Interface Requirements

3.1 User Interfaces

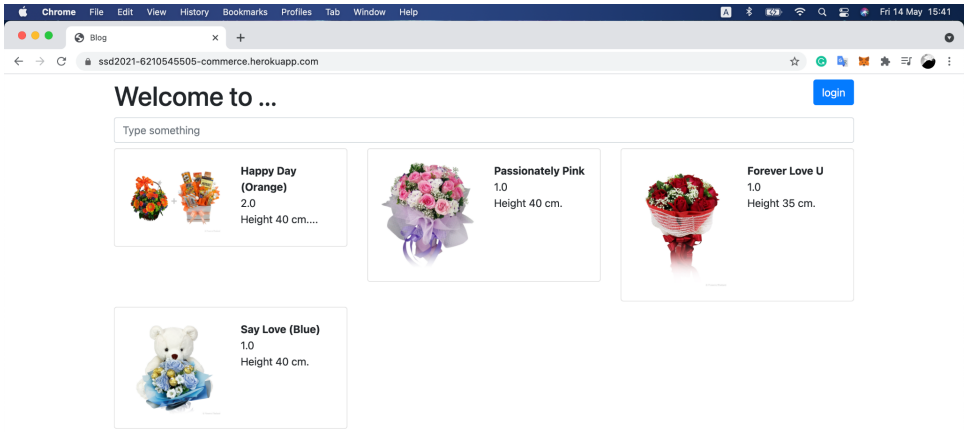


[Implementation using Adobe XD, Heroku (Deployed)]

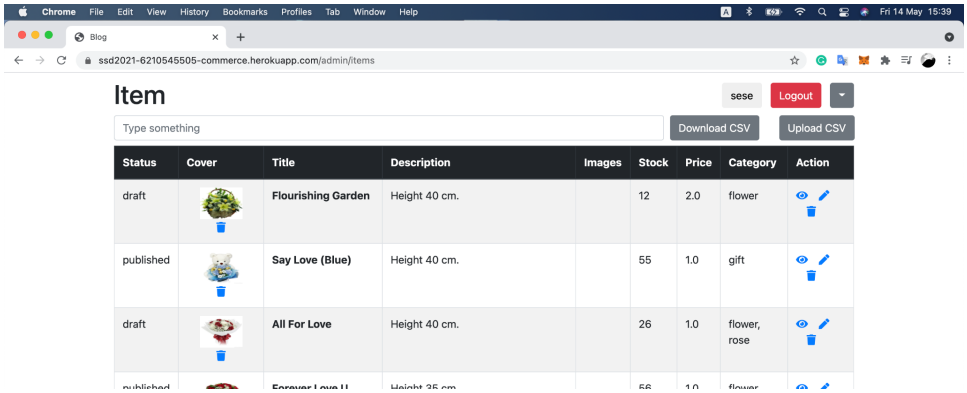
The user interface will consist of four main types of screens: index, new, edit, and show. All four main types will be for each model used in the ECS website.

The index page is the page that contains the list of objects in its model. The pages will follow:

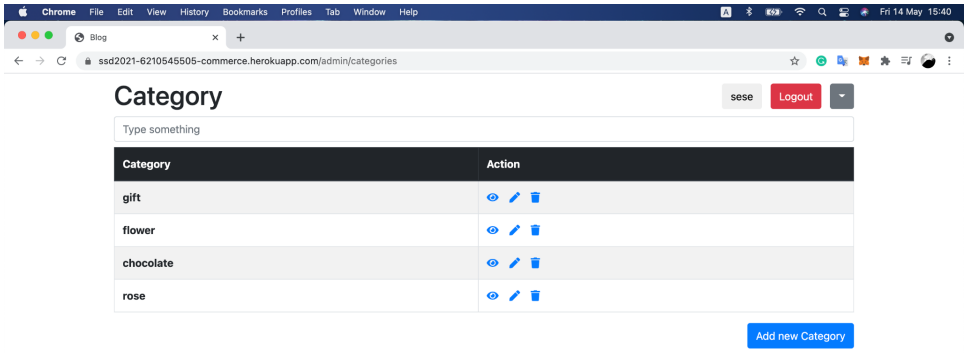
- The Welcome page
  This page will contain all published products, where visitors can click on the product card to redirect to the product show page.



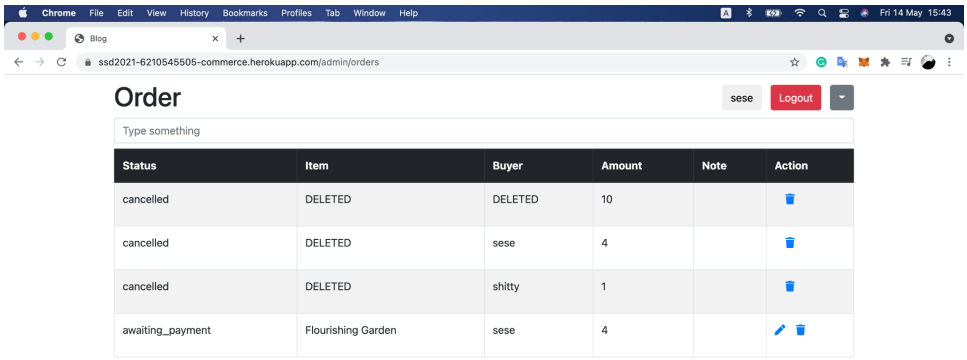
- The Item (Product) index page [Only registered users can get on this page.]
  This page will contain products from all status and action buttons for each of them. The buttons redirect to the item show page, the Item edit page, and the item delete function. Also, this page contains add product sections which are the ‘Add New Item’ on the bottom of the table that redirects to Item (Product) new page and the ‘Upload CSV’ button that receives items information as a CSV file.



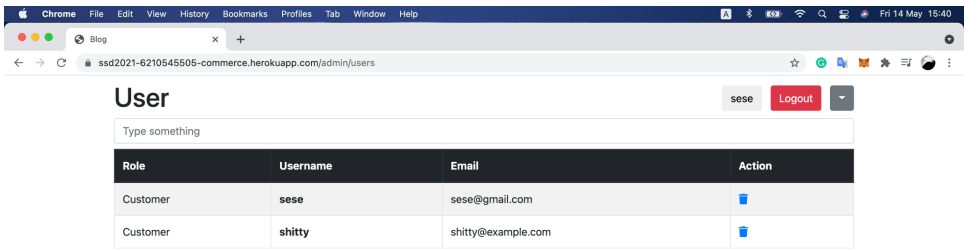
- The Category index page [Only registered users can get on this page.]
  This page contains most the same as the Item index page before.



- The Order index page [Only registered users can get on this page.]
  This page also contains almost the same as the Item index page before, but this page won't have 'show' button since all the detail of each order is shown on the table.

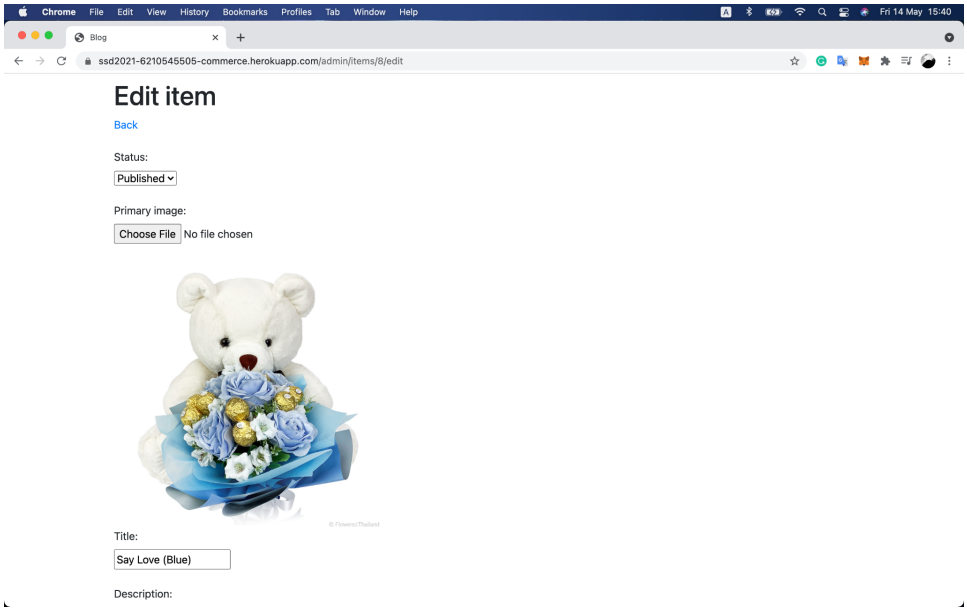


- The User index page [Only registered users can get on this page.]
  This page will contain the role, username, email, and delete function from all registered users.

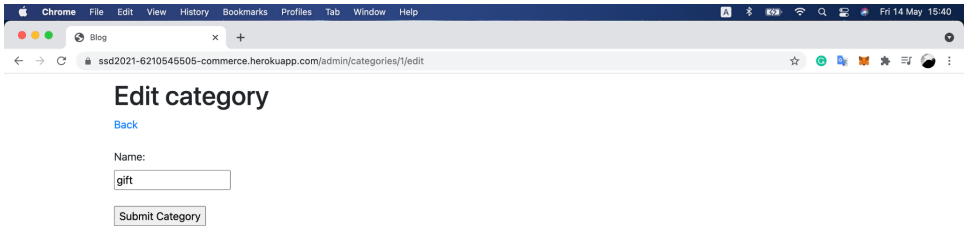


The new and edit page basically uses the same form. The pages will follow:

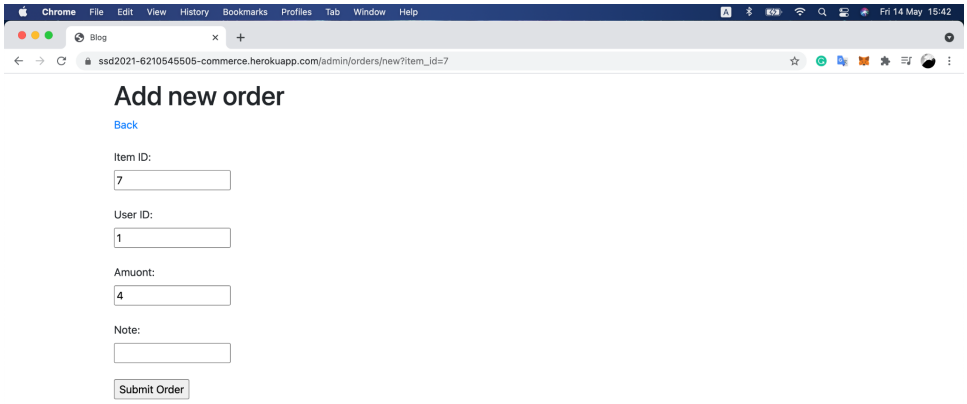
- The Item (Product) new/edit page
  This page contains a form of Item/Product. The form collects data follow:
  - Status
  - Primary image
  - Title
  - Description
  - Supporting images
  - Stock
  - Price
  - Tags
  - Category



- The Category new/edit page
  This page contains a form of Category. The form collects data follow:
  - Category Name

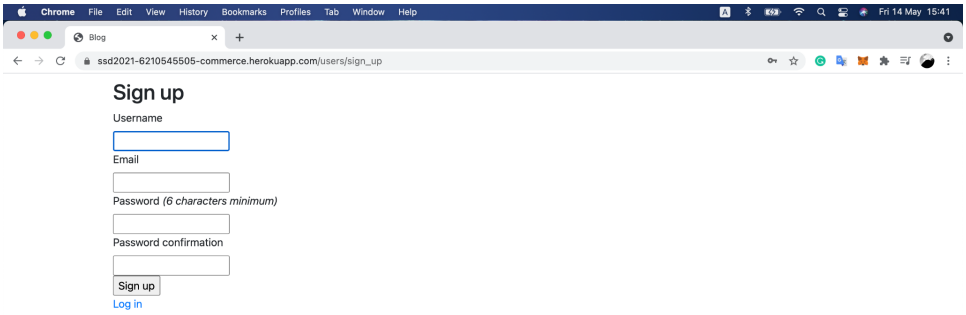


- The Order new/edit page
  This page contains a form of Order. The form collects data follow:
  - Item ID
  - User ID
  - Amount
  - Note

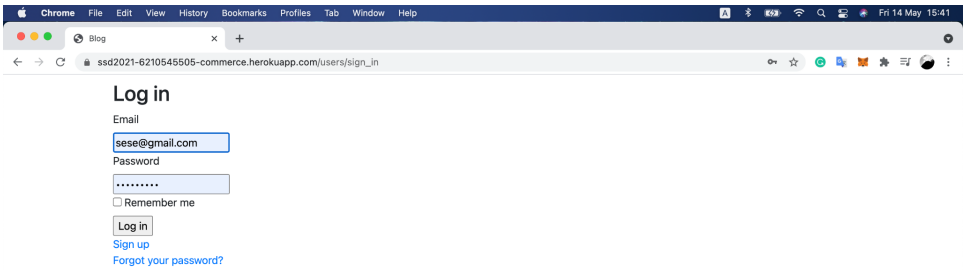


There are 2 special models that have their own UI. There are

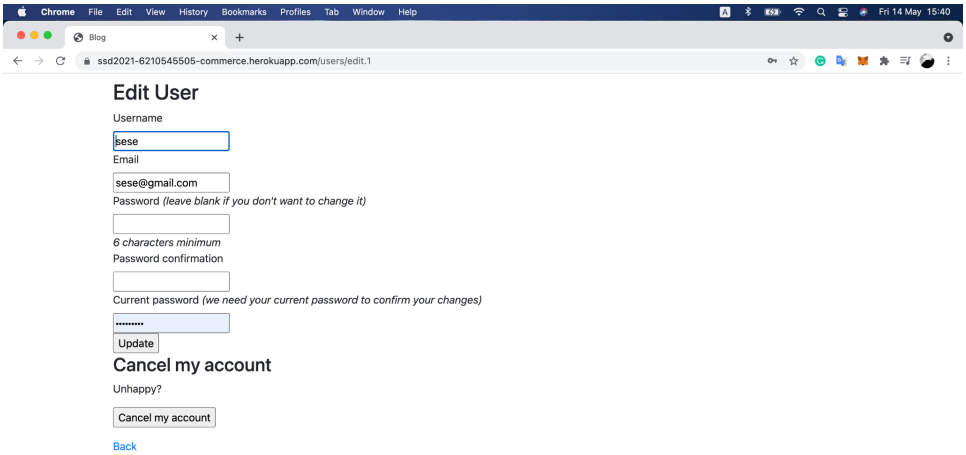
- The User model
  - Sign up



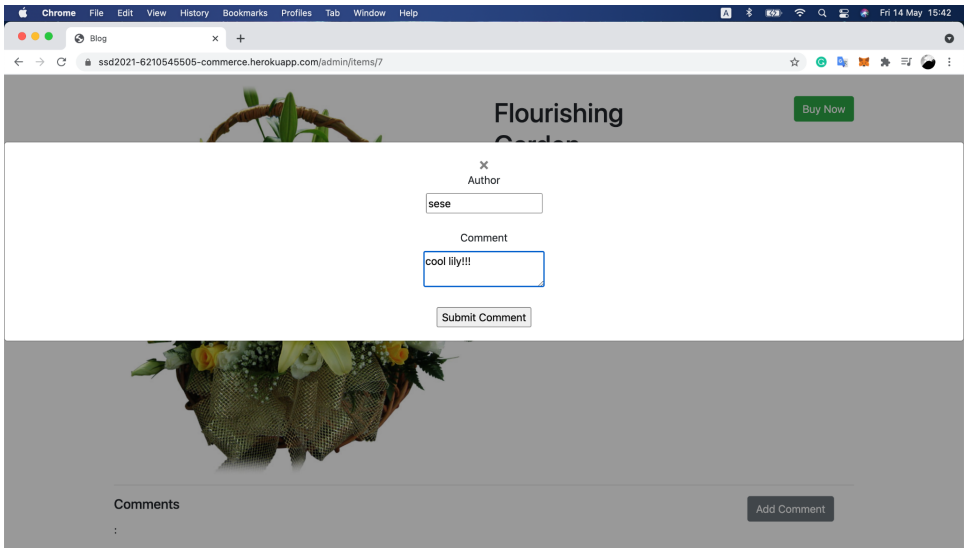
  - Sign in



- Edit Profile

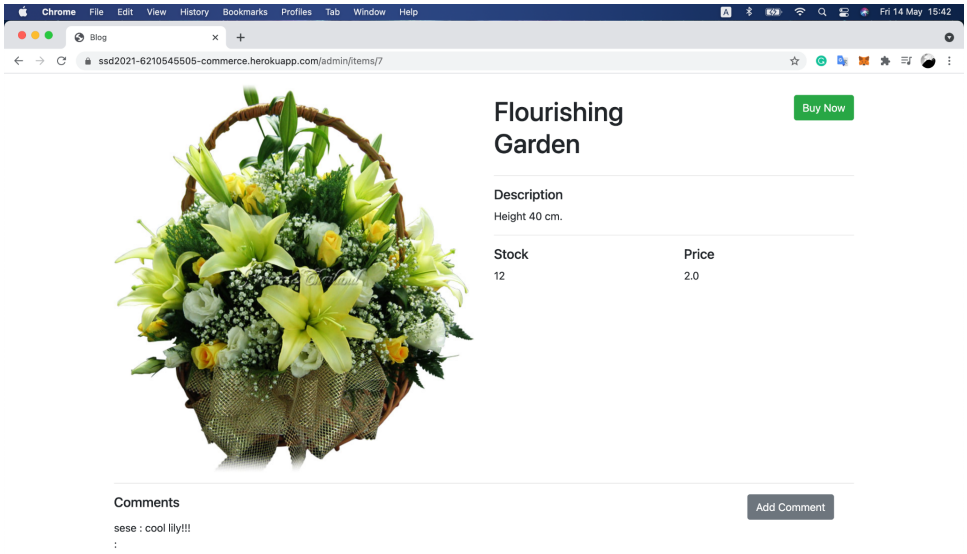


- The Comment model
  - New comment section will contain in the Item/Product show page as a popup
  - View/Show will also contain in the Item/Product show page

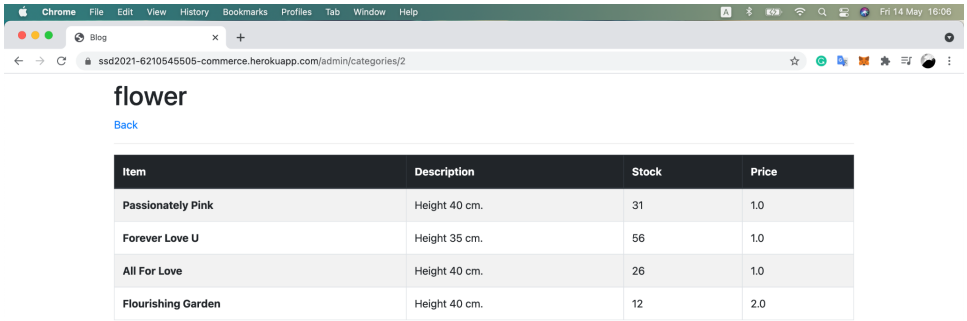


The show page is used to show objects in detail. The pages will follow:

- The Item (Product) show page
  This page also has the ‘Buy Now’ button that will redirect the user to the new order page.



- The Category show page
  This page will show the list of the products that contained in that category.



3.2 Hardware Interfaces

The hardware requirement at the user end is really simple and the website can also run on the hardware that can run a basic simple browser, although the hardware should be good enough during peak times for the web servers

3.3 Software Interfaces

The software used the following.

Software used	Description
Operating system	For the ECS, we try to support users from every OS. [Windows, macOS, Android, etc.]
Database	We use the Postgresql database plugin from Heroku web service to store data for this website and use AWS to store the media.
Framework	We implemented Ruby on Rails (RoR) for this project, which is an open-source framework written with the Ruby programming language

3.4 Communications Interfaces

The client must connect to the Internet to access the Website. The side shall use the HTTP protocol for communication over the internet and for intranet communication.

4. System Features

- Only Admin and account owner are able to delete the user account.
- Order can be created only when the product is published, its stock is more than or equal to the order amount and the order amount is positive.
- Users are only able to create product comments/reviews after sign in.
- Users are able to browse every model on its index page.
- The order status is updated automatically.
- When the parent model attributes update, its child model also updates automatically.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The ECS shall be based on the web and has to be run from a web server. The product shall take initial load time depending on internet connection strength which also depends on the media loaded. The performance also depends upon client hardware.

5.2 Safety Requirements

The database may get crushed at any certain time due to virus or operating system failure. Therefore it is required to take the database backup so that the database is not lost. Proper UPS/ Inverter facility should be there in case of power supply failure.

5.3 Security Requirements

- The system will use a secured database.
- Visitors can just view item/product information but they cannot modify anything before they register and log in.
- Users will have different roles and can only do what their role is capable.

5.4 Software Quality Attributes

There may be multiple admin's creating the system, all of them will have the right to create new functions or changes to the system. But the member (normal registered user) cannot do any changes. The ECS should be open source. The quality of the database is maintained in such a way so that it can be very user-friendly to all the users of the database.

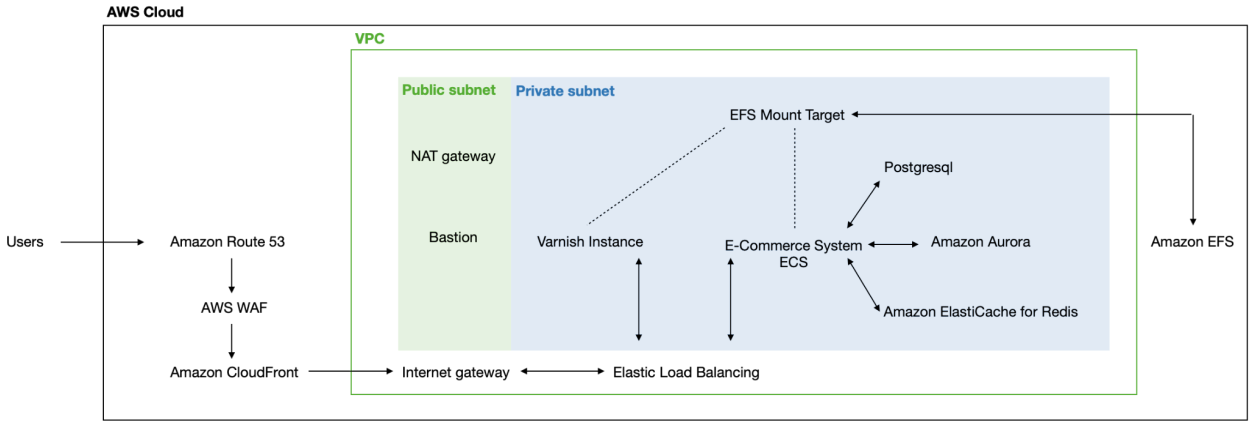
5.5 Business Rules

A business rule is anything that captures and implements business policies and practices. A rule can enforce business policy, make a decision, or infer new data from existing data. This includes the rules and regulations that the system users should abide by. This includes the cost of the system and the discount offers provided. The users should avoid illegal rules and protocols. Neither admin nor members should cross the rules and regulations.

6. Other Requirements

ECS shall handle expected and unexpected errors in ways that prevent loss in information and a long downtime period.

Appendix A: Infrastructure Design



The Content of Design Idea

- Maintain Security & Reliable
  - Amazon Route 53 - checks DNS configuration
  - AWS WAF - web firewall
  - Bastion - control network access
  - NAT Gateway - enable instances in a private subnet

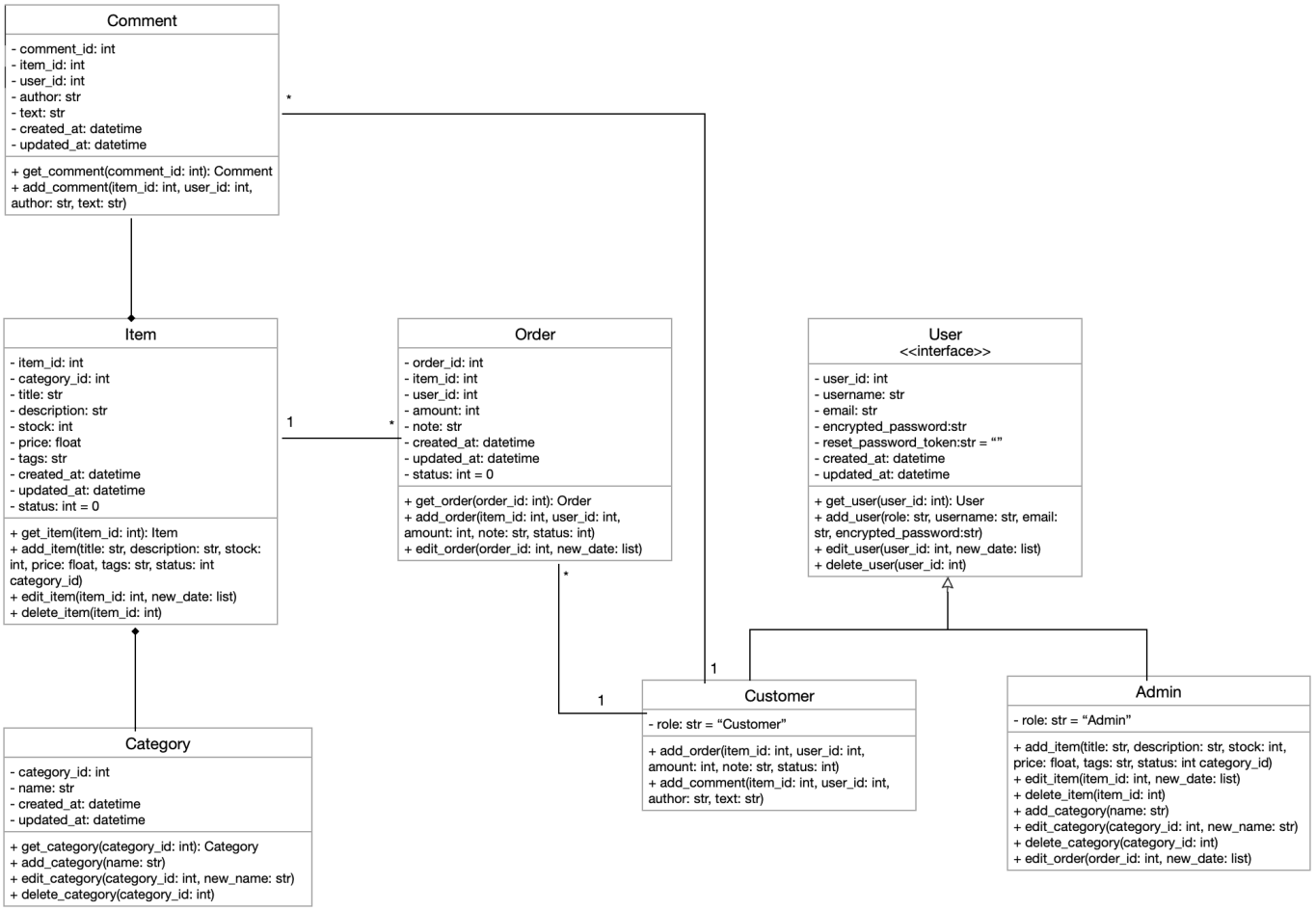
- Maintain Performance
  - CloudFront - speeds up the distribution of static and dynamic web content
  - Varnish Cache - accelerator caching HTTP reverse proxy
  - Amazon ElastiCache for Redis - provides cache for database

- Maintain Scalability
  - Elastic Load Balancing - distributes traffic across
  - Amazon Aurora or Amazon RDS - simplify database administration

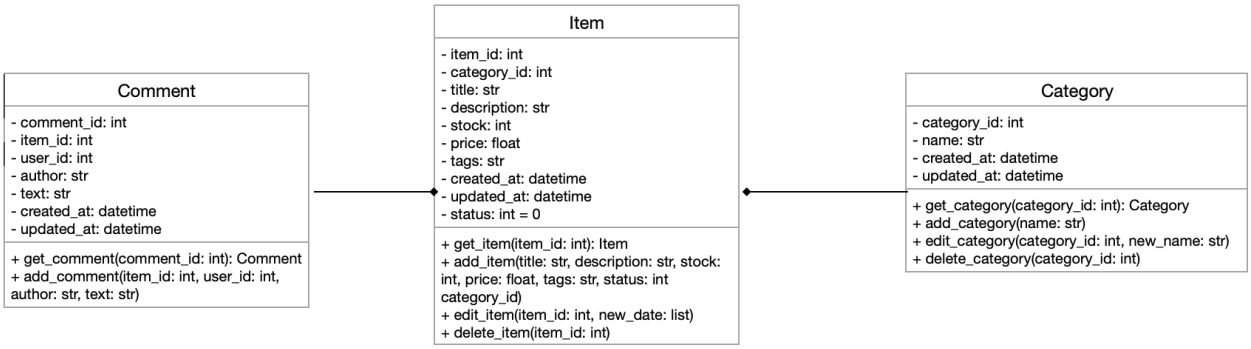
- Recommended by Amazon
  - Amazon EFS - Analytics & machine learning

Appendix B: UML Diagrams

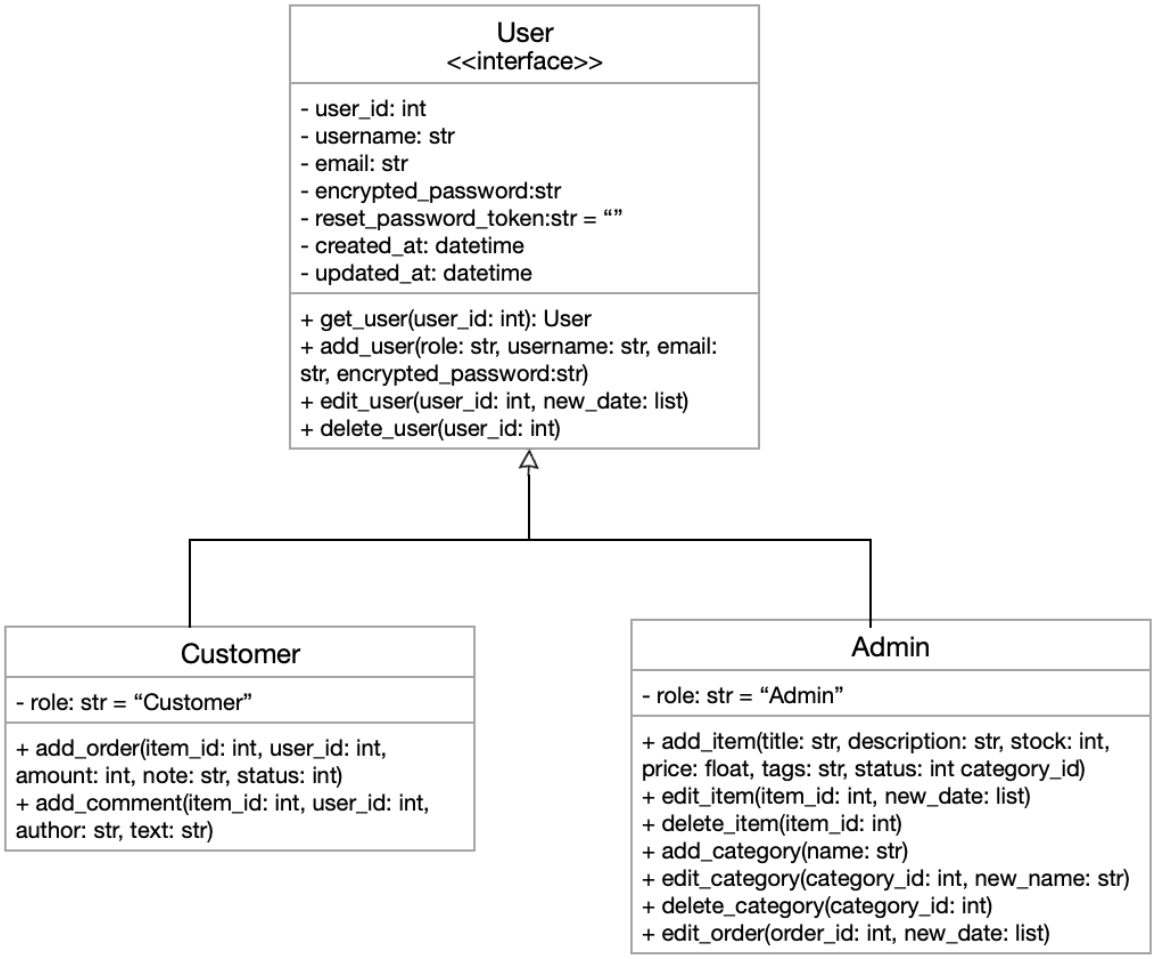
Class Diagram



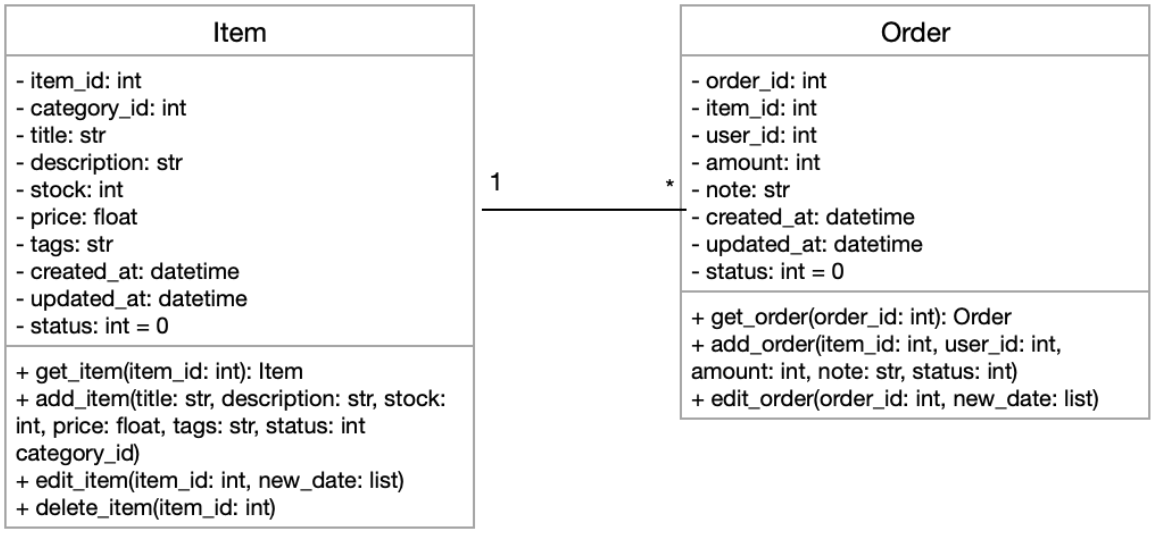
- **Composition**



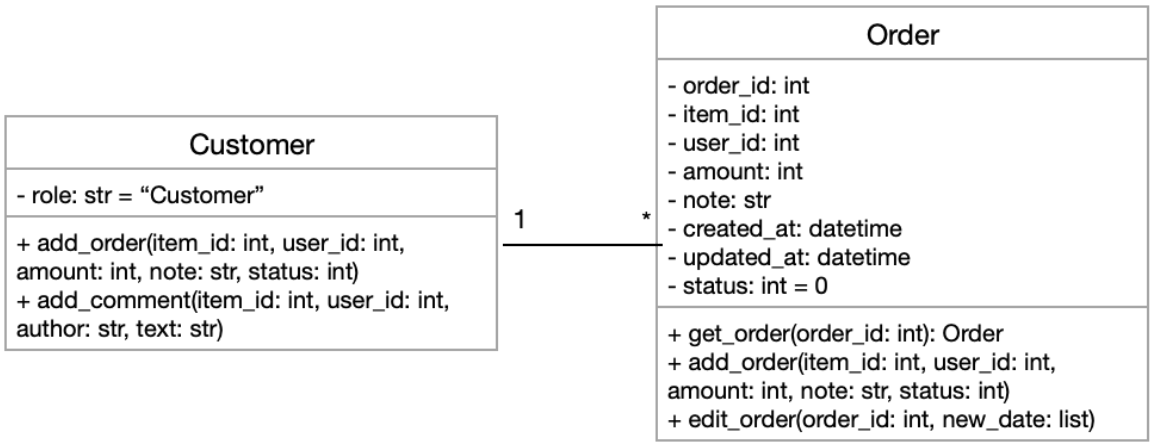
- **Interface**



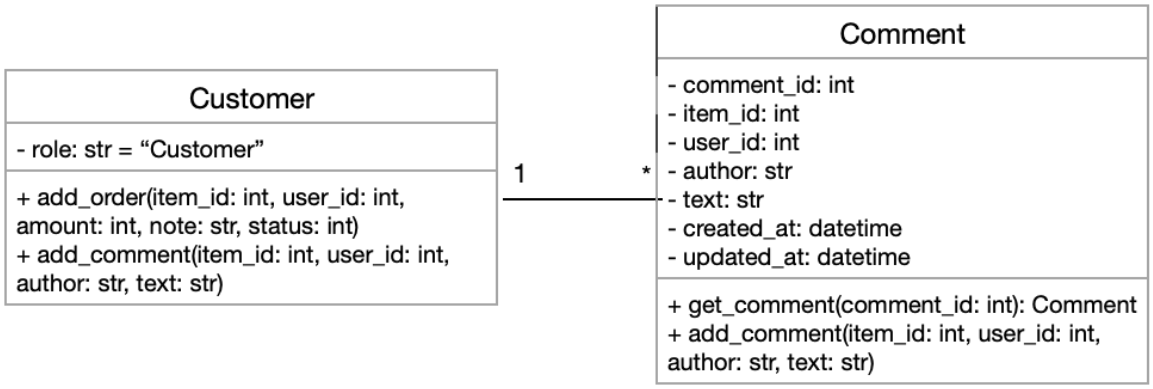
- **Association**
  - Item & Order



- User (Customer) & Order



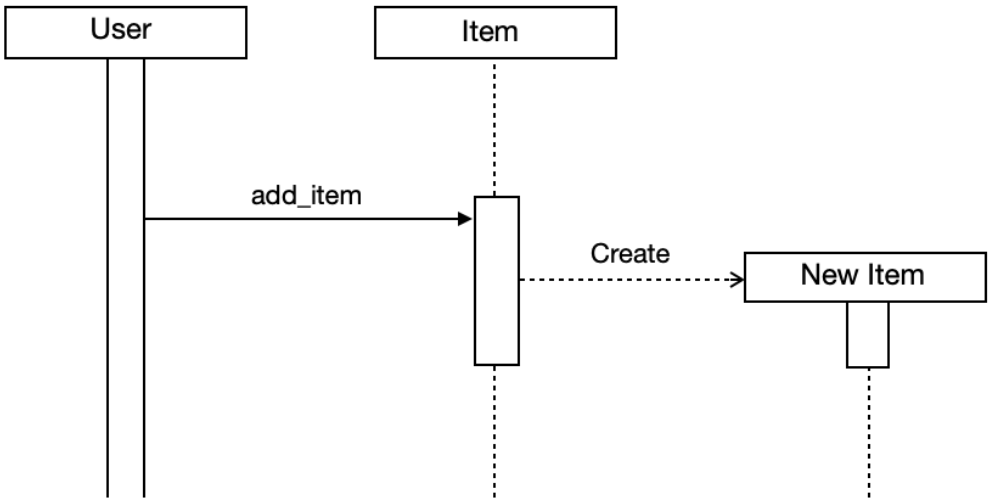
- User (Customer) & Comment



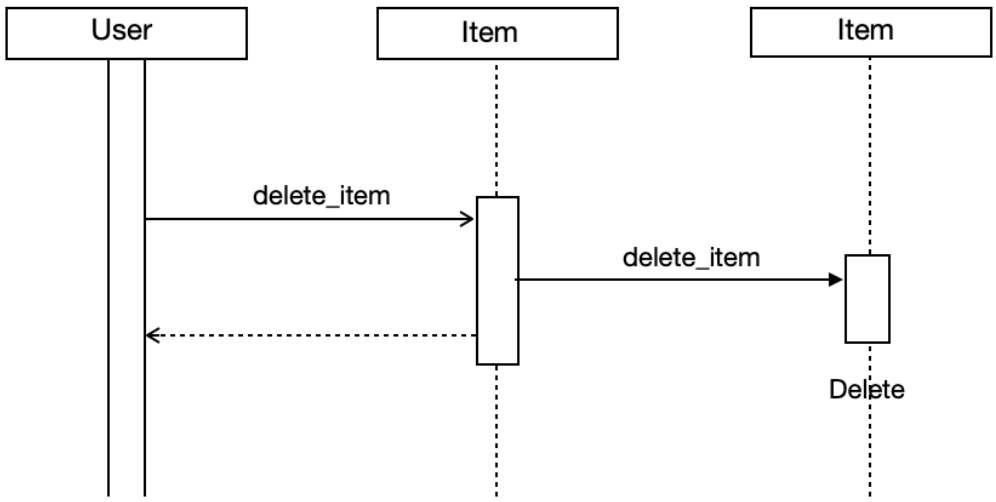
Use Case Diagram

Sequence Diagram Example

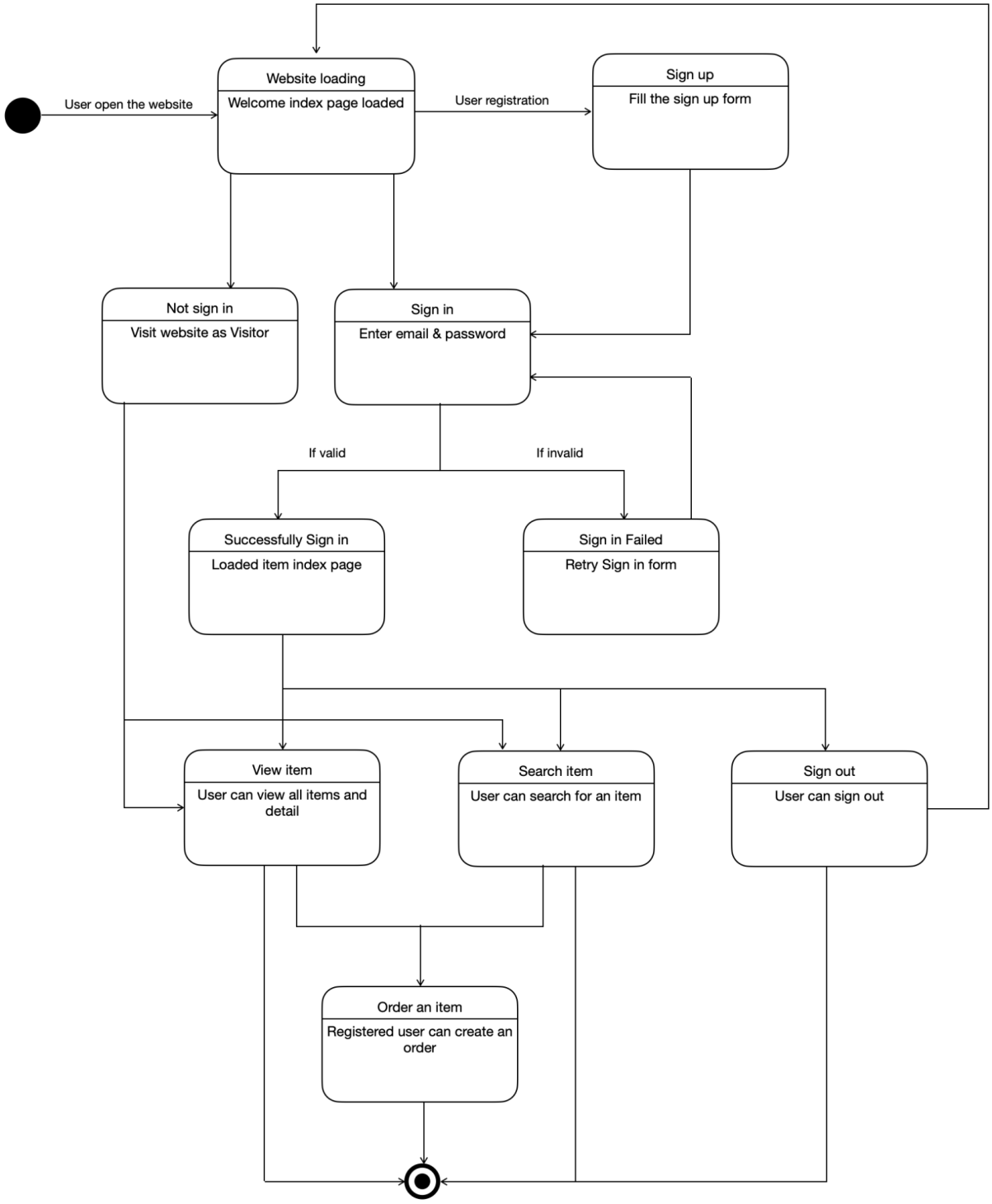
- **Add item use case**



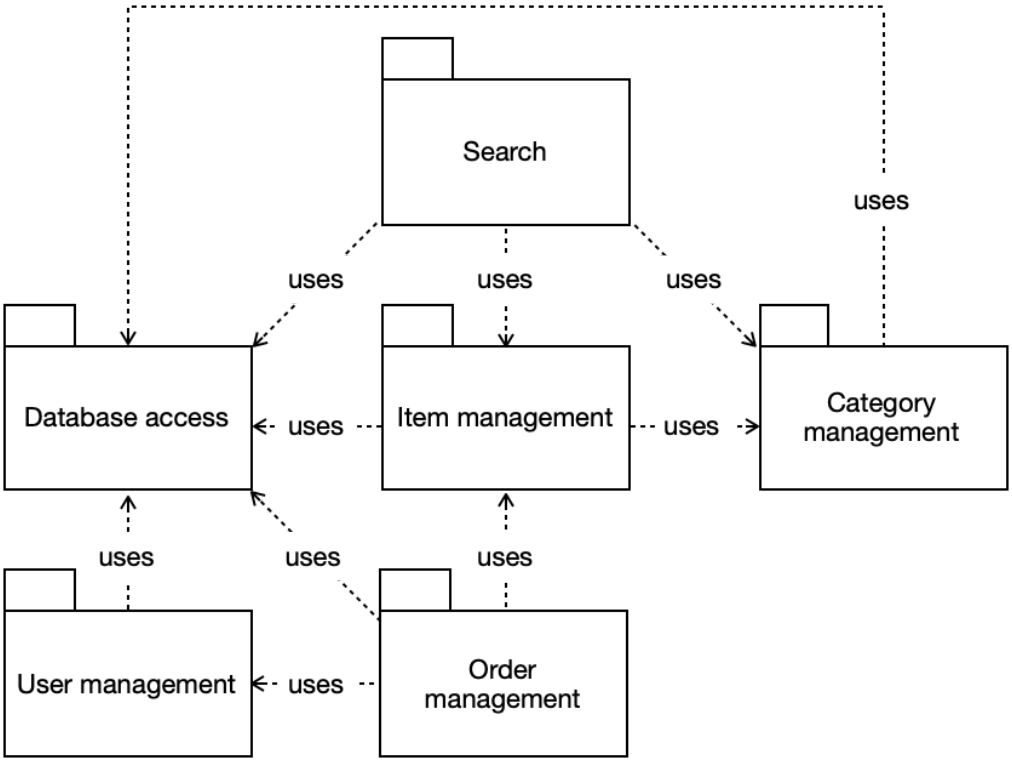
- **Delete item use case**



State Machine Diagram

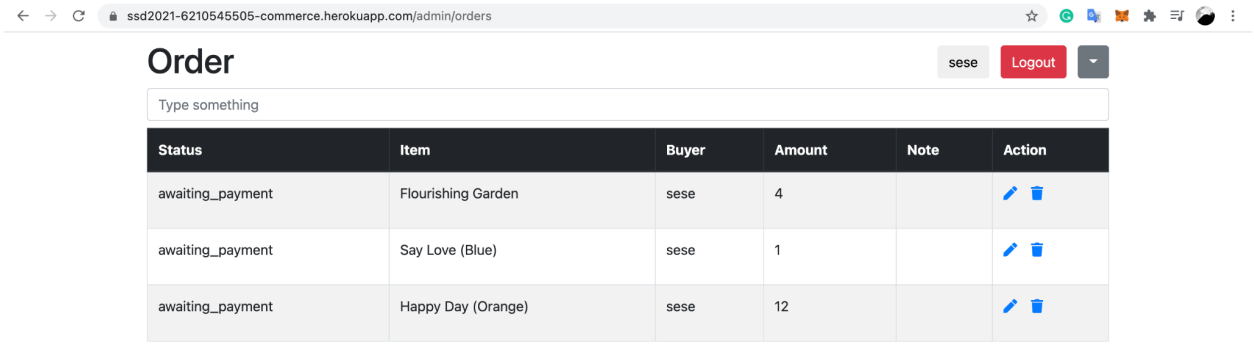


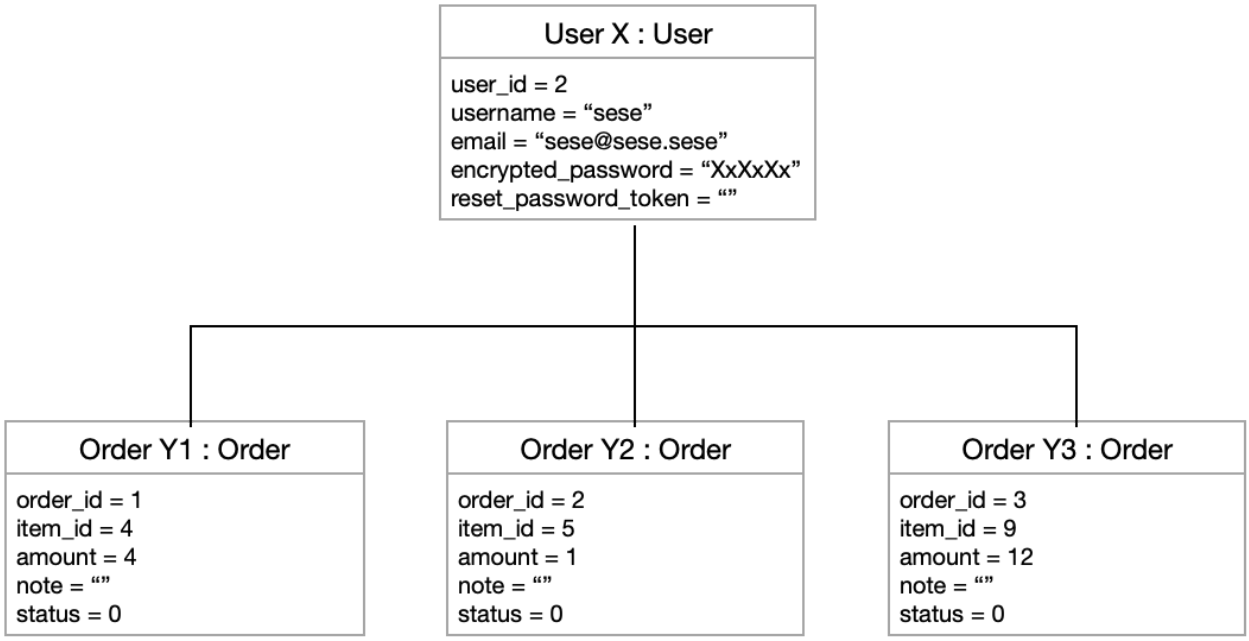
Package Diagram



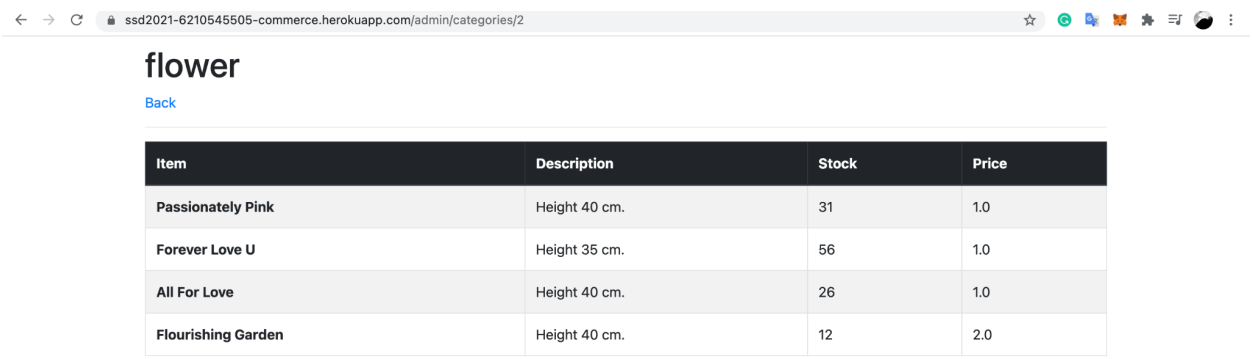
Object Diagram Example

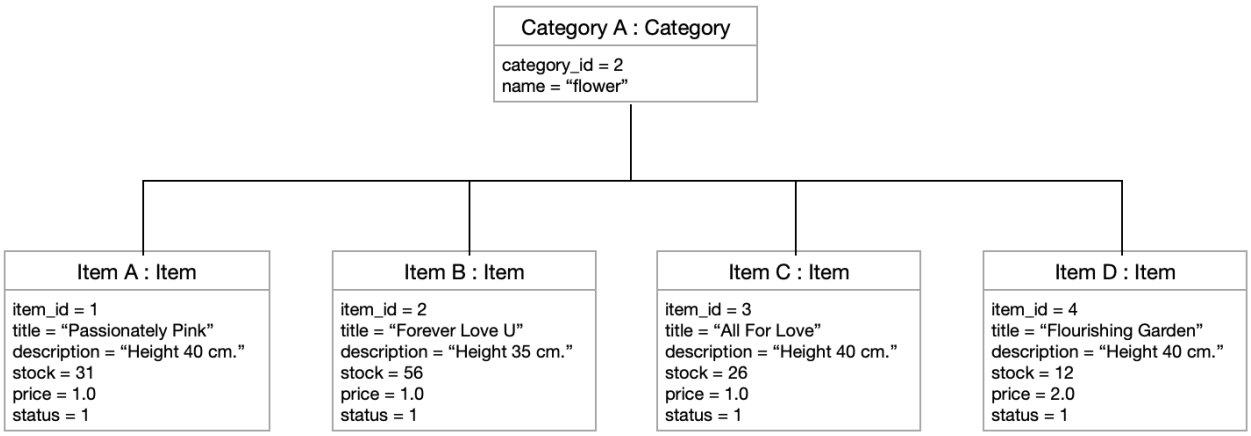
- **User vs Orders**



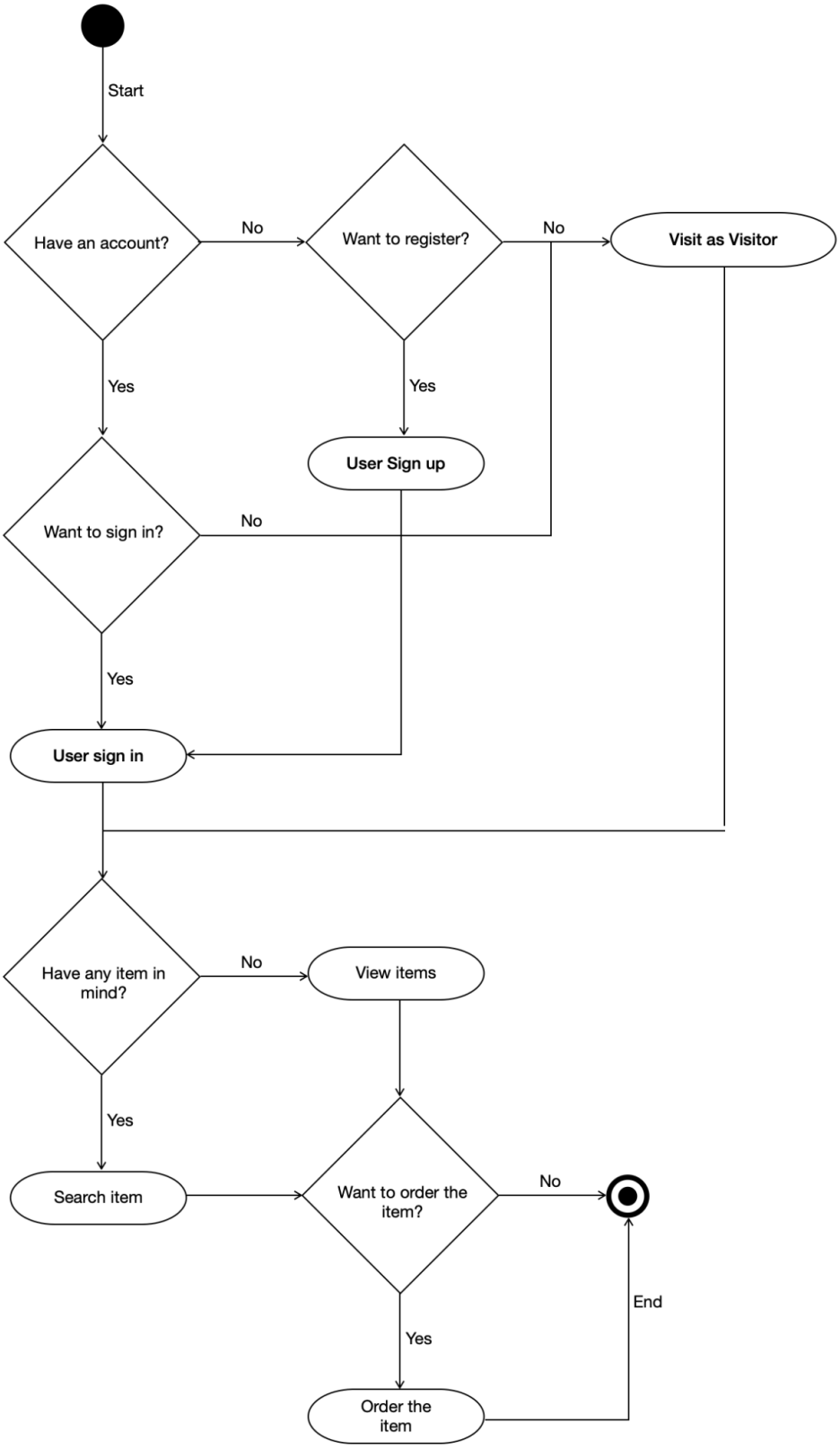


- **Category vs Items**





Activity Diagram



Communication Diagram