
Design Document

for

<ITKart>

Version <1.0>

Prepared by

Group : Team 9

Group Name: Matrix

Akshunya Vijayvargiya	200092	akshunyavijayvargiya@gmail.com
Kushal Gehlot	200541	kushal9427@gmail.com
Subodh Kumar	201007	9703subodh@gmail.com
Kevalkumar Solanki	200991	kevalsolanki49049@gmail.com
Jasjot Singh	200468	jasjotb02@gmail.com
Kumar Arpit	200532	kumararpit987@gmail.com
Samarpreet Singh	200848	singh.samar23@gmail.com
Saurav Kumar	200906	sauravsagu@gmail.com
Mayank	200568	mayankmayank67936@gmail.com
Vaidik Sharma	201079	underscoresharma@gmail.com

Course: CS253

Mentor TA: Swastik Maiti

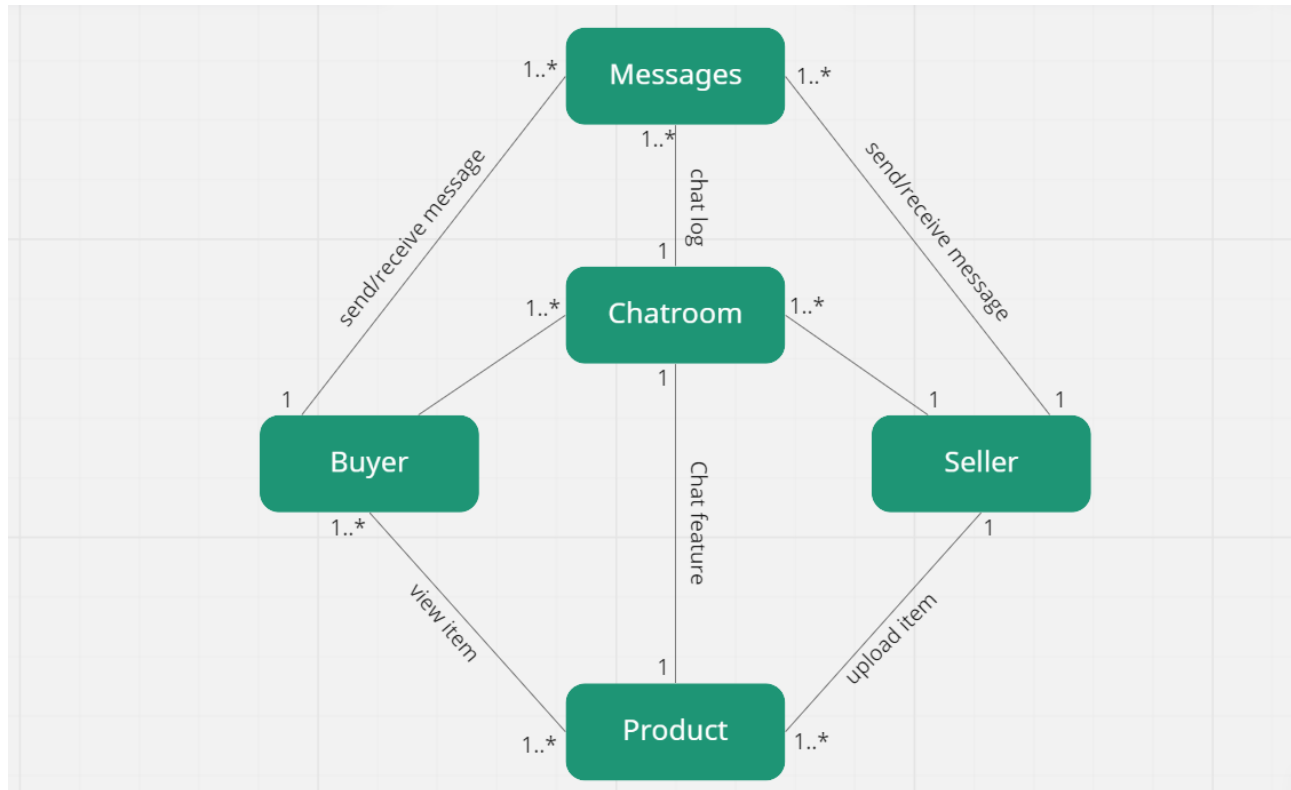
Date: 15/02/2022



CONTENTS	II
1 CONTEXT DESIGN	4
1.1 CONTEXT MODEL	4
1.2 HUMAN INTERFACE DESIGN	4
2 ARCHITECTURE DESIGN	11
3 OBJECT-ORIENTED DESIGN	11
3.1 USE CASE DIAGRAM	11
3.2 CLASS DIAGRAM	11
3.3 SEQUENCE DIAGRAM	11
3.4 STATE DIAGRAM	11
4 PROJECT PLAN	11
APPENDIX A - GROUP LOG	11

1 Context Design

1.1 Context Model



1.2 Human Interface Design

If we divide the user interface into two parts:

1.2.1 Client interface:

Firstly, they will be able to see a login page. After their user verification, they can view the home page. The homepage will contain different facilities available such as buy, sell, bid. So after this, we can see the interface will change from a different point of view.

- **Seller point of view:**

The seller will be able to decide if he wants to sell/bid. There will be options to fill in details (such as photo, price, description/condition, ..) of the product he wants to sell/rent/put in the auction and update them or delete the item itself, if needed. There

will be chat options available if some buyer is interested. The seller can also decide to show share his/her contact details to interested customers.

- **Buyer point of view:**

The buyer will be provided with search option for the product he wants. After that, available options will be shown. He/she can see the details of a product he is interested in and also will be able to chat with the seller. The buyer can report the seller if there is some discrepancy in the product description.

1.2.2 Admin interface:

An admin will be able to see the login details of users to check authentication. Also, the entire database of sold items/ items on rent will be available with the whole history in the database. An admin can see if some user is reported for inappropriate behavior for further action. He can update information, price, add or remove items on-sold.

1.2.3 Hardware interface:

This online software should provide good hardware specifications on mobile/PC. Such as,

- Enough powerful Decent processor
- Enough memory
- Decent internet connection

1.2.4 Software interface:

The client must have the required software to run the website smoothly.

- Operating system(Windows, Linux, Mac,..)
- Modern updated Web Browser (Google Chrome, Mozilla Firefox, MS Edge, ..)

2 Architecture Design



The architecture design of these software will be based on a **Layered architecture model**. It will be basically a two tier system consisting of a client interface and a database made using **MongoDB** **SQLite**.

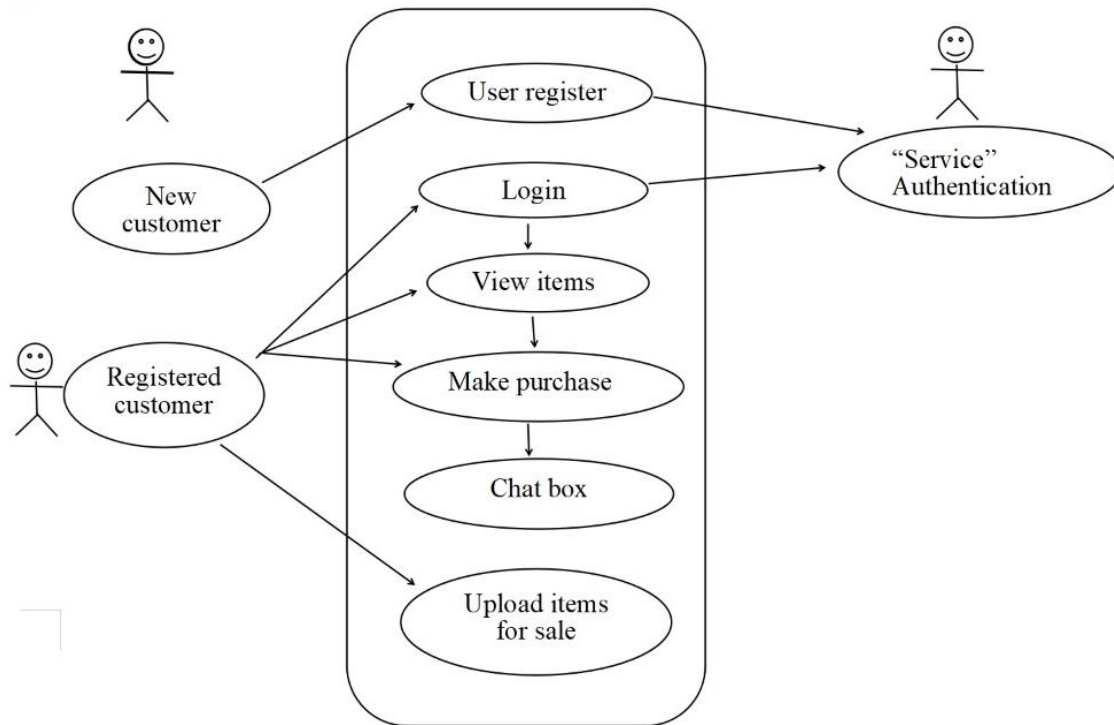
First layer comprises user authentication and role checking. Role checking here means whether the user wants to sell/rent or buy a product. The user would require to enter his login credentials in order to access the website. In Signup, features like confirmation of password and verifying already registered users would be added.

Second layer comprises Client Interface for uploading of product as well as the chat feature that will be enabled once the user is interested in purchasing. It also includes item management, which would be used to add item to the database, and dashboard, which has a list of all the products.

Last layer is the database with user details, synchronisation of chats, etc. This will be responsible for updating the database with new users and items uploaded, sold, etc. The items and selling user would be mapped to each other, which ensures more user-specific functionalities.

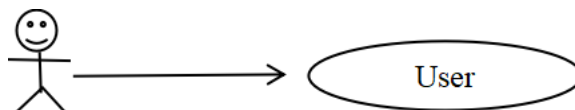
3 Object Oriented Design

3.1 Use Case Diagrams



Use Case #1 : Register Account

Author: Mayank



New Customer

Description - Register a new customer account with the system.

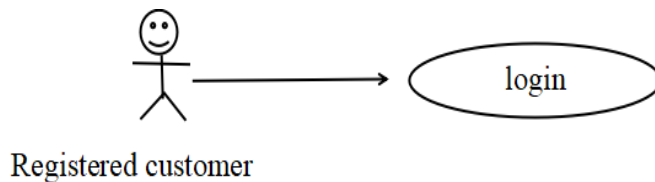
Steps:

1. Customer first clicks on the button or link to initiate the registration process.

2. System prompts the customer to fill out his/her first name, last name, username, email address, and their password.
3. Customers enter fields.
4. System validates the customer's information.
5. System creates a new account for the Customer.
6. System displays an account home page to Customer.

Use Case #2 : Login

Author: Mayank



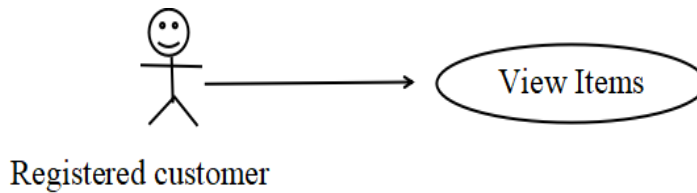
Description - Login to a customer account with the system.

Steps:

1. Customer clicks on the button or link to initiate the login process.
2. System prompts the customer for his/her username and password.
3. System verifies the information.
4. System displays account home page to the Customer

Use Case #3 : View Items

Author: Mayank



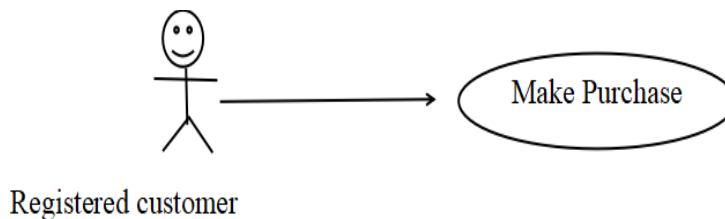
Description - Customer can search the desired product depending on whether he/she needs to purchase/rent the product.

Steps:

1. Customer enters a category description or name of the product in the search engine.
2. System searches for the matching product.
3. System displays the results.

Use Case #4 : Make Purchase

Author: Kushal



Description - Customer places an order.

Steps:

1. Customer clicks the button or link to initiate the confirmation process.
2. System directs the customer to a chat box.

Use Case #5 : Chat Box

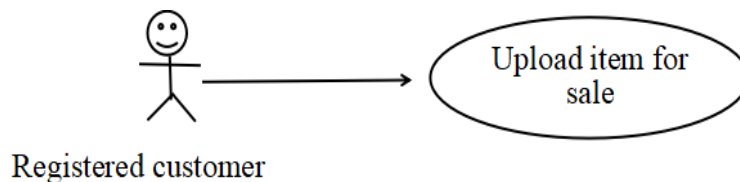
Author: Kushal



Description - Customers get access to chat with sellers for negotiation.

Use Case #6 : Upload item for sale

Author: Kushal

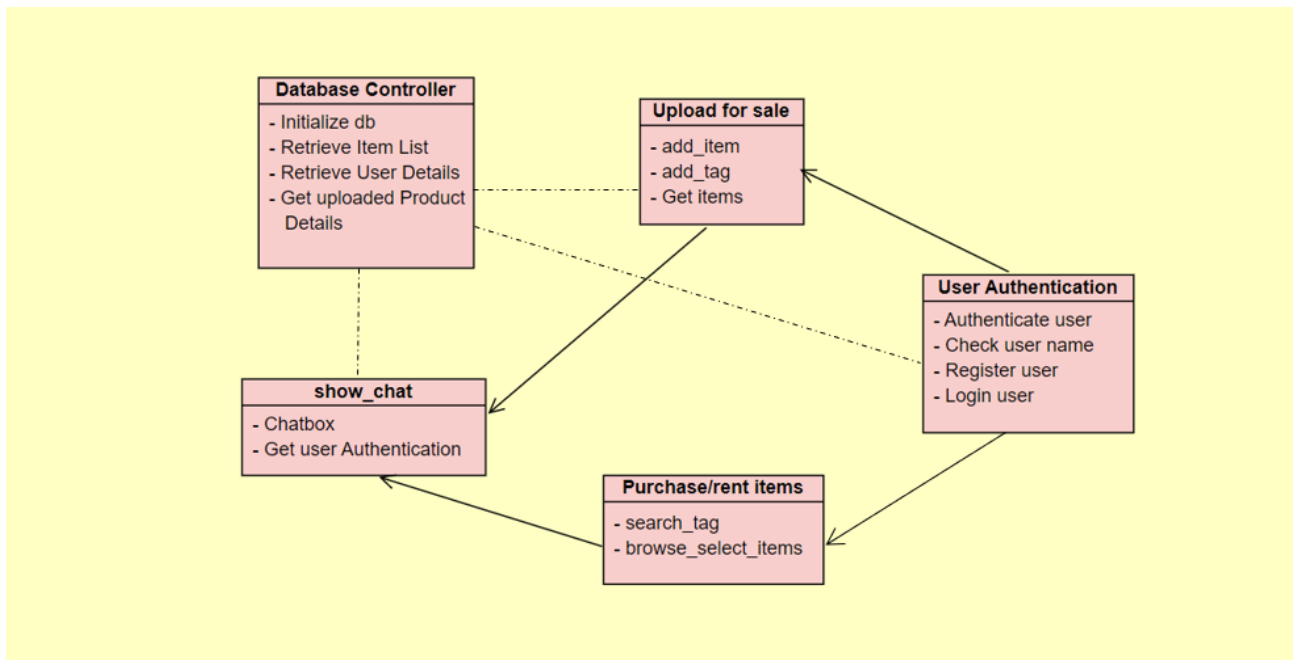


Description - Customers can upload items for sale on the website.

Steps:

1. Customer has to choose whether he wants to sell/rent his product.
2. User has to upload an image of that particular item.
3. Describe the current condition of the item.
4. Click the upload button.
5. System uploads items in sell.

3.2 Class Diagrams



3.2.1

User Authentication: This class is utilized to get user information from the database and is for authenticating the users. The class diagram in Figure 4 shows the methods that are used in this class and the description of each class is listed below.

Authenticate User: This message is used to authenticate a particular user who has provided the login credentials and wishes to login in the system. This method checks the credentials in the database.

Check User Name: This method checks to see if the provided username already exists in the database. If there is an existing user with the same name, then the user is prompted to select another username to create an account.

Register User: This method allows a new user to register for an online shopping-cart account by entering a valid username and password. If the username already exists in database, the user will be prompted to choose another name.

Login User: This method allows the existing users to log in to the database with the credentials they used for first registering into the application.

3.2.2

Upload for Sale: This class is used to process all information regarding the selling of products.

add_item: This method allows the user to upload items for sales and add description to the item to be sold.

add_tag: This method allows the user to add particular tags to his product like book, cycle, color, etc.

Get Items: This method retrieves information for all the items that the user wants to sell.

3.2.3

Purchase/Rent Items: This class is used to process all information regarding the buying of all the products.

browse_select_items: This method provides information about items available for sale in which the user is interested.

Search_Tag: This method allows users to search items associated with a particular tag like rent or purchase, book, cycle, color, subject, etc.

3.2.4

show_chat: This class will be invoked when the user wants to purchase a product and will serve as a connector between buyer and seller.

Get User Authentication: This method fetches the user-authentication information. The show_chat class is invoked once authentication is done.

3.2.5

DB Controller: This class is used for getting users and product information from the database, and it is also used to update the database with the information about new-user registration, product checkout, and user details.

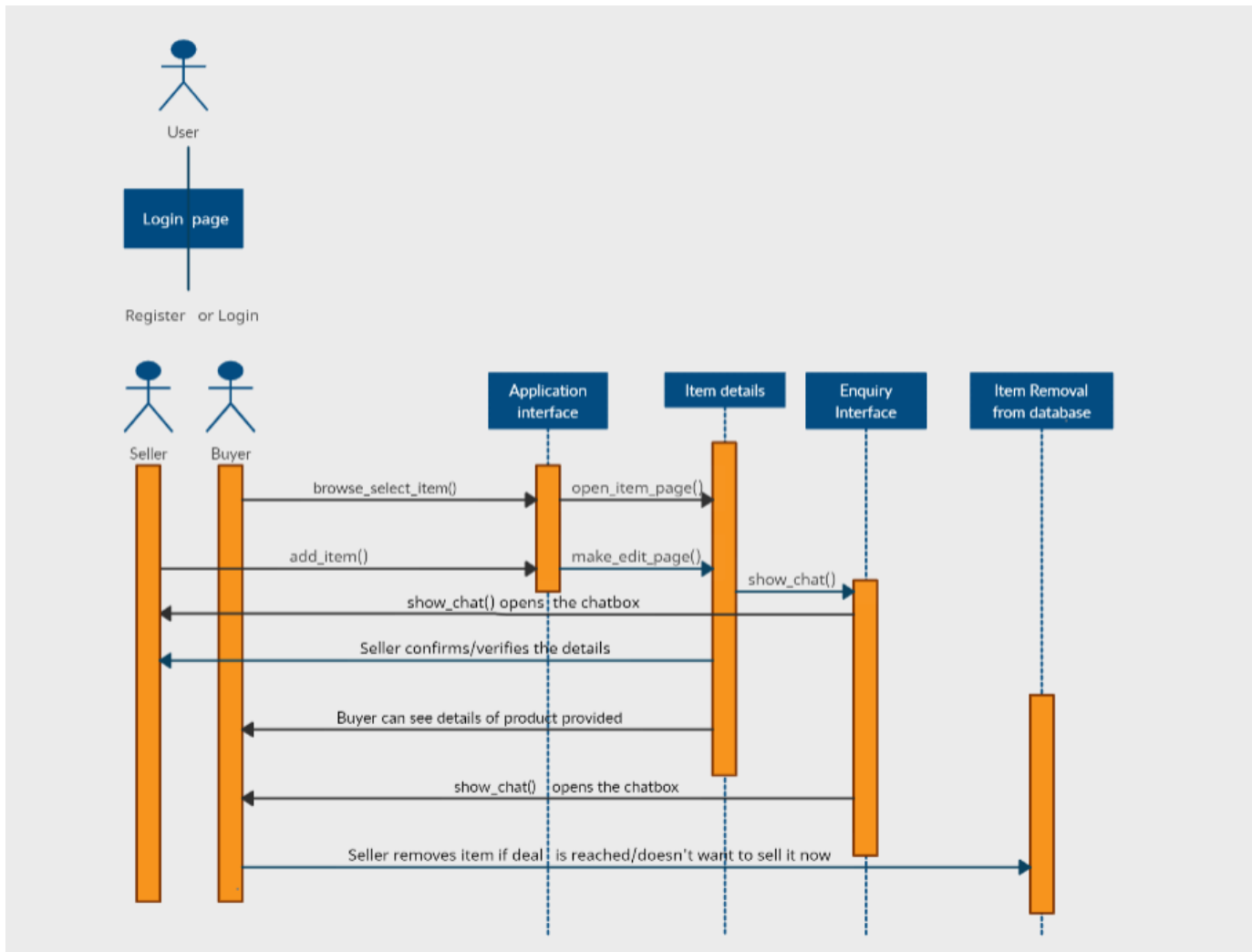
Initialize DB: This method allows the initialization of a database on the first run of the application.

Retrieve Items List: This method fetches all items from the code/workspace into the database and allows the administrator to view information about the items.

Retrieve User's Details: This role locates all the registered users in the database and also fetches any new user who registers by completing the user-authentication form.

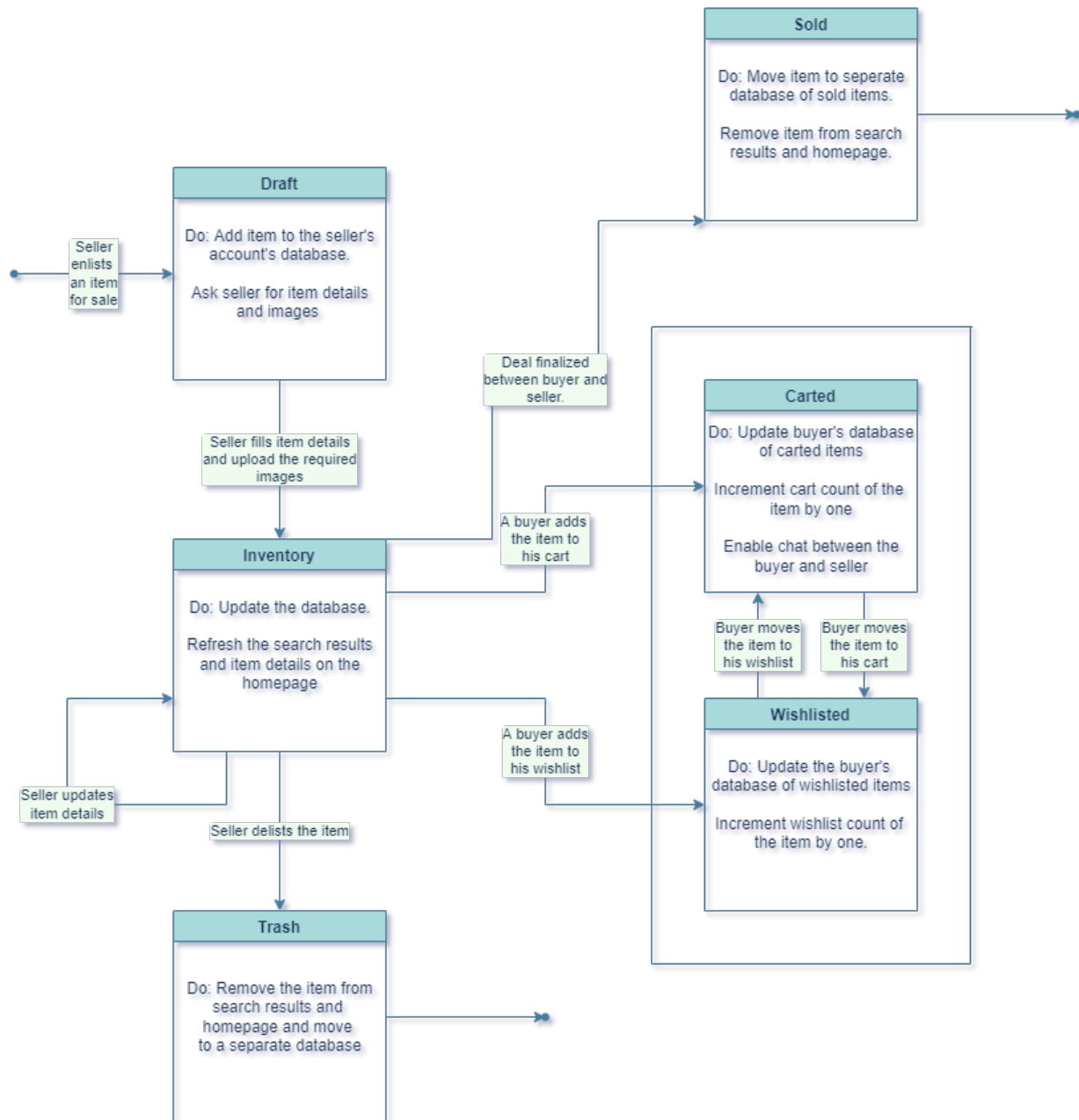
Get Uploaded Product Details: This role inputs the details of the product uploaded by the user. This role updates the row in the database to show the details of products that have been uploaded.

3.3 Sequence Diagrams



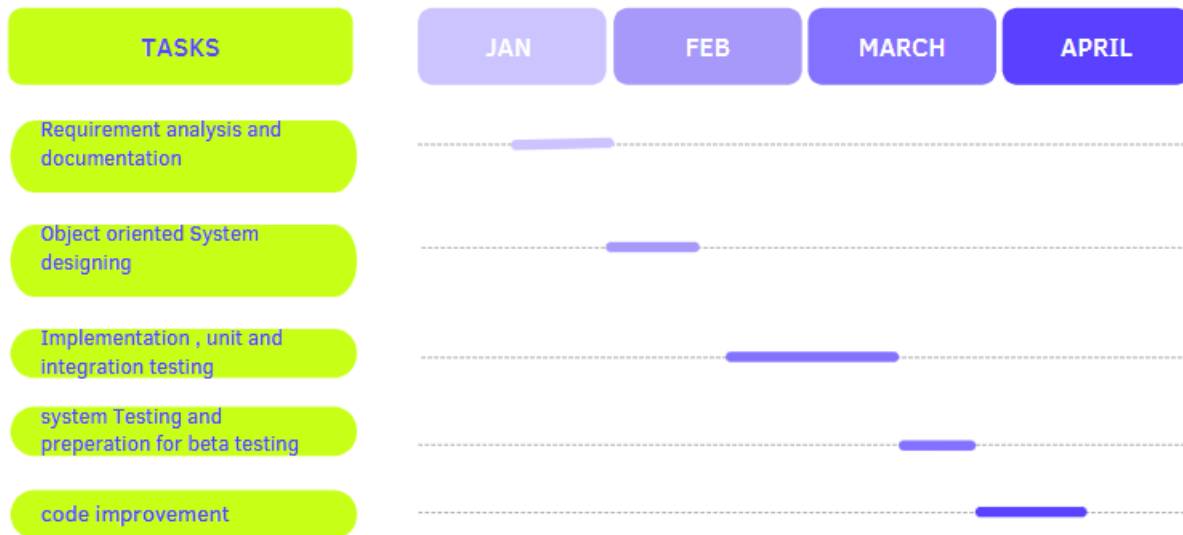
3.4 State Diagrams

State Diagram of an item enlisted for selling:



State	Description
Draft	Initial state of an object when a seller enlists it for selling. It is added to the database with the details and images parameters empty. The item is not visible in search results or on the homepage of the site.
Inventory	The items details are updated by the seller. He/she can update it multiple times which results in an update query to the database. The item is visible in the search results and homepage.
Carted	This is a state of the item on the buyer's side. After carting an item, the buyer may chat with the seller and negotiate.
Wishlisted	This state is also on buyer's side. He/she may save the item to look up at a later time and move to the cart if he wishes to buy it.
Trash	The seller may remove the enlisting from the inventory at any time. The item is removed from the database and stored in a separate one for future references.
Sold	When a deal is finalized for the item, it is removed from the inventory and is no longer visible in search results or on the homepage.

4 Project Plan



- Project idea was finalized on January 16, 2022, which was the starting date of the project.
- Then Requirement generation and analysis were done and finalized requirement documentation was submitted on January 1, 2022.
- Currently Object-oriented design is being made and finalized Design Documented would be completed and submitted by 15th February 2022.
- From 16th February 2022, practical implementation work would commence. It would be completed by February 28, 2022.
- Then testing would be done-
 - Unit testing will be completed by March 6, 2022.
 - Integration testing would be completed by March 12, 2022.
- First code in GitHub Repository and Test Document would be submitted by March 13, 2022.
- March 14 to 21, 2022 would be utilized for system testing.
- Formation of manual for beta testing would be done in time from 21st to 26th March and final System Manual would be submitted by March 27, 2022.
- Improvement of code would be done and then final project code will be submitted by April 10, 2022.

Work Distribution

Different sections of Requirement Document and Design Document were divided between group members for completion as 2 members per section.

Front end design using **React JS, javascript JS frameworks**, HTML and CSS would be done by *Kushal Gehlot, Saurav Kumar and Kevalkumar Solanki*.

Classes implementation would be done as following :

- User Authentication - *Akshunya Vijayvargiya*
- Upload for sale - *Jasjot Singh*
- Purchase/Rent Items - *Samarpreet Singh*
- show_chat - *Mayank*
- DB controller - *Kumar Arpit*

Databasing using MongoDB would be handled by *Vaidik Sharma and Subodh Kumar*.

Update (as of 24th April, 2022) :

We've used django framework, which by default has SQLite as database, hence no database-specific team was needed.

The group contribution (in implementation) is as follows -

Front end Designing - *Kushal Gehlot, Akshunya Vijayvargiya*

Integrating Frontend and Backend - *Kushal and Samarpreet*

Back end Development:

User authentication and functionality - *Samarpreet Singh*

Item and User (one to one mapping) - *Jasjot Singh*

Product management functions/classes - *Samarpreet Singh*

Search engine - *Akshunya Vijayvargiya*

Chatbox and database - *Subodh Kumar*

Appendix A - Group Log

Meeting 1 (5/2/22): 12:00am to 1:00am:

Section-wise Work was distributed between all team members and all the team members worked on specific sections of Design Documentation and regularly discussed the progress of the document.

Work done per person :

Context model - Saurav Kumar

Human Interface design - Kevalkumar Solanki

Architecture Design - Jasjot Singh

Use case diagrams - Kushal Gehlot and Mayank

Class Diagrams - Kushal Gehlot

Sequence Diagrams - Samarpreet Singh

State Diagram - Kumar Arpit

Project plan - Akshunya Vijayvargiya