Design Document

for

store.it

Version 0.1

Prepared by

Group #: 3

Group Name: localghosts

Name	Roll No.	Email
Akanksha Singh	200070	akankshas20@iitk.ac.in
Antreev Singh Brar	190163	antreev@iitk.ac.in
Bhuvan Singla	180199	bhuvans@iitk.ac.in
Deepankur Kansal	180226	deepank@iitk.ac.in
Dipanshu Garg	190306	dipanshu@iitk.ac.in
Harshit Raj	200433	harshitr20@iitk.ac.in
Hitesh Anand	200449	ahitesh20@iitk.ac.in
Manas Gupta	200554	manasg20@iitk.ac.in
Priya Gole	200727	priyagole20@iitk.ac.in
Tushar	190915	tusharb@iitk.ac.in

Course: CS253A

Mentor TA: Mr. Swastik Maiti

Date: 14 February 2022

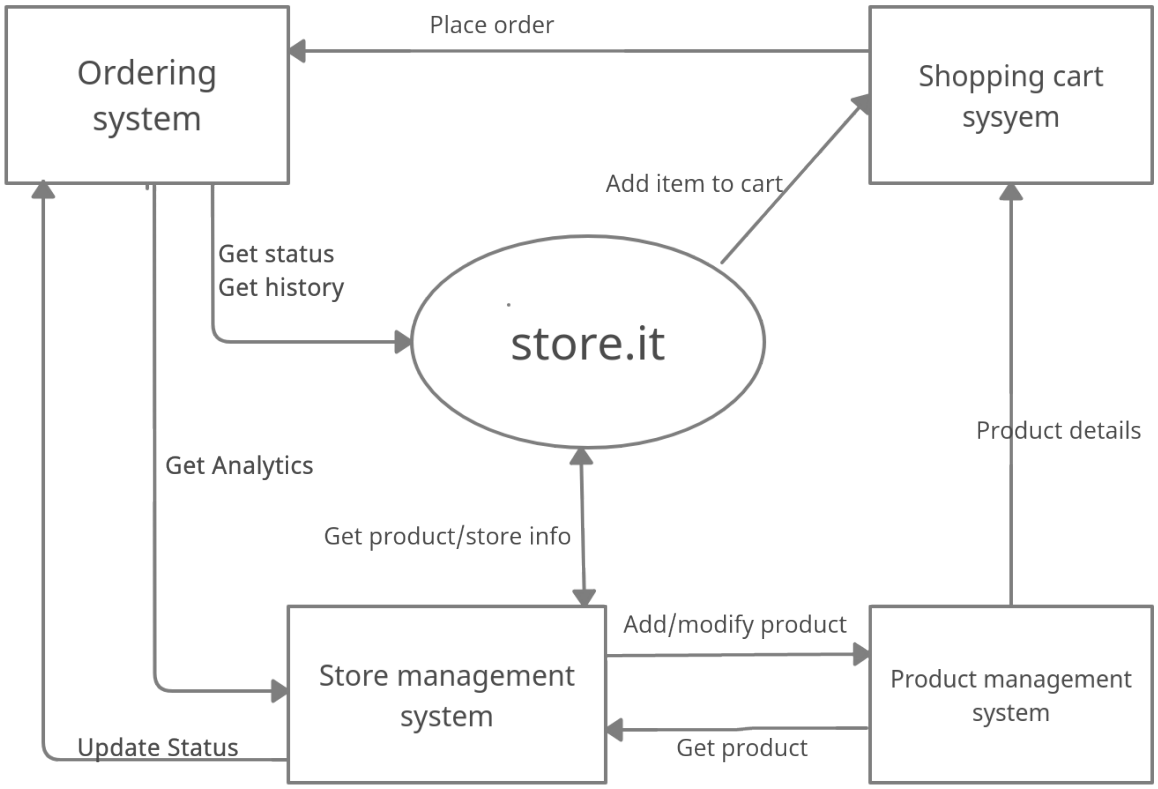
Contents

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
0.1	Akanksha Singh Antreev Singh Brar Bhuvan Singla Deepankur Kansal Dipanshu Garg Harshit Raj Hitesh Anand Manas Gupta Priya Gole Tushar	First Draft	14/02/2022

1 Context Design

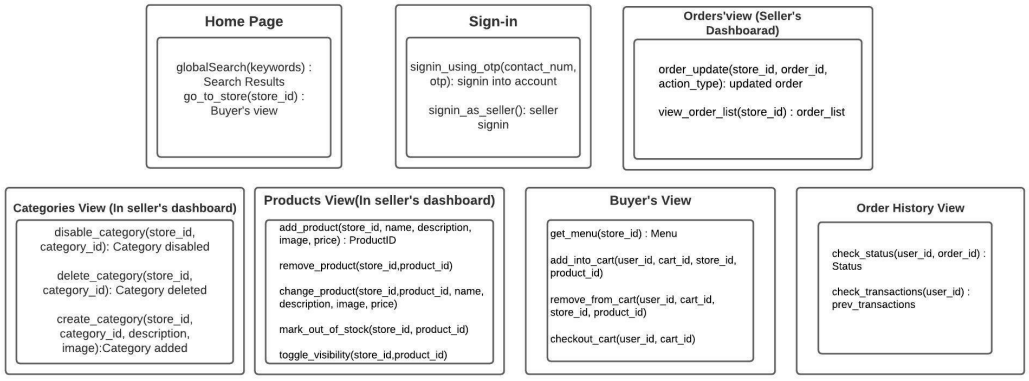
1.1 Context Model

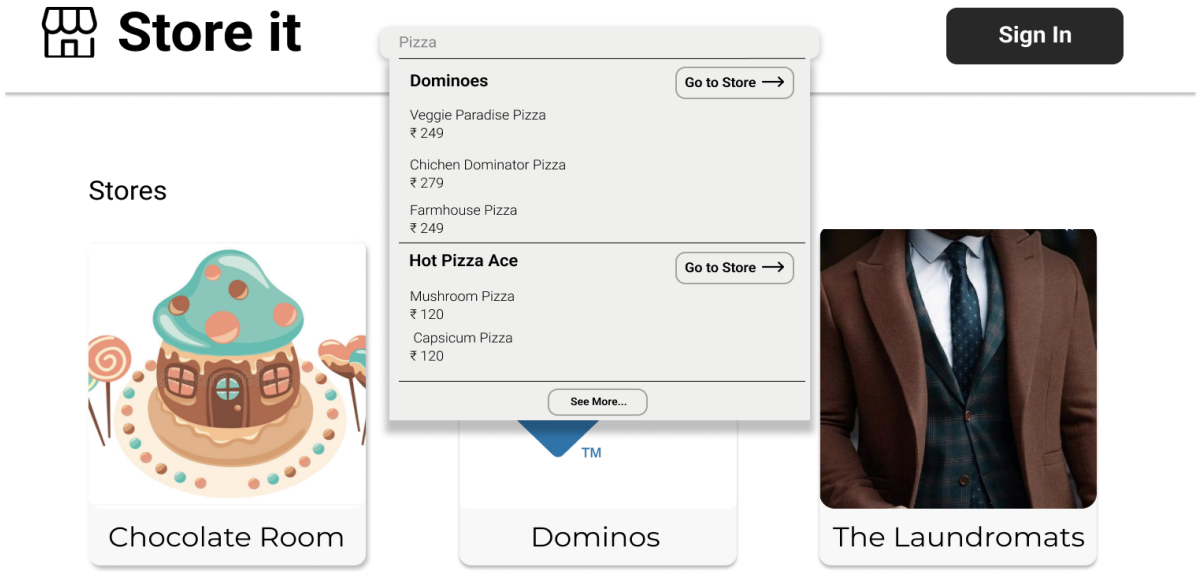


1.2 Human Interface Design

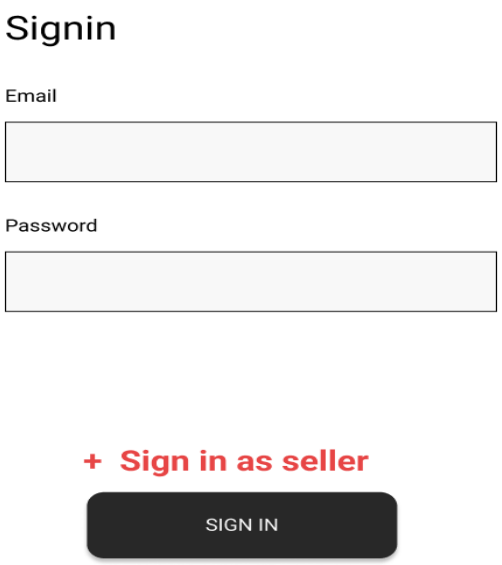
There are seven interfaces in our web-app, each having its own utility. Each interface is meant to be used for either buyers or sellers and facilitates a smooth user experience with the clients. The interface design of store-it with all seven interfaces is given below.

s

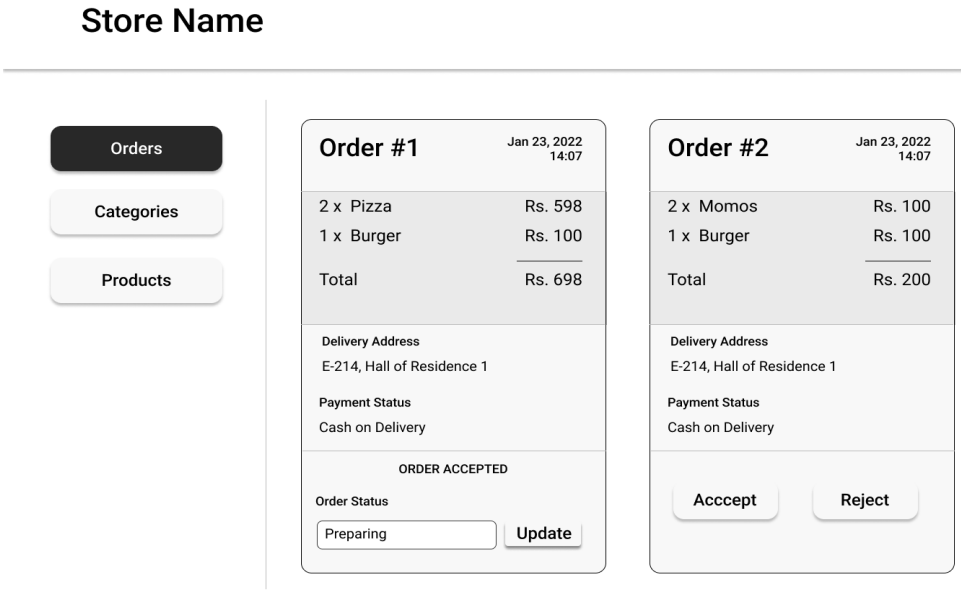

Home Page
globalSearch(keywords) : Search Results
go_to_store(store_id) : Buyer's view
Sign-in
signin_using_otp(contact_num, otp): signin into account
signin_as_seller(): seller signin
Orders'view (Seller's Dashboarad)
order_update(store_id, order_id, action_type): updated order
view_order_list(store_id) : order_list
Categories View (In seller's dashboard)
disable_category(store_id, category_id): Category disabled
delete_category(store_id, category_id): Category deleted
create_category(store_id, category_id, description, image):Category added
Products View(In seller's dashboard)
add_product(store_id, name, description, image, price) : ProductID
remove_product(store_id,product_id)
change_product(store_id,product_id, name, description, image, price)
mark_out_of_stock(store_id, product_id)
toggle_visibility(store_id,product_id)
Buyer's View
get_menu(store_id) : Menu
add_into_cart(user_id, cart_id, store_id, product_id)
remove_from_cart(user_id, cart_id, store_id, product_id)
checkout_cart(user_id, cart_id)
Order History View
check_status(user_id, order_id) : Status
check_transactions(user_id) : prev_transactions


Store it
Sign In
Pizza
Dominoes
Go to Store
Veggie Paradise Pizza
₹ 249
Chichen Dominator Pizza
₹ 279
Farmhouse Pizza
₹ 249
Hot Pizza Ace
Go to Store
Mushroom Pizza
₹ 120
Capsicum Pizza
₹ 120
See More...
Stores
Chocolate Room
Dominos
The Laundromats

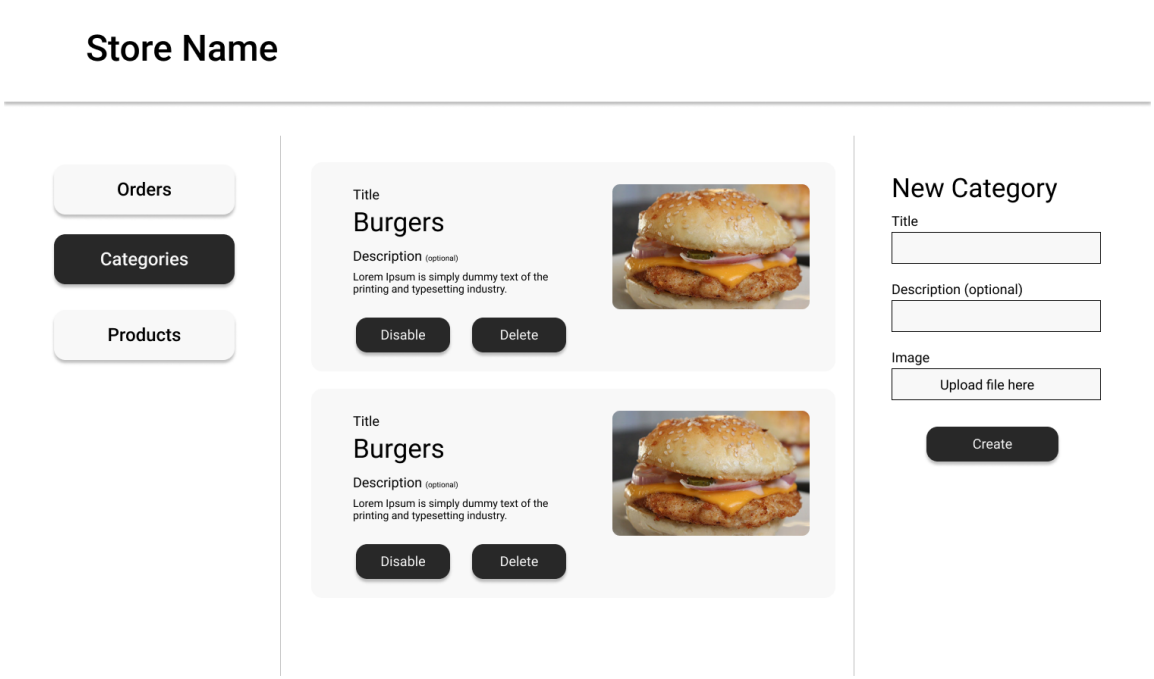
Home Page



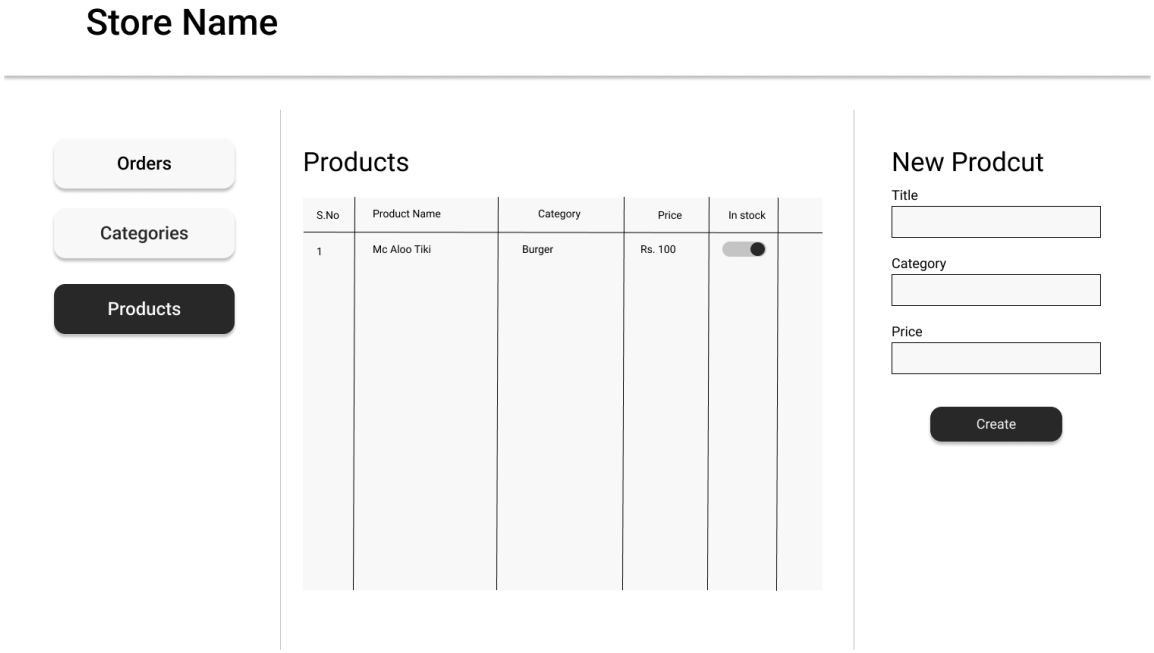
Sign-in



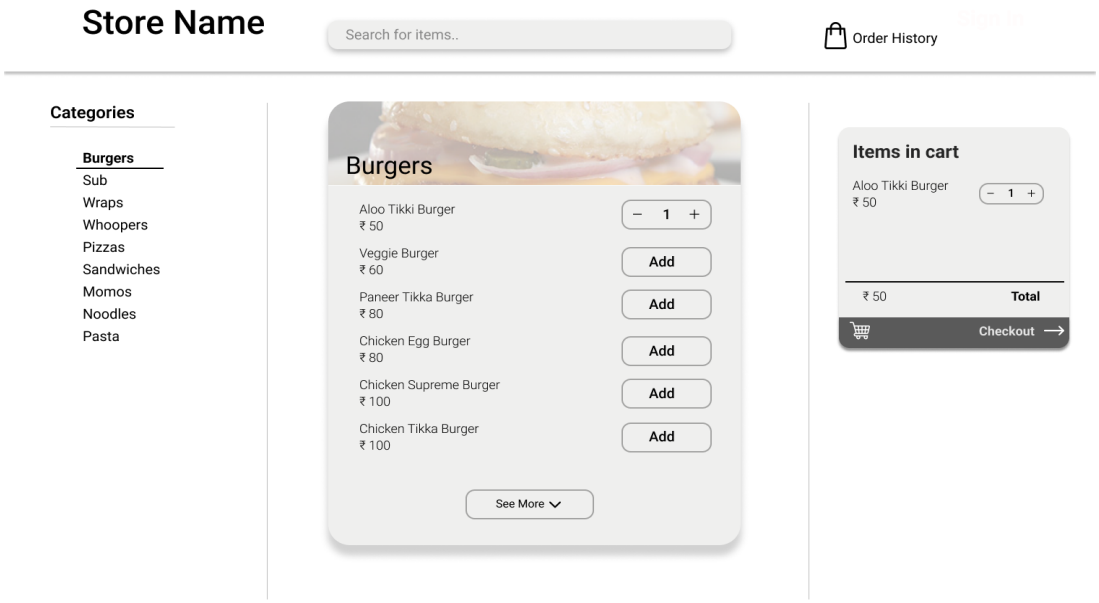
Orders' view (Seller's dashboard)



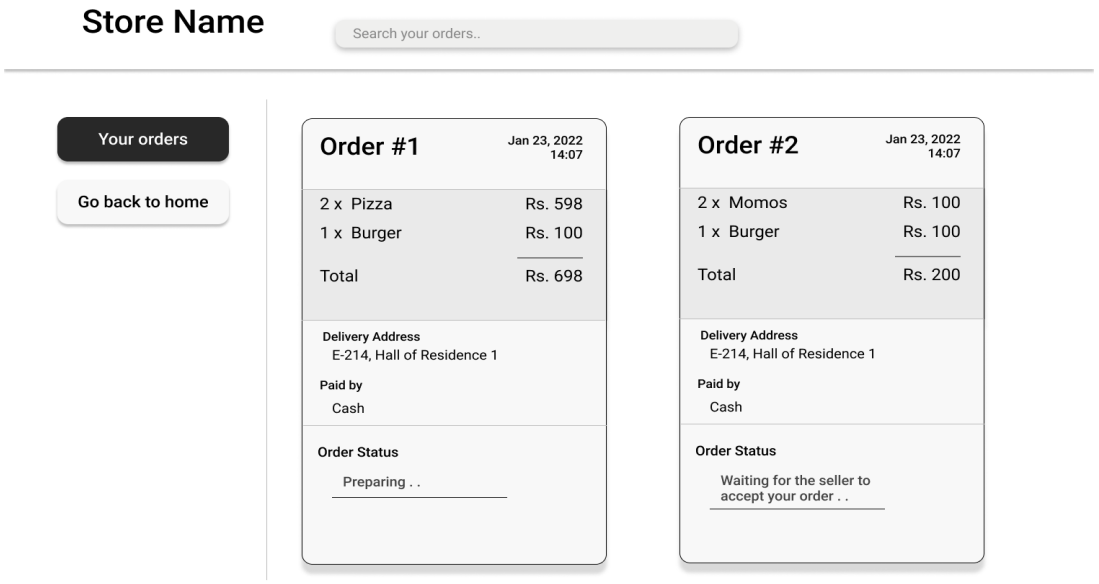
Categories' view (Seller's Dashboard)



Products' view (Seller's dashboard)



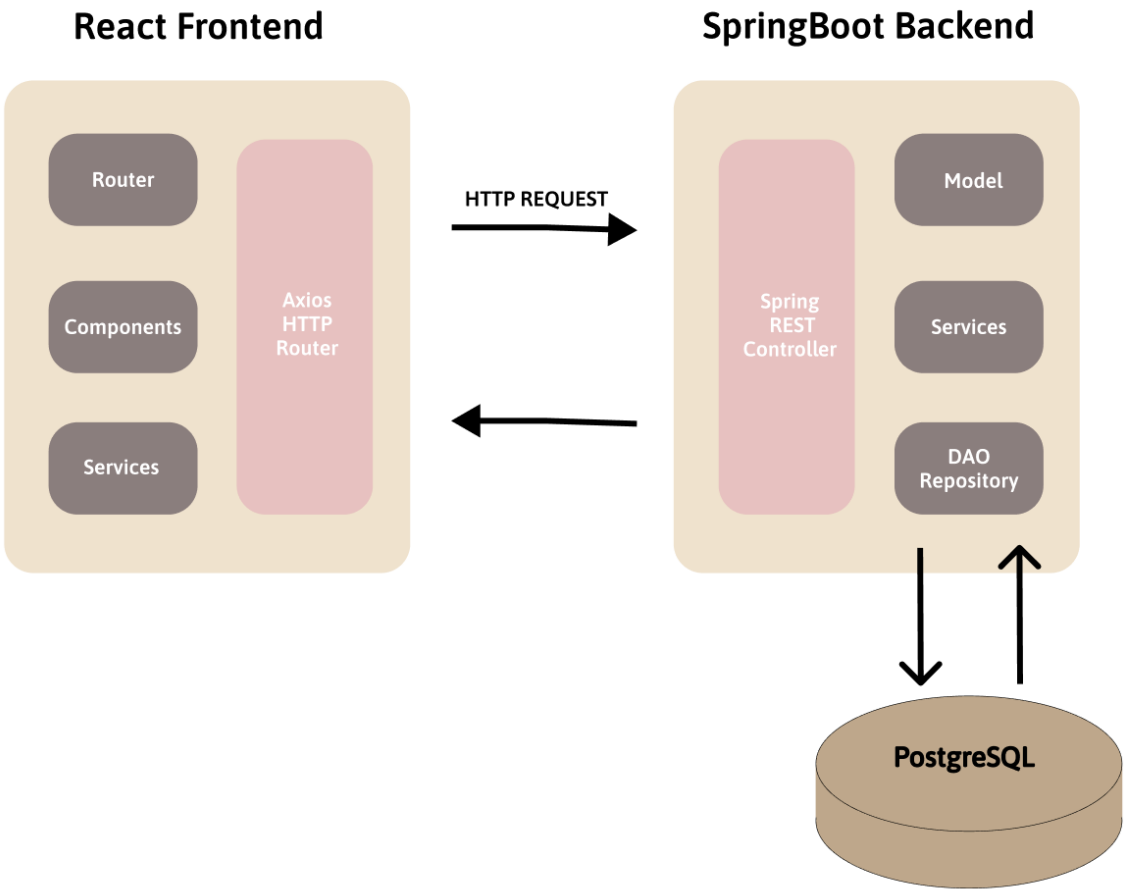
Buyer's view



Order History view

2 Architecture Design

"store.it" is a web-based application implementing a Model-View-Controller model.



Why are we using this: We are using this model since we would be interacting with the data in multiple ways, like checking and receiving requests, managing catalogues, storing transactions. Also, in the near future, one might think of adding some functionalities to it, which makes the Model-View-Controller model as the best fit.

Advantages: One great advantage of this model is that the data can be represented in various manners. Which will prove helpful to us while sorting items based on similar kinds, or from the same stores or in some other fashion. Also, it allows the data to change independently; it would be helpful if something runs out of stock.

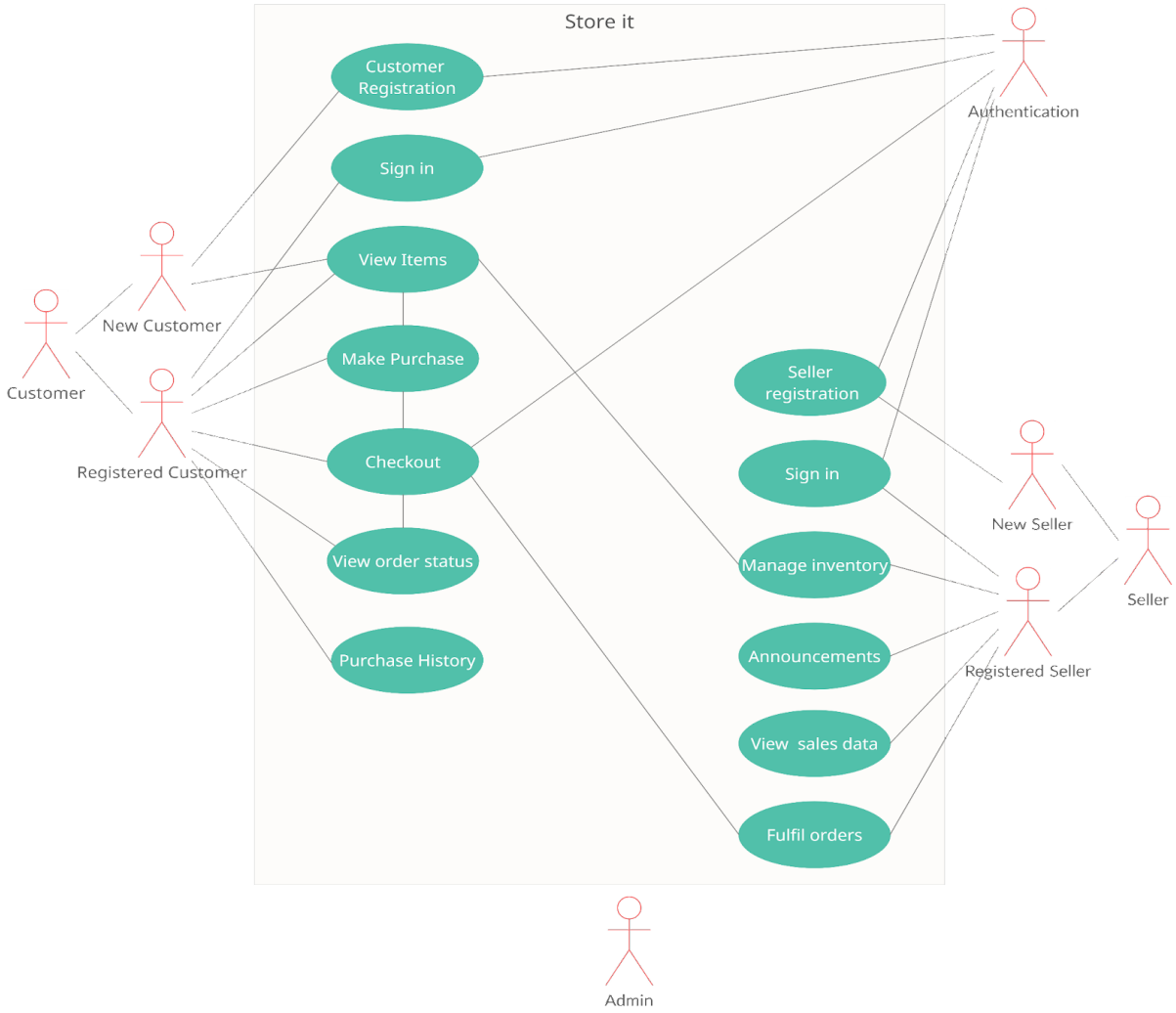
Disadvantages: One disadvantage is that we need to write some additional code for full functionality, even if we know that our implementation is simple in the starting stages.

3 Object-Oriented Design

3.1 Use Case Diagrams

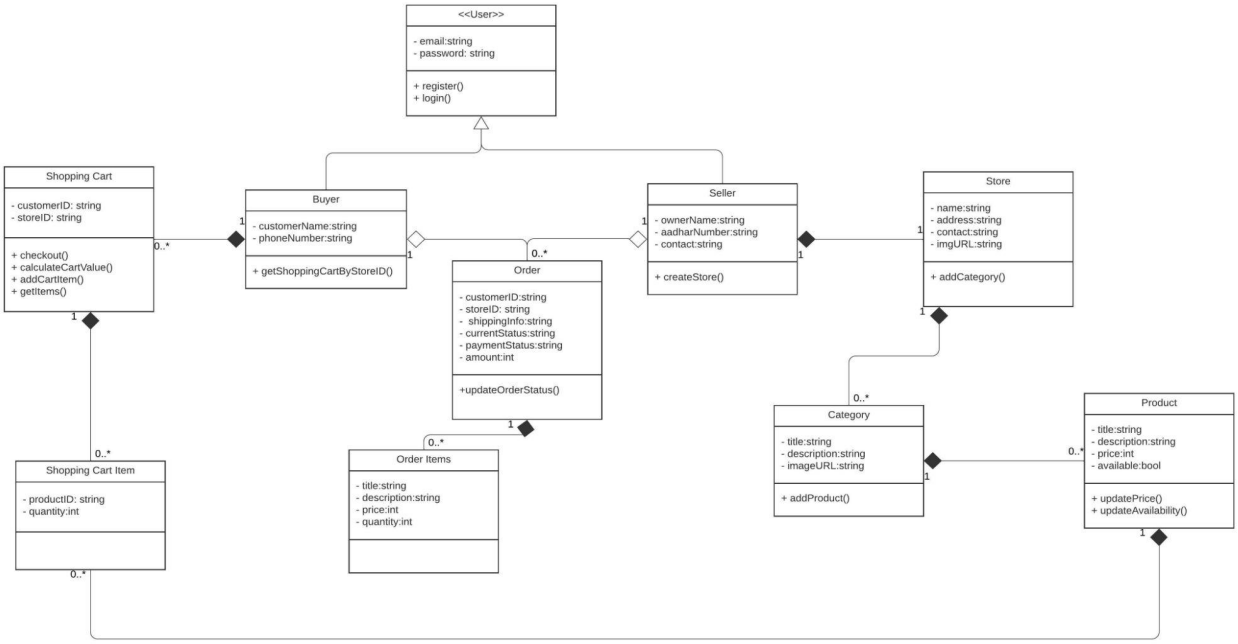
We have given a consolidated diagram showing different use cases provided:

- Shopkeeper registration and catalog creation.
- Browsing catalog of a particular shop
- Items being sold by the shopkeeper and short descriptions and listed prices.
- Adding items to cart.
- Register as a buyer using email and OTP verification.
- Checking out and placing orders.
- Shopkeeper adding/removing items from shop inventory.

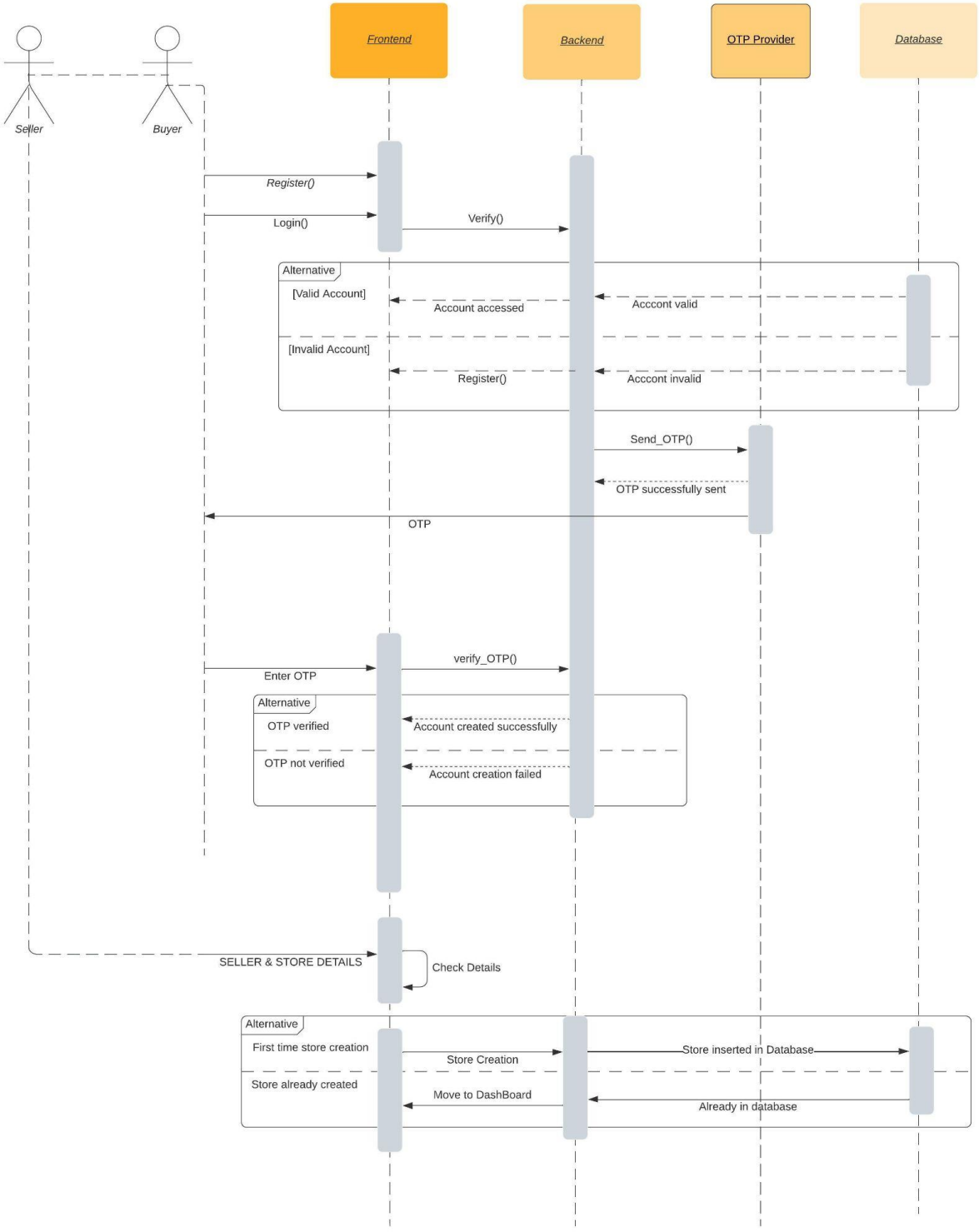


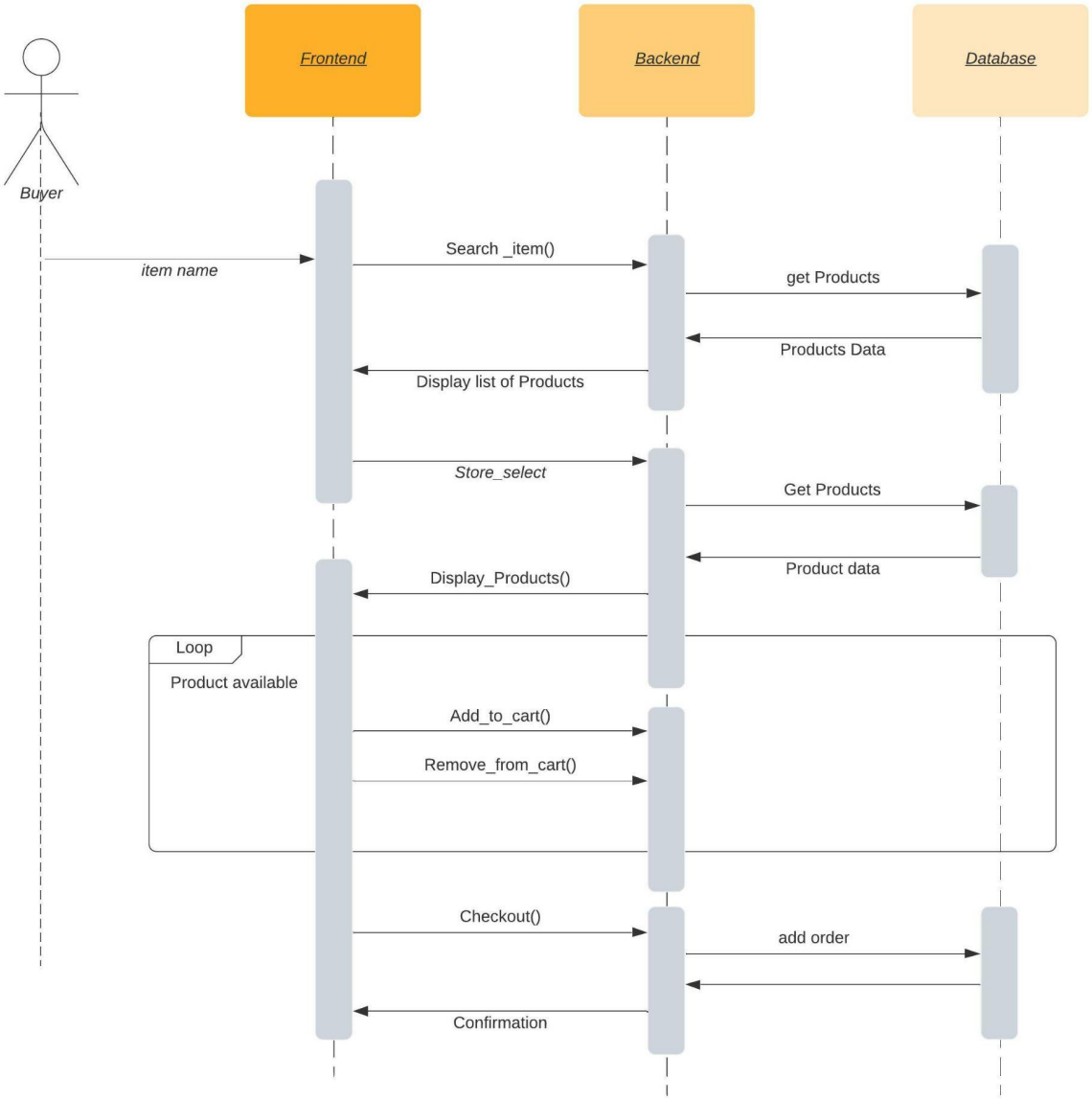
3.2 Class Diagram

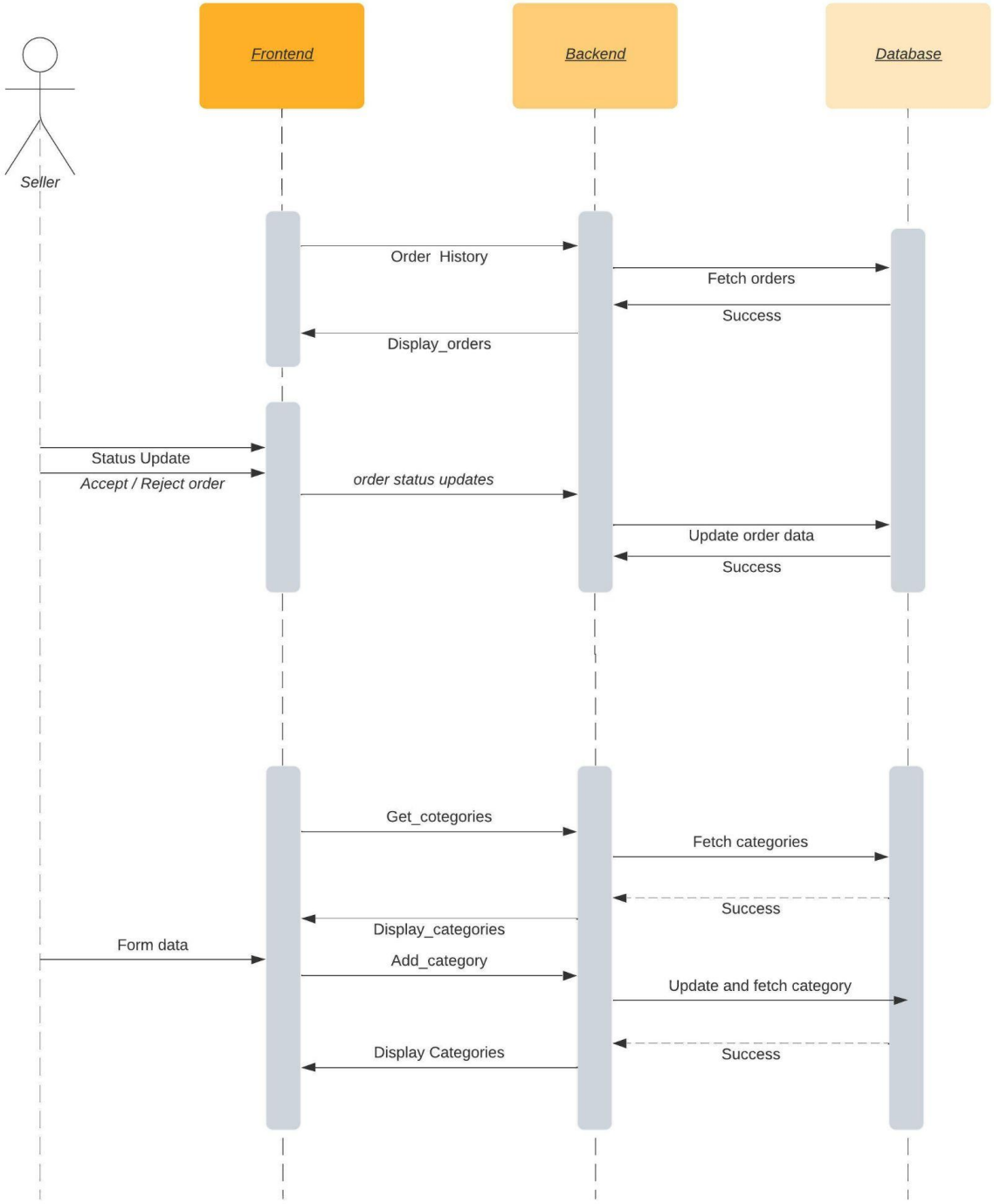
store.it

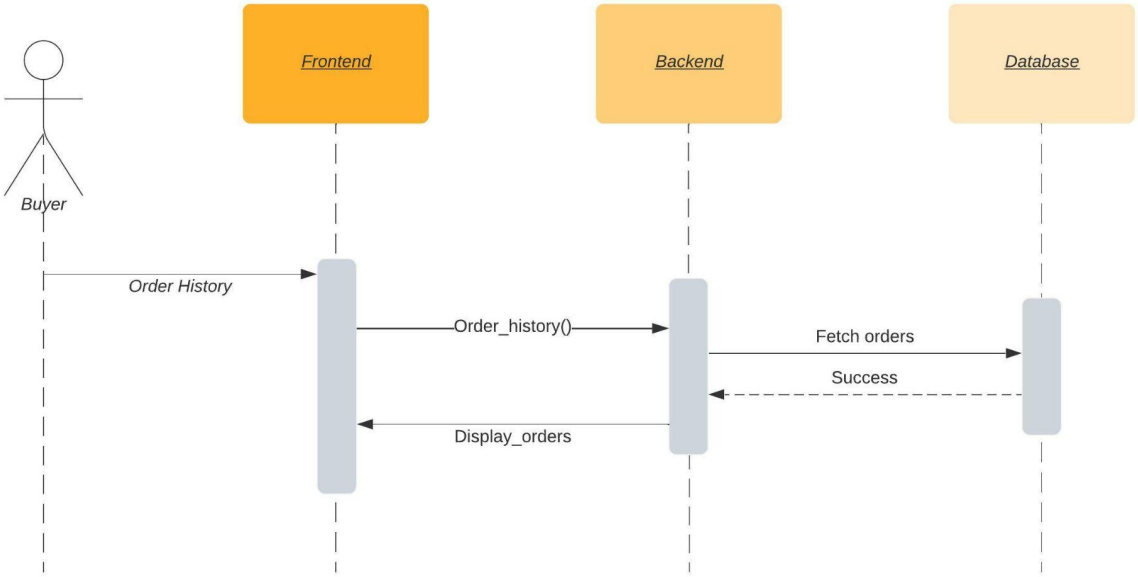


3.3 Sequence Diagrams

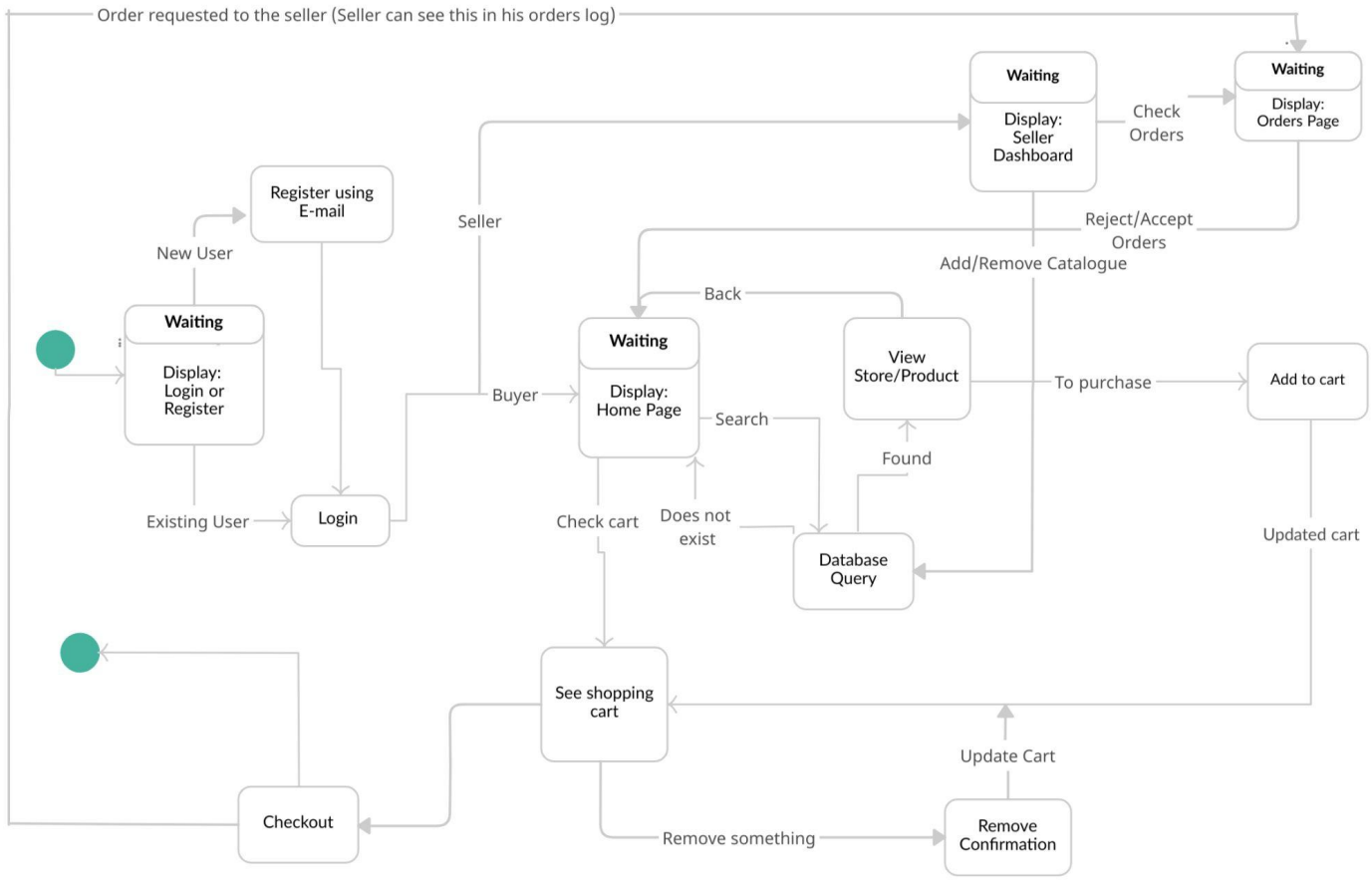



Frontend
Backend
Database
Buyer
item name
Search _item()
get Products
Products Data
Display list of Products
Store_select
Get Products
Product data
Display_Products()
Loop
Product available
Add_to_cart()
Remove_from_cart()
Checkout()
add order
Confirmation


Frontend
Backend
Database
Seller
Order History
Fetch orders
Success
Display_orders
Status Update
Accept / Reject order
order status updates
Update order data
Success
Get_cotegories
Fetch categories
Success
Display_categories
Form data
Add_category
Update and fetch category
Display Categories
Success


Frontend
Backend
Database
Buyer
Order History
Order_history()
Fetch orders
Success
Display_orders

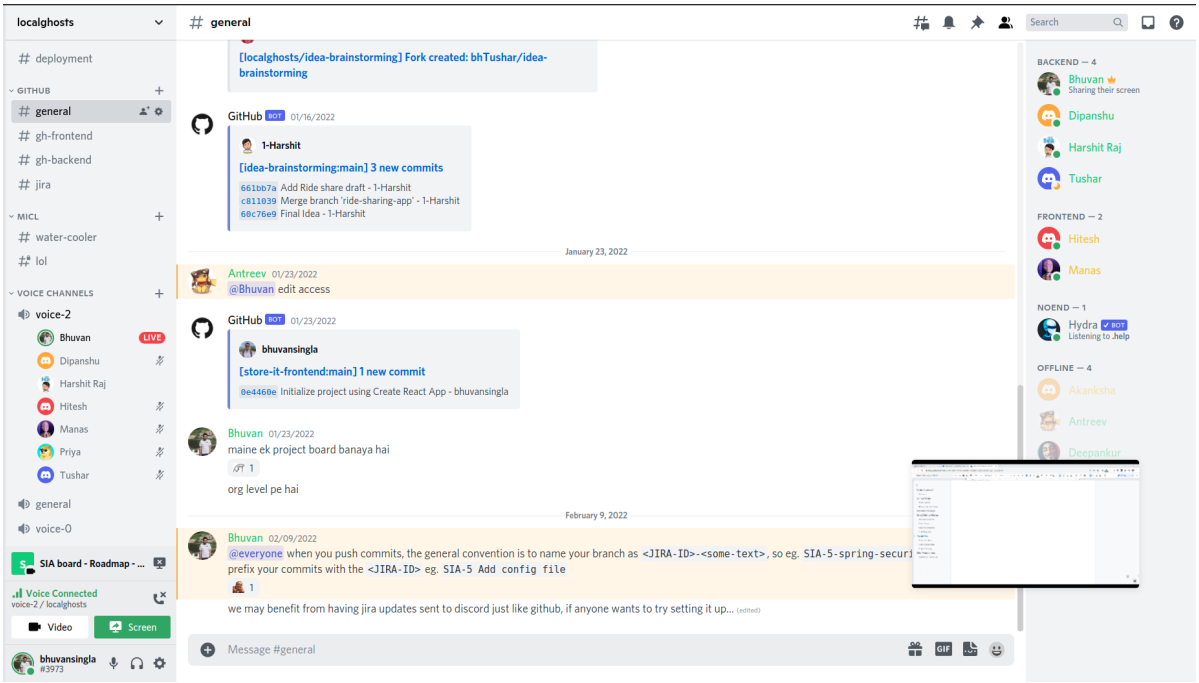
3.4 State Diagrams



4 Project Plan

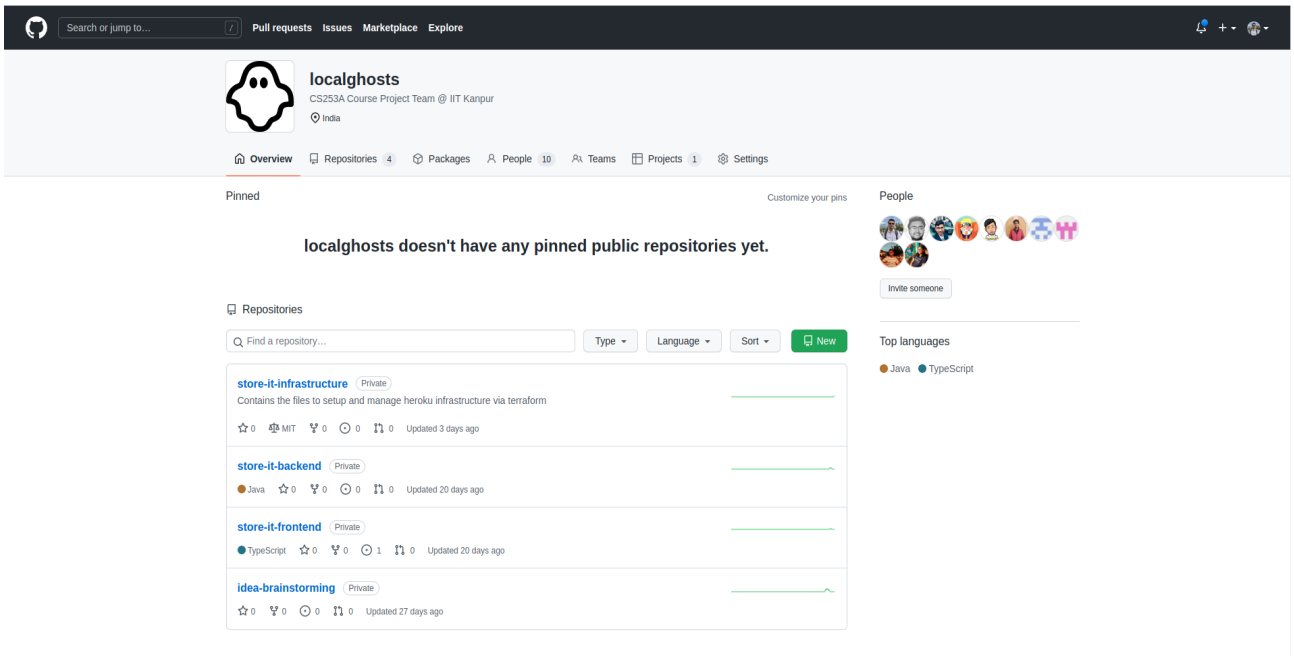
4.1 Communication

- The team communicates internally via a Discord Server both via messaging and voice calls.
- The server is divided into channels based on the scope of work so that discussions are organized based on their context.



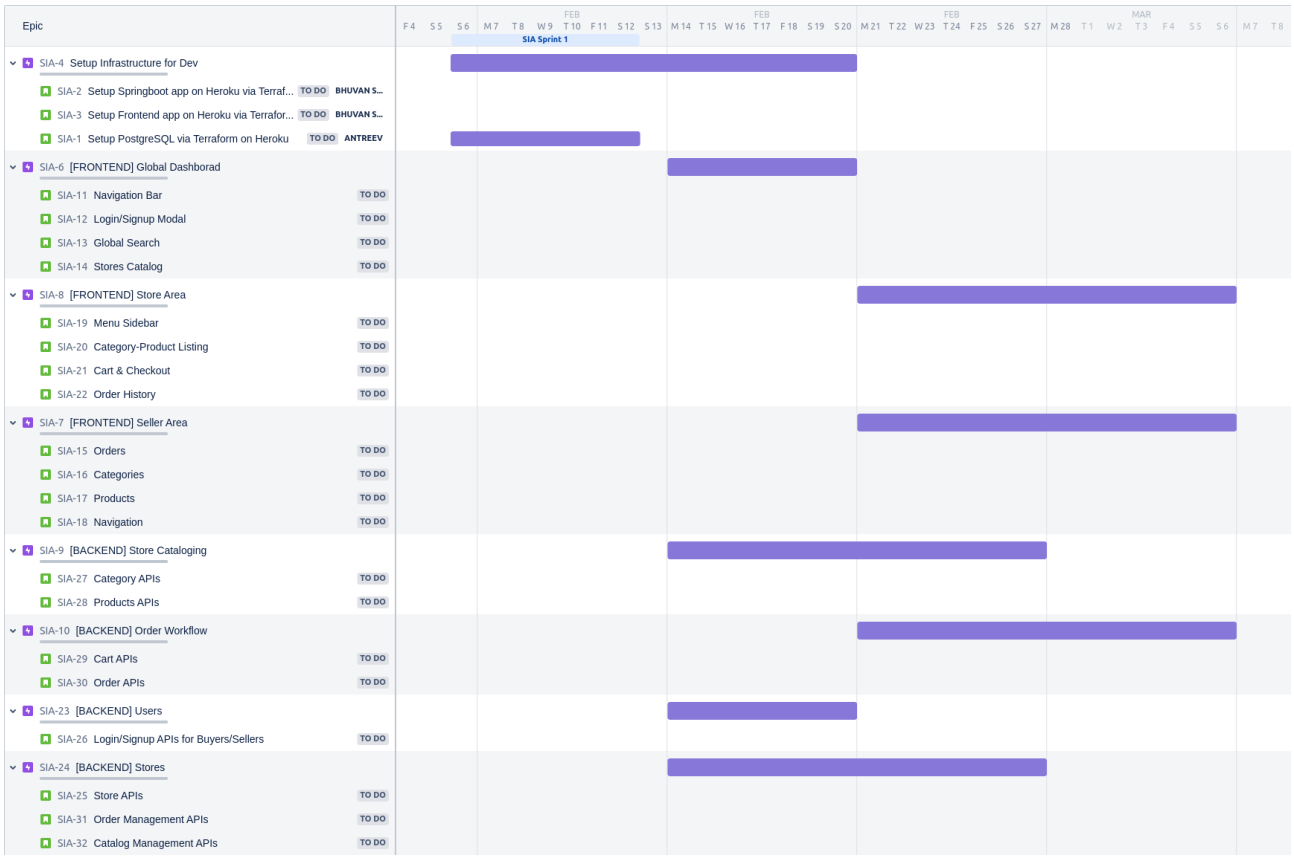
4.2 Code Collaboration

- An organization has been set up on **GitHub** for storing and collaborating on the source code of this project.
- Currently, the organization hosts **four repositories** (all private access):
  - store-it-backend
  - store-it-frontend
  - store-it-infrastructure
  - idea-brainstorming
- We will maintain two **environments (dev and prod)** for the implementation.
  - Dev: where development happens
  - Prod: the hosted version for public use (latest release)
- We aim to keep **high code coverage** of the codebase via unit tests.



4.3 Project Planning

- We have set up a JIRA board for the project planning and developing a Gantt Chart.
- Work has been divided into Epics, which is further divided into Stories.
- We will have weekly sprints where each team member will have clearly defined work.
- Every team member has internally taken the responsibility to focus majorly on either frontend or the backend.
- Our current Gantt chart looks like this:



Appendix A - Group Log

Meeting Minutes	Agenda
29th Jan 2022 2:30 pm - 5:00 pm	Started working on a draft of the Software Design document.
6th Feb 2022 1:00 pm - 4:00 pm	Divided the subsections among the members and completed most of them.
6th Feb 2022 6:00 pm - 7:00 pm	Meet with TA. Updated TA with our current progress. Discussed any issues we were facing. Planned the further steps of development.
12th Feb 2022 3:00 pm - 5:30 pm	Made changes suggested by TA. Completed all the subsections of the document.
12th Feb 2022 6:00 pm - 7:00 pm	Meet with TA. Updated TA with our progress.