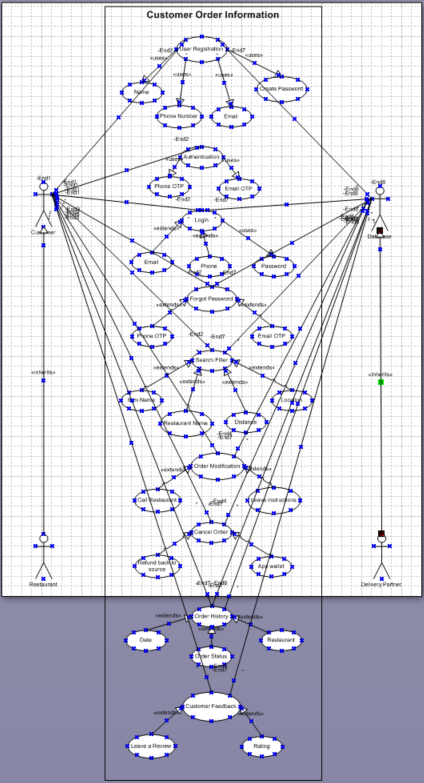
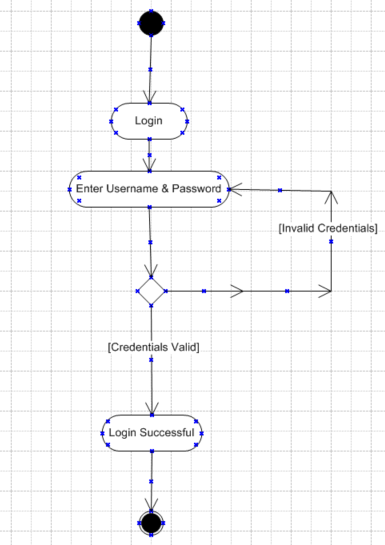
**Waterfall Project – 2**

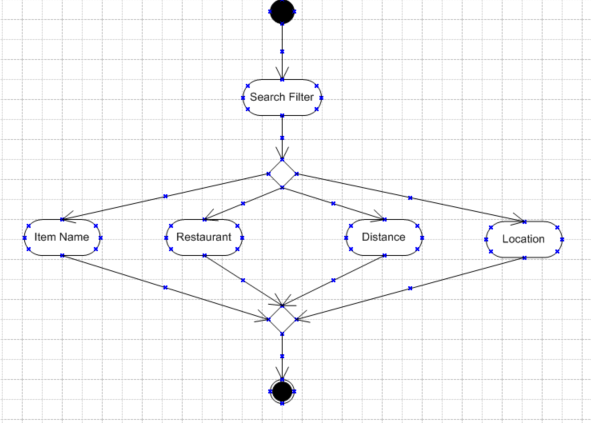
**Document 6-** Please prepare a use case diagram, activity diagram and a use case specification document.

**Solution:   
Use Case Diagram:**

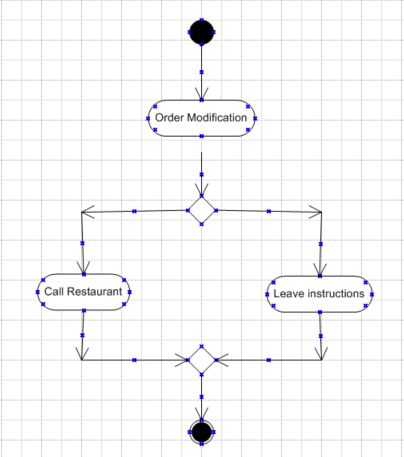
****

**Activity Diagram:**

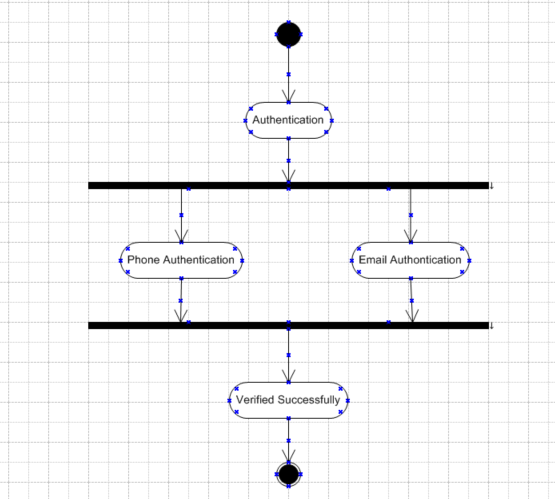




**Branch & Merge:**



**Fork & Join:**

****

**Use Case Specification:**

**Use Case 1:**

1. **Use Case Name:** User Registration
2. **Use Case Description:** New users register to order the food
3. **Actors:** Primary Actors are customers and secondary actor is system used to register
4. **Basic Flow:**

* User must enter name, phone, email and password
* Click on register so that system can validate the details
* System Validated details entered
* Registered Successfully

1. **Alternate Flow:**

* If the user provides existing details in the database, system prompts error and suggests “User Already Exists”

1. **Exceptional Flow:**

* Invalid details lead to error message from system

1. **Pre-Conditions:**

* User must have internet connection
* User must provide valid credential like email, phone etc,

1. **Post-Conditions:**

* User registered successfully.

1. **Assumptions:** User will provide all correct details
2. **Constraints:**

* User phone number must be unique
* User Email must be unique
* Password should meet security criteria like “must have 8 letters, a upper case, a lower case and a special character.”

1. **Dependencies:**

* Email gateway for verification
* SMS service for verification

1. **Inputs and Outputs**

* Inputs are Name, Phone Number, Email address and Password
* Outputs are confirmation message about registration successful

1. **Business Rules:** Depend upon business requirement for example “ User must be a citizen of India.”
2. **Miscellaneous Information:** General Data Protection Regulation (GDPR) compliance for data policy.

**Use Case 2:**

1. **Use Case Name:** Authentication
2. **Use Case Description:** Users must authenticate the phone number and the email address using One Time Password (OTP) before login.
3. **Actors: P**rimary Actor: Customer  
   Secondary Actor: System
4. **Basic Flow:**

* User must verify / authenticate the phone and email
* User Must enter phone number and verify with OTP
* User Must enter valid email id and verify with OTP
* Successful authentication redirects to the next page

1. **Alternate Flow:**

* If biometric authentication is enabled, then user can log in using fingerprint or face recognition.

1. **Exceptional Flow:**

* Incorrect details will lead to an error message.
* Multiple failed attempts will result in account lock.
* Authentication server issue which prevents login

1. **Pre-Conditions:**

* User must have valid email ID
* User must have valid Phone number

1. **Post Conditions:**

* Phone verification successful
* Email verification successful
* User Authenticated

1. **Assumptions:** User will provide all the details correctly
2. **Constraints:**

* User Phone number must be valid
* User Email must be valid
* User must have OTP services activated

1. **Dependencies:**

* Email OTP services / Mail gateway
* SMS services

1. **Inputs and Outputs:** Inputs are Phone number and Email ID  
    Outputs are OTP verified for phone and Email

**Use Case 3:**

1. **Use Case Name:** Login
2. **Use Case Description:** Allows users to access their accounts securely using the valid credentials.
3. **Actors:** Primary Actor: customer  
    Secondary Actor: System
4. **Basic Flow:**

* User Navigates to login page
* User enters email/phone and password
* System validates the credentials
* Successful login redirects to application homepage

1. **Alternate Flow:**

* If “Remember me” is enabled, the user remains logged in for future sessions
* If login is attempted from an unrecognized device, system will prompt for additional verification.

1. **Exceptional Flow:**

* Incorrect credentials will lead to error message and redirect to login again
* Multiple failures to login will temporarily lock the user account.

1. **Pre-Conditions:** User must enter the valid user’s name and password
2. **Post Conditions:** Successfully Logged in
3. **Assumptions:** User will enter the correct user’s name and password created initially
4. **Constraints:**

* User entered valid user name (Email/Phone)
* User entered password correctly which created while registration

1. **Dependencies:**

* Authentication system to verify the entered credentials
* Secure password storage and encryption

1. **Inputs and Outputs:** Inputs are user entering correct username and password

Output is user logged in successfully

**Use Case 4:**

1. **Use Case Name:** Forgot Password
2. **Use Case Description:** Allows users to reset the password if user forgot the password.
3. **Actors:** **Primary Actor** user  
    Secondary Actor System
4. **Basic Flow:**

* User click on Forgot Password on the login page
* System redirect user to verify the email/phone
* User enters the verification code sent and reset the password
* System confirms password reset successful and notifies the user.

1. **Alternate Flow:**

* If the user does not receive the verification code, they can request a resent code.
* If password does not meet eligibility criteria, system prompts and error.

1. **Exceptional Flow:**

* User can select either email verification or phone verification
* If the verification code is expired a new request must be raised

1. **Pre-Conditions:** User password either forgot or incorrect multiple times
2. **Post Condition:** Password reset successful

**Use Case 5:**

1. **Use Case Name:** Search Filter
2. **Use Case Description:** Enables users to search for food using various filter options like Item name, Restaurant, Location, Distance etc,
3. **Actors:** User as Primary Actor  
    System acting as secondary actor
4. **Basic Flow:**

* User enters a search keyword
* System retrieves and displays relevant results
* User Applies filters like Price, rating, delivery time etc,

1. **Alternate Flow:**

* If no results are found system shows “No Results Found”
* System suggests similar items based on search

1. **Exceptional Flow:**

* Slow or failed search due to server error.

1. **Pre-Conditions:**

* User is on the search page
* User enters in the search box what customer want

1. **Post Conditions:** Search results are displayed.

**Use Case 6:**

1. **Use Case Name:** Order Modification
2. **Use Case Description:** System allows users to modify the orders before preparation begins.
3. **Actors:** **Primary – User  
    Secondary – System**
4. **Basic Flow:**

* User selects the active order
* System checks modification eligibility
* User edits items, cooking instructions, quantity or delivery details
* System updates order and recalculate cost.
* User Confirms changes

1. **Alternate Flow:**

* User can contact the restaurant for any order changes or modifications.

1. **Exceptional Flow:**

* Modification window expired
* User cannot modify as system preventing changes

1. **Pre-Conditions:**

* User Must have an active order
* Order Modification should be eligible

1. **Post Conditions:**

* Order Modification done
* Recalculate the cost of the order

1. **Business Rule:**

* User is only eligible for order modification until order starts preparation.

**Use Case 7:**

1. **Use Case Name:** Cancel Order
2. **Use Case Description:** System allows users to cancel order before order preparation begins.
3. **Actors:** User acting as primary actor  
    System acting as secondary actor
4. **Basic Flow:**

* User Select the active order
* System checks if the order is eligible for cancellation or not
* User click on cancel order
* User confirms order cancellation
* System updates the order status and initiates a refund if applicable.

1. **Alternate Flow:**

* System will not show cancellation option if the order is not eligible for cancellation
* User can only cancel before order stars preparation

1. **Pre-Conditions:**

* User Must have an active order
* Order cancellation should be eligible by the system

1. **Post Conditions:**

* User confirms order cancellation
* System process refund if eligible

1. **Business Rule:**

* User or order is only eligible for cancellation before order preparation begin.

**Use Case 8:**

1. **Use Case Name:** Order History
2. **Use Case Description:** System displays users past orders for reference.
3. **Actors:** Users as primary actors  
    System acts as secondary actor
4. **Basic Flow:**

* User Navigates to the order history page
* User can search all the past orders using various filters like Date, Restaurant etc,
* System retrieves past orders
* User select order or re-order option

1. **Alternate Flow:**

* Customer or user can select the last week, month or year orders
* User can only view the order history
* If no orders in the past system shows no order history

1. **Pre-Conditions:**

* User must have ordered in the past to see order history
* User should navigate and select the range to show past orders

1. **Post Conditions:**

* User selects the date range, order or restaurant from the filter options
* System will display the order history as per user selection

**Use Case 9:**

1. **Use Case Name:** Customer Feeback
2. **Use Case Description:** System allows user to leave a review and rating for the food, restaurant and the delivery partner once order is delivered.
3. **Actors:** Primary actors are users  
    Secondary actor is system
4. **Basic Flow:**

* User selects a completed or delivered order.
* Users rates the food, restaurant and delivery partner and leave a review
* User clicks on submit
* System stores feedback and updates overall rating.

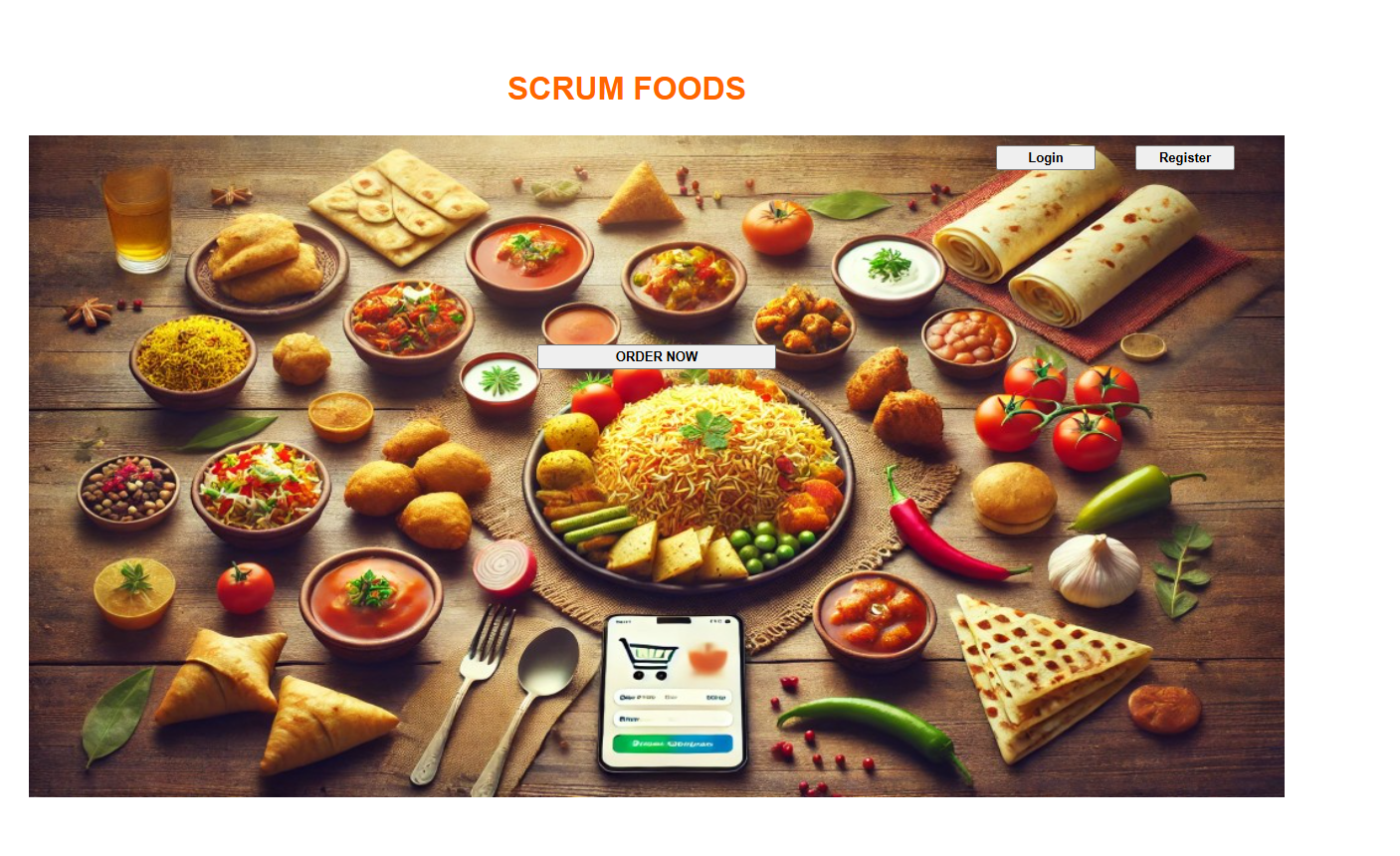
1. **Pre-Conditions:**

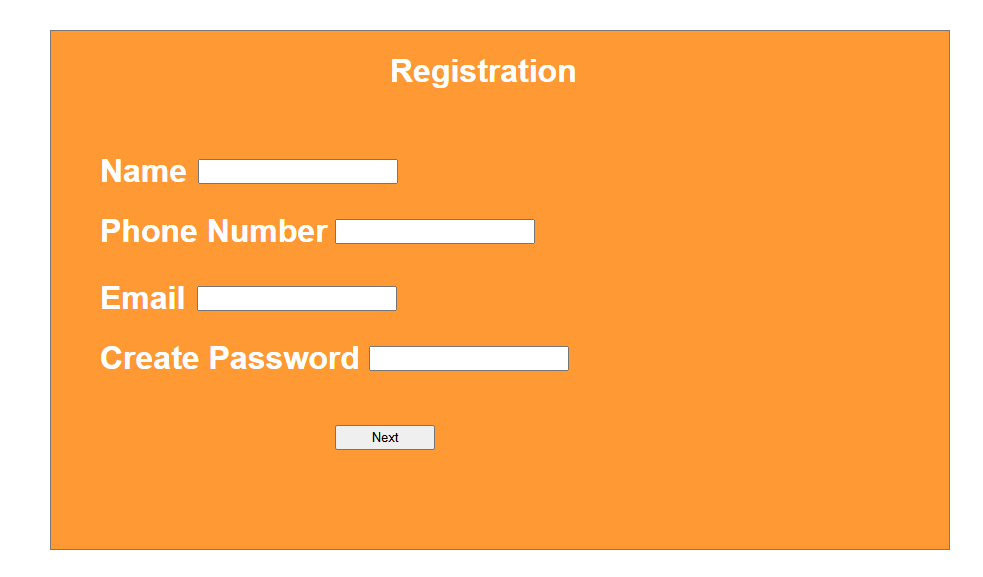
* User must have a delivered order
* User order should have submitted feedback option
* User should have rate option

1. **Post Conditions:**

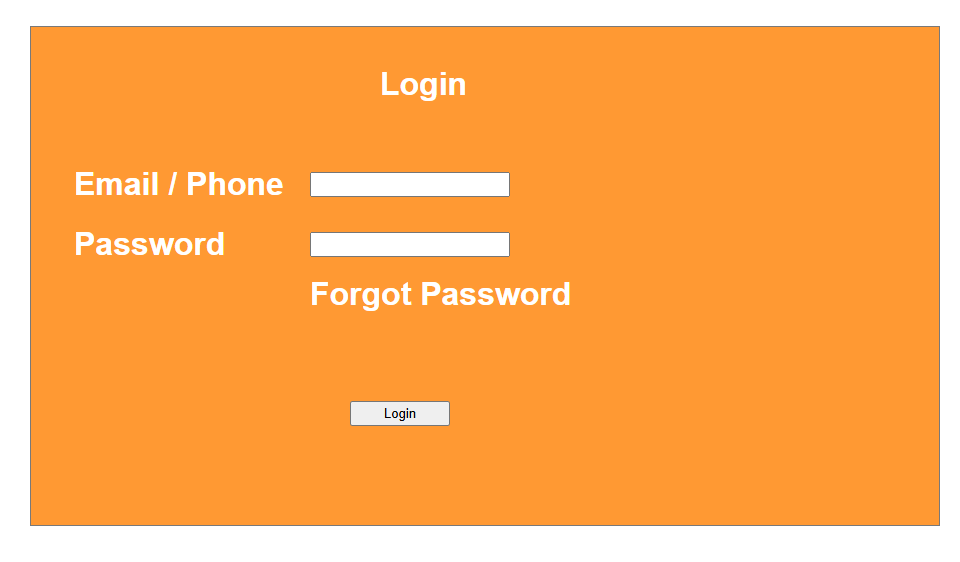
* User give rating to the restaurant, delivery partner and food items
* System prompts feedback submitted option.

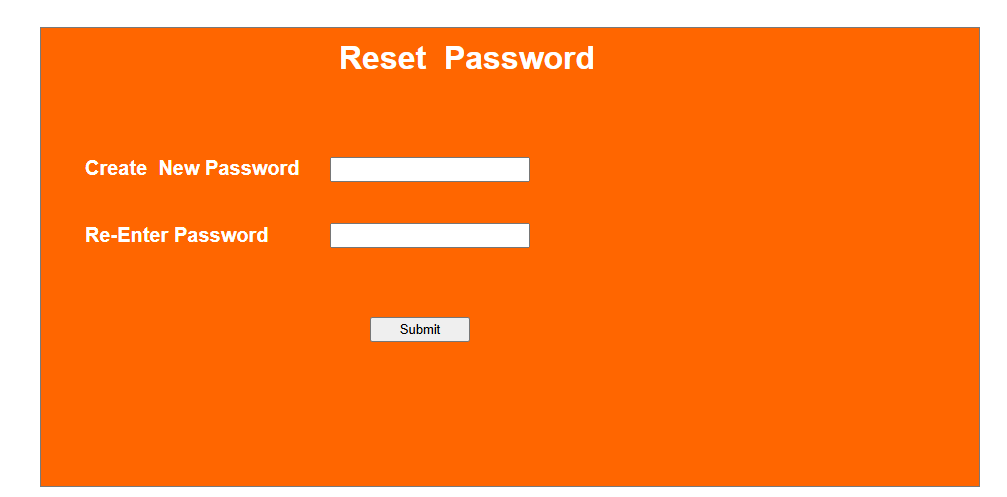
**Document 7- Screens and pages**

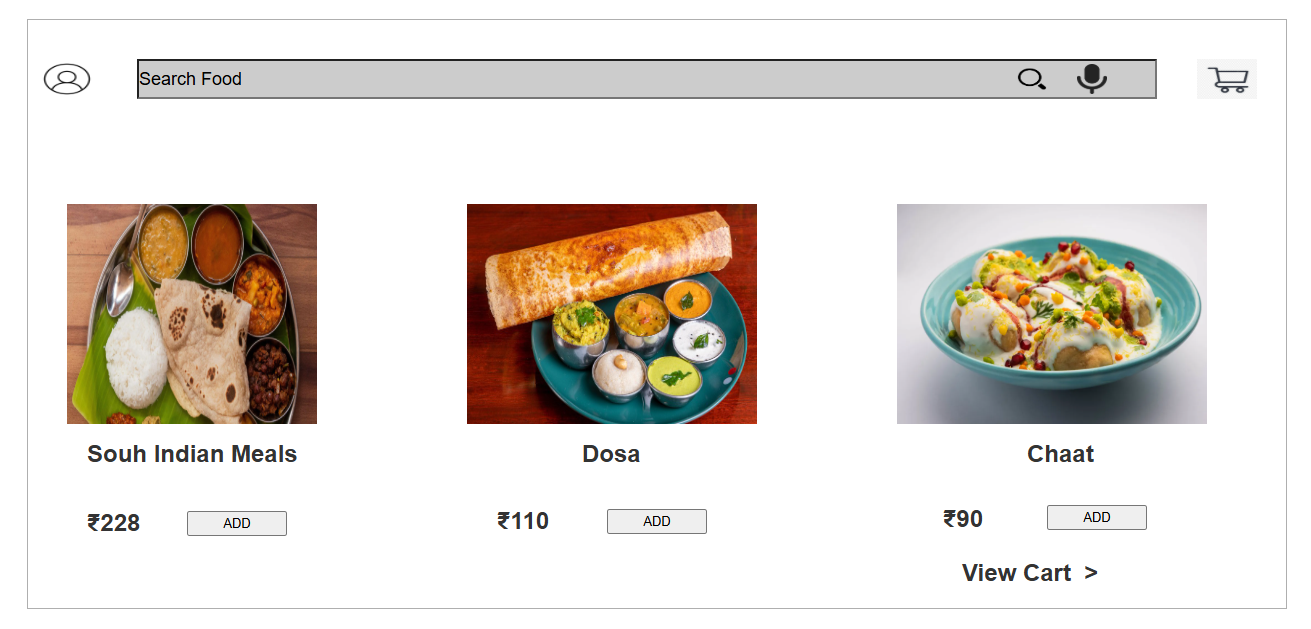
****

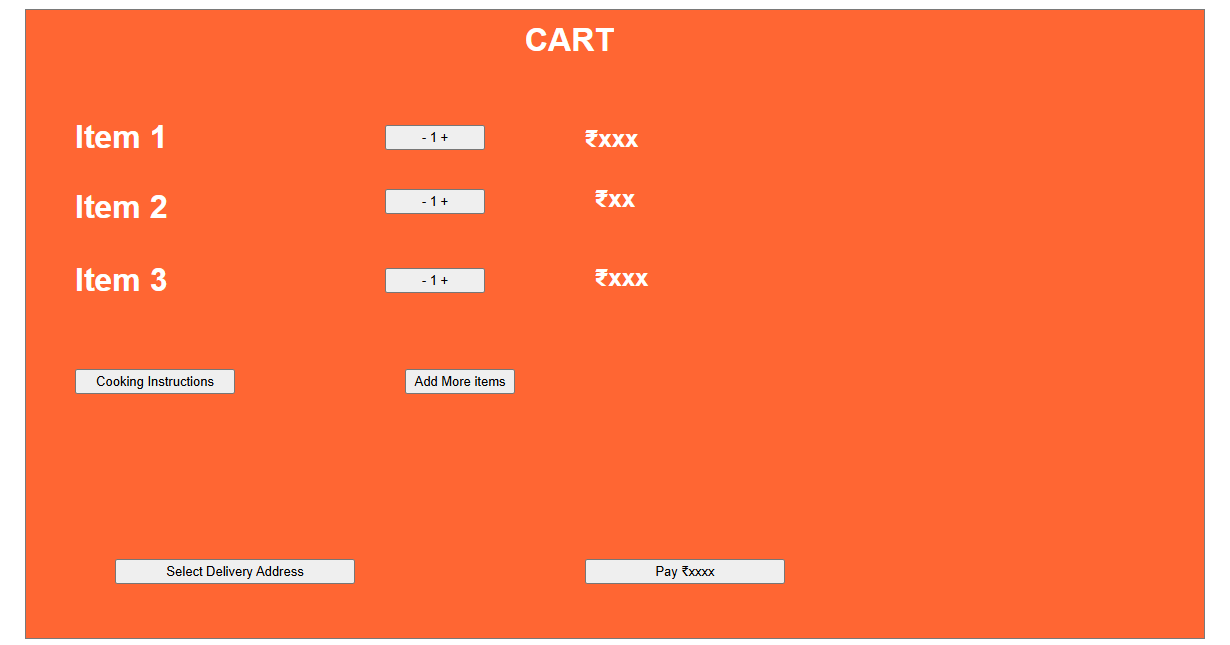
****

****

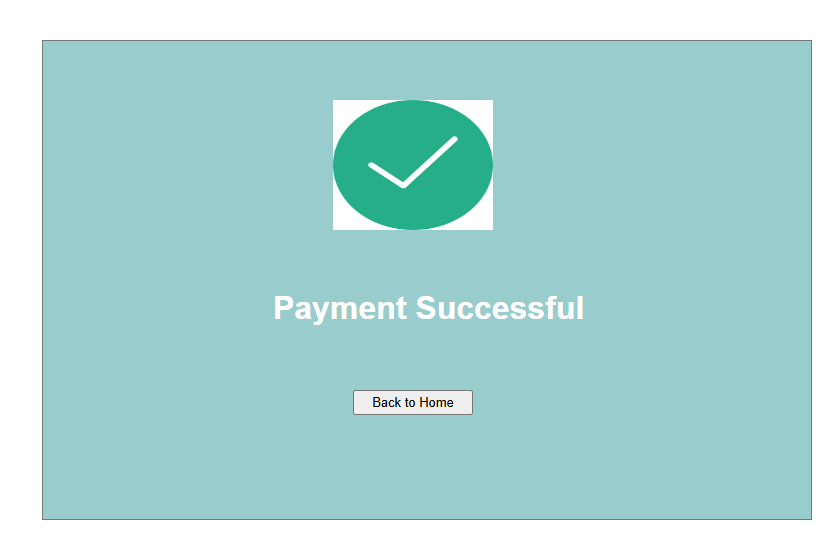
****

****

****

****

****

****

**Document 8- Tools-Visio and Axure**

* I’ve worked a with Microsoft Office Visio and Axure RP Pro, and both have been incredibly useful throughout the project. With **MS Visio**, I have created UML Diagrams (Use Case Diagram) and System flow architecture to map out business process precisely and clearly. It helps in breaking down complex workflows and making them easier for stakeholders and developers to understand without human involvement.
* **Axure RP Pro,** on the other hand has been very useful for wireframing and creating a sample prototype or design or the User Interface (UI) for my project. I used to design the sample prototype , ensuring the user experience is smooth and intuitive. The ability to create dynamic interactions in Axure RP makes it feel like a real ap, which is great for testing and getting feedback.
* Both tools have been a huge help in bringing structure and clarity to my project.

**Document 9- BA experience**

**My experience as BA in following phases:**

1. **Requirements Gathering:**

* Used the MOSCOW technique to prioritize the requirements efficiently.
* Managed stakeholders’ unavailability by identifying alternative contacts to ensure continuity.
* Validated gathered requirements using the FURPS(Functionality, Usability, Reliability, Performance and Supportability) model to maintain quality standards.
* Identified and removed redundant or conflicting requirements to avoid project scope issues.

1. **Requirements Analysis:**

* Developed UML Diagrams (Use Case, Activity) to represent requirements visually.
* Created Activity Diagrams to outline process workflow.
* Collaborated with teams to review diagrams and adjusted them based on feedback.
* Documented finalized requirements in the BRS (Business Requirement Specification) and SRS (Software Requirements Specification).

1. **Design:**

* Derived test cases from use case diagrams to ensure comprehensive validation.
* Engaged with the client to finalize design and solution documents.
* Created positive and negative test cases to cover all possible scenarios.
* Maintained the Requirement Traceability Matrix (RTM) to track requirements throughout the project lifecycle.

1. **Development:**

* Conducted JAD (Joint Application Development) sessions to bridge communication between business team and technical team.
* Addressed development team queries to ensure clarity in implementation.
* Provided visual references like diagrams and process flows to aid development.
* Ensured smooth coordination between clients and developers by conducting regular meetings and recording sessions for absentees.

1. **Testing:**

* Created and reviewed test cases to validate system functionality.
* Conducted high-level testing before formal QA testing.
* Ensured test data was available for accurate validation.
* Updated RTM to maintain requirements traceability throughput the testing phase.
* Organized internal testing before UAT (User Acceptance Testing ) to detect and fix issues early.
* Worked with QA team on regression testing to ensure the system stability.
* Prepared UAT plan, defining scenarios, acceptance criteria and user roles.

**User Acceptance Testing (UAT):**

* Coordinated UAT execution with the end users.
* Collected and documented feedback on system performance.
* Assisted in resolving issues reported during UAT in collaboration with developers.
* Secured UAT sign-off from the client before proceeding with the deployment.

1. **Deployment:**

* Shared the RTM with the client as a part of project closure.
* Created and distributed end-user manuals for seamless adoption.
* Organized training sessions to ensure a smooth transition for users.
* Ensured all key stakeholders participated in training and understood the system.
* Provided post-deployment support, addressing immediate concerns and enhancements.