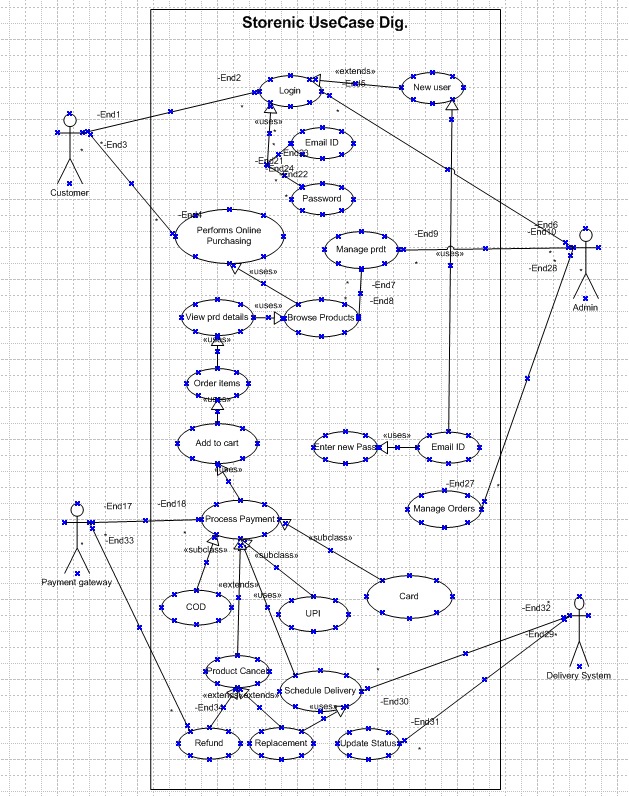
**Document 6- Please prepare a use case diagram, activity diagram and a use case specification document. Use case Diagram -**



**Activity Diagram** -



## **Use Case Specifications -**

## **1. User Registration**

|  |  |
| --- | --- |
| Field | Details |
| Use Case Name | User Registration |
| Use Case Description | Allows new users to create an account on Storenic. |
| Actors | **Primary:** Customer  **Secondary:** Admin (for account approvals) |
| Basic Flow | 1. User clicks on "Sign Up".  2. User enters required details (name, email, password, etc.).  3. System validates inputs and creates an account.  4. System sends an email verification link.  5. User verifies email and account is activated. |
| Alternate Flow | If the user does not verify their email, the account remains inactive. |
| Exceptional Flow | If the email is already registered, an error message is displayed. |
| Pre-Conditions | User must have a valid email and internet access. |
| Post-Conditions | Account is created, and the user can log in. |
| Assumptions | Users provide correct information. |
| Constraints | Password must meet security guidelines. |
| Dependencies | Email verification service. |
| Inputs and Outputs | **Input:** User details  **Output:** Account confirmation email |
| Business Rules | Duplicate emails are not allowed. |
| Miscellaneous Information | None |

## **2. User Login**

|  |  |
| --- | --- |
| Field | Details |
| Use Case Name | User Login |
| Use Case Description | Allows registered users to log in. |
| Actors | **Primary:** Customer  **Secondary:** Admin (for account recovery) |
| Basic Flow | 1. User enters email and password.  2. System validates credentials.  3. User is redirected to the home page. |
| Alternate Flow | If the user forgets their password, they can reset it. |
| Exceptional Flow | If credentials are incorrect, an error message is shown. |
| Pre-Conditions | User must be registered. |
| Post-Conditions | User is logged in successfully. |
| Assumptions | Users remember their credentials. |
| Constraints | Password retry limit: 3 attempts. |
| Dependencies | Authentication system. |
| Inputs and Outputs | **Input:** Login credentials  **Output:** User authentication status |
| Business Rules | Locked out after 3 failed attempts. |
| Miscellaneous Information | Two-factor authentication may be required. |

## **3. Product Browsing and Search**

|  |  |
| --- | --- |
| Field | Details |
| Use Case Name | Product Browsing and Search |
| Use Case Description | Users browse categories or search for products. |
| Actors | **Primary:** Customer  **Secondary:** Admin (for product management) |
| Basic Flow | 1. User searches for a product or navigates through categories.  2. System retrieves and displays matching products. |
| Alternate Flow | If no results are found, recommendations are displayed. |
| Exceptional Flow | If database retrieval fails, an error is shown. |
| Pre-Conditions | Product database must be available. |
| Post-Conditions | User finds desired products. |
| Assumptions | Users enter relevant search terms. |
| Constraints | Search results limited to 50 items per page. |
| Dependencies | Product catalog database. |
| Inputs and Outputs | **Input:** Search query  **Output:** Product list |
| Business Rules | Search must be case-insensitive. |
| Miscellaneous Information | Filters and sorting available. |

## **4. Add to Cart**

|  |  |
| --- | --- |
| Field | Details |
| Use Case Name | Add to Cart |
| Use Case Description | Users add items to their shopping cart. |
| Actors | **Primary:** Customer |
| Basic Flow | 1. User selects a product and clicks "Add to Cart".  2. System adds the item to the cart.  3. User can update or remove items. |
| Alternate Flow | If the item is out of stock, a message is displayed. |
| Exceptional Flow | If session expires, cart is cleared. |
| Pre-Conditions | User must be logged in. |
| Post-Conditions | Cart is updated. |
| Assumptions | User selects valid quantities. |
| Constraints | Cart limit: 20 items. |
| Dependencies | Inventory system. |
| Inputs and Outputs | **Input:** Product selection  **Output:** Updated cart |
| Business Rules | Cannot add more than available stock. |
| Miscellaneous Information | Discounts apply at checkout. |

## **5. Checkout and Payment**

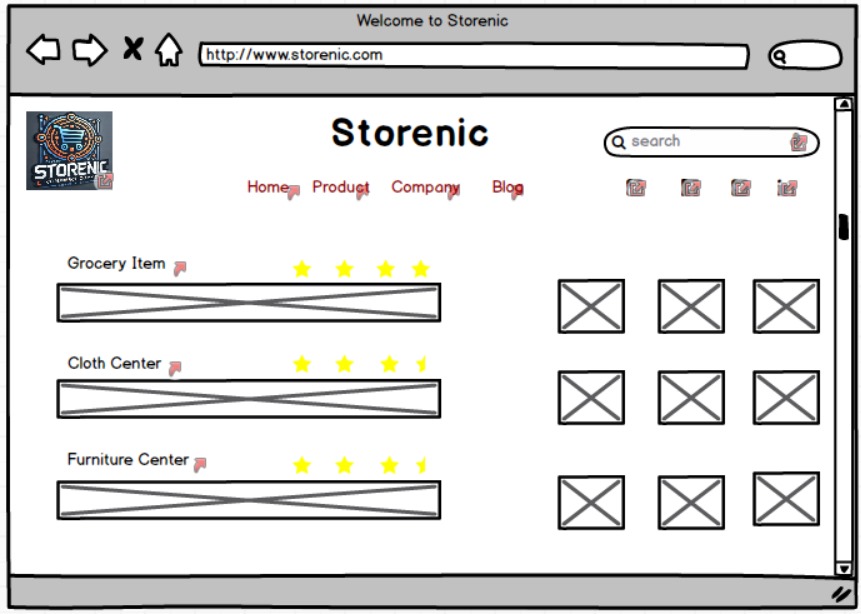
|  |  |
| --- | --- |
| Field | Details |
| Use Case Name | Checkout and Payment |
| Use Case Description | Users complete purchases using online payment methods. |
| Actors | **Primary:** Customer  **Secondary:** Payment Gateway |
| Basic Flow | 1. User proceeds to checkout.  2. Enters shipping details.  3. Selects payment method.  4. Completes payment. |
| Alternate Flow | If the payment fails, user can retry. |
| Exceptional Flow | If internet disconnects, transaction is canceled. |
| Pre-Conditions | User must have valid payment details. |
| Post-Conditions | Order is confirmed. |
| Assumptions | Payment gateway is working. |
| Constraints | Payment processing time: < 10 seconds. |
| Dependencies | Payment service provider. |
| Inputs and Outputs | **Input:** Payment details  **Output:** Order confirmation |
| Business Rules | Secure payment required. |
| Miscellaneous Information | Refunds take 7 days. |

## **6. Manage Orders (Admin)**

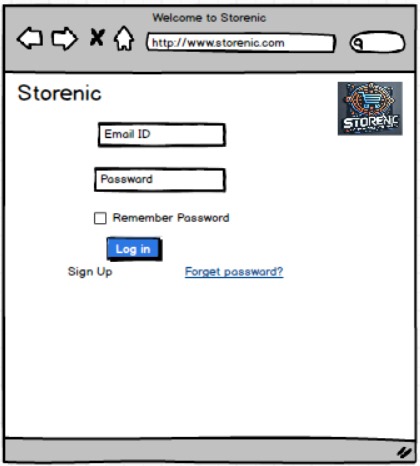
|  |  |
| --- | --- |
| Field | Details |
| Use Case Name | Manage Orders (Admin) |
| Use Case Description | Allows admins to manage and oversee all orders, including order status updates, cancellations, and issue resolutions. |
| Actors | **Primary:** Admin  **Secondary:** Customer |
| Basic Flow | 1. Admin logs into the system.  2. Admin navigates to the "Manage Orders" section.  3. Admin views a list of all orders.  4. Admin selects an order to update status (e.g., "Processing," "Shipped," "Delivered").  5. If needed, admin can cancel orders or provide refunds.  6. Admin confirms the changes, and the system updates order status. |
| Alternate Flow | If a customer requests cancellation, the admin reviews and approves or denies it. |
| Exceptional Flow | If the order ID is invalid, an error message is displayed.  If the system fails to update order status, an error log is generated. |
| Pre-Conditions | Admin must have the necessary access permissions.  Orders must be available in the system. |
| Post-Conditions | Order status is updated, and customers receive notifications. |
| Assumptions | Admin has accurate order details.  Customers may request order changes. |
| Constraints | Admin cannot modify an order that has already been delivered. |
| Dependencies | Order management system, notification system. |
| Inputs and Outputs | **Input:** Order ID, status update, cancellation request  **Output:** Updated order status, notification to customer |
| Business Rules | Only admins with proper permissions can modify orders.  Order cancellations are not allowed after dispatch. |
| Miscellaneous Information | System logs all admin activities for audit purposes. |

**Document 7- Screens and pages**

1. **Home page of Storenic**

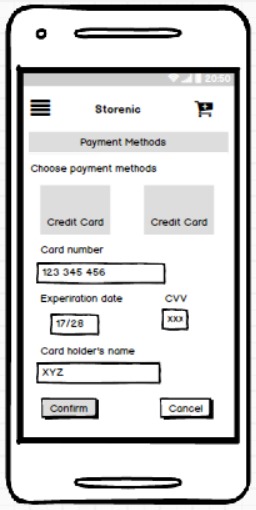


1. **Login Page**





1. **Payment Page**



1. Add to Cart Page



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

I used Visio and Axure for the project. Visio helped me draw diagrams and flowcharts. It was easy to put shapes and lines together to show how things work. Axure let me make prototypes of the app. I created screens and buttons that looked real. Both tools were simple to use and made my work easier.

When I am working on a project, tools like Visio and Axure help me a lot. They make it easy to share ideas and plans with others.

**Document 9- BA experience**

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

1. Requirement Gathering

1. Conducted stakeholder meetings using MoSCoW prioritization and FURPS validation.  
2. Created the Business Requirement Document (BRD) and gathered inputs via prototyping.  
3. Managed stakeholder unavailability by sourcing alternative contacts.

Challenges:

Handled requirement conflicts through impact analysis.

2. Requirement Analysis

1. Created Use Case Diagrams, Activity Diagrams, and RTM to track requirements.  
2. Finalized functional/non-functional requirements in the Functional Specification Document (FSD).  
3. Conducted peer reviews to align with stakeholders.

Challenges:

Managed mid-phase requirement changes via structured Change Requests (CRF).

3. Design

1. Collaborated with UI/UX designers using Axure to create wireframes & mockups.  
2. Ensured easy navigation, accessibility, and responsive UI in the design phase.  
3. Created test case scenarios from Use Case Diagrams.

Challenges:

Balanced UI expectations vs. technical feasibility through discussions with designers & developers.

4. Development

1. Conducted JAD sessions to ensure alignment between business and tech teams.  
2. Verified business rules implementation and assisted in payment gateway integration.  
3. Regularly reviewed progress and provided feedback.

Challenges:

Addressed misinterpretation of requirements through daily standups.

5. Testing

1. Created test cases from use cases and ensured full functional coverage.  
2. Assisted QA team with testing and conducted UAT with stakeholders.  
3. Verified error handling & security measures before deployment.

Challenges:

Managed post-UAT defect fixes with a prioritized bug resolution approach.

6. Deployment

1. Coordinated training sessions & user manuals for smooth onboarding.  
2. Ensured compliance with GDPR & PCI DSS before launch.  
3. Planned post-deployment monitoring and resolved user adaptation challenges.

Challenges:

Addressed post-launch bugs via a structured tracking system.