**Live Project – Waterfall Deliverables -Part 2/2**

**Document 6: Use Case Diagram, Activity Diagram, and Use Case Specification Document.**

 **Use Case Diagram:**

****

**Activity Diagram 1:**

**Login**

****

**Activity Diagram 2:**

**Checking Stock & Managing Inventory**

****

**Activity Diagram 3:**

**Receiving Multiple Orders from supplier**

**Activity Diagram 4:**

**Sales Processing**

****

**Use Cases:**

**1)**

|  |  |
| --- | --- |
| Use Case ID | UC001 |
| Use Case Name | Retail Store Owner Login |
| Created By | Vedant  | Last Updated By | April 1st, 2025 |
| Data Created | February 31st, 2025 | Last Revision Date | April 15th, 2025 |
| Actor | Retail Store Owner, Supplier, System Administrator |
| Description | The store owner logs into the Retail Store Management System (RSMS) using their credentials to access the dashboard and manage store operations. |
| Pre condition | 1. The user must have the internet connectivity.
2. The owner must have an active user account
 |
| Post condition | 1. The user successfully logged in and owner gains access to the dashboard.
 |
| Normal Flow | 1. Owner opens the RSMS login page.
2. Owner enters credentials (username & password).
3. System validates credentials.
4. If valid, system grants access to the dashboard.
 |
| Alternative Flow | 1. If the owner enters incorrect credentials, the system prompts for re entry or password reset.
2. The system will then provide an option for “Forgot user ID/Password”.
3. The user is asked to re enter their credentials.
 |
| Exceptions | 1. If the user forgets their password.
2. If the internet connectivity is not working.
3. If the application is facing some technical issue.
4. If multiple incorrect attempts, the system locks the account.
 |
| Frequency of Use | High |
| Assumptions | 1. It is assumed that the user has registered on the application.
2. It is assumed that the user knows the credentials correctly.
3. Internet connection is available.
 |
| Constraints | Must use registered credentials. |
| Dependencies | Database access for authentication. |
| Input | Username, password |
| Output | Dashboard access |
| Business Rules | Only registered users can log in |

**2)**

|  |  |
| --- | --- |
| Use Case ID | UC002 |
| Use Case Name | Inventory Management |
| Created By | Vedant  | Last Updated By | April 1st, 2025 |
| Data Created | February 31st, 2025 | Last Revision Date | April 15th, 2025 |
| Actor | Retail Store Owner, Supplier |
| Description | The store owner or manager updates product stock in the system. |
| Pre condition | 1. The user must be logged in to the application.
2. Active products must exist in the database.
3. The user has already reached to the desired product through the search option.
 |
| Post condition | 1. Inventory is updated.
 |
| Normal Flow | 1. Owner accesses inventory module.
2. System displays stock levels.
3. Owner updates stock details or marks low stock items.
4. System saves changes.
 |
| Alternative Flow | 1. In case the user wants to change the product after adding a product to the cart.
2. The user wants to change the quantity selected for the product.
3. If stock is below the threshold, the system suggests reorderin.
 |
| Exceptions | 1. If the product is out of stock.
2. If the application is facing some technical issue.
3. If the supplier is unavailable, the order is placed on hold.
 |
| Frequency of Use | High |
| Assumptions | 1. The user has knowledge to add the product to the cart.
2. The product details are up to date and the application is showing the “in stock” and “out of stock” products properly.
3. Supplier can fulfill orders timely.
 |
| Constraints | Stock cannot be negative. |
| Dependencies | Supplier database integration. |
| Input | Stock quantity updates |
| Output | Updated inventory records |
| Business Rules | Stock levels should be checked daily. |

**3)**

|  |  |
| --- | --- |
| Use Case ID | UC003 |
| Use Case Name | Sales Processing |
| Created By | Vedant  | Last Updated By | April 1st, 2025 |
| Data Created | February 31st, 2025 | Last Revision Date | April 15th, 2025 |
| Actor | Cashier |
| Description | A cashier scans items and processes a sale transaction. |
| Pre condition | 1. The cashier must be logged in to the application.
2. Products must be registered in the system.
 |
| Post condition | 1. Sale is recorded.
 |
| Normal Flow | 1. Cashier scans product barcodes.
2. System fetches product details and price.
3. System calculates the total bill.
4. Customer chooses a payment method.
5. Payment is processed, and a receipt is generated.
 |
| Alternative Flow | 1. If the barcode is unreadable, the cashier enters the item manually.
 |
| Exceptions | 1. If the internet connectivity is not working.
2. If payment fails, the transaction is canceled.
 |
| Frequency of Use | High |
| Assumptions | 1. Cashiers are trained to use the system
 |
| Constraints | Only registered products can be scanned. |
| Dependencies | Barcode scanner, payment gateway. |
| Input | Scanned items. |
| Output | Invoice, stock deduction. |
| Business Rules | Sales must be recorded for tax purposes. |

**4)**

|  |  |
| --- | --- |
| Use Case ID | UC004 |
| Use Case Name | Payment Processing |
| Created By | Vedant  | Last Updated By | April 1st, 2025 |
| Data Created | February 31st, 2025 | Last Revision Date | April 15th, 2025 |
| Actor | Customer |
| Description | This use case describes how the system processes payments from customers. |
| Pre condition | 1. The user must be logged in to the application.
2. The user must have added some products to the cart.
3. The application should have secure option to make the payment.
 |
| Post condition | 1. The farmer has successfully made the payment for the products.
 |
| Normal Flow | 1. The customer selects and get the products.
2. The customer comes to billing section.
3. Cashier scan products barcode and generate bill.
4. Customer selects payment method.
5. The application shows different methods to make the payment as :
* UPI
* Credit Card
* Debit Card
* Cash
1. System requests payment details.
2. Payment gateway processes the transaction.
3. Payment confirmation is sent to the system.
4. System generates a receipt
 |
| Alternative Flow | 1. If using UPI, a QR code is generated for payment.
2. In case store gives to discount available on the product then discounted bill generates.
 |
| Exceptions | 1. If payment fails, the customer retries or uses another method.
2. In case the payment is interrupted due to any bank or technical issue.
 |
| Frequency of Use | High |
| Assumptions | 1. Payment gateway is functional
2. The customer would have sufficient amount in their bank account to make the required payment or cash availability.
 |
| Constraints | Must comply with security protocols. |
| Dependencies | Bank servers and payment providers. |
| Input | Card details, UPI ID |
| Output | Payment confirmation, receipt |
| Business Rules | Payments must be secure. |

**5)**

|  |  |
| --- | --- |
| Use Case ID | UC005 |
| Use Case Name | Sales Report Generation |
| Created By | Vedant  | Last Updated By | November 1st, 2025 |
| Data Created | February 31st, 2025 | Last Revision Date | November 15th, 2025 |
| Actor | Store Owner |
| Description | The system generates sales reports for analysis. |
| Pre-condition | 1. Sales transactions must be recorded.
 |
| Post condition | 1. Report is available for analysis.
 |
| Normal Flow | 1. Owner selects report type (daily, weekly, monthly).
2. System fetches sales data.
3. System generates a report.
4. Owner downloads or prints the report.
 |
| Alternative Flow | 1. • If the owner needs customized data, filters can be applied.
 |
| Exceptions | 1. If payment fails, the customer retries or uses another method.
 |
| Frequency of Use | High |
| Assumptions | 1. Sales data is accurate.
 |
| Constraints | 1. Reports must be generated in a specified format.
 |
| Dependencies | 1. Dependencies: Sales database
 |
| Input | Report filters |
| Output | Sales report |
| Business Rules | Reports must align with accounting standards. |

**Mockup 1**

**Login Page**

****

**Mockup 2**

**Inventory overview dashboard**

****

**Mockup 3**

**Stock Availability Page**

****

**Mockup 4**

**Billing And Payment**

****

**Document 8: Tools Visio and Axure**

**Balsamiq** – It is a popular lightweight wireframing tool known for its simplicity and speed, enabling designers to quickly sketch out user interfaces and iterate on design ideas. Its hand drawn style helps emphasize structure and layout over fine details, encouraging team collaboration and rapid ideation. Balsamiq's drag and drop interface, pre built UI components, and real time collaboration features make it easy to create and modify wireframes, while its low fidelity design keeps the focus on functionality. However, its basic interactivity and limited features may not suffice for complex projects requiring high fidelity mock ups.

**Axure RP** – It is a robust tool designed for professionals needing to create detailed and interactive prototypes. It excels in complex projects with features like conditional logic, dynamic content, and advanced interactions. Axure RP's comprehensive environment supports wireframes, flowcharts, mock ups, and specifications, offering extensive widget libraries, adaptive views, and HTML export options. While it provides precision and advanced functionalities, it has a steeper learning curve and higher cost, making it more suitable for professional designers and large-scale projects.

**Document 9: BA Experience**

**Business Analyst Experience in Retail Store Management System (RSMS)**

**1. Requirement Gathering:**

Key Activities:

* Used the MoSCoW technique to categorize requirements into Must Have, Should Have, Could Have, and Won’t Have features.
* Ensured effective communication with the client, even when direct availability was limited. I identified alternative points of contact to avoid project delays.
* Validated requirements using the FURPS technique (Functionality, Usability, Reliability, Performance, Supportability).
* Eliminated duplicate requirements and ensured all business needs were clearly defined.
* Created prototypes and wireframes to give a visual representation of the system.

**2. Requirement Analysis:**

Key Activities:

* Created UML diagrams (Use Case, Activity, and Sequence Diagrams) to depict workflows.
* Developed activity diagrams for major processes like inventory restocking, order checkout, and customer registration.
* Communicated diagrams with the development team and ensured alignment with client expectations.
* Created BRS (Business Requirement Specification) and SRS (Software Requirement Specification) to document all functional and non functional requirements.

**3. Design:**

Key Activities:

* Collaborated with the UI/UX team to finalize screen layouts for the RSMS dashboard.
* Created test scenarios based on Use Case Diagrams for customer transactions, stock management, and order fulfilment.
* Ensured that all test cases covered positive and negative test scenarios.
* Maintained an updated RTM (Requirement Traceability Matrix) to track feature implementation.

**4. Development:**

Key Activities:

* Organized JAD (Joint Application Development) sessions to keep stakeholders engaged.
* Ensured that developers referenced UML diagrams for coding the system components.
* Clarified technical doubts from the development team regarding order processing workflows and stock level automation.
* Ensured compliance with security and performance benchmarks for the RSMS.
* Conducted regular stand up meetings to resolve blockers.

**5. Testing:**

Key Activities:

* Designed test cases covering key functionalities such as customer purchase, supplier order management, and sales reporting.
* Conducted high level testing and documented all defects.
* Requested real world test data to validate system accuracy.
* Facilitated UAT (User Acceptance Testing) and ensured client sign off.

**6. Deployment:**

Key Activities:

* Ensured RTM compliance before final deployment.
* Conducted end user training for store managers and employees.
* Created user manuals and FAQs to help with system adoption.
* Conducted a final review meeting to address post deployment concerns.

Conclusion:

Throughout my Business Analyst journey in RSMS, I learned the importance of structured documentation, effective communication, and proactive problem solving. Each phase had its own challenges, but by implementing agile thinking, prioritization frameworks, and continuous collaboration, I successfully delivered a scalable and efficient Retail Store Management System.

**Adding some more experience points based on my Experience for this project**

**1. Requirement Gathering**

**Techniques:**

Interviews with Retailer Supplier, and management

 Surveys to collect feedback on current processes

 Review of existing documentation and systems

**Activities:**

* Conducting one on one interviews with stakeholders to understand pain points and requirements
* Distributing surveys to gather quantitative data on marks entry processes
* Analyzing existing documents such as marks entry system and procedures manuals
* Shadowing management/higher authority to observe current workflows and identify areas for improvement

**Challenges:**

* Gathering comprehensive requirements from diverse stakeholders
* Ensuring alignment between business needs and technical capabilities
* Managing expectations of stakeholders with varying priorities
* Dealing with resistance to change from online process to existing processes

**Steps to Overcome Challenges:**

* + Establishing a clear communication plan to engage stakeholders throughout the requirement gathering process
* Prioritizing requirements based on business impact and feasibility
	+ Facilitating workshops to resolve conflicting requirements and build consensus
	+ Providing training and support to help stakeholders adapt to new processes

**2. Requirement Analysis:**

**Techniques:**

* + Use Case Analysis to identify user interactions with the system
	+ Business Process Modeling to visualize Retail Store workflows
	+ Data Modeling to define data entities and relationships
	+ Requirement Prioritization to determine critical features

**Activities:**

* + Creating use cases to capture system interactions for different user roles
	+ Modeling current inventory management processes to identify inefficiencies andbottlenecks
	+ Prioritizing requirements based on business value and dependencies

**Challenges:**

* + Managing complexity in inventory processes and data
	+ Balancing conflicting requirements from different stakeholders
	+ Ensuring scalability and flexibility of the system architecture
	+ Addressing regulatory compliance and data privacy requirements

**Steps to Overcome Challenges:**

* + Collaborating closely with Retail industry subject matter experts to understand detailed requirements
	+ Facilitating workshops and discussions to resolve conflicts and reach consensus
	+ Designing a modular and extensible system architecture to accommodate future changes
	+ Conducting thorough analysis of regulatory requirements and incorporating them into the system design

**3. Design:**

**Techniques:**

* System Design to define system components and interactions
* User Interface Design to create intuitive interfaces for users
* Data Design to design databases and data storage mechanisms
* Architecture Design to define the overall system structure

**Activities:**

* + Designing the system architecture with modules for employee management functions
	+ Creating wireframes and mockups for user interfaces
	+ Defining database schemas and data storage mechanisms
	+ Documenting design decisions and rationale for future reference

**Challenges:**

* + Balancing usability with functionality in the user interface design
	+ Ensuring security and privacy of payment system data
	+ Managing design changes and maintaining consistency

**Steps to Overcome Challenges:**

* + Conducting user testing and feedback sessions to iterate on interface designs
	+ Collaborating with IT teams to integrate with existing systems using standardized protocols
	+ Implementing robust security measures such as encryption and access controls
	+ Using version control and documentation tools to track design changes and ensure consistency

**4. Development:**

**Techniques:**

* + Coding according to design specifications
	+ Code reviews and unit testing
	+ Prototyping to validate design concepts
	+ Continuous Integration to integrate code changes

**Activities:**

* Conducting code reviews and unit tests to ensure quality
* Building prototypes to validate key features with stakeholders
* Integrating code changes into the main codebase and deploying to test environments

**Challenges:**

* Meeting deadlines and milestones set in the project plan
* Addressing technical debt and maintaining code quality
* Ensuring compatibility and interoperability with other systems
* Managing dependencies and third-party integrations

**Steps to Overcome Challenges:**

* Breaking down development tasks into smaller, manageable units
* Prioritizing high impact features and functionalities for early delivery
* Allocating time for refactoring and addressing technical debt
* Implementing automated testing and continuous integration practices to detect and fix issues early

**5. Testing: Techniques:**

* + Unit Testing to test individual components
	+ Integration Testing to test interactions between components
	+ System Testing to test the entire system
	+ User Acceptance Testing to validate against user requirements

**Activities:**

* Writing and executing test cases for different levels of testing
* Identifying and reporting bugs and issues
* Conducting regression testing to ensure new features don't break existing functionality
* Facilitating user acceptance testing with stakeholders

**Challenges:**

* + Limited test coverage and resources
	+ Reproducing and fixing complex bugs reported by testers
	+ Coordinating testing efforts across different teams and environments
	+ Balancing time and resources allocated for testing with other project activities

**Steps to Overcome Challenges:**

* Prioritizing test cases based on risk and criticality
* Implementing test automation for repetitive and time-consuming tests
* Establishing clear communication channels between development and testing teams
* Continuously monitoring and adjusting testing efforts based on project priorities and timelines

**6. Project Live/Implementation/Deployment:**

**Techniques:**

* + Deployment Planning to ensure smooth transition to the new system
	+ User Training to familiarize users with the new software
	+ Change Management to address resistance and facilitate adoption
	+ Post Implementation Review to evaluate project success and identify areas forimprovement

**Activities:**

* Planning and coordinating deployment activities with IT and business teams
* Conducting training sessions for Store staff and manager and end users
* Communicating changes and benefits of the new system to stakeholders
* Reviewing project outcomes and gathering feedback for future enhancements

**Challenges:**

* Disruption to Retailer and Supplier system portal during system rollout
* Resistance to change from employees accustomed to old processes
* Ensuring user adoption and proficiency with the new software
* Evaluating project success and identifying lessons learned for future projects

**Steps to Overcome Challenges:**

* Developing a detailed deployment plan with contingencies for potential issues
* Providing comprehensive training and support to address user concerns and build confidence