**Nurturing Process – Waterfall Deliverables – Part -2/2- V2D2 August 2024**

**Waterfall Model Documents**

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

1. *Use Case Diagram*



1. *Activity Diagram*



***Use Case Specifications:***

1. *Registration*

| **Field** | **Description** |
| --- | --- |
| Use Case ID | UC-001 |
| Use Case Name | Seller Registration |
| Use Case Description | Enables new sellers to register on the Flipkart Seller Hub by entering required contact and business details to initiate onboarding. |
| Primary Actor | New Seller |
| Secondary Actor | Seller Hub, Flipkart Server |
| Basic Flow | 1. New Seller navigates to Seller Hub 2. Clicks on "Register" 3. Inputs Phone Number, Email ID, Password, and Address 4. Submits the form 5. System sends verification email 6. Seller verifies email and registration completes |
| Alternate Flow | If email is already registered, system suggests login or password recovery |
| Exceptional Flows | - If mandatory fields are left blank, show validation errors - If email format is invalid, prompt user to re-enter |
| Pre-Conditions | Seller does not already have an account in the system |
| Post-Conditions | Seller account is created and marked as pending for KYC |
| Assumptions | - Internet connection is stable - Seller provides valid and accessible email ID |
| Constraints | - Phone number must be unique - System must validate fields in real time |
| Dependencies | - Email service must be operational - Database must allow new seller entries |
| Inputs and Outputs | Input: Seller details (Email, Phone, Password, Address) Output: Confirmation screen and email verification |
| Business Rules | - All fields are mandatory - Password must be 8+ characters, with one uppercase, lowercase, number |
| Miscellaneous Information | Captcha used to prevent bot registrations |

1. Login

| **Field** | **Description** |
| --- | --- |
| Use Case ID | UC-002 |
| Use Case Name | Seller Login |
| Use Case Description | Allows registered sellers to securely log in to their Flipkart Seller Hub account. |
| Primary Actor | Existing Seller |
| Secondary Actor | Seller Hub Flipkart’s Server |
| Basic Flow | 1. Seller opens the login page 2. Enters valid Email ID and Password 3. System verifies credentials 4. If correct, seller is redirected to the dashboard |
| Alternate Flow | - If credentials are incorrect, show error and offer “Forgot Password” link |
| Exceptional Flows | - After 3 failed attempts, account is locked for 30 minutes - If server is down, show maintenance message |
| Pre-Conditions | Seller must be registered and have an active account |
| Post-Conditions | Seller gains access to the Seller Hub dashboard |
| Assumptions | - Internet is available - Authentication system is up |
| Constraints | - Max 3 login attempts per hour - Passwords are case-sensitive and encrypted |
| Dependencies | - Database for credential verification - Authentication API must be active |
| Inputs and Outputs | Input: Email ID, Password Output: Access granted or error message |
| Business Rules | - Enforce password security policy - Lock account after 3 failed logins |
| Miscellaneous Information | Login logs must be stored for security audits |

1. Forgot Password

| **Field** | **Description** |
| --- | --- |
| Use Case ID | UC-003 |
| Use Case Name | Forgot Password |
| Use Case Description | Allows registered sellers to reset their password securely using a reset link sent to their registered email. |
| Primary Actor | Seller |
| Secondary Actor | Seller Hub System |
| Basic Flow | 1. Seller clicks “Forgot Password” on the login page 2. Enters registered Email ID 3. System sends password reset link 4. Seller clicks the link and sets a new password 5. System confirms and updates password |
| Alternate Flow | - If email ID is not found, display error and allow retry or escalation to support |
| Exceptional Flows | - If the reset link has expired or is reused, show error and prompt re-initiation |
| Pre-Conditions | Seller must have a registered and active email address |
| Post-Conditions | Password is updated and seller can log in with the new credentials |
| Assumptions | - Seller has access to the email inbox - Email service is available |
| Constraints | - Password reset link expires in 15 minutes - New password must meet security criteria |
| Dependencies | - Email delivery service - Secure token validation mechanism |
| Inputs and Outputs | Input: Registered Email ID, New Password Output: Confirmation screen and login prompt |
| Business Rules | - Password must include at least 1 uppercase, 1 lowercase, 1 number, and be 8+ characters long |
| Miscellaneous Information | Reset password attempts should be logged for fraud detection and audit |

1. Documentation KYC

| **Field** | **Description** |
| --- | --- |
| Use Case ID | UC-004 |
| Use Case Name | Upload KYC Documents |
| Use Case Description | Allows sellers to upload legally required documents (PAN, Bank Proof, Brand License, etc.) for verification by Flipkart’s Vendor Ops team. |
| Primary Actor | Seller |
| Secondary Actor | Vendor Ops Executive |
| Basic Flow | 1. Seller logs into Seller Hub 2. Navigates to the KYC section 3. Uploads Pickup Address Proof, PAN Card, Bank Statement, and Brand License 4. Vendor Ops team receives and reviews the documents 5. Status is updated to “Verified” or “Rejected with Reason” |
| Alternate Flow | - If a document is rejected, seller is prompted to re-upload within 72 hours |
| Exceptional Flows | - If file format is unsupported, show error - If file exceeds size limit, prompt compression |
| Pre-Conditions | Seller must be logged in and have a valid Seller ID |
| Post-Conditions | KYC verification status is updated in the Seller Hub |
| Assumptions | - Documents uploaded are legible and in correct format - Vendor Ops SLA is 72 hours |
| Constraints | - File types allowed: JPG, PNG, PDF - Maximum size per file: 2MB |
| Dependencies | - Vendor Ops team availability - File storage and database connection |
| Inputs and Outputs | Input: KYC document files Output: KYC status and confirmation screen |
| Business Rules | - All 4 documents are mandatory - Sellers can reattempt upload 2 times |
| Miscellaneous Information | KYC must be completed before a seller is allowed to list any products |

1. Listing

| **Field** | **Description** |
| --- | --- |
| Use Case ID | UC-005 |
| Use Case Name | Product Listing |
| Use Case Description | Enables sellers to list new products on Flipkart via manual or bulk upload, subject to SDS and QC validation. |
| Primary Actor | Seller |
| Secondary Actor | Catalog Quality Team |
| Basic Flow | 1. Seller logs in to Seller Hub 2. Navigates to “Add New Product” or selects “Bulk Upload” 3. Fills in product details or uploads Excel template 4. Submits listing for QC review 5. Product goes live upon Catalog Team approval |
| Alternate Flow | - If SDS < 80%, system provides feedback and restricts product from going live - Seller can edit and resubmit |
| Exceptional Flows | - If listing format is incorrect or images are missing, system throws validation errors |
| Pre-Conditions | Seller must have completed KYC and onboarding |
| Post-Conditions | Product is live and visible on Flipkart to customers |
| Assumptions | - Catalog QC tool is available - Sellers are aware of SDS guidelines |
| Constraints | - SDS threshold: 80% - Mandatory fields: Title, Images, Key Attributes |
| Dependencies | - Catalog QC system - Internal taxonomy and listing structure |
| Inputs and Outputs | Input: Product details, images, attributes Output: Listing live status or QC feedback |
| Business Rules | - All new listings must be reviewed within 48 hours - Only approved brands can be listed |
| Miscellaneous Information | Listing performance tracked via FDP dashboard after go-live |

1. Dashboard Access

| **Field** | **Description** |
| --- | --- |
| Use Case ID | UC-006 |
| Use Case Name | Dashboard Access |
| Use Case Description | Allows sellers to view key performance indicators (KPIs) including orders, ad spend, inventory, payments, and returns through the Seller Hub dashboard. |
| Primary Actor | Seller |
| Secondary Actor | Seller Hub System |
| Basic Flow | 1. Seller logs in to Seller Hub 2. Clicks on “Dashboard” 3. System retrieves and displays widgets for orders, returns, ads, payments, and inventory 4. Seller can drill down into each metric |
| Alternate Flow | - If one widget fails to load, others are still accessible - Provide fallback messages or retry option |
| Exceptional Flows | - If dashboard service is down, show “Currently Unavailable” notice |
| Pre-Conditions | Seller must be logged in and have transactions associated with their account |
| Post-Conditions | Seller views and/or downloads dashboard data |
| Assumptions | - BI platform (FDP) is integrated - Seller permissions allow access to data |
| Constraints | - Dashboard data refreshes every 24 hours - Data displayed only for active SKUs |
| Dependencies | - FDP (Flipkart Dashboard Platform) - Data pipelines from Ads, Payments, Orders, Inventory |
| Inputs and Outputs | Input: Seller login, dashboard selection Output: Visual metrics, charts, tables |
| Business Rules | - Only the seller’s own data is accessible - Historical data retention for 12 months |
| Miscellaneous Information | Export to Excel/PDF feature available; weekly snapshots also emailed |

**Document 7- Screens and pages**

1. Home Page
2. Login
3. Onboarding Status
4. Listing
5. Dashboard











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

During the execution of the Waterfall project, I used Microsoft Visio to create structured visual diagrams such as Use Case Diagrams and Activity Diagrams. Visio helped me clearly represent the interaction between actors and systems, and map process flows for scenarios like Seller Onboarding and Product Listing. The diagramming capabilities of Visio enabled better communication during stakeholder walkthroughs and helped validate business logic before implementation. Additionally, I used Axure RP to design wireframes and mock screens for key modules of the Seller Hub. These included pages such as Seller Registration, KYC Upload, Product Listing, and Dashboard Overview. Axure allowed me to visually simulate user interactions and present clickable prototypes, which were especially useful for demonstrating functional expectations during UAT. My hands-on experience with both Visio and Axure significantly enhanced the clarity, documentation quality, and visual storytelling of the overall BA deliverables.

**Document 9- BA experience**

***1.*** *Requirement Gathering*

During the initial phase of the project, I led the requirement gathering process using the MoSCoW technique to prioritize inputs from multiple stakeholders across business (Category, Vendor Ops, Ads, Catalog) and IT (Vendor Hub, Hermes, FDP). Given that the primary business sponsor was intermittently unavailable, I proactively identified secondary points of contact from each vertical to ensure continuity in information flow and requirements validation.

Many inputs collected in early workshops were broad, vague, or duplicated across teams. For example, both the Ads team and Vendor Ops shared overlapping requirements around ad spend visibility. Similarly, Catalog and Category teams had repeated concerns related to SDS, which we later merged into a unified Catalog QC module. I used the FURPS model to validate each requirement across Functionality, Usability, Reliability, Performance, and Security.

To further refine the requirements and avoid ambiguity, we used prototyping tools such as Axure to build mock screens for Seller Registration, KYC Upload, and Product Listing. These screens were reviewed by business teams and helped elicit more specific and testable requirements — especially around catalog validations, image format rules, and SDS scoring. This phase also set the foundation for traceability in our RTM, ensuring no critical business goal was missed or misinterpreted.

*2. Requirement Analysis*

In this phase, I translated the elicited requirements into detailed process models and visual documentation to ensure clarity and alignment. I created UML diagrams, including Use Case Diagrams and Activity Diagrams, to map critical business functions such as Seller Registration, KYC Document Upload, Product Listing, and Dashboard Access. These diagrams helped represent the relationship between actors and system functionalities, especially for modules that spanned multiple teams.

Additionally, activity diagrams were created to explain the flow of operations such as FBF stock routing and SLA-based order management. These visuals were shared with cross-functional teams including Catalog, Vendor Ops, Ads, and Engineering for feedback. At times, team members suggested changes — such as modifying the flow of product listing approvals or adjusting SLA breach logic — and as a BA, I incorporated these updates after re-validating them with relevant stakeholders.

I also prepared the BRS (Business Requirement Specification) document, which captured high-level business needs, goals, and stakeholder expectations — for instance, reducing SLA from 10 to 5 days, achieving 80% SDS, and enforcing ad spend compliance. In parallel, I created the SRS (Software Requirement Specification), which broke down these needs into system-level functional and non-functional requirements, including mobile dashboard access, LMS integrations, QC workflows, and role-based access control.

This phase ensured that business logic was well-understood and technically implementable before moving into design and development.

1. ***Design Phase***

In the Design phase, we transitioned from analyzing requirements to preparing system and test specifications. Based on the validated use case diagrams for Seller Registration, Product Listing, FBF stock allocation, Vendor Dashboard, and SLA Reporting, we began drafting detailed test cases.

These test cases were written to cover both positive scenarios—such as successful product uploads and KYC approvals—and negative cases, like incomplete documentation or non-compliant SLAs. Each functionality was carefully broken down to ensure no logical branch or UI element was left untested. For example, when designing test cases for the dashboard view, we covered filters like region-wise AOV, SDS %, Ads spend compliance, and selection design status.

We also created dummy data to simulate real-time scenarios. For instance, mock vendor profiles, trial stock entries across Flipkart’s warehouses, and fake cancellation/return requests helped ensure robustness in functionality. In parallel, we updated the RTM (Requirements Traceability Matrix) to map each business requirement from the BRD/SRS to one or more test cases, confirming test coverage.

Throughout this phase, the design documentation and flow diagrams were reviewed with product managers, category heads, and engineers to ensure a shared understanding before development kicked off. This collaborative and systematic approach ensured a solid foundation for the build stage.

1. ***Development Phase***

During the Development phase, I facilitated multiple Joint Application Development (JAD) sessions involving stakeholders from business (category managers, vendor ops, ads) and the technical team (developers, QA, architects). These sessions were structured to walk through the use cases and logic flows—such as catalog QC scoring, FBF routing, SLA tracking, and LMS certification triggers.

There were situations where some engineers or data stakeholders initially disagreed with how catalog validation logic or ad spend compliance was scoped. As the BA, I addressed these concerns through one-on-one discussions outside group calls. I explained how their cooperation impacted go-live timelines, vendor experience, and Flipkart’s broader goals—like achieving ₹150 Cr AOP and maintaining SLA < 5 days. These direct yet empathetic conversations helped reset expectations and build a collaborative, respectful environment across the cross-functional teams.

We continuously referred to the finalized use case diagrams and activity flows to support coding logic—especially for modules like SDS scoring, SLA tracking by region, and vendor onboarding workflows in Vendor Hub. These visual references helped developers translate business requirements into scalable code units.

Since the teams were spread across Catalog Ops, Engineering, and BI, organizing meetings with everyone present was often a challenge. In such cases, I ensured all sessions were recorded and shared with absentees, and then scheduled one-on-one recaps to keep everyone aligned. I also documented follow-ups, questions raised, and unresolved dependencies clearly for each meeting.

By staying consistent, communicative, and proactive, the development phase progressed smoothly with minimal rework and high traceability to the original BRD and RTM.

1. ***Testing Phase***

In the Testing phase, I led the creation and execution of test cases derived directly from the approved use cases and functional specifications. These test cases covered a wide range of scenarios, including vendor onboarding, SDS scoring validation, SLA monitoring, ad spend tracking, FBF stock routing, and dashboard performance.

High-level testing was conducted across major workflows to ensure that system behavior aligned with business requirements. For instance, I tested the SDS threshold logic across multiple product listings, validated SLA triggers on Hermes-linked flows, and confirmed LMS certification blocks before vendor activation.

To support this testing, I coordinated with the client-side SPOCs (Category and Vendor Ops teams) to acquire realistic test data—such as dummy seller accounts, sample listings, ad campaign IDs, and FC zone mappings. This allowed us to simulate real business scenarios and catch integration issues early.

Throughout testing, the RTM (Requirements Traceability Matrix) was continuously updated to reflect test coverage and outcomes. This ensured that each business and functional requirement was traceable to a test case and marked for closure upon successful validation.

Once the QA team completed internal testing and defect resolution, I obtained formal sign-off from the client on test case results. I also prepared the client for User Acceptance Testing (UAT) by organizing training sessions, walkthroughs of major features (via Axure wireframes and live demos), and furnishing them with a structured UAT test sheet.

This phase ensured that the project was functionally complete, aligned with KPIs, and ready for go-live deployment.

1. Deployment Phase

In the final phase of the project, I facilitated the formal closure and handoff activities. The updated RTM, which traced all business and functional requirements to their respective test cases and UAT results, was forwarded to the client as part of the project closure document. This served as evidence that all deliverables had been fulfilled as per the initial scope defined in the BRD and SRS.

To support end users—including category managers, vendor onboarding teams, catalog reviewers, and seller support—I coordinated the preparation of user manuals for key modules like Seller Registration, Product Listing, Dashboard Access, SDS Monitoring, and FBF Planning. These manuals included annotated screenshots (from Axure wireframes), SOP links, and system navigation guidelines.

I organized and led multiple training sessions to ensure a smooth transition from testing to live operations. These sessions were delivered both live and in recorded format. For critical teams who could not attend the initial session (especially those managing regional operations), I ensured follow-up 1-on-1 walkthroughs were conducted and documented.

Before go-live, I also shared a cutover checklist and issue escalation matrix with all teams. This guaranteed that if any challenge arose post-deployment—whether related to catalog delays, SLA misses, or dashboard access—there was a structured resolution process in place.

The deployment was successful, and the business was able to activate more vendors, improve SDS scores, meet SLA targets, and unlock growth across all four furniture verticals.