**Use case and Activity diagram:**

**Use case**: Hi- level representation the user interaction with the system and their relationships

**Elements**:

**Actors**: is an external entity which interacts with the system. They are outside the system boundary. They can initiate an use case.

* Primary Actors: Initiate the Uses case
* Secondary Actors: Support the use cases but will not initiate the use case, they provide a service to the system or receive output from system.

**Use case:** Each use case represents a complete and meaningful interaction from the actor's perspective. It describes a sequence of actions that provide something of value to the actor.

Elements:

System: Represents a boundary between the system and the users.

Relationship:

* Association: A solid line connecting actor and use case – Indicates interaction between the actor and use case.
* Include: A arrow with a line (labelled as include)to indicate the base use case incorporates the behaviour of another use case.
* Extend: A arrow with a line (labelled as exclude)to indicate the base use case provides additional or optional behaviour.
* Generalization: A solid line indicates that the base use case may inherit the ability of the use case and may have additional capabilities.

Functional use case: Describes what a system should do, and focuses on the features of the system. This is based on the requirements of the client

Non functional use case: Describes how a system should perform and focuses on the performance and quality of system. This is used to improve the performance of the system.

**Activity diagram:** Represents how the process or sequence of activities is carried out. It is a detailed step by step flow of the use case from start to end.

Elements: Start/End, Actions, Decisions, Fork and joint

Start Node: is a solid circle represents the start of the activity flow

End Node: is a solid circle within a circle represents the end of the activity flow

Decision Node: is a diamond shaped box with single input multiple output box. This is the point where a decision needs to be made based on the condition

Connector: is used to determine the flow of the between the different elements

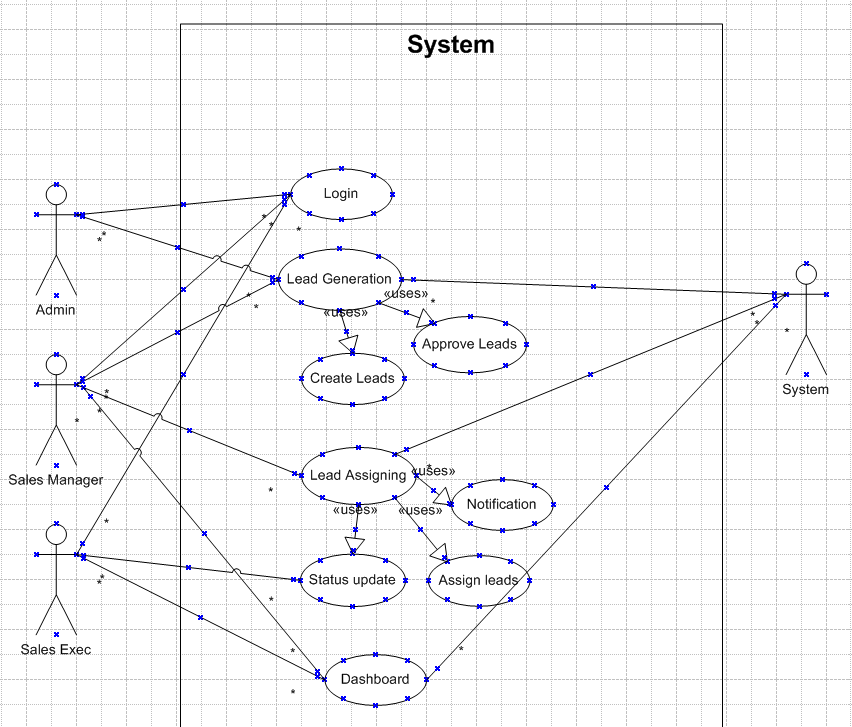
Fork Node: is a solid bar with single input and multiple output flows. This represents the splitting of a single flow of control into two or more parallel (concurrent) flows and all outgoing flows from a fork node are executed concurrently.

Joint Node: is a solid bar with multiple input and single output flow. This represents the synchronization of two or more parallel flows of control. The outgoing flow from a join node is triggered only after all incoming flows have reached the join node.

Action Node: it is a rounded rectangle which represents the task within the activity.

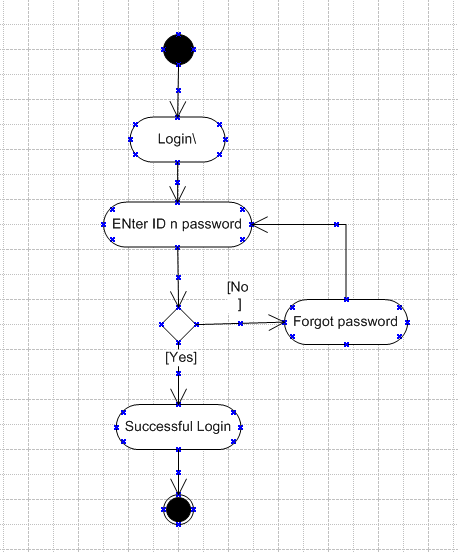
**Document 6- Please prepare a use case diagram, activity diagram and a use case specification document.**

**Use case diagram:**

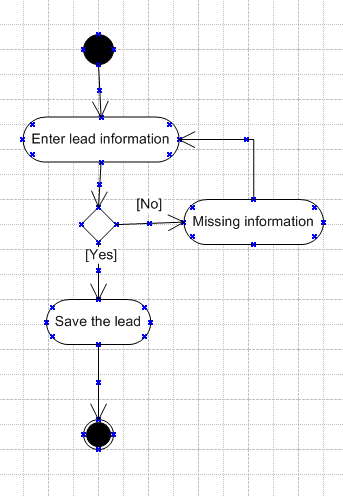


**Activity diagram :**

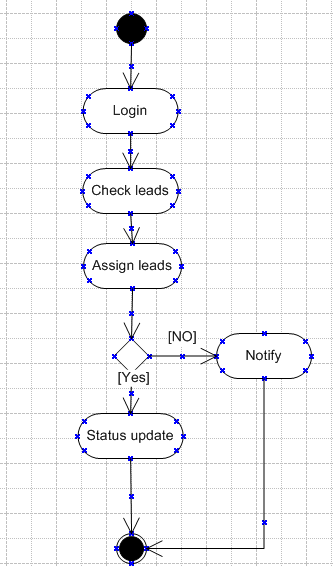
Login Page:



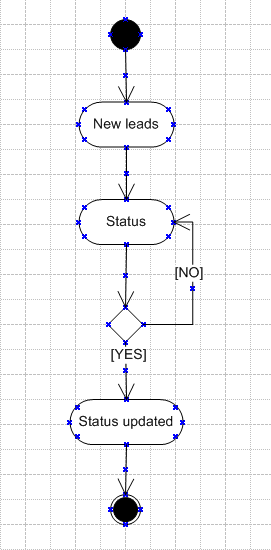
New Lead Generation:



Assign Lead:



Update Leads



**Use case specs**

**Business Requirements**

|  |  |
| --- | --- |
| **BR001** | Lead Capture- The system must allow Sales Representatives to capture new leads with complete details including contact information, client requirements, source, |
| **BR002** | Lead Assignment- Sales Managers must be able to view unassigned leads and assign them to specific Sales Representative |
| **BR003** | Lead Status T-racking- Sales Representatives should be able to update the status of leads (e.g., Follow-up, In Discussion, Closed-Won, Closed-Lost) |
| **BR004** | Notification System- The system must notify Sales Representatives when a lead is assigned to them |
| **BR005** | Dashboard & Analytics- Managers must be able to view performance dashboards, lead conversion reports, and monthly sales insights |

**Use case Specs:**

|  |  |
| --- | --- |
| **Use Case spec 1:- User Login** | |
| **Brief Description** | This use case explains how a user (Sales Representative or Manager) logs into the Lead Management System |
| **Actors** | * Sales Representative * Sales Manager * System (Database/Admin) |
| **Pre-conditions** | * The user must be registered in the system * Internet connection must be available |
| **Basic flow** | * Use case begins with user navigating to the login page * User enters username and password * System validates credentials * On successful validation, user is redirected to the dashboard * Use case ends successfully |
| **Alternate flow** | * Create account * Invalid credentials – error message is shown * User account locked after multiple invalid attempts * System unavailable – display technical error message |
| **Post conditions** | * Successful completion – User is logged into the system * Failure – User unable to access system |
| **Requirements** | * Credentials must be encrypted * Login audit trail must be maintained |

|  |  |
| --- | --- |
| **Use Case spec 2:- Capture New Lead** | |
| **Brief Description** | This use case explains how a Sales Representative adds a new lead into the system. |
| **Actors** | * Sales Representative * System (Database) |
| **Pre-conditions** | * User must be logged in * Required lead details should be available |
| **Basic flow** | * Use case begins with user clicking "Add New Lead" * System displays lead capture form * User enters required details * System validates the input * If valid, the lead is saved to the database * Confirmation message is displayed * Use case ends successfully |
| **Alternate flow** | * Missing mandatory fields – prompt user to complete * Validation failed – show error * Duplicate lead – notify user |
| **Post conditions** | * Lead is saved successfully in the system * Lead creation has failed |
| **Requirements** | * Each lead must have a unique contact number * Entry must include source and status |

|  |  |
| --- | --- |
| **Use Case spec 3:- Assign Lead** | |
| **Brief Description** | This use case explains how a Manager assigns a captured lead to a Sales Representative. |
| **Actors** | * Sales Manager * System * Sales Representative |
| **Pre-conditions** | * Manager must be logged in * Unassigned leads should be available * At least one active Sales Representative |
| **Basic flow** | * Manager selects an unassigned lead * System shows list of available reps * Manager selects a rep * Clicks on "Assign" * System assigns the lead, logs the activity, and sends notification * Use case ends successfully |
| **Alternate flow** | * No active leads – show message |
| **Post conditions** | * Lead is assigned and status updated * Notification is sent |
| **Requirements** | * No Sales Rep should have more than 20 active leads * Assignment should be logged with timestamp |

|  |  |
| --- | --- |
| **Use Case spec 4:- Update Lead Status** | |
| **Brief Description** | This use case describes how a Sales Representative updates the status of an assigned lead. |
| **Actors** | * Sales Manager * System * Sales Representative |
| **Pre-conditions** | * Lead must be assigned to the logged-in user |
| **Basic flow** | * User selects assigned lead * Chooses new status (e.g., Follow-up, Closed) * System saves the update and logs the activity * Use case ends |
| **Alternate flow** | * Invalid status transition – show error |
| **Post conditions** | * Lead status is updated successfully |
| **Requirements** | * Status update must be accompanied by remarks * System must maintain change history |

|  |  |
| --- | --- |
| **Use Case spec 5:- View Dashboard** | |
| **Brief Description** | This use case allows the user to view their personalized dashboard containing KPIs and lead statistics. |
| **Actors** | * Sales Manager * System * Sales Representative |
| **Pre-conditions** | * User must be logged in |
| **Basic flow** | * User clicks on Dashboard * System fetches data and renders charts and reports * User reviews statistics * Use case ends |
| **Alternate flow** | * No data found – show default view * API failure – notify user |
| **Post conditions** | * Dashboard displayed successfully |
| **Requirements** | * Data should be updated in real-time * Managers can view team performance |

RTM

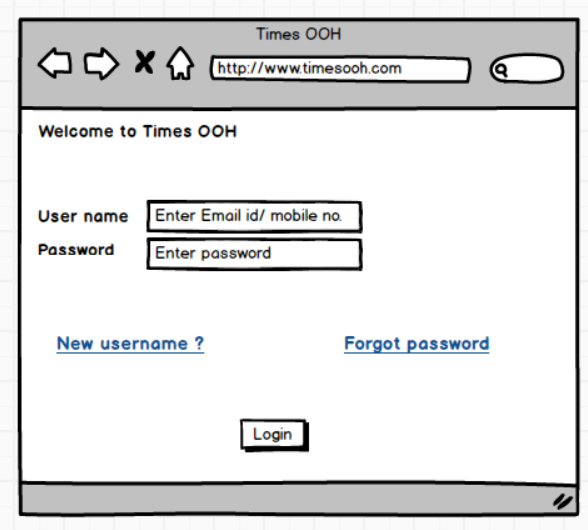
|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Req ID** | **Requirement Description** | **Functional / Non-Functional** | **Design Spec** | **UAT Status** |
| FR001 | User Login to the system | Functional | Yes | Passed |
| FR002 | Capture New Lead | Functional | Yes | Passed |
| FR003 | Assign Lead to Employee | Functional | Yes | Passed |
| FR004 | View Assigned Leads | Functional | Yes | Passed |
| FR005 | Update Lead Status | Functional | Yes | Passed |
| FR006 | Send Notification on Assignment | Functional | Yes | Passed |
| FR007 | Validate Mandatory Fields in Forms | Functional | Yes | Passed |
| FR008 | Login Authentication with Roles | Functional | Yes | Passed |
| FR009 | Generate Reports of Lead Activity | Functional | Yes | Passed |
| FR010 | User Logout | Functional | Yes | Passed |
| NFR001 | Application response time should be under 3 seconds | Non-Functional | Yes | Passed |
| NFR002 | System must be accessible 24/7 | Non-Functional | Yes | Passed |
| NFR003 | Only authenticated users should access the system | Non-Functional | Yes | Passed |
| NFR004 | Mobile responsiveness for Android & iOS | Non-Functional | Yes | Passed |

**Document 7- Screens and pages**

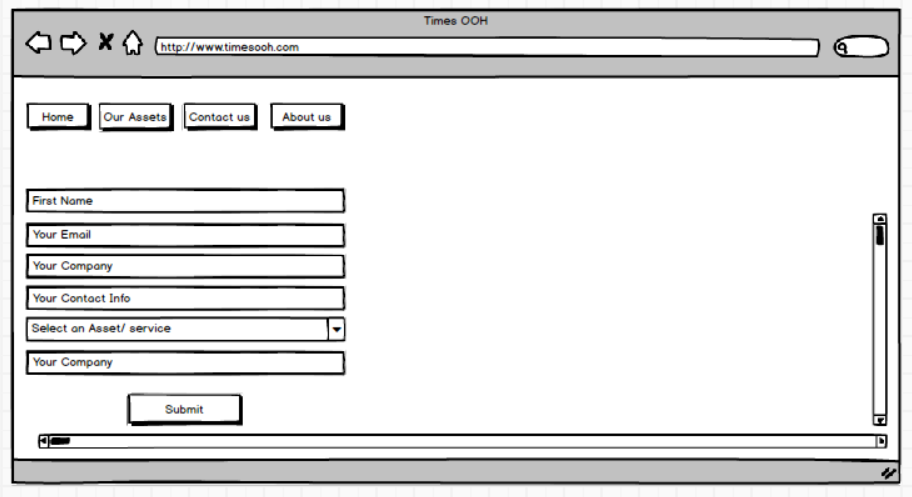
Please follow the following steps to create the mock-ups

* Kindly use balsamic or Axure.

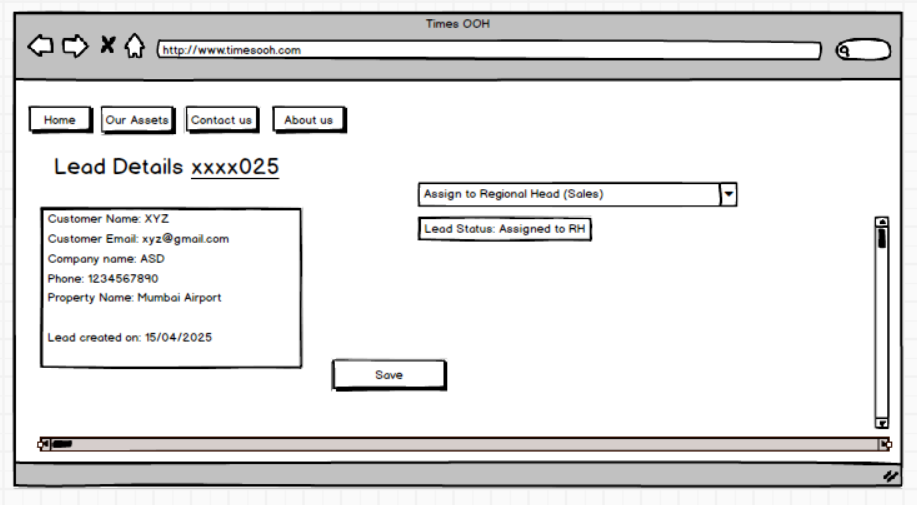
Login Page



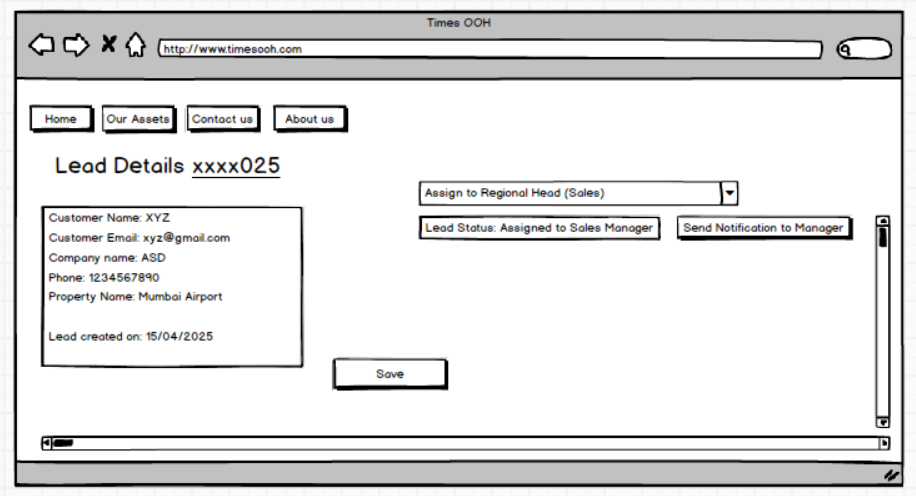
Lead Creation



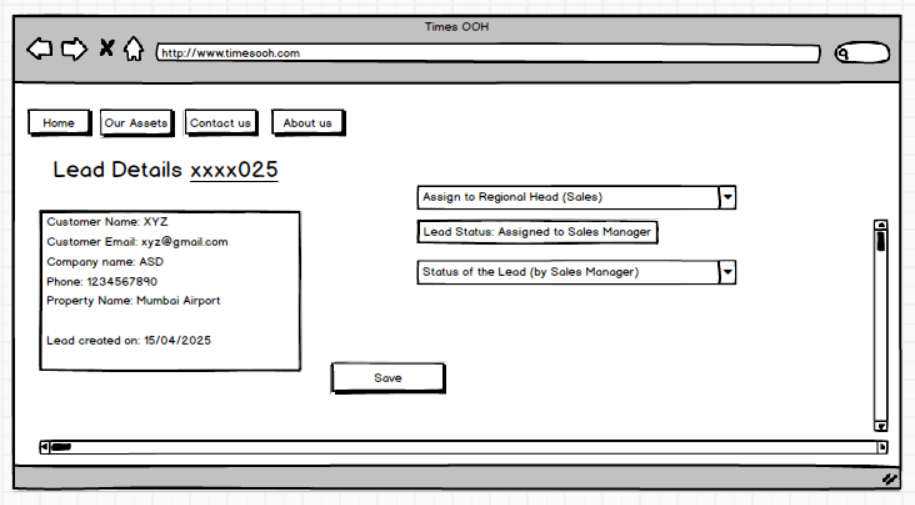
Lead Assign



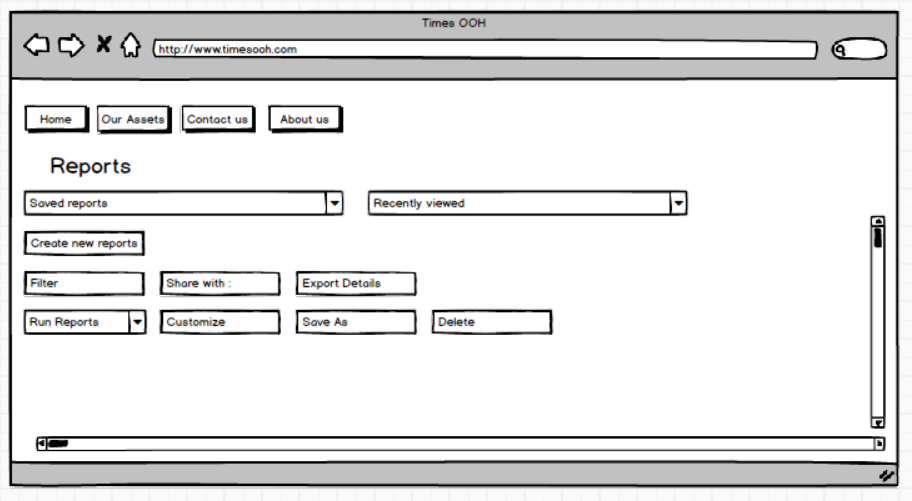
Lead Status:



Status of the Lead



Dashboards:



**Document 8- Tools-Visio and Axure Write a paragraph on your experience using Visio and Axure for the project.**

In this project, Microsoft Visio was extensively used to create use case diagrams, activity diagrams, and process flows for the Lead Management System. Visio provided a highly intuitive and structured interface for visualizing business processes, making it easier to represent system interactions, decision flows, and actor involvement clearly. Its drag-and-drop functionality and support for UML diagrams significantly enhanced the quality and speed of design documentation.

For the user interface (UI) design, Balsamiq was used to create wireframes for the main features like lead creation, lead assignment, and dashboard views. Balsamiq's sketch-style design approach made it ideal for quick mockups, encouraging stakeholder feedback and iterative refinement early in the development lifecycle. The tool’s simplicity helped in rapidly mapping out user journeys and visualizing the screen transitions associated with each use case. Overall, both tools contributed immensely to bridging the gap between business requirements and system design.

**Document 9- BA experience**

My experience as BA in following phases:

**1. Requirement gathering:**

As a Business Analyst for the Lead Management System project in the Outdoor Advertising Industry, my primary responsibility in the requirement gathering phase was to understand the existing processes and problems faced by stakeholders including sales representatives, managers, and administrators.

To initiate the process, I utilized the MoSCoW technique to categorize and prioritize the requirements into Must have, should have, could have, and won’t have. This helped in clearly distinguishing business-critical functionalities like lead creation, assignment, and tracking from the optional or future enhancements.

I conducted one-on-one interviews, JAD (Joint Application Development) sessions, and referred to existing documentation and spreadsheets currently used for lead tracking to gather detailed insights.

To validate the requirements, I applied the FURPS model (Functionality, Usability, Reliability, Performance, and Supportability) to ensure that all angles of the product expectations were addressed.

During elicitation, I came across redundant and duplicate requirements from different users, such as duplicate entries for lead follow-up reminders or multiple fields for the same client detail. I streamlined these to create a cleaner, consolidated requirement list.

Furthermore, I developed prototypes and wireframes using Balsamiq, which allowed stakeholders to visualize the product early and refine their inputs with clarity. This significantly enhanced our understanding and reduced ambiguity in the requirements.

**2. Requirement Analysis:**

My main objective was to translate the gathered business requirements into a format understandable by both technical and non-technical teams.

To begin, I created detailed UML diagrams including use case diagrams and activity diagrams using Microsoft Visio to visually map the system functionalities like lead creation, assignment by managers, and follow-up by sales reps. These visuals helped both the development team and stakeholders clearly understand the system flow.

I also developed use case specifications for each of the identified use cases—like Creating a Lead, assigning a Lead, Updating Lead Status, etc.—detailing actors, basic and alternate flows, pre/post-conditions, and business rules. This helped ensure no functional requirements were overlooked.

During the analysis, I collaborated regularly with the development and testing teams. Some team members had alternative perspectives on certain flows—for example, whether lead reassignments should trigger notifications. As a BA, I took such feedback constructively, consulted stakeholders, and made necessary refinements to the requirements and diagrams.

**3. Design:**

I collaborated closely with the developers to help translate the use case specifications and activity diagrams into technical design documents. These documents outlined the database schema, component interactions, and application flow.

One major responsibility was to **ensure traceability** between requirements and design. I constantly updated the **RTM (Requirement Traceability Matrix)** to reflect any design changes or updates, ensuring every functional and non-functional requirement was accounted for in the system design.

During team interactions, I reviewed the **prototypes and mockups** (built in Balsamiq and Axure) and collected feedback. Any changes from these sessions were documented, communicated to the design team, and incorporated without compromising the timelines.

Design review meetings required detailed walkthroughs. At times, some stakeholders proposed alternative user flows or had concerns about UI/UX design. I facilitated healthy discussions, acknowledged their viewpoints, and coordinated final approvals after consensus.

By the end of this phase, we had a **clear, signed-off design document** and well-defined workflows, setting the stage for a seamless development process.

**4. Development:**

I organized and participated in **JAD (Joint Application Development) sessions** with developers, testers, and UI designers. These sessions helped clarify ambiguous requirements and align the team on technical feasibility. When misunderstandings or conflicts arose during discussions, I handled them diplomatically—offering **one-on-one sessions** to ensure everyone felt heard and supported, maintaining a **positive team environment**.

As development progressed, I acted as a **single point of contact** for the development team to resolve requirement-related queries. Many times, developers reached out for clarification regarding workflows like **lead assignment logic**, **access roles**, or **notification triggers**. I used previously created **UML diagrams, use case specs**, and **process flows** to explain the requirements with precision.

Maintaining clear and timely communication was essential. I scheduled regular **sync-up meetings** with the client and the tech team. If any team member missed a session, I would **record the meeting**, summarize the key points, and conduct **one-on-one catch-ups** to keep everyone in the loop.

Overall, I ensured that the development team remained focused, well-informed, and aligned with business expectations. My contributions played a crucial role in minimizing rework and delays, resulting in a stable and requirement-compliant build ready for the testing phase.

**5. Testing:**

As a Business Analyst, my first responsibility was to **prepare detailed test cases** derived from the finalized **use case specifications** and **RTM**. I wrote both **positive** and **alternate test scenarios** to thoroughly validate the application's behaviour in real-world situations.

I coordinated with the QA team to make sure all test cases were covered and **mapped back to business requirements**, leaving no room for gaps. When testers needed **test data**, I collaborated with the team to obtain realistic and valid datasets—especially important for verifying workflows such as **lead assignment, lead status updates, and user role access**.

I performed **high-level testing** (also known as business testing) to validate core business flows from the user's perspective, like:

* Capturing a new lead
* Assigning leads to sales reps
* Receiving lead update notifications

Before handing off for UAT (User Acceptance Testing), I walked the team through the tested scenarios, addressing any concerns. Once the UAT was completed successfully, I **took official sign-off from the team**, marking the end of the testing phase.

**6. Deployment:**

In the **Deployment Phase** of the Lead Management System project, my role as a Business Analyst extended into ensuring a **smooth transition from development to live usage**.

I also took ownership of coordinating the creation and distribution of **End User Manuals**. These manuals were prepared in collaboration with the QA and development teams and were customized based on user roles—like Sales Manager, Sales Representative, and Admin—making it easier for different stakeholders to understand the system.

To facilitate ease of use, I **planned and organized training sessions** for all end users. This included:

* Live demonstrations of how to use the application
* Walkthroughs of key modules such as lead creation, assignment, updates, and notifications
* Q&A sessions to address doubts

Finally, I worked closely with the development and DevOps teams to **monitor the post-deployment environment**, ensuring there were no critical issues or gaps. Any feedback from the users post-deployment was logged and communicated for potential enhancements.