**Business Analysis Plan**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **PPM #** |  | **Project Name** | | City Transport Ticketing And Real-Time Bus Tracking |
| **Author** | Apaar Jain | **Date** | 16/01/2024 | |

**Table Of Contents**

[**Introduction** 2](#_Toc188287438)

[Purpose 2](#_Toc188287439)

[Scope 3](#_Toc188287440)

[Key Stakeholders 3](#_Toc188287441)

[**Stakeholder Engagement and Communication Approach** 4](#_Toc188287442)

[Stakeholder Register 6](#_Toc188287443)

[Stakeholder Management and Communication Approach 8](#_Toc188287444)

[**Elicitation Approach** 9](#_Toc188287445)

[Identifying Elicitation Sources 9](#_Toc188287446)

[Elicitation Preparation 10](#_Toc188287447)

[Elicitation Techniques 11](#_Toc188287448)

[**Analysis Approach** 13](#_Toc188287449)

[Create and Analyze Models (Incorporated into the Business Analysis Plan) 16](#_Toc188287450)

[Acceptance Criteria 17](#_Toc188287451)

[Requirements Management 19](#_Toc188287452)

[Requirement Verification 19](#_Toc188287453)

[Requirement Validation 20](#_Toc188287454)

[Requirement Prioritization 20](#_Toc188287455)

[Change Management and Traceability 22](#_Toc188287456)

[Product Risk Management 22](#_Toc188287457)

[**Traceability and Monitoring Approach** 25](#_Toc188287458)

[**Solution Evaluation Approach** 28](#_Toc188287459)

[**Sign-Off** 31](#_Toc188287460)

# **Introduction**

## **Purpose**

The purpose of this Business Analysis Plan (BAP) is to define the approach, processes, and activities for performing business analysis throughout the lifecycle of the mobile application development project. This plan aims to obtain a shared understanding and agreement among all stakeholders regarding the business analysis activities, the roles and responsibilities involved, and the skill sets required for successful completion of the analysis work.

By detailing the specific business analysis tasks to be carried out and how they will be executed, this document sets clear expectations for all parties involved, ensuring alignment and avoiding confusion during the execution of the project. It also establishes a framework for tracking progress, identifying any risks or gaps, and making necessary adjustments throughout the project.

The key benefit of this plan is that it ensures all stakeholders are on the same page regarding the approach to business analysis, creating a foundation for effective collaboration, decision-making, and ultimately delivering a product that meets both business and user needs.

## **Scope**

This Business Analysis Plan covers the analysis activities necessary for the development of the mobile application, which will provide users with the ability to book city transport tickets, track real-time bus locations, and generate live, valid tickets for bus conductors. The scope of this planning includes:

* The identification and documentation of business requirements.
* The engagement of stakeholders to ensure all requirements are captured accurately.
* The definition of the processes, roles, and responsibilities for business analysis tasks, ensuring the right resources are allocated to the project.

This plan does **not** cover the technical implementation, development, or testing phases of the project, which will be handled by the development team once the analysis phase is complete.

## **Key Stakeholders**

* **PMC (Project Management Company)**: The primary business sponsor, ensuring that the project aligns with business objectives.
* **End Users (Commuters)**: The primary users who will interact with the mobile app to book transport tickets and track buses.
* **Bus Operators/Conductors**: Individuals who will verify the validity of digital tickets generated by the app.
* **Development Team**: Responsible for translating the business requirements into technical specifications and implementing the solution.
* **Product Managers**: Ensure that the product meets user expectations and business goals throughout the development lifecycle.
* **Third-Party Providers**: Those supplying services such as real-time bus tracking and payment gateway integration.

# **Stakeholder Engagement and Communication Approach**

The Stakeholder Engagement and Communication Approach aims to define the methods, frequency, and channels for engaging and communicating with stakeholders throughout the mobile application development project. This approach will ensure that stakeholders are consistently kept informed, that their needs and expectations are understood and addressed, and that the right information is delivered at the right time, using the most effective communication methods.

By analyzing stakeholder needs, roles, and interests, this approach ensures that all parties involved in the project receive relevant information and feedback, allowing for effective decision-making, alignment, and successful project outcomes.

**Inputs**

1. **Stakeholder Register**:
   * A list of identified stakeholders, including their roles, interests, and influence on the project.
   * Helps determine the communication needs of each stakeholder group.
2. **Business Analysis Plan**:
   * Provides context on how business analysis activities will be carried out and what types of communication will be required at each stage.
   * Ensures that communication is aligned with business analysis goals and objectives.
3. **Project Charter**:
   * Defines the high-level scope and goals of the project.
   * Helps establish the overall expectations and communication requirements for the project’s stakeholders.
4. **Organizational Process Assets**:
   * Existing templates, communication plans, and previous projects that provide insight into communication practices within the organization.

**Tools and Techniques**

1. **Stakeholder Analysis**:
   * A technique to assess stakeholder needs, interests, and influence on the project. This will help categorize stakeholders into groups (e.g., high influence/high interest) and tailor communication methods to each group.
2. **Communication Plan**:
   * A detailed plan that defines the communication goals, methods, frequency, and channels. It will also specify how feedback will be gathered and how concerns will be addressed.
3. **Surveys/Questionnaires**:
   * To gather feedback from stakeholders on their preferences for communication and their level of understanding of the project’s progress and objectives.
4. **Workshops and Focus Groups**:
   * In-person or virtual group discussions where stakeholders can provide feedback, ask questions, and engage with the business analysis team. This helps clarify requirements and align expectations.
5. **Status Reports**:
   * Regular updates on the progress of the business analysis phase, highlighting key achievements, upcoming milestones, and potential issues.
6. **Visual Communication Tools**:
   * Tools such as flowcharts, wireframes, and mockups to help stakeholders visualize requirements, process flows, and the application interface.
7. **Collaboration Tools**:
   * Online platforms (e.g., Microsoft Teams, Slack, or Confluence) for continuous communication and sharing of documents, updates, and feedback.
8. **RACI Matrix**:
   * A tool to clarify the roles and responsibilities of each stakeholder in the decision-making and communication process. This ensures that the right people are engaged at the right time.

**Outputs**

1. **Stakeholder Engagement Plan**:
   * A formalized document detailing how each stakeholder or stakeholder group will be engaged, including their communication preferences, frequency, and method of communication. This plan will also outline specific actions for keeping stakeholders involved and informed throughout the project lifecycle.
2. **Communication Matrix**:
   * A tool that defines the type of communication (e.g., status updates, workshops, reports), the frequency, and the target audience for each communication type. It ensures that every stakeholder receives the information they need at the right time.
3. **Feedback and Issues Log**:
   * A log that tracks feedback, concerns, and issues raised by stakeholders during the project. This will be updated regularly to ensure that all concerns are addressed and stakeholders feel heard.
4. **Regular Status Reports**:
   * Documents providing stakeholders with project updates, progress on business analysis activities, and any potential risks or issues that need attention.
5. **Approved Requirements Documents**:
   * Finalized and approved business requirements, functional specifications, and other key documents that reflect the inputs and feedback from stakeholders throughout the analysis process.

## Stakeholder Register

| **Stakeholder Name** | **Role/Title** | **Interest** | **Influence** | **Engagement Strategy** | **Communication Needs/Preferences** |
| --- | --- | --- | --- | --- | --- |
| **PMC (Project Management Company)** | Business Sponsor / Project Owner | High – Interested in project progress, quality, and alignment with business objectives | High – Has decision-making authority, funding responsibility | Regular high-level status updates, involvement in key decision-making | Monthly executive summaries, meetings as needed |
| **End Users (Commuters)** | Primary Users of the App | High – Need a functional, easy-to-use app for ticket booking and tracking | Medium – Can influence user feedback and experience | Surveys, focus groups, feedback during app testing phases | Regular updates, app usability tests, feedback sessions |
| **Bus Operators/Conductors** | Service Provider / Ticket Validator | Medium – Concerned with validating tickets and operational efficiency | Medium – Can provide feedback on the validation process | Involvement in ticket validation design, user testing | Updates about ticketing system, app validation features |
| **Development Team** | Technical Implementation Team | Medium – Responsible for translating business requirements into technical solution | High – Directly impacts the app’s functionality and success | Regular requirement clarification meetings, sprint reviews (if Agile) | Weekly team meetings, detailed documentation, Jira for tracking issues |
| **Product Managers** | Project Oversight/Requirements Manager | High – Ensures alignment between business goals and product development | Medium – Provides guidance but doesn’t have direct decision-making authority | Periodic progress reports, review of deliverables | Bi-weekly meetings, email updates, product demos |
| **Third-Party Service Providers (Tracking & Payment)** | External Providers for Bus Tracking/Payment Gateway | Medium – Ensures integration is seamless for real-time tracking and payments | Medium – Depend on timely and accurate requirements for integration | Regular coordination meetings to discuss integration requirements | Status calls, integration documentation, email updates |

## Stakeholder Management and Communication Approach

| **Stakeholder Name** | **Communication Goals** | **Communication Method** | **Frequency** | **Responsible Party** | **Message Type** | **Preferred Communication Channel** |
| --- | --- | --- | --- | --- | --- | --- |
| **PMC (Project Management Company)** | Keep informed of overall project progress, risks, and milestones | Status reports, meetings | Monthly or as needed | Business Analyst / Project Manager | Project status, milestone updates, risks | Email, In-person meetings, Reports |
| **End Users (Commuters)** | Ensure the app meets user needs, gather feedback for improvements | Surveys, usability testing, app demos | Bi-monthly, as needed | Business Analyst, UX Team | Feature requests, usability issues, feedback | Online surveys, app testing sessions, email |
| **Bus Operators/Conductors** | Communicate ticket validation process and requirements | Workshops, training sessions | Before app launch, and as needed | Business Analyst, Product Manager | Ticket validation flow, process updates | In-person training, video tutorials, email |
| **Development Team** | Clarify requirements, address technical concerns, review progress | Requirement review meetings, Sprint reviews (if Agile), documentation | Weekly or as needed | Business Analyst, Project Manager | Detailed requirements, clarifications, progress updates | Slack, Jira, Email, Meetings |
| **Product Managers** | Align on business goals and app features, review deliverables | Meetings, product demos, reports | Bi-weekly or as needed | Business Analyst, Project Manager | Business requirements, product features | Email, In-person meetings, Slack |
| **Third-Party Service Providers (Tracking & Payment)** | Ensure smooth integration of bus tracking and payment systems | Coordination meetings, integration testing | Weekly or bi-weekly | Business Analyst, Development Team | Technical requirements, integration progress | Email, Conference calls, Documentation |

# **Elicitation Approach**

The elicitation approach defines how the business analyst will gather and validate the necessary information from stakeholders to ensure that the project requirements are accurately captured and aligned with business goals. This process is crucial for obtaining the insights needed to develop the mobile application for city transport ticketing and bus tracking.

## Identifying Elicitation Sources

**a) Individuals in the Business Analyst Plans:** The business analyst will engage with the following roles to gather critical information for the project:

* **Product Managers**: To understand the business goals, objectives, and expected outcomes of the application.
* **End Users (Commuters)**: To gather insights on user needs, expectations, and pain points in the existing ticketing and transportation systems.
* **Bus Operators/Conductors**: To understand the operational aspects of ticket validation, user interactions, and any challenges faced with current systems.
* **Development Team (Developers)**: To obtain technical insights about the feasibility of implementing the mobile application’s features and integration with existing systems.
* **Third-Party Providers**: For gathering information on technical requirements related to real-time bus tracking and payment gateway integrations.

**b) Models:** The business analyst will also explore models that provide insights into how the system should function. These include:

* **Process Flow Diagrams**: These models will outline how users book tickets, track buses, and validate tickets, ensuring a clear understanding of system functionality.
* **Wireframes/Mockups**: These models will illustrate the user interface design, helping stakeholders visualize how the app will look and feel, which will guide the elicitation of user experience requirements.
* **Use Case Diagrams**: Will help in defining user interactions with the system and will assist in identifying functional requirements.

**c) Other Documented References:** The business analyst will review existing documents to gather background information and ensure all relevant requirements are considered:

* **Existing Documentation on Current Transport Ticketing Systems**: Understanding the pain points and gaps in the current system to inform the design of the new application.
* **Industry Standards and Regulatory Requirements**: To ensure that the application complies with relevant legal and regulatory standards (e.g., accessibility requirements, payment gateway regulations).
* **Competitor Applications**: Analyzing similar applications to understand best practices and any innovative features that could be beneficial.

## Elicitation Preparation

Elicitation preparation is an essential step for ensuring that sessions are organized, effective, and aligned with project goals. It may be formal or informal, depending on the needs of the project. The goal is to ensure that the business analyst is prepared to gather the necessary information and that all stakeholders are aware of the session’s objectives.

**a) Planning the Elicitation Session:**

* **Defining Session Goals**: The business analyst will define specific objectives for each elicitation session. For example, if eliciting feedback from end users, the goal may be to understand their pain points in the current ticket booking system and identify desired features for the new app.
* **Selecting Elicitation Techniques**: Based on the session objectives, the business analyst will determine which techniques will be used to elicit the information:
  + **Interviews**: For in-depth, one-on-one discussions with key stakeholders like product managers or bus operators.
  + **Workshops**: For collaborative sessions involving end users, product managers, and the development team to discuss use cases and gather input on the app’s functionality.
  + **Surveys/Questionnaires**: For gathering broad input from a large group of users (e.g., commuters) on specific features or pain points.
* **Identifying Stakeholders for Each Session**: The business analyst will ensure that the right stakeholders are present for each elicitation session. This may include scheduling interviews or workshops with the identified roles from the stakeholder register (e.g., product managers, end users, etc.).
* **Creating Elicitation Preparation Notes**: Preparation notes are created to guide the session and ensure that all necessary information is captured. These notes should include:
  + **Session Objectives**: Clearly defined goals for what the session will achieve.
  + **Agenda**: A list of topics to be discussed, with time allocated for each.
  + **Questions/Prompts**: A set of questions or prompts to guide the session, ensuring the business analyst probes for the necessary information.
  + **Session Format**: Whether the session will be an interview, workshop, or survey, including the structure and expected outcomes.
  + **Stakeholder Expectations**: Understanding the level of involvement required from each participant, and ensuring their expectations are managed.

**b) Preparing for Probing and Investigative Questions:** Effective elicitation requires a focus on open-ended questions that encourage stakeholders to provide detailed answers.

* **Probing Questions**: These are critical to dig deeper into responses. The business analyst will prepare open-ended, probing questions that encourage stakeholders to elaborate on their answers. For example:
  + "Can you describe a situation where you faced challenges using the current ticketing system?"
  + "What additional features would make the app more useful for you?"
  + "What problems are you trying to solve with real-time bus tracking?"
* **Iterative Questioning**: The business analyst will not settle for the first answer but will ask follow-up questions as necessary. The key to uncovering the right information is in the iterative process of questioning, analyzing answers, and adjusting the questions based on new insights.

**c) Adjusting Expectations and Measuring Progress:** During the elicitation session, it is important for the business analyst to monitor progress against the session's objectives. If new information arises or if an issue is uncovered that requires further investigation, the business analyst should adjust expectations accordingly.

* **Measuring Progress**: After the session, the business analyst will review the notes and determine whether the objectives were met. This helps identify any gaps in the information collected and informs the planning of follow-up sessions.
* **Adjusting Future Sessions**: If certain areas were not fully explored, the business analyst may decide to adjust the approach for future elicitation sessions, refining the questions or techniques used to address the gaps.

## Elicitation Techniques

| **Technique** | **Description** | **Purpose** | **How It Will Be Used in the Project** |
| --- | --- | --- | --- |
| **Interviews** | One-on-one conversations with stakeholders to gather detailed information. | To obtain in-depth understanding of stakeholder needs, expectations, and challenges. | Interviews with **Product Managers** to understand business goals, **Bus Operators** to gather insights on ticket validation, and **End Users** to identify pain points and expectations. |
| **Workshops** | Collaborative sessions involving multiple stakeholders, often used for group decision-making or brainstorming. | To facilitate group discussions and reach consensus on requirements or solutions. | Workshops with **End Users**, **Product Managers**, and **Developers** to define key features, map user journeys, and gather detailed functional requirements. |
| **Surveys/Questionnaires** | Structured forms sent to stakeholders or a large group to gather quantitative or qualitative data. | To gather feedback from a large number of stakeholders efficiently. | **Surveys** sent to **End Users** (commuters) to gather feedback on desired features or issues with the current transport system. |
| **Focus Groups** | A small group of stakeholders discussing a specific topic under the guidance of a facilitator. | To generate ideas, discuss specific issues, or gather opinions from a select group. | A **Focus Group** with **End Users** to discuss the usability of the current system and desired features for the mobile app. |
| **Observation** | The business analyst observes stakeholders interacting with a system or process to understand behaviors and pain points. | To understand how users interact with a system or process, identifying potential issues and opportunities for improvement. | Observing **Bus Operators** and **Commuters** to understand how the current ticketing system works, the challenges faced, and how the app can address them. |
| **Prototyping** | Creating a model or early version of the system to visualize and test requirements. | To validate concepts, features, and user interfaces before full development. | Developing a **prototype** or **wireframe** of the mobile app for **End Users** to interact with and provide feedback on user interface and flow. |
| **Document Analysis** | Reviewing existing documents (e.g., current system manuals, reports, or policies) to extract relevant information. | To understand existing systems, constraints, and requirements. | Reviewing documentation on the **current city transport system**, including ticketing procedures and bus tracking, to ensure the new app addresses current challenges. |
| **Brainstorming** | A group activity to generate creative ideas and solutions through free-form discussion. | To generate ideas for features, processes, or system improvements. | **Brainstorming** sessions with **Product Managers**, **Developers**, and **Bus Operators** to explore innovative features for the app (e.g., user-friendly ticket validation processes). |
| **Use Case Modeling** | Creating use case diagrams to define how users will interact with the system and the expected outcomes. | To capture the functional requirements of the system and ensure all scenarios are covered. | Creating **use case models** to represent how **End Users** will interact with the app (e.g., booking tickets, checking real-time bus location). |
| **Document Reviews** | Reviewing documents, such as existing system specifications, policies, or compliance standards, to identify relevant requirements. | To ensure that requirements are complete and aligned with standards, policies, and regulations. | Reviewing **regulatory guidelines** for accessibility and security standards related to payment gateways and transport systems to ensure the app meets compliance. |
| **Scenario Analysis** | Using hypothetical or real scenarios to explore how the system will behave in different situations. | To analyze potential problems or issues under various conditions. | Analyzing **real-world scenarios** (e.g., how the app handles payment failures, late bus arrivals, or incorrect ticket validations) to ensure robustness and user satisfaction. |
| **Storyboarding** | Creating visual narratives (often in a sequence of images) to illustrate a process or user interaction with the system. | To visualize user experiences and user interface flows. | **Storyboarding** user journeys for **End Users** to visualize how they will interact with the app (e.g., booking a ticket, receiving confirmation). |

# **Analysis Approach**

1**. Understanding and Organizing Information**

The first step in the analysis approach is to **understand** and **organize** the information collected during the elicitation process. This ensures that the business analyst can effectively process all the data and use it to create clear, structured, and usable requirements.

* **Classify and Categorize**: After the information is gathered, it will be classified into categories based on the type of requirement (e.g., functional, non-functional, user experience, system behavior). This process helps to create a structure for the requirements document and ensures that nothing is overlooked.

**Example**:

* + **Functional Requirements**: User ticket booking process, real-time bus tracking, ticket validation.
  + **Non-functional Requirements**: Performance, security, and accessibility requirements for the app.
* **Identify Themes or Patterns**: The business analyst will identify recurring themes or patterns in the data, especially across multiple stakeholder groups. This helps to understand the most important features and functionality from a business perspective.

**Example**: If multiple stakeholders (e.g., **End Users**, **Bus Operators**) emphasize the need for an easy-to-use ticket validation system, it could indicate that the user interface and ticket validation process should be prioritized.

**2. Mapping Business Needs to Requirements**

Once the information is organized, the next step is to map **business needs** to **requirements**. This ensures that all requirements align with the core objectives of the project and reflect the business goals, which include improving user experience, simplifying the ticketing process, and enhancing the efficiency of bus operations.

* **Business Objective Validation**: Ensure that each requirement supports one or more of the project’s business objectives. For example, if the business goal is to increase operational efficiency, the requirement might involve automating the ticket validation process for bus operators.
* **Prioritization**: Work with stakeholders to prioritize requirements based on business value, user needs, technical feasibility, and time constraints. This helps ensure that the most important features are implemented first.

**3. Refining and Validating Requirements**

The next step in the analysis approach is to **refine** and **validate** the requirements. This ensures that the requirements are clear, feasible, complete, and aligned with stakeholder needs.

* **Ensure Completeness**: The business analyst will check if the requirements document includes all the information necessary to guide development. This involves cross-referencing requirements to ensure that no aspect of the project is missing.
* **Clear and Unambiguous Language**: The business analyst will ensure that the requirements are written in clear and concise language, free from ambiguity. This is crucial to avoid misinterpretation by the development team.
* **Validation with Stakeholders**: The requirements will be shared with stakeholders (such as **Product Managers**, **End Users**, **Bus Operators**) for validation. This ensures that the requirements reflect their expectations and needs. Stakeholder feedback will be gathered and incorporated into the requirements as needed.

**4. Documentation and Traceability**

Once the requirements have been refined and validated, the business analyst will document them and establish traceability. Traceability ensures that each requirement can be linked back to its source (e.g., a specific business need or stakeholder input), making it easier to track progress, changes, and ensure alignment throughout the project.

* **Requirements Traceability Matrix**: The business analyst will create a **Requirements Traceability Matrix (RTM)** to map each requirement to the business objectives, stakeholders, and design elements. This helps track which business needs each requirement addresses and ensures alignment throughout the project lifecycle.

**Example of RTM:**

| **Requirement** | **Business Need** | **Stakeholder** | **Design Elements** |
| --- | --- | --- | --- |
| Real-time bus tracking | Improve user experience and operational efficiency | End Users, Bus Operators | Real-time map view, GPS integration |
| Simple ticket validation process | Enhance bus operator efficiency | Bus Operators | QR Code validation, Automated checks |

* **Version Control and Updates**: As the project evolves and new requirements emerge or existing ones change, the requirements will be updated and version-controlled. This ensures that the team works with the most current and agreed-upon requirements.

**5. Analysis Deliverables**

The business analyst will produce key deliverables to document the analysis results and ensure that all stakeholders have a shared understanding of the project’s requirements.

* **Requirements Document**: A comprehensive document that details all business, functional, and non-functional requirements, including acceptance criteria. It will serve as the foundation for the development and testing phases.
* **Requirements Traceability Matrix (RTM)**: A tool that links each requirement to its business need, stakeholder, and design solution to ensure completeness and alignment.
* **Use Cases and User Stories**: Detailed scenarios describing how users will interact with the mobile app (e.g., “As an End User, I want to book a bus ticket easily so that I can travel without delays”). These will be used by the development team to guide coding and testing efforts.

**6. Continuous Communication and Review**

Throughout the analysis phase, the business analyst will continuously communicate with stakeholders to review and refine the requirements. Regular feedback loops will ensure that requirements remain aligned with evolving business goals, user needs, and technical constraints.

* **Review Meetings**: The business analyst will hold regular meetings with key stakeholders to review and discuss the requirements, ensuring all parties remain aligned.
* **Feedback Incorporation**: Stakeholder feedback gathered through reviews will be incorporated into the requirements documentation, ensuring that all concerns are addressed before moving to the development phase.

## Create and Analyze Models (Incorporated into the Business Analysis Plan)

| **Model Type** | **Description** | **Purpose** | **How It Will Be Used in the Project** |
| --- | --- | --- | --- |
| **Scope Models** | Models that define the boundaries of the solution by identifying what is inside and outside the scope. | To establish what will and won’t be included in the solution, ensuring alignment with business goals. | **Scope model** will outline the core features of the mobile app (e.g., ticket booking, real-time bus tracking) and exclude features such as route planning or third-party integrations that aren't part of the initial release. |
| **Process Models** | Models that depict how the solution will be used, showing the flow of activities and interactions. | To demonstrate how the solution will be utilized and to visualize key workflows or processes. | **Process flow diagrams** will show how users (commuters) will book tickets, view real-time bus locations, and validate tickets, helping both stakeholders and developers understand user journeys and functional requirements. |
| **Rule Models** | Models that logically represent the rules and constraints of the solution. | To define and communicate the business rules that govern the system and its processes. | **Business rules model** will be used to define how the app will handle different ticketing scenarios, such as validating tickets, payment methods, fare calculations, and ensuring compliance with regulations. |
| **Data Models** | Models that describe the data used in the system, how it will be structured, and its life cycle. | To ensure proper data management, storage, and flow throughout the system, supporting key functionalities. | **Data models** will outline how the app stores and manages user data (e.g., ticketing information, bus schedules, payment records) and the relationships between different data entities, ensuring data consistency and security. |
| **Interface Models** | Models that show how users will interact with the system or solution. | To define the user interface design, ensuring a clear, intuitive user experience. | **Wireframes/mockups** will be used to visualize how users (commuters, bus operators) will interact with the mobile app, from booking tickets to validating them, ensuring a user-friendly design and smooth interaction flow. |

## Acceptance Criteria

| **Category** | **Acceptance Criteria** | **Description** |
| --- | --- | --- |
| **Functional Requirements** | 1. The app must allow users to book tickets online. | Ensures the primary functionality of the app aligns with user needs for booking tickets. |
|  | 2. The app must display real-time bus location on a map. | Real-time bus tracking is a key feature to provide users with up-to-date information. |
|  | 3. The app must generate a live, valid ticket for each user upon booking. | Valid tickets should be generated to allow users to board the bus without issues. |
|  | 4. The app must support a simple ticket validation process with a green validation symbol. | To enable bus conductors to easily identify valid tickets during travel. |
| **Non-Functional Requirements** | 1. The app must load within 3 seconds for 90% of users. | Ensures that the app's performance meets user expectations for speed and responsiveness. |
|  | 2. The app must be available 99.9% of the time (uptime) to ensure reliability. | Ensures the system operates reliably and can handle peak demand. |
|  | 3. The app must be compatible with both Android and iOS devices. | Ensures the app supports a broad range of users across platforms. |
| **Usability** | 1. The user interface must be intuitive, with no more than three steps to book a ticket. | Ensures a user-friendly experience for end-users and ease of navigation. |
|  | 2. The app must provide clear, understandable error messages and guidance when issues arise (e.g., payment failure). | Ensures users can easily identify and resolve any issues. |
|  | 3. The app must be accessible for users with disabilities (e.g., screen reader support, color contrast). | Ensures inclusivity for all users, including those with visual impairments or other disabilities. |
| **Security & Compliance** | 1. The app must comply with GDPR and other relevant data protection regulations. | Ensures that user data is handled securely and in compliance with data protection laws. |
|  | 2. The app must support secure payment gateways with encryption for transaction data. | Ensures secure transactions for users when purchasing tickets. |
|  | 3. The app must validate user identity during the payment process to prevent fraud. | Ensures that payment data is protected and only legitimate transactions are allowed. |
| **Data Handling** | 1. The app must store user data securely and comply with privacy policies. | Ensures that user data is safely stored and properly handled. |
|  | 2. The app must allow users to access and download their purchase history. | Provides users with easy access to their transaction records and history. |
| **Integration** | 1. The app must integrate with the city’s existing transport system to get real-time data on bus locations. | Ensures that the app provides accurate and timely information about bus routes and locations. |
|  | 2. The app must allow for easy integration with future payment gateways or transport system updates. | Ensures flexibility for future enhancements or system upgrades. |
| **Performance & Scalability** | 1. The app should support at least 10,000 concurrent users without a significant drop in performance. | Ensures that the app can handle high traffic and perform well under peak usage. |
|  | 2. The app should perform optimally on both high-end and low-end devices. | Ensures that the app is accessible and functional for users with different types of devices. |
| **Testing & Quality Assurance** | 1. All identified user stories must be tested and pass acceptance tests. | Ensures that each feature and function has been thoroughly tested and is ready for release. |
|  | 2. The app must undergo user acceptance testing (UAT) with stakeholders prior to launch. | Ensures that stakeholders (including **End Users**, **Product Managers**, and **Bus Operators**) approve the app’s functionality before release. |
| **Support & Maintenance** | 1. The app must include a clear method for users to report bugs or issues. | Ensures that users have a way to report problems and receive support. |
|  | 2. The app must have a defined support process for handling user issues post-launch (e.g., bug fixes, feature updates). | Ensures that a support structure is in place for maintaining and updating the app after launch. |

## Requirements Management

The **Requirements Management** section defines the approach and processes that will be used to manage requirements throughout the project lifecycle, ensuring that they are well-defined, validated, prioritized, and traceable. This section outlines how requirements will be verified and validated, how they will be tracked and managed, and the prioritization techniques that will be used to ensure that the most critical requirements are addressed first.

### Requirement Verification

Verification ensures that the requirements are complete, consistent, clear, feasible, and aligned with the project goals. According to the **Project Management Institute (PMI)** guidelines, verification involves reviewing requirements to ensure that they are accurate and can be met with the available resources and within the project's scope.

**Verification Process:**

* **Requirements Review**: Regular reviews with stakeholders, including project sponsors, product managers, end users, and technical teams, will be conducted to confirm that the requirements are clear, unambiguous, and consistent with the business objectives.
* **Documentation**: Requirements will be documented in a manner that can be easily understood by both technical and non-technical stakeholders, ensuring traceability and alignment with business needs.
* **Cross-Referencing**: Requirements will be cross-referenced with project goals, business objectives, and other related documents to ensure consistency and completeness.

### Requirement Validation

Validation ensures that the requirements meet the stakeholder's needs and align with the business goals. It involves confirming that the right requirements have been captured and that they reflect what the business needs.

**Validation Techniques:**

1. **Delphi Technique**:  
   The **Delphi technique** involves gathering input from a group of experts or stakeholders through multiple rounds of anonymous feedback, refining the requirements after each round. This technique is particularly useful for getting consensus on complex or uncertain requirements and helps mitigate bias from any single stakeholder.  
   **Process**:
   * Select a panel of subject matter experts (SMEs), including bus operators, end-users, and system developers.
   * Ask for their feedback on specific requirements (e.g., real-time bus tracking, ticket validation process).
   * After each round of feedback, summarize and refine the requirements based on expert input.
   * Repeat until consensus is reached on the key requirements for the mobile application.
2. **Traceability Matrix**:  
   The **Traceability Matrix** will be used to validate that each requirement corresponds to its source and aligns with business objectives. It ensures that all requirements are traceable from the business needs through to design and testing, providing a clear linkage between project objectives and requirements.  
   **Process**:
   * Each requirement will be linked to specific business objectives and stakeholder needs in the **Requirements Traceability Matrix (RTM)**.
   * The RTM will ensure that no requirement is overlooked and that all are aligned with the project's scope and goals.
   * The matrix will be reviewed periodically to track changes and ensure alignment throughout the project.

### Requirement Prioritization

**MoSCoW Method:**

The **MoSCoW method** helps categorize requirements based on their priority and importance. Requirements are classified into four categories:

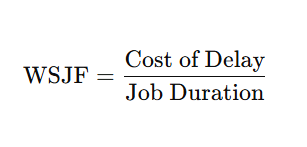
* **Must Have**: These are essential requirements that the app cannot function without. For example, the ability to book a ticket online and validate it.
* **Should Have**: These are important requirements, but the solution can still function without them in the short term. For example, integration with future payment gateways.
* **Could Have**: These are nice-to-have features that would enhance the user experience but are not critical. For example, adding an in-app chat support feature.
* **Won’t Have**: These are features that are out of scope for the current release. For example, route planning and advanced analytics may be excluded from the first version.

The **MoSCoW method** will be applied in collaboration with stakeholders to ensure that only the highest-priority features are included in the first release.

**Weighted Shortest Job First (WSJF):**

**WSJF** is a prioritization technique used to maximize the return on investment by delivering the highest value features first. It evaluates the **cost of delay** divided by the **job duration** to determine the priority of requirements.

**Formula:**



* **Cost of Delay** includes factors such as business value, time-criticality, and risk reduction.
* **Job Duration** represents the effort required to complete the requirement.

**Process**:

* For each requirement, we will calculate the **Cost of Delay** by evaluating the business value, the importance of the requirement, and the impact of delay.
* We will estimate the **Job Duration** based on the effort required to implement the requirement.
* The **WSJF** score will be calculated for each requirement, and those with the highest scores will be prioritized for implementation.

**Example of WSJF Calculation:**

| **Requirement** | **Business Value** | **Time-Criticality** | **Risk Reduction** | **Cost of Delay** | **Job Duration** | **WSJF** |
| --- | --- | --- | --- | --- | --- | --- |
| Real-time bus tracking | High | High | Medium | 9 | 5 | 1.8 |
| Ticket validation system | High | High | High | 10 | 4 | 2.5 |
| User profile management | Medium | Low | Low | 6 | 3 | 2.0 |
| In-app chat support | Low | Low | Low | 3 | 6 | 0.5 |

### Change Management and Traceability

Changes to requirements will be managed carefully to ensure the project stays aligned with the business goals and scope. The **Requirements Traceability Matrix (RTM)** will be used to track changes and ensure that each change is justified and aligned with the project objectives. The RTM will be updated regularly to ensure that all stakeholders are aware of any changes and their impact on the project.

**Change Management Process:**

* **Change Requests**: Any change in the requirements will be documented as a formal change request.
* **Impact Assessment**: Each change request will be assessed for its impact on the project scope, timeline, budget, and resources.
* **Approval Process**: Changes will be approved by the project stakeholders and reviewed by the business analyst to ensure they align with the project’s goals.
* **Update RTM**: The RTM will be updated with any new or revised requirements, ensuring that the traceability and alignment with business objectives are maintained.

## Product Risk Management

**Identifying Product Risks**

The first step in the risk management process is to identify potential risks that could affect the development of the mobile application for city transport ticketing and bus tracking. These risks may include technical challenges, market uncertainties, integration issues, resource constraints, or legal and regulatory considerations.

**Identified Risks for the Project:**

* **Technical Risks**: Issues with integrating real-time bus tracking data into the app, or challenges in ensuring the mobile app works seamlessly across multiple platforms (iOS and Android).
* **Data Security Risks**: Concerns about the handling, storage, and protection of user data, including payment details.
* **User Adoption Risk**: Risk that users may not adopt the mobile app if the user interface is not intuitive or if the app doesn’t meet their expectations.
* **Compliance Risk**: Risk of non-compliance with regulatory standards such as GDPR or data protection laws.
* **Performance Risk**: Risk that the app may fail to handle peak loads or experience performance issues (e.g., crashes or slow loading times during high traffic periods).
* **Integration Risks**: Risk of difficulty in integrating the app with existing city transport systems, including bus schedules, GPS data, and payment processing systems.
* **Financial Risk**: Risk of budget overruns or lack of financial resources to complete the project.
* **Market Risks**: Risk that the product does not meet the actual needs of the users or that competitors introduce a similar product in the market.

**Performing Qualitative Risk Analysis**

Qualitative risk analysis involves evaluating each identified risk in terms of its probability and impact. This helps to prioritize risks and determine the focus areas for mitigation or response planning.

**Risk Evaluation Criteria:**

* **Probability of Occurrence**: The likelihood that a specific risk will materialize.
* **Impact**: The severity of the consequences if the risk materializes.
* **Risk Priority**: A combination of probability and impact that helps identify which risks need immediate attention.

**Qualitative Risk Analysis Table:**

| **Risk** | **Probability** | **Impact** | **Risk Priority** | **Mitigation/Response Strategy** |
| --- | --- | --- | --- | --- |
| **Technical Integration Risk** | High | High | Critical | Develop a prototype for integration testing early. Conduct regular technical reviews. |
| **Data Security Risk** | Medium | High | High | Implement robust encryption for payment data. Regularly audit security measures. |
| **User Adoption Risk** | Medium | Medium | Moderate | Conduct user testing and gather feedback before launch. Focus on intuitive UX design. |
| **Compliance Risk** | Low | High | High | Ensure compliance with GDPR and local data protection laws early in the development process. |
| **Performance Risk** | Medium | High | High | Load testing and performance optimization as part of the development process. |
| **Financial Risk** | Low | Medium | Low | Regular budget tracking and stakeholder communication. |
| **Market Risk** | Medium | Medium | Moderate | Conduct market research and competitive analysis. Gather user feedback early. |

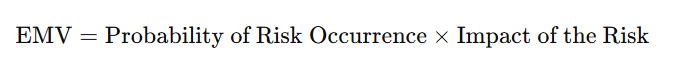
**Performing Quantitative Risk Analysis**

Quantitative risk analysis provides a more in-depth examination of the potential risks identified, typically using data and numerical techniques. This step estimates the potential cost impact or timeline delay for each risk, enabling better preparation for worst-case scenarios.

For this project, **Monte Carlo simulations** and **Expected Monetary Value (EMV)** analysis will be used to quantify risks.

***Expected Monetary Value (EMV) Analysis:***

**EMV** is calculated using the formula:



For example:

* If the risk of **technical integration failure** has a 40% probability of occurring and would result in a delay of $50,000, the EMV for this risk would be:



**Planning Risk Responses**

Once the risks have been identified, analyzed, and prioritized, the next step is to develop strategies for managing those risks. The risk management strategies are divided into **negative risks** (threats) and **positive risks** (opportunities).

**Strategies for Negative Risks (Threats):**

These strategies aim to minimize or avoid the adverse effects of potential risks.

1. **Avoid**: Alter the project plan to eliminate the risk or protect the project from its impact.
   * **Example**: If there is a risk of non-compliance with regulations, the project plan will include legal reviews and updates to meet the necessary standards.
2. **Mitigate**: Reduce the probability and/or impact of the risk to an acceptable level.
   * **Example**: To mitigate the risk of poor performance, regular load testing and optimization efforts will be implemented.
3. **Transfer**: Shift the risk to a third party, such as through insurance or outsourcing.
   * **Example**: Outsourcing data security management to a specialized provider who ensures compliance with data protection laws.
4. **Accept**: Acknowledge the risk and prepare to manage it if it occurs.
   * **Example**: Accepting the possibility of minor market risks but planning for the flexibility to pivot after launch based on user feedback.

**Strategies for Positive Risks (Opportunities):**

These strategies aim to exploit or take advantage of opportunities that may arise from the risks.

1. **Exploit**: Take proactive steps to ensure the opportunity is realized.
   * **Example**: If the mobile app becomes particularly successful with early adopters, we may look for ways to expand the features early.
2. **Share Ownership**: Partner with another organization to share the rewards and risks of the opportunity.
   * **Example**: Collaborating with a third-party payment processor to enhance the app’s payment system features.
3. **Enhance**: Increase the probability of the opportunity happening or the impact if it occurs.
   * **Example**: Accelerating market research efforts to identify a competitive edge, such as unique app features or user experiences.
4. **Accept**: Acknowledge the potential opportunity but choose not to pursue it at this time.
   * **Example**: If the app performs well early in a particular market segment, decide to capitalize on it after completing the primary release.

# **Traceability and Monitoring Approach**

**Traceability of Requirements**

Traceability refers to the process of tracking each requirement from its origin through its lifecycle, ensuring that it is addressed in the final solution and aligned with business objectives. The **traceability matrix** (RTM) will be used to provide a clear, documented link between each requirement and the project goals, design, development, testing, and delivery phases.

**Traceability Activities:**

* **Requirements Mapping**: Every requirement will be mapped to its source (e.g., business need, stakeholder request, regulatory compliance) to ensure alignment with the project’s strategic goals.
* **Traceability Matrix (RTM)**: A Requirements Traceability Matrix will be maintained to track the status of each requirement as it progresses through the lifecycle stages—business analysis, design, development, and testing.
* **Cross-Referencing with Business Objectives**: Requirements will be linked to the high-level business objectives and goals defined at the outset of the project to ensure that each requirement contributes to the overall project success.
* **Linking to Test Cases**: Each requirement will be linked to specific test cases in the testing phase. This ensures that every requirement is validated and verified during testing to confirm that the final product meets the expected criteria.

**Traceability Matrix Example:**

| **Requirement ID** | **Requirement Description** | **Source** | **Business Objective** | **Design Phase** | **Development Phase** | **Testing Phase** | **Status** |
| --- | --- | --- | --- | --- | --- | --- | --- |
| REQ-01 | Allow users to book tickets online | Stakeholder Request | Improve customer experience | In Progress | In Progress | Pending | Open |
| REQ-02 | Display real-time bus location on the app | Business Need | Increase operational efficiency | Completed | Completed | Passed | Closed |
| REQ-03 | Generate a valid ticket with a green validation | Regulatory Requirement | Ensure compliance | In Progress | Not Started | Pending | Open |
| REQ-04 | Multi-platform support for Android and iOS | Market Demand | Expand user base | In Progress | In Progress | Pending | Open |

**Managing Requirement Changes**

Managing changes to requirements is critical to ensure that the project remains within scope and delivers the intended value. Requirements can evolve over time as new insights emerge or market conditions change. The approach for managing these changes involves clear procedures for evaluating, approving, and implementing changes, as well as keeping stakeholders informed.

**Change Management Process:**

* **Change Request Process**: Any proposed changes to the requirements will follow a formal **change request process**. This ensures that changes are documented, evaluated, and approved before implementation.
  + **Initiation**: A stakeholder or team member submits a change request outlining the reason for the change, the potential impact on the project, and any related costs.
  + **Impact Assessment**: The Business Analyst will assess the impact of the change request, including any effect on the scope, schedule, budget, and resources.
  + **Stakeholder Approval**: The change request will be reviewed by key stakeholders, including the project sponsor and product owners, who will decide whether to approve or reject the change.
  + **Implementation**: If the change request is approved, the Business Analyst will update the requirements documentation, the Traceability Matrix, and other relevant project artifacts to reflect the new or modified requirement.
* **Change Control Board (CCB)**: A **Change Control Board** consisting of key stakeholders (e.g., project sponsor, business owners, development team) will review and approve all major requirement changes. This board will ensure that changes are aligned with the overall project goals and that the impact on cost, time, and resources is managed appropriately.
* **Version Control**: A **version control system** will be used to manage changes to requirement documents, ensuring that any updates are properly tracked and previous versions can be referenced if needed. This will also ensure transparency and accountability.
* **Communication of Changes**: All approved changes will be communicated to the relevant team members and stakeholders to ensure that everyone is aligned and aware of the new direction. Updated requirements will be reflected in the project plan, timelines, and task assignments.

**Monitoring Requirements and Progress**

Monitoring is essential to ensure that the requirements are being implemented as planned and that the project is progressing towards its objectives. Progress will be tracked against the defined requirements, with regular checkpoints to evaluate the status of the requirements and their delivery.

**Monitoring Activities:**

* **Regular Status Meetings**: Weekly or bi-weekly status meetings will be held with the project team to discuss the progress of each requirement, any issues or delays, and next steps.
* **Dashboard Tracking**: A **project dashboard** will be created, allowing stakeholders to easily track the status of each requirement in real-time. The dashboard will provide a visual representation of progress, with color-coded indicators to show whether requirements are on track (e.g., green), delayed (e.g., yellow), or at risk (e.g., red).
* **Milestone Reviews**: Major milestones, such as the completion of the design phase, development phase, and testing phase, will be reviewed against the requirements to ensure that they align with the original scope.
* **Testing and Verification**: During the testing phase, each requirement will be verified by the QA team to ensure that it has been correctly implemented. The results will be logged, and any issues will be tracked and resolved.

**Monitoring and Reporting:**

* **Weekly Reports**: A **weekly report** will be prepared that includes:
  + Progress against each requirement.
  + Any changes made to the requirements.
  + Risk and issue tracking.
  + Updated timelines and milestones.
* **Issue Tracking and Resolution**: Any issues identified during the implementation of requirements will be tracked using an issue management tool. The team will work together to resolve these issues in a timely manner.
* **Stakeholder Involvement**: Stakeholders will be actively engaged throughout the monitoring process to review progress, provide feedback, and ensure the project stays aligned with their expectations.

**Traceability and Monitoring Tools**

To support traceability and monitoring, several tools will be used throughout the project lifecycle:

* **Requirements Management Tool**: A dedicated **requirements management tool** (e.g., Jira, Microsoft Azure DevOps, or IBM Rational DOORS) will be used to document, trace, and manage the lifecycle of all requirements.
* **Project Management Software**: Tools like **Trello**, **Asana**, or **Microsoft Project** will be used to track overall project progress and milestones, integrating with the requirements management tool to ensure alignment.
* **Version Control System**: Tools like **Git** or **Subversion** will be used to manage changes to requirement documentation and ensure all versions are tracked.
* **Communication Tools**: **Slack** or **Microsoft Teams** will be used for real-time communication and collaboration on changes and issues.

# **Solution Evaluation Approach**

**1. Solution Evaluation Objectives**

* **Verify Business Goals**: Ensure that the application supports and delivers on the business goals, such as improving user experience and enhancing operational efficiency for city transport services.
* **Measure Performance**: Assess the technical performance of the solution in terms of scalability, speed, security, and stability, particularly during peak usage times.
* **Validate Functional Requirements**: Confirm that all functional requirements (e.g., ticket booking, real-time bus tracking) have been implemented correctly and meet stakeholder expectations.
* **Evaluate User Satisfaction**: Gather feedback from end-users to understand their experience with the app, identifying usability issues or gaps in features.
* **Compliance Check**: Ensure that the application complies with legal and regulatory requirements, including data protection laws and payment security standards.
* **Cost vs Benefit Analysis**: Assess whether the cost of development aligns with the benefits realized post-launch, including user adoption and operational efficiency improvements.

**2. Evaluation Activities**

**Evaluation Phases:**

* **Pre-launch Evaluation**:
  + **User Acceptance Testing (UAT)**: Engage real users in a structured UAT process to validate whether the solution meets their expectations and requirements.
  + **System Testing**: Conduct thorough system testing to verify that the app functions correctly under various conditions (e.g., high traffic, multiple users).
  + **Security Audit**: Ensure the app is secure, especially in terms of payment processing and user data protection.
* **Post-launch Evaluation**:
  + **Monitor Usage Metrics**: Track metrics such as active users, load times, crash rates, and user behavior to understand how the app is performing in real-world conditions.
  + **User Feedback Collection**: Collect feedback from users through surveys, reviews, and in-app feedback mechanisms to assess satisfaction and identify areas for improvement.
  + **Incident and Issue Tracking**: Track any incidents or technical issues reported by users or the support team. Analyze recurring issues and implement fixes as necessary.
* **Ongoing Evaluation**:
  + **Performance Reviews**: Evaluate the system’s ongoing performance in production. Monitor server load, response times, and system uptime.
  + **Feature Usage Analysis**: Track which features are most and least used, helping the team to prioritize future improvements or changes.

**3. Metrics and Key Performance Indicators (KPIs)**

* **Business Metrics**:
  + **Increased Ticket Sales**: Measure any increase in online ticket sales post-implementation.
  + **Operational Efficiency**: Assess whether the app has streamlined ticketing and bus tracking processes, resulting in better resource management.
* **Technical Metrics**:
  + **App Stability**: Track the number of crashes or failures per user session.
  + **Response Time**: Measure how quickly the app loads, processes transactions, and updates bus locations.
  + **Security Compliance**: Track the number of security vulnerabilities or breaches.
* **User Experience Metrics**:
  + **User Satisfaction Score**: Use ratings, feedback, and surveys to assess overall user satisfaction.
  + **Net Promoter Score (NPS)**: Measure how likely users are to recommend the app to others.
  + **App Retention Rate**: Track how many users continue to use the app after their first download.
* **Financial Metrics**:
  + **Cost per Transaction**: Measure the cost of processing a ticket transaction compared to the revenue generated.
  + **ROI (Return on Investment)**: Analyze the app’s financial returns compared to development and operational costs.

**4. Tools and Techniques for Solution Evaluation**

* **User Surveys & Feedback Tools**: Use tools such as **SurveyMonkey**, **Google Forms**, or in-app feedback features to collect qualitative and quantitative user feedback.
* **Analytics Tools**: Utilize tools like **Google Analytics**, **Firebase Analytics**, or **Mixpanel** to track app usage, performance metrics, and user behavior.
* **Performance Monitoring Tools**: Use tools like **New Relic**, **Datadog**, or **AppDynamics** to monitor the technical performance of the app, including load times, crash rates, and server performance.
* **Security Audit Tools**: Conduct security audits using tools like **OWASP ZAP**, **Burp Suite**, or **Qualys** to ensure that the app adheres to security standards.
* **Issue Tracking Tools**: Use project management tools like **Jira** or **Trello** to track and resolve any post-launch technical issues.
* **A/B Testing**: Use **A/B testing** to compare different versions of the app, especially for user interface and experience features, to determine which options result in better user engagement and satisfaction.

**5. Solution Evaluation Deliverables**

* **Evaluation Report**: A detailed report summarizing the findings from the evaluation process, including user feedback, performance metrics, and compliance checks.
* **Recommendations for Improvements**: Based on the evaluation, a list of prioritized recommendations for enhancing the app’s functionality, performance, or user experience.
* **Post-launch Support Plan**: A plan for ongoing support and updates based on user feedback, incident tracking, and performance monitoring.
* **Final Cost vs. Benefit Analysis**: An analysis of whether the costs of development, maintenance, and operations were justified by the benefits achieved, including user adoption and satisfaction.

**6. Evaluation Governance and Roles**

* **Business Analyst**: Responsible for leading the evaluation process, gathering data, analyzing results, and ensuring the solution meets business goals.
* **Project Manager**: Oversees the timeline for evaluations, ensures resources are allocated, and manages post-launch fixes or improvements.
* **QA Team**: Responsible for conducting UAT and system testing, ensuring the solution meets the functional and technical requirements.
* **Development Team**: Assists in the post-launch evaluation by addressing bugs, performance issues, or feature improvements based on evaluation findings.
* **Product Owner**: Reviews the evaluation findings and provides feedback on the product’s alignment with business objectives.

# **Sign-Off**

| **Signatory** | **Role** | **Signature** | **Date** |
| --- | --- | --- | --- |
| John Doe | Project Sponsor | [Signature] | 01/16/2025 |
| Jane Smith | Business Analyst | [Signature] | 01/16/2025 |
| Michael Johnson | Product Owner | [Signature] | 01/16/2025 |
| Emily Brown | Lead Developer | [Signature] | 01/16/2025 |
| Richard Lee | QA Lead | [Signature] | 01/16/2025 |
| Sarah Williams | Stakeholder (PMC) | [Signature] | 01/16/2025 |
| David Clark | IT Infrastructure Manager | [Signature] | 01/16/2025 |

Further versions will be updated after updated requirements or changes.