**Document 1: Definition of Done**

The Definition of Done outlines the criteria that must be met for a product increment or backlog item to be considered complete and potentially shippable. The DOD ensures that the team maintains a consistent level of quality and completeness in their work.

For the LMS, Lead Management software enhancement project, the Definition of Done ensures that each sprint delivers a functional, tested, and approved increment that aligns with Employees, Customers and IT requirements.

The best representation of the DoD is a checklist of activities that validate the value and quality of a user story. This checklist includes:

* Acceptance criteria – Ensuring the feature meets Client and Relationship Manager expectations.
* Quality criteria – Compliance with HR and IRDA policies, security, and performance standards.

For the LMS project, DoD is defined at multiple levels:

* User Story Level – Features meet functional and non-functional requirements.
* Sprint Level – All stories within a sprint are tested and reviewed.
* Release Level – The final product is approved by stakeholders and ready for deployment.

Checklist for DOD:

1. Produced code for presumed functionalities

* Code follows best practices and coding standards
* Code is modular, maintainable, and reusable

1. Assumptions of User Story met

* Requirements are documented and validated
* Business logic aligns with CRM (Client Relationship Management) needs

1. Project builds without errors

* Build process runs successfully
* Versioning and tagging of build artifacts are done

1. Unit tests written and passing

* Unit tests and integration tests pass
* Key functionalities work as expected

1. Project deployed on the test environment identical to production platform

* Test environment matches production settings
* Test data reflects real-world Client Management scenarios

1. Tests on devices/browsers listed in the project assumptions passed
2. Feature ok-ed by UX designer

* Design consistency across the platform is maintained
* User experience is validated

1. QA performed & issues resolved

* Functional, security, and performance testing completed
* All bugs and issues are fixed and verified

1. Feature is tested against acceptance criteria

* All business and functional requirements are satisfied
* Any deviations are documented and approved

1. Feature ok-ed by Product Owner

* Feature is reviewed and signed off by the Product Owner

1. Refactoring completed

* Code is optimized for performance and readability

1. Any configuration or build changes documented

* Any changes in configuration or build process are documented

1. Documentation updated

* Help guides and release notes are revised

1. Peer Code Review performed

* Code is reviewed, and feedback is incorporated

**Document 2- Product Vision**

|  |  |  |  |
| --- | --- | --- | --- |
| **Scrum project name** | **LMS, Lead Management System** | | |
| **Venue** | **Bengaluru, Karnataka** | | |
| **Date:** | **Start Time: 20-Mar-2025** | **End Time: 20-Feb-2025** | **Duration: 1 Year** |
| **Client** | **ICICI Prudential Life Insurance** | | |
| **Stakeholder List:** | **Project Stakeholders:** BA Project Manager Development Team Tech Expert Testing Team Operations  (Network, Training) UI Designer | **Business Stakeholders:** Project Manager Business Owner Business Sponsor Operations Team Subject Matter Expert | **3rd Party Stakeholders:** Auditors Legal Team Vendor Manufacturer |

Scrum Team:

|  |  |
| --- | --- |
| **Role** | **Team Member** |
| **Scrum Master:** |  |
| **Product Owner:** |  |
| **Scrum Developer 1:** |  |
| **Scrum Developer 2:** |  |
| **Scrum Developer 3:** |  |
| **Scrum Developer 4:** |  |
| **Scrum Developer 5:** |  |
| **UI Designer:** |  |
| **Tester:** |  |
| **Business Analyst:** | Ajas Asharaf |

|  |  |  |  |
| --- | --- | --- | --- |
| **Vision:** | Lead Management System (LMS) is to empower ICICI Prudential Life Insurance with a smart, efficient, and automated lead tracking and customer relationship platform. It aims to streamline sales processes, enhance customer engagement, and maximize lead conversions by providing real-time insights, automation, and seamless integration with existing systems.. | | |
| **Target Group** | **Needs** | **Product** | **Value** |
| Sales Representatives, Relationship Managers, Sales Managers, Team Leader’s, Marketing & Campaign Teams, Customer Support Teams, IT Team. | Sales & Lead Management Needs, Automated Follow-Ups – Reduce manual effort with scheduled reminders and AI-driven recommendations, Customer Engagement Needs, Faster Response Times – Enable real-time communication through automated notifications and chatbots, Personalized Customer Journey – Track interactions and offer tailored solutions based on customer profiles.Sales Performance & Reporting Needs **Real-Time Dashboards & Analytics** – Provide data-driven insights to improve conversion rates. **Business Efficiency &** Integration Needs Seamless Integration with CRM & Other Tools – Ensure smooth data flow across existing systems. | The **Lead Management System (LMS)** for **ICICI Prudential Life Insurance** is a **comprehensive digital platform** designed to optimize the sales process, enhance customer relationship management, and maximize lead conversions through automation and data-driven insights. | Value Proposition of the Lead Management System (LMS)  The Lead Management System (LMS) for ICICI Prudential Life Insurance delivers significant value by optimizing sales operations, enhancing customer engagement, and increasing lead conversion efficiency. |

**Document 3: User stories**

**User stories:** are simple descriptions of a feature or requirement from the user's perspective. They help explain how a system should work in a way that is easy to understand, making requirement gathering more straightforward.

**Business Value (BV)** refers to how important a feature (user story) is to the business. A higher BV means the feature has a greater impact.

**Complexity Points (CP)** represent the effort required by developers to implement the feature. They help estimate the time and difficulty involved, often rated using CP points (e.g., Poker cards).

**Acceptance Criteria** define the conditions that a software feature must meet to be considered complete and functional. While user stories describe what a user wants to do, acceptance criteria explain how the system should behave to fulfill that requirement. These criteria ensure the feature meets user expectations and works as intended.

**User story 1 - Employee Login**

|  |  |  |
| --- | --- | --- |
| **User Story: 1** | **Tasks: 1** | **Priority: High** |
| **Value statement:**  AS A USER,  I WANT TO LOGIN  SO THAT I CAN VIEW MY ASSIGNED LEADS | | |
| **BV:** 500 | | **CP:** 1 |
| **Acceptance Criteria:**  User can enter personal details like name, email, and password.  System validates the email format.  User receives a confirmation email upon successful registration. | | |

**User story 2 - Lead Assignment**

|  |  |  |
| --- | --- | --- |
| **User Story:** 2 | **Tasks:** 1 | **Priority: High** |
| **Value statement:**  AS A RELATIONSHIP MANAGER I WANT TO RECEIVE LEADS AUTOMATICALLY FROM VARIOUS SOURCES, SO THAT I CAN ENGAGE WITH POTENTIAL CUSTOMERS EFFICIENTLY. | | |
| **BV:500** | | **CP: 1** |
| **Acceptance Criteria:**  Leads can be captured from multiple sources (website, social media, referrals).  The system should store all leads in a structured database.  Duplicate lead entries should be identified and merged.. | | |

**User story 3 - Lead Prioritization**

|  |  |  |
| --- | --- | --- |
| **User Story**: 3 | **Tasks:** 1 | **Priority:** High |
| **Value statement:**  AS A SALES REPRESENTATIVE, I WANT TO SEE LEADS RANKED BASED ON THEIR POTENTIAL AND LIKELIHOOD TO CONVERT, SO THAT I CAN FOCUS ON HIGH-PRIORITY LEADS FIRST. | | |
| **BV:** 450 | | **CP:** 2 |
| **Acceptance Criteria:**  Leads should be scored based on engagement, demographics, and historical data.  High-priority leads should be highlighted on the dashboard.  Users should be able to filter and sort leads by priority. | | |

**User story 4 - FOLLOW-UP REMINDERS**

|  |  |  |
| --- | --- | --- |
| **User Story:** 4 | **Tasks**: 1 | **Priority:** High |
| **Value statement:**  AS A RELATIONSHIP MANAGER, I WANT TO RECEIVE AUTOMATED REMINDERS FOR FOLLOW-UPS WITH MY LEADS, SO THAT I NEVER MISS AN OPPORTUNITY TO ENGAGE WITH A CUSTOMER. | | |
| **BV:** 500 | | **CP:** 3 |
| **Acceptance Criteria:**  Sales reps should receive reminders based on follow-up schedules.  Notifications should be configurable (email, SMS, in-app alerts).  The system should log completed follow-ups for tracking. | | |

**User story 5 - Customer Profile & Interaction History**

|  |  |  |
| --- | --- | --- |
| **User Story:** 5 | **Tasks:** 1 | **Priority:** Medium |
| **Value statement:**  AS A RELATIONSHIP MANAGER, I WANT TO VIEW A DETAILED PROFILE OF EACH LEAD, INCLUDING THEIR INTERACTION HISTORY AND PREFERENCES, SO THAT I CAN PERSONALIZE MY PITCH. | | |
| **BV:** 350 | | **CP:** 3 |
| **Acceptance Criteria:**  Sales reps should view lead details, past interactions, and status.  Notes and comments should be added for each interaction.  History should be accessible in chronological order. | | |

**User story 6 –  Omnichannel Communication**

|  |  |  |
| --- | --- | --- |
| **User Story:** 6 | **Tasks:** 1 | **Priority:** High |
| **Value statement:**  AS A CUSTOMER, I WANT TO COMMUNICATE WITH THE SALES TEAM THROUGH MY PREFERRED CHANNEL (WHATSAPP, EMAIL, SMS, OR PHONE CALL), SO THAT I HAVE A SEAMLESS EXPERIENCE. | | |
| **BV:** 600 | | **CP:** 2 |
| **Acceptance Criteria:**  Customers should be able to contact sales reps via WhatsApp, email, SMS, or call.  Conversations should be logged under the lead’s profile.  Sales reps should be able to reply directly from the system. | | |

**User story 7 – Real-time Reporting & Analytics**

|  |  |  |
| --- | --- | --- |
| **User Story:** 7 | **Tasks:** 1 | **Priority:** Medium |
| **Value statement:**  AS A RELATIONSHIP MANAGER, I WANT TO ACCESS REAL-TIME DASHBOARDS THAT SHOW TEAM PERFORMANCE, LEAD CONVERSION RATES, AND REVENUE IMPACT, SO THAT I CAN MAKE DATA-DRIVEN DECISIONS. | | |
| **BV:** 450 | | **CP:** 3 |
| **Acceptance Criteria:**  Sales managers should access dashboards with lead performance, conversion rates, and team activity.  Reports should be exportable in Excel/PDF format.  Data should refresh in real-time or at scheduled intervals. | | |

**User story 8 – AI-Powered Lead Recommendations**

|  |  |  |
| --- | --- | --- |
| **User Story:** 8 | **Tasks:** 1 | **Priority:** Medium |
| **Value statement:**  AS A RELATIONSHIP MANAGER , I WANT THE SYSTEM TO SUGGEST THE BEST LEADS TO CONTACT NEXT BASED ON THEIR BEHAVIOR AND INTEREST LEVEL, SO THAT I CAN INCREASE MY CHANCES OF CLOSING DEALS. | | |
| **BV:** 400 | | **CP:** 3 |
| **Acceptance Criteria:**  The system should suggest leads based on past interactions and conversion likelihood.  Recommendations should update dynamically based on new data.  Users should have an option to accept or ignore recommendations.  . | | |

**User story 9 – Integration with Policy Management System**

|  |  |  |
| --- | --- | --- |
| **User Story:** 9 | **Tasks:** 1 | **Priority:** High |
| **Value statement:**  AS AN INSURANCE AGENT, I WANT THE LMS TO INTEGRATE WITH THE POLICY MANAGEMENT SYSTEM, SO THAT I CAN SEAMLESSLY TRACK THE PROGRESS OF LEADS FROM INQUIRY TO POLICY ISSUANCE. | | |
| **BV:** 550 | | **CP:** 2 |
| **Acceptance Criteria:**  Lead details should sync with the policy management system once converted.  Sales reps should track policy status within the LMS.  Policy updates should be reflected in real-time. | | |

**User story 10 – Mobile App Access**

|  |  |  |
| --- | --- | --- |
| **User Story: 10** | **Tasks: 1** | **Priority: High** |
| **Value statement:**  AS A COMPLIANCE OFFICER, I WANT THE SYSTEM TO ENSURE DATA SECURITY AND REGULATORY COMPLIANCE (SUCH AS GDPR AND IRDAI GUIDELINES), SO THAT CUSTOMER INFORMATION IS PROTECTED. | | |
| **BV:** 600 | | **CP:** 2 |
| **Acceptance Criteria:**  The system should follow data security regulations (GDPR, IRDAI).  Role-based access control should be implemented.  All transactions should be logged for audit purposes. | | |

**Document 4: Agile PO Experience**

The Product Owner (PO) plays a key role in shaping the LMS product by combining domain expertise, industry insights, and market needs to ensure the development of a competitive Customer relationship and lead management solution.

**Responsibilities of a PO in the LMS Project:**

* **Market Analysis**

The **market analysis** highlights that an **AI-driven, mobile-first, and integrated LMS** will significantly enhance lead conversion rates and customer engagement for ICICI Prudential. With **growing competition and evolving customer expectations**, the LMS should focus on **automation, omnichannel communication, and predictive analytics** to stay ahead in the market.

* **Enterprise Analysis**

The **LMS project** will provide a **strategic advantage** by **enhancing lead conversion, improving customer relationships, and boosting agent productivity**. With **AI-driven automation, omnichannel integration, and real-time analytics**, ICICI Prudential will **strengthen its market position** and drive business growth.

.

* **Product Vision and Roadmap**

To develop a **smart, AI-powered, and customer-centric** **Lead Management System (LMS)** that **enhances lead tracking, optimizes agent performance, and boosts sales conversion rates** through seamless automation, omnichannel integration, and data-driven decision-making.

The LMS will **revolutionize ICICI Prudential’s sales process** by **leveraging AI, automation, and omnichannel engagement** to ensure **faster conversions, improved customer interactions, and enhanced agent performance.**

* **Managing Product Features in LMS**

Effectively managing product features in the LMS ensures that the system remains **user-friendly, scalable, and aligned with business objectives.** The following approach is used to manage product features throughout the development lifecycle:

| **Feature** | **Description** | **Priority** | | **Status** |
| --- | --- | --- | --- | --- |
| **Lead Capture** | Automate lead collection from multiple sources (web, call, social media) | High | | Implemented |
| **Lead Scoring & Segmentation** | AI-driven ranking of leads based on probability of conversion | High | | Development |
| **Automated Lead Assignment** | Assign leads to agents based on predefined rules | High | | Implemented |
| **Omnichannel Communication** | Engage leads via WhatsApp, Email, SMS & Calls | Medium | | In Progress |
| **Analytics & Reporting** | Custom dashboards for sales tracking & performance insights | High | | Implemented |
| **Mobile App for Agents** | Access leads & update statuses via a mobile interface | Medium | | Backlog |
| **Chatbot for Lead Engagement** | AI-powered chatbot for quick lead responses | Medium | | Future Release |
| **Gamification for Sales Teams** | Leaderboards, incentives, and challenges for motivation | Low | | Future Release |
|  | | |

* **Managing Product Backlog**
* Regular backlog refinement before sprint planning to align with IRDA priorities.
* Categorization of backlog items based on impact (short-term vs. long-term).
* Removing outdated or irrelevant features based on user feedback.
* Strict adherence to the roadmap to prevent scope creep.
* Prioritization based on business value & customer impact.
* **Managing Overall Iteration Progress**

The Iteration manager ensures that each sprint delivers tangible improvements to LMS. They coordinate cross-functional teams, track key milestones, and ensure the development aligns with the overall Lead Management transformation strategy.

* Through this project, I have gained hands-on experience in managing Agile Sprint Meetings, ensuring smooth collaboration between teams and effective product delivery.
* **Sprint Planning Meeting**
* Identify key Lead Management, CRM feature requirements for the sprint.
* Determine feasibility and expected delivery timelines within the Scrum framework.
* Align sprint goals with business priorities (e.g., Lead, automation, compliance tracking).
* **Daily Scrum Meeting**

Helps maintain transparency and track sprint progress effectively.

A 10-minute stand-up where each team member shares updates:

* What they worked on yesterday.
* What they plan to do today.
* Any blockers preventing progress.
* **Sprint Review Meeting**
* Demonstrate the completed LMS, CRM features (e.g., Lead Management, automation, Portfolio management, or analytics dashboards) to stakeholders.
* Gather feedback from Rm’s, Clients, and Internal users.
* Based on feedback, refine product backlog items for enhanced functionality and usability.
* **Sprint Retrospective Meeting**
* Review sprint performance using a burndown chart.
* Discuss what worked well and areas for improvement.
* Identify process optimizations for faster and more efficient product development.
* **Backlog Refinement Meeting**
* Analyze the remaining product backlog with the Scrum team.
* Re-prioritize backlog items based on customer needs, compliance updates, or business goals.
* Plan new sprints focusing on critical LMS, CRM functionalities like performance management automation or AI-based candidate screening.
* Each feature in the LMS follows a structured user story format to ensure clarity and alignment with Client Management needs.

**User Story**

User Story Structure:

As a [type of user],

I want [an action or feature],

So that [benefit or value].

* A list of specific tasks required to complete each user story. These are individual steps or actions needed to implement the feature.

**Tasks**

Tasks: **Lead Assignment**

Develop a LMS TO RECEIVE LEADS AUTOMATICALLY FROM VARIOUS SOURCES

* Indicates the importance or urgency of the user story in the overall project. Priorities are categorized as High, Medium, or Low based on business impact and user needs.

**Priority**

Examples in LMS:

High Priority: Critical for the next release

Medium Priority: Important but can wait for the next iteration

Low Priority: Nice-to-have, can be considered later

* A set of conditions or criteria that must be met for the user story to be considered complete. Ensures the feature meets HR team expectations.

**Acceptance Criteria**

Example for **Lead Assignment**

Leads can be captured from multiple sources (website, social media, referrals).

The system should store all leads in a structured database.

Duplicate lead entries should be identified and merged..

* Defines how the user story benefits the business . Helps in prioritizing feature development.

**Business Value**

Examples in LMS:

High Business Value: Increases efficiency & compliance (e.g., Automated Leads)

Medium Business Value: Adds a useful feature but not critical (e.g., As a Relationship Manager, I want to tag leads with custom labels so that I can categorize them for better follow-up.

Low Business Value: Minimal impact but enhances user experience (e.g.As a user, I want to be able to change the theme of the LMS interface so that I can personalize my experience.)

* An estimation of the effort required to implement the user story, helping with capacity planning and sprint commitments.

**Complexity Points**

Simple (1-3 points): UI enhancements, minor API fixes.

Moderate (3-5 points): Workflow automation, integration with third-party systems.

Complex (5-8 points): Custom AI-driven Leads, compliance automation modules.

* In Scrum, the Product Owner acts as the key liaison between Customers, business stakeholders, and the development team. They gather requirements from Agents, Rm’s and employees to ensure the platform meets business needs.

The Product Owner defines the vision for the **Lead Management System (LMS)** by ensuring that the product aligns with business goals and delivers value to users.

**Document 5: Product and sprint backlog and product and sprint burndown charts**

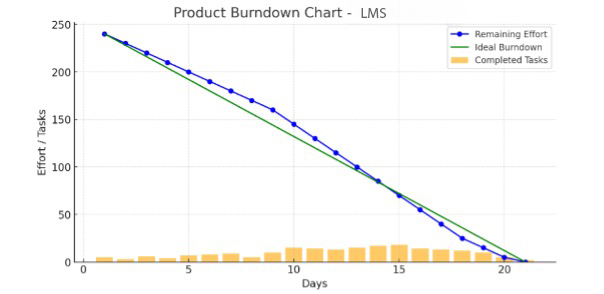
The **Product Backlog** is a prioritized list of features, enhancements, and fixes that the LMS development team will work on. It is continuously refined and updated based on business needs and user feedback.

**Product Backlog Table for LMS**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User story ID** | **User story** | **Tasks** | **Priority** | **BV** | **CP** | **Sprint** |
| US01 | As a RM, I want to add a new lead to the system so that I can track potential customers. | - Create lead entry form- Validate input fields- Store lead in the database | High | 500 | 3 | 1 |
| US02 | As a RM, I want to update a lead’s status so that I can keep track of progress.  . | Implement status dropdown- Save updates to the database- Trigger notifications | High | 400 | 2 | 2 |
| US03 | As a RM, I want to assign leads to agents based on predefined rules so that workload is balanced. | Define assignment logic- Implement auto-assign function- Notify assigned agents | High | 600 | 4 | 1 |
| US04 | As a RM, I want to receive follow-up reminders so that I don’t miss potential sales. | Develop notification service- Integrate with email/SMS- Configure reminder settings | Medium | 350 | 2 | 2 |
| US05 | As a RM, I want to view analytics on lead performance so that I can optimize sales strategies. | Design dashboard UI- Fetch & process lead data- Display reports & graphs | High | 500 | 3 | 2 |
| US06 | As an Admin, I want to manage user roles and permissions, so that I can control access to sensitive data. | Design role-based access control system  Implement admin dashboard for user management  Audit user activity logs | High | 550 | 3 | 3 |
| US07 | . As a system admin, I want to restrict access to sensitive lead data so that only authorized users can view it. | Implement role-based access- Configure permission settings- Test access control | Medium | 300 | 2 | 3 |
| US08 | As a RM, I want an AI-based lead scoring system so that I can focus on high-potential leads. | Define scoring algorithm- Implement AI model- Integrate with lead database | High | 600 | 4 | 4 |
| US09 | As a customer, I want to submit my details through a web form so that I can be contacted by sales. | Create customer web form- Validate inputs- Store data in the LMS | High | 700 | 5 | 4 |
| US10 | As a RM, I want to access LMS on mobile so that I can manage leads on the go. | Develop mobile app UI- Integrate with LMS backend- Optimize for performance | High | 750 | 5 | 5 |

**Product Burndown Chart**

A product burn down chart shows how much work remains for the entire project, whereas a sprint burndown chart shows how much work remains in a specific iteration. A product burn down chart collects a larger amount of data.



**Interpretation:**  
Remaining Effort (Blue Line & Dots) :

* Shows the actual remaining effort/tasks throughout the project timeline.
* The trend indicates how quickly the team is progressing

Ideal Burndown (Green Line):

* Represents the expected pace of progress for smooth project completion.
* If the actual effort stays close to this line, the project is on track.

Completed Tasks (Orange Bars):

* Indicates the number of tasks completed on each day.
* Higher bars suggest productive days, while shorter bars may indicate bottlenecks.

In this chart;

* The blue line follows the green line closely, meaning the project is progressing at a steady pace.
* There are some deviations where the actual effort is above the ideal line, showing slight delays in some phases.
* Completed tasks are consistent, but some fluctuations suggest varying task complexity.
* The project reaches completion within the expected timeline, indicating effective task management.

**Document 6: Sprint meetings**

**Project: Lead Management System (LMS)**

• **Sprint Number:** [Sprint X]

• **Date:** [DD/MM/YYYY]

• **Time:** [HH:MM AM/PM]

• **Location:** [Bengaluru / Virtual Link]

• **Prepared By:** [Ajas Asharaf]

• **Attendees:**

• Product Owner: [Name]

• Scrum Master: [Name]

• Development Team: [Names]

• QA Team: [Names]

**Agenda Topics**

1. **Welcome & Introduction** (5 min)

• Quick check-in and meeting purpose

• Confirm attendees

2. **Sprint Goal Definition** (10 min)

• Product Owner presents the sprint objective

• Align team expectations with business needs

3. **Product Backlog Review** (15 min)

• Product Owner reviews prioritized backlog items

• Clarify user stories, acceptance criteria, and dependencies

4. **Sprint Backlog Creation & Task Breakdown** (25 min)

• Team selects user stories for the sprint

• Break down user stories into tasks

• Assign story points (effort estimation)

5. **Capacity & Availability Check** (10 min)

• Discuss team availability and workload distribution

6. **Definition of Done (DoD) Confirmation** (10 min)

• Verify criteria for task completion

• Ensure alignment on quality standards

7. **Risk & Dependency Identification** (10 min)

• Identify and discuss potential roadblocks

• Plan for cross-team dependencies

8. **Sprint Commitment & Wrap-Up** (5 min)

• Confirm final backlog selection

• Assign responsibilities and action items

• Schedule daily stand-ups and sprint review

**ther Information for Sprint Planning Meeting**

• **Observers:**

• [List any non-participating attendees, such as executives, clients, or stakeholders observing the meeting]

• **Resources:**

• **Project Management Tool:** Jira / Trello / Azure DevOps

• **Documentation & References:** Confluence / Google Drive / SharePoint

• **Development Tools:** [List relevant tools like IDEs, databases, etc.]

• **Testing & CI/CD Tools:** [Jenkins, Selenium, etc.]

• **Special Notes:**

• Any additional comments, reminders, or key takeaways

• Action items for follow-up

• Next meeting schedule (if applicable)

**Sprint review meeting**

**Date:** [DD/MM/YYYY]

• **Time:** [HH:MM AM/PM]

• **Location:** [Meeting Room / Virtual Link]

• **Prepared By:** [Your Name]

• **Attendees:**

• Product Owner: [Name]

• Scrum Master: [Name]

• Development Team: [Names]

• QA Team: [Names]

• Stakeholders (if applicable): [Names]

**Sprint Review Meeting Agenda**

1. **Sprint Status Update**

• Overview of completed user stories and tasks

• Sprint goal achievement status

2. **Things to Demo**

• Live demonstration of completed features

• Walkthrough of functionality and UI changes

• Showcase improvements and bug fixes

3. **Quick Updates**

• Challenges faced and how they were addressed

• Team feedback and retrospective insights

• Any blockers that need resolution

4. **What’s Next**

• Discuss upcoming sprint backlog items

• Prioritization of features for the next sprint

• Dependencies and resource requirements

**Sprint Retrospective Meeting Details**

• **Date:** [DD/MM/YYYY]

• **Time:** [HH:MM AM/PM]

• **Location:** [Meeting Room / Virtual Link]

• **Prepared By:** [Your Name]

• **Attendees:**

• Scrum Master: [Name]

• Product Owner: [Name]

• Development Team: [Names]

• QA Team: [Names]

**Sprint Retrospective Meeting Agenda**

1. **What Went Well**

• Positive aspects of the sprint

• Achievements and successful implementations

• Team collaboration and process improvements

2. **What Didn’t Go Well**

• Challenges and roadblocks faced

• Bottlenecks in the development process

• Areas where improvement is needed

3. **Questions & Discussions**

• Open floor for team feedback

• Suggestions for process optimization

• Identifying action items for improvement

4. **References & Action Items**

• Documented learnings from the sprint

• Agreed-upon changes for the next sprint

• Assigning responsibilities for process improvements

**Daily Stand-up Meeting**

• **Date:** [DD/MM/YYYY]

• **Time:** [HH:MM AM/PM]

• **Location:** [Meeting Room / Virtual Link]

• **Prepared By:** [Your Name]

• **Attendees:**

• Scrum Master: [Name]

• Product Owner: [Name]

• Development Team: [Names]

• QA Team: [Names]

**Agenda**

1. **What did I do yesterday?**

• Each team member shares completed tasks.

2. **What will I do today?**

• Updates on planned tasks for the day.

3. **Are there any blockers?**

• Identifying issues that need resolution.

4. **Quick Team Discussion (if needed)**

• Addressing dependencies or clarifications.

**Daily Stand-up Meeting Report**

**Week “X” (from 01-03-2025 to 07-03-2025)**

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| |  | **Monday (01-Mar)** | **Tuesday (02-Mar)** | **Wednesday (03-Mar)** | **Thursday (04-Mar)** | **Friday (05-Mar)** | **Saturday (06-Mar)** | **Sunday (07-Mar)** | | --- | --- | --- | --- | --- | --- | --- | --- | | **What did you do yesterday?** |  |  |  |  |  |  |  | | **Developer 1** | Completed UI design for the dashboard | Fixed UI bugs | Integrated API for login | Worked on error handling | Started implementing notifications | Finalized dashboard UI | Weekly code review | | **Developer 2** | Backend setup for user authentication | Wrote unit tests for authentication | Fixed API response issues | Optimized database queries | Implemented role-based access | Tested authentication module | Weekly backend review | | **Developer 3** | Designed database schema | Created initial API endpoints | Integrated database with backend | Debugged API responses | Started report generation module | Finalized API security measures | Prepared API documentation | | **What will you do today?** |  |  |  |  |  |  |  | | **Developer 1** | Work on profile page | Enhance UI for settings | Integrate new icons and themes | Test UI responsiveness | Work on mobile view optimization | Fix reported UI issues | Plan for next week’s UI tasks | | **Developer 2** | Fix authentication bugs | Implement JWT token refresh | Code review for authentication module | Work on session management | Conduct security testing | Deploy authentication module | Plan for next sprint | | **Developer 3** | Develop report export feature | Fix API pagination issues | Optimize report query execution | Test report generation speed | Improve database indexing | Conduct final API testing | Document all API changes | | **What (if any) is blocking your progress?** |  |  |  |  |  |  |  | | **Developer 1** | No blockers | API delay in data fetching | Need access to new design assets | UI performance issues | Need feedback from UX team | No blockers | No blockers | | **Developer 2** | No blockers | Dependency on DevOps team | Need access to staging database | Authentication token issue | Need logs from production | No blockers | No blockers | | **Developer 3** | API response time slow | Need test data for reports | Database query performance issues | Need approval for new API endpoint | Awaiting final security audit | No blockers | No blockers | |