**PART -2 DELIVERABLES**

1. **Definition of Done**

The Definition of Done (DoD) ensures that all user stories, sprints, and releases meet the required functionality, quality, and compliance standards before being considered complete. This checklist ensures the team delivers high-quality, production-ready features while maintaining regulatory compliance.

**Definition of Done Checklist:**

* User Story Level DoD

A user story is considered "Done" when:

Development & Code Quality:

* Code for the functionality is written and adheres to coding standards.
* Code is reviewed and approved through Peer Code Review.
* The application builds without errors.
* Unit tests are written with a minimum 80% test coverage and all tests pass.

Testing & Validation:

* The feature is tested against Acceptance Criteria and passes.
* Functional, integration, and regression tests are performed and successful.
* The feature is tested on all required browsers/devices.
* Security and compliance **checks** (e.g., HIPAA, data encryption) are completed.
* The feature is okayed by QA with all defects fixed.

Approval & Documentation:

* The feature is approved by the Product Owner
* Any necessary configuration or build changes are documented.
* User documentation or help guides are updated, if needed.
* Sprint Level DoD

A sprint is considered "Done" when:

* All user stories meet the User Story Level DoD.
* The sprint backlog items are demoed to stakeholders and accepted.
* Integration tests for sprint functionalities are successful.
* No high-priority or critical defects remain unresolved.
* Sprint review and retrospective are completed.
* Release notes are documented.
* Release Level DoD:

A release is considered "Done" when:

* All sprint increments have been tested and validated.
* End-to-end testing across the entire system is successful.
* Performance, security, and compliance audits are completed.
* Data privacy & HIPAA compliance requirements are met.
* Rollback & deployment strategies are documented and validated.
* Production deployment is successful with zero downtime.
* Post-release monitoring and validation are set up
* Final sign-off from the Product Owner and key stakeholders is obtained.

This Definition of Done (DoD) ensures that each feature, sprint, and release meets high standards of quality, security, and compliance in the US healthcare domain. Following this checklist helps deliver a robust, secure, and compliant Patient Healthcare Management Application.

1. **Product Vision Document**



|  |
| --- |
| Scrum Project Name: **Insure Care** |
| Venue:  Date: Start Time: End Time: Duration:  Client: IT Solutions  **Stakeholder List:**  1.Business Owner  2**.** Administrator  3. Healthcare Providers  4. Insurance Companies/Payers  5. IT Support & Help desk |

|  |
| --- |
| **Scrum Team**  Scrum Master: Satya Rathnakar  Scrum Developer 1: Linesh Vegad  Scrum Developer 2: Yogender  Scrum Developer 3: Gowri  Scrum Developer 4: A.Lakshmikala  Scrum Developer 5: Madhuri  Scrum Developer 6: Varun  Scrum Developer 7: Rakesh  Scrum Developer 8: Rajesh |

**Vision:**

The vision for the US patient healthcare management application is to revolutionize the way healthcare providers interact with patient insurance and claims information. The overarching goal is to create a seamless, efficient, and user-friendly platform that empowers healthcare providers to access comprehensive patient insurance details, benefits, authorization information, and claims data in real-time. This will enhance the quality of patient care, reduce administrative burdens, and streamline the healthcare delivery process.

**Target Group:**

**Market Segment:** The product addresses the healthcare technology market, specifically focusing on healthcare providers such as hospitals, clinics, and individual practitioners who need to manage patient insurance and claims information efficiently.

**Target Users and Customers:**

* Primary Users: Healthcare providers including doctors, nurses, administrative staff, and billing specialists.
* Secondary Users: Insurance companies and third-party administrators who may integrate with the platform for seamless data exchange.
* Customers: Healthcare organizations, hospitals, and clinics that will purchase and implement the software for their staff.

**Needs:**

**Problem Solved:** The product addresses the inefficiencies and complexities in managing patient insurance plans, benefits, authorizations, and claims. Healthcare providers often face challenges in accessing timely and accurate information, leading to delays in care, administrative overhead, and potential errors in billing and claims processing.

**Benefits Provided:** The application provides a centralized, real-time platform that simplifies the management of patient insurance information. It reduces administrative workload, minimizes errors, accelerates the authorization process, and ensures that healthcare providers can focus more on patient care rather than paperwork.

**Product:**

**What is the Product?**

The product is a cloud-based healthcare management application designed for healthcare providers to access and manage patient insurance plans, benefits, authorization information, and claims data.

**What Makes it Desirable and Special?**

* Real-Time Access: Providers can access up-to-date insurance and claims information in real-time.
* User-Friendly Interface: Intuitive design that simplifies complex processes.
* Integration Capabilities: Seamless integration with existing Electronic Health Records (EHR) systems and insurance databases.
* Agile Development: Continuous updates and improvements based on user feedback and changing healthcare regulations.

**Feasability:**

Yes, the product is feasible to develop. With the advancement of cloud technologies, APIs for integration with EHRs and insurance systems, and a strong understanding of healthcare workflows, the development team can create a robust and scalable solution. The agile model ensures that the product can evolve based on user needs and regulatory changes.

**Value:**

**How is the Product Going to Benefit the Company?**

**Revenue Growth:** By offering a valuable tool to healthcare providers, the company can generate significant revenue through software licensing, subscription models, and potential partnerships with insurance companies.

**Market Leadership:** Establishing a strong presence in the healthcare technology market can position the company as a leader in patient management solutions.

**Operational Efficiency:** The product can reduce operational costs for healthcare providers, making it an attractive investment for them.

**Business Goals:**

**Customer Acquisition:** Acquire a substantial number of healthcare providers and organizations as customers within the first two years.

**User satisfaction**: Achieve high user satisfaction and retention rates through continuous improvement and responsive customer support.

**Regulatory Compliance:** Ensure the product complies with all relevant healthcare regulations, including HIPAA and HITECH.

**Business Model:**

**Subscription-Based:** Offer monthly or annual subscription plans for healthcare providers based on the number of users or the size of the practice.

**Enterprise Licensing:** Provide enterprise-level licensing for larger healthcare organizations with customized features and support.

**Partnerships:** Collaborate with insurance companies and EHR providers to offer integrated solutions, potentially sharing revenue or charging integration fees.

By focusing on these elements, the company can create a valuable, sustainable, and impactful product that meets the needs of healthcare providers and improves the overall healthcare delivery process.

1. **User Stories:**

|  |  |  |  |
| --- | --- | --- | --- |
| User Story No: 1 | Tasks: Develop a login page with MFA | | Priority: High |
| As a healthcare provider I want to securely log in to the application so that I can access patient information securely. | | | |
| BV: 500 | | CP: 2 | |
| Acceptance Criteria:   1. Providers can log in using their credentials. 2. Multi-factor authentication is implemented. 3. Unauthorized access is prevented. | | | |
| User Story No: 2 | | Tasks: Create Dashboard to display patient summaries | | Priority: High |
| As a provider, I want to view a summary of my patients, so that I can quickly access their information. | | | | |
| BV: 500 | | | CP: 3 | |
| Acceptance Criteria:   1. Patient names, IDs, and recent visits are displayed. 2. Clicking on a patient opens their detailed profile. 3. The dashboard is updated in real-time. | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 3** | Tasks: Implement a search feature to find patients by name or ID. | | Priority: Medium |
| As a provider, I want to search for a specific patient, so that I can quickly retrieve their information. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Search bar is available on the dashboard. 2. Results are displayed instantly as the user types. 3. No results found message is displayed if no match exists. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 4** | Tasks: Display patient insurance plan details. | | Priority: High |
| As a provider, I want to view a patient’s insurance plan details, so that I can verify coverage. | | | |
| BV: 600 | | CP: 3 | |
| Acceptance Criteria:   1. Insurance provider name, plan type, and policy number are displayed. 2. Coverage start and end dates are visible. 3. A link to download the insurance card is available. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 5** | Tasks: Show patient benefits information. | | Priority: Medium |
| As a provider, I want to view a patient’s benefits information, so that I can understand what services are covered. | | | |
| BV: 500 | | CP: 2 | |
| Acceptance Criteria:   1. Deductibles, co-pays, and out-of-pocket maximums are displayed. 2. Covered services (e.g., lab tests, surgeries) are listed. 3. Exclusions and limitations are clearly stated. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 6** | Tasks: Display prior authorization information. | | Priority: High |
| As a provider, I want to view a patient’s prior authorization details, so that I can ensure services are approved. | | | |
| BV: 700 | | CP: 4 | |
| Acceptance Criteria:   1. Authorization status (approved, pending, denied) is displayed. 2. Approved services and their expiration dates are listed. 3. A link to request new authorizations is available. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 7** | Tasks:  Show patient claims history. | | Priority: Medium |
| As a provider, I want to view a patient’s claims history, so that I can track past services and payments. | | | |
| BV: 500 | | CP: 3 | |
| Acceptance Criteria:   1. Claims are listed by date of service. 2. Claim status (paid, denied, pending) is displayed. 3. Details include service codes, amounts billed, and amounts paid. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 8** | Tasks:  Allow providers to submit new authorization requests. | | Priority: High |
| As a provider, I want to submit new authorization requests, so that I can get approval for necessary services. | | | |
| BV: 800 | | CP: 5 | |
| Acceptance Criteria:   1. A form is available to enter service details. 2. Supporting documents can be uploaded. 3. Confirmation of submission is displayed. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 9** | Tasks:  Notify providers of authorization status updates. | | Priority: Medium |
| As a provider, I want to receive notifications about authorization status updates, so that I can stay informed without manually checking. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Notifications are sent via email and in-app alerts. 2. Notifications include the authorization ID and status. 3. Providers can click to view detailed updates. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 10** | Tasks:  Display patient demographic information. | | Priority: Low |
| As a provider, I want to view a patient’s demographic information, so that I can verify their identity. | | | |
| BV: 300 | | CP: 1 | |
| Acceptance Criteria:   1. Name, date of birth, gender, and contact information are displayed. 2. Information is read-only to prevent unauthorized edits. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 11** | Tasks:  Allow providers to download patient reports. | | Priority: Medium |
| As a provider, I want to download patient reports, so that I can share them with other healthcare professionals. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Reports are available in PDF and CSV formats. 2. Reports include insurance, benefits, and claims information. 3. Download links are clearly visible on the patient profile. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 12** | Tasks:  Implement role-based access control. | | Priority: High |
| As an administrator, I want to control access based on user roles, so that only authorized personnel can view sensitive information. | | | |
| BV: 600 | | CP: 4 | |
| Acceptance Criteria:   1. Providers can only view their assigned patients. 2. Administrators can manage user roles and permissions. 3. Unauthorized access attempts are logged and reported. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 13** | Tasks:  Add a feature to track claim status in real-time. | | Priority: High |
| As a provider, I want to track the status of claims in real-time, so that I can monitor payments and denials. | | | |
| BV: 500 | | CP: 3 | |
| Acceptance Criteria:   1. Claim status is updated automatically. 2. Providers can filter claims by status (paid, denied, pending). 3. Notifications are sent for significant status changes. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 14** | Tasks:  Create a patient profile editing feature. | | Priority: Low |
| As a provider, I want to update patient demographic information, so that I can ensure records are accurate. | | | |
| BV: 300 | | CP: 1 | |
| Acceptance Criteria:   1. Providers can edit fields like address and phone number. 2. Changes are logged for audit purposes. 3. Sensitive fields (e.g., SSN) are read-only. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 15** | Tasks:  Add a feature to view patient appointment history. | | Priority: Medium |
| As a provider, I want to view a patient’s appointment history, so that I can track their visits and treatments. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Appointments are listed by date and time. 2. Details include the provider seen and services rendered. 3. Providers can filter by date range. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 16** | Tasks:  Allow providers to send messages to patients. | | Priority: Medium |
| As a provider, I want to send messages to patients, so that I can communicate important updates. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. A messaging interface is available within the application. 2. Messages are encrypted for security. 3. Providers can track message delivery status. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 17** | Tasks:  Add a feature to view patient lab results. | | Priority: High |
| As a provider, I want to view a patient’s lab results, so that I can make informed treatment decisions. | | | |
| BV: 600 | | CP: 3 | |
| Acceptance Criteria:   1. Lab results are displayed in a clear, readable format. 2. Results are categorized by test type and date. 3. Providers can download or print results. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 18** | Tasks:   Implement a feature to track patient medications. | | Priority: Medium |
| As a provider, I want to view a patient’s current medications, so that I can avoid prescribing conflicts. | | | |
| BV: 500 | | CP: 2 | |
| Acceptance Criteria:   1. Medications are listed with dosages and frequencies. 2. Providers can add or remove medications. 3. Alerts are shown for potential drug interactions. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 19** | Tasks:   Add a feature to view patient allergies. | | Priority: Medium |
| As a provider, I want to view a patient’s allergies, so that I can avoid prescribing harmful medications. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Allergies are listed with severity levels. 2. Providers can add or remove allergies. 3. Alerts are shown for potential allergic reactions. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 20** | Tasks:   Create a feature to track patient referrals. | | Priority: Low |
| As a provider, I want to track patient referrals, so that I can ensure they receive necessary care. | | | |
| BV: 300 | | CP: 1 | |
| Acceptance Criteria:   1. Referrals are listed with specialist names and dates. 2. Providers can add new referrals. 3. Referral status (completed, pending) is displayed. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 21** | Tasks:   Add a feature to view patient immunization records. | | Priority: Medium |
| As a provider, I want to view a patient’s immunization records, so that I can ensure they are up-to-date. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Immunizations are listed with dates and types. 2. Providers can add new immunizations. 3. Alerts are shown for upcoming or missed immunizations. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 22** | Tasks:    Implement a feature to track patient vitals. | | Priority: Medium |
| As a provider, I want to track a patient’s vitals over time, so that I can monitor their health trends. | | | |
| BV: 500 | | CP: 3 | |
| Acceptance Criteria:   1. Vitals (e.g., blood pressure, weight) are displayed in graphs. 2. Providers can add new vital readings. 3. Alerts are shown for abnormal values. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 23** | Tasks:   Add a feature to view patient care plans. | | Priority: High |
| As a provider, I want to view a patient’s care plan, so that I can follow their treatment goals. | | | |
| BV: 600 | | CP: 3 | |
| Acceptance Criteria:   1. Care plans are displayed with goals and milestones. 2. Providers can update care plans. 3. Progress toward goals is tracked and displayed. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 24** | Tasks:   Create a feature to track patient progress notes. | | Priority: High |
| As a provider, I want to add and view progress notes for a patient, so that I can document their treatment journey. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Progress notes are displayed in chronological order. 2. Providers can add new notes. 3. Notes are searchable by keywords. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 25** | Tasks:   Add a feature to view patient discharge summaries. | | Priority: Low |
| As a provider, I want to view a patient’s discharge summaries, so that I can understand their post-hospitalization care. | | | |
| BV: 300 | | CP: 1 | |
| Acceptance Criteria:   1. Discharge summaries are displayed with dates and details. 2. Providers can download or print summaries. 3. Summaries include follow-up instructions. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 26** | Tasks:   Add a feature to view patient billing statements. | | Priority: Medium |
| As a provider I want to view a patient’s billing statements so that I can address financial concerns. | | | |
| BV: 400 | | CP: 2 | |
| Acceptance Criteria:   1. Billing statements are displayed with due dates and amounts. 2. Providers can generate new statements. 3. Payment history is visible. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 27** | Tasks:   Implement a feature to view patient appointment history. | | Priority: Medium |
| As a provider I want to view a patient’s appointment history so that I can track their engagement with care. | | | |
| BV: 400 | | CP: 3 | |
| Acceptance Criteria:   1. Appointments are listed with dates, times, and providers. 2. Providers can add new appointments. 3. Missed appointments are highlighted. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 28** | Tasks:   Implement a feature to view patient appointment history. | | Priority: Medium |
| As a provider I want to view a patient’s appointment history so that I can track their engagement with care. | | | |
| BV: 400 | | CP: 3 | |
| Acceptance Criteria:   1. Appointments are listed with dates, times, and providers. 2. Providers can add new appointments. 3. Missed appointments are highlighted. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 29** | Tasks:   Create a feature to view patient educational materials. | | Priority: Low |
| As a provider I want to view educational materials assigned to a patient so that I can reinforce their understanding. | | | |
| BV: 300 | | CP: 2 | |
| Acceptance Criteria:   1. Materials are listed with titles and descriptions. 2. Providers can assign new materials. 3. Patient progress is tracked. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **User Story No: 30** | Tasks:   Add a feature to view patient support group information. | | Priority: Low |
| As a provider I want to view support group information for a patient so that I can recommend additional resources. | | | |
| BV: 300 | | CP: 2 | |
| Acceptance Criteria:   1. Support groups are listed with details and meeting times. 2. Providers can add new groups. 3. Patient attendance is tracked. | | | |

1. **Agile PO Experience**

An Agile Product Owner (PO) for a US Patient Healthcare Management Application plays a crucial role in bridging the gap between business needs, regulatory requirements, and technology teams. The role requires a deep understanding of healthcare insurance workflows, compliance (e.g., HIPAA, ACA), and Agile methodologies to ensure the delivery of a secure, user-friendly, and efficient platform for healthcare providers.

**Key Responsibilities:**

Product Vision & Roadmap:

* Define and communicate the product vision for the patient healthcare management application.
* Align product goals with business objectives, regulatorycompliance, anduser needs (providers, insurers, and patients).
* Develop a strategic roadmap ensuring incremental delivery of value through Agile iterations.

Stakeholder Collaboration:

* Work closely with healthcare providers, payers (insurance companies), compliance teams, and DevOps engineers to gather requirements.
* Conduct user interviews and feedback sessions to refine features like insurance plan retrieval, benefits management, authorization tracking, and claims history.
* Collaborate with Scrum Master, Developers, UX Designers, and QA teams to ensure the feasibility of requirements.

Agile Execution & Iteration Planning

* Participate in Sprint Planning, ensuring the highest-priority features are addressed in each iteration.
* Work with the Scrum Team to clarify business logic, answer queries, and adjust priorities when needed.
* Accept or reject completed user stories based on acceptance criteria during Sprint Reviews.
* Incorporate feedback from retrospectives and user testing to enhance future iterations.

Compliance & Security Considerations:

* Ensure adherence to HIPAA regulations for patient data privacy and security.
* Work with legal and compliance teams to meet CMS, ACA, and payer integration standards.
* Guide development teams in implementing secure APIs, authentication**,** and encryption for data exchanges with insurance providers.

1. **Sprint Meetings**

**Meeting Type 1: Sprint Planning meeting**

|  |  |
| --- | --- |
| Date | March, 22nd , 2025 |
| Time | 10:00 AM - 12:00 PM |
| Location | Virtual (Zoom) |
| Prepared By | Yogesh |
| Attendees | Alice Johnson (Lead Developer)  Bob Brown (QA Engineer)  Emily Davis (UI/UX Designer)  Michael Wilson (Backend Developer)  John Doe (Scrum Master) |

**Agenda Topics**

|  |  |  |
| --- | --- | --- |
| **Topic** | **Presenter** | **Time Alloted** |
| Review of previous sprint outcomes | John Doe | 15 mins |
| Prioritization of backlog items | Jane Smith | 20 mins |
| Discussion on patient insurance plan module | Alice Johnson | 25 mins |
| Claims information integration | Michael Wilson | 20 mins |
| QA testing strategy for new features | Bob Brown | 15 mins |
| UI/UX updates for provider portal | Emily Davis | 15 mins |
| Sprint goal definition | John Doe | 10 mins |

**Other Information**

Observers:

* Sarah Lee (Compliance Officer)
* Yogesh (Business Analyst)

Resources:

* Jira board (updated backlog)
* Confluence (documentation)
* Figma (UI/UX designs)

**Special Notes**:

* Ensure HIPAA compliance is maintained for all new features.
* Focus on improving API response time for claims information retrieval.
* QA team to prioritize testing for insurance plan details due to high user impact.
* UI/UX team to finalize designs for provider portal by mid-sprint.

**Meeting Type 2: Sprint review meeting**

|  |  |
| --- | --- |
| Date | March, 22nd , 2025 |
| Time | 10:00 AM - 12:00 PM |
| Location | Virtual (Zoom) |
| Prepared By | Yogesh |
| Attendees | Alice Johnson (Lead Developer)  Bob Brown (QA Engineer)  Emily Davis (UI/UX Designer)  Michael Wilson (Backend Developer)  John Doe (Scrum Master) |

|  |  |  |  |
| --- | --- | --- | --- |
| **Sprint Status** | **Things to demo** | **Quick Updates** | **What’s Next** |
| Sprint [X] completed successfully. | Patient insurance plan details retrieval. | API integration with insurance providers completed. | Begin Sprint [X+1] planning. |
| All planned user stories delivered | Benefits information display | UI enhancements for better user experience. | Focus on adding patient eligibility verification. |
| No major blockers reported. | Authorization status integration | Performance improvements for faster data loading. | Work on enhancing claims submission feature. |
|  | Claims information dashboard. |  | Address any feedback from stakeholders. |

**Meeting Type 3: Sprint retrospective meeting**

|  |  |
| --- | --- |
| Date | March, 22nd , 2025 |
| Time | 10:00 AM - 12:00 PM |
| Location | Virtual (Zoom) |
| Prepared By | Yogesh |
| Attendees | Alice Johnson (Lead Developer)  Bob Brown (QA Engineer)  Emily Davis (UI/UX Designer)  Michael Wilson (Backend Developer)  John Doe (Scrum Master) |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Agenda** | **What Went Well** | **What Didn’t Go Well** | **Questions** | **Reference** |
| Review Sprint Goals | Successfully integrated insurance plan details API. | Delays in testing due to incomplete test data. | Can we improve test data preparation for the next sprint? | Sprint Goal Document |
| Discuss Blockers | Effective collaboration between dev and QA teams. | Authorization information API had performance issues. | How can we optimize API performance? | Blockers Log |
| Evaluate Team Performance | Completed all planned user stories for claims information. | Missed deadlines for documenting new features**.** | Can we allocate more time for documentation? | Sprint Backlog |
| **Identify Improvements** | Daily stand-ups were productive and kept everyone aligned | Lack of clarity in some user stories led to rework. | How can we improve user story clarity? | Retrospective Notes from Previous Sprint |
| Plan for Next Sprint | Successfully implemented agile best practices for continuous integration. | Communication gaps between remote team members. | Can we schedule more frequent check-ins for remote team members? | Product Roadmap |

**Meeting Type 4: Daily Stand-up meeting**

|  |  |
| --- | --- |
| **Question** | **Name/Role Week ‘X’(from 24-3-25 to 30-3-25)** |

**Mon Tue Wed Thu Fri**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| What did you do yesterday? | Developer 1  Developer 2  Developer 3 | Worked on integrating insurance plan details API.  Designed the UI for patient insurance plan details.  Set up the database schema for storing insurance plan details. | Debugged issues with the benefits display module.  Added filters for searching patient benefits.    Optimized database queries for faster benefits retrieval. | Implemented authorization information retrieval logic.  Worked on improving the authorization workflow UI.  Worked on backend logic for authorization status updates. | Tested claims information API endpoints.  Fixed bugs in the claims information display.  Tested database performance for claims data. | Refactored code for better performance.  Conducted code reviews and merged pull requests.  Documented API endpoints for claims information. |
| What will you do today? | Developer 1  Developer 2  Developer 3 | Start working on integrating patient eligibility checks.  Add responsive design for the insurance plan details page.  Optimize database indexing eligibility check | Begin testing the benefits module for edge cases.  Work on UI improvements for the benefits section.  Work on backend logic for benefits validation. | Work on error handling for authorization API.  Implement a dashboard for authorization status.  Implement logging for authorization API. | Start implementing claims status tracking.  Add pagination to the claims information table.  Start work on claims data export functionality. | Plan for next sprint tasks.  Review and update UI documentat  ion.   |  |  | | --- | --- | | Prepare database performance report. |  | |
| What (if any) is blocking your progress? | Developer 1  Developer 2  Developer 3 | Waiting for clarification on eligibility criteria from the product owner.  Need feedback from the design team on the UI layout.  Database server performance issues causing delays. | No blockers.  No blockers.  No blockers. | No blockers.  No blockers.  No blockers. | No blockers.  No blockers.  No blockers. | No blockers  No blockers  No blockers... |

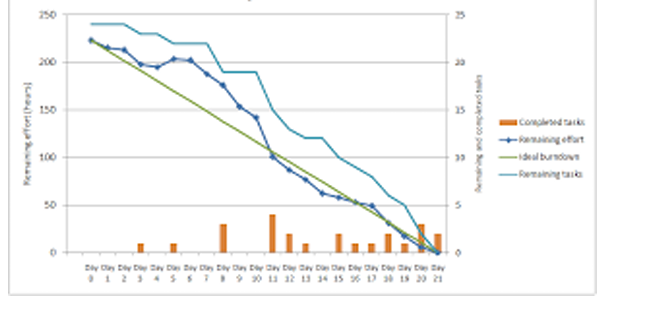
1. **Product backlog**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **User Story ID** | **User Story** | **Tasks** | **Priority** | **BV** | **CP** | **Sprint** |
| US-001 | As a healthcare provider, I want to view a patient's insurance plan details so that I can verify coverage. | Design UI for insurance plan details.  Integrate with insurance provider API. | High | 8 | 5 | 1 |
| US-002 | As a healthcare provider, I want tosee a patient's benefits summary so that I can determine what services are covered. | Create benefits summary UI  Fetch benefits data from insurance API.    Display covered services, co-pays, and deductibles. | High | 9 | 8 | 2 |
| US-003 | As a healthcare provider, I want to check authorization requirements for a procedure so that I can ensure compliance. | Develop authorization lookup feature.  Integrate with authorization API.  Display authorization status and required documents. | Medium | 7 | 7 | 3 |
| US-004 | As a healthcare provider, I want to view a patient's claims history so that I can track past services and payments. | Design claims history UI.  Fetch claims data from insurance API.  Display claim ID, date, service, and status. | Medium | 6 | 6 | 4 |
| US-005 | As a healthcare provider, I want to submit a pre-authorization request so that I can get approval for a procedure**.** | Create pre-authorization request form.   Integrate with authorization API.  Validate form inputs. | High | 8 | 9 | 5 |
| US-006 | As a healthcare provider, I want to receive notifications for authorization updates so that I can stay informed. | Set up notification system.  Integrate with authorization API for updates.  Send email and in-app notifications. | Low | 5 | 5 | 6 |
| US-007 | As a healthcare provider, I want to search for a patient's insurance information by name or ID so that I can quickly access their details. | Add search functionality to UI.  Implement backend search logic.  Display search results in a table. | Medium | 7 | 6 | 7 |
| US-008 | As a healthcare provider, I want to export patient insurance details as a PDF so that I can share them with other providers. | Add export button to UI.  Generate PDF with patient insurance details.  Include branding and formatting. | Low | 4 | 4 | 8 |
| US-009 | As a healthcare provider, I want to see real-time updates on claim status so that I can track progress. | Integrate with claims API for real-time updates.   Display claim status with timestamps. | High | 8 | 7 | 9 |
| US-010 | As a healthcare provider, I want to view a dashboard of all my patients' insurance details so that I can manage them efficiently. | Design dashboard UI.  Fetch and display aggregated insurance data.  Add filters for patient name, insurance type, and status. | Medium | 6 | 8 | 10 |

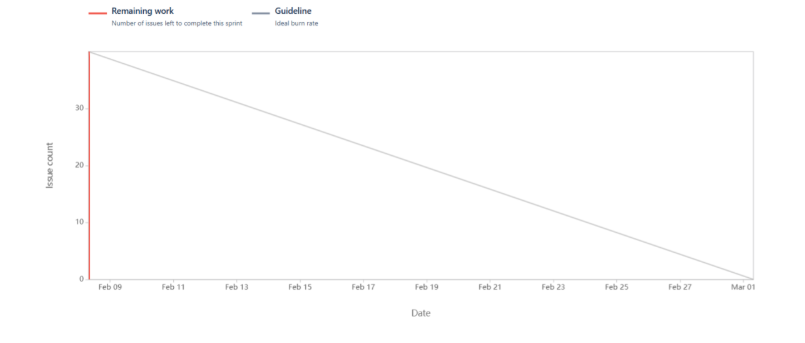
**Sprint Backlog**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **User Story ID** | **User Story** | **Tasks** | **Owner** | **Status** | **Estimated Effort (hours)** |
| US- 101 | As a provider, I want to view a patient's insurance plan details so that I can verify coverage. | Design API endpoint for fetching insurance plan details.  Develop backend service to retrieve insurance plan data.   Create frontend component to display insurance plan details. | John | In Progress | 20 |
| US - 102 | As a provider, I want to see a patient's benefits summaryso that I can determine what services are covered. | Define data model for benefits summary.  Develop API to fetch benefits summary.   Implement frontend UI for benefits summary display. | Emily | To Do | 25 |
| US-103 | As a provider, I want to check authorization requirements for a specific procedure so that I can ensure compliance. | Research and document authorization rules for common procedures.  Develop backend logic to check authorization requirements.  Build frontend form to input procedure details. | Alex | In Progress | 30 |
| US-104 | As a provider, I want to view a patient's claims history so that I can track past services and payments. | Design database schema for claims history.  Develop API to retrieve claims history.  Create frontend table to display claims history. | David | To Do | 35 |
| **US-105** | As a provider, I want to export patient insurance and claimsdata in PDF format so that I canshare it with other stakeholders. | Research PDF generation libraries.  Develop backendservice to generate PDFs.  Add export button to the frontend UI.  Test PDF generation with sample data. | Chris | To Do | 15 |
|  |  |  |  |  |  |

**Product Burndown Chart**

****

**Sprint Burndown**

****