**Agile Documents**

**Document 1: Definition of Done**

As Per Agile Extension to the BABOK® Guide v2, Definition of Done is a technique where the team agrees on, and prominently displays, a list of criteria which must be met before a backlog item is considered done.

That is the team has to create a well-defined, unambiguous, measurable, agreed-upon, and shared Definition of Done between all team members.

The best form of Definition of Done representation is a checklist of activities that has to demonstrate the agreed value and quality of a user story. So, this checklist should include:

* acceptance criteria (to satisfy customer requirements for a product)
* quality criteria (to satisfy quality requirements for a product)

Definition of Done may be defined for different levels of project work. For example, in Agile / Scrum framework these levels of work could be user story, sprint, and release.

## **Levels of Definition of Done**

1. **User Story Level:** Ensures that individual backlog items meet all necessary requirements before completion.
2. **Sprint Level:** Ensures that all user stories in a sprint comply with the agreed-upon DoD.
3. **Release Level:** Ensures that the product is ready for deployment with all necessary approvals, documentation, and compliance.

## **Checklist for Definition of Done**

The following checklist outlines the essential activities required for a backlog item to be considered "Done."

### **1. Produced Code for Presumed Functionalities**

* The functionality described in the user story has been implemented.
* Code follows the defined coding standards and best practices.
* Code does not introduce security vulnerabilities or performance bottlenecks.

### **2. Assumptions of User Story Met**

* All business rules and assumptions stated in the user story have been accounted for.
* There are no unaddressed dependencies that could impact functionality.

### **3. Project Builds Without Errors**

* The latest build of the application compiles and runs successfully without errors or warnings.
* No broken dependencies or unresolved merge conflicts exist.

### **4. Unit Tests Written and Passing**

* Automated unit tests cover key functionalities of the feature.
* Unit test coverage meets the required threshold (e.g., 80% coverage).
* All unit tests pass without failures.

### **5. Project Deployed on a Test Environment Identical to the Production Platform**

* The feature is deployed in an environment that mirrors production to ensure compatibility.
* Deployment scripts and automation configurations are validated.

### **6. Tests on Devices/Browsers Listed in the Project Assumptions Passed**

* The feature has been tested across all specified devices and browsers.
* Cross-browser and responsive design testing are completed (if applicable).

### **7. Feature Ok-ed by UX Designer**

* UI/UX design aligns with the agreed wireframes and mockups.
* User experience is reviewed and signed off by the UX team.

### **8. QA Performed & Issues Resolved**

* Manual and automated testing is completed by the QA team.
* Any reported defects are triaged, fixed, and re-tested.
* Regression testing ensures that new changes do not break existing functionality.

### **9. Feature is Tested Against Acceptance Criteria**

* The feature satisfies all functional and non-functional requirements defined in the acceptance criteria.
* Business stakeholders validate that the implementation meets expectations.

### **10. Feature Ok-ed by Product Owner**

* The Product Owner (PO) reviews and accepts the feature.
* Any feedback from the PO is incorporated before moving to release.

### **11. Refactoring Completed**

* Code refactoring is done where necessary to improve maintainability.
* Code complexity is minimized without affecting functionality.

### **12. Any Configuration or Build Changes Documented**

* Any modifications to configuration settings or build parameters are documented.
* Necessary scripts, environment variables, and dependencies are updated in the documentation.

### **13. Documentation Updated**

* Technical documentation, such as API references and architecture diagrams, is updated.
* User documentation, FAQs, or release notes are provided if applicable.

### **14. Peer Code Review Performed**

* Code has been reviewed by at least one other team member.
* Feedback from the review is addressed before merging changes.
* Code adheres to best practices, security guidelines, and maintainability standards.

**Document 2: Product Vision**

| **Scrum Project Name:** | Graduate Admissions Tracking System |
| --- | --- |
| **Venue:** | Hinjewadi, Pune |
| **Date:** 20 Jan 2025 | **Start Time:** 11:00AM | **End Time:** 01:00PM | **Duration:** 2hrs |
| **Client:** | Nexon University |
| **Stakeholder List:** | Project Stakeholders:* Business Analyst
* Project Manager
* Development Team
* Tech Expert
* Testing Team
* Operations (Network, Training)
* UI Designer
 | Business Stakeholders:* Project Manager
* University Administration
* Admissions Director
* Operations Team
* Subject Matter Expert
 | 3rd Party Stakeholders:* Auditors
* Legal Team
* External Consultants
* Vendors
* Technology Providers
 |
|
|
| **Scrum Team:** |
| **Scrum Master:** | Mugdha |
| **Product Owner:** | Rinsha |
| **Scrum Developer 1:** | Nikhil |
| **Scrum Developer 2:** | Poornima |
| **Scrum Developer 3:** | Pooja |
| **Scrum Developer 4:** | Prutha |
| **Scrum Developer 5:** | Richa |
| **UI Designer:** | Shubham |
| **Tester 1:** | Abhay  |
| **Tester 2:** | Prasad |
| **Business Analyst:** | Swati Kurwade |

| **Vision**:  | To streamline and enhance the graduate admissions process by providing an efficient, transparent, and user-friendly online tracking system. |
| --- | --- |
| **Target Group:**To attract more applicants and universities towards using an automated and paperless graduate admissions system. | **Needs**:* Applicants can track their application progress without needing to contact the admissions office.
* Reduces manual workload for university staff, improving efficiency.
* Enhances transparency and real-time updates in the admission process.
* Saves time for both applicants and administrators.
 | **Product**:The Graduate Admissions Tracking System will enable universities to process applications efficiently while allowing students to track their status online, reducing delays and manual follow-ups. | **Value**:* Reduce manual admission inquiries by 50-80%.
* Improve efficiency in document verification and review.
* Increase applicant satisfaction with real-time status updates.
* Provide a user-friendly and easy-to-navigate platform for applicants and administrators.
 |

**Document 3: User stories**

**User Stories** User Stories are short, simple descriptions of functionality or requirements from the user's perspective.

**Business Value (BV)** refers to how important a feature (User Story) is to the business.

**Complexity Points (CP)** represent the effort required for developers to implement the feature. It is rated as per the CP points known as Poker cards.

**Acceptance criteria** define the conditions a software product must meet to be accepted by the user, customer, or system. They are unique for each user story and describe the feature’s expected behavior.

| **User Story No:** 1 | **Tasks:** Enable applicants to create an account | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** create an account using my email ID and password
* **So that I can** apply for graduate programs online
 |
| **BV:** 300 | **CP:** 5 |
| **Acceptance Criteria:** Applicants should be able to register with an email, receive a verification link, and login successfully. |

| **User Story No:** 2 | **Tasks:** Enable applicants to reset their passwords | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** reset my password if I forget it
* **So that I can** regain access to my account securely
 |
| **BV:** 250 | **CP:** 3 |
| **Acceptance Criteria:** Applicants should receive a password reset link via email, allowing them to set a new password. |

| **User Story No:** 3 | **Tasks:** Allow applicants to fill out application forms | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** complete my application online
* **So that I can** submit my details for graduate admission
 |
| **BV:** 400 | **CP:** 8 |
| **Acceptance Criteria:** The application form should allow applicants to enter personal details, academic history, and upload required documents. |

| **User Story No:** 4 | **Tasks:** Implement an application submission confirmation email | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** receive a confirmation email after submitting my application
* **So that I can** ensure my application has been received
 |
| **BV:** 150 | **CP:** 2 |
| **Acceptance Criteria:** Upon submission, the system should send an email with an acknowledgment and application reference number. |

| **User Story No:** 5 | **Tasks:** Allow administrators to view submitted applications | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** view all submitted applications
* **So that I can** monitor and manage the admission process efficiently
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** Administrators should have a dashboard displaying submitted applications with filtering and sorting options. |

| **User Story No:** 6 | **Tasks:** Allow applicants to track their application status | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** track my application status in real-time
* **So that I can** know where my application stands in the process
 |
| **BV:** 450 | **CP:** 7 |
| **Acceptance Criteria:** Applicants should be able to log in and see their application status (e.g., Under Review, Accepted, Rejected). |

| **User Story No:** 7 | **Tasks:** Implement document upload functionality for applicants | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** upload required documents
* **So that I can** complete my application process
 |
| **BV:** 500 | **CP:** 9 |
| **Acceptance Criteria:** The system should support document uploads with format and size validation. |

| **User Story No:** 8 | **Tasks:** Allow admission officers to review applications | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an Admission Officer**
* **I want to review applications and supporting documents**
* **So that I can assess applicants' eligibility**
 |
| **BV:** 400 | **CP:** 8 |
| **Acceptance Criteria:** Officers should be able to review applications, provide comments, and mark decisions. |

| **User Story No:** 9 | **Tasks:** Implement a role-based access system | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** ensure different roles have appropriate access levels
* **So that I can** maintain system security and data integrity
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** The system should restrict access based on user roles (Applicant, Admission Officer, Admin). |

| **User Story No:** 10 | **Tasks:** Enable communication between applicants and admission officers | **Priority:** |
| --- | --- | --- |
| **Value Statement:*** **As an Applicant**
* **I want to send messages to admission officers**
* **So that I can clarify doubts about my application**
 |
| **BV:** 200 | **CP:** 4 |
| **Acceptance Criteria:** The system should have an in-app messaging feature for applicants and admission officers. |

| **User Story No:** 11 | **Tasks:** Implement interview scheduling functionality | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** schedule an interview with the admission committee
* **So that I can** choose a suitable date and time for my interview
 |
| **BV:** 400 | **CP:** 7 |
| **Acceptance Criteria:** Applicants should be able to view available slots and book an interview; the system should send confirmation emails. |

| **User Story No:** 12 | **Tasks:** Enable payment processing for application fees | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** securely pay my application fee online
* **So that I can** complete my application submission
 |
| **BV:** 450 | **CP:** 8 |
| **Acceptance Criteria:** The system should support online payments, generate receipts, and confirm successful transactions. |

| **User Story No:** 13 | **Tasks:** Allow applicants to withdraw their application | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** withdraw my application if needed
* **So that I can** remove my application from consideration
 |
| **BV:** 300 | **CP:** 5 |
| **Acceptance Criteria:** Applicants should have the option to withdraw their application, triggering an email confirmation. |

| **User Story No:** 14 | **Tasks:** Implement automated email notifications | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** receive email updates about my application
* **So that I can** stay informed about the progress
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** The system should send automated emails for status changes, interview invitations, and submission confirmations. |

| **User Story No:** 15 | **Tasks:** Implement document verification process | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an Admission Officer**
* **I want to verify the submitted documents**
* **So that I can ensure the authenticity of applications**
 |
| **BV:** 500 | **CP:** 9 |
| **Acceptance Criteria:** The system should flag unverified documents and allow officers to mark them as verified or request resubmission. |

| **User Story No:** 16 | **Tasks:** Provide analytics dashboard for administrators | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an Administrator**
* **I want to access application data analytics**
* **So that I can make data-driven admission decisions**
 |
| **BV:** 400 | **CP:** 7 |
| **Acceptance Criteria:** Admins should have access to visual reports and insights on applications, admissions, and trends. |

| **User Story No:** 17 | **Tasks:** Implement automated reminders for applicants | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an Applicant**
* **I want to receive reminders for pending tasks**
* **So that I can complete my application on time**
 |
| **BV:** 300 | **CP:** 5 |
| **Acceptance Criteria:** The system should send automated reminders for incomplete applications, missing documents, and upcoming deadlines. |

| **User Story No:** 18 | **Tasks:** Enable integration with third-party services | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** integrate the system with external services
* **So that I can** enhance system capabilities (e.g., payment gateways, document verification tools)
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** The system should support integration with APIs for secure and efficient third-party service utilization. |

| **User Story No:** 19 | **Tasks:** Allow applicants to accept or reject admission offers | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** accept or reject my admission offer
* **So that I can** confirm or decline my enrollment
 |
| **BV:** 450 | **CP:** 8 |
| **Acceptance Criteria:** Applicants should be able to view their offer, accept or decline it, and receive confirmation emails. |

| **User Story No:** 20 | **Tasks:** Generate application reports for administrators | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an Administrator**
* **I want to generate detailed reports on applications**
* **So that I can track admissions data effectively**
 |
| **BV:** 400 | **CP:** 7 |
| **Acceptance Criteria:** The system should allow admins to generate custom reports on application trends, statuses, and demographics. |

| **User Story No:** 21 | **Tasks:** Implement customizable application forms | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** customize application forms
* **So that I can** collect relevant information from applicants
 |
| **BV:** 400 | **CP:** 7 |
| **Acceptance Criteria:** Admins should be able to modify form fields, add/remove sections, and save changes dynamically. |

| **User Story No:** 22 | **Tasks:** Enable bulk application review | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Admission Officer
* **I want to** review multiple applications at once
* **So that I can** streamline the admission decision process
 |
| **BV:** 500 | **CP:** 8 |
| **Acceptance Criteria:** Officers should be able to filter, select, and evaluate multiple applications in a batch. |

| **User Story No:** 23 | **Tasks:** Implement multi-factor authentication (MFA) | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** enable multi-factor authentication
* **So that I can** secure my account from unauthorized access
 |
| **BV:** 450 | **CP:** 9 |
| **Acceptance Criteria:** The system should support MFA via email/SMS with an option to enable/disable it in user settings. |

| **User Story No:** 24 | **Tasks:** Process application fee waiver requests | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** request a fee waiver
* **So that I can** apply even if I have financial constraints
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** The system should allow applicants to submit waiver requests with supporting documents for admin review. |

| **User Story No:** 25 | **Tasks:** Enable bulk document uploads | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** upload multiple documents at once
* **So that I can** save time during the application process
 |
| **BV:** 350 | **CP:** 5 |
| **Acceptance Criteria:** The system should support batch uploads with validation checks for file format and size. |

| **User Story No:** 26 | **Tasks:** Notify users of system downtimes | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** receive downtime notifications
* **So that I can** plan my application activities accordingly
 |
| **BV:** 250 | **CP:** 4 |
| **Acceptance Criteria:** Users should receive email/portal notifications about planned and unplanned downtimes. |

| **User Story No:** 27 | **Tasks:** Implement fraud detection mechanisms | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** detect fraudulent applications
* **So that I can** maintain the integrity of the admissions process
 |
| **BV:** 500 | **CP:** 9 |
| **Acceptance Criteria:** The system should flag suspicious activities such as duplicate applications and inconsistent document details. |

| **User Story No:** 28 | **Tasks:** Enable data export features for administrators | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** export application data
* **So that I can** analyze and generate reports externally
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** The system should allow data export in multiple formats (CSV, PDF, Excel) with filtering options. |

| **User Story No:** 29 | **Tasks:** Integrate alumni feedback system | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** collect alumni feedback
* **So that I can** improve the admissions process based on past student experiences
 |
| **BV:** 400 | **CP:** 7 |
| **Acceptance Criteria:** The system should allow alumni to submit structured feedback, which can be analyzed by admins. |

| **User Story No:** 30 | **Tasks:** Integrate with student information systems (SIS) | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** sync admission data with SIS
* **So that I can** streamline student onboarding
 |
| **BV:** 500 | **CP:** 9 |
| **Acceptance Criteria:** The system should allow seamless data transfer between the admissions platform and SIS through APIs. |

| **User Story No:** 31 | **Tasks:** Develop mobile application support | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** access the application portal via a mobile app
* **So that I can** track my application status on the go
 |
| **BV:** 500 | **CP:** 10 |
| **Acceptance Criteria:** The mobile app should support login, application submission, status tracking, and notifications. |

| **User Story No:** 32 | **Tasks:** Implement chatbot assistance for FAQs | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** get instant answers to common questions
* **So that I can** resolve my queries without waiting for support
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** The chatbot should provide automated responses based on a predefined knowledge base. |

| **User Story No:** 33 | **Tasks:** Track conditional admissions | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Admission Officer
* **I want to** monitor applicants who have received conditional offers
* **So that I can** ensure they meet the required conditions
 |
| **BV:** 450 | **CP:** 8 |
| **Acceptance Criteria:** The system should track applicants with pending requirements and update status upon fulfillment. |

| **User Story No:** 34 | **Tasks:** Enable collaborative review by multiple officers | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Admission Officer
* **I want to** collaborate with colleagues on application review
* **So that I can** ensure a fair evaluation process
 |
| **BV:** 500 | **CP:** 9 |
| **Acceptance Criteria:** The system should allow multiple officers to add comments, provide ratings, and finalize decisions collaboratively. |

| **User Story No:** 35 | **Tasks:** Process deferral requests | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** request a deferral for my admission
* **So that I can** delay my enrollment if needed
 |
| **BV:** 300 | **CP:** 5 |
| **Acceptance Criteria:** The system should allow applicants to submit deferral requests, which admins can review and approve. |

| **User Story No:** 36 | **Tasks:** Add language support for international applicants | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** access the application portal in multiple languages
* **So that I can** understand the process easily
 |
| **BV:** 400 | **CP:** 7 |
| **Acceptance Criteria:** The system should support multiple languages and allow users to switch between them. |

| **User Story No:** 37 | **Tasks:** Process offline application submissions | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** process applications submitted via offline channels
* **So that I can** ensure all applicants are considered
 |
| **BV:** 350 | **CP:** 6 |
| **Acceptance Criteria:** Admins should be able to manually enter offline applications into the system. |

| **User Story No:** 38 | **Tasks:** Enable custom email templates for notifications | **Priority:** Medium |
| --- | --- | --- |
| **Value Statement:*** **As an** Administrator
* **I want to** customize email templates
* **So that I can** personalize communication with applicants
 |
| **BV:** 300 | **CP:** 5 |
| **Acceptance Criteria:** The system should allow admins to create and modify email templates for various notifications. |

| **User Story No:** 39 | **Tasks:** Implement automated waitlist management | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Admission Officer
* **I want to** manage waitlisted applicants automatically
* **So that I can** efficiently allocate available seats
 |
| **BV:** 500 | **CP:** 9 |
| **Acceptance Criteria:** The system should automatically move applicants from the waitlist based on seat availability. |

| **User Story No:** 40 | **Tasks:** Develop personalized applicant dashboards | **Priority:** High |
| --- | --- | --- |
| **Value Statement:*** **As an** Applicant
* **I want to** see my application status, deadlines, and documents in one place
* **So that I can** track my progress easily
 |
| **BV:** 450 | **CP:** 8 |
| **Acceptance Criteria:** The dashboard should display personalized application data, updates, and next steps. |

**Document 4: Agile PO Experience**

The Product Owner has a vision of the product, keeping in mind the domain/industry experience and the market need. The role of the Product Owner (PO) is pivotal in ensuring that the product aligns with business goals, customer needs, and market demands. The PO acts as the bridge between stakeholders and the Scrum team, ensuring that the development aligns with the strategic objectives of the organization.

* Following are the responsibilities of PO in a project
	+ Market Analysis
		- Conducting a comprehensive analysis of market needs and demands to determine potential opportunities for product development.
		- Researching the availability of similar products in the market and analyzing their strengths and weaknesses to identify differentiation strategies.
		- Assessing competitor strategies, pricing models, and market trends to refine the product positioning and feature set.
		- Example: Analyzing trends in graduate admissions software to understand how universities streamline applications, identify gaps in existing solutions, and propose new features such as AI-driven document verification.
	+ Enterprise Analysis
		- Performing due diligence on market opportunities by evaluating customer segments, revenue potential, and competitive landscape.
		- Understanding the impact of regulatory requirements, industry standards, and technological advancements on the product.
		- Engaging with key stakeholders to validate market assumptions and business objectives.
		- Example: Conducting due diligence on the application processing needs of universities, ensuring compliance with education regulations, and understanding integration with existing student information systems.
	+ Product Vision and Roadmap
		- Defining a clear and compelling product vision that aligns with market needs and business goals.
		- Developing a strategic product roadmap outlining high-level features, milestones, and timelines.
		- Continuously refining and updating the roadmap based on market feedback, business priorities, and technological advancements.
		- Example: Establishing a roadmap for features such as applicant profile creation, document submission, automated verification, review workflows, and decision notifications.
	+ Managing Product Features
		- Engaging with stakeholders to understand their expectations and translating them into actionable product features.
		- Prioritizing epics, user stories, and features based on criticality, business value, and return on investment (ROI).
		- Collaborating with cross-functional teams to ensure the timely delivery of prioritized features while maintaining product quality and usability.
		- Example: Prioritizing features such as automated eligibility checks, scoring algorithms for applications, and dashboards for university staff to track applicant status.
	+ Managing Product Backlog
		- Defining and maintaining a well-organized product backlog to facilitate seamless development processes.
		- Prioritizing user stories based on business value, stakeholder inputs, and technical feasibility.
		- Continuously refining and reprioritizing backlog items to adapt to changing business needs and market conditions.
		- Planning and managing epics to ensure alignment with the overall product vision and strategy.
		- Example: Managing backlog items such as adding a recommendation letter submission feature, optimizing document processing speeds, and integrating third-party payment gateways for application fees.
	+ Managing Overall Iteration Progress
		- Regularly reviewing sprint progress to ensure alignment with the product roadmap and stakeholder expectations.
		- Adjusting sprint priorities and epics when necessary to accommodate emerging business needs and technical challenges.
		- Conducting sprint retrospectives in collaboration with Business Analysts to identify areas of improvement and implement actionable changes.
		- Example: Adjusting sprint priorities when a university requests a new report generation feature or enhancing data security after stakeholder feedback.
* From this project, I have learned how to handle sprint meetings such as:
	+ Sprint Planning Meeting
		- Defining sprint objectives and goals in alignment with business priorities.
		- Breaking down user stories into actionable tasks and assigning them to the development team.
		- Estimating story points and capacity planning for the sprint.
		- Example: Planning a sprint to implement an application tracking module where students can check the status of their submissions in real-time.
	+ Daily Scrum Meeting
		- Monitoring the progress of the development team and identifying any roadblocks.
		- Encouraging collaboration among team members to resolve challenges efficiently.
		- Ensuring that the team remains focused on sprint goals and adheres to Agile principles.
		- Example: Discussing blockers such as slow API response times for document uploads and coordinating with the technical team to resolve them.
	+ Sprint Review Meeting
		- Demonstrating completed features to stakeholders and gathering feedback for improvements.
		- Evaluating whether sprint goals were achieved and discussing potential refinements.
		- Aligning the next steps based on stakeholder inputs and business needs.
		- Example: Showcasing the newly implemented analytics dashboard for tracking application trends and collecting feedback on additional filtering options.
	+ Sprint Retrospective Meeting
		- Analyzing the sprint performance to identify successes and challenges.
		- Discussing potential process improvements to enhance team efficiency and collaboration.
		- Implementing actionable insights to refine future sprint executions.
		- Example: Reviewing why a specific feature took longer to develop than expected and brainstorming ways to improve estimations.
	+ Backlog Refinement Meeting
		- Continuously reviewing and updating the backlog to ensure it remains relevant and prioritized.
		- Breaking down large epics into smaller, manageable user stories.
		- Clarifying user story details to ensure that development teams have a clear understanding of requirements.
		- Example: Breaking down the "Application Fee Payment" epic into stories such as payment gateway integration, error handling, and transaction logs.
* Also, User Story creation and what things will be included in user stories such as:
	+ Story No
		- Assigning a unique identifier to each user story for easy tracking and reference.
	+ Story Format
		- **As a** [user role], **I want** [feature/action], **so that** [benefit/value].
		- Example:

**As an** applicant**, I want to** track my application status in real time**, so that** I can stay informed about my admissions process.

* + Tasks
		- Breaking down user stories into specific tasks required for implementation.
		- Example:
			* Implement an application status tracking page.
			* Integrate email and SMS notifications for status updates.
			* Enable document upload tracking.
	+ Priority
		- Assigning priority levels to user stories based on business value, urgency, and dependencies.
		- Example:
			* **High:** Critical features (e.g., application submission, status tracking).
			* **Medium:** Important but can be refined over time (e.g., chatbot assistance, multi-language support).
			* **Low:** Enhancements that add convenience (e.g., dark mode for UI, personalized notifications).
	+ Acceptance Criteria
		- Defining clear, measurable conditions that must be met for a user story to be considered complete.
		- Example:
* Applicants should receive an email notification when their application status changes.
* The system should display clear status updates (e.g., Submitted, Under Review, Accepted, Rejected).
	+ BV & CP Value
		- Assigning Business Value (BV) to indicate the importance of a feature from a stakeholder perspective.
		- Example:
* **High:** Reduces administrative workload and improves applicant experience.
* **Medium:** Enhances system usability but does not directly impact admissions.
* **Low:** Nice-to-have features with minimal impact.
	+ - Assigning Complexity Points (CP) to assess the technical effort required for implementation.
		- Example:
			* **1-3 points:** Simple tasks (e.g., adding an email confirmation step).
			* **5-8 points:** Moderate complexity (e.g., implementing a dynamic dashboard).
			* **13+ points:** High complexity (e.g., full system integration with SIS and third-party services).
* In Scrum, a Product Owner serves as the liaison between multiple areas of an organization. This person communicates with business stakeholders and collaborates closely with Scrum teams to ensure that all areas of the business remain informed on a project's development. The PO ensures alignment between business needs and development efforts, facilitating a smooth and efficient Agile workflow.
* The Product Owner develops a vision of a product's function and operation, which in turn allows this Scrum team member to define product features and break those features into product backlog items. Through iterative planning and continuous collaboration, the PO ensures that the product evolves to meet user needs and business objectives effectively.

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

**Product Backlog**

It is a **dynamic, prioritized list** of features, enhancements, bug fixes, infrastructure updates, and other tasks that a team may deliver to achieve a specific outcome.

Adding or removing backlog items should be **quick and effortless**, ensuring that only those contributing directly to progress or enabling future advancements remain.

Backlog items can take various formats, with **user stories** being the most common. The team determines the format that best suits their workflow and uses these items as references for the aspects of the solution they may work on.

| **User Story ID** | **User Story** | **Tasks** | **Priority** | **BV** | **CP** | **Sprint** |
| --- | --- | --- | --- | --- | --- | --- |
| US01 | **AS AN** APPLICANT, **I WANT TO** CREATE AN ACCOUNT, **SO THAT** I CAN START MY APPLICATION PROCESS | - Design and implement sign-up page - Validate and store user registration information - Integrate UI with backend | High | 500 | 2 | 1 |
| US02 | **AS AN** APPLICANT, **I WANT** **TO** LOGIN TO THE SYSTEM, **SO THAT** I CAN ACCESS AND UPDATE MY APPLICATION | - Create login UI - Implement user authentication - Test login functionality | High | 500 | 2 | 1 |
| US03 | **AS AN** ADMIN, **I WANT** **TO** TRACK APPLICATION STATUS **SO THAT** I CAN MONITOR PROGRESS | - Design an admin dashboard - Implement application tracking module - Ensure real-time updates | High | 300 | 3 | 2 |
| US04 | **AS AN** APPLICANT, **I WANT TO** UPLOAD REQUIRED DOCUMENTS **SO THAT** MY APPLICATION CAN BE PROCESSED | - Implement document upload feature - Validate file formats - Store documents securely | High | 400 | 2 | 2 |
| US05 | **AS AN** ADMIN, **I WANT TO** REVIEW SUBMITTED APPLICATIONS **SO THAT** I CAN MAKE ADMISSION DECISIONS | - Design application review interface - Implement filters and sorting - Integrate with decision-making workflow | High | 500 | 3 | 2 |
| US06 | **AS AN** APPLICANT, **I WANT TO** RECEIVE NOTIFICATIONS ABOUT MY APPLICATION STATUS **SO THAT** I STAY INFORMED | - Implement notification system - Integrate with email and SMS services - Ensure timely updates | Medium | 300 | 2 | 2 |
| US07 | **AS AN** APPLICANT, **I WANT TO** PAY APPLICATION FEES ONLINE **SO THAT** I CAN COMPLETE MY APPLICATION | - Integrate secure payment gateway - Validate and process payments - Provide payment receipts | High | 400 | 2 | 2 |
| US08 | **AS AN** ADMIN, **I WANT TO** GENERATE REPORTS ON APPLICATION TRENDS **SO THAT** I CAN MAKE DATA-DRIVEN DECISIONS | - Design reporting dashboard - Implement data extraction queries - Ensure data accuracy | Medium | 300 | 3 | 3 |
| US09 | **AS AN** APPLICANT, **I WANT TO** EDIT MY APPLICATION BEFORE FINAL SUBMISSION **SO THAT** I CAN CORRECT ERRORS | - Implement edit functionality - Validate user inputs - Restrict edits after submission | High | 400 | 2 | 3 |
| US10 | **AS AN** APPLICANT, **I WANT TO** REQUEST RECOMMENDATION LETTERS THROUGH THE SYSTEM **SO THAT** MY APPLICATION IS COMPLETE | - Design a recommendation request system - Notify recommenders automatically - Track responses | Medium | 300 | 3 | 3 |

**Sprint Backlog**

The **Sprint Backlog** consists of three key components:

* **Sprint Goal (Why):** The objective the team aims to achieve during the Sprint.
* **Selected Product Backlog Items (What):** The specific items chosen for the Sprint.
* **Actionable Plan (How):** The detailed steps to deliver the Increment.

The Sprint Backlog is **created by and for the Developers**, serving as a highly visible, real-time representation of their planned work. It evolves throughout the Sprint as the team gains new insights.

To ensure transparency and alignment, the Sprint Backlog should be detailed enough to allow Developers to **track progress effectively** during the **Daily Scrum**.

| **User Story ID** | **User Story** | **Tasks** | **Owner** | **Status** | **Estimated Effort** |
| --- | --- | --- | --- | --- | --- |
| US01 | **AS AN** APPLICANT, **I WANT TO** SIGN UP **SO THAT** I CAN CREATE A NEW ACCOUNT | - Design sign-up page - Validate and store registration info - Connect UI with backend | Richa | Done | 6 |
| US02 | **AS AN** APPLICANT, **I WANT TO** LOGIN **SO THAT** I CAN ACCESS MY APPLICATION | - Create login UI - Implement authentication - Connect UI with backend | Nikhil | Done | 8 |
| US03 | **AS AN** APPLICANT, **I WANT TO** RESET MY PASSWORD **SO THAT** I CAN REGAIN ACCESS | - Implement password reset functionality - Design UI for password recovery - Ensure security measures | Prutha | Pending | 5 |
| US04 | **AS AN** APPLICANT, **I WANT TO** VIEW MY APPLICATION STATUS **SO THAT** I KNOW ITS PROGRESS | - Create status tracking page - Retrieve and display real-time updates - Ensure UI responsiveness | Prutha | Pending | 4 |
| US05 | **AS AN** APPLICANT, **I WANT TO** LOGOUT **SO THAT** I CAN SECURELY END MY SESSION | - Implement logout functionality - Add logout option in UI - Ensure session termination | Richa | Done | 7 |
| US06 | **AS AN** ADMIN, **I WANT TO** VIEW ALL APPLICATIONS **SO THAT** I CAN MANAGE ADMISSIONS | - Design admin dashboard - Implement application filtering - Display key application details | Poornima | Done | 5 |
| US07 | **AS AN** APPLICANT, **I WANT TO** PAY APPLICATION FEES **SO THAT** I CAN SUBMIT MY APPLICATION | - Implement payment gateway - Ensure secure payment processing - Provide receipts | Nikhil | Pending | 8 |
| US08 | **AS AN** APPLICANT, **I WANT TO** DOWNLOAD MY PAYMENT RECEIPT **SO THAT** I HAVE A RECORD | - Implement receipt generation - Provide download option | Pooja | Pending | 2 |
| US09 | **AS AN** ADMIN, **I WANT TO** GENERATE REPORTS ON APPLICATIONS **SO THAT** I CAN TRACK TRENDS | - Design report generation system - Implement filters for analysis - Ensure accuracy | Pooja | Done | 4 |
| US10 | **AS AN** APPLICANT, **I WANT TO** ACCESS THE SYSTEM ON MY MOBILE DEVICE **SO THAT** I CAN APPLY ON-THE-GO | - Implement a responsive mobile interface - Ensure compatibility and usability | Poornima | Done | 6 |

### **Product Burndown Chart**

It is a visual tool used in Agile project management to track the amount of work remaining in the entire project over time. It helps teams and stakeholders monitor overall progress and forecast project completion.

* **Purpose:** Shows how much work is left in the project.
* **Scope:** Covers the entire project lifecycle.
* **Data Collected:** Work completed vs. remaining across all sprints.
* **Benefit:** Helps teams and stakeholders assess long-term progress and adjust plans accordingly.

###

###

###

###

###

###

###

###

###

###

###

###

### **Sprint Burndown Chart**

It is a visual tool used in Agile project management focusing on the remaining work within a single sprint. It helps teams measure daily progress and determine if they are on track to complete the sprint goal.

* **Purpose:** Tracks progress within a sprint and forecasts sprint completion.
* **Scope:** Covers a single sprint.
* **Data Collected:** Work completed vs. remaining within the sprint.
* **Benefit:** Allows teams to manage workload, identify bottlenecks, and adjust as needed.



**Document 6: Sprint Meetings**

**Sprint Planning Meeting:**

| **Date:** | 27 Jan 2025 |
| --- | --- |
| **Time:** | 11:00AM to 01:00PM |
| **Location:** | Hinjewadi, Pune |
| **Prepared By:** | Swati Kurwade |
| **Attendees:** | Mugdha | Scrum Master |
|  | Rinsha | Product Owner |
| Nikhil | Developer 1 |
| Richa | Developer 2 |
| Poornima | Developer 3 |
| Swati Kurwade | BA |
| Shubham | UI Designer |
| Abhay | Tester 1 |
|  | Prasad | Tester 2 |

####

#### **Agenda Topics:**

| **Topic** | **Presenter** | **Time Allotted** |
| --- | --- | --- |
| Scopes | Swati Kurwade | 11:00 AM - 11:15 AM |
| Strategy & Planning | SM, BA, PO | 11:15 AM - 11:30 AM |
| Resource Management | Scrum Master | 11:30 AM - 11:50 AM |
| Billing & Invoices | BA | 11:50 AM - 12:00 PM |
| Graduate Dashboard (GD) | Product Owner | 12:00 PM - 01:00 PM |

**Other Information:**

| **Observers:** | Tester Team, SME |
| --- | --- |
| **Resources:** | * Manpower: Business stakeholders from Nexon University, BA, Developers, Project Team
* Time: 6 months
* Budget: Rs. 50,00,000
* Other: Database, Internet, Past Applications, Reports
 |
| **Special Notes:** | Clients will be migrated virtually from the legacy system |

###

### **Sprint Review Meeting:**

| **Date:** | 14 Feb 2025 |
| --- | --- |
| **Time:** | 11:00AM to 01:00PM |
| **Location:** | Hinjewadi, Pune |
| **Prepared By:** | Swati Kurwade |
| **Attendees:** | Mugdha | SM |
|  | Rinsha | PO |
| Nikhil | Dev 1 |
| Richa | Dev 2 |
| Poornima | Dev 3 |
| Abhay | Tester 1 |
| Prasad | Tester 2 |

#### **Sprint Status**

| **Sprint Status** | **Things to Demo** | **Quick Updates** | **What’s Next** |
| --- | --- | --- | --- |
| Sprint 1: Done | User Registration, Document Upload, Application Submission | All features enabled | Deploy in the live environment |

###

### **Sprint Retrospective Meeting:**

| **Date:** | 14 Feb 2025 |
| --- | --- |
| **Time:** | 11:00AM to 01:00PM |
| **Location:** | Hinjewadi, Pune |
| **Prepared By:** | Swati Kurwade |
| **Attendees:** | Mugdha | SM |
|  | Rinsha | PO |
| Nikhil | Dev 1 |
| Richa | Dev 2 |
| Poornima | Dev 3 |
| Abhay | Tester 1 |
| Prasad | Tester 2 |

####

#### **Agenda**

| **What Went Well** | **What Didn’t Go Well** | **Questions** | **Reference** |
| --- | --- | --- | --- |
| All sprint backlogs were covered in time. Sign-off received for the first iteration. | Issues with the registration module code. | UAT and unit testing need completion. | SME, PO |

###

### **Daily Stand-up Meeting**

| **Question** | **Name/Role** | **Week 1 (from 03-02-2025 to 14-02-2025)** |
| --- | --- | --- |
| **Monday** | **Tuesday** | **Wednesday** | **Thursday** | **Friday** | **Sat** | **Sun** |
| **What did you do yesterday?** | **Developer 1** | Worked on user registration | Completed user registration | Completed document upload | Worked on application submission | Completed submission process |  |  |
| **Developer 2**  |
| **Developer 3** |
| **What will you do today?** | **Developer 1** | Finalize registration module | Fix minor issues in registration | Develop payment gateway integration | Test application submission | Deploy and validate workflows |  |  |
| **Developer 2**  |
| **Developer 3** |
| **What (if any) is blocking your progress?** | **Developer 1** | Few bugs | No blockers | Some validation errors | No blockers | Performance testing pending |  |  |
| **Developer 2**  |
| **Developer 3** |