

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.

Checklist for DOD:

- User guides or admin manuals are prepared
- Functional and technical documentation is updated
- Course Assessments and Tests.
- Integration testing completed
- No open critical or high-severity bugs.
- Functionality and Usability
- Student data (e.g., ID, grades, contact details) is stored securely.
- Role-based access is verified
- Accessibility
- User Experience Optimization.
- Technical Optimization.
- Assumptions of User Story met
- Project builds without errors
- Unit tests written and passed
- Project deployed on the test environment identical to production platform
- Tests on devices/browsers listed in the project assumptions passed
- Feature ok-ed by UX designer
- QA performed & issues resolved
- Feature is tested against acceptance criteria
- Feature ok-ed by Product Owner
- Refactoring completed

- Any configuration or build changes documented
- Documentation updated
- Database schema updated in documentation.
- Ready for release or included in the next scheduled deployment.
- Feature is deployed to UAT/staging environment.
- All related API endpoints tested and documented (e.g., for SMS alerts).
- Notifications (email/SMS) are triggered and verified.
- Related reports (e.g., attendance, grades, fee due) are functional and accurate
- Meets **security, legal, or compliance** requirements.
- Follows **performance and accessibility** guidelines.
- Story is marked as **"Done" in the board** (e.g., Jira, Trello).

Document 2- Product Vision

Scrum Project Name:	EVS SMS		
Venue:	Chennai		
Date: 10.03.2025	Start time: 9.00AM	End time: 6.00 PM	Duration: 8 months
Client:	EVS college		
Stakeholder list:	Students Faculty Business Owner Administrator Regional Administrator E-Learning Project Manager Technical Experts		

	Upper Management		
Scrum Team			
Scrum Master:	David George		
Product owner:	Monisha Moudivender		
Scrum Developer 1:	Keerthana		
Scrum Developer 2:	Deepak Kumar		
Scrum Developer 3:	Priyadharshan		
Scrum Developer 4:	Sindhuja		
Scrum Developer 5:	Gracy		

Vision: To create a centralized, secure, and user-friendly Student Management System that streamlines academic administration, enhances data accuracy, real-time communication, generates reports, and supports informed decision-making for educational institutions.

Target Group	Needs	Product	Value
The product addresses the education sector, specifically colleges, universities, and training centers that require digital tools to manage student data efficiently. Target Users and Customers	EVS struggle with student data, manual processes, and delayed reporting. This product solves those problems by: Digitizing student registration, attendance, grading, and course management	The Student Management System is a web and mobile based application that centralizes student and management related functions into a single platform. It is:	The product delivers value by: Reducing administrative costs and paperwork Improving operational efficiency and compliance Enabling timely decisions through real-time data

Administrators	Automating reporting and notifications	Desirable: User-friendly interface, role-based access, supports academic workflows Special: Customizable for different education levels and institutions needs Feasible: Built using scalable, secure technology with modular components	Supporting institutional goals such as academic excellence, student tracking, and digital transformation
Teachers and Faculty	Reducing errors and delays in academic operations		
Students	Providing quick access to accurate records for staff and leadership		
Institutional Management (Upper Management)	Student Management System – Key Benefits Saves time and reduces manual work Easy online access for students and staff Improves communication via alerts Centralizes student data Helps management with quick decisions Reduces errors and ensures data security		Business Goals: Accelerate institutional digitalization Increase user satisfaction and efficiency Provide scalable solutions for regional deployments

Document 3: User stories

User Story No:1	Task:2	Priority:HIGH
Value Statement: AS A STUDENT I WANT TO LOGIN TO MY ACCOUNT SO THAT I CAN VIEW MY PROFILE		
BV:1000		CP:2
Acceptance Criteria: Login page, User ID and password field, login option Basic Flow: Open the application Enter the credentials (User ID, Password) Click login Homepage is shown Alternative Flow: Incorrect password or user id Show error message Prompt to re-enter credentials Exceptional Flow: Showing network error Multiple incorrect credentials leads to lock the account		

User Story No:2	Task:2	Priority:HIGH
Value Statement: AS A FACULTY I WANT TO LOGIN TO MY ACCOUNT SO THAT I CAN UPDATE GRADES		

BV:1000	CP:2
Acceptance Criteria: Login page, User ID and password field, login option Basic Flow: Open the application Enter the credentials (User ID, Password) Click login Homepage is shown Alternative Flow: Incorrect password or user id Show error message Prompt to re-enter credentials Exceptional Flow: Showing network error Multiple incorrect credentials leads to lock the account	

User Story No:3	Task:2	Priority:HIGH
Value Statement: AS AN ADMIN I WANT TO LOGIN TO MY ACCOUNT SO THAT I CAN DO MY JOB		
BV:1000		CP:2
Acceptance Criteria: Login page, User ID and password field, login option		

Basic Flow:

Open the application
Enter the credentials (User ID, Password)
Click login
Homepage is shown

Alternative Flow:

Incorrect password or user id
Show error message
Prompt to re-enter credentials

Exceptional Flow:

Showing network error
Multiple incorrect credentials leads to lock the account

User Story No:4	Task:2	Priority:HIGH
Value Statement: AS AN ADMIN I WANT REGISTER SO THAT I CAN LOGIN TO MY ACCOUNT		
BV:1000		CP:2
Acceptance Criteria: Registration Screen Text Boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number. Click on Register Button. Send Successful Notification to the user Basic Flow: Open login page Click register		

Enter Name, phone number, address, email ID, and other details
Create user id and password
Click submit

Alternative Flow:

Show empty fields and prompt enter details
Invalid character in fields
Fill mandatory field

Expectational Flow:

Show User already exists

User Story No:5	Task:2	Priority:HIGH
Value Statement: AS AN ADMIN I WANT REGISTER STUDENT WITH THEIR DETAILS SO THAT THE STUDENT CAN LOGIN TO HIS ACCOUNT		
BV:1000		CP:2
Acceptance Criteria: Registration Screen Text Boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number. Click on Register Button. Send Successful Notification to the user Basic Flow: Open login page Click register Enter Name, phone number, address, email ID, and other details Create user id and password Click submit		

Alternative Flow:

Show empty fields and prompt enter details

Invalid character in fields

Fill mandatory field

Expectational Flow:

Show User already exists

User Story No:6	Task:2	Priority:HIGH
Value Statement: AS AN ADMIN I WANT REGISTER A FACULTY SO THAT HE CAN LOGIN TO HIS ACCOUNT		
BV:1000		CP:2
Acceptance Criteria: Registration Screen Text Boxes for User Name, Password, Nation ID, Mobile No, Email, Address, Phone Number. Click on Register Button. Send Successful Notification to the user Basic Flow: Open login page Click register Enter Name, phone number, address, email ID,and other details Create user id and password Click submit Alternative Flow: Show empty fields and prompt enter details		

Invalid character in fields
Fill mandatory field

Expectational Flow:

Show User already exists

User Story No:7	Task:1	Priority:MEDIUM
Value Statement: AS A TEACHER I WANT TO MARK THE ATTENDANCE IN MY CLASS SO THAT THEIR PRESENCE AND ABSENCE ARE RECORDED		
BV:500		CP:1
Acceptance Criteria: Teacher should be able to select the class and date from a dropdown/calendar. The system should display a list of enrolled students in that class. Teacher should be able to mark each student as Present, Absent, or Excused. Once submitted, the attendance data should be saved and not editable (unless permitted). System should confirm submission and display a success message. Attendance records should be visible in the reports module. Basic Flow: Teacher selects a class from timetable. System loads the student list. Teacher marks present/absent.		

System saves attendance data.

Alternative Flow:

Teacher uploads attendance via file template.

Exceptional Flow:

System fails to load list displays error.

Unauthorized user tries to mark attendance access denied.

User Story No:8	Task:2	Priority:HIGH
Value Statement: AS AN ADMIN I WANT ENROLL STUDEN IN A COURSE SO THAT HE CAN ATTEND THE CLASS		
BV:1000		CP:2
Acceptance Criteria: Administrator can select a student and view eligible courses based on grade/program. Administrator can assign one or more courses to the student. System should prevent duplicate course enrollment. Confirmation message is shown upon successful enrollment. Enrolled courses should reflect in the student's profile. Basic Flow: Admin logs in and views available courses.		

Student selects desired courses.
System checks course prerequisites and availability.
Admin submits enrollment request.
System confirms enrollment.

Alternative Flow:

Student adds courses to a wishlist first and enrolls later.

Exceptional Flow:

Course full system shows “course not available”.

User Story No:9	Task:2	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO RECORD AND TRACK PAYMENT SO THAT I KNOW THE DUES AND GENERATE REPORTS		
BV:500		CP:2
Acceptance Criteria: Admin can select a student and view fee structure. Admin can enter paid amount and mode of payment. System should update outstanding balance automatically. Payment history should be saved and viewable. A receipt should be generated upon successful entry.		

Basic Flow:

Admin selects report type and filters.

System gathers data.

Admin clicks Generate.

Report displayed and exportable.

Alternative Flow:

Admin schedules report generation automatically.

Exceptional Flow:

No data report empty.

Export fails retry option.

User Story No:10	Task:1	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO SEND NOTIFICATIONS SO THAT STUDENTS STAY UPDATED		
BV:500		CP:1
Acceptance Criteria: Admin can select notification type (e.g., fee reminder, exam schedule). Admin can select recipients (individual or batch). Notification message can be previewed before sending. System should send SMS/email using integrated API. Sent notifications should be logged for reference.		

Basic Flow:

Admin drafts notification.

Selects target group (students, staff).

Schedules or sends immediately.

System delivers via email/SMS.

Alternative Flow:

Admin reuses template for recurring messages.

Exceptional Flow:

Invalid contacts → skipped.

SMS service error → fallback to email.

User Story No:11	Task:2	Priority:MEDIUM
Value Statement: AS A TEACHER I WANT TO ENTER THE GRADES SO THAT STUDENTS CAN SEE THEIR GRADE		
BV:500		CP:1
Acceptance Criteria: Teacher can select the class, subject, and exam term. System displays a list of enrolled students for that subject. Teacher can enter grades for each student. Grades are validated based on max marks. Entered grades are saved and viewable in student profiles and reports.		

Basic Flow:

Alternative Flow:

Exceptional Flow:

User Story No:12

Task:2

Priority:MEDIUM

Value Statement:

AS AN ADMIN
I WANT TO UPDATE EXAM SCHEDULE
SO TEACHERS AND STUDENTS CAN BE PREPARED

BV:500

CP:1

Acceptance Criteria:

Admin can create an exam schedule by selecting class, subject, date, and time.

No exam conflicts (same date/time for a class) should be allowed.

Once published, the schedule should be viewable by teachers and students.

Admin should be able to edit or delete the schedule before the exam date.

Basic Flow:

Admin creates new exam event.
Selects subject, class, date, and room.
Assigns invigilators.
Publishes schedule.

Alternative Flow:

Upload bulk schedule via spreadsheet.

Exceptional Flow:
Room clash system suggests another slot.
Missing invigilator system shows warning.

User Story No:13	Task:2	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO DOWNLOAD REPORT CARDS SO THAT I CAN GENERATE REPORT		
BV:500		CP:2
Acceptance Criteria: Admin can select class and term to generate report cards. Report includes student details, grades, attendance, and remarks. Reports are formatted and exportable as PDF. Reports can be printed or emailed to students/parents. Only authorized users can generate or access reports. Basic Flow: Admin selects report type and filters. System gathers data. Admin clicks Generate. Report displayed and exportable. Alternative Flow: Admin schedules report generation automatically. Exceptional Flow:		

No data report empty.
Export fails retry option.

User Story No:14	Task:2	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO UPDATE STUDENT PROFILE SO THAT DATA CAN BE ACCURATE		
BV:500		CP:2
Acceptance Criteria: Admin can search and view any student's full profile. Admin can update fields like contact info, address, and guardian details. Core academic fields (e.g., student ID, date of birth) are not editable without permission. Change history is logged for audit purposes. Basic Flow: Admin logs in. Opens profile section. Edits details and submits. System validates and updates info. Alternative Flow: Admin approval needed for major changes. Exceptional Flow: Validation fails system highlights error.		

Submission error retry required.

User Story No:15	Task:2	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO ASSIGN ROLE BASED ACCESS SO THAT USER CAN ACCESS WHAT THEY ARE ALLOWED		
BV:500		CP:2
Acceptance Criteria: Roles include Admin, Teacher, Student, and Regional Officer. Admin can assign roles to new or existing users. Each role has defined permissions (e.g., view, edit, delete). Unauthorized access attempts are blocked and logged.		

User Story No:16	Task:2	Priority:MEDIUM
Value Statement: AS A STUDENT I WANT TO VIEW THE TIME TABLE SO THAT I CAN KNOW THE CLASSES		

BV:400	CP:1
Acceptance Criteria: Student can view the timetable based on their class/section. Timetable shows subject name, time, room number, and teacher name. Timetable can be filtered by day or downloaded as PDF. Any updates in the timetable are reflected in real-time. Basic Flow: Student logs in. Clicks “My Timetable”. System displays weekly schedule. Alternative Flow: Filters timetable by subject or day. Exceptional Flow: No timetable available system shows notice.	

User Story No:17	Task:1	Priority:LOW
Value Statement: AS A STUDENT I WANT TO SUBMIT THE FEEDBACK SO THAT THE MANAGEMENT CAN IMPROVE THEMSELF		
BV:200		CP:1
Acceptance Criteria:		

Student can select a feedback type (e.g., academic, facility, faculty, general).

A text field is available to describe the issue or feedback in detail.

Feedback must be saved with a unique ID and submission date.

Acknowledgement message is shown after successful submission.

Admin should be able to view, respond, and update the status of the feedback.

Student should be able to view the response status in their portal.

Basic Flow:

Student selects teacher and subject.

Rates using scale, adds optional comment.

Submits.

System records feedback.

Alternative Flow:

Feedback only includes rating, no comments.

Exceptional Flow:

Already submitted system prevents duplicates.

User Story No:18	Task:1	Priority:LOW
Value Statement: AS A FACULTY I WANT TO SUBMIT THE FEEDBACK SO THAT THE MANAGEMENT CAN IMPROVE THEMSELF		
BV:200		CP:1
Acceptance Criteria:		

Faculty can select a feedback type (e.g., academic, facility, faculty, general).

A text field is available to describe the issue or feedback in detail.

Feedback must be saved with a unique ID and submission date.

Acknowledgement message is shown after successful submission.

Admin should be able to view, respond, and update the status of the feedback.

Student should be able to view the response status in their portal.

Basic Flow:

Faculty selects teacher and subject.

Rates using scale, adds optional comment.

Submits.

System records feedback.

Alternative Flow:

Feedback only includes rating, no comments.

Exceptional Flow:

Already submitted system prevents duplicates.

User Story No:19	Task:1	Priority:LOW
Value Statement: AS AN ADMIN I WANT TO SUBMIT THE FEEDBACK SO THAT THE MANAGEMENT CAN IMPROVE THEMSELF		
BV:200		CP:1
Acceptance Criteria:		

Admin can select a feedback type (e.g., academic, facility, faculty, general).

A text field is available to describe the issue or feedback in detail.

Feedback must be saved with a unique ID and submission date.

Acknowledgement message is shown after successful submission.

Admin should be able to view, respond, and update the status of the feedback.

Student should be able to view the response status in their portal.

Basic Flow:

Admin selects teacher and subject.

Rates using scale, adds optional comment.

Submits.

System records feedback.

Alternative Flow:

Feedback only includes rating, no comments.

Exceptional Flow:

Already submitted system prevents duplicates.

User Story No:20	Task:1	Priority:MEDIUM
Value Statement: AS AN ADMIN I WANT TO GENERATE REPORT SO THAT MANAGEMENT CAN TAKE DECISIONS BASED ON THAT		
BV:500		CP:2
Acceptance Criteria:		

Reports can be filtered by teacher, department, term, and class.

Report shows average rating per feedback parameter (e.g., communication, knowledge, engagement).

Detailed view includes student comments (anonymous if applicable).

Export option available in PDF and Excel formats.

Reports are read-only and cannot be altered once generated.

Option to compare current quarter vs previous quarter performance.

Basic Flow:

Coordinator selects department, term, and teacher.

System aggregates feedback.

Report displayed with summary and export option.

Alternative Flow:

Coordinator compares with previous quarters.

Exceptional Flow:

No feedback available system alerts.

Document 4:

Agile PO Experience The Product Owner has a vision of the product keeping the domain/industry experience and the market need.

❖ Following are the responsibilities of PO in a project

➤ Market Analysis

- Analysis of market need/demand
- Availability of similar products in the market

The Student Management System (SMS) project addresses the growing need for digital transformation in the education sector. With increasing demand for centralized student

data, performance tracking, fee management, and digital communication, SMS solutions are becoming essential for schools, colleges, and training institutes. This project's strengths lie in automation, cloud access, and user-friendly features, making it a competitive solution for institutions seeking efficiency, compliance, and better student engagement, and driven by trends like cloud adoption, mobile accessibility, and integration with learning tools.

➤ Enterprise Analysis

▫ Due diligence on the market opportunity

Enterprise analysis as a knowledge area that describes business analysis activities that take place for an enterprise for the purpose of:

- Identifying business opportunities
- Building a business architecture
- Determining the optimal project investment path for the enterprise
- Implementing new business and technical solutions

The Enterprise Analysis for the Student Management System (SMS) project focuses on aligning the solution with the strategic goals of educational institutions. Schools, colleges, and universities aim to improve operational efficiency, reduce administrative workload, and enhance student experiences. The current manual or semi-digital systems are time-consuming, error-prone, and lack integration, leading to inefficiencies in student tracking, fee collection, and academic performance monitoring. By implementing a centralized, cloud-based SMS, the institution can streamline core processes such as enrollment, attendance, grading, and communication. This solution supports strategic initiatives like digital transformation, compliance with regulatory standards, and improved stakeholder satisfaction (students, parents, faculty, and administrators). Moreover, the system offers scalability for future growth, supports data-driven decision-making through custom reports and dashboards, and integrates with other platforms like LMS and communication tools. Overall, this analysis confirms the SMS project as a high-value investment for institutions aiming to modernize their academic and administrative operations.

Before initiating the Student Management System (SMS) project, thorough due diligence was conducted to assess market opportunities and validate demand. The education sector is witnessing a global shift toward digitization, with institutions actively seeking automated solutions for student data management, academic monitoring, and communication. Trends like cloud computing, mobile accessibility, AI-powered insights, and system integration (with LMS and finance systems) further widen the scope for

innovation and differentiation. The need for compliance with privacy regulations like GDPR and FERPA also increases the demand for secure, role-based data systems. This due diligence confirms a strong market opportunity for deploying an agile, user-friendly, and cost-effective SMS tailored to evolving institutional needs.

➤ **Product Vision and Roadmap**

- Product vision keeping the need analysis in mind
- Product roadmap with high-level features and timeline

Product Vision

To develop a centralized, user-friendly, and secure Student Management System (SMS) that automates student-related processes—such as registration, course enrollment, attendance, grading, exam scheduling, and notifications—enhancing administrative efficiency, improving student experience, and supporting data-driven decision-making for educational institutions.

Key Needs Addressed:

- Manual data entry and errors
- Delayed communication
- Lack of real-time reports
- Inefficient tracking of attendance, grades, and fees

Product Roadmap

Sprint-wise Product Roadmap

Sprint 1: Core Setup – Registration & Course Management

- Student registration with validation
- Admin login and role-based access (Student, Teacher, Admin)
- User profile creation and management
- Course creation and editing
- Assign courses to teachers
- Student course enrollment

Sprint 2: Attendance & Timetable Management

- Teacher marks daily attendance
- Student views attendance status

- Attendance report generation (daily/monthly)
- Admin creates class timetable
- Students and teachers view their schedules
- Exam scheduling feature

Sprint 3: Grade & Exam Management

- Enter and update student grades
- View grades (student and teacher view)
- Generate performance reports
- View exam dates and subjects

Sprint 4: Fee Management & Notification System

- Generate and manage fee invoices
- Record and track payments
- View fee payment history
- Send automated SMS/email notifications for fees, grades, attendance
- Admin sends bulk announcements

Sprint 5: Reports, Dashboard & Mobile View

- Student-wise academic summary reports
- Course and attendance reports
- Fee status reports
- Admin dashboard with key metrics
- Mobile-responsive UI for student portal
- Accessibility and UI refinements

Sprint 6: UAT, Go-Live & Support

- Conduct User Acceptance Testing
- Integrate stakeholder feedback
- Final bug fixes and performance improvements
- Deployment and go-live
- Admin/teacher training
- Prepare user manuals and provide post-launch support

➤ Managing Product Features

- Managing stakeholder expectations and prioritizing needs
- Prioritization of the epics, stories, and features based on criticality and ROI involved

Managing Stakeholder Expectations & Prioritizing Needs

Stakeholder Engagement & Expectation Management:

Conduct interviews, workshops, and surveys with key stakeholders (administrators, teachers, students).

Maintain a communication channel (weekly updates, backlog review sessions)

Set realistic expectations by showing what’s feasible within budget, time, and technical limits.

Use MoSCoW prioritization

Share early prototypes or wireframes to align vision and reduce scope creep.

Prioritization of Epics, Stories, and Features Based on Criticality & ROI

Prioritization Criteria:

- Criticality (Is it core to system functioning?)
- ROI (Will it save time, money, effort, or improve user satisfaction?)
- Dependencies (Is this feature required to unlock others?)
- Regulatory/Compliance Need (Is it legally or institutionally required?)
- User Impact (How many users benefit?)

Epic/Feature	Criticality	ROI Impact	Priority
Student Registration	High	High	Must-Have (Sprint 1)
Attendance Tracking	High	Medium	Must-Have (Sprint 2)
Grade Management	Medium	High	Should-Have (Sprint 3)
Notification via SMS	Medium	High	Should-Have (Sprint 4)

- Managing Product Backlog
 - Prioritization of user stories
 - Reprioritization based on stakeholders' needs
 - Epics planning

Prioritization of User Stories

Goal: Ensure high-value features are delivered first based on business need, user impact, and technical readiness.

Factors Considered:

- Business value
- Urgency from stakeholders
- Dependencies with other modules
- Team capacity

Reprioritization Based on Stakeholder Needs

Why It Happens:

- Change in policies (e.g., new exam rules)
- Stakeholders' feedback during demos

Approach:

- Hold regular backlog refinement meetings (every sprint or biweekly)
- Use Impact Mapping to understand new priorities
- Reassess user stories using updated inputs (cost, effort, urgency)
- Maintain version history of backlog to track changes

- Managing Overall Iteration Progress
 - Sprint progress review
 - Reprioritization of sprints and epics if needed
 - Sprint retrospectives with Business Analyst

Sprint Progress Review

Purpose: To monitor whether the team is on track to complete the committed work and deliver value.

Key Activities:

- **Daily Stand-ups:** Developers, testers, BA, and Scrum Master sync on:
 - What was done yesterday?
 - What will be done today?
 - Any blockers?
- **Mid-Sprint Review:**
 - Check burn-down chart.
 - Compare planned vs actual task progress.
 - Ensure user stories meet acceptance criteria.
- Tracks burn-down
- Validates progress aligns with MVP scope
- Accepts/rejects stories based on DoD
- Resolves ambiguity in stories
- Supports developers/testers with functional questions
- Verifies progress meets user needs

Reprioritization of Sprints and Epics (If Needed)

When It's Needed:

- Stakeholders change priorities mid-project.
- Dependencies shift (e.g., one module delay impacts another).
- A feature is not delivering expected business value.
- Coordinates with stakeholders on trade-offs
- Performs impact analysis on functionality and data
- Suggests which features can be split or deferred

Approach:

- Conduct backlog refinement before each sprint.
- Use BA-led stakeholder sessions to evaluate change impact.
- Reorder stories or epics in backlog using MoSCoW or ROI analysis.
- Possibly replan future sprints to accommodate new direction

Sprint Retrospectives (with Business Analyst)

Purpose: To reflect and improve team processes after each sprint.

- Notes feedback for next sprint planning

- Suggests process improvements
- Highlights recurring misunderstandings or story refinements needed

Format:

- What went well?
- What didn't go well?
- What can we improve?
- What actions will we take next sprint?

BA's Role in Retrospective:

- Shares feedback from stakeholders and UAT.
 - Highlights recurring issues (e.g., unclear requirements, missed acceptance criteria).
 - Suggests process improvements (e.g., earlier involvement in design/testing).
 - Encourages better collaboration with QA and Dev teams.
- ❖ From this project I have learned how to handle sprint meetings such as
- Sprint planning meeting
 - Daily scrum meeting
 - Sprint review meeting
 - Sprint retrospective meeting
 - Backlog refinement meeting

From this project, I have learned how to handle sprint meetings as a Product Owner.

I gained practical experience leading and participating in all key Agile ceremonies, ensuring that each sprint stayed aligned with the product vision and delivered business value.

Sprint Planning

Objective: Define the sprint goal and decide which backlog items will be delivered.

In **Sprint Planning**, I prioritized user stories based on stakeholder needs and ROI, clearly communicated the sprint goal, and collaborated with the team and Business Analyst to finalize scope and clarify acceptance criteria.

What I Did as PO:

- Presented the highest priority user stories from the product backlog.
- Explained the business value and expected outcomes for each story (e.g., "Student Registration" is critical for onboarding).
- Collaborated with the team to ensure acceptance criteria were well-defined and clear.
- Answered any open questions from the team and clarified scope boundaries.
- Ensured the sprint was feasible, with a manageable workload.

What I Learned:

- How to align sprint goals with the product vision.
- The importance of clear and well-prioritized backlog before sprint planning.
- Collaborating with the team helps avoid ambiguity and scope creep.

Daily Stand-Up

Objective: Keep track of team progress and resolve blockers quickly.

During **Daily Stand-ups**, I monitored progress, addressed blockers, and made quick decisions to help the team stay on track with sprint commitments.

What I Did as PO:

- Attended stand-ups as an observer or contributor.
- Listened for any impediments affecting story progress.
- Provided quick decisions when priorities shifted or clarification was needed.
- Aligned with the BA on user concerns or stakeholder expectations.

What I Learned:

- Regular touchpoints help keep the team aligned with the sprint goal.
- Quick decision-making can reduce delays.
- Supporting the BA and team during execution boosts confidence and efficiency.

Sprint Review (Demo)

Objective: Present completed work to stakeholders and collect feedback.

In the **Sprint Review**, I facilitated demos, collected stakeholder feedback, and evaluated completed stories against the Definition of Done before accepting or rejecting them.

What I Did as PO:

- Facilitated the demo session alongside the team.
- Showed how completed features (e.g., attendance module, timetable) addressed stakeholder needs.
- Gathered feedback from stakeholders, noted change requests.
- Accepted or rejected stories based on Definition of Done (DoD) and acceptance criteria.
- Updated the product backlog based on review outcomes.

What I Learned:

- Stakeholder feedback is key to refining the product direction.
- Ensuring clarity in acceptance criteria avoids rework.
- Validating delivered value helps build stakeholder trust.

Sprint Retrospective

Objective: Reflect on what went well and what can be improved in the next sprint.

Through **Sprint Retrospectives**, I contributed to discussions on what went well, what could be improved, and helped define actionable steps to improve team efficiency and product quality.

What I Did as PO:

- Participated as a team member, respecting the Scrum Master's facilitation.
- Shared thoughts on collaboration, communication, and requirements flow.
- Discussed how priority changes or unclear requirements affected the sprint.
- Agreed on action items for improvement (e.g., refining stories earlier with the BA).

What I Learned:

- Continuous improvement boosts team morale and efficiency.
- It's important to listen actively and support the team's feedback.
- A collaborative mindset strengthens team ownership and product quality.

Backlog Refinement (Ongoing)

Objective: Keep the product backlog clear, prioritized, and ready for future sprints.

I also actively participated in **Backlog Refinement** sessions, continuously updating and reprioritizing the product backlog to reflect changing business priorities.

What I Did as PO:

- Worked closely with the BA to groom stories ahead of planning.
- Reprioritized epics and stories based on stakeholder feedback, business urgency, and ROI.
- Ensured dependencies were mapped and addressed early.
- Adjusted story scope when required, based on team capacity.

What I Learned:

- A well-refined backlog avoids chaos during sprint planning.
- Prioritization is not static – it must adapt to stakeholder needs and external changes.
- BA support in analyzing impact makes refinement faster and smarter.

❖ Also, User stories creation and what things will be included in user stories such as

- Story no
- Tasks
- Priority
- Acceptance criteria
- BV & CP value

User Story Creation – Product Owner Role in SMS Project

As a Product Owner in the Student Management System (SMS) project, I was responsible for defining and managing the product backlog by creating well-structured user stories that clearly communicated business needs to the development team. Ensure that each user story clearly aligned with the product vision, and delivered measurable value. Here's how I handled user story creation for the SMS project:

Writing user stories in a user-centric format:

“As a [role], I want to [action], so that [benefit].”

This ensured that each story reflected real user needs and delivered measurable value.

User Story Numbering

Assigned a unique Story No for easy tracking and referencing during sprint planning, daily stand-ups, and reviews.

Tasks Breakdown

Broke down each user story into sub-tasks in collaboration with the Scrum team. Tasks were developer- and tester-friendly and helped define effort clearly.

Priority Assignment

Prioritized stories based on:

- Business urgency
- Stakeholder need
- Feature dependency

Used MoSCoW or numeric scale (1 – High, 2 – Medium, 3 – Low).

Example: Registration (Priority 1), Notifications (Priority 3)

Acceptance Criteria

Defined clear, testable Acceptance Criteria for every user story using bullet format to ensure shared understanding.

Example:

- User must be able to register with email and password
- Error message displayed for invalid input
- Data must be saved in student profile table

Business Value (BV) & Complexity Points (CP)

BV (Business Value): Defining Business Value to indicate the importance or impact of each story, which helped prioritize the backlog effectively. Assigned by PO based on ROI, stakeholder need, compliance.

CP (Complexity Point): Estimated in collaboration with the development team to gauge the effort and guide sprint planning by using Planning Poker or Fibonacci scale.

PO Contributions

- Collaborated with stakeholders to gather requirements
- Translated needs into refined user stories with full clarity
- Ensured all stories were ready for sprint planning (Definition of Ready)
- Participated in backlog grooming and story reprioritization based on stakeholder feedback
- Validated completed stories against acceptance criteria during sprint review

❖ 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 keep all areas of the business informed on a project's development.

❖ 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.

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

Product backlog:

User story ID	User story	Tasks	Priority	BV	CP	Sprint
US001	As an Administrator, I want to register a new student into the system with their personal, academic, and contact details, So that the student can be enrolled in courses and tracked throughout their academic lifecycle.	Register with mobile number/email id Enrolling for the classes	HIGH	1000	3	1
US002	As a Teacher, I want to mark daily attendance for students in my assigned class, So that their presence and absence are recorded accurately and reflected in reports.	Mark attendance	MEDIUM	500	2	2

US003	As a student I want a detailed description of my course So that I can choose my course	Add option to add description	MEDIUM	300	2	2
US004	As an Administrator, I want to enroll a student in available courses based on their grade or program, So that they can attend classes and access course materials.	Course enrollment	Medium	500	1	3
US005	As an Administrator, I want to record and track fee payments for each student, So that I can monitor dues and generate financial reports.	Generate report	Medium	200	2	3
US006	As an Administrator, I want to send SMS or email notifications for attendance, exams, or fee dues, So that students and parents stay informed and engaged.	Update notification	High	500	3	1
US007	As a Teacher, I want to enter and view grades for students based on their performance in exams, So that their academic progress is tracked and report cards can be generated.	Update grades	High	1000	2	2

US008	As a student I want to select the payment mode So that I can make payment of my choice	Option to select the payment mode	High	500	3	2
US009	As a student I want to be able to rate my experience on a scale of 1 to 5 So that other users will get to know	Rate experience on a scale of 1 to 5	Low	200	1	3
US010	As a admin I want to provide schedule So that students can check their classes' timing	Providing class schedule	Medium	500	1	3

Sprint backlog:

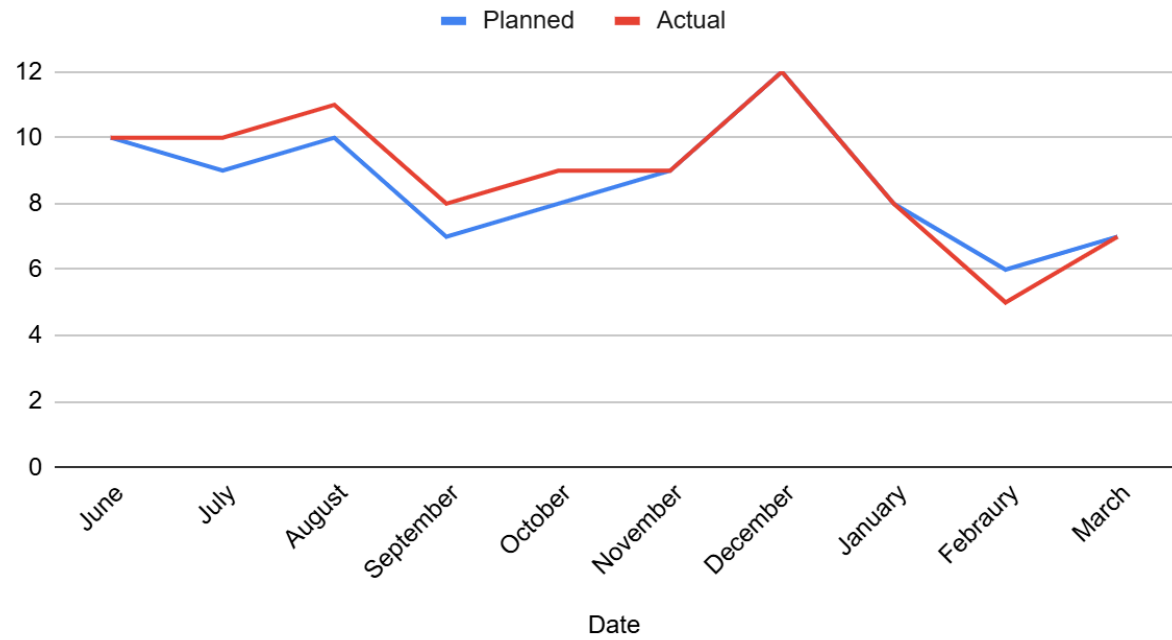
User story ID	User story	Tasks	Owner	Status	Estimated effort
US001	As an Administrator, I want to register a new student into the system with their personal, academic, and contact details, So that the student can be enrolled in courses and tracked	Register with mobile number/email id Enrolling for the classes	Monisha	Completed	15

	throughout their academic lifecycle.				
US002	As a Teacher, I want to mark daily attendance for students in my assigned class, So that their presence and absence are recorded accurately and reflected in reports.	Mark attendance	Monisha	Completed	5
US003	As a student I want a detailed description of my course So that I can choose my course	Add option to add description	Monisha	Completed	10
US004	As an Administrator, I want to enroll a student in available courses based on their grade or program, So that they can attend classes and access course materials.	Course enrollment	Monisha	Completed	8

	As an Administrator, I want to record and track fee payments for each student, So that I can monitor dues and generate financial reports.				
US005		Generate report	Monisha	Completed	10

Product burndown

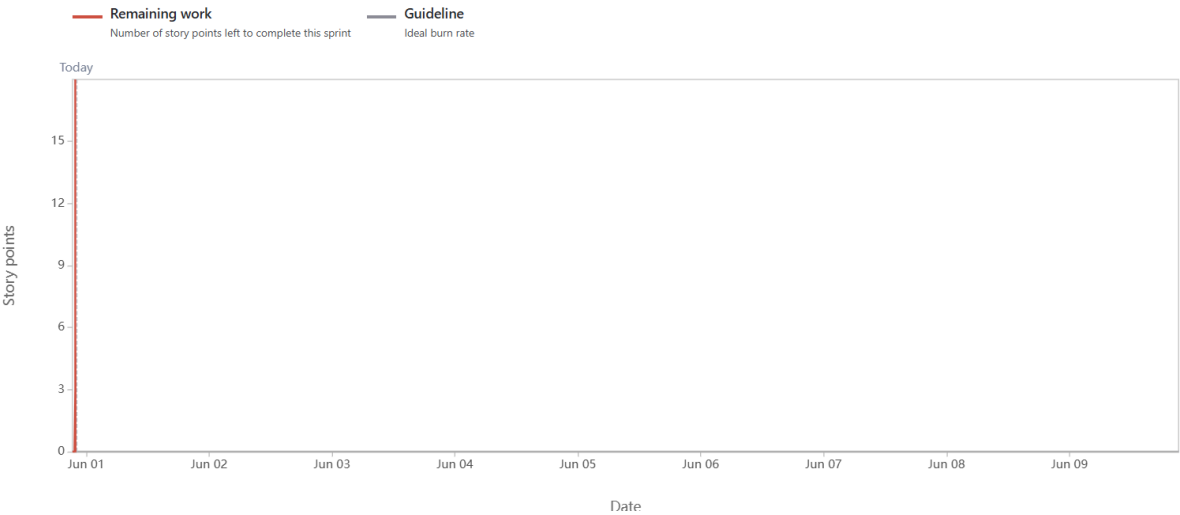
Planned and Actual



Sprint burndown

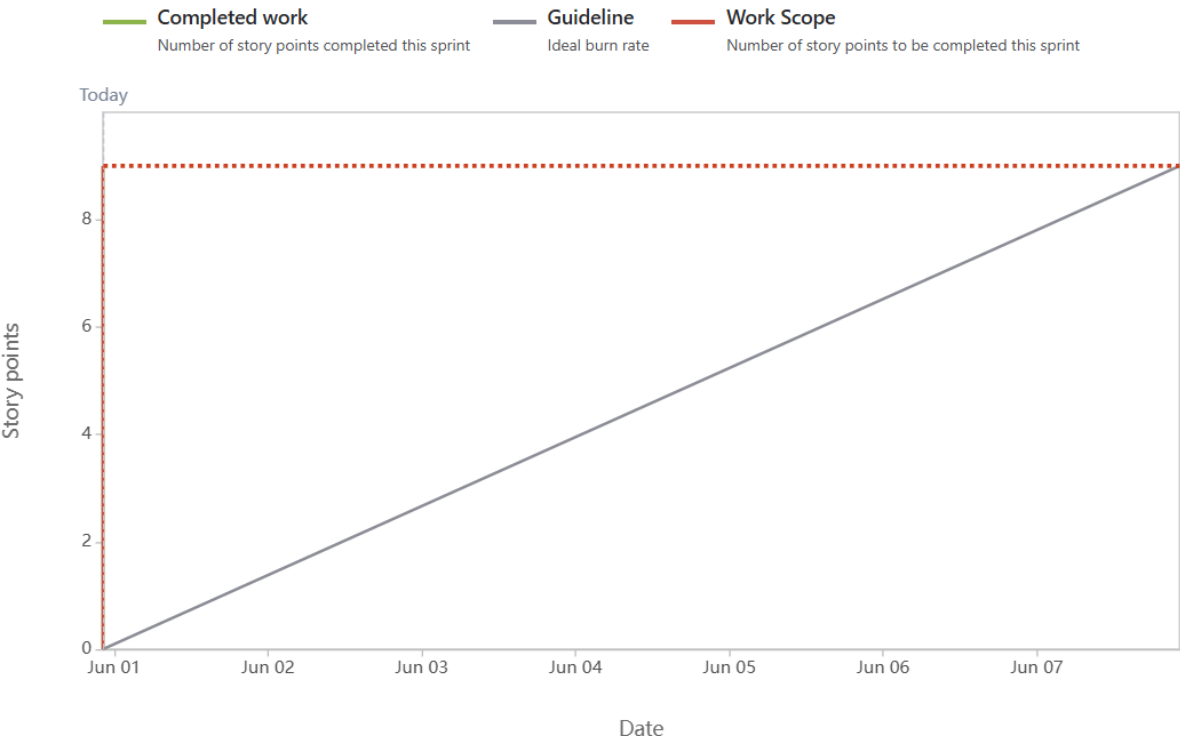
Date - May 31st, 2025 - June 9th, 2025

Sprint goal - create registration for the application



Date - May 31st, 2025 - June 7th, 2025

Sprint goal - create login



Document 6: Sprint meetings

Meeting Type 1: Sprint Planning meeting

Date	21.02.2025
Time	11.00 AM – 1.00 PM
Location	Microsoft Teams
Prepared By	David George
Attendees	Development Team, Scrum Master and Product Owner

Agenda Topics

Topic	Presenter	Time allotted
Plan next sprint	Product Owner	11.00 AM – 11.20 AM
Estimating stories	Scrum Master	11.20 AM – 11.40 AM
Clarify team availability	Scrum Master	11.40 AM – 12.00 PM
Go through backlog items for the next sprint one by one	Product Owner	12.00 PM – 12.20 PM
Agree on sprint goal	Scrum team	12.20 PM – 12.40 PM
Start next sprint	Scrum Master	12.40 PM – 1.00 PM

Other Information

Observers	Development Team
Resources	8
Special Notes	-

Meeting Type 2: Sprint review meeting

Date	29.02.2025
Time	11.00 AM to 12.00 PM
Location	Microsoft Teams
Prepared By	Product Owner
Attendees	Product Owner, Scrum Team, Scrum Master, Management, Customers

Sprint status	Things to demo	Quick updates	What's next
Closed	The initial goal of the sprint was	Accepted	Review what have been discussed
Closed	Requirements that were met during the sprint	Few changes were made	Making sure every we need any more requirement
Accepted	What went well or went poorly during the sprint	A couple of developers did not attend the meeting	Getting updates from everyone
Closed	Any adjustments that are going to be made moving forward	Took notes	Update the team

Closed	Adjustments were made	Detailed discussion about adjustments	Update the team
Closed	Feedback from any key stakeholders	Feedbacks were noted	Make use of the previous feedbacks to get in track
Closed	Closing information about the next sprint	Was very useful	Getting into a research on the closing information

Meeting Type 3: Sprint retrospective meeting

Date	22.02.2025
Time	11.00 AM – 11.45 AM
Location	Microsoft Teams
Prepared By	Scrum Master
Attendees	Development team, Scrum Master, Product Owner.

Agenda	What went well	What didn't go well	Questions	Reference
Coding	Good collaboration	Couldn't cope up with lots of meetings	Can we do more coding and less meeting?	May be timing of meetings can be reduced
User stories	8 out of 10 stories are done	A couple of stories were rejected by PO. Need to work on story	Do we need to work on story acceptance criteria?	All stories 5 points or less. Unless it is impossible

		acceptance criteria		
Bug fixing	The DB support team came through with a bug fix which unblocked a critical story	Too many in-sprint bug being found in story acceptance testing	How to avoid it in the future	Accelerate adoption of the auto unit test framework. At least 50% of all stories through this next sprint

Meeting Type 4: Daily Stand-up meeting

Question	Name/Role	Week "X" (from dd-mm-yyyy to dd-mm-yyyy)						
		Monday	Tuesday	Wednesday	Thursday	Friday	Saturday	Sunday
What did you do yesterday?	Developer 1	I worked on			Check ed current product metrics and	Completed the API for student registration.	Designed and develop	
	Developer 2	implementi	Worked on the user registration page	Went over some new wireframes with Daniel	recreate all dashboards	Integrated student profile update module with database.	d the attendance marking UI for teachers.	
	Developer 3	functionalit y	Tested the payment page	Had a sync-up meeting with Rakesh for the new marketing campaigns	Brainstorm ing meeting on how to boost user growth		Connecte d grade entry form with backend API.	
		Worked on fixing some bugs in the search feature	Created the view based on the table and moved the code from dev to QA	Created marketing ads images for Daniel	Created Marketing Landing			
		Created the procedures and tables in the dev environme						

		nt						
What will you do today?	Developer 1 Developer 2 Developer 3	I plan on working on the user registration page Planning to test the payment page Creating the view based on the table and move the code from dev to QA	Will go over some new wireframes with Daniel Will sync up a meeting with Rakesh for the new marketing campaigns Marketing ads images for daniel	Check current product metrics and recreate all dashboards Brainstorming meeting on how to boost user growth Marketing Landing page layout		Tie up loose ends with billing Create new POs Create new UI	Work on student dashboard UI (for timetable and notifications). Begin responsive adjustments for mobile view.	Start work on course enrollment validation s. Add unit tests for registration and profile modules
What (if any) is blocking your progress ?	Developer 1 Developer 2 Developer 3	I don't have any blockers at the moment I'm blocked by a dependency on the backend team to provide the database object Blocked by the source team to	Blocked by the source team Need to talk to Brandon about new landing pages I don't have any blockers at the moment	Don't have any blockers at the moment Blocked by Brandon on DB Don't have any blockers at the moment		1. Blocked by the source team 2. Need to talk to Brandon about new landing pages 3. I don't have any blockers at the moment	Waiting on updated API docs for exam schedule from backend team. Need DB export access to verify some report queries.	Need clarification on course prerequisites logic from the BA. Waiting on updated API docs for exam schedule from

		generate the files					backend team.	
--	--	-----------------------	--	--	--	--	------------------	--