SPMP
For
Syllabank

07/July/2025

Prepared By :

Chanpirun Srorn , Channon Teng , Vattanac Ny

Paragon International University
8 St. 315, Boeung Kok 1, Toul Kork, Phnom Penh

Revision History

Version	Date	Name	Description
0.1	28/06/2025	Chanpirun Scrom, Chanon Teng, Vattanac Ny	Initial draft: Goals, scope, Definitions, Assumption added
0.2	29/06/2025	Chanpirun Scrom, Chanon Teng, Vattanac Ny	Added Design Constraints, User Characteristics, and Product Environment
0.3	30/6/2025	Chanpirun Scrom, Chanon Teng, Vattanac Ny	Completed Nonfunctional Requirements and Potential System Evolution sections
0.4	1/7/2025	Chanpirun Scrom, Chanon Teng, Vattanac Ny	

Table of Contents

7. **Support Process Plans**
    7.1 Configuration-Management Plan
    7.2 Release & Deployment Plan
    7.3 Training & Knowledge-Transfer Plan
    7.4 Maintenance & Warranty Plan

1.Overview

1.1 Purpose and Scope

The purpose of this project is to enhance the existing **SyllaBank** system to comply with the new 2025 university syllabus policy. This enhancement focuses on expanding the structure and flexibility of syllabus input and preview features. It targets faculty members, department heads, and curriculum reviewers who rely on accurate and policy-compliant syllabi.

This section outlines the scope of the upgrade across the core system functionalities.

Task Management

Instructors will be able to manage their syllabi creation process by saving drafts, submitting for review, and receiving approval feedback. Reviewers will be able to mark review stages (e.g., “Needs Revision”, “Approved”) within the system.

File Storage

Syllabi data is stored in a structured format (JSON) within the database. Rendered versions (HTML or PDF) may be exported or cached temporarily for fast access. Version control will allow instructors to revert to previous drafts.

Scheduling

The upgraded schedule input introduces a three-level structure: week → day → hour, with input fields for topics, delivery methods, assignments, assessments, and CLO tags. This supports compliance with the new curriculum policy and enhances clarity in time allocation.

1.2 Goals and Objectives

The primary goal is to make SyllaBank compliant with the 2025 syllabus policy by supporting:

- A structured course schedule (week → day → hour)
- Dynamic section formatting (paragraph ↔ table)
- CLO–PLO mapping in learning outcomes

Objectives:

- Improve data accuracy and structure
- Reduce instructor workload through reusable templates
- Ensure rendered syllabi match new policy formatting standards

- Enable better analytics and curriculum tracking for higher people (Dean, HOD, Provoke)

1.3 Project Deliverables

- Enhanced Course Schedule Input Interface
- CLO–PLO Mapping System
- Dynamic Section Format switching
- Upgraded Preview & Print Renderer
- Generate Pre-data for syllabi

1.4 Assumptions and Constraints

Assumptions:

- All instructors have access to SyllaBank
- Existing syllabi are stored in a consistent format
- Users will adopt the new format

Constraints:

- Development must stay within the existing Laravel + Next.js architecture
- Limited team member availability

1.5 Success Criteria

- The team finished the primary scope of the project
- All the scope will be complete.

1.6 Definitions

This section provides definitions of key terms, acronyms, and abbreviations used throughout this Software Project Management Plan (SPMP). It includes both domain-specific concepts (related to syllabus management) and project management terms essential for executing and understanding this upgrade project.

General Terms

- **Academic Staff (Faculty)** – University staff involved in academic activities, including lecturers, heads of department (HoDs), deans, and the Provost's Office.
- **Actor** – A user or system that interacts with the Syllabus Management System (e.g., Student, Lecturer, Dean).
- **Approve Syllabi** – Final confirmation performed by the Provost's Office for syllabi to be accepted into the official repository.
- **Compile** – The act of collecting and organizing multiple syllabi for institutional use or external requests.
- **Dean (of Faculty)** – Oversees all departments within their faculty and coordinates with HoDs to ensure timely and accurate syllabus submissions.
- **Head of Department (HoD)** – Academic authority responsible for reviewing syllabi submitted by lecturers and providing initial validation ("vouching").
- **Lecturer** – Creates and submits syllabi for assigned courses; interacts with the system to edit, track, and respond to reviews.
- **May** – Indicates an optional feature or action. For example, "The student may request a public link."
- **Must / Shall** – Indicates a required condition or feature; a mandatory element of the system.
- **Non-affiliated Individuals** – External entities such as university admission officers or partner institutions who may request syllabi via secure links.
- **Provost's Office** – Academic leadership responsible for ensuring quality assurance, syllabus compliance with accreditation, and final approval.
  Request Form – A submission from students or academic staff to request access to or share syllabi.
- **Role-Based Access Control (RBAC)** – A system model where permissions are assigned to users based on their role (e.g., student, dean, admin).
- **Syllabus** – A formal academic document outlining course details, learning outcomes, grading, and content structure.
- **Student** – End-user of the system, primarily interested in accessing or requesting syllabi related to their enrolled courses.
- **Vouch Syllabi** – The process by which an HoD confirms that a submitted syllabus is acceptable for review by a Dean.

Abbreviations and Acronyms

- **ACC** – Accreditation Committee of Cambodia
- **API** – Application Programming Interface
- **CS** – Computer Science
- **DB** – Database
- **ESP** – English for Specific Purposes
- **HoD** – Head of Department
- **ICT** – Information and Communication Technology
- **IPO** – Input–Process–Output (model used to structure system functionality)
- **PDF** – Portable Document Format
- **PoC** – Proof of Concept
- **RBAC** – Role-Based Access Control
- **SRS** – Software Requirements Specification
- **UAT** – User Acceptance Testing
- **UI** – User Interface
- **UX** – User Experience
- **2FA** – Two-Factor Authentication

2. Startup Plan

2.1 Team Organization

Srorn Chan Pirun	**Project Manager**	Oversees timeline, task assignments, reporting, risk, and stakeholder communication
Chhanon Teng **Srorn Chan Pirun** **Vattanac Ny**	**Frontend Developer**	Implements the UI for schedule editor, formatting toggle, and preview
Chhanon Teng	**Backend Developer**	Handles API changes, data models, and migration logic

Vattanac Ny		
Chhanon Teng **Vattanac Ny** **Srorn Chan Pirun**	**QA / Tester**	Designs test cases, conducts unit/UAT testing, ensures release quality
Chhanon Teng **Srorn Chan Pirun** **Vattanac Ny**	**Doc & Support Lead**	Updates user guides and delivers training materials to instructors

2.1 Project Communications

Communication will occur regularly to ensure coordination and issue resolution. Tools and protocols used include:

- Weekly Team Meetings – Mondays, 1:00 PM via Google Meet
- Sprint Reviews / Demos – Every 3 day with Steering Committee
- Documentation Sharing – Google Drive
- Real-Time Chat – Telegram group for immediate coordination

2.3 Technical Process

The development will follow an iterative XP-inspired approach, blending structured planning with short, focused development sprints.

Lifecycle Highlights:

- 3-Week Sprints – Plan → build → test → review
- Early Prototyping – For editor UI and learning outcome mapping
- End-of-Sprint Reviews – Internal testing + academic feedback

- Final UAT – With real instructor users before deployment

2.4 Tools

Purpose	Tool	Notes
Code Repository	GitHub (Private Repo)	Feature branches + code review
Frameworks	Laravel (Backend), Next.js (Frontend)	Compatible with existing stack
IDE	VS Code / PhpStorm	Developer-preferred
Version Control	Git + GitFlow	Tag releases (vYY.MM)
CI/CD	GitHub Actions → DigitalOcean	Auto-deploy to staging
Database	PostgreSQL	Schema adjusted for new format
Testing	Playwright, Postman, PHPUnit	UI, API, and unit testing
Documentation	Google Docs, Markdown	End-user and technical

Communication	Telegram, Google Meet, Gmail	Informal and formal coordination

2.5 Project Estimation

User Story

User Story ID	As a...	I want to...	So that...	Why It Matters / Complexity Factors	Story Point
1	Lecturer	input week-by-week course schedule with nested days and hours	I can structure my teaching activities precisely	Complex UI structure, nested inputs, dynamic state	13
2	Lecturer	specify learning outcomes using a multiline text area	I can list multiple outcomes clearly	Simple UI upgrade, minimal logic	5
3	Lecturer	define delivery methods per day and hour (e.g., Lecture, Quiz, Lab)	I can explain what happens in each session	Nested under schedule, slight complexity	8
4	Lecturer	assign reading materials, assignments, and assessments per week	I can align them with delivery and outcomes	Input field extensions, tied to nested structure	5
5	Lecturer	use a pre-filled weekly template to speed up syllabus creation	I can save time while still complying with detailed structure	JSON injection + conditionally rendering defaults	5
6	Lecturer	copy previous weeks' data to create new ones with adjustments	I can avoid repetitive typing	Requires state copy, key updates, and conflict prevention	8
7	Lecturer	choose between paragraph or table view input for each section	I can work in the format I prefer	Requires toggle logic + dual rendering paths	8
8	Lecturer	preview nested course schedule rendering	I can confirm the structure aligns with	Requires complete parsing and layout	5

		before final submission	expectations	reflection	
9	Lecturer	have my syllabus data stored and loaded using JSON structure	I can benefit from flexible and structured data	Backend + frontend logic, affects all sections	13
Total					70

Team Composition:

- 1 Project Manager (PM)
- 1 Frontend Developer
- 1 Backend Developer

Based on the completed user stories, the total estimated effort is **70 Story Points (SP)**. Using a combination of Agile and RAD methodologies, the development will be carried out iteratively with a focus on rapid prototyping, early feedback, and continuous improvement.

Estimation on Team

Role	SP Done	Mandays	MD with Un	Rate per Ma	Total Cost
Project Manage	28	5.75	6.61	500	1653
Frontend	21	5.75	6.61	300	1322
Backend	21	6.9	7.93	400	2380
					5355

Estimation on Operational Cost

Item	Estimated Cost
Office	500
Utilities	50
Database Hosting	100
Load Balancer	50

We add the total combine we'll get **6055$.**

3. WorkPlan

3.1 Activities and Tasks

The SyllaBank upgrade is broken into four major phases: planning, design, development, and deployment. Each phase is divided into specific tasks to ensure structure, traceability, and delivery readiness.

Phase 1: Planning and Requirement Finalization

- Review updated new syllabus policy
- Analyze existing SyllaBank system capabilities and limitations
- Conduct meetings with instructors for feedback
- Finalize Software Requirements Specification (SRS)
- Define updated data structure for week-day-hour input and PLO mapping

Phase 2: Design

- Design dynamic input UI (table/paragraph toggle for each section)
- Design enhanced schedule editor (week → day → hour)
- Define PLO mapping table structure
- Wireframe preview and rendering layout

Phase 3: Development

- Implement schedule input module (frontend + backend)
- Implement PLO mapping feature
- Add toggleable formatting component (table ↔ paragraph)
- Refactor syllabus preview renderer to support deeply nested schedule
- Add export-to-PDF and print functionality

Phase 4: Testing & Deployment

- Unit testing and integration testing

- QA testing using real-world syllabus data
- Finalize user guides and quick-start instructions
- Deploy to staging and then production

3.3 Iteration Plans

Development will occur over 1 sprints, each lasting 2–3 weeks. Tasks are planned using a sprint backlog, and each sprint includes planning, development, review, and testing.

Iteration 1: UI Foundations & Structure

- Week-day-hour structure component
- Dynamic input fields
- Basic database adjustments

Iteration 2: Core Functionality

- Table ↔ paragraph toggle for all key sections
- CLO ↔ PLO mapping logic
- Preview module redesign

Iteration 3: Final Features & Review Flow

- Instructor save/submit/review flow
- Version control and PDF export
- Internal review and QA

Iteration 4 (Buffer):

- UAT revisions
- Documentation and handoff
- Bug fixing and polishing

4. Control Plan

4.1 Monitoring and Control

To ensure the SyllaBank project progresses smoothly and remains aligned with its goals, a structured monitoring process will be implemented:

- Progress Tracking: Weekly stand-up meetings will be held to review completed tasks and upcoming goals using a Kanban board (e.g., Trello or GitHub Projects).
- Milestone Reviews: Milestones such as "Frontend UI Completion", "Course Schedule Module", and "Syllabus Preview Renderer" will serve as checkpoints for progress validation.
- Deliverable Reviews: Each deliverable will be reviewed by peers and the team leader to ensure it meets the SRS and SPMP criteria.
- Issue Management: Issues encountered during development will be logged and prioritized using an issue tracker.
- Change Management: Any major scope or requirement change will be documented, evaluated for feasibility, and approved by all team members before implementation.

Project Implementation Date

- 20 / June / 2025 : Completed project charter (approved) and Discuss scope.
- 25 / June / 2025 : Initialize the project’s source code repository
- 30 / June / 2025 : Completed SDA documetation.
- 8 / June / 2025 : Completion of final coding phase and analysis comparing the original code with the newly developed version.
- 13 / June / 2025 : Completed SRS and SPMP documentations.
- 15 / June / 2025 : Checked and finalized code & documentations.

4.2 Project Measurements

Date	Sprint phase	Milestone	Status
20/June/2025	Sprint1:Initialization	Project Charter finalized and Scope Discussion Completed	Completed
25/June/2025	Sprint2:Setup phase	Initialized Git Repository and Set Up Project Skeleton	Completed
30/June/2025	Sprint3:Architecture Sprint	Completed SDA Documentation	Completed
8/July/2025	Sprint 4: Final Coding	Finalized Coding and Comparison with Original Source	Completed
13/July/2025	Sprint 5: Documentation Delivery	Finalized SRS and SPMP Documents	Completed
15/July/2025	Sprint 6: Wrap-up & Review	Reviewed, Cleaned, and Finalized All Code & Documentation	Completed

4.3 Risk Management Plan

4.4 Configuration Management Plan

4.5 Verification and Validation Plan