



CCS 3310 – Software Engineering Method

Lab Sheet – 03

Attendance Management System

22UG2 - 0004 Lahiru Dilshan

22UG2 - 0002 Supun Dilshan

22UG2 - 0035 Sahan Hansaja

22UG2 - 0009 Deemantha Rathnayaka

22UG2 - 0560 Adithya Ramanayake

Agile Software Development Method

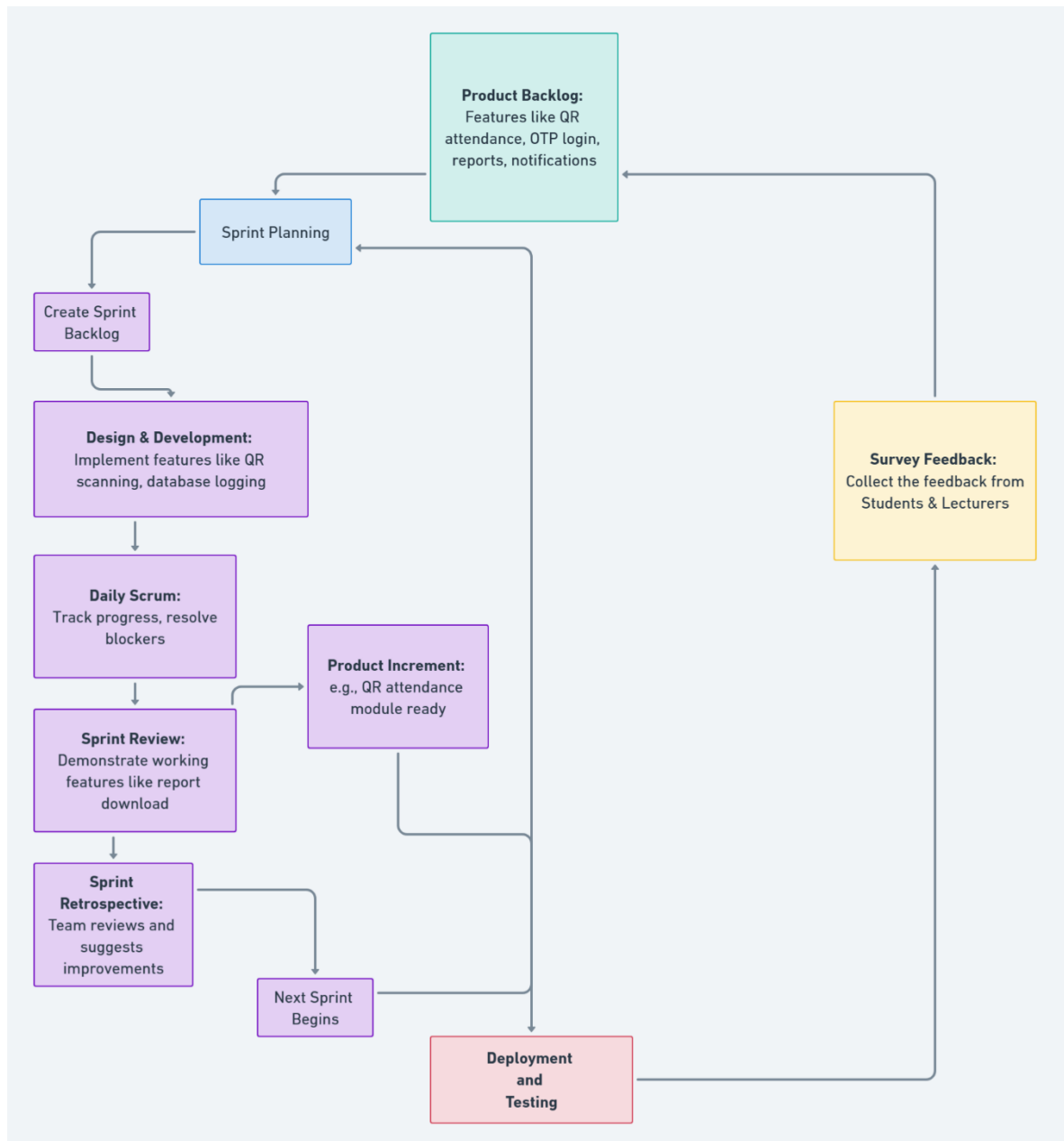
- Description For Agile SDLC:
 - Agile Scrum is an iterative and incremental software development framework that divides the project into short development cycles called sprints, typically lasting 1–2 weeks. Scrum focuses on continuous feedback, adaptability, and close collaboration between stakeholders and developers. Key roles include the Product Owner, Scrum Master, and Development Team, with regular ceremonies like sprint planning, daily stand-ups, reviews, and retrospectives.

- How this align with the Attendance Management System?
 - The Student Attendance Management System project includes dynamic and evolving requirements such as QR-based check-in, OTP support, real-time notifications, and admin control panels.
 - These features require ongoing refinement based on user feedback from lecturers and students. Scrum allows the team to deliver core functionalities early and iteratively improve the system based on actual classroom use.
 - Since the system targets multiple roles (students, lecturers, and admins) and needs flexibility in implementation and integration (e.g., QR scanning, notification logic), the Agile Scrum model offers the ideal balance between structure and adaptability.
 - The short sprints also align well with academic timelines, allowing incremental progress and regular evaluation.

Model Mapping to Your Project

Scrum Phase	Your Project Step
Product Backlog	Collect and list all required features: QR / OTP check-in, attendance report generation, student/lecturer login, notification system, and admin panel.
Sprint Planning	Select features for Sprint 1 (e.g., login system, user roles) and break them into tasks. Define sprint goals.
Sprint Backlog	Create a task list from selected Product Backlog items: UI for login, backend auth, role management.
Sprint (2 weeks)	Sprint 1: Implement login and authentication Sprint 2: QR check-in system Sprint 3: Attendance history and reports Sprint 4: Push notifications and admin controls
Daily Stand-ups	Share daily progress (individually or with team), identify blockers (e.g., QR logic, DB issues), and plan daily tasks.
Sprint Review	Present completed features to peers / instructor (e.g., demo QR check-in and view report), gather feedback.
Sprint Retrospective	Reflect on sprint process: What went well, what needs improvement, adjust team workflows or tech stack if needed.
Product Increment	After each sprint, deliver a working feature: a usable system that evolves with each cycle (e.g., working check-in system by Sprint 2).

Attendance Management System Model Diagram



Reflection & Questions

1. What challenges might you face using this model in your project?

One challenge we might face is sticking to the structure of regular sprints and stand-up meetings, especially with academic deadlines and personal time limitations. Since we're a small team (or sometimes working solo), it's not always easy to strictly follow Scrum roles like Product Owner or Scrum Master. Also, managing continuous feedback and making quick changes between sprints can get overwhelming without proper time management.

2. How would you adapt if requirements change mid-way?

If requirements change in the middle of the project—like if the client suddenly wants to add fingerprint attendance or another login method—I'd go back to the product backlog, re-prioritize the tasks, and include the new features in an upcoming sprint. That way, I won't disturb the current workflow, but I can still adapt and include the updated needs in the development cycle smoothly.

3. If time was limited, which model would you switch to and why?

If we were really short on time, we probably switch to the Incremental Model. It would let me build the most important parts first (like login and QR check-in) and then gradually add the rest if there's time. That way, we still have a working system to submit, even if it doesn't have every single feature.