Project Name: **Student Attendance Management System**
Responsible(s): Pratyush Kakkar (20022933) Gabriel Hernandez Gomez (20033649) Aigbomian Tehillah Eghonghon (20024813)

Document Versions and Revisions			
Date	**Comment**	**Author(s)**	**Version**
	Document creation	Luciana Nascimento	1.0

Student Attendance Management System

1 Project Vision

Goal: The goal of creating a student attendance management system is to make tracking and managing student attendance easier and more efficient for educational institutions. This system simplifies the daily recording of attendance by automating the process, making it accurate and convenient for both teachers and students. By consistently recording attendance data, it enables staff to quickly generate clear reports,which support daily management and allow for easy tracking of attendance patterns. Additionally, the system helps in checking student eligibility based on attendance, replacing traditional manual methods with a faster, more reliable digital solution.

Scope:

- Automate the process of recording student attendance
- Ensure accurate and reliable record-keeping
- Generate relevant and desired reports
- Keep the system simple and easy to use
- Provide communication between students and administration regarding attendance.

2 Requirements Specification

2.1 Functional Requirements

Id	Functional Requirement Description	State	Priority
FR01	The system shall allow teachers to mark the attendance of every student upon entry into class (Present / Absent)	**Accepted**	**High**
FR02	The system shall allow students, teachers, and administrators to generate reports of the student's attendance record.	**Accepted**	**High**
FR03	The system shall update the attendance record within 50 minutes of saving.	**Accepted**	**High**
FR04	The system shall allow administrators to designate the scheduled times for classes each day.	**Accepted**	**High**
FR05	The system shall allow administrators to add students and teachers to the System.	**Accepted**	**High**
FR06	The system shall flag students with attendance less than 80% and notify both students and administrators.	**Accepted**	**Medium**
FR07	The system shall restrict teachers from editing attendance 24 hours after the class, only administrators can make changes afterward.	**Accepted**	**High**

FR08	The system shall provide users with access through a web-based interface.	**Accepted**	**High**
FR09	The system shall allow students and teachers to view the scheduled class times.	**Proposed**	**Low**
FR10	The system shall allow students to request a justifiable absence leave.	**Proposed**	**Low**
FR11	The system shall allow students to report any discrepancies in their attendance records.	**Proposed**	**Medium**

2.2 Non-Functional Requirements

Code	Non-Functional Requirement Description	State	Priority
Performance and Robustness Requirements:			
NFR01	The system should support up to 3,000 simultaneous users without noticeable drops in performance.	Accepted	Medium
Maintainability Requirements:			
NFR02	The system shall support adding new features, including viewing schedules and reporting discrepancies in attendance records.	Accepted	Low
Portability Requirements:			
NFR03	The web application shall be compatible with major browsers, including Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, and Opera.	Accepted	High
Security Requirements:			
NFR04	The system shall enforce user authentication before granting access to view or edit student attendance records.	Accepted	High
NFR05	The system shall store user passwords securely using strong hashing algorithms to prevent unauthorized access	Accepted	High
NFR06	The web application shall use HTTPS to encrypt data between the server and users, ensuring secure communication.	Accepted	High

Usability Requirements			
NFR07	The system shall have an intuitive and user-friendly interface that is easy for all users to navigate.	Accepted	High
NFR08	The system shall provide a short user guide to assist new users in understanding the functionality.	Accepted	High
NFR09	The system shall enable users to update and view attendance in no more than four steps.	Accepted	Medium

2.3 Constraints

- Browser Support: It must be compatible with major browsers such as Google Chrome, Safari, Mozilla Firefox, Microsoft Edge, and Opera.

- Security Constraints: The system shall implement strict authentication and authorization protocols to restrict access to sensitive information exclusively to authorized users.

- Usability Constraints:The system interface must be intuitive and navigable for all user groups with minimal training.

- Scalability Constraints:The system must be designed to scale with increasing user loads and growing datasets, ensuring future expansions do not degrade system performance.

- Development Constraints: Time limitations for the development of version 1.0 may restrict the inclusion of advanced features, such as student self-check-in or discrepancy reporting.

- Legal and Compliance Constraints: The system must adhere to data protection laws such as GDPR to ensure compliance with privacy regulations.

2.4 Stakeholders

Students:

- Students are the primary users of the system, accessing it to monitor their attendance records.
- Their attendance data directly impacts their academic performance and compliance with institutional requirements.

Teachers:

- Teachers play a critical role in the system, using it to record daily attendance, generate attendance reports, and ensure the accuracy of student records for their classes.
- They are responsible for maintaining compliance with institutional policies on attendance tracking.

Administrators:

- Administrators oversee the management and operation of the system, including user role assignments, scheduling classes, and modifying attendance records when necessary.
- They ensure the system's data integrity and availability for all authorized users.

School Management:

- They use the insights provided by the system to make informed decisions on policy, resource allocation, and institutional performance.

System Developers:

- The developers and IT personnel are responsible for designing, implementing, and maintaining the system to meet the functional and non-functional requirements.

Support Staff:

- The support staff assists users in resolving any technical issues or difficulties encountered while using the system.

Compliance Officers:

- These stakeholders ensure that the system adheres to relevant data protection laws (such as GDPR or equivalent regulations).

2.5 User Stories

US for FR1:

As a Teacher, I want to record the attendance of each student as they enter the classroom so that I can maintain an accurate record of who was present on a daily basis. This will allow me to keep track of each student's attendance history for administrative purposes and ensure that records are up-to-date for all authorized users.

Acceptance Criteria:

- The teacher must log into the system with a valid username and password to ensure that only authorized personnel can record attendance.
- After logging in, the system should display a list of all students enrolled in the teacher's class.
- For each student, the teacher should be able to mark them as "Present" or "Absent," as they enter the class.
- The system will automatically save the teacher's input for each student and generate a timestamp to confirm when attendance was recorded.
- Once recorded, the attendance information should be accessible to all authorized users, including admins, other teachers, and the students themselves within the time frame.

US for FR2:

As a Student, I want to view my attendance records so that I can keep track of my attendance history, including detailed information for each class. This will help me stay aware of my attendance rate and understand how it may impact my academic performance.

Acceptance Criteria:

- The student must log into the system using their assigned username and password to ensure that only authorized users have access to their attendance data.
- The records should be detailed, showing the date, specific class or subject, and attendance status for each session (e.g., "Present," or "Absent,").
- The system should also provide an overall attendance percentage to give students a clear understanding of their attendance rate.
- The attendance data should be organized chronologically, allowing students to view their attendance history across different dates and classes.

US for FR3:

As an Administrator, I want to designate scheduled times for classes each day, so that a clear and organized timetable is available for all users in the system, enabling students, teachers, and other administrators to reference the daily class schedule as needed.

Acceptance criteria:

- The system interface should be intuitive and easy for administrators to use, allowing them to learn how to navigate and manage class schedules efficiently with only 30 minutes of training.
- Administrators must log into the system using a unique username and password to ensure they have the proper authorization to modify or create new class times.
- The system should provide options for administrators to input details such as the class name, instructor, and start and end times.
- Once the schedule is created or updated, it should be visible to all authorized users, including teachers, students, and other administrators.

US for FR4:

As an Administrator**,** I want to add students to classes, as well as add teachers to the system. Depending on whether the user is a student or teacher, the system should grant appropriate permissions—such as the ability to mark attendance for teachers or view attendance percentage for students. This will ensure that user roles are clearly defined and that access is appropriately restricted.

Acceptance criteria:

- The system interface should be intuitive and easy for administrators to use, allowing them to learn how to navigate and manage class schedules efficiently with only 30 minutes of training.
- Administrators must log into the system using a unique username and password to ensure they have the proper authorization to modify or create other users' inputs.
- Administrators' modifications and additions to the system shall be immediately reflected on the system for other users.
- Upon completing an update or addition, the system should provide clear feedback to the administrator to confirm that the action was successful.
- Any errors or issues should also be displayed with instructions to help administrators resolve them promptly.

US for FR5:

As a Student, I want to receive a clear notification in the system's interface when my attendance falls below 80% so that I am aware of my attendance status and can take corrective action if needed. Additionally, I want the administration to be notified if I have not logged into the system frequently enough to view these updates, ensuring that someone is aware of my attendance status and can follow up if necessary.

Acceptance criteria:

- The student must log into the system using their assigned username and password to ensure that only authorized users have access to their attendance data.
- When a student's attendance falls below 80%, the system should immediately generate a clear, visible notification on the student's interface, alerting them of their current attendance status.
- The interface should also allow students to dismiss notifications only after they acknowledge them, reducing the risk of missed warnings.

US for FR6:

As an Administrator, I want to restrict teachers from modifying student attendance records after 24 hours of the class session to prevent discrepancies between administration and teaching staff regarding attendance records. After this 24-hour window, only authorized administrative staff should have permission to modify attendance records to maintain data integrity and accountability.

Acceptance criteria:

- Teachers should be allowed to modify attendance records within a 24-hour window following the class session.
- After 24 hours have passed, the system should automatically lock attendance records for that session, preventing any further modifications by teachers.
- The system should maintain an audit trail for attendance records, logging the name of the person who made changes, the date, time, and type of modification.
- This audit trail should be accessible to administrators to review any modifications made by both teachers and administrative staff, providing transparency.

US for FR7:

As an authorized user of the system, I want to access the system remotely from any device, as it is a web-based interface, allowing me to manage or view necessary information without being restricted to a specific device or location. This flexibility will enable me to stay up-to-date with important information from any place with internet access.

Acceptance criteria:

- Users must authenticate with a secure username and password before accessing the system remotely to maintain security.
- Multi-factor authentication should be available as an optional or required feature for added security during remote access.
- The system should be fully accessible and responsive on various devices, including desktops, laptops, tablets, and smartphones, with optimal display on all screen sizes.
- The layout, style, and functionality should remain consistent across devices, so users do not need to learn different processes for different devices, enhancing ease of use.

3. Life-cycle model analysis and selection:

INCREMENTAL DEVELOPMENT VS AGILE (XP)

Extreme Programming (XP)

Extreme Programming (XP) is an Agile methodology aimed at providing high-quality software through rapid iterations and close collaboration with stakeholders. It highlights adaptive planning, regular releases, constant feedback, and a strong dedication to high coding standards. XP promotes the ability of teams to swiftly adapt to changes, integrating user and stakeholder feedback as development unfolds.

The advantages of XP include:

- Teams can swiftly react to shifting requirements and user input, making XP particularly suitable for projects with uncertain or evolving needs.
- Short development cycles facilitate frequent updates, enabling stakeholders to observe ongoing progress and provide continuous feedback.

Incremental Development

Incremental Development is a conventional software development strategy that breaks the system into smaller, manageable segments (increments), each contributing functionality to the final product. Each increment builds on what has been developed previously, allowing portions of the system to be delivered in stages. Although the entire system becomes available only when all increments are complete, each increment offers a fully functional subset of the system, permitting partial deployment and testing.

Key benefits of Incremental Development include:

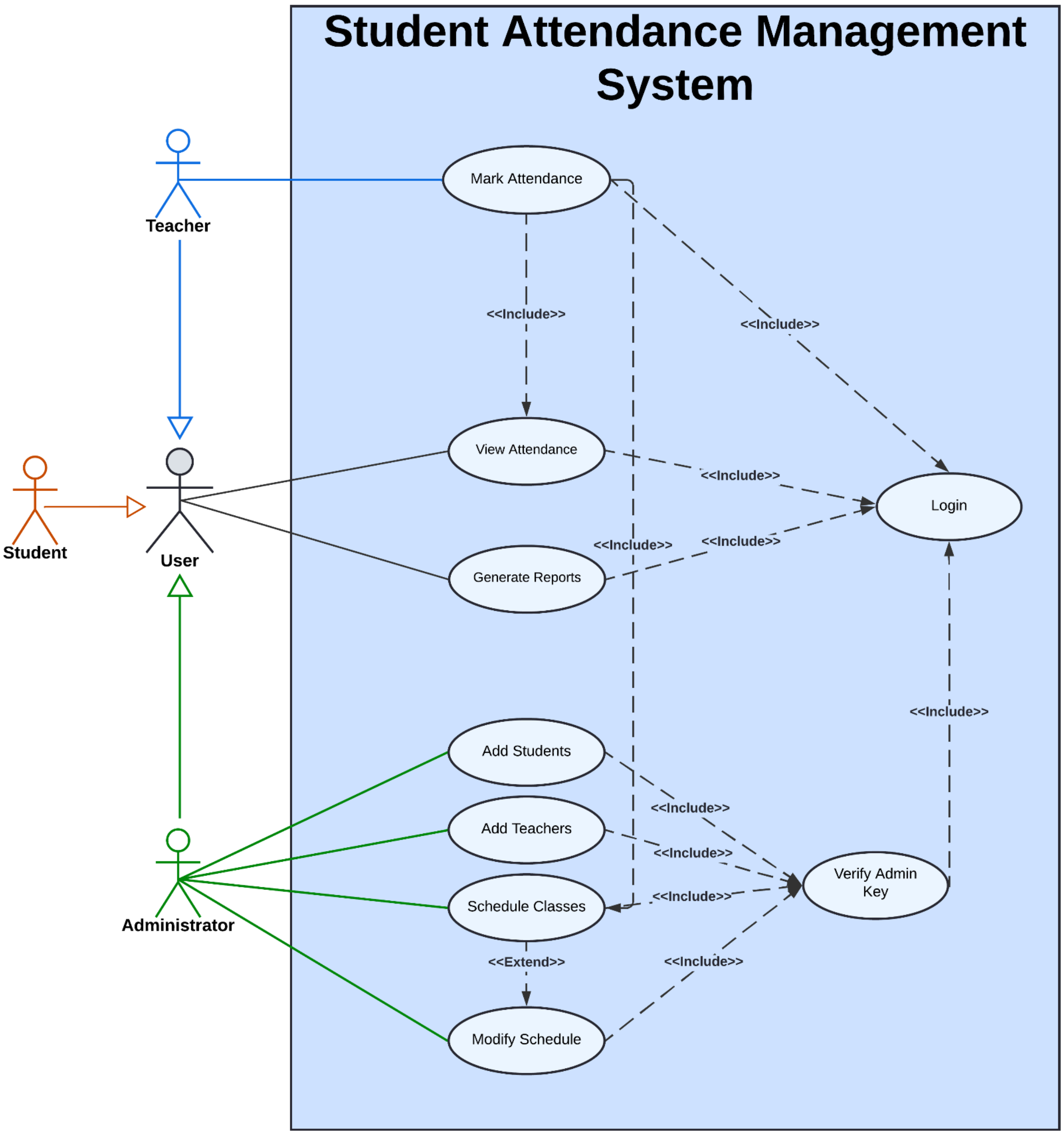
- By constructing the system incrementally, teams maintain better oversight over development schedules, allowing issues to be identified early within each increment.
- Development is divided into smaller sections, enabling more manageable distribution of resources and better budget management.
- The development team can focus on different areas of expertise with different releases
- Training can begin early, even though some functions are missing
- If a problem emerges, it affects only the current increment instead of the entire system. This compartmentalization lowers the risk and complexity associated with solutions.
- Provides opportunities for feedback and improvement throughout the development process.

After careful review of both life cycle models, Incremental Development comes as the more appropriate option for this system. Its iterative approach delivers progress through manageable phases, allowing for clear monitoring of each stage. This model also offers stability by managing risks linked to major changes late in the process, making it ideal for projects with well-defined requirements where stability and control are essential.

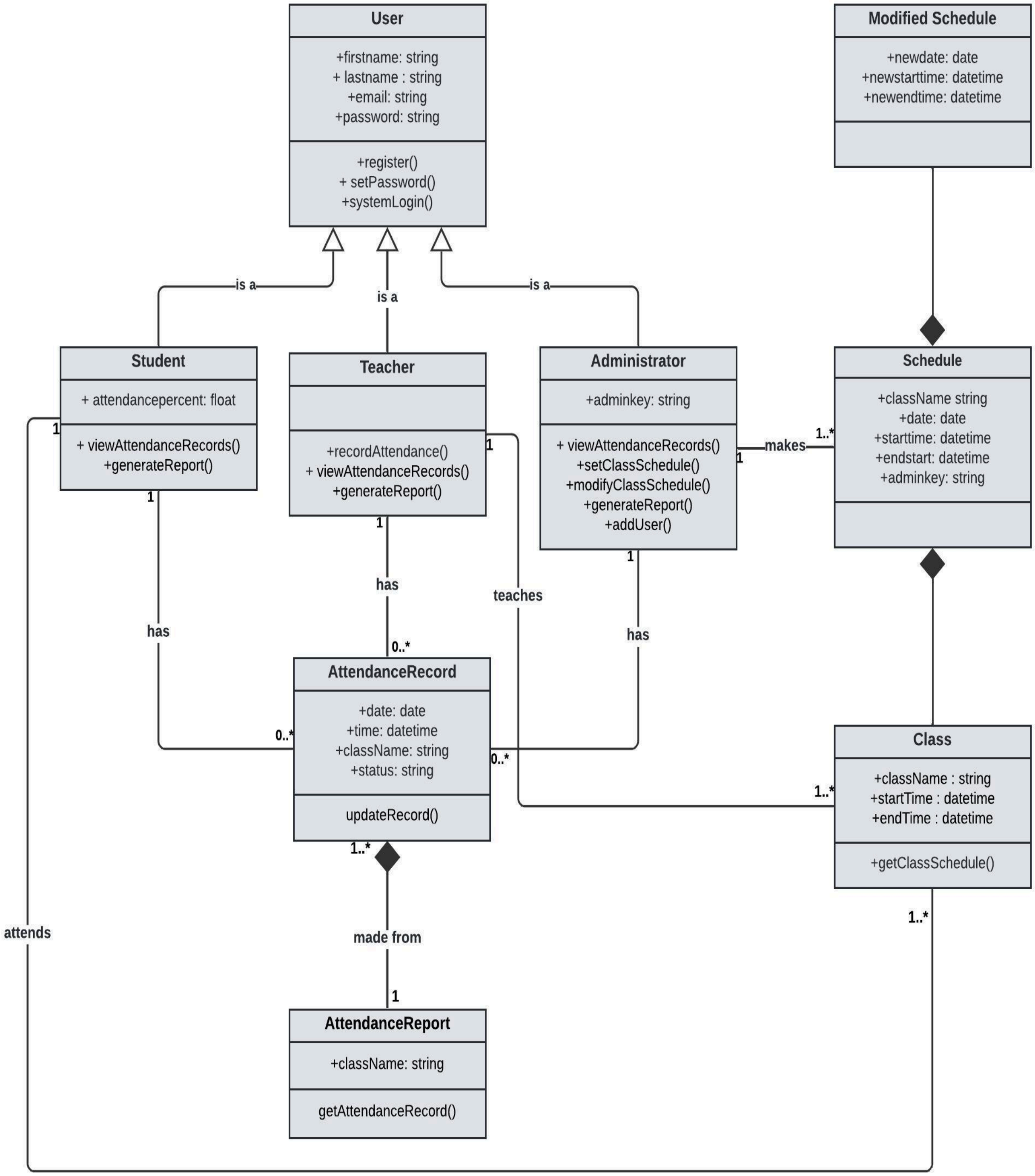
Although Extreme Programming provides significant flexibility and adaptability, the need for continuous customer input and rapid adjustments is less critical for this particular project. The Incremental Development model, focusing on delivering a system systematically and in stages, aligns effectively with the project's needs for predictable delivery and lowered risk, thereby making it the optimal choice.

4. Initial Domain Models

4.1 Use Case Diagram



4.2 Class Diagram



5. Release Plan

Version 1.0: Initial Release

Stakeholders: Administration Staff, Teachers, Students.

Goals:

- Implement tracking attendance and organizing course schedules.
- Enable core user interactions: teachers marking attendance, administrators managing schedules, and students viewing attendance records.
- Notify students whose attendance drops below 80%.

Features:

- **Teacher Role:**
  - Ability to mark students as **“Present” or “Absent”** at the beginning of each class.
  - Modify attendance discrepancies in a 24 hour frame after a class.
- **Administration Role:**
  - Manage class schedules and student rosters.
  - Manage attendance discrepancies, with the authority to edit or correct attendance records as needed.
- **Student Role:**
  - Limited access to view personal attendance records only, without editing capabilities.

Version 2.0: Enhanced Attendance Records

Goals:

- Provide more detailed attendance tracking, allowing teachers to mark students as “Present,” “Absent,” or **“Late.”**

Features:

- **Teacher Role:**
  - Enhanced attendance marking with an additional “Late” option, improving attendance accuracy.

Version 3.0: Student Self-Check-In and Schedule Visibility

Goals:

- Introduce self-check-in functionality for students using a physical or digital card at the beginning of each class.
- Enable all users (administrators, teachers, students) to view course schedules alongside attendance records.

Features:

- **Student Role:**
  - Self-check-in functionality via card tap, enabling students to record their own attendance as “Present” or “Late” if applied.
- **System-Wide Feature:**
  - Course schedule visibility for all users, integrated into the attendance management system.

Version 4.0: Full Attendance Tracking and Discrepancy Management

Goals:

- Enhance attendance tracking by requiring students to tap in **at the beginning and the end of each class.**
- Differentiate between partial and full attendance based on check-in/check-out times.
- Introduce a support system for students to report discrepancies or request justified absences.

Features:

- **Student Role:**
  - Two-step attendance verification with check-in at the start and check-out at the end of class, allowing tracking of full or partial attendance.
  - Access to a “Discrepancy Ticket” feature, enabling students to notify administration of any errors in attendance records.
  - A section to request justified absence leaves, which the administration can review and approve or deny.

Conclusion

This phased release plan for the system establishes a clear path for delivering a trustworthy platform designed to support teachers, students, and administrative staff in managing attendance with ease and accuracy. By releasing essential features in incremental versions, each phase introduces significant enhancements that progressively improve the system's functionality, user experience, and administrative control.

The initial version will prioritize core needs by equipping teachers with the ability to mark attendance and allowing students to access their records, while administrators can manage schedules and resolve discrepancies. Each subsequent version then builds upon this foundation. With Version 2.0's introduction of additional attendance options, such as marking students "Late," the system gains more detailed tracking capabilities. Version 3.0 goes further to empower students with self-check-in functionality and provides all users with visibility into course schedules, enhancing the system's utility and user engagement.

Finally, Version 4.0 elevates the attendance tracking process by enabling full or partial attendance based on entry and exit times, and it introduces essential tools for students to report discrepancies and request excused absences. This comprehensive set of features not only provides an accurate, real-time attendance record but also offers users the transparency and flexibility needed to manage their schedules effectively.

This plan also counts for potential user feedback after each release, enabling the team to respond to evolving user needs and implement improvements based on real-world usage. By following this structured release plan, the system will ultimately deliver a secure, user-friendly, and efficient solution that can handle the needs of educational institutions, improve attendance accuracy, and promote accountability among students. This iterative release approach ensures that each stakeholder group gains functionality that meets their needs.