

1-User Story

User Story 1: Student Scans QR Code

As a Student, I want to mark my attendance by scanning the lecture's QR code, so that I am accurately and quickly recorded as present for that class.

Acceptance Criteria:

- **Given:** I am logged into the EduTrack app and I am registered for the course.
 - **When:** I scan the valid, time-limited QR code for the lecture while I am in the correct location(lecture hall)
 - **Then:** The system confirms "Attendance Recorded," and my attendance status for that session is updated to "Present."
-

User Story 2: Student Uses Fingerprint

As a Student, I want to mark my attendance using my fingerprint on the classroom device, so that I have a reliable alternative if I forget my phone or the app has a problem.

Acceptance Criteria:

- **Given:** I have my fingerprint registered in the university system, and the lecture session is active.
 - **When:** I place my finger on the in-classroom biometric scanner.
 - **Then:** The system validates my fingerprint and records my attendance for that lecture.
-

User Story 3: Faculty Creates Session

As a Teacher, I want to create a new attendance session for my lecture, so that the system generates a unique QR code for my students to scan.

Acceptance Criteria:

- **Given:** I am logged into the EduTrack faculty interface and have selected my scheduled course.
 - **When:** I click the "Start Session" button.
 - **Then:** The system generates a unique, time-limited QR code and displays it for me to project in the hall.
-

User Story 4: Faculty Views Real-time Summary

As a Faculty/Teacher, I want to view a real-time attendance summary for my active lecture, so that I can see who is present and who is absent.

Acceptance Criteria:

- **Given:** I have an active lecture session running.
 - **When:** I navigate to the "Session Dashboard" page.
 - **Then:** The system displays a live-updating list of all registered students, showing their status as "Present" or "Absent."
-

User Story 5: Faculty Makes Manual Correction

As a Faculty/Teacher, I want to manually correct a student's attendance record, so that I can mark them as "excused" or fix an error.

Acceptance Criteria:

- **Given:** I am viewing the attendance report for a past lecture.
 - **When:** I select a student's name, choose "Edit Status," and change their status from "Absent" to "Excused."
 - **Then:** The system saves the change and requires me to enter a reason for the manual correction.
-

User Story 6: Academic Affairs Sends Warnings

As an Academic Affairs Department user, I want to identify and send warning notifications to students with low attendance, so that we can proactively support at-risk students and enforce university policy.

Acceptance Criteria:

- **Given:** I am logged into the Academic Affairs dashboard.
 - **When:** I run the "Low Attendance Report" for students with attendance below 75%.
 - **Then:** The system generates a list of all students who meet the criteria and allows me to send a pre-defined warning notification to them.
-

User Story 7: Academic Affairs Generates Final Report

As an Academic Affairs Department user, I want to generate an end-of-term attendance report for all classes, so that I have an accurate, official record for analysis and accreditation.

Acceptance Criteria:

- **Given:** The academic semester "Fall 2025" has officially ended.
 - **When:** I request the "End-of-Term Summary Report" for "Fall 2025."
 - **Then:** The system generates a comprehensive PDF report detailing final attendance percentages for every student in every course.
-

User Story 8: System Updates FMS

As the Faculty Management System (FMS) Admin, I want the EduTrack system to automatically send attendance data to the FMS, so that student attendance percentages are always up-to-date without manual data entry.

Acceptance Criteria:

- **Given:** A faculty member has concluded an attendance session in EduTrack.
- **When:** The session is closed and the data is finalized.

- **Then:** The EduTrack system's integration service automatically pushes the new attendance records to the FMS database, updating the relevant student records.
-

Checking Our User Stories (INVEST)

To make sure our user stories are good and easy to work with, we checked them with the **INVEST** model. It's a simple checklist to see if the stories are clear, useful, and ready for development.

I - Independent: Our stories don't depend on each other. A developer can build the "Student Uses Fingerprint" feature without waiting for the "Student Scans QR Code" feature to be finished. They are separate tasks.

N - Negotiable: These stories are a starting point for a conversation, not a final command. "Faculty Creates Session" says *what* the teacher needs (a QR code) but doesn't say *how* the button must look or where it must be. We can negotiate those details later.

V - Valuable: Every story clearly explains *why* it's needed in the "so that" part. For students: "...so that I am accurately and quickly recorded as present." For teachers: "...so that I can see who is present and who is absent." For admin: "...so that we can proactively support at-risk students."

E - Estimable: The stories are clear and focused. A developer can read "mark my attendance by scanning" and understand the task well enough to guess if it's a big job or a small one. The "Acceptance Criteria" make this even easier.

S - Small: Each story is one small piece of work. Instead of one giant, scary story like "As a teacher, I want to manage attendance," we broke it down into small, simple tasks like "Create Session," "View Real-time Attendance," and "Correct Attendance Record." This is much easier to build.

T - Testable: This is the best part. Our "Acceptance Criteria" for every story are the tests. For "Student Scans QR Code," we just follow the "Given-When-Then" steps. If it works as described, the story is 100% done.

Identifying External Events:

1-Event: A Student wants to mark their attendance.

.Use Case: Record Attendance via QR

2-Event: A Student wants to use the backup attendance method

Use Case: Record Attendance via Fingerprint

3-Event: A Faculty member wants to start the attendance process for their lecture.

- **Use Case:** Create Attendance Session

4-Event: A Faculty member wants to see who is present *during* the lecture.

- **Use Case:** View Real-time Attendance

5-Event: An Academic Affairs user wants a list of students with poor attendance.

- **Use Case:** Generate At-Risk Student Report

6-Event: A Faculty member needs to fix an error or mark an excused absence.

- **Use Case:** Correct Attendance Record

External Event checklist:

- ✓ An external agent wants to perform a transaction
- ✓ An external agent wants some information
- ✓ An external agent wants to change some data

Identifying Temporal Events:

1-Event: The academic term ends, and a final, official report is needed.

Use Case: Generate End-of-Term Report

Trigger: End of the academic term.

2- Event: A lecture session is finalized, and the data must be sent to the FMS.

Use Case: Push Data to FMS

Trigger: Upon finalization/closure of a lecture session.

Temporal Event checklist:

- ✓ Internal outputs needed
- ✓ Internal statements and documents
- ✓ Internal statements and documents
- ✓ External outputs needed
- ✓ Statements, status reports, bills, reminders

Event Type	Event (Description)	Stakeholder / Actor	Use Case
External	A Student wants to mark their attendance.	Student	Record Attendance via QR
External	A Student wants to use the backup attendance method.	Student	Record Attendance via Fingerprint
External	A Faculty member wants to start the attendance process for their lecture.	Faculty member	Create Attendance Session
External	A Faculty member wants to see who is present during the lecture.	Faculty member	View Real-time Attendance
External	An Academic Affairs user wants a list of students with poor attendance.	Academic Affairs user	Generate At-Risk Student Report
External	A Faculty member needs to fix an error or mark an excused absence.	Faculty member	Correct Attendance Record

Event Type	Event (Description)	Stakeholder / Actor	Use Case
Temporal	The academic term ends, and a final, official report is needed.	Academic Affairs / Time	Generate End-of-Term Report
Temporal	A lecture session is finalized, and the data must be sent to the FMS.	FMS Admin / System	Push Data to FMS

Brief use case description

Use Case	Brief Use Case Description
Record Attendance via QR	<i>Student scans the valid QR code, and the system validates the code, location, and time, then updates the student's status to "Present."</i>
Record Attendance via Fingerprint	<i>Student places their finger on the biometric scanner, and the system validates the fingerprint and records their attendance for the lecture</i>
Create Attendance Session	<i>Faculty/Teacher selects their course and clicks "Start Session," and the system generates and displays a unique, time-limited QR code.</i>

Use Case	Brief Use Case Description
View Real-time Attendance	<i>Faculty/Teacher navigates to the "Session Dashboard," and the system displays a live-updating list of students and their attendance status.</i>
Generate At-Risk Student Report	<i>Academic Affairs user runs the "Low Attendance Report," and the system generates a list of all students below the set attendance threshold</i>
Correct Attendance Record	<i>Faculty/Teacher selects a student's record and enters a new status (e.g., "Excused"); the system saves the change and requires a reason.</i>
Generate End-of-Term Report	<i>Academic Affairs user requests the "End-of-Term Summary Report," and the system generates a comprehensive PDF report of all final attendance percentages.</i>
Push Data to FMS	<i>The system is triggered when a faculty member closes a session; it automatically pushes the new attendance records to the FMS database.</i>

Uml:

