QR Code-Based System for Tracking Student Attendance

Changu Doreen Chankie-Madoda (Student Number: 7883857)
Dang Tuan Nguyen (Student Number: 8450535)
Ngo Minh Thu Le (Student Number: 8725111)
Dai Duong Phan (Student Number: 8786112)
Justin James Quinn (Student Number: 6453399)
Deepak Kumar Sunar (Student Number: 8381586)
Supervisor: Dr. Khoa Nguyen

Group 04: JC3DN

October 18, 2025

Abstract

Student attendance during lectures plays an important role in student development by facilitating active learning and improving understanding, and has been shown to produce positive academic outcomes. Mandating attendance appears a promising method to improve attendance rates and overall performance, however there are currently no effective and efficient methods of tracking student attendance. Traditional attendance tracking methods commonly rely on time-consuming methods, such as paper records or spreadsheets. While some alternative methods have been proposed, these still suffer from issues of proxy attendance, excessive time requirements, or significant processing required to extract meaningful data, and are difficult to implement in mixed-setting classes. This paper explores a novel approach of tracking student attendance utilising QR codes and an online platform. This platform, AttendEase, allows lecturers to create QR codes for their lectures which students can scan to mark their attendance. Utilising multiple checkpoints, GPS tracking and realtime data eliminates many of the limitations of other approaches. The platform generates meaningful statistics from collected data, allowing students to track their own attendance, and lecturers to see detailed attendance reports and statistics. These statistics allows better decisions to be made regarding attendance requirements. Overall, this project highlights the effective nature of this method in efficiently tracking student attendance, and with continued development could help to revolutionise student attendance tracking in tertiary environments.

Contents

1 Introduction

1.1 Background

Student attendance plays a critical role in higher education, as it is strongly linked to engagement, learning outcomes, and overall academic performance. Numerous studies highlight that consistent attendance contributes positively to student success, while irregular attendance correlates with poor academic achievement and increased dropout risks [3, 6]. Attendance data is also valuable for institutions, supporting quality assurance, funding decisions, and compliance with regulatory requirements. Beyond academic purposes, accurate attendance records provide important benefits in student welfare and safety. They allow institutions to quickly account for students in emergencies such as fire evacuations, natural disasters, or security incidents [10]. They also assist in monitoring vulnerable or at-risk students, enabling timely interventions and support services. Furthermore, accurate records contribute to transparency and fairness in administering attendance-based assessments or participation marks, helping to build trust between students and academic staff [14].

In professional and vocational programs, reliable attendance monitoring is even more critical, as accrediting bodies often require documented participation in labs, clinical placements, and workshops to ensure graduates meet industry standards [2]. In such contexts, attendance records serve not only as an academic requirement but also as evidence of competency, readiness for practice, and adherence to legal or ethical obligations. Despite its importance, attendance is often monitored inconsistently or inefficiently, especially in large classes where manual tracking methods create administrative burdens and lead to unreliable records [7, 10].

Across universities worldwide, reliance on outdated manual systems such as paper registers and spreadsheets continues to undermine efficiency and data reliability. These approaches are time-consuming, susceptible to human error, and open to manipulation through proxy attendance [7, 10]. In large cohorts, the process of calling names or passing attendance sheets consumes valuable teaching time, while the resulting records are prone to inaccuracies and loss. This challenge has been recognised globally, with researchers calling for modernised, technology-supported solutions to make attendance monitoring both efficient and reliable [9, 14].

At the University of Wollongong (UOW), attendance is not systematically monitored during lectures and is only inconsistently recorded in tutorials and laboratory sessions, often at the discretion of individual tutors. Even in cases where attendance is considered “mandatory,” records are commonly collected using manual methods such as sign-in sheets or spreadsheets, which are labour-intensive and vulnerable to errors and manipulation [10]. These challenges are not unique to UOW but are reflective of broader issues in higher education, where outdated methods hinder effective engagement monitoring and institutional accountability [6, 7, 10]. The COVID-19 pandemic further exposed these weaknesses, as the need for contactless processes made manual sign-ins impractical and highlighted the necessity for secure, efficient, and technology-driven alternatives [7].

1.2 Literature Review

1.2.1 Existing QR Code-Based Solutions

The integration of QR code technology into educational settings has garnered significant attention in recent years, particularly for its potential to streamline administrative processes such as attendance tracking. Numerous studies have explored the limitations of traditional attendance systems and the advantages of QR-based alternatives. Manual sign-in sheets and spreadsheet-based registers, while still prevalent, are widely recognised as inefficient, error-prone, and susceptible to proxy attendance [6, 10]. These methods also impose administrative burdens on instructors and reduce valuable teaching time [7].

To address these challenges, various technological solutions have been proposed. QR code-based systems have emerged as a popular choice due to their low cost, ease of deployment, and compatibility with smartphones [8, 14]. However, existing implementations often lack robust security features such as user authentication, geolocation verification, and real-time validation, which limits their effectiveness in preventing misuse [9, 11].

Several enhancements have been proposed to improve the reliability of QR-based attendance systems. For instance, Masalha and Hirzallah (2014) [7] introduced a system combining QR codes with facial recognition and GPS, achieving an 87% reduction in fraud. Similarly, Sweidan et al. (2021) [14] implemented time-limited QR codes with fingerprint verification, reporting 99.3% accuracy. These studies demonstrate the feasibility of integrating multi-factor authentication and contextual verification to enhance system integrity.

1.2.2 Existing Non-QR Code-Based Solutions

Beyond QR codes, alternative technologies such as RFID, NFC, GPS, Bluetooth beacons, and biometric systems have been explored. RFID and NFC offer high accuracy but are costly and require specialised infrastructure [2, 11]. GPS-based systems provide location verification but suffer from indoor inaccuracy and privacy concerns [13, 16]. Bluetooth and Wi-Fi-based systems offer passive monitoring but may lack individual-level verification [12]. Biometric systems, while highly accurate, raise ethical concerns and involve significant setup costs [4, 5].

1.2.3 Limitations of Existing Works

Traditional attendance systems in educational institutions, often paper-based or manually recorded, have been associated with challenges such as time consumption, inaccuracy, data loss, compromised confidentiality and opportunities for manipulation (Liew2021; Orah2023). In response, educational institutions have increasingly turned to digital solutions, among which Quick Response (QR) codes have emerged as one of the most promising. Originally developed in 1994 by Denso Wave for the automotive industry, QR codes gained global popularity with the rise of smartphones due to their ability to encode data compactly and be scanned instantly. In educational contexts, QR codes have been adopted for resource distribution, feedback collection, and attendance monitoring, offering advantages of low cost, speed, and ease of use [8, 14]. Their adoption was accelerated during the COVID-19 pandemic, where contactless attendance-taking became essential for maintaining safe and efficient classroom operations [7].

Beyond QR codes, other attendance-tracking technologies have been trialed with varying degrees of success. Manual registers remain common due to their simplicity but are slow, error-prone, and unreliable in large classes [10]. RFID and NFC systems provide fast and accurate identification but are costly to deploy and maintain at scale [2, 11]. Wi-Fi and Bluetooth beacon systems offer passive monitoring and higher acceptance but often lack individual-level verification [12]. GPS-based systems improve fraud resistance through geolocation checks, yet suffer from indoor inaccuracy, privacy concerns, and battery drain [13, 16]. Biometric systems, including fingerprint and facial recognition, provide high accuracy and fraud prevention but raise ethical concerns around privacy and come with significant setup costs [4, 5].

In the context of hybrid learning environments, the need for systems that can accommodate both in-person and online attendance has become increasingly important. The proposed system addresses this by incorporating dynamic QR code generation, geofencing, and secure authentication, while also allowing online students to check in via platform-integrated links. This approach ensures inclusivity and minimises opportunities for proxy attendance without requiring continuous tracking.

Recent research by Bekavac et al. (2024) [1] introduces the concept of 'quishing' QR code-based phishing attacks and highlights the importance of visual integrity in QR code design. Their study proposes 'Integrity by Design' strategies, such as embedding logos, using transparent backgrounds, and overlapping design elements, to make tampering visually detectable. While our system does not implement these design features directly, their findings support our use of time-sensitive QR codes and geolocation verification to ensure authenticity. Their user study further validates the importance of visual cues in promoting user vigilance, aligning with our goal of reducing proxy attendance and enhancing trust in the attendance process.

While each method addresses some challenges, none balance cost-effectiveness, scalability, usability, and security adequately for modern higher education needs. Many existing attendance systems lack secure authentication, making them vulnerable to proxy attendance [7, 11], and fail to integrate with Learning Management Systems [6], limiting their utility for educators. Additionally, few solutions offer hybrid compatibility to support both in-person and online learners [9, 14]. Some rely on costly infrastructure such as biometric or RFID systems [2, 5], while others fail to verify continuous student presence during lectures. Furthermore, design-based security enhancements (such as those proposed in QR code integrity research [1] are often overlooked.

1.3 Aims

The main objectives of this project are as follows:

1. To automate and streamline attendance tracking by implementing a dynamic QR code system that reduces time wastage, eliminates manual roll calls, and provides immediate confirmation for students and lecturers.

2. To enhance accuracy, security, and reliability through geolocation verification, real-time validation, encryption, time-limited codes, and multifactor authentication, thereby minimising proxy attendance and safeguarding user privacy.

3. To improve usability and adaptability by offering a seamless user experience supporting both in-person and online attendance tracking within hybrid learning environments, and enabling efficient reporting for academic and administrative purposes.

2 Developed System - AttendEase

2.1 Implemented Improvements to Existing Methods

This project (AttendEase), builds on the feasibility and acceptance demonstrated in earlier QR code studies while addressing persistent limitations identified in the literature. The proposed system enhances QR-based attendance by generating a unique, time-sensitive QR code for each lecture session, embedded with a timestamp that defines its validity period. Once expired, students can still scan it to be marked as late. To reinforce continuous engagement, the system introduces a mid-session button for the validation process where students must press or click on it to confirm their sustained presence.

Students' physical presence in the classroom is verified through geofencing combined with secure multi-factor authentication. To further strengthen data protection, the system implements HTTPS encryption with TLS 1.3 to secure all network traffic between client devices and the server. This protocol ensures that sensitive information such as login credentials, GPS coordinates, and attendance records is encrypted during transmission, preventing unauthorised access or eavesdropping. By safeguarding data in transit, the system reinforces user trust and aligns with best practices in secure web application design. The system also supports hybrid attendance tracking via platform-integrated links, allowing simultaneous monitoring of both in-person and online learners. To ensure inclusivity, it incorporates accessibility features such as manual attendance entry by lecturers for students unable to scan QR codes due to device limitations or visual impairments, platform-integrated links for online check-ins that bypass camera use, and support for multiple device types including laptops and tablets.

Real-time dashboards will provide lecturers with immediate insights into engagement, reducing administrative burdens while improving transparency. These features align with validated approaches in the literature and are fully implemented within the scope of the project, offering a secure, scalable, and transparent solution tailored to modern higher education environments.

2.2 Novelty

Although many existing attendance management tools address isolated aspects such as digital sign-ins, QR code scanning, or location-based verification, few offer a unified solution that integrates all these elements into a cohesive framework. The proposed platform is designed to overcome this fragmentation by delivering a comprehensive system that combines secure user authentication, precise geolocation tracking, and real-time data capture.

A key innovation lies in its dual QR code mechanism, which requires students to scan QR codes twice during a class (once at the beginning and once at the end) to ensure complete participation. These codes have a narrow time-window to use, eliminating opportunities for code reuse or unauthorised sharing. When a student scans a code, the platform validates their GPS location against a predefined location, ensuring both the authenticity of the sign-in and the student's physical presence. This dual verification process represents a marked advancement over other QR-based solutions, addressing the persistent problem of proxy attendance while maintaining a seamless user experience. By integrating these technical safeguards into a single, web-based interface accessible across devices, the platform introduces a genuinely novel approach to classroom attendance management.

Unlike other systems which tend to focus on tracking in-person attendance, this project focuses on a hybrid mixed-setting delivery for online and offline students. This aligns with current university trends where lectures are delivered both through in-person lectures and online streams of lectures made available to students online. As this system can be used for both modes of delivery simultaneously, this system can be used to accurately track and monitor students regardless of how they access live lectures. Additionally, this design also allows for additional students to be manually added by the lecturer so that students without access to devices can still be recorded, which is a major limitation of many other online attendance systems. These features make the proposed system much more applicable to a modern real-world university setting.

Combined with the automated attendance notifications to students, these features combined create a highly effective novel system that overcomes many weaknesses encountered when using traditional attendance-tracking methods.

These features come together to harmoniously to create a novel attendance-tracking system.

2.3 Use Cases and Impact

The proposed attendance verification system revolutionises educational and professional environments by automating the traditionally manual, error-prone process into a seamless, real-time solution that enhances accuracy, accountability, and efficiency across diverse settings, from higher education classrooms and corporate training programs to large-scale seminars and workshops. By eliminating time-consuming roll calls, it significantly reduces administrative burdens, freeing faculty and organisers to prioritise teaching, student engagement, and curriculum delivery while providing advanced reporting dashboards with actionable insights into attendance trends, participation patterns, and compliance statistics, empowering evidence-based decisions on resource allocation, student support strategies, and program design that ultimately drive improved learning outcomes and institutional performance. Students and participants experience a fast, intuitive sign-in process with minimal interaction, fostering inclusivity, minimising disruptions, and ensuring fair, transparent recognition of their presence in an equitable environment. Importantly, the framework is scalable and adaptable, enabling adoption across diverse contexts, from single classrooms to entire universities, and from small workshops to large corporate training programs. Its ability to grow with institutional or organisational needs makes it a sustainable long-term solution.

2.4 Challenges in Developing a New System

Despite its potential, several implementation challenges must be addressed to ensure equitable access and operational success. A major concern relates to accessibility, as the requirement to scan QR codes may disadvantage users with visual impairments, motor disabilities, or reliance on assistive technologies. In addition, students using older devices or those without functioning cameras may face barriers to participation. To mitigate these issues, the system incorporates a contingency feature allowing lecturers or administrators to manually record attendance for affected individuals, ensuring that technological limitations do not lead to exclusion. Verifying the physical location of remote students poses an additional difficulty. However, the platform's dynamic QR code generation and time-limited token authentication provide safeguards against proxy attendance to some extent.

Data privacy and security present another critical consideration. The system must safeguard sensitive information such as geolocation data, device identifiers, and user authentication details through robust encryption protocols, token expiration policies, and secure data storage practices that comply with applicable legal and institutional requirements. Institutions adopting this platform must also consider the operational aspects, including user training, device readiness, and provision of technical support to prevent disruption during live lectures. Integrating the platform with an organisation's existing systems, such as learning management systems (LMS), student information systems (SIS), or institutional single sign-on (SSO), may require additional development effort and technical coordination. Careful planning is necessary to ensure interoperability, minimise disruptions during rollout, and maintain consistency of records across multiple platforms.

3 Methodology

This section outlines the research and development methodology adopted for the QR Code-Based Student Attendance System. It includes the literature review process, the design and development approach, and the data collection strategies used to support system implementation and testing.

3.1 Research Methodology

A comprehensive literature review was conducted to understand the limitations of existing attendance tracking systems and to identify opportunities for innovation. Academic databases such as ProQuest, IEEE Xplore, and ResearchGate were used to source relevant studies. Keywords included: "QR code attendance", "multimodal authentication", "GPS-based student tracking", "RFID attendance systems", and "biometric verification in education".

3.2 Development Methodology

The project followed the Design Science Research (DSR) methodology, which emphasises the creation and evaluation of practical solutions to real-world problems. The DSR process involved:

1. Problem Identification: Manual attendance systems were found to be inefficient, error-prone, and susceptible to fraud.

2. Objective Definition: Develop a secure, scalable, and hybrid-compatible attendance system.

3. Design and Development: Build a web-based platform using modern technologies.

4. Evaluation: Test the system for accuracy, usability, and security.

5. Communication: Document and present the findings and outcomes.

The development process was structured using a Waterfall model, dividing the project into distinct phases: requirements gathering, design, implementation, testing, and deployment. Each phase was executed sequentially with clear deliverables and team responsibilities.

3.3 Development Process

In the project, development took a top-down development approach with higher-level aspects of the project being planned and implemented, and broken down to achieve the more detailed, functional components.

Throughout development, different group members worked together to create the different features of the project. During this process GitHub was used for code version control and to manage changes. This allowed group members to independently work on sections of code during development to implement different features, which after testing changes were committed to individual branches, and merged together when appropriate to implement these features and create the final project.

Microsoft's VSCode was chosen for development as it allows users to easily push, pull, and switch between GitHub branches, allowing the team to work on various branches and features throughout the course of the project.

XAMPP was also used to host a local copy of the designed database, so that changes could be made and tested locally on a real version of the database before being added to the GitHub repository.

UMLet software was used for conceptual database design to facilitate better cooperation and understanding between group members.

Google Cloud Platform (GCP) was used to deploy the project live for testing purposes in a production-ready environment. Additionally, a CI/CD pipeline was integrated via GitHub Actions to trigger automatic deployment of the latest code or new features, ensuring they could be tested seamlessly in a live, production-ready environment.

3.4 Data Collection

To support system functionality and testing, both simulated and real-world data were collected and created.

Simulated user data was created to represent students and lecturers, including names, email addresses, student/staff IDs, and course details. This data was used to test login, attendance recording, and dashboard features while maintaining privacy compliance.

Accurate GPS coordinates were manually collected from lecture halls and buildings across the University of Wollongong campus using GPS-enabled mobile devices. These coordinates were used to define geofencing boundaries for each venue, enabling the system to validate student presence during QR code check-ins.

Collected data was essential for validating geolocation-based attendance logic, simulating realistic user behavior, ensuring the system could differentiate between physical and remote attendance, and testing the hybrid attendance model. By combining simulated user profiles with real location data, the team was able to simulate the check-in process to gather data for testing, as well as create samples of real-world check-in data. This closely mirrored actual usage conditions, enhancing the reliability and credibility of the final system.

4 System Design and Architecture

4.1 Overview of System Design

The QR Code Attendance System is designed as a modular web-based application that supports secure, real-time attendance tracking for both in-person and online learners. It integrates frontend and backend components with a structured database and RESTful APIs, enabling dynamic QR code generation, geolocation verification, multiple authentication methods, and dashboard analytics. The system is built to be scalable, secure, and adaptable to hybrid learning environments.

4.1.1 Functional Requirements

The functional requirements that the system must meet are as follows:

- Generate unique QR codes for each lecture session with embedded timestamps
- Allow students to scan QR codes via mobile devices to mark attendance
- Record attendance in real time and distinguish between on-time and late arrivals
- Verify student location using geofencing to ensure physical presence
- The system shall allow login via email-password or Microsoft SSO with UOW email to integrate with UOW's current authentication method
- The system shall require students to check in at the start and end of each session to verify their presence
- Support hybrid attendance tracking for both in-person and online students
- Generate attendance reports and analytics for lecturers and students
- Send notifications via email to students once they reach the non-attendance/absenteeism threshold

4.1.2 Nonfunctional Requirements

The system should meet the following quality attributes:

- Security: Ensure data integrity, secure authentication, and protection against proxy attendance or other bad actors
- Usability: Provide a simple and intuitive interface for students and lecturers
- Portability: Be accessible via web browsers and mobile devices across platforms
- Reliability: Maintain high system uptime, support database backups, and handle concurrent usage
- Accessibility: Include features for students with limited devices or visual impairments

4.1.3 Key Features

Additionally, there are a number of key features for the project:

- Role-Based Interfaces: Customised portals for students and lecturers, offering tailored access to attendance tools, validation workflows, and reporting dashboards for intuitive and secure user experiences.
- Hybrid Attendance Compatibility: Supports both in-person and online learners through QR scanning and platform-integrated links, enabling inclusive attendance tracking across physical and virtual environments.
- Real-Time Dashboard Updates: Provides lecturers with live dashboards that refresh instantly as students check in, offering immediate visibility into attendance status and engagement levels.
- Advanced Analytics & Reporting: Generates detailed attendance reports and visual summaries to support academic and administrative decision-making with real-time data insights.
- Automated Notifications: Sends alerts to students approaching absenteeism thresholds, helping institutions intervene early and support student retention and welfare.
- Accessibility & Responsive Design: Designed to accommodate students with limited devices or impairments, offering manual entry options, camera-free check-ins, and compatibility across smartphones, tablets, and laptops.
- Venue Change Adaptability: Allows lecturers to update classroom locations mid-session, with the system dynamically adjusting geofencing parameters to validate attendance accordingly.
- Edge Case Resilience: Handles last-second scans, GPS fluctuations, and other anomalies using tolerance thresholds and fallback logic to ensure fair and accurate attendance recording.
- Integrated Support System: Provides students and lecturers with access to expedient assistance through in-app help forms, contact options, and troubleshooting guides. This feature ensures users can resolve issues quickly without leaving the platform, improving overall satisfaction and reducing technical barriers.

- Interactive User Manual: Offers step-by-step guidance, helping users understand how to use features such as QR scanning, geolocation validation, and dashboard navigation. The manual is searchable and context-aware, appearing dynamically based on user actions.
- Smart FAQ Section: Includes a searchable, categorised FAQ module that addresses common questions related to attendance logging, login issues, hybrid mode usage, and accessibility options.

4.1.4 Stakeholders identification

The stakeholder identification diagram in Figure 1 illustrates all key stakeholders involved in the QR Attendance System (AttendEase Platform) and their respective interests, motivations, and interactions with the system. The diagram serves to clarify the diverse expectations and dependencies among technical, academic, and business-oriented participants in the project.

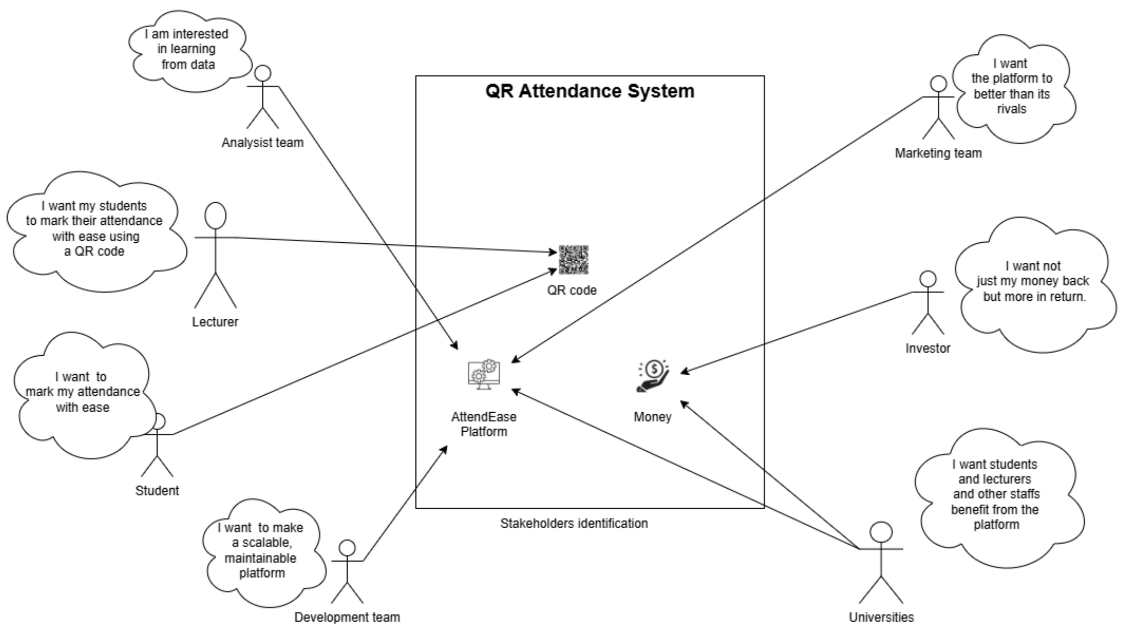

Figure 1: Stakeholder Identification Diagram for the QR Attendance System

The central component of the diagram is the *QR Attendance System*, also referred to as the *AttendEase Platform*, which facilitates attendance marking through the use of QR codes generated by the system. Each stakeholder is represented as an actor connected to the system, accompanied by a speech bubble describing their specific goals or interests.

1. **Lecturer:** The lecturer's primary goal is to allow students to mark their attendance easily through QR codes. They interact with the system to generate, display, and view analysis QR codes for their classes.
2. **Student:** Students seek a convenient and efficient method to record their attendance using their personal devices. Their interaction mainly involves scanning QR codes and verifying successful check-ins.
3. **Development Team:** This team is responsible for building and maintaining the AttendEase platform. Their objective is to ensure that the system is scalable, secure, and easy to maintain over time.

4. **Analyst Team:** Analysts are interested in the data produced by the system. Their focus is on analyzing attendance patterns, generating insights, and contributing to data-driven decision-making.

5. **Marketing Team:** This team aims to promote the platform and position it competitively in the market. Their goal is to ensure that the platform outperforms rivals and attracts a broad user base.

6. **Investor:** Investors provide financial support for the project and expect a profitable return. They are interested in the system's adoption rate, revenue generation, and long-term growth potential.

7. **Universities:** As institutional stakeholders, universities expect both students and lecturers to benefit from the platform. They seek improvements in attendance accuracy, record management, and administrative efficiency.

The *QR code*, representing the core mechanism for attendance marking, and the *Money*, symbolizing the financial investment and returns for using the platform.

4.1.5 Use Case Diagram

Figure 2 illustrates the main use cases of the QR Attendance System, focusing on the interactions between two primary actors: *Student* and *Lecturer*. The system facilitates attendance tracking and management through a QR code-based mechanism, allowing both actors to perform role-specific actions within the platform.

1. **Student:** The student actor interacts with the system primarily to manage and monitor their attendance. Their key use cases include:

    - *Log in:* Students authenticate themselves to access the attendance system securely.
    - *Check attendance:* Students scan QR codes during sessions to register their attendance.
    - *View attendance records:* Students can view their past attendance history, allowing them to track participation and compliance with attendance requirements.
    - *View analytics dashboard:* Students access an analytical view of attendance data for their attendance records.

2. **Lecturer:** The lecturer actor has more administrative privileges related to session and attendance management. Their use cases include:

    - *Log in:* Lecturers authenticate themselves to access the attendance system securely.
    - *Create QR sessions:* Lecturers generate QR codes corresponding to study sessions or lectures for student check-ins.
    - *Mark attendance manually:* Lecturers can record attendance for students who could not check in due to specific reasons.
    - *Send QR code via email:* Lecturers can distribute session QR codes to students via email for convenience.
    - *View real-time tracking:* Lecturers monitor student check-ins in real time to ensure attendance is accurately recorded.
    - *Generate reports:* Lecturers can produce attendance reports summarizing student participation across sessions.

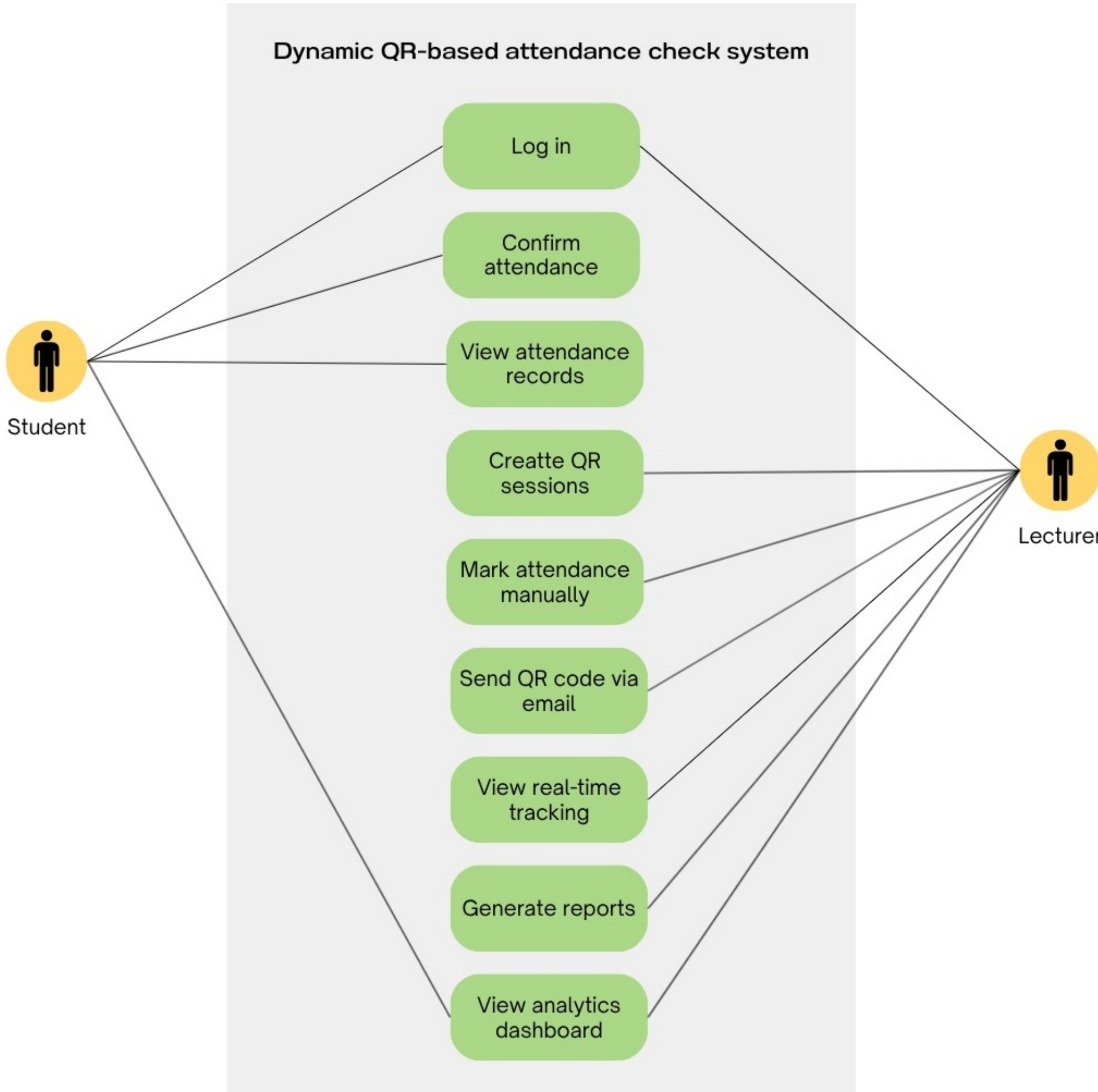

Figure 2: Use case diagram highlighting the features of our QR code-based check in system and the relationships both Student and Lecturer users have with them

- *View analytics dashboard:* Lecturers access an analytical view of attendance data, including statistics and performance metrics for decision-making and academic evaluation.

3. **Shared Use Cases:** Both students and lecturers share access to certain functionalities, such as the ability to *log in* and *view attendance records*, ensuring consistency and transparency in attendance data management.

4.2 System Workflow

4.2.1 Account Log-in

The login process is common to both student and lecturer users. As shown in Figure 3, there are two login options; organic (username and password) or Microsoft SSO login. If the user select the SSO login option, the user will be prompted to login with their Microsoft account. If the user successfully logs in, they will be taken to the resource they are trying to access. If the user unsuccessfully logs in, they will be prompted to try again.

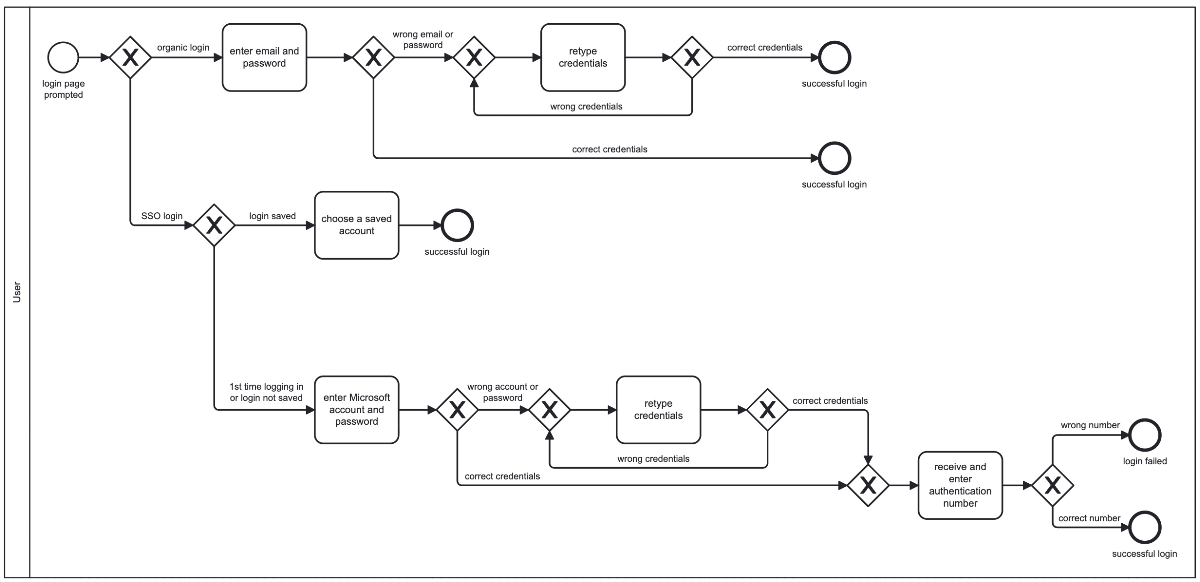

Figure 3: Diagram outlining the general process of a logged-out user logging in

4.2.2 Generate QR Codes

Figure 4 outlines the procedure for a lecturer to generate and configure a QR code. After logging in, the lecturer first navigates to the QR page and selects the relevant course. The lecturer then fills in the form they are presented with, including details such as room, time, location verification, and check-in windows. Once completed, the lecturer can click the 'Generate QR code' button to create the QR code. If changes are desired to the setting chosen, the lecturer can edit the form on the page and click the 'Update QR code' button to save their changes.

4.2.3 Share QR Code via Email

Figure 5 illustrates the process that lecturers follow to send a QR code via email to enrolled students. After generating the QR code, the lecturer can use the share button to automatically email it to students.

This function allows lecturers to share the code immediately after generating it, saving time compared to other alternative paths, such as manually downloading the code to embed it in presentation slides, or logging into the website during the lecture to display the code live. In cases where students do not have camera access to scan QR codes, lecturers need to send the link to those specific students. This is not only time consuming but also an unnecessary and avoidable disruption during class time.

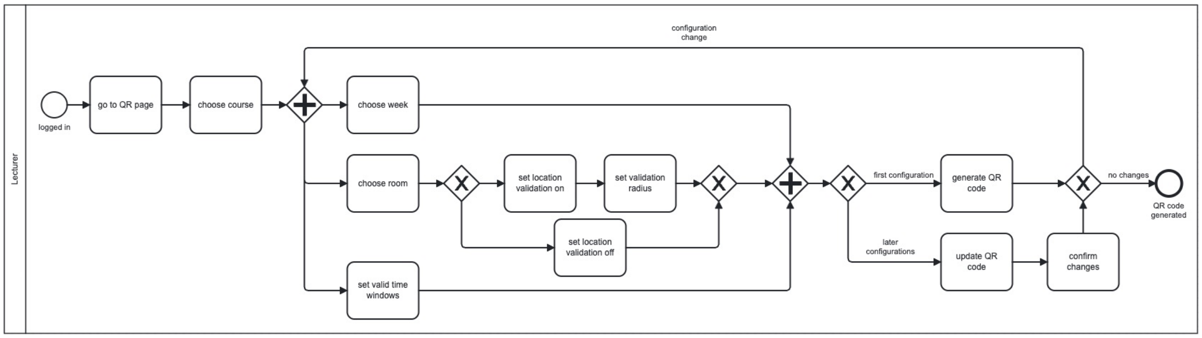

Figure 4: Diagram outlining the process of a lecturer creating a QR code

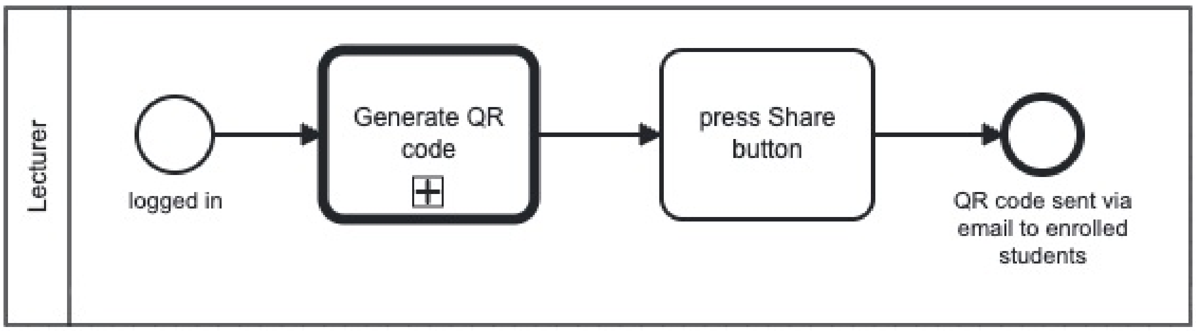

Figure 5: Diagram outlining the process of a lecturer sharing a QR code

4.2.4 Manually Mark Attendance

This process enables lecturers to mark student attendance manually in case students are not accessible to QR codes or links. After logging in and generating a QR code, the lecturer chooses to view the student list and clicks 'Check in' on students by request. Then, the lecturer confirms the action, ending with the student attendance being updated to 'Manual'. For manual attendance check, the student only needs to check in once for full attendance.

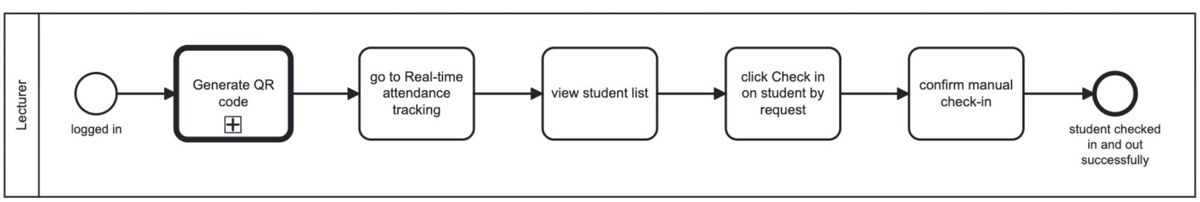

Figure 6: Diagram outlining the process of a lecturer manually checking in a student

4.2.5 View Real-time Attendance Tracking

Figure 7 illustrates the workflow for lecturers to access and monitor real-time attendance tracking. After logging in and creating a QR code, the lecturer proceeds to the attendance page, selects 'Live sessions', chooses the relevant subject, and presses 'View tracking', enabling live oversight of student participation. If any student checks in, the system records the attendance activity and updates real-time attendance statistics. The student name will then show up on the lecturer's tracking screen.

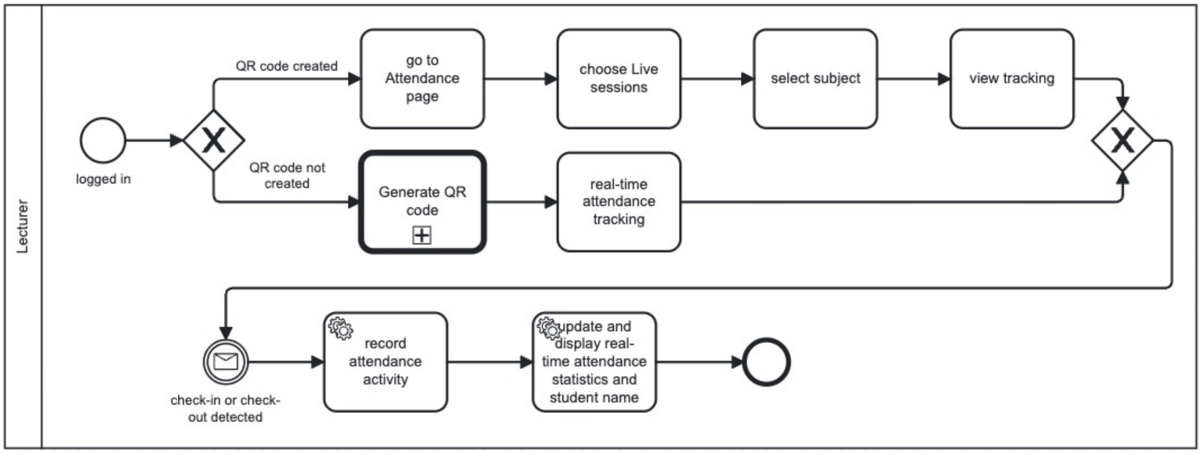

Figure 7: Diagram outlining the process of a lecturer viewing real-time attendance data

4.2.6 Attendance Confirmation

From the perspective of online and in-person students confirming their attendance, each take a very similar path as shown in Figure 8, with the addition of a step for online students, allowing them to be marked as online.

Regardless of check-in location, students start by scanning the QR code provided to them by their lecturer, using their devices. This QR code is generally provided to students via addition to lecture slides, although it may also be communicated via other means, such as email. Scanning this QR code redirects the user to a URL specific for each lecture.

After being directed from the QR code to the site, the user will be prompted to sign in with their account (if not already signed in), or will be directed straight to the page for that lecture (if student already signed in). Here, the user will then need to grant GPS location data permissions to confirm their location. The user then follows the prompts to submit their attendance.

Based on this GPS location, the system determines if the user is in person, or an online student. Students in the correct location (in-person students) will have their attendance confirmed. Online students however will be prompted if they want to confirm their attendance as an online student.

If a lecture has a second attendance confirmation, the lecturer will prompt students to sign in again. Students can then rescan the QR code, or refresh the page in their browser to repeat the attendance confirmation process again.

4.2.7 Student Users Viewing Records and Statistics

Similar to lecturers, students can also access a dashboard of their attendance within a course. Here they will be presented with a set of basic analytics for their enrolled subjects. The user can select a subject they wish to view specific analytics and information by clicking on it, or can click on “View Analytics” on the right-hand side of the screen to see more detailed overall analytics. This function allows students to identify any subjects at risk of not meeting the attendance requirements and better manage their study plan.

Figure 9 shows how students can access their attendance history. After logging in, the student navigates to the ‘Records’ page to view the complete attendance history. Filtering options are also available to help the student better manage their view, including courses and subject IDs, class types, weeks, campuses, buildings, and rooms.

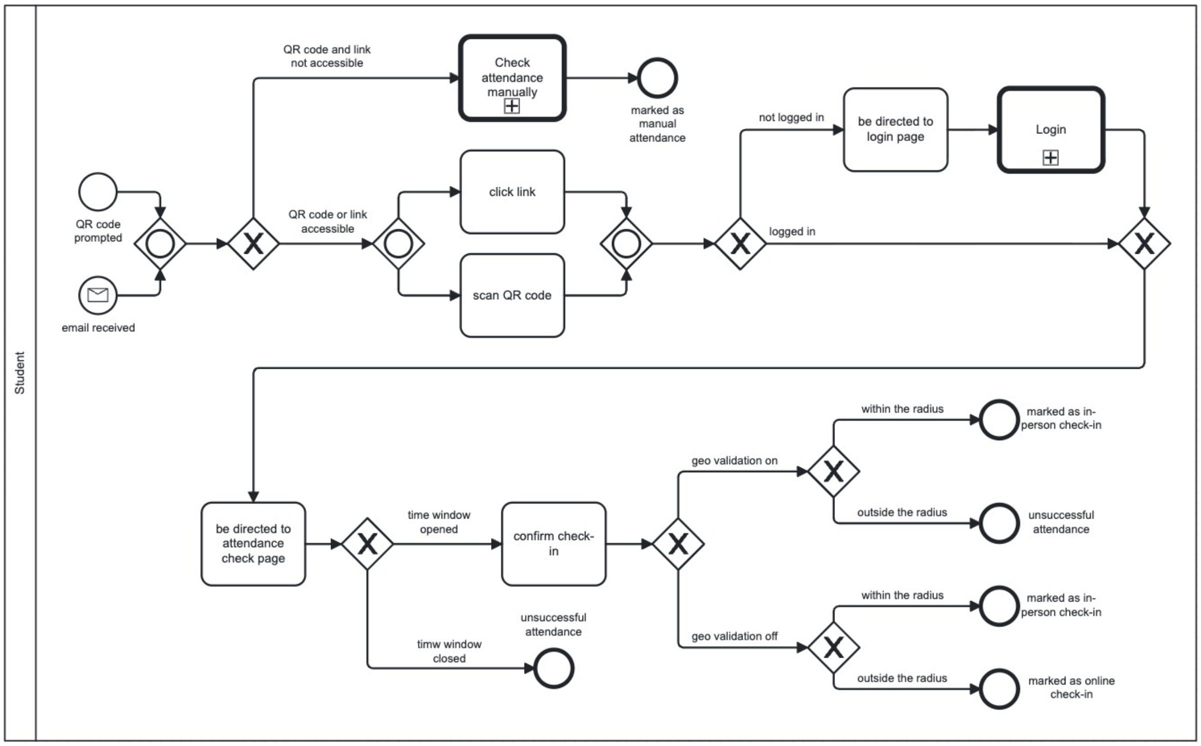

Figure 8: Diagram outlining the general process of a logged-out student user confirming their attendance during a lecture on the AttendEase. Flows for both in-person and online student attendance are included.

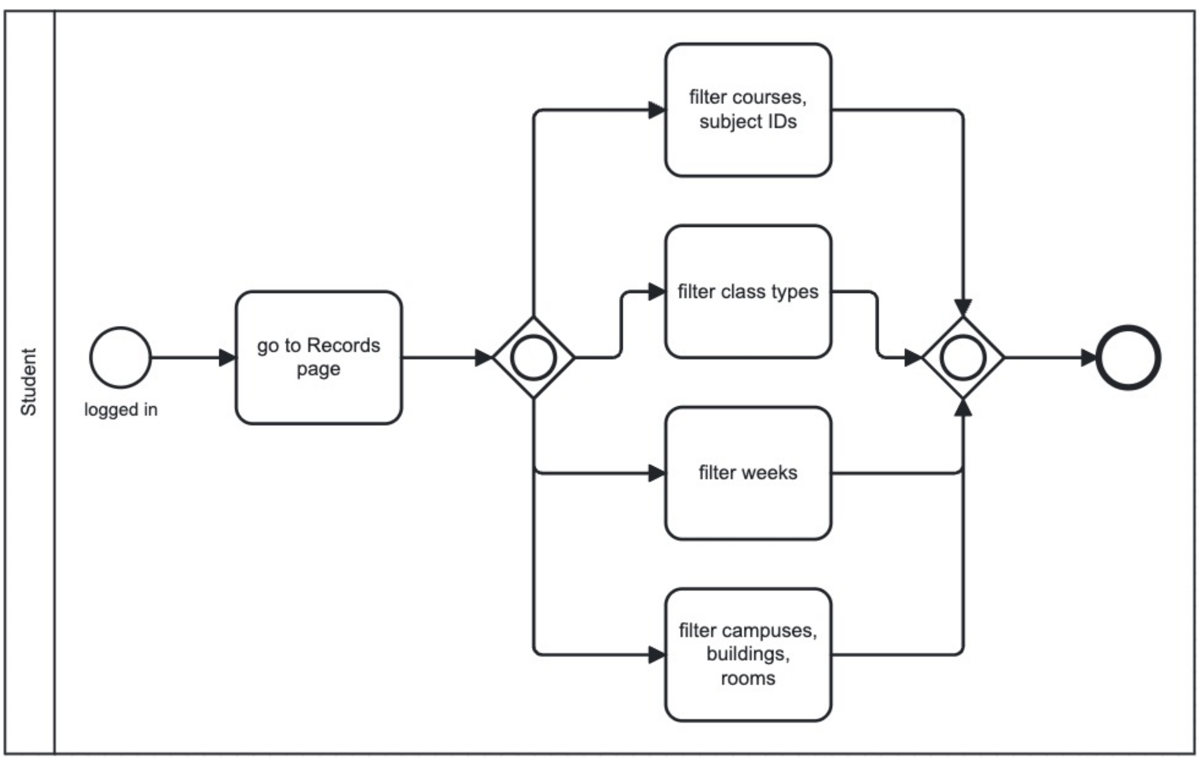

Figure 9: Process of a student viewing their attendance records

4.2.8 Viewing Student Statistics

On the AttendEase platform, lecturers can see advanced analytics of their classes. Lecturers will need to sign into the AttendEase platform with their account (if not already signed in). Once signed in, the lecturer can see their attendance dashboard with the statistics for their classes. The dropdown menu in the top right corner can be used to select a specific class to see statistics for. There are also several buttons available for the lecturer to refine their search.

Initiating from a logged-in state, the lecturer navigates to the home or dashboard page. Following this, the lecturers can interact with the dashboard by choosing a course, specifying a session type 'Lecture' or 'Tutorial', and selecting an analytics type. These options enable customised data visualisation, allowing the lecturer to filter and refine the displayed metrics and facilitating informed decision-making through structured data exploration.

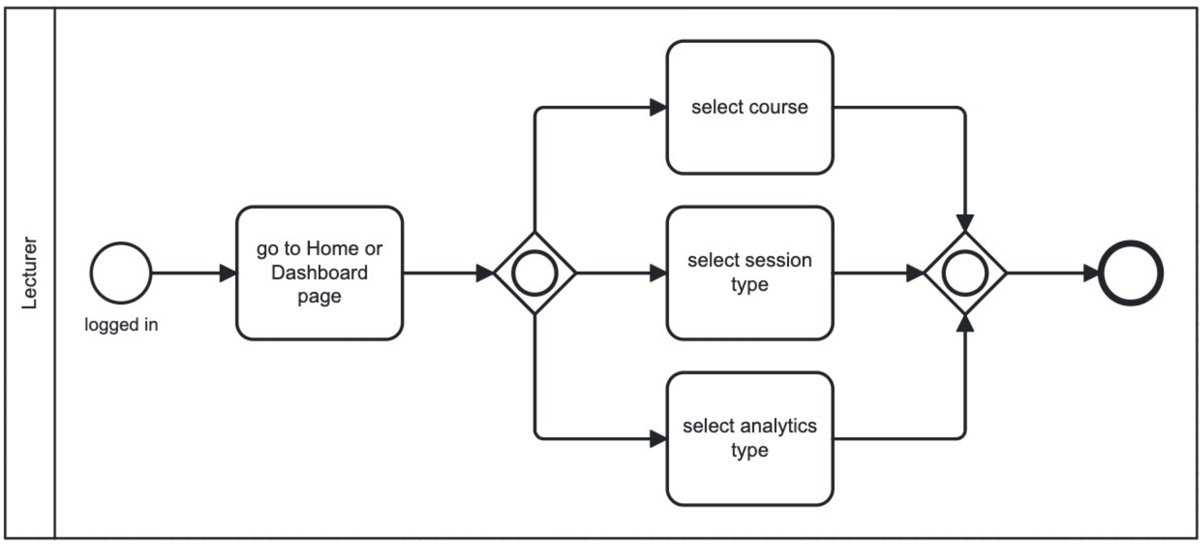

Figure 10: Diagram showing the process of a lecturer viewing student analytics/statistics for classes

4.2.9 Generate Analytics Reports

Apart from viewing analytics, lecturers can also generate a report and send it via email, as shown in Figure 11. Commencing from a logged-in state, the lecturer also navigates to the home or dashboard page and then accesses the 'Email report' section. A parallel gateway then branches the flow into concurrent activities to gather requisite parameters: choosing the report type, selecting the date range, specifying the course, and entering the email address. After filling out all those fields, the lecturer proceeds to send the report to the specified email address. The lecturer can use this function to send warnings to students at risk of not meeting attendance requirements.

4.3 Database Schema

During the course of the project, the group created a Unified Modeling Language (UML) class diagram highlighting the tables required for the AttendEase database, and their relationships, as shown in Figure 12.

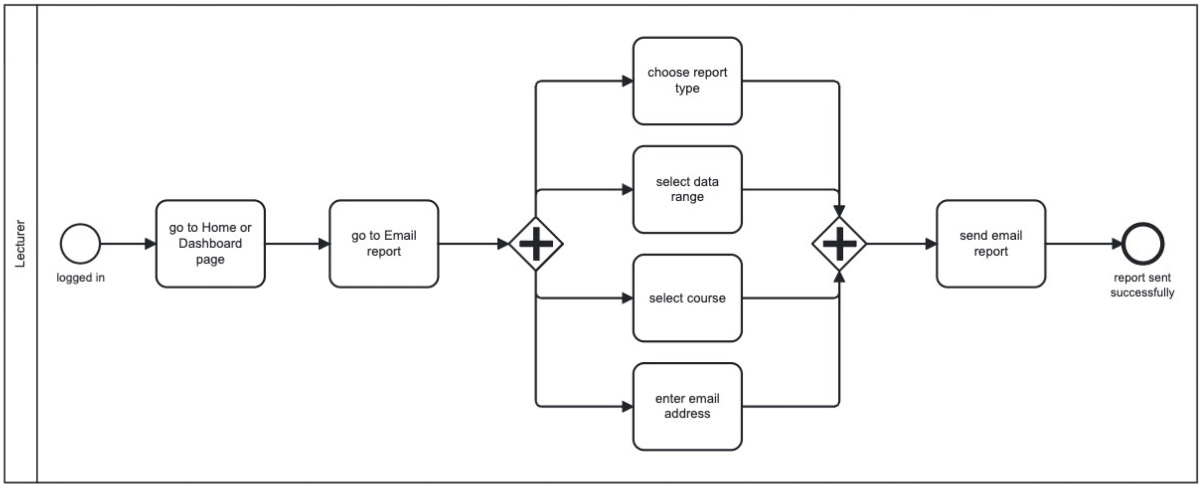

Figure 11: Diagram showing the process of a lecturer generating analytics reports

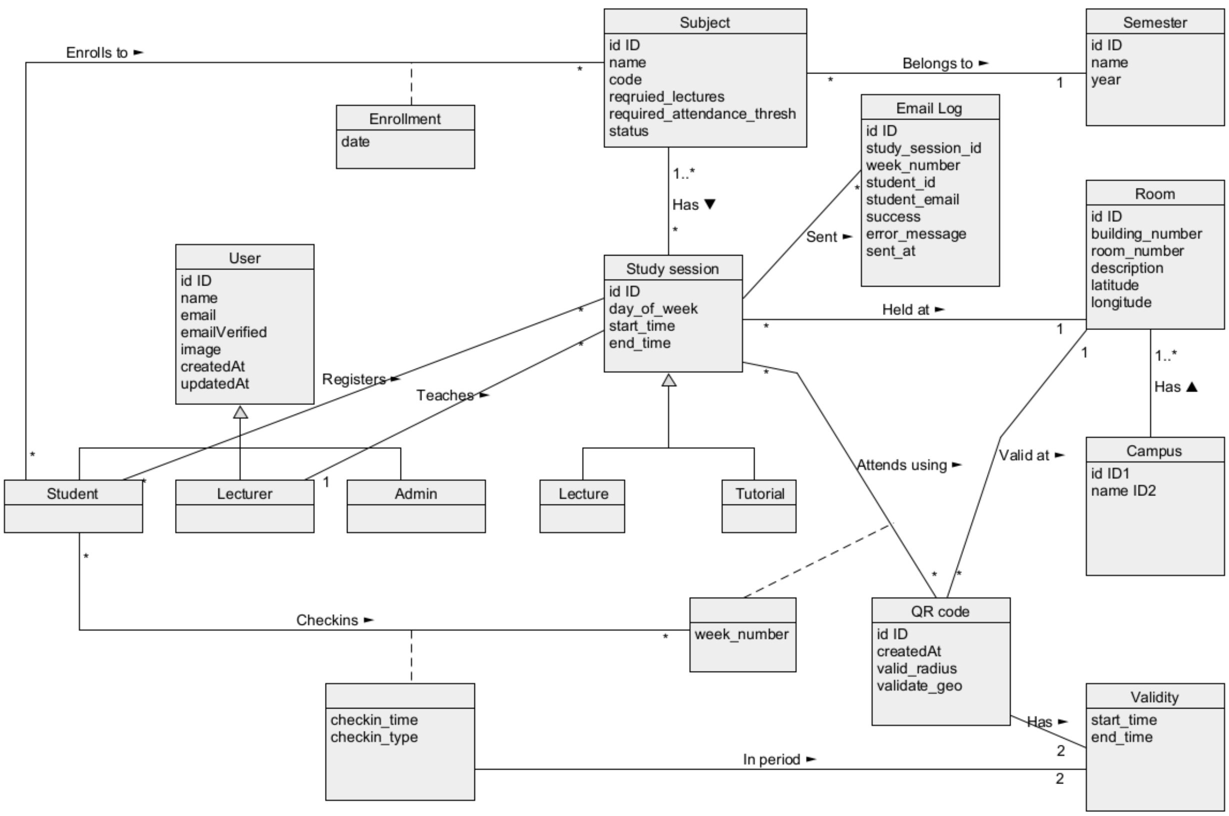

Figure 12: Diagram showing the finalised conceptual design for the AttendEase database

4.3.1 System Scope and Integration

This relational database, built using MySQL, serves as the foundational data layer for a comprehensive attendance management ecosystem that spans multiple operational domains within educational institutions. The system integrates with authentication services through the better-auth framework, providing secure user management capabilities that support role-based access control for students, lecturers, and administrative staff. The academic management component handles the complex hierarchical relationships between campuses, buildings, rooms, subjects, and semesters, enabling institutions to organise their physical and academic resources effectively. The QR code generation and validation subsystem manages the creation of time-sensitive, location-specific attendance tokens that can be distributed to students during class sessions. Geospatial capabilities are leveraged through latitude and longitude coordinate storage, enabling location-based validation with configurable radius parameters. Temporal data management is supported through datetime fields with automatic timestamping and interval-based calculations for validity periods. The attendance tracking module captures and validates student check-ins, incorporating geospatial verification to ensure location compliance. Additionally, the system supports multi-modal attendance recording, accommodating in-person, online, and manual attendance scenarios to adapt to different teaching environments and exceptional circumstances.

4.3.2 Business Rules and Constraints

The database implements numerous business rules and constraints that reflect the operational realities of educational attendance management. The enrollment system ensures that students can only register once for each subject. QR code validation is enforced through a database trigger that restricts each QR code to two validity periods per class, one at the beginning and one at the end, ensuring accurate tracking of full class participation. Location-based validation is achieved through configurable radius parameters that define acceptable proximity to designated rooms. Time-based constraints ensure that attendance can only be recorded during valid periods. The system supports flexible attendance requirements through configurable thresholds that can be set per subject, enabling different programs to maintain their specific attendance standards. The study session scheduling system prevents conflicts through unique constraints while supporting various class formats including lectures and tutorials.

4.4 System Architecture

The system architecture of the AttendEase application, illustrated in Figure 13, depicts the overall structure of the deployment and interaction between different system components. AttendEase is a web-based QR code attendance management system built using the Next.js framework and integrated with a MySQL database hosted on Google Cloud SQL. The architecture emphasizes scalability, automation, and secure database connectivity.

The source code of the Next.js application is maintained on a GitHub repository. Whenever the source code is updated, a Continuous Integration and Continuous Deployment (CI/CD) pipeline is triggered using GitHub Actions. The pipeline automates the process of building, and deploying the application.

During the deployment process, GitHub Actions build the Next.js source code into a Docker image. This Docker image encapsulates the entire application along with its runtime dependencies, ensuring consistent behavior across different environments. Once the image is successfully built, it is pushed to the Google Cloud Container Registry, which serves as a secure storage for containerized images.

After the Docker image is stored in the Container Registry, Google Cloud Run automatically pulls the image and creates a container instance. Google Cloud Run is a fully managed compute platform that runs containerized applications in a serverless environment. It handles the scaling of container instances based on incoming traffic and ensures high availability.

Once deployed, Google Cloud Run exposes a unique HTTPS endpoint for the application. This URL allows external users to securely access the AttendEase website through the web browser. The platform automatically manages SSL/TLS certificates, enabling encrypted communication between clients and the application.

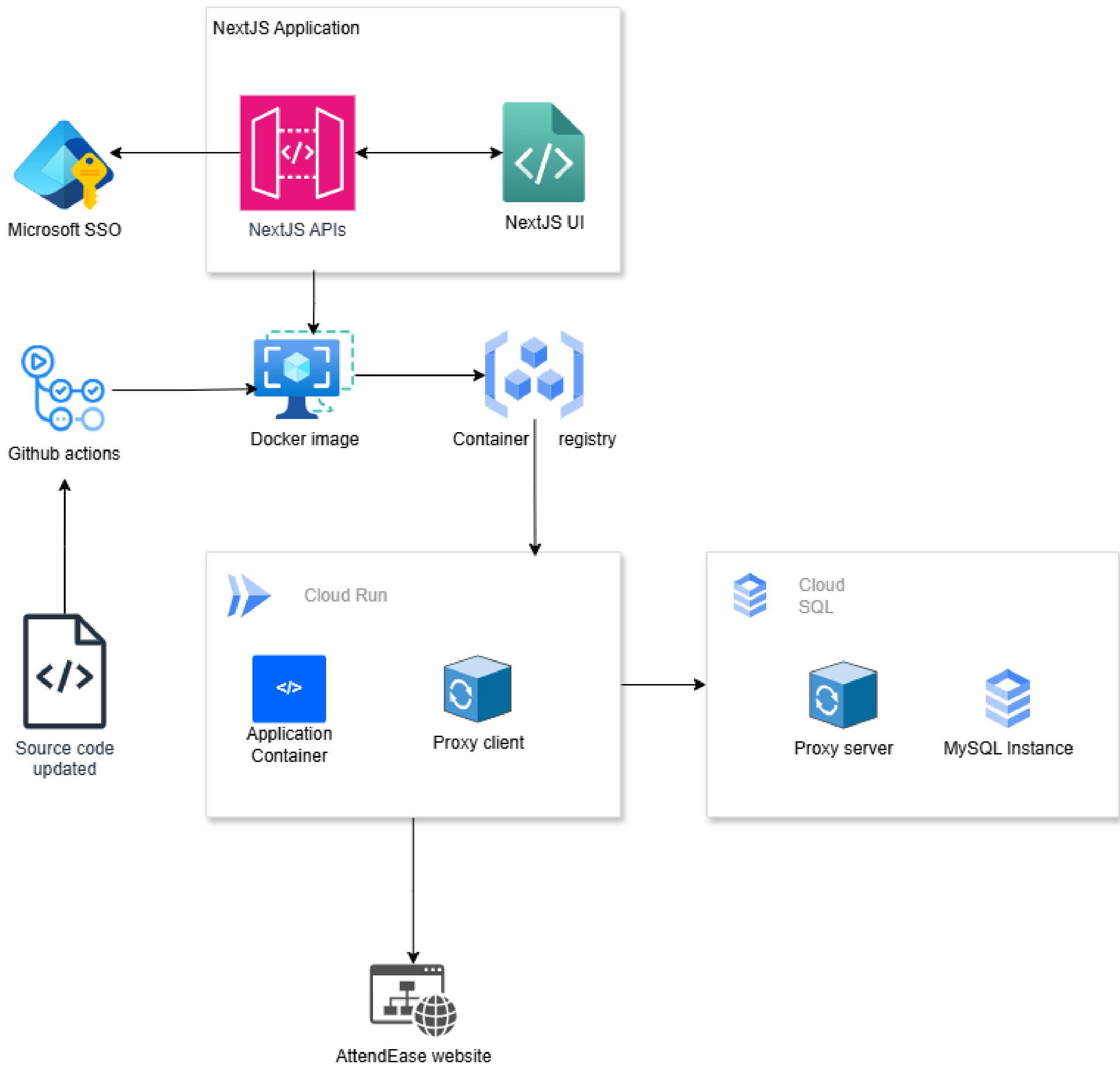

Figure 13: System Architecture of AttendEase

The AttendEase application relies on a MySQL database hosted in Google Cloud SQL. Instead of directly exposing the database to the public internet, the architecture employs a Cloud SQL Proxy for secure connectivity. The Cloud SQL Proxy client, running inside the Cloud Run container, securely communicates with the Cloud SQL Proxy server, which in turn connects to the MySQL instance.

This setup provides multiple layers of protection, including authentication, encrypted connections, and prevention of unauthorized external access. The proxy mechanism ensures that the database credentials are not exposed within the application and that all traffic between the application and the database remains internal to Google Cloud's infrastructure.

The AttendEase web application provides both frontend and backend functionalities using a unified Next.js framework. The user interface (UI) is built with Next.js pages and components, while the backend APIs are implemented within the same framework, enabling efficient client–server communication.

The system also supports user authentication through Microsoft Single Sign-On (SSO), ensuring secure and centralized access control. Once authenticated, users interact with the AttendEase platform via the web interface hosted on Cloud Run, while all API operations securely communicate with the Cloud SQL database through the proxy layer.

The AttendEase architecture leverages modern cloud technologies to deliver a scalable, secure, and maintainable web application. By integrating GitHub Actions for automated deployment, Cloud Run for serverless hosting, and Cloud SQL with proxy-based connectivity for database management, the system achieves both operational efficiency and robust security.

5 Implementation

5.1 Technical Overview

The AttendEase application is developed using a modern web architecture that integrates both frontend and backend components within a unified Next.js framework. This section provides a detailed technical overview of the system's implementation, covering the frontend, backend, and automated deployment pipeline.

The frontend delivers a responsive interface for students, lecturers, and administrators to manage attendance records. Built with Next.js for client-side and server-side rendering, it utilizes *Radix UI* components customized with *Tailwind CSS* for accessible, maintainable interfaces. The frontend communicates with the backend through API calls to Next.js routes, ensuring clear separation between UI and server logic.

The backend handles data processing, authentication, and secure MySQL database communication via *mysql2* library and Unix socket paths. Authentication uses the *Better-Auth* library supporting email-password and Microsoft SSO (restricted to `uowmail.edu.au` domain). The backend incorporates *Swagger UI* for API documentation, enabling developers to visualize and test endpoints through an interactive interface.

Deployment is automated using *GitHub Actions* CI/CD pipeline with two stages: (1) *Application Build and Deployment* - builds Docker images and deploys to Google Cloud Run with automatic scaling and SSL, and (2) *Database Initialization* - connects via Cloud SQL Proxy to execute SQL scripts for schema setup and data seeding.

5.2 Frontend

The frontend of the AttendEase system serves as the user interface that connects students, lecturers to the backend services. It is developed using the React-based `Next.js` framework. The frontend design emphasizes usability, responsiveness, and maintainability to deliver an intuitive and modern experience across all devices.

The technology stack includes:

- Next.js 15 and React 19 with TypeScript for server/client components and routing.
- Tailwind CSS 4 and shadcn/ui component system for styling and UI components.
- TanStack React Query for state management and data fetching.
- React Hook Form + Zod for forms and validation.
- Recharts for analytics and QR scanning for mobile attendance.

The application architecture follows Next.js App Router conventions with a structured routing system. The root layout composes the application shell: global styles, theme provider, data-fetching provider, header/footer, and a global toaster. Routing is organized into logical groups:

- `(auth)`: login, signup, password flows.
- `(protected)`: authenticated area with role-specific groups:
  - `(student)`: scan, records, analytics dashboard.
  - `(lecturer)`: QR generation, live tracking, analytics and reporting.

Implementation and User Interface The frontend application is organized into role-specific interfaces designed to reflect the permissions and functionalities available to each user group. Each interface is dynamically rendered based on the authenticated user's role and access permissions retrieved from the backend, ensuring a personalized and secure user experience.

- **Student Interface:** Provides access to course details, attendance history, and QR code scanning functionality for attendance check-in. Students can view attendance confirmation in real time and receive notifications of successful check-ins or errors.
- **Lecturer Interface:** Enables lecturers to manage study sessions, generate QR codes for each session, and monitor attendance submissions. Lecturers can view live student attendance data and session summaries through graphical dashboards.

Data Management and API Communication The frontend communicates with the backend through RESTful APIs documented with Swagger UI. All user actions trigger API requests that return structured JSON data, with comprehensive error handling and response validation.

Authentication and Session Handling The frontend integrates with the backend's system, securely exchanging credentials or Microsoft SSO tokens via HTTPS. Session tokens are stored in browser cookies/local storage, with client-side route protection redirecting unauthorized users to login. This complements the backend's RBAC system for consistent authorization enforcement.

Key Components and Screens

- **Authentication forms**: Login and related forms use React Hook Form + Zod for schema-driven validation, with Microsoft SSO initiated through the Better Auth client. The authentication interface, as shown in Figure 14, provides multiple entry points including traditional email/password authentication and Microsoft SSO integration for seamless institutional access.

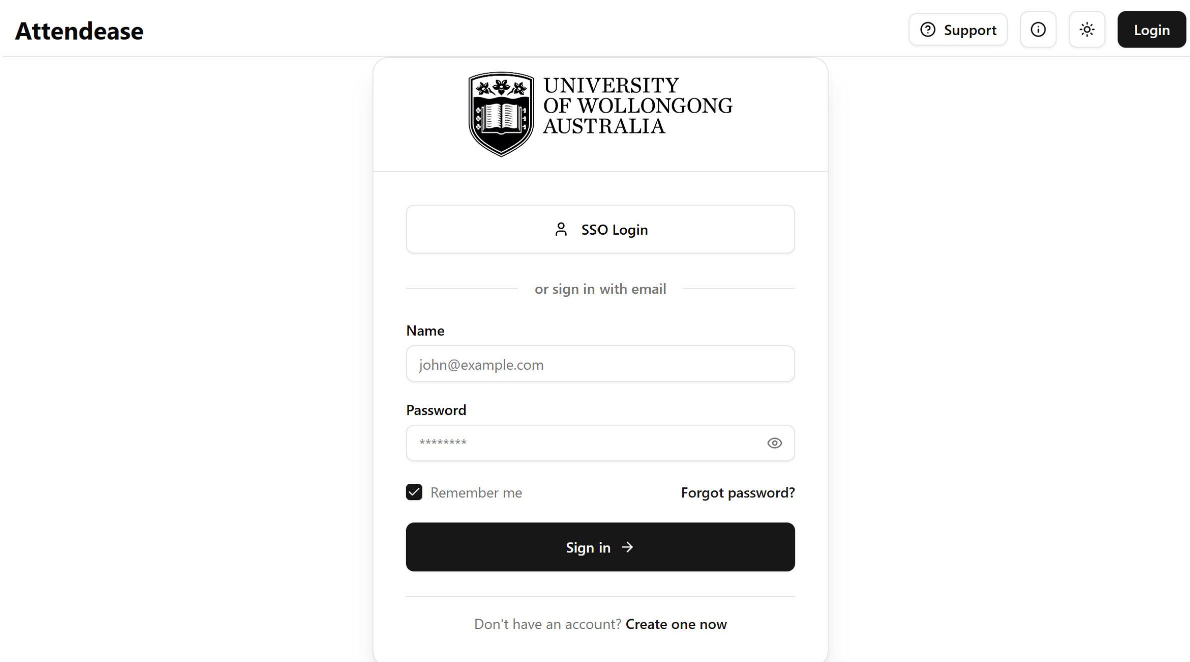

Figure 14: Authentication entry points: Email/Password and Microsoft SSO.

- **Lecturer QR generation**: A guided flow powered by a context provider and the `NewQrGeneration` UI, with tabs/sections for Location & Validation (room, radius, geo toggle) and Time Windows (entry/exit). A sticky panel renders the generated QR for immediate sharing. The process begins with course selection as shown in Figure 15, followed by the main QR generation interface (Figure 16) which provides comprehensive configuration options.

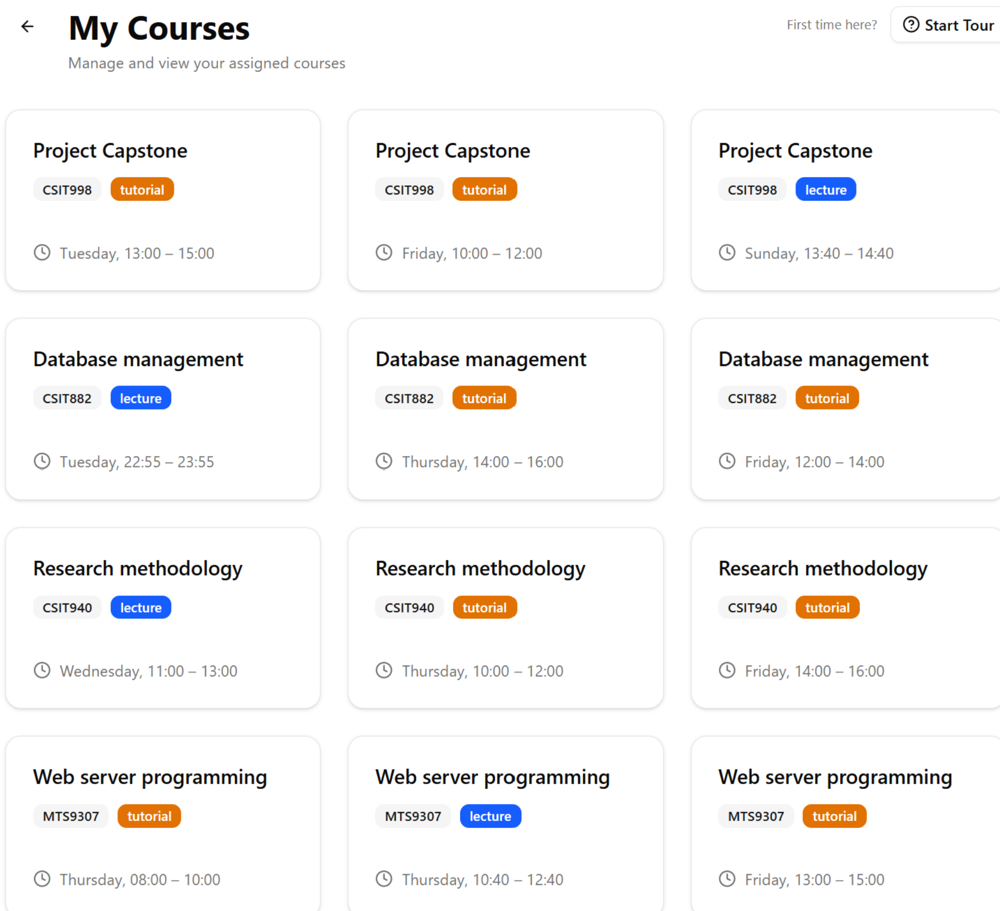

Figure 15: Course selection interface for lecturer.

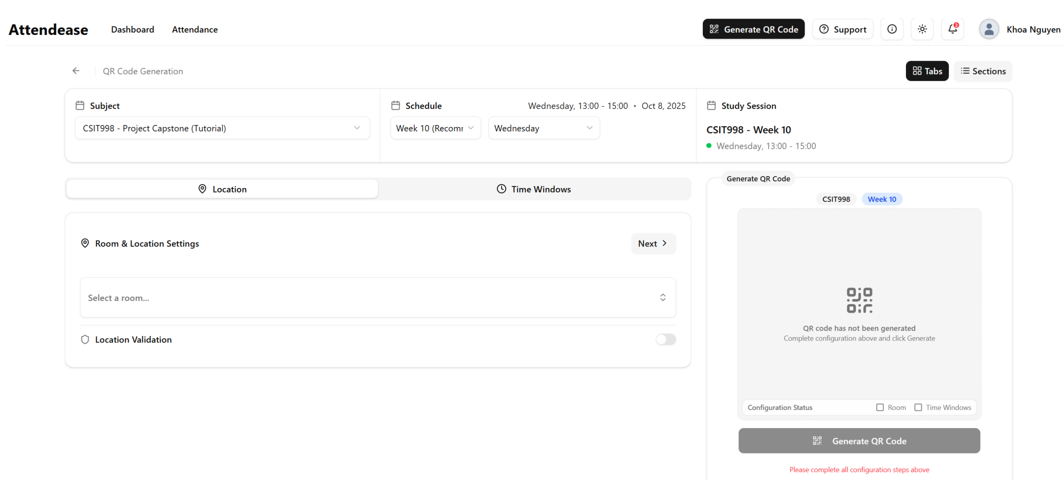

Figure 16: Main QR generation interface with configuration tabs for location and time settings.

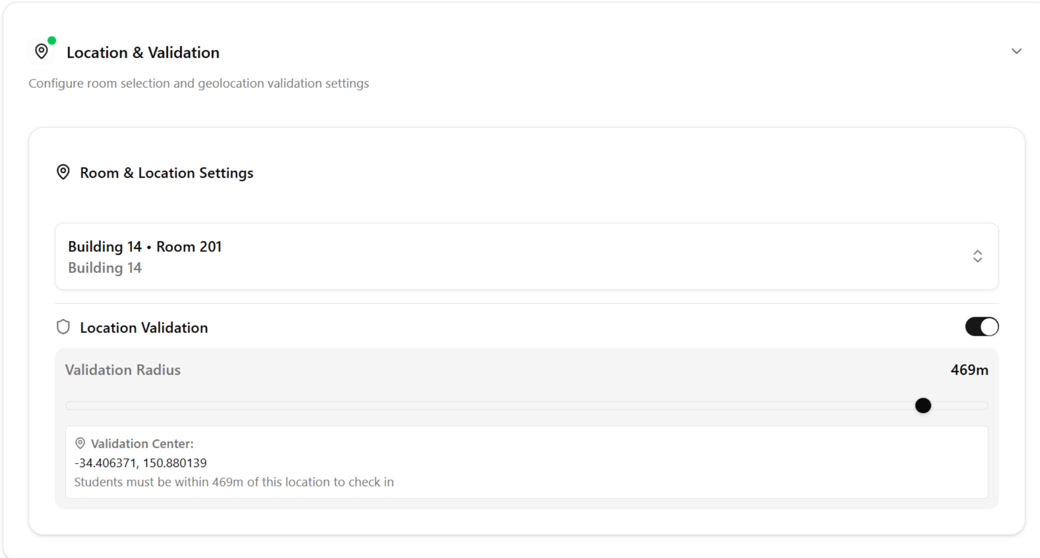

Figure 17: Location & Validation tab: room selection, radius configuration, and geolocation validation settings.

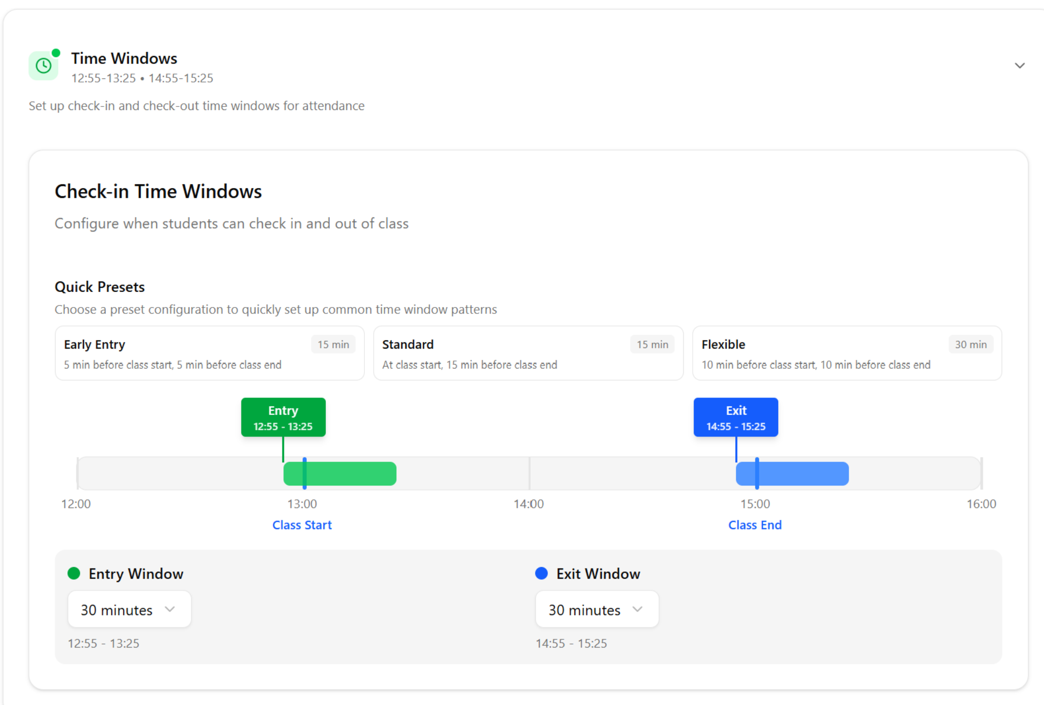

Figure 18: Time Windows tab: configuration of entry and exit validity windows for attendance tracking.

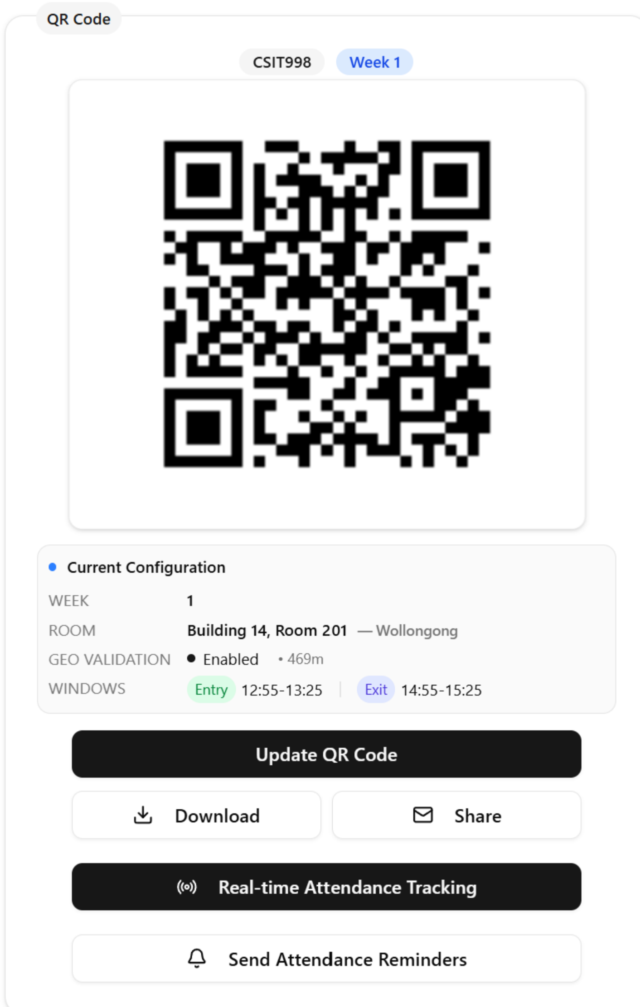

Figure 19: Generated QR code preview ready for display or sharing, accessible through the sticky panel.

- **Real-time tracking and analytics**: Role dashboards render live lists and charts. Recharts visualizations are themed via `ChartContainer` and employ tooltips/legends consistent with tokens defined in `globals.css`. The real-time dashboard, illustrated in Figure 20, provides lecturers with live attendance data, student participation metrics, and interactive charts that update dynamically as students check in and out of sessions.

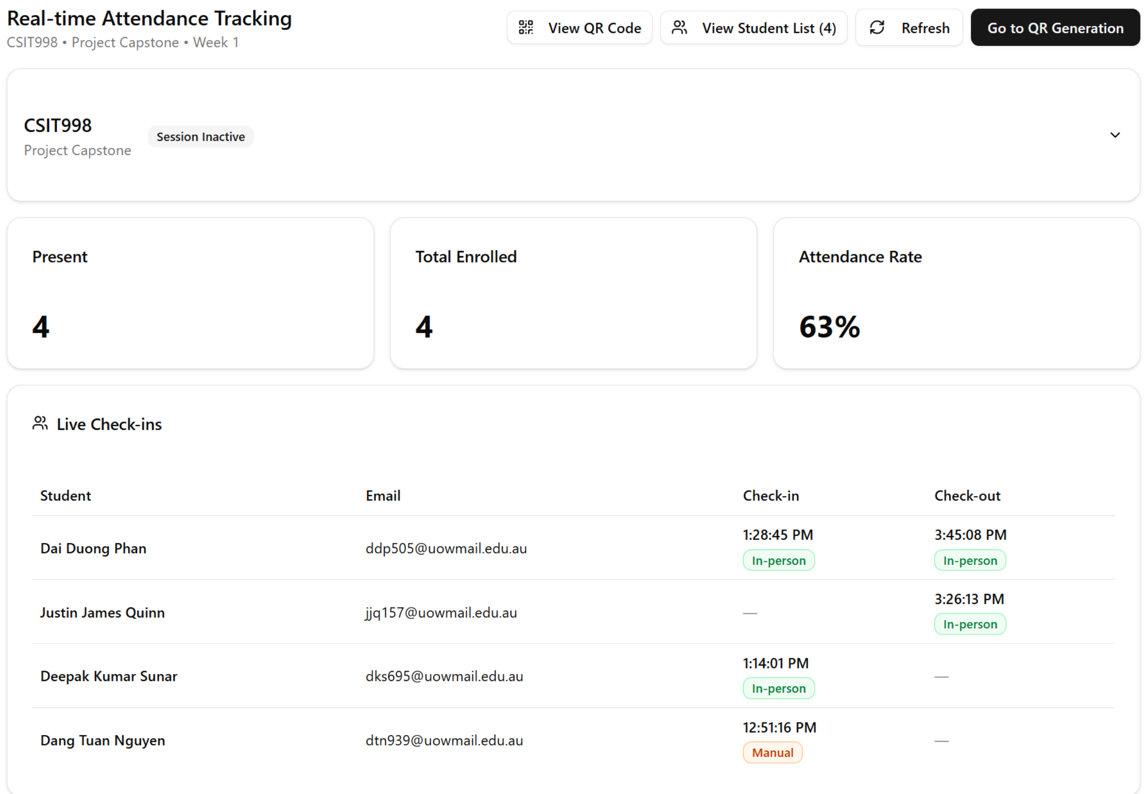

Figure 20: Real-time attendance tracking during an active session.

- **Student scanning and check-in**: Orchestrates the two-step check-in flow (entry/exit), uses a custom geolocation hook and Haversine distance for geofencing logic, and presents dialogs for online fallback and geo-validation errors. The camera-based QR scanner and navigates to the scanned session URL. The student check-in interface, shown in Figure 21, demonstrates the mobile-optimized scanning experience with real-time validation feedback and geolocation verification.

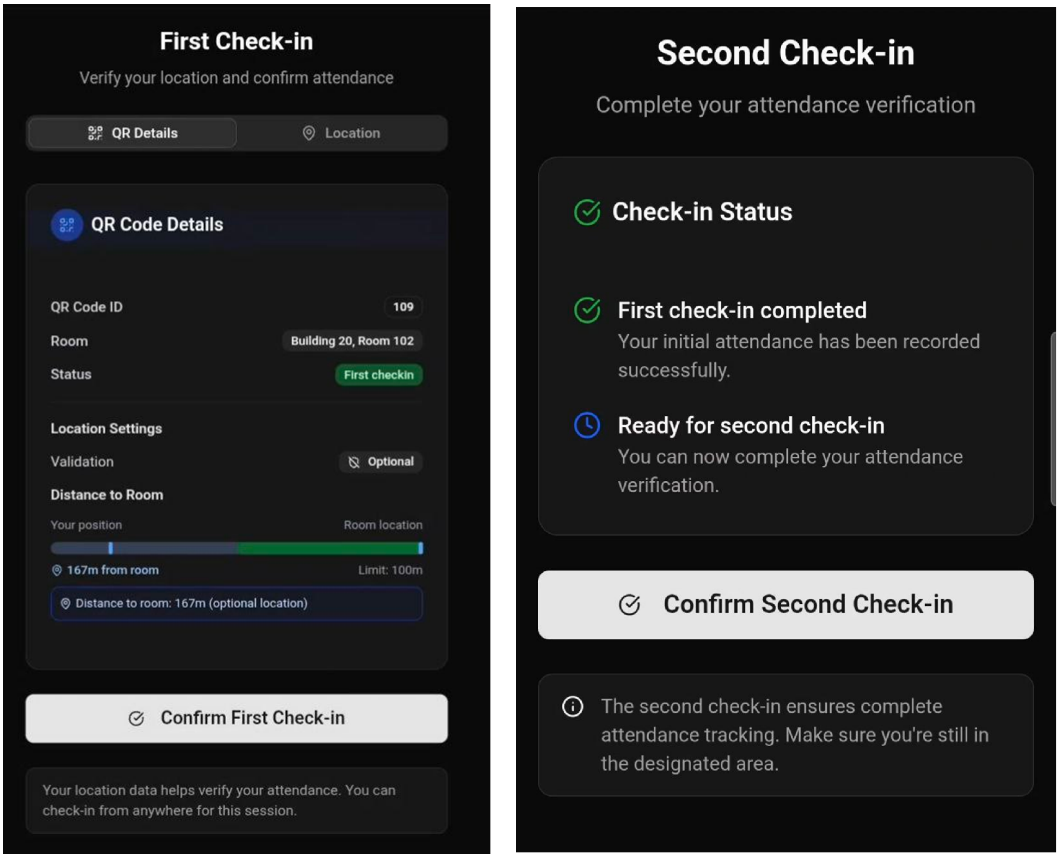

Figure 21: Student check-in interface with QR code scanning and geolocation validation.

- **Student records**: Provides students with access to their attendance history and records, allowing them to view their participation in different study sessions and track their attendance patterns over time. The student records dashboard, displayed in Figure 22, offers a comprehensive view of attendance history with detailed session information, attendance status, and personal analytics to help students monitor their engagement.

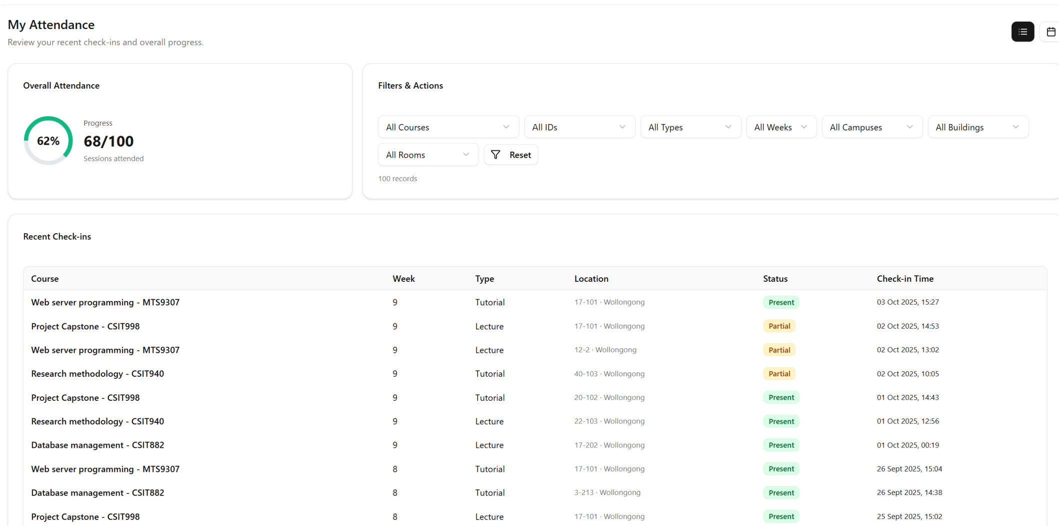

Figure 22: Student records interface showing attendance history and session details.

- **Manual check-in**: Enables lecturers to manually record student attendance for cases where students cannot use QR code scanning due to device issues, accessibility needs, or technical difficulties. The manual check-in interface, shown in Figure 23, provides lecturers with a comprehensive student list where they can select individual students and mark their attendance status, ensuring inclusive participation for all students regardless of their technical capabilities.

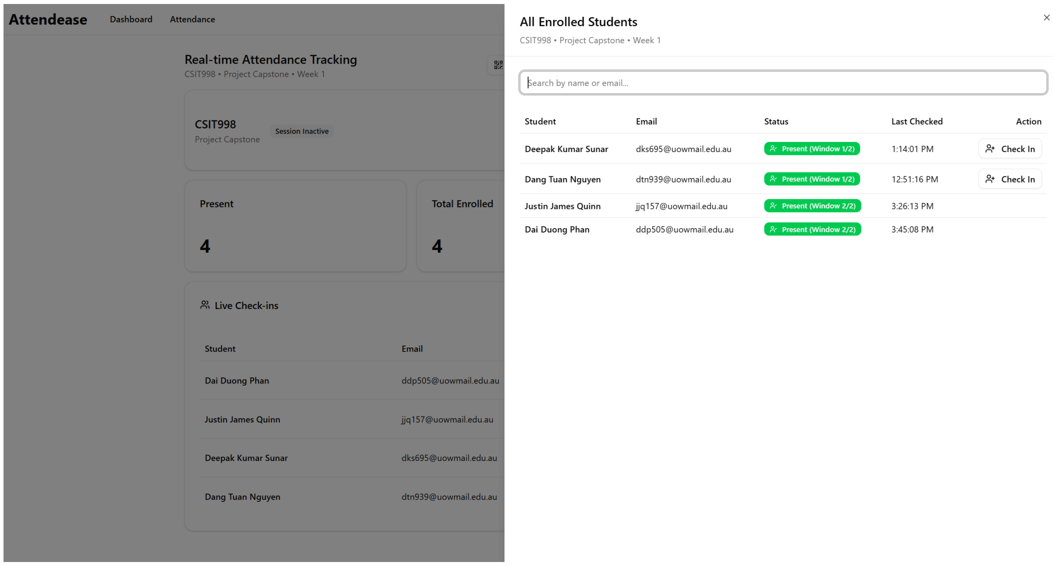

Figure 23: Manual check-in interface for lecturers to record student attendance manually.

5.3 Backend

The backend manages business logic, database operations, and user authentication using Next.js API routes. It connects to MySQL via the `mysql2` library and Cloud SQL Proxy for secure database communication. All credentials are managed through environment variables and GitHub Secrets.

API Design and Structure The backend exposes RESTful APIs organized by user roles: **Auth** (authentication/SSO), **Student** (QR scanning, attendance history), **Lecturer** (session management, analytics), **Common** (profile management), **Statistics** (attendance analytics), and **System** (automated operations). All endpoints are documented using OpenAPI specifications for seamless frontend integration.

API Documentation with Swagger UI The backend integrates `Swagger UI` for automatic API documentation generation, providing an interactive interface for developers to view endpoints, inspect request/response structures, and execute test calls. Complete endpoint listings are available in Appendix C.

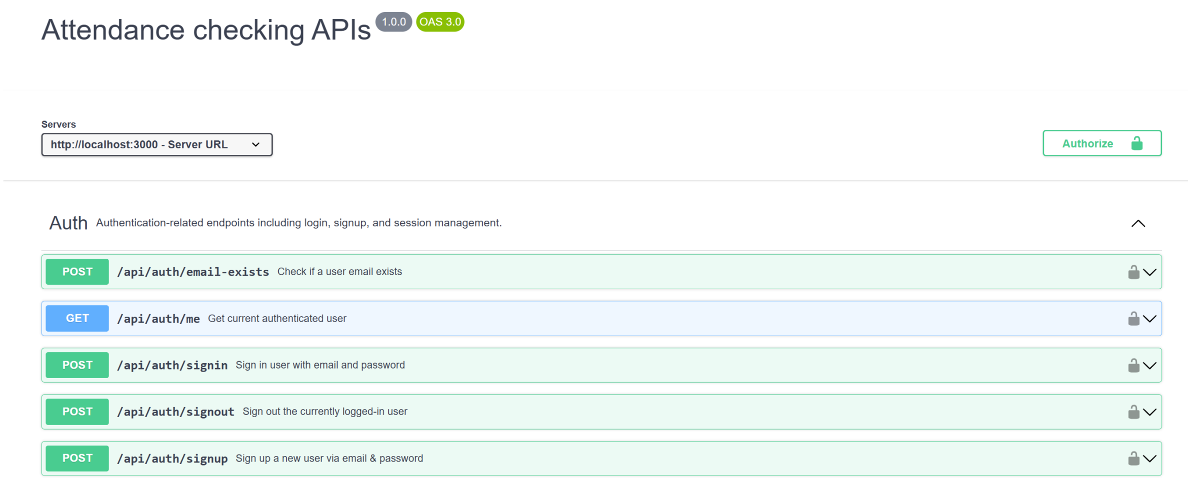

Figure 24: Swagger UI for the backend API documentation.

Authentication and Security Authentication is handled via `BetterAuth` library supporting email-password and Microsoft SSO (restricted to `uowmail.edu.au` domain). Role-based access control ensures API security, with credentials managed through GitHub Secrets and environment variables.

Deployment The system uses automated CI/CD via GitHub Actions with two workflows: *Application Deployment* (builds and deploys to Google Cloud Run) and *Database Initialization* (executes SQL scripts on Cloud SQL). All credentials are managed through GitHub Secrets, and Cloud SQL Proxy ensures secure database connections.

5.4 Analytics and Reporting

The AttendEase system provides comprehensive analytics converting attendance data into actionable insights for students and lecturers. The analytics module features enhanced visualization, predictive modeling, and automated reporting to support data-driven attendance management decisions.

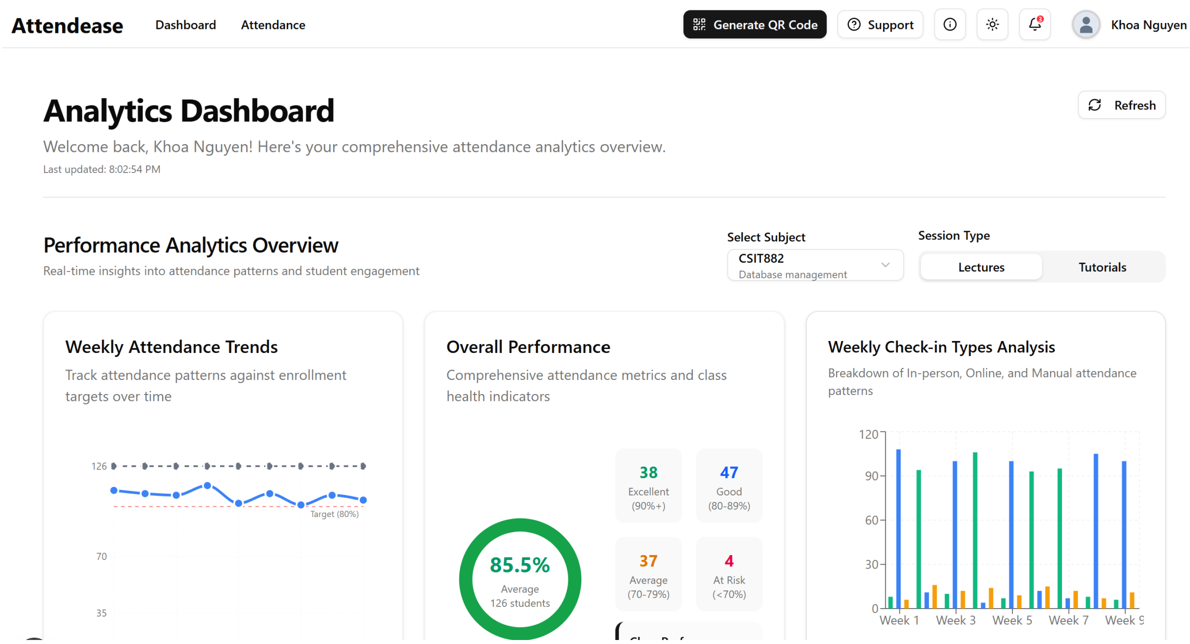

Figure 25: Lecturer analytics/dashboard: course KPIs and trends.

Lecturer Analytics Dashboard The system provides hierarchical analytics from semester-wide statistics to individual student records, including course-level analytics, session-type analysis, and weekly performance tracking. Advanced visualization tools include color-coded attendance heatmaps, distribution charts, and day-of-week analysis for optimized scheduling decisions.

The system maintains detailed attendance records with complete history, cumulative percentages using a scoring system (2 check-ins = 100%, 1 check-in = 50%, 0 = 0%), and personal trend analysis. At-risk student identification automatically flags students below 80% attendance with early warning systems.

Automated email reporting generates personalized attendance summaries after each lecture session, including current week/semester percentages, remaining absence allowance, and at-risk warnings. Lecturers can configure reports for 'This Week', 'This Month', or 'This Semester' periods.

Meta-analytics enable lecturers to assess teaching effectiveness through overall attendance rates, tutorial-specific metrics, student engagement comparisons, and identification of high-performing versus struggling classes.

The Advanced Analytics module provides predictive analysis through day-of-week attendance patterns, time-based semester analysis (early/mid/late periods), student risk prediction with Critical Risk (¡60%), Moderate Risk (60-75%), and Watch List categories, attendance forecasting using linear regression, and performance benchmarking against historical data and institutional averages.

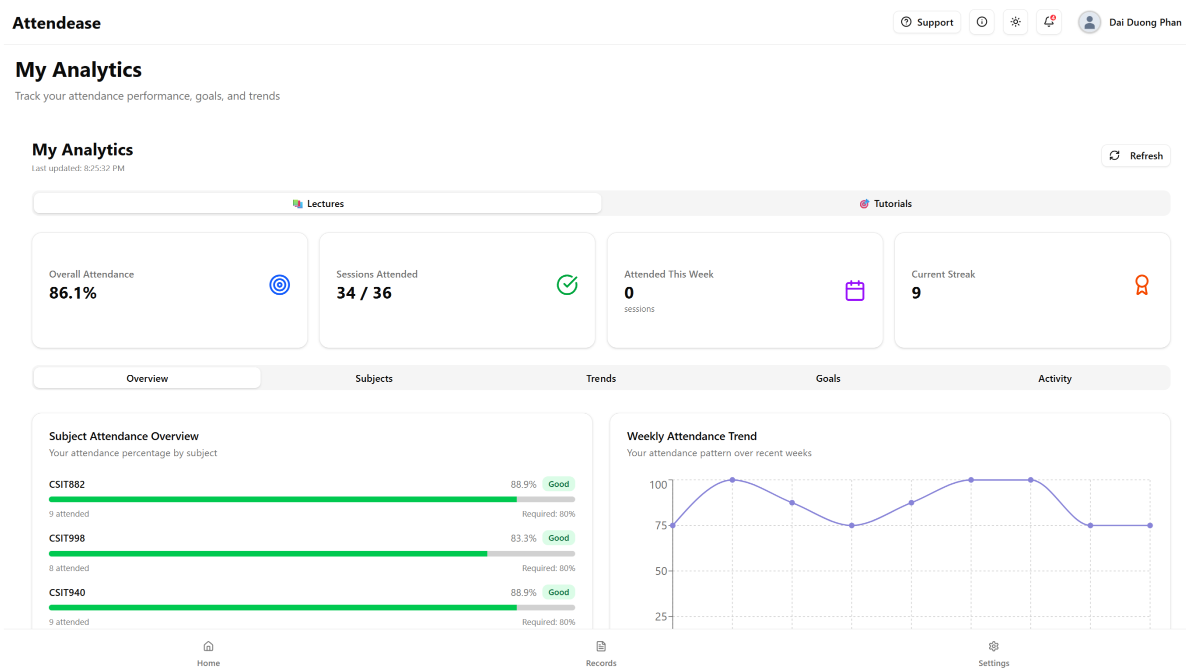

Figure 26: Student dashboard: subjects and recent attendance.

Student Analytics Dashboard Students access comprehensive attendance records through an intuitive dashboard displaying overall percentages, class counts, status indicators (on-track/warning/at-risk), and color-coded progress tracking. The system provides individual attendance percentages for each course with week-by-week records, class averages, and trend indicators, along with progress bars, goal completion indicators, trend charts, and calendar views.

Chronological check-in events include session details, timestamps, geolocation verification, and attendance points (100%/50%/0%) with filtering capabilities. Clear visualization of 80% threshold progress includes buffer zone indicators, risk assessments, and end-of-semester attendance forecasting with warning alerts and early intervention notifications.

5.5 Testing and Evaluation

We used a layered, risk-driven approach aligned with agile sprints to validate correctness, reliability, security, and usability across student and lecturer workflows. Testing covered:

- *Methodology*: fast feedback during implementation plus pre-release validation; focused unit tests and realistic end-to-end user journeys for lecturer and student roles.
- *Environments*: local (Node.js/Next.js, MySQL), production-like GCP staging (Cloud Run + Cloud SQL with TLS 1.3), and real devices (Android/iOS/desktop) under varied network conditions.
- *Types*: Unit (utilities, hooks, API client), Integration/E2E (QR creation, in-person/online/manual check-ins, real-time tracking, analytics), Acceptance (representative scenarios), and Non-functional (security, performance, compatibility, accessibility).
- *Results*: Unit — 9 suites / 30 cases, 100% pass, coverage 87% lines / 82% branches / 91% functions; Integration/E2E — 14 scenarios, 100% pass; Acceptance — 27 scenarios, 100% pass.

- *Non-functional highlights*: TLS 1.3, authenticated routes, RBAC, no GPS persistence; major browsers verified and responsive layouts across 320–1920px.
- *Notable fixes*: week anchor off-by-one in date labeling; duplicate check-in race (UI disable + DB unique constraint + 409 handling); clearer messaging for denied geolocation permissions.
- *Known limitation*: robust verification of online attendance remains future work.

Detailed test cases are provided in Appendix A.

6 Discussion

6.1 Benefits and Implications

The attendEase platform offers a range of benefits that address long-standing challenges in attendance management while introducing forward- looking implications for educational institutions. Its implementation not only improves operational efficiency and security but also promotes inclusivity, data informed decision making, and institutional accountability.

6.1.1 Benefits

- **Efficiency and Time saving:**AttendEase automates attendance tracking through dynamically generated QR codes, eliminating manual roll calls and paper-based registers. This digital transformation significantly reduces the administrative workload for lecturers and staff. The system's real-time dashboards offer instant visibility into student attendance, enabling instructors to identify absences or late arrivals during live sessions. As a result, classroom time can be used more effectively for teaching and engagement rather than administrative tasks.
- **Enhanced Accuracy and Security:** The system combines geolocation verification, time-sensitive QR codes, and multiple authentication to ensure accurate and secure attendance records. Proxy attendance is minimised through dual verification (QR scan and GPS confirmation). Sensitive information is safeguarded using TLS 1.3 encryption, randomised QR code URLs, and strict role-based access controls that limit data exposure to authorised users only. These measures align with modern privacy regulations such as the General Data Protection Regulation (GDPR), ensuring ethical and secure handling of student data.
- **Hybrid Learning Compatibility:**Recognizing the evolution of post-pandemic education, AttendEase supports both in-person and online attendance tracking. Students attending remotely can sign in via a link that does not require camera access, ensuring inclusivity. For those without devices or with accessibility challenges, manual check-in options are available for lecturers to record attendance. This design supports hybrid, blended, and flexible delivery modes, aligning with modern university teaching environments.
- **Advanced Analytics and Reporting:**The analytics module provides multi-level insights, including course-level summaries, session-type comparisons, and weekly performance tracking. Lecturers can visualise attendance trends, identify at-risk students, and take timely action to improve engagement. Students benefit from personal dashboards that track their attendance history and send automated notifications when they approach absenteeism thresholds. This two-way visibility promotes accountability and encourages early intervention to support academic success.

- **Accessibility and Inclusivity:**AttendEase prioritises accessibility through responsive web design, device flexibility, and manual input features for those unable to scan QR codes. Planned enhancements include screen reader compatibility, keyboard navigation, and multi-language support—ensuring equal participation for users with disabilities or diverse linguistic backgrounds. Such features strengthen the system's commitment to inclusivity and equitable access to digital tools.

- **Cost-Effectiveness:**Unlike biometric or RFID-based attendance systems, AttendEase relies on existing smartphone and web technologies. This low-cost approach reduces implementation expenses and eliminates the need for specialised hardware. Combined with reduced paper usage and streamlined administrative processes, the system provides a sustainable and financially efficient attendance management model.

- **Student Engagement and Accountability:**The system's dual check-in process (at the start and mid of each session) fosters continuous engagement and discourages superficial participation. Transparent attendance records build fairness and trust in participation-based grading, while giving students clearer feedback on their engagement patterns.

6.1.2 Implications

- **Institutional Impact:**AttendEase promotes data-driven governance by turning attendance information into actionable insights for student support, curriculum planning, and resource allocation. Automated data collection enhances compliance with regulatory and accreditation requirements, particularly in professional programs that mandate attendance records. The system also strengthens transparency and fairness in attendance-based assessment, fostering trust between students and staff.

- **Privacy and Ethical Considerations:**The use of geolocation data and authentication measures necessitates strong data governance. AttendEase addresses this through user consent, minimal data collection, and short-term processing of GPS data. Nonetheless, institutions adopting the system must maintain clear privacy policies, communicate ethical data practices, and ensure compliance with GDPR and local regulations to avoid perceptions of digital surveillance.

- **Technical and Operational Implications:**The successful deployment of AttendEase depends on smooth integration with existing institutional systems such as Learning Management Systems (LMS), Student Information Systems (SIS), and Single Sign-On (SSO) services. Institutions may require technical support, staff training, and user orientation programs to ensure effective adoption. The challenge of verifying online attendance and preventing misuse remains an area for continued development.

- **Future Development and Research Implications:**The project sets the stage for further research into online attendance verification, learning analytics, and the correlation between attendance and academic performance. Future iterations could integrate predictive analytics, AI-driven engagement metrics, and advanced accessibility features. Professionally, the project highlights the growing importance of secure, hybrid-ready educational technologies in post-pandemic learning environments.

6.2 Challenges Faced

Throughout the course of the project, several challenges were faced and overcome. These include changes to the database design as well as skill development.

6.2.1 Changes to Database Design

Throughout the project, there were several instances in which the database had to be redesigned. This was done in order to accommodate changes to the needs of the users, and to more closely align with the proposed functions of the system.

The initial design of the database was created based on a UML diagram to describe all of the classes and data elements expected to be required for the planned system. After this database was implemented, and the system began to be developed on top of it, several discrepancies were found between the implemented database and the needs of the system, such as handling multiple logins during a session, and handling enrolments into tutorials and lectures.

While these changes were mostly iterative with only small changes, and involved careful planning before changes were implemented, these changes consumed a large amount of time due to the creation of new sample testing data, realigning the code with the modified database, and testing for errors. Overall, these changes helped to ensure that the end product meets the needs of potential users as closely as possible, and works as a highly-functional system.

6.2.2 Skill Development

The group contains members with diverse backgrounds and skill sets. While this helped to create an effective system and associated documentation by leveraging each member's skill sets, group members also encountered activities that were not within their skill sets, and for which they needed to improve their skill sets. This was also highlighted through the implementation of more advanced features, such as implementing analytics tools, which many members were unfamiliar with. Overall, through developing these skills, the group was able to work more effectively together to create the finalised system, and these skills have helped prepare the group for future endeavours.

6.2.3 Time Limitations

Throughout the project, one major limitation was the available time. The time from project conception to delivery was from March to October. This put a time pressure on the project. As a result, consideration and was needed to define the scope of the project. To attempt to keep on track a schedule was developed to plan for the expected length of tasks so that they could be more efficiently completed. Despite this planning, as complications arose adjustments were required to adjust the scope of the project and prioritise tasks to ensure that the project could be delivered in a functional state by the specified time limit. This resulted in some planned features, such as verification methods to confirm that online students access the online lecture they confirm their attendance during.

6.2.4 Verification for Online Students

Although due to time limitations, none was implemented, throughout the project, the team investigated several methods to verify online student attendance. This feature is important as without it, there is no way to determine whether a student who is not at the correct location is actually attending the online version of the lecture, or has gained access to the QR code link via other means, such as having it sent to them by a friend attending the class. Several methods to counteract this issue were investigated, with varying degrees of effectiveness.

The first proposed method to account for this was the inclusion of a link redirect system that tracks user access to the online streaming link. This system would take the online access link from an online platform, such as Webex or Zoom, and generate a link which the lecturer could add to the Moodle page, or otherwise distribute to students. This link would take students to a page where they would be prompted to sign in with their account (if not already signed in) before being automatically redirected to the online streaming platform. For students who are already signed in, they would be automatically be redirected without having to take any further action. This would place a low burden on both students and lecturers as it does not deviate significantly from the current workflow generally in place. Before students are redirected, the system would record information about the account accessing the link, the time, and the link accessed. This could then automatically be compared against attendance confirmations to automatically confirm students as online students without any need from further input by students or lecturers. This method could be effective as by clicking on the link and being redirected, it can at least ensure that students have actually accessed the online streaming link.

Another proposed method for verifying online student attendance is to use the inbuilt Moodle logs export feature to allow lecturers to export this information. This however is a less elegant solution as it requires lecturers to manually export this data from Moodle and import it into the platform for each individual lecture, greatly increasing the burden placed on lecturers. Both of these methods however as limited in their effectiveness as some students who are not actually attending online may click the link to have their online attendance verified, and then close the page down, so that they are marked as an online student even if they did not actually watch the lecture. Another limitation is that some students may save the link from the previous instance of the lecture to join, bypassing the online tracking link, and being marked as not visiting the online lecture, even when they did.

Another method investigated was exporting data directly from streaming platforms such as Webex and Zoom. This information could then be imported into the platform to verify online attendance. Like the Moodle export method, this method would also add a burden to the lecturer in manually exporting and importing this data. These platforms, may also track information such as the duration and join/end time of viewers, allowing more accurate verification that students were actually present for the duration of the lecture. These platforms however by default allow students with the link to join meetings either anonymously or with any registered account, which may result in students who joined using these means not being marked as present.

In order to minimise the effort required to export and import information from these platforms, the Webex API was investigated to attempt to automate the importing of meeting data. This however proved challenging as the Webex API requires the individual meeting ID for each instance as well as an API key to access participant information. These meeting IDs and API keys would need to be manually added, greatly increasing the burden on lecturers, especially with the enforced 12-hour validity period of Webex API keys. As a result, this method was deemed not viable.

Each of these methods also exports information in different formats, further complicating the import process, by requiring different steps for each import method to extract the necessary information. Overall, while no ideal solution could be found to verify online attendance, some combination of features could be used.

6.3 Limitations

While measures through planning and research have aimed to reduce limitations and deficiencies within the software project, some still exist. Further efforts will be required to minimise or eliminate these limitations when possible in order to ensure a maximally effective and accurate system.

6.3.1 Sign in For Students Without Mobile Devices

The proposed platform assumes that most, if not all students using it will have access to a mobile device to scan the QR code presented by the lecturer to enable them to sign in, however some students may not have access to a device for a multitude of reasons. This may include students who do not own a device, as well as those who are unable to use their device for reasons such as insufficient battery charge. This would render these students unable to sign in, resulting in them being marked as absent, regardless of if they are present or not.

In order to try and counteract this, the project has implemented a manual check-in option, which allows the lecturer to manually add students as present into the system. This however is an inelegant solution as this takes up additional class time to check students in and disrupts the flow of the lecture, similar to in manual paper-based sign in methods currently in use. This solution is also not effective for online students who would not be present to ask the lecturer to mark them as present.

6.3.2 Online Sign-in Bypass

In the current system, the system only verifies that online students have signed in at the correct time using the QR code and that they have indicated that they are an online student at the time of sign in. This system does not verify that they have actually viewed the lecture, and could be bypassed by a student who is not present being sent a copy of the QR code by another student who is present, and simply indicating that they are an online student when they confirm their attendance. This will result in students erroneously being marked as attending the lecture online when they in fact did not.

The proposed verification solutions presented (by either tracking access to the lecture, or exporting streaming platform attendance statistics) go a long way to reducing this issue of fraudulent online sign ins by verifying that a given student has either accessed the online meeting or verifying a specific duration of attendance. They can however still be bypassed by students either accessing the online link and then closing it to be marked as having accessed the platform, or in methods that track duration, simply opening the link and leaving the meeting playing for the duration of the lecture, even if they are not actually attending.

Additionally, some students after accessing the online lecture link in a given week may opt to save this link to access it directly in subsequent weeks, bypassing Moodle or other link tracking methods. This will result in some students not being verified as attending the lecture online when these verification methods are used, even if the students actually attend.

Some online streaming platforms allow people to join lectures with just the link, either anonymously without signing in, or with their own accounts which may not be linked to their identity or their university details. As a result, even if the attendance details are exported directly from the streaming platform, some students who attended would not be marked as present, as the record corresponding to their attendance would not be able to be linked to their account. Settings within the streaming platform would need to be configured to mandate that students access the online lecture stream using an account with their registered university account details (if possible within the settings made available to the lecturer on any respective platform) to ensure that any attendees are able to be accounted for, although this may not be possible on some platforms and may inconvenience some users.

As a result, in order to be maximally inclusive of all students and ensure that as many online students are able to have their online attendance verified, multiple sources of attendance information would need to be combines, which adds complexity, and room for easier exploitation by students.

In any of these scenarios, even if measures are taken to ensure that students are accessing the online lecture platform during the duration of the lecture, this does not ensure that the students are actually paying attention during the lecture, as opposed to opening the lecture and doing other tasks, or opening the lecture and leaving for some portion of it. While measures, such as using the webcam in a device to track student attentiveness could be implemented, this would pose a privacy risk and greatly increase the processing required by the platform, making it unlikely to be a realistic solution. As a result, alternate minimally-invasive measures of determining whether students are actually present and attentive during online lectures may need to be developed.

6.4 Future Work

6.4.1 Online Student Tracking

Implementing online tracking for students to verify online attendance could make online attendance statistics more meaningful by ensuring that online those who access the online lecture can mark themselves as such. By combining sources of information, such as link access and duration of access, it could be better determined whether students actually access the online lecture. Combining information from multiple sources such as Moodle and the streaming platforms themselves would be a complex procedure, but could be used to make a more thorough determination. Additionally, development of a tracking tool that can determine how long users watch a lecture, such as embedding an online stream within a webpage that gathers analytics could be conducted, although this would be an intensive procedure. By focusing further feature development to try to address this issue of students registering as online students without viewing the online lecture, lecturers could place more faith in the attendance statistics being delivered to them in order to make better determinations about the class.

6.4.2 Improved Accessibility Features

When developing any platform, accessibility is an important consideration. Increasing accessibility allows a wider range of individuals to be potential users, and can help ensure that noone is excluded from being able to use the application. Vision, hearing, and mobility are three of the most common types of disability [15]. Some common disabilities such as visual impairment may make a complex application difficult to navigate. By building in screenreader support, keyboard navigation, high-contrast and colour-blindness-friendly colour modes, verbal navigation, and other similar features users with visual impairments could better use the application. Additionally, such as those with motor or visual impairments may find it difficult to scan a QR code, and alternate methods for these students could be designed. To help users from linguistically-diverse backgrounds, an option for multiple languages could be integrated to help users better navigate and use the application. All of these changes would make the application more accessible and usable by a wider range of users.

6.4.3 Check-in For Students Without Device Access

Another feature related to access is to better account for students who do not have access to devices with which they can confirm their attendance. As the proposed solution still currently requires students who do not have access to a device to be manually signed in by their lecturer. Further research could help to develop a process which could streamline this process to reduce the disruption caused by signing in students without access to a device. This process may include combining the system with other proposed methods for confirming attendance such as card readers at entry points, or the ability to confirm attendance with other devices which could be borrowed by the institution. Overall, this would improve the overall time efficiency of the sign-in process, allowing a greater proportion of class time to be utilised for teaching.

6.4.4 Enhanced Analytics

In its current state, the project only provides a basic level of statistics and analytics to students and lecturers. Enhancing the software to provide users with enhanced analytics could allow users to develop more meaningful insights from the data collected by the system. This could include using trends to predict future attendance and long-term trends, comparisons between classes over time, and comparing student attendance with performance in assessments. Leveraging features like these could help to identify students who need additional help or encouragement, and to better design resources that can help improve student outcomes.

6.5 Scalability and Additional Uses

While AttendEase is currently designed to be used in an academic setting through recording attendance at lectures and other university classes, the system could potentially be expanded to track attendance for other events. This includes for both academic and non-academic settings, bringing the benefits of a fast, and efficient method of user sign-in and analytics tracking. Potential uses include for:

- Emergency roll call: compared to traditional methods of roll call (similar to traditional methods in lectures) at muster stations or other meeting points to quickly and efficiently verify the presence of staff and students. Combined with manual check in for those without access to devices, this approach could drastically reduce the amount of time required to verify the presence of each person, which is especially important in the context of emergencies.
- During testing and examinations: by utilising this check-in system at the start and end of examinations, this system could be used to quickly verify whether all students were present, helping reduce administrative workload. Additionally, as students would need to sign in on their device to confirm attendance, this could reduce the risk of another person taking an exam on behalf of a student. One downside of this approach however is that students are generally not permitted to have electronic devices on them for examinations. The additional time of requiring students to scan a QR code and then stow their devices elsewhere could negatively effect the feasibility of this approach.
- Events and Workshops: The system could be modified to support attendance tracking for academic seminars, career fairs, and student development workshops, providing insights into student engagement beyond the classroom.
- Library and Resource Access: QR-based check-ins can be used to monitor usage of study spaces, labs, and equipment, helping optimise resource allocation and usage patterns.
- Clinical and Industry Placements: For students in health, education, or engineering programs, the system can be adapted to log attendance and engagement during off-campus placements, integrating with external partner systems where needed.

7 Conclusion

Overall, this project proposes an effective and novel method of tracking lecture attendance in universities through QR codes that accounts for both online and offline students in a modern mixed delivery environment. With further development and refinement, this system could become even more effective accurately tracking student attendance and engagement. Implementing this system could ensure that students maintain an adequate and consistent level of attendance throughout their classes, allowing them to benefit from active participation and learning, and ultimately leading to better student success.

8 Acknowledgment

We would like to sincerely express our gratitude to our supervisor, Dr Khoa Nguyen, for his unwavering guidance, support, and encouragement throughout the course of this project. His patience, dedication, and commitment to our success have been invaluable. His insightful advice consistently inspired us to improve and stay focused on our objectives. We are especially grateful for the time and effort he invested in reviewing our work and providing constructive feedback, which played a crucial role in shaping the direction and quality of this report. This work is a reflection of his mentorship and the collaborative spirit he fostered within our team.

References

[1] Bekavac, L., Mayer, S., Strecker, J.: Qr-code integrity by design. In: Extended Abstracts of the CHI Conference on Human Factors in Computing Systems. pp. 1–9. ACM (2024). https://doi.org/10.1145/3613905.3651006

[2] Benyó, B., Sodor, B., Doktor, T., Fordos, G.: Student attendance monitoring at the university using nfc. In: Wireless Telecommunications Symposium (WTS), 2012. pp. 1–5 (04 2012). https://doi.org/10.1109/WTS.2012.6266137

[3] Credé, M., Roch, S.G., Kieszczynka, U.M.: Class attendance in college: A meta-analytic review of the relationship of class attendance with grades and student characteristics. Review of Educational Research **80**(2), 272–295 (2010). https://doi.org/10.3102/0034654310362998, https://doi.org/10.3102/0034654310362998

[4] Duraisamy, B.: Student attendance tracking management using face biometric smart system (06 2024). https://doi.org/10.1109/CONIT61985.2024.10626516

[5] Fridman, L., Weber, S., Greenstadt, R., Kam, M.: Active authentication on mobile devices via stylometry, application usage, web browsing, and gps location. IEEE Systems Journal **11**(2), 513–521 (Jun 2017). https://doi.org/10.1109/jsyst.2015.2472579, http://dx.doi.org/10.1109/JSYST.2015.2472579

[6] Liew, K.J., Tan, T.H.: Qr code-based student attendance system. In: 2021 2nd Asia Conference on Computers and Communications (ACCC). pp. 10–14 (2021). https://doi.org/10.1109/ACCC54619.2021.00009

[7] Masalha, F., Hirzallah, N.: A students attendance system using qr code. International Journal of Advanced Computer Science and Applications **5**(3) (2014). https://doi.org/10.14569/IJACSA.2014.050310, http://dx.doi.org/10.14569/IJACSA.2014.050310

[8] Masih, E.A.: Feasibility of using qr code for registration & evaluation of training and its ability to increase response rate - the learners' perception. Nurse Education Today **111**, 105305 (2022). https://doi.org/10.1016/j.nedt.2022.105305, https://doi.org/10.1016/j.nedt.2022.105305

[9] Nuhi, A., Memeti, A., Imeri, F., Cico, B.: Smart attendance system using qr code. In: 2020 9th Mediterranean Conference on Embedded Computing (MECO). pp. 1–4 (2020). https://doi.org/10.1109/MECO49872.2020.9134225

[10] Orah: The hidden costs: How manual attendance tracking damages schools and disrupts parent engagement (2023), https://www.orah.com/blog/hidden-costs-of-manual-attendance, accessed: 02-05-11

[11] Patel, A., Joseph, A., Survase, S., Nair, R.: Smart student attendance system using qr code. SSRN Electronic Journal (01 2019). https://doi.org/10.2139/ssrn.3370769

[12] Shene, A., Aldridge, J., Alamleh, H.: Privacy-preserving zero-effort class attendance tracking system. pp. 1–4 (04 2021). https://doi.org/10.1109/IEMTRONICS52119.2021.9422481

[13] Sunehra, D., Priya, P.L., Bano, A.: Children location monitoring on google maps using gps and gsm technologies. 2016 IEEE 6th International Conference on Advanced Computing (IACC) pp. 711–715 (2016), https://api.semanticscholar.org/CorpusID:26875269

[14] Sweidan, S., Alshareef, S., Darabkh, K.: Sata: A new students attendance tracking application. In: 2021 9th International Conference on Information and Education Technology (ICIET). pp. 41–46 (03 2021). https://doi.org/10.1109/ICIET51873.2021.9419593

[15] Varadaraj, V., Deal, J.A., Campanile, J., Reed, N.S., Swenor, B.K.: National prevalence of disability and disability types among adults in the us, 2019. JAMA Network Open **4**(10), e2130358–e2130358 (10 2021). https://doi.org/10.1001/jamanetworkopen.2021.30358, https://doi.org/10.1001/jamanetworkopen.2021.30358

[16] Yang, S., Song, Y., Ren, H., Huang, X.: An automated student attendance tracking system based on voiceprint and location. In: 2016 11th International Conference on Computer Science & Education (ICCSE). pp. 214–219 (2016). https://doi.org/10.1109/ICCSE.2016.7581583

A Detailed Test Cases

The following test cases are documented in a consistent format to support repeatable execution and auditing.

Test Case ID: TC_LOGIN_001
Test Scenario: User login with valid credentials
Description: Verify that a registered user can log in successfully and is redirected appropriately.
Preconditions: User account exists and is verified.
Test Steps:

1. Navigate to login page.
2. Enter valid email and password.
3. Click *Log in*.

Test Data: email: `student@example.edu`, password: `Password123!`
Expected Result: Login succeeds; user lands on their dashboard.
Actual Result: As observed in staging, login succeeded and redirected to dashboard.
Status: Pass

Test Case ID: TC_UNIT_001
Test Scenario: Week number to calendar date mapping
Description: Validate `getQrDateForWeek()` maps semester start + week/day to correct ISO date.
Preconditions: None.
Test Steps:

1. Call with semesterStart=2025-02-24, week=3, day=Wednesday.

Test Data: Inputs above.
Expected Result: Returns 2025-03-12.
Actual Result: 2025-03-12.
Status: Pass

Test Case ID: TC_UNIT_002
Test Scenario: Validity window computation (start/end)
Description: Ensure start/end timestamps computed with offsets produce non-overlapping windows.
Preconditions: None.
Test Steps:

1. Compute windows for 09:00–10:00 with mid-session window 09:45–09:50.

Test Data: Offsets in minutes.
Expected Result: First window [08:55,09:15], second [09:50,10:05] (example policy).
Actual Result: Matches policy; no overlap.

Status: Pass

Test Case ID: TC_UNIT_003
Test Scenario: Haversine distance calculation correctness
Description: Validate meters computed between two known coordinates.
Preconditions: None.
Test Steps:

1. Calculate distance between known points with published result.

Test Data: Coordinates with expected 111,195 m per 1° lat.
Expected Result: Error ¡ 1 m for small deltas.
Actual Result: Within tolerance.
Status: Pass

Test Case ID: TC_UNIT_004
Test Scenario: Geofence decision function
Description: Given radius R and measured meters D, returns inRange boolean.
Preconditions: None.
Test Steps:

1. Evaluate at D=R-1, D=R, D=R+1.

Test Data: R=50.
Expected Result: true, true (inclusive), false.
Actual Result: Matches expectation.
Status: Pass

Test Case ID: TC_UNIT_005
Test Scenario: Duplicate check-in prevention logic
Description: Idempotency key or unique constraint prevents duplicates within same window.
Preconditions: None.
Test Steps:

1. Simulate two check-ins with same (studentId, sessionId, windowId).

Test Data: Identical payloads.
Expected Result: One record persists; second returns conflict.
Actual Result: Conflict detected; single record.
Status: Pass

Test Case ID: TC_UNIT_006
Test Scenario: Check-in payload schema validation
Description: Zod/validator enforces required fields and types.
Preconditions: None.
Test Steps:

1. Validate missing required field.
2. Validate wrong type.

Test Data: Omitting sessionId; string for latitude.
Expected Result: Validation errors; request rejected.
Actual Result: Errors raised; no DB writes.
Status: Pass

Test Case ID: TC_UNIT_007
Test Scenario: Date label formatting
Description: Format function outputs expected human-readable label for UI (e.g., ”Wed 12 Mar”).
Preconditions: None.
Test Steps:

1. Format known date.

Test Data: 2025-03-12.
Expected Result: ”Wed 12 Mar” (locale as configured).
Actual Result: Matches.
Status: Pass

Test Case ID: TC_LOGIN_002
Test Scenario: User login with invalid credentials
Description: Verify that invalid credentials are rejected with an error.
Preconditions: None.
Test Steps:

1. Navigate to login page.
2. Enter invalid password.
3. Click *Log in*.

Test Data: email: `student@example.edu`, password: `WrongPass!`
Expected Result: Error shown; user remains on login.
Actual Result: Error message displayed; no session created.
Status: Pass

Test Case ID: TC_QR_GEN_001
Test Scenario: Lecturer generates first QR for a session
Description: Validate QR generation with room, radius, time windows, and geo validation.
Preconditions: Logged in as **Lecturer**; at least one course/session configured.
Test Steps:

1. Open `/qr-generation`.
2. Select course/session and week.

3. Set room, radius; enable geo validation; configure time windows.

4. Click *Generate*.

Test Data: Sample session CSIT998 Wk 3; radius 50 m; two validity windows.
Expected Result: Toast success; QR displayed; `dateLabel` reflects computed date.
Actual Result: Behaved as expected; label and QR rendered.
Status: Pass

Test Case ID: TC_QR_UPD_001
Test Scenario: Update existing QR parameters
Description: Validate updates to radius, windows, and geo validation flag.
Preconditions: Logged in as **Lecturer**; existing QR for the session.
Test Steps:

1. Open existing QR.

2. Change radius/windows/geo validation.

3. Click *Update*.

Test Data: Radius 30 m $\rightarrow$ 60 m; adjust end window by +10 minutes.
Expected Result: Update succeeds; new validity windows active.
Actual Result: Parameters applied; windows reflected in tracking view.
Status: Pass

Test Case ID: TC_QR_ANCHOR_001
Test Scenario: Week/day override and anchor logic
Description: Validate `getQrDateForWeek()` and earliest-QR anchor behaviour across weeks.
Preconditions: Logged in as **Lecturer**; two consecutive weeks with QRs generated.
Test Steps:

1. Generate week N.

2. Generate week N+1.

3. Compare `dateLabel` changes.

Test Data: Weeks 5 and 6, same weekday/time.
Expected Result: `dateLabel` matches computed calendar date; anchor logic consistent.
Actual Result: Labels correct per computation.
Status: Pass

Test Case ID: TC_CHECKIN_001
Test Scenario: Student in-person first check-in (within radius)
Description: Verify in-radius validation enables check-in and records first attendance.
Preconditions: Logged in as **Student**; live QR within first validity; student enrolled; location permission granted.
Test Steps:

1. Open `/scan?qr_code_id=<id>`.
2. Allow geolocation.
3. Stand within radius.
4. Click *Check-in*.

Test Data: Radius 50 m; campus coordinates.
Expected Result: Success response; first check-in timestamp shown.
Actual Result: Success; timestamp displayed; record persisted.
Status: Pass

Test Case ID: TC_CHECKIN_002
Test Scenario: Student in-person second check-in (later window)
Description: Validate second validity window flow and recording.
Preconditions: Logged in as **Student**; first check-in completed; second window active.
Test Steps:

1. Refresh page when second window opens.
2. Ensure in radius.
3. Click *Check-in*.

Test Data: Second window start/end configured.
Expected Result: Success; second check-in recorded; no duplicates.
Actual Result: Second record stored; no duplication.
Status: Pass

Test Case ID: TC_CHECKIN_003
Test Scenario: Geo validation disabled
Description: Confirm distance check bypass when QR has `validate_geo = false`.
Preconditions: Logged in as **Student**; QR generated with geo validation disabled.
Test Steps:

1. Open scan URL.
2. Attempt check-in from outside radius.

Test Data: Default radius retained; flag disabled.
Expected Result: Check-in allowed without accurate position.
Actual Result: Check-in succeeded outside nominal radius.
Status: Pass

Test Case ID: TC_CHECKIN_004
Test Scenario: Online check-in path (allowed)
Description: Verify online confirmation dialog and record type.
Preconditions: Logged in as **Student**; QR allows online check-in; student out of radius.

Test Steps:

1. Open scan URL.
2. Trigger online dialog.
3. Confirm online attendance.

Test Data: Out-of-radius location; online option enabled.
Expected Result: Record stored with type = Online.
Actual Result: Type = Online recorded.
Status: Pass

Test Case ID: TC_CHECKIN_005
Test Scenario: Already checked in during same validity
Description: Ensure duplicate check-in in a window is prevented.
Preconditions: Logged in as **Student**; one check-in exists for active window.
Test Steps:

1. Attempt second check-in in same window.

Test Data: Active window unchanged.
Expected Result: Error or disabled action; no new record.
Actual Result: Duplicate prevented; UI disabled.
Status: Pass

Test Case ID: TC_CHECKIN_006
Test Scenario: Out of radius with geo validation enabled
Description: Validate enforcement of geofenced radius.
Preconditions: Logged in as **Student**; QR with geo validation enabled.
Test Steps:

1. Move outside configured radius.
2. Attempt check-in.

Test Data: Radius 50 m; location offset greater than 80 m.
Expected Result: Action disabled or error; no record.
Actual Result: Check-in blocked; no persistence.
Status: Pass

Test Case ID: TC_CHECKIN_007
Test Scenario: Invalid/expired QR ID
Description: Verify handling of nonexistent or expired QR.
Preconditions: Logged in as **Student**.
Test Steps:

1. Open scan URL with invalid QR ID.

Test Data: `/scan?qr_code_id=invalid`.
Expected Result: 404-style message; no action available.
Actual Result: Error page displayed; no record created.
Status: Pass

Test Case ID: TC_AUTHZ_001
Test Scenario: Not enrolled attempts check-in
Description: Confirm authorization enforcement for enrollment.
Preconditions: Logged in as **Student** who is not enrolled in the subject.
Test Steps:

1. Open valid scan URL.
2. Attempt check-in.

Test Data: Non-enrolled student account.
Expected Result: 403 denied; no record created.
Actual Result: 403 observed; no changes in DB.
Status: Pass

Test Case ID: TC_ANALYTICS_001
Test Scenario: Day-of-week patterns analytics
Description: Validate loading skeleton and chart rendering or empty state.
Preconditions: Logged in as **Lecturer**; analytics data available.
Test Steps:

1. Navigate to analytics.
2. Apply filters.

Test Data: Course with varying weekly attendance.
Expected Result: Skeleton then charts or empty state.
Actual Result: Expected sequence observed.
Status: Pass

Test Case ID: TC_EMAIL_001
Test Scenario: Email attendance summary (if route enabled)
Description: Verify API surface and success feedback.
Preconditions: Logged in as **Lecturer**; email route enabled.
Test Steps:

1. Trigger email summary action.
2. Observe UI and network response.

Test Data: Recipient list derived from enrolled students.
Expected Result: Success toast/log; API returns success.

Actual Result: Success message displayed; 2xx observed in Network.
Status: Pass

Test Case ID: TC_SSO_001
Test Scenario: Microsoft SSO login with valid UOW account
Description: Verify SSO path authenticates and redirects correctly.
Preconditions: None; SSO provider reachable.
Test Steps:

1. Click *Sign in with Microsoft*.
2. Complete UOW SSO flow.

Test Data: Valid UOW credentials.
Expected Result: Authenticated session; redirected to role dashboard.
Actual Result: Session established; redirect successful.
Status: Pass

Test Case ID: TC_RBAC_001
Test Scenario: Student denied access to lecturer QR management
Description: Ensure role-based access control prevents unauthorized access.
Preconditions: Logged in as **Student**.
Test Steps:

1. Navigate to lecturer QR generation route.

Test Data: Student account.
Expected Result: 403 or redirect to student area.
Actual Result: Access denied; no sensitive data shown.
Status: Pass

Test Case ID: TC_QR_EXP_001
Test Scenario: Check-in exactly at validity start boundary
Description: Validate inclusive start boundary behaviour.
Preconditions: Logged in as **Student**; live QR validity window about to start.
Test Steps:

1. Attempt check-in at exact start timestamp.

Test Data: Start time T0.
Expected Result: Check-in accepted.
Actual Result: Accepted at boundary.
Status: Pass

Test Case ID: TC_QR_EXP_002
Test Scenario: Check-in exactly at validity end boundary
Description: Validate exclusive end boundary behaviour.
Preconditions: Logged in as **Student**; first window about to end.
Test Steps:

1. Attempt check-in at exact end timestamp.

Test Data: End time T1.
Expected Result: Rejected if end is exclusive; otherwise accepted per spec.
Actual Result: Rejected at end boundary.
Status: Pass

Test Case ID: TC_GEO_PERM_001
Test Scenario: Location permission denied
Description: Ensure UI prompts and prevents in-person check-in when permission is denied.
Preconditions: Logged in as **Student**; browser location permission explicitly denied.
Test Steps:

1. Open scan URL.
2. Deny location permission.

Test Data: Chrome site settings deny.
Expected Result: UI shows prompt/instruction; in-person action disabled; online path shown if allowed.
Actual Result: Matches expectation.
Status: Pass

Test Case ID: TC_NET_001
Test Scenario: Network failure and retry
Description: Verify resilient handling of transient network errors.
Preconditions: Logged in as **Student**.
Test Steps:

1. Trigger check-in.
2. Simulate offline/timeout.
3. Retry after connectivity restores.

Test Data: DevTools offline or throttling.
Expected Result: Graceful error; no duplicate records; retry succeeds.
Actual Result: Works as expected.
Status: Pass

Test Case ID: TC_CONC_001
Test Scenario: Concurrency duplicate prevention
Description: Prevent double submission when pressing check-in rapidly.
Preconditions: Logged in as **Student**; valid window active.
Test Steps:

1. Tap check-in multiple times quickly.

Test Data: Mobile device rapid taps.
Expected Result: One attendance record; UI disables during request.
Actual Result: Single record persisted.
Status: Pass

Test Case ID: TC_LATE_001
Test Scenario: Late check-in flow after first window expires
Description: Verify late status recorded when outside first window but within designated late policy.
Preconditions: Logged in as **Student**; first window ended; late policy enabled.
Test Steps:

1. Attempt check-in post-window per policy.

Test Data: Late threshold configured.
Expected Result: Attendance stored with status = Late.
Actual Result: Late status saved.
Status: Pass

Test Case ID: TC_ANALYTICS_002
Test Scenario: Analytics empty state
Description: Validate user feedback when no data matches filters.
Preconditions: Logged in as **Lecturer**.
Test Steps:

1. Apply filters with no matching sessions.

Test Data: Date range outside semester.
Expected Result: Empty state message; no errors.
Actual Result: Empty state shown.
Status: Pass

Test Case ID: TC_EMAIL_002
Test Scenario: Email summary failure path
Description: Surface backend failure gracefully.
Preconditions: Logged in as **Lecturer**.
Test Steps:

1. Trigger email summary with simulated 5xx.

Test Data: Force 500 via mock/staging flag.

Expected Result: Error toast; no crash; retry option.
Actual Result: Error surfaced; UI stable.
Status: Pass

Test Case ID: TC_MANUAL_001
Test Scenario: Lecturer manual attendance entry
Description: Verify lecturer can add a student who lacks a device.
Preconditions: Logged in as **Lecturer**; session active.
Test Steps:

1. Open attendance management.
2. Add student manually.

Test Data: Student ID enrolled in subject.
Expected Result: Manual record created; visible in analytics.
Actual Result: Record present and counted.
Status: Pass

Test Case ID: TC_PRIV_001
Test Scenario: Privacy—GPS coordinates not persisted
Description: Ensure system does not store raw GPS after validation.
Preconditions: Logged in as **Student**; perform in-person check-in.
Test Steps:

1. Complete check-in.
2. Inspect network payloads and DB records.

Test Data: Normal in-radius check-in.
Expected Result: DB contains attendance result without raw GPS coordinates.
Actual Result: No GPS stored beyond transient validation.
Status: Pass

B Calculation Methodology

B.1 Basic Attendance Calculations

Session Attendance Scoring

$$\text{Session Points} = \begin{cases} 100, & \text{if check-in count} \geq 2 \\ 50, & \text{if check-in count} = 1 \\ 0, & \text{if check-in count} = 0 \end{cases} \tag{1}$$

Overall Attendance Percentage

$$\text{Attendance \%} = \frac{\sum \text{Session Points}}{(\text{Total Sessions} \times 100)} \times 100 \tag{2}$$

Where:

- $\sum$ Session Points = Sum of all earned points across all sessions
- Total Sessions = Count of all QR sessions created for the course
- Maximum possible points per session = 100

SQL Implementation for Attendance Calculation

```
ROUND(
  (SUM(CASE
    WHEN checkin_count >= 2 THEN 100
    WHEN checkin_count = 1 THEN 50
    ELSE 0
  END) / (COUNT(sessions) * 100)) * 100,
  1
)
```

On-Time Percentage Calculation

$$\text{On-Time }\% = \frac{\text{Check-ins within 15 minutes of start time}}{\text{Total check-ins}} \times 100 \tag{3}$$

Allowable Absences Calculation

$$\text{Classes Can Miss} = \lceil \text{Total Sessions} \times 0.2 \rceil - \text{Classes Already Missed} \tag{4}$$

Where:

- 0.2 represents the 20% allowable absence rate (80% requirement)
- $\lceil\rceil$ denotes ceiling function for conservative estimates
- Negative values indicate attendance requirement not met

B.2 Advanced Analytics Formulas

Time-Based Semester Segmentation

$$\text{Early Period} : \text{weeks}[0 : \lceil n/3 \rceil] \tag{5}$$

$$\text{Mid Period} : \text{weeks}[\lceil n/3 \rceil : \lceil 2n/3 \rceil] \tag{6}$$

$$\text{Late Period} : \text{weeks}[\lceil 2n/3 \rceil : n] \tag{7}$$

$$\text{Period Average} = \sum(\text{attendance}_i)/\text{period\_length} \tag{8}$$

Where n = total number of weeks in semester

Risk Categorization Algorithm

$$\text{Risk Level} = \begin{cases} \text{Critical}, & \text{if attendance} < 60\% \\ \text{Moderate}, & \text{if } 60\% \leq \text{attendance} < 75\% \\ \text{Watch List}, & \text{if trend} = \text{'declining'} \wedge \text{attendance} \geq 75\% \\ \text{Low Risk}, & \text{if attendance} \geq 75\% \wedge \text{trend} \neq \text{'declining'} \end{cases} \tag{9}$$

Trend Direction Detection

$$\text{Recent Average} = \frac{1}{3} \times \sum(\text{attendance}_i) \text{ for } i \in \text{last 3 weeks} \tag{10}$$

$$\text{Earlier Average} = \frac{1}{3} \times \sum(\text{attendance}_i) \text{ for } i \in \text{first 3 weeks} \tag{11}$$

$$\text{Trend Change} = \text{Recent Average} - \text{Earlier Average} \tag{12}$$

$$\text{Trend Direction} = \begin{cases} \text{'improving'}, & \text{if Trend Change} > 2\% \\ \text{'declining'}, & \text{if Trend Change} < -2\% \\ \text{'stable'}, & \text{otherwise} \end{cases} \tag{13}$$

Attendance Forecasting (Linear Projection)

$$\text{Recent\_Avg} = \frac{1}{4} \times \sum(\text{attendance}_i) \text{ for } i \in \text{last 4 weeks} \tag{14}$$

$$\text{Earlier\_Avg} = \frac{1}{4} \times \sum(\text{attendance}_i) \text{ for } i \in \text{weeks 4-8} \tag{15}$$

$$\text{Trend\_Rate} = \text{Recent\_Avg} - \text{Earlier\_Avg} \tag{16}$$

$$\text{Forecast\_Week}_1 = \text{Recent\_Avg} + \text{Trend\_Rate} \tag{17}$$

$$\text{Forecast\_Week}_2 = \text{Forecast\_Week}_1 + (\text{Trend\_Rate} \times 0.7) \tag{18}$$

Note: Forecasts are bounded [0, 100]

Performance Distribution Metrics

$$\text{Peak Attendance} = \max\{\text{attendance}_i | i \in \text{all weeks}\} \tag{19}$$

$$\text{Average Weekly} = \frac{1}{n} \times \sum(\text{attendance}_i) \text{ for all } n \text{ weeks} \tag{20}$$

$$\text{Weeks Above Target} = |\{\text{week}_i | \text{attendance}_i \geq 80\%\}| \tag{21}$$

Check-in Type Distribution

$$\text{Type Percentage} = \frac{\text{Count\_type}}{\text{Total\_checkins}} \times 100 \tag{22}$$

Where Type $\in$ {in-person, online, manual}

Minimum Points Required

$$\text{Required Points} = \frac{\text{Total Possible Points} \times \text{Required \%}}{100} \tag{23}$$

$$\text{Total Possible Points} = \text{Session Count} \times 100 \tag{24}$$

$$\text{Attended Sessions} = |\{\text{session}_i | \text{checkin\_count}_i > 0\}| \tag{25}$$

Consistency Variance Metric

$$\text{Total Variance} = \sum |\text{attendance}_i - \text{attendance}_{i-1}| \text{ for } i = 1 \text{ to } n - 1 \tag{26}$$

$$\text{Consistency Rating} = \begin{cases} \text{'excellent'}, & \text{if Total Variance} < 5 \\ \text{'good'}, & \text{if } 5 \leq \text{Total Variance} < 10 \\ \text{'moderate'}, & \text{if } 10 \leq \text{Total Variance} < 15 \\ \text{'high variability'}, & \text{if Total Variance} \geq 15 \end{cases} \tag{27}$$

Benchmarking Gap Analysis

$$\text{Performance Gap} = \text{Target Percentage} - \text{Current Average} \tag{28}$$

$$\text{Progress Ratio} = \min\left(\frac{\text{Current Average}}{\text{Target}} \times 100, 100\right) \tag{29}$$

$$\text{Status} = \begin{cases} \text{'achieved'}, & \text{if Current Average} \geq \text{Target} \\ \text{'on track'}, & \text{if Gap} \leq 5\% \\ \text{'needs focus'}, & \text{if Gap} > 5\% \end{cases} \tag{30}$$

At-Risk Student Count

```
SELECT COUNT(*) as at_risk_count
FROM (
  SELECT student_id,
    ROUND((SUM(points) / (COUNT(*) * 100)) * 100, 1) as attendance_pct
  FROM student_attendance
  GROUP BY student_id
  HAVING attendance_pct < 80
) at_risk_students
```

Performance Level Distribution

$$\text{Excellent} = |\{\text{student}_i | \text{attendance}_i \geq 90\%\}| \tag{31}$$

$$\text{Good} = |\{\text{student}_i | 80\% \leq \text{attendance}_i < 90\%\}| \tag{32}$$

$$\text{Average} = |\{\text{student}_i | 70\% \leq \text{attendance}_i < 80\%\}| \tag{33}$$

$$\text{Poor} = |\{\text{student}_i | \text{attendance}_i < 70\%\}| \tag{34}$$

C API Endpoints

This appendix provides an overview of the available backend API endpoints, as documented and visualized using *SwaggerUI*. Each category corresponds to a logical group of routes within the backend server, providing a clear interface for frontend-backend integration.

Auth Endpoints The **Auth** category contains endpoints related to authentication, user registration, and session management. These APIs handle credential verification, user creation, and access token validation.

- **POST /api/auth/email-exists**
  Checks whether a user email already exists in the system. This endpoint is used during signup validation to prevent duplicate accounts.
- **GET /api/auth/me**
  Retrieves the currently authenticated user's information, useful for maintaining session persistence on the frontend.
- **POST /api/auth/signin**
  Signs in a user using email and password credentials. Returns session information or authentication tokens upon successful login.
- **POST /api/auth/signout**
  Logs out the currently authenticated user, clearing the active session.
- **POST /api/auth/signup**
  Registers a new user with email and password credentials, creating an entry in the system's user database.

Student Endpoints The **Student** category includes endpoints that allow students to interact with their enrolled subjects, view session details, and check attendance using QR codes.

- **POST /api/student/attendance/checkin**
  Enables a student to check in to a class or study session by scanning a QR code. This endpoint records the attendance in the database.
- **GET /api/student/subjects**
  Retrieves the list of subjects in which the student is enrolled, including grouped study sessions and related attendance data.

Lecturer Endpoints Lecturer endpoints support lecturers in managing study sessions, generating and distributing QR codes, and monitoring attendance data.

- **POST /api/lecturer/manual-checkin**: Manually check in a student.
- **GET /api/lecturer/qr-codes/all**: Retrieve all QR codes for the lecturer's study sessions.
- **GET /api/lecturer/rooms**: List all available rooms.
- **POST /api/lecturer/send-attendance-emails**: Send automatic attendance reminder emails.

- **POST /api/lecturer/send-qr-email**: Email QR code links or images to enrolled students.
- **GET /api/lecturer/session-types**: Get session types taught by the current lecturer.
- **GET /api/lecturer/study-session/{id}/at-risk-students**: Retrieve students at risk based on attendance.
- **GET /api/lecturer/study-session/{id}/checkin-list**: View the list of students who checked in for a session.
- **GET /api/lecturer/study-session/{id}/details**: Get detailed study session information.
- **GET /api/lecturer/study-session/{id}/qr/{qr_code_id}**: Retrieve QR code data for a session.
- **POST /api/lecturer/study-session/{id}/qr**: Generate a new QR code for a study session.
- **PUT /api/lecturer/study-session/{id}/qr**: Update an existing QR code for a study session.
- **GET /api/lecturer/study-session/{id}/rooms**: List rooms used for a specific study session.
- **GET /api/lecturer/study-session/{id}/student-list**: Retrieve enrolled students for a specific session.
- **GET /api/lecturer/study-session/{id}/timeline**: Fetch session timeline and event data.
- **GET /api/lecturer/subjects**: Retrieve subjects and grouped study sessions taught by the lecturer.

Common Endpoints Common endpoints provide utilities available to all authenticated users, including QR code validity checks.

- **GET /api/qr/{qr_code_id}**: Retrieve QR code validity, session, and geofence information.

Statistics Endpoints These endpoints provide analytical data and attendance statistics for lecturers and administrators to monitor performance trends and course analytics.

- **GET /api/analytics/attendance-distribution**: Retrieve attendance performance distribution.
- **GET /api/analytics/available-courses**: Retrieve courses available for analytics.
- **GET /api/analytics/key-metrics**: Get attendance-related metrics and insights.
- **GET /api/analytics/lecturer-trends**: Retrieve lecturer-specific performance analytics.
- **GET /api/analytics/student-performance**: Retrieve performance data for an individual student.
- **GET /api/analytics/subject-summary**: Get aggregated statistics per subject.
- **GET /api/analytics/weekly-attendance**: Retrieve attendance data per week.

System Endpoints System-level endpoints handle internal automation, maintenance tasks, and email scheduling. These endpoints require elevated privileges for access.

- **POST /api/system/faker**: Generate dummy users and simulate check-ins for testing.
- **GET /api/system/init**: Initialize system services and schedulers.
- **POST /api/system/lecture-end-trigger**: Trigger end-of-lecture attendance emails.
- **POST /api/system/send-stu-report**: Send student attendance reports via email.

Analytics Endpoints These endpoints extend statistical insights, focusing on individual attendance check-in behavior and frequency.

- **GET /api/analytics/checkin-types**: Retrieve weekly breakdown of in-person vs online check-ins.
- **GET /api/analytics/student-recent-checkins**: Retrieve recent attendance check-ins for a student.