



In Partial Fulfillment of the Requirements for the

CS 314 – Software Engineering 2

**A COMPREHENSIVE MOBILE-BASED APPROACH TO SCHOOL EVENT
ATTENDANCE MONITORING**

Presented to:

Dr. Unife O. Cagas

Prepared by:

Nilo Jr. Olang

Arjhay-Jun P. Pastrana

Doña Jeva N. Estoque

BSCS-3A: Computer Science

May 2025



Table of Contents

CHAPTER 1 : INTRODUCTION	1
Project Context	1
General Objective	2
Specific Objectives.....	2
Scope and Limitations.....	2
CHAPTER 2 : TECHNICAL SPECIFICATIONS	4
System Overview	4
System Requirements.....	4
Use Case Diagram.....	5
Class Diagram	6
Activity Diagram	7
Sequence Diagram	8
Architectural Design.....	9
System Components.....	10
Layers in the Architecture	11
CHAPTER 3 : IMPLEMENTATION AND TESTING	13
Data Gathering	13
Implementation Method	14
Testing Phase	14
Testing Objectives	14
Testing Methods	15
Test Environment.....	15



Test Cases and Results	16
Testing Metrics	16
System Performance	17
CHAPTER 4 : DEPLOYMENT AND MAINTENANCE	18
Deployment Environment.....	18
Deployment Process.....	18
User Accessibility.....	18
Data Integrity and Backup.....	18
System Scalability.....	19
Bug Fixing and Updates.....	19
Performance Monitoring.....	19
Future Enhancements.....	19
USING THE WEB-BASED EVENT ATTENDANCE SYSTEM.....	19
Dashboard Overview	20
Checking In to an Event.....	21
Managing Events	22
Managing Universities.....	23
CONCLUSION.....	24
REFERENCES:	25
USER MANUAL	27



CHAPTER 1

INTRODUCTION

Project Context

School events are important because they help students stay engaged, improve learning experiences, and build a sense of community. However, keeping track of attendance at these events is still a big challenge, especially at big universities like Surigao del Norte State University (SNSU). Many schools continue to use outdated manual logging or paper-based sign-in methods, which are exhausting, prone to mistakes, and vulnerable to attendance fraud, including proxy sign-ins. Many institutions, including SNSU, have yet to put in place a thorough and fraud-resistant framework in spite of technological improvements. Moreover, the absence of real-time monitoring of attendance records hinders administrators from efficiently managing event attendance. A system that provides instant verification and automates attendance tracking can help schools improve efficiency and reduce errors.

Several studies have researched into ways to make school attendance tracking better like new technology offers solutions to automate and improve attendance monitoring. Studies have shown that GPS tracking, QR code scanning, and biometric verification can significantly enhance accuracy and efficiency in attendance systems [1]. But these systems also have drawbacks, such as technical limitations, security flaws, and issues with system dependability. Additional studies demonstrate the efficacy of RFID-based attendance tracking, which improves visibility and accountability by sending parents real-time SMS warnings [7]. However, existing solutions face challenges related to data privacy, system reliability, and high implementation costs [2].

Despite these advancements, many schools continue to struggle with attendance fraud, slow processing times, and data management inefficiencies. The absence of a fraud-resistant, real-time, and user-friendly system limits event organizers from efficiently managing attendance records. Administrators are also unable to quickly detect and resolve attendance irregularities due to the lack of real-time tracking tools. Some schools use time-consuming, prone to errors, and slow manual verification procedures in an effort to address this problem. Furthermore, the usability of some digital attendance solutions is limited since they are not compatible with all devices. As a result, SNSU and similar institutions need a system that is safe, effective, and flexible in order to guarantee accurate attendance monitoring during educational events.

This research proposes a comprehensive mobile-based event attendance monitoring system that integrates GPS-based tracking, QR code scanning, and selfie verification with event backgrounds. This approach aims to eliminate attendance fraud, enhance efficiency, and provide real-time monitoring for administrators. According to research on biometric authentication and location-based tracking, these methods efficiently and user-friendly confirm a student's physical attendance at an event [3]. Students won't need to download extra apps because the suggested solution is made to work with mobile web browsers. The system guarantees accessibility, security, and efficiency by utilizing cloud storage and secure authentication techniques, which makes it the perfect choice for schools like SNSU wanting to enhance event attendance tracking.

General Objective

This study's primary goal is to develop and deploy a mobile web-based attendance system that enhances event attendance tracking's precision, security, and effectiveness. The technology offers a secure and dependable method of tracking attendance at SNSU events through GPS tracking, QR code scanning, and biometric verification.

Specific Objectives

1. Develop an Event Attendance Management.
2. QR Code-Based Attendance Tracking.
3. Locate and Visual Verification Features.
4. Automated Reporting and Data Analytics.

Scope and Limitations

This project was conducted at Surigao del Norte State University (SNSU) between the second and first semesters of the 2025-2026 academic year to develop and test the Event Attendance System. The system aims to enhance attendance tracking for school events by integrating QR code scanning, GPS tracking, and selfie verification. The proposed system will be tested at SNSU and evaluated for its accuracy, security, and efficiency.



The core features of the event attendance system include:

1. *Develop a System and Fraud Prevention.*

- 1.1 Develop an event attendance system based on Django to avoid misinformation and cheating
- 1.2 Secure methods of authentication should be implemented to confirm student attendance properly.
- 1.3 Design a structured database for efficient attendance record storage and retrieval.

2. *Locate and Visual Verification Features.*

- 2.1 Integrate location-based technologies to track student at scheduled event and verify participation.
- 2.2 Implement selfie verification with event backgrounds to prevent proxy attendance and enhance security.

3. *Perform a Performance Evaluation on System.*

- 3.1 Deliver real-time attendance updates and automate data entry to reduce administrative workload.
- 3.2 Compare proposed system performance against traditional methods in terms of accuracy, efficiency, and user satisfaction.
- 3.3 Conduct usability assessments, compile feedback, and refine features to enhance the overall experience and utility for user experience.

However, there are certain limitations to the Event Attendance System, including:

- The system requires a good internet connection for real-time updates, which might fail in areas with poor network coverage.
- Environmental variables including buildings, weather, and device limits can impact GPS accuracy.
- Poor lighting and photos might result in misleading rejection during verification.
- The system is intended for school-related events and may require adjustments for larger events.
- GPS monitoring and face verification raise privacy concerns and require strict data protection procedures to ensure user security.

CHAPTER 2

TECHNICAL SPECIFICATIONS

System Overview

The proposed Mobile-Based School Event Attendance Monitoring System is designed to provide a reliable and efficient way to track attendance at school events. The system uses GPS tracking, QR code scanning, and selfie verification to confirm student presence at an event. It will be developed as a mobile application, with an admin dashboard accessible on both mobile and desktop devices for monitoring and managing attendance records.

System Requirements

Hardware Requirements

To develop and run the system efficiently, the following hardware is recommended:

For Development (Laptops/PCs)

- Processor: Minimum Intel Core i3 (8th Gen) or AMD Ryzen 5 (3000 Series)
- RAM: At least 8 GB (16 GB recommended for better performance)
- Storage: Minimum 256 GB SSD (512 GB recommended)

For Mobile Devices (Users & Admins)

- Operating System: Android 10 (Oreo) or later
- RAM: At least 4 GB
- Browser: Chrome, Firefox, Edge, Safari
- Camera: Required for selfie verification
- GPS Support: Required for location tracking

Software Requirements

- VS Code – For coding and testing the web application
- Python Django – Web framework for backend development
- Sqlite3 – Database for storing user, event details, and attendance records
- APIs & Services:
 - Mapbox API – For GPS tracking and displaying event locations
 - QR Code Generator & Scanner – For event-based check-in
 - Django REST Framework – For API development

TECHNICAL FIGURES

Use Case Diagram

The **Figure 1** use case diagram of the event attendance monitoring system, illustrating the interactions between users and the system's main functionalities. The diagram showcases the roles of students and admins, along with their corresponding use cases, such as attendance verification, event creation, and survey distribution.

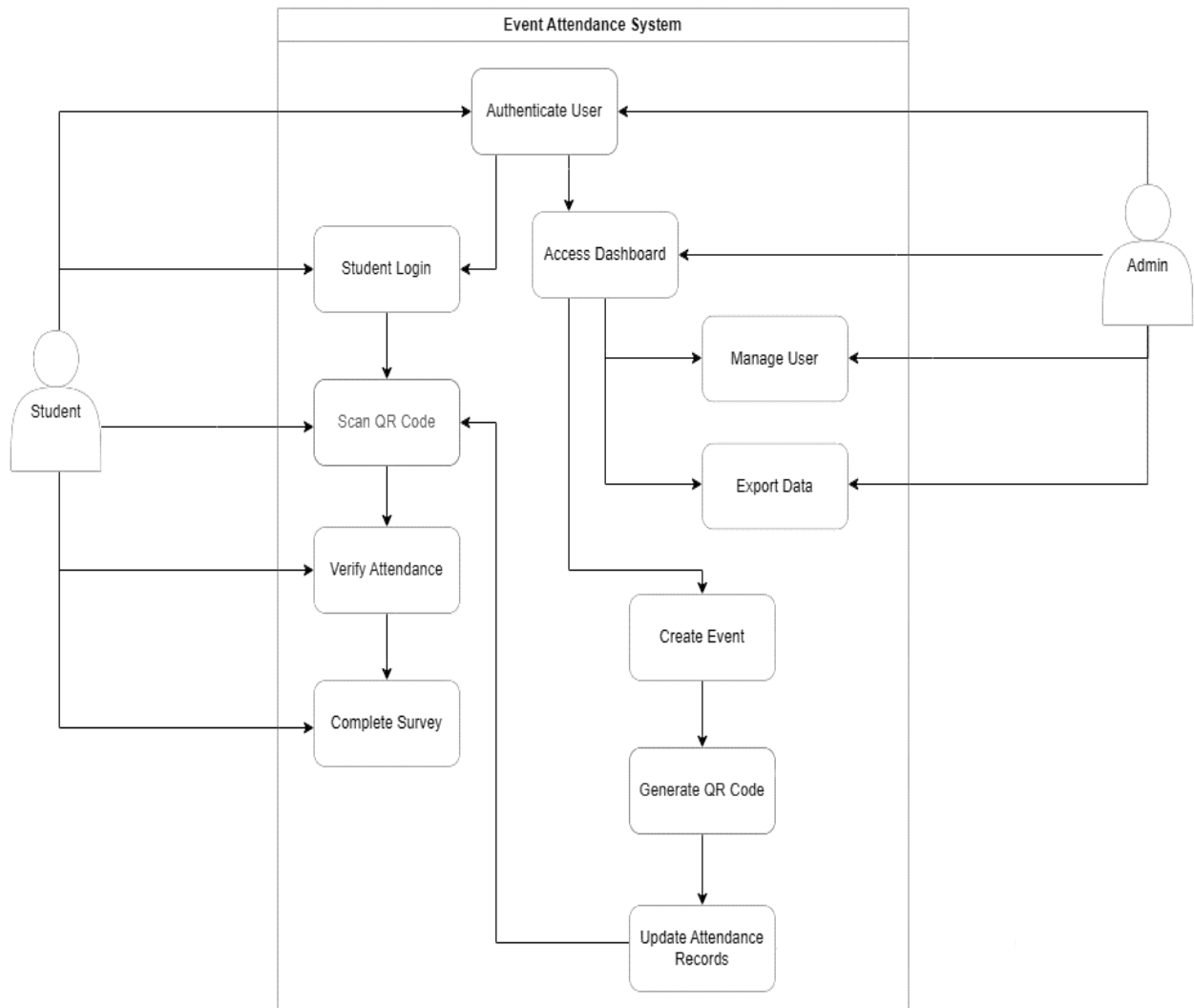


Figure 1: Use-case Diagram for Event Attendance

The use case diagram (Figure 3) visually represents the key functionalities of the Event Attendance System and the positions of its actors. Students participate with the system through signing in, scanning QR codes, checking attendance, and filling out questionnaires. Administrators may manage users, create events, and export attendance data. The system automates essential tasks including user identification, QR code production, and attendance record changes. These interactions guarantee that the event attendance monitoring system is safe and efficient, with real-time data updates and record keeping for the SNSU.

Class Diagram

The **Figure 2** provides a structured representation of the system, showcasing the relationships between different classes and their attributes. It highlights how various components interact to ensure an efficient event attendance monitoring process.

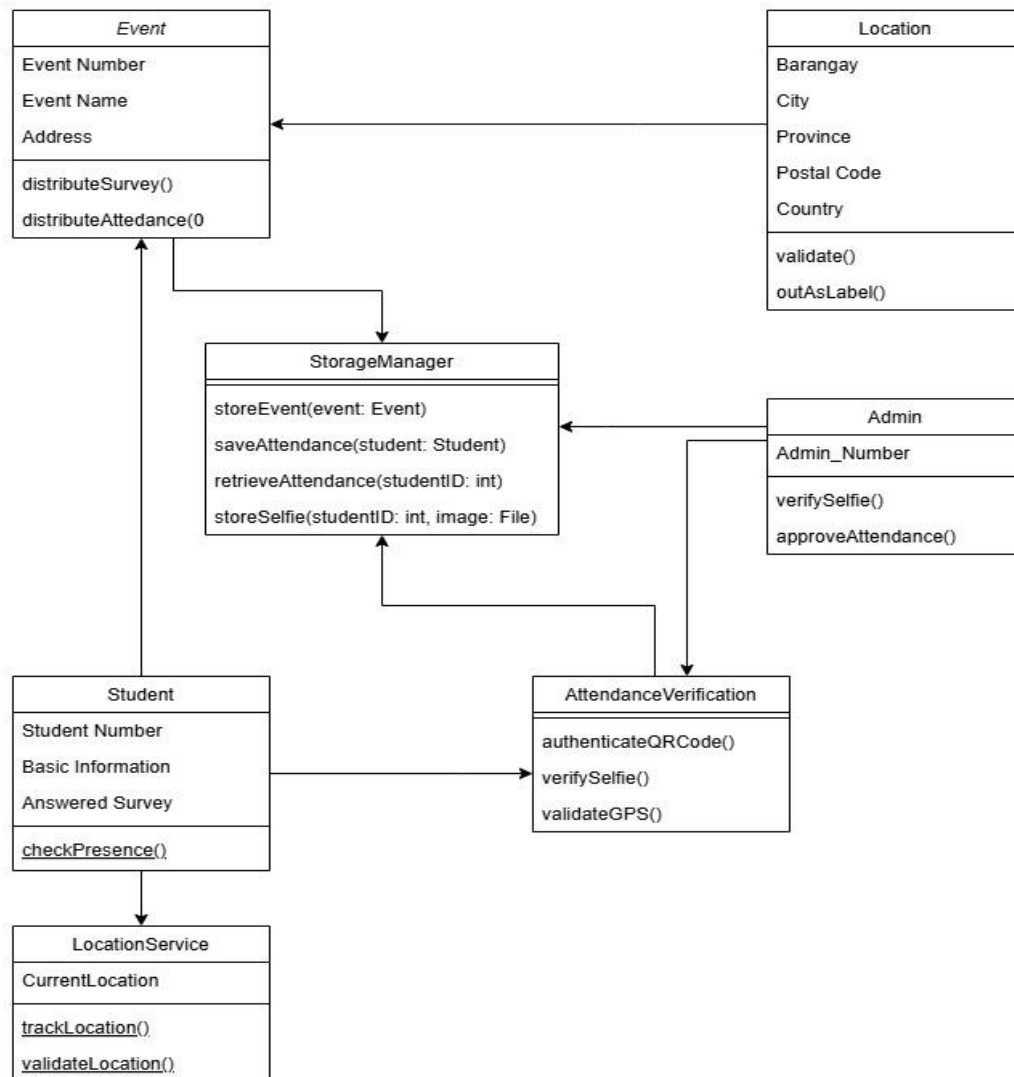


Figure 2: Class Diagram for Event Attendance

The class diagram defines the core components of the system and their interconnections. The Event class stores event details such as name and address, enabling functionalities like attendance tracking and survey distribution while integrating GPS location verification. The Student class represents attendees, incorporating methods like `IsPresentInEvent()` to confirm attendance through GPS, QR codes, and selfies, reducing fraudulent entries. The Admin class oversees the entire process, using methods such as `VerifySelfie()` to validate attendance by comparing

selfies to event backgrounds and ApproveStudentAttendance() to handle exceptions. The LocationService class manages real-time GPS tracking via updateLocation(). Additionally, the StorageManager class securely handles event data, attendance logs, and media files, ensuring efficient data retrieval and storage.

Activity Diagram

The **Figure 3** represents the web application workflow, from user action that determines whether the user is an admin. If the user is an admin, they can generate a QR code, request a unique ID, and authenticate themselves to create an event.

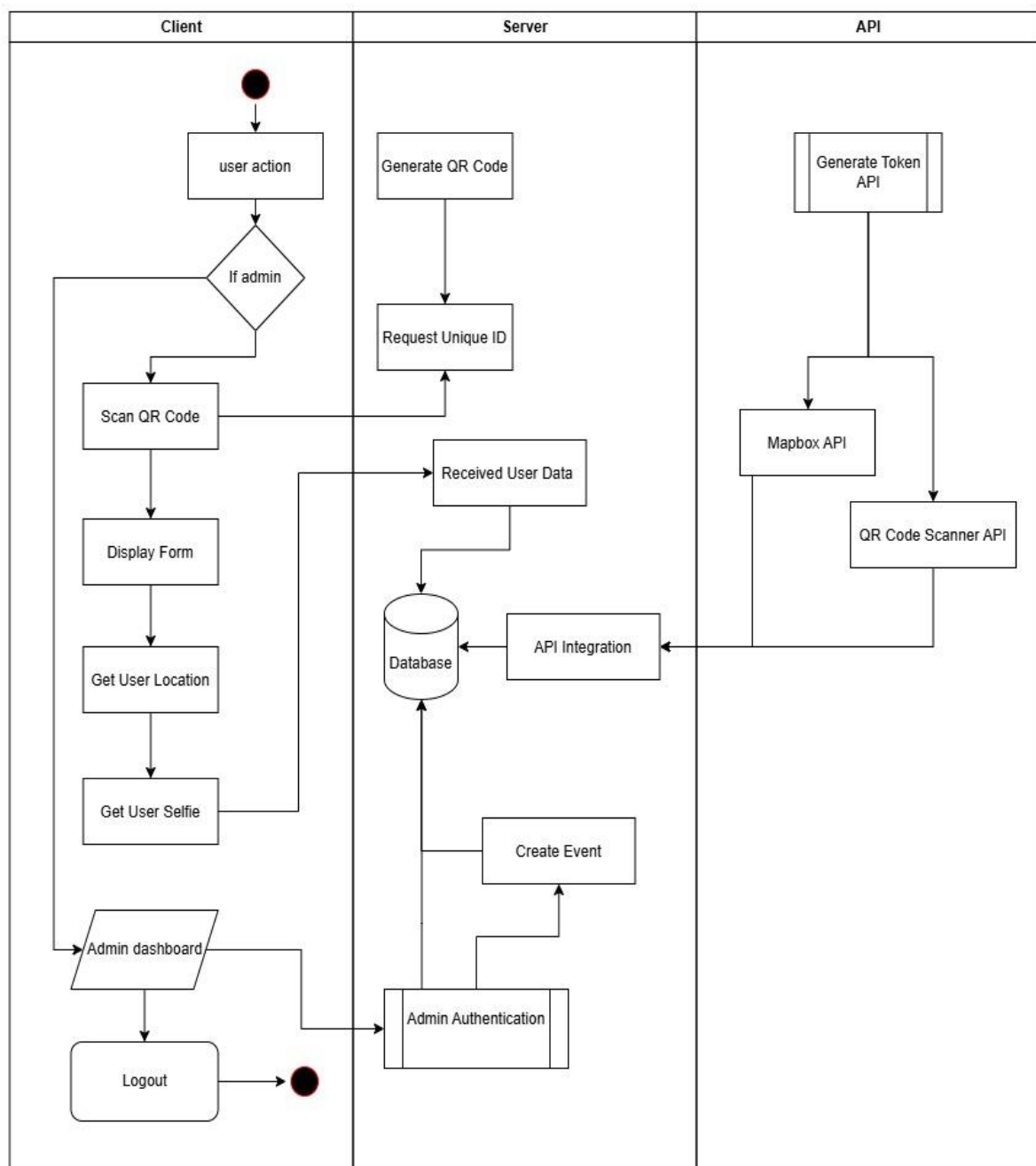


Figure 3: Activity Diagram for Event Attendance

The activity diagram outlines the structured workflow of the web application, detailing the interactions between the client, server, and external APIs. It distinguishes between regular users and admins, where admins are responsible for creating events and managing attendance, while regular users scan a QR code, input their location, and capture a selfie for verification. The system integrates APIs such as the QR Code Scanner and Mapbox for automated data retrieval and location tracking. Admin authentication is required before event creation, enhancing security and restricting access to authorized users. The use of QR codes streamlines the process, while selfie and location capture provide an extra layer of verification.

Sequence Diagram

The **Figure 4** sequence diagram interact among the system components in the event attendance monitoring process. It outlines how students, admins, the database, and the system communicate to facilitate attendance tracking, authentication, and survey distribution.

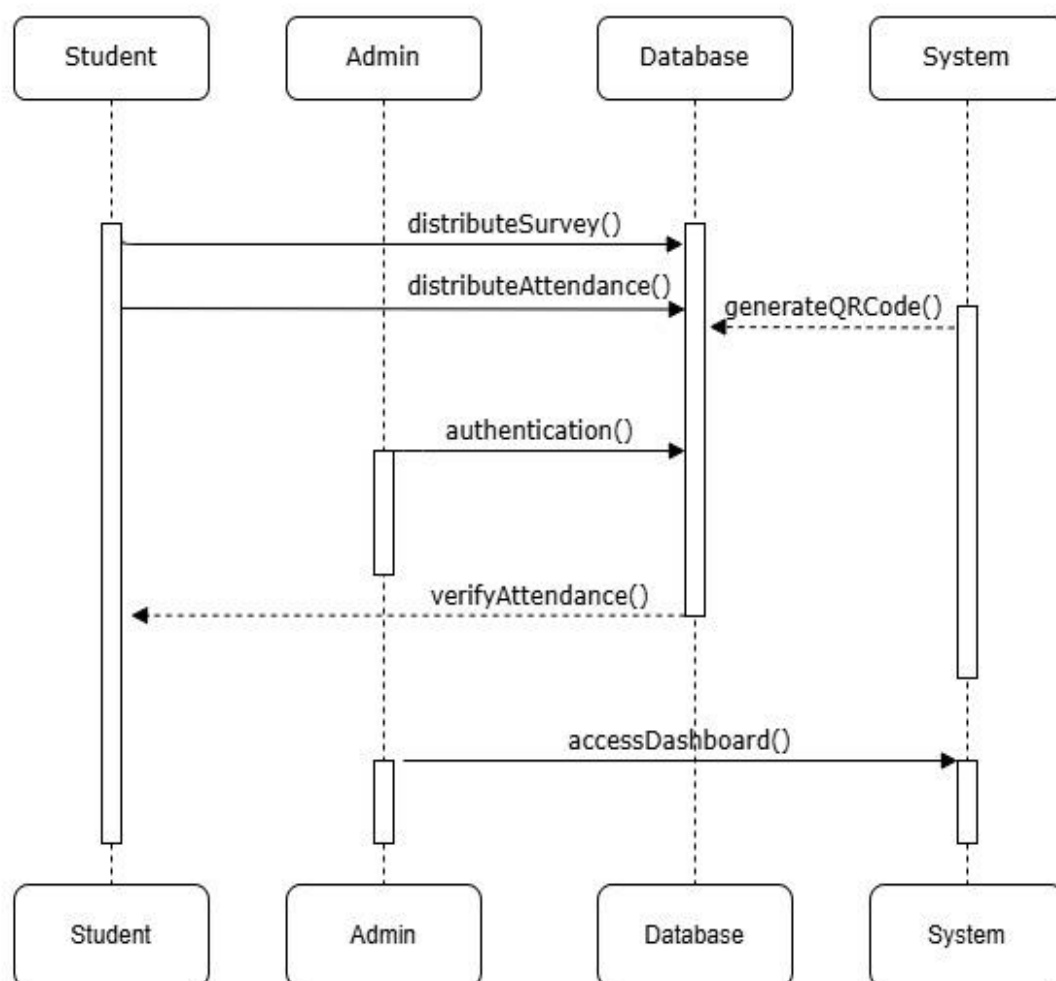


Figure 4: Sequence Diagram for Event Attendance

The sequence diagram details the structured exchange of information in the system, emphasizing the role of each participant. The process begins with students receiving surveys and attendance distribution from the database. The admin initiates authentication to manage event access and verifies attendance before updating the database. A QR code is generated by the system to facilitate attendance tracking, ensuring a secure and streamlined process. Finally, admins access the dashboard to monitor attendance records.

Architectural Design

The architectural design of the Mobile-Based School Event Attendance Monitoring System follows a client-server model, ensuring a modular, scalable, and efficient approach for real-time attendance tracking. The system integrates GPS tracking, QR code scanning, and selfie verification while maintaining a secure and user-friendly experience for both students and administrators.

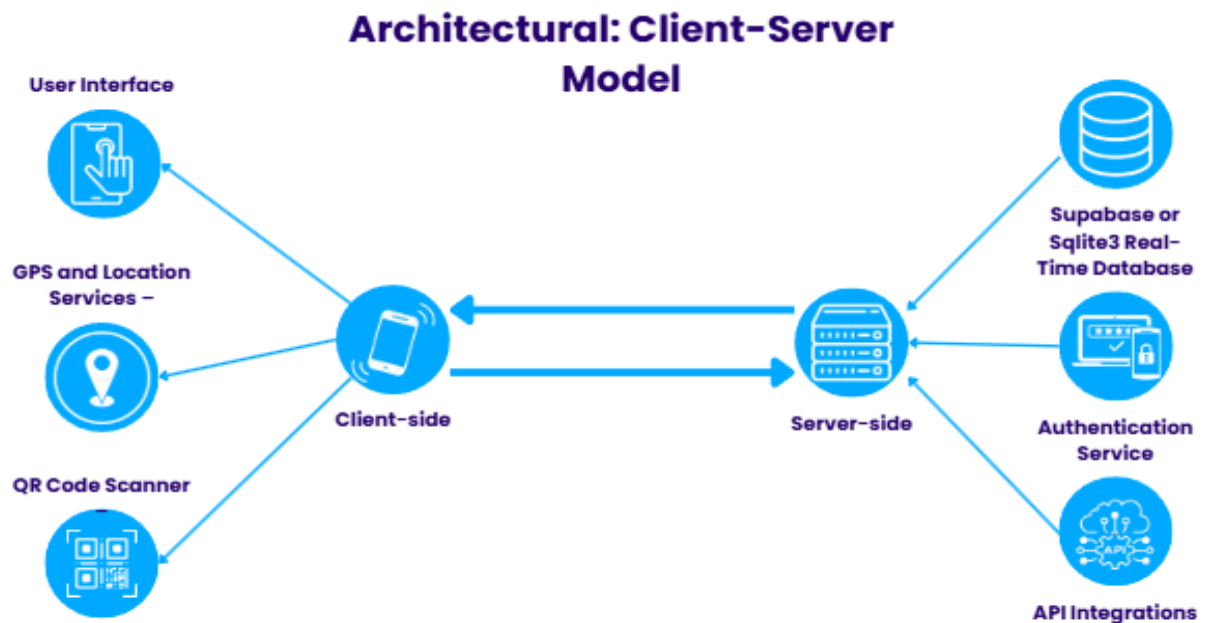


Figure 5. Architectural: Client-Server Model

Architectural Style: Client-Server Model

- The client-side runs on mobile devices (Android) and provides an interactive interface for students to check in at school events using GPS, QR codes, and selfies.
- The server-side processes and stores attendance data, user profiles, and event details using a cloud-based database. This ensures real-time updates and secure data management.

Key Components

1. Frontend (Client-Side) – Web-based UI
 - Built using HTML, CSS, JavaScript, and Django Templates
 - Collects student location (GPS), scans QR codes, and captures selfies for verification.
 - Sends attendance data to the server for processing.
2. Cloud Database & Backend (Server-Side)
 - Uses PostgreSQL or Sqlite3 for storing attendance records, student information, and event details.
 - Implements user authentication and event management
 - Provides API endpoints for QR scanning and GPS validation
3. Admin Dashboard (Web & Mobile Access)
 - Accessible via web browsers on desktop and mobile
 - Allows administrators to monitor attendance records in real time.
 - Provides tools for verifying attendance and generating reports.

System Components

The Web-Based School Event Attendance Monitoring System consists of two main components: the frontend (web application) and the backend (server-side services). These components work together to ensure a smooth, accurate, and real-time attendance tracking experience for school events.

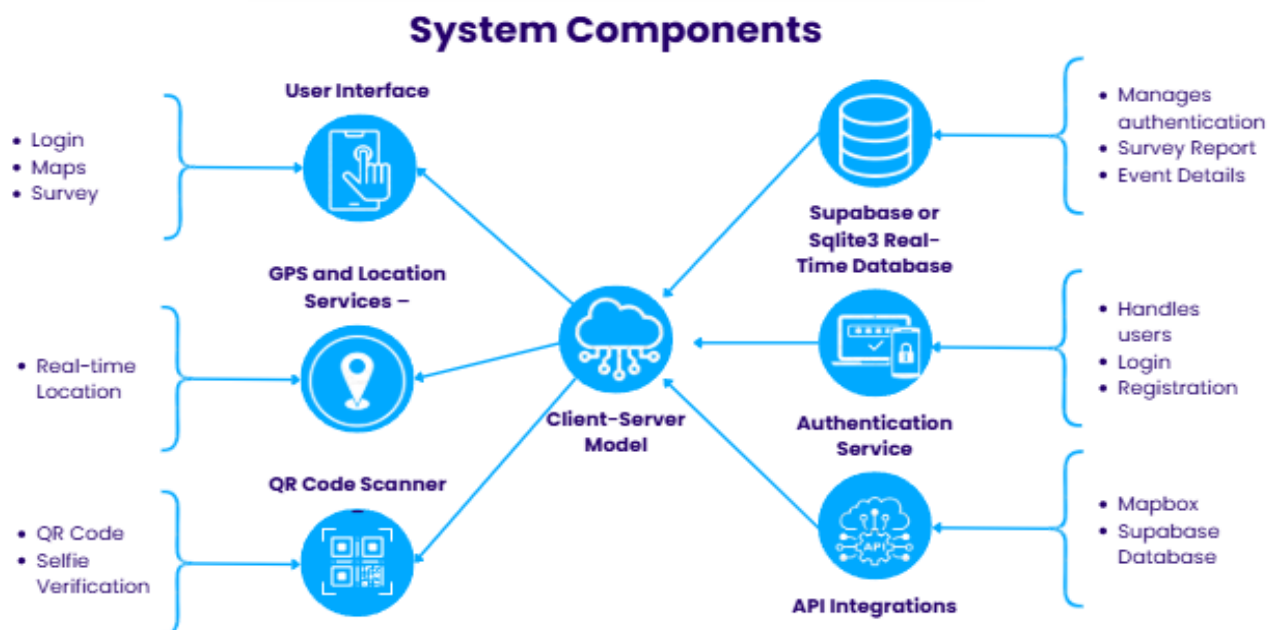


Figure 6. Architectural: System Components

Client-Side (Web Application)

- User Interface (UI) – Built using Django templates and JavaScript for an interactive experience.
- GPS and Location Services – Uses Mapbox API for real-time location tracking
- QR Code Scanner – Allows students to scan event-specific QR codes to confirm their attendance.

Server-Side (Cloud Services)

- PostgreSQL or Sqlite3 Real-Time Database
 - Manages authentication, attendance records, and event details
- Authentication Service
 - Manages user login and registration, ensuring only authorized users can access the system.
- API Integrations
 - Mapbox API – For geolocation and mapping of event venues.
 - QR Code Generator & Scanner API – For generating unique event QR codes and verifying attendance.

Layers in the Architecture

The Web-Based School Event Attendance Monitoring System follows a layered architecture to ensure clear separation of concerns and efficient system operation. It consists of three main layers: Presentation Layer (Frontend), Application Layer (Middleware), and Data Layer (Backend).

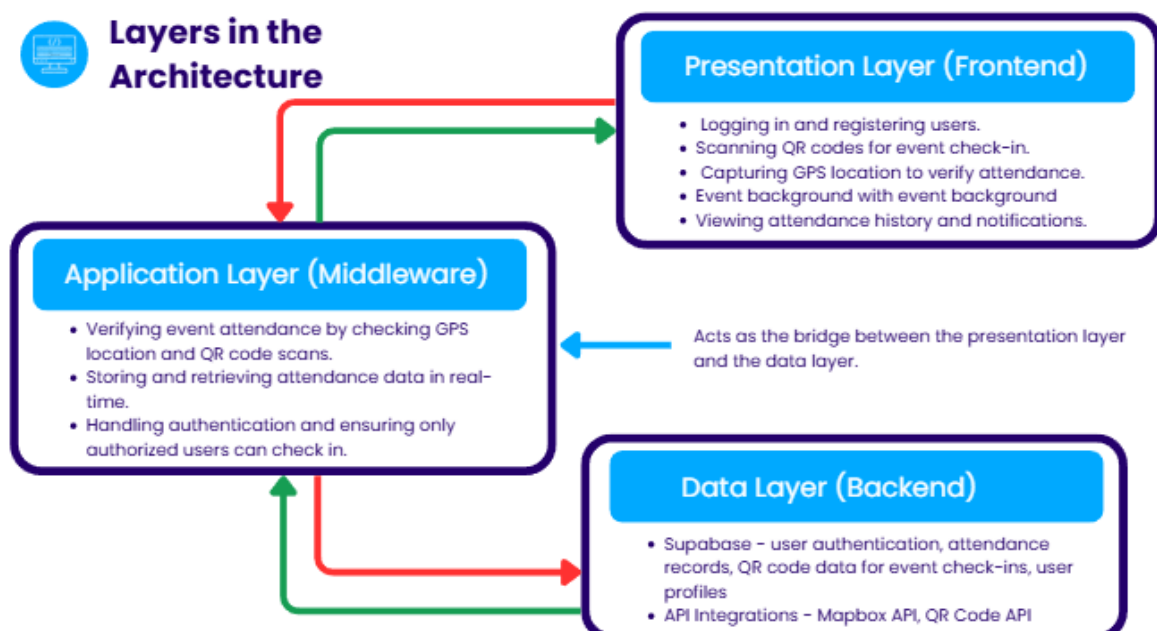


Figure 7. Architecture: Layers



a. Presentation Layer (Frontend)

- Logging in and registering users.
- Scanning QR codes for event check-in.
- Capturing GPS location to verify attendance.
- Taking selfies with the event background for additional verification.
- Viewing attendance history and notifications.

b. Application Layer (Middleware)

- Manages core logic of the system, including:
 - Verifying event attendance by checking GPS location and QR code scans.
 - Storing and retrieving attendance data in real-time.
 - Handling authentication and ensuring only authorized users can check in.
- Acts as the bridge between the frontend (web-based) and backend (database and APIs).

c. Data Layer (Backend)

- Manages all data storage and retrieval operations securely.
- PostgreSQL handles:
 - User authentication (login, registration).
 - Attendance records (event details, timestamps, location).
 - QR code data for event check-ins.
 - User profiles (student/admin roles).
- API Integrations:
 - Mapbox API – For tracking event locations and verifying GPS coordinates.
 - QR Code API – For generating and validating event check-in codes.

CHAPTER 3

IMPLEMENTATION AND TESTING

The system development follows a combination of the Agile and Rapid Application Development (RAD) Methodologies, ensuring an incremental and iterative approach. The key phases include:

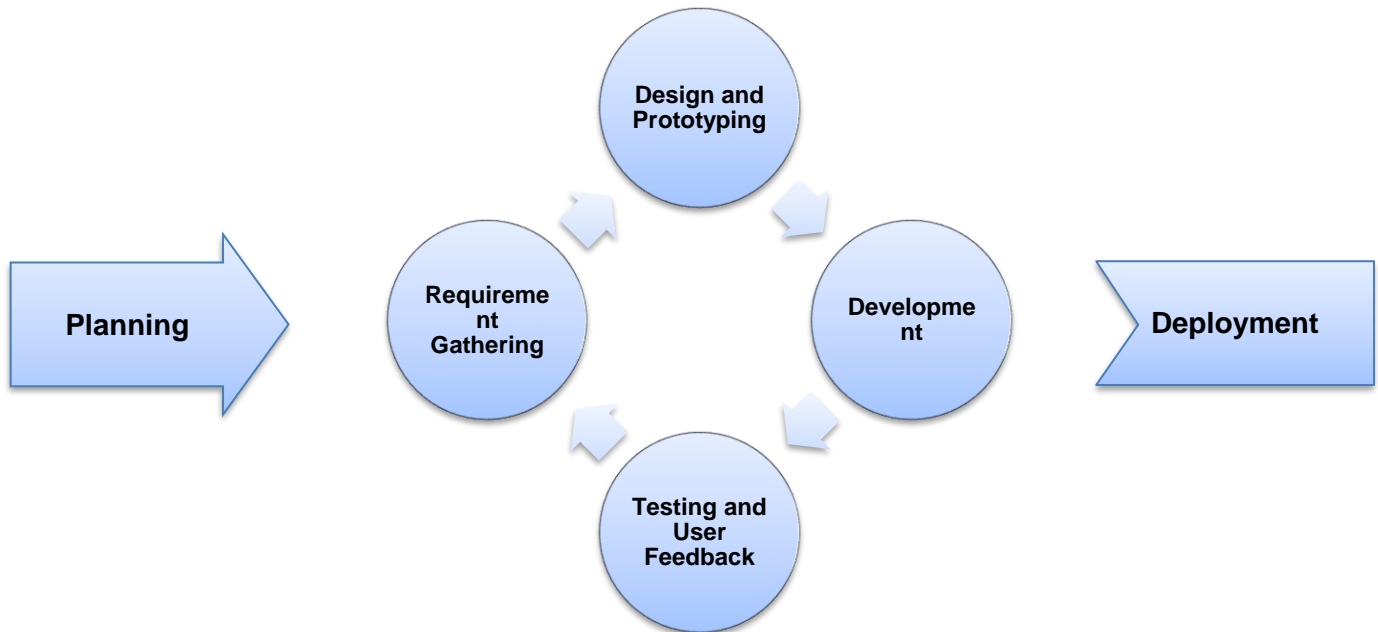


Figure 8. Agile and RAD Methodology Model

1. *Planning* – Identifying system requirements based on literature review and user feedback.
2. *Requirement Gathering* - the system features and functionalities were identified base on the data collected from the event organizer.
3. *Design and Prototyping* – Creating the system architecture, UI/UX design, and database structure.
4. *Development* – Developing the system using Django, SQLite, Mapbox API, and QR Code generator.
5. *Testing and User Feedback* – Conducting unit testing, integration testing, and user acceptance testing, also gathering user feedback, and making refinements.
6. *Deployment* – Deploying the system for a real event, which will release the official version of the system to use in school events.

Data Gathering

Data collection for evaluation and analysis will be conducted through:

1. *System Logs* – Monitoring system-generated logs to measure response times and data accuracy.
2. *User Surveys* – Collecting feedback from students and administrators about ease of use, effectiveness, and security.
3. *Comparison with Traditional Methods* – Measuring time efficiency and accuracy against manual sign-in sheets.

Implementation Method

The application was implemented using a structured development process that combined Agile and Rapid Application Development (RAD) methodologies. This approach ensured iterative development, continuous user feedback, and efficient system refinement.

Development Tools

- VS Code – Used for coding and debugging.
- Django Framework – Backend development.
- SQLite3/PostgreSQL – Database management.

APIs & Services

- Mapbox API – GPS tracking.
- QR Code Scanner & Generator API – Event check-in.
- Django REST Framework – API development.
- Render – Hosting services for production.

Testing Phase

The testing phase ensured that the Mobile-Based School Event Attendance Monitoring System functioned correctly, met performance standards, and provided a secure and reliable attendance tracking experience.

Testing Objectives

The testing process aimed to validate the system's functionality by ensuring that key features, such as QR code scanning, GPS tracking, and selfie verification, were implemented correctly. It also focused on verifying performance by assessing system responsiveness, data processing speed, and real-time updates. Additionally, security and fraud prevention mechanisms were tested, including authentication, data encryption, and fraud detection, to ensure a secure attendance tracking process. Lastly, the testing phase evaluated usability and user experience, ensuring that the interface was intuitive and accessible for both students and administrators.



Testing Methods

In **unit testing**, each feature was tested separately to make sure it worked correctly. This included checking the QR code scanner, GPS tracking, and selfie verification. The goal was to confirm that every function could run properly on its own without depending on other parts of the system.

For **integration testing**, the focus was on ensuring that different parts of the system worked well together. This involved testing how the frontend interface communicated with the backend system. The team checked whether the GPS coordinates were correctly recorded and whether the QR code scanner properly logged attendance.

During **system testing**, the entire application was tested under real-world conditions. Different event scenarios were simulated to see how the system performed when multiple students checked in at the same time. The team also tested the system's ability to handle different internet conditions, such as strong and weak network signals, to verify its reliability.

Lastly, **user acceptance testing (UAT)** was conducted with actual students and administrators at SNSU. The goal was to gather feedback on the system's usability, security, and overall performance. Users tested the system by checking in at events, scanning QR codes, and verifying their attendance. Their feedback helped identify areas that needed improvement before the final deployment.

Test Environment

The system was tested in an environment that met the hardware and software requirements outlined in Chapter 2. To simulate real-world scenarios, test data was carefully configured. Sample student and administrator accounts were created to test the login system, event creation, and attendance monitoring features. Multiple events with different locations were set up to evaluate GPS accuracy, while QR codes were generated for each event to verify proper scanning functionality. The system was also tested against potential fraudulent activities, such as proxy attendance, by attempting to use mismatched selfies or incorrect GPS locations. Furthermore, various network conditions were simulated to assess system performance under weak or strong internet connectivity. This comprehensive test environment allowed for an accurate evaluation of the system's performance, reliability, and security under different conditions.

Test Cases and Results

Test Case	Expected Outcome	Actual Outcome	Status
QR Code Scanning	Student successfully checks in using QR code		
GPS Location Tracking	System records student location at event		
Selfie Verification	System should be able to saved selfie from the attendees		
Fraud Prevention (Proxy Check-in)	System denies attendance if trying to enter another data		
Admin Dashboard	Admin views real-time attendance data		
Offline Mode Test	System should notify users of connectivity issues		

Testing Metrics

To evaluate the scalability and responsiveness of the Event Attendance System on a local server, various performance and load tests will be conducted.

- *Load Testing:* Simulate multiple concurrent users accessing the system to measure its ability to handle heavy traffic. This will be done using tools like Apache JMeter or Locust to determine the maximum number of users the system can support before performance degrades.
- *Stress Testing:* Push the system beyond its normal load conditions to assess its stability under extreme scenarios, such as multiple simultaneous QR code scans or high-volume event check-ins.
- *Functional Testing:* Ensure that 90% of all system functionalities are tested, covering key operations such as user authentication, event management, QR code scanning, and attendance tracking.



System Performance

- *Response Time Analysis:* Measure the average time taken for key system actions, such as QR code scanning, form submissions, and event data retrieval, ensuring minimal delays.
- *Database Query Performance:* Analyze database response times under normal and high-traffic conditions to detect slow queries and optimize data retrieval.
- *Resource Utilization Monitoring:* Track CPU, memory, and disk usage during test cases to ensure the local server can handle multiple requests efficiently.
- *Page Load Time Testing:* Conduct speed tests to measure how quickly the web pages load under different conditions, ensuring an optimal user experience.
- *GPS & Selfie Verification Accuracy:* Validate the accuracy of GPS location logging and measure selfie verification processing times to ensure real-time attendance verification is reliable.

CHAPTER 4

DEPLOYMENT AND MAINTENANCE

Deployment Environment

The Event Attendance System is initially deployed on a local server, ensuring a controlled environment for testing and optimization before potential migration to cloud-based hosting. The system is designed to run on a Django backend with a PostgreSQL database, supporting smooth data transactions. To ensure compatibility, the system is tested on different web browsers, including Google Chrome, Mozilla Firefox, and Microsoft Edge.

Deployment Process

The deployment process follows a structured workflow to ensure reliability and efficiency. Initially, code and database migrations are performed to set up the system. Version control is managed using Git, allowing for seamless updates and tracking of modifications. The deployment involves configuring server settings, static files, and database connections to ensure optimal performance.

User Accessibility

The system is accessible through both mobile and desktop browsers, providing a user-friendly experience for event organizers and attendees. Authentication is managed through user login credentials, ensuring role-based access control for attendees, event organizers, and administrators. The system also supports QR code-based check-ins, allowing for quick and efficient attendance verification.

Data Integrity and Backup

Regular backups of attendance records, event details, and user profiles are performed to ensure data integrity. When deployed on Render hosting services, backups are managed using `pg_dump` for PostgreSQL, allowing efficient data dumps and restoration. If the system is running on a local server, the Django-DBBackup package is utilized to automate scheduled backups and store data securely. The backup process follows a scheduled routine, ensuring that all critical data is preserved in secured storage locations to prevent data loss in case of system failures. Database consistency checks are conducted to prevent anomalies and ensure accurate record-keeping.



System Scalability

The current deployment utilizes Render for online hosting, ensuring scalability and reliability as user demand increases. The system is optimized for handling a larger volume of users and events, leveraging cloud infrastructure for efficient performance. Additionally, indexing and caching techniques are considered to enhance database query efficiency and system responsiveness.

Bug Fixing and Updates

The system undergoes continuous monitoring to detect and resolve issues. An error logging mechanism is implemented to track system failures, and regular testing is conducted to identify and fix bugs. User feedback is collected to refine the user interface and functionality. The system updates follow a structured process to apply patches and introduce new features without disrupting ongoing operations.

Performance Monitoring

To maintain optimal performance, the system is monitored using performance tracking tools that assess server response times, database queries, and user load handling. Real-time logging and diagnostic tools help administrators track resource utilization and ensure the system runs efficiently. Stress testing is conducted periodically to verify that the system can handle an increasing number of users without performance degradation.

Future Enhancements

The system is designed with future expansion in mind. Possible enhancements include AI-based attendance validation, real-time event analytics, and mobile application integration to improve accessibility. Feedback from users will guide future updates, ensuring continuous improvement and relevance to event management needs.

USING THE WEB-BASED EVENT ATTENDANCE SYSTEM

The Event Attendance System aims to simplify how event participation is recorded, and the users can access this system from web browser in either desktop or mobile. When an attendee visits the website, he needs to log in with account details if already a user or sign up if it is his first time.

Dashboard Overview

The Dashboard Module provides administrators and moderators with an overview of system activity and real-time data tracking. Upon logging in, administrators have access to:

- Today's Logins, Attendees, Unverified Attendees – Displays a list of users who have logged in on the current day, the number of attendees checked in, and those whose attendance status is yet to be verified, providing insights into system activity.
- Attendance Overview – Shows attendance statistics for active events, including the number of attendees checked in.
- System Logs – Tracks all system activities such as user logins, attendance submissions, and event modifications.
- System Usage Metrics – Displays resource usage statistics to help monitor performance and detect any unusual system behavior.

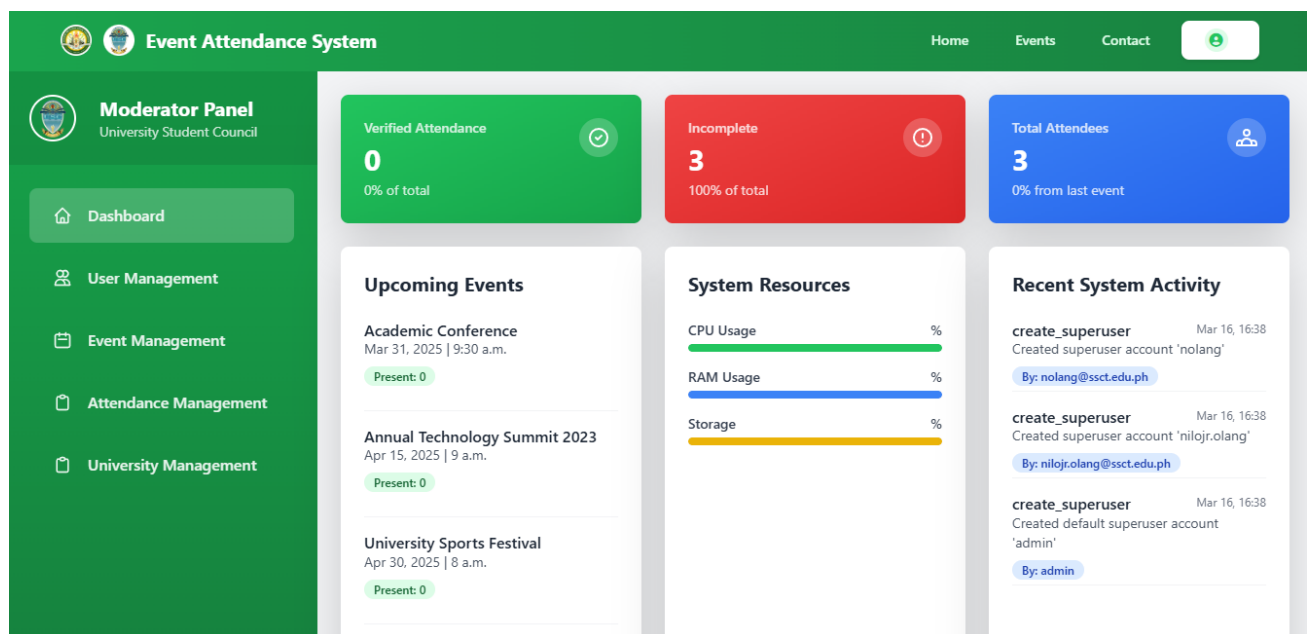
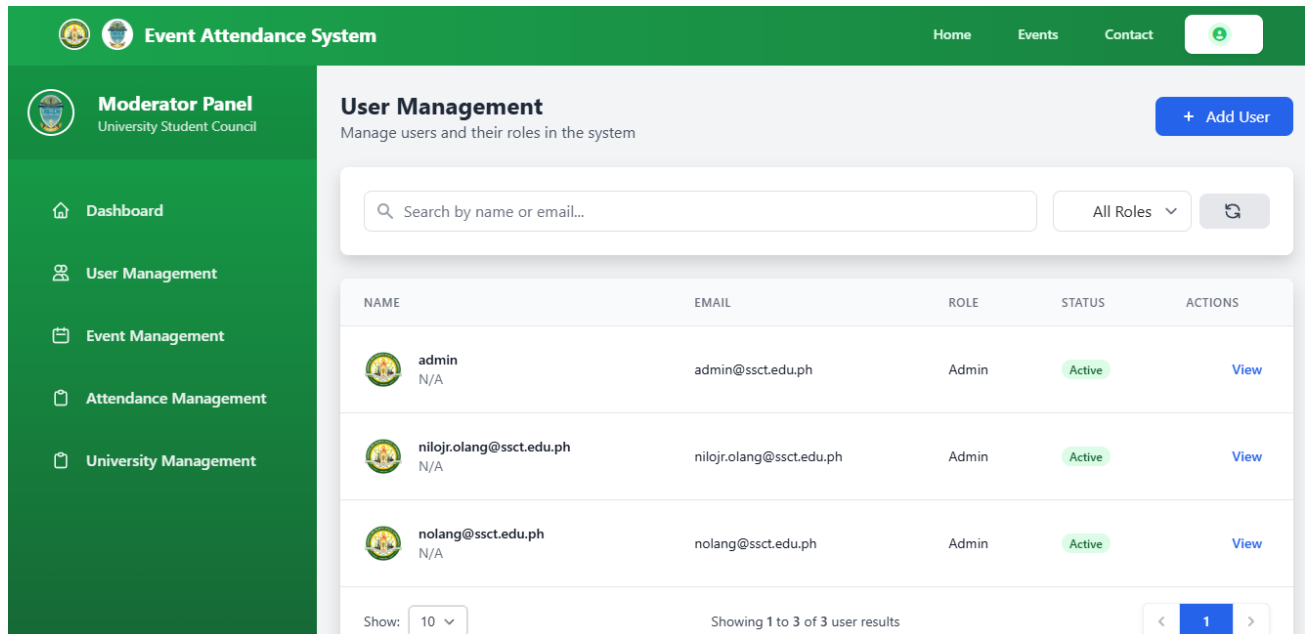


Figure 9. Dashboard overview

Managing Users

- Administrators can manage user accounts through the user management module.
- Users can be assigned different roles, such as attendees, event organizers, and administrators.
- The system allows administrators to edit, suspend, or remove user accounts as needed.

4. User authentication settings, including password resets and access control, are managed within this **module**.



NAME	EMAIL	ROLE	STATUS	ACTIONS
admin N/A	admin@ssct.edu.ph	Admin	Active	View
nilojr.olang@ssct.edu.ph N/A	nilojr.olang@ssct.edu.ph	Admin	Active	View
nolang@ssct.edu.ph N/A	nolang@ssct.edu.ph	Admin	Active	View

Figure 10. User management overview

Checking In to an Event

1. Once logged in, attendees can view a list of available events.
2. To check in, users can either scan a QR code provided by the event organizer or enter an event code manually.
3. The system verifies attendance using QR code scanning, GPS verification, or selfie authentication.
4. If successful, the user's attendance is recorded in real-time and reflected on the administrator's dashboard.

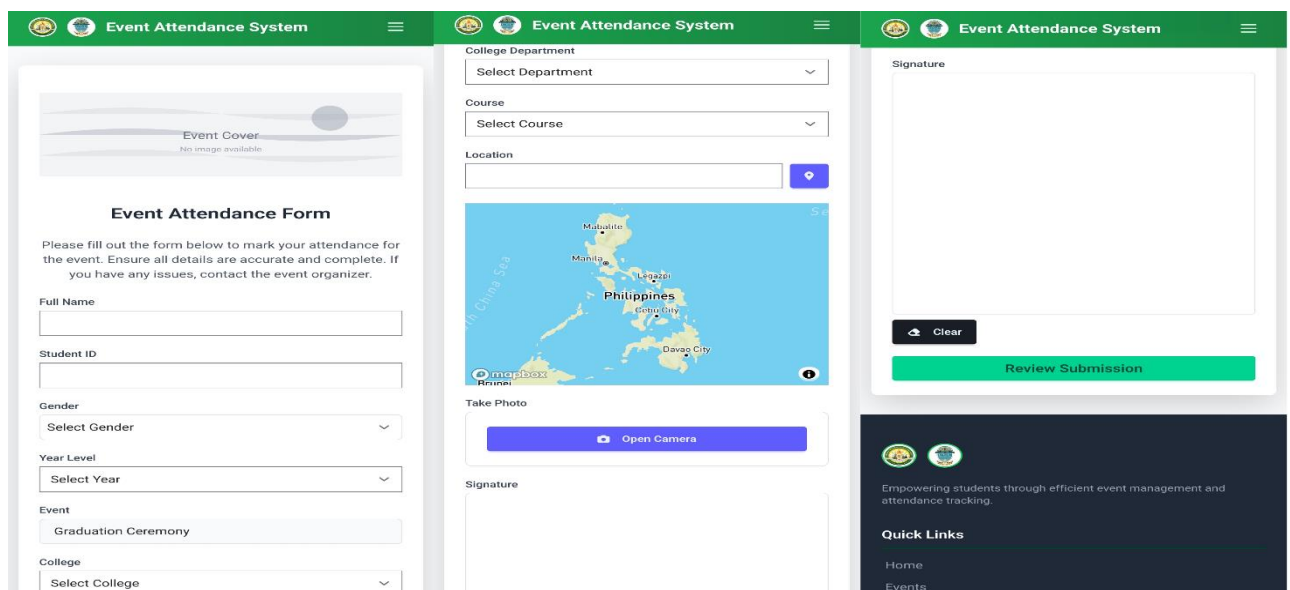
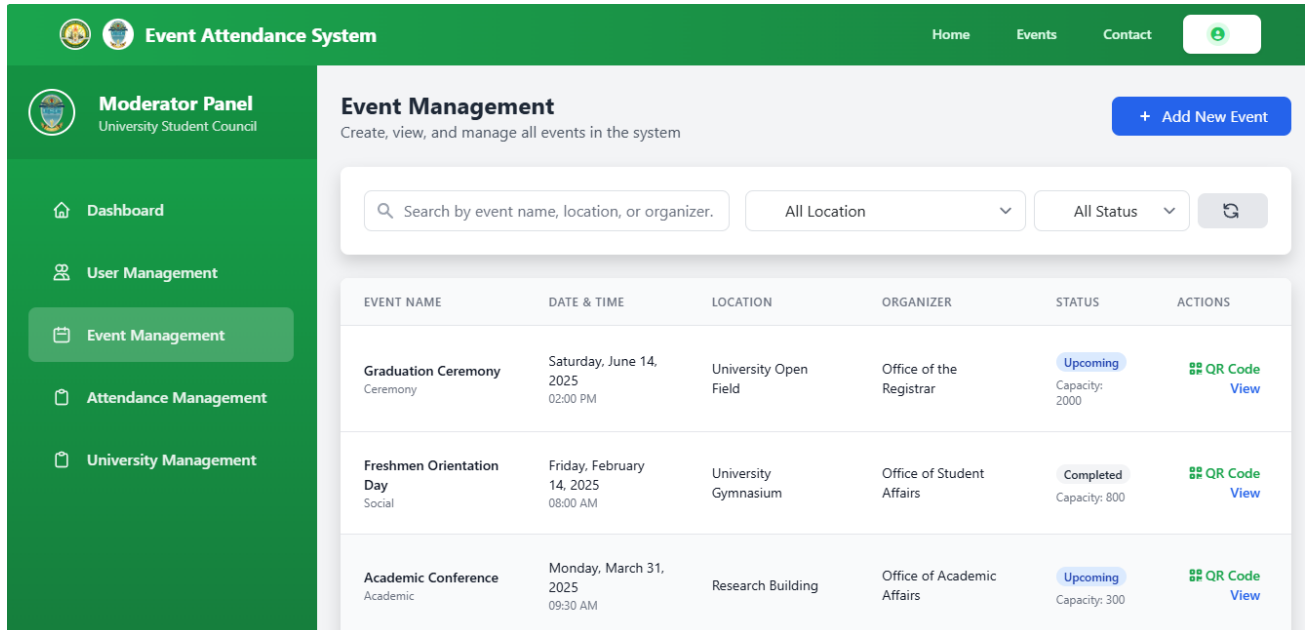


Figure 11. Event attendance Form overview

Managing Events

1. Organizers can create and manage events from the event management module.
2. They can specify event details, including location, date, and time.
3. The system generates a unique QR code for each event, which can be distributed to attendees.



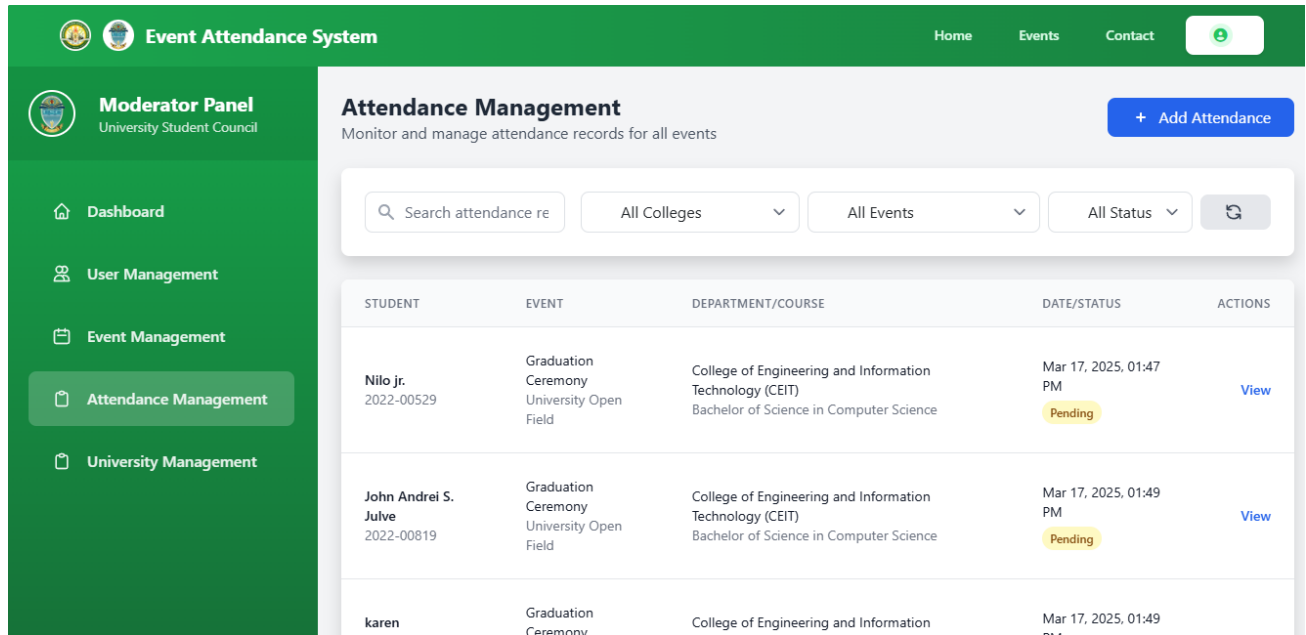
The screenshot shows the 'Event Management' overview page of the 'Event Attendance System'. The page has a green header with navigation links: Home, Events, Contact, and a user profile icon. A left sidebar for the 'Moderator Panel' (University Student Council) includes links to Dashboard, User Management, Event Management (highlighted), Attendance Management, and University Management. The main content area is titled 'Event Management' with a subtitle 'Create, view, and manage all events in the system' and an 'Add New Event' button. Below this is a search bar and filters for 'All Location' and 'All Status'. A table lists three events:

EVENT NAME	DATE & TIME	LOCATION	ORGANIZER	STATUS	ACTIONS
Graduation Ceremony Ceremony	Saturday, June 14, 2025 02:00 PM	University Open Field	Office of the Registrar	Upcoming Capacity: 2000	QR Code View
Freshmen Orientation Day Social	Friday, February 14, 2025 08:00 AM	University Gymnasium	Office of Student Affairs	Completed Capacity: 800	QR Code View
Academic Conference Academic	Monday, March 31, 2025 09:30 AM	Research Building	Office of Academic Affairs	Upcoming Capacity: 300	QR Code View

Figure 12. Event Management overview

Viewing Attendance Records

- a. Event organizers can log in and navigate to the attendance management module.
- b. Here, they can filter attendance records by event name, date, or attendee details.
- c. Attendance data can be exported in various formats for documentation purposes.



Event Attendance System Home Events Contact

Moderator Panel
University Student Council

- Dashboard
- User Management
- Event Management
- Attendance Management**
- University Management

Attendance Management

Monitor and manage attendance records for all events

+ Add Attendance

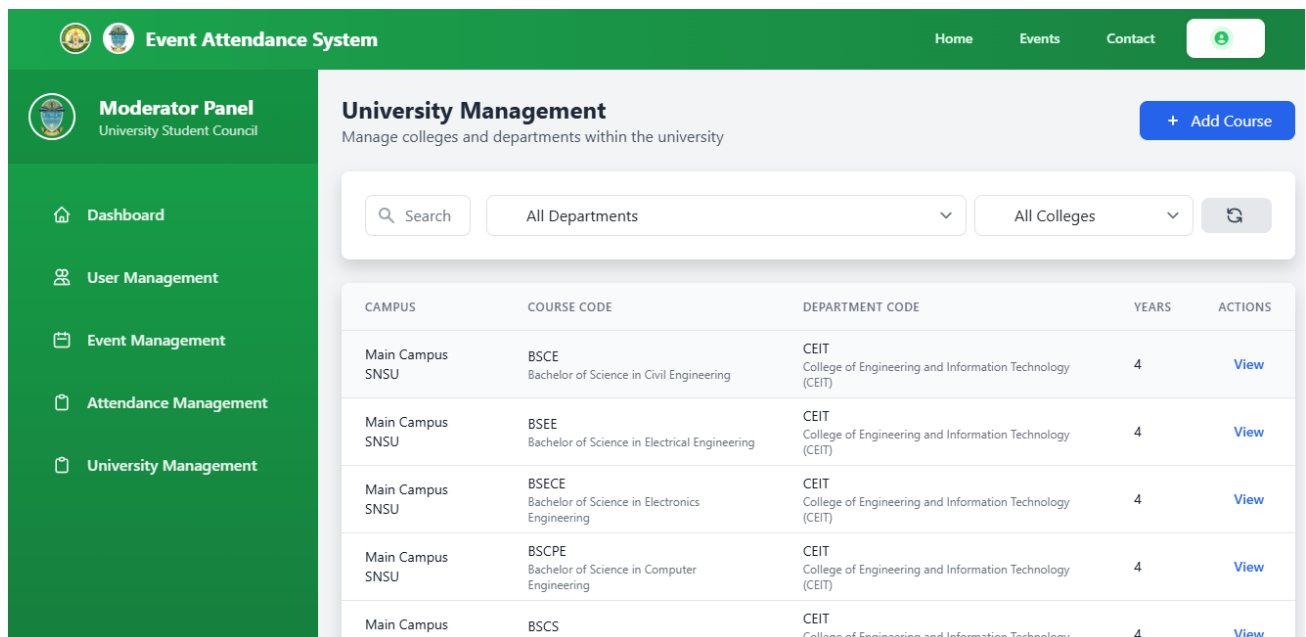
Search attendance re All Colleges All Events All Status

STUDENT	EVENT	DEPARTMENT/COURSE	DATE/STATUS	ACTIONS
Nilo Jr. 2022-00529	Graduation Ceremony University Open Field	College of Engineering and Information Technology (CEIT) Bachelor of Science in Computer Science	Mar 17, 2025, 01:47 PM Pending	View
John Andrei S. Julve 2022-00819	Graduation Ceremony University Open Field	College of Engineering and Information Technology (CEIT) Bachelor of Science in Computer Science	Mar 17, 2025, 01:49 PM Pending	View
karen	Graduation Ceremony	College of Engineering and Information Technology (CEIT)	Mar 17, 2025, 01:49 PM	View

Figure 13. Attendance Management overview

Managing Universities

- The system supports multiple universities by allowing administrators to register and manage them.
- Each university has its own set of events, users, and attendance records.
- Administrators can update university details, associate users with institutions, and oversee event participation.
- Universities can generate reports specific to their events and attendance data.



Event Attendance System Home Events Contact

Moderator Panel
University Student Council

- Dashboard
- User Management
- Event Management
- Attendance Management
- University Management**

University Management

Manage colleges and departments within the university

+ Add Course

Search All Departments All Colleges

CAMPUS	COURSE CODE	DEPARTMENT CODE	YEARS	ACTIONS
Main Campus SNSU	BSCE Bachelor of Science in Civil Engineering	CEIT College of Engineering and Information Technology (CEIT)	4	View
Main Campus SNSU	BSEE Bachelor of Science in Electrical Engineering	CEIT College of Engineering and Information Technology (CEIT)	4	View
Main Campus SNSU	BSECE Bachelor of Science in Electronics Engineering	CEIT College of Engineering and Information Technology (CEIT)	4	View
Main Campus SNSU	BSCPE Bachelor of Science in Computer Engineering	CEIT College of Engineering and Information Technology (CEIT)	4	View
Main Campus SNSU	BSCS Bachelor of Science in Computer Science	CEIT College of Engineering and Information Technology (CEIT)	4	View

Figure 14. University Management overview



CONCLUSION

The Mobile-Based School Event Attendance Monitoring System improves the efficiency, security and accuracy of attendance tracking in school events. The system is capable of integrating QR code scanning, GPS based location tracking and selfie verification to prevent fraudulence in attendance and ease the verification process. The use of cloud based storage enables easy access of the attendance records in real time, which in turn lightens the administrative burden and enhance on data management. Furthermore, the user feedback and testing of the system indicate that it is easy to use, reliable and applicable to a variety of school events. However, there are some possible drawbacks which might be the limitations of the system, for example dependence on the Internet and accuracy of GPS which can be problematic in areas with poor network coverage.

Furthermore, the system's user-centered design ensures accessibility for both students and administrators, making attendance tracking a seamless process. The integration of data encryption and authentication protocols enhances security, protecting sensitive user information from unauthorized access. Through continuous maintenance and user support, the system remains reliable, with regular updates improving performance, usability, and scalability. Its ability to generate real-time attendance reports also aids school administrators in making data-driven decisions to enhance event management strategies. Moving forward, future developments could focus on biometric authentication, machine learning for anomaly detection, and expanded compatibility with various mobile operating systems. Additionally, incorporating AI-based analytics could provide deeper insights into attendance trends and student engagement patterns. By continuously refining its features and expanding its capabilities, the Mobile-Based School Event Attendance Monitoring System has the potential to become a standardized solution for educational institutions, ensuring an accurate, fraud-resistant, and efficient attendance tracking process for years to come.



REFERENCES:

- [1] Chiang, T.-W., Yang, C.-Y., Chiou, G.-J., Lin, F. Y.-S., Lin, Y.-N., Shen, V. R. L., Juang, T.-Y., & Lin, C.-Y. (2022). Development and Evaluation of an Attendance Tracking System Using Smartphones with GPS and NFC. *Applied Artificial Intelligence*, 36(1). DOI: <https://doi.org/10.1080/08839514.2022.2083796>
- [2] Saracostti, M., De Toro, X., Rossi, A., Lara, L. G., & Sotomayor, M. B. (2022). Implementation of a web-based system to measure, monitor, and promote school engagement strategies. A Chilean experience. *Frontiers in Psychology*, 13. DOI: <https://doi.org/10.3389/fpsyg.2022.980902>
- [3] Tsai, M.-F. and Li, M.-H. 2022. Intelligent attendance monitoring system with spatio-temporal human action recognition. *Soft Computing*. 27, 8 (Oct. 2022), 5003–5019. DOI: <https://doi.org/10.1007/s00500-022-07582-y>
- [4] Farag, W. and Abouelela, M.M.M. 2022. Low-Cost Active Monitoring of Attendance using Passive RFID Technology. *JITEKI : Jurnal Ilmiah Teknik Elektro Komputer dan Informatika*. 8, 4 (Dec. 2022), 552. DOI: <https://doi.org/10.26555/jiteki.v8i4.25168>
- [5] Mulay, S.S. 2022. Student attendance application. *Indian Scientific Journal Of Research In Engineering And Management*. 06, 11 (Nov. 2022). DOI: <https://doi.org/10.55041/ijrsrem16984>
- [6] Caytairo-Silva, N. et al. 2024. Optimizing Attendance Management in Educational Institutions Through Mobile Technologies: A Machine Learning and Cloud Computing Approach. *International journal of interactive mobile technologies*. 18, 12 (Jun. 2024), 112–128. DOI: <https://doi.org/10.3991/ijim.v18i12.46917>
- [7] Satria, D., Hidayat, T., Hidayat, M.A. and Zakaria, Z. 2018. Application of SMS Gateway on Attendance Detection Systems using RFID. 1, 2 (Oct. 2018). DOI: <https://doi.org/10.32672/JNKTI.V1I2.778>
- [8] Mrabet, H. and Ait Moussa, A. 2020. IoT-School Attendance System Using RFID Technology. *International Journal of Interactive Mobile Technologies (ijim)*. 14, 14 (Aug. 2020), 95–108. DOI: <https://doi.org/10.3991/IJIM.V14I14.14625>
- [9] Syahrul, M. and Wicaksono, F. 2020. Design and Application of a Portable Fingerprint System for Student Attendance Web-Based and Telegram Using



Raspberry Pi. 879, 1 (Jul. 2020), 012101. DOI: <https://doi.org/10.1088/1757-899X/879/1/012101>

[10] B, A., C, A., K. N, C. and R, R. 2018. Attendance Monitoring System Using Face Recognition. International journal of engineering research and technology. 6, 13 (Apr. 2018).

[11] Adderley, R., Sanim Mohd Fauzi, S., Ayub, I.M., Fikri Jamaluddin, M.N., Puspitasari, I. and Okfalisa, O. 2020. Design and development of activity attendance monitoring system based on RFID. Indonesian Journal of Electrical Engineering and Computer Science. 17, 1 (Jan. 2020), 500–507. DOI: <https://doi.org/10.11591/IJEECS.V17.I1.PP500-507>

[12] Febrifyaning, R., Hamidah, M.N., Arizal, A. and Setyatama, F. 2021. Smart school management system of presence monitoring teacher as a realtime with rad (rapid application development) method at khadijah high school surabaya. 6, 1 (Jun. 2021).

[13] Vinay, S.V., Kumar, V.S., Shukla, S., Kumar, R. and Devanagavi, G.D. 2020. Attendance monitoring Using IOT for transportation Systems. (Mar. 2020).

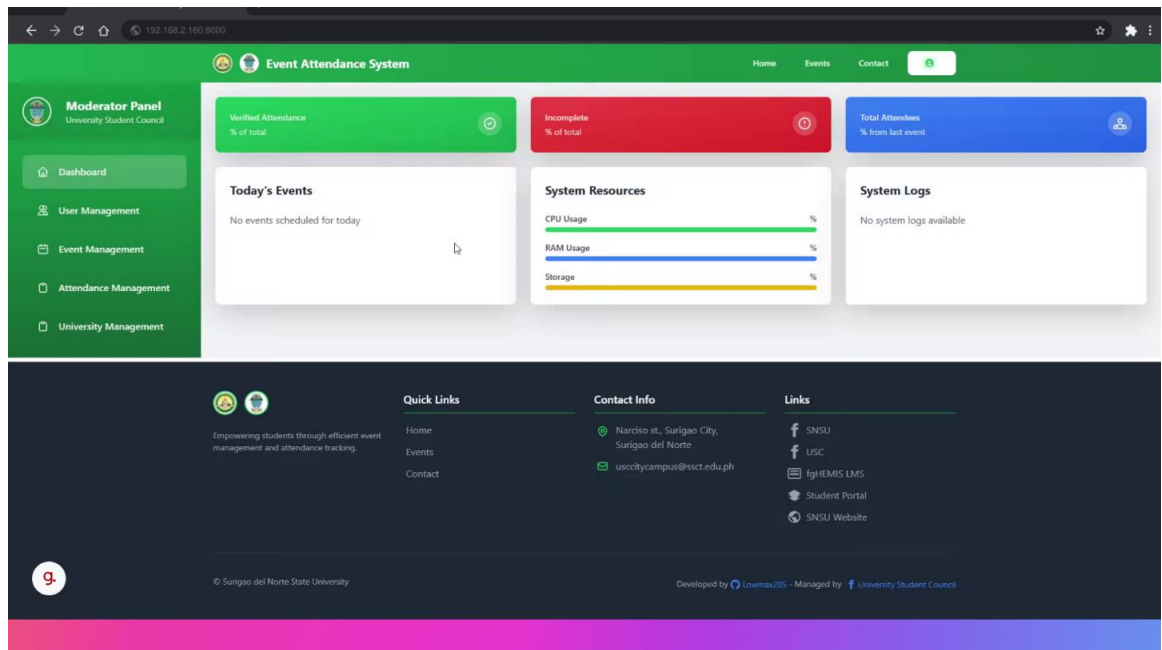
USER MANUAL

How to add user in admin

This guide will walk you through the process of adding a user in the admin panel

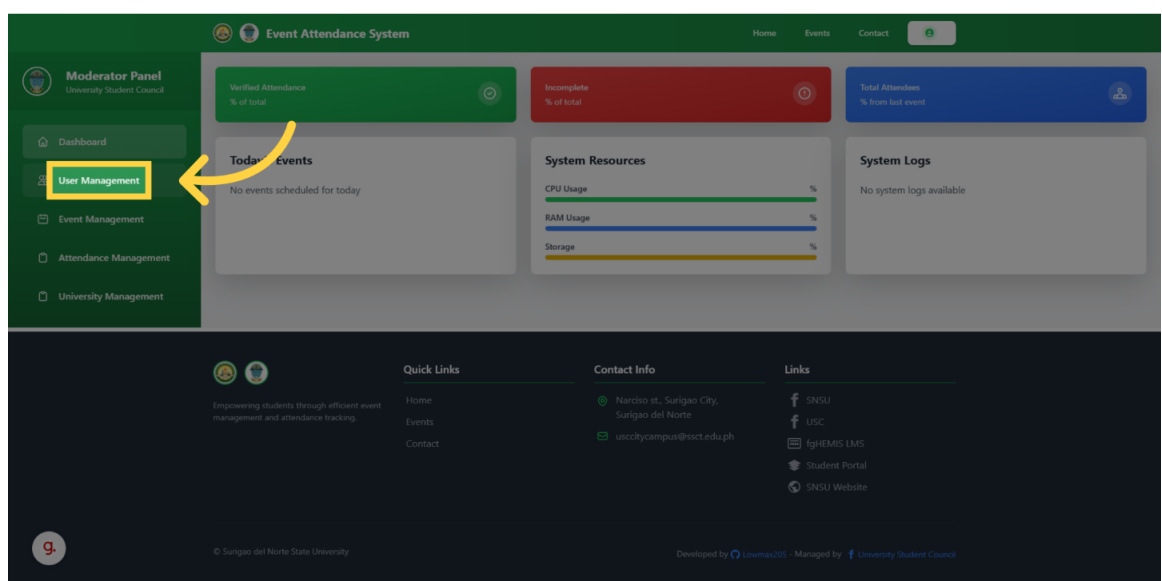
1. Introduction

By following these instructions, you will be able to efficiently create new user accounts within the system.



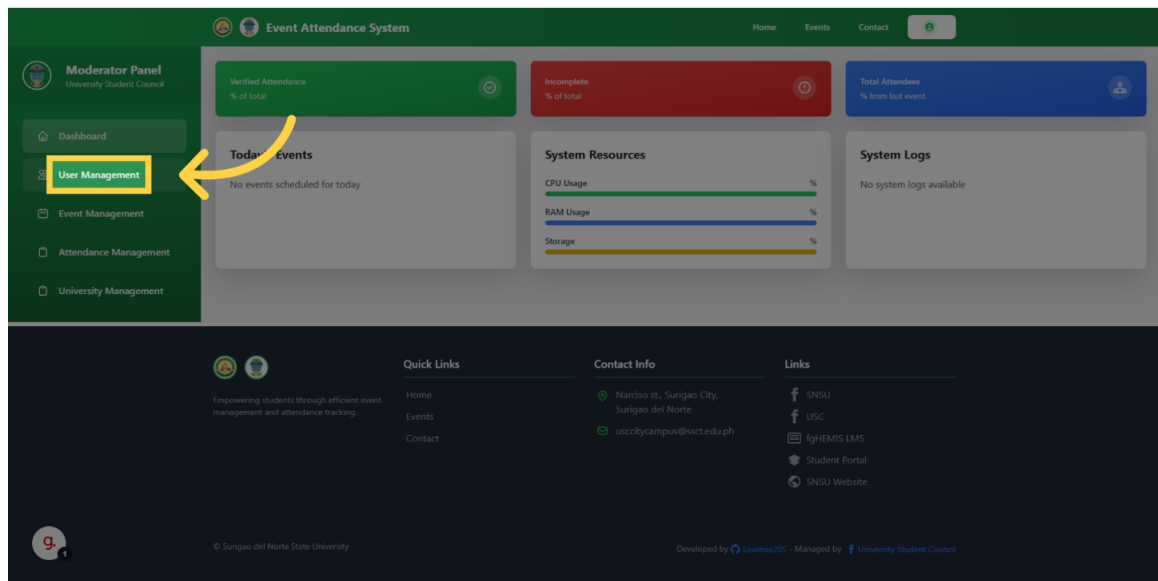
2. Click "User Management"

Access the User Management section.



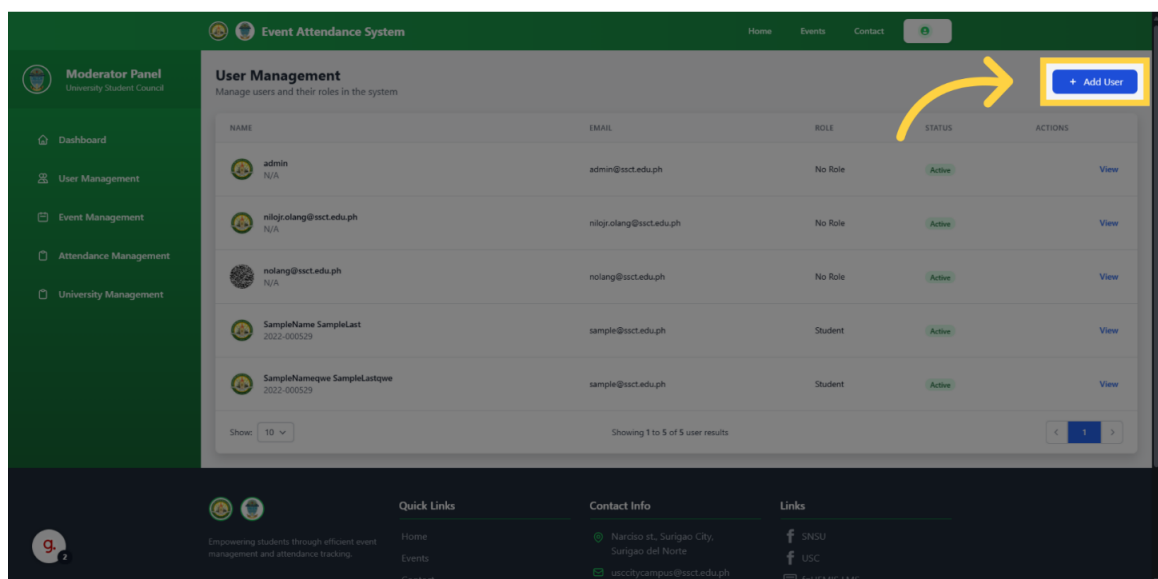
3. Click "User Management"

Navigate to the User Management area.



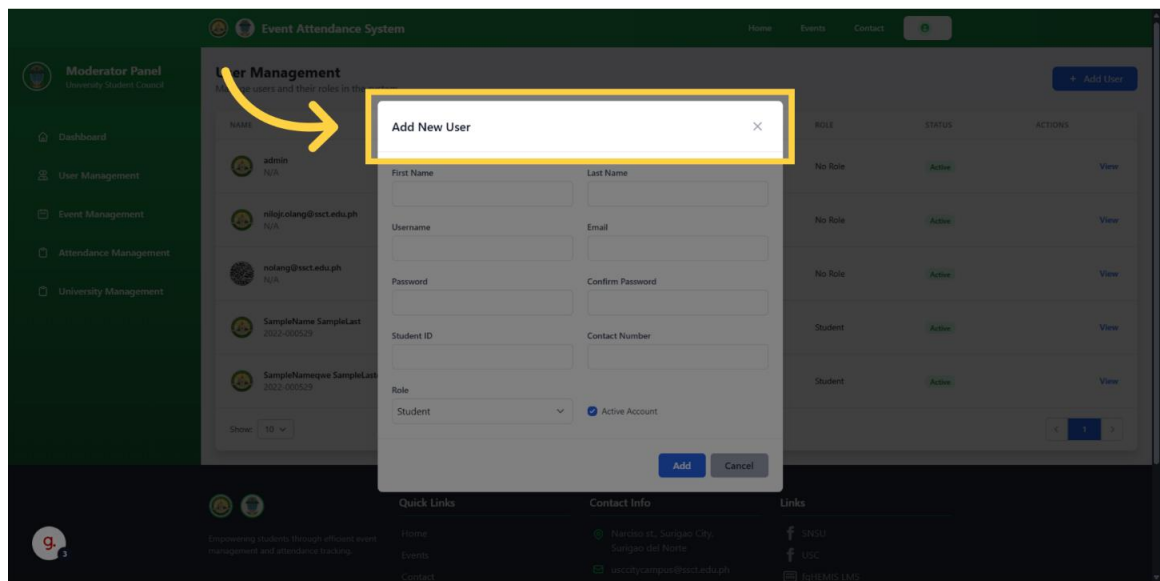
4. Click "Add User"

Select the option to add a new user.



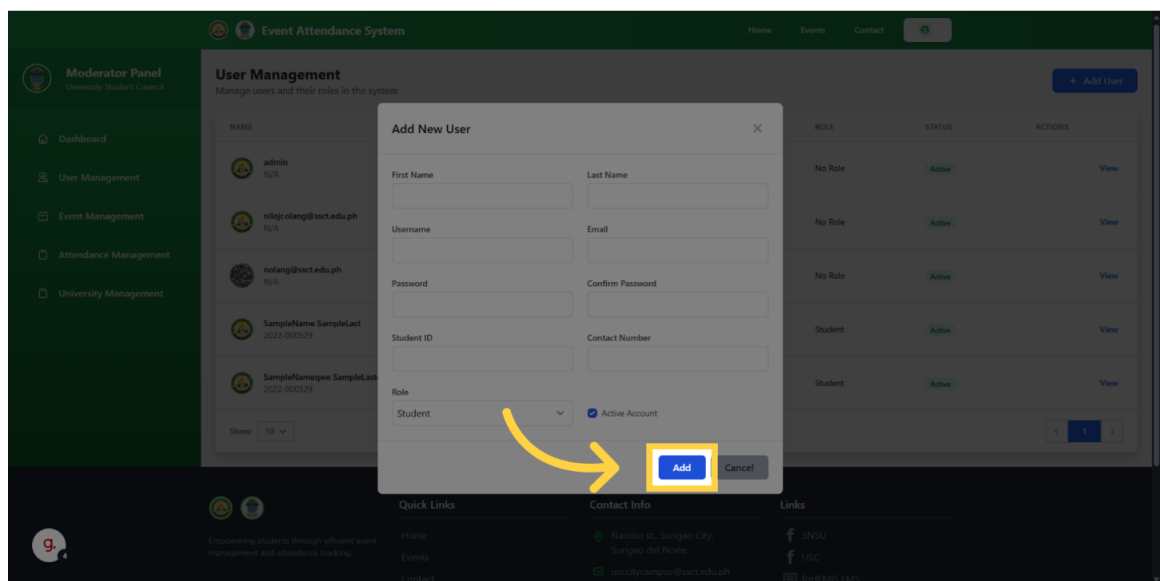
5. Add New User

Add a new user profile and fill-up the form below.



6. Click "Add"

Save the user information.



7. User created successfully"

Acknowledge the successful creation of the user.



The screenshot displays the 'Event Attendance System' interface. On the left is a 'Moderator Panel' sidebar with options: Dashboard, User Management, Event Management, Attendance Management, and University Management. The main area is titled 'User Management' with the subtitle 'Manage users and their roles in the system'. A table lists users with columns for NAME, EMAIL, ROLE, STATUS, and ACTIONS. A yellow arrow points from the 'Add User' button to a notification box in the top right corner that says 'User created successfully' with a green checkmark icon. The table contains six rows of user data, including 'admin', 'nilojolrang@sct.edu.ph', 'nolang@sct.edu.ph', and three sample users. At the bottom, there are pagination controls showing 'Showing 1 to 6 of 6 user results'.

NAME	EMAIL	ROLE	STATUS	ACTIONS
admin N/A	admin@sct.edu.ph	No Role	Active	View
nilojolrang@sct.edu.ph N/A	nilojolrang@sct.edu.ph	No Role	Active	View
nolang@sct.edu.ph N/A	nolang@sct.edu.ph	No Role	Active	View
SampleName SampleLast 2022-000529	sample@sct.edu.ph	Student	Active	View
SampleNameqwe SampleLastqwe 2022-000529	sample@sct.edu.ph	Student	Active	View
SampleNameqwe23 SampleLastqwe23 2022-000529	sample@sct.edu.ph23	Student	Active	View

This guide covered the steps to add a user in the admin panel. From accessing the User Management section to confirming the successful creation of a new user, each step is outlined the process of adding users effectively. This manual guide is applicable to other module like Event, Attendance, and University Management following with the form of each management.