

CampusConnect-Design and Development of an Automated Attendance Management System

Rajaram Deshpande

Department of Artificial Intelligence and Machine Learning,
YSPM's Yashoda Technical Campus, Satara,
India

Samiksha Nikam

Department of Computer Application,
YSPM's Yashoda Technical Campus, Satara,
India

ABSTRACT

The CampusConnect application is designed as a Smart Attendance System modeled on the concept of Management Information Systems (MIS) widely adopted in organizational environments. It utilizes technologies such as Android Studio, Flutter, Dart, Firebase RTDB, Google forms and Google sheets to streamline daily attendance processes. The primary objective behind developing this system is to eliminate proxy attendance, reduce repetitive manual tasks, and minimize the dependency on paper-based records. The application creates a structured environment where attendance data is collected digitally and stored securely. Along with attendance tracking, the system supports maintaining student progress records and generating monthly analytical reports that can assist faculty and administrators in decision-making. By integrating data handling and reporting into one centralized platform, Campus Connect helps improve accuracy, transparency, and time efficiency in academic management, making it suitable for implementation in colleges and universities. The primary objective behind developing this system is to eliminate proxy attendance and reduce manual tasks [1], [9].

Keywords

Academic Record Management, Digital Attendance, Firebase RTDB, Proxy Prevention, QR Code Attendance, Smart Attendance System.

1. INTRODUCTION

The campus Connect application is developed using multiple technologies provided by Google, including Android Studio, Flutter, Dart, Firebase Real-Time Database (RTDB), Google Forms, and Google Sheets. The system is structured as a mobile-based platform consisting of three main functional modules: Admin module, Faculty module, and Student module. Each module performs a distinct set of operations designed to support automated attendance handling and ensure quick access to academic data. Several institutions have adopted mobile and cloud-supported attendance solutions to improve accuracy and transparency [5], [7]. QR-code-based mechanisms are widely used to reduce impersonation [9].

The Admin Module is responsible for managing the core data of the system. It allows administrative users to add, update, and maintain records related to students, faculty members, and subjects. This module also controls the operational settings of the application, ensuring that all modules work in coordination. Through Firebase RTDB, the admin can securely store and Retrieve data in real time, maintaining continuity and consistency across the system.

The Faculty Module is designed to simplify the process of marking and verifying attendance. Faculty members can generate session-specific QR codes and share them with

students during class. This QR code remains valid for only 90 seconds, which significantly helps prevent proxy attendance [1], [9]. When students submit attendance responses through Google Forms, the records are automatically stored in Google Sheets. This digital approach reduces manual entry, minimizes errors, and provides faculty members with immediate access to consolidated attendance data for review or reporting.

The Student Module focuses on attendance submission and personal academic visibility. Students scan the QR code shared by faculty to access the attendance form and submit their details. Additionally, the module allows students to view their profile information and attendance status, ensuring transparency and encouraging accountability for class participation. Overall, CampusConnect provides an integrated system that replaces manual attendance registers with an efficient digital solution, reduces paperwork, and improves accuracy and reliability in attendance management across educational institutions.

2. SYSTEM WORKFLOW

The Campus Connect application is primarily focused on implementing a smart attendance system that simplifies data collection and handling. The current system supports daily attendance marking, record management, and secure data storage. It is designed in a modular structure and can later be expanded into a complete campus management ecosystem. The application consists of three main modules: Admin, Faculty, and Student. The overall system flow is illustrated in Figure 1.1.

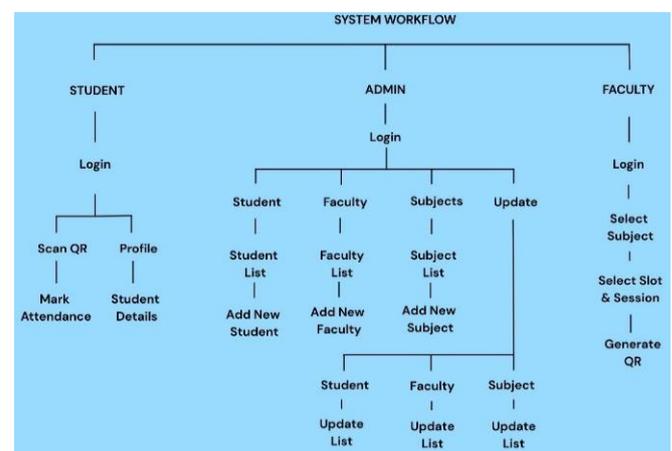


Fig.1.1: System Flow Diagram

2.1 MODULE DESCRIPTION

Admin: The Admin module provides full control over the system. It allows administrative users to add, update, or delete

information related to students, faculty members, and subjects. All major data handling tasks are carried out through this module. The administrator may be a Head of Department, Principal, or any institutional authority responsible for academic administration. Through this module, the admin ensures that all operational data remains accurate and up to date. Admin module screen shots are shown in fig 1.2.

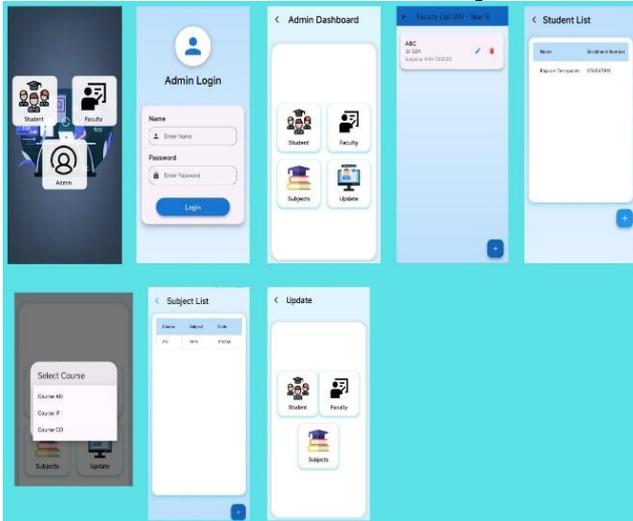


Fig1.2 Admin Module

Faculty: The Faculty module is intended for teaching staff to carry out attendance marking. After logging in, the faculty selects the assigned subject and chooses the session type, which may be theory or practical. The system is structured to display time slots based on the selected session. The faculty then generates a QR code linked to the attendance form. This QR code remains valid for only 90 seconds, which significantly helps prevent proxy attendance. Attendance submissions are automatically recorded and stored for verification. The faculty module screenshot is shown in fig 1.3

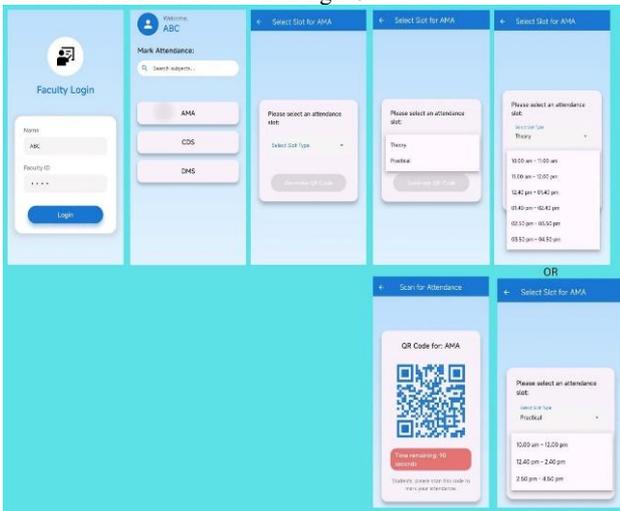


Fig.1.3: Faculty Module

Student: The Student module enables students to mark their attendance and view personal profile details. On the home screen, students can either Scan QR or View Profile. The profile section displays basic information such as name and enrolment number. On selecting the scan option, the device camera opens to read the QR code generated by the faculty. Once the QR is scanned, the system redirects the student to the attendance form. After submitting the form, the attendance

marking process is completed. The fig 1.4 illustrate screen shot for student module.

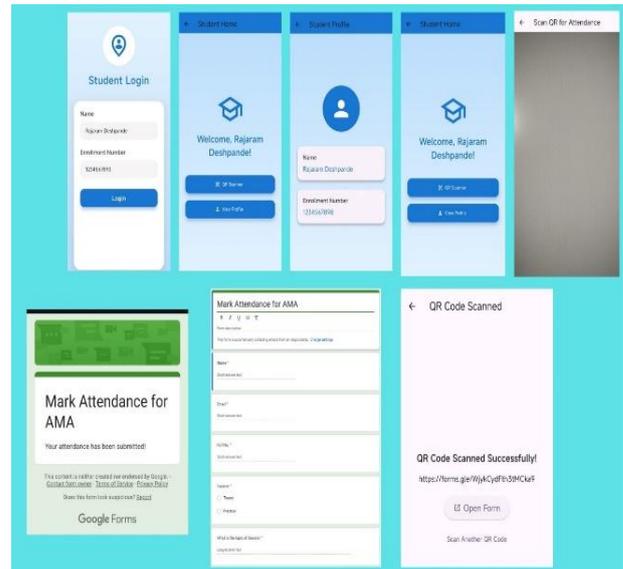


Fig.1.4: Student Module

2.2 DATA STORAGE & DATABASE

For storing attendance data submitted by students, the system makes use of Google Forms integrated directly with Google Sheets. When a student submits the attendance form, the response is automatically recorded in a linked Google Sheet. Each entry includes a timestamp, date, and the student information entered during submission. This automated process removes the need for manual entry, reduces errors, and ensures that attendance records remain accurately organized and easily accessible for review or analysis.

For storing institutional data such as student details, faculty records, and subject information, the system uses Google Firebase Real-Time Database (RTDB). The RTDB is connected to the application through Android Studio, allowing dynamic storage and retrieval of data. The database operates in real time, meaning any update performed by the administrator is immediately reflected across the system for all authorized users. This is particularly useful for functions such as login verification, user authentication, and applying validation logic within the application developed using Dart.

Firestore initially provides free storage suitable for development and educational project use. However, when the system is deployed on a larger scale—such as at the institutional or university level—additional storage and higher data handling capacity may require shifting to a paid plan. Despite this, Firestore remains a suitable choice due to its secure data structure, ease of integration, and strong data protection features.

By using Google Forms, Google Sheets, and Firestore RTDB together, the system ensures reliable data collection, organized storage, and efficient access control. This combination supports transparency, accuracy, and security in academic record management. Cloud databases such as Firestore enable real-time synchronization and secure access to academic records [5], [8].

2.3 TECHNOLOGY STACK

The CampusConnect application has been developed using several technologies provided by Google. The selection of

these tools was based on their compatibility, ease of integration, and suitability for mobile application development.

Code Editor:

The development environment used for this project is Android Studio. It offers a comprehensive interface for building, debugging, and testing applications. Android Studio also provides built-in support for connecting with various databases such as Firebase, Cloudinary, MySQL, MongoDB, and PostgreSQL, which makes it versatile and efficient for application development

Frontend Development:

The user interface and overall layout of the application have been developed using the Flutter framework. Flutter is well-known for its ability to create smooth, responsive, and visually consistent mobile applications. The programming language used with Flutter is Dart, which allows access to a wide range of UI components and design features. Dart provides a secure and structured environment, making it suitable for designing modern applications.

Backend Development:

For backend functionality, Dart is also used to handle communication between the application and Firebase Real-Time Database (RTDB). This integration allows the application to store and retrieve data instantly, ensuring that updates are reflected across the system in real time. The use of Firebase RTDB provides reliability, scalability, and secure data handling within the application [5], [8].

2.4 RESULT AND PERFORMANCE ANALYSIS

The fig 1.5 represents the output generated by the system. Students submit their attendance through a Google Form, and the responses are automatically recorded in Google Sheets, as the form is directly linked to the sheet [7].

To validate the performance of the proposed CampusConnect system, real-time testing was carried out with 10 students across 4 subjects over a period of 3 days. Each day consisted of 6 lecture sessions, resulting in a total of 16 attendance events. The system therefore processed 180 possible attendance submissions.

During each lecture, the faculty generated a QR code valid for 90 seconds. Students scanned the code and submitted their attendance using the mobile interface. The responses were automatically stored in Google Sheets, while user authentication and master data were handled through Firebase RTDB.

The manual roll-call method typically required between 5–10 minutes per lecture depending on class interaction and verification. Using CampusConnect, attendance submission was generally completed within about 1 minute. This indicates an approximate 75–80% reduction in time, enabling instructors to dedicate more time to teaching activities.

Timestamp	Name	Email	Session	What is the topic of Session	Comments	Email address	Roll No.
05/02/2026 13:06:00	Shikhar Sarika Som	sanikaashikha19@gmail.com	Theory	Evaluation of operating system		sanikaashikha19@gmail.com	55
05/02/2026 13:06:29	Sanchita jivan zende	sanchitazende19@gmail.com	Theory	Evolution of operating system		sanchitazende19@gmail.com	45
06/02/2026 17:25:01	Soham Tanaji Salunkhe	sohamtanajisalunkhe45@gmail.com	Theory	OOPS using Java		sohamtanajisalunkhe45@gmail.com	69
06/02/2026 17:45:26	Sanchita jivan zende	sanchitazende@gmail.com	Theory	Generation operating system		sanchitazende19@gmail.com	65
06/02/2026 18:00:36	Sarika Somrath Shik	sanikaashikha19@gmail.com	Theory	Applications of java		sanikaashikha19@gmail.com	57
06/02/2026 18:01:11	Siddhi sanjay mandav	siddhi527@gmail.com	Theory	Applications of java		siddhi527@gmail.com	28
06/02/2026 18:04:00	Sarika Somrath Shik	sanikaashikha19@gmail.com	Theory	Application of java		sanikaashikha19@gmail.com	57
06/02/2026 18:18:19	Sung Shrinant Mohin	mohitesung75@gmail.com	Theory	OOP using Java		mohitesung75@gmail.com	30
06/02/2026 14:18:25	Sarika Somrath Shik	sanikaashikha19@gmail.com	Theory	Features of Java		sanikaashikha19@gmail.com	57
06/02/2026 17:14:29	Sung Shrinant Mohin	mohitesung75@gmail.com	Theory	OOPs using Java		mohitesung75@gmail.com	30
06/02/2026 17:22:45	Soham Tanaji Salunkhe	sohamtanajisalunkhe45@gmail.com	Theory	Oops using java		sohamtanajisalunkhe45@gmail.com	49
06/02/2026 18:47:20	Siddhi sanjay mandav	siddhi527@gmail.com	Theory	Features of algorithm		siddhi527@gmail.com	28
10/02/2026 14:48:25	Sung Shrinant Mohin	mohitesung75@gmail.com	Theory	OOPs using Java		mohitesung75@gmail.com	30
10/02/2026 15:00:01	Sarika Somrath Shik	sanikaashikha19@gmail.com	Theory	Features of Java		sanikaashikha19@gmail.com	57

Fig.1.5: Attendance Sheet

3. REFERENCES

- [1] Munivrana, L., & Pleština, V. (2025). Smarttendance: A BLE-based mobile application for real-time student attendance tracking. In Proceedings of the International Conference on Smart and Connected Systems. IEEE. <https://ieeexplore.ieee.org/abstract/document/11091720>
- [2] Patel, P. (2025). An approach to location-based QR attendance system and mobile application integration for lecture and laboratory oversight: Smart QR attendance application with Flutter and Firebase. In International Conference on Data Management, Analytics & Innovation (pp. 245–255). Springer. https://link.springer.com/chapter/10.1007/978-981-96-6537-2_18
- [3] Alawathugoda, K., & Razeeth, M. S. S. (2024). Smart student record book for university education on the Sri Lanka education system. In International Conference on Advances in Computing and Technology. IEEE. <https://ieeexplore.ieee.org/abstract/document/11025289>
- [4] Darshana Band, Garima Agrawal, Harshada Deo, Vaishnavi Dhengekar, Rushikesh Dadanje Prof. Yugandhara Thakare, (2024), International Research Journal of Engineering and Technology (IRJET), 11(4).
- [5] Kumar, A., & Raj, S. (2023). Design of a mobile attendance system using Firebase Cloud and Flutter framework. International Journal of Computer Applications, 184(22), 25–30. <https://doi.org/10.5120/ijca2023920113>
- [6] Ahmed, T., & Rahman, M. (2022). IoT-enabled smart attendance tracking using Flutter and Google Firebase. Journal of Emerging Technologies in Learning, 17(4), 112–122. <https://doi.org/10.3991/ijet.v17i04.28745>
- [7] Singh, R., & Gupta, N. (2022). Enhancing academic management through smart mobile attendance systems using cloud databases. Procedia Computer Science, 200, 543–550. <https://doi.org/10.1016/j.procs.2022.01.221>
- [8] Bhattacharya, S., & Das, P. (2021). Firebase and Flutter-based student monitoring system for higher education institutions. In Smart Innovations in Computing and Communications (pp. 379–389). Springer. https://doi.org/10.1007/978-981-33-6996-6_35
- [9] Jain, A., & Mehta, V. (2020). Smart attendance system

using QR code and cloud integration. *International Journal of Advanced Research in Computer Science*, 11(6), 70–75. <https://doi.org/10.26483/ijarcs.v11i6.6623>

[10] Mr.g. karunakar, ch. Asha sai, k.v.s.m chandana sree, k.

Anil kumar,(2020). Smart attendance monitoring system (sams): a face recognition-based attendance system for classroom environment. ISSN 2581–4575 Page 195. https://ijrdst.org/public/uploads/paper/458_approvedpaper.pdf