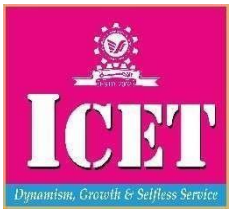
COLLEGE OF ENGINEERING AND TECHNOLOGY

Mulavoor P.O, Muvattupuzha, Kerala – 686673



BLUETOOTH AUTOMATED ATTENDANCE MARKING SYSTEM

PROJECT REPORT

JAMESY JOSEPH

(ICE21CS039)

PARVATHY VASUDEVAN

(ICE21CS051)

RAHMATH BEEVI ASHARAF

(ICE21CS053)

in partial fulfilment for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE AND ENGINEERING
APJ ABDUL KALAM TECHNOLOGICAL UNIVERSITY

2024

COLLEGE OF ENGINEERING AND TECHNOLOGY

Mulavoor P.O, Muvattupuzha, Kerala – 686673

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that the project report entitled **"BLUETOOTH AUTOMATED ATTENDANCE MARKING SYSTEM**" is a bonafide record of the project report presented by JAMESY JOSEPH (ICE21CS039), PARVATHY VASUDEVAN (ICE21CS051), RAHMATH BEEVI ASHARAF (ICE21CS053) during the academic year 2023 - 2024 towards the partial fulfilment of the requirement of the award of B.Tech Degree in Computer Science and Engineering of APJ Abdul Kalam Technological University, Thiruvananthapuram.

Ms. SHABNA C H	**Ms. HONEY JOHN & Ms. AZIYA SHIRIN V.S**	**Dr. LINO ABRAHAM VARGHESE**
Assistant Professors CSE Department	Assistant Professors CSE Department	Professor & HOD CSE Department
Project Guide	**Project Coordinators**	**Head of the Department**

DECLARATION

We Jamesy Joseph, Parvathy Vasudevan, Rahmath Beevi Asharaf hereby declare that this project report entitled “Bluetooth Automated Attendance Marking System” is the Bonafide work of mine carried out under the supervision of Ms. Honey John and Ms. Aziya Shirin V.S. We declare that to the best of our knowledge, the work report here in does not form part of any project report or dissertation the basis of which a degree or award was conferred on an earlier occasion on any other candidate the content of this report is not being presented by any other student to this or any other university for the award of degree.

Signature:
Name of the Student: JAMESY JOSEPH
Uni.Register No: ICE21CS039

Signature:
Name of the Student: PARVATHY VASUDEVAN
Uni.Register No: ICE21CS051

Signature:
Name of the Student: RAHMATH BEEVI ASHARAF
Uni.Register No: ICE21CS053

Signature:
Name of the Guide: Ms. SHABNA C H

Signature:
Name of the Coordinator: Ms.HONEY JOHN

Signature:
Name of the Coordinator: Ms.AZIYA SHIRIN V.S

Countersigned with Name: Dr. LINO ABRAHAM VARGHESE
HOD, Computer Science & Engineering Ilahia
College of Engineering and Technology

Date:
Place: Muvattupuzha

ACKNOWLEDGEMENT

Apart from the efforts of our, the success of this project report depends largely on the encouragement and guidelines of many others. We take this opportunity to express our gratitude to the people who have been instrumental in the successful completion of this project.

We would like to show our heartfelt gratitude towards **Prof. Dr. K. A. Navas**, Principal, Ilahia College of Engineering and Technology for granting us the permission to work this project. Also, we would like to show our greatest gratitude towards our head of the department of computer science & Engineering, **Dr. Lino Abraham Varghese** and project guide **Ms. Shabna C H,** Our project Coordinators **Ms.Honey John and Ms.Aziya Shirin V.S,** for their valuable advice and guidance.

Finally, we express our gratitude and thanks to all our teachers and other faculty members of the Department of Computer Science & Engineering, for their sincere and friendly cooperation in completing this project.

Date:
Place: Muvattupuzha

JAMESY JOSEPH

PARVATHY VASUDEVAN

RAHMATH BEEVI ASHARAF

ABSTRACT

Efficient attendance management is crucial for educational institutions to monitor student participation and ensure academic accountability. This project presents the development and implementation of "TrackMate: Bluetooth Automated Attendance Marking System," an Android application that leverages Bluetooth technology and biometric authentication to create a seamless and secure attendance tracking system. The TrackMate system addresses the limitations of traditional attendance trackers by automating the process and minimizing manual intervention.

The TrackMate system enhances attendance tracking by establishing a Bluetooth connection between the teacher's device and students' smartphones, incorporating biometric authentication to prevent proxy attendance. It features an Android application for interface, a Bluetooth communication module for data exchange, and a secure database for storing student details and attendance records. Extensive testing, including unit, integration, and user acceptance tests, demonstrated TrackMate's high accuracy and reliability across various classroom scenarios. The system offers significant advantages over traditional methods, such as reduced manual errors, lower administrative burdens, and enhanced security. The centralized database supports comprehensive data analysis and reporting, facilitating data-driven decision-making for educational institutions.

In conclusion, the TrackMate system presents an innovative approach to attendance management, combining Bluetooth technology, biometric authentication, and database management in an Android application. This project contributes to the optimization of administrative processes in educational institutions, paving the way for improved efficiency, accountability, and student engagement. Future enhancements may include additional features such as automated notifications, integration with student information systems, and advanced analytics capabilities.

CONTENTS

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

Bluetooth technology has revolutionized the way we interact with devices, offering seamless connectivity over short distances. One innovative application of Bluetooth automated attendance marking systems, where smartphones or other Bluetooth-enabled devices are used to automate attendance recording processes. This technology eliminates the need for manual attendance taking, streamlining the process and reducing human error. The Bluetooth Automated Attendance Marking System leverages the Bluetooth protocol's ability to detect nearby devices without the need for pairing or extensive setup. By utilizing this feature, the system can identify students or participants carrying Bluetooth-enabled devices as they enter a designated area, such as a classroom or a meeting room. This information is then captured and recorded automatically, creating an efficient and accurate attendance tracking system.

Key components of such a system typically include hardware components like Bluetooth chips, sensors, and possibly switches for activation, along with software components for data processing, storage, and user interface. The system may also integrate with cloud services for centralized data management and accessibility. Overall, Bluetooth Automated Attendance Marking Systems offer a modern and convenient solution for organizations and educational institutions to manage attendance efficiently, saving time and resources while enhancing accuracy and reliability. Attendance tracking is a fundamental aspect of educational institutions, serving as a basis for monitoring student participation and academic progress. Traditionally, attendance has been recorded manually, relying on teachers to mark student presence in a tracker. However, this conventional method is prone to human errors, time-consuming, and lacks real-time data access. As technology continues to advance, there is a growing need for more efficient and reliable attendance management systems.

The motivation for implementing TrackMate extends beyond the desire for more efficient attendance management. The system seeks to enhance the overall educational experience for both teachers and students. By automating attendance tracking, teachers can focus more on instructional activities, fostering a conducive learning environment. Students benefit from a streamlined and accountable attendance system that encourages active participation and facilitates timely interventions when needed. Ultimately, the motivation behind TrackMate is to bridge the gap between traditional attendance trackers and modern technology. By leveraging Bluetooth connectivity, biometric authentication, and database management, TrackMate aims to provide educational institutions with an innovative solution that improves efficiency, accuracy, and student engagement.

CHAPTER 2

LITERATURE SURVEY

Literature review plays an important role in understanding the artificial intelligence domain and implementation of the working system. We have included five literature papers with proper explanations.

Rudra Malali.etal [1] proposed a system and published in the International Journal of Innovative Technology and Exploring Engineering (IJITEE) in June 2019, presents a detailed exploration of using Bluetooth technology for automating attendance management in educational institutions. The authors propose an Android application-based solution that leverages Bluetooth Low Energy (BLE) technology for seamless and efficient attendance tracking. The system's architecture incorporates features such as automatic detection of nearby devices, real-time data synchronization with a central server, and a user-friendly interface for both students and teachers. The paper discusses the implementation of the system, including the software design, database management, and integration with Bluetooth hardware. It also evaluates the system's performance in terms of accuracy, reliability, and user experience, highlighting its potential benefits for simplifying attendance management processes in educational settings.

Sangam Dange.etal [2] proposed a system using NFC, as discussed in the International Journal of Innovative Science and Research Technology, represents an innovative approach to automating attendance tracking in educational environments. By integrating NFC technology alongside Bluetooth, the app aims to enhance the accuracy and reliability of attendance monitoring. NFC tags are strategically placed in key locations, allowing students to tap their smartphones against these tags to register their attendance. This combination of Bluetooth and NFC offers a robust solution that minimizes manual effort and improves the efficiency of attendance management processes. The app's utilization of NFC not only simplifies the attendance logging process but also ensures real-time data synchronization and security. Each NFC tag is uniquely identified, reducing the risk of fraudulent attendance entries. Moreover, the seamless integration of NFC with Bluetooth technology provides a user-friendly experience, making attendance tracking convenient for both students and faculty. Overall, the Bluetooth-based Attendance Management App using NFC represents a significant advancement in leveraging mobile technology to streamline attendance monitoring in educational institutions.

Rayyan Banjar.etal [3], from the Computer Engineering Department at Umm Al Qura University presents a robust solution for automating attendance tracking using smartphones. This system utilizes Bluetooth technology to detect students' presence within designated areas equipped with Bluetooth beacons. By leveraging the Bluetooth capabilities of smartphones, the system accurately records attendance based on proximity, eliminating the need for manual attendance taking. One key aspect of this system is its user-friendly interface, which simplifies the attendance logging process for both students and faculty members. The system's implementation of Bluetooth technology ensures seamless connectivity and real-time data synchronization, enhancing the overall efficiency of attendance management in educational institutions. Overall, the Bluetooth-based Smartphone Attendance System offers a reliable and modern approach to attendance tracking, contributing to streamlined administrative processes and improved accuracy in attendance monitoring.

Arulogun O.T.etal [4] presents a comprehensive solution for automating attendance tracking in educational institutions. The system utilizes Radio Frequency Identification (RFID) technology, which involves tagging students with RFID cards or tags that emit signals containing unique identification information. When students enter or leave the premises, RFID readers capture these signals, allowing the system to record attendance accurately and in real-time. This approach eliminates manual attendance taking, reduces errors, and enhances efficiency in attendance management. The research likely delves into the technical aspects of RFID implementation, system architecture, data security measures, and the benefits gained from deploying such a system in educational settings. It may also discuss challenges encountered and potential areas for future enhancements or research in RFID-based attendance management systems.

V. Rajesh.etal [5] proposed a system using BLE technology, researched by V. Rajesh.etal [5] from Anil Neerukonda Institute of Technology and Sciences in 2019, presents an innovative approach to attendance tracking in educational environments. BLE technology, known for its low power consumption and short-range wireless communication capabilities, is utilized to automate the attendance process. Students are provided with BLE-enabled devices, such as smartphones or ID cards, which communicate with BLE beacons installed in classrooms or strategic locations on campus. As students enter these areas, their devices automatically connect to the beacons, enabling the system to record their attendance without manual intervention. The research likely explores the technical aspects of BLE implementation, system architecture, integration with attendance management software, data privacy and security considerations, and the advantages of using BLE technology for attendance management. It may also discuss user experience, scalability, and potential enhancements or future research directions in BLE-based attendance systems.

CHAPTER 3

PROPOSED SYSTEM

The proposed system Bluetooth Automated Attendance Marking System, TrackMate is an Android application that leverages Bluetooth technology and biometric authentication to automate and streamline the attendance tracking process in educational institutions. The system offers a robust and efficient solution that enhances accuracy, reliability, and security in attendance management.

3.1 INTRODUCTION

The key features of this system are:

Bluetooth Connectivity:
The TrackMate application establishes a Bluetooth connection between the teacher's device and the students' smartphones. This connection enables real-time communication and attendance tracking within the classroom environment. Bluetooth technology ensures seamless and reliable data exchange, even in crowded or interference-prone settings.

Attendance Detection:
TrackMate utilizes the Bluetooth connection to detect student presence automatically. When a student enters the classroom, their smartphone establishes a connection with the teacher's device, signaling their attendance. Conversely, when a student's device moves out of range or disconnects, the system marks them as absent.

Biometric Authentication:
To ensure accurate identification, TrackMate incorporates biometric authentication, such as fingerprint or facial recognition, from the student's device. This authentication process validates the student's identity, minimizing the possibility of proxy attendance.

Centralized Database:
All attendance records and student details are securely stored in a centralized database. The database allows easy access to historical attendance data and facilitates comprehensive reporting and analysis. Teachers and administrators can generate reports, track student attendance patterns, and identify trends for further assessment.

User-friendly Interface:
The TrackMate application provides a user-friendly interface for both teachers and students. Teachers can quickly view and manage attendance records, monitor student presence, and generate reports. Students have access to their attendance history and receive notifications or reminders for upcoming classes.

Data Security and Privacy:
TrackMate ensures the security and privacy of student data through encryption and strict access controls. Only authorized personnel, such as teachers and administrators, have access to attendance records and sensitive information.

3.2 ADVANTAGES

Automation and Efficiency:
The TrackMate system automates the attendance tracking process, eliminating the need for manual recording. This automation saves time and reduces administrative burdens, allowing teachers to focus more on instructional activities.

Real-time Data Access:

TrackMate provides real-time access to attendance data for teachers, administrators, and parents. They can instantly view attendance records, track student presence, and address attendance issues promptly.

Accurate and Reliable Attendance Tracking: With Bluetooth connectivity, TrackMate offers accurate and reliable attendance tracking. The system detects student presence or absence in real-time, minimizing errors and proxy attendance.

Enhanced Security:
The integration of biometric authentication adds an extra layer of security to the attendance system. It ensures that only authorized individuals can mark attendance, reducing the risk of identity fraud or proxy attendance.

Centralized Database:
TrackMate stores attendance records and student details in a centralized database. This allows for easy access, comprehensive reporting, and data analysis. Teachers and administrators can generate reports, identify attendance patterns, and make data-driven decisions.

User-friendly Interface:
TrackMate provides a user-friendly interface for both teachers and students. Teachers can easily manage attendance records and generate reports, while students can access their attendance history and receive notifications or reminders for upcoming classes.

Improved Student Engagement:
TrackMate promotes student accountability and engagement by providing transparency and easy access to attendance records. Students can track their own attendance, stay informed, and take ownership of their academic progress.

Data Security and Privacy:
TrackMate ensures the security and privacy of student data through encryption and strict access controls. Only authorized personnel have access to attendance records, protecting student privacy and data integrity.

Scalability and Compatibility:
The proposed system is built on Android technology, making it compatible with a wide range of smartphones and devices. It can be easily scaled and adapted to accommodate the needs of various educational institutions.

Future Enhancements:

The TrackMate system provides a foundation for future enhancements and integration. It can be extended to include additional features like automated notifications, integration with student information systems, or advanced analytics capabilities.

3.3 DISADVANTAGES

- Incompatibility with Modern Technology: Older systems may not be compatible with modern technologies, such as smartphones or biometric authentication, limiting the ability to integrate advanced features or streamline the attendance process.
- Lack of Student Engagement: Traditional attendance systems often lack mechanisms to actively engage students in the attendance tracking process. Students may not have visibility into their own attendance records or receive timely notifications regarding their attendance status
- Android-exclusive: The software that are specifically designed to run on devices that uses the Android operating system
- Offline Functionality: Apps that require a constant internet may lack offline functionality. This means users cannot use certain features or access content when they are offline, The application should not allow teachers to mark attendance even when there's no internet connection.
- Maintenance and Updates: Bluetooth technology evolves over a time,and maintaining compatibility with newer Bluetooth versions and devices requires ongoing maintenance and updates to the system
- Mobility Constraints: Users are restricted in terms of mobility since they must remain within the Wi-Fi range to use the app effectively. This can be particularly problematic for apps intended for use in dynamic or mobile environments.

CHAPTER 3

SYSTEM DESIGN

System design is the process of defining the architecture, components, modules, interfaces, and data for a system to satisfy specified requirements. It involves making decisions about how various elements of a system will work together to achieve the desired functionality, performance, scalability, and other quality attributes.

3.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. The Fig 3.1 given below shows the architecture of the system and Fig 3.2 shows the block diagram.

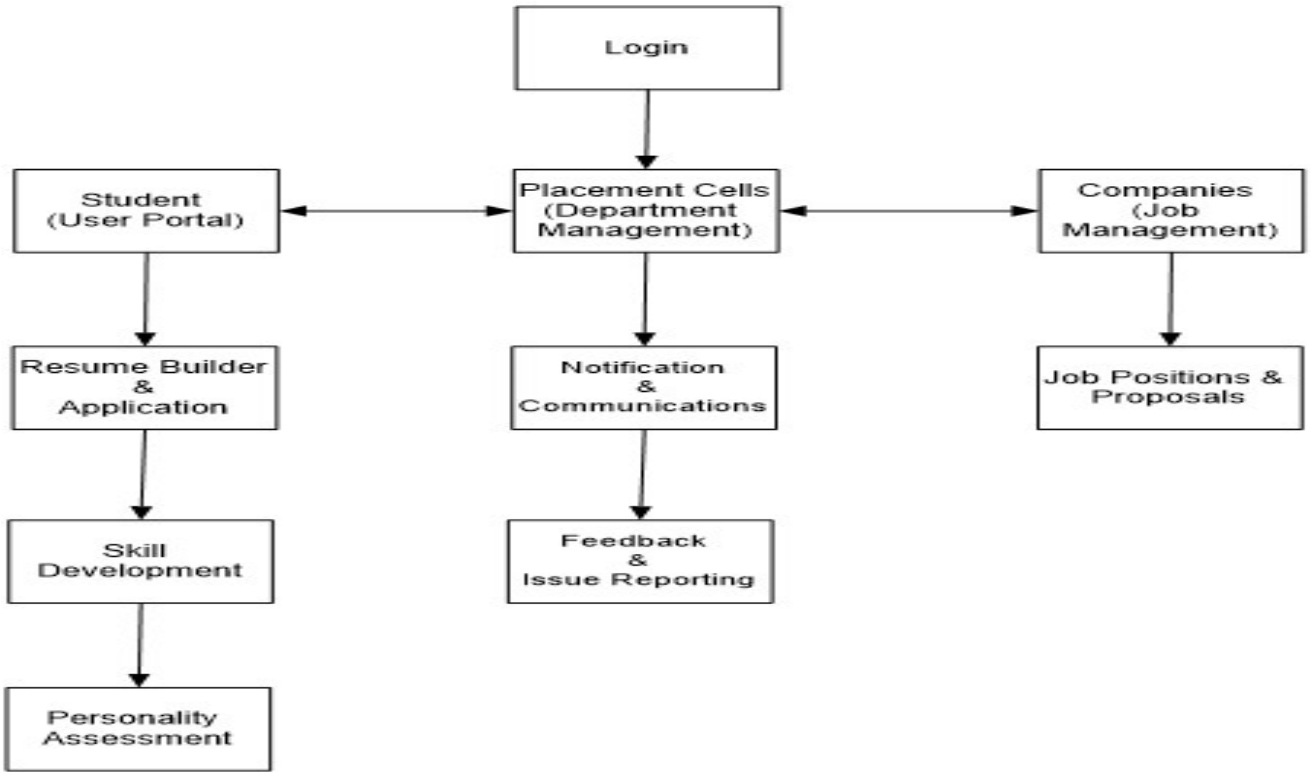


Fig 3.1. System Architecture

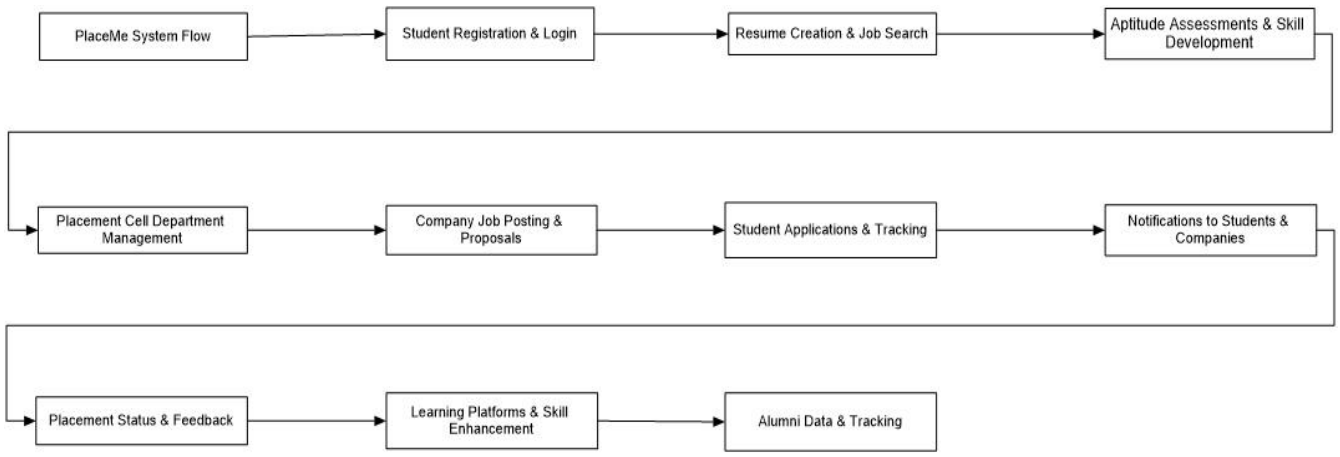


Fig 3.2. Block Diagram

Student Registration & Login

The Student Registration and Login Module in the College Placement App enables students to create an account and access the app's features securely. Key features include user-friendly registration forms, validation, unique username generation, password encryption, email/mobile verification, secure login, password recovery, session management, and authorization checks. Functional requirements include registering as a new student, logging in, recovering forgotten passwords, updating profiles, and logging out. The database schema includes tables for students, login history, and profile information, ensuring a seamless and secure experience for students to access the app's placement services.

Resume Creation & Job Search

The Resume Creation and Job Search Module in the College Placement App enables students to craft professional resumes and search for job opportunities. Students can build new using pre-designed templates, search and filter job listings by location, industry, and job type, and apply directly through the app. The app provides a common template for all students making it as unified template for all. Additional features include resume building and uploading, job application tracking, company profiles and reviews, interview scheduling and reminders, and push notifications for job matching and application deadlines, streamlining the job search process and connecting students with potential employers to enhance their chances of securing placements.

Aptitude assessment and skill Development

The Aptitude Assessment and Skill Development Module in the College Placement App evaluates student's cognitive abilities, technical skills, and personality traits to identify areas for improvement. It offers standardized aptitude tests, technical skill developments, and personality evaluations, providing personalized development plans and recommendations. Students gain access to online courses, tutorials, and resources, with progress tracking and analytics to monitor growth, ultimately empowering them to bridge skill gaps and enhance their employability and placement prospects.

Placement Cell Department Management

The Placement Cell Development & Management Module in the College Placement App streamlines college placement processes, enabling administrators to manage student placements efficiently. It features company relationship management, job posting and vacancy tracking, student resume and profile management, interview scheduling and coordination, placement statistics and reporting, and automated notifications and reminders, ultimately optimizing placement operations and fostering strong industry connections to improve student employment outcomes.

Company Posting and Proposal

The Company Posting and Proposal Module allows companies to post job openings, internships, and project proposals directly to the college's placement platform. Companies can create profiles, specify requirements, and upload job descriptions, while college administrators review, approve, and match suitable students. Students receive notifications and apply directly through the app, streamlining the hiring process and facilitating connections between companies and emerging talent.

Student Application and Tracking

The Student Application and Tracking Module enables students to apply for job openings and track their application status in real-time. Students can view eligible job postings, submit resumes and cover letters, and receive notifications on application status, interview schedules, and outcome updates. College administrators can monitor student applications, manage deadlines, and analyze placement statistics, ensuring seamless communication and efficient placement processes.

Notification to Students & Companies

The Notification Module sends automated alerts and updates to students and companies, ensuring timely communication and seamless placement processes. Notifications include job posting alerts, application status updates, interview schedules, company visit announcements, and placement offer notifications, keeping students informed and engaged, while also updating companies on application receipts, student selections, and interview schedules.

Placement Status & Feedback

The Placement Status & Feedback Module tracks and displays real-time placement statistics, allowing students, colleges, and companies to monitor progress. Students can provide feedback on their placement experience, while companies can rate student performance, providing valuable insights for future improvements. The module also includes features such as placement analytics, student evaluation reports, and company satisfaction surveys, facilitating data-driven decision-making and enhancing the overall placement ecosystem.

Learning Platforms & Skill Enhancement

The Learning Platforms & Skill Enhancement Module provides students with access to curated online courses, tutorials, and resources to develop industry-relevant skills. Partnering with leading learning platforms, this module offers personalized learning paths, skill assessments, and certifications, enabling students to bridge skill gaps and enhance employability. Key features include course recommendations, progress tracking, and badges/certificates, empowering students to upskill and reskill for better placement outcomes.

Alumni Data & Tracking

The Alumni Data and Tracking Module stores and manages alumni information, tracking their career progression and achievements. The module enables colleges to maintain lifelong connections with alumni, gather insights on career outcomes, and showcase success stories. Key features include alumni profiles, placement details, contact details. Therefore students can connect with their alumni easily.

3.3 DATA FLOW DIAGRAM

Data Flow Diagram (DFD) also called Data Floe Diagram is commonly used during problem analysis. They are quite general and are not limited to problem analysis for software requirement specification. DFD is very useful in understanding a system and is effectively used during analysis. Fig 3.3.1, Fig 3.3.2, Fig 3.3.3, Fig 3.3.4, Fig 3.3.5 shows the different levels of dataflow diagram.

DFD Diagram (Level – 0)

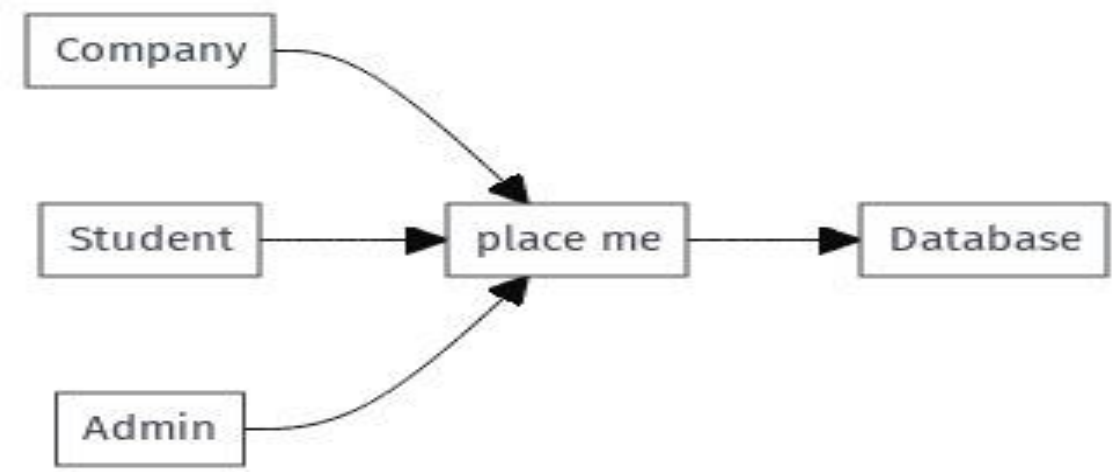


Fig 3.3.1. DFD level 0

DFD Diagram (Level – 1 Admin)

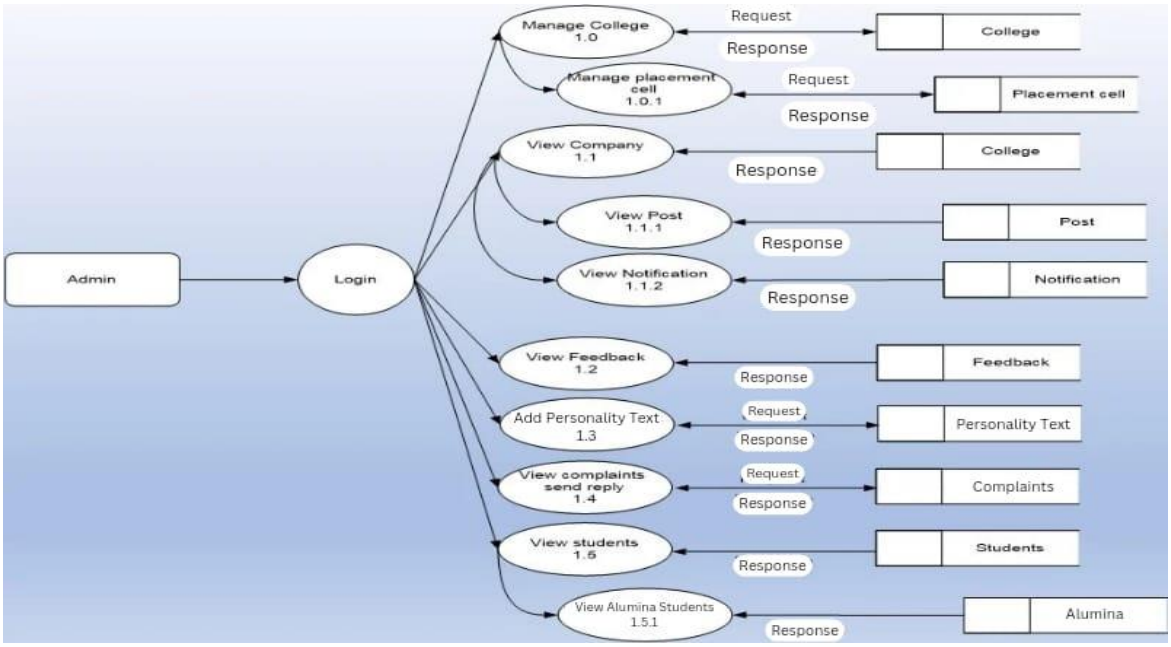


Fig 3.3.2. DFD level 1 (Admin)

DFD Diagram (Level – 1 Company)

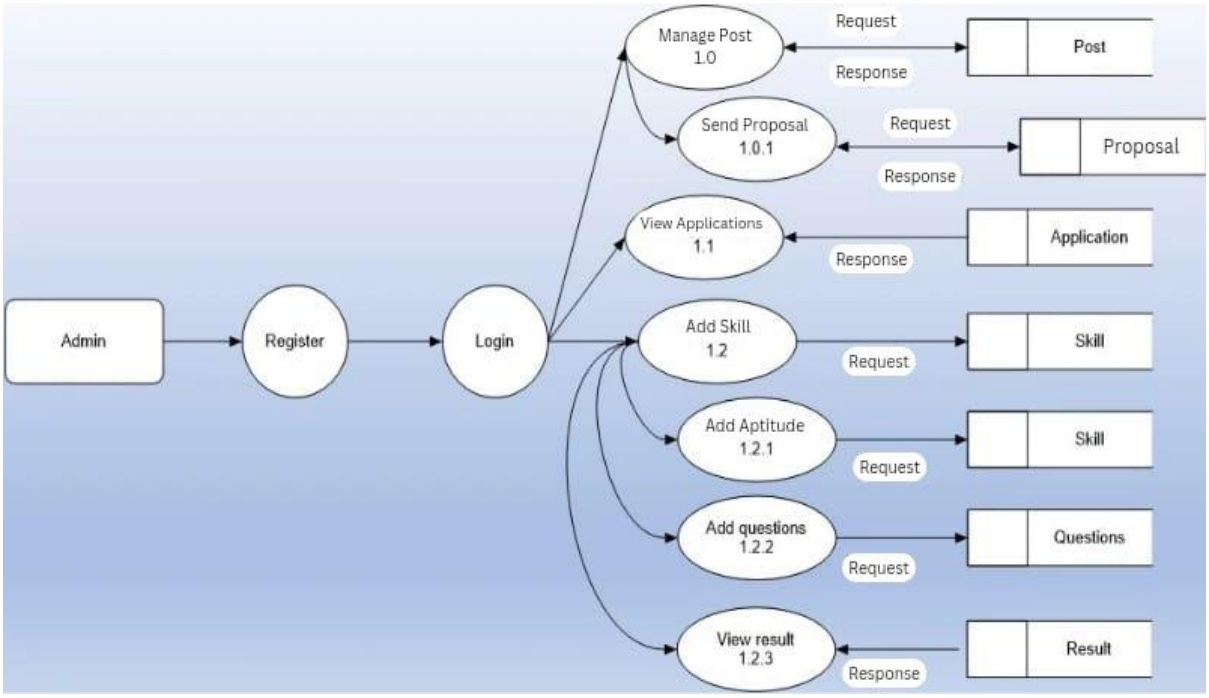


Fig 3.3.3. DFD level 1 (Company)

DFD Diagram (Level – 1 Placement Cell)

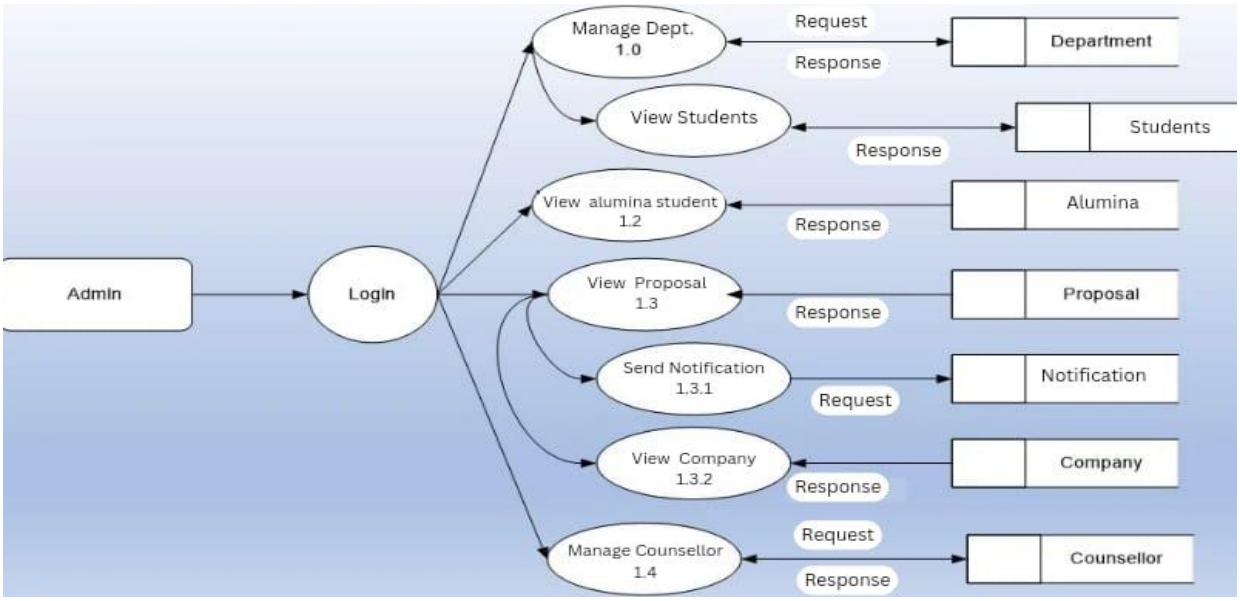


Fig 3.3.4. DFD level 1 (Placement Cell)

DFD Diagram (Level – 1 Students)

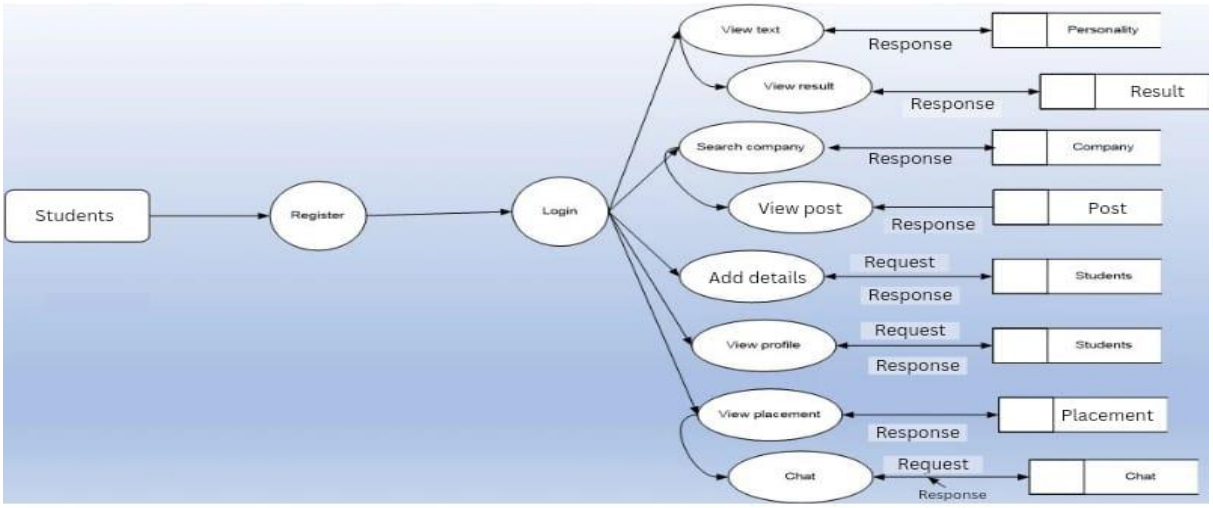


Fig 3.3.5. DFD level 1 (Students)

3.4 UML DIAGRAM

Unified Modeling Language (UML) diagrams visually represent software systems, illustrating components, relationships, and interactions. Uses include defining requirements, modeling complexity, facilitating communication, and guiding implementation, with common types including Class, Use Case, Sequence, and Deployment diagrams. Fig 3.4 shows the UML Diagram.

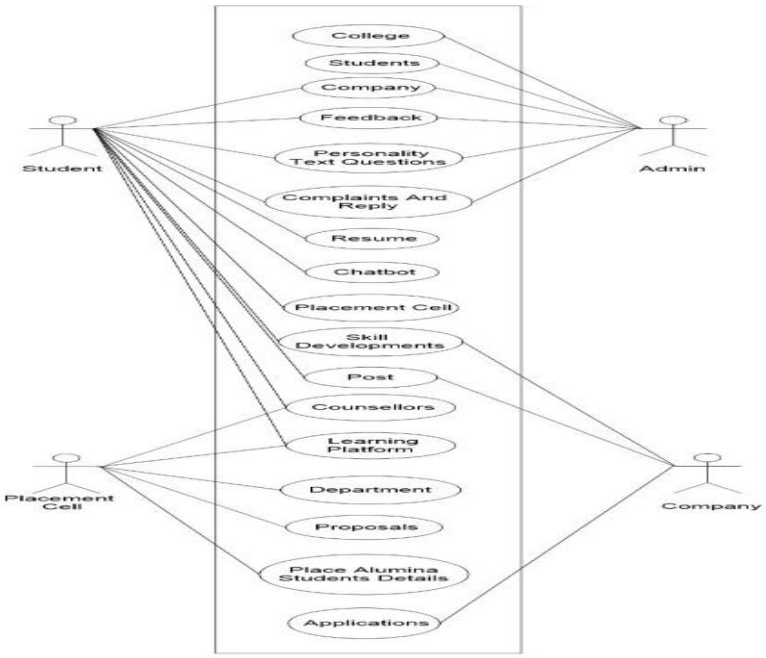


Fig 3.4. UML Diagram

3.5 DATABASE SCHEMA

A database schema is a blueprint outlining database structure, including tables, relationships, and data types. It ensures data integrity, scalability, and performance, and is used for data modeling, database optimization, query optimization, and data migration. Fig 3.5 shows the database schema.

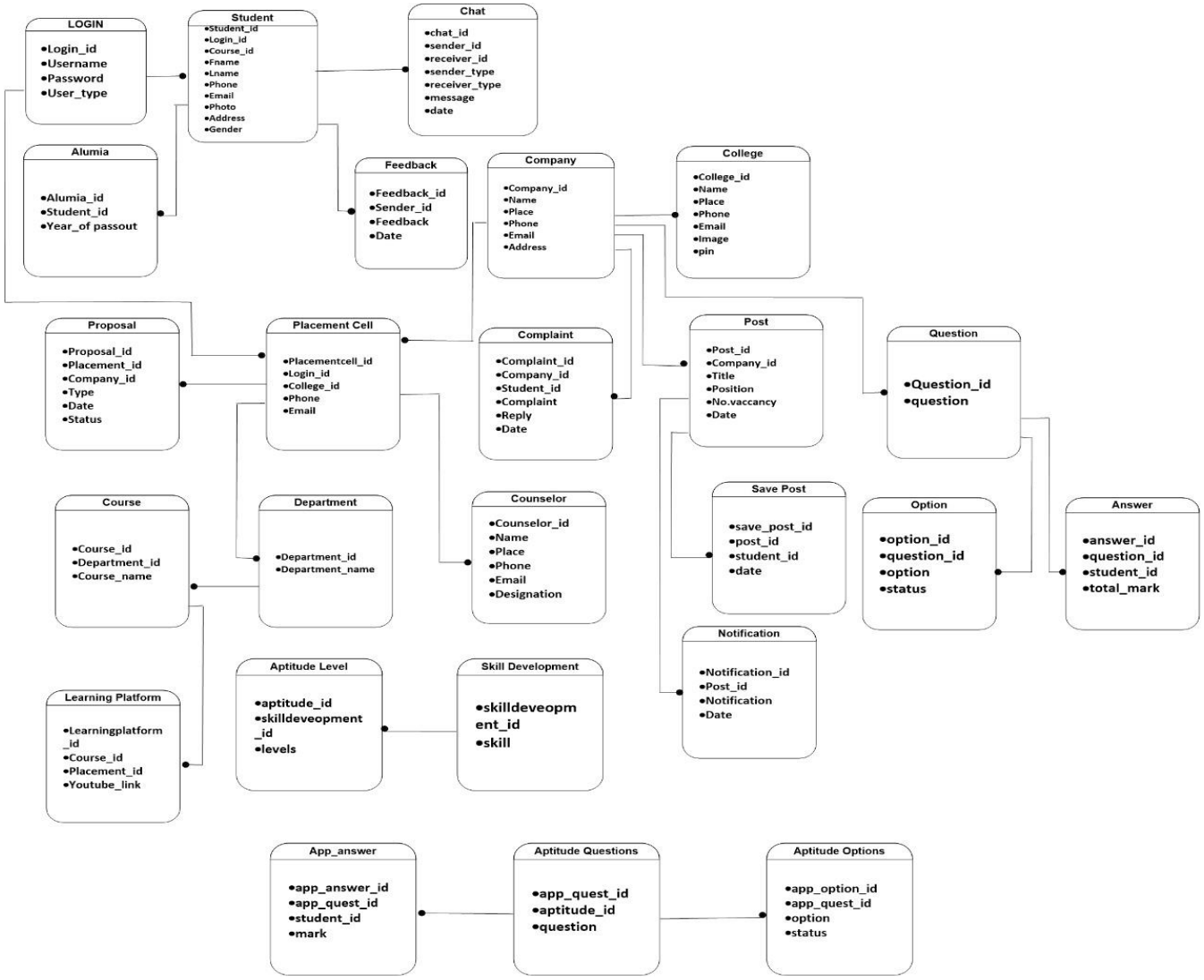


Fig 3.5. Database Schema

3.6 GUI DESIGN

Graphical User Interface (GUI) design is used to create intuitive, visually appealing, and user-friendly interfaces for software applications, websites, mobile apps, gaming consoles, and devices, enhancing user experience and interaction. Fig 3.6.1, Fig 3.6.2, Fig 3.6.3, Fig 3.6.4, Fig 3.6.5 shows the different pages of graphical user interfaces.

3.6.1 Home Page

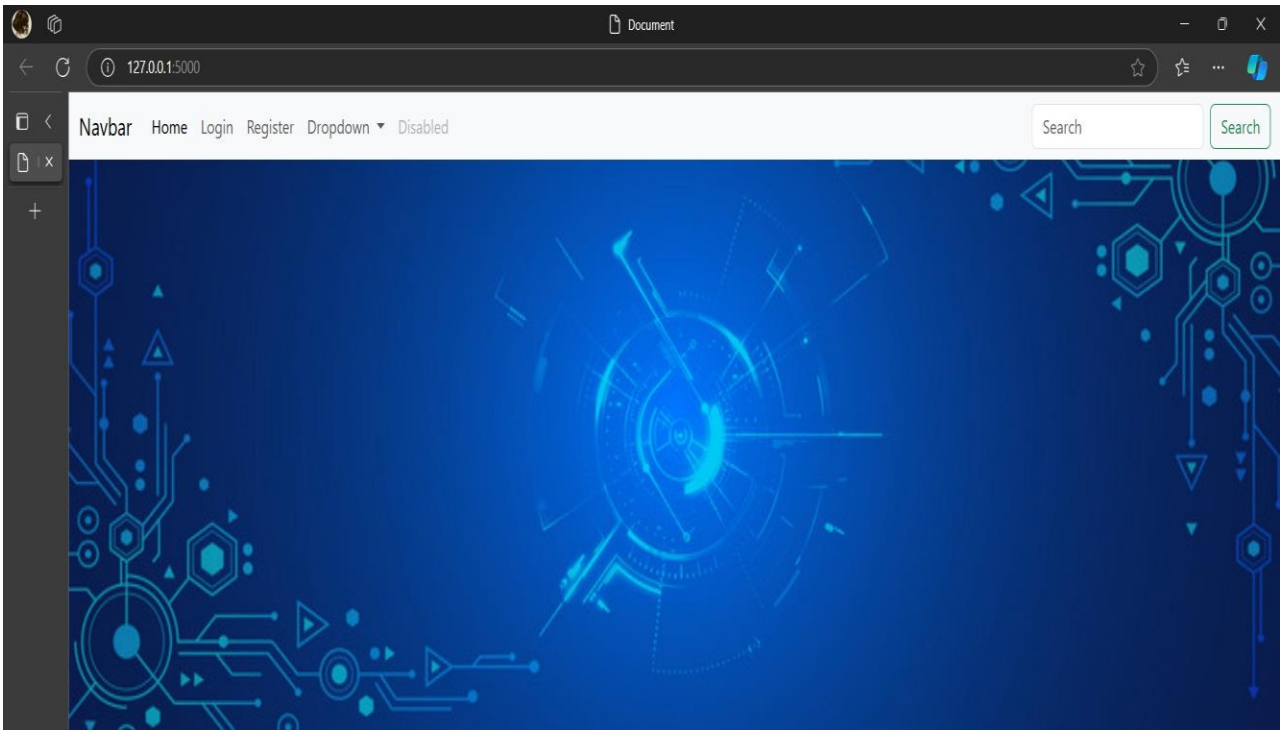


Fig 3.6.1. Home Page

3.6.2 Student Registration

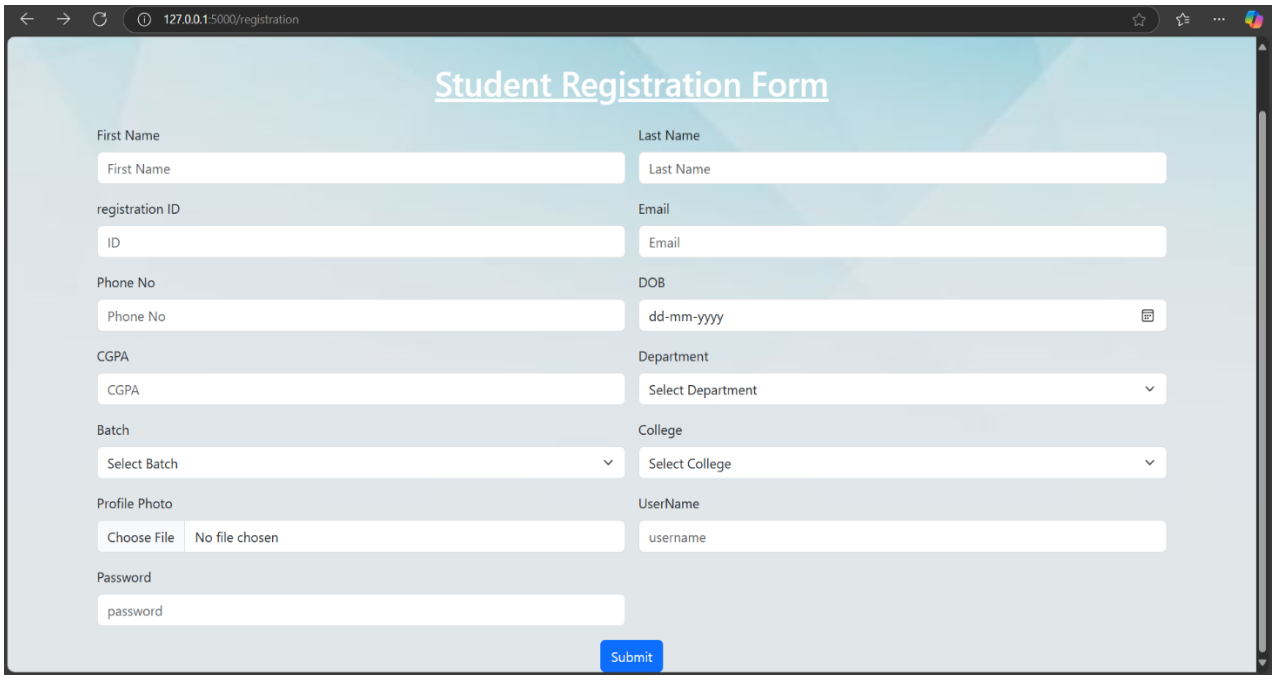


Fig 3.6.2. Student Registration

3.6.3 Placement Cell Registration

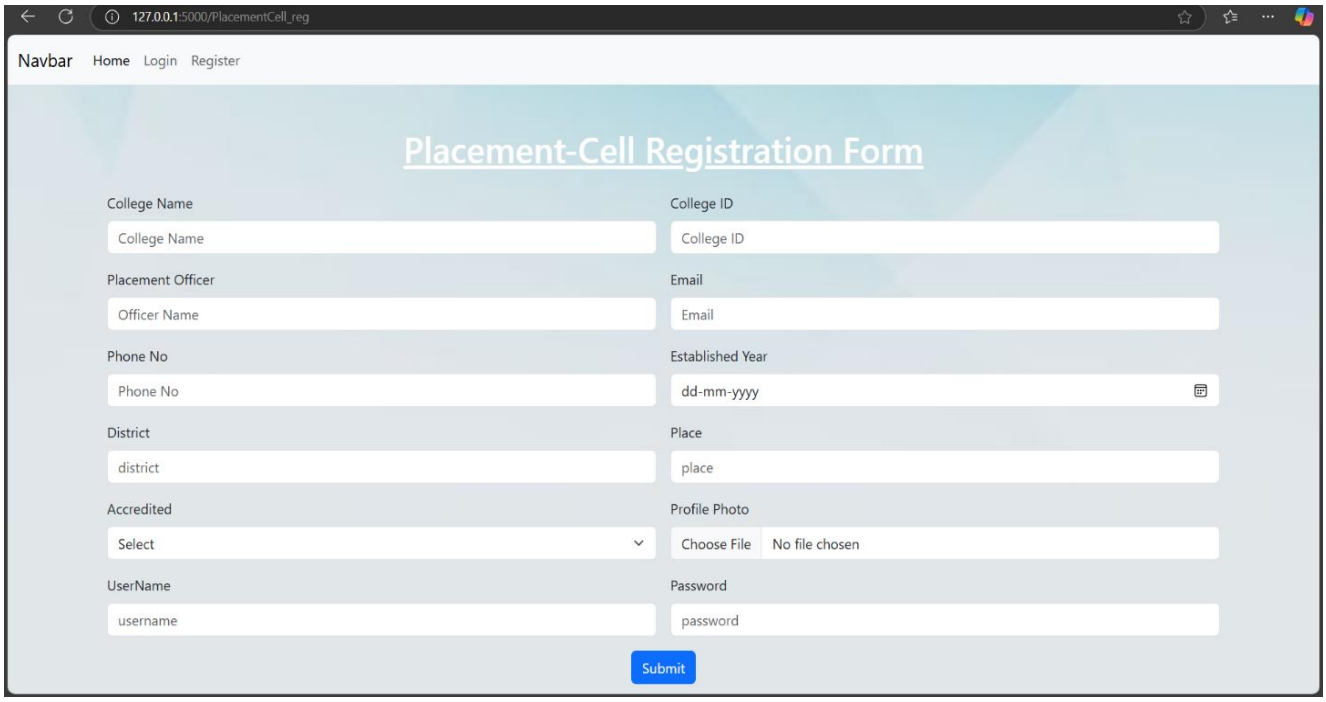


Fig 3.6.3. Placement Cell Registration

3.6.4 Login / Sign-In Page

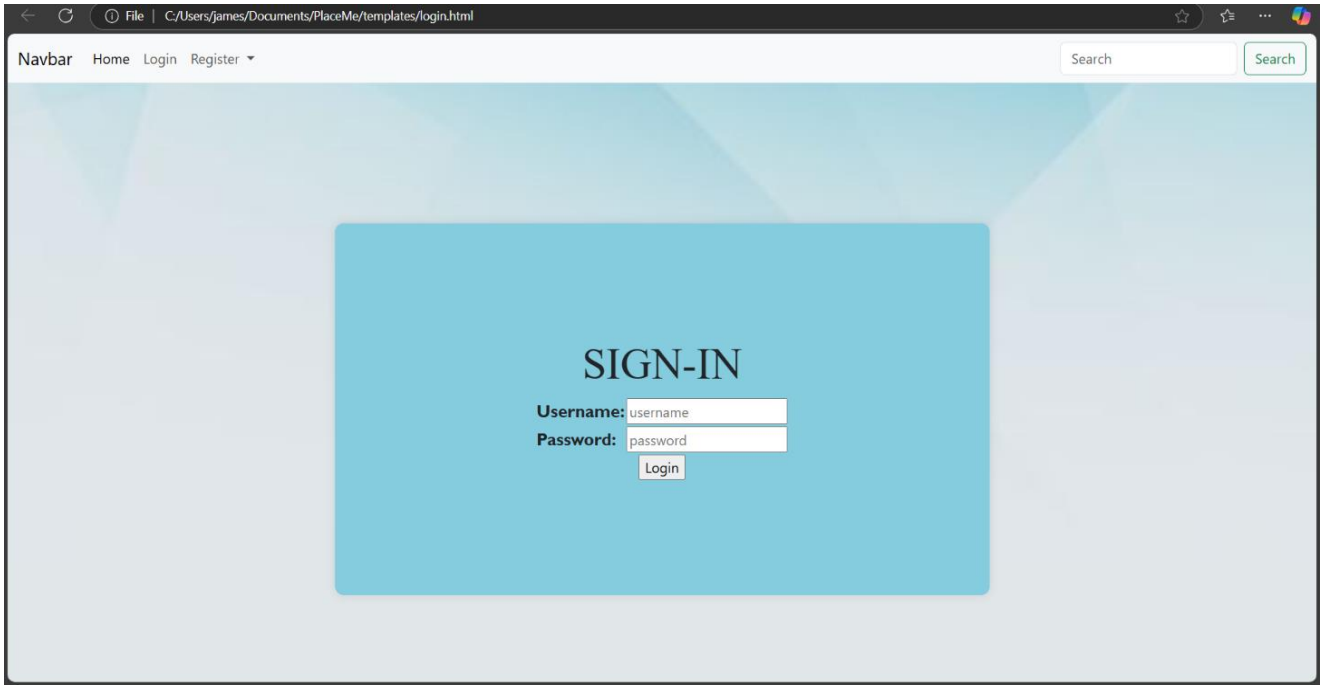


Fig 3.6.4. Login / Sign-In Page

3.6.5 Placement Cell Home Page

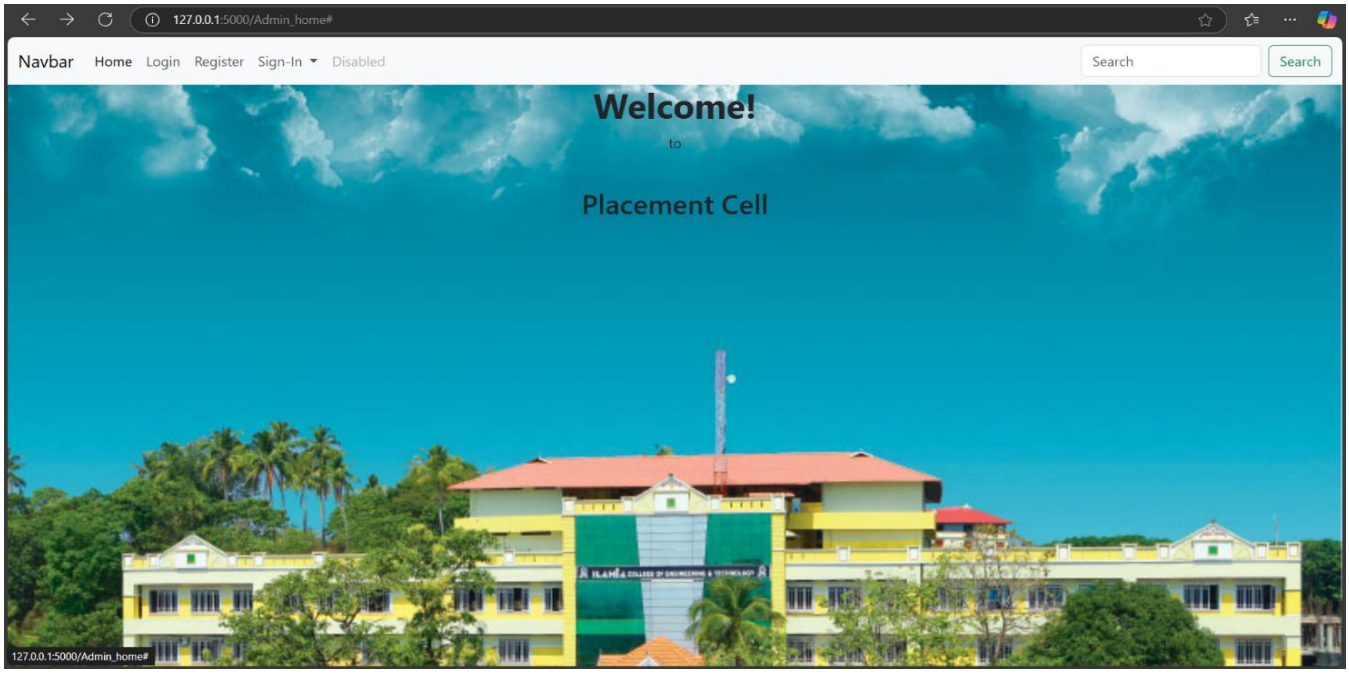


Fig 3.6.5. Placement Cell Home Page

CHAPTER 5

SYSTEM REQUIREMENTS

System requirements are the minimum hardware and software specifications necessary for a computer system to effectively run a specific software application or operating system. These specifications cover aspects like processor speed, RAM, storage space, and graphics capabilities, ensuring smooth performance on the user's device.

5.1 SOFTWARE REQUIREMENTS

Operating System: Windows 10 (or compatible OS) for development.

Integrated Development Environment (IDE): Android Studio, the official IDE for Android development, which provides a comprehensive environment for coding, testing, and debugging Android applications.

Kotlin Development Kit: Required for developing Android applications using Kotlin programming language.

Android Software Development Kit (SDK): Provides necessary tools, libraries, and APIs for Android app development.

Gradle Build System: Used to automate the build process and manage dependencies for the Android app.

Version Control System: Git for source code management and collaboration.

Additionally, during the development process, emulators or physical Android devices can be used for testing the TrackMate application. The Android Emulator, provided by Android Studio, allows for virtual device testing on the computer.

5.2 HARDWARE REQUIREMENTS

The selection of hardware configuring is a very task related to the software development, particularly inefficient RAM may affect adversely on the speed and corresponding on the efficiency of the entire system. The processor should be powerful to handle all the operations. The hard disk should have the sufficient to solve the database and the application.

Hardware Used for Development:

CPU: Intel i5 Processer, Memory: 8 GB, Cache: 6 MB, Hard Disk: 512 GB, Monitor: 15.6”, Keyboard:

Standard 108 keys Enhanced, Keyboard Mouse: Optical Mouse

Minimum Hardware Required for Implementation:

CPU: Pentium V Processor, Memory: 1 GB Above, Cache: 512 KB Above, Hard Disk: 256 GB Above,

Monitor: Any, Keyboard: Any.

CHAPTER 6

SOLUTION METHODOLOGY

The solution methodology for a Bluetooth automated attendance marking system involves agile or waterfall development, with security measures and compliance integrated. Continuous feedback drives usability improvements, aiming to enhance efficiency for marking attendance.

6.1 ARCHITECTURE DESIGN

The system architecture diagram is an abstract depiction of the system's component architecture. It provides a succinct description of the system's component architecture in order to assist in component-component relationships and system functioning. The figures 6.1 a, 6.1 b and 6.1 c show the architecture design of Admin, Teacher, and Student respectively.

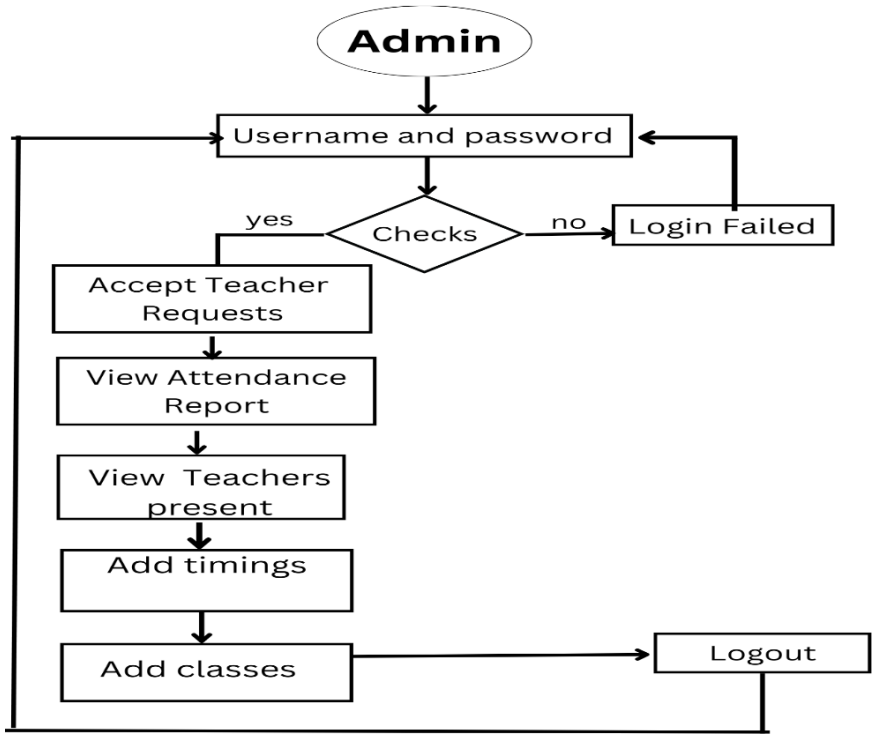


Fig 6.1 (a) Architecture Design of Admin

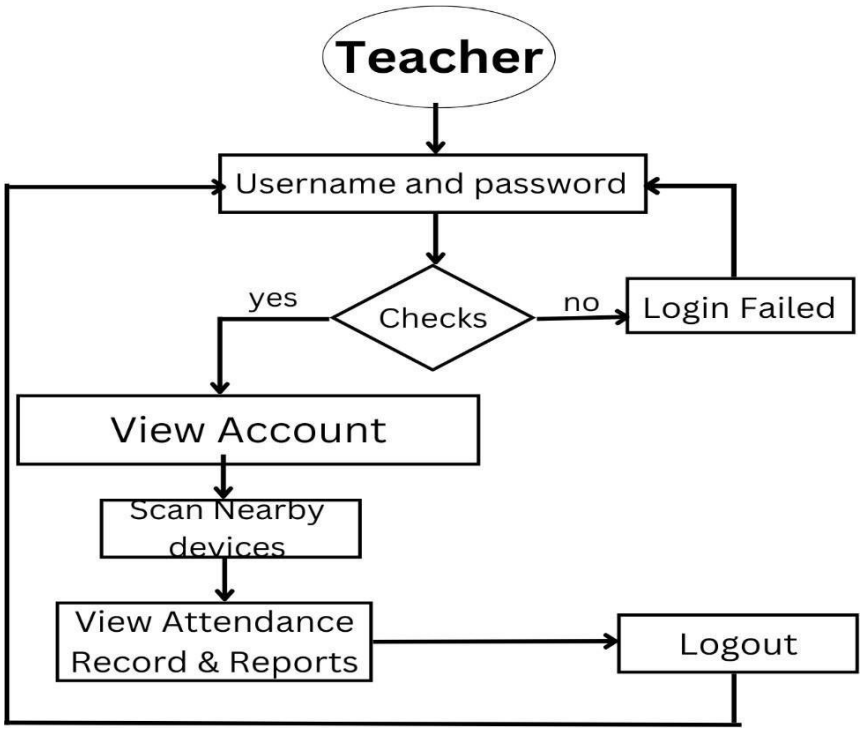


Fig 6.1(b) Architecture Design of Teacher

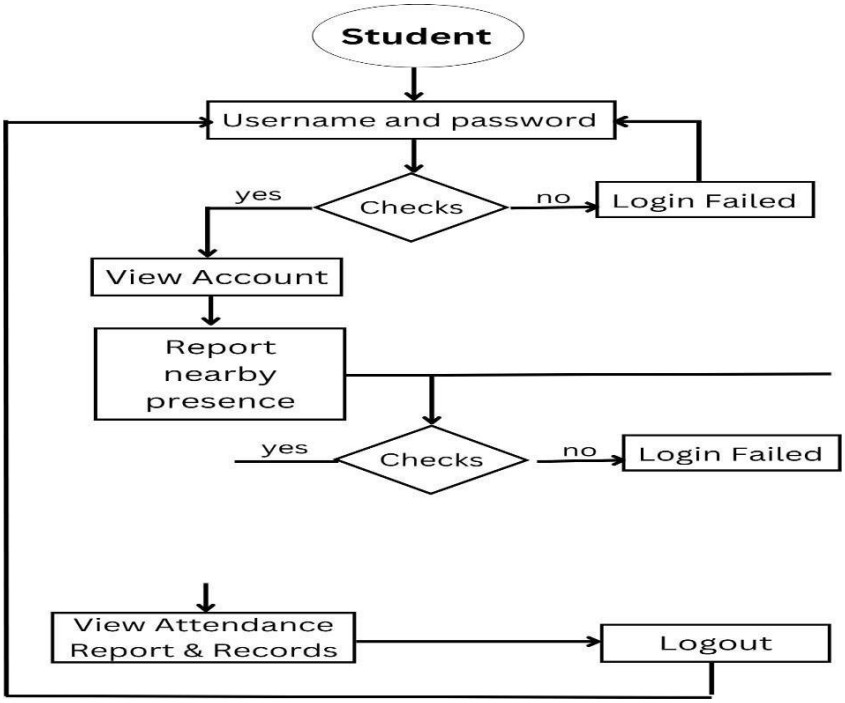


Fig 6.1 (c) Architecture Design of Student

6.2 USE CASE DIAGRAM

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well. The use cases are represented by either circles or ellipses.

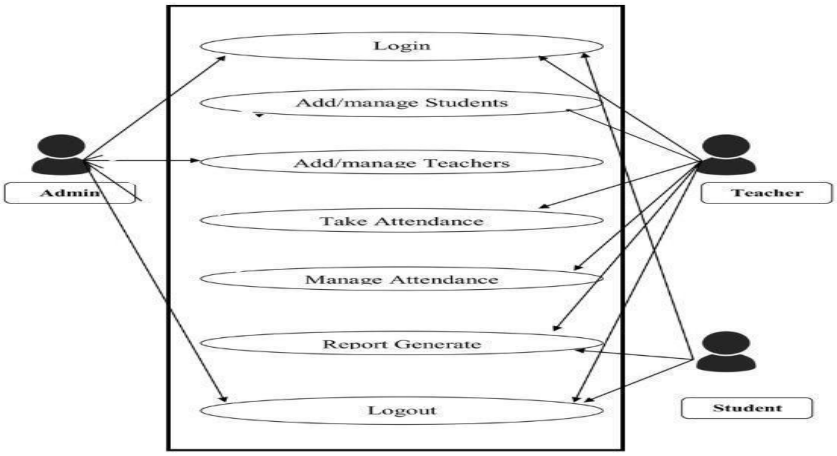


Fig 6.2: Use Case Diagram

6.3 ACTIVITY DIAGRAM

An activity diagram visually presents a series of actions or flow of control in a system similar to a flowchart or a data flow diagram. Activity diagrams are often used in business process modeling. They can also describe the steps in a use case diagram.

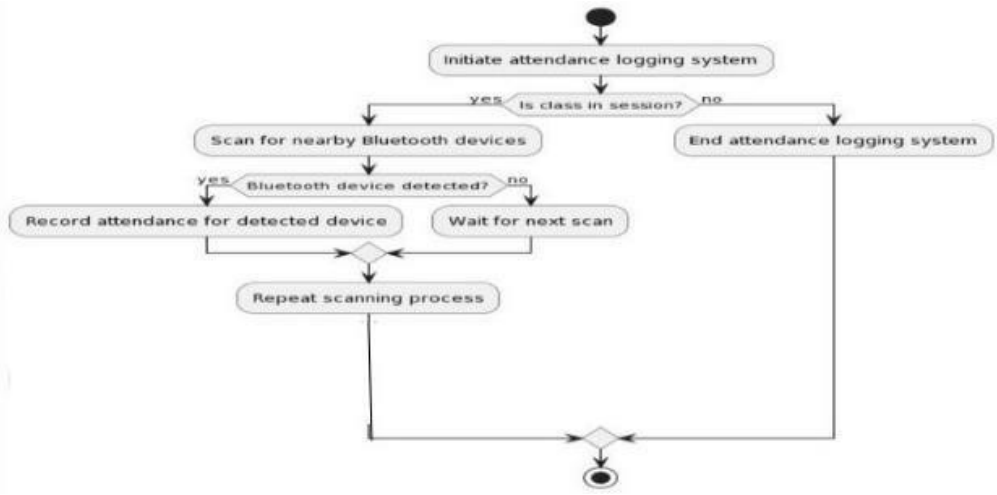


Fig 6.3: Activity Diagram

6.4 DATAFLOW DIAGRAM

Data Flow Diagram (DFD) also called Data Floe Diagram is commonly used during problem analysis. They are quite general and are not limited to problem analysis for software requirement specification. DFD is very useful in understanding a system and is effectively used during analysis.

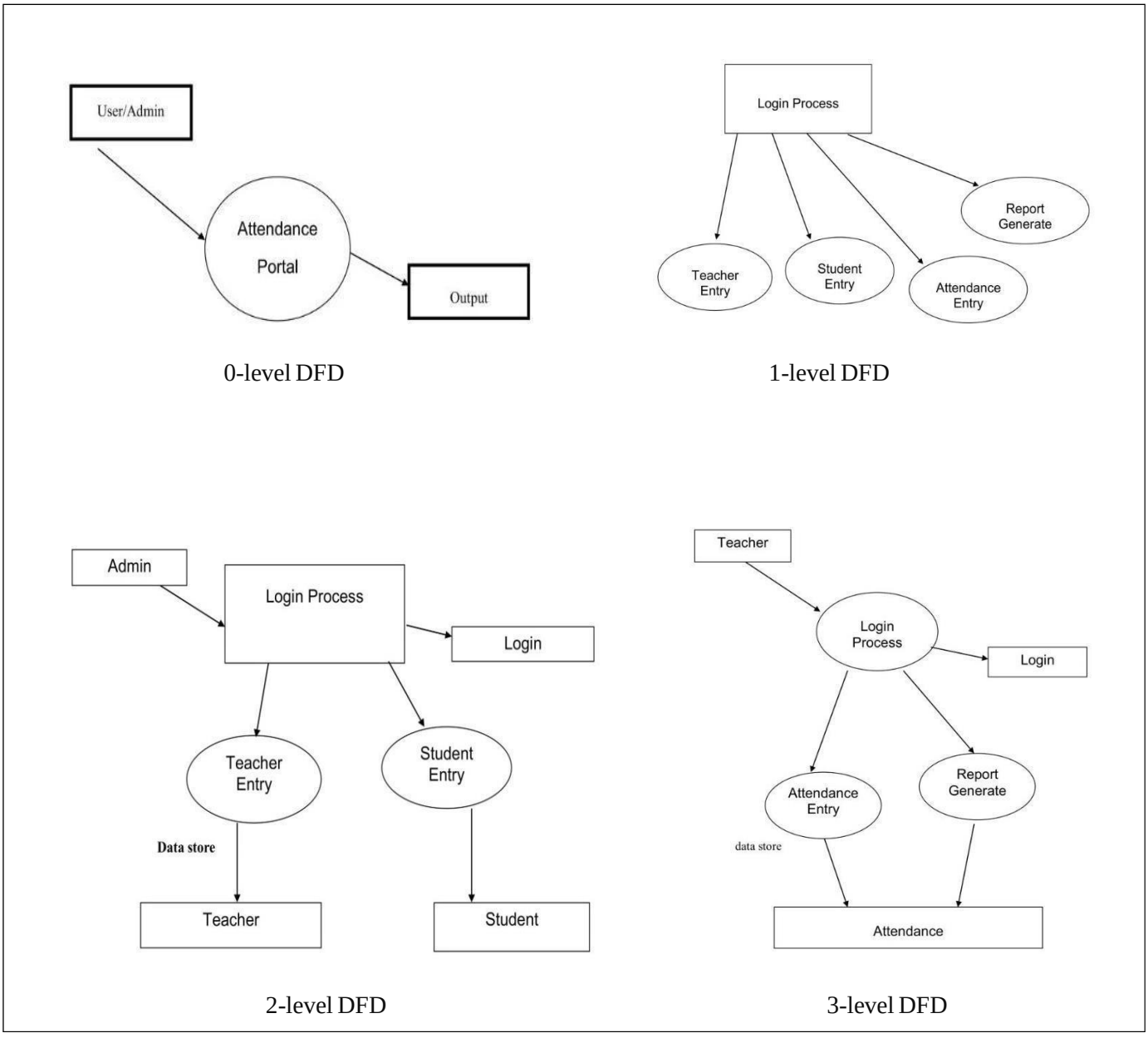


Fig 6.4: Data Flow Diagram

6.5 SEQUENCE DIAGRAM

A sequence diagram or system sequence diagram shows process interactions arranged in time sequence in the field of software engineering. It depicts the processes involved and the sequence of messages exchanged between the processes needed to carry out the functionality.

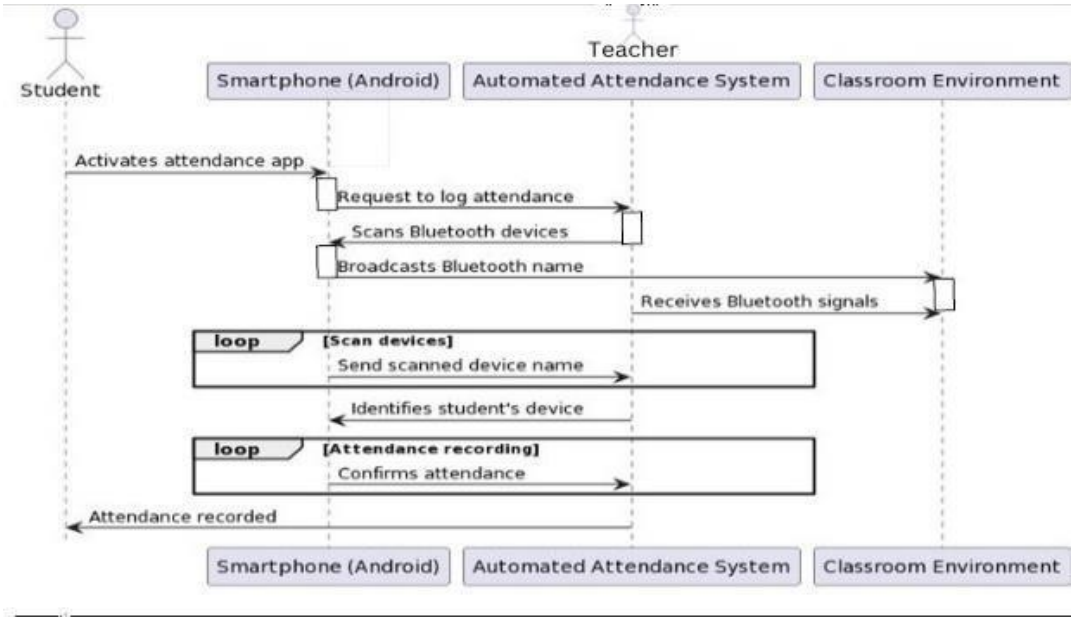


Fig 6.5: Sequence Diagram

CHAPTER 7
IMPLEMENTATION

Implementing a Bluetooth automated attendance marking system involves developing user-friendly software solutions, integrating with external systems, and providing training for efficient usage. The aim is to improve the efficiency of attendance by automating it.

7.1 BACKEND DEVELOPMENT

Programming Language: JavaScript

Development Environment: Visual Studio Code (VS Code)

Framework: Express.js (Node.js-based web application framework)

Database: JSON DB (Filesystem)

API Development and Testing: Postman (API development and testing tool)

7.2 ANDROID APPLICATION DEVELPOMENT

Programming Language: Kotlin

Integrated Development Environment (IDE): Android Studio

7.3 DEPLOYMENT

Backend Server:
Hosting: The backend server is hosted on a local machine or a local server accessible within the network. Localhost IP Address: The backend server can be accessed using the local host IP address (e.g., 127.0.0.1) or the IP address of the hosting machine. Port Configuration: The backend server is configured to listen on a specific port (e.g., 8080) to handle incoming requests. Firewall and Network Configuration: Ensure that the hosting machine's firewall settings and network configuration allow incoming connections to the backend server.

Android Application:
Build Configuration: Build the Android application using Android Studio, specifying the appropriate backend server URL and port number in the application's configuration files. APK Generation: Generate the APK file of the Android application using Android Studio's build tools. Manual Installation: Distribute the APK file to the target devices either by transferring it directly or hosting it on a local server accessible within the network. Enable Unknown Sources: To install the Android application outside the Google Play Store, users must enable the "Unknown Sources" option in the device's security settings to allow installations from sources other than the Play Store.

7.4 SECURITY CONSIDERATIONS

Access Control: Implement appropriate access control measures to restrict unauthorized access to the backend server and the Android application. Encryption: Utilize secure communication protocols (e.g., HTTPS) to protect sensitive data transmission between the Android application and the backend server. Authentication and Authorization: Implement secure authentication mechanisms for teachers, students, and administrators to ensure only authorized users can access the system and perform necessary actions. Data Privacy: Comply with data privacy regulations and best practices to protect the personal information of users stored in the system's database.

7.5 TESTING

Testing is a critical phase in the development process of Bluetooth Automated Attendance Marking to ensure its functionality, performance, and reliability. Here are some important aspects of testing for the project:

Unit Testing:
Backend: Write unit tests for individual backend components, APIs, and database interactions using testing frameworks such as Mocha, Chai, or Jest. Test for expected behavior, edge cases, and error handling. Android Application: Conduct unit testing for specific modules or classes within the Android application using frameworks like JUnit or Espresso. Verify the correctness of functionality, handle different scenarios, and validate input/output.

Integration Testing
Backend and Database: Perform integration tests to ensure seamless interaction between the backend server and the database. Test API endpoints, data retrieval, storage, and validation. Android Application and Backend: Validate the integration between the Android application and the backend server. Test the communication via API calls, data synchronization, and handling of responses.

User Acceptance Testing:
Engage teachers, students, and administrators to participate in user acceptance testing. Provide them with a test environment and gather feedback on the system's usability, functionality, and user experience. Test various use cases, including attendance tracking, biometric authentication, user management, class scheduling, and report generation. Address any issues, bugs, or usability concerns identified by users and make necessary improvements.

Performance Testing:
Simulate high-load scenarios to test the system's performance and response times. Use tools like Apache JMeter to generate concurrent requests and measure the system's ability to handle them. Evaluate the performance of Bluetooth connectivity, biometric authentication processes, database operations, and API response times. Identify any bottlenecks or performance issues and optimize the system accordingly.

Security Testing:
Conduct security testing to identify vulnerabilities and ensure the system's security measures.Test for

potential security breaches, such as unauthorized access, data leakage, injection attacks, or session hijacking. Implement secure coding practices, encryption protocols, and user authentication mechanisms to mitigate security risks.

Compatibility Testing:
Test the compatibility of the Android application on different devices, screen sizes, and Android versions. Ensure that the application functions correctly on a variety of devices, including smartphones and tablets, with different resolutions and hardware specifications.

Regression Testing:
Perform regression testing after bug fixes, updates, or new feature implementations to ensure that the existing functionality has not been adversely affected.

7.6 DATA COLLECTION

A data collection of interrelated data stored with minimum redundancy to serve many users quickly and efficiently. The general objective is to make information access easy, quick, inexpensive, and flexible for the users. The general theme behind a database is to integrate all information. Database design is recognized as a standard of management information system and is available virtually for every computer system. In database design several specific objectives are considered:

Ease of learning and use, Controlled redundancy, Data independence, More information at low cost, Accuracy and integrity, Recovery from failure, Privacy and security, Performance.

There are 4 collections:

i) Collection_timings

Field	Type	Key
id	int	Primary Key
hour	int	
min	int	

ii) Collection_class

Field	Type	Key
id	int	Primary Key
name	Varchar(5 0)	

iii) Collection_teacher

Field	Type	Key
id	int	Primary Key
username	varchar(2 0)	
password	varchar(2 0)	
name	varchar(2 0)	
requests	array	foreign key

iv) Collection_student

Field	Type	Key
username	varchar(5 0)	
id	int	Primary Key
password	varchar(5 0)	
class	varchar(2 0)	Foreign key
name	varchar(2 0)	
status	varchar(2 0)	
bluetooth address	int	

CHAPTER 8

DESIGN GOAL

Most of the Bluetooth attendance marking System is inefficient. Many Bluetooth automated systems fall short in providing traceability, immutability, transparency, audit, and security features. They also fail to provide accurate reports. This research objective is to find out the solution by developing a Bluetooth automated attendance Marking System which overcomes all the disadvantages of the other Bluetooth Attendance Marking Systems. It is a user-friendly system. With the help of this system, the admin can easily view the activities of the users and can give approval and refusal to the proposals of the user. And every teacher can easily give a request to the admin and the student can easily find out the needful teacher. This is a Bluetooth Automated Attendance Marking System, that comes with live, better tracking and unique identification features. It has enhanced efficient, unique identification features for users and has a safe and secure database.

The Bluetooth Automated Attendance Marking System is designed with the following goals in mind:

Modularity: The folder is divided into several modules, each responsible for a specific functionality of the Bluetooth Automated Marking System. This modular design allows for easy maintenance, testing, and debugging of the system.

Scalability: The folder is designed to be scalable, allowing for the addition of new features and functionalities as needed. The modular design facilitates the addition of new modules without affecting the existing ones.

Security: The folder is designed with security in mind, ensuring that sensitive data is protected and secure. The system uses access controls to prevent unauthorized access to the data.

Usability: The folder is designed to be user-friendly, with a clear and intuitive interface. The system provides clear instructions and feedback to the user, making it easy to use even for those with limited technical expertise.

Performance: The folder is designed to ensure optimal performance of the system. The system uses efficient algorithms and data structures to ensure fast and responsive performance.

Maintainability: The folder is designed to be maintainable, with clear and concise code, and proper documentation. This ensures that the system can be easily maintained and updated as needed.

By adhering to these design goals, the Bluetooth Automated Attendance Marking System provides a robust, scalable, and user-friendly solution for managing blood banks, hospitals, donors, and patients.

8.1. MODULE DESIGN

Module Design for “TrackMate” Bluetooth Automated Attendance Marking System:

User Management Module:

This module handles user registration, login, and profile management functionalities. It allows teachers, students, and parents to create accounts, update their profiles, and manage their login credentials. It also includes features for password reset and user role assignment.

Attendance Tracking Module:

The Attendance Tracking module is responsible for tracking student attendance in real-time. It includes functionalities to detect Bluetooth signals from the beacons placed in the classrooms and record the presence of students. This module also manages attendance sessions, tracks late entries or early exits, and handles exceptions or special cases.

Biometric Authentication Module:

The Biometric Authentication module captures and verifies the biometric data of students during the attendance process. It integrates with biometric authentication devices (e.g., fingerprint scanners or facial recognition cameras) to capture and compare biometric information for authentication purposes. This module ensures the accuracy and security of student identification.

Data Storage and Retrieval Module:

The Data Storage and Retrieval module handles the storage and retrieval of attendance data, student profiles, and other relevant information. It interacts with the database or data storage system to store attendance records, student details, class schedules, and other related data. This module also supports efficient retrieval and querying of data for reporting and analysis purposes.

Reporting and Analytics Module:

The Reporting and Analytics module generates attendance reports and provides statistical analysis and insights. It allows teachers, students, and parents to view attendance reports for individual students, classes, or specified time periods. This module also includes features for visualizing attendance trends, identifying patterns, and generating actionable insights to improve attendance management.

Administration and Settings Module:

The Administration and Settings module provides administrative functionalities for system configuration and management. It allows administrators to manage user roles, set up classrooms and beacons, define attendance rules and exceptions, and customize system settings according to the institution's requirements.

Integration and API Module:

The Integration and API module facilitates integration with external systems or services. It includes APIs and interfaces for integrating with third-party biometric authentication services, messaging services, or other external systems. This module enables seamless communication and data exchange between TrackMate and other systems.

CHAPTER 9

RESULT AND DISCUSSION

The existing manual attendance tracking systems suffer from several limitations that hinder their effectiveness and reliability. Firstly, manual recording is prone to errors and manipulation, leading to inaccuracies in attendance data. Real-time data access is lacking, causing delays in addressing attendance issues. Proxy attendance is a significant concern, compromising the integrity of attendance records. Generating reports and analyzing data is inefficient, and security measures are often inadequate. Administrative burden, lack of student engagement, limited accessibility, and incompatibility with modern technology further contribute to the shortcomings of manual systems.

In contrast, the proposed Bluetooth Attendance Automated system offers a multitude of advantages that address these limitations comprehensively. Automation and efficiency streamline the attendance tracking process, saving time and reducing administrative workload. Real-time data access empowers stakeholders to make informed decisions promptly. The system ensures accurate and reliable attendance tracking through Bluetooth connectivity and biometric authentication, enhancing security and minimizing fraudulent activities.

A centralized database facilitates easy access, reporting, and analysis of attendance data. The user-friendly interface promotes engagement among teachers and students, fostering transparency and accountability. Data security and privacy are prioritized, and the system's scalability and compatibility make it adaptable to diverse educational environments. Future enhancements such as automated notifications and advanced analytics further augment the system's capabilities, promising continuous improvement and innovation in attendance management.

Output Design:

The output design of the TrackMate Bluetooth Attendance Tracker focuses on providing clear and informative outputs to teachers, students, and administrators. Here are the key aspects of the output design:

Attendance Reports:

Teacher: Teachers can generate attendance reports for individual classes or specific time periods. The reports include information such as student names, dates, and attendance status (present, absent, late). These reports can be viewed on the screen and also exported in various formats like PDF or Excel. Administrator: Administrators have access to comprehensive attendance reports across all classes. They can generate reports based on different criteria, such as class-wise attendance summaries or individual student attendance records.

Attendance Summaries:

Teacher: Teachers can view attendance summaries for each class, displaying the overall attendance percentage, total present/absent counts, and any exceptional entries (e.g., late arrivals or early departures). These summaries help teachers track attendance trends and identify patterns. Student: Students can view

their attendance summaries, showing the overall attendance percentage, total present/absent counts, and any exceptional entries. This allows students to monitor their attendance and take necessary actions if required.

Class Schedules:

Teacher: Teachers can access and view class schedules, which include details such as class timings, room numbers, and dates. This helps teachers manage their classes effectively and ensure proper attendance tracking. Student: Students can access their class schedules to know when and where their classes are scheduled. This allows them to plan their attendance and be prepared for each class.

Notifications and Alerts:

Teacher: Teachers receive notifications and alerts regarding attendance updates, such as absentees, late arrivals, or exceptions. These notifications help teachers stay informed and take appropriate actions if needed. Administrator: Administrators receive notifications and alerts for important system updates, pending requests (e.g., student enrollment or teacher requests), or any critical issues related to attendance management.

Student Requests:

Teacher: Teachers receive notifications and requests from students who want to join their classes. Teachers can review these requests and accept or reject them accordingly, managing their class memberships. Administrator: Administrators handle student requests for enrollment, verifying and approving or rejecting them based on eligibility criteria.

The output design aims to provide relevant and actionable information to users, helping them effectively manage attendance, track student records, and make informed decisions. The outputs are presented in a clear and easily understandable format, facilitating efficient communication and enhancing the overall user experience.

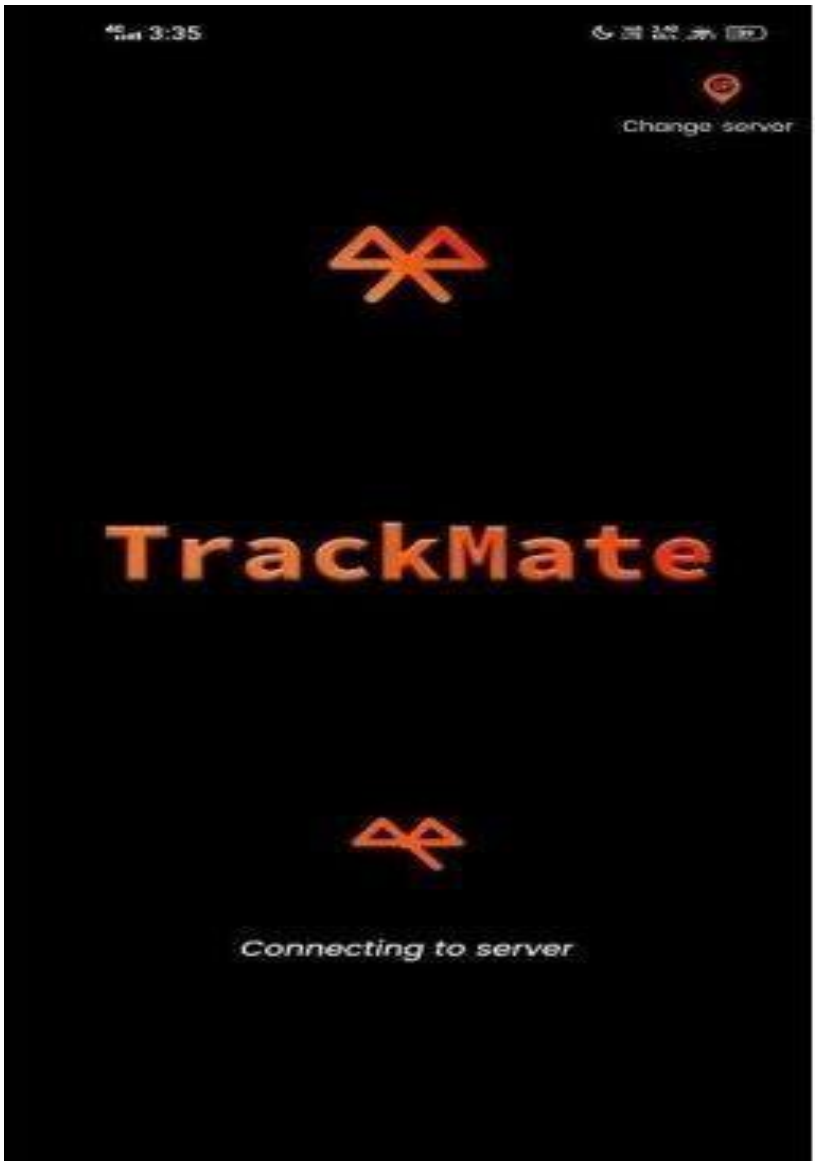


Fig 9.1 Home page

Fig 9.2 Login Page

Fig 9.3 Student Home Page

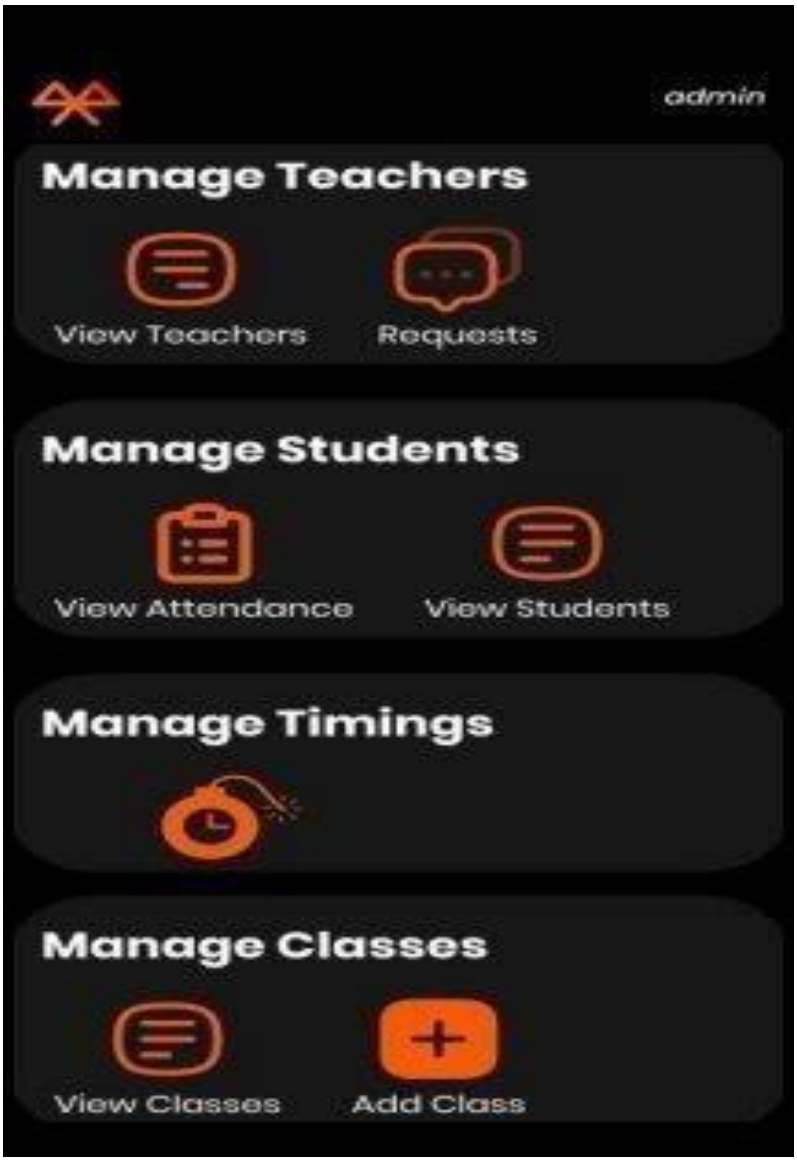


Fig 9.4 Admin Home Page

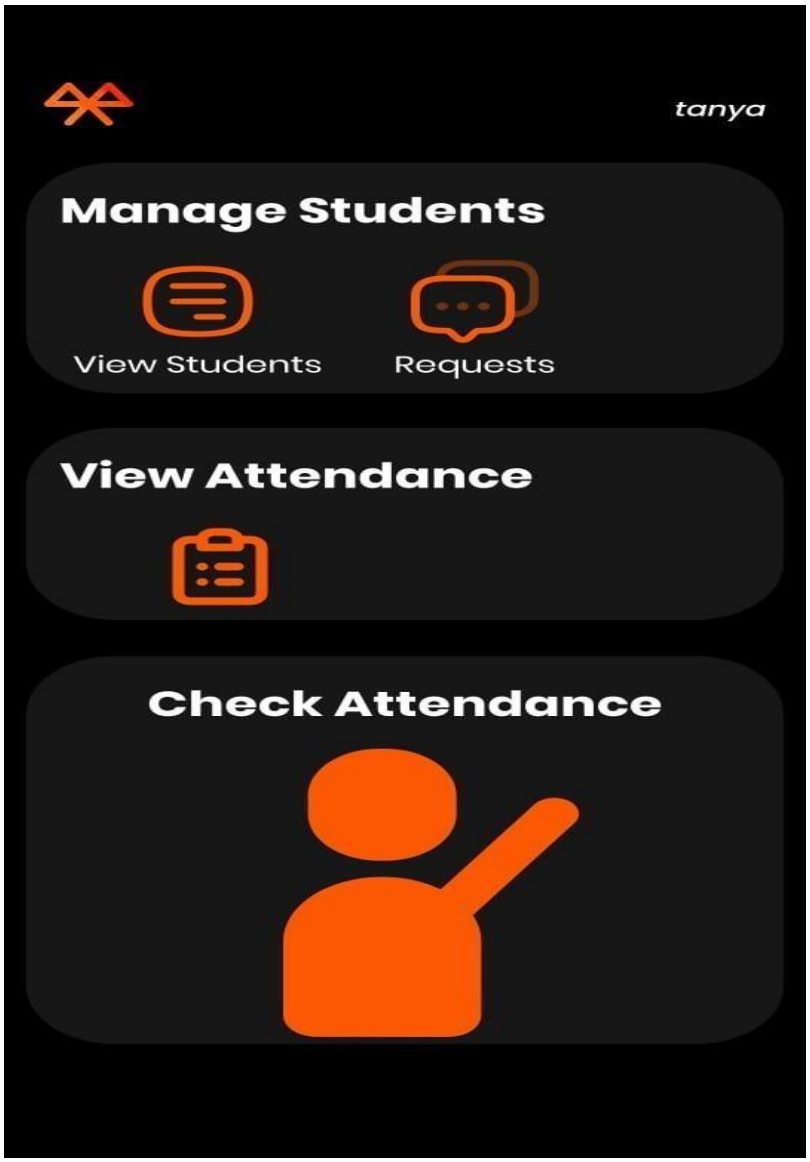


Fig 9.5 Teacher Home Page

CHAPTER 10

CONCLUSION AND FUTURE SCOPE

In conclusion, the Bluetooth Automated Attendance Marking System project not only revolutionizes attendance tracking in schools but also lays a robust foundation for future advancements and enhancements. By integrating Bluetooth and biometric authentication, the system not only offers convenience and accuracy but also provides distinct interfaces tailored for teachers, students, and administrators. Developed using Agile methodologies and leveraging Express.js and Android Studio, the project ensures both technical feasibility and economic viability, making it a sustainable solution for educational institutions.

Looking ahead, the project's future enhancements hold great potential. Real-time analytics can unveil attendance patterns and student engagement levels, empowering decision-makers with valuable insights. Incorporating machine learning algorithms can further optimize scheduling by predicting attendance trends. Geolocation-based attendance tracking within school premises adds another layer of accuracy and convenience.

Biometric upgrades such as facial recognition or fingerprint scanning enhance security and streamline attendance verification. Improving user interfaces ensures a seamless experience, while communication integration through automated notifications keeps stakeholders informed about attendance and schedules. Customization options allow schools to tailor policies, reports, and system integrations to their specific needs.

Enabling an offline mode ensures continuity of operations even without internet connectivity, with data synchronization capabilities for seamless updates. Continuous security updates are prioritized to safeguard student information and maintain system integrity. In conclusion, the TrackMate Bluetooth Attendance Tracker project not only meets current needs but also paves the way for ongoing innovation and improvement in attendance management within educational environments.

REFERENCES

[1] Rudra Malai, Naman Jangid, Pranjali Satish Deshmukh, Halgaon Prasad S “Bluetooth Automated Attendance Management using Android Application” , IJITEE June 2019.

[2] Sangam Dange “NFC technology using system Bluetooth-based Attendance Management App” by (International Journal of Innovative Science and Research Technology ISSN No: -2456-2165, IJEECS,2021).

[3] Rayyan Banjar, Hassan Samkari and Anas Basalamah “A Bluetooth Based Smartphone Attendance System” , Computer Engineering Department, Umm AI Qura University.

[4] Arulogun O.T,Olantunbosun,A.Fakolujo O.A & Olaniyi,O.M “RFID-Based Students Attendance Management System” .

[5] V. Rajesh, Ramu, K. Harshitha, CH.V. Sai Teja “Automatic Attendance Management system using BLE technology” [ANIL NEERUKONDA INSTITUTE OF TECHNOLOGY AND SCIENCES],2019.

APPENDIX

ADMIN

```
const username = "admin";
const password = "12345";
const express = require('express');
const router = express. Router();
const fs = require('fs');
const path = require('path');
const app = require('../app.js');

function server(req, res, func) {
    try {
        func(req, res);
    }
    catch (err) {
        console.log(err);
        res.status(500).send();
    }
}
router.get('/', (req, res) => {
    res.send('heyyy');
});
router.post('/auth', (req, res) => {
    server(req, res, (req, res) => {
        if (req.body.username == username && req.body.password == password) {
            console.log('admin authenticated');
            res.status(200).send();
        }
        else {
            console.log('invalid admin credentials');
            res.status(401).send();
        }
    });
});

router.get('/requests', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile('teachers_requests.json');
        console.log('sending teacher requests');
        res.send(data);
    });
```

```
});
router.post('/respond', (req, res) => {
    server(req, res, (req, res) => {
        let data = app.readFile('teachers_requests.json');
        data['requests'] = data['requests'].filter((teacher) => {
            return teacher.username != req.body.teacher.username;
        });
        app.writeFile('teachers_requests.json', data);
        if (req.body.accept) {
            data = app.readFile('teachers.json');
            req.body.teacher.requests=[];
            data['teachers'] [req.body.teacher.username] = req.body.teacher;
            app.writeFile('teachers.json', data);
            console.log('Added new teacher');
        }
        else {
            console.log('Declined request');
        }
        res.send();
    });
});

router.get('/teacher/list', (req, res) => {
    server(req, res, (req, res) => {
        res.send(app.readFile('teachers.json'));
    });
});
router.post('/teacher/remove', (req, res) => {
    server(req, res, (req, res) => {
        let data = app.readFile('teachers.json');
        if (data['teachers'] [req.body.username]) {
            delete data['teachers'] [req.body.username];
        }
        else {
            res.status(403);
        }
        app.writeFile('teachers.json', data);
        res.send();
        console.log('teacher deleted');
    });
});

router.post('/class/add', (req, res) => {
    server(req, res, (req, res) => {
        const path = `./data/classes/${req.body.class}.json`;
        fs.access(path, fs.constants.F_OK, (err) => {
            if (err) {
```

```
        fs.writeFileSync(path, JSON.stringify({"students":{},"attendance":{}}));
```

```
                console.log('new class added');
                res.send();
            }
            else {
                console.log('class already exist');
                res.status(403).send();
            }
        });
    });
});

router.post('/class/remove', (req, res) => {
    server(req, res, (req, res) => {
        const path = `./data/classes/${req.body.class}.json`;
        fs.unlink(path, (err) => {
            if (err) {
                console.log(err);
                res.status(403).send();
            } else {
                console.log('Class deleted successfully');
                res.send();
            }
        });
    });
});

router.post('/class/view', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile(`classes/${req.body.class}.json`);
        res.send(data['students']);
        console.log('viewing class');
    });
});
router.get('/class/list', (req, res) => {
    server(req, res, (req, res) => {
        fs.readdir("./data/classes", (err, files) => {
            if (err) {
                console.log('Error reading classes', err);
                res.status(500).send();
            } else {
                const classes = files.map((file) => {
                    return path.parse(file).name;

          });


                }
            });
```

```
});
console.log('classesare', classesses);
res.send({
            "classesses": classes
            }
            );
```

```
});
router.post('/student/remove', (req, res) => {
    server(req, res, (req, res) => {
        let data = app.readFile(`classes/${req.body.class}.json`);
        if (data['students'][req.body.username]) {
            delete data['students'][req.body.username];
        }
        else {
            res.status(403);
        }
        app.writeFile(`classes/${req.body.class}.json`, data);
        res.send();
        console.log('student deleted');
    });
});

router.post('/attendance', (req, res) => {
    server(req, res, (req, res) => {
        res.send(app.readFile(`classes/${req.body.class}.json`)["attendance"]);
        console.log('attendance returned');
    });
});
router.get('/timings', (req, res) => {
    server(req, res, (req, res) => {
        console.log('sending timings')
        res.send(app.readFile('timings.json'));
    });
});
router.post('/timings', (req, res) => {
    server(req, res, (req, res) => {
        console.log('updating timings')
        app.writeFile('timings.json', req.body);
        res.send();
    });
});

module.exports = router;
```

TEACHER

```
const express = require('express');
const router = express.Router();
const fs = require('fs');
const path = require('path');
const app = require('../app.js');

function server(req, res, func) {
    try {
        func(req, res);
    }
    catch (err) {
        console.log(err);
        res.status(500).send();
    }
}
router.get('/', (req, res) => {
    res.send('heyyy');
});
router.post('/auth', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile('teachers.json');
        if (data['teachers'][req.body.username]['password'] == req.body.password) {
            console.log('teacher authenticated');
            res.status(200);
        }
        else {
            res.status(401);
            console.log('invalid teacher credentials');
        }
        res.send();
    });
});

router.post('/new', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile('teachers.json');
        if (data['teachers'][req.body.username]) {
            console.log('username exists');
            res.status(401);
        }
        else {
```

```
            console.log('new teacher');
            let requests = app.readFile('teachers_requests.json');
            requests['requests'].push(req.body);
            app.writeFile('teachers_requests.json', requests);
            res.status(200);
        }
        res.send();
    });
});

router.get('/class/list', (req, res) => {
    server(req, res, (req, res) => {
        fs.readdir("./data/classes", (err, files) => {
            if (err) {
                console.log('Error reading classes', err);
                res.status(500).send();
            } else {
                const classes = files.map((file) => {
                    return path.parse(file).name;
                });
                console.log('classes are ', classes);
                res.send({ "classes": classes });
            }
        });
    });
});

router.post('/class/view', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile(`classes/${req.body.class}.json`);
        res.send(data['students']);
        console.log('viewing class');
    });
});

router.post('/requests', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile('teachers.json');
        const requests = data['teachers'][req.body.username]['requests'];
        console.log('sending students requests');
        res.send({ "requests": requests });
    });
});
```

```
router.post('/respond', (req, res) => {
```

```
    server(req, res, (req, res) => {
        let data = app.readFile('teachers.json');
        const student = req.body.student;
        data['teachers'][req.body.username]['requests']                                    =
data['teachers'][req.body.username]['requests'].filter((student) => {
            return student.username != req.body.student.username;
        });
        app.writeFile('teachers.json', data);
        if (req.body.accept) {
            data = app.readFile(`classes/${student.class}.json`);
            data['students'][student.username] = student;
            app.writeFile(`classes/${student.class}.json`, data);
            console.log('Added new student');
        }
        else {
            console.log('Declined student request');
        }
        res.send();
    });

});
router.post('/attendance', (req, res) => {
    server(req, res, (req, res) => {
        let data = app.readFile(`classes/${req.body.class}.json`);
        for (student in req.body.attendance) {
            if (!data['attendance'][student]) {
                data['attendance'][student] = [];
            }
            data['attendance'][student].push(req.body.attendance[student]);
        }
        app.writeFile(`classes/${req.body.class}.json`, data);
        console.log('Attendance marked successfully');
    res.send();
    });
});


router.get('/attendance', (req, res) => {
    server(req, res, (req, res) => {
        res.send(app.readFile(`classes/${req.body.class}.json`)["attendance"]);
        console.log('attendance returned');
    });
});
module.exports = router;
```

STUDENT

```
const express = require('express');
const router = express.Router();
const path = require('path');
const fs = require('fs');
const app = require('../app.js');

function server(req, res, func) {
    try {
        func(req, res);
    }
    catch (err) {
        console.log(err);
        res.status(500).send();
    }
}
router.get('/', (req, res) => {
    res.send('heyyy');
});
router.get('/class/list', (req, res) => {
    server(req, res, (req, res) => {
        fs.readdir("./data/classes", (err, files) => {
            if (err) {
                console.log('Error reading classes', err);
                res.status(500).send();
            } else {
                const classes = files.map((file) => {
                    return path.parse(file).name;
                });
                console.log('classes are ', classes);
                res.send({ "classes": classes });
            }
        });
    });
});

router.get('/teacher/list', (req, res) => {
    server(req, res, (req, res) => {
        res.send(app.readFile('teachers.json'));
    });
});
router.post('/auth', (req, res) => {
```

```
    server(req, res, (req, res) => {
        let data = app.readFile(`classes/${req.body.class}.json`);
        if (data['students'][req.body.username]['password'] == req.body.password) {
            data['students'][req.body.username]['id'] = req.body.id;
            app.writeFile(`classes/${req.body.class}.json`, data);
            console.log('successful');
            res.status(200);
        }
        else {
            res.status(401);
            console.log('oops');
        }
        res.send();
    });
});

router.post('/new', (req, res) => {
    server(req, res, (req, res) => {
        const data = app.readFile(`classes/${req.body.class}.json`);
        if (data[req.body.username]) {
            console.log('username exists');
            res.status(401);
        }
        else {
            console.log('new student');
            let data = app.readFile('teachers.json');
            data['teachers'][req.body.teacher]['requests'].push(req.body);
            app.writeFile('teachers.json', data);
            res.status(200);
        }
        res.send();
    });
});

router.get('/timings', (req, res) => {
    server(req, res, (req, res) => {
        console.log('sending timings')
        res.send(app.readFile('timings.json'));
    });
});
router.post('/status', (req, res) => {
    server(req, res, (req, res) => {
        let data = app.readFile(`classes/${req.body.class}.json`);
        data['students'][req.body.username]['status'] = req.body.status;
```

```
app.writeFile(`classes/${req.body.class}.json`, data);
```

```
        res.send();
    });
});

router.post('/attendance', (req, res) => {
    server(req, res, (req, res) => {
        res.send(app.readFile(`classes/${req.body.class}.json`)["attendance"]);
        console.log('attendance returned');
    });
});

module.exports = router;
```