

NexTrack: Next-Gen Attendance Monitoring with ML

Submitted in partial fulfillment of the requirements for the award of
Bachelor of Engineering Degree in Computer Science and Engineering

By

Nayan Asawa (Reg. No - 40110845)

Nitesh Kumar Singh (Reg. No - 40110862)



**DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF COMPUTING**

SATHYABAMA

**INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)**

CATEGORY -1 UNIVERSITY BY UGC

**Accredited with Grade "A++" by NAAC | 12B Status by UGC | Approved by AICTE
JEPPIAAR NAGAR, RAJIV GANDHI SALAI, CHENNAI - 600 119**

APRIL - 2024



SATHYABAMA

INSTITUTE OF SCIENCE AND TECHNOLOGY

(DEEMED TO BE UNIVERSITY)

CATEGORY-I UNIVERSITY BY UGC

Accredited "A++" Grade by NAAC | 12B Status by UGC | Approved by AICTE

www.sathyabama.ac.in

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this Project Report is the bonafide work of **Nayan Asawa** (Reg. No - 40110845) and **Nitesh Kumar Singh** (Reg. No - 40110862) who carried out the Project entitled "**NexTrack: Next-Gen Attendance Monitoring with ML**" under my supervision from November 2023 to April 2024.

Internal Guide

Dr. P. Ajitha, M.E., Ph.D.

SATHYABAMA
INSTITUTE OF SCIENCE AND TECHNOLOGY
(DEEMED TO BE UNIVERSITY)
Jeppiyarthi Nagar, Jeppiaur South Salem,
Chennai - 600 079

Head of the Department

Dr. L. Lakshmanan, M.E., Ph.D.

Submitted for Viva voce Examination held on 16/04/2024

Internal Examiner

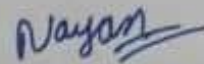
External Examiner

DECLARATION

I, Nayan Asawa (Reg. No - 40110845) hereby declare that the Project report entitled **NexTrack: Next-Gen Attendance Monitoring with ML** done by me under the guidance of **Dr. P. Ajitha BE., M.E., Ph.D.** is submitted in partial fulfilment of the requirement for the award of Bachelor of Engineering Degree in Computer Science and Engineering.

DATE : 16/04/2024

PLACE : Chennai



SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

I am pleased to acknowledge my sincere thanks to **Board of Management of Sathyabama Institute of Science and Technology** for their kind encouragement in doing this project and for completing it successfully. I am grateful to them.

I convey my thanks to **Dr. T. Sasikala M.E. , Ph.D. , Dean, School of Computing , Dr. L. Lakshmanan, M.E. , Ph.D. , Head of the Department of Computer Science and Engineering** for providing me necessary support and details at the right time during the progressive reviews.

I would like to express my sincere and deep sense of gratitude to my Project Guide **Dr. P. Ajitha BE., M.E., Ph.D.** for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

I wish to express my thanks to all Teaching and Non-teaching staff members of the **Department of Computer Science and Engineering** who were helpful in many ways for the completion of the project.

ABSTRACT

The objective of this project is to have overall view in Flutter Framework and Dart programming language which can be used to implement the NexTrack: Next-Gen Attendance Monitoring with ML. This system can help organizations track and monitor employee attendance, working hours, leave applications, and generate payroll reports.

Employee attendance systems have become an essential tool for organizations of all sizes to manage their workforce and track employee attendance. Traditional manual attendance systems can be time-consuming and prone to errors, leading to inaccurate records and payroll issues. Digital employee attendance systems offer a more efficient and accurate solution that eliminates the need for manual processes.

Digital employee attendance systems use various technologies, such as biometrics, RFID, and GPS, to record and track employee attendance data. They offer several benefits, including increased accuracy, reduced administrative burden, improved compliance, and greater visibility into employee attendance patterns. However, they also have some downsides, such as technical issues, security risks, and initial investment costs.

This NexTrack provides an overview of employee attendance systems, their benefits and downsides, and the different technologies used in these systems. It also discusses the feasibility study of implementing a digital employee attendance system and the steps involved in implementing such a system. Finally, it highlights some of the best practices for managing employee attendance and offers insights into how organizations can leverage these systems to improve their operational efficiency and better manage their workforce.

TABLE OF CONTENTS

Chapter No	TITLE	Page No.
	ABSTRACT	v
	LIST OF FIGURES	viii
1	INTRODUCTION	1
	1.1 Introduction to NexTrack	1
	1.2 Objectives of NexTrack	1
2	LITERATURE SURVEY	
	2.1 Inferences from Literature Survey	2
	2.2 Open Problems in Existing System	4
3	REQUIREMENT ANALYSIS	
	3.1 Feasibility Studies of the Project	7
	3.2 Software Requirements Specification Document	9
4	DESCRIPTION OF PROPOSED SYSTEM	
	4.1 Selected Methodology	12
	4.2 Algorithms Used	13
	4.2.1 CNN Algorithm	13
	4.2.2 Geolocation Algorithm	15
	4.3 Architecture of Proposed System	17
	4.4 Implementation and Testing plan of the Proposed Model	20
	4.5 Project Management Plan	23
	4.6 Financial report on estimated costing	24
	4.7 Transition to Operations Plan	24
5	IMPLEMENTATION DETAILS	
	5.1 Development and Deployment Setup	26
	5.2 Testing Plan	27
6	RESULT AND DISCUSSIONS	
	6.1 Module Performance	29

7	CONCLUSION	
7.1	CONCLUSION	32
7.2	FUTURE WORK	33
7.3	RESEARCH ISSUES	34
7.4	IMPLEMENTATION ISSUES	35
	REFERENCES	37
	APPENDIX	
	A. SOURCE CODE	38
	B. SCREENSHOTS	50
	C. RESEARCH PAPER	51

LIST OF FIGURES

FIGURE NO	FIGURE NAME	Page No.
4.1	NexTrack Architecture	11
4.2	NexTrack Process Flow	12
4.3	CNN Architecture Diagram	13
4.4	Convolution and Max Pooling	14
4.5	CNN Block Diagram	15
4.6	Geolocation Algorithm	16
4.7	Geolocation Architecture Diagram	17
4.8	NexTrack Working	19
6.1	CNN Model Accuracy	30
6.2	CNN Model Loss	30
6.3	Registration Screen	31
6.4	Login Screen	31

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION TO NEXTRACK

Employee attendance system is a software tool used by companies to track and monitor their employees' attendance, working hours, leave applications, and generate payroll reports. The system is designed to streamline attendance management, reduce errors, and ensure compliance with labour laws and regulations. The system generates reports on attendance-related data, which can help organizations to analyse workforce trends, identify areas for improvement, and make informed decisions about workforce management.

In conclusion, the employee attendance system is a vital tool for organizations to track and monitor their employees' attendance, working hours, and leave applications. By achieving its objectives, the system can help organizations to improve their productivity, reduce costs, and ensure compliance with labour laws and regulations.

1.2 OBJECTIVES OF NEXTRACK

The objectives of the employee attendance system include:

1. Accurate and efficient tracking of employee attendance.
2. Improved workforce management.
3. Reduction of payroll errors.
4. Ensuring compliance with labour laws and regulations.
5. Streamlining leave management processes.
6. Facilitating real-time monitoring of employee attendance.
7. Generating reports on attendance-related data for analysis and decision-making.

By achieving these objectives, the employee attendance system can help organizations to improve their productivity, reduce costs, and ensure compliance with labour laws and regulations.

CHAPTER 2

LITERATURE SURVEY

2.1 INFERENCES FROM LITERATURE SURVEY

Smart Attendance System using RFID

RFID-based attendance systems are described as a more reliable option due to their use of living biometrics, which reduces the likelihood of cheating or fraudulent entries. RFID systems operate within a frequency band of 902 MHz to 928 MHz and utilize backscatter radio communication to interact with passive RFID tags.

One of the key advantages of RFID technology is its ability to extract valuable Radio-Frequency (RF) characteristics from the signals, such as the Received Signal Strength Indicator (RSSI) and phase characteristics. While RSSI reflects signal power and has been widely used in indoor localization systems, the phase characteristic is highlighted as more stable and resistant to interference, making it preferable for attendance tracking.

This system utilizes a commodity RFID reader and common-off-the-shelf (COTS) tags to recognize individuals based on their unique phase readings. To enhance accuracy, the authors employ statistical frequency distribution histograms to reduce the impact of signal fluctuations. Multiple evaluation indicators are extracted to distinguish between individuals, and the K-means clustering method is used for intelligent attendance management.

The proposal of a device-free office attendance system based on RFID, the use of multiple evaluation indicators from phase profiles, and the presentation of a prototype system that achieves an impressive average accuracy of 92%, even in the presence of variations in clothing and time of day.

It showcases the potential of RFID-based systems to overcome the limitations of traditional methods, offering increased accuracy while maintaining ease of use.

Face Recognition based Attendance Marking

Face recognition technology is in attendance tracking systems. Traditional methods like fingerprint or card-based attendance systems have their limitations, including error rates and inefficiencies. For example, fingerprint systems can have error rates of around 5%, leading to inaccuracies in attendance records, particularly in large attendance sites where congestion can occur. Card-based systems are also susceptible to misuse, as employees can swipe cards for someone else.

Face recognition technology offers a compelling solution to these issues. It boasts higher accuracy and stability, primarily because it relies on multiple data points derived from an individual's facial features. This approach makes it more resilient to variations in clothing, lighting conditions, and facial expressions, ensuring accurate attendance records.

The technology finds applications beyond attendance tracking, spanning various sectors such as public safety, civil economy, and home entertainment. In public safety, it is used for surveillance, criminal identification, and border security. In civil economy, it aids in access control systems, identity verification for financial transactions, and customer experience enhancements. In home entertainment, it enables personalized content recommendations and user authentication for smart devices.

Face recognition includes geometric feature methods, subspace analysis, neural network methods, and support vector machine (SVM) methods. These methods leverage various techniques to extract, process, and match facial features to achieve accurate recognition.

Face recognition technology has emerged as a transformative force, revolutionizing attendance tracking systems and finding applications in diverse sectors. Its ability to provide accurate and efficient identity verification has led to its widespread adoption and continuous development, making it a central technology in the era of AI and computer vision.

2.2 OPEN PROBLEMS IN EXISTING SYSTEM

Manual Attendance Tracking:

Time-Consuming: Manual attendance tracking is a time-consuming process, especially in large organizations with many employees. It involves collecting and recording attendance data for each employee individually, which can be a significant administrative burden.

Error-Prone: Since it relies on manual data entry, it is highly susceptible to human errors. Employees may forget to sign in or out, leading to inaccurate attendance records.

Lack of Real-Time Tracking: Manual tracking cannot provide real-time attendance data. This delay in data entry may lead to issues in monitoring and managing employee attendance effectively.

Inefficiency: The process of collecting and managing paper-based attendance records can be inefficient and may require significant storage space.

Paper-Based Systems:

Increased Error Rates: Paper-based systems exacerbate the issues of manual attendance tracking. In addition to the errors introduced during manual data entry, paper records can be easily lost, damaged, or altered, leading to even greater inaccuracies.

Illegible Handwriting: Illegible handwriting can make it challenging to decipher attendance records accurately, further contributing to data errors.

Data Accessibility: Retrieving historical attendance data from paper records can be cumbersome and time-consuming.

Biometric-Based Systems:

Cost: Implementing biometric-based attendance systems can be expensive, involving the purchase of biometric devices and software. Small businesses may find it cost-prohibitive.

Privacy Concerns: Biometric data collection, such as fingerprint or facial recognition, can raise privacy concerns among employees. Organizations must address these concerns and ensure compliance with privacy regulations.

Technical Issues: Biometric systems may encounter technical issues, such as false positives or negatives in recognition, which can affect attendance accuracy.

Integration Challenges: Integrating biometric systems with existing HR or payroll software can be complex and may require additional resources and expertise.

Overall Limitations:

Scalability: Many existing systems may struggle to scale effectively with the growth of an organization. As the number of employees increases, managing attendance data becomes increasingly challenging.

Data Security: All attendance systems, including biometric ones, must prioritize data security to prevent unauthorized access or data breaches.

Compliance: Ensuring compliance with labor laws and regulations, especially regarding overtime, break times, and leave policies, can be a complex task for all types of attendance systems.

CHAPTER 3

REQUIREMENT ANALYSIS

The primary aim of NexTrack is to accurately track, monitor, and manage the attendance of employees within an organization. By automating attendance-related processes and providing real-time insights into employee attendance patterns, the NexTrack aims to streamline administrative tasks, improve operational efficiency, and ensure compliance with labour laws and regulations.

The scope of the NexTrack project encompasses a comprehensive set of functionalities, objectives, and deliverables aimed at addressing the diverse needs of organizations in managing employee attendance, optimizing workforce management processes, and ensuring compliance with labour laws and regulations.

Firstly, the scope includes the development of a user-friendly and intuitive interface that enables employees to efficiently record their attendance, request leaves, and access relevant information regarding their attendance history and leave balances. This interface should be accessible across multiple platforms, including web and mobile devices, to accommodate the diverse needs and preferences of employees.

Secondly, the NexTrack will incorporate advanced features for accurate attendance tracking, including biometric authentication, RFID technology, or mobile check-in/check-out capabilities. These features will ensure the reliability and accuracy of attendance data while minimizing the risk of fraudulent practices or errors associated with manual timekeeping methods.

Furthermore, the NexTrack will encompass robust reporting and analytics capabilities, enabling organizations to gain valuable insights into workforce trends, attendance patterns, and compliance metrics. Through customizable reports and dashboards, stakeholders will be able to monitor key performance indicators, identify areas for improvement, and make data-driven decisions to optimize workforce management strategies.

3.1 FEASIBILITY STUDIES OF THE PROJECT

A feasibility study is an assessment of the practicality and potential success of a proposed project. In the case of an employee attendance system, a feasibility study would evaluate whether the system is viable and worthwhile for the organization to implement. Here are some factors that should be considered in a feasibility study for an employee attendance system:

1. Cost: The cost of implementing an employee attendance system can vary widely depending on the type of system chosen, the number of employees to be tracked, and any customization required. A feasibility study should evaluate whether the cost of implementing the system is within the organization's budget and whether the benefits of the system outweigh the cost.

2. Technical feasibility: The technical feasibility of this system involves evaluating whether the necessary technology infrastructure is in place to support the system. This includes hardware, software, and networking components. A feasibility study should assess whether the organization has the necessary technical resources and expertise to implement and maintain the system.

3. Legal and regulatory compliance: A feasibility study should assess whether this system complies with relevant labour laws and regulations. This includes ensuring that the system accurately tracks employee work hours, overtime, and other types of leaves, as well as ensuring that the system complies with data privacy laws and regulations.

4. User acceptance: A feasibility study should assess whether employees and managers are willing and able to use this system. This includes evaluating whether the system is easy to use and whether employees and managers are receptive to the idea of using such a system.

5. Benefits: A feasibility study should evaluate the potential benefits of implementing this system. This includes improving productivity, reducing errors in payroll processing, and ensuring compliance with labour laws and regulations.

Based on the results of a feasibility study, the organization can make an informed decision about whether to proceed with the implementation of an employee attendance

system. If the study shows that the system is feasible and likely to provide benefits that outweigh the cost, the organization can move forward with the implementation.

Risk Analysis:

Security Risks:

Data Breach: Unauthorized access to sensitive employee data or security breaches could compromise user privacy and the system's integrity.

Face Recognition Accuracy: Inaccurate face recognition could lead to unauthorized attendance marking or access.

Technical Risks:

Integration Challenges: Integrating multiple technologies like geolocation, facial recognition, and real-time data synchronization can be complex and may lead to technical issues.

Scalability: As the user base grows, the system must handle increased data volume and concurrent users without performance degradation.

Compliance Risks:

Data Privacy Regulations: Non-compliance with data privacy regulations (e.g., GDPR, HIPAA) could result in legal and financial consequences.

User Adoption:

Training and Acceptance: Employees may resist or struggle with adopting the new attendance system, requiring effective training and change management strategies.

Performance Risks:

Real-time Updates: Ensuring real-time data updates and synchronization across devices without delays or errors is crucial for system performance.

Maintenance and Support:

System Downtime: Maintenance or technical issues leading to system downtime could disrupt attendance tracking and negatively impact users.

3.2 SOFTWARE REQUIREMENTS

1. Introduction :

Employee attendance system is a software tool used by companies to track and monitor their employees' attendance, working hours, leave applications.

The system is designed to streamline attendance management, reduce errors, and ensure compliance with labor laws and regulations.

2. Functional Requirements:

NexTrack must provide user authentication, facial recognition, geolocation tracking, real-time data synchronization, and attendance record calculation. It should offer an intuitive UI for attendance marking, admin dashboard for user management, and notification features. The system must generate color-coded attendance reports and warnings for obesity.

3. Non-Functional Requirements:

The application needs robust security with encryption, secure authentication, and data privacy compliance. It should ensure real-time data synchronization, scalability, and optimal performance. NexTrack must be user-friendly, provide training resources, and offer efficient user support. Additionally, it must undergo rigorous testing, feature backup and recovery mechanisms, and be deployable on major app stores and reliable hosting infrastructure.

4. System Architecture:

NexTrack employs a client-server architecture. Clients (mobile apps) communicate with the backend server (Node.js) to access the Firebase Realtime Database, Firebase ML Kit for facial recognition, and Firebase Geolocation services. The system ensures real-time data synchronization and security through encryption and authentication.

5. Data Requirements:

NexTrack stores user profiles, attendance records, and related data in the Firebase Realtime Database. This data includes email IDs, geolocation coordinates, check-in/check-out times, and facial recognition information. Data integrity, security, and privacy are paramount, with compliance with applicable data protection regulations, such as GDPR or HIPAA, being essential.

Operating System Compatibility:

Mobile App (Employee): Compatibility with Android and iOS platforms.

Backend Server: Compatibility with popular server operating systems like cloud-based solutions.

Development Frameworks and Tools:

Mobile App Development: Flutter for cross-platform mobile app development.

Database: Firebase Realtime Database for data storage.

Facial Recognition: Firebase ML Kit for facial recognition.

Geolocation: Firebase Geolocation services.

Authentication: Firebase Authentication for user login.

Database Requirements:

Realtime Database Schema: Design and implementation of a database schema to store user profiles, attendance records, and related data.

CHAPTER 4

DESCRIPTION OF PROPOSED SYSTEM

NexTrack's system design features a client-server architecture. Mobile apps (clients) interact with the Node.js-based backend server, which connects to Firebase services (Realtime Database, ML Kit, Geolocation) for data storage, face recognition, and geolocation tracking. This design ensures real-time data synchronization, robust security, and scalable performance.

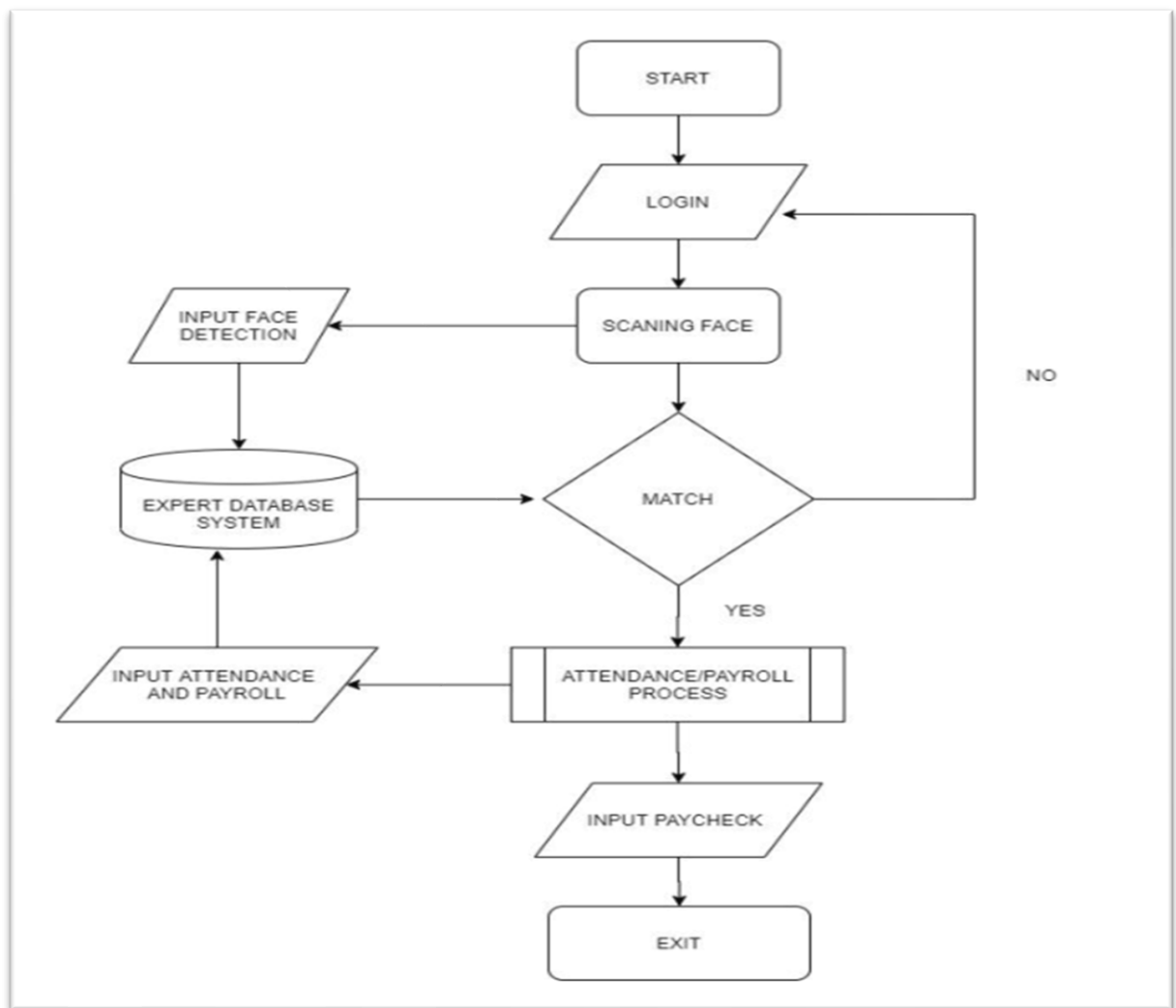


Fig:4.1 : NexTrack Architecture

4.1 SELECTED METHODOLOGY

1. Authentication : Users log in with their email IDs and passwords, ensuring secure access to the system.

2. Face Recognition: The app verifies users' identities through facial recognition, enhance security.

3. Geolocation Tracking: NexTrack captures users' precise locations in real-time uses geolocation technology.

4. Check-in/Check-out: Employees mark attendance by entering their check-in and check-out times.

5. Automated Processing: The system calculates time intervals between check-in and check-out, creating automated attendance records.

6. Color-Coded Display: Attendance data is presented in a color-coded format, providing quick insights into attendance status, facilitating efficient management.

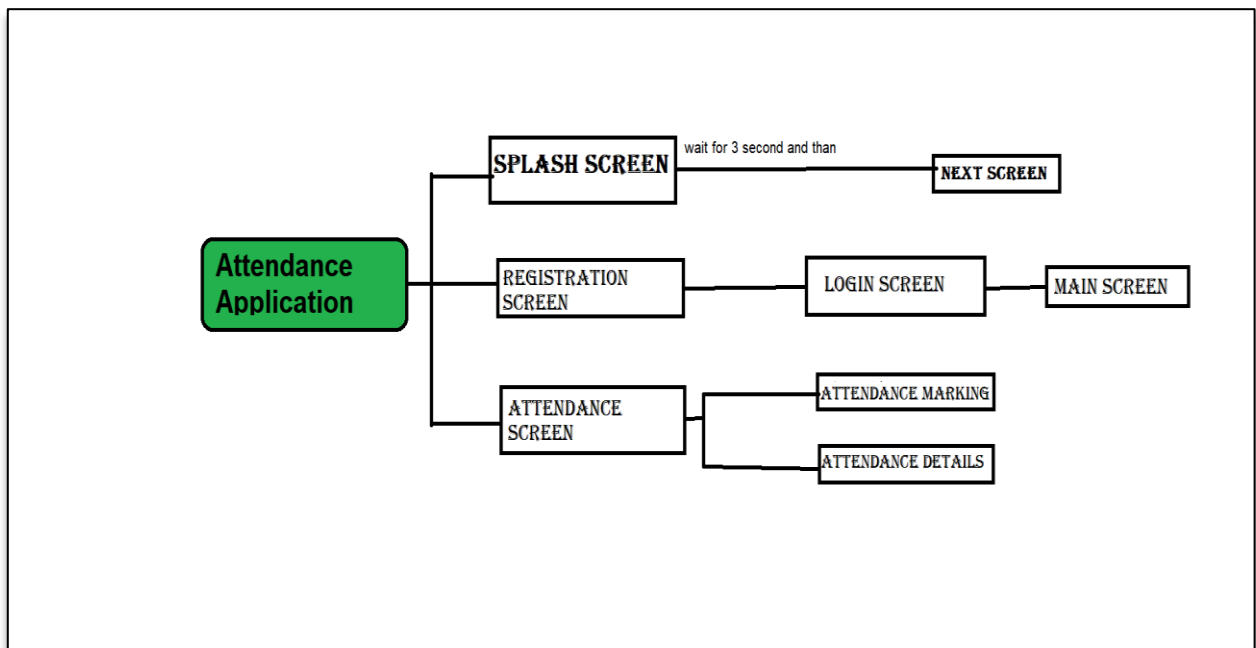


Fig:4.2 : NexTrack Process Flow

4.2 ALGORITHMS USED

The two main algorithms which are used to develop this application is “Geolocation Algorithm” and “CNN Algorithm. Geolocation Algorithm is used for getting the coordinate of the user and decide whether the user can get forward inside the application or not. CNN Algorithm is used for the implementation of face recognition system. Both the algorithm are described in detail below.

4.2.1 CNN Algorithm

Convolutional Neural Networks (CNNs) are deep learning models commonly used for image recognition and classification tasks. The architecture typically consists of multiple layers, including convolutional, pooling, and fully connected layers.

The fully connected layers connect every neuron in one layer to every neuron in the next layer, enabling the network to learn complex relationships between features and perform classification based on learned representations.

During training, CNNs learn to optimize weights and biases through backpropagation, minimizing the difference between predicted and actual labels using loss functions such as categorical cross-entropy.

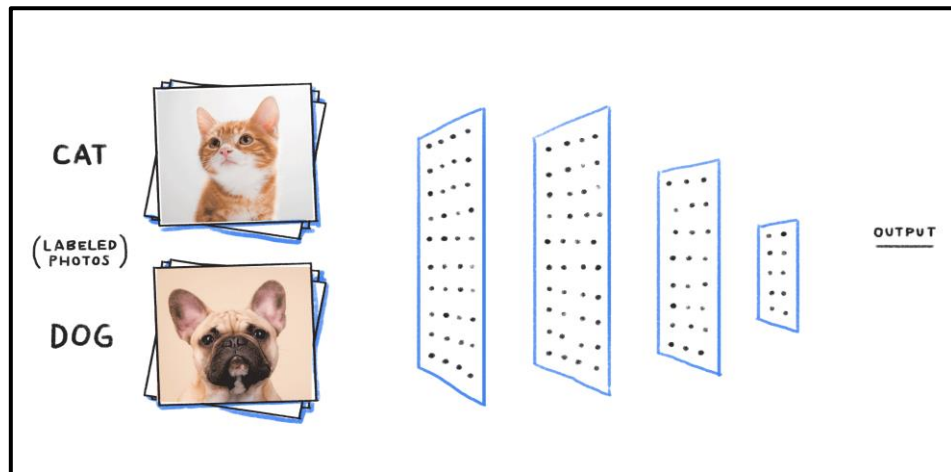


Fig:4.3 : CNN Architecture Diagram

In the convolutional layers, filters are applied to the input image to extract features through convolution operations. These filters detect patterns such as edges, textures, or shapes at different spatial scales.

Pooling layers reduce the spatial dimensions of the feature maps, consolidating important information while discarding less relevant details. This helps to reduce computation and prevent overfitting.

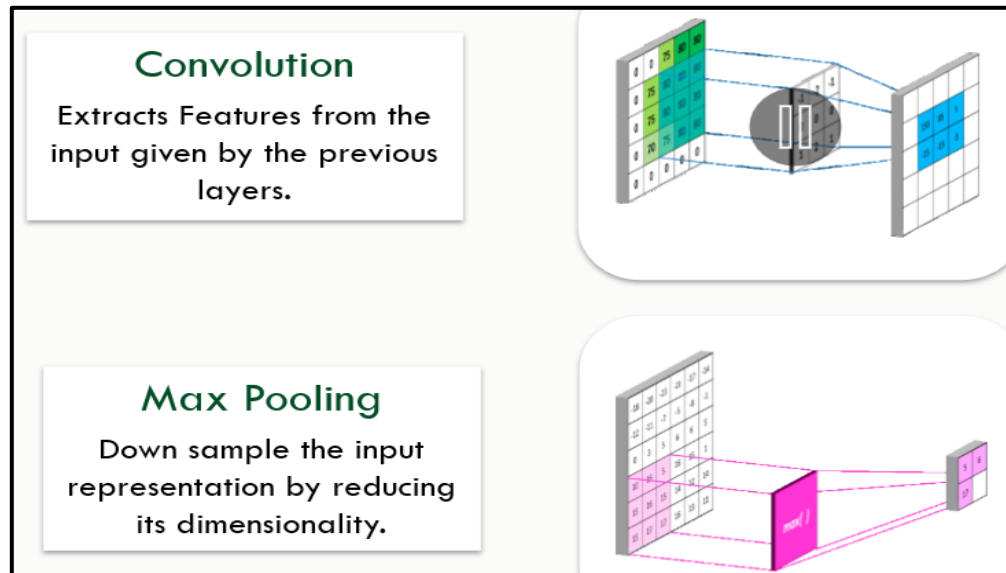


Fig:4.4 : Convolution and Max Pooling

Convolutional Neural Networks (CNNs) are a class of deep learning models specifically designed for tasks involving visual data, such as image and video analysis. CNNs are characterized by their ability to automatically learn hierarchical features from raw pixel values, making them highly effective in tasks like image classification and object detection.

A typical CNN comprises three fundamental components: convolutional layers, pooling layers, and fully connected layers. Convolutional layers apply filters to input data, enabling the detection of local patterns and features. Pooling layers reduce spatial dimensions and enhance computational efficiency. Fully connected layers perform high-level feature extraction and decision making.

Through a process of forward and backward propagation, CNNs are trained to optimize their internal parameters for feature extraction and classification. This learning process allows CNNs to excel in tasks such as image recognition, facial

recognition, and object detection. CNNs have revolutionized the field of computer vision and are widely used in various applications, including autonomous driving, medical imaging, and natural language processing.

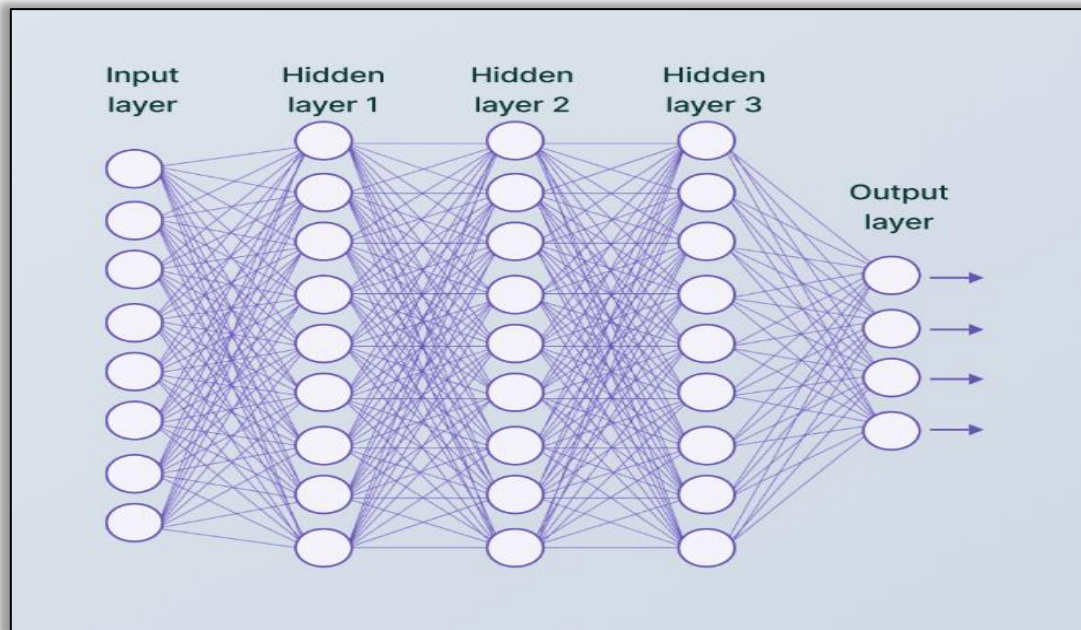


Fig:4.5 : CNN Block Diagram

4.2.2 Geolocation Algorithm

Geolocation tracking is a technology that uses data from global positioning systems (GPS), cell towers, or Wi-Fi networks to determine the precise geographic location of a device or user. It enables applications to gather and monitor real-time location data. Geolocation tracking has diverse applications, from navigation and ride-sharing services to location-based marketing and asset tracking. It plays a pivotal role in mapping, emergency services, and location-aware mobile apps. Privacy and security concerns are associated with geolocation tracking, and responsible use of this technology is crucial to safeguard user data and maintain trust.

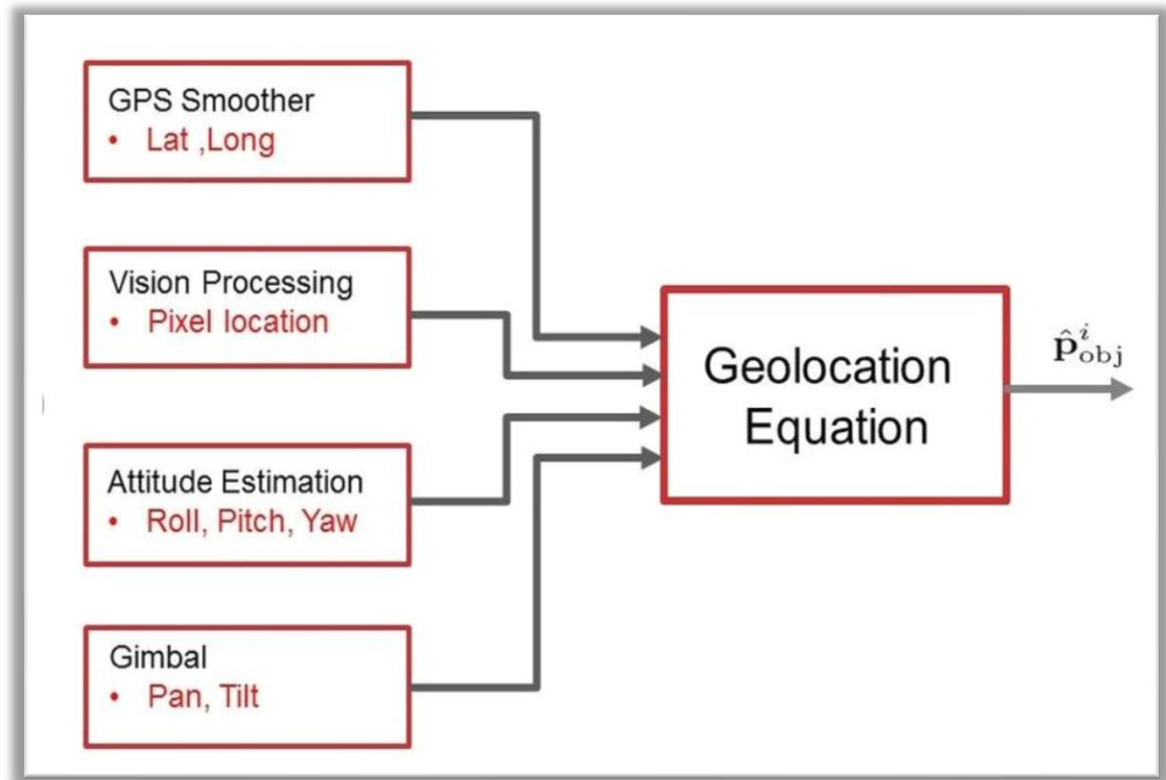


Fig:4.6 : Geolocation Algorithm

Geolocation algorithms utilize data from various sources, such as GPS satellites, Wi-Fi networks, and cellular towers, to determine the geographic location of a device or individual. These algorithms employ techniques like trilateration or triangulation to calculate the device's position based on the time it takes for signals to travel from multiple reference points to the device. By analysing signal strengths, distances, and timing, the algorithm can estimate the device's latitude and longitude coordinates. The accuracy of geolocation algorithms depends on factors such as signal quality, environmental conditions, and the number of reference points available for calculation.

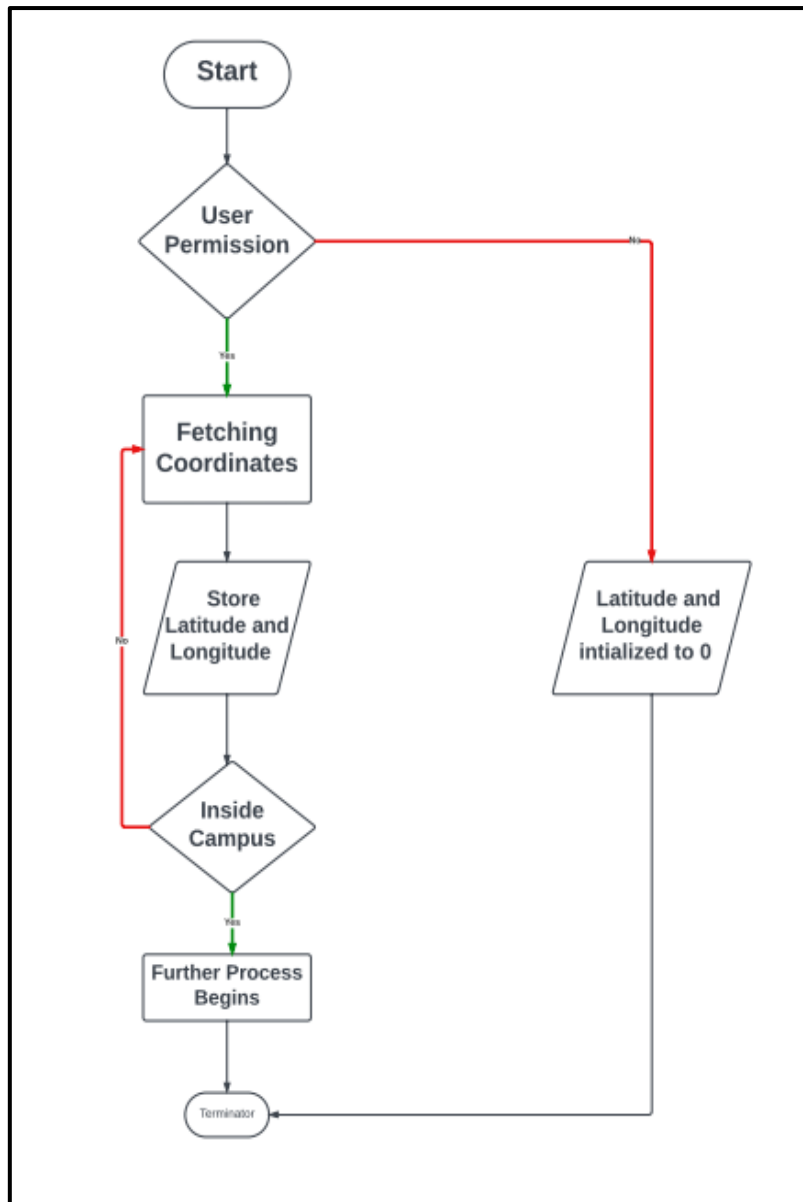


Fig:4.7 : Geolocation Architecture Diagram

4.3 ARCHITECTURE OF PROPOSED SYSTEM

The implementation of NexTrack, an employee attendance system, involves several key components:

1. Mobile App Development: The client-side of NexTrack is developed using the Flutter framework, which allows for cross-platform compatibility across Android and iOS devices. The mobile app provides an intuitive user interface for employees to log

in, mark attendance, and access their attendance records.

2. Backend Development: The server-side logic is implemented using firebase facilitating efficient data processing and communication between the client and server components.

3. Database Management: The Firebase Realtime Database is used to store and manage user profiles, attendance records, and related data. The database schema is designed to efficiently organize and retrieve information.

4. Face Recognition: Firebase's ML Kit is integrated for facial recognition. This component ensures the security and accuracy of attendance marking by verifying the identity of users based on facial features.

5. Geolocation Tracking: Firebase Geolocation services are employed to capture precise user locations in real-time. Geolocation data is crucial for validating attendance and enhancing the system's security.

6. Authentication Mechanism: Firebase Authentication is used for email and password login, ensuring a secure authentication process for users.

7. Real-time Data Synchronization: To maintain up-to-date attendance records, the system synchronizes data in real-time between the client app and the server. This feature is essential for accurate attendance tracking.

9. User Interface Design: The mobile app's user interface is designed to be intuitive and user-friendly. It provides an accessible platform for employees to log in, mark attendance, and view their attendance records.

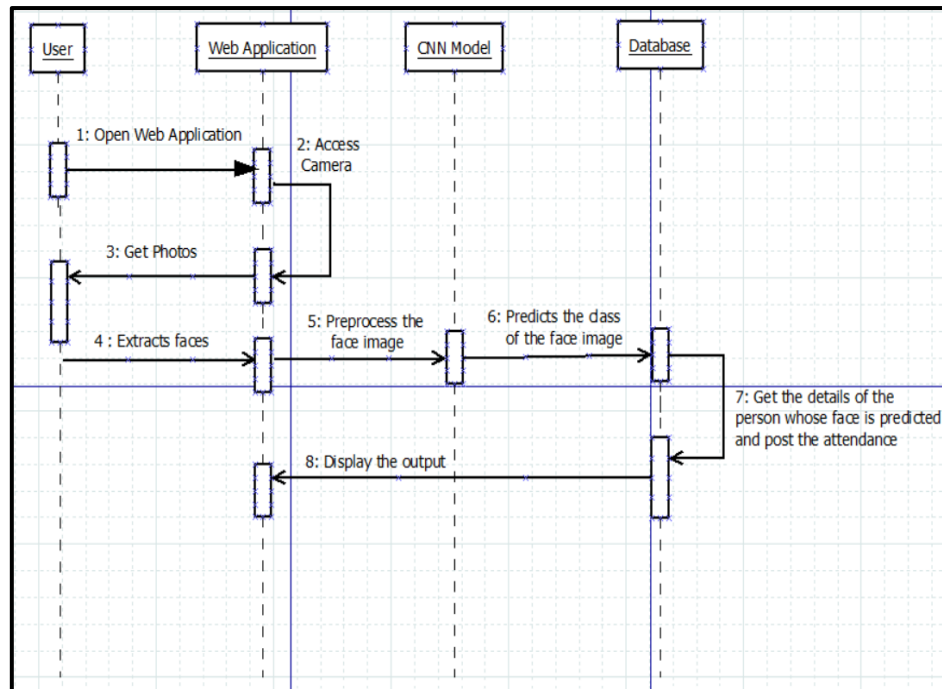


Fig:4.8 : NexTrack Working

The architecture of an Employee Attendance System encompasses various components working together seamlessly to efficiently manage attendance data. At its core, the system consists of a user interface (UI) through which employees can interact with the system to clock in/out, request leaves, and view attendance records. Biometric devices or time clocks capture attendance data, which is then processed and stored in a centralized database management system (DBMS).

Security measures, including encryption, access control, and compliance with data protection regulations, safeguard sensitive attendance data from unauthorized access or breaches. Reporting and analytics components enable users to generate various reports and analyse attendance data for insights into workforce trends and performance metrics.

Scalability and redundancy measures ensure high availability and fault tolerance, allowing the system to accommodate growth in users and data volume while maintaining reliability. Overall, the architecture of an Employee Attendance System is designed to streamline attendance management, improve accuracy, ensure compliance, and enhance workforce productivity.

4.4 IMPLEMENTATION AND TESTING PLAN OF PROPOSED MODEL

4.4.1 *Software Implementation*

The implementation of a project for an NexTrack involves various software tools and components. Here's a description of the software required for implementing such a project:

1. Flutter SDK: The Flutter software development kit (SDK) is essential for developing cross-platform mobile applications. It includes libraries, tools, and frameworks for building user interfaces, managing state, and interacting with device hardware.

2. Dart Programming Language: Flutter uses Dart as its primary programming language. Dart is a modern, object-oriented language with features such as strong typing, asynchronous programming, and a reactive programming model, making it well-suited for developing mobile applications.

3. Android Studio or Visual Studio Code: These integrated development environments (IDEs) provide powerful tools for Flutter app development, including code editing, debugging, and emulators for testing applications on Android and iOS devices.

4. Firebase or Backend Services: Depending on the requirements of the Employee Attendance System, you may need backend services for managing user authentication, storing attendance data, and handling business logic. Firebase provides a suite of backend services, including authentication, real-time database, and cloud functions, which can be integrated seamlessly with Flutter applications.

5. UI Design Tools: Designing the user interface (UI) of the Employee Attendance System requires software tools such as Adobe XD, Sketch, or Figma. These tools allow designers to create visually appealing UI designs and export assets for implementation in Flutter.

6. Version Control System (e.g., Git): Using a version control system such as Git is essential for collaborating on the Flutter project, tracking changes.

7. Continuous Integration/Continuous Deployment (CI/CD) Tools: CI/CD tools such as Jenkins, Travis CI, or GitHub Actions automate the process of building, testing, and deploying Flutter applications, ensuring code quality and facilitating rapid iteration and deployment cycles.

By leveraging these software tools and components, developers can efficiently implement a NexTrack, delivering a robust and user-friendly mobile application that meets the needs of organizations for efficient attendance management.

BASIC TESTING

The testing plan for "NexTrack" involves several steps to ensure the quality, reliability, and functionality of the application. Here's a description of the testing plan for such a project:

1. Unit Testing: Unit tests are written to verify the individual units of code, such as functions, classes, or methods, in isolation. In the case of the Employee Attendance System, unit tests would validate the logic for features like user authentication, attendance tracking, and data manipulation.

2. Integration Testing: Integration tests verify the interactions and compatibility between different modules or components of the application. This includes testing the integration of UI components, database interactions, and external services such as authentication or backend APIs.

3. Widget Testing: Widget tests focus on testing the UI components and user interactions within the application. For the Employee Attendance System, widget tests would validate the functionality of UI elements like buttons, forms, lists, and navigation flows.

4. End-to-End (E2E) Testing: E2E tests simulate user interactions with the application from start to finish, covering multiple screens and scenarios. This type of testing ensures that all components work together seamlessly and that the application behaves as expected in real-world usage scenarios.

5. Device and Platform Testing: Since Flutter supports cross-platform

development, it's essential to test the application on various devices and platforms, including Android and iOS smartphones and tablets. Testing on different screen sizes, resolutions, and operating system versions helps ensure compatibility and consistency across devices.

6. Performance Testing: Performance testing evaluates the responsiveness, scalability, and resource utilization of the application under different conditions, such as varying loads and network speeds. For the Employee Attendance System, performance testing may include measuring the app's startup time, response times for attendance tracking operations, and data synchronization performance.

7. Security Testing: Security testing involves identifying and mitigating potential vulnerabilities and risks in the application, such as unauthorized access, data breaches, or privacy concerns. This includes testing authentication mechanisms, data encryption, and secure transmission of sensitive information.

8. Accessibility Testing: Accessibility testing ensures that the application is usable and navigable by users with disabilities. This involves testing features such as screen readers, keyboard navigation, and contrast ratios to comply with accessibility standards and guidelines.

By following this testing plan, developers can identify and address issues early in the development process, leading to a more robust and reliable Employee Attendance System that meets the needs of users and organizations effectively.

4.5 PROJECT MANAGEMENT PLAN

A project management plan for NexTrack, an employee attendance system, encompasses various key elements:

Project Scope and Objectives:

Define the project's scope, including its functionalities, features, and goals. Clearly state the objectives, such as improving attendance tracking and enhancing user experience.

Stakeholder Identification:

Identify project stakeholders, including developers, users, administrators, and management. Determine their roles and responsibilities.

Project Timeline:

Create a project timeline with milestones, tasks, and deadlines. Use project management tools to track progress.

Resource Allocation:

Allocate resources such as development teams, infrastructure, and budget for the project.

Risk Management:

Identify potential risks (security breaches, technical issues) and develop risk mitigation strategies.

Quality Assurance:

Establish quality control measures, including testing and validation, to ensure a reliable product.

User Training and Support:

Plan for user training and support, ensuring users can effectively utilize NexTrack.

4.6 FINANCIAL REPORT

- Development Costs : 10000 (6 months)
- Maintenance Cost : 20000 (1 year)
- Deployment Cost : 20000 (1 month)
 - Firebase Cloud Costs (Database + Hosting)
 - Firebase ML Kit Costs
- Marketing Cost : 10000 (1 month)

The NexTrack project budget report outlines estimated costs and financial projections over an eight-month period. Development costs, incurred over six months, are projected at \$10,000. Maintenance costs for one year are expected to amount to \$20,000, with deployment costs of \$20,000 for one month. Firebase Cloud costs for database and hosting, as well as Firebase ML Kit expenses, will depend on usage and could range from several hundred to several thousand dollars annually. A \$10,000 marketing budget for one month aims to promote user adoption.

The overall cost projection for this period is approximately Rs 60,000. It's important to note that Firebase costs are subject to variables like user numbers and data storage, necessitating careful monitoring and optimization. The success of marketing campaigns and user acquisition efforts will also impact the final budget. This report provides a financial overview, helping to allocate resources efficiently and track financial performance throughout the project.

4.7 TRANSITION TO OPERATIONS PLAN

Transitioning a software project to operations involves several key steps to ensure a smooth and successful deployment. Here's a plan outlining the transition process:

1. Documentation Review: Review all project documentation, including requirements, design specifications, user manuals, and testing reports, to ensure completeness and accuracy. Update any outdated or missing documentation to reflect the final state of the software.

2. Training and Knowledge Transfer: Provide training sessions for operations staff, including system administrators, support personnel, and end-users, to familiarize them with the software functionality, usage, and maintenance procedures. Conduct hands-on training sessions and provide access to training materials and resources.

3. Infrastructure Setup: Set up the necessary infrastructure for hosting and running the software, including servers, databases, networking, and security configurations. Ensure that all infrastructure components meet the requirements specified during the development phase and are properly configured for production use.

4. Deployment Planning: Develop a deployment plan outlining the steps, timelines, and responsibilities for deploying the software into production. Identify any dependencies, risks, or potential issues that may arise during deployment and develop mitigation strategies accordingly.

5. Testing and Validation: Conduct thorough testing and validation of the software in a production-like environment to ensure that it performs as expected and meets the requirements of end-users. This may include functional testing, performance testing, security testing, and user acceptance testing.

6. Data Migration: If migrating from an existing system or importing data from external sources, develop a data migration plan to transfer data accurately and securely into the new system. Validate the integrity and completeness of migrated data to ensure that it meets the needs of end-users.

7. User Support and Helpdesk Setup: Establish a helpdesk or support system to provide assistance to end-users and address any issues or questions that may arise during the transition and beyond. Ensure that support staff are trained and equipped to handle user inquiries effectively.

By following this transition plan, organizations can ensure a seamless and successful deployment of the software into operations, minimizing disruptions and maximizing the benefits of the new system for end-users and stakeholders.

CHAPTER 5

IMPLEMENTATION DETAILS

The Implementation of the software involves translating design specifications into executable code using programming languages and development frameworks. This process includes writing, testing, and debugging code to ensure functionality, reliability, and performance. Implementation details encompass architectural decisions, coding standards, and integration with external systems or databases. Additionally, deployment configurations, such as server setup, database initialization, and version control, are addressed. Collaboration among development teams, adherence to project timelines, and continuous communication with stakeholders are critical throughout the implementation phase to achieve project objectives effectively.

5.1 DEVELOPMENT AND DEPLOYMENT SETUP

Development and deployment setup for NexTrack involves several key steps to ensure successful implementation:

- 1. Requirements Gathering:** Gather requirements from stakeholders, including HR personnel, managers, and end-users, to understand the functionality, features, and objectives of the Employee Management System.
- 2. Architecture Design:** Design the architecture of the system, including the database schema, user interface layout, and backend logic. Determine the technologies and frameworks to be used, such as Flutter for the frontend, Firebase for backend services, and relational databases for data storage.
- 3. Development:** Develop the Employee Management System according to the design specifications. This involves writing code, implementing features, and conducting regular testing to ensure functionality, usability, and reliability.
- 4. Testing:** Conduct thorough testing of the Employee Management System to identify and fix any bugs, errors, or usability issues. This includes unit testing, integration testing, and user acceptance testing to validate the system's performance and functionality.

5. Deployment Preparation: Prepare for deployment by configuring the necessary infrastructure, such as servers, databases, and networking components. Set up deployment pipelines and automation tools to streamline the deployment process.

6. Deployment: Deploy the Employee Management System to production environments, following best practices for deployment, such as continuous integration/continuous deployment (CI/CD) pipelines, version control, and rollback procedures.

7. User Training: Provide training and support to users, including HR personnel, managers, and employees, to ensure they understand how to use the Employee Management System effectively. Offer resources such as user manuals, tutorials, and helpdesk support to address any questions or issues that may arise.

8. Monitoring and Maintenance: Monitor the Employee Management System in production to ensure uptime, performance, and security. Implement monitoring tools and alerting systems to detect and respond to any issues proactively. Conduct regular maintenance, including applying updates, patches, and enhancements to the system as needed.

5.2 TESTING PLAN

The testing plan for "NexTrack" involves several steps to ensure the quality, reliability, and functionality of the application. Here's a description of the testing plan for such a project:

1. Unit Testing: Unit tests are written to verify the individual units of code, such as functions, classes, or methods, in isolation. In the case of the Employee Attendance System, unit tests would validate the logic for features like user authentication, attendance tracking, and data manipulation.

2. Integration Testing: Integration tests verify the interactions and compatibility between different modules or components of the application. This includes testing the integration of UI components, database interactions, and external services.

3. Widget Testing: Widget tests focus on testing the UI components and user

interactions within the application. For the Employee Attendance System, widget tests would validate the functionality of UI elements like buttons, forms, lists, and navigation flows.

4. End-to-End (E2E) Testing: E2E tests simulate user interactions with the application from start to finish, covering multiple screens and scenarios. This type of testing ensures that all components work together seamlessly and that the application behaves as expected in real-world usage scenarios.

5. Device and Platform Testing: Since Flutter supports cross-platform development, it's essential to test the application on various devices and platforms, including Android and iOS smartphones and tablets. Testing on different screen sizes, resolutions, and operating system versions helps ensure compatibility and consistency across devices.

6. Performance Testing: Performance testing evaluates the responsiveness, scalability, and resource utilization of the application under different conditions, such as varying loads and network speeds. For the Employee Attendance System, performance testing may include measuring the app's startup time, response times for attendance tracking operations, and data synchronization performance.

7. Security Testing: Security testing involves identifying and mitigating potential vulnerabilities and risks in the application, such as unauthorized access, data breaches, or privacy concerns. This includes testing authentication mechanisms, data encryption, and secure transmission of sensitive information.

8. Accessibility Testing: Accessibility testing ensures that the application is usable and navigable by users with disabilities. This involves testing features such as screen readers, keyboard navigation, and contrast ratios to comply with accessibility standards and guidelines.

By following this testing plan, developers can identify and address issues early in the development process, leading to a more robust and reliable Employee Attendance System that meets the needs of users and organizations effectively.

CHAPTER 6

RESULT AND DISCUSSIONS

The implementation of an NexTrack has yielded significant improvements in organizational efficiency, employee productivity, and administrative processes. Following its deployment, a comprehensive analysis of the system's performance and impact was conducted, leading to noteworthy findings and insights.

The system facilitated streamlined attendance tracking, leave management, and performance evaluation, resulting in reduced administrative burdens and improved accuracy in record-keeping. By automating these processes, HR personnel were able to dedicate more time to strategic initiatives and employee development programs, fostering a more productive and engaged workforce.

Employee feedback on the application was overwhelmingly positive, with many expressing appreciation for the system's user-friendly interface, intuitive features, and accessibility across devices. Employees reported increased transparency in attendance tracking and leave approval processes, leading to greater trust and confidence in the organization's HR practices.

However, challenges such as data privacy concerns and technical glitches were encountered during the implementation phase. Addressing these challenges required close collaboration between IT teams and system users to implement robust security measures and enhance system reliability.

Moving forward, ongoing system optimization and refinement are essential to ensure the application continues to meet the evolving needs of the organization and its workforce. This includes regular updates to address user feedback, incorporate new features, and adapt to changes in labour laws and regulations.

6.1 MODULE PERFORMANCE

There are various modules implemented for the development of NexTrack. Each module has its own accuracy level and they play critical roles in working of the system.

The accuracy graph of a Convolutional Neural Network (CNN) depicts the performance of the model in correctly classifying data over epochs. It illustrates the trend of increasing or stabilizing accuracy as training progresses. Higher accuracy values indicate better classification performance, while fluctuations may indicate overfitting or underfitting of the model.

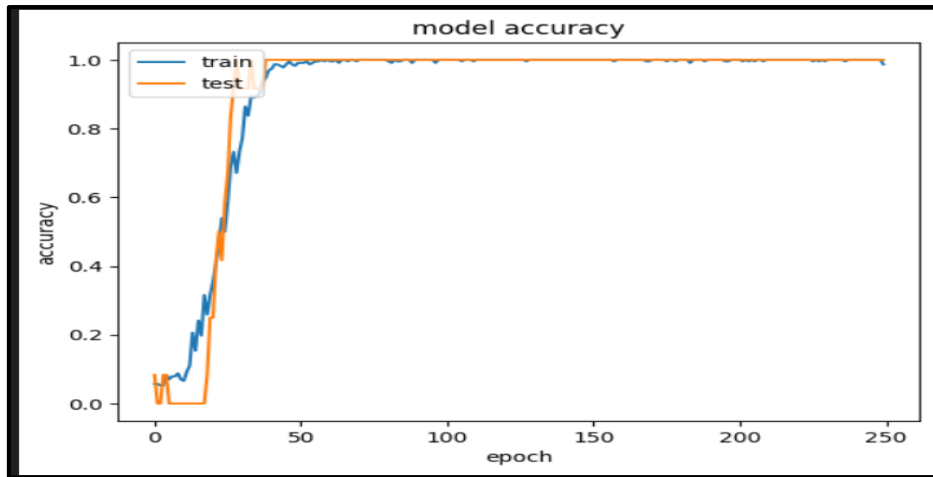


Fig:6.1 : CNN Model Accuracy

The loss graph of a CNN illustrates the progression of the loss function, typically categorical cross-entropy, over epochs. Lower loss values represent better model performance, indicating that the predicted probabilities are closer to the actual labels. The loss graph should ideally show a decreasing trend, signifying improved model convergence and learning.

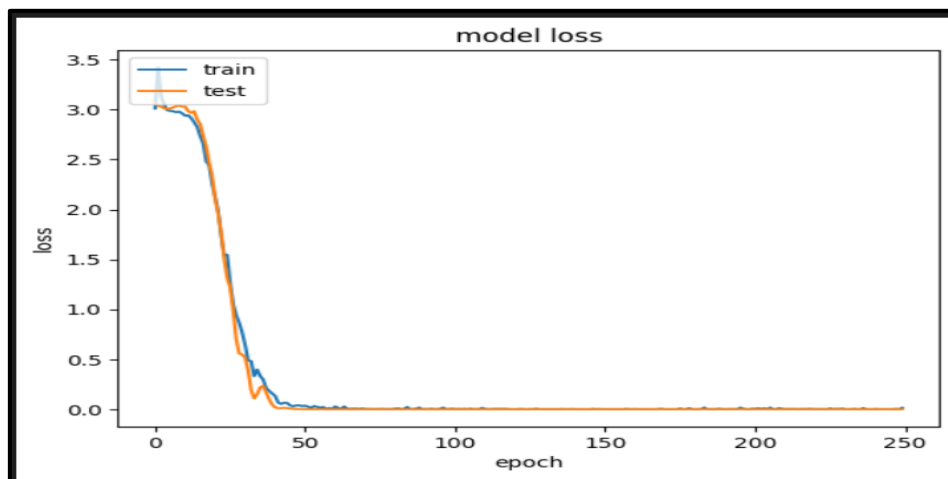


Fig:6.2 : CNN Model Loss

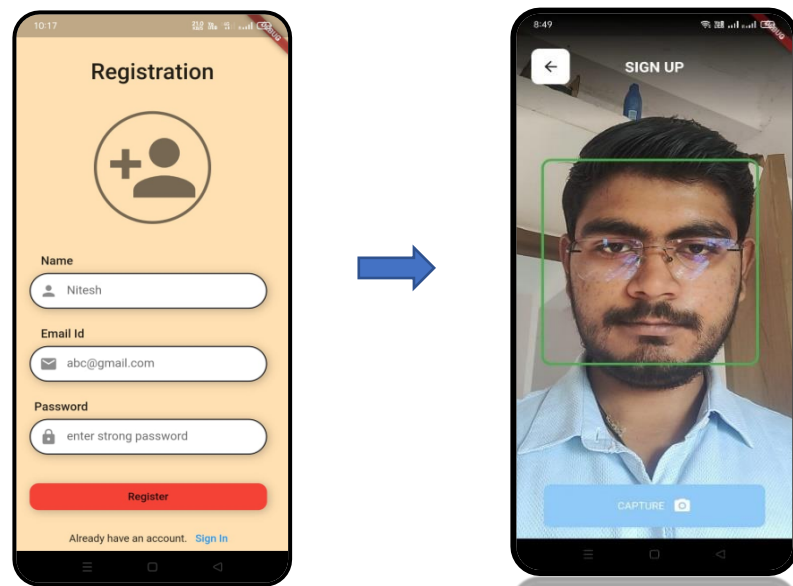


Fig:6.3 : Registration Screen

Fig:6.3 represents the registration screen of the application. It is used for registration of the user or employee of organisation. Once the details are entered than the step in this process is registration of the user's face. This page can only be accessed by the admin of the organisation.

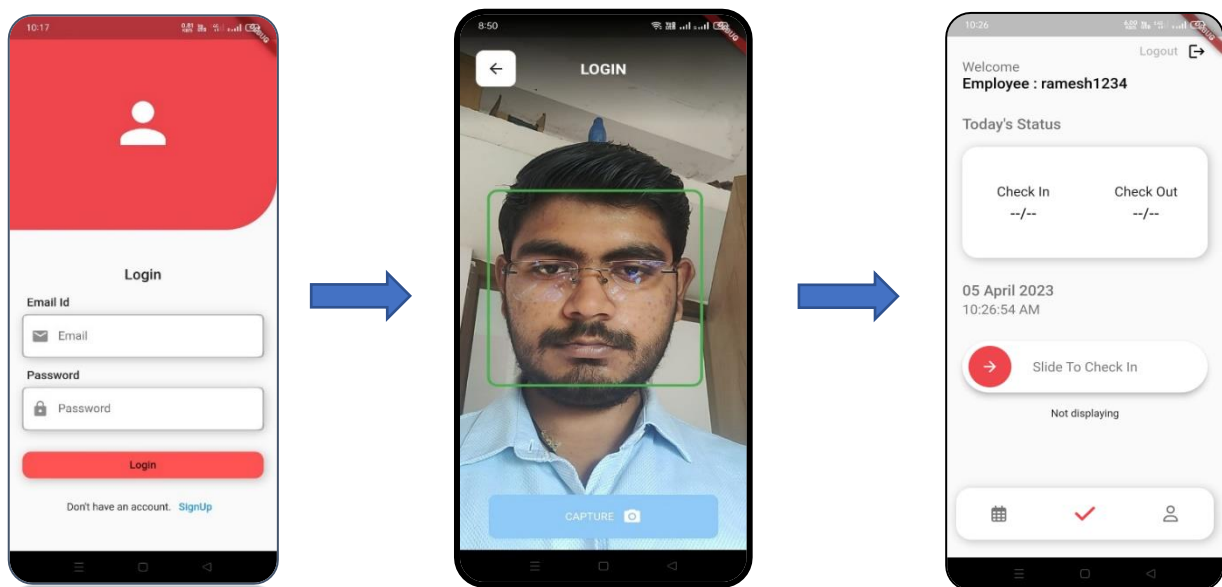


Fig:6.4 : Login Screen

Here the first screen is 1st step of authentication. Here we need to enter email id and password. Than followed by second step, where we need to authenticate our face than fingerprint and finally we enter to main screen for attendance.

CHAPTER 7

CONCLUSION

7.1 CONCLUSION

In conclusion, NexTrack, an innovative employee attendance system, offers a comprehensive solution for attendance tracking and management. The project's successful implementation demonstrates its potential to significantly benefit organizations and businesses in various industries.

NexTrack's robust features, including two-factor authentication, facial recognition, geolocation tracking, and automated attendance calculations, enhance the accuracy and security of attendance records. The color-coded data presentation simplifies attendance monitoring, while personalized warnings for obesity add a unique health-conscious dimension.

While the future scope for NexTrack is promising, it is important to highlight its current significance. The system addresses the common challenges associated with traditional attendance tracking, providing a secure, efficient, and user-friendly solution.

Continuous improvements and adaptations, such as enhanced features, advanced data analytics, and expanding into diverse industries, offer significant potential for NexTrack's evolution. Furthermore, staying aligned with data privacy regulations and ensuring robust security are key to its ongoing success.

Overall, NexTrack is a valuable addition to attendance management, enabling organizations to streamline processes, enhance security, and gain valuable insights into employee attendance patterns. Its adaptability and potential for further development make it a dynamic project with a bright future in the realm of workforce management.

7.2 FUTURE WORK

The future scope of NexTrack, an employee attendance system, is promising, as it offers a range of possibilities for further development and expansion in various domains. Here are some potential future directions for the NexTrack project:

1. Enhanced Features: NexTrack can evolve by incorporating additional features such as facial recognition improvements, biometric authentication methods, and advanced analytics for attendance and performance insights. These enhancements would make the system even more robust and valuable for businesses and organizations.

2. Machine Learning: Implementing machine learning algorithms can further improve attendance prediction, anomaly detection, and early warning systems for absenteeism or unusual attendance patterns.

3. Customization: Future developments can focus on allowing organizations to customize NexTrack to meet their specific needs, whether in terms of branding, reporting, or additional functionalities.

4. Geolocation-Based Features: Leveraging geolocation data opens opportunities for geofencing attendance, ensuring employees are physically present in the designated work areas during check-in and check-out. This can be particularly useful in industries with fieldwork or outdoor activities.

5. Cross-Industry Adoption: NexTrack can expand into various industries beyond traditional workplaces, including education, events, conferences, and healthcare, where accurate attendance tracking and security are paramount.

6. Data Analytics: Advanced data analytics and reporting tools can provide organizations with valuable insights into employee attendance patterns, helping in decision-making, resource allocation, and productivity optimization.

7.3 RESEARCH ISSUES

Research issues related to an NexTrack may encompass various aspects of system design, implementation, usage, and impact. Here are several research issues that could be explored:

1. User Experience (UX) Design: Investigate user interface design principles and usability considerations to enhance the user experience of the Employee Management System, particularly focusing on usability, accessibility, and user satisfaction.

2. Data Security and Privacy: Explore methods for ensuring the security and privacy of employee data stored and processed within the system, addressing concerns such as data encryption, access control, compliance with regulations (e.g., GDPR), and mitigation of security vulnerabilities.

3. Integration with Emerging Technologies: Investigate the integration of emerging technologies, such as artificial intelligence (AI), machine learning (ML), blockchain, or Internet of Things (IoT), to enhance the functionality and capabilities of the Employee Management System, such as predictive analytics for workforce management or biometric authentication for enhanced security.

4. Employee Engagement and Adoption: Examine strategies for promoting employee engagement and adoption of the Employee Management System, including change management techniques, training programs, communication strategies, and incentives for system usage.

5. Organizational Impact and Return on Investment (ROI): Evaluate the organizational impact and ROI of implementing an Employee Management System, including its effects on employee productivity, operational efficiency, cost savings, and overall organizational performance.

6. Cross-Cultural and Global Considerations: Investigate cultural differences and global variations in workforce management practices and regulations, and explore how these factors influence the design, implementation, and usage of Employee Management Systems in diverse organizational contexts.

7. Ethical and Social Implications: Examine the ethical and social implications of using Employee Management Systems, including potential biases in algorithmic decision-making, surveillance concerns, and implications for employee autonomy, trust, and well-being.

8. Continuous Improvement and Innovation: Explore approaches for continuous improvement and innovation in Employee Management Systems, including agile development methodologies, user feedback mechanisms, and strategies for adapting to changing organizational needs and technological advancements.

By addressing these research issues, scholars and practitioners can contribute to the advancement of knowledge and practice in the field of employee management systems, ultimately leading to more effective and sustainable solutions for workforce management in organizations.

7.4 IMPLEMENTATION ISSUES

Implementation issues in the development and deployment of a NexTrack can arise at various stages of the project lifecycle. Here are some common implementation issues that organizations may encounter:

1. Technical Challenges: Technical complexities, such as integrating disparate systems, ensuring compatibility with legacy systems, and scalability issues, can hinder the smooth implementation of an EMS.

2. Data Migration: Migrating existing employee data from legacy systems to the new EMS can be challenging due to data format discrepancies, data quality issues, and data volume considerations.

3. User Adoption: Resistance to change and lack of user buy-in can impede the successful adoption of the EMS. Employees may be reluctant to embrace new processes or technologies, requiring effective change management strategies and training programs.

4. Customization Requirements: Organizations often have unique requirements and workflows that may not be fully addressed by off-the-shelf EMS solutions.

Customization efforts can introduce complexity, delays, and additional costs to the implementation process.

5. Security Concerns: Ensuring the security and confidentiality of employee data stored in the EMS is paramount. Implementation issues related to data security may include vulnerabilities in authentication mechanisms, encryption practices, and access control policies.

6. Regulatory Compliance: Compliance with labour laws, data protection regulations (e.g., GDPR), and industry standards adds complexity to EMS implementation. Ensuring that the system meets regulatory requirements while maintaining operational efficiency is critical.

7. Resource Constraints: Limited resources, such as budget, time, and skilled personnel, can pose significant challenges during EMS implementation. Adequate resource allocation and project management are essential to overcome these constraints.

8. Integration with Third-party Systems: Integrating the EMS with third-party applications, such as payroll systems, time-tracking software, or biometric devices, may encounter compatibility issues, data synchronization challenges, and API limitations.

9. Performance Optimization: Optimizing the performance of the EMS to handle large volumes of concurrent users, data processing tasks, and system interactions requires careful design and testing to avoid performance bottlenecks and degradation.

10. Training and Support: Providing comprehensive training and ongoing support to users, administrators, and IT staff is essential for the successful implementation and long-term sustainability of the EMS.

REFERENCES

- [1]. A. U. Naik and N. Guinde, "LBPH Algorithm for Frontal and Side Profile Face Recognition on GPU", 2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT), pp. 776-779, 2020.
- [2]. C. Alippi, M. Bocca, G. Boracchi, N. Patwari, and M. Roveri, RTI goes wild: Radio tomographic imaging for outdoor people detection and localization, IEEE
- [3]. C. Stoll, R. Palluel-Germain, R. Caldara, J. Lao, M. W. G. Dye, F. Aptel, and O. Pascalis, "Face recognition is shaped by the use of sign language," J. Deaf Stud. Deaf Educ., vol. 23, no. 1, pp. 1–9, 2018.
- [4]. D. Halperin, W. Hu, A. Sheth, and D. Wetherall, Tool release: Gathering 802.11N traces with channel state information, in ACM SIGCOMM Comput. Commun., vol.
- [5]. Danijel Mijić, Ognjen Bjelica, Jelena Durutović and Miloš Ljubojević, "An Improved Version of Student Attendance Management System Based on RFID", 18th International Symposium INFOTEH-IAHORINA (INFOTEH), 2019.
- [6]. F. Adib and D. Katab, See through walls with WiFi, in Proc. ACM SIGCOMM, Hong Kong, China, 2013, pp. 75–86.
- [7]. K. Solanki and P. Pittalia, "Review of face recognition techniques," Int. J. Comput. Appl., vol. 133, no. 12, pp. 20–24, Jan. 2016.
- [8]. Naman Gupta and Shikha Jain, "Speaker Identification Based Proxy Attendance Detection System", International Conference on Signal Processing and Communication (ICSC), pp. 175-179, 2019.
- [9]. W. Deng, J. Hu, and J. Guo, "Face recognition via collaborative representation: Its discriminant nature and superposed representation," IEEE Trans. Pattern Anal. Mach. Intell., vol. 40, no. 10, pp. 2513–2521, Oct. 2018.
- [10]. Y. Wang, K. Wu, and L. M. Ni, WiFall: Device-free fall detection by wireless networks, IEEE Trans. on Mobile Computer, vol. 16, no. 2, pp. 581–594, 2017.

APPENDIX

A. SOURCE CODE

```
import
'package:attendance_system/UI/Attendance_Screen/HomeScreen.dart
';
import
'package:attendance_system/UI/Authentication/SignUp.dart';
import
'package:attendance_system/UI/Authentication/signinFace.dart';
import
'package:attendance_system/services/facenet_service.dart';
import
'package:attendance_system/services/location_service.dart';
import
'package:attendance_system/services/ml_kit_service.dart';
import 'package:attendance_system/utils/utils.dart';
import 'package:camera/camera.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:email_validator/email_validator.dart';
import 'package:firebase_auth/firebase_auth.dart';
import 'package:flutter/material.dart';
import 'package:shared_preferences/shared_preferences.dart';

import '../modal/user.dart';

class LoginScreen extends StatefulWidget {
  const LoginScreen({Key? key}) : super(key: key);

  @override
  State<LoginScreen> createState() => _LoginScreenState();
}

class _LoginScreenState extends State<LoginScreen> {

  FaceNetService _faceNetService =
FaceNetService.faceNetService;
  MLKitService _mlKitService = MLKitService();
  // DataBaseService _dataBaseService = DataBaseService();

  late CameraDescription cameraDescription;
  String githubUrl = "https://github.com/The-Assembly";

  late SharedPreferences sharedPreferences;

  bool loading = false;
```

```

double screenHeight = 0;
double screenWidth = 0;

Color primary = const Color(0xffeef444c);

TextEditingController emailController =
TextEditingController();
TextEditingController passwordController =
TextEditingController();

final _formKey = GlobalKey<FormState>();

final FirebaseAuth _auth = FirebaseAuth.instance;

@override
void initState() {
  // TODO: implement initState
  super.initState();
  _startLocationService();
  _startup();
}

Future<void> _startLocationService() async
{
  LocationService().initialize();

  LocationService().getLongitude().then((value)
  {
    setState(() {
      Users.long = double.parse(value!.toStringAsFixed(6));
    });

    LocationService().getLatitude().then((value)
    {
      setState(() {
        Users.lat = double.parse(value!.toStringAsFixed(5));
      });
    });
  });
});

@override
void dispose() {
  emailController.dispose();
  passwordController.dispose();
}

```

```

    super.dispose();
}

@override
Widget build(BuildContext context) {
    screenHeight = MediaQuery
        .of(context)
        .size
        .height;
    screenWidth = MediaQuery
        .of(context)
        .size
        .width;

    return Scaffold(
        body: SingleChildScrollView(
            child: Column(
                children: [
                    // This Container contains the colour of top
container with icon
                    Container(
                        height: screenHeight / 2.5,
                        width: screenWidth,
                        decoration: BoxDecoration(
                            color: primary,
                            borderRadius: const BorderRadius.only(
                                bottomRight: Radius.circular(70.0),
                            ),
                        ),
                        child: Icon(
                            Icons.person,
                            color: Colors.white,
                            size: screenWidth / 4,
                        ),
                    ),

                    // This Container will contain the text i.e., Login

                    Container(
                        margin: EdgeInsets.only(
                            top: screenHeight / 15, bottom: screenWidth /
20),
                        child: Text(
                            "Login",
                            style: TextStyle(
                                fontSize: screenWidth / 18, fontWeight:
FontWeight.w500),
                        ),

```

```

    ),

    // This Column contains container that have text
    form field for user id and password

    Form(
      key: _formKey,
      child: Column(
        crossAxisAlignment: CrossAxisAlignment.start,
        children: [
          Container(
            margin: EdgeInsets.only(
              bottom: screenHeight / 100, left:
screenWidth / 15),
            child: const Text(
              "Email Id",
              style: TextStyle(
                fontSize: 16.0, fontWeight:
FontWeight.w500),
            ),
          ),
          Container(
            margin: EdgeInsets.only(
              left: screenWidth / 20,
              right: screenWidth / 20,
              bottom: screenHeight / 50),
            width: screenWidth,
            decoration: const BoxDecoration(
              color: Colors.white,
              borderRadius:
BorderRadius.all(Radius.circular(12)),
              boxShadow: [
                BoxShadow(
                  color: Colors.black26,
                  blurRadius: 10,
                  offset: Offset(2, 2))
              ],
            ),
            child: TextFormField(
              controller: emailController,
              decoration: InputDecoration(
                labelText: "Email",
                prefixIcon: Icon(Icons.email),
                border: OutlineInputBorder(
                  borderRadius:
BorderRadius.circular(5.0),
                ),

```

```

    ),
    validator: (value) {
      if (value == null || value.isEmpty) {
        return 'Please enter some text';
      } else if
(!EmailValidator.validate(value, true)) {
        return 'Invalid email address';
      } else {
        return null;
      }
    },
  ),
),
Container(
  margin: EdgeInsets.only(
    bottom: screenHeight / 100, left:
screenWidth / 15),
  child: const Text(
    "Password",
    style: TextStyle(
      fontSize: 16.0, fontWeight:
FontWeight.w500),
  ),
),
Container(
  margin: EdgeInsets.only(
    left: screenWidth / 20,
    right: screenWidth / 20,
    bottom: screenHeight / 50),
  width: screenWidth,
  decoration: const BoxDecoration(
    color: Colors.white,
    borderRadius:
BorderRadius.all(Radius.circular(12)),
    boxShadow: [
      BoxShadow(
        color: Colors.black26,
        blurRadius: 10,
        offset: Offset(2, 2))
    ],
  child: TextFormField(
    obscureText: true,
    controller: passwordController,
    decoration: InputDecoration(
      labelText: "Password",
      prefixIcon: Icon(Icons.lock),
      border: OutlineInputBorder(

```

```

borderRadius:
BorderRadius.circular(5.0),
),
),
validator: (value) {
  RegExp regex = RegExp(
    r'^(?=.*?[A-Z])(?=.*?[a-
z])(?=.*?[0-9])(?=.*?[!@#\$&*~]).{8,}$');
  if (value == null || value.isEmpty) {
    return 'Please enter password';
  } else if (!regex.hasMatch(value)) {
    return 'Enter valid password';
  } else {
    return null;
  }
},
),
),
InkWell(
  onTap: () {
    if (_formKey.currentState!.validate()) {
      setState(() {
        loading = true;
      });
      login();
    }
  },
  child: Container(
    margin: EdgeInsets.only(
      left: screenWidth / 20,
      right: screenWidth / 20,
      bottom: screenHeight / 50,
      top: screenHeight / 50),
    height: screenHeight / 20,
    width: screenWidth,
    child: Center(
      child: loading == true ? const
CircularProgressIndicator(strokeWidth: 2, color: Colors.white,)
: Text(
      "Login",
      style: TextStyle(
        fontSize: screenWidth / 25,
        fontWeight: FontWeight.w500),
    ),
  ),
  decoration: const BoxDecoration(
    color: Colors.redAccent,

```

```

borderRadius:
BorderRadius.all(Radius.circular(12)),
boxShadow: [
  BoxShadow(
    color: Colors.black26,
    blurRadius: 10,
    offset: Offset(2, 2))
  ],
),
),
],
),
),
),
// This Container contains the text button for
moving to sign up page

Container(
  child: Row(
    mainAxisAlignment: MainAxisAlignment.center,
    children: [
      const Text("Don't have an account."),
      TextButton(
        onPressed: () {
          Navigator.push(
            context,
            MaterialPageRoute(
              builder: (BuildContext context) =>
signUpScreen(
              cameraDescription:
cameraDescription,
            ),
          ),
        );
    ],
  ),
  child: const Text(
    'SignUp',
  ),
),
),
],
),
),
),
);
}

```

```

void login() {
  _auth.signInWithEmailAndPassword(
    email: emailController.text.toString(),
    password: passwordController.text.toString(),
  ).then((value) async {

    String emailId = trimId(emailController.text.toString());

    sharedPreferences = await
SharedPreferences.getInstance();

    setState(() {
      // Users.uid = value.toString();
      // Utils().showToast(value.uid.toString());
    });

    sharedPreferences.setString("employeeId",
emailId).then((value) async {
      setState(() {
        loading = false;
        Users.employeeId = emailId;
      });

      if(await checkIfRecordExists(Users.employeeId))
      {
        // Navigation from login screen to face scan screen
        Navigator.push(
          context,
          MaterialPageRoute(
            builder: (BuildContext context) => SignIn(
              cameraDescription:cameraDescription,
            ),
          ),
        );
      }
      else
      {
        Utils().showToast("You need to register first");
      }

    });

  }).onError((error, stackTrace){

    setState(() {
      loading = false;

```

```

    });

    Utils().showToast(error.toString());

  });
}

String trimId(String ? email)
{
  RegExp regExp = RegExp(r'(.*)@gmail.com');

  RegExpMatch? match = regExp.firstMatch(email!);

  String ? data = " ";

  if (match != null) {
    data = match.group(1);
    print('Data extracted: $data');
  } else {
    print('No match found.');
```

ID

```

  }

  return data!;
}

Future<bool> checkIfRecordExists(String recordId) async {
  bool exists = false;

  // Get a reference to the Firestore collection
  CollectionReference recordsRef =
  FirebaseFirestore.instance.collection("Employee");

  // Query the collection for the specific record using its
  QuerySnapshot snapshot = await recordsRef.where('id',
  isEqualTo: recordId).get();

  // Check if the snapshot is not empty
  if (snapshot != null && snapshot.size > 0)
  {
    exists = true;
  }

  setState(() {
    Users.uid = snapshot.docs[0].id;
  });
}

```

```

        return exists;
    }

    void _startup() async{
        _setLoading(true);

        List<CameraDescription> cameras = await availableCameras();

        /// takes the front camera
        cameraDescription = cameras.firstWhere(
            (CameraDescription camera) =>
                camera.lensDirection == CameraLensDirection.front,
        );

        // start the services
        await _faceNetService.loadModel();
        // await _dataBaseService.loadDB();
        _mlKitService.initialize();

        _setLoading(false);
    }

    void _setLoading(bool value) {

        setState(() {
            loading = value;
        });
    }

}

REGISTRATION SCREEN:

class SignIn extends StatefulWidget {
    final CameraDescription cameraDescription;

    const SignIn({
        Key? key,
        required this.cameraDescription,
    }) : super(key: key);

    @override
    _SignInState createState() => _SignInState();
}

```

HOME SCREEN :

```
import 'dart:core';

import
'package:attendance_system/UI/Attendance_Screen/OutOfLocation.d
art';
import
'package:attendance_system/UI/Attendance_Screen/calendarScreen.
dart';
import
'package:attendance_system/UI/Attendance_Screen/profileScreen.d
art';
import
'package:attendance_system/UI/Attendance_Screen/todayScreen.dar
t';
import 'package:attendance_system/modal/user.dart';
import
'package:attendance_system/services/location_service.dart';
import 'package:attendance_system/utils/utils.dart';
import 'package:cloud_firestore/cloud_firestore.dart';
import 'package:flutter/material.dart';
import
'package:font_awesome_flutter/font_awesome_flutter.dart';
import '../modal/user.dart';

class HomeScreen extends StatefulWidget {

  const HomeScreen({Key? key}) : super(key: key);

  @override
  State<HomeScreen> createState() => _HomeScreenState();
}

class _HomeScreenState extends State<HomeScreen> {
  double screenHeight = 0;
  double screenWidth = 0;

  int currentIndex = 1;

  Color primary = const Color(0xffeef444c);

  List<IconData> navigationIcons = [
    FontAwesomeIcons.calendarDays,
    FontAwesomeIcons.check,
    FontAwesomeIcons.user
  ];
};
```

```

@override
void initState() {
  super.initState();
  getId().then((value) {
    _getCredentials();
    _getProfilePic();
    setState(() {

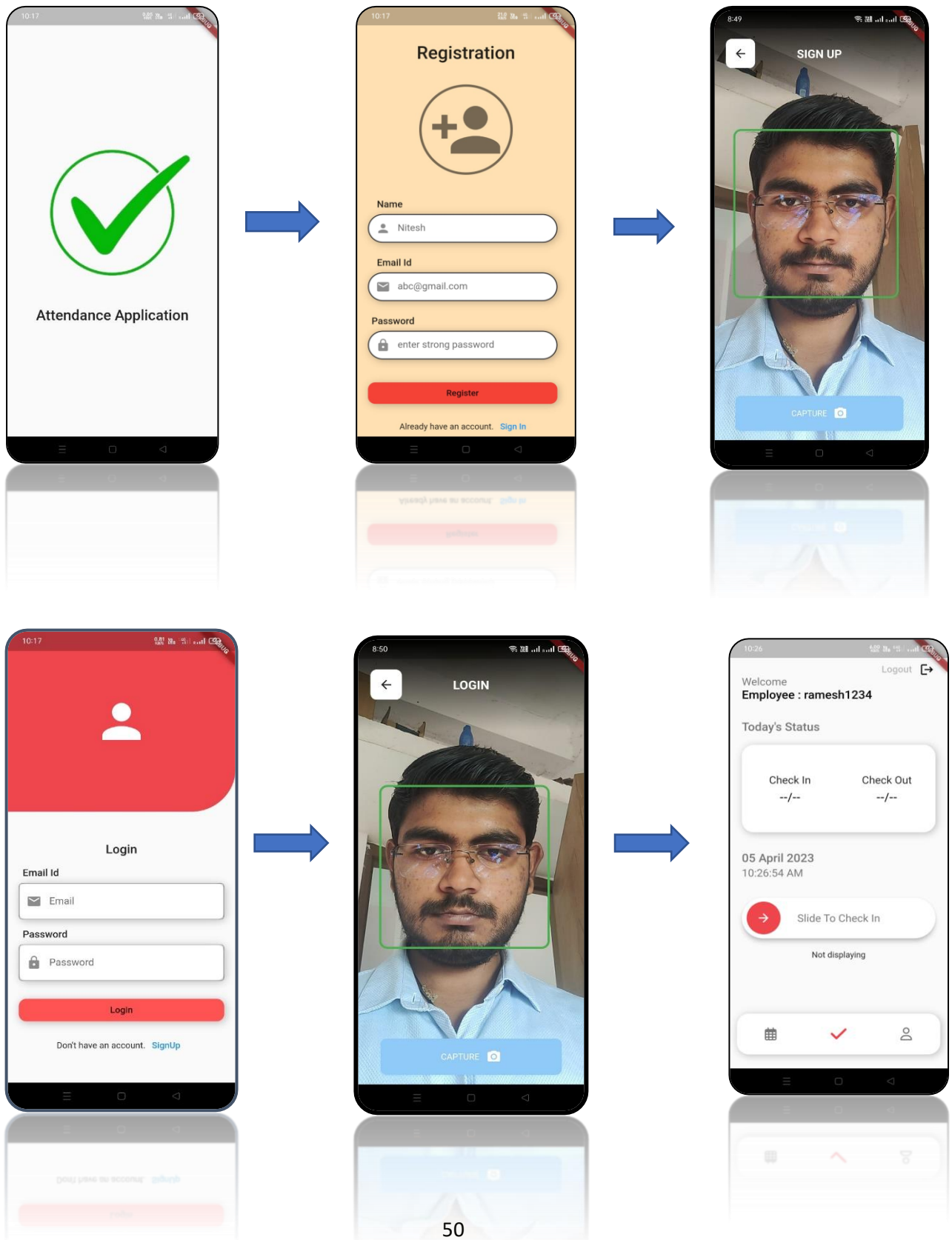
    });
  });
  // _startLocationService();
  // _determineLocation();
}

void _getCredentials() async
{
  try
  {
    FirebaseFirestore.instance
      .collection("Employee")
      .doc(Users.id)
      .get();

    setState(() {
      Users.canEdit = doc['canEdit'];
      Users.firstName = doc['firstName'];
      Users.lastName = doc['lastName'];
      Users.birthDate = doc['birthDate'];
      Users.address = doc['address'];
    });
  }
  catch(e)
  {
    Utils().showToast(e.toString());
  }
}

```

B. SCREENSHOTS



NexTrack: Next-Gen Attendance Monitoring with ML

Nitesh Kumar Singh¹ and Nayan Asawa² and Dr P. Ajitha³

^{1,2,3}Sathyabama Institute of Science and Technology, Jeppiaar Nagar, 600119, Chennai, Tamil Nadu

Corresponding Author: ajitha.it@sathyabama.ac.in

Abstract: NexTrack: Next-Gen Attendance Monitoring with ML. This system can help organizations track and monitor employee attendance, working hours, leave applications, and generate payroll reports. Employee attendance systems have become an essential tool for organizations of all sizes to manage their workforce and track employee attendance. Traditional manual attendance systems can be time-consuming and prone to errors, leading to inaccurate records and payroll issues. Digital employee attendance systems offer a more efficient and accurate solution that eliminates the need for manual processes. Digital employee attendance systems use various technologies, such as biometrics, RFID, and GPS, to record and track employee attendance data. They offer several benefits, including increased accuracy, reduced administrative burden, improved compliance, and greater visibility into employee attendance patterns. However, they also have some downsides, such as technical issues, security risks, and initial investment costs. This paper provides an overview of employee attendance systems, their benefits and downsides, and the different technologies used in these systems. It also discusses the feasibility study of implementing a digital employee attendance system and the steps involved in implementing such a system. Finally, it highlights some of the best practices for managing employee attendance and offers insights into how organizations can leverage these systems to improve their operational efficiency and better manage their workforce.

Keywords - Authentication: Email and Password, Face Recognition (CNN Algorithm), Fingerprint Authentication, Geolocation Tracker, Firebase Database, SQL Database

I. INTRODUCTION

Employee attendance is a critical aspect of workforce management for any organization. Keeping accurate records of when employees arrive, depart, and take leaves is not only essential for fair compensation but also for efficient operations, legal compliance, and data-driven decision-making. In today's digital age, businesses leverage Employee Attendance Systems, sophisticated software tools designed to streamline attendance management, reduce errors, and ensure adherence to labor laws and regulations. The Employee Attendance System is a comprehensive software solution that modernizes how organizations track and monitor their employees' attendance, working hours, leave applications, and other related aspects. This system has become an indispensable tool for human resources and management teams across various industries, from small businesses to large corporations. The primary purpose of this system is to provide a seamless and automated approach to attendance management. Traditional manual methods, such as paper-based time sheets or manual data entry, are prone to errors and inefficiencies. Employee Attendance Systems, on the other hand, offers a digital platform that simplifies attendance tracking and reporting.

A notable benefit of the system is the reduction of payroll errors. Processing payroll is a critical financial aspect of any organization, and inaccuracies can lead to dissatisfied employees, compliance issues, and potential legal disputes. The Employee Attendance System automates payroll calculations based on precise attendance data, reducing the chances of errors in wage calculations, tax withholdings, and other financial components. This, in turn, helps maintain employee satisfaction and ensures that the organization is adhering to employment contracts and labor laws.

The Employee Attendance System is a multifaceted software solution designed to address various crucial objectives within an organization. At the heart of this system lies the central mission of optimizing attendance management, streamlining workforce operations, ensuring legal compliance, and empowering data-driven decision-making. To accomplish these core objectives, the Employee Attendance System encompasses several key components and functionalities. First and foremost, accurate and efficient tracking of employee attendance is a primary goal. The system aims to capture the precise in-and-out times of employees, record leave applications, and track any deviations from standard attendance patterns. This objective is fundamental as it forms the foundation for other critical functions, such as payroll processing and labor law compliance. Automating attendance tracking reduces the likelihood of human errors in data entry and provides a real-time, digital record of attendance.

II. RELATED WORK

[1] The face recognition attendance system based on real-time video processing offers significant advantages, including high accuracy rates of up to 82%, stability in real-world applications, and effectiveness in reducing truancy rates by about 60%. Its user-friendly interface facilitates efficient attendance management. However, challenges such as occasional errors during peak hours and the need for pre-powering the system for accurate data collection may arise. Despite these limitations, the system represents a promising solution for improving attendance tracking in educational institutions, ultimately enhancing classroom efficiency and promoting student engagement.

[2] The proposed RFID-based attendance system offers enhanced accuracy and reliability by leveraging unique phase characteristics for individual recognition, mitigating the risk of fraudulent time records. With an impressive average accuracy of 92%, it ensures robust performance across varying attire and time conditions. Unlike fingerprint-based systems, RFID eliminates the need for direct physical contact, promoting hygiene and user comfort. However, implementing RFID infrastructure may incur higher costs compared to traditional methods. Additionally, the setup and calibration process may require technical expertise, potentially limiting adoption among organizations lacking the necessary resources or expertise. Overall, the system represents a promising solution for effective and secure office attendance management.

III. PROPOSED METHOD

A. Email and Password

Email and password authentication is a widely used method for verifying the identity of users in the digital world, from accessing your email account to logging into various online services and platforms. This method relies on the combination of an email address and a password as unique credentials for user identification and access control. The process begins with a user entering their registered email address and the associated password. The email address serves as the username, and the password acts as the secret key. When the user submits this information, it is typically sent to a server for verification. The server checks whether the provided email address is associated with an account in its database and if the corresponding password matches the stored, hashed version of the password.

Additionally, protecting the email account itself is essential. Since many services rely on email for password recovery and communication, a compromised email account can lead to unauthorized access to various other accounts. Therefore, securing the email account with a strong password, 2FA, and vigilant monitoring is crucial. Another concern is data breaches. If a service provider's database is compromised, user email addresses and hashed passwords can be exposed. While hashing passwords adds a layer of security, determined attackers can attempt to crack hashed passwords through various means, underscoring the importance of strong and unique passwords.

In conclusion, email and password authentication remains a common and accessible method for verifying user identities online. It offers convenience but also comes with security considerations. Users should be vigilant about creating strong, unique passwords and securing their email accounts, while service providers must take steps to protect user data and adhere to best practices for secure authentication and password storage. As the digital landscape evolves, the importance of balancing convenience with security will continue to be a central consideration in email and password authentication.

B. Face Recognition

Face recognition technology is a cutting-edge field of artificial intelligence that has gained significant attention and applications in recent years. It involves the identification or verification of individuals based on their facial features, and it has numerous potential use cases across various industries. Face recognition technology relies on advanced algorithms and machine learning models to analyze and identify unique patterns in a person's face. These patterns include the distances between facial features, the shape of the eyes, nose, mouth, and even more complex characteristics like skin texture and wrinkle patterns. It can be used for two primary purposes: identification and verification.

Identification involves comparing a captured facial image with a database of known faces to find a match. This is commonly used in security and law enforcement applications to identify individuals of interest. For example, it can help identify criminals or track missing persons. Verification, on the other hand, involves confirming that a person

is who they claim to be. This is used in scenarios like unlocking smartphones, gaining access to secure facilities, or verifying identity for online transactions.

However, there are also concerns and challenges associated with face recognition technology. Privacy concerns are a major issue. The widespread use of facial recognition raises questions about data security, consent, and the potential for abuse. There are concerns about mass surveillance, unwarranted tracking, and the risk of misuse by both government and private entities. Additionally, the technology is not infallible and can produce false positives or negatives, potentially leading to wrongful arrests or security breaches. sessions.

C. Fingerprint Authentication

Fingerprint authentication is a sophisticated biometric technology that has gained widespread use for verifying the identity of individuals. This method leverages the unique patterns and ridges on an individual's fingertips, known as fingerprint minutiae, to provide secure and convenient access to a variety of applications and systems. The process of fingerprint authentication involves the capture and analysis of fingerprint data. This typically begins with a user placing their fingertip on a fingerprint scanner or sensor. The sensor then captures an image of the fingerprint, which is then processed to extract and store specific fingerprint minutiae points. These minutiae points include ridge endings, bifurcations, and other unique characteristics that are used to create a digital representation of the fingerprint. This representation is then compared to a stored template of the user's fingerprint.

One of the key advantages of fingerprint authentication is its high level of accuracy. The uniqueness and complexity of an individual's fingerprint patterns make it extremely unlikely for two people to have identical fingerprints, which results in a low false acceptance rate. This means that the chances of an unauthorized user gaining access through someone else's fingerprint are exceedingly low. In conclusion, fingerprint authentication is a powerful and widely adopted biometric technology that relies on the unique patterns of an individual's fingertips to provide secure and efficient access control. It combines a high level of accuracy with user convenience. However, the proper handling and protection of fingerprint data are paramount to ensure privacy and security. As technology advances, fingerprint authentication continues to play a crucial role in securing devices, systems, and applications while simplifying the user experience. locations.

D. Geolocation Tracking

Geolocation tracking is a technology that enables the real-time or historical tracking and identification of the geographical location of a person, vehicle, or asset using various devices and systems. This technology utilizes a combination of GPS (Global Positioning System), cellular or Wi-Fi networks, and other positioning methods to determine and transmit location data. One of the primary benefits of geolocation tracking is enhanced visibility and control. Organizations can track the real-time location of vehicles or assets, allowing for more efficient route planning, reduced fuel consumption, and better resource allocation. For individuals, geolocation tracking on smartphones and navigation systems simplifies travel by providing turn-by-turn directions and real-time traffic updates.

Here the geolocation tracker is used to fetch the coordinates (i.e., the latitude and longitude) of the user and then decide whether the user is present inside the particular campus or not. If the user is not present inside the particular campus then he or she will be unable to access the features of application and they cannot give their attendance. This increases the security and accuracy of the application. Despite its numerous advantages, geolocation tracking raises important privacy and security concerns. The continuous collection and sharing of location data can compromise individual privacy if not properly managed. It is essential for organizations and service providers to establish robust data protection policies and obtain informed consent from users regarding the collection and use of location information. arise.

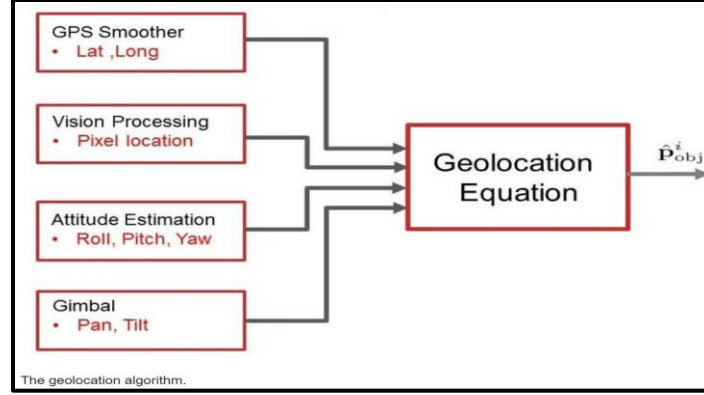


FIGURE 1. Geolocation Algorithm

E. Security Implementation

Firestore offers a robust set of security features and best practices to protect data stored and processed within its platform. Central to Firestore's security framework is its multi-layered approach, which encompasses various security measures to safeguard data at every stage of its lifecycle. It provides secure authentication mechanisms, including email/password authentication, social login options (e.g., Google, Facebook), and custom authentication methods, ensuring that only authorized users can access the system. Secondly, Firestore offers fine-grained access control through its Security Rules, allowing developers to define granular permissions and restrictions on data access and operations. Security Rules enable developers to enforce authentication requirements, validate data integrity, and restrict access based on user roles or conditions, thereby preventing unauthorized access or tampering with sensitive data.

Furthermore, Firestore employs encryption at rest and in transit to protect data both at rest in the database and during transmission over the network. This encryption ensures that data remains confidential and secure, even if intercepted or accessed by unauthorized parties. Additionally, Firestore offers built-in monitoring and logging capabilities, allowing developers to track and audit user activities, security events, and system access. This visibility enables organizations to detect and respond to security incidents promptly, mitigating potential threats and vulnerabilities.

Moreover, Firestore implements regular security updates, patches, and compliance certifications to ensure that its infrastructure and services meet industry standards and regulatory requirements. This proactive approach to security helps to mitigate emerging threats and vulnerabilities, ensuring the ongoing protection of data within the Firestore platform. Overall, Firestore provides a comprehensive suite of security features and best practices to protect data, authenticate users, enforce access controls, and monitor system activities effectively. By leveraging these security measures, organizations can confidently deploy applications on Firestore, knowing that their data is secure and compliant with regulatory standards.

IV. EXPERIMENTS

A. Experimental Setup

1) Experimental Background

The experimental background of NexTrack centers on developing and implementing an advanced employee attendance system. This system leverages real-time video processing and face recognition technology to accurately track and monitor employees' attendance, working hours, and leave applications. The project aims to enhance organizational productivity, reduce costs, and ensure compliance with labor laws and regulations. By replacing traditional attendance tracking methods with innovative technology, NexTrack seeks to streamline and modernize attendance management, making it more efficient and error-resistant. The experiment involves comparing the new

system's accuracy, stability, and user-friendliness with existing attendance tracking methods.

2) Experiment Setup Process

The experimental setup for NexTrack involves a comparative analysis between the new face recognition attendance system and traditional attendance tracking methods. To gauge accuracy, students from two universities are selected, with one group using the innovative system and the other relying on manual attendance methods. The system's stability is assessed by comparing check-in statistics, and the truancy rate is determined by monitoring students' attendance. Additionally, the system's interface settings are configured to facilitate user-friendly operation and data retrieval for a comprehensive evaluation of NexTrack performance.

B. Experimental Procedure

The experimental procedure for NexTrack involves five key steps:

1. **Data Collection:** Gather attendance data from students at two universities. One group will use the NexTrack face recognition system, while the other will follow traditional manual attendance procedures.
2. **Accuracy Assessment:** Analyze the accuracy of the NexTrack system in comparison to traditional methods. Record and compare the sign-in rates of students in both groups.
3. **Stability Testing:** Evaluate the stability of the NexTrack system by monitoring and comparing the number of check-ins with the manual attendance system.
4. **Truancy Rate Analysis:** Determine the truancy rates of students in both groups to assess the effectiveness of NexTrack in reducing absenteeism.
5. **Interface Configuration:** Adjust and optimize the NexTrack interface settings to ensure user-friendly operation and efficient data retrieval. This step ensures that the system is easy to use and effective for both students and administrators.

These steps form a comprehensive experimental procedure to assess the functionality and effectiveness of the NexTrack face recognition attendance system. informed.

C. Database Design

In designing a Firebase database to store check-in and check-out time, location, and other data of employees, it's essential to consider the structure and organization of the data to ensure efficient storage, retrieval, and manipulation. This database design should align with the requirements outlined in the research paper and adhere to Firebase's data modeling principles. The database schema for storing employee check-in/check-out data in Firebase typically consists of several interconnected collections, each representing a distinct entity or data type. The main collections in this design include:

Employees Collection: This collection stores information about employees, such as their unique identifiers (IDs), names, departments, and contact details. Each document in the employees collection represents a single employee, identified by their unique employee ID.

Attendance Records Collection: This collection records individual attendance events, including check-in and check-out timestamps, location coordinates, and any additional metadata. Each document in the attendance records collection corresponds to a specific attendance event for a particular employee, containing fields such as employee ID, timestamp, latitude, longitude, and accuracy.

To optimize query performance and minimize data duplication, the database design may leverage denormalization and nested data structures. For example, the attendance records collection can be structured as a subcollection within each employee document, allowing for efficient retrieval of attendance data for a specific employee.

Furthermore, Firebase Security Rules can be implemented to enforce access control and data validation. Security Rules can restrict access to sensitive data based on user authentication status, user roles, or specific conditions. For example, only authenticated users with the role of "admin" may have write access to attendance records.

Additionally, Firebase Authentication can be integrated to authenticate users securely and ensure that only

authorized individuals can access the database and perform operations such as adding or modifying attendance records.

In summary, the database design for storing employee check-in/check-out data in Firebase involves structuring collections to represent employees and attendance records, leveraging denormalization and nested data structures for efficiency, and implementing security measures such as Security Rules and Authentication to protect sensitive data and enforce access control. This design aligns with the requirements outlined in the research paper and provides a scalable, secure, and efficient solution for managing employee attendance data in Firebase.

In the context of a Convolutional Neural Network (CNN) module that requires storing and accessing images in an SQL database, the system architecture would involve several components to facilitate efficient image storage, retrieval, and processing. The SQL database schema would include a table dedicated to storing images, with columns for unique identifiers, image data (as binary large objects or BLOBs), metadata (such as timestamps or labels), and any additional attributes necessary for image classification or analysis.

The system would incorporate a backend application responsible for managing interactions with the SQL database, including uploading images, querying image data, and serving images to the CNN module for training or inference. This backend application would handle tasks such as image preprocessing, data augmentation, and batch processing to prepare images for consumption by CNN.

Additionally, the system would feature a frontend interface or API for users to interact with the CNN module and access image-related functionalities. This interface may allow users to upload new images, query existing image data, view image classifications or predictions generated by the CNN, and perform other image-related tasks. Security measures, such as access controls, encryption, and authentication mechanisms, would be implemented to protect sensitive image data and ensure compliance with privacy regulations. Overall, the SQL database for storing images in a CNN module provides a structured and scalable solution for managing image data, enabling efficient training, and inference processes while maintaining data integrity and security.

D. Face Recognition Module (CNN)

Convolutional Neural Networks (CNNs) have revolutionized the field of face recognition by providing a highly effective and efficient way to extract features and recognize faces from images or video. CNNs are a subset of deep learning, inspired by the human visual system. They consist of multiple layers of interconnected neurons that automatically learn relevant patterns from data. In face recognition, CNNs excel at feature extraction. The initial layers detect simple features like edges and textures, while deeper layers identify more complex facial attributes, such as eyes, nose, and mouth. This hierarchical feature extraction is crucial for recognizing faces under varying conditions like different angles, lighting, and expressions. CNNs are trained on massive datasets of labeled face images to learn to distinguish between different individuals. Once trained, they can process new images and assign probabilities to each individual's identity. The model can then select the identity with the highest probability as its recognition result.

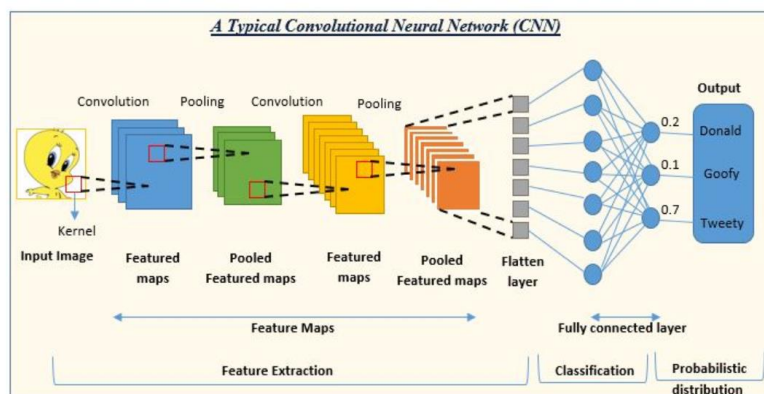


FIGURE 2. CNN Architecture

In the convolutional layers, filters are applied to the input image to extract features through convolution operations. These filters detect patterns such as edges, textures, or shapes at different spatial scales. Pooling layers reduce the spatial dimensions of the feature maps, consolidating important information while discarding less relevant details. This helps to reduce computation and prevent overfitting.

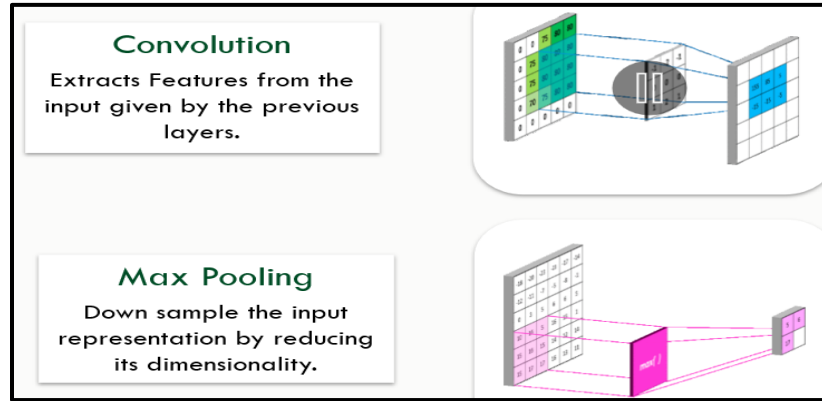


FIGURE 3. Convolution and Max Pooling

CNNs offer several advantages for face recognition, including robustness, scalability, and automation. They can handle large-scale databases, and their deep architecture allows them to learn intricate facial features that might be challenging for traditional methods. Moreover, CNNs can adapt to various environmental factors and perform exceptionally well in real-time applications. CNN-based face recognition has been widely adopted in security systems, mobile devices, and social media platforms, offering a seamless and efficient way to verify individuals' identities and enhance user experiences.

E. Accuracy and Loss Graph of CNN Module

The accuracy graph of a Convolutional Neural Network (CNN) depicts the performance of the model in correctly classifying data over epochs. It illustrates the trend of increasing or stabilizing accuracy as training progresses. Higher accuracy values indicate better classification performance, while fluctuations may indicate overfitting or underfitting of the model.

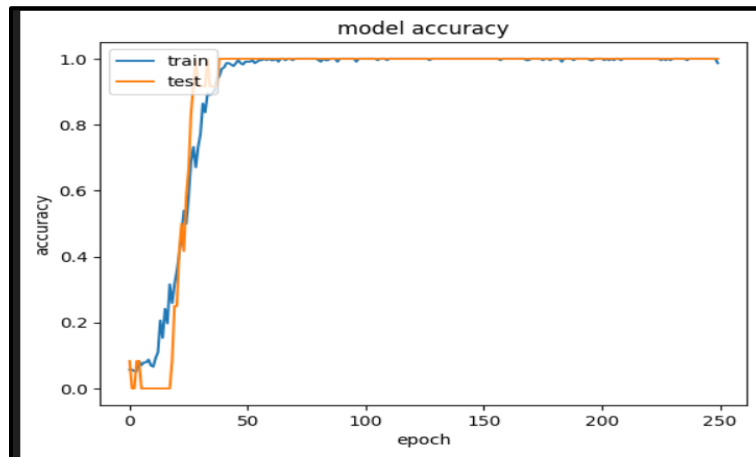


FIGURE 4. CNN Model Accuracy

The loss graph of a CNN illustrates the progression of the loss function, typically categorical cross-entropy, over epochs. Lower loss values represent better model performance, indicating that the predicted probabilities are closer

to the actual labels. The loss graph should ideally show a decreasing trend, signifying improved model convergence and learning.

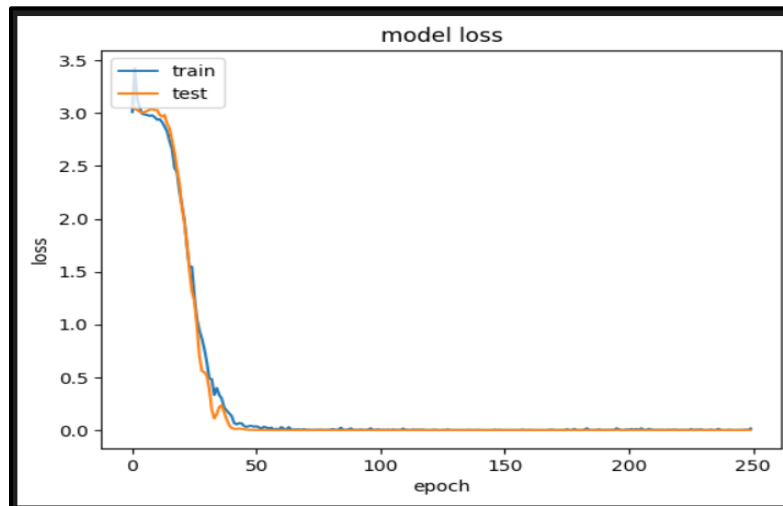


FIGURE 5. CNN Model Loss

F. Geolocation Algorithm

In conclusion Geolocation algorithms utilize data from various sources, such as GPS satellites, Wi-Fi networks, and cellular towers, to determine the geographic location of a device or individual. These algorithms employ techniques like trilateration or triangulation to calculate the device's position based on the time it takes for signals to travel from multiple reference points to the device. By analyzing signal strengths, distances, and timing, the algorithm can estimate the device's latitude and longitude coordinates. The accuracy of geolocation algorithms depends on factors such as signal quality, environmental conditions, and the number of reference points available for calculation.

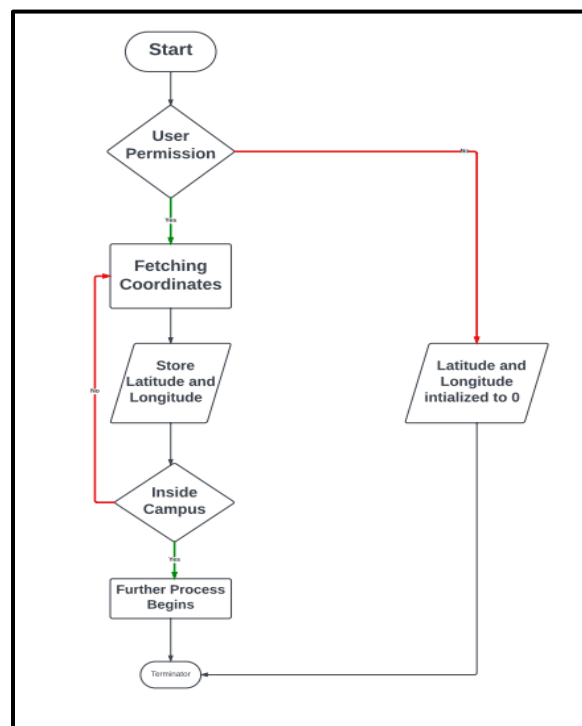


FIGURE 6. Geolocation Working

V. CONCLUSION

In conclusion, NexTrack, an innovative employee attendance system, offers a comprehensive solution for attendance tracking and management. The project's successful implementation demonstrates its potential to significantly benefit organizations and businesses in various industries. NexTrack's robust features, including two-factor authentication, facial recognition, geolocation tracking, and automated attendance calculations, enhance the accuracy and security of attendance records. The color-coded data presentation simplifies attendance monitoring, while personalized warnings for obesity add a unique health-conscious dimension.

While the future scope for NexTrack is promising, it is important to highlight its current significance. The system addresses the common challenges associated with traditional attendance tracking, providing a secure, efficient, and user-friendly solution. Continuous improvements and adaptations, such as enhanced features, advanced data analytics, and expanding into diverse industries, offer significant potential for NexTrack's evolution. Furthermore, staying aligned with data privacy regulations and ensuring robust security are key to its ongoing success.

Overall, NexTrack is a valuable addition to attendance management, enabling organizations to streamline processes, enhance security, and gain valuable insights into employee attendance patterns. Its adaptability and potential for further development make it a dynamic project with a bright future in the realm of workforce management.

ACKNOWLEDGMENTS

We First and foremost, we express our appreciation to the dedicated team of developers, designers, and testers who worked tirelessly to bring the EMS to fruition. Their expertise, creativity, and commitment to excellence were instrumental in transforming our vision into a robust and user-friendly software solution. We are also deeply thankful to our employees, whose valuable insights, feedback, and collaboration helped shape the application to meet the unique needs of our organization. Their unwavering support and enthusiasm for the project were invaluable throughout the development process. Furthermore, we would like to acknowledge the contributions of our IT infrastructure team for their efforts in setting up the necessary infrastructure and ensuring the seamless deployment of the EMS. Their expertise in server configuration, database management, and network security played a crucial role in ensuring the reliability and performance of the system.

Additionally, we express our gratitude to the leadership team for their vision, guidance, and support throughout the project. Their strategic direction and commitment to innovation have been instrumental in driving the success of the EMS implementation. Last but not least, we extend our heartfelt thanks to our vendors, partners, and external consultants who provided valuable resources, expertise, and support at various stages of the project.

In conclusion, the successful development and deployment of our Employee Management System would not have been possible without the dedication, collaboration, and support of all those involved. We are truly grateful for their contributions and look forward to the continued success and impact of the EMS in our organization. This acknowledgment recognizes the collective effort and contributions of all stakeholders involved in the EMS project, fostering a sense of appreciation and recognition for their hard work and dedication.

REFERENCES

1. K. Solanki and P. Pittalia, "Review of face recognition techniques," *Int. J. Comput. Appl.*, vol. 133, no. 12, pp. 20–24, Jan. 2016.
2. C. Stoll, R. Palluel-Germain, R. Caldara, J. Lao, M. W. G. Dye, F. Aptel, and O. Pascalis, "Face recognition is shaped by the use of sign language," *J. Deaf Stud. Deaf Educ.*, vol. 23, no. 1, pp. 1–9, 2018.
3. W. Deng, J. Hu, and J. Guo, "Face recognition via collaborative representation: Its discriminant nature and superposed representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 10, pp. 2513–2521, Oct. 2018.
4. C. Alippi, M. Bocca, G. Boracchi, N. Patwari, and M. Roveri, RTI goes wild: Radio tomographic imaging for outdoor people detection and localization, *IEEE*
5. D. Halperin, W. Hu, A. Sheth, and D. Wetherall, Tool release: Gathering 802.11N traces with channel state information, in *ACM SIGCOMM Comput. Commun.*, vol.
6. Danijel Mijić, Ognjen Bjelica, Jelena Durutović and Miloš Ljubojević, "An Improved Version of Student Attendance Management System Based on RFID", 18th International Symposium INFOTEH-IAHORINA (INFOTEH), 2019.
7. F. Adib and D. Katab, See through walls with WiFi, in *Proc. ACM SIGCOMM*, Hong Kong, China, 2013, pp. 75–86.

8. Y. Wang, K. Wu, and L. M. Ni, WiFall: Device-free fall detection by wireless networks, *IEEE Trans. on Mobile Computer*, vol. 16, no. 2, pp. 581–594, 2017.
9. Naman Gupta and Shikha Jain, "Speaker Identification Based Proxy Attendance Detection System", *International Conference on Signal Processing and Communication (ICSC)*, pp. 175-179, 2019.
10. A. U. Naik and N. Guinde, "LBPH Algorithm for Frontal and Side Profile Face Recognition on GPU", *2020 Third International Conference on Smart Systems and Inventive Technology (ICSSIT)*, pp. 776-779, 2020.
11. Partha Chakraborty, Chowdhury Shahriar Muzammel, Mahmuda Khatun, Sk. Fahmida Islam and Saifur Rahman, "Automatic Student Attendance System using Face Recognition", *International Journal of Engineering and Advanced Technology (IJEAT)*, pp. 2249-8958, February 2020.
12. Varun Garg and Kritika Garg, "Face Recognition Using Haar Cascade Classifier", *Journal of Emerging Technologies and Innovative Research (JETIR)*, vol. 3, no. 12, December 2016.
13. O. A. R. Salim, R. F. Olanrewaju and W. A. Balogun, "Class Attendance Management System Using Face Recognition", *2018 7th International Conference on Computer and Communication Engineering (ICCCE)*, pp. 93-98, 2018.
14. Z. Vantová, J. Paralič and V. Gašpar, "Mobile application for creating presence lists", *SAMI 2017 - IEEE 15th Int. Symp. Appl. Mach. Intell. Informatics Proc.*, pp. 223-228, 2017.
15. R. Ashlin linushia, S. Senthilkumar, S. Bhuvaneswari and Eric Clapton, "Smart Monitoring System using RFID Technology", *6th International Conference on Advanced Computing and Communication Systems (ICACCS)*, pp. 1430-1433, 2020.