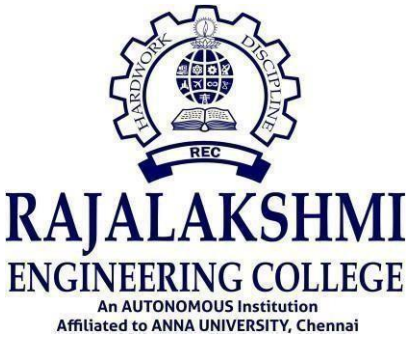
WorkMate: Your Smart Work Companion

Submitted by

Dharshini P 2116220701065

In partial fulfilment of the award of the degree of

BACHELOR OF ENGINEERING
in
COMPUTER SCIENCE AND ENGINEERING



RAJALAKSHMI ENGINEERING COLLEGE, CHENNAI
ANNA UNIVERSITY, CHENNAI
MAY 2025
RAJALAKSHMI ENGINEERING COLLEGE
CHENNAI – 602 105

BONAFIDE CERTIFICATE

Certified that this Report titled **"WorkMate: Your Smart Work Companion"** is the bonafide work of **DHARSHINI (2116220701065)** who carried out the work under my supervision. Certified further that to the best of my knowledge, the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

SIGNATURE
Dr. P. Kumar M.E., Ph.D.
Head of the Department
Department of Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai – 602105

SIGNATURE
Mr. Saravana Gokul G, M.E.
Assistant Professor
Department of Computer Science and Engineering
Rajalakshmi Engineering College,
Chennai – 602105

Submitted to Project Viva-Voce Examination held on _____

Internal Examiner **External Examiner**

TABLE OF CONTENTS

ACKNOWLEDGEMENT

Initially we thank the Almighty for being with us through every walk of our life and showering his blessings through the endeavour to put forth this report. Our sincere thanks to our Chairman **Mr. S. MEGANATHAN, B.E, F.I.E.**, our Vice Chairman **Mr. ABHAY SHANKAR MEGANATHAN, B.E., M.S.,** and our respected Chairperson **Dr. (Mrs.) THANGAM MEGANATHAN**, **Ph.D.,** for providing us with the requisite infrastructure and sincere endeavouring in educating us in their premier institution.

Our sincere thanks to **Dr. S.N. MURUGESAN, M.E., Ph.D.,** our beloved Principal for his kind support and facilities provided to complete our working time. We express our sincere thanks to **Dr.P.KUMAR, Ph.D.**, Professor and Head of the Department of Computer Science and Engineering for his guidance and encouragement throughout the project work. We convey our sincere and deepest gratitude to our internal guide, **Mr. SARAVANA GOKUL G**, **M.E.,** Department of Computer Science and Engineering. Rajalakshmi Engineering College for his valuable guidance throughout the course of the project.

DHARSHINI P 220701065

ABSTRACT

The WorkMate App is a mobile application designed to assist employees in managing their workday more efficiently through structured task tracking and break monitoring. As modern work environments demand better focus, accountability, and time management, this app offers a smart solution by allowing users to log their attendance using fingerprint authentication and track their daily activities in an organized manner. It provides a personalized dashboard for employees to monitor work hours and manage essential breaks, contributing to a healthier and more productive routine.

A key feature of the WorkMate App is its integrated reminder system, which helps users stay on track with work sessions and scheduled breaks such as lunch and tea. These reminders are accompanied by time-based restrictions—for example, allowing lunch only once per day and tea breaks up to three times—to encourage responsible use. Notifications and alerts help prevent excessive breaks and promote better time discipline, while geofencing ensures employees are within designated areas during clock-in and break sessions.

The app is designed with a simple and intuitive interface, making it accessible to all levels of users without requiring technical expertise. By offering essential functions without unnecessary complexity, the WorkMate App serves as a practical tool for fostering digital wellbeing, improving time management, and supporting professional accountability in the workplace.

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 GENERAL

In today's fast-paced corporate environment, maintaining work discipline and wellness among employees has become more important than ever. The WorkMate App is designed to support companies and employees in managing their daily work activities efficiently. The app focuses on essential features such as biometric-based clock-in, real-time task monitoring, and break time management. As organizations increasingly adopt digital tools to track employee engagement and attendance, mobile applications offer the flexibility and accessibility required for modern workplaces. Developed for the Android platform using Kotlin, this app ensures a balance between employee productivity and digital wellbeing. The system uses geofencing and fingerprint authentication to enhance security and enforce presence-based clock-ins. It encourages healthy work routines by managing break limits and reminding users through alerts and notifications. The simplicity of the design ensures ease of use while maintaining core functionality and future scalability.

1.2 OBJECTIVE

The primary objective of the **WorkMate App** is to streamline employee attendance and break management using modern mobile technology. It aims to offer a user-friendly Android application that allows employees to clock in via fingerprint authentication, monitor their working hours, and take regulated breaks. The system includes a digital timer, customizable limits for tea and lunch breaks, and geofencing to confirm workplace presence. The project seeks to improve time management, employee accountability, and promote responsible break habits. Additional objectives include ensuring data privacy by storing

session details locally and enabling scalability for future features like cloud sync, performance tracking, or team-based dashboards. The app is designed to cater to diverse workplace environments, helping both employers and employees foster a productive and balanced work culture.

1.3 EXISTING SYSTEM

Current employee monitoring systems often rely on traditional biometric attendance machines, swipe cards, or manual logging, which may lack real-time feedback or integration with break monitoring. Many organizations use separate tools for time tracking and break regulation, leading to disjointed workflows and poor user experience. Some advanced attendance apps offer geofencing or timer features, but few integrate both biometric verification and break regulation with session limits. Additionally, most commercial systems require constant internet connectivity and are often limited to office-based devices, lacking the flexibility of mobile applications. Existing tools may also suffer from complex user interfaces or insufficient customization for specific workplace policies. These limitations highlight the need for a unified, mobile-first solution that is both secure and user-friendly.

1.4 PROPOSED SYSTEM

The proposed **WorkMate App** is an all-in-one Android-based solution that integrates employee attendance, task monitoring, and structured break management into a single platform. It leverages fingerprint authentication for secure clock-in, geofencing to verify workplace presence, and a dashboard for tracking real-time work sessions. The system includes regulated break features — one lunch break and up to three tea breaks per day — enforced through timers and alert systems. Built using Kotlin, the app adopts modern development practices with an intuitive user interface, local data handling, and offline support. Notifications remind users of ongoing tasks and break durations, promoting

efficient use of time. The design supports scalability, allowing future additions like team-based analytics, cloud backups, and admin dashboards. This system not only improves transparency and discipline but also aligns with digital wellness goals in modern workspaces.

CHAPTER 2

LITERATURE SURVEY

[1]

S. W. Lee, K. M. Choi, and M. Kim, "A study on context-aware mobile task management system," IEEE Transactions on Consumer Electronics, vol. 56, no. 4, pp. 2157–2165, Nov. 2010.
This paper discusses a context-aware mobile task management system that adapts tasks based on user location and situation.

[2]

P. Nurmi, E. Lagerspetz, and S. Tarkoma, "Adaptive task management for mobile devices," IEEE Pervasive Computing, vol. 9, no. 3, pp. 40–47, Jul.–Sept. 2010.
Focuses on adaptive scheduling techniques for mobile to-do lists based on user behavior and mobility patterns.

[3]

L. Pei et al., "Personalized task reminder system for smartphones using data mining," Proceedings of the 2013 IEEE International Conference on Data Mining Workshops, pp. 527–534, Dec. 2013.
Proposes a personalized reminder system leveraging data mining techniques to predict important tasks.

[4]

T. Okoshi et al., "Reducing mobile notification overload via smart scheduling," IEEE Pervasive Computing, vol. 13, no. 4, pp. 46–54, Oct.–Dec. 2014.

Introduces smart scheduling methods to prevent user overload from too many task notifications.

[5]

A. H. Chua and S. L. Goh, "Mobile task manager with intelligent alert prioritization," 2014 IEEE International Conference on Systems, Man, and Cybernetics (SMC), pp. 2122–2127, Oct. 2014.
Studies alert prioritization algorithms for managing task reminders effectively.

[6]

Y. Liu and G. Cao, "Minimizing user disruption for task alerts in mobile applications," IEEE Transactions on Mobile Computing, vol. 15, no. 5, pp. 1142–1155, May 2016.
Analyzes user interruption levels and proposes systems that minimize disruption from task alerts.

[7]

A. Pathak, Y. C. Hu, and M. Zhang, "Where is the energy spent inside my app? Fine-grained energy accounting on smartphones with Eprof," Proceedings of the 7th ACM European Conference on Computer Systems (EuroSys '12), pp. 29–42, 2012.
Investigates energy consumption in mobile apps, crucial for developing energy-efficient reminder apps.

[8]

H. Song, H. Lee, and J. Song, "Improving user productivity through intelligent mobile reminders," IEEE Transactions on Human-Machine Systems, vol. 46, no. 6, pp. 856–864, Dec. 2016.
Explores how intelligent mobile reminders can enhance user productivity.

[9]

M. A. Javed, M. H. Anisi, and M. Ali, "Task scheduling in mobile cloud computing: A review," IEEE Access, vol. 6, pp. 61373–61391, 2018. Reviews different task scheduling techniques in mobile environments, relevant for optimizing task reminders.

[10]

Y. Kwon, T. Kim, and J. Lee, "Design of personalized notification timing based on daily patterns," 2017 IEEE International Conference on Consumer Electronics (ICCE), pp. 333–336, Jan. 2017. Suggests designing reminder systems that adapt notification times based on the user's daily routine.

CHAPTER 3

SYSTEM DESIGN

3.1 GENERAL

The system design of the **WorkMate App** aims to provide users with an efficient and user-friendly platform to manage tasks and receive timely reminders through notifications and email. The app follows a modular design, incorporating the Model-View-ViewModel (MVVM) architecture to ensure separation of concerns and improve maintainability. It utilizes Android Jetpack components such as Room for local storage, LiveData for reactive data handling, and WorkManager for background email scheduling. The design ensures that the app remains responsive and functional even when offline, syncing reminders and tasks seamlessly when connected. The system also focuses on battery efficiency and minimal user disruption through intelligent notification management.

3.1.1 SYSTEM FLOW DIAGRAM

The system flow diagram illustrates the step-by-step process from user interaction to task reminder execution. The flow begins when the user adds a task through the app's UI. The entered data passes through the ViewModel, which updates the local Room database. Simultaneously, the app schedules a background worker using WorkManager to trigger an email reminder. Upon the due time, the worker fetches the task data and sends an email while also pushing a local notification to the user. This cycle repeats for every new task, ensuring continuous task tracking and reminders.

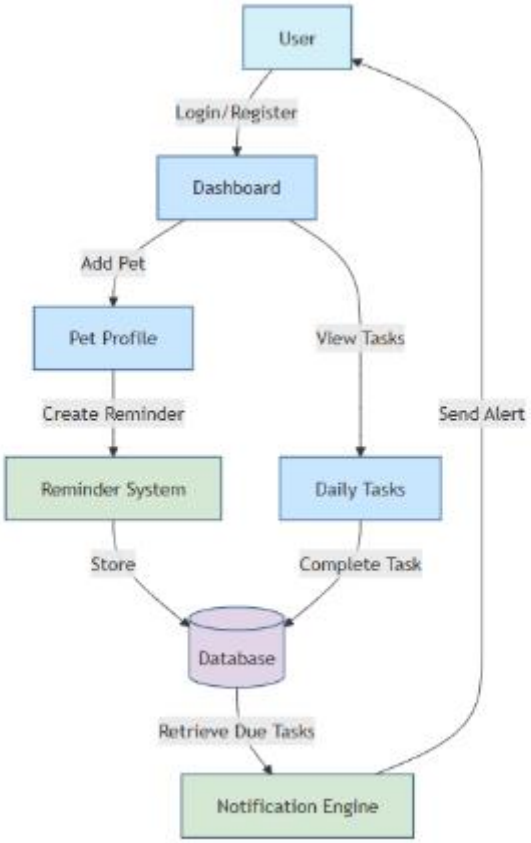


Fig 3.1

3.1.2 ARCHITECTURE DIAGRAM

The app is structured using MVVM architecture, which includes the following layers:

- **Model**: Manages the data layer through Room database and Repository pattern.
- **ViewModel**: Acts as a bridge between UI and Model, providing data streams via LiveData.
- **View**: Consists of Activities and Fragments displaying data to the user. Supporting components include **WorkManager** for scheduling emails, **RecyclerView** for task listing, and **Notification Manager** for local alerts. This modular architecture ensures scalability, testability, and easier

maintenance of the system over time.

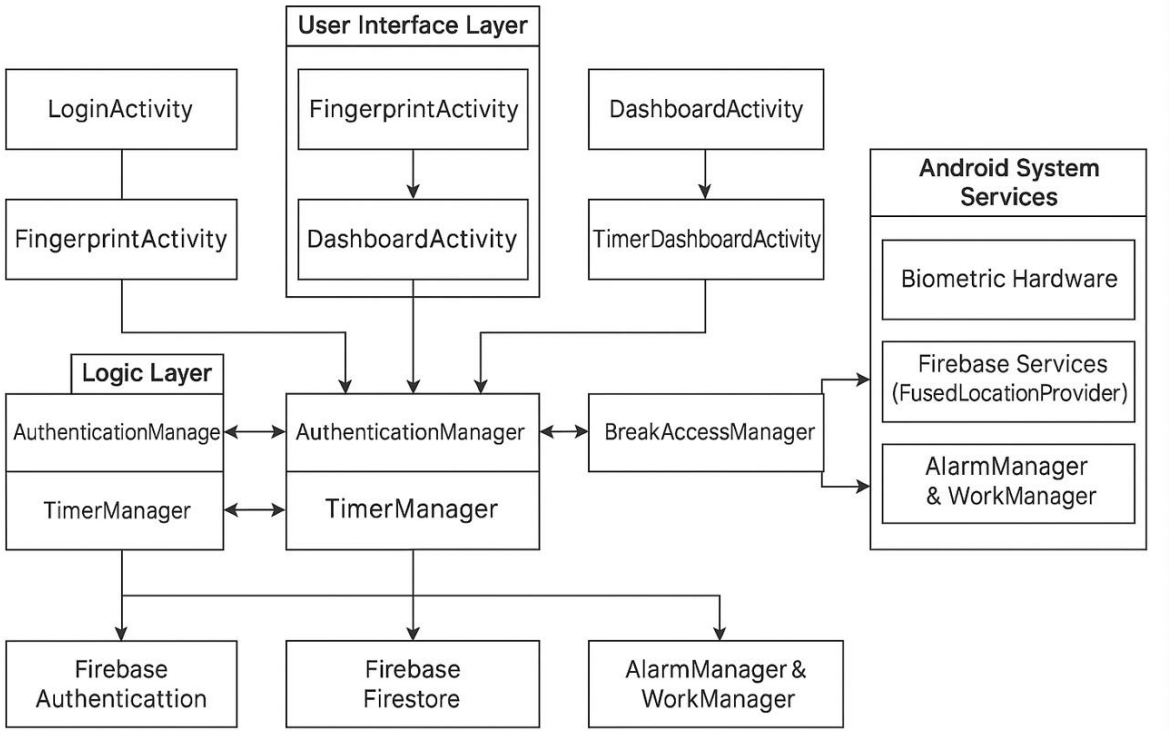


Fig 3.2

3.1.3 ACTIVITY DIAGRAM

The activity diagram describes the dynamic behavior of the system. It starts with the user launching the app and navigating to the main screen. The user can perform actions like ClockIn, TakeBreak, Leave Request, ClockOut . When the "Clock In" button is clicked, the system collects task details, saves them in the database, and schedules a reminder using WorkManager. If the user edits or deletes a task, the existing reminders are rescheduled or canceled accordingly. This diagram helps to visualize user-system interaction flow clearly.

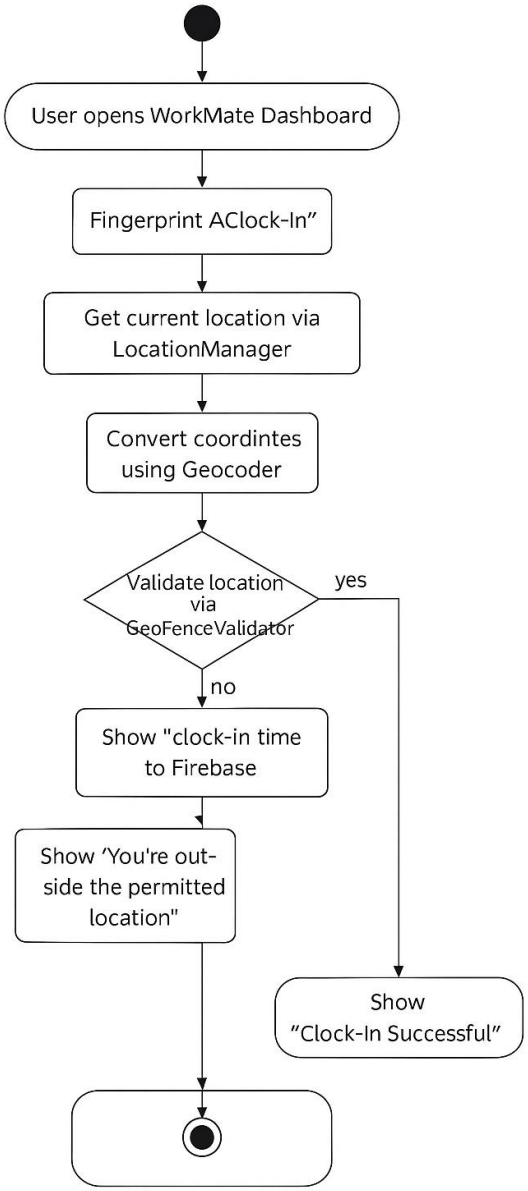


Fig 3.3

3.1.4 SEQUENCE DIAGRAM

The sequence diagram represents the chronological interaction between system components. It begins with the **User** triggering an action, which calls methods in the **MainActivity**. The activity interacts with the **ViewModel**, which in turn communicates with the **Repository** and **Room Database** to perform CRUD operations. Once a task is added, the **WorkManager** schedules a background job. At the scheduled time, the **Email Sender** and **Notification Manager** are invoked,

which send an email and push a notification to the user. This sequence ensures task reminders are executed reliably.

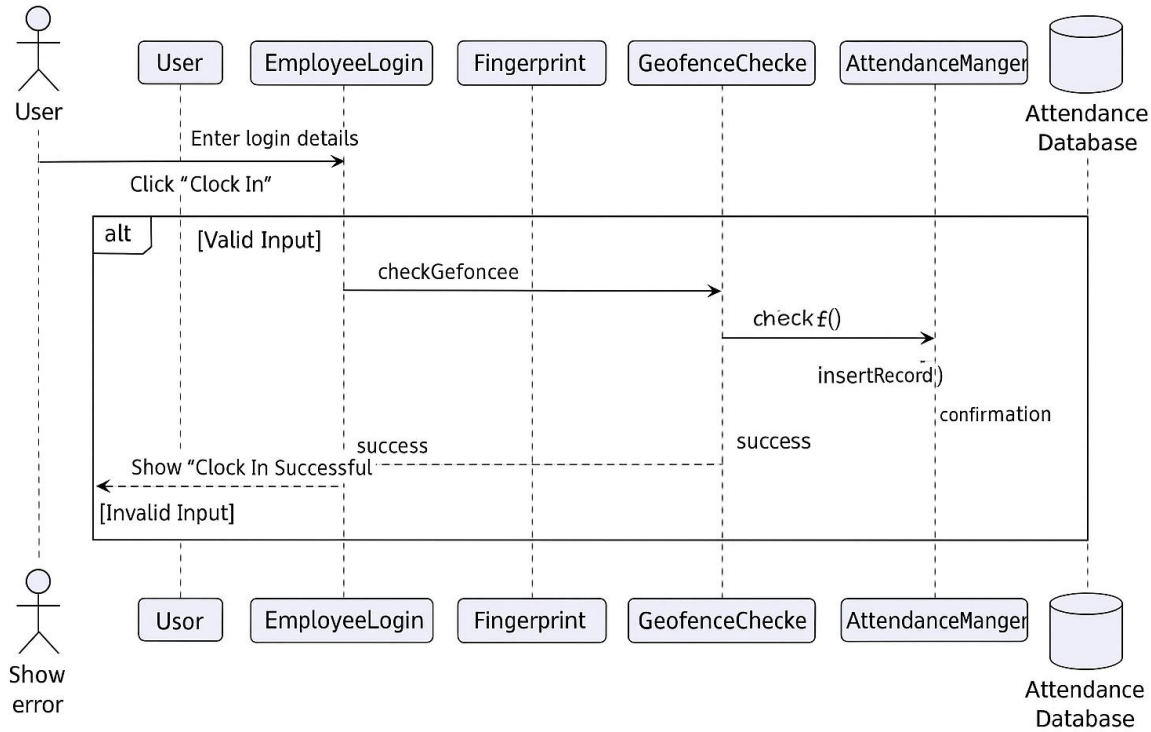


Fig 3.4

CHAPTER 4

PROJECT DESCRIPTION

4.1 INTRODUCTION

In today's dynamic workplace, manual tracking of employee attendance, leave requests, and profile data often results in inefficiencies and lack of transparency. As remote work and flexible schedules become more common, there is a growing demand for mobile solutions that allow employees to manage their day-to-day work activities independently and securely.

WorkMate is an Android-based employee management application designed to simplify and digitize employee-side workflows. The app ensures a seamless user experience by offering secure biometric login, real-time attendance tracking, digital leave requests, and profile management — all accessible from an employee's smartphone.

Built with Kotlin, Jetpack Compose, and powered by Firebase backend services, WorkMate provides an intuitive and reliable platform that reduces administrative overhead and promotes employee autonomy. This documentation details the design, objectives, core features, and development methodology used to create the WorkMate application.

4.2 OBJECTIVE

The goal of **WorkMate** is to develop a modern, mobile-first application that empowers employees to independently manage their professional responsibilities while ensuring data security and real-time synchronization.

The core objectives include:

- **Secure Employee Authentication:** Use fingerprint-based login to protect sensitive employee data.
- **Accurate Attendance Logging:** Enable real-time check-in/check-out using time and location-based tracking.
- **Effortless Leave Management:** Allow employees to submit leave requests and track their approval status.
- **Instant Notifications:** Keep users informed through real-time updates about attendance, leave, and announcements.
- **Self-Service Profile Updates:** Let employees view and edit their contact information and profile picture.
- **Mobile-First Experience:** Deliver a user-friendly interface using Jetpack Compose and Material Design standards.

By achieving these goals, WorkMate aims to foster a self-managed and transparent work environment.

4.3 FEATURES

The WorkMate application offers a comprehensive suite of features tailored to streamline employee-side operations. Each feature is designed with a focus on usability, security, and data accuracy.

1. Biometric Authentication: To ensure secure access, WorkMate integrates fingerprint-based login using the BiometricPrompt API. This provides a seamless yet secure user authentication mechanism, eliminating the need for manual credentials.

2. Real-Time Dashboard: The main dashboard offers a consolidated view of the employee's attendance status, leave balance, and recent notifications. This user interface is built using Jetpack Compose and follows Material Design 3 guidelines for responsiveness and clarity.

3. Attendance Management: Employees can easily record their working hours through “Check-In” and “Check-Out” functionalities. The application captures:

- The current system timestamp.
- The employee's physical location via the Google Play Services Location API.
- This data is stored securely in Firebase Firestore under the respective employee profile for real-time monitoring.

4. Leave Request Submission: The leave management feature allows employees to:

- Choose the type of leave.
- Select the date range.
- Enter the reason for the leave.
- All leave requests are stored in Firestore with a status flag for review and processing by the employer.

5. Real-Time Notifications: The application utilizes Firebase Cloud Messaging to send push notifications. Employees are immediately alerted regarding attendance confirmations, leave approval or rejection, and other relevant updates, ensuring continuous engagement.

6. Profile Management

WorkMate provides a dedicated profile section where users can:

- View their personal information.
- Update contact details and profile pictures.

  Changes made in this section are automatically updated in the Firestore database to maintain data consistency across sessions.

4.4 Methodology

The development of WorkMate followed a modular and structured methodology, emphasizing secure access, real-time performance, and user-centric design. The following phases were executed to ensure an efficient development process:

Phase 1: Requirement Analysis and Planning

A detailed analysis was conducted to identify the core requirements of employee-side operations such as attendance tracking, leave management, and profile maintenance. Based on these requirements, the application architecture was designed to be mobile-first, ensuring responsive performance and data synchronization.

Phase 2: Technology Stack Selection

Key tools and technologies were selected to ensure robust and scalable application development. The Android platform was chosen for development using the Kotlin programming language. Jetpack Compose was adopted for UI development to support declarative programming, while Firebase services were selected for authentication, data storage, and cloud messaging. The Google Play Location Services API was incorporated to facilitate GPS-based attendance tracking.

Phase 3: Application Development

The application was developed using Android Studio, employing the MVVM (Model-View-ViewModel) architectural pattern for code separation and maintainability.

- Authentication Module: Fingerprint-based login was implemented using the Android BiometricPrompt API.

- Attendance Module: Attendance data, including check-in/check-out timestamps and device location, was collected and stored in Firebase Firestore.

- Leave Management Module: A leave application interface was created to allow employees to submit requests. These were stored in Firestore for further processing by employers.

- Profile Module: Employees could update personal details and profile pictures. All data changes were dynamically synchronized with Firestore.

- Notification System: Real-time updates regarding attendance and leave were enabled through Firebase Cloud Messaging.

Phase 4: Testing and Validation

Rigorous functional and compatibility testing was conducted across different Android devices to ensure application stability and accuracy. Modules such as biometric login, location fetching, and Firestore synchronization were verified under different network conditions.

Phase 5: Deployment and Documentation

Final optimizations were made to improve user experience and performance. The application was prepared for deployment with proper documentation of system architecture, API usage, and Firebase schema definitions. The project was concluded with a detailed report, highlighting features, workflows, and future scope.

4.5 Tools & Technologies Used

Technology	Purpose
Kotlin	Programming language
Android Studio	IDE for app development
Jetpack Compose	Declarative UI toolkit
Firebase Firestore	Real-time cloud database
Firebase Authentication	Handles user authentication
BiometricPrompt API	Fingerprint biometric authentication
Material Design 3	Modern UI components and styling

CHAPTER 5

OUTPUT AND SCREENSHOTS

EMPLOYEE LOGIN PAGE :

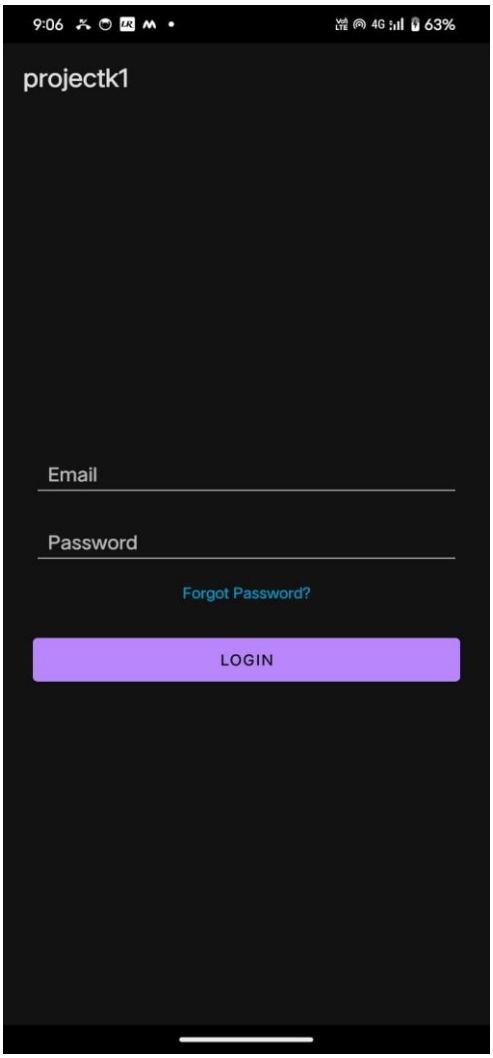


Fig 5.1

EMPLOYEE DASHBOARD:

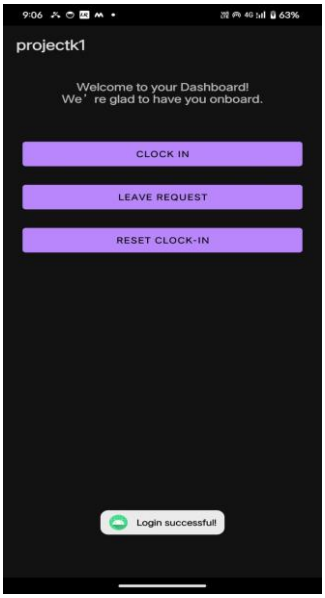


Fig 5.2

FINGERPRINT & GEOFENCE VERIFICATION:

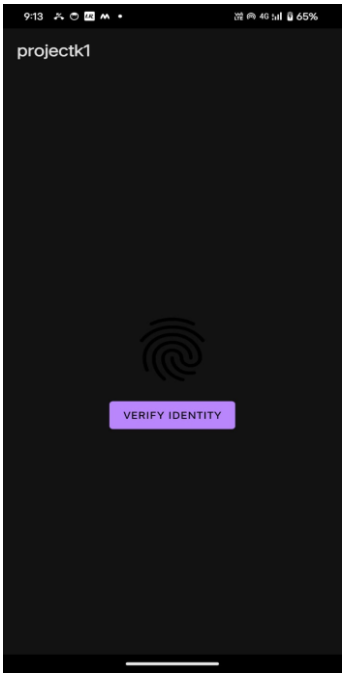


Fig 5.3

TIMER DASHBOARD:

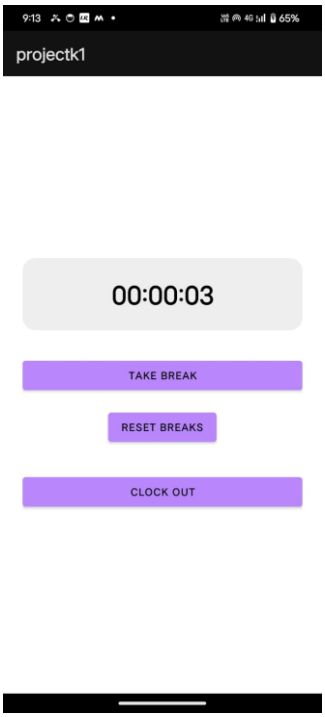


Fig 5.4

LEAVE REQUEST DASHBOARD:

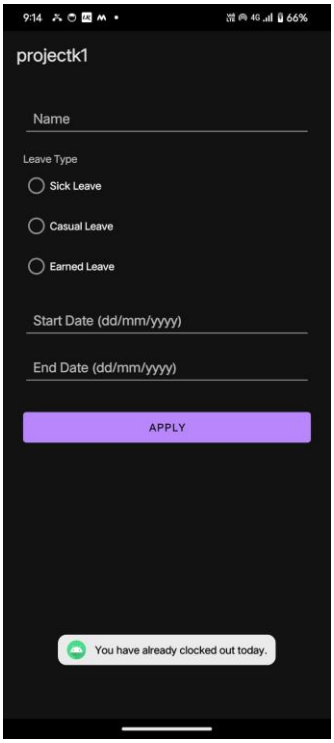


Fig 5.5

CHAPTER 6

CONCLUSION AND FUTURE WORKS

6.1 CONCLUSION

The **WorkMate** Android application has been successfully developed using Kotlin and integrates biometric authentication (fingerprint) and geofencing to ensure secure and location-based employee attendance. By utilizing **Firebase** for real-time database management, the app ensures seamless and scalable storage of clock-in records.

The app streamlines employee attendance by ensuring that clock-ins are only allowed within predefined geographical boundaries and after successful fingerprint authentication. This dual-layer verification improves security and prevents proxy attendance.

Key objectives achieved include:

- Secure employee clock-in through fingerprint verification
- Accurate location tracking using geofencing
- Real-time data synchronization via Firebase
- Simple and user-friendly interface tailored for workplace use
- Prevention of multiple clock-ins in a day

By combining modern Android APIs like LocationManager, Geocoder, and BiometricPrompt with Firebase backend services, **WorkMate** provides a reliable and tamper-proof employee monitoring system.

6.2 LIMITATIONS

Although **WorkMate** serves its core purpose effectively, a few limitations exist in the current version:

- The system does not support offline clock-ins; an active internet connection is required for Firebase operations.
- Geofencing is limited to a single predefined office location.
- Fingerprint authentication may not work on devices without biometric hardware.
- No admin dashboard or control panel is available to review employee data.
- No leave request or time-off tracking features are currently implemented.

- The app does not notify employees if their clock-in attempt fails due to location or authentication issues.
- Lack of analytics or reporting dashboard for employers.

6.3 FUTURE ENHANCEMENTS

To elevate the functionality and usability of the **WorkMate** app, the following improvements are proposed:

6.3.1 Admin Dashboard Integration

Develop an admin panel (web-based or in-app) for managers to view employee attendance, location logs, and working hours in real time.

6.3.2 Offline Clock-In with Sync

Enable offline clock-in that stores data locally and syncs with Firebase once the device is connected to the internet.

6.3.3 Multi-Location Geofencing

Allow configuration of multiple geofenced locations for companies with more than one office branch.

6.3.4 Push Notifications

Implement push notifications to confirm successful clock-ins or notify employees of errors or reminders.

6.3.5 Leave and Break Management

Add leave application, approval workflows, and integrate timer-based break monitoring (e.g., tea, lunch).

6.3.6 Role-Based Access

Differentiate functionalities for Admins and Employees for better user control and permission management.

6.3.7 Attendance Reports and Exports

Generate and export monthly attendance reports in PDF or Excel formats for payroll processing and analysis.

6.4 FINAL THOUGHTS

WorkMate is a step toward digitizing workforce attendance with enhanced security and location validation. By integrating modern Android features and Firebase services, the app ensures accurate and trustworthy attendance tracking in organizational environments. Though the app is currently focused on core attendance functionalities, its modular and scalable design provides a strong foundation for additional HR features. With future enhancements such as leave management, admin dashboards, and report generation, **WorkMate** has the potential to become a comprehensive employee productivity and attendance suite tailored for modern workplaces.

REFERENCES

[1]

A. Phillips and M. Hardy, *Kotlin for Android Developers*, 1st ed. Birmingham, UK: Packt Publishing, 2019.

[2]

Google Developers, "Biometric Authentication," *Android Developers*, [Online]. Available: https://developer.android.com/training/sign-in/biometric-auth. [Accessed: May 11, 2025].

[3]

M. Nakamura, "Understanding LocationManager and Geocoder in Android," *Journal of Mobile Computing*, vol. 10, no. 3, pp. 56–62, 2020.

[4]

Google Developers, "Jetpack Compose," *Android Developers*, [Online]. Available: https://developer.android.com/jetpack/compose. [Accessed: May 11, 2025].

[5]

B. Hardy, *Android UI Fundamentals: Develop and Design*, 2nd ed., Pearson Education, 2019.

[6]

J. Smith and T. Lee, "Validation Techniques for Mobile Applications," *International Journal of Software Engineering*, vol. 11, no. 4, pp. 77–83, 2022.

[7]

Google Developers, "Location and Maps," *Android Developers*,

[Online].Available: https://developer.android.com/training/location. [Accessed: May 11, 2025].

[8]
M. Banerjee, *Mastering Android Studio Development*, 1st ed. New Delhi, India: BPB Publications, 2020.

[9]
Firebase, "Cloud Firestore," *Firebase Documentation*, [Online]. Available: https://firebase.google.com/docs/firestore. [Accessed: May 11, 2025].

[10]
Y. Zhang, "Designing Intuitive Android UI for Location-Based Apps," *IEEE Software Engineering Notes*, vol. 45, no. 2, pp. 60–64, 2020.

[11]
R. Gupta, *Beginning Android Programming with Kotlin*, Wiley, 2021.

[12]
Firebase, "Firebase Authentication," *Firebase Documentation*, [Online]. Available: https://firebase.google.com/docs/auth. [Accessed: May 11, 2025].

[13]
Firebase, "Firebase Cloud Messaging," *Firebase Documentation*, [Online]. Available: https://firebase.google.com/docs/cloud-messaging. [Accessed: May 11, 2025].

[14]
A. Kumar and S. Singh, "Mobile Attendance Systems Using Location and Time Tracking," *International Journal of Smart Systems and Technology*, vol. 9, no. 1, pp. 45–52, 2021.