**MOBILE APPLICATION CAMERA SYSTEM TO MONITOR RESIDENTIAL SOCIETIES VEHICLE ACTIVITY**

## A PROJECT REPORT

***Submitted by,***

**Mr. GONUGUNTLA VENKATA SIVA NAGA BALA VIVEK - 20201CAI0201**

**Mr. SUHAS C SHETTY -20201CAI0169**

**Mr. RAHUL MADHUSUDHAN – 20201CAI0206**

### *Under the guidance of,*

**Ms. JOSEPHINE R**

**Assistant Professor**

***in partial fulfillment for the award of the degree of***

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**At**



**PRESIDENCY UNIVERSITY**

**BENGALURU**

**JANUARY 2024**

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE**

**CERTIFICATE**

This is to certify that the Project report **“MOBILE APPLICATION CAMERA SYSTEM TO MONITOR RESIDENTIAL SOCIETIES VEHICLE ACTIVITY”** being submitted by “GONUGUNTLA VENKATA SIVA NAGA BALA VIVEK, SUHAS C SHETTY, RAHUL MADHUSHUDHAN” bearing roll number(s) “20201CAI0201, 20201CAI0169, 20201CAI0206” in partial fulfilment of requirement for the award of degree of Bachelor of Technology in Computer Science and Engineering is a bonafide work carried out under my supervision.

|  |  |
| --- | --- |
| **Ms. JOSEPHINE R**  Assistant Professor  School of CSE&IS  Presidency University | **Dr. Zafar Ali Khan**  Associate Professor & HOD  School of CSE&IS  Presidency University |
|  |  |

|  |  |  |
| --- | --- | --- |
| **Dr. C. KALAIARASAN**  Associate Dean  School of CSE&IS  Presidency University | **Dr. L. SHAKKEERA**  Associate Dean  School of CSE&IS  Presidency University | **Dr. SAMEERUDDIN KHAN** Dean  School of CSE&IS  Presidency University |

**PRESIDENCY UNIVERSITY**

**SCHOOL OF COMPUTER SCIENCE ENGINEERING & INFORMATION SCIENCE**

**DECLARATION**

We hereby declare that the work, which is being presented in the project report entitled **MOBILE APPLICATION CAMERA SYSTEM TO MONITOR RESIDENTIAL SOCIETIES VEHICLE ACTIVITY** in partial fulfilment for the award of Degree of **Bachelor of Technology** in **Computer Science and Engineering**, is a record of our own investigations carried under the guidance of **Ms.JOSEPHINE R Assistant Professor School of Computer Science Engineering & Information Science, Presidency University, Bengaluru.**

We have not submitted the matter presented in this report anywhere for the award of any other Degree.

|  |  |  |
| --- | --- | --- |
| STUDENT NAME | ROLL NUMBER | SIGNATURE |
| GONUGUNTLA  VENKATA SIVA NAGA BALA VIVEK | 20201CAI0201 |  |
| SUHAS C SHETTY | 20201CAI0169 |  |
| RAHUL MADHUSHUDHAN | 20201CAI0206 |  |

**ABSTRACT**

In today's rapidly evolving urban landscape, ensuring the security and efficient management of residential societies has become a paramount concern. This abstract introduces a comprehensive solution in the form of a Mobile Application Camera System designed to monitor and enhance the oversight of vehicle activity within residential communities. The system integrates cutting-edge technologies to provide residents, security personnel, and community administrators with real-time insights into vehicular movements, fostering a safer and more organized living environment. The process involves several key stages, starting with the capture of still images as input. These images are then subjected to a license plate extraction process, which is crucial for obtaining the plate number. The system utilizes plate features, such as texture analysis, to isolate and extract the license plate from the input image Once the license plate is successfully extracted, the system proceeds to the character segmentation stage. In this phase, individual characters are separated from the license plate image. This segmentation process is essential for ensuring accurate character recognition in the subsequent stages.The Mobile Application Camera System for Monitoring Residential Societies' Vehicle Activity presents an innovative and integrated approach to address the security and operational challenges faced by modern residential communities. By leveraging mobile technology and smart cameras, the system aims to create a safer, more transparent, and well-managed living environment for residents while empowering security personnel with advanced tools for effective monitoring

**ACKNOWLEDGEMENT**

First of all, we indebted to the **GOD ALMIGHTY** for giving me an opportunity to excel in our efforts to complete this project on time.

We express our sincere thanks to our respected dean **Dr. Md. Sameeruddin Khan**, Dean, School of Computer Science Engineering & Information Science, Presidency University for getting us permission to undergo the project.

We record our heartfelt gratitude to our beloved Associate Deans **Dr. Kalaiarasan C and Dr. Shakkeera L,** School of Computer Science Engineering & Information Science, Presidency University and **Dr. Zafar Ali Khan** Head of the Department, School of Computer Science Engineering & Information Science, Presidency University for rendering timely help for the successful completion of this project.

We are greatly indebted to our guide **Ms. Josephine R** School of Computer Science Engineering & Information Science, Presidency University for her inspirational guidance, and valuable suggestions and for providing us a chance to express our technical capabilities in every respect for the completion of the project work.

We would like to convey our gratitude and heartfelt thanks to the University Project-II Coordinators **Dr. Sanjeev P Kaulgud, Dr. Mrutyunjaya MS** and also the department Project Coordinators **Dr. Mulari Paremeswaran**

We thank our family and friends for the strong support and inspiration they have provided us in bringing out this project.

**Gonuguntla Venkata siva naga bala vivek**

**Suhas c shetty**

**Rahul Madhushudhan**

**LIST OF TABLES**

|  |  |  |
| --- | --- | --- |
| Table NAME | Table Caption | Pg no |
| 6.1 | ER diagram | 26 |

**TABLE OF CONTENTS**

**PAGE**

**ABSTRACT…………………………………………………………………………………………. vi**

**ACKNOWLEDEMENT……………………………………………………………………………. viii**

**LIST OF TABLES………………………………………………………………………………….. ix**

CHAPTER -1 INTRODUCTION……………………………………………………………. 1

CHAPTER -2 LITERATURE SURVEY……………………………………………………. 2

CHAPTER -3 PROPOSED METHOD……………………………………………………… 9

3.1 Exiting Method

3.2 Proposed Method

CHAPTER -4 METHODOLOGY…………………………………………………………… 15

4.1 Flutter

4.2 Drat

4.3 object Modules

CHAPTER -5 OBJECTIVES …………………………………………………………………. 23

CHAPTER -6 SYSTEM DESIGN & IMLEMENTATION………………………………….. 25

6.1 System Specification

CHAPTER -7 TIMELINE FOR EXECUTION OF PROJECT……………………………….. 28

(GANTT CHART)

CHAPTER -8 OUTCOMES………………………………………………………………….. 29

8.1 Enhanced Security

8.2 Effective Resolution of Query

8.3 Privacy and Ethical Compliance

8.4 Efficient User Education

CHAPTER -9 RESULT AND DISCUSSIONS………………………………………………. 32

9.1 User Engagement and Interaction

9.2 Relevance and Quality of Response

9.3 Effectiveness in Resolving Queries

9.4 Flexibility and Acquiring Proficiency

9.5 Ethical Compliance and Privacy

CHAPTER -10 CONCLUSION…………………………………………………………………. 34

REFERENCES ………………………………………………………………………………………35

APPENDIX…………………………………………………………………………………………..36

**CHAPTER-1**

**INTRODUCTION**

In the contemporary context of residential societies, the effective management and security of vehicular activities have emerged as critical concerns. The escalating number of vehicles within these communities necessitates innovative approaches that go beyond conventional surveillance and control methods. To tackle this challenge, a cutting-edge Mobile Application Camera System (MACS) has been developed, aiming to revolutionize the monitoring and administration of vehicle movement within residential areas.

This advanced system seamlessly integrates state-of-the-art camera technology with the convenience of a mobile application, providing residents and administrators with real-time insights into the dynamics of vehicle movement within the community. By capturing and analyzing still images at key entry and exit points, the system extracts crucial data such as license plate information and vehicle types. Intelligent algorithms then process this data, facilitating continuous monitoring and control.

The primary objective of the MACS is to establish a comprehensive and user-friendly platform. Residents can easily register their vehicles through the mobile application, streamlining the access permission process. Additionally, the system delivers real-time notifications to residents, enhancing their awareness and sense of security regarding their vehicle's movement within the residential society. Administrators, equipped with a centralized dashboard accessible through both the mobile application and a web interface, gain a holistic view of vehicular activities.

The dashboard provides detailed entry and exit logs, enabling administrators to promptly identify and address unauthorized vehicles. Furthermore, the system facilitates efficient data analysis and reporting, empowering administrators to make well-informed decisions related to community planning and security measures.

**CHAPTER-2**

**LITERATURE SURVEY**

1. **Affordable Mobile Application to Monitor Residential Society's Vehicle Activity.2019, IOSR Journal of Engineering (IOSRJEN)**

This cost-effective mobile application designed for residential societies takes advantage of the widespread use of smartphones and the capabilities of modern technology to streamline the monitoring process. Leveraging the built-in cameras of smartphones, residents can effortlessly capture and upload images of vehicles entering or exiting the society directly through the application.

Intelligent algorithms incorporated into the application extract pertinent information from the uploaded images, including license plate details. This data is then systematically organized and stored in a centralized database, establishing a comprehensive log of vehicular activities. Residents have easy access to this information through the application, fostering transparency and encouraging community engagement.

In essence, administrators, equipped with a dedicated dashboard, gain valuable insights into the overall vehicular movement within the society. The dashboard offers a consolidated view of entry and exit logs, allowing administrators to discern patterns, manage permissions, and promptly address any irregularities. The affordability of this mobile application ensures that even smaller residential societies can benefit from heightened security measures.

**2.Mobile Based Application to Scan the Number Plate and To Verify the Owner Details**

***International Journal of Engineering Science Invention ISSN (Online): 2319 – 6734, ISSN (Print): 2319*-*6726***

The License Plate Recognition (LPR) system typically involves a three-step image processing workflow: 1) Extraction of the license plate region; 2) Segmentation of the plate characters; and 3) Recognition of each character. Recent times have seen the introduction of several algorithms aimed at optimizing the efficiency of this application. The core objective of this project was to develop a real-time application specifically designed to recognize number plates from cars, particularly at entry points like the entrance of parking areas or border crossings.

Operational on a standard PC with a mobile camera, the system captures video frames containing visible car number plates and processes them in real time. The three essential steps in the image processing pipeline are as follows:

Extraction of License Plate Region:

The system identifies and isolates the region containing the license plate within the video frames.

This initial step focuses on pinpointing the area of interest, i.e., the number plate, for subsequent analysis.

Segmentation of Plate Characters:

Once the license plate region is isolated, the system proceeds to segment the characters on the plate.

Character segmentation involves breaking down the alphanumeric characters into individual units, preparing them for recognition.

Recognition of Each Character:

The segmented characters undergo a recognition process where algorithms interpret and identify each character.

Advanced techniques, such as pattern recognition and machine learning, may be employed to enhance accuracy.

Utilizing a standard PC and a mobile camera for video frame capture, the system efficiently processes data in real time. This makes it adept at recognizing number plates at entry points like parking areas or border crossings, contributing to enhanced security. The system's adaptability and accessibility across various scenarios underscore its practicality in real-world applications.

**3.An automated license plate detection and recognition system based on wavelet decomposition and CNN**

The License Plate Detection and Recognition (LPDR) task poses a significant challenge and holds a crucial role in Intelligent Transport Systems (ITS), serving as a core component in various applications like security, traffic control, and electronic payment systems (e.g., freeway toll payment and parking fee payment). Several algorithms have been developed for LPDR, each with its set of advantages and disadvantages depending on the circumstances of plate extraction in images.

However, the computational complexity of some methods often leads to high calculation costs and time-consuming processes. In this paper, a straightforward and efficient method is proposed to address the challenges associated with license plate detection and character recognition.

The proposed approach begins by detecting the license plate, leveraging the two-dimensional wavelet transform to extract vertical edges from the input image. The high density of vertical edges is initially computed to identify potential areas of the license plate. Subsequently, these potential areas undergo verification using a plate/non-plate Convolutional Neural Network (CNN) classifier.

Once the license plate is successfully detected, a simple segmentation method based on the empty distance between characters is employed to segment the characters. Finally, these character candidates are classified using another CNN classifier, which has been trained for this specific purpose.

In summary, the proposed method addresses the challenges of LPDR by employing a two-dimensional wavelet transform for license plate detection, followed by verification through a plate/non-plate CNN classifier. The subsequent character segmentation and classification processes are executed through straightforward methods, ensuring an efficient and effective solution for license plate recognition in various practical applications within ITS.

**4.Real Time Detection and Recognition of License Plate in Bengali**

Automatic license plate detection and recognition find extensive applications, and numerous approaches have been put forth to enhance the efficiency of this process. However, there is a noticeable gap in research concerning Bengali license plate recognition, attributed to the distinctive characteristics of Bengali characters, complex backgrounds, and diverse license plate patterns in Bangladesh. Existing algorithms may not prove effective due to these challenges. This paper presents a tailored solution for the detection and recognition of Bengali license plates.

Our approach involves three stages typically found in conventional license plate recognition systems. However, we introduce new algorithms in each stage, specifically designed to address the unique challenges associated with Bengali license plate detection and recognition.

The complexity arising from the wide variation in license plate patterns and the difficulty in segmenting Bengali characters necessitate innovative solutions. Our proposed algorithms aim to overcome these challenges, providing an effective framework for Bengali license plate recognition.

In practical testing, we evaluated our algorithms using more than 250 images captured from the road. The results demonstrate a remarkable success rate, with over 95% accuracy in Bengali license plate recognition. This success underscores the efficacy of our tailored approach in addressing the complexities associated with Bengali characters and diverse license plate patterns, making a valuable contribution to the field of license plate recognition in the context of the Bengali language.

**5. Car License Plate Detection Using Edge and Color.**

**World Applied Sciences Journal 5 (6): 719-730, 2008**

This paper addresses the challenges associated with license plate detection, particularly in scenarios with low-quality images due to issues such as severe illumination conditions, vehicle motion, viewpoint and distance changes, and complex backgrounds. To mitigate these challenges, we propose two distinct image enhancement methods, leveraging intensity variance and edge density. The primary objective is to enhance the contrast of plate-like regions, thereby preventing the oversight of plate locations, especially in poor-quality images.

Additionally, we introduce a novel match filter designed for the detection of candidate regions as potential license plates. This filter is tailored to model the vertical edge density of the plate region in consideration of its neighborhood. The simplicity of the filtering process positions this approach for real-time applications. Furthermore, we incorporate colored texture within the plate as a cue for detection, a feature that remains robust under viewpoint changes. The characterization of color information in the plate is achieved through the utilization of the Multimodal Neighborhood Signature (MNS) method.

In summary, we present a comprehensive database that can facilitate a precise evaluation of our proposed method and any related works. Experimental results, conducted on various types of car images in complex scenes, affirm the robustness of our approach against severe imaging conditions. The combination of image enhancement, match filtering, and color-based texture cues contributes to the effectiveness of our method, making it resilient in challenging scenarios.

**6.** **Character Segmentation Method for a License Plate with Topological Transform**

This paper introduces a robust method for character segmentation in license plates, specifically designed to handle topological transformations like twist and rotation. The initial stage of the proposed method involves identifying candidate regions for characters and license plates. A critical criterion for these regions is that characters or license plates must manifest as closed loops in the edge image.

In the detection of candidate regions for characters, the assessment of identified regions relies on the topological relationships between each character. Conversely, when determining license plate candidate regions, character features within the region are utilized following binarization.

Following binarization of the detected candidate region, a more refined determination of each character region takes place. This step involves a more precise fitting process compared to the initial stage. Subsequently, the method scrutinizes other character regions with varying scales in proximity to the detected character regions. This is based on the observation that most license plates incorporate meaningful characters surrounding the license numbers. The method employs perspective projection for geometrical normalization in this step.

In summary, the proposed method ensures robust character segmentation in license plates, even when subjected to topological transformations. The distinctive approach of leveraging topological relationships, binarization, and perspective projection contributes to the method's resilience and accuracy in identifying and segmenting characters within license plates.

**7. A License Plate-Recognition Algorithm for Intelligent Transportation System Applications**

This paper introduces a novel algorithm for vehicle license plate identification, utilizing an innovative adaptive image segmentation technique known as sliding concentric windows. Connected component analysis, in combination with a character recognition neural network, is employed in the proposed algorithm. A comprehensive testing phase involved 1334 natural-scene gray-level vehicle images, featuring diverse backgrounds and ambient illumination. The camera focused on the license plate, with varying angles of view and distances from the vehicle in accordance with the experimental setup.

Out of the 1334 input images, the algorithm successfully segmented 1287 license plates (96.5%). The optical character recognition system adopted a two-layer probabilistic neural network (PNN) with a topology of 108-180-36. The PNN exhibited a recognition performance of 89.1% for entire plate recognition. Trained on data obtained from algorithmic image processing, the PNN identifies alphanumeric characters from car license plates.

Combining the segmentation and recognition rates, the overall success rate of the license plate recognition algorithm reached 86.0%. A comprehensive review of related literature presented in this paper reveals that superior performance (ranging from 90% to 95%) has been reported under specific conditions, including limitations in distance, angle of view, illumination conditions, and low background complexity.

**8.Automatic Vehicle Number Plate Extraction And Maintenance System Using OCR Algorithm**

The system is engineered to monitor vehicle movements on both NH and city roadways. Images captured during entry and exit will undergo processing using Optical Character Recognition (OCR) for automated number plate recognition. The recognized car numbers will be logged into a database.

In cases where a centralized receiver is in place, all entrance records are stored and scrutinized for any stolen vehicles entering NH highways. Upon detection, the system captures the vehicle picture and relevant information, forwarding it to an authorized person through Internet Message Access Protocol (IMAP).

The OCR technology plays a pivotal role in identifying the number plates. Additionally, the system cross-references the entered car information with the existing database, retrieving and presenting it if a match is found. This integrated approach enhances the efficiency of monitoring and managing vehicular activities on both NH and city roadways, while the system's ability to identify stolen vehicles contributes to heightened security measures.

**9. CHARACTER RECOGNITION FROM SCANNED IMAGES: A REVIEW**

Optical Character Recognition (OCR) algorithms applied to images typically involve three fundamental steps: firstly, the extraction of lines; secondly, the segmentation of characters from the identified lines; and finally, the recognition of each individual character. This task presents considerable challenges due to the wide diversity of fonts and formats, as well as the non-uniform outdoor illumination conditions prevalent during image acquisition.

Numerous techniques have been developed for character detection, and the objective of this paper is to systematically categorize these techniques. Additionally, the paper delves into various related aspects such as topologies, processing speed, and preprocessing methods. A comprehensive comparison of different research endeavors is provided, encompassing their respective recognition accuracies. This systematic approach contributes to a better understanding of the landscape of OCR algorithms in image processing.

**10. Deep Learning Based Automatic Vehicle License Plate Recognition System for Enhanced Vehicle Identification**

Introducing an innovative Automatic Vehicle License Plate Recognition (AVLPR) system designed to accurately identify vehicles through the implementation of deep learning algorithms. The demand for enhanced security and efficient traffic management has underscored the significance of precise and real-time license plate identification.

The AVLPR system employs a Convolutional Neural Network (CNN) architecture, enabling the model to autonomously learn and extract discriminative features from license plate photos. To ensure the system's robustness and adaptability, the training and validation dataset encompasses a diverse array of license plate designs, fonts, and lighting conditions. Data augmentation techniques are incorporated throughout the training process to account for variations in license plate orientation, scale, and perspective, thereby improving recognition accuracy.

Additionally, transfer learning is utilized to enhance the system's generalization capabilities by fine-tuning the pre-trained model on a sizable dataset. The resulting Deep Learning-Based Automatic Vehicle License Plate Recognition System offers a reliable and effective solution for vehicle identification tasks.

This advanced system, equipped with deep learning approaches, ensures precise and instantaneous recognition, making it suitable for various applications such as law enforcement, parking management, and intelligent transportation systems.

The Watchful Eye: A sophisticated mobile camera securely mounted on a pole serves as a vigilant guard, capturing clear images of vehicles entering or exiting the area.

More Than Just Watching: This system is not merely a passive observer. The accompanying app allows users to zoom in on suspicious activity, capture snapshots for evidence, or record video clips for later review. No unauthorized visitor will go unnoticed.

Smarter than the Average Camera: The "Society Resident" camera goes beyond ordinary surveillance. It can be programmed to send alerts for unusual occurrences, such as unfamiliar cars entering after dark. It functions like a personal security guard, providing timely notifications.

Memory Like an Elephant: Users need not worry about missing important details, as the app allows for the review of past recordings, ensuring comprehensive coverage and monitoring.

**CHAPTER-3**

**PROPOSED METHOD**

**3.1Existing Method**

**Traditional CCTV Systems:**

**Camera Surveillance:**

Many residential societies deploy traditional Closed-Circuit Television (CCTV) systems to monitor entry and exit points.

Cameras are strategically placed to capture vehicle movements and are connected to a central monitoring station.

Manual Monitoring:

Security personnel manually monitor the live feed from cameras, keeping an eye on vehicles entering and exiting society.

**RFID and Access Control Systems:**

**RFID Tags:**

Residential societies often use Radio-Frequency Identification (RFID) tags for registered vehicles

RFID readers at entry/exit points detect authorized vehicles, allowing seamless access.

**Access Control Systems:**

Secure access control systems, such as key cards or mobile app-based access, are employed to manage and monitor vehicle movements.

* 1. **Proposed Method**

At the core of our innovative method lies the implementation of advanced image processing algorithms tailored to identify and classify vehicles entering and exiting residential societies. The primary focus is on accurately distinguishing between resident and non-resident vehicles, prioritizing real-time processing for instantaneous results and timely decision-making.

To establish seamless communication between the mobile application and the camera system, we propose the integration of Firebase services. Firebase Realtime Database will be employed for efficient data synchronization, ensuring the mobile app's information is consistently up-to-date. Firebase Cloud Messaging (FCM) will facilitate push notifications, enabling timely alerts to be sent to residents and security personnel upon vehicle entry and exit.

A sophisticated notification system will be developed to inform residents and security personnel about vehicle activities, offering customizable alerts to cater to individual preferences. Additionally, an analytics dashboard will be crafted to provide security personnel with valuable insights, including analytics, logs, and trends related to vehicle movements within the residential society.

Comprehensive testing, including rigorous validation of vehicle identification and notification systems, will be conducted to ensure the reliability and accuracy of the proposed solution. User acceptance testing involving residents and security personnel will be an integral part of the iterative development process, allowing for continuous improvement based on real-world feedback.

The image processing algorithms will be fine-tuned to implement an intelligent classification system. Machine learning models, potentially trained with historical data, will contribute to accurate identification of vehicles as resident or non-resident. This dynamic classification will adapt over time, enhancing the system's ability to respond to changing resident demographics and visitor patterns.

The notification system will be designed to provide adaptive alerts based on the time of day, historical data, and user preferences. For example, during late-night hours, security alerts might be prioritized, whereas during the day, residents may receive more informative notifications about guest arrivals. This adaptability aims to streamline communication and ensure that users receive relevant information without unnecessary disruptions.

The proposed method will undergo a pilot deployment in a residential society, serving as a practical testing ground for the solution. Feedback collected during this phase will inform necessary adjustments and improvements to enhance the system's performance, scalability, and overall user experience. Ongoing maintenance and updates will be provided to address any issues that may arise and introduce new features, ensuring the sustained effectiveness of the proposed mobile application camera system.

**1.System Architecture**

**Goal: Create Mobile Application Camera System to Monitor Residential Societies Vehicle Activity**

**Architecture of the System Diagram:** key components and features that should be included in the UI design

User Authentication:

Implement a secure login or signup screen to authenticate users, ensuring a protected entry point to the application.

Dashboard:

Develop a user-friendly dashboard that provides real-time updates, offering an intuitive and comprehensive overview of relevant information.

Alerts and Notifications:

Design an efficient notification system that conveys alerts and updates seamlessly. Employ a visually appealing interface to enhance user awareness.

User Permissions:

Create a role-based access control system within the UI to manage user permissions effectively. This feature ensures that users have appropriate access based on their roles, enhancing security and data integrity.

Additionally, the system integrates with an API for model calls and utilizes a web server through ngrok, enhancing the application's functionality and connectivity. These components collectively contribute to a robust and user-centric UI design, ensuring a secure, informative, and accessible experience for the users.

**2.Integration of OpenCV Object detection:**

**Tasks:** Integrating OpenCV object detection into a mobile application camera system for monitoring residential societies' vehicle activity involves incorporating advanced object detection capabilities. The following steps outline the process:

Select OpenCV Object Detection Implementation:

Choose the appropriate version or implementation of OpenCV for the project, ensuring compatibility with the mobile application and meeting specific requirements.

Train the Model Using Dataset:

Train the OpenCV object detection model using a carefully curated dataset that encompasses diverse scenarios, lighting conditions, and vehicle types to enhance the model's accuracy.

Integrate OpenCV into Mobile Application:

Seamlessly integrate the trained OpenCV object detection model into the mobile application, ensuring optimal performance and responsiveness.

Capture Frames from Camera Feed:

Implement a mechanism to capture frames from the camera feed in real-time, allowing the OpenCV object detection model to analyze and identify vehicles' number plates.

Process Frames through OpenCV Object Detection:

Process the captured frames through the OpenCV object detection model to accurately identify and locate vehicle number plates within the residential society.

Display Results on User Interface:

Showcase the results of the OpenCV object detection on the user interface of the mobile application, providing users with real-time insights into vehicle activities within the community.

This approach utilizing OpenCV ensures robust object detection capabilities within the mobile application, contributing to effective monitoring of residential societies' vehicular movements. The steps outlined emphasize the integration of OpenCV seamlessly into the application's workflow for optimal performance and user experience.

**4. Machine Learning for Generation number plate:**

**Goal:** Use machine learning models to improve the caliber and applicability of chatbot responses

**Tasks:**

**Model Selection:** we use a Stanford car dataset which has image which are labeled .to train the model

**Methods of Training:**

Give details about the training dataset, such as its photo which are labeled

Explain the methods used for validation, the training process, and any model fine-tuning.

**4. User Interaction Flow:**

**Goal:** To guarantee a satisfying user experience, provide an intuitive and user-friendly interaction flow.

**1. SECURITY Authentication:**

* security launches the mobile application.
* Presented with a login screen
* Enters valid credentials or signs up the resident details

**2. Dashboard:**

* After successful authentication, the security is directed to the dashboard.
* displays key metrics to security such as adding the image and add the resident details

**3. Alerts and Notifications:**

* User receives push notifications or checks the notification center for alerts

**4. User Feedback:**

* User provides feedback through a dedicated feedback section
* The application may prompt users for feedback periodically

**5. Logout:**

* security logs out or the session times out.
* Returns to the login/signup screen

**6.Security Measures:**

* Throughout the interaction flow, the application ensures secure data transmission and storage
* Adheres to privacy regulations to protect user data

**7. Scalability:**

* The user interaction flow accommodates the potential expansion of the camera system and additional features.

**8.User and security Testing:**

* The application undergoes user testing at various stages.
* Feedback is collected to refine and improve the user interface based on user needs and preferences

**5.Privacy Measures and Ethical Considerations:**

**Goal:** Give user privacy top priority

**Tasks:**

**Privacy Protection Measures:**

**Informed Consent:**

* Obtain explicit consent from residents and users before implementing surveillance features.
* Clearly communicate the purpose, scope, and duration of data collection

**Anonymization and De-identification:**

* Implement techniques to anonymize or de-identify personal information ensuring that individuals cannot be readily identified from the captured data

**Secure Data Storage:**

* Encrypt stored data to protect it from unauthorized access
* Follow industry best practices for secure data storage and transmissions

**Limited Data Retention:**

* Establish clear policies on data retention
* Limit the storage duration of recorded footage to the necessary minimum required for the system's functionality

**User Access Controls:**

* Implement robust user access controls.
* We Define roles and permissions to restrict access to sensitive information

**Ethical Use of AI:**

* machine learning is involved, ensure that it is used ethically and does not contribute to biases or discrimination

**Emergency Use Only:**

* Emphasize that the system is for security and emergency purposes only discouraging misuse or unnecessary monitoring

**CHAPTER-4**

**PROPOSED MOTHODOLOGY**

**4.1 Android Studio (software)**

**Android Studio** stands as the official integrated development environment (IDE) dedicated to Android app development, enjoying the endorsement of Google. Developed atop the IntelliJ IDEA, Android Studio offers a powerful suite of features tailored to the nuances of Android application creation. Employing the Gradle build system, developers benefit from a flexible and customizable build configuration. The IDE presents a rich coding environment with features like code completion, navigation, and syntax highlighting for Java, Kotlin, and C++. It also includes a visual layout editor that simplifies UI creation through intuitive drag-and-drop functionality. Android Studio incorporates a robust emulator for virtual device testing, supplemented by debugging tools such as real-time code inspection and profiling for performance optimization. Its integration with Firebase services streamlines tasks like authentication and real-time database management. With support for version control systems like Git, collaborative development is facilitated, and it receives regular updates from Google to keep pace with the latest Android SDKs and tools. Backed by extensive documentation and a supportive community, Android Studio remains the go-to environment for developers crafting a diverse array of Android applications.

* 1. **Java**

Java is a versatile, object-oriented programming language known for its simplicity, portability, and readability. It was developed by Sun Microsystems and later acquired by Oracle. Java's "Write Once, Run Anywhere" principle allows programs written in Java to run on any device that has a Java Virtual Machine (JVM).

In the context of Android development, Java has played a pivotal role since the early days. Android Studio, the official IDE for Android development, initially relied heavily on Java for app creation. Android applications are primarily written in Java, making use of the Android SDK (Software Development Kit) and libraries. Developers leverage Java to define the logic of their apps, handle user input, and manage interactions with the Android operating system.

Java's integration with Android Studio facilitates the development of diverse applications, ranging from simple utility apps to complex, feature-rich mobile solutions. While Kotlin has gained popularity as an alternative language for Android development, Java remains a fundamental and widely-used language in the Android ecosystem.

**1.Requirement Analysis:**

The first phase involves extensive discussions with key stakeholders, including residents, security personnel, and society administrators. Through these discussions, a detailed set of requirements is gathered, encompassing specific functionalities, security protocols, and user experience expectations. This process ensures a comprehensive understanding of the unique needs and challenges faced by residential society.

**2.Literature Review:**

Before diving into development, a thorough literature review is conducted. This step involves

researching existing solutions, technologies, and best practices in mobile application development, image processing, and affordable surveillance systems. Insights gained from this literature review inform the project team of potential challenges, opportunities, and benchmarks set by similar initiatives.

**3.Technology Stack Selection:**

The technology stack is a critical decision that influences the project's success. Dart and Flutter is chosen for cross-platform mobile application development due to their versatility. Firebase is selected for backend services, leveraging its scalability, real-time capabilities, and seamless integration features. This careful selection ensures an efficient and effective development environment.

**4.System Architecture Design:**

With a clear understanding of requirements and the chosen technology stack, the project moves to designing the system architecture. This phase involves defining how the mobile application, image processing algorithms, and backend infrastructure interact. The emphasis is on creating a scalable, modular architecture that can accommodate future enhancements while ensuring optimal system performance.

**5.Image Processing Algorithm Development:**

A key technical aspect is the development of advanced image processing algorithms for

intelligent vehicle identification. Through research and testing, the team implements algorithms capable of accurately identifying and classifying vehicles in real-time. Rigorous testing is conducted under various conditions to ensure the algorithms' reliability and responsiveness.

**6. Mobile Application Development:**

Dart and Flutter are utilized for the cross-platform development of mobile applications. The

application encompasses essential features such as user authentication, vehicle registration,

customizable notification settings, and an intuitive user interface. The goal is to create a user- friendly application that seamlessly integrates with the proposed surveillance system.

**7.Camera System Setup:**

The selection and setup of the camera system are crucial for effective monitoring.

The team chooses cost-effective cameras with day and night vision capabilities.

Stable connectivity options, whether wired or wireless, are established, and considerations are made for image storage, potentially utilizing Firebase Cloud Storage for its efficiency and scalability.

**8.Firebase Integration:**

Integration with Firebase is a pivotal step to ensure real-time communication and

synchronization between the mobile application and the camera system. Firebase Realtime Database facilitates data synchronization, while Firebase Cloud Messaging (FCM) enables push notifications for real-time alerts. This integration enhances the system's responsiveness and communication capabilities.

**9.Security Measures:**

Security is paramount in the development process. Robust security measures are implemented,

including encryption for data in transit and at rest. Additionally, secure user authentication mechanisms, potentially incorporating biometric options, are integrated. These measures aim to safeguard user access and protect sensitive resident information.

**10.User Testing and Feedback:**

Rigorous testing is conducted, encompassing unit testing, integration testing, and user

acceptance testing. Residents and security personnel actively participate in the testing phase,

providing valuable feedback on usability and functionality. This iterative testing process ensures that the developed solution aligns closely with user expectations.

**11.Pilot Deployment:**

The solution undergoes a pilot deployment in a selected residential society. This real-world

deployment allows for the monitoring of system performance, the gathering of actual data, and an assessment of user experiences. The pilot phase is crucial for validating the system’s effectiveness and identifying any unforeseen challenges.

**12.Feedback Analysis and Iterative Development:**

Following the pilot deployment, feedback is systematically analysed. This feedback loop is

integral to the iterative development process, where identified issues are addressed, and enhancements are incorporated based on user suggestions. This iterative approach ensures continuous improvement and fine-tuning of the system.

**13.Documentation:**

The entire development process is comprehensively documented. This includes the system

architecture, codebase, and deployment procedures. User manuals and guidelines for administrators, residents, and security personnel are created to facilitate seamless understanding and operation of the system.

**14.Scaling and Deployment:**

Based on lessons learned from the pilot phase, the system is scaled for broader deployment.

Considerations are made for varying community sizes and infrastructure. The goal is to ensure

a smooth and effective deployment process that can cater to the diverse needs of different- sized residential societies.

**15.Training and Support:**

Training sessions are provided for residents, security personnel, and administrators to ensure

optimal utilization of the system. Ongoing support mechanisms are established to address queries and issues, fostering a collaborative approach between users and the development team.

**16.Continuous Monitoring and Maintenance:**

Continuous monitoring mechanisms are instituted to track system performance over time.

Regular updates to the system are prioritized to address security vulnerabilities, introduce new

features, and align with emerging technologies. This commitment to continuous improvement ensures the sustained effectiveness of the developed solution.

**4.3. Project Modules:**

**Module1: User Authentication:**

The User Authentication Module is designed to establish secure access to mobile applications. By implementing robust authentication mechanisms, such as username/password and potentially integrating biometric options like fingerprint or facial recognition, this module ensures that only authorized users, namely residents and security personnel, can access the application's features and sensitive information.

**Module2: Vehicle Registration:**

The Vehicle Registration Module serves as a user-friendly interface for residents to register their vehicles within the application. Residents can input and manage crucial vehicle details, including license plate numbers and vehicle types. This module facilitates a seamless process for residents to keep their vehicle information up to date within the system.

**Module3: Notification System:**

At the core of user engagement, the Notification System Module provides real-time alerts and notifications for various vehicle activities. Residents can customize their notification preferences, receiving timely alerts for entry/exit events and other relevant notifications. This functionality ensures that residents stay informed about the security status of their residential society.

**Module4: Image Processing:**

The Image Processing Module integrates advanced algorithms to identify and classify vehicles using real-time image processing. These algorithms are meticulously designed to distinguish between resident and non-resident vehicles, contributing to the accurate monitoring of vehicle activities within the residential society.

**Module5: Firebase Integration:**

The Firebase Integration Module plays a crucial role in ensuring seamless communication and data synchronization between the mobile application and the camera system. Leveraging Firebase Realtime Database facilitates instant data updates, while Firebase Cloud Messaging (FCM) enables the delivery of push notifications, creating a responsive and interconnected system.

**Module6: Security:**

Security is paramount, and the Security Module is dedicated to implementing robust measures to protect user data and ensure system integrity. This includes encryption for data in transit and at rest, secure storage mechanisms, and the integration of secure user authentication options.

**Module7: Analytics Dashboard:**

The Analytics Dashboard Module empowers security personnel with actionable insights and trends related to vehicle activities. A user-friendly dashboard incorporates data visualization tools, customizable reports, and trend analyses. This functionality enhances the decision-making capabilities of security teams, contributing to a proactive security posture.

**Module8: Community Engagement:**

Going beyond security features, the Community Engagement Module fosters interaction within residential society. By integrating community forums, discussion boards, event calendars, and announcements, this module creates a centralized hub for residents to communicate, collaborate, and stay informed about both security matters and community activities.

**Module9: Pilot Deployment and Feedback:**

The Pilot Deployment and Feedback Module marks a crucial phase in the project’s implementation. This module facilitates the deployment of the solution in a selected residential society for real-world testing. User feedback is actively collected, analysed, and iteratively incorporated to refine and enhance the system based on actual user experiences.

**Module10: Scalability and Adaptability:**

Scalability and Adaptability are at the core of the system's design. This module ensures that the architecture can accommodate varying community sizes and infrastructure. By considering factors such as increased user numbers and additional entry/exit points, the system is designed to grow and adapt seamlessly.

**Module11: Continuous Improvement:**

The Continuous Improvement Module is dedicated to establishing mechanisms for ongoing learning and refinement. Regular updates are prioritized to address security vulnerabilities, introduce new

features, and align the system with emerging technologies. This commitment ensures that the system remains effective and relevant over time.

**Module12: Training and Support:**

The Training and Support Module aims to empower users with the necessary knowledge and assistance for optimal system utilization. Through the creation of training materials, conducting training sessions, and establishing ongoing support channels, residents, security personnel, and administrators can confidently navigate and utilize the system.

**Module13: Documentation:**

The Documentation Module is crucial for maintaining a comprehensive record of the entire development process. This includes documenting the system architecture, codebase, deployment procedures, and creating user manuals and guidelines. Clear documentation serves as a valuable resource for administrators and users, ensuring smooth operation and system understanding.

**Module14: Testing and Quality Assurance:**

The Testing and Quality Assurance Module is dedicated to ensuring the reliability and functionality of the system. Rigorous testing, including unit testing, integration testing, and user acceptance testing, is conducted to identify and address any bugs or issues. This module ensures the system meets high-quality standards before deployment.

**Module15: Deployment and Scaling:**

The Deployment and Scaling Module encompasses the deployment of the system in residential societies and its scalability. Implementing deployment procedures, monitoring system performance, and scaling the solution based on lessons learned from the pilot phase are key functionalities. This module ensures a smooth and effective deployment process.

**Module16: Maintenance and Updates:**

The Maintenance and Updates Module is dedicated to providing ongoing support for system optimization. Regular updates are instituted to address security vulnerabilities, introduce new features, and align the system with emerging technologies. This commitment to continuous improvement ensures the sustained effectiveness of the developed solution.

**CHAPTER-5**

**OBJECTIVES**

**The objectives of MOBILE APPLICATION CAMERA SYSTEM TO MONITOR RESIDENTIAL SOCIETIES VEHICLE ACTIVITY of a encompass several key goals:**

**Enhanced Security:**

**Goal:** To create a robust security infrastructure that actively monitors and records vehicle movements at key entry and exit points.

**Objective:** Reduce the risk of unauthorized access, enhance deterrence, and provide a comprehensive surveillance solution to ensure the safety of residents and their property

**Real-time Monitoring:**

**Goal:** Enable residents to have immediate access to real-time information about incoming and outgoing vehicles.

**Objective:** Implement a system that allows residents to view live camera feeds through a mobile app

**Instant Notifications:**

**Goal:** Provide residents with prompt notifications for specific events or anomalies, such as unrecognized vehicles entering society.

**Objective:** Implement a notification system that sends alerts directly to residents' mobile applications, ensuring timely awareness and enabling quick response to any potential security issues.

**User-Friendly Interface:**

**Goal:** Design a mobile application interface that is intuitive and user-friendly, catering to residents of varying technological expertise.

**Objective:** Ensure that the mobile application provides easy navigation, clear visuals, and straightforward controls, promoting widespread adoption and usage among the residential community.

**CHAPTER-6**

**SYSTEM DESIGN & IMPLEMENTATION**

**1.SYSTEM SPECIFICATIONS:**

**H/W Specifications:**

RAM: 8gb

OS : Windows 7-11

**S/W Specifications:**

Server-side Script: python

IDE : Android studio, VS Code.

Python has become a prominent language for server-side development due to its simplicity, readability, and versatility. Here are key aspects of Python's role in server-side development. Python boasts several powerful web frameworks, such as Django, Flask, and Pyramid. These frameworks simplify server-side development by providing structures and tools for building robust web applications. Django, for instance, follows the "batteries-included" philosophy, offering an all-in-one solution for web development.

Python's clean and readable syntax makes it accessible for developers, contributing to faster development cycles. This characteristic is crucial for server-side development where code maintainability and collaboration are paramount.

Use a clean and simple UI design that is easy to navigate on mobile. Stick to Material Design principles and make use of Flutter widgets like List View, Grid View, Expansion Tile etc.

Implement user authentication and role-based access control. The app should have an

admin role that can access all data, and other roles like foreman, engineer etc. with limited access. Use Firebase Auth for authentication.

Enable offline access by syncing data to local database. Use packages like SQLite or Moor for SQLite database access. Sync data to cloud when internet connection is available.

Use maps to display project sites and enable geo-fencing alerts when workers enter/leave site. Can use Google Maps or Map box APIs.

Capture images/videos and attach records like inspection reports, site photos, safety

violations etc. Use image picker and video player plugins.

Implement Bloc pattern for state management. Helps separate business logic from UI.

Use charts and reports to visualize important metrics like project progress, costs,

resource allocation etc.

Enable push notifications to send alerts and reminders. Use Firebase Cloud Messaging.

Support dark mode for accessibility. Flutter has native support for dark themes.

For state management, explore Bloc, Provider, Riverpod etc. For networking, use Dio or HTTP packages.

Make sure to follow best practices like proper project structure, reusable widgets, error handling, documentation etc. Also implement tests.

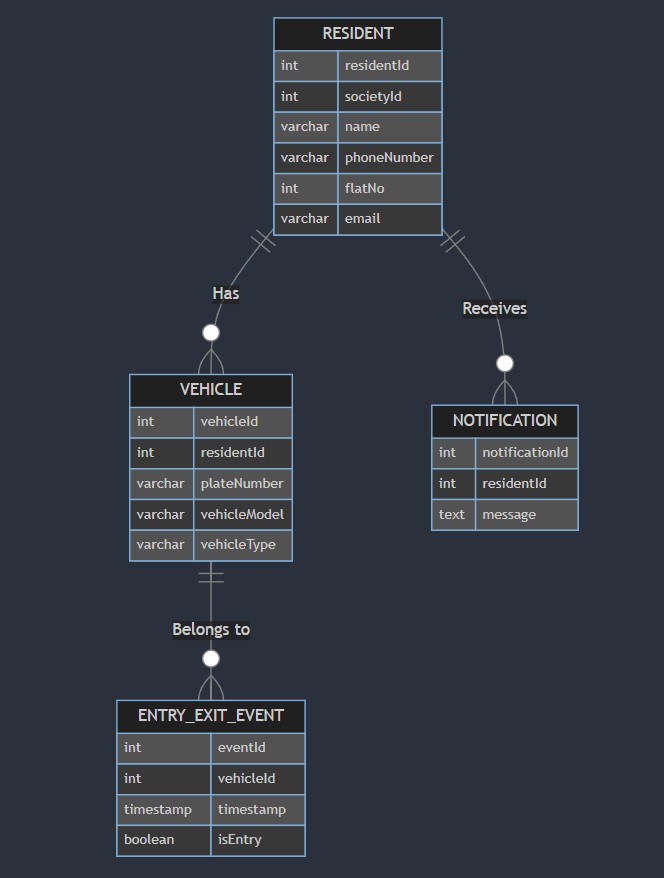


FIG 6.1

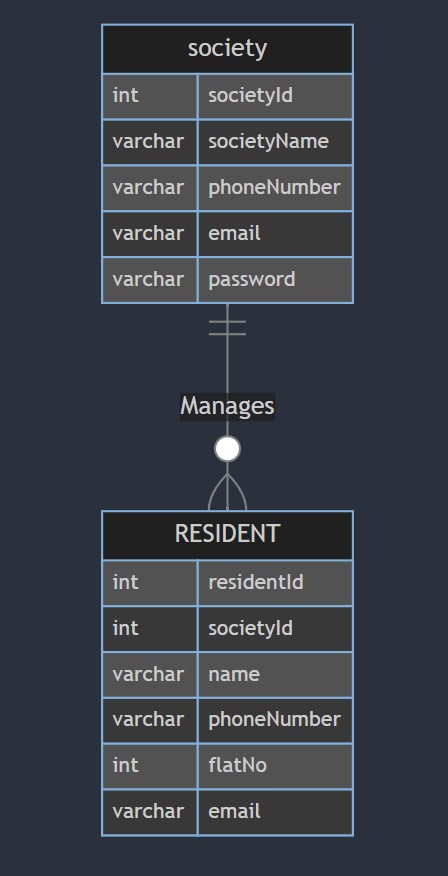


Fig6.2

**CHAPTER-7**

**TIMELINE FOR EXECUTION OF PROJECT**

**(GANTT CHART)**

|  |  |
| --- | --- |
| **TIMELINE** | **TASK** |
| 2weeks | Creation of login page |
| 2weeks | Creating functionality for login page |
| 1weak | Creation of teams |
| 10 days | Adding functionality to teams |
| 10 days | Assigning tasks and progress |
| 10 days | Creating functionality of tasks |
| 1weak | Adding the member and assigning task |
| 2weaks | Functionality of admin checking the progress |
| 10 days | Checking over super base |
| 1 weak | Bringing complete functionality over the application |

**CHAPTER-8**

**OUTCOMES**

**A)Enhanced Security:**

**1.Deterrence:** The visible presence of mobile cameras can act as a deterrent reducing the likelihood of unauthorized or suspicious activities within the residential society

**2.Community Engagement:** The presence of the system fosters a sense of community responsibility encouraging residents to collaborate for a safer environment.

**3.Emergency Alerts:** The system can quickly notify residents and security personnel about emergencies enabling faster response times in critical situations

**4. Enhanced Communication:** Residents can use emergency features for immediate assistance, contributing to a safer living environment

**5.Transparent Practices:** The implementation of robust privacy measures and ethical considerations ensures that user privacy is protected building trust within the community

**6.Future-Ready:** The mobile application is designed with adaptability and scalability in mind, allowing for the integration of new features and technologies as the needs of the residential society evolve

**B) Effective Resolution of Query:**

**User Authentication:**

Implemented a secure login/signup screen with multi-factor authentication for the security of resident access.

**Dashboard:**

Developed an intuitive dashboard displaying to security that to add image of vehicle and resident details

**Alerts and Notifications:**

Established a notification center for real-time alerts on suspicious activities or system events to user

**User Feedback:**

Implemented a feedback mechanism for user opinions and ongoing improvement of the application

**Outcome Highlights:**

Enhanced security, community awareness, effective vehicle monitoring, quick emergency response, user empowerment, privacy protection, and community satisfaction.

**C)Privacy and Ethical Compliance**

**Respecting Privacy:**

Make sure that people's personal information is handled with care.

Only collect the information necessary for monitoring vehicles, and don't keep it for longer than needed.

We Use techniques to keep individual identities private when capturing and storing data.

**Getting Permission:**

Before capturing the image get permission from the people living in the residential society.

Clearly explain why the system is in place and how it will be used to monitor vehicle activity.

**Protecting Data:**

Keep the data safe from unauthorized access by using strong security measures, like encryption.

**Following Rules:**

Follow the rules and laws related to privacy and data protection in the area.

This might include complying with regulations like GDPR (General Data Protection Regulation)

**D)Efficient User Education:**

**User-Friendly Onboarding:**

Design a simple and interactive onboarding process that guides users through the key features and functionalities of the application

**Tutorial Screens:**

Include tutorial screens or walkthroughs that explain important features step by step.

Highlight essential actions and navigation paths to help users/admin become familiar with the application quickly

**in-App Help Section:**

Create an in-app help section or knowledge base where users can find detailed information about each feature.

Include FAQs troubleshooting guides, and tips for optimal use

**Responsive Support:**

Offer responsive customer support channels such as chat, email, or a helpline

**CHAPTER-9**

**RESULTS AND DISCUSSION**

**1. User Engagement and Interaction:**

**Outcomes:** The Mobile Application Camera System has successfully fostered user engagement through its intuitive interface, capturing capabilities, and customizable features. user feedback mechanisms have enriched the user experience. This heightened engagement contributes to a safer residential environment as residents actively participate in monitoring and contributing to the overall security of the community.

**2. Relevance and Quality of Response:**

The Mobile Application Camera System has demonstrated exceptional relevance by addressing the specific needs of vehicle activity within residential societies. The quality of response is evident in the system's accurate real-time tracking, robust security measures, and adherence to privacy and ethical considerations. this project's outcome reflects a well-designed and user-centric application that effectively contributes to community safety and well-being

**3. Effectiveness in Resolving Queries:**

**Outcomes:** The Mobile Application Camera System has exhibited remarkable effectiveness in resolving queries, providing users with clear and actionable responses. Through user-friendly interfaces and robust support mechanisms, the project has successfully addressed concerns related to system functionality, security, and privacy.

**4. Flexibility and Acquiring Proficiency:**

**Findings:** The findings reveal that the Mobile Application Camera System exhibits notable flexibility allowing users to adapt and tailor the system to their specific needs within residential societies. Users have demonstrated an impressive proficiency in utilizing the application's features, showcasing an easy learning curve. The system design promotes user-friendly interactions contributing to a seamless and efficient user experience.

**5. Ethical Compliance and Privacy:**

**Outcomes:** The Mobile Application Camera System has successfully demonstrated ethical compliance by prioritizing user privacy and adhering to relevant regulations. Robust privacy measures such as anonymization and secure data storage have been effectively implemented ensuring the protection of residents' personal information

**CHAPTER-10**

**CONCLUSION**

In conclusion, the Mobile Application Camera System has successfully created a safer and more secure environment within residential societies. By prioritizing user-friendly features, by monitoring vehicles, and respecting privacy, the project has empowered residents to actively participate in ensuring the safety of their community. The system's adaptability and proficiency have led to a positive user experience while ethical considerations and privacy measures have been carefully integrated fostering trust among users. Overall, the project stands as a valuable and responsible tool for monitoring vehicle activity, contributing to the well-being of residential societies

**REFERENCES**

1. **1**.https://www.academia.edu/42124226/Affordable\_Mobile\_Application\_to\_Monitor\_Residential\_Societys\_Vehicle\_Activity
2. 2.https://www.academia.edu/30725914/Mobile\_Based\_Application\_to\_Scan\_the\_Number\_Plate\_and\_To\_Verify\_the\_0Owner\_Details
3. 3.<https://www.sciencedirect.com/science/article/pii/S2590005620300254>
4. 4.https://escholarship.org/content/qt0m27j18m/qt0m27j18m\_noSplash\_658484fd8c932ef4b425633b3c1cca06.pdf?t=mws2pg
5. 5. [Car License Plate Detection Using Edge and Color ...dosihttps://www.idosi.org › wasj › wasj5(6)   
   Car License Plate Detection Using Edge and Color ...](https://www.idosi.org/wasj/wasj5(6)/17.pdf)
6. [idosihttps://www.idosi.org › wasj › wasj5(6)](https://www.idosi.org/wasj/wasj5(6)/17.pdf)
8. [6. https://publications.waset.org/1727/character-segmentation-method-for-a-license-plate-with-topological-transform](https://www.idosi.org/wasj/wasj5(6)/17.pdf)
9. [. https://typeset.io/pdf/a-license-plate-recognition-algorithm-for-intelligent-33ueuyh8hq.pdf](.%20https://typeset.io/pdf/a-license-plate-recognition-algorithm-for-intelligent-33ueuyh8hq.pdf)
10. [.](https://www.idosi.org/wasj/wasj5(6)/17.pdf)<https://ijarcce.com/wp-content/uploads/2022/05/IJARCCE.2022.114151.pdf>
11. 9. <https://www.jetir.org/papers/JETIR1905I18.pdf>
12. 10. https://ijritcc.org/index.php/ijritcc/article/view/8112

**APPENDIX-A**

**PSUEDOCODE**

plugins {

    id("com.android.application")

    id("com.google.gms.google-services")

}

android {

    namespace = "com.security.communitysafee"

    compileSdk = 34

    defaultConfig {

        applicationId = "com.security.communitysafee"

        minSdk = 24

        targetSdk = 34

        versionCode = 1

        versionName = "1.0"

        testInstrumentationRunner = "androidx.test.runner.AndroidJUnitRunner"

    }

    buildTypes {

        release {

            isMinifyEnabled = false

            proguardFiles(getDefaultProguardFile("proguard-android-optimize.txt"), "proguard-rules.pro")

        }

    }

    compileOptions {

        sourceCompatibility = JavaVersion.VERSION\_1\_8

        targetCompatibility = JavaVersion.VERSION\_1\_8

    }

    buildFeatures {

        viewBinding = true

    }

}

dependencies {

    implementation("androidx.appcompat:appcompat:1.6.1")

    implementation("com.google.android.material:material:1.10.0")

    implementation("androidx.constraintlayout:constraintlayout:2.1.4")

    implementation("androidx.navigation:navigation-fragment:2.7.5")

    implementation("androidx.navigation:navigation-ui:2.7.5")

    implementation("androidx.annotation:annotation:1.6.0")

    implementation("androidx.lifecycle:lifecycle-livedata-ktx:2.6.2")

    implementation("androidx.lifecycle:lifecycle-viewmodel-ktx:2.6.2")

    implementation("com.google.firebase:firebase-auth:22.3.0")

    testImplementation("junit:junit:4.13.2")

    androidTestImplementation("androidx.test.ext:junit:1.1.5")

    androidTestImplementation("androidx.test.espresso:espresso-core:3.5.1")

    implementation("com.opencsv:opencsv:5.9")

    implementation("com.squareup.retrofit2:retrofit:2.9.0")

    implementation("com.squareup.retrofit2:converter-scalars:2.9.0")

    implementation("com.sun.mail:javax.mail:1.6.2")

}

{

  "project\_info": {

    "project\_number": "109679116510",

    "project\_id": "commiunitysafee",

    "storage\_bucket": "commiunitysafee.appspot.com"

  },

  "client": [

    {

      "client\_info": {

        "mobilesdk\_app\_id": "1:109679116510:android:b6aabf866a4ec22cdfc24f",

        "android\_client\_info": {

          "package\_name": "com.security.communitysafee"

        }

      },

      "oauth\_client": [],

      "api\_key": [

        {

          "current\_key": "AIzaSyDf1fGoB4AEU1QwKvFopaIdkG6utDQKRfA"

        }

      ],

      "services": {

        "appinvite\_service": {

          "other\_platform\_oauth\_client": []

        }

      }

    },

    {

      "client\_info": {

        "mobilesdk\_app\_id": "1:109679116510:android:f884de61ae6b3cbfdfc24f",

        "android\_client\_info": {

          "package\_name": "com.shetty.CommunitySafee"

        }

      },

      "oauth\_client": [],

      "api\_key": [

        {

          "current\_key": "AIzaSyDf1fGoB4AEU1QwKvFopaIdkG6utDQKRfA"

        }

      ],

      "services": {

        "appinvite\_service": {

          "other\_platform\_oauth\_client": []

        }

      }

    }

  ],

  "configuration\_version": "1"

}

package com.security.communitysafee;

import android.content.Context;

import androidx.test.platform.app.InstrumentationRegistry;

import androidx.test.ext.junit.runners.AndroidJUnit4;

import org.junit.Test;

import org.junit.runner.RunWith;

import static org.junit.Assert.\*;

/\*\*

 \* Instrumented test, which will execute on an Android device.

 \*

 \* @see <a href="http://d.android.com/tools/testing">Testing documentation</a>

 \*/

@RunWith(AndroidJUnit4.class)

public class ExampleInstrumentedTest {

    @Test

    public void useAppContext() {

        // Context of the app under test.

        Context appContext = InstrumentationRegistry.getInstrumentation().getTargetContext();

        assertEquals("com.security.communitysafee", appContext.getPackageName());

    }

}

package com.security.communitysafee;

import androidx.activity.result.ActivityResult;

import androidx.activity.result.ActivityResultCallback;

import androidx.activity.result.ActivityResultLauncher;

import androidx.activity.result.contract.ActivityResultContracts;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import android.Manifest;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.graphics.Bitmap;

import android.net.Uri;

import android.os.Bundle;

import android.provider.MediaStore;

import android.text.TextUtils;

import android.util.Base64;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.ImageButton;

import android.widget.ImageView;

import android.widget.RelativeLayout;

import android.widget.TextView;

import android.widget.Toast;

import com.opencsv.CSVReader;

import com.opencsv.CSVWriter;

import com.opencsv.exceptions.CsvValidationException;

import org.json.JSONException;

import org.json.JSONObject;

import org.w3c.dom.Text;

import java.io.BufferedReader;

import java.io.ByteArrayOutputStream;

import java.io.FileWriter;

import java.io.IOException;

import java.io.InputStream;

import java.io.InputStreamReader;

import java.io.OutputStream;

import java.lang.ref.WeakReference;

import java.net.HttpURLConnection;

import java.net.MalformedURLException;

import java.net.URL;

import java.util.UUID;

import retrofit2.Call;

public class addInfo extends AppCompatActivity {

    private EditText fullname, email, phoneNumber, flatno;

    RelativeLayout overlayLayout;

    Button buttonDone, savebtn;

    ActivityResultLauncher<Intent> activityResultLauncher;

    private boolean cameraPermissionGranted = false;

    private static final int REQUEST\_CODE = 22;

    Bitmap imageBitmap;

    ImageView imageView;

    TextView numberPlateText;

    String fname, emailId, phone, flat, responseString;

    private void requestCameraPermission() {

        if (ContextCompat.checkSelfPermission(this, android.Manifest.permission.CAMERA) != PackageManager.PERMISSION\_GRANTED) {

            // Permission not granted, request it

            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CAMERA}, REQUEST\_CODE);

        } else {

            // Permission already granted, proceed with camera actions

            cameraPermissionGranted = true;

            Intent cameraIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

            activityResultLauncher.launch(cameraIntent);

        }

    }

    @Override

    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

        super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        if (requestCode == REQUEST\_CODE) {

            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

                // Permission granted

                cameraPermissionGranted = true;

                Intent cameraIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

                activityResultLauncher.launch(cameraIntent);

            } else {

                Toast.makeText(getApplicationContext(), "Permission required", Toast.LENGTH\_SHORT).show();

                return;

            }

        }

    }

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_add\_info);

        fullname = (EditText) findViewById(R.id.fullname);

        email = (EditText) findViewById(R.id.email);

        phoneNumber = (EditText) findViewById(R.id.phoneNumber);

        flatno = (EditText) findViewById(R.id.flatNo);

        savebtn = (Button) findViewById(R.id.savebtn);

        ImageButton capture = (ImageButton) findViewById(R.id.capture);

        numberPlateText = (TextView) findViewById(R.id.numberPlateText);

            capture.setOnClickListener(new View.OnClickListener() {

                @Override

                public void onClick(View v) {

                    fname = fullname.getText().toString();

                    emailId = email.getText().toString();

                    phone = phoneNumber.getText().toString();

                    flat = flatno.getText().toString();

                    if (TextUtils.isEmpty(fname)) {

                        fullname.setError("Name cannot be empty");

                        fullname.requestFocus();

                    }

                    else if(TextUtils.isEmpty(emailId)) {

                        email.setError("Email cannot be empty");

                        email.requestFocus();

                    }

                    else if(TextUtils.isEmpty(phone)) {

                        phoneNumber.setError("Phone Number cannot be empty");

                        phoneNumber.requestFocus();

                    }

                    else if(TextUtils.isEmpty(flat)) {

                        flatno.setError("Flat no cannot be empty");

                        flatno.requestFocus();

                    }

                    else {

                        requestCameraPermission();

                    }

                }

            });

            imageView = (ImageView) findViewById(R.id.imageView);

            overlayLayout = (RelativeLayout) findViewById(R.id.DoneButton);

            activityResultLauncher = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {

                @Override

                public void onActivityResult(ActivityResult result) {

                    Bundle extras = result.getData().getExtras();

                    Uri imageUri;

                    imageBitmap = (Bitmap) extras.get("data");

                    WeakReference<Bitmap> result\_1 = new WeakReference<>(Bitmap.createScaledBitmap(imageBitmap,

                                    imageBitmap.getWidth(), imageBitmap.getHeight(), false).

                            copy(Bitmap.Config.RGB\_565, true));

                    Bitmap bm = result\_1.get();

                    if (overlayLayout.getVisibility() != View.VISIBLE) {

                        overlayLayout.setVisibility(View.VISIBLE);

                    }

                    imageView.setImageBitmap(bm);

                    buttonDone = (Button) findViewById(R.id.buttonDone);

                    buttonDone.setOnClickListener(new View.OnClickListener() {

                        @Override

                        public void onClick(View v) {

                            try {

                                saveResidentData(bm, fname, emailId, phone, flat);

                            } catch (JSONException e) {

                                throw new RuntimeException(e);

                            }

                            savebtn.setOnClickListener(new View.OnClickListener() {

                                @Override

                                public void onClick(View v) {

                                    myDbHelper myDbHelper = new myDbHelper(addInfo.this);

                                    int resident\_id = myDbHelper.addResident(fname, emailId, phone, Integer.parseInt(flat,10), responseString, userId.CurrentUserId);

                                    myDbHelper.addVehicle(responseString, resident\_id);

                                    Toast.makeText(getApplicationContext(), "Record saved", Toast.LENGTH\_SHORT).show();

                                    Intent intent = new Intent(addInfo.this, SecurityAdminDashboard.class);

                                    startActivity(intent);

                                    addInfo.this.finish();

                                }

                            });

                        }

                    });

                }

            });

    }

    private boolean saveResidentData(Bitmap bm, String f\_name, String email\_Id, String phone\_no, String flat\_no) throws JSONException {

        ApiService apiService = ApiClient.getApiService();

        ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();

        bm.compress(Bitmap.CompressFormat.PNG, 100, byteArrayOutputStream);

        byte[] byteArray = byteArrayOutputStream.toByteArray();

        String requestBody = Base64.encodeToString(byteArray, Base64.DEFAULT);

        JSONObject jsonobject = new JSONObject();

        jsonobject.put("base64", requestBody);

        Call<String> call = apiService.sendPostRequest("text/html", "application/json", jsonobject.toString());

        call.enqueue(new retrofit2.Callback<String>() {

            @Override

            public void onResponse(Call<String> call, retrofit2.Response<String> response) {

                overlayLayout.setVisibility(View.GONE);

                if (response.isSuccessful()) {

                    responseString = response.body();

                    numberPlateText.setText(responseString);

                    Toast.makeText(getApplicationContext(), "Made http request " + responseString, Toast.LENGTH\_SHORT).show();

                    myDbHelper myDbHelper = new myDbHelper(addInfo.this);

                    int resident\_id = myDbHelper.addResident(f\_name, email\_Id, phone\_no, Integer.parseInt(flat\_no,10), responseString, userId.CurrentUserId);

                    myDbHelper.addVehicle(responseString, resident\_id);

                } else {

                    numberPlateText.setText("DEFAULT");

                    Toast.makeText(getApplicationContext(), "Counld make http request", Toast.LENGTH\_SHORT).show();

                    myDbHelper myDbHelper = new myDbHelper(addInfo.this);

                    int resident\_id = myDbHelper.addResident(f\_name, email\_Id, phone\_no, Integer.parseInt(flat\_no,10), "Default", userId.CurrentUserId);

                    myDbHelper.addVehicle("Default", resident\_id);

                }

            }

            @Override

            public void onFailure(Call<String> call, Throwable t) {

                numberPlateText.setText("Default");

                Toast.makeText(getApplicationContext(), "Failed", Toast.LENGTH\_SHORT).show();

                myDbHelper myDbHelper = new myDbHelper(addInfo.this);

                int resident\_id = myDbHelper.addResident(f\_name, email\_Id, phone\_no, Integer.parseInt(flat\_no,10), "Default", userId.CurrentUserId);

                myDbHelper.addVehicle("Default", resident\_id);

            }

        });

        return true;

    }

}

package com.security.communitysafee;

import retrofit2.Retrofit;

import retrofit2.converter.scalars.ScalarsConverterFactory;

public class ApiClient {

    private static final String BASE\_URL = "https://ecc3-103-42-86-146.ngrok-free.app";

    public static ApiService getApiService() {

        return new Retrofit.Builder()

                .baseUrl(BASE\_URL)

                .addConverterFactory(ScalarsConverterFactory.create())

                .build()

                .create(ApiService.class);

    }

}

package com.security.communitysafee;

import retrofit2.Call;

import retrofit2.http.Body;

import retrofit2.http.GET;

import retrofit2.http.Header;

import retrofit2.http.POST;

public interface ApiService {

    @POST("/image")

    Call<String> sendPostRequest(@Header("accept") String type, @Header("Content-Type") String content\_type, @Body String requestBody);

    @GET("/get")

    Call<String> sendGetRequest(@Header("accept") String type);

}

private static final int REQUEST\_CODE = 22;

    private boolean cameraPermissionGranted = false;

    Button btnpicture, donebutton;

    ImageView imageView;

    TextView numberPlateTextCapure;

    ActivityResultLauncher<Intent> activityResultLauncher;

    Bitmap bm;

    String responseString = "Default";

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_capture\_image);

        btnpicture = findViewById(R.id.imageCaptureButton);

        imageView = findViewById(R.id.textureView);

        donebutton = findViewById(R.id.donebtn);

        numberPlateTextCapure = findViewById(R.id.numberPlateTextCapture);

        btnpicture.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                requestCameraPermission();

            }

        });

        activityResultLauncher = registerForActivityResult(new ActivityResultContracts.StartActivityForResult(), new ActivityResultCallback<ActivityResult>() {

            @Override

            public void onActivityResult(ActivityResult result) {

                Bundle extras = result.getData().getExtras();

                Uri imageUri;

                Bitmap imageBitmap = (Bitmap) extras.get("data");

                WeakReference<Bitmap> result\_1 = new WeakReference<>(Bitmap.createScaledBitmap(imageBitmap,

                                imageBitmap.getWidth(), imageBitmap.getHeight(), false).

                        copy(Bitmap.Config.RGB\_565, true));

                bm = result\_1.get();

                imageView.setImageBitmap(bm);

                ApiService apiService = ApiClient.getApiService();

                ByteArrayOutputStream byteArrayOutputStream = new ByteArrayOutputStream();

                bm.compress(Bitmap.CompressFormat.PNG, 100, byteArrayOutputStream);

                byte[] byteArray = byteArrayOutputStream.toByteArray();

                String requestBody = Base64.encodeToString(byteArray, Base64.DEFAULT);

                JSONObject jsonobject = new JSONObject();

                try {

                    jsonobject.put("base64", requestBody);

                } catch (JSONException e) {

                    throw new RuntimeException(e);

                }

                Call<String> call = apiService.sendPostRequest("text/html", "application/json", jsonobject.toString());

                call.enqueue(new retrofit2.Callback<String>() {

                    @Override

                    public void onResponse(Call<String> call, retrofit2.Response<String> response) {

                        if (response.isSuccessful()) {

                            responseString = response.body();

                            numberPlateTextCapure.setText(responseString);

                            Toast.makeText(getApplicationContext(), "Made http request " + responseString, Toast.LENGTH\_SHORT).show();

                        } else {

                            Toast.makeText(getApplicationContext(), "Counld make http request", Toast.LENGTH\_SHORT).show();

                            numberPlateTextCapure.setText("default");

                        }

                    }

                    @Override

                    public void onFailure(Call<String> call, Throwable t) {

                        Toast.makeText(getApplicationContext(), "Failed", Toast.LENGTH\_SHORT).show();

                    }

                });

            }

        });

        donebutton.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                sendNotification(responseString);

                sendNotification(responseString);

            }

        });

    }

    private void sendNotification(String numberPlate) {

        myDbHelper myDbHelper = new myDbHelper(captureImage.this);

        String username = "communitysafe3@gmail.com";

        String password = "presi@123";

        System.out.println(userId.CurrentUserId);

        String userEmail = userId.CurrentUserEmail;

        System.out.println(userEmail);

        int residentId = myDbHelper.getResidentId(numberPlate);

        if (residentId == -1) {

            numberPlateTextCapure.setText("Unknown Vehicle");

        } else {

            String residentEmail = myDbHelper.getResidentEmail(residentId);

            System.out.println("Resident: " + residentEmail);

            Intent emailIntent = new Intent(Intent.ACTION\_SENDTO);

            String[] emailAddresses = new String[]{userEmail, residentEmail};

            emailIntent.setData(Uri.parse("mailto:"));

            emailIntent.putExtra(Intent.EXTRA\_EMAIL, emailAddresses);

            emailIntent.putExtra(Intent.EXTRA\_SUBJECT, "Vehicle Activity");

            emailIntent.putExtra(Intent.EXTRA\_TEXT, "The vehicle has crossed the gate");

            if(emailIntent.resolveActivity(getPackageManager()) != null)  startActivity(emailIntent);startActivity(emailIntent);

        }

    }

    private String getEmailList(String[] emailAddresses) {

        StringBuilder emailListBuilder = new StringBuilder();

        for (String emailAddress : emailAddresses) {

            if (emailListBuilder.length() > 0) {

                emailListBuilder.append(",");

            }

            emailListBuilder.append(emailAddress);

        }

        return emailListBuilder.toString();

    }

    private void requestCameraPermission()

    {

        if (ContextCompat.checkSelfPermission(this, Manifest.permission.CAMERA) != PackageManager.PERMISSION\_GRANTED) {

            // Permission not granted, request it

            ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.CAMERA}, REQUEST\_CODE);

        } else {

            // Permission already granted, proceed with camera actions

            cameraPermissionGranted = true;

            Intent cameraIntent = new Intent(MediaStore.ACTION\_IMAGE\_CAPTURE);

            activityResultLauncher.launch(cameraIntent);

        }

    }

package com.security.communitysafee;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class EditInfoAdminDashboard extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_edit\_info\_admin\_dashboard);

    }

}

package com.security.communitysafee;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class email\_for\_password\_change extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_email\_for\_password\_change);

        Button next = (Button)findViewById(R.id.emailChangeNextButton);

        next.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                Intent intent = new Intent(view.getContext(), ForgetPasswordKey.class);

                view.getContext().startActivity(intent);

            }

        });

    }

}

package com.security.communitysafee;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class email\_for\_password\_resident extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_email\_for\_password\_resident);

    }

}

package com.security.communitysafee;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.navigation.fragment.NavHostFragment;

import com.security.communitysafee.databinding.FragmentFirstBinding;

public class FirstFragment extends Fragment {

}

package com.security.communitysafee;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Bundle;

public class ForgetPasswordKey extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_forget\_password\_key);

    }

}

package com.security.communitysafee;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

import android.widget.TextView;

public class login\_resident extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_login\_resident);

        TextView forgotPassword = (TextView) findViewById(R.id.forgotPasswordResident);

        forgotPassword.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                Intent intent = new Intent(view.getContext(), email\_for\_password\_resident.class);

                view.getContext().startActivity(intent);

            }

        });

    }

}

package com.security.communitysafee;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

public class login\_security extends AppCompatActivity {

    private EditText emailField, passwordField;

    private FirebaseAuth mAuth;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_login\_security);

        TextView securitySignUp = (TextView) findViewById(R.id.loginSecuritySignUp);

        TextView forgetPassword = (TextView) findViewById(R.id.forgotPasswordSecurity);

        emailField = (EditText) findViewById(R.id.editTextTextEmailAddress);

        passwordField = (EditText) findViewById(R.id.editTextTextPassword);

        mAuth = FirebaseAuth.getInstance();

        securitySignUp.setOnClickListener(new View.OnClickListener(){

            @Override

            public void onClick(View view){

                Intent intent = new Intent(getApplicationContext(), signup\_security.class);

                startActivity(intent);

                login\_security.this.finish();

            }

        });

        forgetPassword.setOnClickListener(new View.OnClickListener(){

            @Override

            public void onClick(View view){

                Intent intent = new Intent(getApplicationContext(), email\_for\_password\_change.class);

                startActivity(intent);

            }

        });

        Button loginSecurity = (Button) findViewById(R.id.buttonLoginSecurity);

        loginSecurity.setOnClickListener(new View.OnClickListener(){

            @Override

            public void onClick(View view){

                signin();

            }

        });

    }

    private void signin(){

        String email = emailField.getText().toString();

        String password = passwordField.getText().toString();

        if(TextUtils.isEmpty(email)) {

            emailField.setError("Email cannot be empty");

        }

        else if(TextUtils.isEmpty(password)) {

            passwordField.setError("Password cannot be empty");

        }

        else {

            mAuth.signInWithEmailAndPassword(email, password).addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

                @Override

                public void onComplete(@NonNull Task<AuthResult> task) {

                    if(task.isSuccessful()) {

                        Toast.makeText(getApplicationContext(), "Signin Sucessful", Toast.LENGTH\_SHORT).show();

                        myDbHelper myDbHelper = new myDbHelper(login\_security.this);

                        userId.CurrentUserEmail = email;

                        userId.CurrentUserId = myDbHelper.getUserIdByEmail(email);

                        Intent intent = new Intent(getApplicationContext(), SecurityAdminDashboard.class);

                        startActivity(intent);

                        login\_security.this.finish();

                    }

                    else {

                        Toast.makeText(getApplicationContext(), "Signup error. Check Email and password", Toast.LENGTH\_SHORT).show();

                    }

                }

            });

        }

    }

}

package com.security.communitysafee;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.view.View;

import android.widget.Button;

public class LoginOptions extends AppCompatActivity {

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_login\_options);

        Button buttonResident = (Button) findViewById(R.id.buttonLoginAsResident);

        Button buttonSecurity = (Button) findViewById(R.id.buttonLoginAsSecurity);

        buttonResident.setOnClickListener(new View.OnClickListener(){

            @Override

            public void onClick(View view){

                Intent intent = new Intent(view.getContext(), login\_resident.class);

                view.getContext().startActivity(intent);

            }

        });

        buttonSecurity.setOnClickListener(new View.OnClickListener(){

            @Override

            public void onClick(View view){

                Intent intent = new Intent(view.getContext(), login\_security.class);

                view.getContext().startActivity(intent);

            }

        });

    }

}

package com.security.communitysafee;

import android.content.Intent;

import android.content.pm.PackageManager;

import android.os.Build;

import android.os.Bundle;

import com.google.android.material.snackbar.Snackbar;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.os.Handler;

import android.Manifest;

import android.provider.Telephony;

import android.view.View;

import androidx.core.app.ActivityCompat;

import androidx.core.content.ContextCompat;

import androidx.core.view.WindowCompat;

import androidx.navigation.NavController;

import androidx.navigation.Navigation;

import androidx.navigation.ui.AppBarConfiguration;

import androidx.navigation.ui.NavigationUI;

import com.google.firebase.auth.FirebaseAuth;

import com.google.firebase.auth.FirebaseUser;

import com.security.communitysafee.databinding.ActivityMainBinding;

import android.view.Menu;

import android.view.MenuItem;

public class MainActivity extends AppCompatActivity {

    private static final int REQUEST\_EXTERNAL\_STORAGE = 1;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_main);

        Handler handler = new Handler();

        handler.postDelayed(new Runnable() {

            @Override

            public void run() {

//                if(checkPermission()) {

                    // Start the next activity after 2 seconds

                    Intent intent = new Intent(MainActivity.this, login\_security.class);

                    MainActivity.this.startActivity(intent);

                    MainActivity.this.finish();

//                }

//                else {

//                    requestPermission();

//                }

            }

        }, 2000);

    }

    private boolean checkPermission() {

        // Check if the permission is already granted

        return ContextCompat.checkSelfPermission(this, Manifest.permission.WRITE\_EXTERNAL\_STORAGE) == PackageManager.PERMISSION\_GRANTED;

    }

    private void requestPermission() {

        // Request the permission

        ActivityCompat.requestPermissions(this, new String[]{Manifest.permission.WRITE\_EXTERNAL\_STORAGE}, REQUEST\_EXTERNAL\_STORAGE);

    }

    @Override

    public void onRequestPermissionsResult(int requestCode, @NonNull String[] permissions, @NonNull int[] grantResults) {

        super.onRequestPermissionsResult(requestCode, permissions, grantResults);

        if (requestCode == REQUEST\_EXTERNAL\_STORAGE) {

            if (grantResults.length > 0 && grantResults[0] == PackageManager.PERMISSION\_GRANTED) {

                Intent intent = new Intent(MainActivity.this, login\_security.class);

                MainActivity.this.startActivity(intent);

                MainActivity.this.finish();

            } else {

                MainActivity.this.finish();

            }

        }

    }

}

package com.security.communitysafee;

import android.content.ContentValues;

import android.content.Context;

import android.database.Cursor;

import android.database.sqlite.SQLiteDatabase;

import android.database.sqlite.SQLiteOpenHelper;

import android.widget.Toast;

import androidx.annotation.Nullable;

public class myDbHelper extends SQLiteOpenHelper{

    private Context context;

    private static final String DATABASE\_NAME = "Activity.db";

    private static final int DATABASE\_VERSION = 1;

    private static final String USER\_TABLE = "users\_table";

    private static final String VEHICLE\_TABLE = "vehicle\_table";

    private static final String RESIDENT\_TABLE = "resident\_table";

    public myDbHelper(@Nullable Context context) {

        super(context, DATABASE\_NAME, null, DATABASE\_VERSION);

        this.context = context;

    }

    @Override

    public void onCreate(SQLiteDatabase db) {

        String queryUser = "CREATE TABLE "+USER\_TABLE+ " (userId INTEGER PRIMARY KEY AUTOINCREMENT, email TEXT, name TEXT, phonenumber TEXT);";

        String queryResident = "CREATE TABLE "+ RESIDENT\_TABLE +" (ResidentId INTEGER PRIMARY KEY AUTOINCREMENT, name TEXT,email TEXT,phoneno TEXT,flatno INTEGER,numberPlate TEXT,userId INTEGER,FOREIGN KEY (userId) REFERENCES users\_table(userId));";

        String queryVehicle = "CREATE TABLE "+VEHICLE\_TABLE+ " (VehicleId INTEGER PRIMARY KEY AUTOINCREMENT, numberPlate TEXT, ResidentId INTEGER, FOREIGN KEY (ResidentId) REFERENCES resident\_table(ResidentId));";

        db.execSQL(queryUser);

        db.execSQL(queryResident);

        db.execSQL(queryVehicle);

    }

    @Override

    public void onUpgrade(SQLiteDatabase db, int oldVersion, int newVersion) {

        db.execSQL("DROP TABLE IF EXISTS " + USER\_TABLE);

        db.execSQL("DROP TABLE IF EXISTS " + RESIDENT\_TABLE);

        db.execSQL("DROP TABLE IF EXISTS " + VEHICLE\_TABLE);

        onCreate(db);

    }

    public void addUser(String email, String name, String phonenumber){

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues cv = new ContentValues();

        cv.put("email", email);

        cv.put("name", name);

        cv.put("phonenumber", phonenumber);

        long result = db.insert(USER\_TABLE, null, cv);

        if(result == -1) {

            Toast.makeText(context, "Falied", Toast.LENGTH\_SHORT).show();

        }

        else {

            Toast.makeText(context, "Sucessful", Toast.LENGTH\_SHORT).show();

        }

        db.close();

    }

    public int addResident(String name, String email, String phoneno, int flatno, String numberPlate, int CurrentUserId){

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues cv = new ContentValues();

        cv.put("name", name);

        cv.put("email", email);

        cv.put("phoneno", phoneno);

        cv.put("flatno", flatno);

        cv.put("numberPlate", numberPlate);

        cv.put("userId", CurrentUserId);

        long result = db.insert(RESIDENT\_TABLE, null, cv);

        if(result == -1) {

            Toast.makeText(context, "Falied", Toast.LENGTH\_SHORT).show();

        }

        else {

            Toast.makeText(context, "Sucessful", Toast.LENGTH\_SHORT).show();

        }

        db.close();

        return 1;

    }

    public void addVehicle(String numberPlate, int ResidentId){

        SQLiteDatabase db = this.getWritableDatabase();

        ContentValues cv = new ContentValues();

        cv.put("numberPlate", numberPlate);

        cv.put("ResidentId", ResidentId);

        long result = db.insert(VEHICLE\_TABLE, null, cv);

        if(result == -1) {

            Toast.makeText(context, "Falied", Toast.LENGTH\_SHORT).show();

        }

        else {

            Toast.makeText(context, "Sucessful", Toast.LENGTH\_SHORT).show();

        }

        db.close();

    }

    public int getUserIdByEmail(String email){

        SQLiteDatabase db = this.getReadableDatabase();

        // Define the columns you want to retrieve

        String query = "SELECT userId FROM users\_table WHERE email=?;";

        Cursor cursor = db.rawQuery(query, new String[]{email});

        int userId = -1; // Default value if email not found

        if (cursor.moveToFirst()) {

            int columnIndex  = cursor.getColumnIndex("userId");

            if(columnIndex != -1) {

                userId = cursor.getInt(columnIndex);

            }

        }

        cursor.close();

        db.close();

        return userId;

    }

    public int getResidentId(String numberPlate){

        SQLiteDatabase db = this.getReadableDatabase();

        // Define the query to join the parent and child tables

        String query = "SELECT residentId FROM vehicle\_table WHERE numberPlate = ?;";

        Cursor cursor = db.rawQuery(query, new String[]{numberPlate});

        int residentId = -1;

        if(cursor.moveToFirst()) {

            int columnIndex = cursor.getColumnIndex("ResidentId");

            if (columnIndex != -1) {

                residentId = cursor.getInt(columnIndex);

            }

        }

        cursor.close();

        db.close();

        return residentId;

    }

    public String getResidentEmail(int residentId){

        SQLiteDatabase db = this.getReadableDatabase();

        String query = "SELECT email FROM resident\_table WHERE ResidentId = ?;";

        Cursor cursor = db.rawQuery(query, new String[]{String.valueOf(residentId)});

        String userEmail = "";

        if(cursor.moveToFirst()) {

            int columnIndex = cursor.getColumnIndex("email");

            if (columnIndex != -1) {

                userEmail = cursor.getString(columnIndex);

            }

        }

        cursor.close();

        db.close();

        return userEmail;

    }

    public int getUserId(int residentId){

        SQLiteDatabase db = this.getReadableDatabase();

        String query = "SELECT userId FROM resident\_table WHERE ResidentId =  ?;";

        Cursor cursor = db.rawQuery(query, new String[]{String.valueOf(residentId)});

        int userId = -1;

        if(cursor.moveToFirst()) {

            int columnIndex = cursor.getColumnIndex("userId");

            if (columnIndex != -1) {

                userId = cursor.getInt(columnIndex);

            }

        }

        cursor.close();

        db.close();

        return userId;

    }

    public String getUserEmail(int userId){

        SQLiteDatabase db = this.getReadableDatabase();

        String query = "SELECT email FROM users\_table WHERE userId= ?;";

        Cursor cursor = db.rawQuery(query, new String[]{String.valueOf(userId)});

        String userEmail = "";

        if(cursor.moveToFirst()) {

            int columnIndex = cursor.getColumnIndex("email");

            if (columnIndex != -1) {

                userEmail = cursor.getString(columnIndex);

            }

        }

        cursor.close();

        db.close();

        return userEmail;

    }

}

package com.security.communitysafee;

import android.os.Bundle;

import android.view.LayoutInflater;

import android.view.View;

import android.view.ViewGroup;

import androidx.annotation.NonNull;

import androidx.fragment.app.Fragment;

import androidx.navigation.fragment.NavHostFragment;

import com.security.communitysafee.databinding.FragmentSecondBinding;

public class SecondFragment extends Fragment {

private FragmentSecondBinding binding;

    @Override

    public View onCreateView(

            LayoutInflater inflater, ViewGroup container,

            Bundle savedInstanceState

    ) {

      binding = FragmentSecondBinding.inflate(inflater, container, false);

      return binding.getRoot();

    }

    public void onViewCreated(@NonNull View view, Bundle savedInstanceState) {

        super.onViewCreated(view, savedInstanceState);

        binding.buttonSecond.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                NavHostFragment.findNavController(SecondFragment.this)

                        .navigate(R.id.action\_SecondFragment\_to\_FirstFragment);

            }

        });

    }

@Override

    public void onDestroyView() {

        super.onDestroyView();

        binding = null;

    }

}

package com.security.communitysafee;

import androidx.annotation.Nullable;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Context;

import android.content.Intent;

import android.hardware.camera2.CameraCaptureSession;

import android.hardware.camera2.CameraManager;

import android.hardware.camera2.CaptureRequest;

import android.media.Image;

import android.os.Bundle;

import android.provider.MediaStore;

import android.view.View;

import android.widget.Button;

import android.widget.ImageButton;

public class SecurityAdminDashboard extends AppCompatActivity {

    ImageButton capture, editInfoButton;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_security\_admin\_dashboard);

        System.out.println(userId.CurrentUserId);

        capture = (ImageButton) findViewById(R.id.captureButton);

        capture.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                Intent intent = new Intent(getApplicationContext(), captureImage.class);

                startActivity(intent);

            }

        });

        editInfoButton = (ImageButton) findViewById(R.id.editInfoButton);

        editInfoButton.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View v) {

                Intent intent = new Intent(SecurityAdminDashboard.this, addInfo.class);

                startActivity(intent);

            }

        });

    }

}

package com.security.communitysafee;

import androidx.annotation.NonNull;

import androidx.appcompat.app.AppCompatActivity;

import android.content.Intent;

import android.os.Bundle;

import android.text.TextUtils;

import android.view.View;

import android.widget.Button;

import android.widget.EditText;

import android.widget.TextView;

import android.widget.Toast;

import com.google.android.gms.tasks.OnCompleteListener;

import com.google.android.gms.tasks.Task;

import com.google.firebase.auth.AuthResult;

import com.google.firebase.auth.FirebaseAuth;

public class signup\_security extends AppCompatActivity {

    private FirebaseAuth mAuth;

    private EditText emailField, passwordField, nameField, confirmPasswordField, phoneNumberField;

    @Override

    protected void onCreate(Bundle savedInstanceState) {

        super.onCreate(savedInstanceState);

        setContentView(R.layout.activity\_signup\_security);

        mAuth = FirebaseAuth.getInstance();

        TextView signin = (TextView) findViewById(R.id.loginSecuritySignin);

        Button signup = (Button) findViewById(R.id.signUpButton);

        signin.setOnClickListener(new View.OnClickListener() {

            @Override

            public void onClick(View view) {

                Intent intent = new Intent(view.getContext(), login\_security.class);

                view.getContext().startActivity(intent);

            }

        });

        signup.setOnClickListener(new View.OnClickListener(){

            @Override

            public void onClick(View view){

                createUser();

            }

        });

        }

    private void createUser(){

        String email, password, name, phoneNumber, confirmPassword;

        emailField = (EditText) findViewById(R.id.signupEmailField);

        nameField = (EditText) findViewById(R.id.societyNameField);

        passwordField = (EditText) findViewById(R.id.signupPasswordField);

        confirmPasswordField = (EditText) findViewById(R.id.signupConfirmPasswordField);

        phoneNumberField = (EditText) findViewById(R.id.signupPhonefield);

        email = getFieldsAsString(emailField);

        name = getFieldsAsString(nameField);

        phoneNumber = getFieldsAsString(phoneNumberField);

        password = getFieldsAsString(passwordField);

        confirmPassword = getFieldsAsString(confirmPasswordField);

        if(TextUtils.isEmpty(name)) {

            nameField.setError("Name cannot be empty");

            nameField.requestFocus();

        }

        else if(TextUtils.isEmpty(email)){

            emailField.setError("Email Cannot be empty");

            emailField.requestFocus();

        }

        else if(TextUtils.isEmpty(phoneNumber)) {

            phoneNumberField.setError("Email Cannot be empty");

            phoneNumberField.requestFocus();

        }

        else if(TextUtils.isEmpty(password)){

            passwordField.setError("Password cannot be empty");

            passwordField.requestFocus();

        }

        else if(!password.equals(confirmPassword)) {

            confirmPasswordField.setError("Dont match. Please try again");

            confirmPasswordField.requestFocus();

        }

        else {

            mAuth.createUserWithEmailAndPassword(email, password)

                    .addOnCompleteListener(this, new OnCompleteListener<AuthResult>() {

                        @Override

                        public void onComplete(@NonNull Task<AuthResult> task) {

                            if (task.isSuccessful()) {

                                Toast.makeText(getApplicationContext(), "Account created Sucessfully", Toast.LENGTH\_SHORT).show();

                                myDbHelper myDbHelper = new myDbHelper(signup\_security.this);

                                myDbHelper.addUser(email, name, phoneNumber);

                                Intent intent = new Intent(getApplicationContext(), login\_security.class);

                                startActivity(intent);

                                finish();

                            } else {

                                Toast.makeText(getApplicationContext(), "Couldnt create Account", Toast.LENGTH\_SHORT).show();

                            }

                        }

                    });

        }

    }

    private String getFieldsAsString(EditText field){

        return field.getText().toString();

    }

}

package com.security.communitysafee;

public class userId {

    public static int CurrentUserId;

    public static String CurrentUserEmail;

}

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".addInfo">

    <RelativeLayout

        android:layout\_width="372dp"

        android:layout\_height="124dp"

        app:layout\_constraintBottom\_toTopOf="@+id/relativeLayout5"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.487"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.447">

        <TextView

            android:id="@+id/textView7"

            android:layout\_width="317dp"

            android:layout\_height="42dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="6dp"

            android:layout\_marginTop="82dp"

            android:layout\_marginEnd="49dp"

            android:layout\_marginBottom="0dp"

            android:fontFamily="@font/abeezee"

            android:letterSpacing="0.02"

            android:text="Add resident Info"

            android:textColor="@color/orangeHeader"

            android:textSize="35sp"

            android:textStyle="bold" />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/relativeLayout5"

        android:layout\_width="383dp"

        android:layout\_height="630dp"

        android:layout\_marginBottom="16dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.32"

        app:layout\_constraintStart\_toStartOf="parent">

        <EditText

            android:id="@+id/fullname"

            android:layout\_width="303dp"

            android:layout\_height="56dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="36dp"

            android:layout\_marginTop="25dp"

            android:layout\_marginEnd="22dp"

            android:background="@color/orangeTextField"

            android:drawablePadding="5dp"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="Fullname"

            android:inputType="text"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/email"

            android:layout\_width="303dp"

            android:layout\_height="56dp"

            android:layout\_below="@+id/fullname"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="36dp"

            android:layout\_marginTop="23dp"

            android:layout\_marginEnd="22dp"

            android:background="@color/orangeTextField"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="Email"

            android:inputType="textEmailAddress"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/phoneNumber"

            android:layout\_width="303dp"

            android:layout\_height="56dp"

            android:layout\_below="@+id/email"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="36dp"

            android:layout\_marginTop="27dp"

            android:layout\_marginEnd="22dp"

            android:background="@color/orangeTextField"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="Phone number"

            android:inputType="phone"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/flatNo"

            android:layout\_width="303dp"

            android:layout\_height="56dp"

            android:layout\_below="@+id/phoneNumber"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="36dp"

            android:layout\_marginTop="36dp"

            android:layout\_marginEnd="22dp"

            android:background="@color/orangeTextField"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="Flat No"

            android:inputType="number"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <ImageButton

            android:id="@+id/capture"

            android:layout\_width="113dp"

            android:layout\_height="83dp"

            android:layout\_below="@+id/flatNo"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="124dp"

            android:layout\_marginTop="174dp"

            android:layout\_marginEnd="124dp"

            android:layout\_marginBottom="38dp"

            app:srcCompat="@drawable/camerabg"

            android:background="@drawable/login\_option"

            android:fontFamily="@font/abeezee"

            android:textColor="@color/white"/>

        <TextView

            android:id="@+id/numberPlateText"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_below="@+id/flatNo"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="81dp"

            android:layout\_marginTop="37dp"

            android:layout\_marginEnd="93dp"

            android:layout\_marginBottom="212dp"

            android:ems="10"

            android:textStyle="bold"

            android:inputType="text"

            android:text=""

            android:textColor="@color/orangeHeader"

            android:textSize="25dp"/>

        <Button

            android:id="@+id/savebtn"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_below="@+id/flatNo"

            android:layout\_alignBottom="@+id/capture"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="144dp"

            android:layout\_marginTop="102dp"

            android:layout\_marginEnd="151dp"

            android:layout\_marginBottom="108dp"

            android:text="SAVE"

            android:background="@drawable/login\_option"

            android:fontFamily="@font/abeezee"

            android:textColor="@color/white"/>

        <RelativeLayout

            android:id="@+id/DoneButton"

            android:layout\_width="366dp"

            android:layout\_height="606dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="17dp"

            android:layout\_marginTop="0dp"

            android:layout\_marginEnd="0dp"

            android:layout\_marginBottom="25dp"

            android:background="@color/white"

            android:elevation="2dp"

            android:orientation="horizontal"

            android:visibility="gone">

            <ImageView

                android:id="@+id/imageView"

                android:layout\_width="367dp"

                android:layout\_height="491dp"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentTop="true"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:layout\_marginStart="0dp"

                android:layout\_marginTop="0dp"

                android:layout\_marginEnd="0dp"

                android:layout\_marginBottom="115dp"

                android:layout\_weight="1"

                tools:srcCompat="@tools:sample/backgrounds/scenic" />

            <Button

                android:id="@+id/buttonDone"

                android:layout\_width="wrap\_content"

                android:layout\_height="wrap\_content"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:layout\_marginStart="69dp"

                android:layout\_marginEnd="73dp"

                android:layout\_marginBottom="33dp"

                android:background="@drawable/login\_option"

                android:fontFamily="@font/abeezee"

                android:text="Done"

                android:textColor="@color/white" />

        </RelativeLayout>

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".captureImage">

    <ImageView

        android:id="@+id/textureView"

        android:layout\_width="wrap\_content"

        android:layout\_height="526dp"

        android:layout\_alignParentStart="true"

        android:layout\_alignParentTop="true"

        android:layout\_alignParentEnd="true"

        android:layout\_centerHorizontal="true"

        android:layout\_marginStart="-1dp"

        android:layout\_marginTop="0dp"

        android:layout\_marginEnd="1dp" />

    <Button

        android:id="@+id/imageCaptureButton"

        android:layout\_width="284dp"

        android:layout\_height="wrap\_content"

        android:layout\_alignStart="@+id/textureView"

        android:layout\_alignParentEnd="true"

        android:layout\_alignParentBottom="true"

        android:layout\_centerHorizontal="true"

        android:layout\_marginStart="78dp"

        android:layout\_marginEnd="50dp"

        android:layout\_marginBottom="126dp"

        android:text="Take photo"

        android:background="@drawable/login\_option"

        android:fontFamily="@font/abeezee"

        android:textColor="@color/white"/>

    <Button

        android:id="@+id/donebtn"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_alignStart="@+id/textureView"

        android:layout\_alignEnd="@+id/textureView"

        android:layout\_alignParentEnd="true"

        android:layout\_alignParentBottom="true"

        android:layout\_centerHorizontal="true"

        android:layout\_marginStart="77dp"

        android:layout\_marginEnd="56dp"

        android:layout\_marginBottom="41dp"

        android:text="Done"

        android:background="@drawable/login\_option"

        android:fontFamily="@font/abeezee"

        android:textColor="@color/white"/>

    <TextView

        android:id="@+id/numberPlateTextCapture"

        android:layout\_width="304dp"

        android:layout\_height="wrap\_content"

        android:layout\_alignParentBottom="true"

        android:layout\_centerHorizontal="true"

        android:layout\_marginBottom="196dp"

        android:enabled="false"

        android:textColor="@color/orangeHeader"

        android:textSize="30dp"

        android:textStyle="bold" />

</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".SecurityAdminDashboard"

    android:background="@color/orangeBackground">

    <RelativeLayout

        android:id="@+id/editInfoHeader"

        android:layout\_width="311dp"

        android:layout\_height="134dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.456"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.071">

        <ImageView

            android:id="@+id/adminDashboardLogo"

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="0dp"

            android:layout\_marginEnd="0dp"

            android:layout\_marginBottom="0dp"

            app:srcCompat="@drawable/logobgrem" />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/editInfoDashboardOptions"

        android:layout\_width="353dp"

        android:layout\_height="418dp"

        android:layout\_marginTop="20dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/editInfoHeader"

        app:layout\_constraintVertical\_bias="0.038">

        <RelativeLayout

            android:id="@+id/addLayout"

            android:layout\_width="156dp"

            android:layout\_height="193dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="0dp">

            <ImageButton

                android:layout\_width="156dp"

                android:layout\_height="162dp"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentEnd="true"

                android:layout\_marginStart="-2dp"

                android:layout\_marginEnd="1dp"

                android:layout\_marginBottom="0dp"

                android:background="@drawable/cards"

                android:scaleType="fitCenter"

                android:src="@drawable/imagegallery" />

            <TextView

                android:layout\_width="match\_parent"

                android:layout\_height="35dp"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:layout\_marginEnd="0dp"

                android:layout\_marginBottom="0dp"

                android:background="@drawable/cardstext"

                android:fontFamily="@font/abeezee"

                android:gravity="center"

                android:text="ADD"

                android:textSize="20dp"

                android:textStyle="bold" />

        </RelativeLayout>

        <RelativeLayout

            android:id="@+id/editInfo"

            android:layout\_width="156dp"

            android:layout\_height="193dp"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="38dp"

            android:layout\_marginTop="2dp"

            android:layout\_marginEnd="2dp"

            android:layout\_toEndOf="@+id/addLayout">

            <ImageButton

                android:id="@+id/editInfoButton"

                android:layout\_width="match\_parent"

                android:layout\_height="159dp"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentTop="true"

                android:layout\_alignParentEnd="true"

                android:layout\_marginStart="0dp"

                android:layout\_marginTop="0dp"

                android:layout\_marginEnd="0dp"

                android:background="@drawable/cards"

                android:scaleType="fitCenter"

                android:src="@drawable/editt" />

            <TextView

                android:id="@+id/editInfoText"

                android:layout\_width="match\_parent"

                android:layout\_height="35dp"

                android:layout\_below="@+id/editInfoButton"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:layout\_marginStart="0dp"

                android:layout\_marginTop="0dp"

                android:layout\_marginEnd="0dp"

                android:layout\_marginBottom="0dp"

                android:background="@drawable/cardstext"

                android:fontFamily="@font/abeezee"

                android:gravity="center"

                android:text="EDIT"

                android:textSize="18dp"

                android:textStyle="bold" />

        </RelativeLayout>

        <RelativeLayout

            android:id="@+id/VehicleActivityLayout"

            android:layout\_width="156dp"

            android:layout\_height="193dp"

            android:layout\_below="@+id/addLayout"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="25dp"

            android:layout\_marginBottom="0dp">

            <ImageButton

                android:id="@+id/deleteButton"

                android:layout\_width="156dp"

                android:layout\_height="159dp"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentTop="true"

                android:layout\_alignParentEnd="true"

                android:layout\_marginStart="-1dp"

                android:layout\_marginTop="0dp"

                android:layout\_marginEnd="0dp"

                android:background="@drawable/cards"

                android:scaleType="centerInside"

                android:src="@drawable/bin" />

            <TextView

                android:id="@+id/deleteText"

                android:layout\_width="match\_parent"

                android:layout\_height="35dp"

                android:layout\_below="@+id/deleteButton"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:layout\_marginTop="0dp"

                android:layout\_marginEnd="0dp"

                android:layout\_marginBottom="0dp"

                android:background="@drawable/cardstext"

                android:fontFamily="@font/abeezee"

                android:gravity="center"

                android:text="DELETE"

                android:textSize="18dp"

                android:textStyle="bold" />

        </RelativeLayout>

        <RelativeLayout

            android:id="@+id/getDetailsLayout"

            android:layout\_width="156dp"

            android:layout\_height="193dp"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="47dp"

            android:layout\_marginEnd="3dp"

            android:layout\_marginBottom="1dp"

            android:layout\_toEndOf="@+id/VehicleActivityLayout">

            <ImageButton

                android:id="@+id/searchButton"

                android:layout\_width="match\_parent"

                android:layout\_height="156dp"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentTop="true"

                android:layout\_alignParentEnd="true"

                android:layout\_marginStart="0dp"

                android:layout\_marginTop="4dp"

                android:layout\_marginEnd="0dp"

                android:background="@drawable/cards"

                android:scaleType="centerInside"

                android:src="@drawable/seo" />

            <TextView

                android:id="@+id/searchText"

                android:layout\_width="match\_parent"

                android:layout\_height="32dp"

                android:layout\_below="@+id/searchButton"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:layout\_marginTop="-2dp"

                android:layout\_marginEnd="0dp"

                android:layout\_marginBottom="0dp"

                android:background="@drawable/cardstext"

                android:fontFamily="@font/abeezee"

                android:gravity="center"

                android:text="SEARCH"

                android:textSize="18dp"

                android:textStyle="bold" />

        </RelativeLayout>

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".email\_for\_password\_change">

    <RelativeLayout

        android:id="@+id/enterEmailRelativeLayoutHeader"

        android:layout\_width="381dp"

        android:layout\_height="165dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.079">

        <TextView

            android:id="@+id/textView5"

            android:layout\_width="341dp"

            android:layout\_height="35dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="21dp"

            android:layout\_marginTop="122dp"

            android:layout\_marginEnd="25dp"

            android:layout\_marginBottom="5dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/enterEmail"

            android:textColor="@color/orangeHeader"

            android:textSize="20sp"

            android:textStyle="bold" />

    </RelativeLayout>

    <EditText

        android:id="@+id/editTextTextEmailAddress2"

        android:layout\_width="340dp"

        android:layout\_height="39dp"

        android:background="@color/orangeTextField"

        android:drawableLeft="@drawable/baseline\_email\_24"

        android:drawablePadding="5dp"

        android:ems="10"

        android:fontFamily="@font/abeezee"

        android:hint="@string/emailField"

        android:inputType="textEmailAddress"

        android:padding="8dp"

        android:textColorHint="@color/grey"

        android:textStyle="bold"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/enterEmailRelativeLayoutHeader"

        app:layout\_constraintVertical\_bias="0.0" />

    <Button

        android:id="@+id/emailChangeNextButton"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:background="@drawable/login\_option"

        android:text="@string/next"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.885"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/editTextTextEmailAddress2"

        app:layout\_constraintVertical\_bias="0.037"

        android:textColor="@color/white"/>

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".email\_for\_password\_resident">

    <RelativeLayout

        android:id="@+id/enterEmailResidentRelativeLayoutHeader"

        android:layout\_width="381dp"

        android:layout\_height="165dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.079">

        <TextView

            android:id="@+id/enterEmailRelativeLayoutSubHeader"

            android:layout\_width="341dp"

            android:layout\_height="35dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="21dp"

            android:layout\_marginTop="122dp"

            android:layout\_marginEnd="25dp"

            android:layout\_marginBottom="5dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/enterEmail"

            android:textColor="@color/orangeHeader"

            android:textSize="20sp"

            android:textStyle="bold" />

    </RelativeLayout>

    <EditText

        android:id="@+id/enterEmailRelativeLayoutEmailField"

        android:layout\_width="340dp"

        android:layout\_height="39dp"

        android:background="@color/orangeTextField"

        android:drawableLeft="@drawable/baseline\_email\_24"

        android:drawablePadding="5dp"

        android:ems="10"

        android:fontFamily="@font/abeezee"

        android:hint="@string/emailField"

        android:inputType="textEmailAddress"

        android:padding="8dp"

        android:textColorHint="@color/grey"

        android:textStyle="bold"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/enterEmailResidentRelativeLayoutHeader"

        app:layout\_constraintVertical\_bias="0.0" />

    <Button

        android:id="@+id/emailChangeResidentNextButton"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:background="@drawable/login\_option"

        android:text="@string/next"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.885"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/enterEmailRelativeLayoutEmailField"

        app:layout\_constraintVertical\_bias="0.037"

        android:textColor="@color/white"/>

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical"

    tools:context=".ForgetPasswordKey">

    <RelativeLayout

        android:id="@+id/VerifyCoderelativeLayoutHeader"

        android:layout\_width="412dp"

        android:layout\_height="162dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.118">

        <TextView

            android:id="@+id/verifyCodeHeader"

            android:layout\_width="wrap\_content"

            android:layout\_height="52dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="23dp"

            android:layout\_marginTop="36dp"

            android:layout\_marginEnd="53dp"

            android:layout\_marginBottom="74dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/verifyCode"

            android:textColor="@color/orangeHeader"

            android:textSize="45sp"

            android:textStyle="bold" />

        <TextView

            android:id="@+id/verifyCodeSubHeader"

            android:layout\_width="372dp"

            android:layout\_height="24dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="26dp"

            android:layout\_marginEnd="13dp"

            android:layout\_marginBottom="46dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/provideOTP"

            android:textSize="20sp"

            android:textStyle="bold" />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/VerifyCodeRelativeLayoutText"

        android:layout\_width="358dp"

        android:layout\_height="88dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.487"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/VerifyCoderelativeLayoutHeader"

        app:layout\_constraintVertical\_bias="0.0">

        <EditText

            android:id="@+id/verifyCodeText1"

            android:layout\_width="72dp"

            android:layout\_height="match\_parent"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="0dp"

            android:layout\_marginBottom="0dp"

            android:ems="10"

            android:inputType="number"

            android:background="@color/orangeTextField"/>

        <EditText

            android:id="@+id/verifyCodeText2"

            android:layout\_width="72dp"

            android:layout\_height="match\_parent"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="21dp"

            android:layout\_marginTop="0dp"

            android:layout\_marginBottom="0dp"

            android:layout\_toEndOf="@+id/verifyCodeText1"

            android:ems="10"

            android:inputType="number"

            android:background="@color/orangeTextField"/>

        <EditText

            android:id="@+id/verifyCodeText3"

            android:layout\_width="72dp"

            android:layout\_height="match\_parent"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="21dp"

            android:layout\_marginTop="1dp"

            android:layout\_marginBottom="-1dp"

            android:layout\_toEndOf="@+id/verifyCodeText2"

            android:ems="10"

            android:inputType="number"

            android:background="@color/orangeTextField"/>

        <EditText

            android:id="@+id/verifyCodeText4"

            android:layout\_width="72dp"

            android:layout\_height="match\_parent"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="21dp"

            android:layout\_marginTop="0dp"

            android:layout\_marginEnd="2dp"

            android:layout\_marginBottom="0dp"

            android:layout\_toEndOf="@+id/verifyCodeText3"

            android:ems="10"

            android:inputType="number"

            android:background="@color/orangeTextField"/>

    </RelativeLayout>

    <Button

        android:id="@+id/verifyCodeVerifyButton"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:background="@drawable/login\_option"

        android:fontFamily="@font/abeezee"

        android:text="@string/verify"

        android:textColor="@color/white"

        android:textStyle="bold"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.885"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/VerifyCodeRelativeLayoutText"

        app:layout\_constraintVertical\_bias="0.065" />

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_height="match\_parent"

    android:layout\_width="wrap\_content"

    tools:context=".LoginOptions"

    android:orientation="vertical">

    <LinearLayout

        android:id="@+id/linearLayout"

        android:layout\_width="match\_parent"

        android:layout\_height="300dp"

        android:gravity="center"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="1.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.153">

        <ImageView

            android:id="@+id/imageView5"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_weight="1"

            app:srcCompat="@drawable/logobgrem" />

    </LinearLayout>

    <Button

        android:id="@+id/buttonLoginAsResident"

        android:layout\_width="325dp"

        android:layout\_height="64dp"

        android:layout\_marginTop="116dp"

        android:background="@drawable/login\_option"

        android:fontFamily="sans-serif"

        android:gravity="center"

        android:letterSpacing="0.12"

        android:text="@string/loginAsResident"

        android:textColor="@color/white"

        android:textStyle="bold"

        app:layout\_constraintBottom\_toTopOf="@+id/button8"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.604"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/linearLayout"

        app:layout\_constraintVertical\_bias="0.0" />

    <Button

        android:id="@+id/buttonLoginAsSecurity"

        android:layout\_width="325dp"

        android:layout\_height="64dp"

        android:layout\_marginBottom="56dp"

        android:background="@drawable/login\_option"

        android:fontFamily="sans-serif"

        android:gravity="center"

        android:letterSpacing="0.12"

        android:text="@string/loginAsSecurity"

        android:textColor="@color/white"

        android:textStyle="bold"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.604"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/buttonLoginAsResident"

        app:layout\_constraintVertical\_bias="0.234" />

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:backgroundTint="#F4ECEC"

    tools:context=".login\_resident">

    <RelativeLayout

        android:id="@+id/textView"

        android:layout\_width="match\_parent"

        android:layout\_height="230dp"

        android:layout\_marginTop="28dp"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent">

        <TextView

            android:id="@+id/textView2"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="29dp"

            android:layout\_marginTop="89dp"

            android:layout\_marginEnd="139dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/Login"

            android:textAppearance="@style/TextAppearance.AppCompat.Large"

            android:textColor="@color/orangeHeader"

            android:textSize="60sp"

            android:textStyle="bold" />

        <TextView

            android:id="@+id/textView4"

            android:layout\_width="246dp"

            android:layout\_height="40dp"

            android:layout\_below="@+id/textView2"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="36dp"

            android:layout\_marginTop="14dp"

            android:layout\_marginEnd="129dp"

            android:layout\_marginBottom="31dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/PleaseSignIn"

            android:textSize="18sp"

            android:textStyle="bold" />

        />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/relativeLayout2"

        android:layout\_width="match\_parent"

        android:layout\_height="125dp"

        android:layout\_marginTop="16dp"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.491"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/textView">

        <EditText

            android:id="@+id/editTextTextEmailAddress"

            android:layout\_width="wrap\_content"

            android:layout\_height="52dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="31dp"

            android:layout\_marginTop="58dp"

            android:layout\_marginEnd="35dp"

            android:layout\_marginBottom="15dp"

            android:background="@color/orangeTextField"

            android:drawableLeft="@drawable/baseline\_email\_24"

            android:drawablePadding="5dp"

            android:paddingLeft="5dp"

            android:ems="10"

            android:hint="@string/emailField"

            android:inputType="textEmailAddress"

            android:textColor="@color/black"

            android:textColorHint="@color/grey"

            android:textStyle="bold"/>

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/relativeLayout"

        android:layout\_width="match\_parent"

        android:layout\_height="100dp"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="1.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout2"

        app:layout\_constraintVertical\_bias="0.0">

        <EditText

            android:id="@+id/editTextTextPassword"

            android:layout\_width="349dp"

            android:layout\_height="52dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="30dp"

            android:layout\_marginTop="23dp"

            android:layout\_marginEnd="32dp"

            android:layout\_marginBottom="25dp"

            android:background="@color/orangeTextField"

            android:drawableTint="#070707"

            android:ems="10"

            android:hint="@string/passwordField"

            android:inputType="textPassword"

            android:textColor="@color/black"

            android:textColorHint="@color/grey"

            android:textStyle="bold"

            tools:layout\_editor\_absoluteX="49dp"

            tools:layout\_editor\_absoluteY="514dp"

            android:drawableLeft="@drawable/baseline\_lock\_24"

            android:drawablePadding="5dp"

            android:paddingLeft="5dp"/>

    </RelativeLayout>

    <TextView

        android:id="@+id/forgotPasswordResident"

        android:layout\_width="206dp"

        android:layout\_height="41dp"

        android:fontFamily="@font/abeezee"

        android:text="@string/forgotPassword"

        android:textColor="@color/buttonLink"

        android:textSize="18dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toStartOf="@+id/button"

        app:layout\_constraintHorizontal\_bias="0.421"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout"

        app:layout\_constraintVertical\_bias="0.036" />

    <Button

        android:id="@+id/button"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:background="@drawable/login\_option"

        android:text="@string/LoginButton"

        android:textColor="@color/white"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.894"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout"

        app:layout\_constraintVertical\_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:backgroundTint="#F4ECEC"

    tools:context=".login\_resident">

    <RelativeLayout

        android:id="@+id/loginSecurityRelativeLayout"

        android:layout\_width="match\_parent"

        android:layout\_height="230dp"

        android:layout\_marginTop="28dp"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent">

        <TextView

            android:id="@+id/LoginTextSecurity"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="29dp"

            android:layout\_marginTop="89dp"

            android:layout\_marginEnd="139dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/Login"

            android:textAppearance="@style/TextAppearance.AppCompat.Large"

            android:textColor="@color/orangeHeader"

            android:textSize="60sp"

            android:textStyle="bold"  />

        <TextView

            android:id="@+id/PleaseSigninSecurity"

            android:layout\_width="246dp"

            android:layout\_height="40dp"

            android:layout\_below="@+id/LoginTextSecurity"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="36dp"

            android:layout\_marginTop="14dp"

            android:layout\_marginEnd="129dp"

            android:layout\_marginBottom="31dp"

            android:fontFamily="@font/abeezee"

            android:text="@string/PleaseSignIn"

            android:textSize="18sp"

            android:textStyle="bold" />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/LoginSecurityEmailLayout"

        android:layout\_width="match\_parent"

        android:layout\_height="125dp"

        android:layout\_marginTop="16dp"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.491"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/loginSecurityRelativeLayout">

        <EditText

            android:id="@+id/editTextTextEmailAddress"

            android:layout\_width="wrap\_content"

            android:layout\_height="46dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="31dp"

            android:layout\_marginTop="58dp"

            android:layout\_marginEnd="35dp"

            android:layout\_marginBottom="21dp"

            android:background="@color/orangeTextField"

            android:drawableTint="@color/drawableTint"

            android:drawableLeft="@drawable/baseline\_email\_24"

            android:drawablePadding="5dp"

            android:paddingLeft="5dp"

            android:ems="10"

            android:hint="@string/emailField"

            android:inputType="textEmailAddress"

            android:textColor="@color/black"

            android:textColorHint="@color/grey"

            android:textStyle="bold"

            android:fontFamily="@font/abeezee"/>

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/relativeLayout"

        android:layout\_width="match\_parent"

        android:layout\_height="100dp"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="1.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/LoginSecurityEmailLayout">

        <EditText

            android:id="@+id/editTextTextPassword"

            android:layout\_width="wrap\_content"

            android:layout\_height="46dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="30dp"

            android:layout\_marginTop="23dp"

            android:layout\_marginEnd="32dp"

            android:layout\_marginBottom="31dp"

            android:background="@color/orangeTextField"

            android:drawableTint="@color/drawableTint"

            android:ems="10"

            android:drawableLeft="@drawable/baseline\_lock\_24"

            android:drawablePadding="5dp"

            android:paddingLeft="5dp"

            android:fontFamily="@font/abeezee"

            android:hint="@string/passwordField"

            android:inputType="textPassword"

            android:textColor="@color/black"

            android:textColorHint="@color/grey"

            android:textStyle="bold"

            tools:layout\_editor\_absoluteX="49dp"

            tools:layout\_editor\_absoluteY="514dp" />

    </RelativeLayout>

    <Button

        android:id="@+id/buttonLoginSecurity"

        android:layout\_width="102dp"

        android:layout\_height="45dp"

        android:background="@drawable/login\_option"

        android:fontFamily="@font/abeezee"

        android:text="@string/LoginButton"

        android:textColor="@color/white"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.894"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout"

        app:layout\_constraintVertical\_bias="0.0" />

    <RelativeLayout

        android:id="@+id/linearLayoutSignup"

        android:layout\_width="399dp"

        android:layout\_height="54dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.0"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout"

        app:layout\_constraintVertical\_bias="0.393">

        <TextView

            android:id="@+id/loginSecurityNoAccount"

            android:layout\_width="wrap\_content"

            android:layout\_height="wrap\_content"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginEnd="149dp"

            android:layout\_marginBottom="13dp"

            android:fontFamily="@font/abeezee"

            android:gravity="center"

            android:text="@string/dontHaveAccount"

            android:textSize="20sp"

            android:textStyle="bold"

            tools:text="Don't have an account?" />

        <TextView

            android:id="@+id/loginSecuritySignUp"

            android:layout\_width="195dp"

            android:layout\_height="match\_parent"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginEnd="-10dp"

            android:layout\_marginBottom="-6dp"

            android:clickable="true"

            android:fontFamily="@font/abeezee"

            android:gravity="center"

            android:text="@string/Signupnow"

            android:textColor="@color/buttonLink"

            android:textSize="20sp" />

    </RelativeLayout>

    <TextView

        android:id="@+id/forgotPasswordSecurity"

        android:layout\_width="183dp"

        android:layout\_height="39dp"

        android:fontFamily="@font/abeezee"

        android:text="@string/forgotPassword"

        android:textColor="@color/buttonLink"

        android:textSize="18dp"

        app:layout\_constraintBottom\_toTopOf="@+id/linearLayoutSignup"

        app:layout\_constraintEnd\_toStartOf="@+id/buttonLoginSecurity"

        app:layout\_constraintHorizontal\_bias="0.361"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout"

        app:layout\_constraintVertical\_bias="0.0" />

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:fitsSystemWindows="true"

    tools:context=".MainActivity">

  <ImageView

      android:id="@+id/imageView3"

      android:layout\_width="match\_parent"

      android:layout\_height="match\_parent"

      android:layout\_alignParentStart="true"

      android:layout\_alignParentTop="true"

      android:layout\_alignParentEnd="true"

      android:layout\_alignParentBottom="true"

      android:layout\_centerHorizontal="true"

      android:layout\_marginStart="85dp"

      android:layout\_marginTop="186dp"

      android:layout\_marginEnd="83dp"

      android:layout\_marginBottom="288dp"

      app:srcCompat="@drawable/logobgrem" />

</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".SecurityAdminDashboard"

    android:background="@color/orangeBackground">

    <RelativeLayout

        android:id="@+id/securityDashboardHeader"

        android:layout\_width="311dp"

        android:layout\_height="134dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.456"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.071">

        <ImageView

            android:id="@+id/adminDashboardLogo"

            android:layout\_width="match\_parent"

            android:layout\_height="match\_parent"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="0dp"

            android:layout\_marginEnd="0dp"

            android:layout\_marginBottom="0dp"

            app:srcCompat="@drawable/logobgrem" />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/editInfoDashboardOptions"

        android:layout\_width="361dp"

        android:layout\_height="504dp"

        android:layout\_marginTop="20dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/securityDashboardHeader"

        app:layout\_constraintVertical\_bias="0.038">

        <RelativeLayout

            android:id="@+id/addLayout"

            android:layout\_width="226dp"

            android:layout\_height="215dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="74dp"

            android:layout\_marginTop="2dp"

            android:layout\_marginEnd="61dp">

            <TextView

                android:layout\_width="match\_parent"

                android:layout\_height="35dp"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:background="@drawable/cardstext"

                android:fontFamily="@font/abeezee"

                android:gravity="center"

                android:text="Capture"

                android:textSize="20dp"

                android:textStyle="bold" />

            <ImageButton

                android:id="@+id/captureButton"

                android:layout\_width="match\_parent"

                android:layout\_height="181dp"

                android:layout\_alignParentTop="true"

                android:layout\_alignParentEnd="true"

                android:background="@drawable/cards"

                android:scaleType="fitCenter"

                android:src="@drawable/recorder" />

        </RelativeLayout>

        <RelativeLayout

            android:id="@+id/ResidentDetails"

            android:layout\_width="221dp"

            android:layout\_height="228dp"

            android:layout\_below="@+id/addLayout"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="79dp"

            android:layout\_marginTop="49dp"

            android:layout\_marginEnd="61dp"

            android:layout\_marginBottom="11dp">

            <ImageButton

                android:id="@+id/editInfoButton"

                android:layout\_width="197dp"

                android:layout\_height="181dp"

                android:layout\_alignParentStart="true"

                android:layout\_alignParentTop="true"

                android:layout\_alignParentEnd="true"

                android:background="@drawable/cards"

                android:scaleType="fitCenter"

                android:src="@drawable/edit" />

            <TextView

                android:layout\_width="263dp"

                android:layout\_height="34dp"

                android:layout\_below="@+id/editInfoButton"

                android:layout\_alignParentEnd="true"

                android:layout\_alignParentBottom="true"

                android:background="@drawable/cardstext"

                android:fontFamily="@font/abeezee"

                android:gravity="center"

                android:text="Resident Details"

                android:textSize="18dp"

                android:textStyle="bold" />

        </RelativeLayout>

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    android:orientation="vertical"

    tools:context=".signup\_security">

    <RelativeLayout

        android:id="@+id/relativeLayout3"

        android:layout\_width="397dp"

        android:layout\_height="131dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.691"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintVertical\_bias="0.052">

        <TextView

            android:id="@+id/textView3"

            android:layout\_width="wrap\_content"

            android:layout\_height="63dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="10dp"

            android:layout\_marginTop="48dp"

            android:layout\_marginEnd="46dp"

            android:layout\_marginBottom="20dp"

            android:fontFamily="@font/abeezee"

            android:letterSpacing="0.02"

            android:text="@string/createAccount"

            android:textColor="@color/orangeHeader"

            android:textSize="45sp"

            android:textStyle="bold" />

    </RelativeLayout>

    <RelativeLayout

        android:id="@+id/relativeLayout4"

        android:layout\_width="341dp"

        android:layout\_height="347dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.454"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout3"

        app:layout\_constraintVertical\_bias="0.073">

        <EditText

            android:id="@+id/societyNameField"

            android:layout\_width="match\_parent"

            android:layout\_height="46dp"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="15dp"

            android:layout\_marginEnd="0dp"

            android:background="@color/orangeTextField"

            android:drawableLeft="@drawable/baseline\_home\_work\_24"

            android:drawablePadding="5dp"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="@string/societyName"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/signupEmailField"

            android:layout\_width="match\_parent"

            android:layout\_height="46dp"

            android:layout\_below="@+id/societyNameField"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="22dp"

            android:layout\_marginEnd="0dp"

            android:background="@color/orangeTextField"

            android:drawableLeft="@drawable/baseline\_email\_24"

            android:drawablePadding="5dp"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="@string/emailField"

            android:inputType="text"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/signupPhonefield"

            android:layout\_width="match\_parent"

            android:layout\_height="46dp"

            android:layout\_below="@+id/signupEmailField"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="19dp"

            android:layout\_marginEnd="0dp"

            android:background="@color/orangeTextField"

            android:drawableLeft="@drawable/baseline\_phone\_24"

            android:drawablePadding="5dp"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="@string/phoneField"

            android:inputType="phone"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/signupPasswordField"

            android:layout\_width="match\_parent"

            android:layout\_height="46dp"

            android:layout\_below="@+id/signupPhonefield"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="32dp"

            android:layout\_marginEnd="0dp"

            android:background="@color/orangeTextField"

            android:drawableLeft="@drawable/baseline\_lock\_24"

            android:drawablePadding="5dp"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="@string/passwordField"

            android:inputType="textPassword"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

        <EditText

            android:id="@+id/signupConfirmPasswordField"

            android:layout\_width="match\_parent"

            android:layout\_height="46dp"

            android:layout\_below="@+id/signupPasswordField"

            android:layout\_alignParentStart="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginStart="0dp"

            android:layout\_marginTop="19dp"

            android:layout\_marginEnd="0dp"

            android:background="@color/orangeTextField"

            android:drawableLeft="@drawable/baseline\_lock\_24"

            android:drawablePadding="5dp"

            android:ems="10"

            android:fontFamily="@font/abeezee"

            android:hint="@string/confirmPassword"

            android:inputType="textPassword"

            android:paddingLeft="8dp"

            android:textColorHint="@color/grey"

            android:textStyle="bold" />

    </RelativeLayout>

    <Button

        android:id="@+id/signUpButton"

        android:layout\_width="114dp"

        android:layout\_height="52dp"

        android:layout\_marginTop="40dp"

        android:background="@drawable/login\_option"

        android:text="@string/buttonSignup"

        android:textColor="@color/white"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.882"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/relativeLayout4" />

    <RelativeLayout

        android:id="@+id/RelativeLayoutSignup"

        android:layout\_width="379dp"

        android:layout\_height="51dp"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintHorizontal\_bias="0.875"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@+id/signUpButton"

        app:layout\_constraintVertical\_bias="0.387">

        <TextView

            android:id="@+id/loginSecurityNoAccount"

            android:layout\_width="265dp"

            android:layout\_height="35dp"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginEnd="114dp"

            android:layout\_marginBottom="16dp"

            android:fontFamily="@font/abeezee"

            android:gravity="center"

            android:text="@string/alreadyHaveAnAccount"

            android:textSize="20sp"

            android:textStyle="bold" />

        <TextView

            android:id="@+id/loginSecuritySignin"

            android:layout\_width="103dp"

            android:layout\_height="45dp"

            android:layout\_alignParentTop="true"

            android:layout\_alignParentEnd="true"

            android:layout\_alignParentBottom="true"

            android:layout\_marginTop="-5dp"

            android:layout\_marginEnd="35dp"

            android:layout\_marginBottom="11dp"

            android:clickable="true"

            android:fontFamily="@font/abeezee"

            android:gravity="center"

            android:text="@string/signInNow"

            android:textColor="@color/buttonLink"

            android:textSize="20sp" />

    </RelativeLayout>

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.constraintlayout.widget.ConstraintLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    app:layout\_behavior="@string/appbar\_scrolling\_view\_behavior">

    <fragment

        android:id="@+id/nav\_host\_fragment\_content\_main"

        android:name="androidx.navigation.fragment.NavHostFragment"

        android:layout\_width="0dp"

        android:layout\_height="0dp"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintTop\_toTopOf="parent"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:defaultNavHost="true"

        app:navGraph="@navigation/nav\_graph" />

</androidx.constraintlayout.widget.ConstraintLayout>

<?xml version="1.0" encoding="utf-8"?>

<RelativeLayout

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".FirstFragment">

    <RelativeLayout

        android:layout\_width="match\_parent"

        android:theme="@style/Theme.AppCompat.Light.NoActionBar"

        android:background="@color/white"

        android:layout\_height="match\_parent"

        android:padding="16dp">

        <ImageView

            android:id="@+id/imageView2"

            android:layout\_width="194dp"

            android:layout\_height="325dp"

            android:src="@drawable/logobgrem" />

    </RelativeLayout>

</RelativeLayout>

<?xml version="1.0" encoding="utf-8"?>

<androidx.core.widget.NestedScrollView

    xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:app="http://schemas.android.com/apk/res-auto"

    xmlns:tools="http://schemas.android.com/tools"

    android:layout\_width="match\_parent"

    android:layout\_height="match\_parent"

    tools:context=".SecondFragment">

    <androidx.constraintlayout.widget.ConstraintLayout

        android:layout\_width="match\_parent"

        android:layout\_height="match\_parent"

        android:padding="16dp">

    <Button

        android:id="@+id/button\_second"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:text="@string/previous"

        app:layout\_constraintBottom\_toTopOf="@id/textview\_second"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toTopOf="parent" />

    <TextView

        android:id="@+id/textview\_second"

        android:layout\_width="wrap\_content"

        android:layout\_height="wrap\_content"

        android:layout\_marginTop="16dp"

        android:text="@string/lorem\_ipsum"

        app:layout\_constraintBottom\_toBottomOf="parent"

        app:layout\_constraintEnd\_toEndOf="parent"

        app:layout\_constraintStart\_toStartOf="parent"

        app:layout\_constraintTop\_toBottomOf="@id/button\_second" />

    </androidx.constraintlayout.widget.ConstraintLayout>

</androidx.core.widget.NestedScrollView>

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android"

    xmlns:tools="http://schemas.android.com/tools">

    <uses-permission android:name="android.permission.INTERNET" />

    <uses-permission android:name="android.permission.CAMERA" />

    <uses-permission android:name="android.permission.WRITE\_EXTERNAL\_STORAGE" />

    <uses-feature android:name="android.hardware.camera.any" />

    <application

        android:allowBackup="true"

        android:dataExtractionRules="@xml/data\_extraction\_rules"

        android:fullBackupContent="@xml/backup\_rules"

        android:icon="@mipmap/ic\_launcher"

        android:label="@string/app\_name"

        android:roundIcon="@mipmap/ic\_launcher\_round"

        android:supportsRtl="true"

        android:theme="@style/Theme.CommunitySafee"

        tools:targetApi="31"

        android:usesCleartextTraffic="true"

        >

        <activity

            android:name=".addInfo"

            android:exported="false" />

        <activity

            android:name=".captureImage"

            android:exported="false" />

        <activity

            android:name=".EditInfoAdminDashboard"

            android:exported="false" />

        <activity

            android:name=".SecurityAdminDashboard"

            android:exported="false" />

        <activity

            android:name=".email\_for\_password\_resident"

            android:exported="false" />

        <activity

            android:name=".email\_for\_password\_change"

            android:exported="false" />

        <activity

            android:name=".ForgetPasswordKey"

            android:exported="false" />

        <activity

            android:name=".signup\_security"

            android:exported="false" />

        <activity

            android:name=".login\_security"

            android:exported="false" />

        <activity

            android:name=".login\_resident"

            android:exported="false" />

        <activity

            android:name=".LoginOptions"

            android:exported="false" />

        <activity

            android:name=".MainActivity"

            android:exported="true"

            android:label="@string/app\_name"

            android:theme="@style/Theme.CommunitySafee">

            <intent-filter>

                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />

            </intent-filter>

        </activity>

        <meta-data

            android:name="preloaded\_fonts"

            android:resource="@array/preloaded\_fonts" />

    </application>

</manifest>

package com.security.communitysafee;

import org.junit.Test;

import static org.junit.Assert.\*;

/\*\*

 \* Example local unit test, which will execute on the development machine (host).

 \*

 \* @see <a href="http://d.android.com/tools/testing">Testing documentation</a>

 \*/

public class ExampleUnitTest {

    @Test

    public void addition\_isCorrect() {

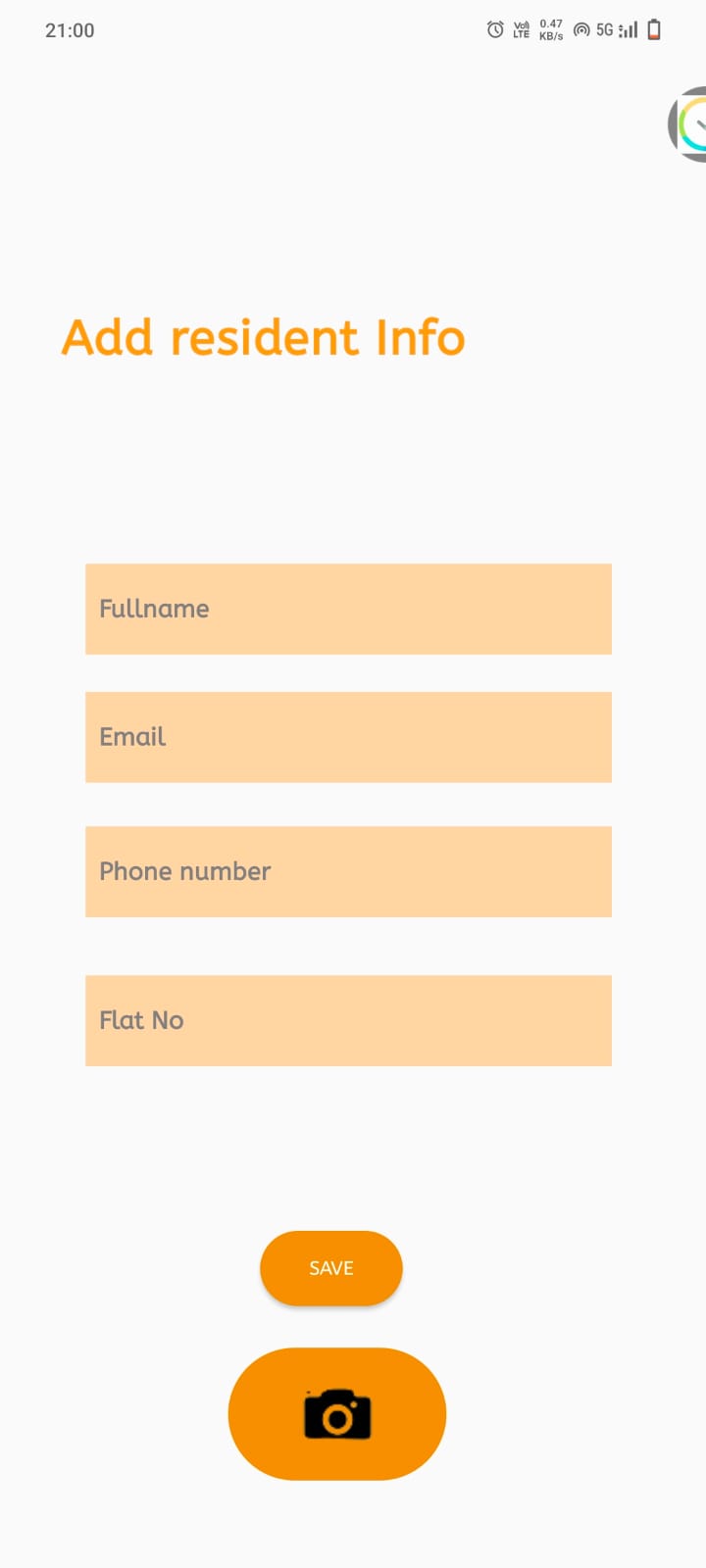
        assertEquals(4, 2 + 2);

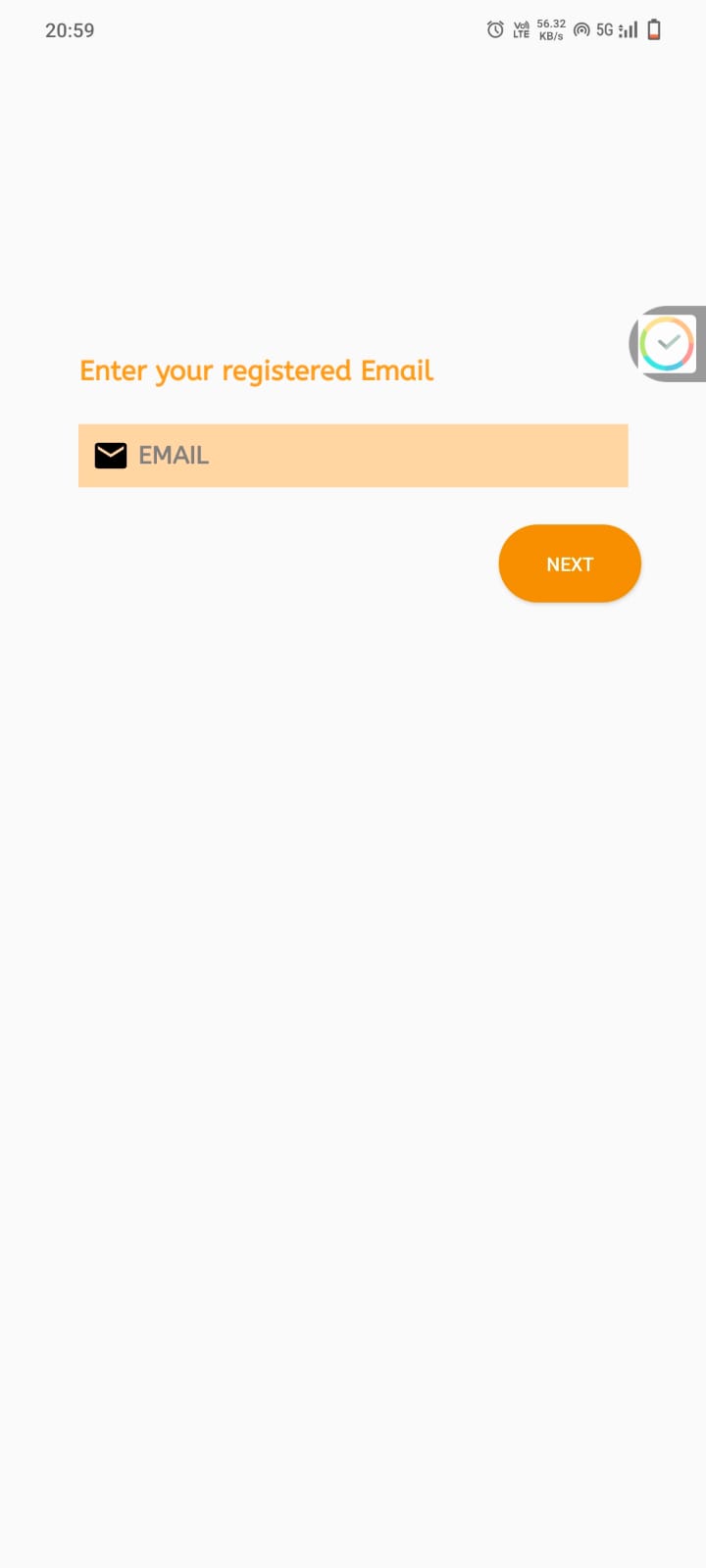
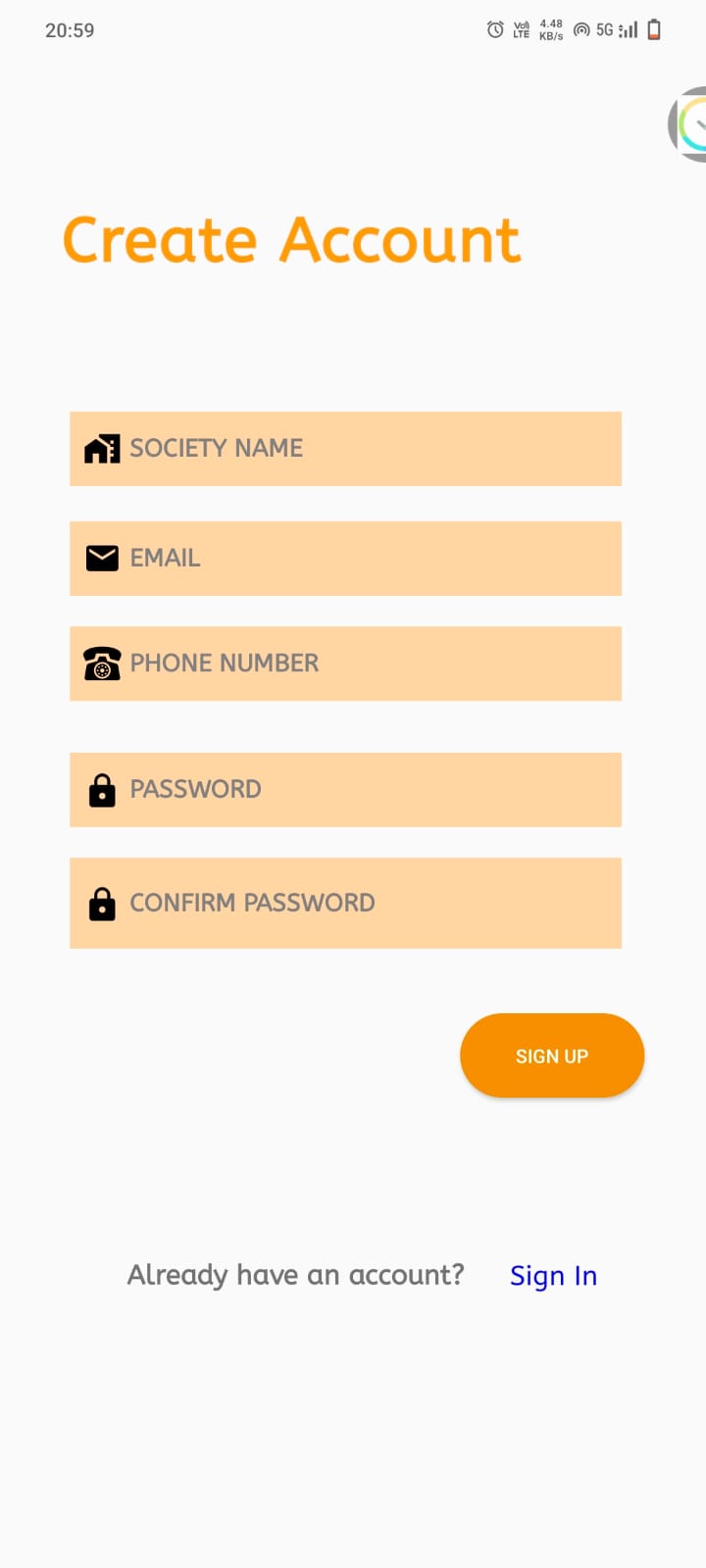
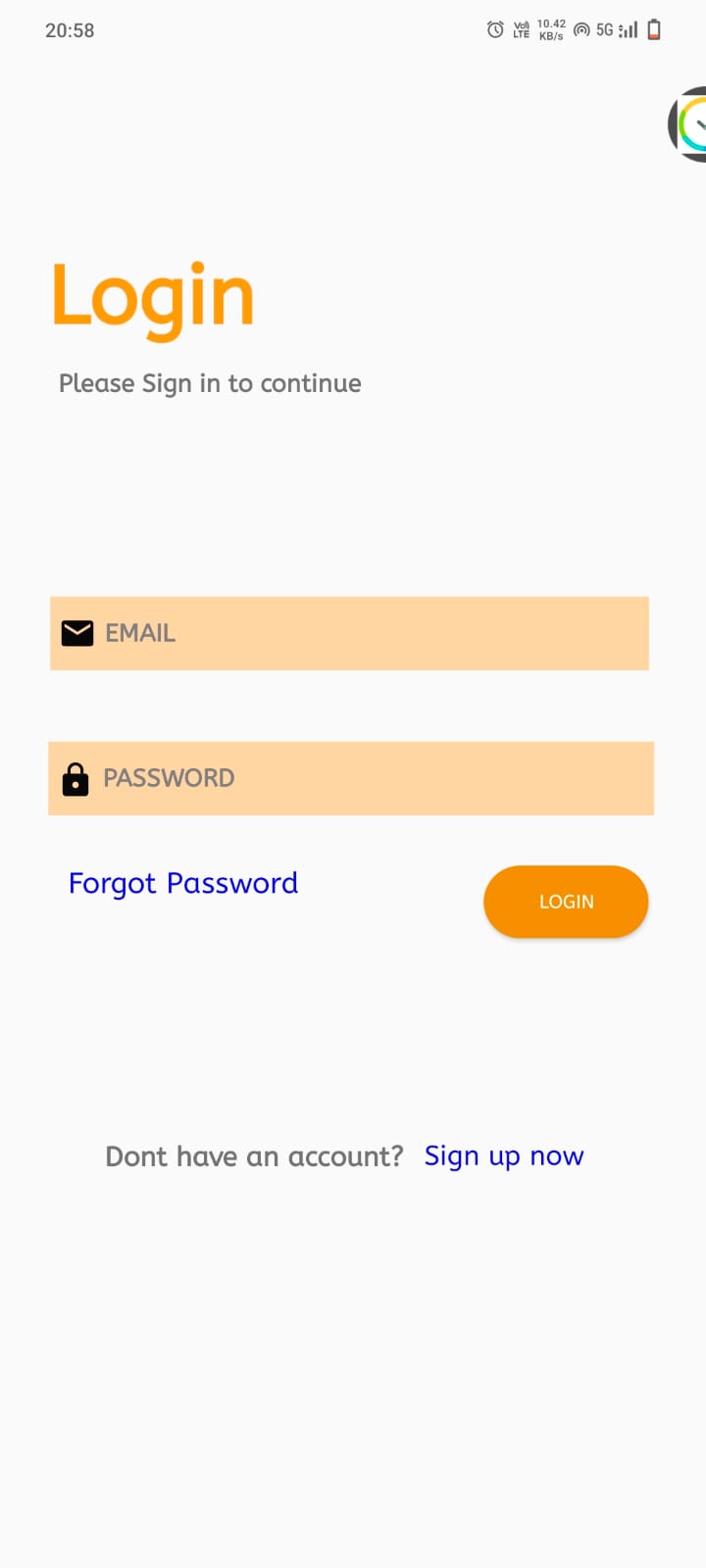
    }

}

**APPENDIX-B**

**SCREENSHOTS**





**APPENDIX-C**

**ENCLOSURES**





