A logo with a tower and a building in the middle

Description automatically generatedMINISTRY OF EDUCATION AND TRAINING

**UNIVERSITY OF ECONOMICS AND FINANCE**

**PROJECT REPORT**

**Mobile Device Programming**

**TOPIC**

**Customer Management**

**Major: Information Technology**

**Minor: Software Engineering**

**Group: UEF\_M\_02E\_Group 5**

**Academic advisor: Lê Viết Linh**

**Ho Chi Minh City, 2024**

Members of Group 5:

|  |  |
| --- | --- |
| **Name** | **Student ID** |
| Mai Anh Khôi | 215051703 |
| Bùi Tấn Sắc | 215051759 |

**1. Introduction**

The **Customer Management** project is developed to create a modern customer management system that helps businesses store, analyze, and manage customer data in a smart and efficient way. Through this project, we aim to provide a tool that optimizes customer care processes, enhances user experience, and increases the competitive edge of businesses in the market.

**1.1. Purpose**

The **Customer Management** project aims to build a comprehensive customer management system that helps businesses effectively store, manage, and analyze customer data. The main objectives include optimizing customer care processes, improving service quality, supporting data analysis to better understand customer needs and behaviors, providing detailed reports to support business decision-making, and ensuring the security of customer information.

**1.2. Scope**

The scope of the **Customer Management** project includes building a comprehensive customer management system with key functions such as customer information management, tracking transaction and interaction history, customer segmentation and grouping, and providing reporting and data analysis. The system also supports integration with existing systems, ensures customer data security, and manages user access rights. The goal is to optimize customer care and data analysis processes, helping businesses improve operational efficiency while protecting personal data.

**1.3. Overview**

The Customer Management project is designed to meet the increasing demands of businesses in managing and optimizing customer relationships. This system will provide a comprehensive platform for storing, managing, and analyzing customer data, thereby supporting businesses in developing effective customer care strategies.

With the rapid development of technology and increased competition in the market, understanding customer needs and behaviors has become more important than ever. This project not only helps businesses systematically track and manage customer information but also provides data analysis tools that enable businesses to make accurate and timely decisions.

The system will include key features such as managing personal information of customers, tracking transaction history, segmenting customers based on behaviors and needs, as well as generating detailed reports to support the decision-making process. Additionally, ensuring data security and compliance with personal data protection regulations is also a crucial factor in this project.

In summary, the Customer Management project is not just a customer management tool but also a strategic solution that helps businesses enhance their competitive advantage and increase customer satisfaction.

**2. Design Considerations**

**2.1. Constraints**

**Technology Stack:**

* **Front-End:** XML, Java
* **Database:** SQLite (DB Browser for SQLite)

**2.2. System environment**

The system environment for the Customer Management project includes hardware, software, and services on which the application will run. This environment needs to be designed to ensure the stability, performance, and scalability of the system.

**2.3. Design Methodology**

The design methodology for the **Customer Management** project includes a systematic process to ensure the application meets user requirements. The main steps involve gathering requirements through interviews and surveys, designing the user interface using Figma to create wireframes and mockups, designing the system architecture and database, developing the application using Android Studio with Java, testing the components and the entire system, and deploying the application to the production environment. The project also adopts an Agile development methodology to ensure flexibility and adaptability to changes in requirements. The combination of systematic design steps and the Agile methodology helps ensure that the final product is an effective and reliable customer management tool.

**3. Architecture**

The architecture of the **Customer Management** project is designed for scalability, maintainability, and performance, utilizing the MVC (Model-View-Controller) pattern. The **Model** contains the data and business logic, representing customers and transactions with methods to manage data. The **View** is the user interface designed with **Figma** and implemented in **Android Studio**, facilitating user interactions, while the **Controller** handles these interactions, retrieves data from the model, and updates the view. The project employs **SQLite or Room** for database management, utilizes **RESTful APIs** for server communication via Retrofit, and implements **LiveData and ViewModel** for effective state management. Additionally, unit and UI testing ensure the application functions correctly. Overall, this architecture aims to create a reliable and effective customer management tool.

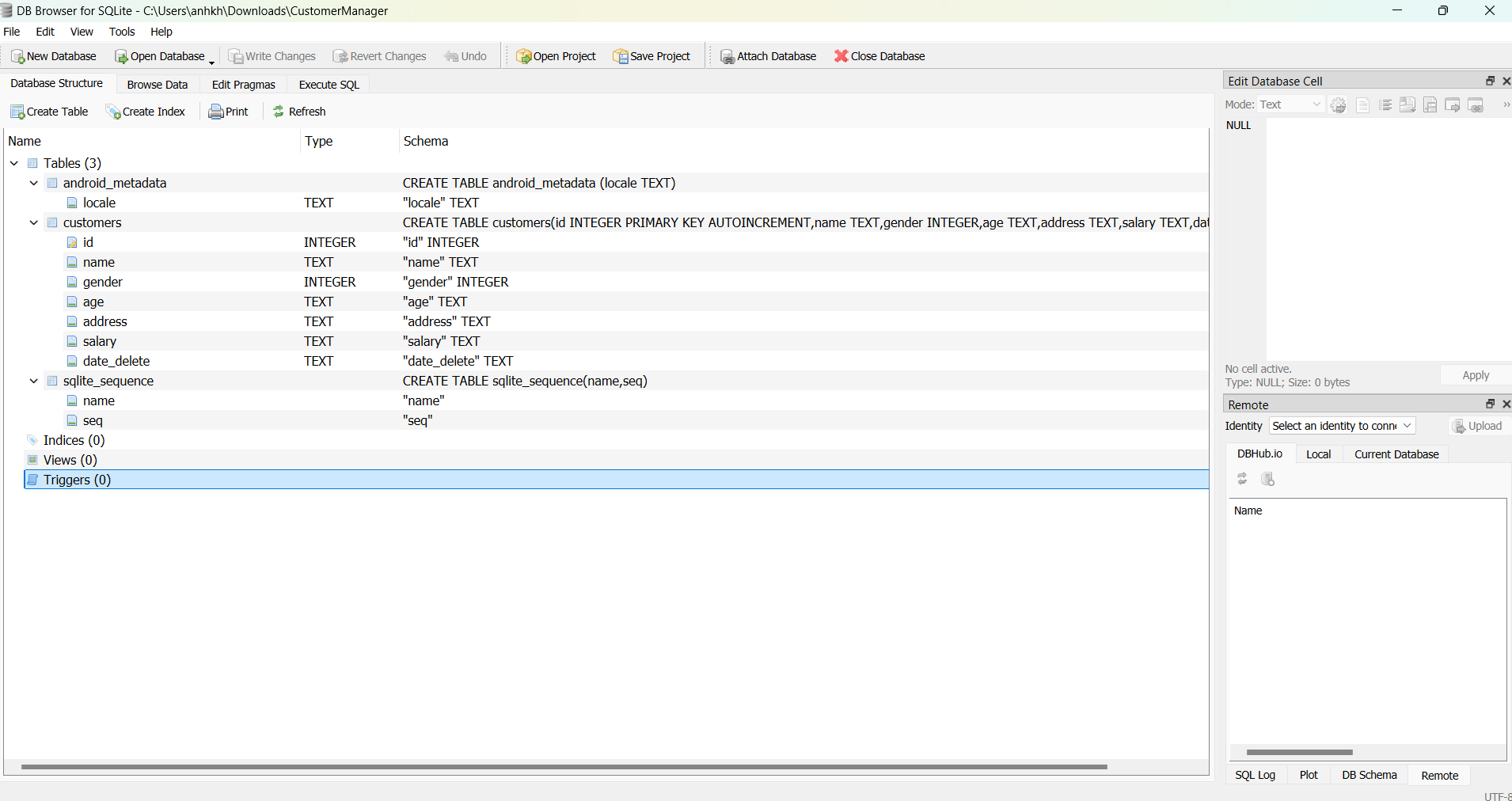
**4. Database: SQLite**

**4.1 What is SQLite?**

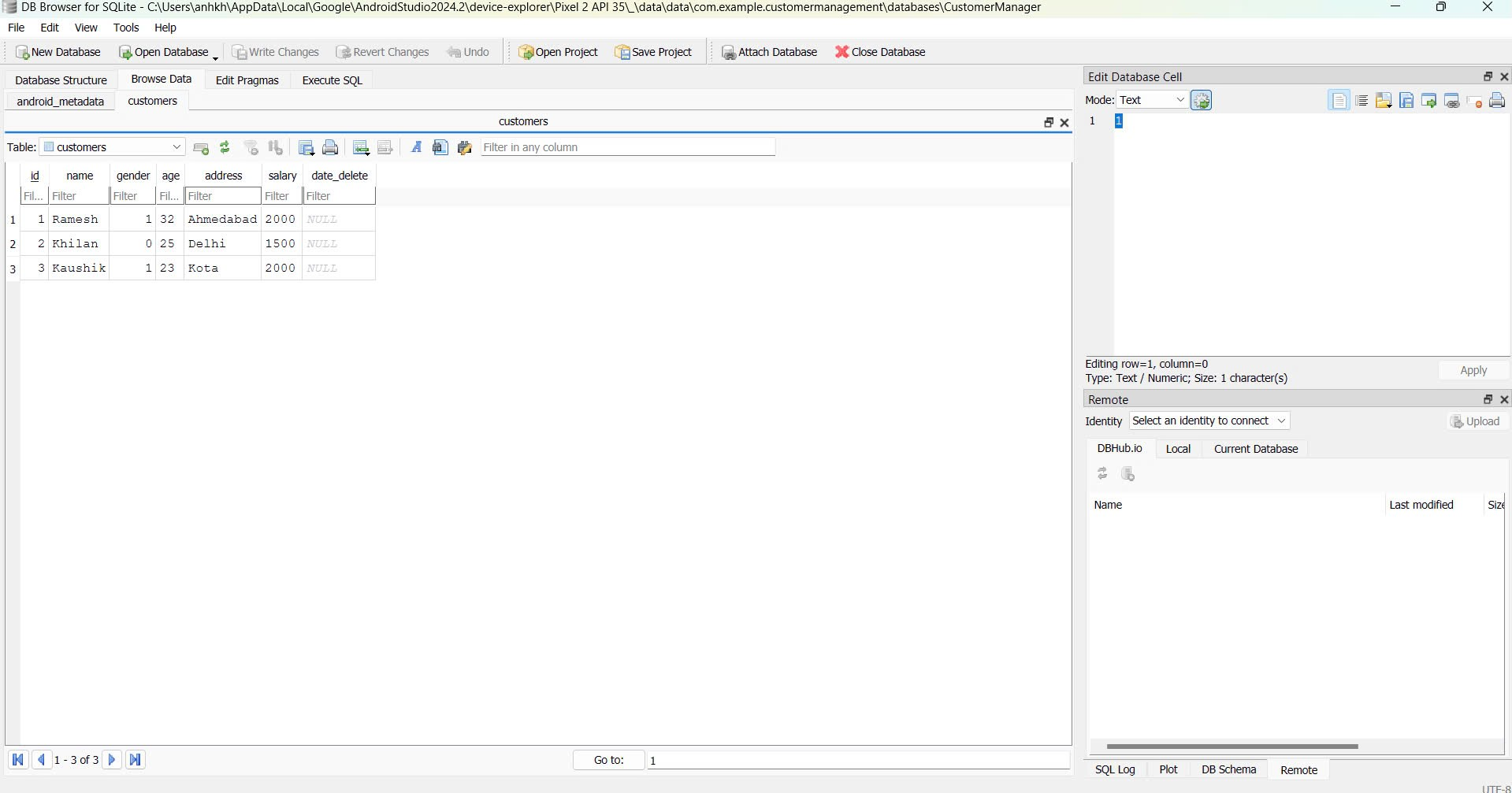
SQLite is a lightweight and popular relational database management system widely used in Android applications. It is suitable for storing local data on mobile devices.

**4.2 Advantage**

* Lightweight and built-in: SQLite is a compact database system that doesn't require a separate database server and is built into Android.
* No installation or management needed: SQLite doesn't require server configuration or additional services, making it easy to deploy and use.
* Fast retrieval speed: For small to medium-sized data, SQLite operates quickly and efficiently, making it suitable for mobile applications.
* Standard SQL support: SQLite supports standard SQL queries, making it easy for developers to work with data.

****

Picture 1. The interface of DB Browser for SQLite.

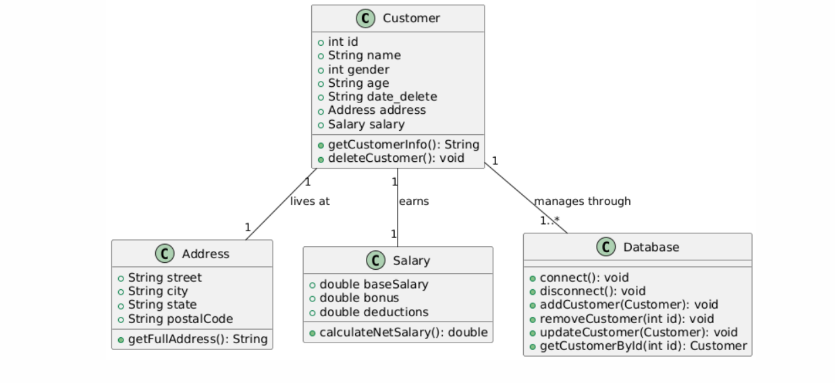
****

Picture 2. Browse Data

**5. IMPLEMENTATION RESULTS**

**5.1 Class Diagram**

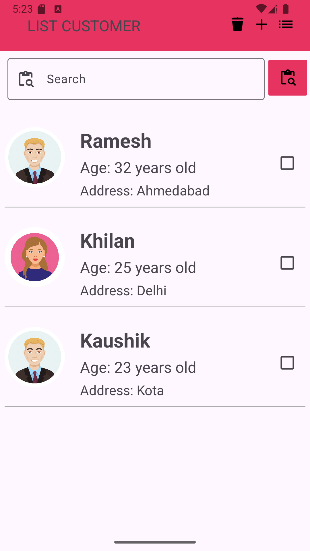
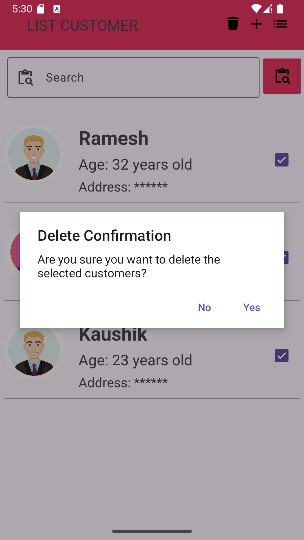
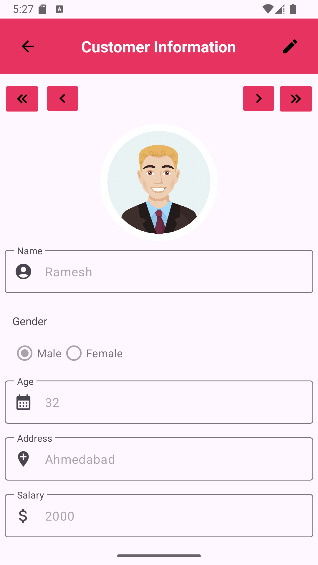
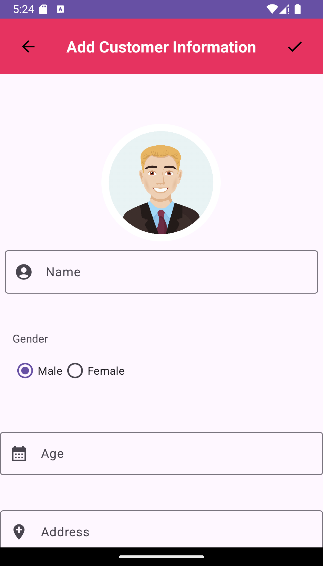
The class diagram describes a customer management system with the main classes being **Customer**, **Address**, **Salary**, and **Database**. The **Customer** class contains information about the customer such as ID, name, gender, age, address (represented by the **Address** class), and salary (represented by the **Salary** class). **Customer** has methods to retrieve and delete customer information. The **Address** class manages address details, while the **Salary** class calculates the net salary based on the base salary, bonuses, and deductions. The **Database** manages multiple customers and provides methods for connecting, adding, removing, and updating customer information.



Picture 3. Class diagram in the application

**6. Main screens and features of the application.**

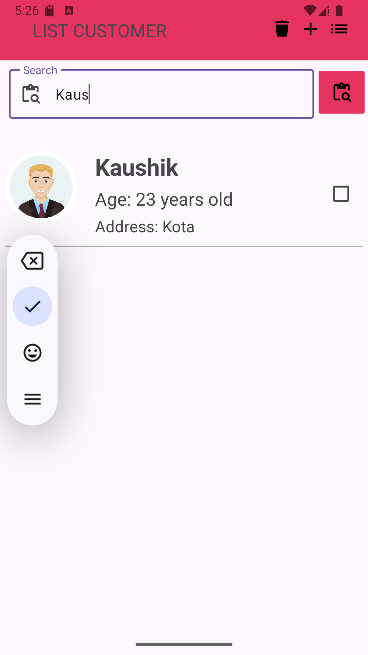
**6.1 Home Screen**

****

Picture 4. Home Screen

|  |  |
| --- | --- |
| **Screen** | Home |
| **Description** | The Home screen of the application provides an intuitive interface for users to manage customer information.   * **Display the customer list**: * The screen will display a list of customers with the following information:   + - **Name**: The name of the customer.     - **Age:** Customer age     - **Address:** Customer address * **Add Customer information:**    + Users can add new customers to the list.   + The interface will include data fields: Name, Gender, Age, Address, Salary and Amount salary * **Edit customer information:**   + Users can edit the information of an existing customer.   + The input fields will be pre-filled with the customer's current information and can be changed by the user as needed. * **Delete customer information:**   + The user can delete one or more selected customers using the checkbox.   + When deleted, customer-related information will be permanently deleted from the system. |

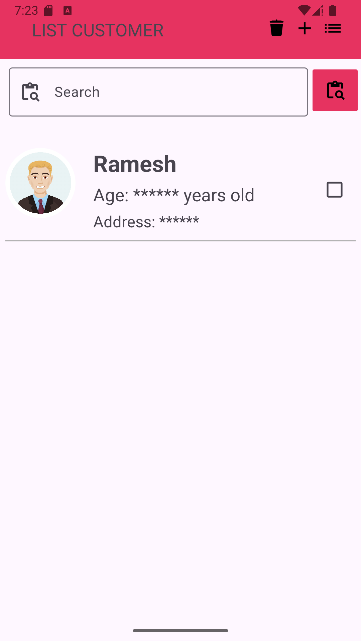
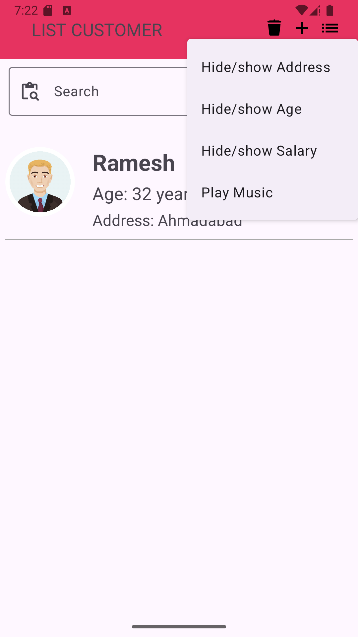
**6.2 Search Screen**

****

Picture 5. Search Screen

|  |  |
| --- | --- |
| **Screen** | Search |
| **Description** | The application's Search screen allows users to search for customer information based on customer name   * **Search by customer name:** * Users can enter the name of the person assigned the job into the search box. * The resulting list will include all customer information along with the corresponding name. * **Display search results:** * Search results will be displayed as a list with information similar to the Home screen, including: Name, age, address. * If there are no satisfactory results, the system will display a message that no suitable customers were found. |
| **Validation** | **Search for customer names:**   * Does not distinguish between upper and lower case letters. For example, a search with the keyword "Task A" will return results named "task a", "Task A", or "TASK A". |

**6.2 Setting Screen**

****

Picture 6. Setting Screen

|  |  |
| --- | --- |
| **Screen** | Setting screen |
| **Description** | The app's Settings screen gives users the ability to customize some information and display according to their needs.   * **Hide/Show toggle**   + When switching to the (Show) state, information in the list will be displayed such as address, age, salary   + When switching to the (Hide) state, information will be hidden |
| **Validation** |  The state of the toggle button (Hide/Show) will be saved in the user's settings, helping to maintain their selection when reopening the application.   When the user changes the toggle button state, the app automatically updates the interface to reflect the new selection. |

**7. Testing App**

**7.1 Test Case for GUI Home Screen:**

**Test Case 1: Add customer information**

* Description: Add new customer with valid information.
* Input:

Name: “Ramesh”

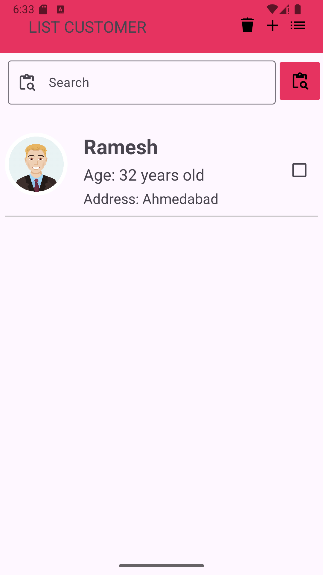
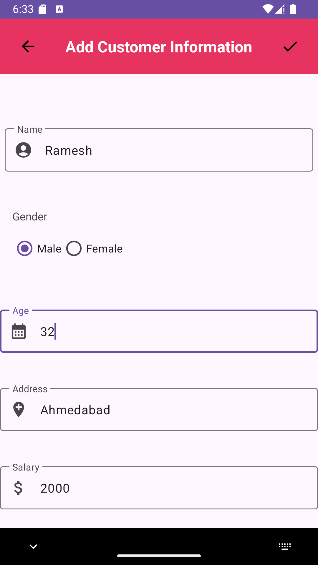
Gender: Male

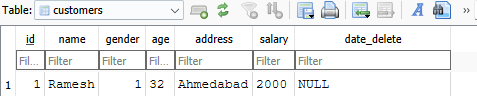
Age: “32”

Address: “Ahmedabad”

Salary: “2000”

* Expected result: New customer successfully added, displayed in the list.





**Test Case 2: Edit customer information**

* **Description: Edit information of an existing task.**
* Input:

Name: “Ramesh”

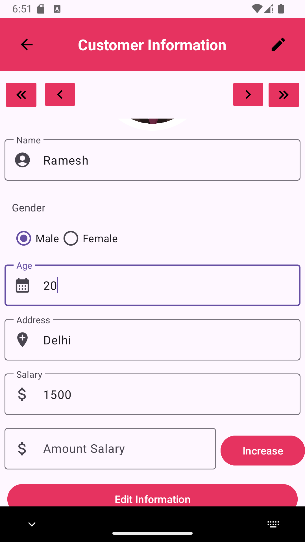
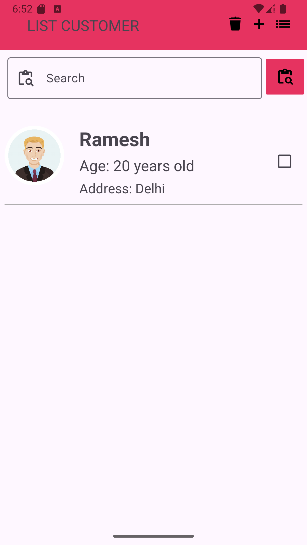
Gender: Male

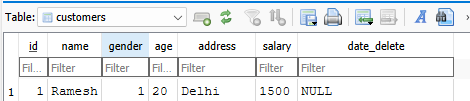
Age: “20”

Address: “Delhi”

Salary: “1500”

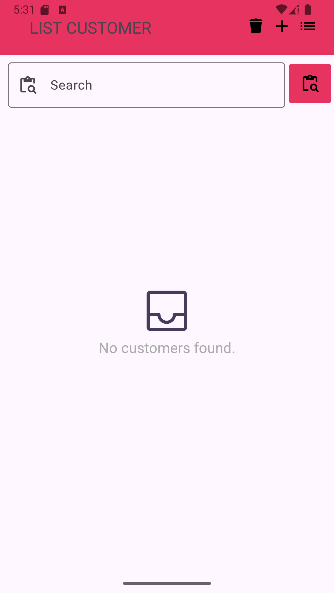
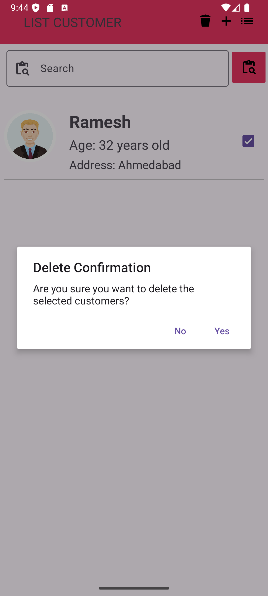
* **Expected result: Mission information has been updated successfully.**

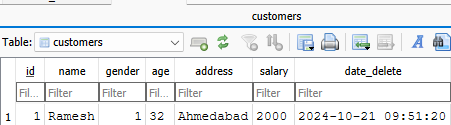
****



**Test Case 3: Delete customer information**

* Description: Delete an existing task.
* Input: Check the box the customer wants to delete and then click the Trash icon on the toolbar to delete.
* **Expected result:** The customer has been successfully deleted and is no longer visible in the list.

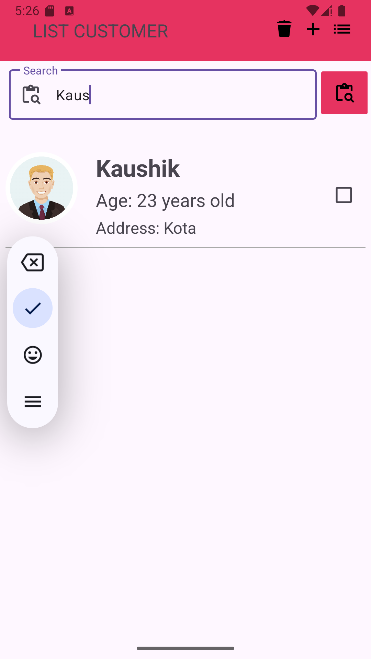




**7.2 Test Case for GUI Search Screen:**

**Test Case 4: Search by customer name**

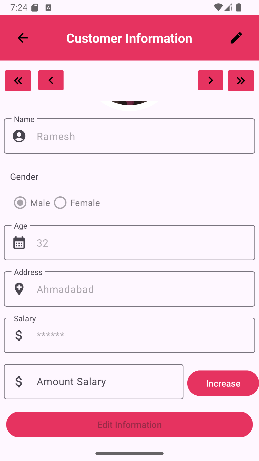
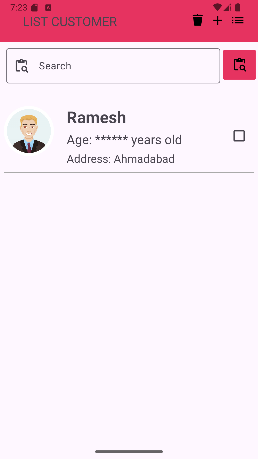
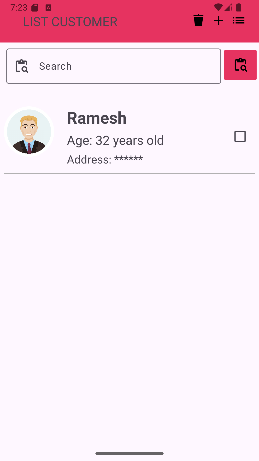
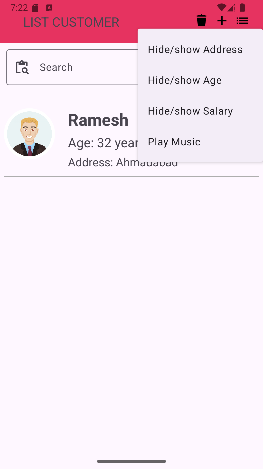
* **Description: Search name by valid customer name.**
* **Search: “Kausik”**
* **Expected result: Display a list of customers with the name "Kaushik"**

****

**7.3 Test Case for GUI Setting Screen:**

**Test Case 5: Search by customer name**

* **Description: Show/Hide address, age and salary**
* **Input: Switch from Visible to Hidden.**
* **Expected results: address, age, salary information will be displayed/hidden depending on the user's choice.**

****

**8. Conclusion and recommendations**

**8.1 Conclusion**

The Customer Management project has developed an effective customer management system that meets the needs and challenges of managing customer relationships in the digital age. By applying the MVC architecture and leveraging powerful development technologies, the project optimizes customer care processes and systematically stores and analyzes data. The integration of SQLite/Room for database management and RESTful APIs for server connectivity has established a solid foundation for the application. The Agile development approach enables the team to quickly adapt to changing requirements, ensuring product quality. With notable features such as customer information management and detailed reporting, the system enhances customer experience and improves business efficiency, helping companies maintain strong relationships with customers and enhance their competitive edge in the market.

**8.2 Meaning**

The Customer Management project holds significant meaning for both businesses and customers. It optimizes the processes of managing and caring for customers, helping businesses save time and enhance operational efficiency. By storing and analyzing customer data, companies can gain deeper insights into customer needs and behaviors, allowing them to adjust marketing strategies and provide better services. The system also strengthens the relationship between businesses and customers by tracking transaction histories, creating personalized experiences, and building customer loyalty. Finally, the application of modern technology in customer management not only enhances competitive ability but also improves brand image. In summary, the project is not just a management tool but a strategic solution that helps businesses grow sustainably and increase value for customers.

**8.3 Scope of application**

This application can be applied in many different fields, from project management, office work to software development teams. It is suitable for large as well as small organizations, helping to improve workflow and increase productivity. Besides, the application can also be expanded to integrate other features such as performance analysis, progress reporting and resource management, meeting the increasingly diverse needs of users.

REFERENCES

1. **Google Developers (2020)**, Android Installation. FlutterFire Document. Date of access 10/10/2024 <https://firebase.flutter.dev/docs/manual-installation/android>
2. **Oracle (2021)**, The Java™ Tutorials. Getting Started. Date of access 10/10/2024 <https://docs.oracle.com/javase/tutorial/java/index.html>
3. **DB Browser for SQLite. Date of access 10/10/2024 https://sqlitebrowser.org/dl/**
4. **Android Developers (2021)**, App Architecture. Date of access 10/10/2024 <https://developer.android.com/jetpack/guide>