Vocational Training Authority of Sri Lanka

District Center – Inamaluwa

Report on Contact Mobile App Development

Android App Development

Table of Content

Introduction

The Contact Mobile App simplifies contact management for Android users. It is a lightweight, privacy-focused solution with offline functionality and intuitive features such as adding, editing, and deleting contacts with image association.

Objective

The app's primary goals are:

1. Provide CRUD functionality for contact management.
2. Enable offline operations with local data storage using SQLite.
3. Ensure user privacy by avoiding cloud storage.
4. Optimize performance for devices with limited resources.

Key Features

1. Contact Management

- **Add Contact:** Users can input a contact's name, number, and upload an image.
- **Edit Contact:** Modify details or update the image. Delete functionality is also integrated into this screen.
- **View Contacts:** Scrollable contact list displaying names and numbers.

2. Offline Functionality

- Entirely local data storage ensures that the app works without internet connectivity.

3. Image Handling

- Users can attach images from their device's gallery.
- Images are stored securely in the app's private directory.

Development Process

1. Planning

Defined objectives and challenges, structured progress using the Waterfall Model.

2. Design

- Developed a clean, user-friendly UI using Kotlin.
- Designed an efficient SQLite database schema.

3. Implementation

- Created intuitive screens for viewing, adding, and editing contacts.
- Integrated SQLite for managing contact data and image associations.

4. Testing

Conducted unit and usability tests to validate the app's features and ensure offline functionality.

System Specifications

Hardware Requirements

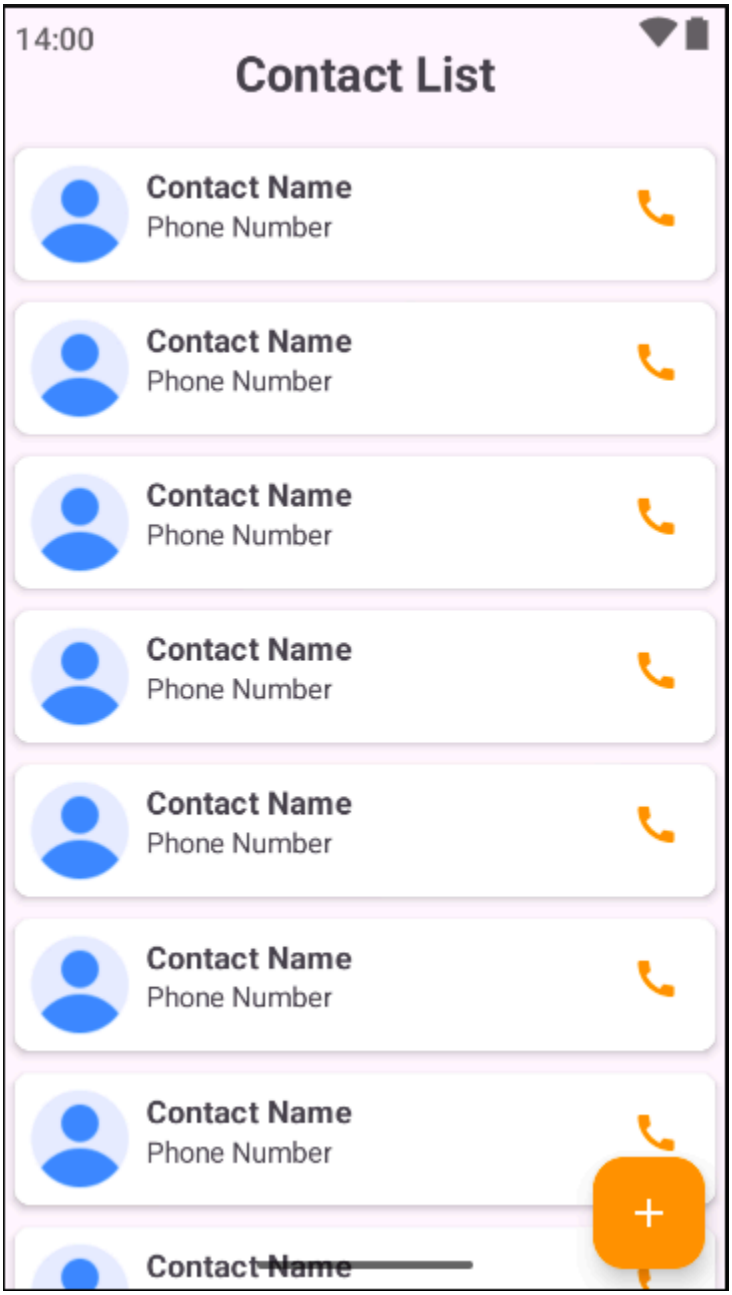
- **Processor:** Minimum ARM Cortex-A7; recommended ARM Cortex-A53 or higher.
- **Memory:** Minimum 1 GB RAM; recommended 2 GB or more.
- **Storage:** Minimum 10 MB free space.

Software Requirements

- **Operating System:** Android 5.0 (API Level 21) or higher.
- **Programming Language:** Kotlin.
- **Database:** SQLite for local storage.

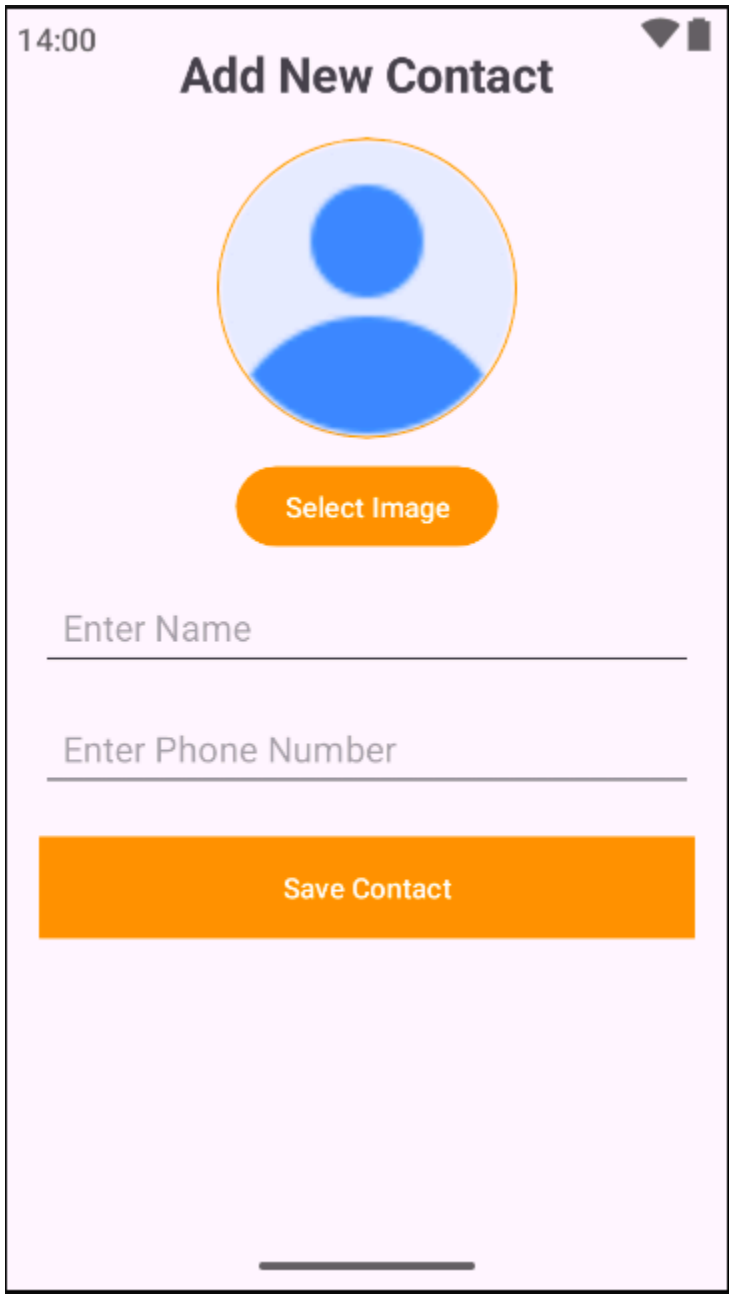
Output Screens

1. Contact List Screen



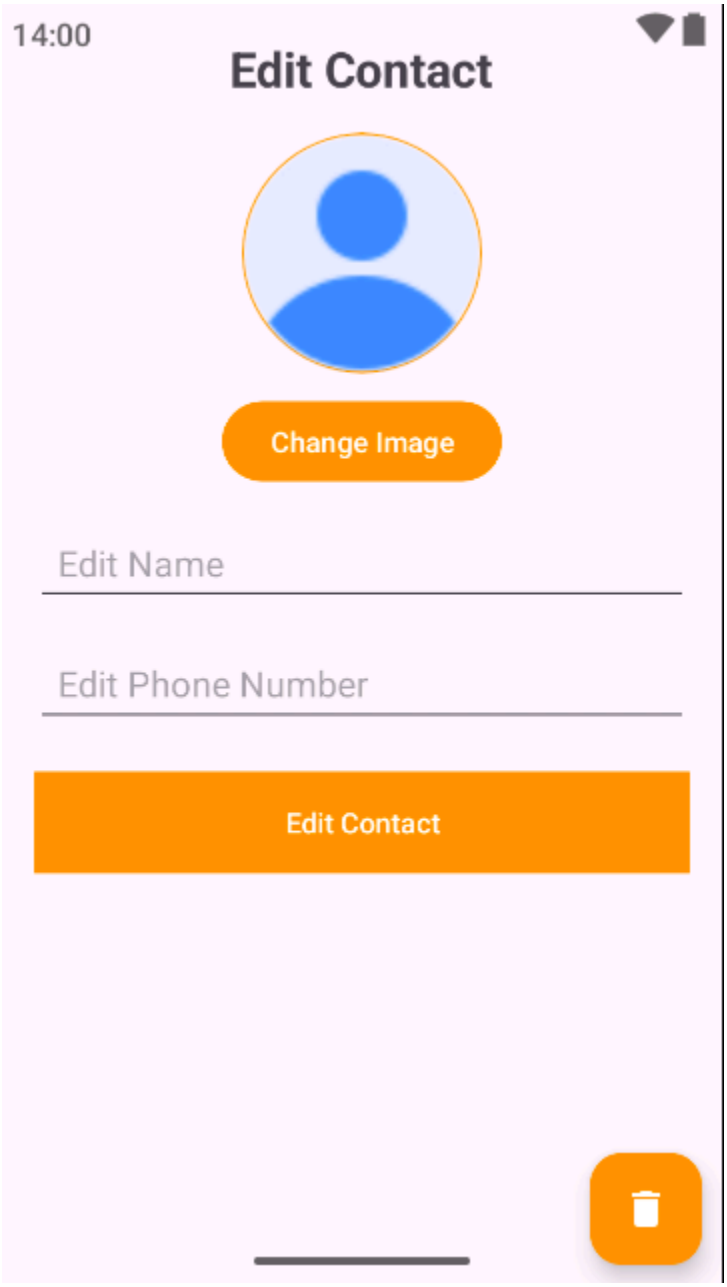
Description: Displays all saved contacts with image ,names and numbers.

2. Add Contact Screen



Description: Allows users to input a contact's name, mobile number, and upload an image.

3. Edit Contact Screen



Description: Enables editing contact details, updating the image, and deleting the contact.

Challenges and Solutions

1. Ensuring smooth performance on older devices.

- **Solution:** Optimized database queries and compressed images to reduce memory usage.

2. Maintaining data privacy.

- **Solution:** Stored all data locally with optional database encryption.

3. Simplifying the user interface.

- **Solution:** Adopted Material Design principles for a clean and intuitive layout.

Future Enhancements

1. **Additional Features:** Contact grouping, email, and address fields.
2. **Backup and Restore:** Integration with cloud services for optional backups.
3. **Synchronization:** Enable multi-device synchronization.
4. **Localization:** Add support for multiple languages.

Conclusion

The Contact Mobile App addresses common challenges in contact management by offering a lightweight, privacy-focused, and offline-capable solution. Future updates will expand its features, ensuring it continues to meet evolving user needs.