ADVANCED ANDROID APP DEVELOPMENT

(Project Semester January-May 2024)

Chat Application

Submitted by

Shaik Mahammad Rafi

Registration No 12013757

Programme and Section KO203

Course Code CSE227

Under the Guidance of

Dr. Subhita (20260)

Discipline of CSE/IT

Lovely School of Computer Science of Engineering

Lovely Professional University, Phagwara

DECLARATION

I, SHAIK MAHAMMAD RAFI, student of Computer Science & Engineering under CSE/IT Discipline at, Lovely Professional University, Punjab, hereby declare that all the information furnished in this project report is based on my own intensive work and is genuine.

Date: 17-04-2024 Signature: **Shaik Mahammad Rafi**

Registration.no.**12013757** Name of the student: **Shaik Mahammad Rafi**

CERTIFICATE

This is to certify that SHAIK MAHAMMAD RAFI bearing Registration no. 12013757 has completed CSE227 project titled, Recipe application. To the best of my knowledge, the present work is the result of his/her original development, effort and study.

Signature and Name of the Supervisor

Designation of the Supervisor

School of Computer Science and Engineering

Lovely Professional University

Phagwara, Punjab.

Date: - 17-04-2024

ACKNOWLEDGEMENT

Primarily I'd thank God for being able to complete my project with success. Then I'dlike to thank my mentor **Dr. Subhita (20260)**, whose valuable guidance has been the ones that helped me patch this project and make it full proof success in contribution towards the completion of this project.

Finally, I'd rather thanks to **Lovely Professional University,** who gave me this golden opportunity to learn many new things, to learn another aspect of life.

- Shaik Mahammad Rafi

CONTENTS:

INTRODUCTION:

Introducing an end-to-end encrypted chat application built in Kotlin, designed to prioritize user privacy and security without compromising on usability. Leveraging modern cryptographic techniques, this application ensures that only the intended recipient can decrypt and read messages, providing a secure channel for communication.

Using Kotlin's robust features and libraries, the application implements strong encryption protocols such as AES (Advanced Encryption Standard) and RSA (Rivest-Shamir-Adleman) to encrypt messages before transmission. Each user is assigned a unique key pair, ensuring confidentiality and integrity of messages exchanged.

Furthermore, the application employs key exchange mechanisms like Diffie-Hellman to securely negotiate encryption keys between users, minimizing the risk of eavesdropping or interception by malicious actors.

User experience is paramount, with a sleek and intuitive interface facilitating seamless communication while transparently handling encryption and decryption in the background. End-to-end encryption is the cornerstone of this application, empowering users to communicate freely and confidently, knowing their conversations are shielded from unauthorized access or surveillance.

OBJECTIVES:

The main objective of our recipe application is to empower users to explore, discover, and create culinary delights effortlessly. We aim to provide a user-friendly platform where individuals can access a diverse range of recipes, cooking videos, and culinary inspiration, enhancing their cooking experience and fostering a sense of culinary adventure and creativity.

1) Enhanced User Experience: Ensure that users have a seamless and enjoyable experience browsing, searching, and accessing recipes within the application.User can read the trending.

2) Increased Engagement: Encourage users to engage with the application by regularly exploring new recipes, watching cooking videos, and interacting with the content.

3) Content Variety and Quality: Curate a diverse range of recipes across different categories such as breakfast, lunch, dinner, and vegetarian options to cater to various dietary preferences and occasions.

4) Personalization: Allow users to personalize their experience by adding recipes to their favorites list, enabling them to easily access their preferred recipes for future reference.

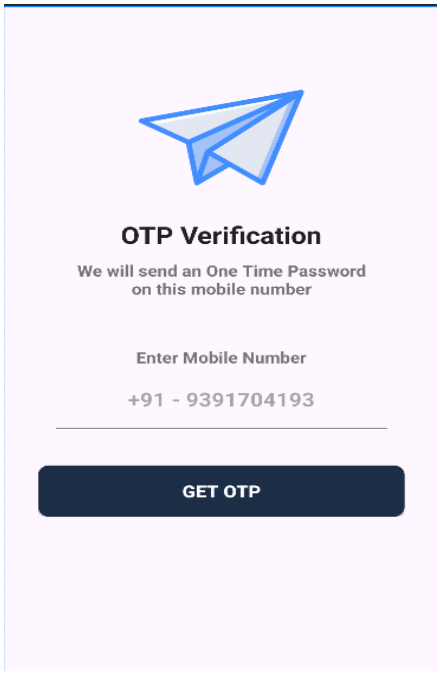
5) User Retention and Loyalty: Implement features such as the ability to add recipes as favorites to encourage users to return to the application regularly and build long-term loyalty.

Topics Covered in this project:

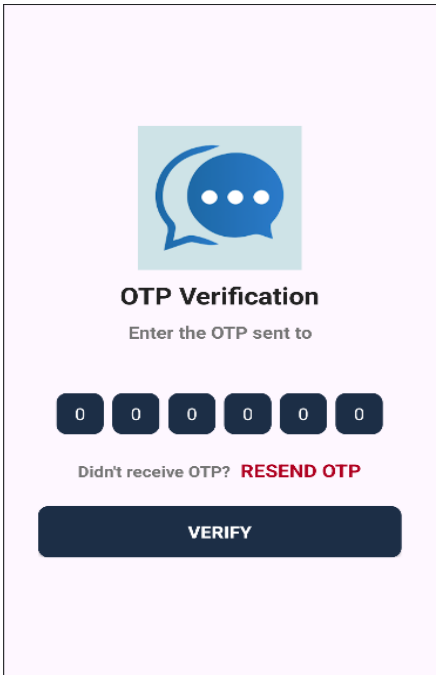
- Scroll Views (Recycler View, Nested Scroll View)
- Intents (Both Implicit and Explicit)
- Progress Bar
- Fragments
- ViewPager2
- Material3 Bottom Navigation Drawer
- Card View
- Speech to Text converter
- Floating Action Button
- Animations (fade in, & fade out)
- Firebase Authentication (Mobile)
- Firebase Fire Store
- Firebase cloud Storage
- Firebase Cloud Messaging
- Notification Manager
- Custom Alert Dialog
- Room Database
- Shared Preferences
- Permissions (Internet, Media, Contacts)

Screen Shots:

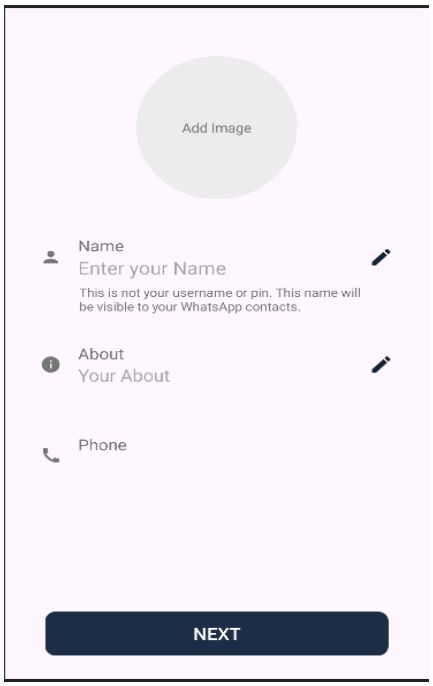
Mobile Authentication

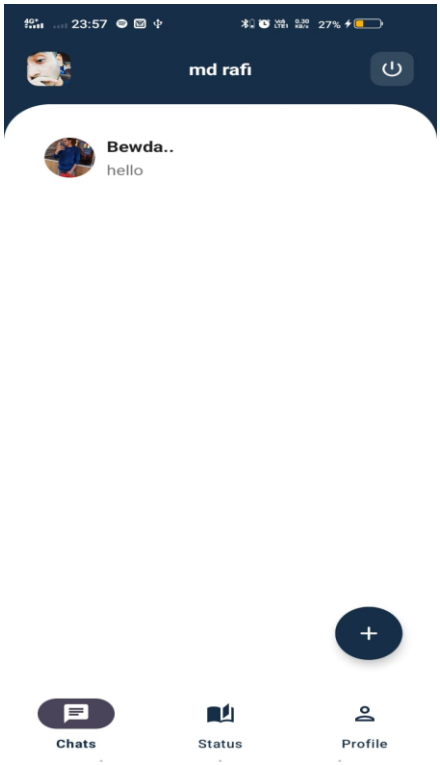


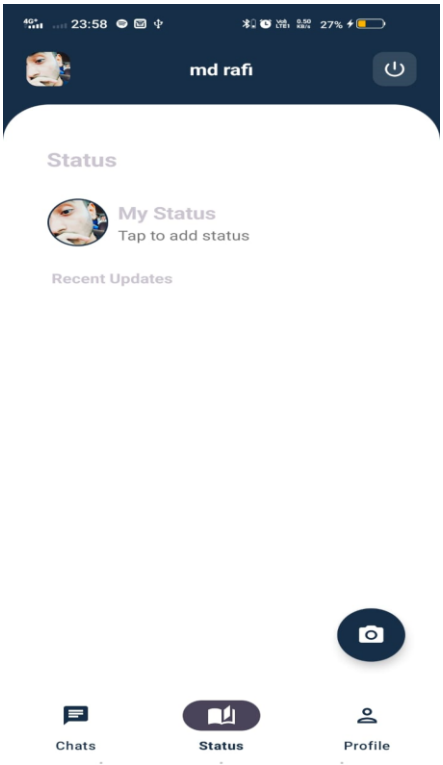
Verify OTP

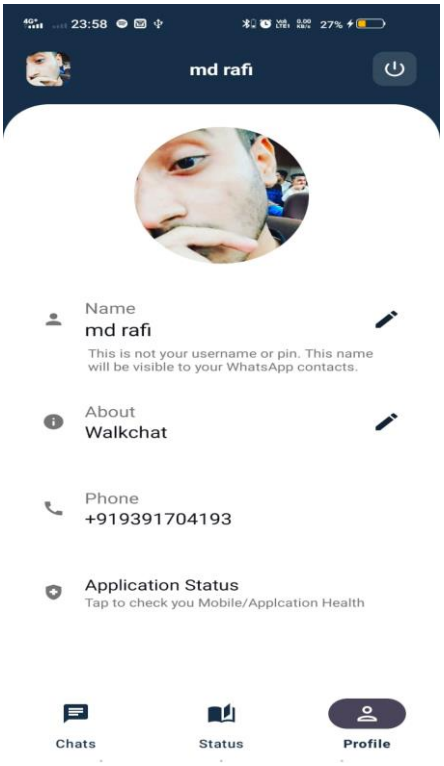


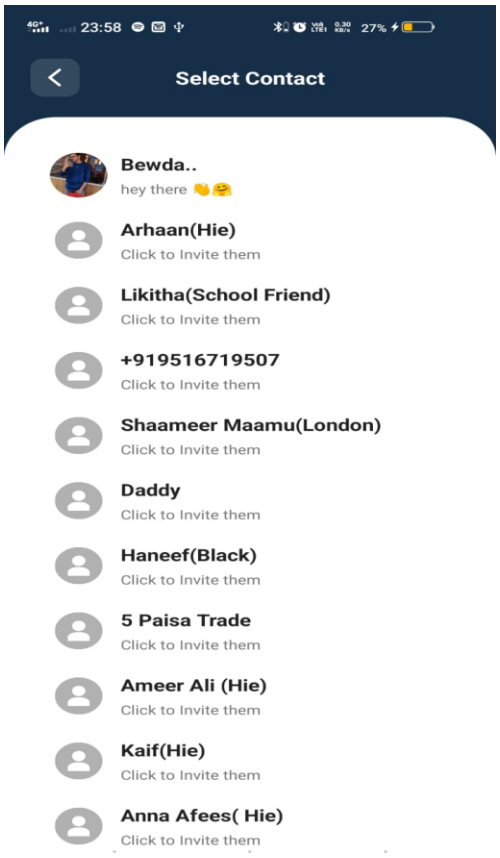
Profile Page

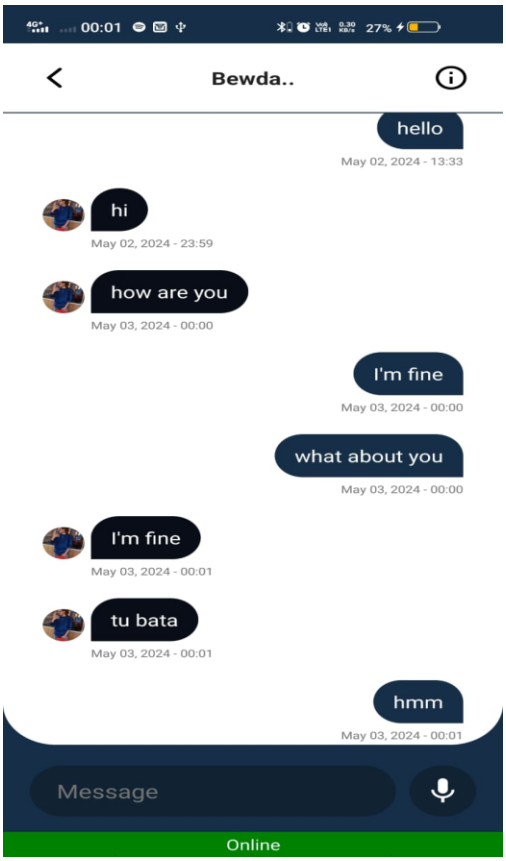


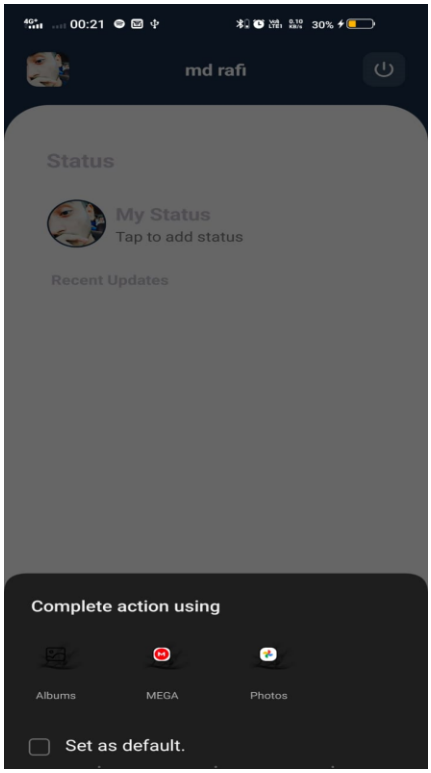


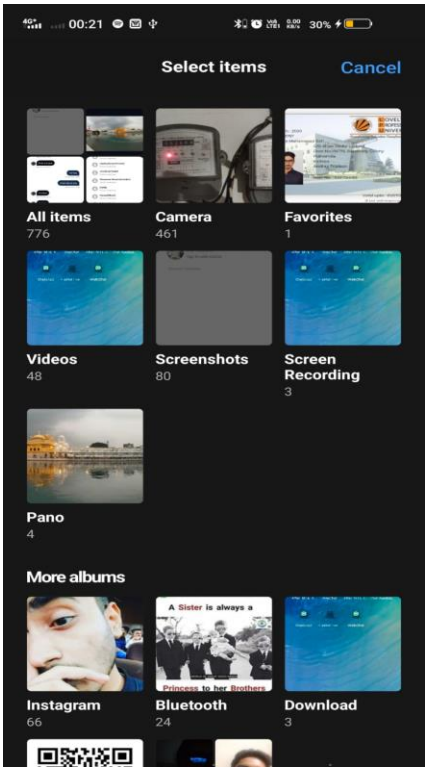


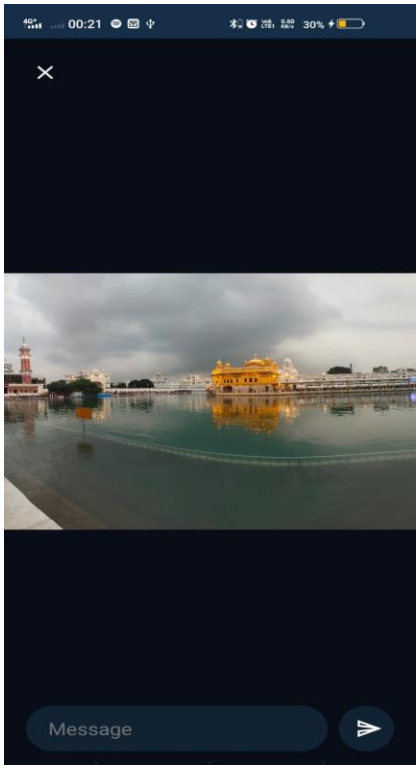



23:58
27%
Select Contact
Bewda..
hey there 👋🤗
Arhaan(Hie)
Click to Invite them
Likitha(School Friend)
Click to Invite them
+919516719507
Click to Invite them
Shaameer Maamu(London)
Click to Invite them
Daddy
Click to Invite them
Haneef(Black)
Click to Invite them
5 Paisa Trade
Click to Invite them
Ameer Ali (Hie)
Click to Invite them
Kaif(Hie)
Click to Invite them
Anna Afees(Hie)
Click to Invite them


00:01
27%
Bewda..
hello
May 02, 2024 - 13:33
hi
May 02, 2024 - 23:59
how are you
May 03, 2024 - 00:00
I'm fine
May 03, 2024 - 00:00
what about you
May 03, 2024 - 00:00
I'm fine
May 03, 2024 - 00:01
tu bata
May 03, 2024 - 00:01
hmm
May 03, 2024 - 00:01
Message
Online


00:21
30%
md rafi
Status
My Status
Tap to add status
Recent Updates
Complete action using
Albums
MEGA
Photos
Set as default.


00:21
30%
Select items
Cancel
All items
776
Camera
461
Favorites
1
Videos
48
Screenshots
80
Screen Recording
3
Pano
4
More albums
Instagram
66
Bluetooth
24
Download
3


00:21
30%
Message

Functionalities:

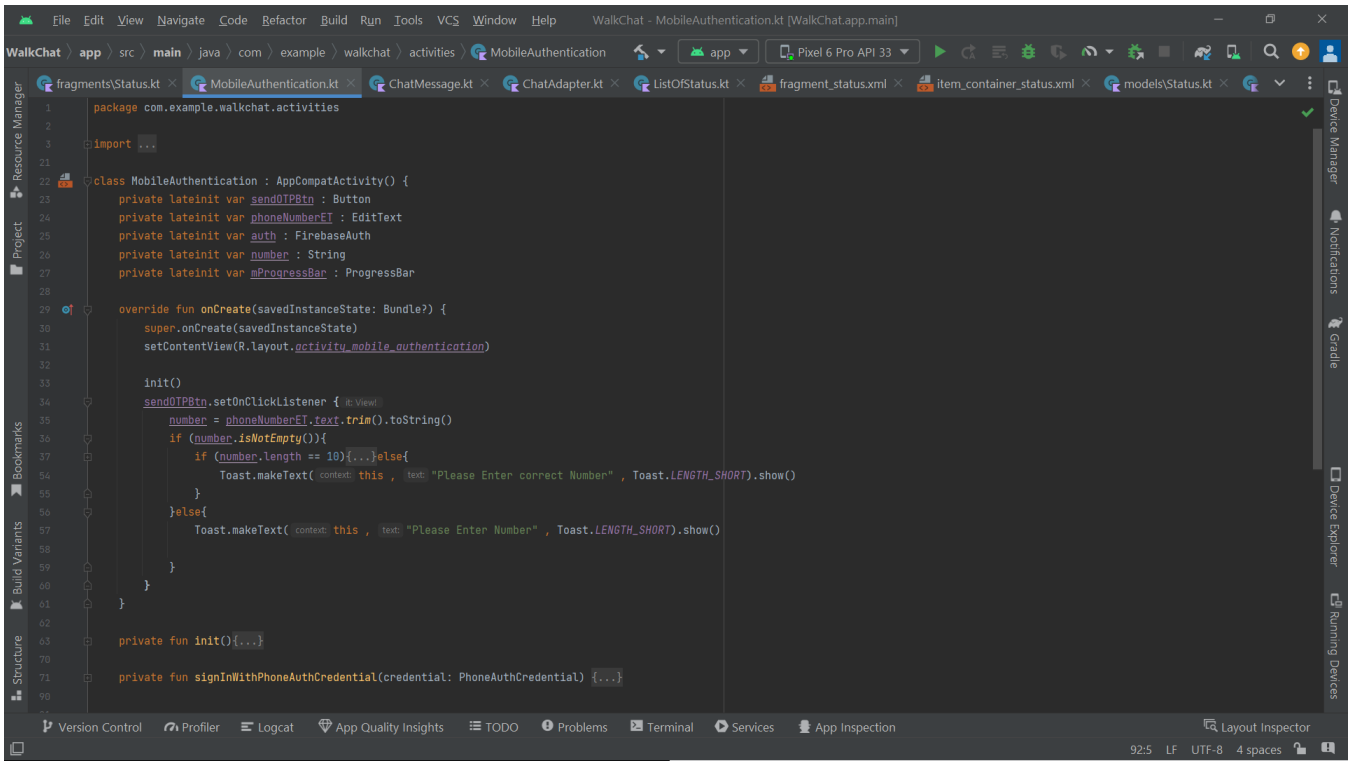
Activity: 1 – Mobile Authentication

Key components include:

- Utilization of Firebase AUTH for user authentication.
- Integration of Phone Auth Options and Phone Auth Provider to send and verify OTPs.
- Handling of various states in the authentication process through callbacks, such as successful verification, failed verification, and OTP sent.

The activity features user feedback via toasts and UI updates using progress bars. The code ensures a smooth and secure user authentication process, enhancing the overall user experience.

.

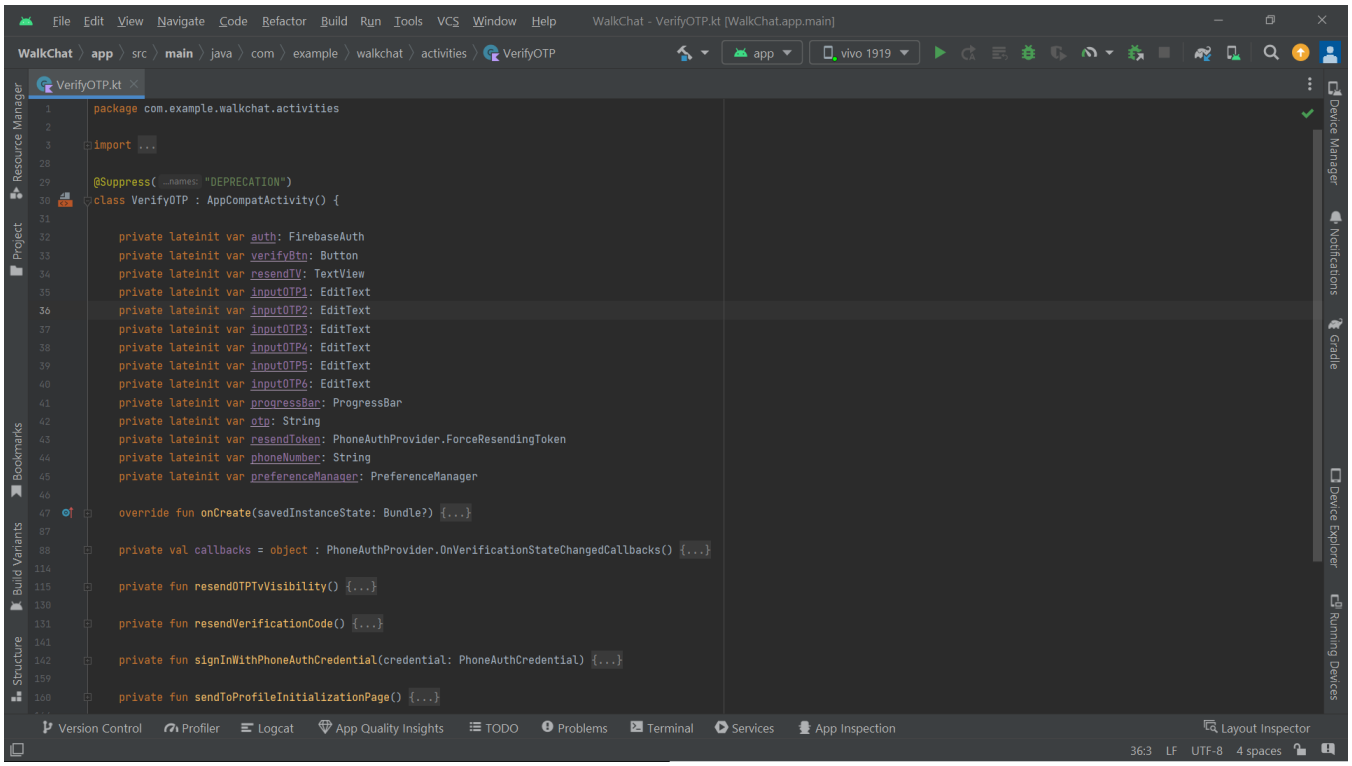


Activity: II – OTP Verification

Key features include:

- Integration of Firebase's authentication services, such as PhoneAuthProvider and FirebaseAuth, for OTP verification.
- Dynamic UI updates to enable/disable the resend OTP button after a cooldown period.
- Error handling for invalid OTP inputs and too many verification attempts.
- Seamless transition to the next activity upon successful verification, sending the user to the profile initialization page.

Additionally, the activity enhances user experience by organizing OTP input fields and providing smooth navigation between them using a custom EditTextWatcher class. Overall, the code demonstrates effective utilization of Firebase Phone Authentication for secure and efficient phone number verification in an Android application.



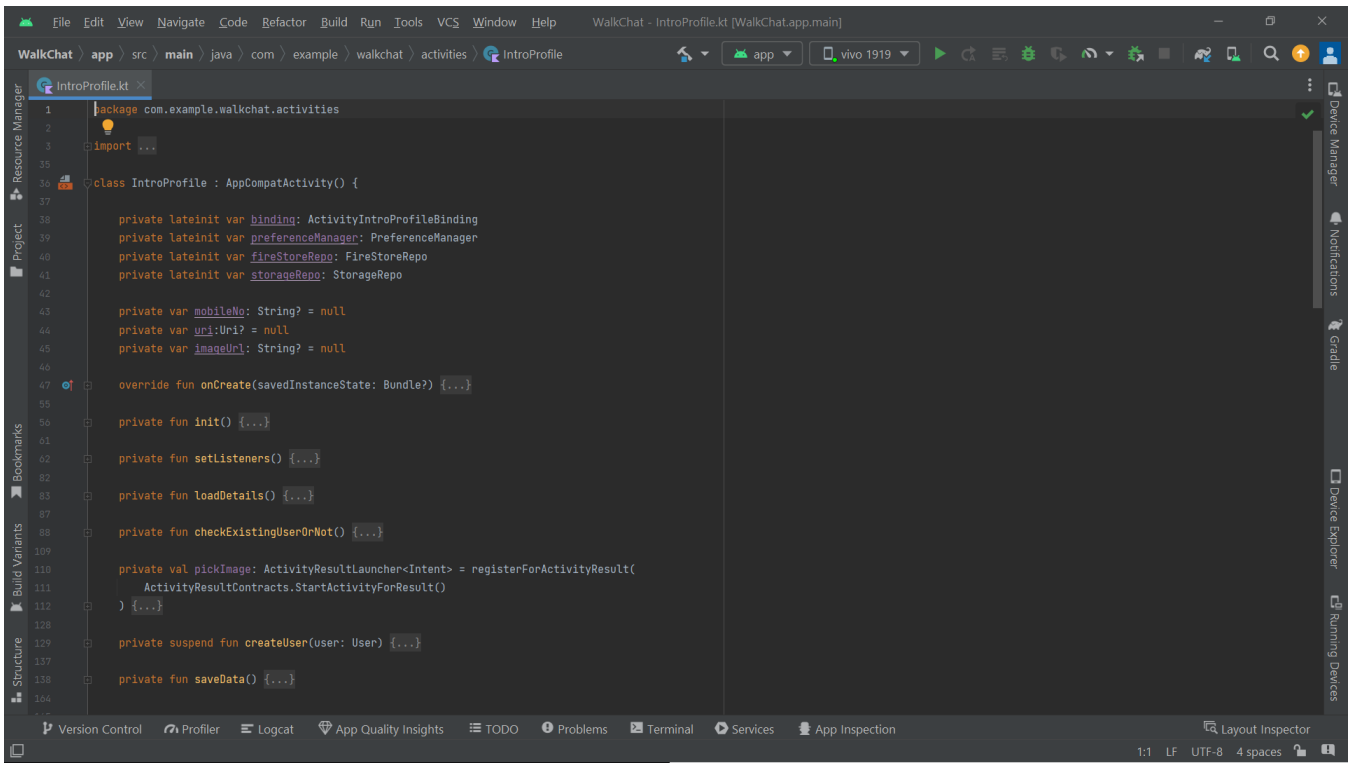
Activity: III – Profile Page

Key features include:

- Integration of Firebase services for user authentication and Firestore database.
- Use of coroutines for asynchronous tasks such as uploading images and updating user information.
- Implementation of activity result launcher for picking images from the device gallery.
- Custom dialog implementation for updating user information.
- Data validation to ensure that essential profile information such as name is provided before proceeding.

Management of UI state to show loading progress during data processing.

Overall, the activity provides a seamless user experience for initializing their profile within the application, ensuring essential user information is captured accurately and efficiently.

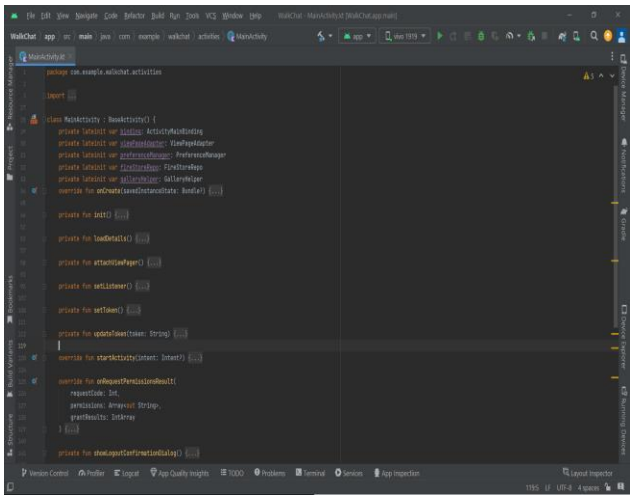


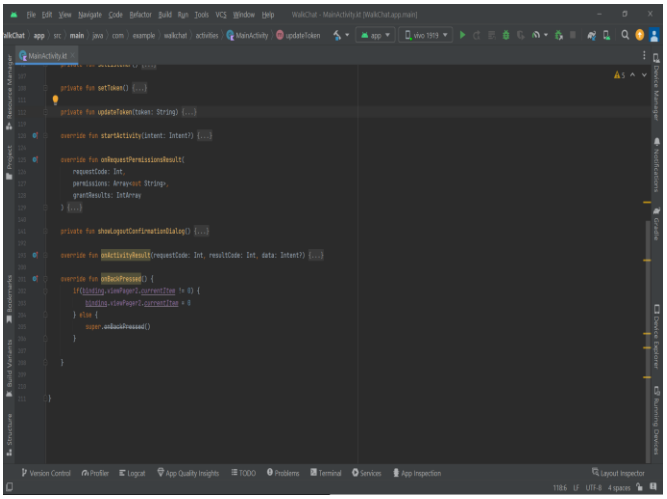
Activity: IV – Main Activity

Key features include:

- Integration of ViewPager2 to facilitate horizontal swiping between different sections.
- Bottom navigation setup for easy navigation between chat, status, and profile sections.
- Implementation of Firebase Messaging to update the FCM token and manage push notifications.
- Utilization of coroutines for asynchronous tasks such as updating the FCM token on Firestore.
- Custom logout confirmation dialog to ensure user confirmation before logging out.
- Integration of GalleryHelper to handle opening the device gallery for selecting images.
- Management of activity transitions to provide a smoother user experience.

Overall, the MainActivity enhances user engagement by providing a seamless interface for accessing various features within the application.



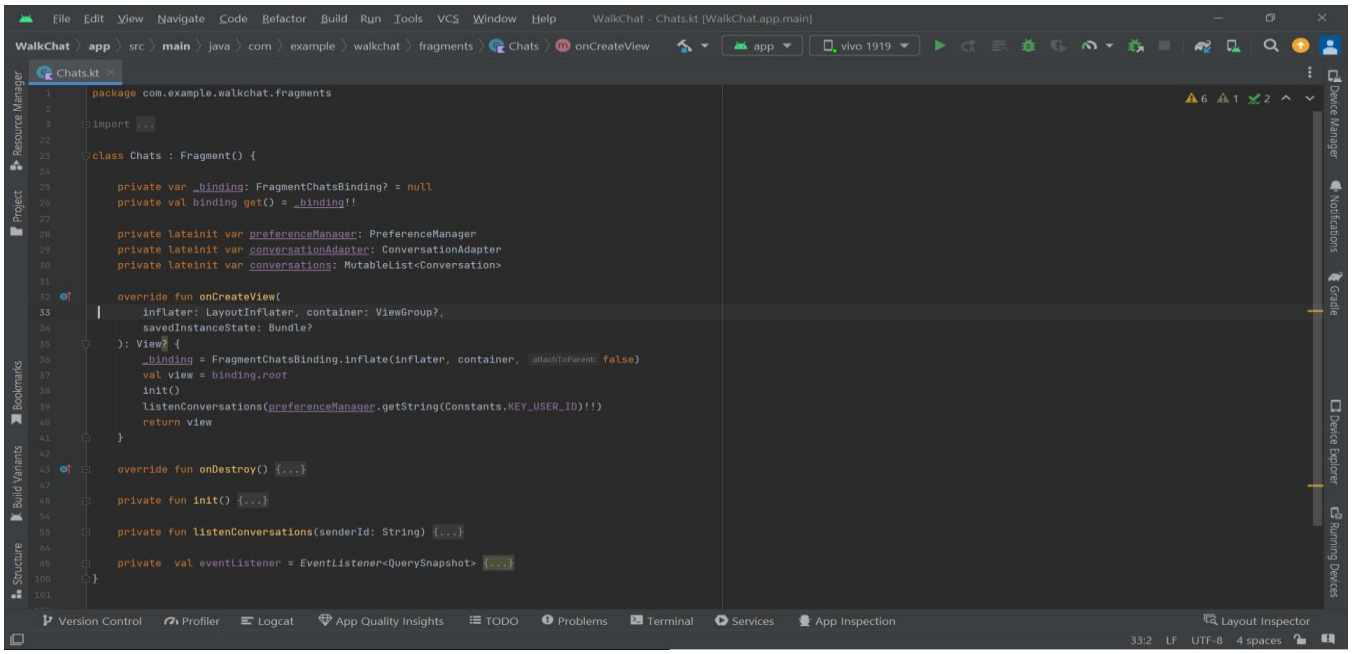


Fragment: I – Conversations

Key features include:

- Initialization of RecyclerView and ConversationAdapter for displaying conversations.
- Integration of Firestore to listen for real-time updates on conversations.
- Sorting conversations based on the timestamp of the most recent message.
- Handling document changes such as addition and modification of conversation details.
- Displaying progress bar during data loading and updating the UI accordingly.

Overall, the Chats fragment provides users with an up-to-date view of their conversations, ensuring a seamless communication experience within the application.

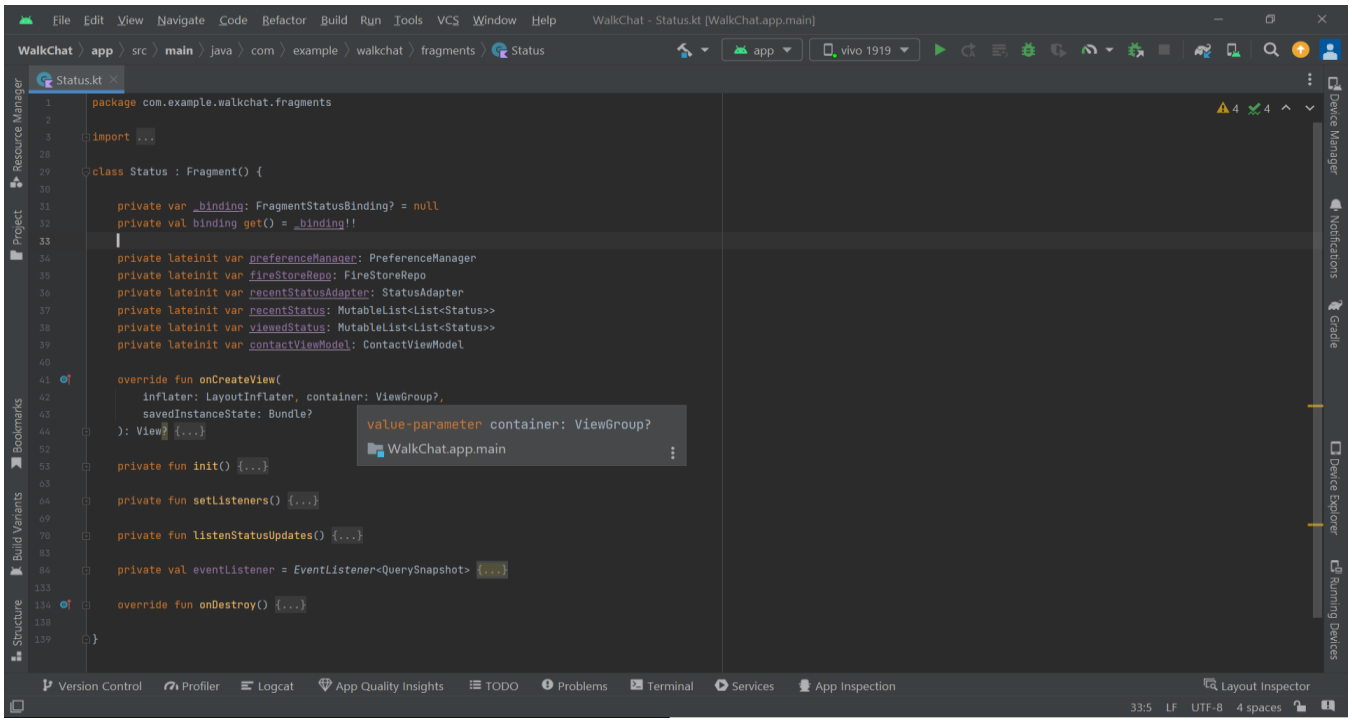


Fragment: II – Status

Key features include:

- Initialization of RecyclerView and StatusAdapter for displaying recent status updates.
- Integration of Firestore to listen for real-time updates on status changes.
- Sorting status updates based on the timestamp of the most recent update.
- Grouping status updates by user and sorting them within each group.
- Handling document changes such as addition of new status updates.

Overall, the Status fragment provides users with an up-to-date view of recent status updates from their contacts, ensuring they stay informed about their network's activities.

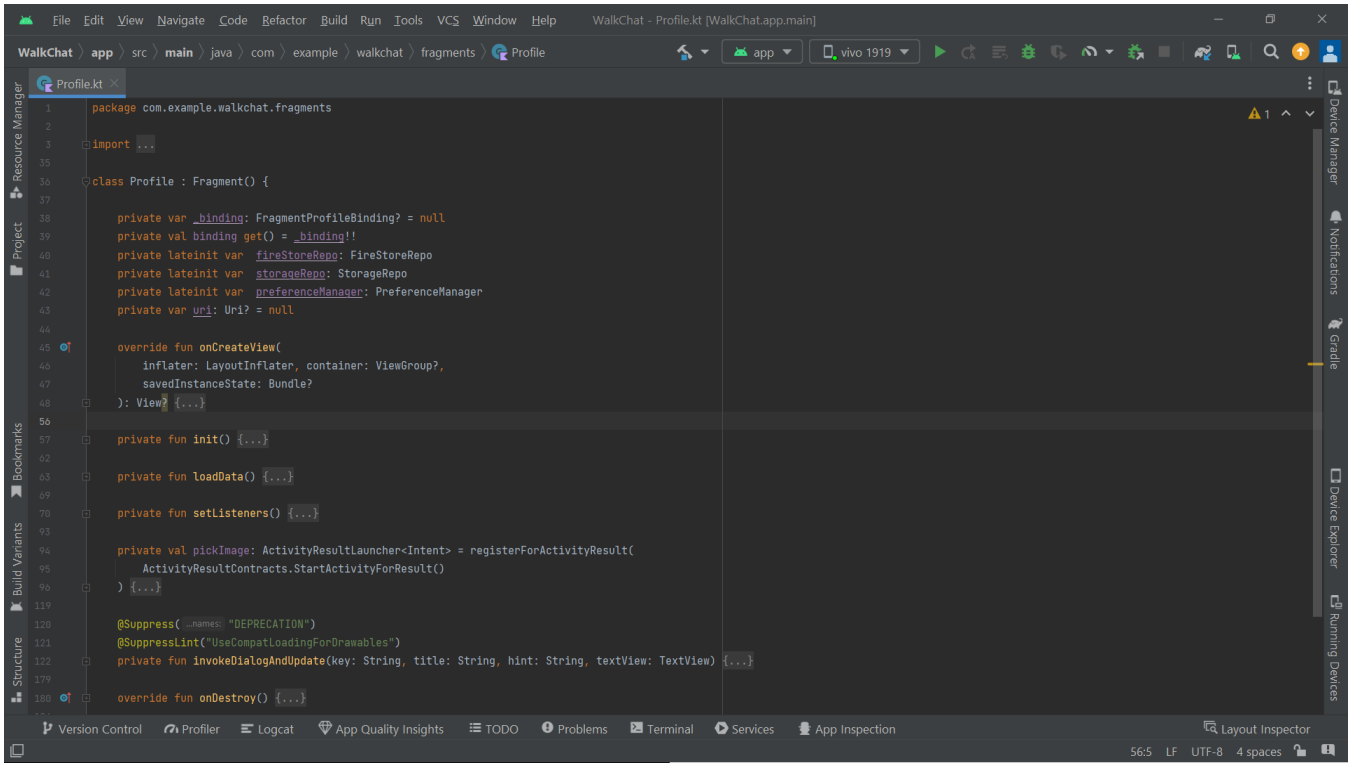


Fragment: III – Profile (User)

Key features include:

- Loading user profile data such as name, about, and mobile number from SharedPreferences.
- Providing functionality to update the profile picture by selecting an image from the device's gallery.
- Implementing dialogs for updating username and about information, with real-time updates to the UI and Firestore database.
- Utilizing Glide library for image loading and displaying default profile images if the user hasn't set one.
- Handling lifecycle events to prevent memory leaks.

Overall, the Profile fragment offers a user-friendly interface for managing profile information and updating it in real-time.

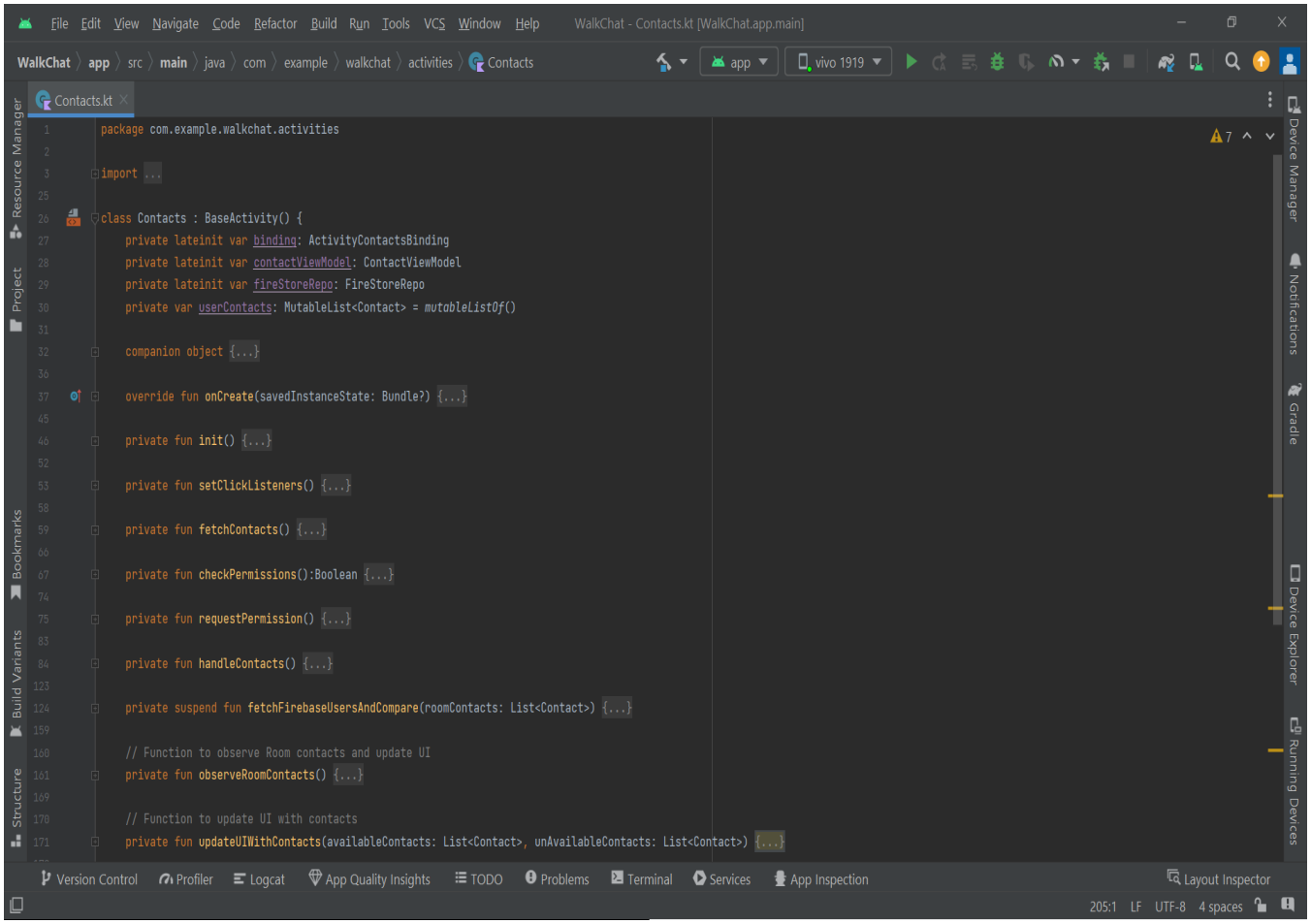


Activity: V – Contacts

Key features include:

- Requesting permission to access device contacts if not already granted.
- Handling the permission request result and proceeding with contact handling if granted.
- Fetching device contacts and comparing them with Firebase users to update their information in the Room database.
- Observing changes in Room database contacts and updating the UI accordingly.
- Displaying a progress bar while fetching and updating contacts to indicate loading.

Overall, the `Contacts` activity provides a seamless experience for users to interact with their contacts while efficiently managing data synchronization between local and remote sources.



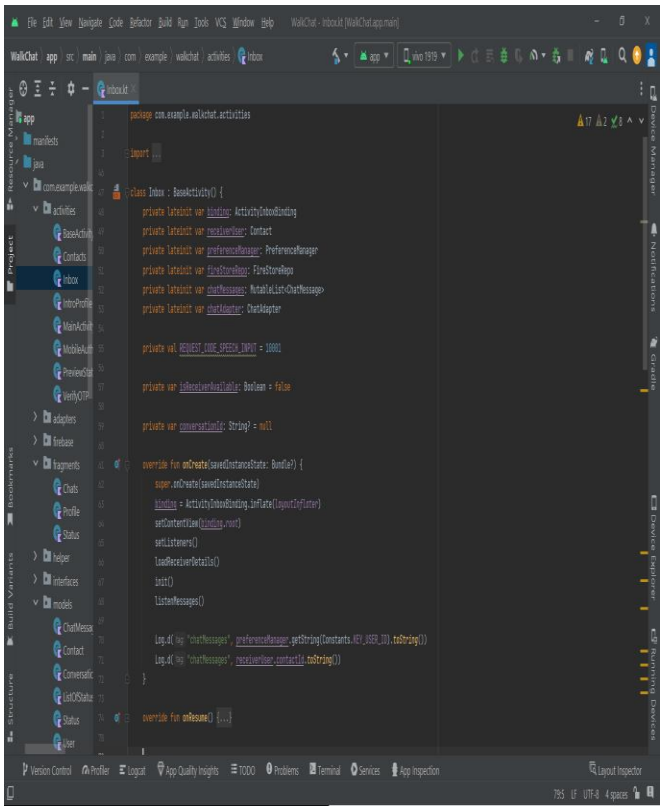
Activity: VI – Inbox

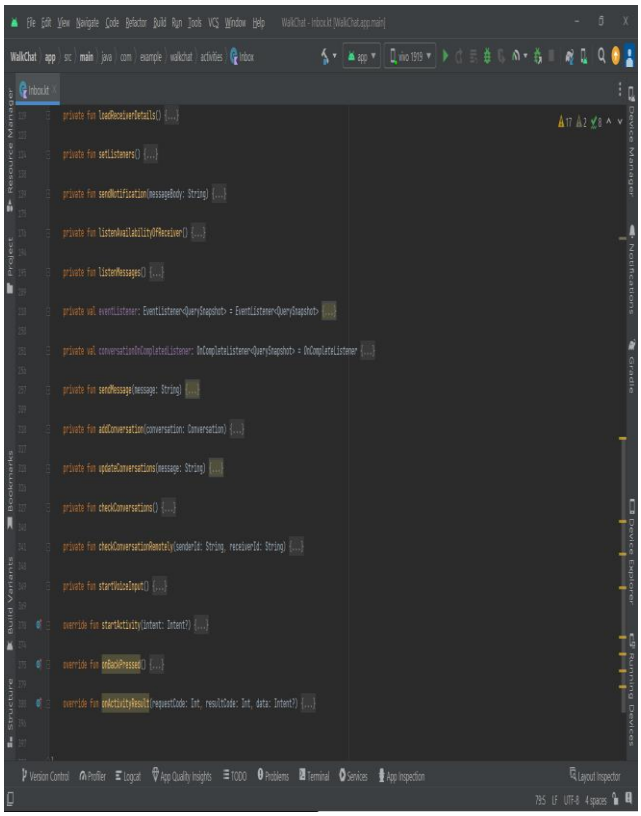
Key features include:

- Loading receiver details from the intent.
- Initializing necessary variables and adapters.
- Setting listeners for UI elements.
- Listening to messages using Firestore.
- Sending messages and updating conversations in Firestore.
- Handling speech input using speech recognition.
- Sending notifications to users when they are offline

.

Overall, the Inbox activity provides a comprehensive interface for users to exchange messages and interact with each other within the application.

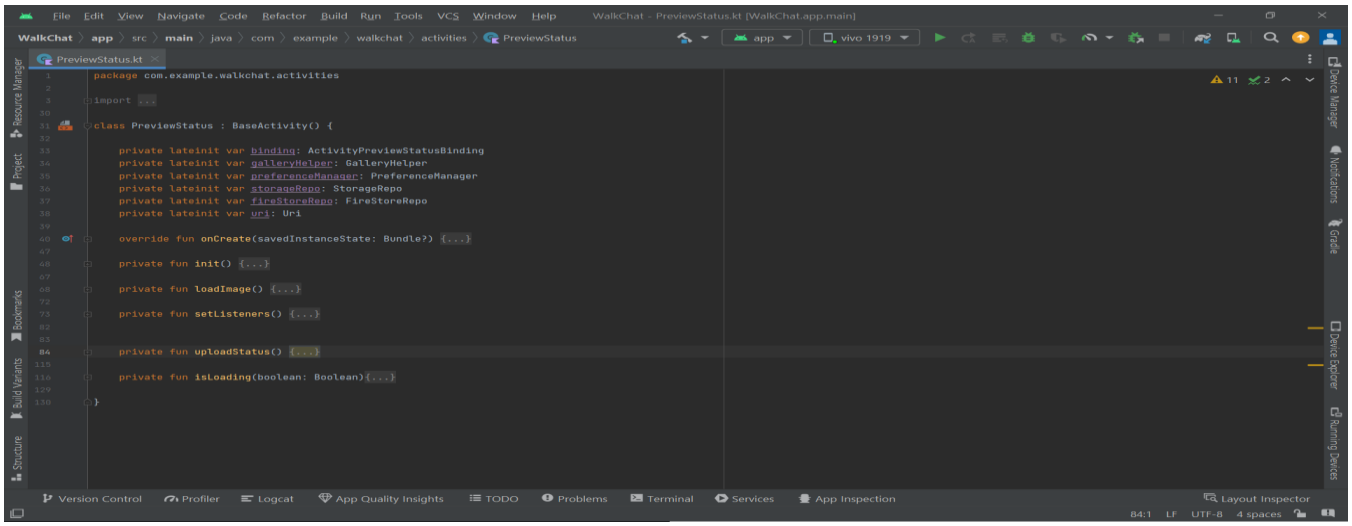




Activity: VII – Status

Key features Include:

- Initialization: It initializes necessary variables, such as galleryHelper, preferenceManager, storageRepo, and fireStoreRepo. It also checks the intent to determine the mode of operation: either uploading a new status or viewing a preview.
- Load Image: If the mode is upload, it loads the image from the URI passed in the intent using Glide and displays it in the ImageView.
- Set Listeners: It sets click listeners for the send button and back button.
- Upload Status: When the send button is clicked, it initiates the upload process. It first shows a progress bar to indicate loading. Then, it uploads the status image to Firebase Storage and adds a corresponding entry to Firestore. After completion, it shows a toast message indicating success or failure and finishes the activity.
- isLoading: This function manages the visibility and enable/disable state of UI elements based on the loading state. When loading is true, it disables the send button, input message, and back button, and shows the progress bar.



References

[1] Scroll View: https://developer.android.com/reference/android/widget/ScrollView

[2] Rating Bar: https://developer.android.com/reference/android/widget/RatingBar

[3] Intents: https://developer.android.com/guide/components/intents-filters

[4] Progress Bar: https://developer.android.com/reference/android/widget/ProgressBar

[5] Fragments: https://developer.android.com/guide/fragments

[6] Camera Intent: https://developer.android.com/training/camera/photobasics

[7] Splash Screen: https://developer.android.com/guide/topics/ui/declaring-layout

[8] Firebase: https://firebase.google.com/

[9] Creating Swipe Views with Tabs: https://developer.android.com/training/animation/screen-slide

[10] Creating Bottom Navigation Drawer: https://developer.android.com/guide/navigation/navigation-ui#bottom_nav

[11] Card View: https://developer.android.com/guide/topics/ui/layout/cardview

[12] Speech to Text Converter: https://developer.android.com/reference/android/speech/SpeechRecognizer

[13] Floating Action Button: https://developer.android.com/guide/topics/ui/floating-action-button

Bibliography

1. ANDROID APPLICATION DEVELOPMENT ALL-IN-ONE FOR DUMMIES by BARRY BURD, JOHN PAUL MUELLER, WILEY
2. ANDROID APPLICATION DEVELOPMENT by BARRY BURD, WILEY
3. ANDROID APPLICATION DEVELOPMENT by PRADEEP KOTHARI, DREAMTECH PRESS