



Send - Receive SMS Using Firebase

Grace Sharonnie Wijaya

Goals for this project

Create Chat Application to
read and send messages

Implementing Firebase
Authentication and Database



01

UI Design



Project Setup

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="my_primary">#1A214A</color>
  <color name="my_secondary">#FFED50</color>

  <color name="black">#191919</color>
  <color name="white">#FFFFFF</color>

  <color name="off_white">#D5D2D2</color>
  <color name="grey">#878585</color>
</resources>
```

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns:tools="http://schemas.android.com/tools">
  <!-- Base application theme. -->
  <style name="Base.Theme.SMSProject" parent="Theme.Material3.DayNight.NoActionBar">
    <item name="colorPrimary">@color/my_primary</item>
    <item name="colorPrimaryVariant">@color/my_primary</item>
    <item name="colorOnPrimary">@color/black</item>

    <item name="colorSecondary">@color/my_secondary</item>
    <item name="colorSecondaryVariant">@color/my_secondary</item>
    <item name="colorOnSecondary">@color/black</item>

    <item name="android:statusBarColor">@attr/colorPrimaryVariant</item>

    <item name="android:windowBackground">@color/off_white</item>
  </style>

  <style name="Theme.SMSProject" parent="Base.Theme.SMSProject" />
</resources>
```

```
<?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:background="@color/my_primary"
    tools:context=".SplashActivity">

    <LinearLayout
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        android:gravity="center"
        android:orientation="vertical">

        <ImageView
            android:layout_width="160dp"
            android:layout_height="160dp"
            android:src="@drawable/chat_icon"
            android:tint="@color/my_secondary" />

        <TextView
            android:layout_width="wrap_content"
            android:layout_height="wrap_content"
            android:text="SMS Project"
            android:textSize="40sp"
            android:textColor="@color/white"
            android:textStyle="bold"
            android:fontFamily="serif-monospace"
            android:layout_marginTop="38dp"/>
    </LinearLayout>
</RelativeLayout>
```



Chat App

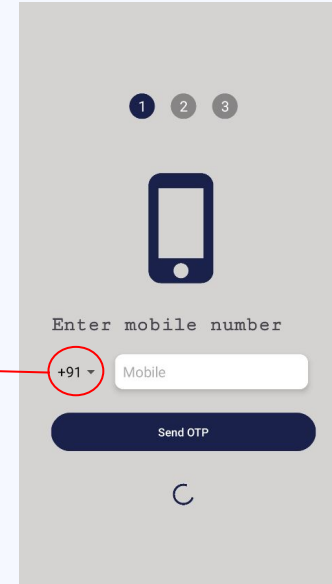


Phone Number Login

layout

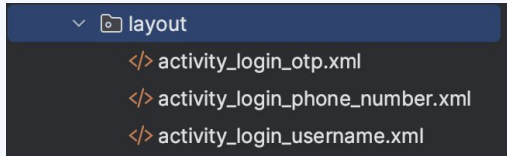
- </> activity_login_otp.xml
- </> activity_login_phone_number.xml
- </> activity_login_username.xml

```
dependencies { this: DependencyHandlerScope
    implementation("androidx.core:core-ktx:1.12.0")
    implementation("androidx.appcompat:appcompat:1.6.1")
    implementation("com.google.android.material:material:1.11.0")
    implementation("androidx.constraintlayout:constraintlayout:2.1.4")
    implementation("com.google.firebase:firebase-firestore:24.11.0")
    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.hbb28:ccp:2.5.0")
    implementation(platform("com.google.firebase:firebase-bom:32.8.0"))
    implementation("com.google.firebase:firebase-auth")
}
```



The image shows a mobile app UI for phone number login. At the top, there are three numbered steps: 1, 2, and 3. Step 1 is active. Below the steps is a smartphone icon. The text "Enter mobile number" is displayed. Below this text is a dropdown menu showing "+91" and a text input field labeled "Mobile". Below the input field is a dark blue button labeled "Send OTP". At the bottom, there is a circular arrow icon.

OTP and Username



1 2 3

*

Enter OTP sent to
your phone number

OTP

NEXT

C

Resend OTP in 30 seconds

This is a mockup of an OTP (One-Time Password) screen. It features a light gray background. At the top, there are three small circles containing the numbers 1, 2, and 3. Below them is a large, dark blue asterisk icon. The text 'Enter OTP sent to your phone number' is centered in a monospaced font. Below the text is a white input field with the placeholder text 'OTP'. Underneath the input field is a dark blue button with the text 'NEXT'. At the bottom, there is a circular refresh icon and the text 'Resend OTP in 30 seconds'.

1 2 3

Person icon

Enter your username

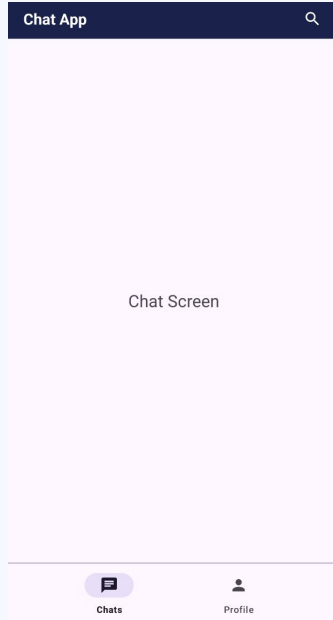
Username

Finish

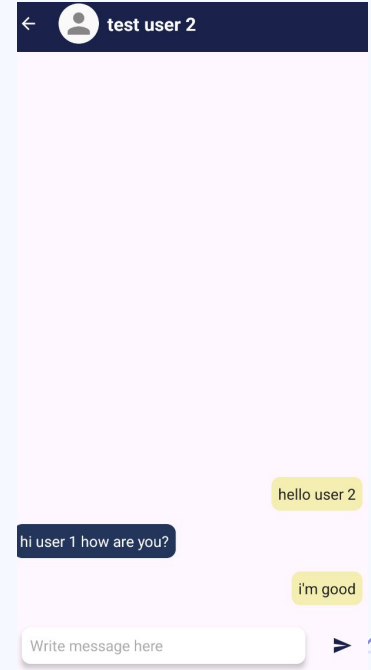
C

This is a mockup of a username entry screen. It features a light gray background. At the top, there are three small circles containing the numbers 1, 2, and 3. Below them is a dark blue icon of a person. The text 'Enter your username' is centered in a monospaced font. Below the text is a white input field with the placeholder text 'Username'. Underneath the input field is a dark blue button with the text 'Finish'. At the bottom, there is a circular refresh icon.

Main Page Search User



Chat Page





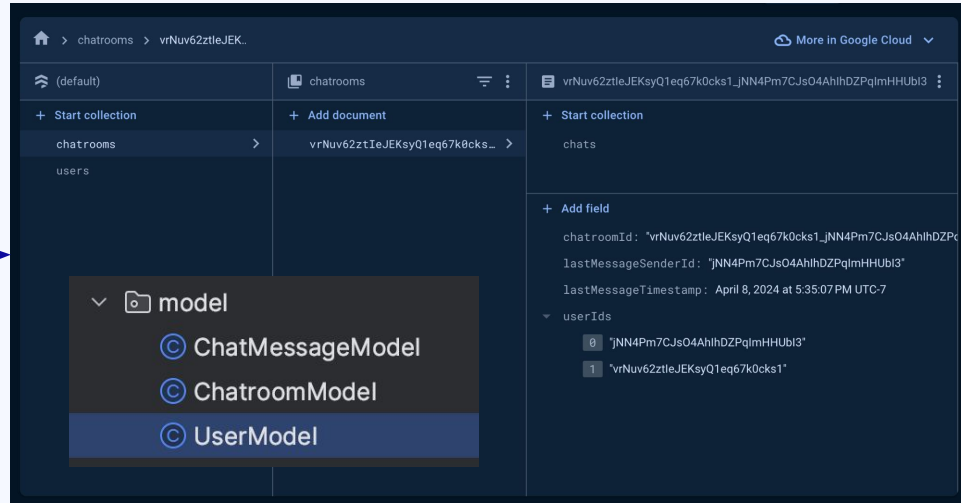
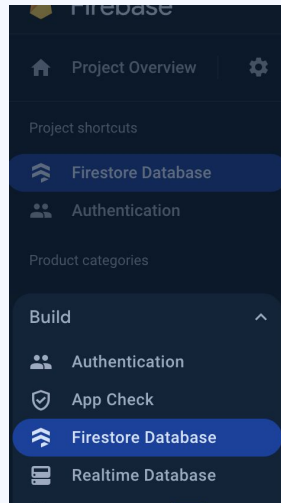
02

Firestore

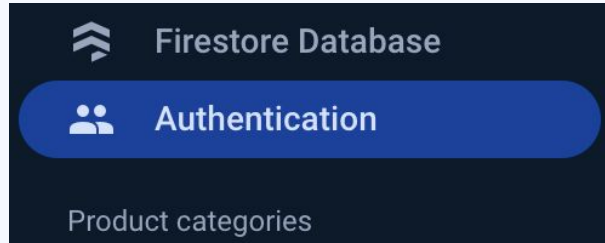


Connect Project & Create Database

```
implementation("com.google.firebase:firebase-firestore:24.11.0")
```



Phone Authentication OTP



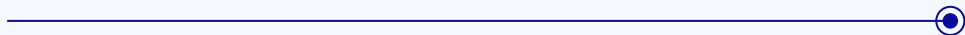
3f:d4:e3:21:a5:3f:44:d4:35:43:19:bc:95:6d:da:e8:74:0f:7c:75	SHA-1
3a:67:be:2c:60:ed:99:a5:86:6f:9b:31:67:93:00:d0:08:07:2c:44:db:b0:96:43:ea:be:1a:6d:e8:ae:b4:c4	SHA-256

Add SHA-1 & SHA-256 Fingerprint



03

SMS



Create Chatroom Model

- Declare variables to put into firebase
- **getOrCreateChatroomModel()**: fetches the chatroom model from Firestore using the chatroomId. If the chatroom doesn't exist, it creates a new one with the current user's ID and the other user's ID

```
void getOrCreateChatroomModel(){
    FirebaseUtil.getChatroomReference(chatroomId).get().addOnCompleteListener(task -> {
        if (task.isSuccessful()){
            chatroomModel = task.getResult().toObject(ChatroomModel.class);
            if (chatroomModel==null){
                Date now = new Date();
                Timestamp timestamp = new Timestamp(now.getTime());
                //first time chatting
                chatroomModel = new ChatroomModel(
                    chatroomId,
                    Arrays.asList(FirebaseUtil.currentUserId(),otherUser.getUserId()),
                    timestamp,
                    lastMessageSenderId: ""
                );
                FirebaseUtil.getChatroomReference(chatroomId).set(chatroomModel);
            }
        }
    });
}
```

```
public class ChatroomModel {
    3 usages
    String chatroomId;
    3 usages
    List<String> userIds;
    3 usages
    Date lastMessageTimestamp;
    3 usages
    String lastMessageSenderId;

    no usages
    public ChatroomModel() {
    }

    1 usage
    public ChatroomModel(String chatroomId, List<String> userIds, Date lastMessageTimestamp, String lastMessageSenderId) {
        this.chatroomId = chatroomId;
        this.userIds = userIds;
        this.lastMessageTimestamp = lastMessageTimestamp;
        this.lastMessageSenderId = lastMessageSenderId;
    }
}
```

Send Message

- Declare variables to put into firebase
- **sendMessageToUser()**: creates a new ChatMessageModel object with the message, sender ID, and timestamp. It then updates the chatroom's last message timestamp and sender ID, adds the message to the Firestore database

```
void sendMessageToUser(String message){
    Date now = new Date();
    Timestamp timestamp = new Timestamp(now.getTime());

    chatroomModel.setLastMessageTimestamp(timestamp);
    chatroomModel.setLastMessageSenderId(FirebaseUtil.currentUser());
    FirebaseUtil.getChatroomReference(chatroomId).set(chatroomModel);

    ChatMessageModel chatMessageModel = new ChatMessageModel(message, FirebaseUtil.currentUser(), timestamp);
    FirebaseUtil.getChatRoomMessageReference(chatroomId).add(chatMessageModel)
        .addOnCompleteListener(new OnCompleteListener<DocumentReference>() {
            @Override
            public void onComplete(@NonNull Task<DocumentReference> task) {
                if (task.isSuccessful()){
                    messageInput.setText("");
                }
            }
        });
}
```

```
public class ChatMessageModel {
    3 usages
    private String message;
    3 usages
    private String senderId;
    3 usages
    private Date timestamp;

    no usages
    public ChatMessageModel() {
    }

    1 usage
    public ChatMessageModel(String message, String senderId, Date timestamp) {
        this.message = message;
        this.senderId = senderId;
        this.timestamp = timestamp;
    }
}
```

Resources

https://youtube.com/playlist?list=PLgpnJydBcnPB-aQ6P5hWCHBjy8LWZ9x4w&si=ByNu1yQ_Fvme5SbL