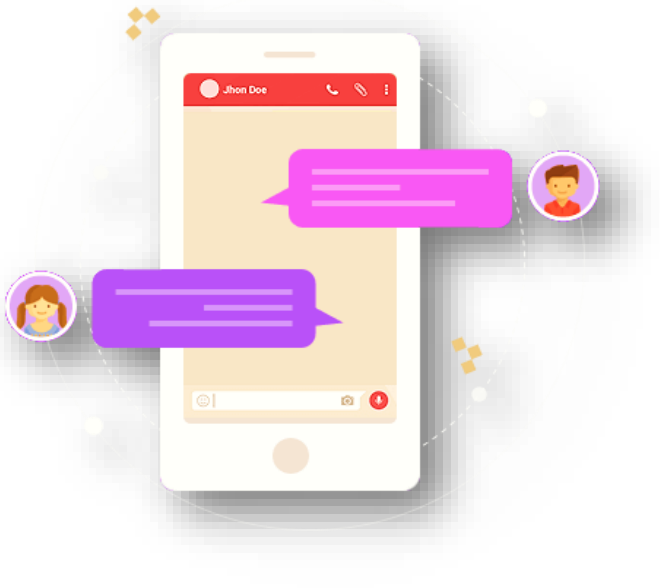
Manarat International University

Department of Computer Science & Engineering

Software Development (CSE312)

Project Title: MIU Messenger

Category: Mobile Application



Team Name: Unique Developer

Starting Git Repository Link: ChattingApplication

Final Git Repository Link: MIU Messenger

Promo Video Link: Presentation on MIU Messenger

Student Name: Md. Bodrul Alam

Student ID: 1846CSE00711

Objectives

Communication through internet is becoming vital these days. An online communication allows the users to communicate with other people in a fast and convenient way. Considering this, the online communication application must be able share the texts or images or any other files in a faster way with minimum delay or with no delay. Firebase is one of the platforms which provides a real-time database and cloud services which allows the developer to make these applications with ease. Instant messaging can be considered as a platform to maintain communication. Android provides better platform to develop various applications for instant messaging compared to other platforms such as iOS. The main objective of this paper is to present a software application for the launching of a real time communication between MIU members. Because in this application able to create account those users who has MIU domain mail like (example@manarat.ac.bd). The system developed on android will enable the users to communicate with another users through text messages with the help of internet. The system requires both the device to be connected via internet. This application is based on Android with the backend provided by google Firebase.

Introduction

This document is the requirements document of a chat application project that is MIU Messenger.

1. Purpose and Scope

   The purpose of the MIU Messenger is to allow users be able to the chat with each other of members of Manarat International University, like a normal chat application. Group chat is allowed in this application. Also users able to see portal, news, events and Academic calendar of Manarat University.

   Scope of the project, University has provided domain mail for all members and who has domain mail only they would create an account in this application. User can share Texts, Photos and other files through internet.

2. Target Audience

   Here has some restrictions about audience. The target audience is only MIU members who has MIU domain mail.

Features Description

There so many features in this application for users.

Login:

When user will run this application they will welcomed in log in page. If they have an account of this application they can log in. Where email and password is required. If don't have account then there has a option for going to signup page.

Signup:

In this page, user will create an account for MIU Messenger user name, Manarat domain mail and pass word is required.

Navigation Drawer:

After logged in, user will go to the main activity where will see the Individual chats and when they click or move from left to right they will see the navigation drawer. Where has two part header and menu. In header, user's personal information will be shown. In menu, the menu items are Individual chats, group chats, portal, news , events ,notice, academic calendar, settings, and logout.

Logout:

When user wants they can log out from this application. They will not lose their data because there has real time database management system.

Individual Chat:

Here user can chat individually with each other.

Group Chat:

Here has one group for all members

Portal:

This is for MIU students from MIU website.

News:

MIU news will be show from MIU website.

Events:

MIU events will be show from MIU website.

Notice:

MIU notice will be show from MIU website.

Academic Calendar:

MIU academic calendar will be show from MIU website.

Settings:

Here user can set up their profile. They can edit username, profile picture, about any time.

Functional:

Message send and receive:

Sender will send message it can be text, photos or other files and receiver will receive it automatically.

Last message and time count:

When user send message the last message and time will be count.

Presence:

When a users open this application and if they connected by internet then receiver will see that the sender is in online otherwise offline. The amazing implement is when sender will typing something the receiver able to see that sender is typing.

Reaction on Message:

When user touch or click the on a message there will be open a popup of reaction user can choose the reaction for message.

Delete Message:

When user will long touch on left side or right side on a message there will be open a popup there will be option for delete message and cancel the popup. If they click on delete the message will be deleted.

Platform

This is a mobile application. This application developed for android system. Only android user can use this application.

Libraries/Dependencies

```
implementation 'androidx.appcompat:appcompat:1.2.0'
implementation 'com.google.android.material:material:1.3.0'
implementation'androidx.constraintlayout:constraintlayout:2.0.4'

implementation 'androidx.navigation:navigation-fragment:2.3.4'
implementation 'androidx.navigation:navigation-ui:2.3.4'
implementation 'androidx.lifecycle:lifecycle-livedata-ktx:2.3.1'
implementation 'androidx.lifecycle:lifecycle-viewmodel-ktx:2.3.1'
implementation "androidx.multidex:multidex:2.0.1"
implementation 'androidx.swiperefreshlayout:swiperefreshlayout:1.0.0'
implementation 'com.squareup.picasso:picasso:2.71828'
implementation 'com.github.sharish:ShimmerRecyclerView:v1.3'
implementation 'de.hdodenhof:circleimageview:3.1.0'

implementation 'com.github.bumptech.glide:glide:4.11.0'

annotationProcessor 'com.github.bumptech.glide:compiler:4.11.0'

implementation "com.github.pgreze:android-reactions:1.4

implementation platform('com.google.firebase:firebase-bom:26.8.0')
implementation 'com.google.firebase:firebase-analytics'
implementation 'com.google.firebase:firebase-auth'
implementation 'com.google.firebase:firebase-database'
implementation 'com.google.firebase:firebase-storage'
```

```
implementation 'com.google.firebase:firebase-messaging'
implementation 'androidx.legacy:legacy-support-v4:1.0.0'
implementation 'com.google.firebase:firebase-messaging'
testImplementation 'junit:junit:4.+'
androidTestImplementation 'androidx.test.ext:junit:1.1.2'
androidTestImplementation 'androidx.test.espresso:espresso-core:3.3.0'
```

Database

For this application we have used Firebase Database. This is the real time database management system where will be store all user's data in online storage. User can access from any android device which is connected with internet.

Source code

Main Activity:

```
public class MainActivity extends AppCompatActivity {
    private DrawerLayout drawer;
    private NavigationView navigationView;
    private Toolbar toolbar;
    private ActionBarDrawerToggle toggle;
    FirebaseAuth auth;
    FirebaseDatabase database;
    ActivityMainBinding binding;
    FragmentManager fragmentManager;
    FragmentTransaction fragmentTransaction;
    DrawerHeaderBinding bind;
    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        binding = ActivityMainBinding.inflate(getLayoutInflater());
        setContentView(binding.getRoot());
        auth = FirebaseAuth.getInstance();
        database = FirebaseDatabase.getInstance();
        toolbar = findViewById(R.id.toolbar);
        setSupportActionBar(toolbar);
        drawer = findViewById(R.id.drawer_layout);
        navigationView = findViewById(R.id.nav_view);
        View header= navigationView.getHeaderView(0);
        database.getReference().child("users").child(FirebaseAuth.getInstance().getUid())
                .addValueEventListener(new ValueEventListener() {
                    @Override
                    public void onDataChange(@NonNull DataSnapshot snapshot) {
                        Users user = snapshot.getValue(Users.class);
                        ImageView userProfilePic = header.findViewById(R.id.userProfilePic);
                        TextView userName = header.findViewById(R.id.userName);
                        TextView userEmail = header.findViewById(R.id.userEmail);
                        TextView about = header.findViewById(R.id.about);
                            userName.setText(user.getUserName());
                            userEmail.setText(user.getEmail());
                            about.setText(user.getAbout());
Picasso.get().load(user.getProfilePic()).placeholder(R.drawable.avatar).into(userProfilePic);
```

```
                }
                @Override
                public void onCancelled(@NonNull DatabaseError error) {
                }
            });
        toggle = new ActionBarDrawerToggle(this,drawer,toolbar,R.string.navigation_drawer_open,R.string.navigation_drawer_close);
        drawer.addDrawerListener(toggle);
        toggle.setDrawerIndicatorEnabled(true);
        toggle.syncState();
        //load layout
        fragmentManager = getSupportFragmentManager();
        fragmentTransaction = fragmentManager.beginTransaction();
        fragmentTransaction.add(R.id.fragment_container,new ChatsFragment());
        fragmentTransaction.commit();
    navigationView.setNavigationItemSelectedListener(new NavigationView.OnNavigationItemSelectedListener() {
        @Override
        public boolean onNavigationItemSelected(@NonNull MenuItem item) {
            drawer.closeDrawer(GravityCompat.START);
            switch (item.getItemId()){
                case R.id.nav_logout:
                    auth.signOut();
                    Toast.makeText(MainActivity.this, "Logged Out",Toast.LENGTH_SHORT).show();
                    Intent intentSignIn = new Intent(MainActivity.this,SignInActivity.class);
                    startActivity(intentSignIn);
                    finish();
                    break;
                case R.id.nav_settings:
                    fragmentManager = getSupportFragmentManager();
                    fragmentTransaction = fragmentManager.beginTransaction();
                    fragmentTransaction.replace(R.id.fragment_container, new SettingsFragment());
                    fragmentTransaction.commit();
                     break;
                case R.id.nav_idividual:
                    fragmentManager = getSupportFragmentManager();
                    fragmentTransaction = fragmentManager.beginTransaction();
                    fragmentTransaction.replace(R.id.fragment_container, new ChatsFragment());
                    fragmentTransaction.commit();
                    break;
                case R.id.nav_groups:
                    Intent intent = new Intent(MainActivity.this, GroupChatActivity.class);
                    startActivity(intent);
                    break;
                case R.id.nav_portal:
                    fragmentManager = getSupportFragmentManager();
                    fragmentTransaction = fragmentManager.beginTransaction();
                    fragmentTransaction.replace(R.id.fragment_container, new PortalFragment());
                    fragmentTransaction.commit();
                    break;
                case R.id.nav_news:
                    fragmentManager = getSupportFragmentManager();
                    fragmentTransaction = fragmentManager.beginTransaction();
                    fragmentTransaction.replace(R.id.fragment_container, new NewsFragment());
                    fragmentTransaction.commit();
                    break;
                case R.id.nav_academicCalender:
                    fragmentManager = getSupportFragmentManager();
                    fragmentTransaction = fragmentManager.beginTransaction();
fragmentTransaction.replace(R.id.fragment_container, new AcademicCalenderFragment());
```

```
                fragmentTransaction.commit();
                break;
            case R.id.nav_notice:
                fragmentManager = getSupportFragmentManager();
                fragmentTransaction = fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.fragment_container, new NoticeFragment());
                fragmentTransaction.commit();
                break;
            case R.id.nav_events:
                fragmentManager = getSupportFragmentManager();
                fragmentTransaction = fragmentManager.beginTransaction();
                fragmentTransaction.replace(R.id.fragment_container, new EventFragment());
                fragmentTransaction.commit();
                break;

        }
        return true;

    }
});

}
protected void onResume() {
    super.onResume();
    String currentId = FirebaseAuth.getInstance().getUid();
    database.getReference().child("presence").child(currentId).setValue("Online");
}
@Override
protected void onPause() {
    super.onPause();
    String currentId = FirebaseAuth.getInstance().getUid();
    database.getReference().child("presence").child(currentId).setValue("Offline");
}
}
```

Activity_Layout:

```
<?xml version="1.0" encoding="utf-8"?>
<androidx.drawerlayout.widget.DrawerLayout 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:id="@+id/drawer_layout"
    android:layout_width="match_parent"
    android:layout_height="match_parent"
    android:fitsSystemWindows="true"
    tools:openDrawer="start">
    <include
        layout="@layout/app_toolbar"
        android:layout_width="match_parent"
        android:layout_height="match_parent"
        />

    <include
        layout="@layout/content_main"
        android:layout_width="match_parent"
        android:layout_height="match_parent" />
```

```
<com.google.android.material.navigation.NavigationView
    android:id="@+id/nav_view"
    android:layout_width="wrap_content"
    android:layout_height="match_parent"
    android:layout_gravity="start"
    android:fitsSystemWindows="true"
    android:visibility="visible"
    app:headerLayout="@layout/drawer_header"
    app:menu="@menu/drawer_menu" />

</androidx.drawerlayout.widget.DrawerLayout>
```

Future plan

This application is implementing for MIU members. If user's has any requirements that will be better for this application developer team will try to implement it. Some features will add the developer team in future.

1. End-to-end encryption:
   End-to-end encryption (E2EE) is a system of communication where only the communicating users can read the messages. The messages are encrypted by the sender but the third party does not have a means to decrypt them, and stores them encrypted. The recipients retrieve the encrypted data and decrypt it themselves.

2. Group chat:

   There will be available some necessary groups for users some common groups and some for batch and department.

3. Email Varification

4. Push notifications:
   Database will send a notification when a new message come from sender.

5. Seen and Unseen Message:
   When a user will see the message it will be marked as seen otherwise unseen.

6. Audio/Video Chat

7. Canned response

8. Post

9. Bus Schedule of MIU