

# Adaptive Mail: A Flexible Email Client App

## Introduction

### 1.1. Overview

#### Project Description

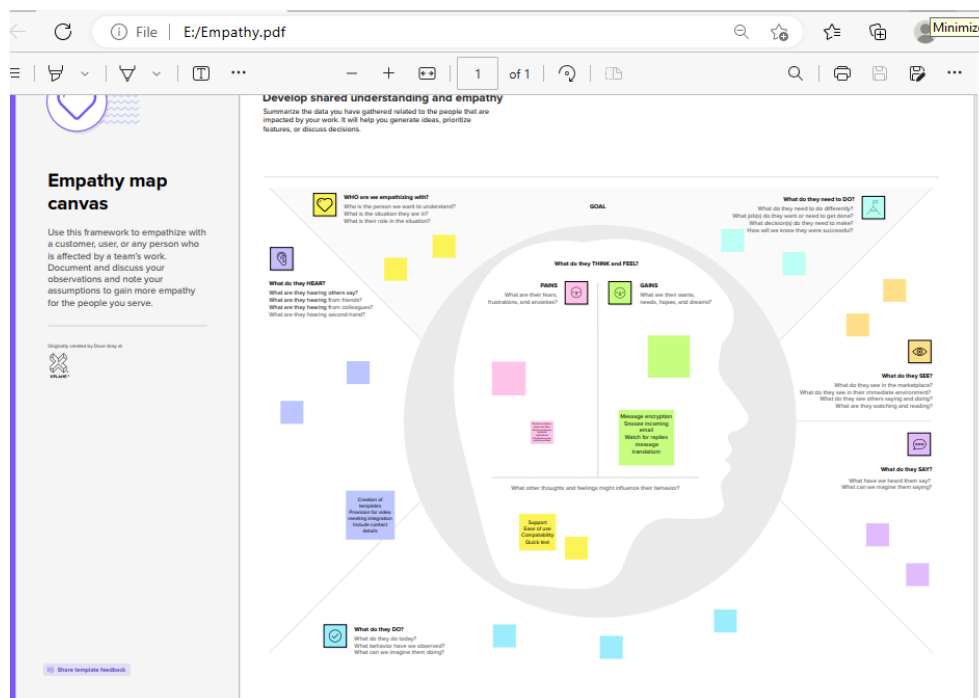
**Adaptive Mail: A Flexible Email Client App** makes use of Android Compose UI toolkit to build an app to allow users to send and receive messages. It maintains the history of previous messages as well.

#### Purpose

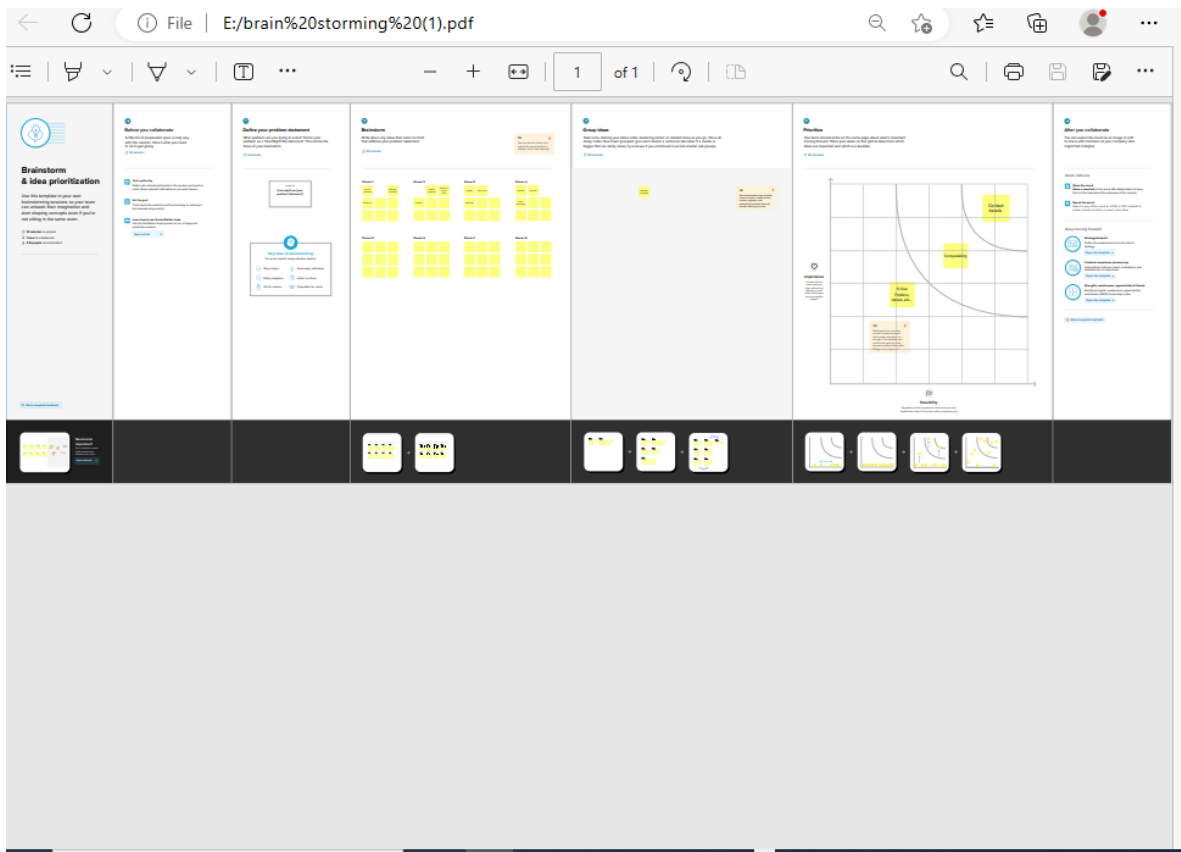
The Adaptive mail app can be used to send and receive text messages easily.

#### Problem Definition and thinking

### 2.1. Empathy Map

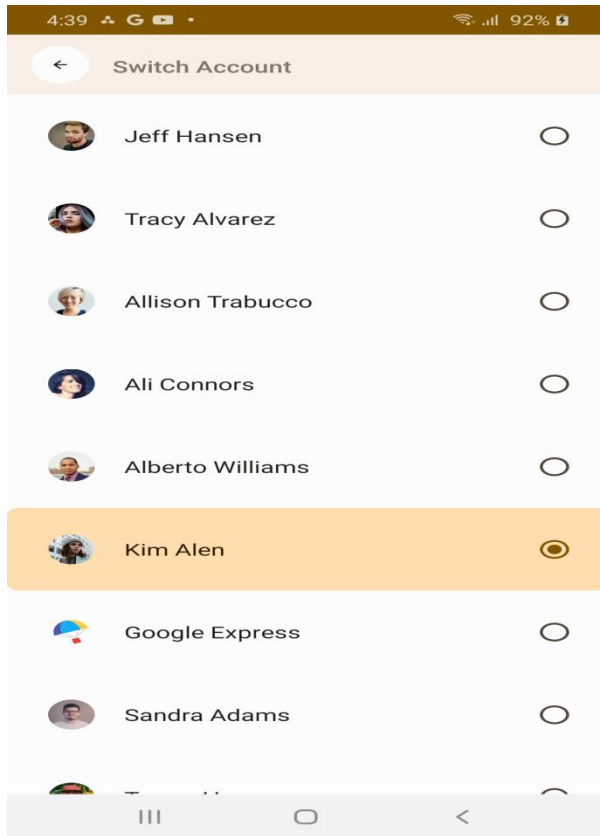


2.2. Ideation and Brainstorming Map

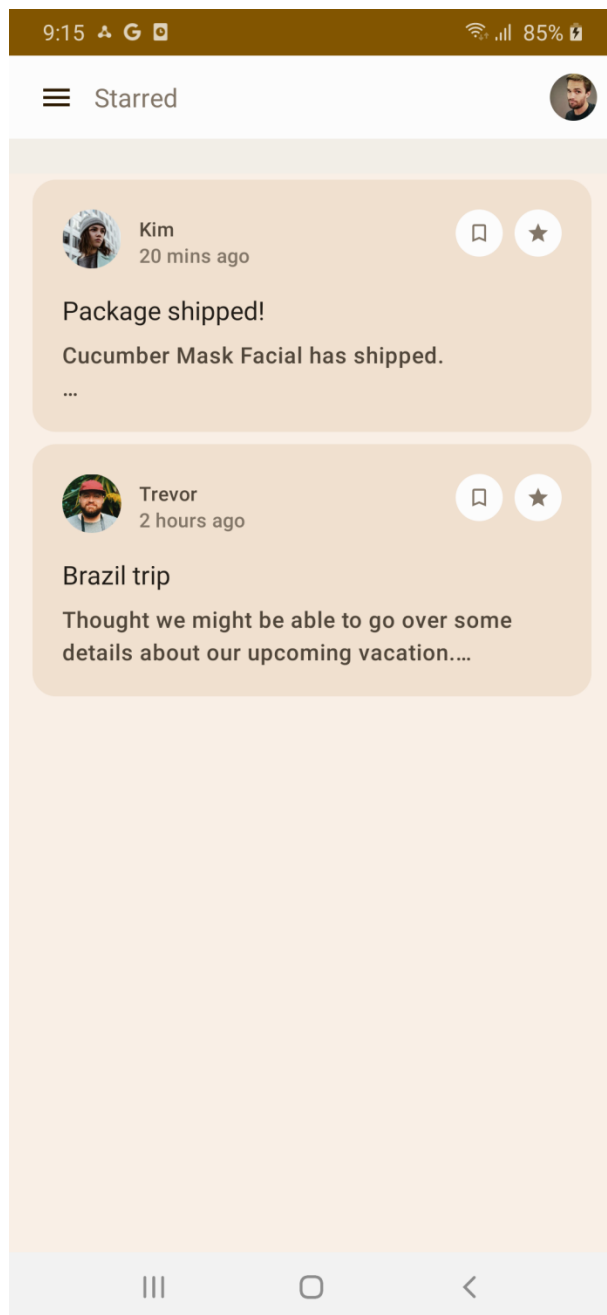


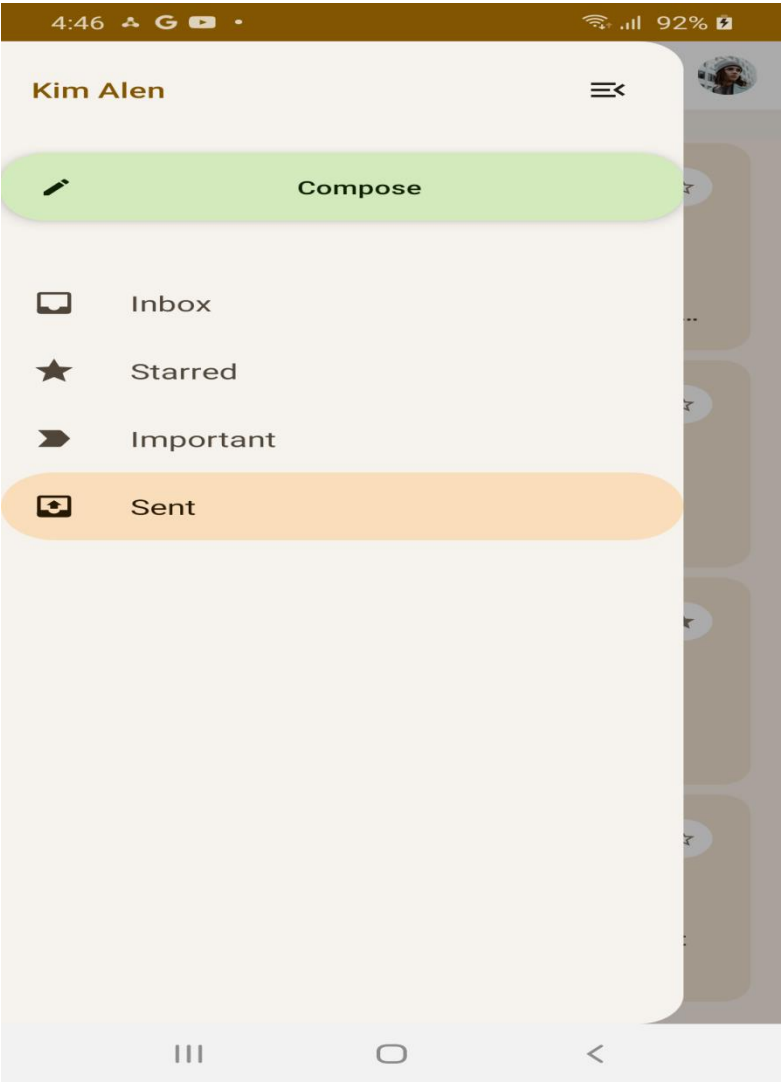
## Result

### Switch Other Account

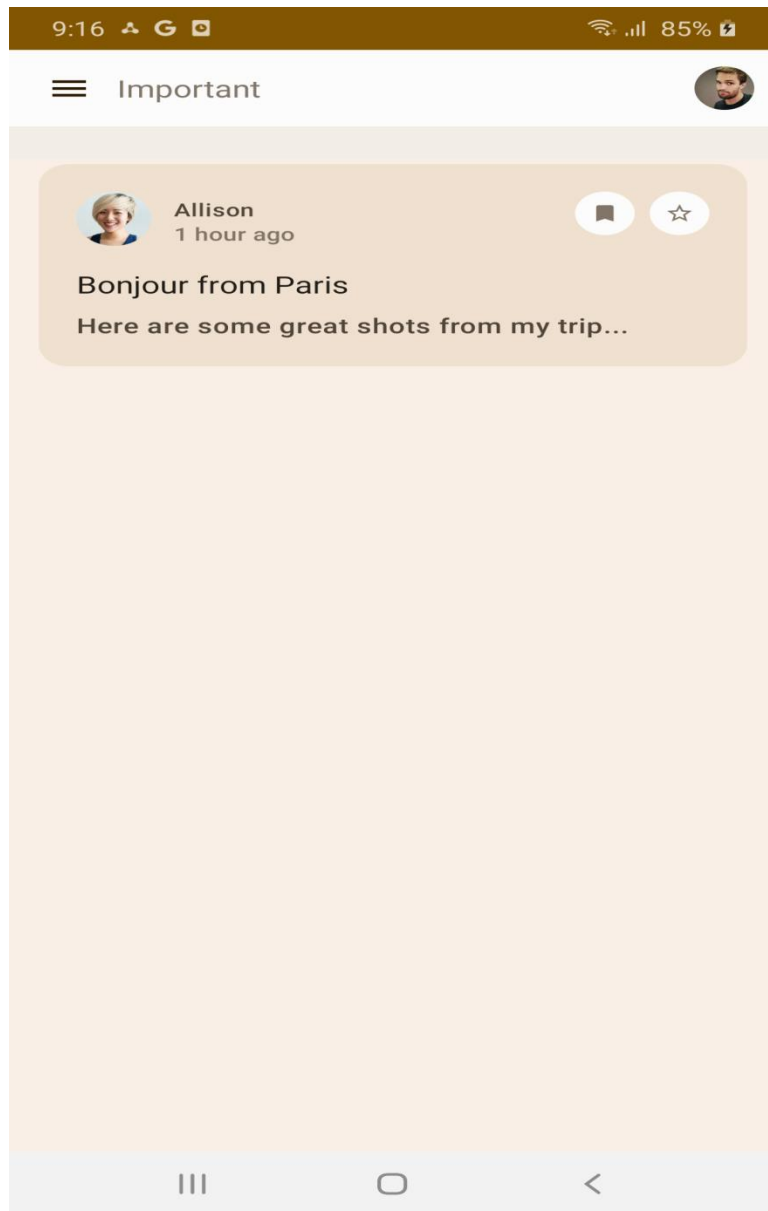


## Starred

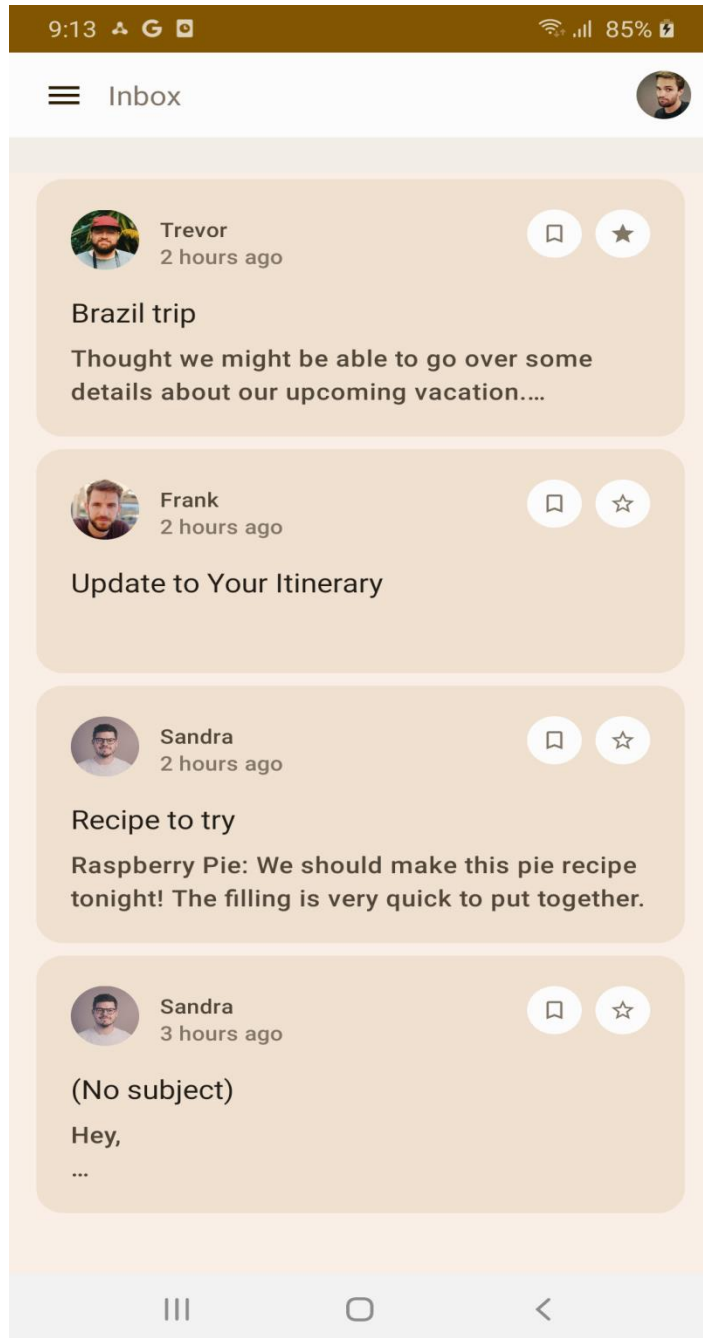




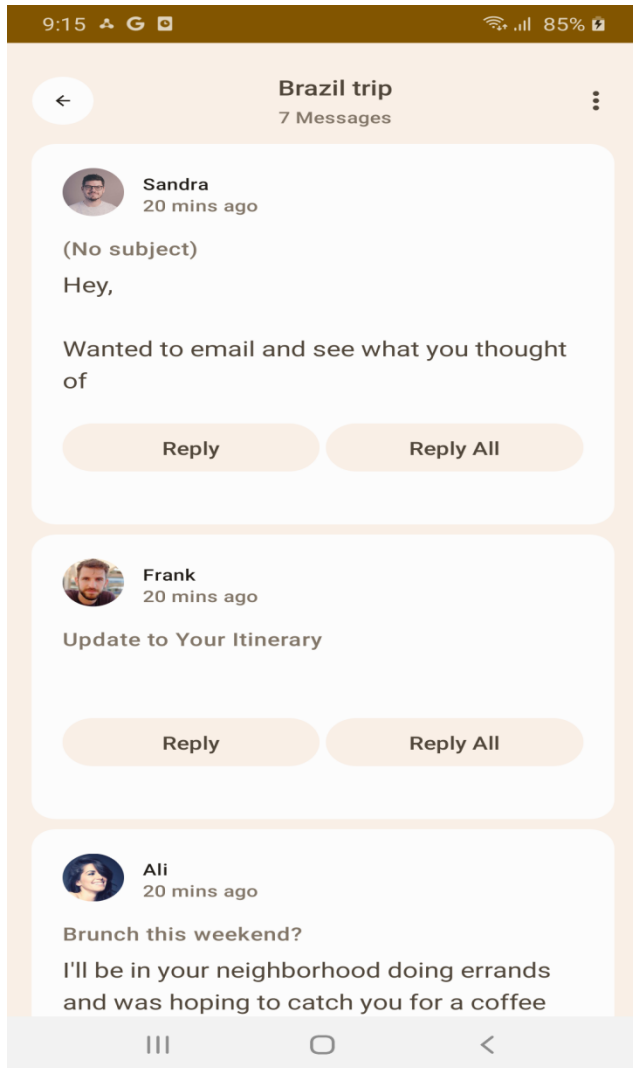
## Important Mails



## Inbox






## Detail









**Compose**


4:38




 92% 




Compose





From: (current account)


 **Kim**

To: (multiple selection available)

 **Jeff**

 **Tracy**

 **Allison**





Subject


Trinfy Award

Compose email

You make as proud











## **Advantages**

Adaptive mail provides faster and easy mean of communication. One can send message to any person at any place of world by just clicking mouse. It is effective and cheap means of communication because single message can be send to multiple people at same time.

## **Disadvantages**

The mails can be source of various spams which may fill up inbox and to delete mail consumes lot of time. It is an informal method of communication. To use the facility of e-mail, user must have an access to internet and there are many parts of world where people does not have access to Internet.

## **Conclusion**

Mailis one of most widely used services of Internet. This service allows an Internet user to send a message in formatted manner to other Internet user in any part of the world.

## **Future Scope**

The app could be modified in such a way that it supports file transfer and multimedia files.

## **Appendix**

### **A. Source Code**

#### **AndroidManifest.xml**

```
<?xml
version="1.0"
encoding="utf-
8"?>

    <manifest xmlns:android="http://schemas.android.com/apk/res/android"
        xmlns:tools="http://schemas.android.com/tools" >
        <application
            android:allowBackup="true"
            android:dataExtractionRules="@xml/data_extraction_rules"
```

```

        android:fullBackupContent="@xml/backup_rules"
        android:icon="@mipmap/ic_launcher"
        android:label="@string/app_name"
        android:supportsRtl="true"
        android:theme="@style/Theme.EmailApplication"
        tools:targetApi="31" >
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="@string/title_activity_register"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".MainActivity"
            android:exported="false"
            android:label="MainActivity"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".ViewMailActivity"
            android:exported="false"
            android:label="@string/title_activity_view_mail"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".SendMailActivity"
            android:exported="false"
            android:label="@string/title_activity_send_mail"
            android:theme="@style/Theme.EmailApplication" />
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.EmailApplication" >
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
</manifest>

```

```
package
```

```
com.example.emailapplication
```

```
import android.content.Context
```

```

import android.content.Intent
import android.os.Bundle
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.background
import androidx.compose.foundation.layout.*
import androidx.compose.material.*
import androidx.compose.runtime.*
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.layout.ContentScale
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import
androidx.compose.ui.text.input.PasswordVisualTransformation
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
import
com.example.emailapplication.ui.theme.EmailApplicationTheme
class LoginActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
            LoginScreen(this, databaseHelper)
        }
    }
}
@Composable
fun LoginScreen(context: Context, databaseHelper:
UserDatabaseHelper) {
    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }
    Column(
        modifier =
Modifier.fillMaxSize().background(Color.White),

```

```

horizontalAlignment = Alignment.CenterHorizontally,
verticalArrangement = Arrangement.Center
    ) {
        Image(
            painterResource(id = R.drawable.email_login),
            contentDescription = ""
        )
        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Login"
        )
        Spacer(modifier = Modifier.height(10.dp))
        TextField(
            value = username,
            onChange = { username = it },
            label = { Text("Username") },
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )
        TextField(
            value = password,
            onChange = { password = it },
            label = { Text("Password") },
            visualTransformation = PasswordVisualTransformation(),
            modifier = Modifier.padding(10.dp)
                .width(280.dp)
        )
        if (error.isNotEmpty()) {
            Text(
                text = error,
                color = MaterialTheme.colors.error,
                modifier = Modifier.padding(vertical = 16.dp)
            )
        }
        Button(
            onClick = {
                if (username.isNotEmpty()
                    &&password.isNotEmpty()) {
                    val user = databaseHelper.getUserByUsername(username)
                    if (user != null &&user.password ==
                        password) {

```



```

        error = "Successfully log in"
context.startActivity(
        Intent(
            context,
MainActivity::class.java
        )
    )
    //onLoginSuccess()
}
} else {
    error = "Please fill all fields"
}
},
colors =
ButtonDefaults.buttonColors(backgroundColor =
Color(0xFFd3e5ef)),
modifier = Modifier.padding(top = 16.dp)
) {
    Text(text = "Login")
}
Row {
TextButton(onClick = {context.startActivity(
        Intent(
            context,
RegisterActivity::class.java
        )
    ))
    { Text(color = Color(0xFF31539a),text = "Sign up")
}
TextButton(onClick = {
    })
    {
        Spacer(modifier = Modifier.width(60.dp))
Text(color = Color(0xFF31539a),text = "Forget password?")
    }
}
}
}
private fun startMainPage(context: Context) {
val intent = Intent(context, MainActivity::class.java)
ContextCompat.startActivity(context, intent, null)
}

```

```

package
com.example.emailapplication

import androidx.test.platform.app.InstrumentationRegistry
import androidx.test.ext.junit.runners.AndroidJUnit4
import org.junit.Test
import org.junit.runner.RunWith
import org.junit.Assert.*
/**
 * Instrumented test, which will execute on an Android device.
 *
 * See [testing
documentation](http://d.android.com/tools/testing).
 */
@RunWith(AndroidJUnit4::class)
class ExampleInstrumentedTest {
    @Test
    fun useAppContext() {
        // Context of the app under test.
        val appContext =
            InstrumentationRegistry.getInstrumentation().targetContext
        assertEquals("com.example.emailapplication",
            appContext.packageName)
    }
}

```

## build.gradle

```

plugins
{
    id 'com.android.application'
    id 'org.jetbrains.kotlin.android'
}
android {
    namespace 'com.example.emailapplication'
    compileSdk 33
    defaultConfig {
        applicationId "com.example.emailapplication"
        minSdk 21
        targetSdk 33
        versionCode 1
        versionName "1.0"
        testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    }
}

```

```

vectorDrawables {
    useSupportLibrary true
}
}
buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    }
}
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
kotlinOptions {
    jvmTarget = '1.8'
}
buildFeatures {
    compose true
}
composeOptions {
    kotlinCompilerExtensionVersion '1.2.0'
}
packagingOptions {
    resources {
        excludes += '/META-INF/{AL2.0,LGPL2.1}'
    }
}
}
dependencies {
    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-compose:1.3.1'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-tooling-preview:$compose_ui_version"
    implementation 'androidx.compose.material:material:1.2.0'
    implementation 'androidx.room:room-common:2.5.0'
    implementation 'androidx.room:room-ktx:2.5.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_ui_version"
}

```

```
debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"
}
```

## **settings.gradle**

```
pluginManagement
{
    repositories {
        google()
        mavenCentral()
        gradlePluginPortal()
    }
}
dependencyResolutionManagement {
    repositoriesMode.set(RepositoriesMode.FAIL_ON_PROJECT_REPOS)
    repositories {
        google()
        mavenCentral()
    }
}
rootProject.name = "Email Application"
include ':app'
```