

Flexible email client app

SUBMITTED BY:

Vanniya Raja K

Vel Prakash B

Rukumani T

Nambi Rajan B

Introduction

Overview:

- ❖ The app simulates a messaging interface, allowing the user to send and receive messages, and view a history of previous messages.
- ❖ A email client is a software program that you download and install on your computer to send and receive emails directly on your desktop.
- ❖ Microsoft continually updates the program to make

sure outlook operates at a high level.

- ❖ Users register into the application.

Purpose:

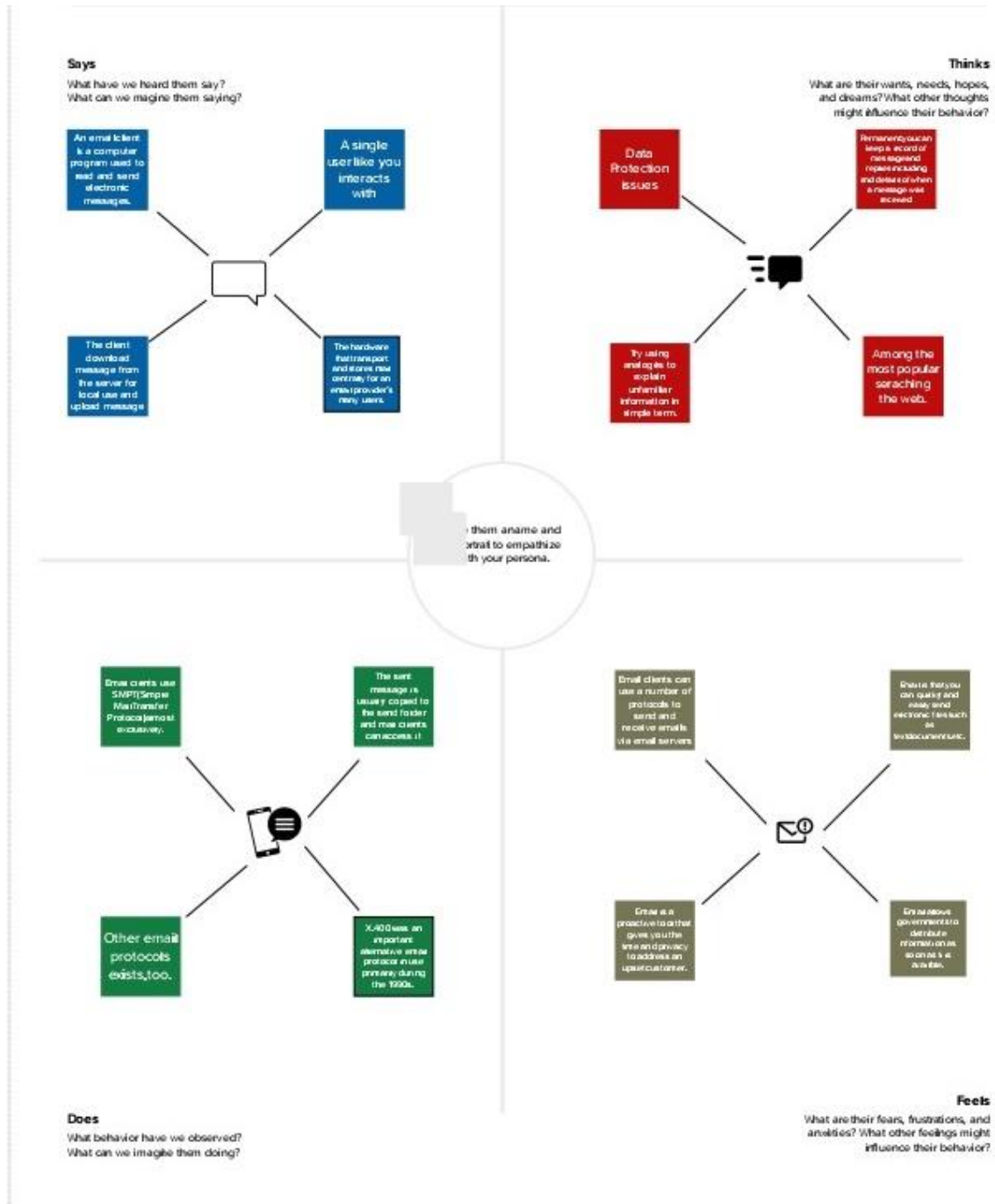
- ❖ An email client is a program that lives on your computer and lets you send or receive emails.
- ❖ An email client is a desktop program that allows users to access their email on their

computer without having to log in via the web.

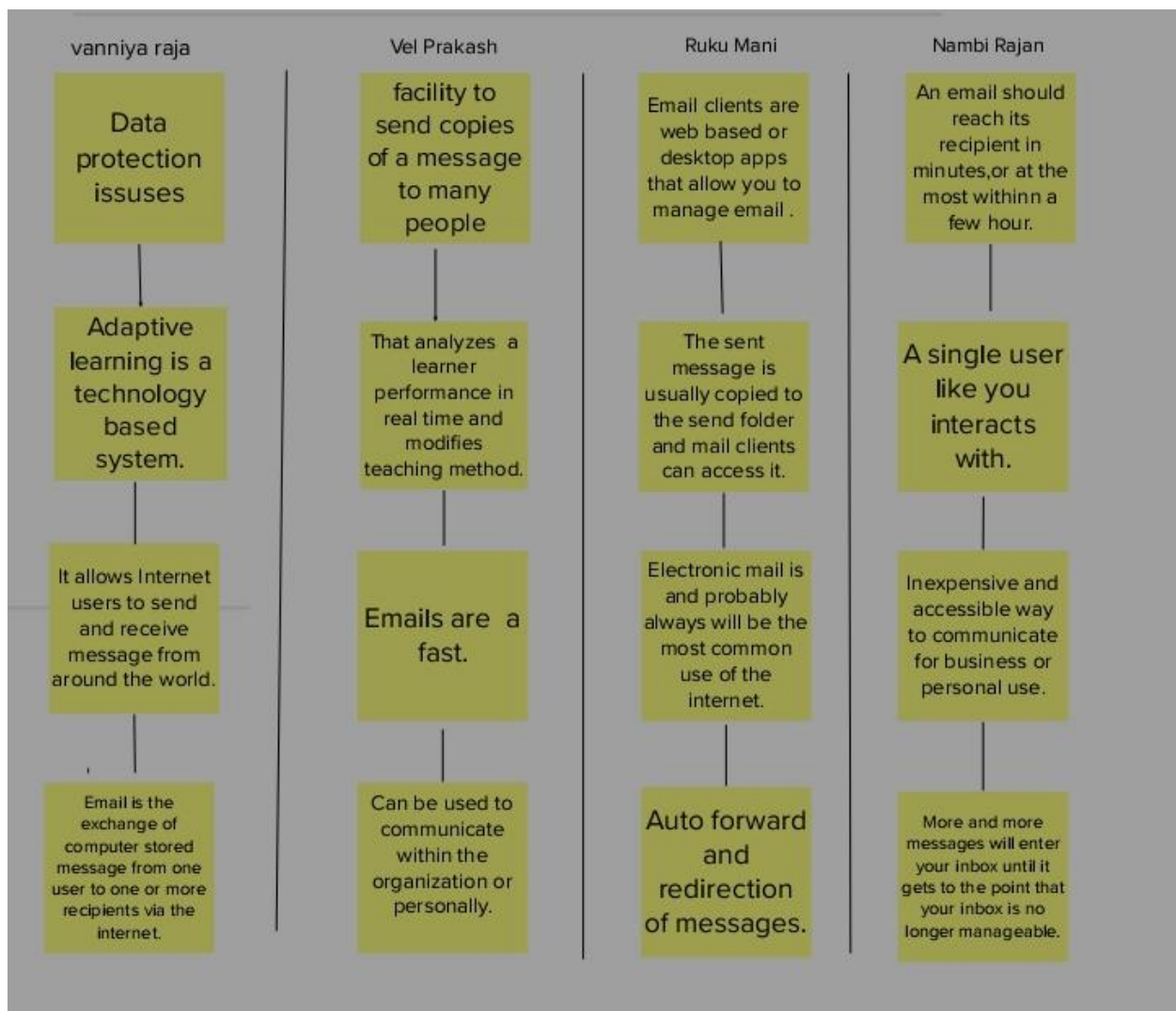
- ❖ Email client is a desktop application that enables configuring one or more email addresses to receive, read, compose and send emails from that email address through the desktop interface.

Problem Definition & Design Thinking

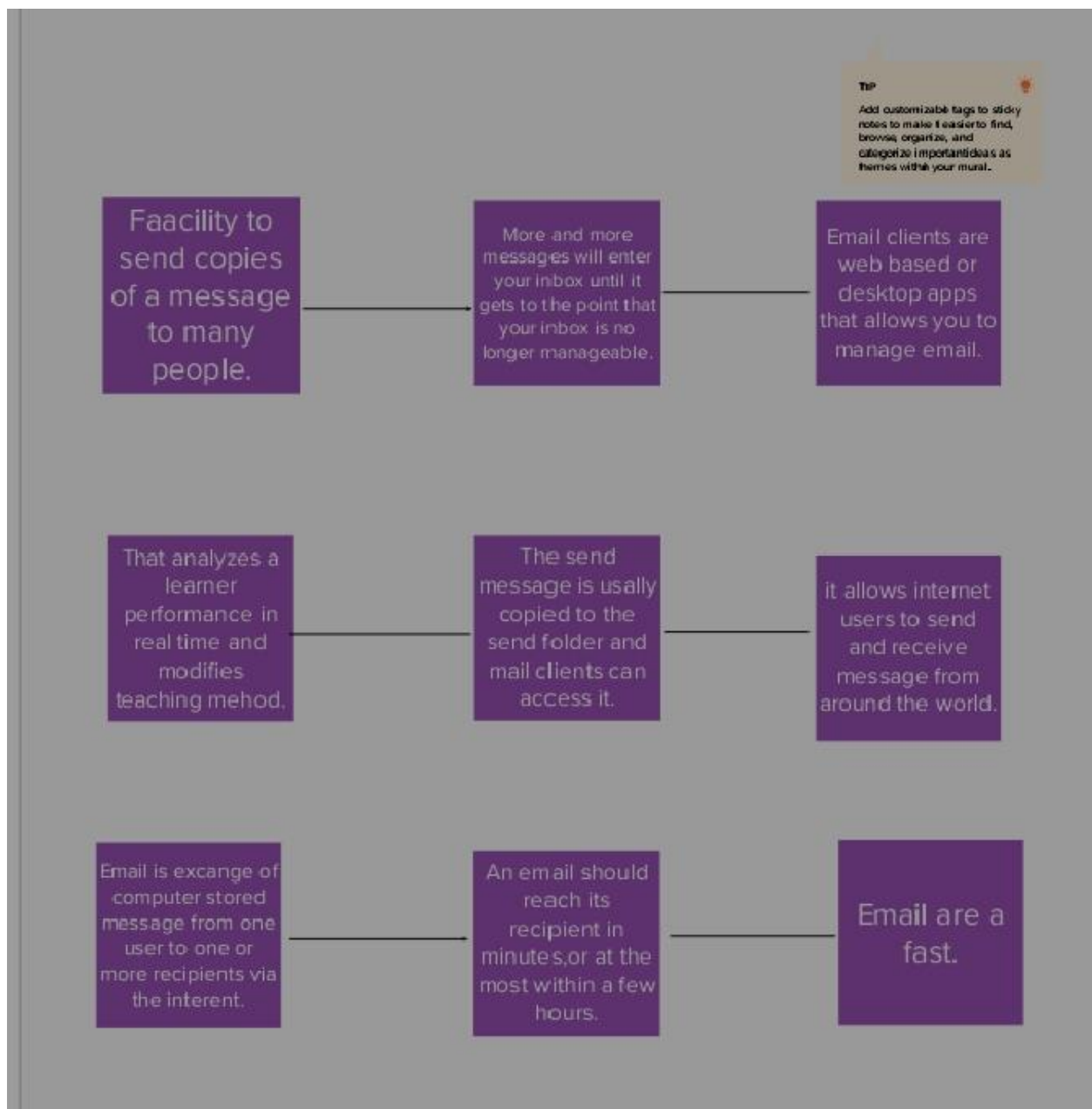
Empathy map:

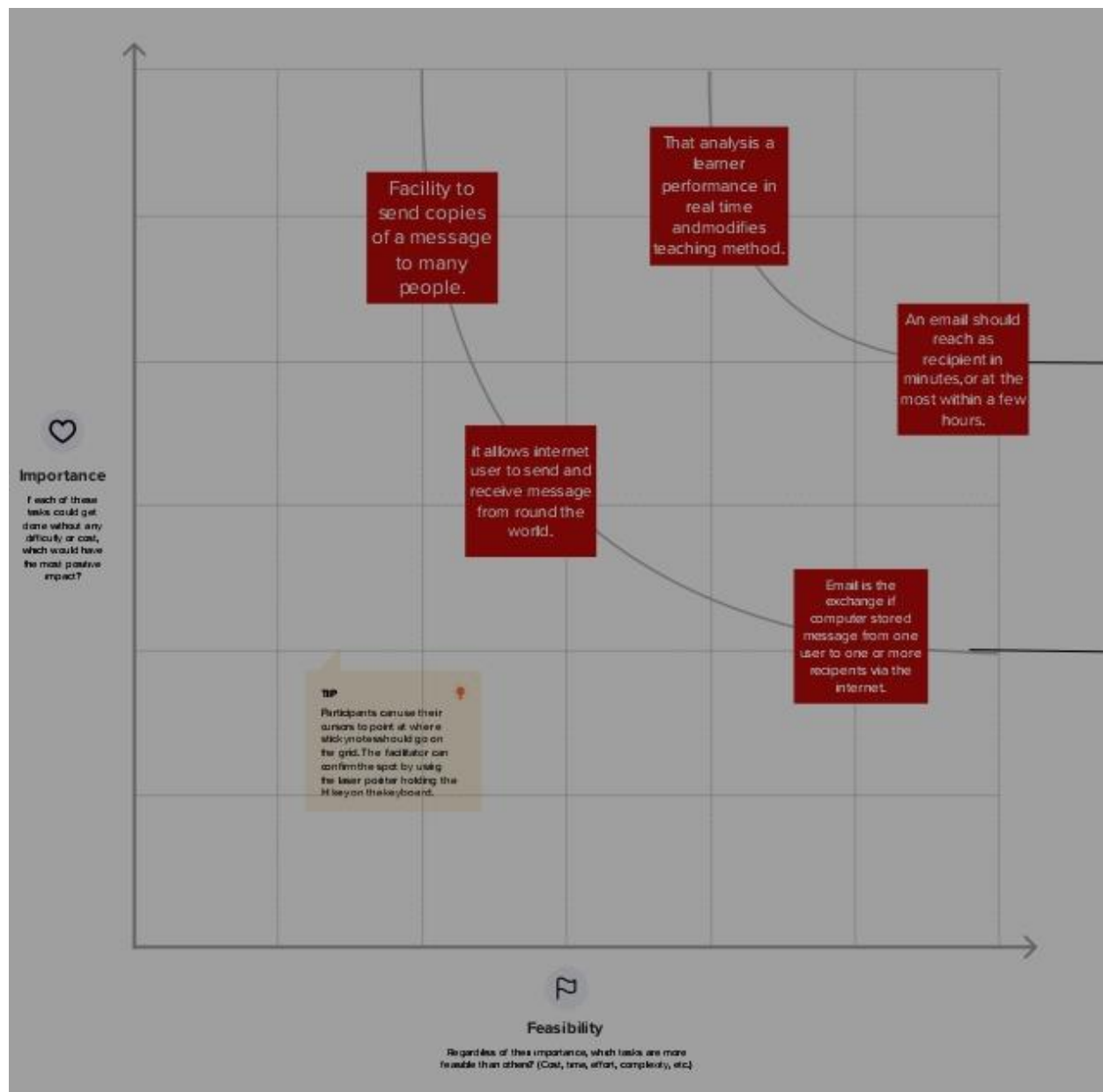


Ideation and Brainstorming:



Important ideas





Result

Home Screen



Send Email

View Emails

Send Mail

Receiver Email-Id

Email address

Mail Subject

Subject

Mail Body

Body

Send Email

View Mails

Receiver_Mail: bnambirajan39@gmail.com
Subject: hi
Body: hi

Receiver_Mail: bnambirajan39@gmail.com
Subject: hi
Body: hi

Receiver_Mail: bnambirajan39@gmail.com
Subject: hi
Body: hi

Receiver_Mail: velprakashbaga@gmail.com
Subject: seven stars
Body: naangathan

Receiver_Mail: velprakashbaga@gmail.com
Subject: seven stars
Body: naangathan

Receiver_Mail: velprakashbaga@gmail.com
Subject: seven stars
Body: naangathan

Receiver_Mail: rajanethaji25@gmail.com
Subject: 7 Star's
Body: hi

Receiver_Mail: rukumani136@gmail.com
Subject: 7 Star's
Body: Good Afternoon

Receiver_Mail: rukumani136@gmail.com
Subject: 7 Star's
Body: Good Afternoon

View Mails

Receiver_Mail: bnambirajan39@gmail.com
Subject: hi
Body: hi

Receiver_Mail: velprakashbaga@gmail.com
Subject: seven stars
Body: naangathan

Receiver_Mail: velprakashbaga@gmail.com
Subject: seven stars
Body: naangathan

Receiver_Mail: velprakashbaga@gmail.com
Subject: seven stars
Body: naangathan

Receiver_Mail: rajanethaji25@gmail.com
Subject: 7 Star's
Body: hi

Receiver_Mail: rukumani136@gmail.com
Subject: 7 Star's
Body: Good Afternoon

Receiver_Mail: rukumani136@gmail.com
Subject: 7 Star's
Body: Good Afternoon

Receiver_Mail: rajanethaji25@gmail.com
Subject: hello
Body: sending msg

Receiver_Mail: rajanethaji25@gmail.com
Subject: hello
Body: sending msg

Advantage:

We can send and receive email message and save them on our computer.

They are delivered at once around the world. No other form of return communication is as fast as an email.

The language used in emails is simple and informal and hence easy to compose.

Emails do not use paper.
They are environment friendly
and save a lot trees forms begin
cut down.

Disadvantage:

Emails may carry virus.

These are the small programs that harm our computer system.

They can read out your email address book and send them self to a number of people around the world.

Many people send unwanted email to other. These are called spam email.

Your mail box may get flooded with emails after a certain time show you have to empty it's from to time to time.

Application:

Emails are most commonly used for professional communications, as they have made formal communications much more accessible and simplified.

Without using mails, communication inside and outside organizations will become quite challenging, as

a secure medium for data transmission is required for such communications.

Conclusion:

It provides closure for the reader while reminding the reader of the contents and importance of the paper.

“Expressing gratitude and kind regards is always a good idea!” Professional email sign-off examples include:
Sincerely.

Future scope:

In 2020 approximately 306 billion e-mails were sending and receive everyday worldwide.

In 2021, about 319.6 billion emails were sending and received daily that 4.1% more emails sent than the previous year. The number of email users globally is only expected to grow in the next four years.

The figure is projected
increase to over 375 billion daily
emails in 2028.

Appendix

Source code:

Androidmanifest.xml:

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

```
<manifest  
xmlns:android="http://sche  
mas.android.com/apk/res/a  
ndroid"
```

```
xmlns:tools="http://schemas  
.android.com/tools" >
```

<application

android:allowBackup="true"

**android:dataExtractionRules
="@xml/data_extraction_rule
s"**

**android:fullBackupContent="
@xml/backup_rules"**

**android:icon="@drawable/e
mail"**

**android:label="@string/app_
name"**

android:supportsRtl="true"

**android:theme="@style/The
me.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/The
me.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/The
me.EmailApplication" >**

<intent-filter>

**<action
android:name="android.inten
t.action.MAIN" />**

**<category
android:name="android.inten
t.category.LAUNCHER" />**

</intent-filter>

</activity>

</application>

</manifest>

Mainactivity.kt:

package

**com.example.emailapplicatio
n**

import

android.content.Context

import android.content.Intent

import android.os.Bundle

import

androidx.activity.Component

Activity

```
import  
androidx.activity.compose.set  
Content
```

```
import  
androidx.compose.foundation  
Image
```

```
import  
androidx.compose.foundation  
background
```

```
import  
androidx.compose.foundation  
layout.*
```



```
import  
androidx.compose.material.*
```

```
import  
androidx.compose.runtime.C  
omposable
```

```
import  
androidx.compose.ui.Alignme  
nt
```

```
import  
androidx.compose.ui.Modifie  
r
```

```
import  
androidx.compose.ui.graphics  
Color
```

```
import  
androidx.compose.ui.layout.C  
ontentScale
```

```
import  
androidx.compose.ui.res.pain  
terResource
```

```
import  
androidx.compose.ui.text.fon  
t.FontWeight
```

```
import  
androidx.compose.ui.tooling.  
preview.Preview
```

```
import  
androidx.compose.ui.unit.dp
```

```
import  
androidx.compose.ui.unit.sp
```

```
import  
androidx.core.content.Context  
Compat
```

```
import  
androidx.core.content.Context  
Compat.startActivity
```

```
import  
com.example.emailapplicatio  
n.ui.theme.EmailApplicationT  
heme
```

```
class MainActivity :  
ComponentActivity() {  
    override fun  
onCreate(savedInstanceState  
e: Bundle?) {  
  
    super.onCreate(savedInstanceState)
```

```
setContent {  
    // A surface container  
    using the 'background' color  
    from the theme  
    Surface(  
        modifier =  
        Modifier.fillMaxSize().backgr  
        ound(Color.White),  
    ) {  
        Email(this)  
    }  
}
```

```
    }  
  }  
}
```

@Composable

fun Email(context: Context) {

Text(

text = "Home Screen",

modifier =

Modifier.padding(top =

**74.dp, start = 100.dp, bottom
= 24.dp),**

```
color = Color.Black,  
fontWeight =  
FontWeight.Bold,  
fontSize = 32.sp  
)
```

```
Column(  
    horizontalAlignment =  
Alignment.CenterHorizontally  
,  
    verticalArrangement =  
Arrangement.Center
```

```
) {  
    Image(  
        painterResource(id =  
R.drawable.home_screen),  
        contentDescription = ""  
    )
```

```
Button(onClick = {  
    context.startActivity(  
        Intent(  
            context,
```


SendMailActivity::class.java

)

)

},

colors =

ButtonDefaults.buttonColors(

backgroundColor =

Color(0xFFadbef4))

) {

Text(

text = "Send Email",

```
modifier =  
Modifier.padding(10.dp),  
color = Color.Black,  
fontSize = 15.sp  
)  
}
```

```
Spacer(modifier =  
Modifier.height(20.dp))
```

```
Button(onClick = {  
context.startActivity(
```

```
Intent(  
    context,
```

```
ViewMailActivity::class.java
```

```
)
```

```
)
```

```
},
```

```
colors =
```

```
ButtonDefaults.buttonColors(
```

```
backgroundColor =
```

```
Color(0xFFadbef4))
```

```
) {
```

```
Text(  
    text = "View  
Emails",  
    modifier =  
Modifier.padding(10.dp),  
    color = Color.Black,  
    fontSize = 15.sp  
)  
}  
  
}
```

}

Registeractivity.kt:

package

**com.example.emailapplicatio
n**

import

android.content.Context

import android.content.Intent

import android.os.Bundle

```
import  
androidx.activity.Component  
Activity
```

```
import  
androidx.activity.compose.se  
tContent
```

```
import  
androidx.compose.foundatio  
n.Image
```

```
import  
androidx.compose.foundatio  
n.background
```

```
import  
androidx.compose.foundation  
layout.*
```

```
import  
androidx.compose.material.*
```

```
import  
androidx.compose.runtime.*
```

```
import  
androidx.compose.ui.Alignme  
nt
```

```
import  
androidx.compose.ui.Modifie  
r
```

```
import  
androidx.compose.ui.graphics  
Color
```

```
import  
androidx.compose.ui.layout.C  
ontentScale
```

```
import  
androidx.compose.ui.res.pain  
terResource
```

```
import  
androidx.compose.ui.text.fon  
t.FontFamily
```



```
import  
androidx.compose.ui.text.fon  
t.FontWeight
```

```
import  
androidx.compose.ui.text.inp  
ut.PasswordVisualTransforma  
tion
```

```
import  
androidx.compose.ui.tooling.  
preview.Preview
```

```
import  
androidx.compose.ui.unit.dp
```

```
import  
androidx.compose.ui.unit.sp  
  
import  
androidx.core.content.Context  
tCompat  
  
import  
com.example.emailapplication  
n.ui.theme.EmailApplicationT  
heme
```

```
class RegisterActivity :  
ComponentActivity() {
```

```
private lateinit var  
databaseHelper:  
UserDatabaseHelper  
  
override fun  
onCreate(savedInstanceState:  
Bundle?) {  
  
    super.onCreate(savedInstanceState)  
  
    databaseHelper =  
    UserDatabaseHelper(this)  
  
    setContent {
```

```
RegistrationScreen(this,  
databaseHelper)
```

```
}
```

```
}
```

```
}
```

```
@Composable
```

```
fun
```

```
RegistrationScreen(context:
```

**Context, databaseHelper:
UserDatabaseHelper) {**

**var username by remember
{ mutableStateOf("") }**

**var password by remember
{ mutableStateOf("") }**

**var email by remember {
mutableStateOf("") }**

```
var error by remember {  
mutableStateOf("") }
```

```
Column(  
    modifier =  
    Modifier.fillMaxSize().backgr  
ound(Color.White),  
    horizontalAlignment =  
    Alignment.CenterHorizontally  
,  
    verticalArrangement =  
    Arrangement.Center
```

```
) {  
    Image(  
        painterResource(id =  
R.drawable.email_signup),  
        contentDescription = "",  
        modifier =  
Modifier.height(300.dp)  
    )  
    Text(  
        fontSize = 36.sp,  
        fontWeight =  
FontWeight.ExtraBold,
```

```
fontFamily =  
FontFamily.Cursive,  
text = "Register"  
)
```

```
Spacer(modifier =  
Modifier.height(10.dp))
```

```
TextField(  
value = username,  
onValueChange = {  
username = it },
```



```
label = {  
Text("Username") },  
modifier = Modifier  
    .padding(10.dp)  
    .width(280.dp)  
}
```

```
TextField(  
    value = email,  
    onValueChange = {  
email = it },
```

```
label = { Text("Email")  
},  
  
modifier = Modifier  
    .padding(10.dp)  
    .width(280.dp)  
)
```

```
TextField(  
    value = password,  
    onValueChange = {  
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() &&  
email.isNotEmpty()) {  
        val user = User(  
            id = null,  
            firstName =  
username,  
            lastName =  
null,  
            email = email,
```

```
password =  
password  
)
```

```
databaseHelper.insertUser(u  
ser)
```

```
error = "User  
registered successfully"
```

```
// Start  
LoginActivity using the  
current context
```

```
context.startActivity(  
    Intent(  
        context,
```

```
LoginActivity::class.java  
    )  
)
```

```
} else {  
    error = "Please fill  
all fields"
```

```
        }  
    },  
    colors =  
    ButtonDefaults.buttonColors(  
        backgroundColor =  
        Color(0xFFd3e5ef),  
        modifier =  
        Modifier.padding(top =  
        16.dp)  
    ) {  
        Text(text = "Register")  
    }  
}
```


**Spacer(modifier =
Modifier.width(10.dp))**

**Spacer(modifier =
Modifier.height(10.dp))**

**Row() {
 Text(
 modifier =
Modifier.padding(top =
14.dp), text = "Have an
account?"
)**

```
TextButton(onClick = {  
  
context.startActivity(  
    Intent(  
        context,  
  
LoginActivity::class.java  
    )  
    )  
})  
  
{
```

```
        Spacer(modifier =  
Modifier.width(10.dp))
```

```
        Text(color =  
Color(0xFF31539a),text =  
"Log in")
```

```
    }
```

```
 }
```

```
 }
```

```
}
```

```
private fun  
startLoginActivity(context:  
Context) {
```

```
val intent = Intent(context,  
LoginActivity::class.java)
```

```
ContextCompat.startActivity(  
context, intent, null)  
}
```

LoginActivity.kt:

```
package  
com.example.emailapplicatio  
n
```

import

android.content.Context

import android.content.Intent

import android.os.Bundle

import

androidx.activity.Component

Activity

import

androidx.activity.compose.se

tContent

```
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.Alignme  
nt
```

```
import  
androidx.compose.ui.Modifie  
r
```

```
import  
androidx.compose.ui.graphic  
s.Color
```

```
import  
androidx.compose.ui.layout.C  
ontentScale
```

```
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.Context  
Compat
```

```
import  
com.example.emailapplicatio
```

**n.ui.theme.EmailApplicationT
heme**

```
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().backgr  
ound(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,
onValueChange = {
username = it },
label = {
Text("Username") },**

```
modifier =  
Modifier.padding(10.dp)  
    .width(280.dp)  
)
```

```
TextField(  
    value = password,  
    onValueChange = {  
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)
```


}