Compose Input: A Demonstration of Text Input and Validation with Android Compose

TEAM ID:NM2024TMID04485

DEPARTMENT OF INFORMATION TECHNOLOGY

V.S.B.COLLEGE OF ENGINEERING TECHNICAL CAMPUS,COIMBATORE

Description

"Compose Input: A Demonstration of Text Input and Validation with Android Compose" refers to showcasing how to implement and manage user input fields using **Jetpack Compose**, Google's modern toolkit for building native UI in Android applications. This topic typically includes demonstrating how to:

1. Create Text Input Fields

•Use the TextField or OutlinedTextField components to allow users to enter text.
•Customize these fields with labels, placeholders, and icons.

2. Handle State and Input Changes

•Utilize remember and mutableStateOf to manage the state of input fields.
•Update UI dynamically as the user types.

3. Implement Validation

•Validate user inputs such as email formats, passwords, or other criteria.
•Display error messages or feedback directly below the input field if validation fails.

4. Enhance Usability

•Add features like:
•Input masking (e.g., phone numbers or credit card formatting).
•Keyboard actions to move between fields or submit forms.
•Clear buttons or focus management for user convenience.

5. Styling and Customization

•Apply custom themes and styles to input fields.
•Adjust colors, fonts, and dimensions to fit the app’s design.

AdminActivity.kt

```
package com.example.surveyapplication

import android.os.Bundle
import android.util.Log
import androidx.activity.ComponentActivity
import androidx.activity.compose.setContent
import androidx.compose.foundation.Image
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.lazy.LazyColumn
import androidx.compose.foundation.lazy.LazyRow
import androidx.compose.foundation.lazy.items
import androidx.compose.material.MaterialTheme
import androidx.compose.material.Surface
import androidx.compose.material.Text
import androidx.compose.runtime.Composable
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.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.surveyapplication.ui.theme.SurveyApplicationTheme
class AdminActivity : ComponentActivity() {
    private lateinit var databaseHelper: SurveyDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = SurveyDatabaseHelper(this)
        setContent {
            val data = databaseHelper.getAllSurveys();
            Log.d("swathi", data.toString())
            val survey = databaseHelper.getAllSurveys()
            ListListScopeSample(survey)
        }   }   }
```

```
@Composable
fun ListListScopeSample(survey: List<Survey>) {

    Image(
        painterResource(id = R.drawable.background), contentDescription = "",
        alpha =0.1F,
        contentScale = ContentScale.FillHeight,
        modifier = Modifier.padding(top = 40.dp) )

    Text(
        text = "Survey Details",
        modifier = Modifier.padding(top = 24.dp, start = 106.dp, bottom = 24.dp),
        fontSize = 30.sp,
        color = Color(0xFF25b897) )
    Spacer(modifier = Modifier.height(30.dp))
    LazyRow(
        modifier = Modifier
            .fillMaxSize()
            .padding(top = 80.dp),
    horizontalArrangement = Arrangement.SpaceBetween  ) {
        item {
          LazyColumn {
             items(survey) { survey ->
                    Column(
                        modifier = Modifier.padding(
                            top = 16.dp,
                            start = 48.dp,
                            bottom = 20.dp
                        )
                    ) {
                        Text("Name: ${survey.name}")
                        Text("Age: ${survey.age}")
                        Text("Mobile_Number: ${survey.mobileNumber}")
                        Text("Gender: ${survey.gender}")
                        Text("Diabetics: ${survey.diabetics}")
                    }          }          }    } }     }
```

LoginActivity.kt

```
package com.example.surveyapplication

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.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
import com.example.surveyapplication.ui.theme.SurveyApplicationTheme

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.survey_login), contentDescription = "")
 Text(
      fontSize = 36.sp,
      fontWeight = FontWeight.ExtraBold,
      fontFamily = FontFamily.Cursive,
      color = Color(0xFF25b897),
      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()          }
                    if (user != null && user.password == "admin") {
                        error = "Successfully log in"
                        context.startActivity(
                            Intent(
                                context,
                                AdminActivity::class.java                )         )            }
                    else {
                        error =  "Invalid username or password            }           } else {
                    error = "Please fill all fields          }       },
            colors = ButtonDefaults.buttonColors(backgroundColor = Color(0xFF84adb8)),
            modifier = Modifier.padding(top = 16.dp)     ) {
            Text(text = "Login")      }
        Row {
            TextButton(onClick = {context.startActivity(
                Intent(
                    context,
                    RegisterActivity::class.java          )        )}        )
            { Text(color = Color(0xFF25b897),text = "Register") }
            TextButton(onClick = {       })       {
                Spacer(modifier = Modifier.width(60.dp))
                Text(color = Color(0xFF25b897),text = "Forget password?")         }      }   }}
private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```

MainActivity.kt

```
package com.example.surveyapplication

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.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.style.TextAlign
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.surveyapplication.ui.theme.SurveyApplicationTheme

class MainActivity : ComponentActivity() {
    private lateinit var databaseHelper: SurveyDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = SurveyDatabaseHelper(this)
        setContent {
            FormScreen(this, databaseHelper)
        }
    }
}
```

```
@Composable
fun FormScreen(context: Context, databaseHelper: SurveyDatabaseHelper) {

    Image(
        painterResource(id = R.drawable.background), contentDescription = "",
        alpha =0.1F,
        contentScale = ContentScale.FillHeight,
        modifier = Modifier.padding(top = 40.dp)
        )




    // Define state for form fields
    var name by remember { mutableStateOf("") }
    var age by remember { mutableStateOf("") }
    var mobileNumber by remember { mutableStateOf("") }
    var genderOptions = listOf("Male", "Female", "Other")
    var selectedGender by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }
    var diabeticsOptions = listOf("Diabetic", "Not Diabetic")
    var selectedDiabetics by remember { mutableStateOf("") }

    Column(
        modifier = Modifier.padding(24.dp),
        horizontalAlignment = Alignment.Start,
        verticalArrangement = Arrangement.SpaceEvenly
    ) {

        Text(
            fontSize = 36.sp,
            textAlign = TextAlign.Center,
            text = "Survey on Diabetics",
            color = Color(0xFF25b897)
        )
```

```
Spacer(modifier = Modifier.height(24.dp))
   Text(text = "Name :", fontSize = 20.sp)
     TextField(
        value = name,
        onValueChange = { name = it },
     )
  Spacer(modifier = Modifier.height(14.dp))
     Text(text = "Age :", fontSize = 20.sp)
     TextField(
        value = age,
        onValueChange = { age = it },
     )
     Spacer(modifier = Modifier.height(14.dp))
     Text(text = "Mobile Number :", fontSize = 20.sp)
     TextField(
        value = mobileNumber,
        onValueChange = { mobileNumber = it },
     )
     Spacer(modifier = Modifier.height(14.dp))
     Text(text = "Gender :", fontSize = 20.sp)
     RadioGroup(
        options = genderOptions,
        selectedOption = selectedGender,
        onSelectedChange = { selectedGender = it }
     )
     Spacer(modifier = Modifier.height(14.dp))

     Text(text = "Diabetics :", fontSize = 20.sp)
     RadioGroup(
        options = diabeticsOptions,
        selectedOption = selectedDiabetics,
        onSelectedChange = { selectedDiabetics = it }        )
     Text(
        text = error,
        textAlign = TextAlign.Center,
        modifier = Modifier.padding(bottom = 16.dp)
     )
```

```
// Display Submit button
      Button(
          onClick = {  if (name.isNotEmpty() && age.isNotEmpty() && mobileNumber.isNotEmpty() && genderOptions.isNotEmpty() && diabeticsOptions.isNotEmpty()) {
              val survey = Survey(
                  id = null,
                  name = name,
                  age = age,
                  mobileNumber = mobileNumber,
                  gender = selectedGender,
                  diabetics = selectedDiabetics                  )
              databaseHelper.insertSurvey(survey)
              error = "Survey Completed"
          } else {
              error = "Please fill all fields"           }          },
          colors = ButtonDefaults.buttonColors(backgroundColor = Color(0xFF84adb8)),
          modifier = Modifier.padding(start = 70.dp).size(height = 60.dp, width = 200.dp)       ) {
          Text(text = "Submit")       }    } }
@Composable
fun RadioGroup(
    options: List<String>,
    selectedOption: String?,
    onSelectedChange: (String) -> Unit) {
    Column {
        options.forEach { option ->
            Row(
                Modifier
                    .fillMaxWidth()
                    .padding(horizontal = 5.dp)          ) {
                RadioButton(
                    selected = option == selectedOption,
                    onClick = { onSelectedChange(option) }              )
                Text(
                    text = option,
                    style = MaterialTheme.typography.body1.merge(),
                    modifier = Modifier.padding(top = 10.dp),
                    fontSize = 17.sp              )        }     }  }}
```

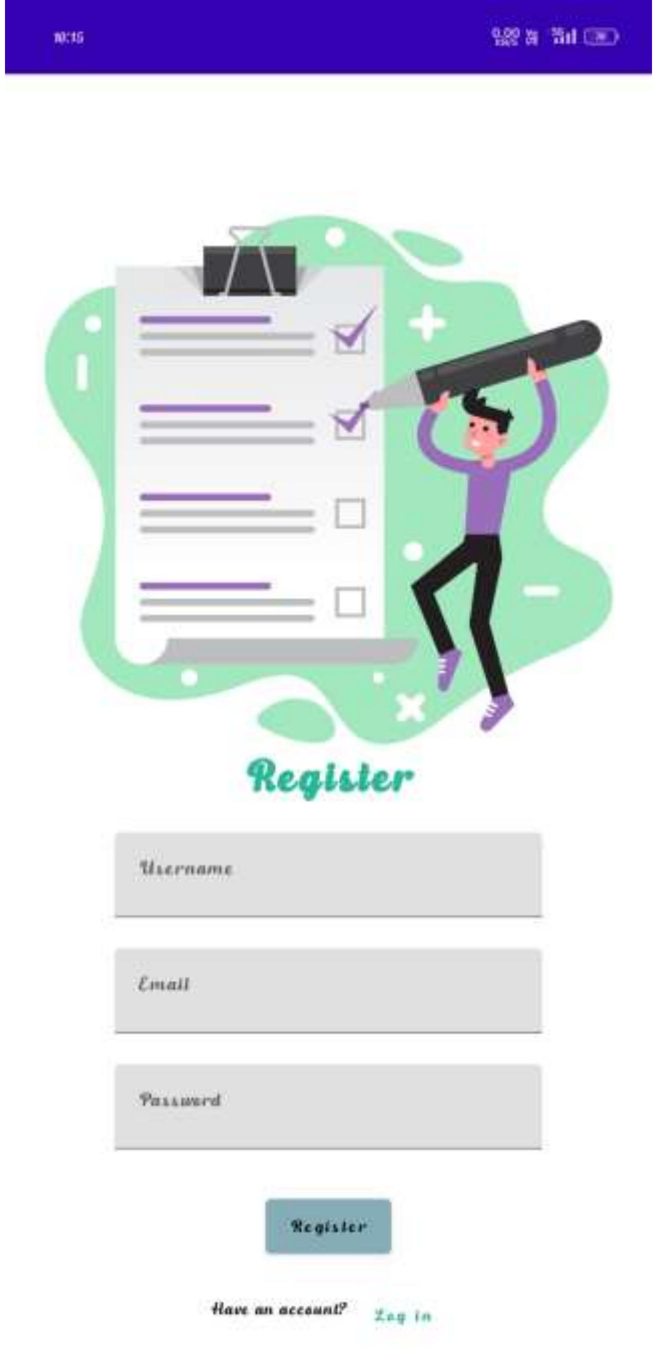
RegisterActivity.kt

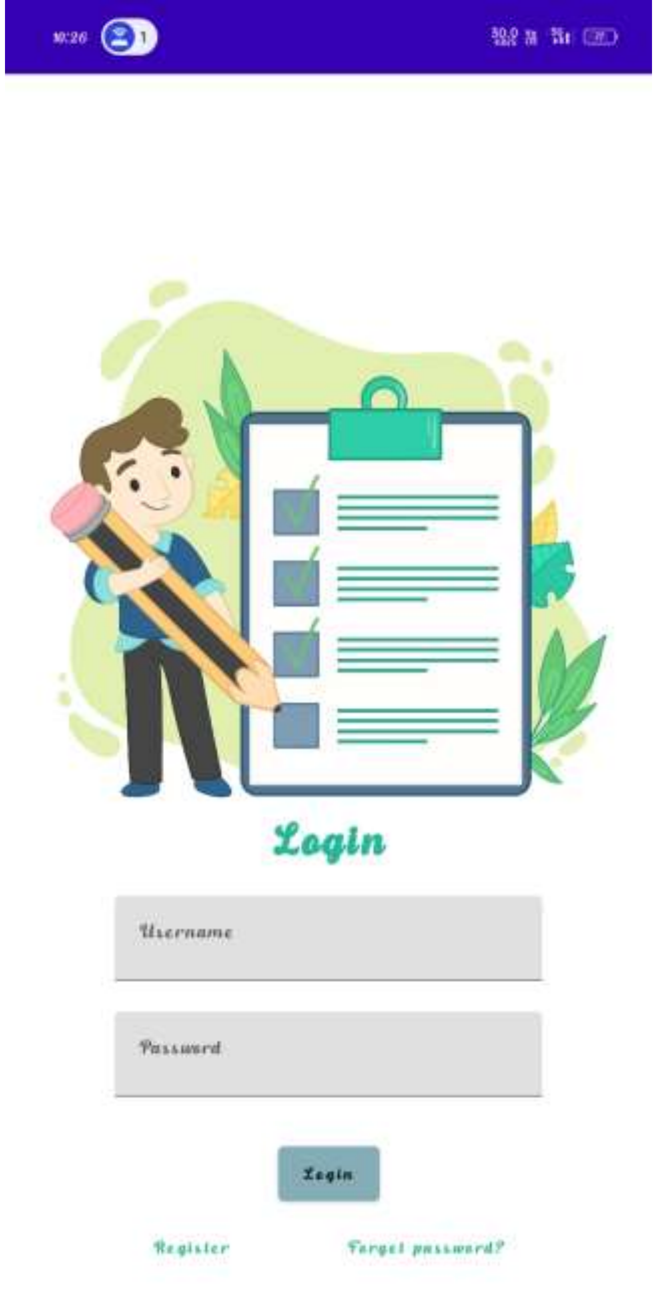
```
package com.example.surveyapplication

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.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import androidx.core.content.ContextCompat
import com.example.surveyapplication.ui.theme.SurveyApplicationTheme

class RegisterActivity : ComponentActivity() {
    private lateinit var databaseHelper: UserDatabaseHelper
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        databaseHelper = UserDatabaseHelper(this)
        setContent {
                RegistrationScreen(this,databaseHelper)
        }
    }
}
```

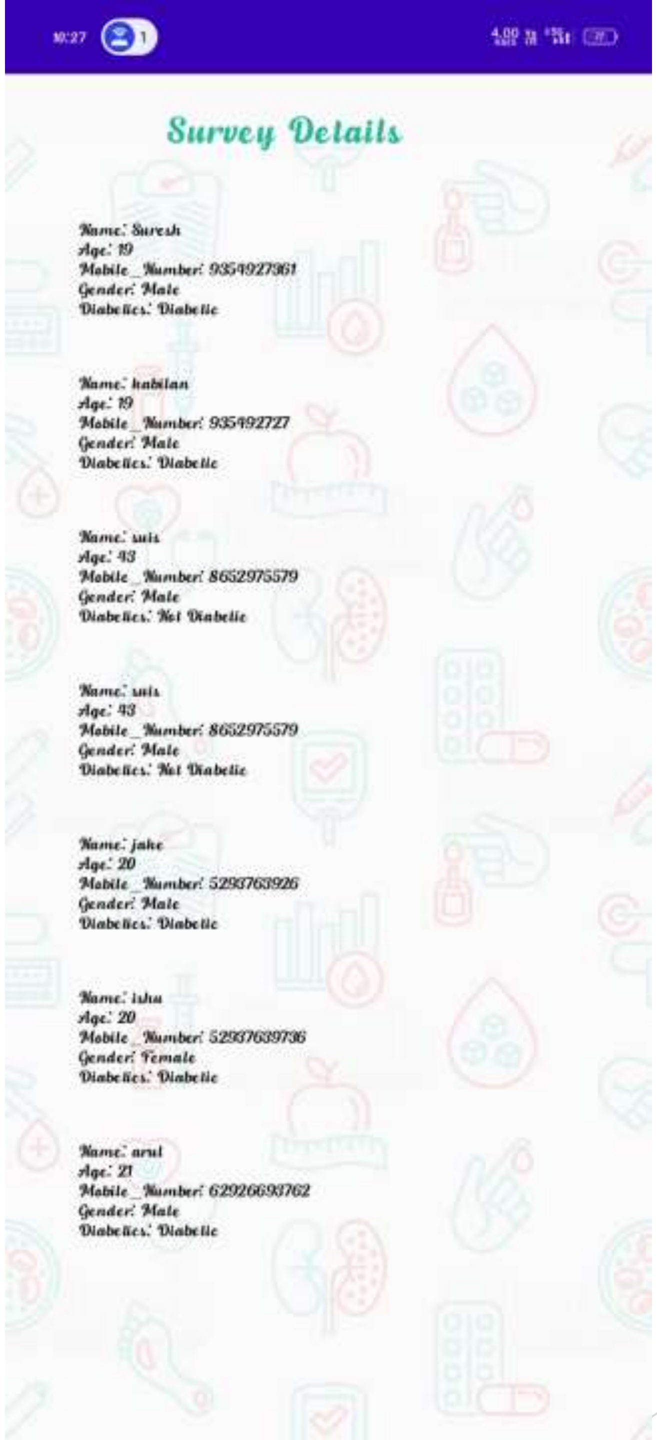
```
@Composable
fun RegistrationScreen(context: Context, databaseHelper: UserDatabaseHelper) {

    var username by remember { mutableStateOf("") }
    var password by remember { mutableStateOf("") }
    var email by remember { mutableStateOf("") }
    var error by remember { mutableStateOf("") }
    Column(
        modifier = Modifier.fillMaxSize().background(Color.White),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) {
        Image(painterResource(id = R.drawable.survey_signup), contentDescription = "")
        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            color = Color(0xFF25b897),
            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(user)
                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(0xFF84adb8)),
        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(0xFF25b897),text = "Log in")
            }
        }
    }
}
private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}
```


10:15
Register
Username
Email
Password
Register
Have an account?
Log in


10:26
Login
Username
Password
Login
Register
Forgot password?



Survey Details

Name: Suresh
Age: 19
Mobile_Number: 9354927361
Gender: Male
Diabetics: Diabetic

Name: kabilan
Age: 19
Mobile_Number: 935492727
Gender: Male
Diabetics: Diabetic

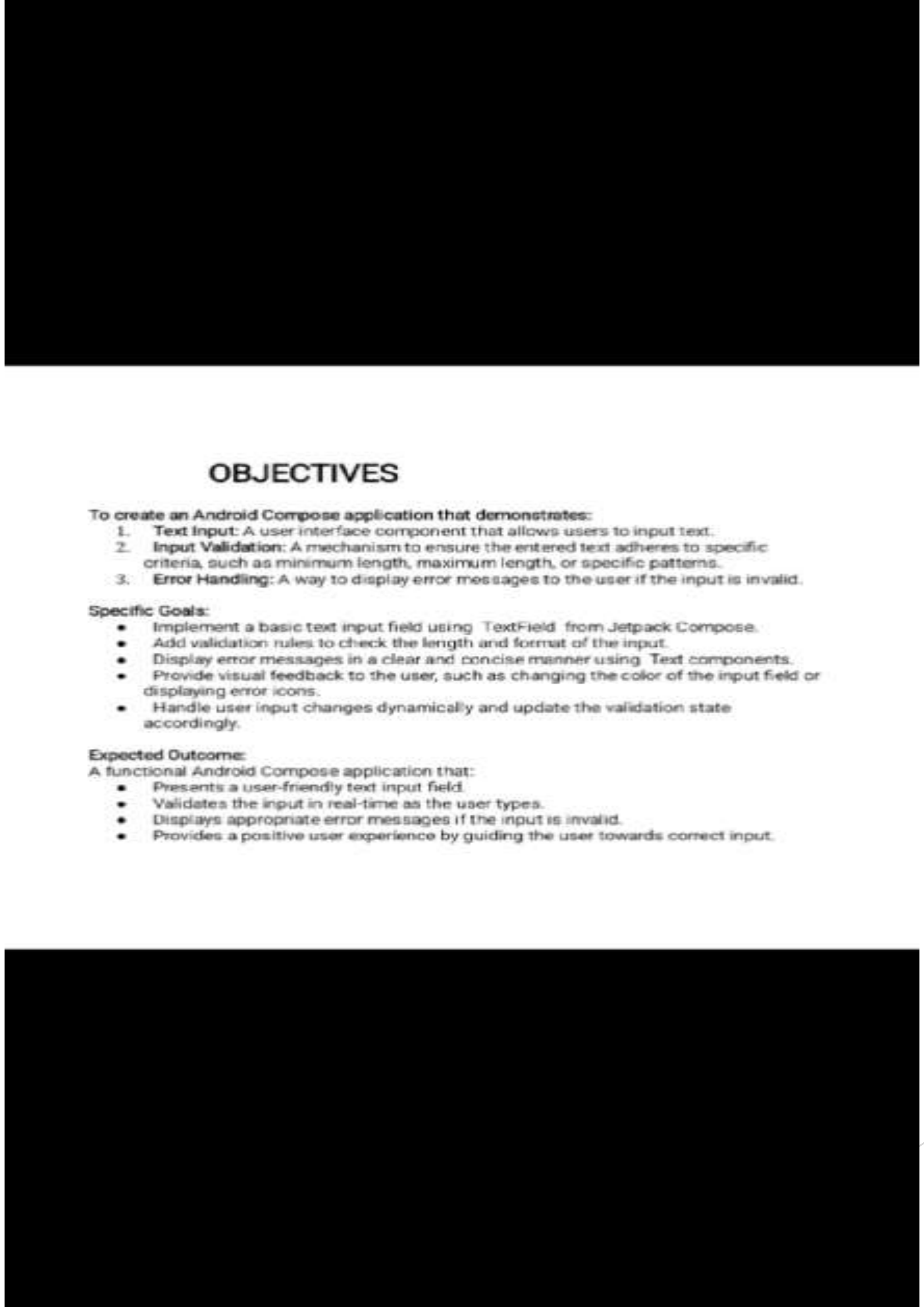
Name: suis
Age: 43
Mobile_Number: 8652975579
Gender: Male
Diabetics: Not Diabetic

Name: suis
Age: 43
Mobile_Number: 8652975579
Gender: Male
Diabetics: Not Diabetic

Name: jake
Age: 20
Mobile_Number: 5293763926
Gender: Male
Diabetics: Diabetic

Name: isha
Age: 20
Mobile_Number: 52937639736
Gender: Female
Diabetics: Diabetic

Name: arul
Age: 21
Mobile_Number: 62926693762
Gender: Male
Diabetics: Diabetic



OBJECTIVES

To create an Android Compose application that demonstrates:

1. **Text Input:** A user interface component that allows users to input text.
2. **Input Validation:** A mechanism to ensure the entered text adheres to specific criteria, such as minimum length, maximum length, or specific patterns.
3. **Error Handling:** A way to display error messages to the user if the input is invalid.

Specific Goals:

- Implement a basic text input field using TextField from Jetpack Compose.
- Add validation rules to check the length and format of the input.
- Display error messages in a clear and concise manner using Text components.
- Provide visual feedback to the user, such as changing the color of the input field or displaying error icons.
- Handle user input changes dynamically and update the validation state accordingly.

Expected Outcome:

A functional Android Compose application that:

- Presents a user-friendly text input field.
- Validates the input in real-time as the user types.
- Displays appropriate error messages if the input is invalid.
- Provides a positive user experience by guiding the user towards correct input.