C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

1.INTRODUCTION

1.1 Overview

The article covers the basics of setting up a Compose project and creating UI elements, such as text fields and buttons, as well as implementing input validation using Kotlin's standard library functions. The article also demonstrates how to handle user input events and display error messages when user input is invalid. Overall, the article is a useful resource for developers looking to get started with Android Compose and learn best practices for text input and validation in mobile app development. creating text fields, validating user input, displaying error messages, and implementing different types of input fields such as text, numbers, and passwords.

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

1.2 PURPOSE

❖ **USES**

- Learning Android Compose
- Developing app with text input
- Building responsive user interfaces
- Ensuring data accuracy
- Improving user experience

Text input and validation are essential components of many Android apps, such as sign-up and login screens, search bars, and forms. With Android Compose, developers can create these components with a declarative and efficient approach, making the development process faster and more straightforward.

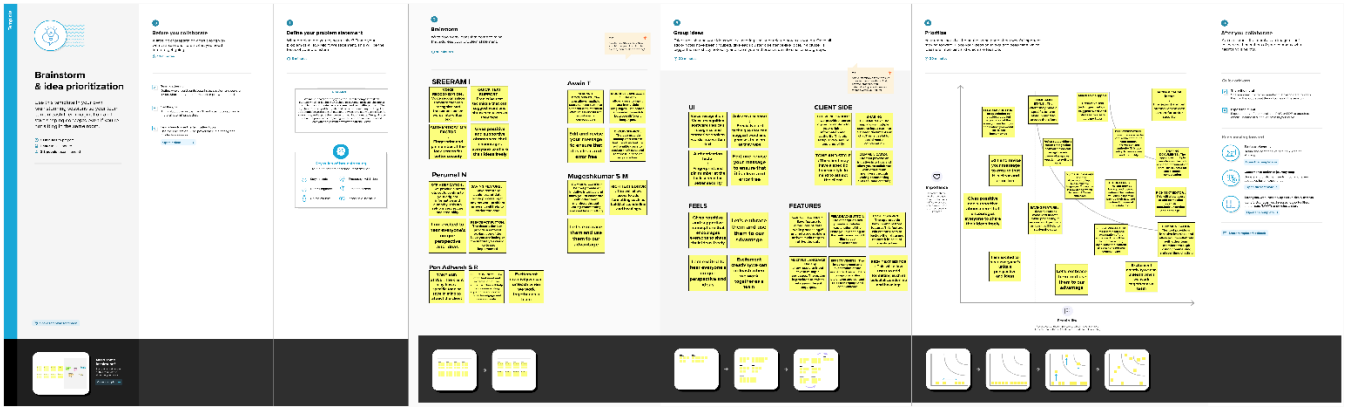
C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

2.PROBLEM DEFINITION AND DESIGN THINKING

2.1 EMPATHY MAP

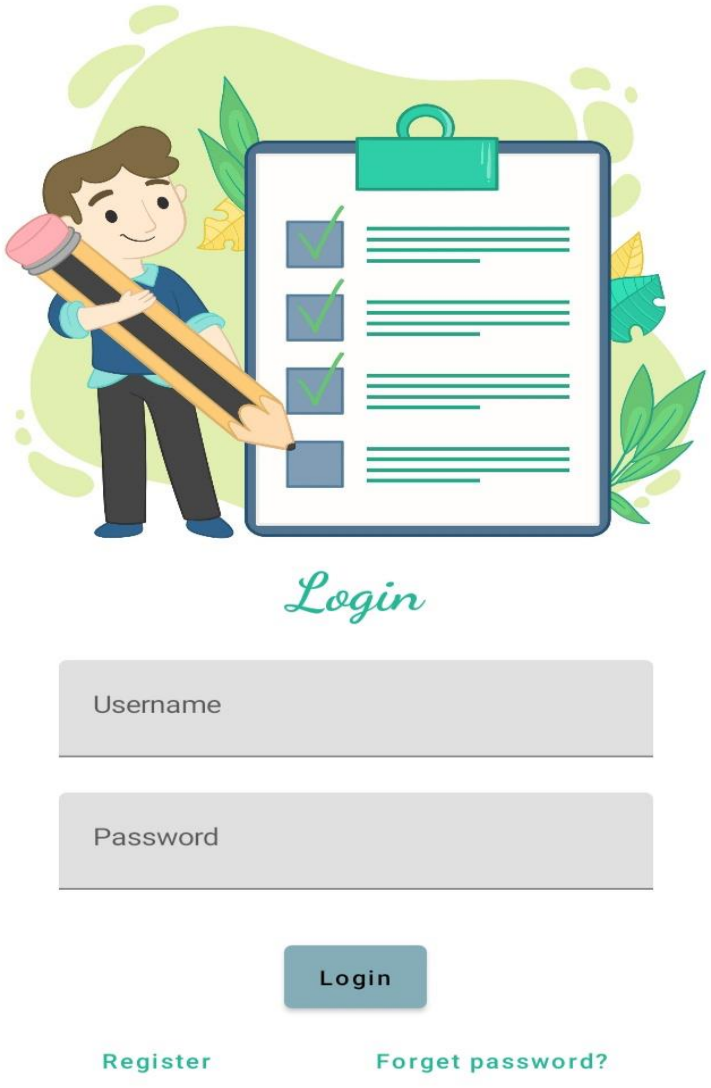
C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

2.2 BRAIN STORMING

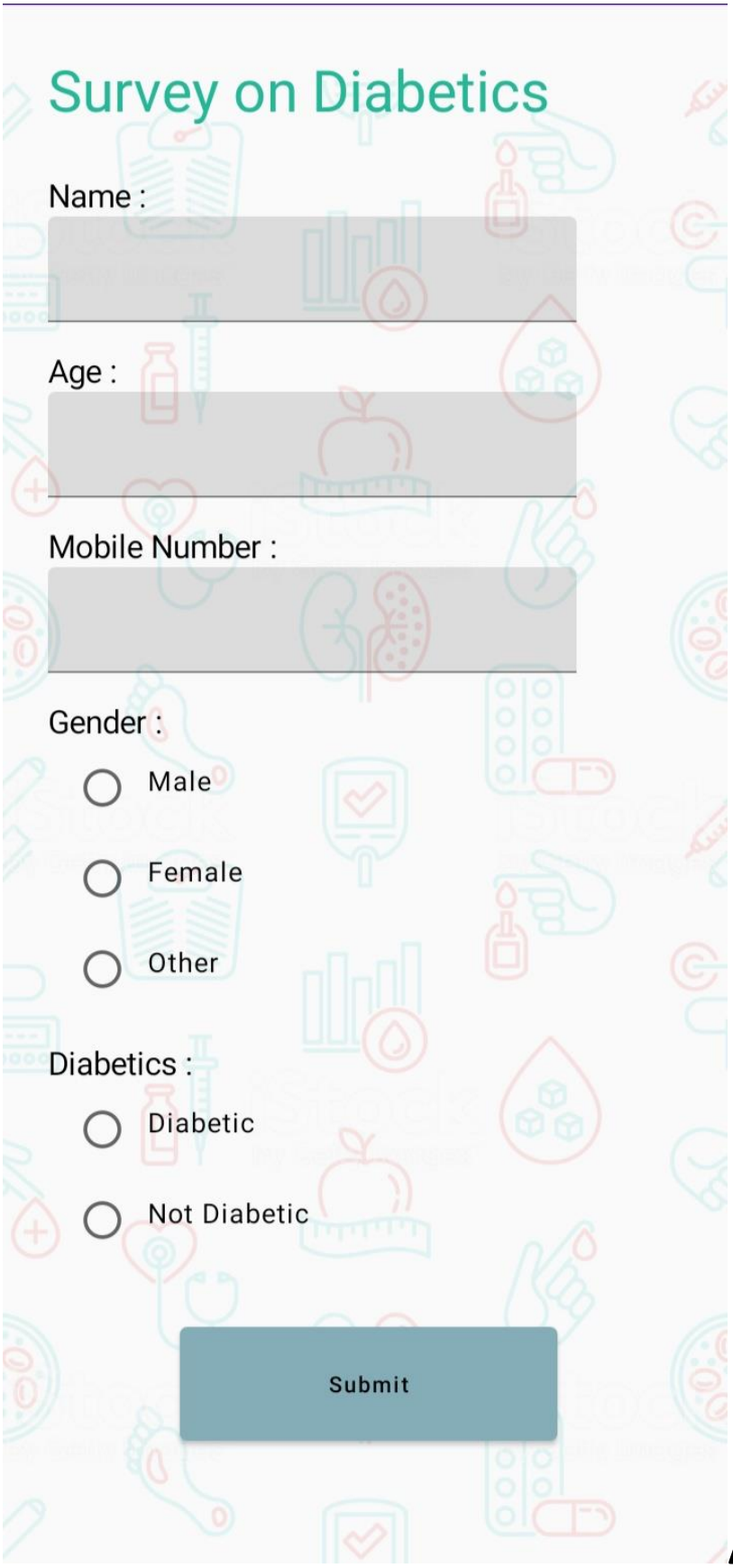


C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

3.RESULT



C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose



ADVANTA

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

4. ADVANTAGE AND DISADVANTAGE

4.1 ADVANTAGES

- Improved User Experience
- Simplified Development
- Increased Accessibility
- Consistent Design
- Improved Performance

4.2 DISADVANTAGES

- Steep Learning Curve
- Limited Compatibility
- Potential for Bugs
- Lack of Third-Party Support
- Limited Community Support

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

5.APPLICATIONS

- **Form-Based Applications:**

Applications that require users to fill out forms, such as registration forms, login forms, or contact forms, can benefit from using Android Compose for text input and validation. Compose makes it easy to create dynamic forms that respond to user input and provide real-time validation feedback.

- **Messaging Applications:**

Messaging applications that allow users to send and receive messages can benefit from using Android Compose for text input and validation.

- **Note-Taking Applications:**

Note-taking applications that allow users to create and edit notes can benefit from using Android Compose for text input and validation. Compose can help ensure that notes are properly formatted and free of errors before they are saved.

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

6.CONCLUSION

In conclusion, Android Compose for text input and validation has its advantages, such as its declarative syntax, real-time preview, and ease of customization. However, there are also potential disadvantages, including a steep learning curve, limited compatibility, potential for bugs, lack of third-party support, and limited community support. Before deciding to use Android Compose for text input and validation, it's important to carefully consider these factors and determine if the benefits outweigh the potential drawbacks for your specific use case. Overall, Android Compose is an exciting new technology that has the potential to simplify and streamline Android app development, and it's worth exploring for developers who are interested in building modern, dynamic user interfaces.

7.FUTURE SCOPE

The future scope of Demonstration of Text Input And Validation with Android Compose is promising, with potential developments such as increased adoption, better compatibility, improved tooling, enhanced capabilities, and a more robust community, making it easier for developers to create apps with powerful text input and validation features.

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

8.APPENDIX

A. SOURCE CODE

```
    <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.SurveyApplication"
        tools:targetApi="31">
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="@string/title_activity_register"
            android:theme="@style/Theme.SurveyApplication" />
        <activity
            android:name=".MainActivity"
            android:exported="false"
            android:label="MainActivity"
            android:theme="@style/Theme.SurveyApplication" />
        <activity
            android:name=".AdminActivity"
            android:exported="false"
            android:label="@string/title_activity_admin"
            android:theme="@style/Theme.SurveyApplication" />
        <activity
            android:name=".LoginActivity"
            android:exported="true"
            android:label="@string/app_name"
            android:theme="@style/Theme.SurveyApplication">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />

                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
```

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

B. MAIN ACTIVITY

```
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)
        }
    }
}
```

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

C. REGISTER ACTIVITY

```
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 {
```

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

D. LOGIN ACTIVITY

```
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 {
```

C0MPOSE INPUT:A Demonstration of Text Input And Validation with Android Compose

TEAM: 6

1. **SREERAM I [TEAM LEADER]**
2. **PERUMAL.N**
3. **ASWIN.T**
4. **MUGESH KUMAR.S.M**
5. **PON ADHARSH S.R**