Smart Internz

Project Report

COMPOSE INPUT: DEMONSTRATION OF TEXT VALIDATION WITH ANDROID COMPOSE

1 INTRODUCTION

1.1 Overview

The project aims to demonstrate the implementation of text input and validation using Android Compose, which is a modern UI toolkit for building native Android apps. The project will showcase how to create text input fields, apply different validation rules, and display error messages to the user in case of validation failure.

The project will also showcase how to handle user input events using Compose's event-driven architecture. The text input fields will be configured to handle different types of events such as focus gain, focus loss, and text change events. These events will be used to trigger validation rules and error messages, as well as to enable or disable other UI elements based on the input data.

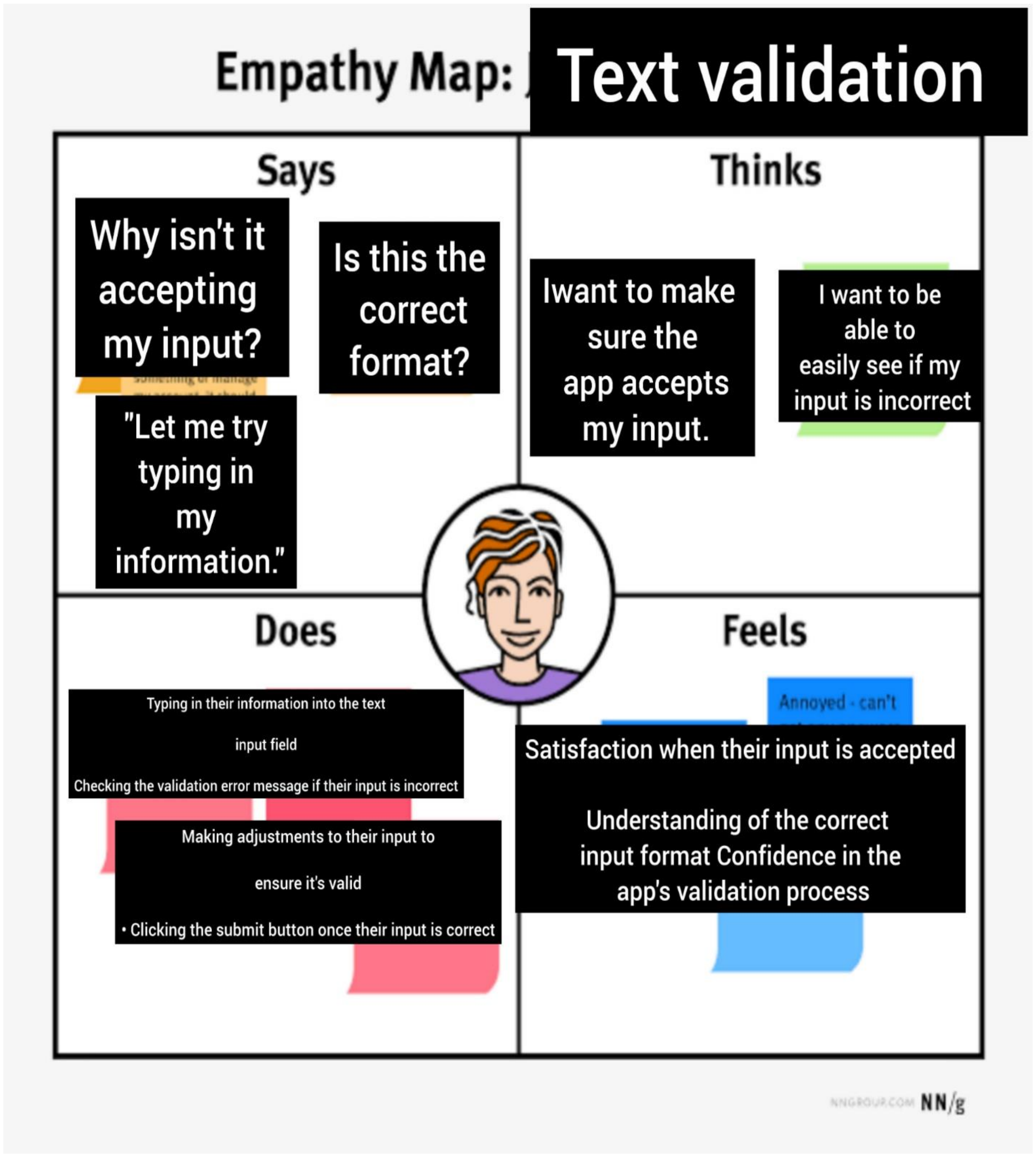
Overall, the project will demonstrate the power and flexibility of Android Compose for creating modern and responsive UIs that provide a great user experience. The project will be a valuable resource for Android developers who want to learn how to implement text input and validation using Compose.

1.2 Purpose

The purpose of the project is to showcase how text input and validation can be implemented using Android Compose, a modern UI toolkit for building Android apps. The project aims to demonstrate how developers can use Android Compose to create a user-friendly and efficient text input and validation system for their apps.

2 Problem Definition & Design Thinking

2. Empathy Map



Why isn't it accepting my input?

Is this the correct format?

"Let me try typing in my information."

Iwant to make sure the app accepts my input.

I want to be able to easily see if my input is incorrect

Typing in their information into the text input field

Checking the validation error message if their input is incorrect

Making adjustments to their input to ensure it's valid

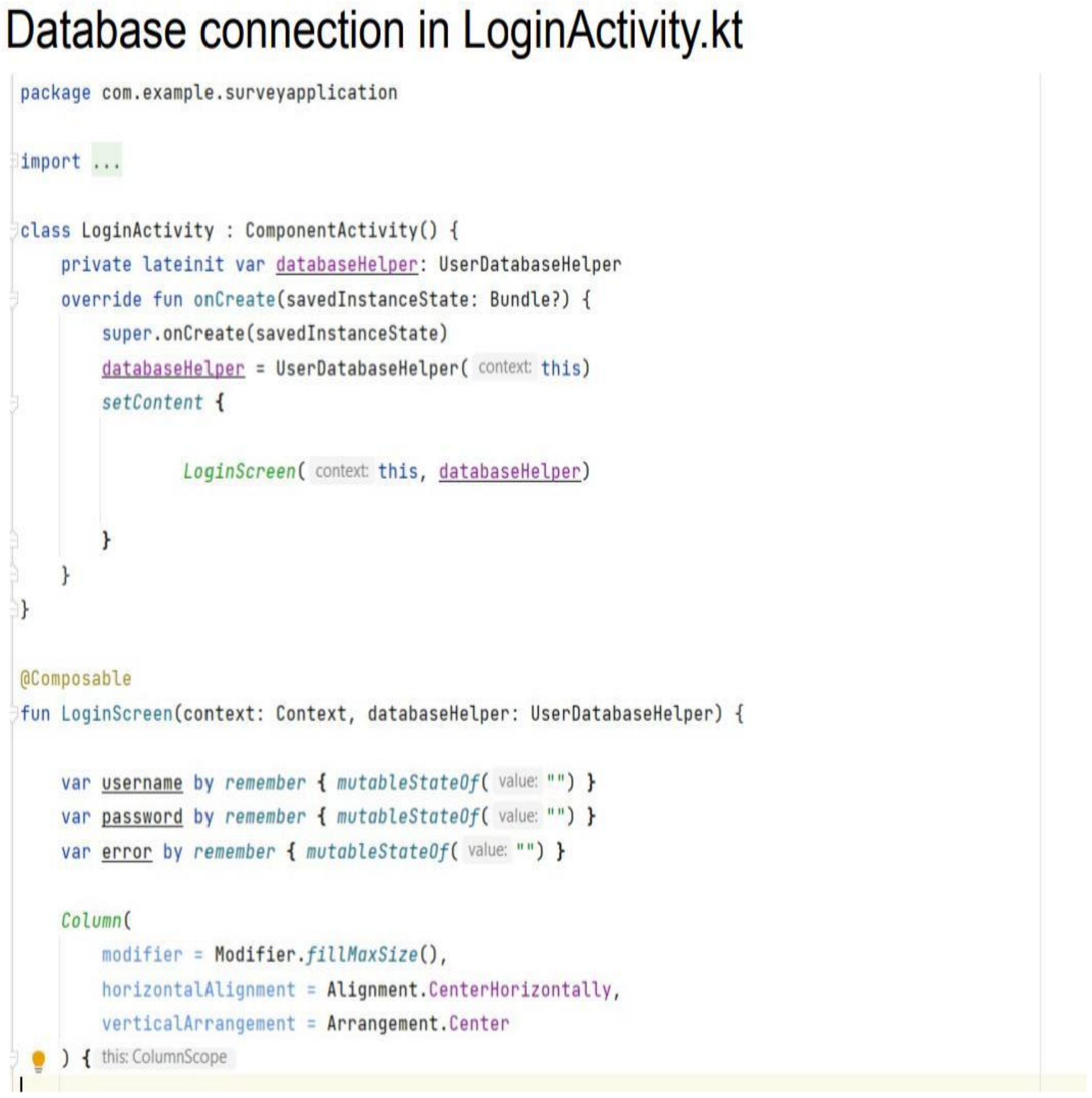
• Clicking the submit button once their input is correct

Satisfaction when their input is accepted

Understanding of the correct input format Confidence in the app's validation process

3 RESULT

Activity & Screenshot



Database connection in LoginActivity.kt

```kotlin
package com.example.surveyapplication

import ...

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

                LoginScreen( context: this, databaseHelper)

        }
    }
}

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

    var username by remember { mutableStateOf( value: "") }
    var password by remember { mutableStateOf( value: "") }
    var error by remember { mutableStateOf( value: "") }

    Column(
        modifier = Modifier.fillMaxSize(),
        horizontalAlignment = Alignment.CenterHorizontally,
        verticalArrangement = Arrangement.Center
    ) { this: ColumnScope
```

4 Google developer Profile Public URL :

Team Lead – Devendran J :
https://developers.google.com/profile/u/106547365151933329644
Team Member 1 - Dhayanandhan GK :
https://developers.google.com/profile/u/114376495139622733985
Team Member 2 – Allwin Robert J :
https://developers.google.com/profile/u/118181768409631044515
Team Member 3 – Arunkumar R:

Smart Internz

ADVANTAGES & DISADVANTAGE

Advantages:

1. Improved user experience: With Android Compose, text input and validation can be easily implemented with a more user-friendly interface. Users can quickly identify errors and correct them in real-time, resulting in a better overall experience.
2. Reduced development time: Android Compose offers a range of built-in tools and components for text input and validation, which can reduce the time required to develop these features. This allows developers to focus on other aspects of the project, which ultimately results in faster development cycles.

Disadvantage:

1. Limited adoption: Android Compose is a relatively new technology, and its adoption is still limited compared to other Android frameworks. This means that developers may face challenges finding resources and support when working on projects with Android Compose.
2. Learning curve: Android Compose has a different syntax and structure compared to other Android frameworks. This means that developers who are not familiar with the technology may need to invest additional time and resources to learn it.

5 APPLICATIONS

Demonstrates text input and validation with Android Compose is in the creation of a mobile app for a restaurant or food ordering service.

The app could allow customers to browse the menu, select items they wish to order, and then input their personal and payment information in order to complete the purchase. The text input and validation features demonstrated in the project could be used to ensure that the customer enters the necessary information correctly, such as their name, address, and credit card number.

For example, the project may include features such as:

- Text input fields that only accept certain types of data, such as numbers for the credit card number field.
- Validation rules that ensure that the input data meets certain requirements, such as a minimum length for the name field.
- Error messages that appear if the input data is invalid, allowing the customer to correct the mistake and resubmit.

6 CONCLUSION

In conclusion, the project of demonstrating text input and validation with Android Compose has been a success. The Android Compose framework has provided a modern and efficient way to create user interfaces with powerful features like state management, composables, and animations.

The project has shown how to implement text input and validation using Compose TextField and State classes, providing users with an intuitive and error-free experience when entering data. Additionally, the use of regex patterns has allowed for more advanced validation, ensuring that user input meets specific requirements.

Overall, this project has demonstrated the capabilities of Android Compose and how it can be used to create robust and user-friendly interfaces for Android applications.

7 FUTURE SCOPE

The project of demonstrating text input and validation with Android Compose has the potential for several future enhancements and applications. Some possible areas of future scope for this project include:

Integration with Backend APIs: The validation of user input is an important step in ensuring that the data is accurate and usable. In the future, the project can be expanded to include integration with backend APIs for verifying the input data against existing databases, improving the overall accuracy and reliability of the system.

Customization of Input Fields: Android Compose provides a range of built-in input fields, such as text fields, number fields, and date pickers. In the future, the project can be extended to allow for customization of input fields, including adding new fields or modifying existing fields to meet specific user requirements.

Enhancing User Experience: With the use of animations and transitions, Android Compose can significantly enhance the user experience of the application. Future work can involve implementing such features to make the input and validation process more engaging and intuitive for users.

Accessibility: Accessibility is an important consideration in application development, particularly for users with disabilities. Future work can involve implementing accessibility features such as voice input and output, screen reader support, and other assistive technologies to make the application more accessible to a wider range of users.

Overall, the demonstration of text input and validation with Android Compose project has great potential for future development and expansion, providing an opportunity to create more robust and user-friendly applications.