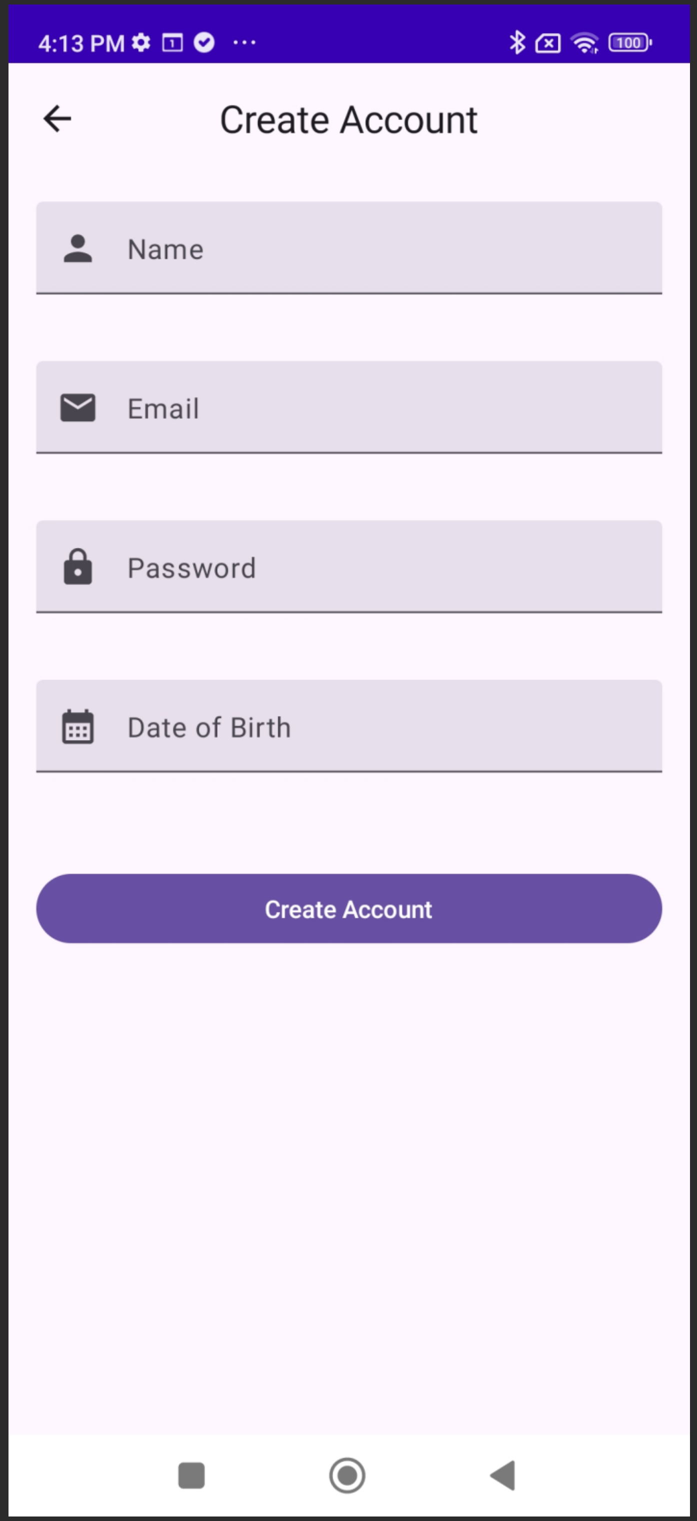
A More Efficient Way to Perform Form Validation in Jetpack Compose

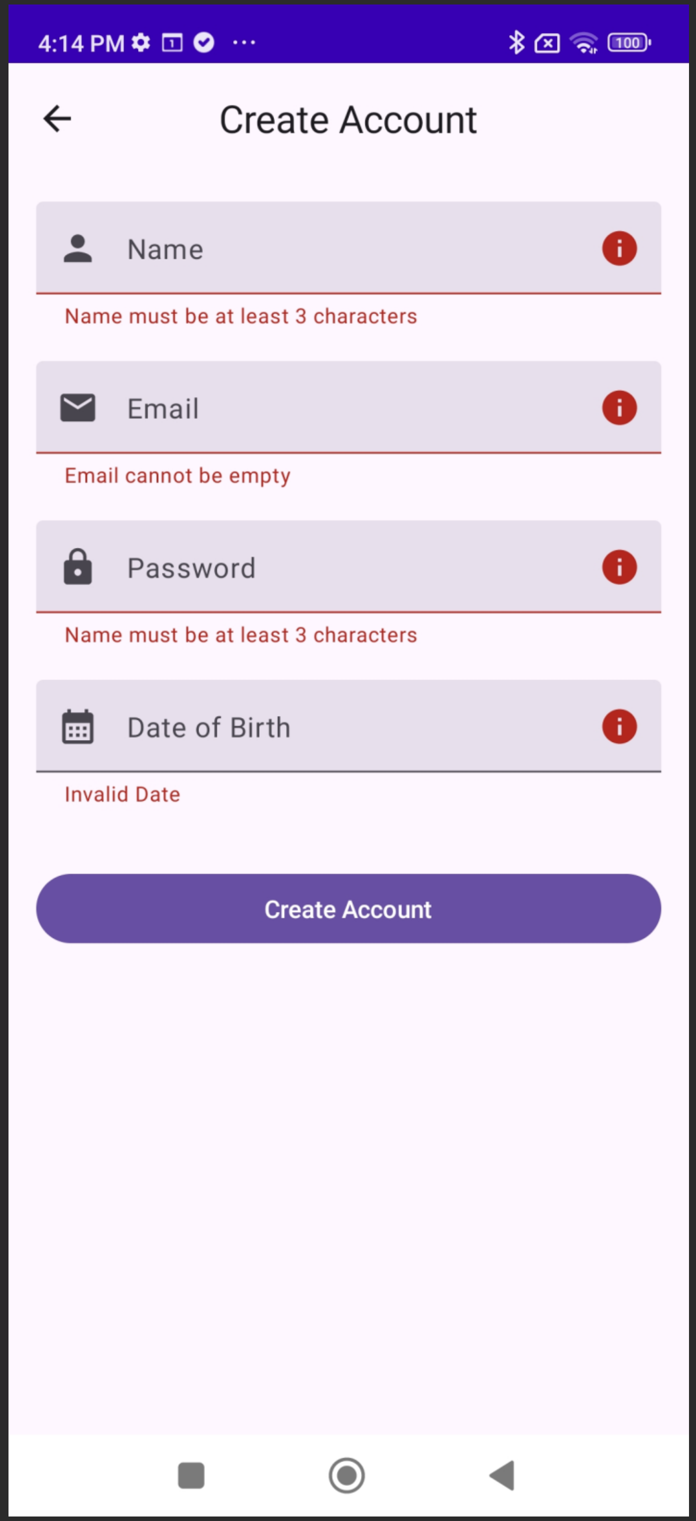
Belal Khan

@probelalkhan

in/probelalkhan

Form Validation





Name must be at least 3 characters

Email cannot be empty

Name must be at least 3 characters

Invalid Date

droidcon academy

Why Form Validation?

- User experience
- Server load
- Accuracy
- Efficiency

TextField

```kotlin
TextField(
    modifier = Modifier
        .fillMaxWidth(),
    value = "",
    onValueChange = { },
    singleLine = true,
    isError = false, //@Update Error State
    supportingText = {
        Text(text = "Error Message or Supporting Message")
    },
    placeholder = { Text(text = "Hint") }
)
```

Validation Result

```kotlin
data class ValidationResult(
    @StringRes val errorMessage: Int? = null
) {
    val isValid: Boolean
        get() = errorMessage == null
}
```

Input Validator

```kotlin
interface InputValidator {
    fun validate(input: String): ValidationResult
}
```

Input Validator

```
class NameValidator : InputValidator {
    override fun validate(input: String): ValidationResult {
        return if (input.length < 3) {
            ValidationResult(R.string.error_name_length)
        } else {
            ValidationResult()
        }
    }
}
```

Validator Factory

```
object ValidatorFactory {

    fun create(): Map<CreateAccountParam, InputValidator> = mapOf(
        CreateAccountParam.FULL_NAME to NameValidator(),
        CreateAccountParam.EMAIL to EmailValidator(),
        .
        .
        .
    )

}
```

Form Event

```kotlin
sealed class CreateAccountEvent {
    data class NameChanged(val name: String) : CreateAccountEvent()
    data class EmailChanged(val email: String) : CreateAccountEvent()
    .
    .
    .
    object CreateAccount : CreateAccountEvent()
}
```

Form State

```kotlin
data class CreateAccountState(
    val name: String = "",
    @StringRes val nameError: Int? = null,

    val email: String = "",
    @StringRes val emailError: Int? = null,

    .
    .
    .
)
```

droidcon academy

Get the Starter Project

Summary

- Created Create Account Form with Jetpack Compose
- Understood how TextField displays additional information
- Created a Form State using Data Class
- Created Form Events using Sealed Class
- Created an Input Validator interface and used it for validating different inputs.