

Wanderlust: A Personalized Travel Planning And Tracking App

INTRODUCTION

1.1 Overview

A project built using the Android Compose UI toolkit. It demonstrates how to create a simple travel app using the Compose libraries. It also features a personalized feed of recommended accommodations based on the locations.

1.2 purpose

By end of this project:

- we will be able to work on Android studio and build an app.
 - we will be able to integrate the database accordingly.
-
- Planning and organizing your own trip is of course great. You're your own tour guide and you have everything in your own hands. But there are a few things to take into consideration when you're thinking about doing everything on (things to take into account) of planning your own trip.
-
- If you like it when everything is organized for you, or you don't have a great amount of time you want to (or can) put into researching everything and planning your trip, and you don't mind paying a little extra for the comfort of not having to worry about anything, then go with an organized trip.

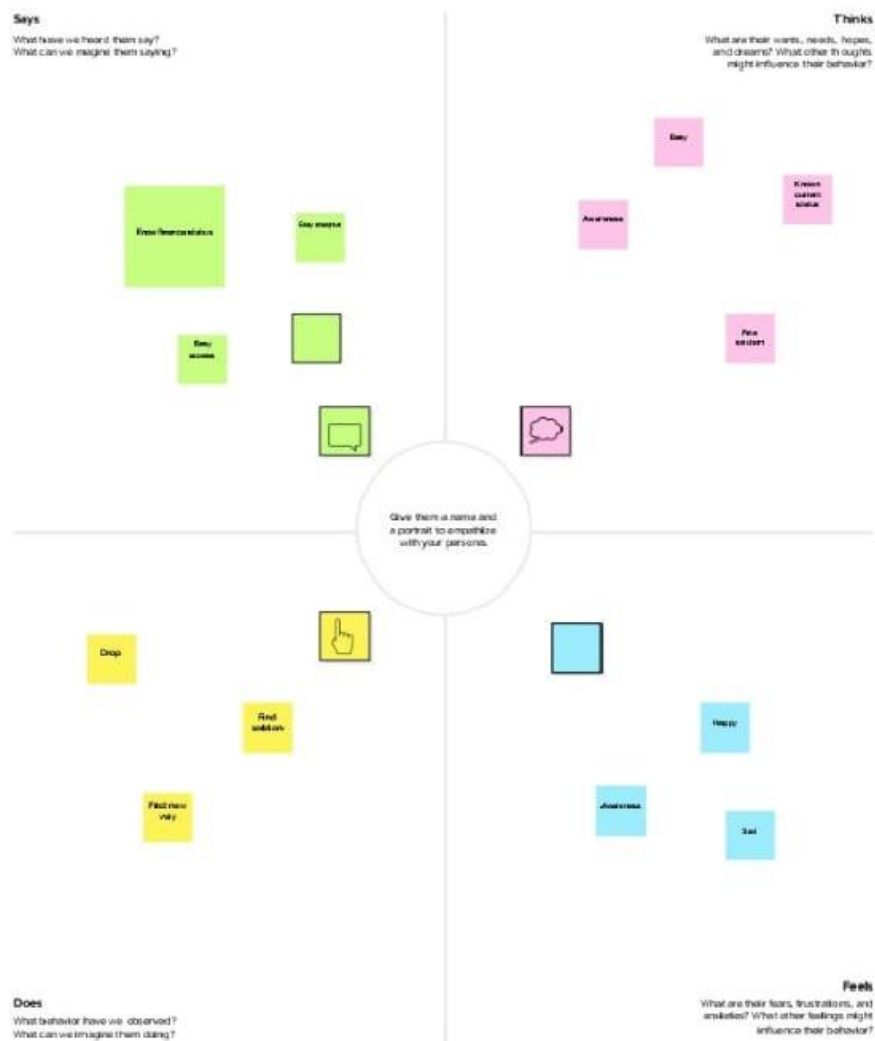
Problem definition& design thinking

2.1 empathy map:



Build empathy

The information you add here should be representative of the observations and research you've done about your users.



Result

Login Page :



Login

Login

[Register](#)

[Forget password?](#)

RegisterPage :



Register

Register

Have an account? [Log in](#)

MainPage :

Wanderlust Travel



Bali

Super saver pack with less than \$10000
7days/2persons



Paris

Super saver pack with less than \$10000
7days/2persons



Singapore

Bali



Day 1: Arrival and Relaxation

Arrive in Bali and check into your hotel or accommodation.

Spend the day relaxing and getting acclimated to the island.

If you have time, explore the nearby area or head to the beach.

Day 2: Ubud Tour

Start your day early and head to Ubud, a cultural and artistic hub in Bali.

Visit the Monkey Forest and the Ubud Palace.

Take a tour of the Tegalalang Rice Terrace, a beautiful UNESCO World Heritage Site.

End your day with a traditional Balinese dance performance.

Day 3: Temple Hopping

Visit some of Bali's most famous temples, such as Tanah Lot and Uluwatu.

Take in the stunning views of the ocean and cliffs.

Enjoy a sunset dinner at one of the many restaurants near the temples.

ADVENTAGES

- You're free as a bird, you can choose wherever you want to go (within budget of course)
- You have the freedom to create your own itinerary. And **travel at your own pace**. Going where ever you want to go. And staying in places as long as you want to. That's a big plus for me!
- If you have a chronic pain or fatigue disorder, a 15-23 days organized tour around a country isn't always a possibility anymore. For me at least it isn't. They have a too busy program. So the advantage of going on my own is that I get to go to only the things I'm actually physical capable of.
- If you need a day of **relaxing between busy days**, just plan a day of laying by the pool or on the beach.
- You can **shop around** for the best hotel offers.
- **Choose where you want to stay**. Like in the busy center of a town (if you like going out at night) or in a more quiet area. In a beautiful hotel by the beach, or in a less expensive hotel at a 5 minute walking distance from the beach.
- You can **shop around** for the best flight offers. And choose your own seats.
- If you book your own hotel and flights, and you book them often with the same booking site, you'll may get **discounts** on next bookings. I get a nice discount at
- You can **shop around** for nice day tours and compare prices and conditions.
- The joy of the vacation starts before you even left your house. All the research and planning just **fills you with anticipation**. And for me, is part of the fun of traveling. Knowing I did it all by myself and creating my own itinerary.
- It's often **cheaper** when you book everything yourself. When you let an organization create your trip or if you go on an organized round tour in a country (with a group of people), the price is often much higher than when you go by yourself.

DISADVANTAGES

- There's a lot of planning involved, which **takes a lot of time**. I mean, a lot! Researching everything, looking up reviews, going to Pinterest and websites for the best tips on things you want to see and do. Searching for which hotels to choose. Booking everything. It's really a lot of work.
- If something goes wrong, there's no one to hold accountable for it. It's on you. And you **need to figure things out yourself**. There's no travel guide you can turn to.
- If you travel with someone else, and you're the one who has done all the planning, **you turn into the travel guide** while you're on holiday. The other person will look to you for all the knowledge and answers.
- **You have to have a credit card**. Without one, there's no way you can arrange all the hotels, flights and more. And in other countries (like the United States) everyone seems to have a credit card (at least, a lot of people seem to do), but in Europe that isn't always the case. And it's much harder to get a credit card. So that's something you have to take into account. If you want to travel by yourself, make sure you have a credit card. With enough spending limit on it.
- When you go on an organized vacation/round trip in a country, with a group of other people and a tour guide, you'll have less to worry about. Everything is arranged and the guide probably has done that trip multiple times, so no surprises there. All the transport is arranged. When you plan your own travel, **you'll always have to plan ahead** and make sure you know how to get from point A to point B. And spend time in advance figuring out for example how the transport system works in a country, how taxi companies operate, which ferry to take etc.
- **You're more likely to get scammed**, if you don't do enough research up front. For example by taxi's and (fake) tour guides. So again, this takes time to look up and figure out.

APPLICATION

Features that make a travel app successful

Intelligent booking systems

Intelligent booking systems help travelers and tourists perform several functions including setting up reminders, price comparison, timely notifications on new discounts, save future bookings and categorization in one platform.

This is a feature that can fit excellently within any body's lifestyle and is very easy to use making the user experience a buzz.

Search engine optimization (specificity in everything)

Most travelers browse for certain locations and the services offered there. For example thrilling places in Europe or Africa like Maasai Mara. If they want to visit an exotic destination they can type "London resorts" and they will get almost everything about the best places.

Although this seems pretty straightforward, what actually allows apps to cover this are several aspects, including having a seamless design and strong social media presence which makes sure that users find what they need conveniently. And this is where SEO comes in handy.

Effective SEO has the capability to direct users in the right direction every time. And a good travel SEO guide is enough to help travel and tourism agencies take notes about the next strategic move to make.

Additionally, apps that are optimized for local SEO enhance the user experience by offering localized suggestions and is served to travel based on their preferences and location.

The apps utilize mobile location insights to not only offer value to users but also optimize user experience and efficiency of planning.

In-app cab booking capabilities

Mobile travel Apps also help users make flight, train, hotel and cab bookings with a few clicks without necessarily moving out of the application. Booking a cab this way makes it possible for the user to see the exact location and the estimated arrival time of the cab.

However, it also expands the scope of information by giving valuable details such as registration number, drivers contact info, cars color etc. Amongst the thoroughly detailed information advantage, there's also the security factor, which makes traveling safer.

Geolocation features

After arriving at their destination, most travelers want to understand everything around them. This includes how to get the club, restaurant or hotel. And this where [geolocation features on travel apps](#) come in handy.

It helps people get around comfortably and easily without worrying about getting lost in the middle of nowhere. This is very important especially for people who are traveling to a place they have never been to before.

Climate or weather forecasting

Real-time weather forecasting features help travelers plan and make timely flight bookings. It also helps users pack the right clothes according to the current weather.

There are numerous apps with amazing features, which are suitable for different continents, regions and travel type. Some even go as far as allowing users to know the temperature of the sea, humidity rate, hurricane forecaster etc.

Trip reviews

With the help of this feature, tourists can review a destination and share their real-life experience with others. It helps raise awareness and better the services of cafes, hotels and other related businesses.

Streamlining of transactions

Moreover, with the help of this feature, the admin can monitor the gas distribution per vendor and can charge them accordingly.

Whether it is providing clients with digital receipts, confirmations or going completely paperless, travel firms use mobile apps to streamline various transactions and provide customers with a quick and straightforward experience.

This not only minimizes the need for companies to provide hard copy documents but also reduces the necessary personnel as well as the [resources](#) required to complete transactions.

Conclusions

- The travel industry is a multi-billion-dollar industry today, thanks to the ever-evolving travel application development solutions that offer better, reliable and faster services to users.
- Use of smartphones for travel planning and booking is surging considerably and the competition is not only good for industry players but also offers incredible incentives and benefits to travellers.

APPENDIX

User.kt

```
import androidx.room.ColumnInfo
import androidx.room.Entity
import androidx.room.PrimaryKey

@Entity(tableName = "user_table")
data class User(
    @PrimaryKey(autoGenerate = true) val id: Int?,
    @ColumnInfo(name = "first_name") val firstName: String?,
    @ColumnInfo(name = "last_name") val lastName: String?,
    @ColumnInfo(name = "email") val email: String?,
    @ColumnInfo(name = "password") val password: String?,

)
```

UserDao.kt

```
import androidx.room.*

@Dao
interface UserDao {

    @Query("SELECT * FROM user_table WHERE email = :email")
    suspend fun getUserByEmail(email: String): User?

    @Insert(onConflict = OnConflictStrategy.REPLACE)
    suspend fun insertUser(user: User)

    @Update
    suspend fun updateUser(user: User)

    @Delete
    suspend fun deleteUser(user: User)
```

```
}
```

UserDatabase.kt

```
import android.content.Context
import androidx.room.Database
import androidx.room.Room
import androidx.room.RoomDatabase

@Database(entities = [User::class], version = 1)
abstract class UserDatabase : RoomDatabase() {

    abstract fun userDao(): UserDao

    companion object {

        @Volatile
        private var instance: UserDatabase? = null

        fun getDatabase(context: Context): UserDatabase {
            return instance ?: synchronized(this) {
                val newInstance = Room.databaseBuilder(
                    context.applicationContext,
                    UserDatabase::class.java,
                    "user_database"
                ).build()
                instance = newInstance
                newInstance
            }
        }
    }
}
```

UserDatabaseHelper.kt

```
import android.annotation.SuppressLint
import android.content.ContentValues
import android.content.Context
```

```

import android.database.Cursor
import android.database.sqlite.SQLiteDatabase
import android.database.sqlite.SQLiteOpenHelper

class UserDatabaseHelper(context: Context) :
    SQLiteOpenHelper(context, DATABASE_NAME, null,
DATABASE_VERSION) {

    companion object {
        private const val DATABASE_VERSION = 1
        private const val DATABASE_NAME = "UserDatabase.db"

        private const val TABLE_NAME = "user_table"
        private const val COLUMN_ID = "id"
        private const val COLUMN_FIRST_NAME = "first_name"
        private const val COLUMN_LAST_NAME = "last_name"
        private const val COLUMN_EMAIL = "email"
        private const val COLUMN_PASSWORD = "password"
    }

    override fun onCreate(db: SQLiteDatabase?) {
        val createTable = "CREATE TABLE $TABLE_NAME (" +
            "$COLUMN_ID INTEGER PRIMARY KEY AUTOINCREMENT,
" +
            "$COLUMN_FIRST_NAME TEXT, " +
            "$COLUMN_LAST_NAME TEXT, " +
            "$COLUMN_EMAIL TEXT, " +
            "$COLUMN_PASSWORD TEXT" +
            ")"

        db?.execSQL(createTable)
    }

    override fun onUpgrade(db: SQLiteDatabase?, oldVersion:
Int, newVersion: Int) {
        db?.execSQL("DROP TABLE IF EXISTS $TABLE_NAME")
        onCreate(db)
    }
}

```

```

fun insertUser(user: User) {
    val db = writableDatabase
    val values = ContentValues()
    values.put(COLUMN_FIRST_NAME, user.firstName)
    values.put(COLUMN_LAST_NAME, user.lastName)
    values.put(COLUMN_EMAIL, user.email)
    values.put(COLUMN_PASSWORD, user.password)
    db.insert(TABLE_NAME, null, values)
    db.close()
}

@SuppressLint("Range")
fun getUserByUsername(username: String): User? {
    val db = readableDatabase
    val cursor: Cursor = db.rawQuery("SELECT * FROM
$TABLE_NAME WHERE $COLUMN_FIRST_NAME = ?", arrayOf(username))
    var user: User? = null
    if (cursor.moveToFirst()) {
        user = User(
            id =
cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
            firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
            lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
            email =
cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
            password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
        )
    }
    cursor.close()
    db.close()
    return user
}

@SuppressLint("Range")

```



```

fun getUserById(id: Int): User? {
    val db = readableDatabase
    val cursor: Cursor = db.rawQuery("SELECT * FROM
$TABLE_NAME WHERE $COLUMN_ID = ?", arrayOf(id.toString()))
    var user: User? = null
    if (cursor.moveToFirst()) {
        user = User(
            id =
cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
            firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),
            lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
            email =
cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
            password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
        )
    }
    cursor.close()
    db.close()
    return user
}

```

```

@SuppressLint("Range")
fun getAllUsers(): List<User> {
    val users = mutableListOf<User>()
    val db = readableDatabase
    val cursor: Cursor = db.rawQuery("SELECT * FROM
$TABLE_NAME", null)
    if (cursor.moveToFirst()) {
        do {
            val user = User(
                id =
cursor.getInt(cursor.getColumnIndex(COLUMN_ID)),
                firstName =
cursor.getString(cursor.getColumnIndex(COLUMN_FIRST_NAME)),

```

```

        lastName =
cursor.getString(cursor.getColumnIndex(COLUMN_LAST_NAME)),
        email =
cursor.getString(cursor.getColumnIndex(COLUMN_EMAIL)),
        password =
cursor.getString(cursor.getColumnIndex(COLUMN_PASSWORD)),
    )
    users.add(user)
} while (cursor.moveToNext())
}
cursor.close()
db.close()
return users
}
}

```

LoginActivity.kt

```

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.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.ContextCompat

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.trav),
contentDescription = "")

        Text(
            fontSize = 36.sp,

```

```

        fontWeight = FontWeight.ExtraBold,
        fontFamily = FontFamily.Cursive,
        text = "Login"
    )
    Spacer(modifier = Modifier.height(10.dp))

    TextField(
        value = username,
        onChange = { username = it },
        label = { Text("Username") },
        modifier = Modifier.padding(10.dp)
            .width(280.dp)
    )

    TextField(
        value = password,
        onChange = { 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
                )
            )

        }
        else {
            error = "Invalid username or password"
        }

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

},
modifier = Modifier.padding(top = 16.dp)
) {
    Text(text = "Login")
}
Row {
    TextButton(onClick = {context.startActivity(
        Intent(
            context,
            RegisterActivity::class.java
        )
    ))
    { Text(text = "Register") }
    TextButton(onClick = {
    })

    {
        Spacer(modifier = Modifier.width(60.dp))
    }
}

```

```

        Text(text = "Forget password?")
    }
}
}
}
private fun startMainPage(context: Context) {
    val intent = Intent(context, MainActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}

```

RegisterActivity.kt

```

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.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.ContextCompat

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.tra),
contentDescription = "")

        Text(
            fontSize = 36.sp,
            fontWeight = FontWeight.ExtraBold,
            fontFamily = FontFamily.Cursive,
            text = "Register"
        )
    }
}
```

```

Spacer(modifier = Modifier.height(10.dp))
TextField(
    value = username,
    onChange = { username = it },
    label = { Text("Username") },
    modifier = Modifier
        .padding(10.dp)
        .width(280.dp)

)

TextField(
    value = email,
    onChange = { email = it },
    label = { Text("Email") },
    modifier = Modifier
        .padding(10.dp)
        .width(280.dp)

)

TextField(
    value = password,
    onChange = { 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"

                context.startActivity(
                    Intent(
                        context,
                        LoginActivity::class.java
                    )
                )

            } else {
                error = "Please fill all fields"
            }
        },
        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(text = "Log in")
    }
}

}

private fun startLoginActivity(context: Context) {
    val intent = Intent(context, LoginActivity::class.java)
    ContextCompat.startActivity(context, intent, null)
}

```

MainActivity.kt

```

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.clickable
import androidx.compose.foundation.layout.*
import androidx.compose.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
import androidx.compose.material.Card

```

```

import androidx.compose.material.Text
import androidx.compose.runtime.Composable
import androidx.compose.ui.Alignment
import androidx.compose.ui.Modifier
import androidx.compose.ui.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.text.font.FontWeight
import androidx.compose.ui.text.style.TextAlign
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp

class MainActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelApp(this)
        }
    }
}

@Composable
fun TravelApp(context: Context) {
    Column(
        modifier = Modifier
            .padding(20.dp)
            .verticalScroll(rememberScrollState())

    ) {

        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120,
40, 251)),

            fontFamily = FontFamily.Cursive,
            text = "Wanderlust Travel"

```

```

    )

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

    Card(
        modifier = Modifier
            .fillMaxWidth()
            .height(250.dp)
            .clickable {
                context.startActivity(
                    Intent(context,
BaliActivity::class.java)

                )
            },
        elevation = 8.dp
    )
    {
        Column(
            horizontalAlignment =
Alignment.CenterHorizontally
        ) {
            Image(
                painterResource(id = R.drawable.bali),
contentDescription = "",
                modifier = Modifier
                    .height(150.dp)
                    .scale(scaleX = 1.2F, scaleY = 1F)
            )

            Text(
                text = stringResource(id =
R.string.place_1),
                fontSize = 18.sp
            )
        }
    }

```

```

        Text(
            text = stringResource(id =
R.string.description),
            fontWeight = FontWeight.Light,
            fontSize = 16.sp,
            textAlign = TextAlign.Center,
        )

        Text(
            text = stringResource(id =
R.string.plan), color = Color.Gray,
            fontSize = 16.sp
        )
    }
}

```

```

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

```

```

Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
        .clickable {
            context.startActivity(
                Intent(context,
ParisActivity::class.java)
            )
        },
    elevation = 8.dp
) {
    Column(
        horizontalAlignment =
Alignment.CenterHorizontally
    ) {

```

```
        Image(
            painterResource(id = R.drawable.paris),
contentDescription = "",
            modifier = Modifier
                .height(150.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
```

```
        Text(
            text = stringResource(id =
R.string.place_2),
            fontSize = 18.sp
        )
```

```
        Text(
            text = stringResource(id =
R.string.description),
            fontWeight = FontWeight.Light,
            fontSize = 16.sp,
            textAlign = TextAlign.Center,
        )
```

```
        Text(
            text = stringResource(id =
R.string.plan), color = Color.Gray,
            fontSize = 16.sp
        )
```

```
    }
}
```

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

```
//03
```

```
Card(
    modifier = Modifier
        .fillMaxWidth()
        .height(250.dp)
```

```

        .clickable {
            context.startActivity(
                Intent(context,
SingaporeActivity::class.java)

            )
        },
        elevation = 8.dp
    )
    {
        Column(
            horizontalAlignment =
Alignment.CenterHorizontally
        ) {
            Image(
                painterResource(id =
R.drawable.singapore), contentDescription = "",
                modifier = Modifier
                    .height(150.dp)
                    .scale(scaleX = 1.2F, scaleY = 1F)
            )

            Text(
                text = stringResource(id =
R.string.place_3),
                fontSize = 18.sp
            )

            Text(
                text = stringResource(id =
R.string.description),
                fontWeight = FontWeight.Light,
                fontSize = 16.sp,
                textAlign = TextAlign.Center,
            )

            Text(

```

```

                text = stringResource(id =
R.string.plan), color = Color.Gray,
                fontSize = 16.sp
            )
        }
    }

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

```

BaliActivity.kt

```

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.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
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.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

```



```

class BaliActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {

                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    PlaceOne()
                }
            }
        }
    }
}

@Composable
fun PlaceOne() {
    Column(modifier = Modifier.background(color = Color.White)
        .padding(20.dp)
        .verticalScroll(rememberScrollState()))
    {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40,
251)),

            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_1),
        )
        Image(
            painterResource(id = R.drawable.bali),
            contentDescription = "",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
        )
    }
}

```

```
        .scale(scaleX = 1.2F, scaleY = 1F)
    )
    Text(
        color=Color.Black,
        text = "Day 1: Arrival and Relaxation\n" +

            "Arrive in Bali and check into your hotel
or accommodation.\n" +

            "Spend the day relaxing and getting
acclimated to the island.\n" +

            "If you have time, explore the nearby area
or head to the beach.\n" +

            "\n" +

            "Day 2: Ubud Tour\n" +

            "Start your day early and head to Ubud, a
cultural and artistic hub in Bali.\n" +

            "Visit the Monkey Forest and the Ubud
Palace.\n" +

            "Take a tour of the Tegalalang Rice
Terrace, a beautiful UNESCO World Heritage Site.\n" +

            "End your day with a traditional Balinese
dance performance.\n" +

            "\n" +

            "Day 3: Temple Hopping\n" +

            "Visit some of Bali's most famous temples,
such as Tanah Lot and Uluwatu.\n" +

            "Take in the stunning views of the ocean
and cliffs.\n" +

            "Enjoy a sunset dinner at one of the many
restaurants near the temples.\n" +

            "\n" +

            "Day 4: Waterfalls and Beaches\n" +

            "Take a day trip to Bali's beautiful
waterfalls, such as Tegenungan or Gitgit.\n" +
```

```

        "Spend the afternoon at one of Bali's
world-renowned beaches, like Seminyak or Nusa Dua.\n" +
        "\n" +
        "Day 5: Island Hopping\n" +

        "Take a day trip to one of Bali's
neighboring islands, such as Nusa Lembongan or Gili Islands.\n"
+
        "Snorkel or scuba dive in the clear waters
and relax on the beach.\n" +
        "\n" +
        "Day 6: Cultural Activities\n" +
        "Visit a traditional Balinese village and
learn about the island.\n" +
        "\n" +
        "Day 7: Departure\n" +
        "Explore the surrounding area and take in
the stunning sunset views.\n" +
        "Have dinner at a local restaurant before
returning to your accommodation."
    )

}
}

```

ParisActivity.kt

```

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.foundation.rememberScrollState
import androidx.compose.foundation.verticalScroll
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.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class ParisActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {
                // A surface container using the 'background'
                color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting()
                }
            }
        }
    }
}

@Composable
fun Greeting() {

    Column(
        modifier = Modifier.background(color = Color.White)
    )
}

```

```

        .padding(20.dp)
        .verticalScroll(rememberScrollState())
    ) {
        Text(
            fontSize = 40.sp,
            color = Color(android.graphics.Color.rgb(120, 40,
251)),
            fontFamily = FontFamily.Cursive,
            text = stringResource(id = R.string.place_2),
        )
        Image(
            painterResource(id = R.drawable.paris),
contentDescription = "",
            modifier = Modifier
                .padding(16.dp)
                .fillMaxWidth()
                .height(200.dp)
                .scale(scaleX = 1.2F, scaleY = 1F)
        )
        Text(
            color=Color.Black,
            text = "Day 1: Arrival and Introduction\n" +

                "Check into your accommodation and freshen
up\n" +

                "Take a stroll around the neighborhood to
get acquainted\n" +

                "Visit the Eiffel Tower, preferably in the
evening when it is lit up\n" +

                "Have a relaxing dinner at a nearby
restaurant\n" +

                "\n" +

                "Day 2: Art and History\n" +

                "Visit the Louvre Museum to see some of the
world's most famous art pieces\n" +

```

"Stroll through the Tuileries Garden and the Place de la Concorde\n" +

"Visit the Orsay Museum, which houses a large collection of impressionist art\n" +

"Have dinner at a local French restaurant\n" +

"\n" +

"Day 3: French Culture and Food\n" +

"Visit the Montmartre neighborhood to see the famous Basilique du Sacré-Cœur and Place du Tertre\n" +

"Explore the historic neighborhood of Le Marais\n" +

"Try some delicious French pastries at a local bakery\n" +

"Have dinner at a brasserie to taste some classic French cuisine\n" +

"\n" +

"Day 4: Architecture and Gardens\n" +

"Visit the Palace of Versailles, a UNESCO World Heritage site, and explore its beautiful gardens\n" +

"Walk along the Champs-Élysées and stop at the Arc de Triomphe\n" +

"Visit the Sainte-Chapelle, a beautiful Gothic chapel with stunning stained-glass windows\n" +

"Have dinner at a local restaurant in the 7th arrondissement\n" +

"\n" +

"Day 5: Shopping and Sightseeing\n" +

"Visit the Notre-Dame Cathedral and climb up to the top for a stunning view of the city\n" +

```

        "Explore the Latin Quarter and visit the
        Panthéon\n" +
        "Go shopping at the famous Galeries
        Lafayette or Printemps department stores\n" +
        "Have dinner at a local bistro\n" +
        "\n" +
        "Day 6: Parisian Parks and Museums\n" +
        "Visit the Musée Rodin and explore its
        beautiful gardens\n" +
        "Stroll through the Luxembourg Gardens and
        visit the Luxembourg Palace\n" +
        "Visit the Centre Pompidou, a modern art
        museum in the Marais neighborhood\n" +
        "Have dinner at a local restaurant in the
        Latin Quarter\n" +
        "\n" +
        "Day 7: River Cruise and Farewell\n" +
        "Take a boat cruise along the Seine River
        to see the city from a different perspective\n" +
        "Visit the Musée de l'Orangerie, which
        houses Monet's famous water lilies paintings\n" +
        "Have a farewell dinner at a Michelin-
        starred restaurant"
    )
}
}

```

SingaporeActivity.kt

```

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.foundation.rememberScrollState

```

```

import androidx.compose.foundation.verticalScroll
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.draw.scale
import androidx.compose.ui.graphics.Color
import androidx.compose.ui.res.painterResource
import androidx.compose.ui.res.stringResource
import androidx.compose.ui.text.font.FontFamily
import androidx.compose.ui.tooling.preview.Preview
import androidx.compose.ui.unit.dp
import androidx.compose.ui.unit.sp
import com.example.travelapp.ui.theme.TravelAppTheme

class SingaporeActivity : ComponentActivity() {
    override fun onCreate(savedInstanceState: Bundle?) {
        super.onCreate(savedInstanceState)
        setContent {
            TravelAppTheme {
                // A surface container using the 'background'
                color from the theme
                Surface(
                    modifier = Modifier.fillMaxSize(),
                    color = MaterialTheme.colors.background
                ) {
                    Greeting2()
                }
            }
        }
    }
}

@Composable
fun Greeting2() {

```



```

Column(
    modifier = Modifier.background(color = Color.White)
        .padding(20.dp)
        .verticalScroll(rememberScrollState())
) {
    Text(
        fontSize = 40.sp,
        color = Color(android.graphics.Color.rgb(120, 40,
251)),
        fontFamily = FontFamily.Cursive,
        text = stringResource(id = R.string.place_3),
    )
    Image(
        painterResource(id = R.drawable.singapore),
        contentDescription = "",
        modifier = Modifier
            .padding(16.dp)
            .fillMaxWidth()
            .height(200.dp)
            .scale(scaleX = 1.2F, scaleY = 1F)
    )
    Text(
        color = Color.Black,
        text = "Day 1:\n" +

```

```

            "Morning: Visit Gardens by the Bay and
marvel at the Supertree Grove and the Flower Dome and Cloud
Forest conservatories.\n" +

```

```

            "Afternoon: Explore the Marina Bay Sands
complex, which includes a casino, luxury shopping mall, and
observation deck with a stunning view of the city.\n" +

```

```

            "\n" +

```

```

            "Day 2:\n" +

```

```

            "Morning: Explore the historic district of
Chinatown, including the Buddha Tooth Relic Temple and Museum
and the Sri Mariamman Temple.\n" +

```

"Afternoon: Visit the nearby Clarke Quay for lunch and to explore its waterfront restaurants, bars, and shops.\n" +

"\n" +

"Day 3:\n" +

"Morning: Take a tour of the UNESCO-listed Botanic Gardens, one of the world's most famous and significant tropical gardens.\n" +

"Afternoon: Head over to the National Museum of Singapore, which houses a vast collection of historical and cultural artifacts.\n" +

"\n" +

"Day 4:\n" +

"Morning: Visit the Singapore Zoo and admire the wildlife, including orangutans, tigers, and elephants.\n" +

"Afternoon: Head over to Sentosa Island and relax at one of its many beaches or try some of the many attractions such as Universal Studios Singapore or Adventure Cove Waterpark.\n" +

"\n" +

"Day 5:\n" +

"Morning: Go on a nature walk at MacRitchie Reservoir, which offers hiking trails and stunning views of the city skyline.\n" +

"Afternoon: Visit Little India, a vibrant and colorful neighborhood, and explore the shops, temples, and food stalls.\n" +

"\n" +

"Day 6:\n" +

"Morning: Explore the trendy neighborhood of Tiong Bahru, known for its hip cafes and boutiques, as well as its Art Deco architecture.\n" +

```

        "Afternoon: Visit the National Gallery
Singapore, which houses the largest public collection of modern
art in Singapore and Southeast Asia.\n" +
        "\n" +
        "Day 7:\n" +

        "Morning: Take a day trip to the nearby
island of Pulau Ubin, where you can rent a "
    )
}
}

```

AndroidManifest.xml

```

<xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools">

    <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.TravelApp"
        tools:targetApi="31">
        <activity
            android:name=".RegisterActivity"
            android:exported="false"
            android:label="RegisterActivity"
            android:theme="@style/Theme.TravelApp" />
        <activity
            android:name=".SingaporeActivity"

```

```

        android:exported="false"
        android:label="@string/title_activity_singapore"
        android:theme="@style/Theme.TravelApp" />
<activity
    android:name=".ParisActivity"
    android:exported="false"
    android:label="@string/title_activity_paris"
    android:theme="@style/Theme.TravelApp" />
<activity
    android:name=".BaliActivity"
    android:exported="false"
    android:label="@string/title_activity_bali"
    android:theme="@style/Theme.TravelApp" />
<activity
    android:name=".MainActivity"
    android:exported="true"
    android:label="@string/app_name"
    android:theme="@style/Theme.TravelApp"/>
<activity
    android:name=".LoginActivity"
    android:exported="true"
    android:label="@string/app_name"
    android:theme="@style/Theme.TravelApp">
    <intent-filter>
        <action
            android:name="android.intent.action.MAIN" />

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

</manifest>

```

