ANDROID STUDIO – EXPERIENCE BASED PROJECT LEARNING

Wanderlust: A personalized travel planning and tracking app

Submitted by

SRIRAM S	-	711022104053
KALANIDHI A	-	711022104022
KARNA P	-	711022104024
SIVASINGH V	-	711022104050

BACHELOR OF COMPUTER SCIENCE AND ENGINEERING

IN

FIFTH SEMESTER

COMPUTER SCIENCE AND ENGINEERING

INFO INSTITUTE OF ENGINEERING, COIMBATORE – 641107

NOVEMBER/DECEMBER - 2024

BONAFIDE CERTIFICATE

Certified that this project Wanderlust: "A personalized travel planning and tracking app" the Bonafide work of SRIRAM S (711022104053), KALANIDHI A (711022104022), KARNA P(711022104024),SIVA SINGH(711022104050) who carried out the project work under any supervision.

SIGNATURE

STAFF COORDINATOR

Mrs. A. SARANYA M.E.,

ASSISTANT PROFESSOR

DEPT. COMPUTER SCIENCE AND ENGINEERIN

INFO INSTITUTE OF ENGINEERING,

KOVILPALAYAM COIMBATORE - 641107

SIGNATURE

HEAD OF THE DEPARTMENT

Dr. G. SELVAVINAYAGAM Ph.D.,

HEAD OF THE DEPARTMENT

DEPT. COMPUTER SCIENCE AND ENGINEERING

INFO INSTITUTE OF ENGINEERING,

KOVILPALAYAM COIMBATORE - 641107

INTERNAL EXAMINER

EXTERNAL EXAMINER

ACKNOWLEDGEMENT

We sincerely thank to Tamil Nadu Skill Development Corporation (TNSDC), Naan Mudhalvan" Platform and ANDROID STUDIO – EXPERIENCE BASED PROJECT LEARNING (EBPL) for encouragement towards our project work for providing necessary skill training.

We sincerely thank our Principal Dr. N. KOTTISWARAN, M.E., Ph.D., and Head of the Department Dr. G. SELVAVINAYAGAM, M.E., Ph.D., and also Staff Coordinator Mrs. A. SARANYA M.E for her encouragement towards our project works.

We also thank our project guide and our parents for the complete and whole hearted support, motivation guidance and help in making our project activities.

Abstract:

The *Personalized Travel Planning and Tracking App* is an Android-based application designed to assist users in planning, organizing, and tracking their travel experiences. The app leverages Android's native features to provide a tailored travel experience, helping users efficiently manage trip details, itineraries, and travel activities. With a focus on personalization, the app allows users to input preferences such as destination choices, accommodation types, budget, and travel interests. It then generates a customized travel plan based on these inputs.

Key features include location-based services, real-time tracking, and offline functionality, allowing users to track their journey, access trip details, and receive recommendations even without internet access. The app integrates with third-party APIs to provide dynamic data, including weather updates, local attractions, and transportation options. Furthermore, users can set reminders for important travel tasks, track their expenses, and share their experiences with others.

This app aims to simplify the travel planning process by offering a seamless, intuitive, and highly personalized solution, making travel experiences more enjoyable and stress-free.

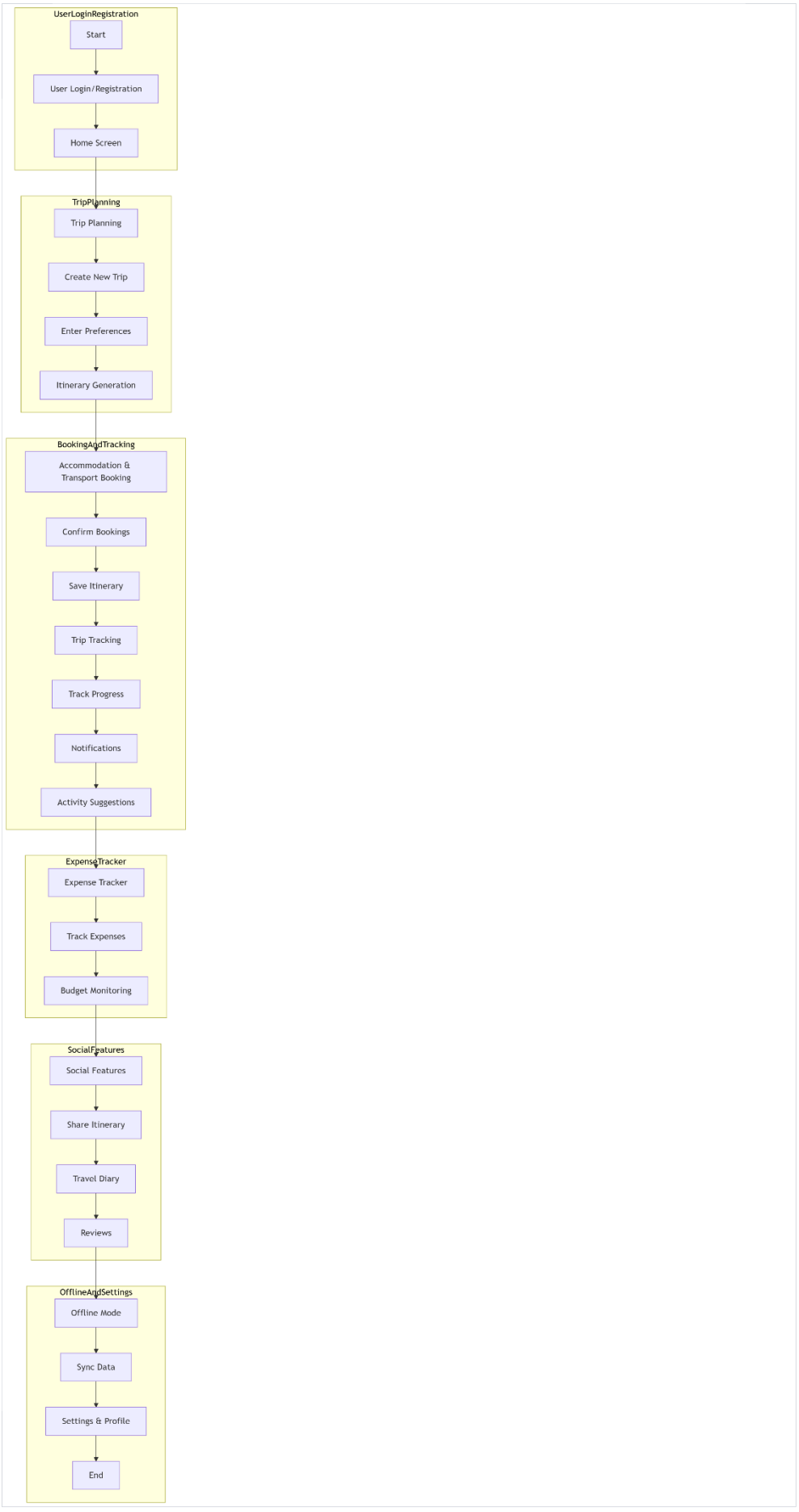
Introduction:

Traveling has become an essential part of modern life, but the process of planning and organizing a trip can often be overwhelming. The *Personalized Travel Planning and Tracking App* aims to revolutionize the way travelers approach their journeys by offering a comprehensive, intuitive, and personalized mobile solution. Designed for Android, the app empowers users to plan their trips from start to finish, while ensuring their travel experience is smooth and enjoyable.

This app provides a personalized approach to travel planning, offering customized recommendations based on user preferences such as destination, budget, and activity interests. With an easy-to-navigate interface, users can effortlessly create itineraries, manage bookings, and track their journey in real-time. The app also offers unique features like offline access to key trip details, real-time notifications, and expense tracking to help users stay organized and informed throughout their travels.

By integrating various tools and resources, including weather updates, local attractions, and transportation options, the app not only simplifies the planning process but also enhances the travel experience, making it both enjoyable and stress-free.

Data Flow Diagram:



Use Case Diagram:

Software Requirements

1. Programming Languages:

Kotlin: Primary language for Android app development, offering modern, concise, and efficient coding.

Java: Used for legacy support and some Android SDK components.

2. Development Tools:

Android Studio: IDE for designing, testing, and deploying Android applications.

Gradle: Build automation tool for managing dependencies and app builds.

3. System Software:

Android OS: Minimum supported version is Android 5.0 higher.

Firebase: Backend services for real-time database syncing, authentication, push notifications, and analytics.

Program code:

plugins

```
{
   id 'com.android.application'
   id 'org.jetbrains.kotlin.android'
}


android {
   namespace 'com.example.travelapp'
```

```
compileSdk 33

defaultConfig {
    applicationId "com.example.travelapp"
    minSdk 21
    targetSdk 33
    versionCode 1
    versionName "1.0"

    testInstrumentationRunner "androidx.test.runner.AndroidJUnitRunner"
    vectorDrawables {
                useSupportLibrary true
    }
}

buildTypes {
    release {
        minifyEnabled false
        proguardFiles getDefaultProguardFile('proguard-android-optimize.txt'), 'proguard-rules.pro'
    }
}
compileOptions {
    sourceCompatibility JavaVersion.VERSION_1_8
    targetCompatibility JavaVersion.VERSION_1_8
}
kotlinOptions {
    jvmTarget = '1.8'
}
buildFeatures {
    compose true
}
composeOptions {
    kotlinCompilerExtensionVersion '1.2.0'
}
```

```
    packagingOptions {
        resources {
            excludes += '/META-INF/{AL2.0,LGPL2.1}'
        }
    }
}

dependencies {

    implementation 'androidx.core:core-ktx:1.7.0'
    implementation 'androidx.lifecycle:lifecycle-runtime-ktx:2.3.1'
    implementation 'androidx.activity:activity-compose:1.3.1'
    implementation "androidx.compose.ui:ui:$compose_ui_version"
    implementation "androidx.compose.ui:ui-tooling-preview:$compose_ui_version"
    implementation 'androidx.compose.material:material:1.2.0'
    implementation 'androidx.room:room-common:2.5.0'
    implementation 'androidx.room:room-ktx:2.5.0'
    testImplementation 'junit:junit:4.13.2'
    androidTestImplementation 'androidx.test.ext:junit:1.1.5'
    androidTestImplementation 'androidx.test.espresso:espresso-core:3.5.1'
    androidTestImplementation "androidx.compose.ui:ui-test-junit4:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-tooling:$compose_ui_version"
    debugImplementation "androidx.compose.ui:ui-test-manifest:$compose_ui_version"

    // Adding Room dependencies

    implementation 'androidx.room:room-common:2.5.0'
    implementation 'androidx.room:room-ktx:2.5.0'
}
```

build.gradle

```
buildscript {
    ext {
        compose_ui_version = '1.2.0'
```

```
    }
}// Top-level build file where you can add configuration options common to all sub-projects/modules.
plugins {
    id 'com.android.application' version '7.4.1' apply false
    id 'com.android.library' version '7.4.1' apply false
    id 'org.jetbrains.kotlin.android' version '1.7.0' apply false
}
```

gradle.xml

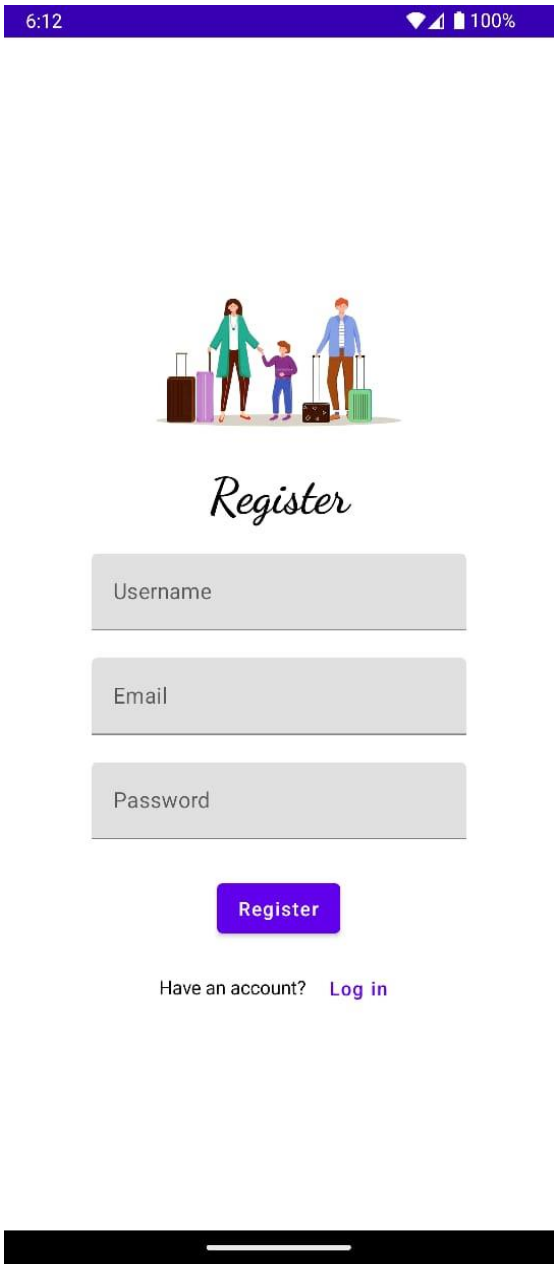
```
<?xml version="1.0" encoding="UTF-8"?>
<project version="4">
  <component name="GradleMigrationSettings" migrationVersion="1" />
  <component name="GradleSettings">
    <option name="linkedExternalProjectsSettings">
      <GradleProjectSettings>
        <option name="testRunner" value="CHOOSE_PER_TEST" />
        <option name="externalProjectPath" value="$PROJECT_DIR$" />
        <option name="gradleJvm" value="jbr-21" />
        <option name="modules">
          <set>
            <option value="$PROJECT_DIR$" />
            <option value="$PROJECT_DIR$/app" />
          </set>
        </option>
        <option name="resolveExternalAnnotations" value="false" />
      </GradleProjectSettings>
    </option>
  </component>
</project>
```

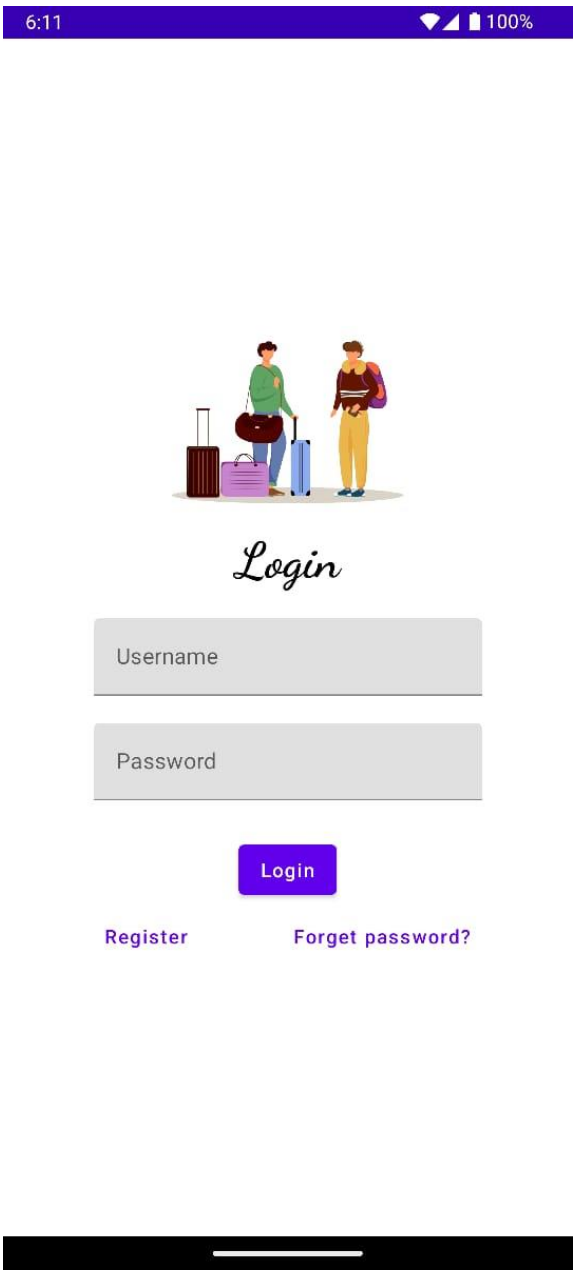
Project_Default.xml

```
<component name="InspectionProjectProfileManager">
  <profile version="1.0">
    <option name="myName" value="Project Default" />
```

```
    <inspection_tool class="PreviewAnnotationInFunctionWithParameters" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewApiLevelMustBeValid" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewDimensionRespectsLimit" enabled="true" level="WARNING"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewFontScaleMustBeGreaterThanZero" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewMultipleParameterProviders" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewMustBeTopLevelFunction" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewNeedsComposableAnnotation" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewNotSupportedInUnitTestFiles" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
    <inspection_tool class="PreviewPickerAnnotation" enabled="true" level="ERROR"
enabled_by_default="true">
      <option name="composableFile" value="true" />
    </inspection_tool>
  </profile>
</component>
```

Output:


6:12
100%
Register
Username
Email
Password
Register
Have an account?
Log in


6:11
100%
Login
Username
Password
Login
Register
Forget password?

6:13 100%

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.

Day 4: Waterfalls and Beaches
Take a day trip to Bali's beautiful waterfalls, such as Tegenungan or Gitgit.
Spend the afternoon at one of Bali's world-renowned beaches, like Seminyak or Nusa Dua.

6:13 100%

Wanderlust Travel

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

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

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

Conclusion:

This travel application aims to streamline the entire travel planning process, from initial trip idea to return home. It provides users with the flexibility to manage their entire trip, including planning, booking, tracking, and budgeting, all from a single platform. By integrating social features and offline functionality, the app ensures users can stay connected and organized throughout their journey, even in areas with poor internet connectivity.

The application is built to offer a seamless user experience with intuitive navigation and reliable performance. By catering to both experienced travelers and first-timers, it provides value through custom trip planning, integrated booking systems, and tools for tracking activities and expenses.

Future Enhancements:

1. **AI-Powered Itinerary Suggestions**: Use machine learning to recommend personalized itineraries based on user preferences, past trips, or popular destinations.
2. **Augmented Reality (AR) Integration**: Provide location-based services like nearby attractions, transport routes, and accommodation availability through AR.
3. **Real-Time Trip Assistance**: Offer a chatbot or live assistant for real-time support, answering questions or adjusting bookings.
4. **Sustainability Features**: Recommend eco-friendly options for accommodations, transport, and activities based on user preferences.
5. **Collaborative Planning**: Enable multiple users to plan and share itineraries together for group or family trips.
6. **Integration with Wearables**: Sync trip details with wearable devices (smartwatches) for updates, reminders, and notifications.
7. **Expanded Social Features**: Add a community or forum for sharing travel experiences, tips, and photos.
8. **Customizable Travel Alerts**: Offer personalized notifications for flight delays, weather changes, or local events affecting the itinerary.
9. **Multi-Destination Travel**: Enhance support for multi-destination trips with automatic itinerary adjustments for multiple cities or countries.