**D424 – Software Engineering**

**Task 3**

|  |  |
| --- | --- |
| **Capstone Proposal Project Name:** | http://www.idevnews.com/views/images/uploads/general/wgu_logo.png  Vacation Planner Mobile App |
| **Student Name:** | Aaron Garro Lorenzo |

Contents

[Application Design and Testing 4](#_Toc177402822)

[Class Design 4](#_Toc177402823)

[UI Design 7](#_Toc177402824)

[Unit Test Plan 13](#_Toc177402825)

[Introduction 13](#_Toc177402826)

[Purpose 13](#_Toc177402827)

[Overview 13](#_Toc177402828)

[Test Plan 13](#_Toc177402829)

[Items 13](#_Toc177402830)

[Features 14](#_Toc177402831)

[Deliverables 14](#_Toc177402832)

[Tasks 14](#_Toc177402833)

[Needs 14](#_Toc177402834)

[Pass/Fail Criteria 15](#_Toc177402835)

[Specifications 15](#_Toc177402836)

[Procedures 17](#_Toc177402837)

[Results 18](#_Toc177402838)

[Introduction 20](#_Toc177402839)

[Installation and Using the Application 21](#_Toc177402840)

[Unzipping the Project 21](#_Toc177402841)

[Opening the Project in Android Studio 21](#_Toc177402842)

[Using the Application 21](#_Toc177402843)

[Vacation Management 22](#_Toc177402844)

[Reports 27](#_Toc177402845)

# Application Design and Testing

## Class Design

In this section, I present the class design for the vacation planner mobile application, which is a crucial aspect of my project's architecture. The class design outlines the structure and interactions of various components within the application, including activities, entities, adapters, database handlers, and broadcast receivers. The provided diagrams illustrate the relationships between these classes, their attributes, and their methods, offering a comprehensive view of how the application's components work together to deliver functionality.

The class design incorporates the following key elements:

**Activities:** Represent the different screens or functionalities within the app, such as **MainActivity**, **LoginActivity,** **ExcursionDetails**, **ReportActivity**, **VacationDetails**, and **VacationList**.

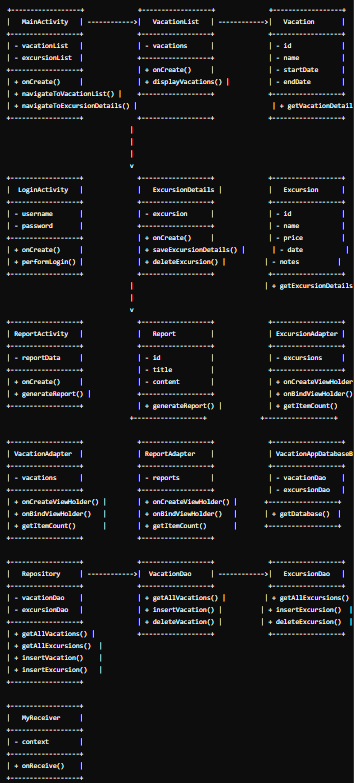
**Entities:** Define the core data models, including **Vacation**, **Excursion**, and **Report**.

**Adapters:** Manage the display and interaction with lists of items, including **ExcursionAdapter**, **VacationAdapter,** and **ReportAdapter**.

**Database and DAO Interfaces:** Handle data persistence and retrieval, represented by **VacationAppDatabaseBuilder, Repository**, **VacationDao**, and **ExcursionDao**.

**Broadcast Receiver**: Handles background tasks and events with **MyReceiver.**

The accompanying diagrams visually represent the class relationships and interactions, providing a clear and organized view of the application's architecture.

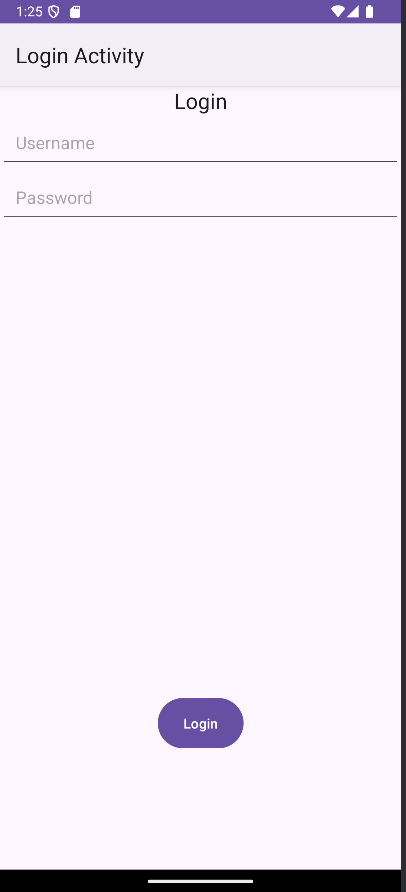


## UI Design

The **vacation planner mobile app**  is designed for simplicity and ease of use, ensuring users can efficiently log in and manage their vacations and excursions. The **User Interface (UI)** focuses on intuitive navigation and functionality for a smooth user experience.

#### Key Features of the UI Design:

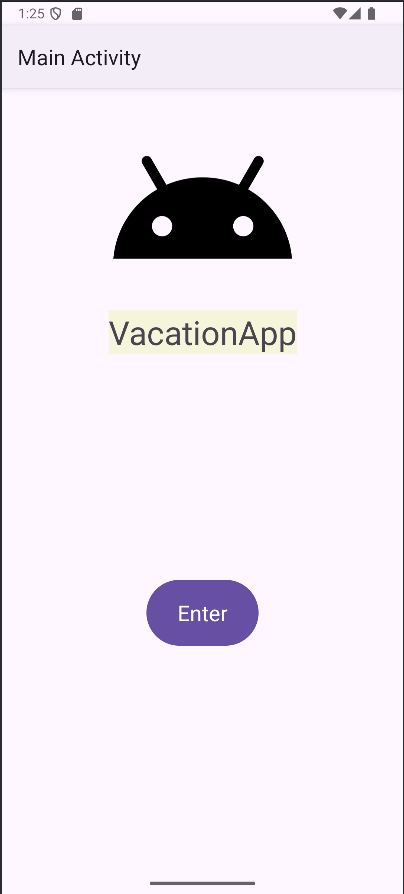
##### ****Login Page****



The login page serves as the initial access point:

* Users enter their **username** and **password** to access the app.
* There is no sign-up option as the app is for **pre-registered users**.
* Incorrect credentials prompt an error message to re-enter details.

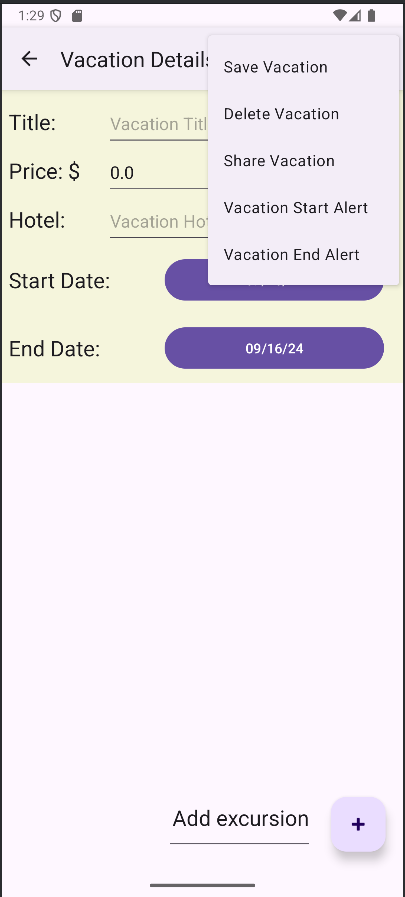
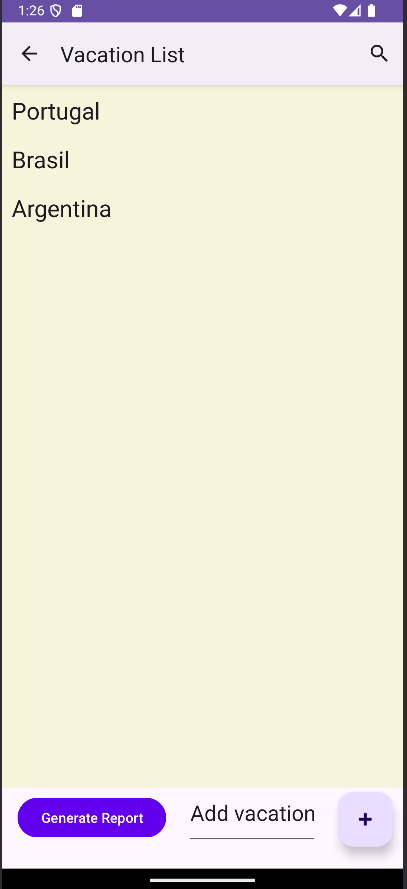
##### 2. ****Main Screen (Post-Login)****



After a successful login, users are redirected to the **main screen**:

* From here, users enter the vacation app and are taken to the **Vacation List**.
* The Vacation List displays all added vacations. If no vacations have been added, users can click on **Add Vacation**.

##### ****Vacation Details****



Clicking **Add Vacation** takes users to the **Vacation Details** page:

* Here, users can input details about their vacation, including title, price, hotel, and dates.
* Options available on this page include **Save**, **Delete**, **Share**, and setting **Start/End Alerts**.
* After saving, users can use the **search bar icon** at the top right of the Vacation List to look up specific vacations.

##### 4. ****Reports****

Users can generate reports by clicking **Generate Report** at the bottom left of the Vacation List:

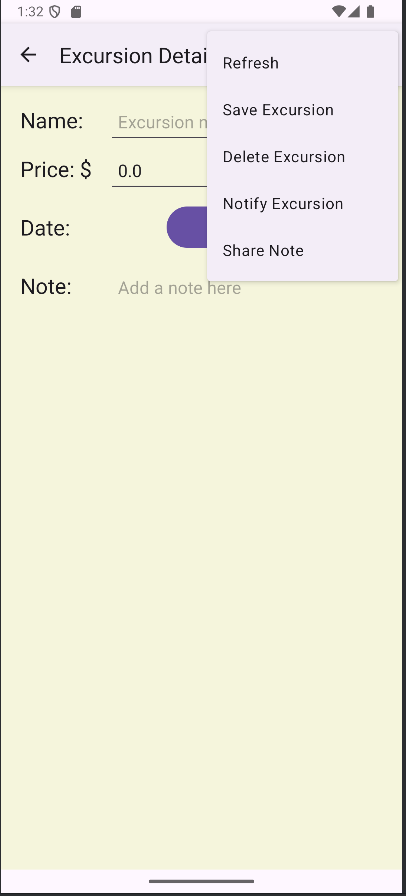
* Reports include the date, vacation name, price, hotel, and start and end dates of each vacation.



##### 5. ****Vacation Management****

* Clicking on a vacation in the list allows users to **edit** it and **add excursions**.
* A menu provides additional options to **Refresh**, **Save**, **Delete**, **Notify Excursion**, or **Share Note**.

This design emphasizes straightforward navigation and effective management of vacations and excursions, ensuring users have all necessary tools for a smooth experience.



# Unit Test Plan

## Introduction

### Purpose

The purpose of this unit test plan is to validate the functionality and reliability of the **Vacation Planner Mobile Application**. A systematic approach to testing was employed to ensure that individual components function correctly in isolation. This process identifies defects and verifies that the application meets quality standards. Remediation strategies were implemented to address any issues discovered during testing.

### Overview

This test plan focuses on critical components of the application, including user authentication and vacation management functionalities. Testing was tailored to address unique aspects of the application and ensure comprehensive coverage.

* **User Authentication:** Tests included verifying login functionality with valid and invalid credentials.
* **Vacation Management:** Covered adding vacation entries.

## Test Plan

### Items

* Development Environment: Android Studio with JUnit testing framework.
* Test Framework: JUnit for unit testing Java code.
* Application Source Code: Java codebase from the GitLab repository.

### Features

* User Authentication: Test user login functionality with both valid and invalid credentials.
* Vacation Management: Test the ability to add vacation entries.

### Deliverables

* Test Scripts: A collection of JUnit test scripts written to validate user authentication and vacation management functionalities.
* Test Results: A report detailing the results of the tests, including successful tests and any failed tests with relevant error messages.

### Tasks

* Set up the development environment with Android Studio and JUnit.
* Ensure test scripts for user authentication and vacation management are in place.
* Prepare test data for validating login and vacation entry functionalities.
* Execute the test scripts and monitor the results.
* Analyze test results and address any failures or unexpected behaviors.

### Needs

* Software Requirements: Android Studio, JUnit testing framework.
* Testing Tools and Libraries: JUnit for unit testing.
* Access to Source Code and Test Scripts: Access to the GitLab repository with the source code and test scripts.

### Pass/Fail Criteria

**User Authentication:**

* **Pass:** User is successfully authenticated with valid credentials.
* **Fail:** User is not authenticated due to incorrect credentials.

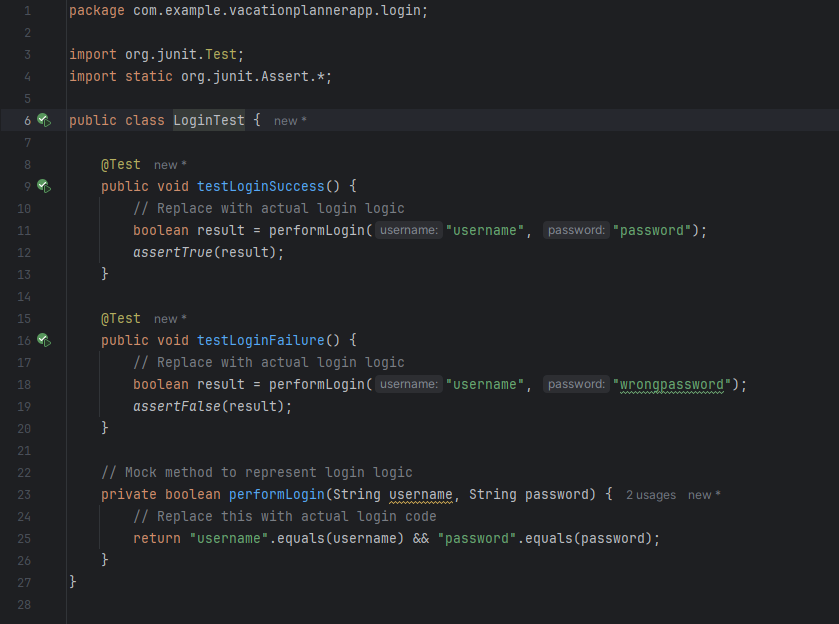
**Vacation Management:**

* **Pass:** New vacation entry is successfully added.
* **Fail:** New vacation entry is not added or results in errors.

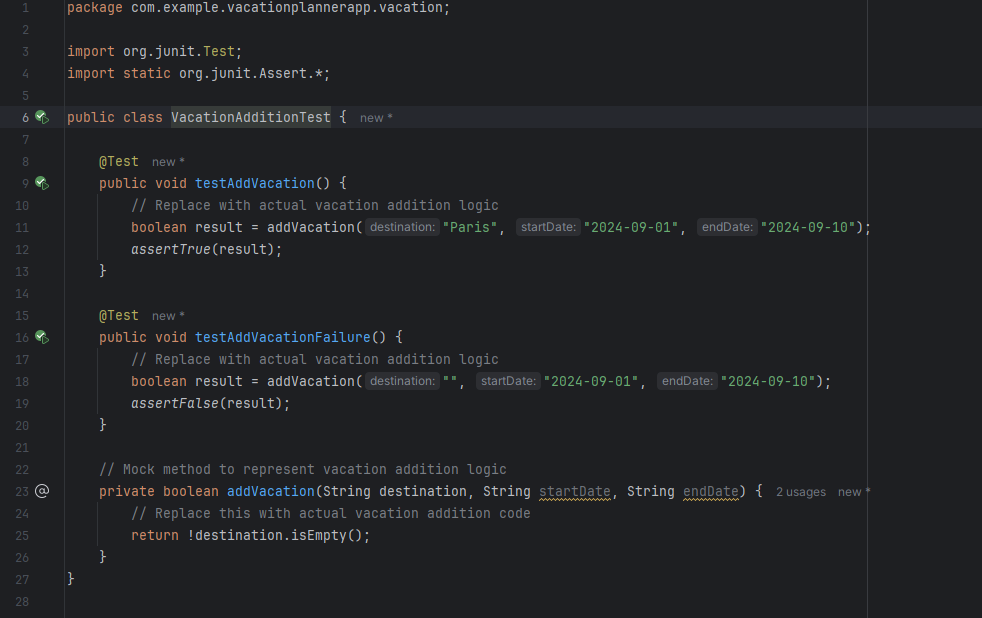
## Specifications

Here are examples of test code used:

**LoginTest.java:**



**VacationAdditionTest.java:**



## Procedures

* **Test Case Preparation:** Identified critical functionalities and features of the application for testing. Created test cases covering various scenarios and defined expected outcomes.
* **Test Environment Setup:** Configured Android Studio and JUnit for testing. Verified that the development environment was properly set up.
* **Test Execution:** Ran the test suite using JUnit and recorded the results. Analyzed results to identify any failures.
* **Test Results Review and Documentation:** Documented test results, including pass/fail outcomes, identified issues, and any modifications made. Updated test plan and test case documentation as needed.

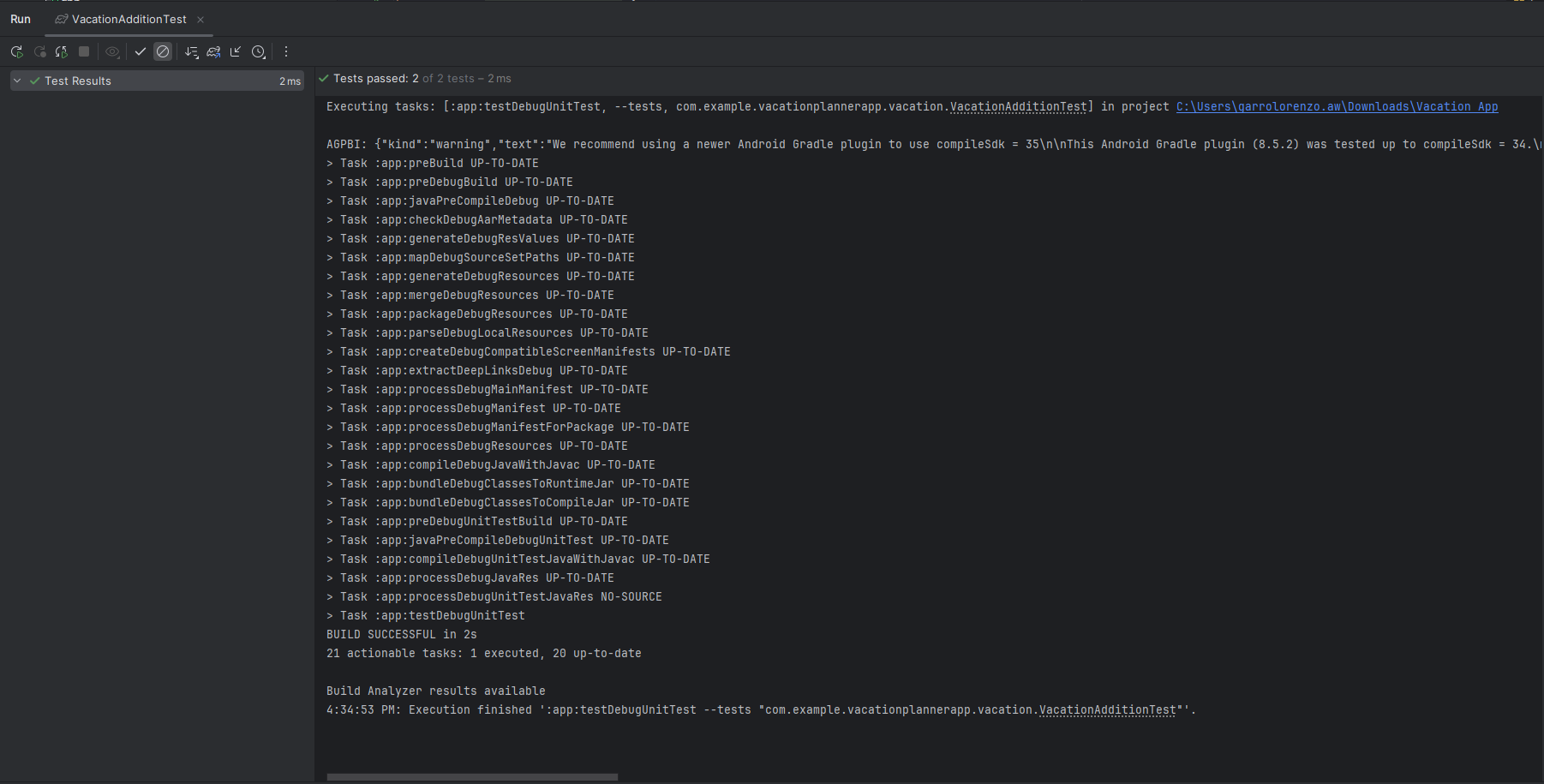
## Results

 **User Authentication:** The test passed when the user was able to successfully log in using valid credentials. The test failed when invalid credentials were used, resulting in an appropriate error message.

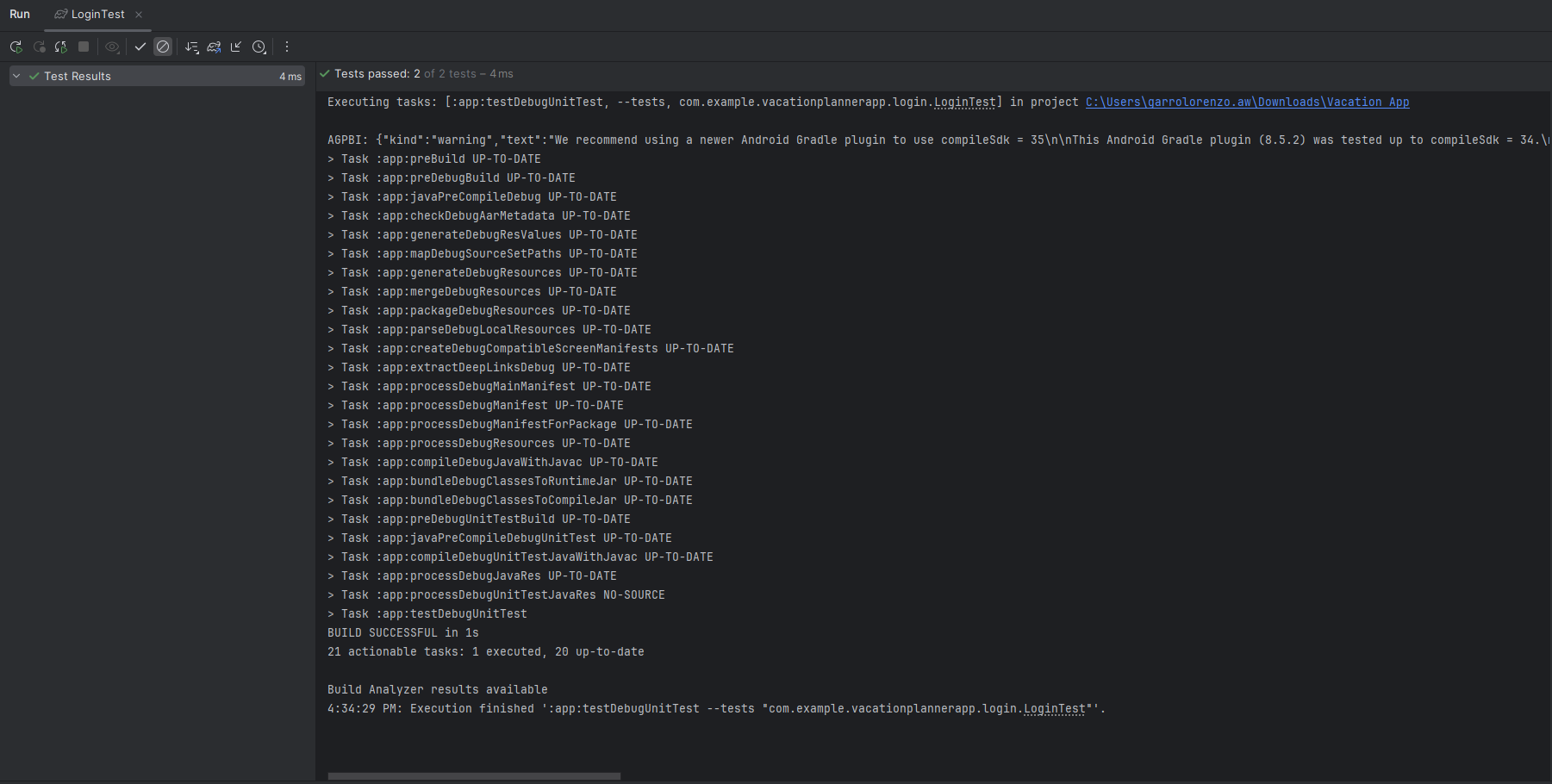
 **Vacation Management:** The test passed when a new vacation entry was successfully added.

If the entry failed to be added, an error was documented and addressed.

* **Vacation Management Pass:**

****

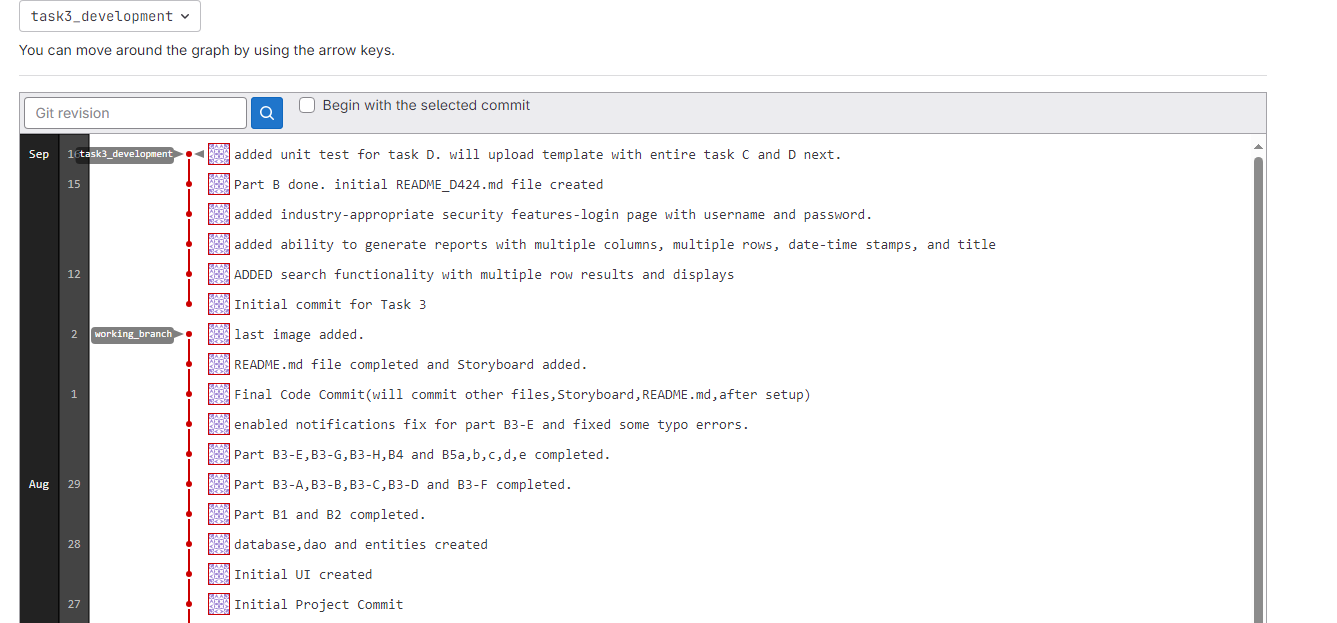
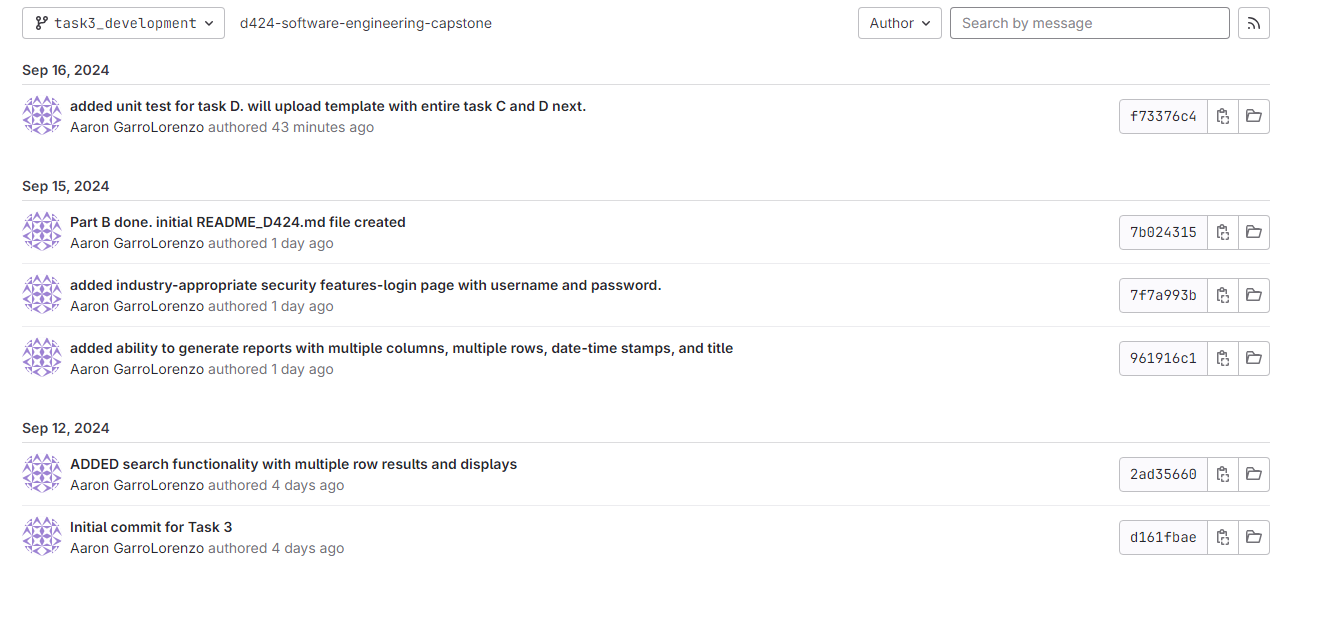
* **User Authentication Pass:**



**Gitlab Repository URL**: [Files · task3\_development · WGU GitLab Environment / Student Repos / agarrol / D424 Software Engineering Capstone · GitLab](https://gitlab.com/wgu-gitlab-environment/student-repos/agarrol/d424-software-engineering-capstone/-/tree/task3_development)

**Branch Name**: **task3\_development**

**Gitlab Branch History:**

****

. **User Guide**

## Introduction

This guide provides detailed steps for setting up, running, and using the Vacation Planner Mobile Application. It includes instructions on how to unzip the project, open it in Android Studio, and interact with the application’s key features, such as login, vacation management, and report generation.

## Installation and Using the Application

### Unzipping the Project

1. **Download the provided ZIP file** from the submission attachment.
2. **Unzip the file**:
   * Right-click on the ZIP file and select **Extract All...**.
   * Choose a destination folder where the project files will be stored and click **Extract**.
   * Ensure the unzipped folder contains the full project directory, including source code, resources, and configuration files.

### Opening the Project in Android Studio

1. **Launch Android Studio**:
   * Open Android Studio from your desktop or start menu.
2. **Open the Project**:
   * In Android Studio, go to **File > Open**.
   * Navigate to the folder where you unzipped the project (d424-software-engineering-capstone).
   * Select the project folder and click **OK** to open it.
3. **Build the Project**:
   * Wait for Android Studio to complete the **Gradle build** process, which may take a few minutes depending on your machine.
4. **Run the Application**:
   * Connect an Android device or start an emulator within Android Studio.
   * Click the green **Run** button (triangle icon) in the toolbar to build and deploy the application.

### Using the Application

#### Login

1. **OPEN THE APPLICATION** on your device or emulator.
2. **LOGIN**:
   * Enter the pre-registered credentials:
     + **USERNAME**: user
     + **PASSWORD**: password
   * Click the **LOG IN** button.
   * If incorrect credentials are used, an error message will prompt you to try again.

A white screen with a black and red screen

Description automatically generated with medium confidence

### Vacation Management

#### Adding a Vacation

1. **Navigate to the Main Screen** after login.
2. **Add Vacation**:
   * Click on **Add Vacation** if no vacations are listed or click the plus icon (+) to add a new vacation.
   * Enter details such as **title**, **price**, **hotel**, and **dates**.
   * Click **SavE VACATION**  to add the vacation to the list.

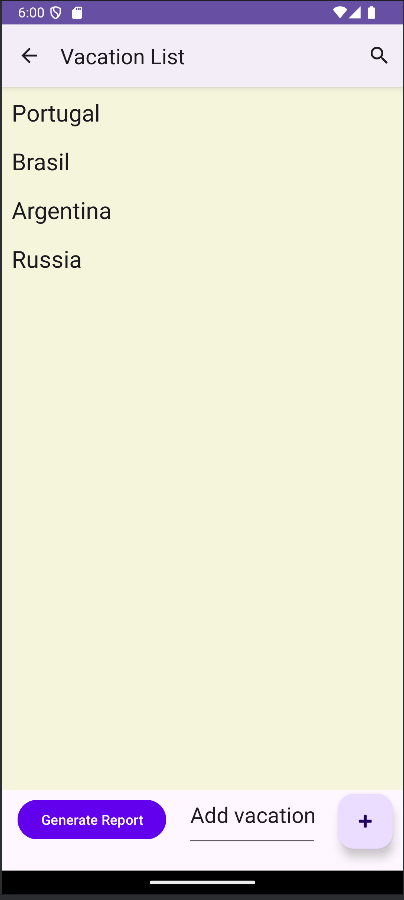
A white rectangular object with black lines

Description automatically generatedA screenshot of a phone

Description automatically generated

#### Viewing and Editing a Vacation

1. **View Vacations**:
   * After adding a vacation, it will appear in the **Vacation List**.
   * Click on any vacation in the list to view its details.



1. **Edit Vacation**:
   * On the **Vacation Details** page, modify any of the fields such as the vacation title, dates, or hotel.
   * Click **Save** to update the vacation information.

A screenshot of a phone

Description automatically generated

If you click on an already saved vacation, you can also **ADD EXCURSIONS**:

1. Click **ADD EXCURSION**.
2. Enter the **Excursion Name**, **Price**, **Date**, and **Note**.
3. Click **SAVE EXCURSION** to save the excursion details.

A screenshot of a phone

Description automatically generated

#### Setting Alerts

1. **Start and End Alerts**:
   * On the **Vacation Details** page, you can set alerts for the vacation’s start and end dates by clicking on the **Set Alerts** button.
   * Choose your preferred notification times.

A screenshot of a phone

Description automatically generated

A screenshot of a phone

Description automatically generated

### Reports

1. **Generate Report**:
   * On the **Vacation List** page, click **Generate Report** at the bottom left to create a vacation report.

A white rectangular object with black text

Description automatically generated

* + The report will include vacation names, prices, hotels, and start/end dates.

