|  |  |  |
| --- | --- | --- |
|  | | |
| BuNest App  Software Engineering Project Report | | |
| MOEEZ MUSTAFA (01-134212-088) |  | MUJADIL HAZQEEL (01-134212-177) |

**ABSTRACT**

Our project introduces an innovative mobile application, "BuNest" designed as an Airbnb-inspired platform built using React Native. BuNest aims to streamline the reservation process for university classrooms, facilitating the scheduling of both regular lectures and makeup sessions.

The application offers a user-friendly interface, enabling professors, students, and administrators to effortlessly browse, select, and reserve available classrooms on their university campus. Leveraging the power of React Native, BuNest ensures a seamless and responsive cross-platform experience for users on iOS and Android devices.

Key Features:

1. Classroom Listings:

2. Reservation Management:

3.User Profiles:

4. Real-Time Updates:

BuNest aims to enhance the efficiency of university resource utilization, providing a centralized platform for managing classroom reservations. By leveraging the flexibility of React Native, this application ensures a consistent and engaging experience for users, promoting effective communication and collaboration within the academic community.

**TABLE OF CONTENT:**

|  |  |  |
| --- | --- | --- |
| Sno | Name | Page |
| 1 | Problem Statement | 3 |
|  | Process Model | 3 |
|  | Introduction | 4 |
|  | SRS | 5 |
|  | Context Level diagram | 7 |
|  | Use Case | 8 |
|  | ER diagram | 12 |
|  | Sequence diagram | 12 |
|  | Activity Diagram | 13 |
|  | Timeline chart | 14 |
|  | Screenshots | 14 |
|  | Risk Analysis | 17 |
|  | Testing | 19 |
|  | Conclusion | 25 |

**PROBLEM STATEMENT**

Classroom reservation often involve cumbersome administrative processes, leading to scheduling conflicts, miscommunications, and underutilization of available spaces. This issue becomes more pronounced when accommodating both regular lectures and makeup sessions, requiring a dynamic and user-friendly solution.

Challenges:

Inefficient Reservation Systems: Existing methods of reserving university classrooms are often manual, time-consuming, and prone to errors, resulting in scheduling conflicts that disrupt the academic calendar.

Lack of Real-time Information: Current systems may not provide real-time updates on classroom availability, making it difficult for professors and students to find suitable spaces for impromptu makeup sessions or supplementary classes.

Communication Gaps: There is a lack of centralized communication channels for professors, students, and administrators to coordinate and share information about classroom reservations, leading to confusion and overlapping bookings.

Underutilization of Resources: Many classrooms remain unused during specific time slots, while others are overbooked, highlighting the need for a more balanced and optimized allocation of university resources.

**PROCESS MODEL**

BuNest App follows **INCREMENTAL** model because initially software requirements are well defined, but the overall scope of development effort is a purely linear process. There may be other requirements of the user which will be known later. So, those requirements can the implemented and delivered in the following next increments. Our project is a short-term project of 2 months and team members consist only two people.

**Introduction**

**Purpose:**

BuNest is primarily developed for **Google Cloud** **LHR** and **Ejad labs.**Purpose of App is providing an efficient and user-friendly platform for reserving university classrooms, focusing on both regular lectures and makeup sessions. The system aims to streamline the reservation process, optimizing the utilization of university resources and minimizing scheduling conflicts. By fostering improved communication among professors, students, and administrators, BuNest enhances the overall user experience with features such as real-time updates, and personalized user profiles.

**Scope:**

BuNest is developed with the primary purpose of providing an efficient and user-friendly platform for reserving university classrooms, focusing on both regular lectures and makeup sessions. The system aims to streamline the reservation process, optimizing the utilization of university resources and minimizing scheduling conflicts. By fostering improved communication among professors, students, and administrators, BuNest enhances the overall user experience with features such as real-time updates, and personalized user profiles

**SYSTEM REQUIREMENT SPECIFICATION**

**Functional Requirements:**

**User Registration and Authentication:**

Users should be able to create accounts with a valid university email address.

The system should authenticate users securely, ensuring data privacy.

**Responsive Design:**

The UI should be responsive, providing a seamless experience across different devices, including desktops, tablets, and mobile phones.

**Classroom Listings:**

The platform should display a comprehensive catalog of available university classrooms.

Classroom details should include capacity, available equipment, and location information.

**Reservation Management:**

Users (professors, students, and administrators) should be able to browse and reserve classrooms for regular lectures and makeup sessions.

A calendar view should allow users to identify and select available time slots.

**Real-Time Updates:**

The system should provide real-time updates on classroom availability to prevent conflicts.

Users should receive notifications for reservation confirmations, changes, or cancellations.

**User Profiles:**

Users should have the ability to create, update, and manage their profiles.

User profiles should display reservation history and preferences.

**Non-Functional Requirements:**

**Performance:**

The system should provide a responsive user experience with minimal latency.

It should handle a concurrent user load equivalent to the peak usage times at the university.

**Security:**

User authentication and data transfer should be encrypted to ensure the security of user information.

The system should have mechanisms in place to prevent unauthorized access.

**Scalability:**

The platform should be designed to scale with growing user and classroom data.

It should accommodate future enhancements without significant reengineering.

**Compatibility:**

The application should be compatible with major web and mobile devices (iOS, Android).

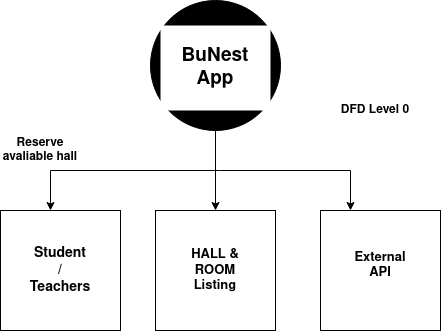
**Regulatory Compliance:**

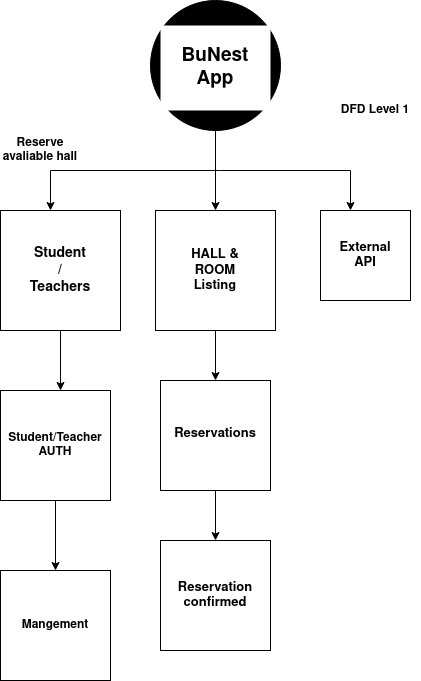
The system should adhere to data protection regulations and university policies.

It should comply with relevant legal and ethical standards.

**DATA FLOW DIAGRAM:**

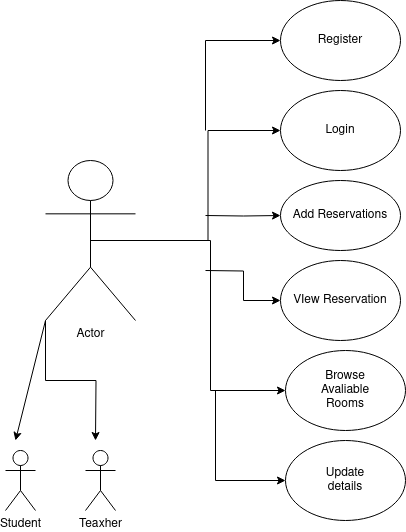
**Context Level Diagram:**

**Level 0 DFD**

**DFD Level 1**

**USE CASE**

**Diagram:**

 **use case diagram**

**Use Case Description:**

**Register:**

Use Case ID: UC-01

Use Case Name: Register

Actors:

User (Guest)

Description: The "Register" use case (UC-01) represents the process through which a new user (Guest) creates an account on the BuNest App. This use case involves providing necessary information for account creation and includes validation steps to ensure accurate and secure registration.

Preconditions:

The user is not currently registered in the BuNest system.

The user accesses the registration page.

Basic Flow:

The user accesses the registration page on the BuNest system.

The system prompts the user (Guest) to provide necessary information, including a valid university email address, a password, and user type selection (professor, student, or administrator).

The user enters the required information and submits the registration form.

The system validates the provided information, checking for the correctness of the email format, password strength, and user type selection.

If the information is valid, the system creates a new user account and sends a verification email to the provided university email address.

The user receives the verification email and clicks on the provided link to confirm and activate the account.

After successful verification, the system confirms the registration (UC-01) and redirects the user to the login page.

Alternative Flow:

If the provided information is not valid (e.g., invalid email format, weak password), the system prompts the user to correct the errors and resubmit the registration form.

Postconditions:

A new user account is created in the BuNest system.

Exceptional Conditions:

If the user does not Register, the system allows them to contact support.

**Add Reservation**

Use Case ID: UC-03

Use Case Name: Reserve Classroom

Actors:

User (Professor or Student)

Description: The "Reserve Classroom" use case (UC-03) involves the process through which a registered user (Professor or Student) reserves a university classroom for either a regular lecture or a makeup session. This use case enables users to browse available classrooms, select a suitable time slot, and confirm their reservation.

Preconditions:

The user is registered and logged into the BuNest App.

The user has navigated to the reservation section of the system.

Basic Flow:

The user logs into the BuNest system and navigates to the reservation section.

The system displays a list of available classrooms with details such as capacity, equipment, and location.

The user browses the available classrooms and selects a specific classroom for reservation.

The user chooses a date and time for the reservation from an interactive calendar view.

The system validates the selected date and time for availability, ensuring there are no scheduling conflicts.

If the selected slot is available, the user confirms the reservation.

The system updates the database with the reservation details, marking the selected classroom as booked for the specified date and time.

The user receives a confirmation notification with the reservation details.

Alternative Flow:

If the selected slot is not available due to a scheduling conflict, the system notifies the user and prompts them to choose an alternative date or time.

Postconditions:

The selected classroom is reserved for the specified date and time.

The user receives a confirmation notification.

Exceptional Conditions:

If the user encounters an issue during the reservation process, they can contact customer support for assistance.

**Browse Room:**

Use Case ID: UC-05

Use Case Name: Browse/Select Properties

Actors:

User (Professor or Student)

Description: The "Browse/Select Properties" use case (UC-05) involves the process through which a registered user (Professor or Student) explores available university classrooms, reviews their details, and selects a specific classroom for potential reservation within the BuNest system.

Preconditions:

The user is registered and logged into the BuNest system.

The user has navigated to the section for browsing and selecting available classrooms.

Basic Flow:

The user logs into the BuNest system and navigates to the section for browsing and selecting available classrooms.

The system displays a comprehensive catalog of university classrooms, highlighting details such as capacity, available equipment, and location.

The user utilizes search and filter functionalities to narrow down the list of available classrooms based on specific criteria.

The system presents the user with a list of filtered classrooms, each accompanied by relevant information.

The user reviews the details of various classrooms, including capacity, equipment, and location, to make an informed decision.

The user selects a specific classroom of interest for potential reservation.

Alternative Flow:

If the user has a specific classroom in mind, they may use a direct search option to locate and select that classroom.

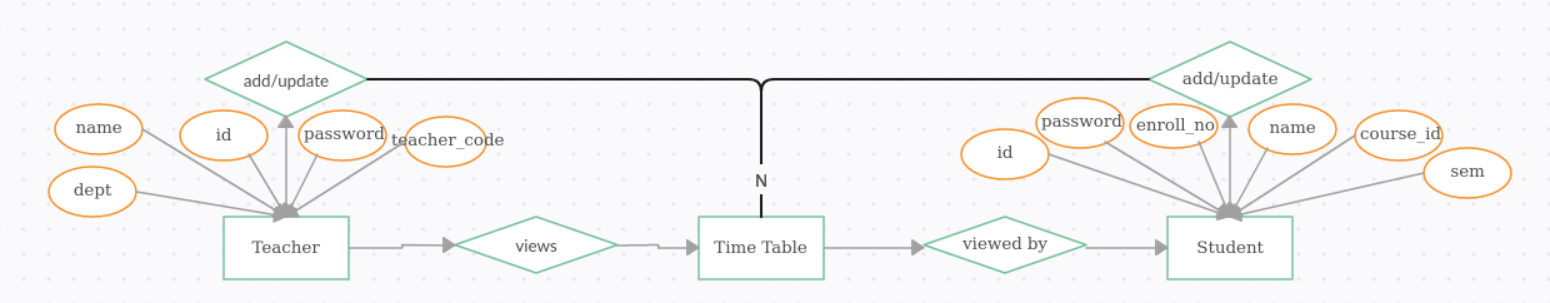
Postconditions:

The user has reviewed available classrooms and selected a specific classroom of interest.

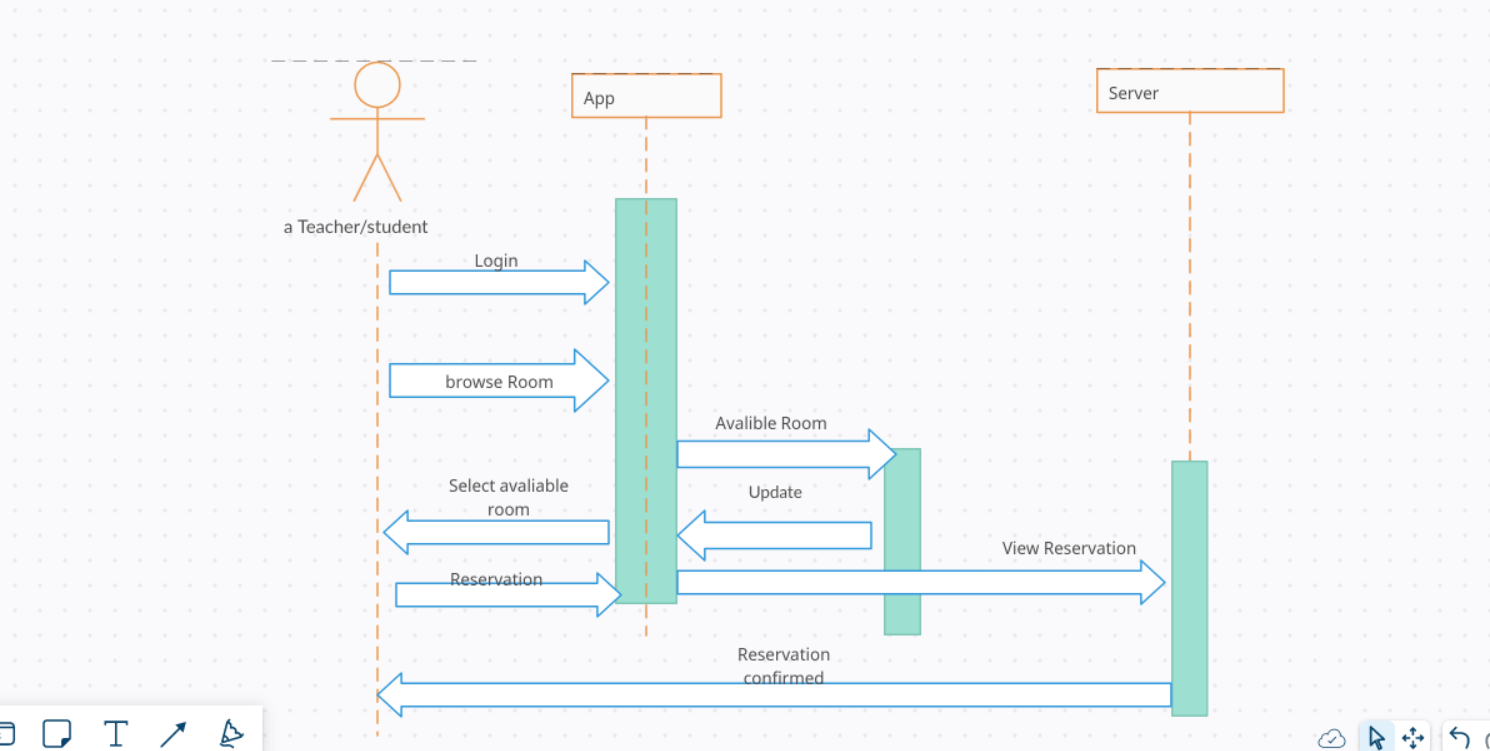
Exceptional Conditions:

If the user encounters difficulties in finding suitable classrooms, the system helps through search and filter functionalities or prompts the user to contact customer support.

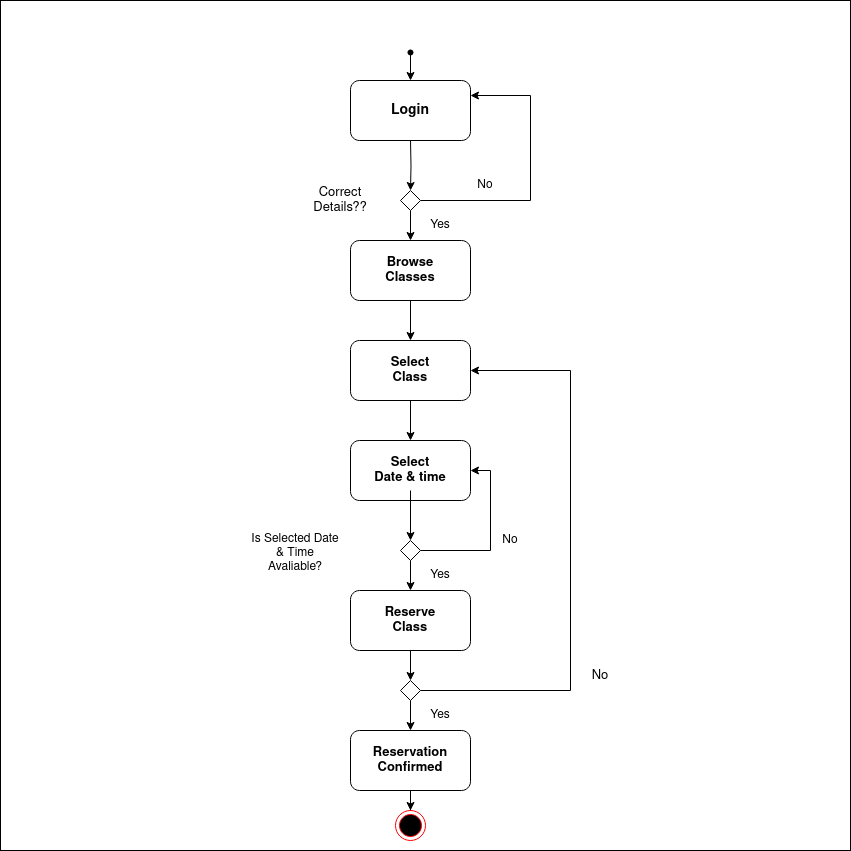
**ER DIAGRAM:**



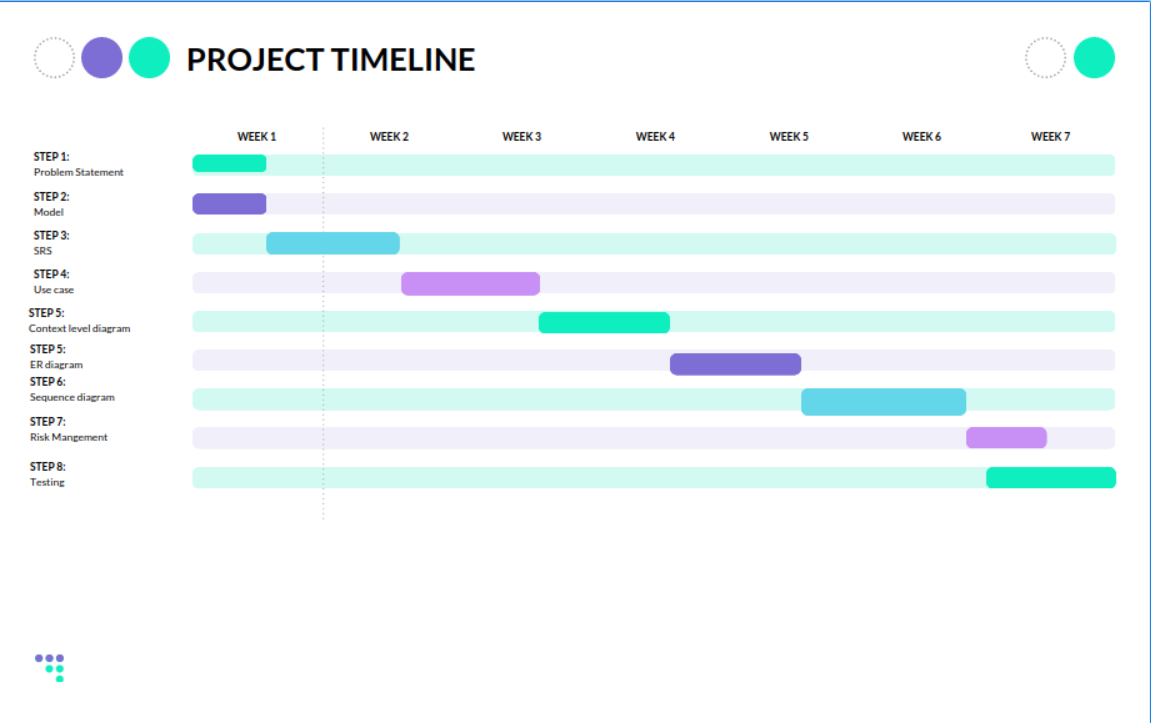
**SEQUENCE DIAGRAM**



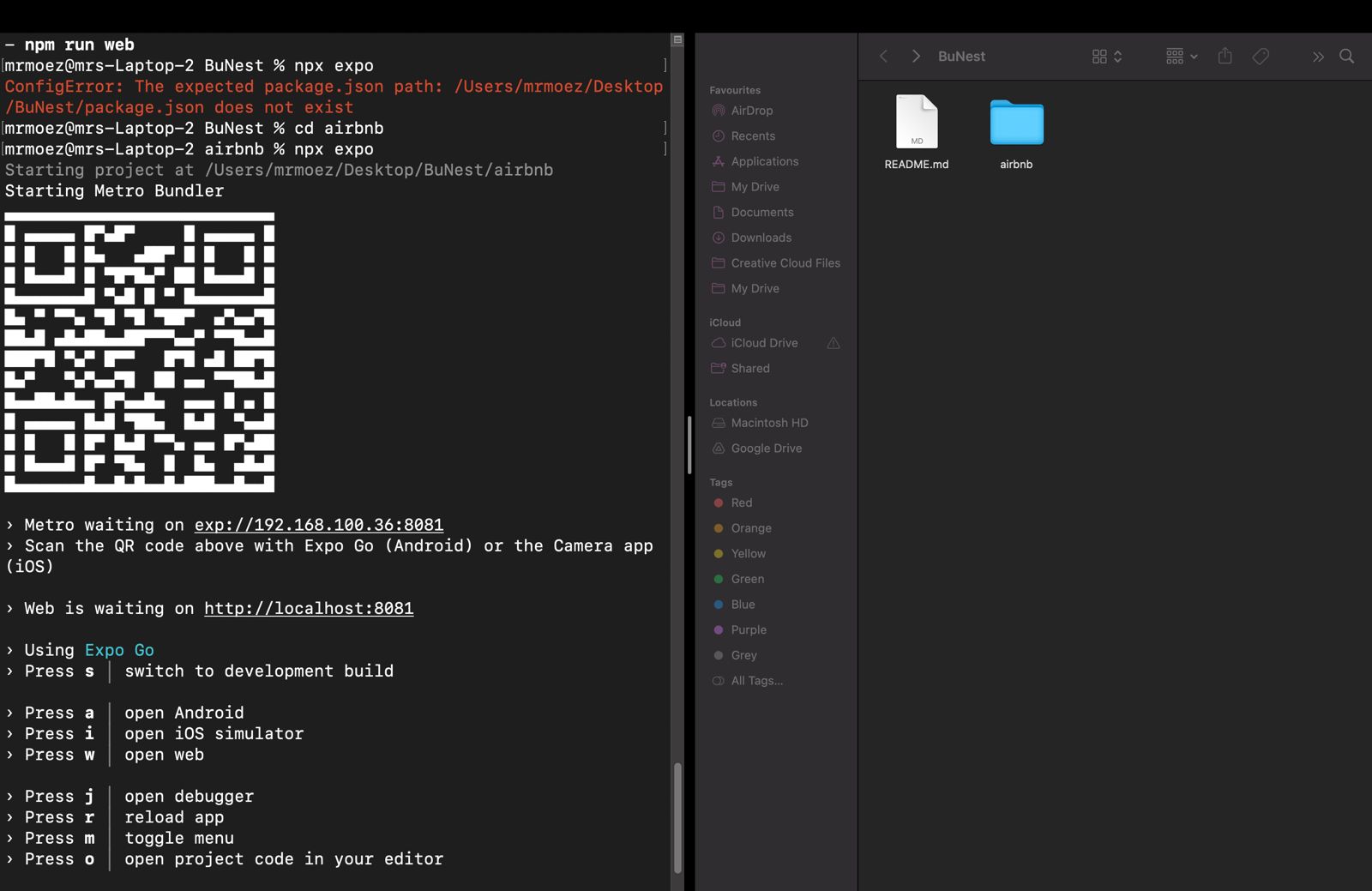
**ACTIVITY DIAGRAM**



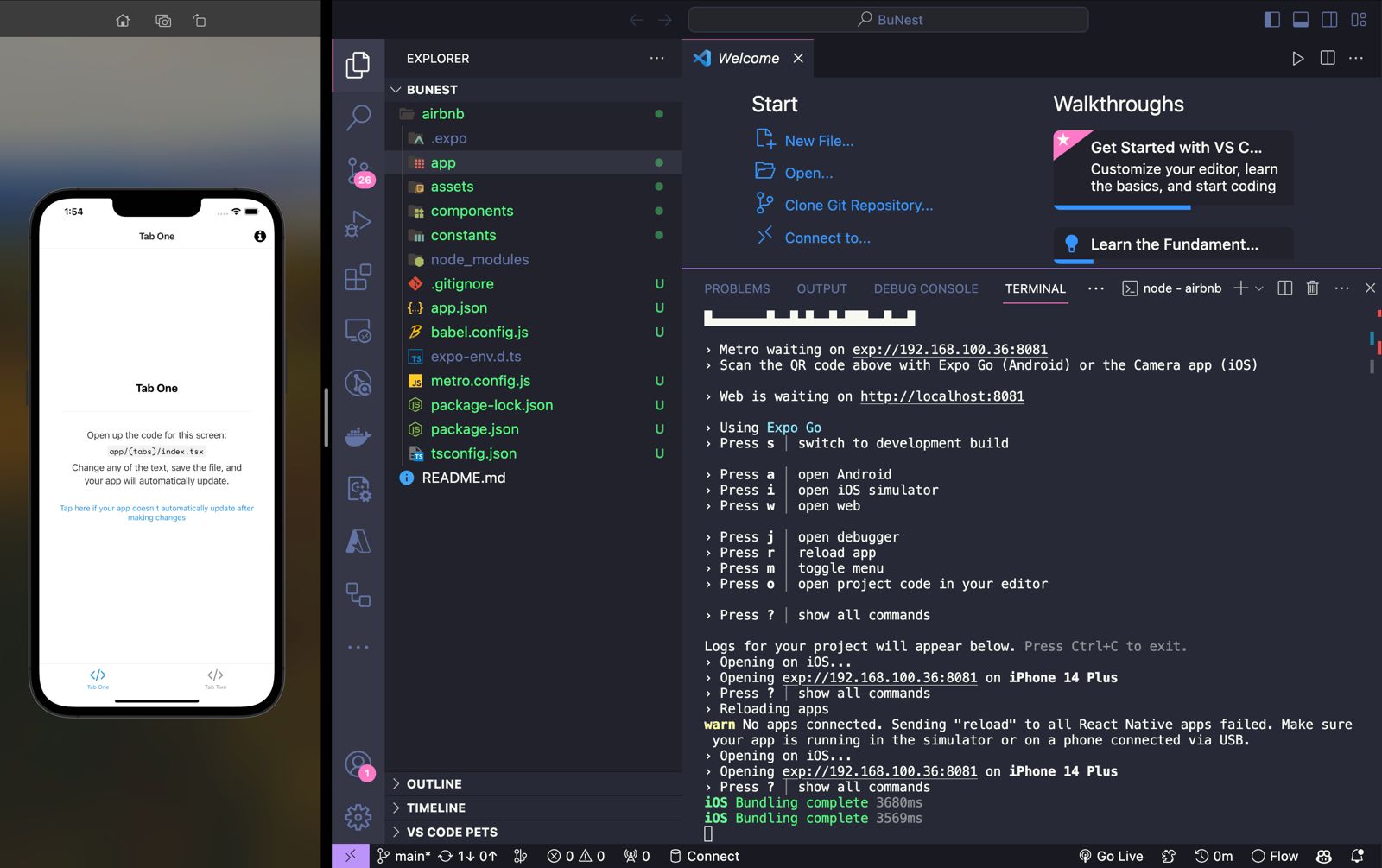
**TIMELINE CHART**



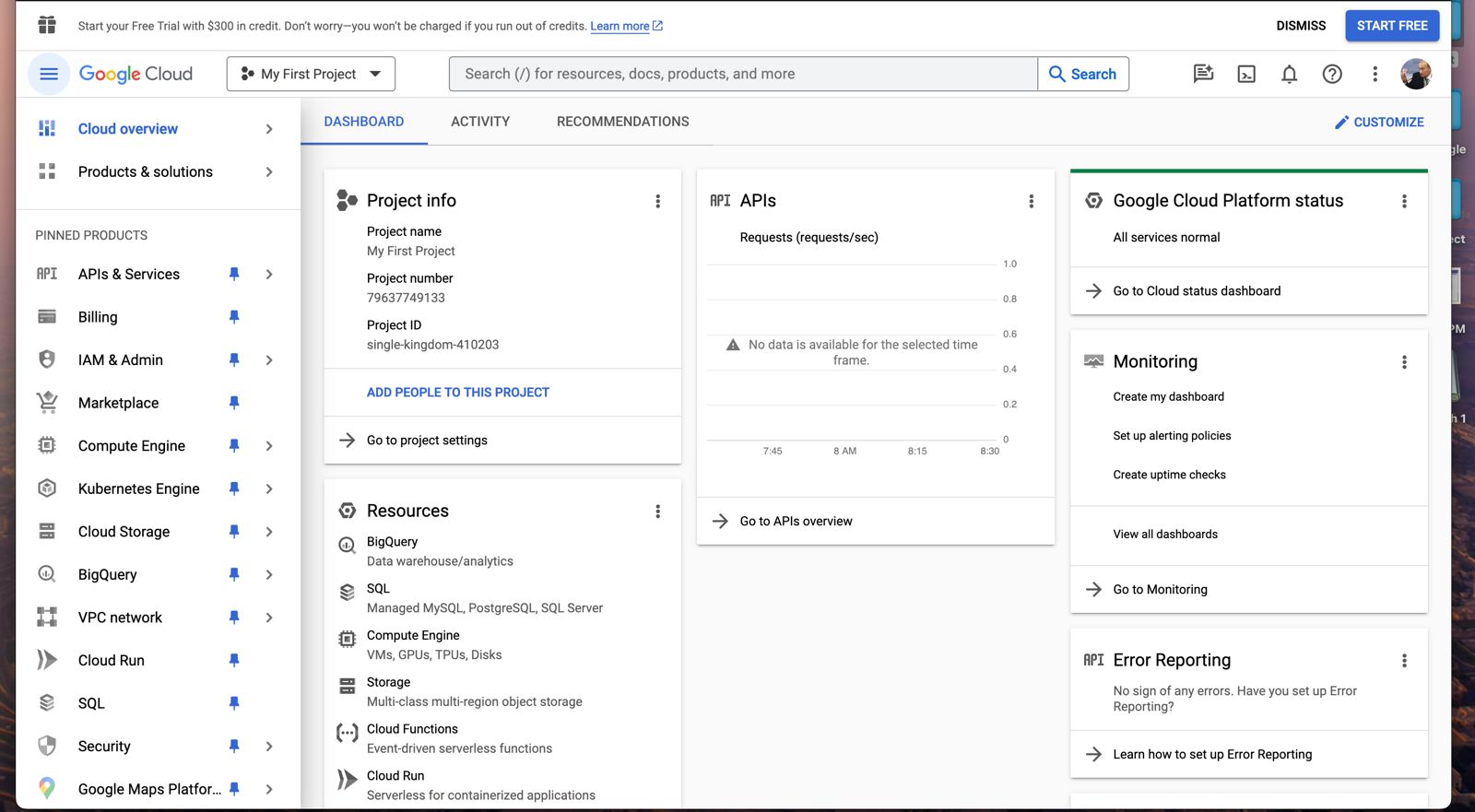
**SCREENSHOTS**



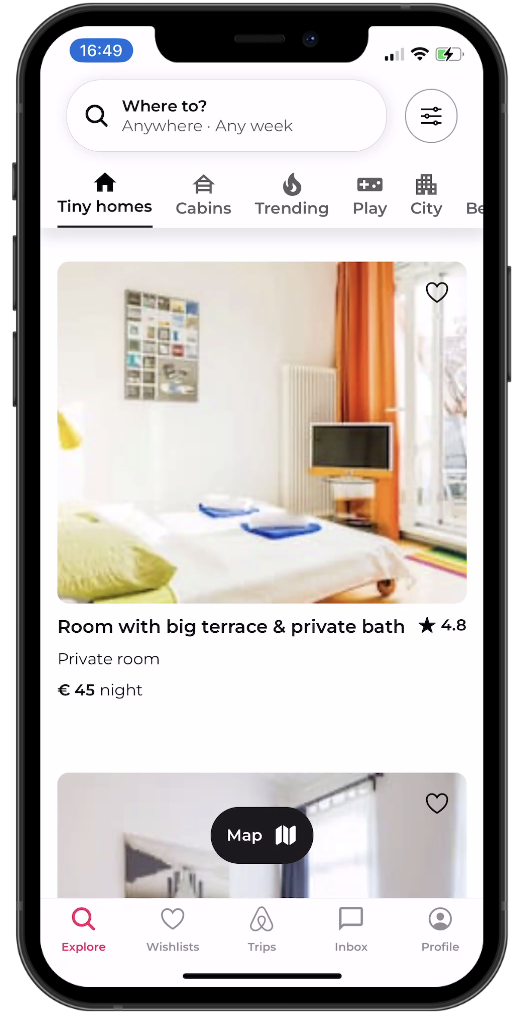
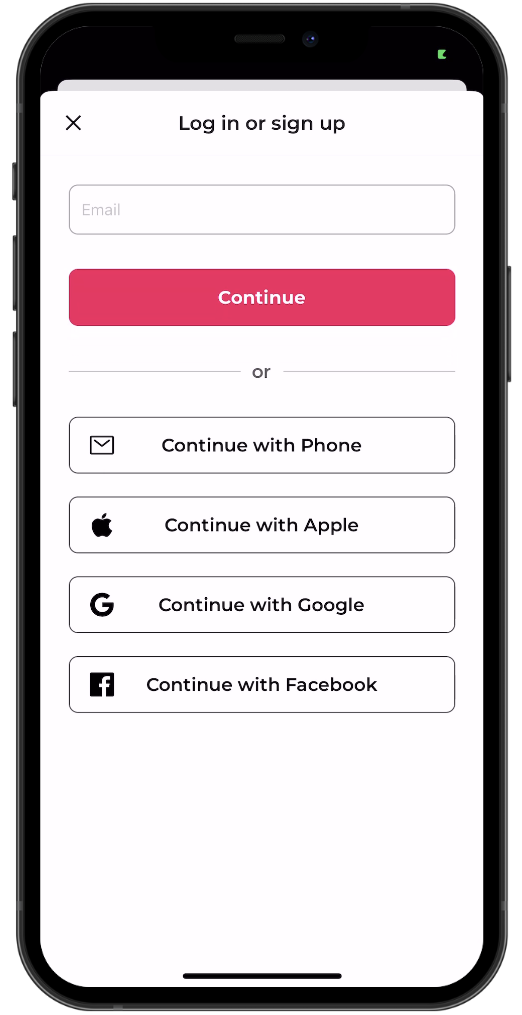
Expo (For cross platform App)



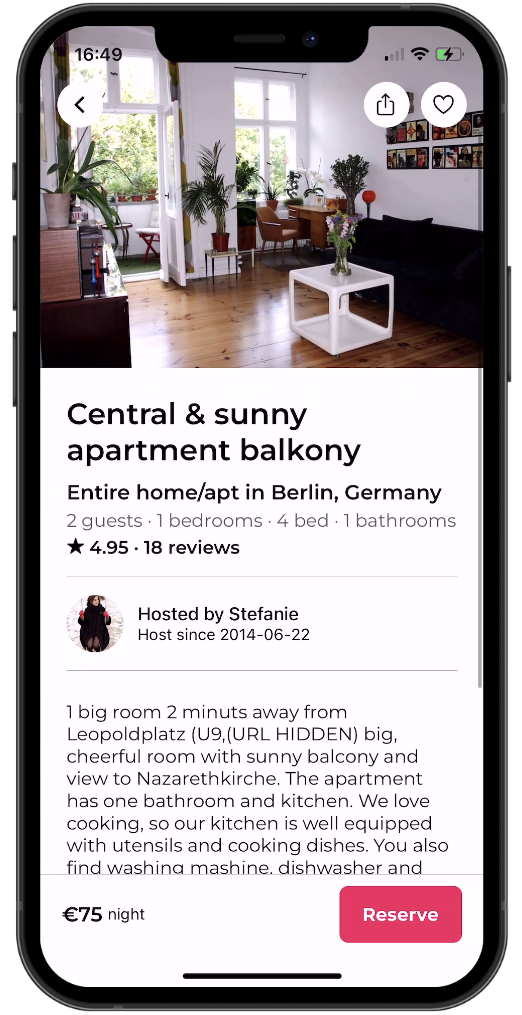
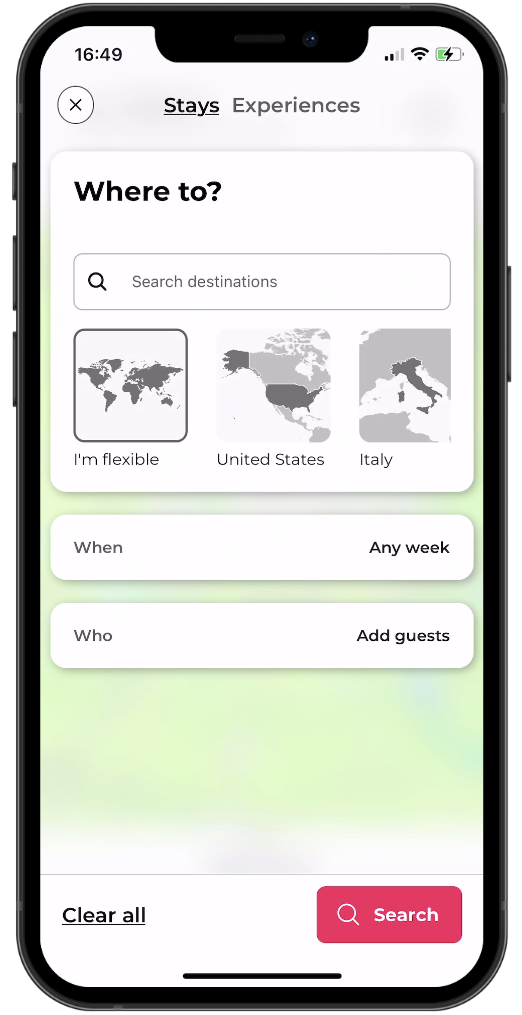
Intial UI (platform choose to run app)



Google Console (API)



Login Main

Resevation tab Date & time

**RISK MANGEMENT**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **S.No** | **Risk** | **Probability** | **Impact** | **RMMA Plan** |
| 1 | Team member illness | 35% | Modrate | Other person does the work or delay 1 to 2 days |
| 2 | Data loss | 5% | High | Github commits |
| 3 | Changes in university policies affecting classroom reservations | 25% | Modrate | Have a rapid development and update plan in case of policy changes. |
| 4 | Dissatisfaction with the UI | 10% | Low | Discussions |
| 5 | Deadline | 15% | Low | Get help |

**PSEDUO CODE (page view)**

import { View } from 'react-native';

import React, { useMemo, useState } from 'react';

import ListingsBottomSheet from '@/components/ListingsBottomSheet';

import listingsData from '@/assets/data/airbnb-listings.json';

import ListingsMap from '@/components/ListingsMap';

import listingsDataGeo from '@/assets/data/airbnb-listings.geo.json';

import { Stack } from 'expo-router';

import ExploreHeader from '@/components/ExploreHeader';

const Page = () => {

const items = useMemo(() => listingsData as any, []);

const getoItems = useMemo(() => listingsDataGeo, []);

const [category, setCategory] = useState<string>('Tiny homes');

const onDataChanged = (category: string) => {

setCategory(category);

};

return (

<View style={{ flex: 1, marginTop: 80 }}>

{/\* Define pour custom header \*/}

<Stack.Screen

options={{

header: () => <ExploreHeader onCategoryChanged={onDataChanged} />,

}}

/>

<ListingsMap listings={getoItems} />

<ListingsBottomSheet listings={items} category={category} />

</View>

);

};

export default Page;

Full Code Is available on github:[BuNest](https://github.com/moezmustafa/BuNest)

**TESTING**

App Testing is a method to check whether the actual software product matches expected requirements and to make sure that the App product is bug free. It includes execution of App components to evaluate properties. The purpose of software testing is to identify errors, or missing requirements Software

The project team does manual testing for our system, which is a process of testing software manually, without using automated tools. In this process, a tester manually executes test cases and checks the software for defects, errors, and other issues that can affect the software's quality and functionality. Manual testing can be performed at different levels of software testing, such as unit testing, integration testing, system testing, and acceptance testing.Manual testing is a critical part of the software development life cycle, and it is often used complement automated testing.

**TEST CASES:**

**Test Case 1**: Verify User Login with Valid Credentials

Test Case ID: BU-TC-001

Test Objective:

To verify that a user can log in with valid credentials in the BuNest app.

Test Type: Unit Testing

Input:

Username: user123

Password: pass@123

Expected Result:

The user is now logged in.

The user is directed to the home page.

Actual Result: Passed

**Test Case 2**: Verify User Login with Invalid Username

Test Case ID: BU-TC-002

Test Objective:

To verify that a user cannot log in with an invalid username in the BuNest app.

Test Type: Unit Testing

Input:

Username: invalidUser

Password: pass@123

Expected Result:

The login attempt fails.

An error message is displayed, indicating invalid credentials.

Actual Result: Passed

**Test Case 3**: Verify User Login with Invalid Password

Test Case ID: BU-TC-003

Test Objective:

To verify that a user cannot log in with an invalid password in the BuNest app.

Test Type: Unit Testing

Input:

Username: user123

Password: wrongPass

Expected Result:

The login attempt fails.

An error message is displayed, indicating invalid credentials.

Actual Result: Passed

**Test Case 4**: Verify User Login with Empty Username

Test Case ID: BU-TC-004

Test Objective:

To verify that a user cannot log in with an empty username in the BuNest app.

Test Type: Unit Testing

Input:

Username:

Password: pass@123

Expected Result:

The login attempt fails.

An error message is displayed, indicating the need for a valid username.

Actual Result: Passed

**Test Case 5**: Verify Successful Reservation of a Hall

Test Case ID: BU-TC-005

Test Objective:

To verify that a user can successfully reserve a hall in the BuNest app.

Test Type: Unit Testing

Input:

Select a hall with available slots.

Choose a valid date and time for reservation.

Expected Result:

The hall is successfully reserved.

The user receives a confirmation message.

The reserved hall is marked as unavailable for the selected date and time.

Actual Result: Passed

**Test Case 6**: Verify Reservation with Invalid Date

Test Case ID: BU-TC-006

Test Objective:

To verify that a user cannot reserve a hall with an invalid date in the BuNest app.

Test Type: Unit Testing

Input:

Select a hall with available slots.

Choose a past date for reservation.

Expected Result:

The reservation attempt fails.

An error message is displayed, indicating an invalid date.

Actual Result: Passed

**Test Case 7**: Verify Reservation with Invalid Time

Test Case ID: BU-TC-007

Test Objective:

To verify that a user cannot reserve a hall with an invalid time in the BuNest app.

Test Type: Unit Testing

Input:

Select a hall with available slots.

Choose a time outside the available slots.

Expected Result:

The reservation attempt fails.

An error message is displayed, indicating an invalid time.

Actual Result: Passed

**Test Case 8**: Verify Successful User Registration

Test Case ID: BU-TC-008

Test Objective:

To verify that a new user can successfully register an account in the BuNest app.

Test Type: Unit Testing

Input:

New username: newUser123

New password: securePass@456

Email: newuser@example.com

Full Name: New User

Expected Result:

The user registration is successful.

The user is redirected to the login page.

A confirmation email is sent to the provided email address.

Actual Result: Passed

**Test Case 9**: Verify Registration with Existing Username

Test Case ID: BU-TC-009

Test Objective:

To verify that a user cannot register an account with an existing username in the BuNest app.

Test Type: Unit Testing

Input:

Existing username: exiUser

New password: newPass@789

Email: newuser@example.com

Full Name: Another User

Expected Result:

The registration attempt failed.

An error message is displayed, indicating that the username is already in use.

Actual Result: Passed

**Test Case 10**: Verify Registration with Existing Email

Test Case ID: BU-TC-010

Test Objective:

To verify that a user cannot register an account with an existing email address in the BuNest app.

Test Type: Unit Testing

Input:

New username: newUser456

New password: strongPassword@123

Email: existinguser@example.com

Full Name: New Name

Expected Result:

The registration attempt fails.

An error message is displayed, indicating that the email address is already in use.

Actual Result: Passed

**Test Case 11**: Verify Registration with Empty Fields

Test Case ID: BU-TC-011

Test Objective:

To verify that a user cannot register an account with empty required fields in the BuNest app.

Test Type: Unit Testing

Input:

New username: ""

New password: ""

Email: ""

Full Name: ""

Expected Result:

The registration attempt fails.

Error messages are displayed, indicating the need to fill in all required fields.

Actual Result: Passed

**CONCLUSION**

The BuNest project has been an ambitious endeavor, aiming to provide a comprehensive solution for managing university classrooms and hall reservations. Throughout the project lifecycle, various aspects have been addressed, including system design, development, and testing. Here's a comprehensive conclusion that encapsulates the key components of the project:

System Architecture and Design:

The architectural foundation of BuNest was meticulously planned, leveraging modern technologies such as React Native, Expo, and Cleck for a cross-platform, user-friendly, and secure application. The system's design prioritized intuitive user interfaces, robust authentication mechanisms, and efficient reservation workflows.

Development:

The development phase was marked by collaborative efforts, with a dedicated team implementing features such as user registration, authentication, reservation functionality, and viewing reservations. The use of incremental models allowed for iterative development, ensuring that each module was refined and integrated seamlessly into the system.

Testing and Quality Assurance:

The testing phase was comprehensive, covering user authentication, reservation processes, and overall system performance. Test cases were designed to validate functionality, security, and user experience. The results indicated a robust and reliable application, with identified issues promptly addressed to meet high-quality standards.

Incremental Model and Agile Methodology:

The project's adoption of an incremental model facilitated a systematic approach to development, enabling the team to prioritize features and respond to evolving requirements. The Agile methodology further ensured flexibility and adaptability, fostering collaboration between team members and stakeholders throughout the project lifecycle.

User Experience:

A key focus of the project was enhancing the user experience. The BuNest app provides an intuitive platform for users to effortlessly reserve classrooms or halls, view reservations, and manage their academic activities. The user interfaces were designed with clarity and simplicity to cater to a diverse user base within university settings.

Future Recommendations:

While the BuNest app has achieved a commendable level of functionality and reliability, continuous improvement is essential. Future iterations could explore additional features such as notifications, analytics for space utilization, and integration with university calendars. User feedback and iterative updates would contribute to the app's ongoing success.

Overall Impact:

The BuNest project holds the potential to significantly impact university operations, providing an efficient and modernized approach to classroom and hall reservations. By streamlining reservation processes and enhancing user accessibility, the app stands as a valuable tool for both students and faculty members, contributing to an optimized learning environment.

In conclusion, the BuNest project represents a successful integration of technology into the academic domain. The collaborative efforts of the development team, adherence to best practices, and a commitment to user experience have resulted in a robust and functional application. As the BuNest app moves toward deployment, it stands poised to make a positive contribution to university life, offering a sophisticated solution for managing academic spaces.