



**Faculty of Information Technology  
Future University - Sudan**  
**Khartoum-Sudan**

**Hotels Portal Reservation and Management System  
Mobile Application for The Ministry of Tourism Sudan**

A Graduation Project Presented to the  
**FACULTY OF INFORMATION TECHNOLOGY**  
In Partial Fulfillment of the Requirements for the Degree of  
**BACHELOR OF SCIENCE (HONS) IN INFORMATION TECHNOLOGY**

By

**Mohammed Jamal Abdalla** **202003073**

**Mozan Abdelsamie Mohamed** **202003115**

Project Supervisor

**Dr. Emmalyn Capuno**

**December 2025**

## Dedication

To *Ustaza*. Zainab Ahmed Eltoum (Mama Hindiyya) ,My grandmother and the first teacher I have had the honor of learning.

To my Father and my Mother, who taught me how to be a person and to always strive to be a better person than I was yesterday.

To my Siblings, who have always supported me unconditionally.

To Musaab Mahjoub , Rudaina Al-Khair , and all my friends whom I wish were here to share this accomplishment with . and to all my family , friends and teachers **this work is dedicated to you.**

# Acknowledgment

**First and foremost, we thank Allah, the most Compassionate and the most Merciful.**

We would like to express our deepest gratitude to our supervisor **Dr. Emmalyn Capuno**, whose guidance, support, and expertise were invaluable throughout the development of this graduation project. Her insightful feedback and continuous encouragement helped shape this work significantly.

We would also like to thank the faculty members and staff of the Faculty of Information Technology for providing a supportive academic environment and for equipping us with the knowledge and tools necessary to complete this project successfully.

We would also like to thank Awab Ahmed, Abdelrahman Badreldin, Almigdad Zuhair, and Mohammed Al-Madid for their Continuous support and help in troubleshooting all of our issues.

Finally, We want to extend our thanks to anyone who contributed, directly or indirectly, to the completion of this project.

## **Abstract**

There is a challenge in Sudan's hospitality sector for both guests and hotel owners to find, compare, and manage hotel reservations, especially since most of these processes are currently done manually and many hotels lack digital infrastructure. The purpose of this project is to develop a mobile application that enables guests to view available hotels across different states of Sudan, compare prices and amenities, and easily make reservations, while also allowing hoteliers to manage their bookings in an organized and efficient manner through a centralized system.

The Ministry and the country would benefit from the proposed system as it supports the revitalization of tourism in the post-war period by providing a unified digital platform that improves data collection, policy-making, and the promotion of Sudan as a safe and attractive destination. The system also helps hotel owners expand their business reach, attract more customers, and modernize their operations. By facilitating these procedures through modern technology instead of requiring guests to visit hotels physically, the proposed system will simplify the booking process, support economic recovery, and enhance the overall experience for both guests and hoteliers.

## الملخص

لطالما وجدت مشكلة في السودان في قطاع الضيافة والسياحة لكلا السياح و ملاك الفنادق حيث من الصعوبة بمكان الحصول على حجوزات فندقية ، فضلا عن إمكانية المقارنة بين أسعار الفنادق المختلفة. بروزت هذه المشاكل بصورة أكبر في الآونة الأخيرة نسبة لعدم مواكبة التغيرات الحديثة و إنعدام البنية التقنية . يهدف هذا المشروع إلى تطوير تطبيق للهواتف الذكية يمكن الزائرين من استئجار الفنادق المتاحة عبر ولايات السودان المختلفة ، مقارنة أسعارها ، و حجز غرف بها بكل سهولة. كما يسمح لمديري الفنادق بإدارة الحجوزات بطريقة مرتبة و فعالة عبر نظام مركزي .

يهدف التطبيق لتحقيق الإستفادة القصوى للوزارة و الدولة كونه يعزز لثورة تقنية في مجال السياحة في Sudan ما بعد الحرب بتوفير منصة رقمية موحدة تحسن من جمع البيانات ، صنع القوانين ، و ترويج السودان كوجهة آمنة و جذابة للجميع. يساعد التطبيق أيضاً ملاك الفنادق في التوسيع في نطاق أعمالهم بجلب المزيد من الضيوف و تجديد طريقة عملهم و ذلك بتسهيل هذه العمليات عبر تقنيات حديثة بدلاً عن الحاجة لزيارة الفنادق يمكن استخدام التطبيق من تسهيل عملية الحجز ، دعم التعافي الاقتصادي للبلاد و تحسين التجربة بشكل عام للضيوف و مراء الفنادق.

# Table of Contents

Dedication	II
Acknowledgment	III
Abstract	IV
الملخص	V
<b>TABLE OF CONTENTS</b>	<b>VI</b>
List of Tables	IX
List of Figures	X
<b>CHAPTER ONE</b>	
1. Introduction	14
1.1. Background Of the Study	14
1.1.1. Nature Of the Project	14
1.1.2. Site Of the Project	15
1.2. Problem Statement	16
1.2.1. General Problems	16
1.2.2. Specific Problems	16
1.3. Project Objectives	17
1.3.1. General Objectives	17
1.3.2. Specific Objectives	17
1.4. Significance of The Study	18
1.5. Scope and Limitation	18
1.5.1 Scope	18
1.5.2 Limitation	19

## **CHAPTER TWO**

2.1. Related Studies #1	21
2.1.1 Title of the Reviewed Study	21
2.1.2 Proponents	21
2.1.3 Program and Year	21
2.1.4 Abstract	21
2.1.5 Synthesis (Similarities & Differences)	21
2.1.6. References	22
2.2. Related Studies #2	22
2.2.1 Title of the Reviewed Study	22
2.2.2 Proponents	22
2.2.3 Program and Year	22
2.2.4 Abstract	22
2.2.5 Synthesis (Similarities & Differences)	23
2.3 Related Studies #3	24
2.3.1 Title of the Reviewed Study	24
2.3.2 Proponents	24
2.3.3 Program and Year	24
2.3.4 Abstract	24
2.3.5 Synthesis (Similarities & Differences)	24
2.3.6. References	24

## **CHAPTER THREE**

3.1. Area of Technology	27
3.1.1. Mobile Application	27
3.1.2. Web Application	27
3.2. Software Methodology	27
3.2.1. Agile Software Development Methodology	27
3.2.2. Why agile methodology	28

3.3. Resources Requirement	29
3.3.1. Resource Requirement for Development	29
3.3.1.1. Hardware Resources Requirement for development	29
3.3.1.2. Software Resources Requirement for development	29
3.3.2. Resource Requirement for Deployment	29
3.3.2.1 Hardware Resources Requirement For Deployment	29
3.3.2.2. Software Resources Requirement For Deployment	30
3.3.2.3 Network Infrastructure	30
<b>CHAPTER FOUR</b>	
4.1 UML Design	33
4.1.1. Use case Diagram	33
4.1.2. Activity Diagram	35
4.2 Database	38
4.2.1 Entity-Relationship Diagram (Firestore Data Model)	38
4.2.2 Data Dictionary	39
<b>CHAPTER FIVE</b>	
5.1 Users prototype Interface	49
5.2 Testing and Implementations	68
5.2.1 Alpha and Beta Test results	68
5.2.1.1 Alpha Testing	68
5.2.1.2 Beta Testing	68
5.2.2 Implementation	69
5.3 Acceptability Test	69
5.4 Sample Generated Report	71

## **CHAPTER SIX**

6.1 Conclusion	74
6.2 Recommendation	74

## **BIBLIOGRAPHY**

7.1 References	77
7.2 Attachments	78
7.3 E-Books	78

## **APPENDICES**

8.1 USER MANUAL	80
8.2 PROGRAM CODES	85
8.3 SYSTEM TESTING EVALUATION TOOL	104
8.4 PROPONENTS' CURRICULUM VITAE	105

## **List of Tables**

Table 3.1: Hardware Resources Requirement for development	29
Table 3.2: Software Resources Requirement for development	29
Table 3.3: Hardware Resources Requirement for Deployment	30
Table 3.4: Software Resources Requirement for Deployment	30
Table 3.5: Network Infrastructure	31
Table 4.1 guests table	39
Table 4.2 notifications table	40
Table 4.3 hotels table	41
Table 4.4 generated_reports table	41
Table 4.5 rooms table	42
Table 4.6 reviews table	43

Table 4.7 bookings table	44
Table 4.8 admins table	45
Table 4.9 notifications table	45
Table 4.10 ministry_reports table	46
Table 4.11 activities table	46
Table 5.1 Acceptability Test	70

## List of Figures

Figure 1.1: Site of The Ministry of Tourism and Antiquities and Wildlife	15
Figure 1.2: The Ministry of Tourism and Antiquities and Wildlife Location	16
Figure 2.1: Hansonic Hotel Home Page	22
Figure 2.2: Nubia Apartments Home Page	23
Figure 2.3: Manage Rooms page – Admin view	25
Figure 3.1: Agile Methodology	28
Figure 3.2: Network Infrastructure	31
Figure 4.1 Use case diagram	34
Figure 4.2 Guest activity diagram	35
Figure 4.3 Hotel admin activity diagram	36
Figure 4.4 Ministry admin activity diagram	37
Figure 4.5 firestore diagram	38

Figure 5.1: Guest-Welcome Screen	48
Figure 5.2 Guest-Login Screen	48
Figure 5.3 SignUp Screen	48
Figure 5.4 Home Screen (1)	49
Figure 5.5 Home Screen (2)	49
Figure 5.6 Sidebar Screen	49
Figure 5.7 Search Hotels Screen page	50
Figure 5.8 Search Results Screen	50
Figure 5.9 Filter Hotels Screen	50
Figure 5.10 Hotels Info Screen	51
Figure 5.11 Hotels Rooms Screen	51
Figure 5.12 Select Rooms	51
Figure 5.13 Confirm Booking Screen	52
Figure 5.14 Booking Confirmed Screen	52
Figure 5.15 My Bookings Screen	52
Figure 5.16 Leave Review Screen	53
Figure 5.17 My Reviews Screen	53
Figure 5.18 Guest-Notifications Screen	53
Figure 5.19 Guest-Profile Screen	54
Figure 5.20 Guest-Settings Screen (1)	54
Figure 5.21 Guest-Settings Screen (2)	54
Figure 5.22 Favorites Screen	55
Figure 5.23 About Us Screen (1)	55

Figure 5.24 About Us Screen (2)	55
Figure 5.25 Help Center Screen (1)	56
Figure 5.26 Help Center Screen (2)	56
Figure 5.27 AI Assistant Screen	56
Figure 5.28: Admins-Welcome Page	57
Figure 5.29: Admins-Login Page	57
Figure 5.30: Ministry Dashboard Page (1)	58
Figure 5.31: Ministry Dashboard Page (2)	58
Figure 5.32: System Activities Page	58
Figure 5.33: Users Management Page	59
Figure 5.34: Registered Hotels Page	59
Figure 5.35: Register New Hotel Page	59
Figure 5.36: Hotel Information Page (1)	60
Figure 5.37: Hotel Information Page (2)	60
Figure 5.38: Ministry-Profile Page	60
Figure 5.39: Ministry-Reports Dashboard Page	61
Figure 5.40: Ministry-Custom New Reports Page	61
Figure 5.41: Hotel Dashboard Page (1)	62
Figure 5.42: Hotel Dashboard Page (2)	62
Figure 5.43: Hotel Bookings Management Page	62
Figure 5.44: Booking Details Page (1)	63
Figure 5.45: Booking Details Page (2)	63

Figure 5.46: Guests Reviews Page	63
Figure 5.47: Guests History Page	64
Figure 5.48: Room Management Page	64
Figure 5.49: Add New Room Page	64
Figure 5.50: Edit Room Page	65
Figure 5.51: Room Details Page (1)	65
Figure 5.52: Room Details Page (2)	65
Figure 5.53: Hotel-Profile Page (1)	66
Figure 5.54: Hotel-Profile Page (2)	66
Figure 5.55: Hotel-Notifications Page	66
Figure 5.56: Hotel-Settings Page (1)	67
Figure 5.57: Hotel-Settings Page (2)	67
Figure 5.58: Hotel-Reports Page	67
Figure 5.59: Ministry- Platform Booking Activity Report	71
Figure 5.60: Ministry-Hotel Summary Report	71
Figure 5.61: Ministry-Platform Revenue Summary Report	71
Figure 5.62: Ministry-Users Report	71
Figure 5.63: Hotel-Booking Summary (1) Report	72
Figure 5.64: Hotel-Booking Summary (2) Report	72
Figure 5.65: Hotel-Revenue Report	72

# **Chapter One**

## **Introduction**

# Chapter One

## Introduction

### 1. Introduction

In the rapidly evolving landscape of the 21st century, technological advancements have become the driving force behind transformative changes, reshaping the way we live, work, and connect with the world.

Initially, In the late 1990s and early 2000s web apps were the primary way users interacted with online services, including hotel reservations and travel planning; but the rise of smartphones created a demand for more accessible, efficient, and personalized solutions. This led to the development of mobile apps in the late 2000s, which now dominate various industries such as e-commerce, healthcare, banking, and hospitality. The hospitality and tourism industries are rapidly evolving, and mobile apps are at the heart of that transformation.

#### 1.1 Background of the study

##### 1.1.1 Nature of the project

Mobile Application is a software program that is specifically designed to operate on mobile devices such as smartphones, tablets, and smartwatches. Such applications can be downloaded and installed from app stores or other online platforms. The apps are intended to offer users various functionalities such as entertainment, communication, online services, etc.

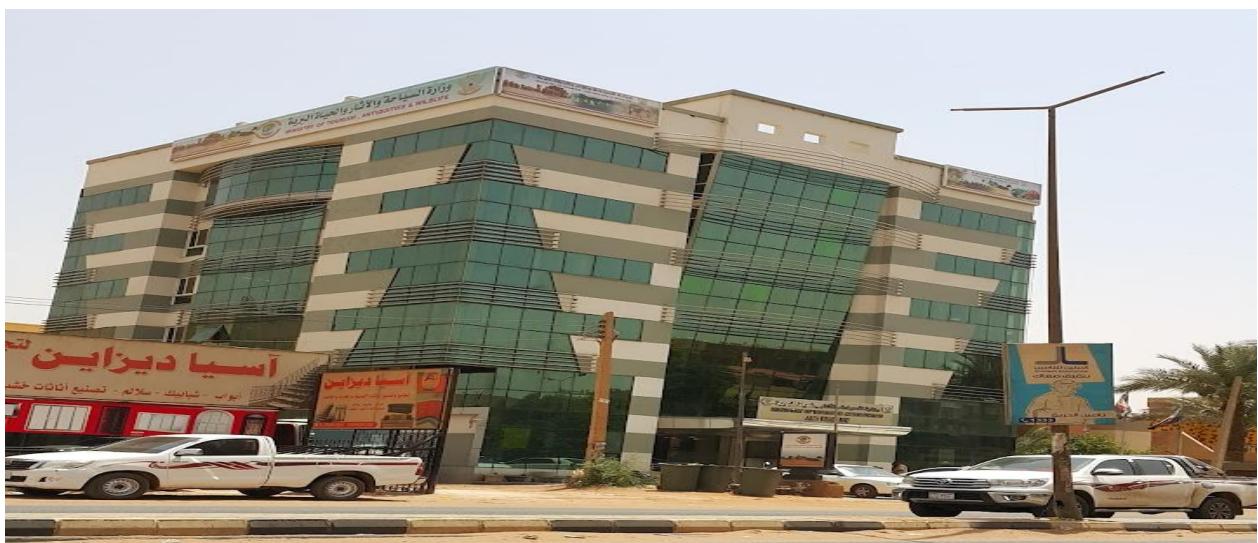
Hotels portal reservation and management system is a software application that allows the guests to book rooms online and the hoteliers to manage their online reservations. This system provides users with a wide range of hotel options in the states of Sudan, and the ability to compare prices, amenities, and reviews before making a reservation, all in one place instead of searching the hotels individually via the internet or going to them physically.

Sudan has faced massive challenges due to the current civil war, which has disrupted its economy, displaced millions, and severely impacted industries like tourism and hospitality. Once the war is finished and peace is restored, rebuilding Sudan's economy and infrastructure will be a priority. A reliable hotel booking system can attract international tourists and business travelers, showcasing Sudan as a safe and welcoming destination. Many local hotels may struggle to recover after the war and most of them lack digital infrastructure. Therefore, this project aims to develop a mobile application to connect those travelers and locals to the hotels and make their booking easier, and also help the hoteliers reach a broader audience, manage bookings efficiently, and offer competitive pricing.

### **1.1.2 Site of the Project**

The Ministry of Tourism and Antiquities and Wildlife located in Mashtal Street, Khartoum State-Sudan. The formal ministry structure has changed names and merged with other sectors several times throughout Sudan's history. After independence in 1956, Sudan established the Department of Tourism and Antiquities First tourism promotion office opened in 1967 in Khartoum, the National Corporation for Tourism and Hotels was established In 1971, The Wildlife Conservation General Administration was formed in 1975. Following the Comprehensive Peace Agreement in 2005, tourism development became a priority. The Ministry was reorganized as the Ministry of Tourism, Antiquities and Wildlife in 2010. Since then, the ministry is responsible for managing and promoting the country's rich cultural heritage, archaeological sites, wildlife conservation, and tourism development, including roads, hotels, and transport, while promoting Sudan as a destination for cultural and natural tourism. In 2018 the ministry joined the African Tourism Board (ATB).

The ongoing conflict since April 2023 has caused significant damage to tourism infrastructure in conflict areas, and the hotel sector, which includes a range of categorized and uncategorized hotels, guesthouses, and hostels, has also sustained damage from the conflict. The Ministry of Tourism has endured a loss of essential information and resources. As the country moves toward recovery in post-war, rebuilding the tourism and hospitality sectors to help in restoring the country's economic stability will be crucial.



**Figure 1.1: Site of The Ministry of Tourism and Antiquities and Wildlife**



**Figure 1.2: The Ministry of Tourism and Antiquities and Wildlife Location**

## 1.2 Problem Statement

### 1.2.1 General Problem

The main problem is after the current war is finished, when foreign travelers or locals are searching for hotels to stay in Sudan they need to visit global booking platforms or search through the website of each hotel separately (if it exists). It is difficult to find hotels in Sudan and compare between them when there are no shared online platforms to search through.

### 1.2.2 Specific problems

1. Difficulties to search and know about hotels in Sudan to compare the options without visiting them physically.
2. Waste of time, effort and cost to go to each hotel location individually to find a convenient accommodation to stay in.
3. Lack of digital infrastructure in Sudan to make online reservations and manage them easily and efficiently.
4. Lack of tourism awareness and negative perception about hotels in Sudan.
5. There are a few hotels integrating with global travel and booking platforms, Sudan does not have a dedicated local web or mobile app for its hotels.
6. The current war has severely impacted Sudan's economy, it will need local businesses to help in the recovery process and build a new image of Sudan as a safe and welcoming destination.

## **1.3 Project Objectives**

### **1.3.1 General Objective**

The main objective is to develop a mobile application for hotels booking system for the Ministry of Tourism and Antiquities and Wildlife in Sudan to attract tourists, and connect travelers and locals with a wide range of hotels options in Sudan based on the destined state, providing them with all the information they need and allowing them to compare and choose their accommodation and book their rooms easily.

### **1.3.2 Specific Objectives**

1. To design a mobile module that allows to create an account and login, ensuring secure authentication based on the user role (guest/admin).
2. To design a mobile module that allows guests to search for hotels across Sudan, filtered by location, price range, rates, and amenities.
3. To design a mobile module that allows to provide the guests with detailed hotel profiles, showcasing images, descriptions, amenities, rooms, user reviews, to aid decision-making.
4. To design a mobile module that allows guests to compare and choose their accommodation and book rooms instantly with real-time availability.
5. To design a mobile module that allows guests to manage their bookings, including viewing or canceling bookings and manage their account including editing profile or delete account.
6. To design a web module that allows to create an admin dashboard for ministry admins to approve and register hotels, monitor all hotel listings, bookings, and user activity.
7. To design a web module that allows to create an admin dashboard for hoteliers' admins to manage their profiles, including updating hotel information, images, room details and adding new rooms.
8. To design a web module that allows hoteliers to manage booking requests and cancellations efficiently.
9. To design a web module that allows the Ministry to generate reports to aid in tourism and hotels trends, booking statistics, revenue insights and user management, and the hoteliers' admins to generate bookings and revenue reports .

## 1.4 Significance of the study

The designed mobile application will be significant to the following:

- **The Ministry:** The project will help the ministry by providing a centralized digital platform to monitor and promote Sudan's hospitality sector post-war, and enhance tourism data collection for policy-making and infrastructure development.
- **The country:** The project will help in the economic recovery and growth post-war by boosting tourism, supporting local businesses and creating job opportunities; to rebuild Sudan as a safe and attractive destination.
- **Hoteliers:** The project will help hotel owners to grow their business in the digital age by competing in the market, reaching broader customers, attracting international travelers, and managing bookings digitally all in one place.
- **Guests:** The project will help potential guests in saving time and effort by comparing hotels in one platform instead of manual individual searches and allowing easy booking with instant confirmation.
- **Researchers:** The project will provide valuable opportunities for researchers by enhancing technical capabilities in mobile app development, implementing a real-world hotel booking system and creating digital solutions for post-war tourism.
- **Future researchers:** The project will enable future researchers to use this case study as a source of data and a methodology when conducting similar topics in hospitality technology or digital solutions for developing economies.

## 1.5 Scope and Limitation

### 1.5.1 Scope:

The designed mobile application functions as an intuitive and user-friendly online platform that connects guests with wide range of hotels in Sudan, providing guests, hotel admins and the ministry admins with the following functionalities:

#### Guests can:

- **Create** personal accounts and access the mobile application with secure login.
- **Search/filter/compare** hotels by location, price, rates and amenities.
- **View** detailed hotel profiles with images, descriptions, rooms, reviews and amenities.
- **Book** rooms with real-time availability checks.
- **View/cancel** existing bookings.
- **Submit** ratings and reviews for stayed hotels.

#### Hotel Admins can:

- **Limited-Access** to the admin dashboard for their hotel only with secure login.
- **Update/add** hotel information, room types, and pricing.
- **Respond** to booking requests.
- **View/manage** room details, bookings and view reviews.

- **Generate** two types of reports, such as: booking summary and revenue reports

#### **Ministry Admins can:**

- **Full-Access** to the admin dashboard with secure login..
- **Activate/deactivate** user accounts.
- **Register/approve** hotel profiles on the platform.
- **Verify/approve** hotel registrations.
- **Send** notifications to guests for their account registration and booking status.
- **Monitor** all hotel listings, bookings, and user activity.
- **Generate** various types of reports, such as: platform revenue summary, hotel listing summary, hotel booking activities, and users listing.
- **Oversee** the entire platform.

#### **1.5.2 Limitations**

- The application does not support online payment.
- The application does not support translation to Arabic or other languages.
- The application is not connected to the hotel's systems.
- The application does not offer transportation or tour booking services.
- The application only shows hotels in Sudan.
- The application does not support iOS devices.

# **CHAPTER TWO**

## **REVIEW OF RELATED STUDIES**

# CHAPTER TWO

## REVIEW OF RELATED STUDIES

### **2. REVIEW OF RELATED STUDIES**

#### **2.1. Related study #1**

##### **2.1.1. Title of the reviewed study**

Online Hotel Reservation System

##### **2.1.2. Proponents**

Richard Bemile - Methodist University College Ghana

Akwasi Achampong - University of South Africa

Emmanuel Danquah - Takoradi Polytechnic

##### **2.1.3. Program & Year**

IJISET - International Journal of Innovative Science, Engineering & Technology, Vol. 1

Issue 9 - November 2014.

##### **2.1.4. Abstract**

The hotel industry is a business venture for the owner and a solace for the traveler and/or tourist. A customer can get stranded in the quest to secure a hotel room to pass the night if he has not made adequate plans by the existing system. Through this study, it was realized that for a customer to be guaranteed a room, he or she has to physically come to the hotel since the attendants paid more attention to that. He could also send a friend or relative who lives around the neighbourhood of the hotel to do the booking for him. There is nothing to bond the hotel and the customer in person that he has indeed booked for a room. This study took Hansonic Hotel as its case. It looked at creating an online reservation system to enable customers to choose the room they wanted after a virtual tour to guarantee him a room.

##### **2.1.5. Synthesis (Similarities & Difference)**

###### **Similarities**

- Both projects are online hotel booking systems.
- Both systems provide room information for customers for easy reservation.
- Both systems have administrator's domain with required account registration for information management and reports generating.
- Both systems do not support online payment.

###### **Difference**

- The related study is a web-based application, while our project is a mobile application.
- The related study takes a specific hotel as a case; which is Hansonic hotel, while our project takes all hotels of Sudan.
- The related study does not require account registration for the customers.
- The related study has an existing manual system.

## 2.1.6. References

- [www.ijiset.com](http://www.ijiset.com) Vol. 1 Issue 9
- [https://www.researchgate.net/publication/274079295\\_Online\\_Hotel\\_Reservation\\_System](https://www.researchgate.net/publication/274079295_Online_Hotel_Reservation_System)

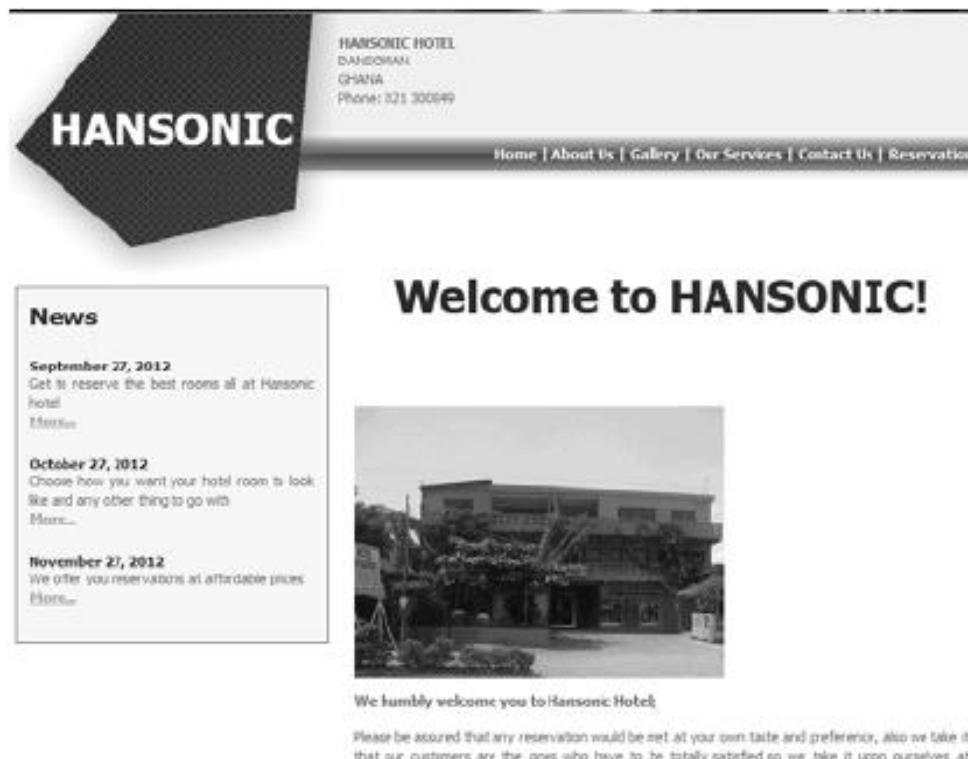


Figure 2.1: Hansonic Hotel Home Page

## 2.2. Related study #2

### 2.2.1. Title of the Reviewed Study

Online Reservation System for Nubia Apartments

### 2.2.2. Proponents

Manat Allah Mamoun Ali - 201603004

Amani Sirelkhatim Awad - 201603008

### 2.2.3. Program & Year

Bachelor Of Science (Honors) in Information Technology - 2021

### 2.2.4. Abstract

Online reservations for hotel apartment buildings are needed specially in Sudan because it attracts foreigners and Sudanese living abroad due to their need for a home-like experience for their holidays. The idea of the project is to increase Nubia Apartments clients using a computerized online reservation system, the system will allow the customer to make reservations through the internet and see the available apartments. The “Online Reservation System for Nubia Apartments” aims to simplify the reservation process and save time and effort for both customer and Nubia management.

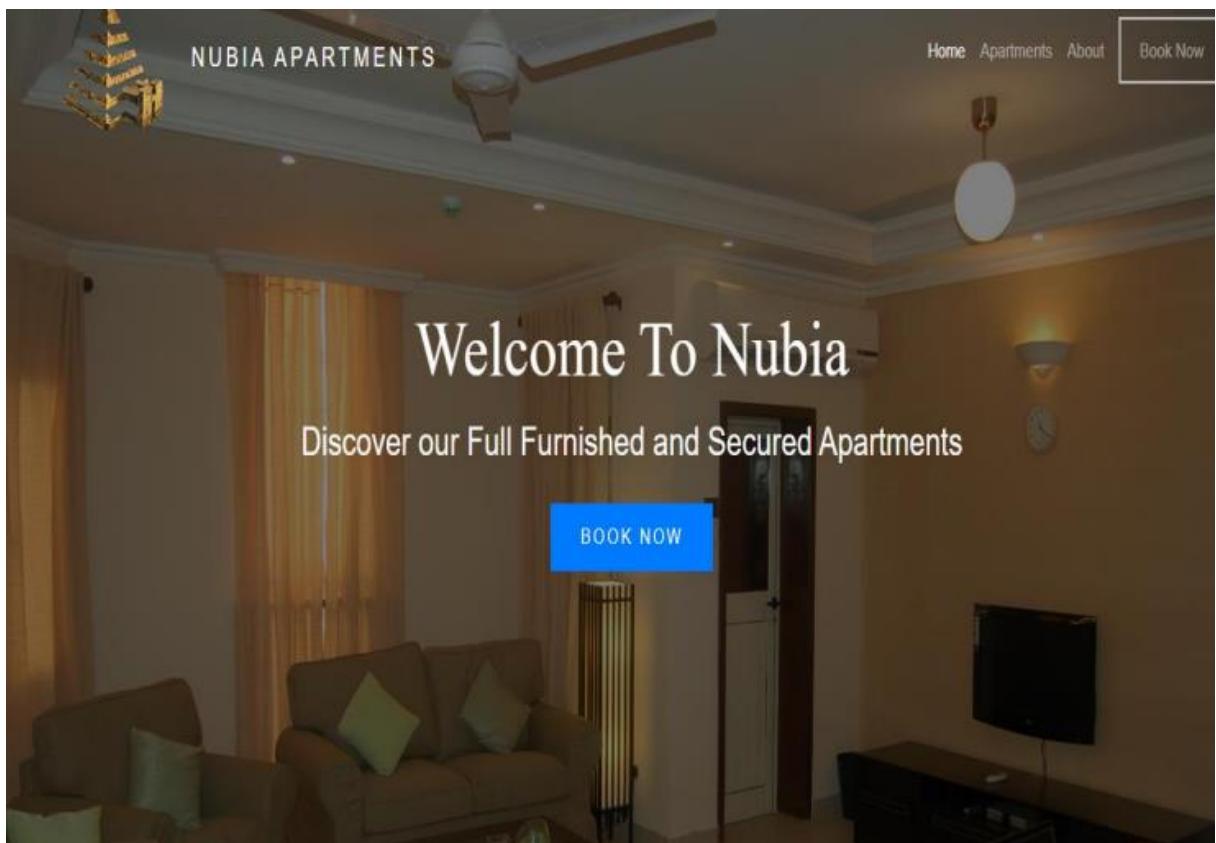
## 2.2.5. Synthesis (Similarities & Difference)

### Similarities

- Both projects are booking systems.
- Both systems provide the customer with information and features to compare before making a reservation.
- Both systems require account registration for both customers and admins
- Both systems do not support online payment.

### Differences

- The related study is a web-based application, while our project is a mobile application.
- The related study is for apartments reservation, while our project is for hotels booking.
- The related study is dedicated for Nubia Apartments, while our project is for all hotels in Sudan.



**Figure 2.2: Nubia Apartments Home Page**

### **2.3. Related study #3**

#### **2.3.1. Title of the Reviewed Study**

Research on Hotel Management System

#### **2.3.2. Proponents**

W.P.S.W. Weerasinghe - Sri Lanka Institute of Information Technology, SRI LANKA

K.D.M.I. Jayathilaka - Sri Lanka Institute of Information Technology, SRI LANKA

W.V.C. Prasadi - Sri Lanka Institute of Information Technology, SRI LANKA

M.D.K.M Goonetilleke - Sri Lanka Institute of Information Technology, SRI LANKA

D. I. De Silva - Sri Lanka Institute of Information Technology, SRI LANKA

Piyumika Samarasekara - Sri Lanka Institute of Information Technology, SRI LANKA

#### **2.3.3. Program & Year**

International Journal of Engineering and Management Research, Volume-12, Issue-5 - October 2022

#### **2.3.4. Abstract**

With rapid growth of economy and tourism, there is an intensified competition that can be seen in the hotel industry today. To be in the competition, they need to continuously improve their management techniques and procedures. “Online Hotel Management system” is software developed to focus on these factors. Through this system, it will be able to manage various functions including room and hall reservations, ordering food, and managing employees and suppliers. We intend to develop this web application using React JS, Express JS, Node JS, and MongoDB. This system addresses hotel management issues while avoiding issues that arise when tasks are carried out manually. In terms, the main objective of this whole process is to automate the day today manual tasks of this system. Therefore, this online hotel management system is designed to find a more practical, well organized, and quick way of processing the service from the hotel for both nearby and distant customers by giving a more user friendly and more GUI oriented experience.

#### **2.3.5. Synthesis (Similarities & Difference)**

##### **Similarities**

- Both projects are systems for hotels.
- Both systems provide the customer with information and features to allow an easy room booking experience.
- Both systems have admin front-end and system back-end.
- Both systems require account registration for both customers and admins.

##### **Differences**

- The related study is an e-commerce web-based application, while our project is a mobile application.
- The related study is fundamentally a management system that allows the customers to also book rooms, their system consists of 4 main functions such as Restaurant management, Room reservation management, Review management and Employee and Supplier management, while our project is fundamentally a booking system and only provides reservation management.
- The related study supports online payment, while our project does not.

### 2.3.6. References

- <https://doi.org/10.31033/ijemr.12.5.27>
- [https://www.researchgate.net/publication/365151133\\_Research\\_on\\_Hotel\\_Management\\_System](https://www.researchgate.net/publication/365151133_Research_on_Hotel_Management_System)

Wijaya Beach Hotel Edit Room Edit Restaurants Edit Employee Edit Suppliers All Users Logout

Search by Room Name ADD ROOM ALL BOOKINGS

Start Date End Date SEARCH Adult CHILDREN SEARCH BedRoom SEARCH

**Room 001**  
Delux Sea-Front Room  
**Features:**  
2x King-sized four-poster bed Air Conditioning & fan Spacious balcony Modern en-suite bathroom  
**Rent Per Day:** LKR 13000.00 /-

Max Count: 04 Adult: 07  
Children: 02 BedRoom: 02

**Room 002**  
Delux Room  
**Features:**  
1x King-sized four-poster bed Day-bed/Sofa(suitable as extra single bed), Air Conditioning & fan, Spacious balcony, Modern en-suite bathroom, WiFi Internet access  
**Rent Per Day:** LKR 11000.00 /-

Max Count: 06 Adult: 02  
Children: 03 BedRoom: 03

**VIEW DETAILS** **UPDATE ROOM** **DELETE ROOM**

Figure 2.3: Manage Rooms page – Admin view

# **CHAPTER THREE**

# **TECHNICAL BACKGROUND**

# **CHAPTER THREE**

## **TECHNICAL BACKGROUND**

### **3. TECHNICAL BACKGROUND**

#### **3.1. Area of Technology**

##### **3.1.1. Mobile Application**

Mobile Application is a software program that is specifically designed to operate on mobile devices such as smartphones, tablets, and smartwatches. Such applications can be downloaded and installed from app stores or other online platforms. The Hotels portal reservation and management system is designed as a mobile-first application, ensuring users can easily access hotel listings and make reservations via smartphones. Mobile applications offer high accessibility, interactive user interfaces, and real-time functionality, making them ideal for travel and hospitality solutions.

The technology used for mobile development is Flutter, a powerful open-source UI toolkit developed by Google for cross-platform applications. Flutter allows for the development of native-like applications using a single codebase, meaning apps can run on both iOS and Android without needing separate development efforts. The programming language used is Dart, which enables smooth and structured development, optimized for front-end and real-time applications.

##### **3.1.2. Web-Based Application**

A web-based application is a software application that runs on a web server and can be accessed through a web browser. Unlike mobile applications, web applications do not require installation on a device and can be accessed from any device with internet connectivity. Although the primary platform for our system is mobile, certain administrative and backend functionalities are accessible via web-based tools provided by technologies like Firebase Console, Firestore Database and Firebase Authentication. A web-based dashboard is designed primarily for hotel administrators and ministry officials to manage bookings, monitor hotel activity, and generate reports. Web platforms allow centralized management and secure role-based access to data.

#### **3.2. Software Methodology**

##### **3.2.1. Agile Software Development Methodology**

The project follows the Agile methodology, which is widely adopted throughout the software development life cycle. Agile is known for its iterative approach, collaboration, and responsiveness to change, making it ideal for projects requiring continuous refinement and user feedback like hospitality projects. The iterative process of the cycle phases begins with planning, where requirements are gathered, and the vision for the software is established. Development then takes place in short sprints, where features are built incrementally and tested continuously to maintain quality. The testing phase is embedded within each sprint, ensuring that

bugs and issues are addressed early. Deployment follows, where functional updates are released frequently rather than waiting for a final version. Finally, feedback is gathered from users and stakeholders to refine the application and inform the next development cycle, creating an ongoing loop of innovation and improvement.

### 3.2.2. Why agile methodology

The hospitality industry is dynamic, and user needs may change rapidly. Agile allows for adjustments to be made based on user feedback and market trends, ensuring the application remains relevant and user-friendly, agile also encourages ongoing interaction between developers, hoteliers, and ministry officials, ensuring that every feature aligns with actual industry needs, and not to mention Sudan's hospitality industry is recovering, meaning new requirements such as hotel registrations or expanded admin permissions may emerge. Agile development adapts dynamically to such changes.

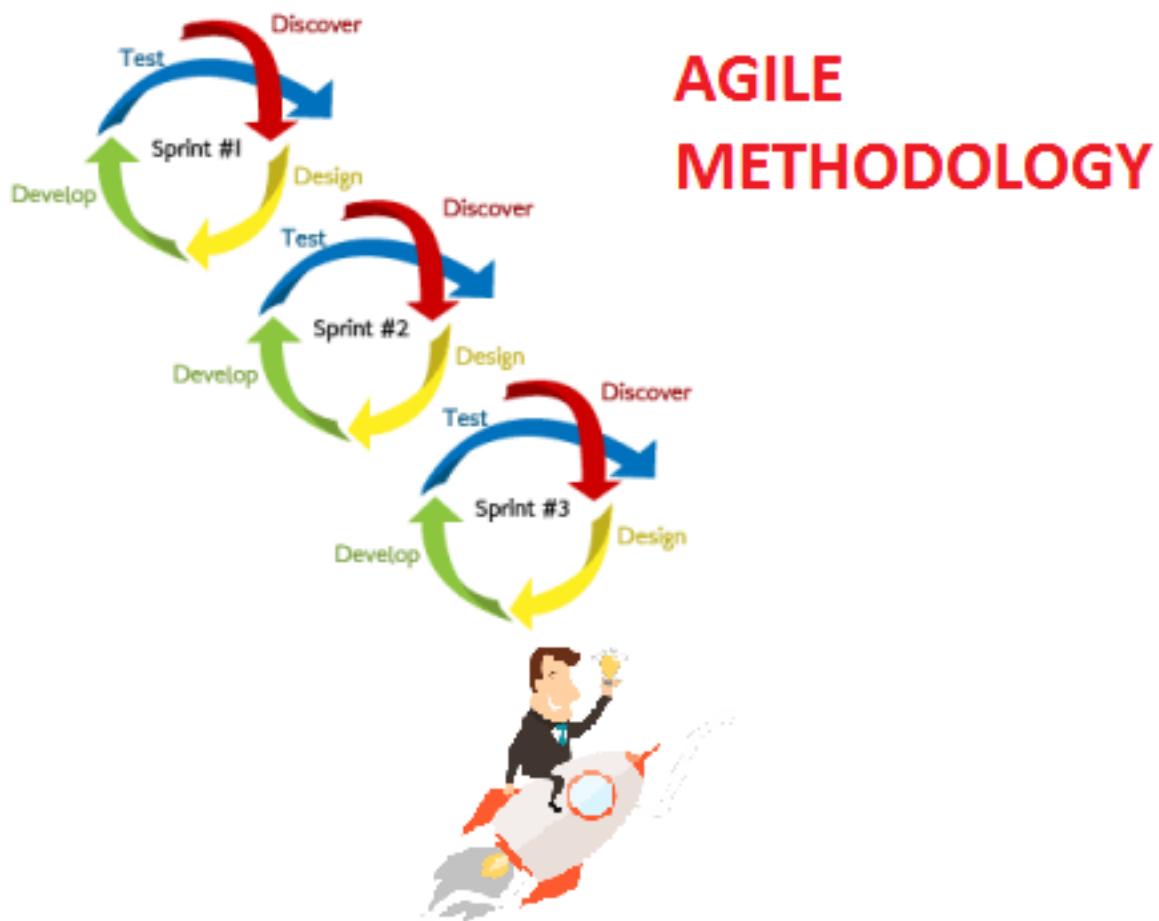


Figure 3.1: Agile Methodology

### **3.3. Resources Requirement**

#### **3.3.1 Resources Requirement For Development**

##### **3.3.1.1. Hardware Resources Requirement for development**

Description	Specification
Processor	Intel Core i5
Random Access Memory (RAM)	8GB
Hard Drive	256GB

**Table 3.1: Hardware Resources Requirement for development**

##### **3.3.1.2. Software Resources Requirement for development**

Description	Specification
Operating System	Windows 10
IDE	Visual Studio Code
SDK	Android / Flutter
Database Management	Firebase Firestore
Programming Language	Dart
Backend Functions	Firebase Cloud Functions
Cloud Storage	Firebase Storage

**Table 3.2: Software Resources Requirement for development**

#### **3.3.2 Resources Requirement For Deployment**

##### **3.3.2.1 Hardware Resources Requirement For Deployment**

Description For PC	Specification
Processor	Intel core i5 or higher
Random Access Memory (RAM)	4 GB Or higher
Hard Drive	125 GB Or higher

Description For Mobile	Specification
Processor	1.3 GHz Or higher (supports OpenGL ES 3.0+)
Random Access Memory (RAM)	4 GB Or higher
Internal Storage	16 GB or higher

**Table 3.3: Hardware Resources Requirement for Deployment**

### 3.3.2.2. Software Resources Requirement For Deployment

Description For PC.	Specification
Operating System	Windows 10 or higher
Web browsers	Google Chrome, Firefox or any browser
Web Console	Firebase Console
Server side	Firebase Hosting, Cloud Functions, Firestore
Description For Mobile	Specification
Operating System	Android 8.0 or higher
Application Host	Google Play Store/Firebase
Internet Access	Required

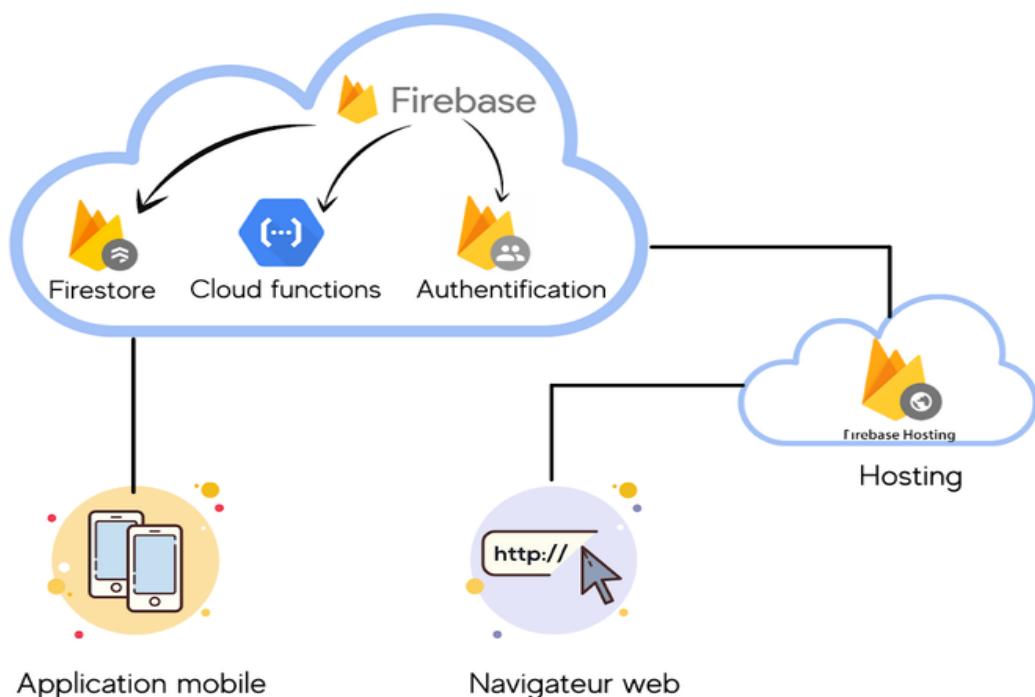
**Table 3.4: Software Resources Requirement for Deployment**

### 3.3.2.3 Network Infrastructure

Device	Specification	Usage
Internet Connection	Broad band/Wi-Fi/4G/5G	Provides high-speed access for mobile devices to connect with Firebase services.
Routers	Supports up to 1 Gbps	Ensures fast and stable connections between users' devices and the cloud infrastructure.
Firebase Cloud	Google Cloud hosted	Handles backend services like Authentication, Firestore Database, Cloud Functions, and Cloud Storage.
Network Security	HTTPS, Firebase Authentication, Firestore Rules	Secures all communications with encrypted protocols and ensures data privacy and role-based access control.

CDN	Google CDN	Firebase Hosting uses Google's CDN for faster static asset delivery.
Domain/DNS	custom domain	for the web panel, Points the domain to Firebase Hosting with DNS configuration. (e.g., admin.hotelsudan.com)
Firewall	Hardware-based or router-integrated, supports stateful inspection	Protects internal network from unauthorized access and filters traffic.

**Table 3.5: Network Infrastructure**



**Figure 3.2: Network Infrastructure**

# **CHAPTER FOUR**

# **ANALYSIS AND DESIGN**

# **CHAPTER FOUR**

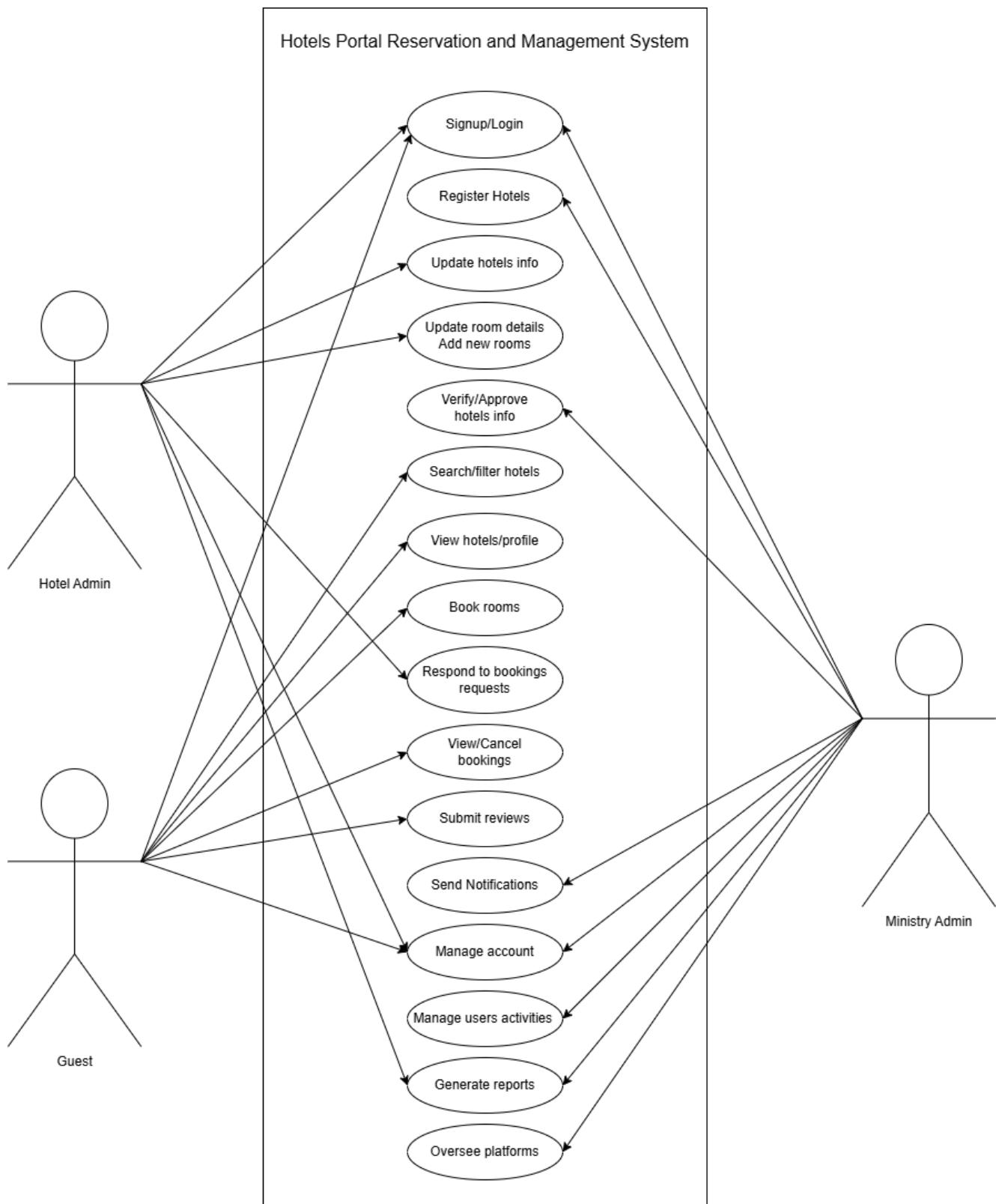
## **ANALYSIS AND DESIGN**

### **4.1 UML Design**

UML (Unified Modeling Language) Design is a standardized visual language used to model the structure and behavior of software systems. It provides a set of diagram types—such as use case diagrams, and activity diagrams—These diagrams serve as blueprints for development, allowing developers, analysts, and stakeholders to understand the system’s components, relationships, and workflows without having to read through raw code.

#### **4.1.1 Use Case Diagram**

A Use Case Diagram is a UML diagram that identifies the different types of users (actors) in a system and the various actions (use cases) they can perform. It focuses on “what” the system should do rather than “how” it does it. Actors can be human users or external systems, and use cases represent the services or functionalities the system provides. The diagram uses simple symbols—stick figures for actors and ovals for use cases—connected by lines to indicate relationships.



**Figure 4.1 Use case diagram**

#### 4.1.2 Activity diagram

An Activity Diagram is a UML behavioral diagram that models the sequence of activities and decision points in a process. It represents the workflow from start to end, showing parallel processes, conditional branching, and loops. It uses symbols such as rounded rectangles for activities, diamonds for decision points, and arrows to indicate flow direction.

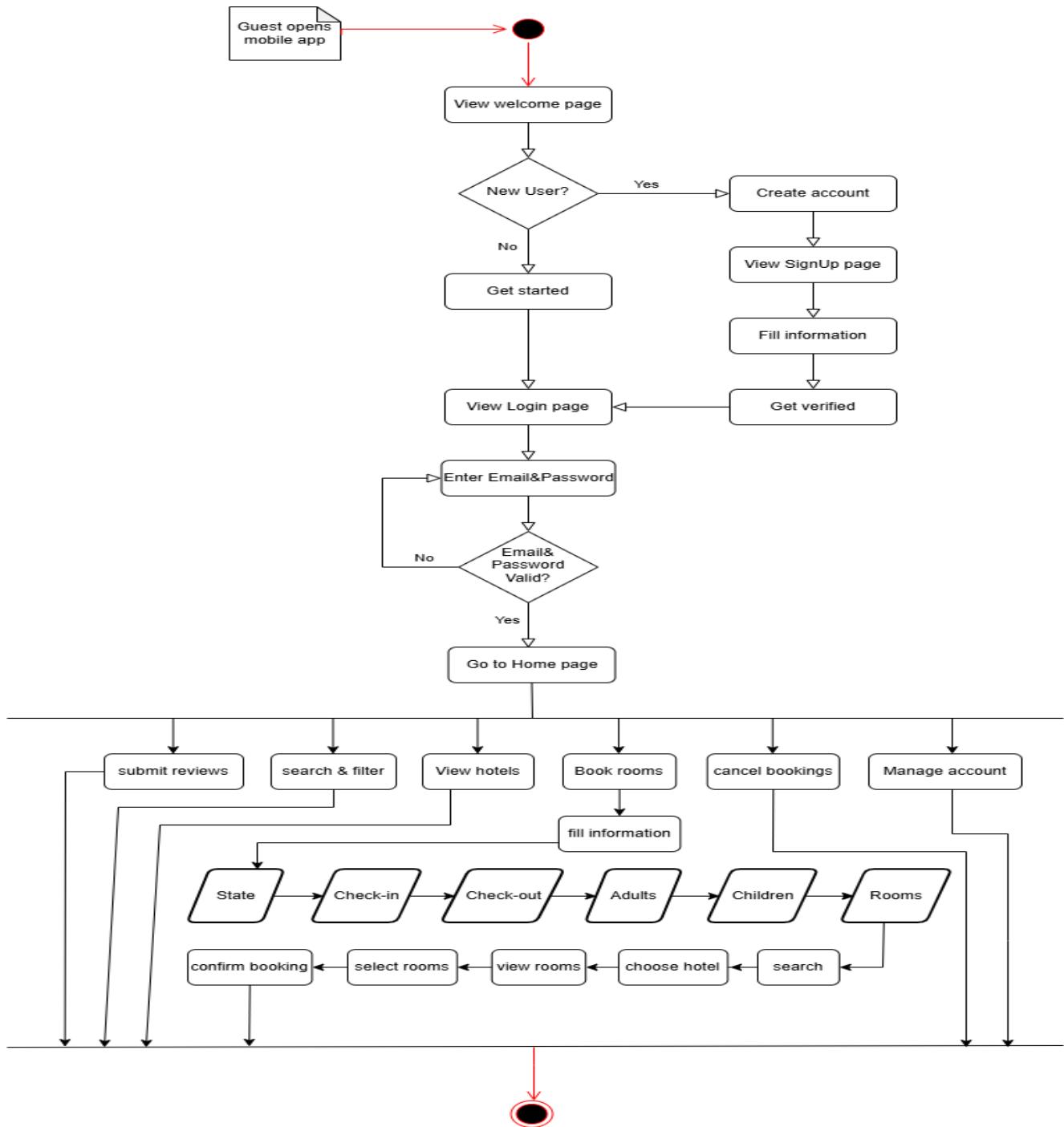
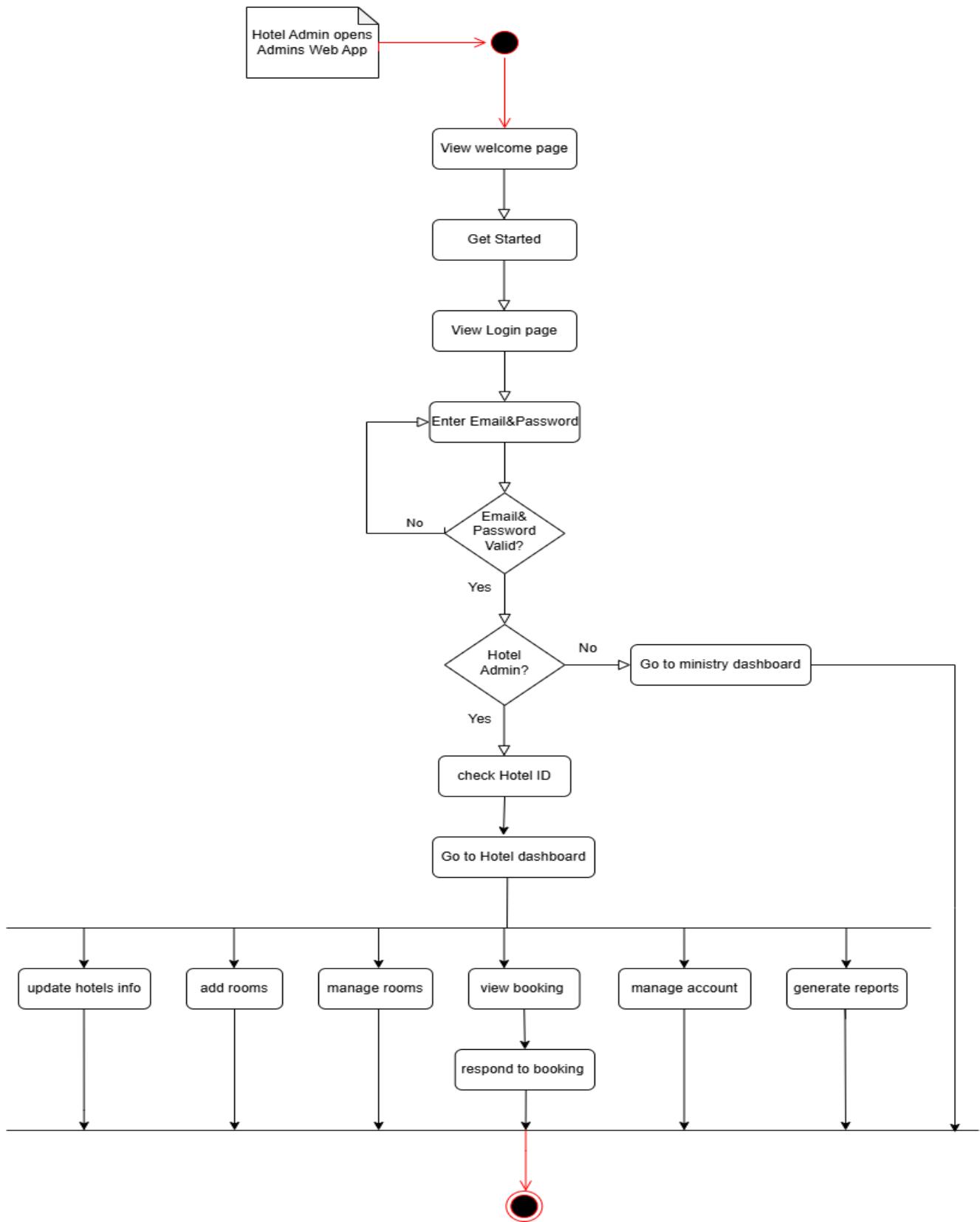
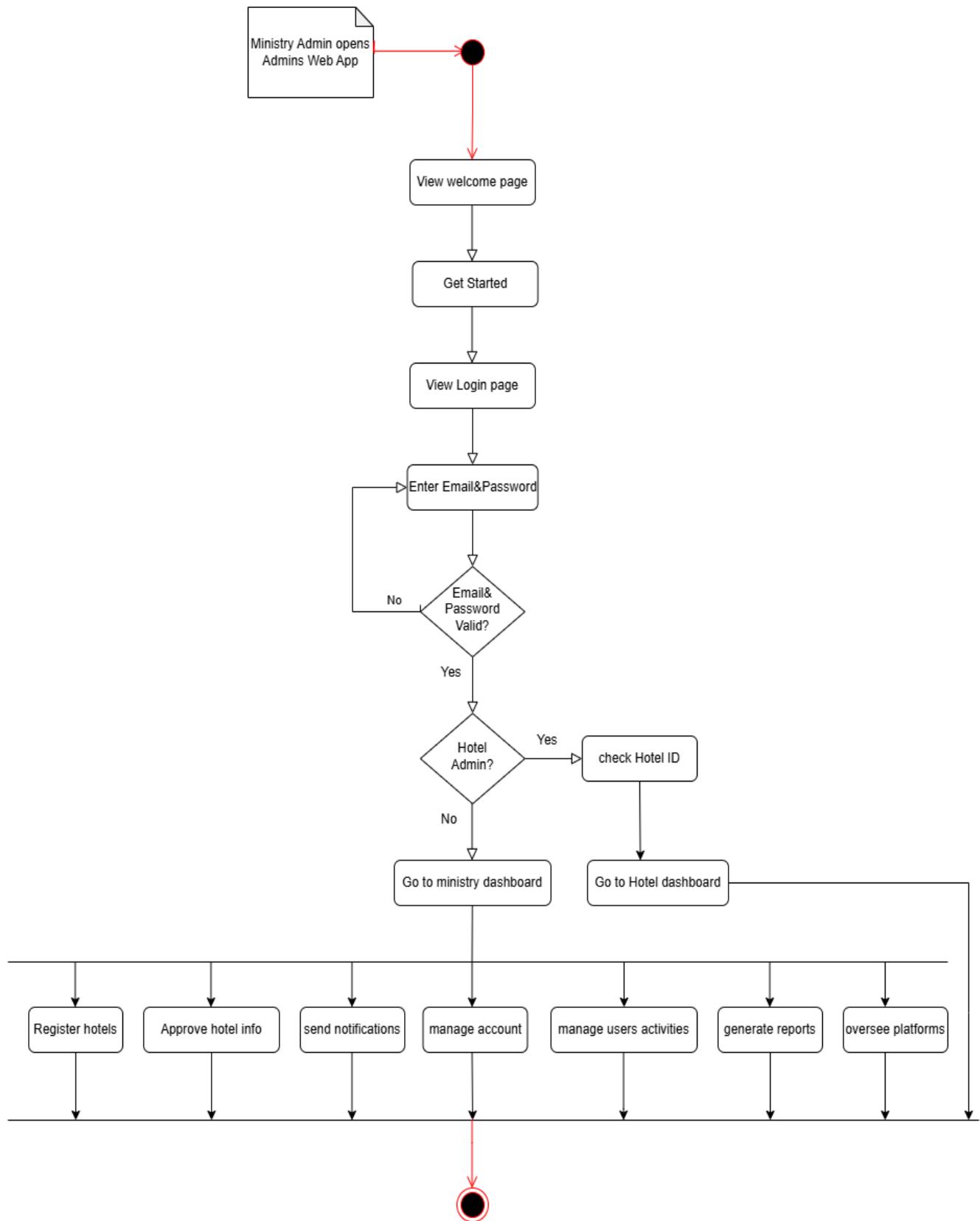


Figure 4.2 Guest activity diagram



**Figure 4.3 Hotel admin activity diagram**



**Figure 4.4 Ministry admin activity diagram**

## 4.2 Database

A database is an organized collection of data that is stored and accessed electronically. In software systems, the database acts as the backbone for storing all the information the application needs to function. Database design involves structuring the data in a way that supports efficient storage, retrieval, updating, and security. Depending on the type of database (relational or NoSQL), the design may involve tables with rows and columns or collections of documents and key-value pairs.

### 4.2.1 Entity-Relationship Diagram (Firestore Data Model)

An Entity-Relationship Diagram is a type of data modeling diagram used to visually represent the entities in a system and the relationships between them. The ERD is crucial for understanding how data is linked and how it will be stored in the database.

The Firestore Data Model defines how data is structured within Firestore using collections, documents, and subcollections. Unlike relational databases, Firestore does not use joins or foreign keys—instead, it relies on document references and hierarchical nesting to represent relationships.

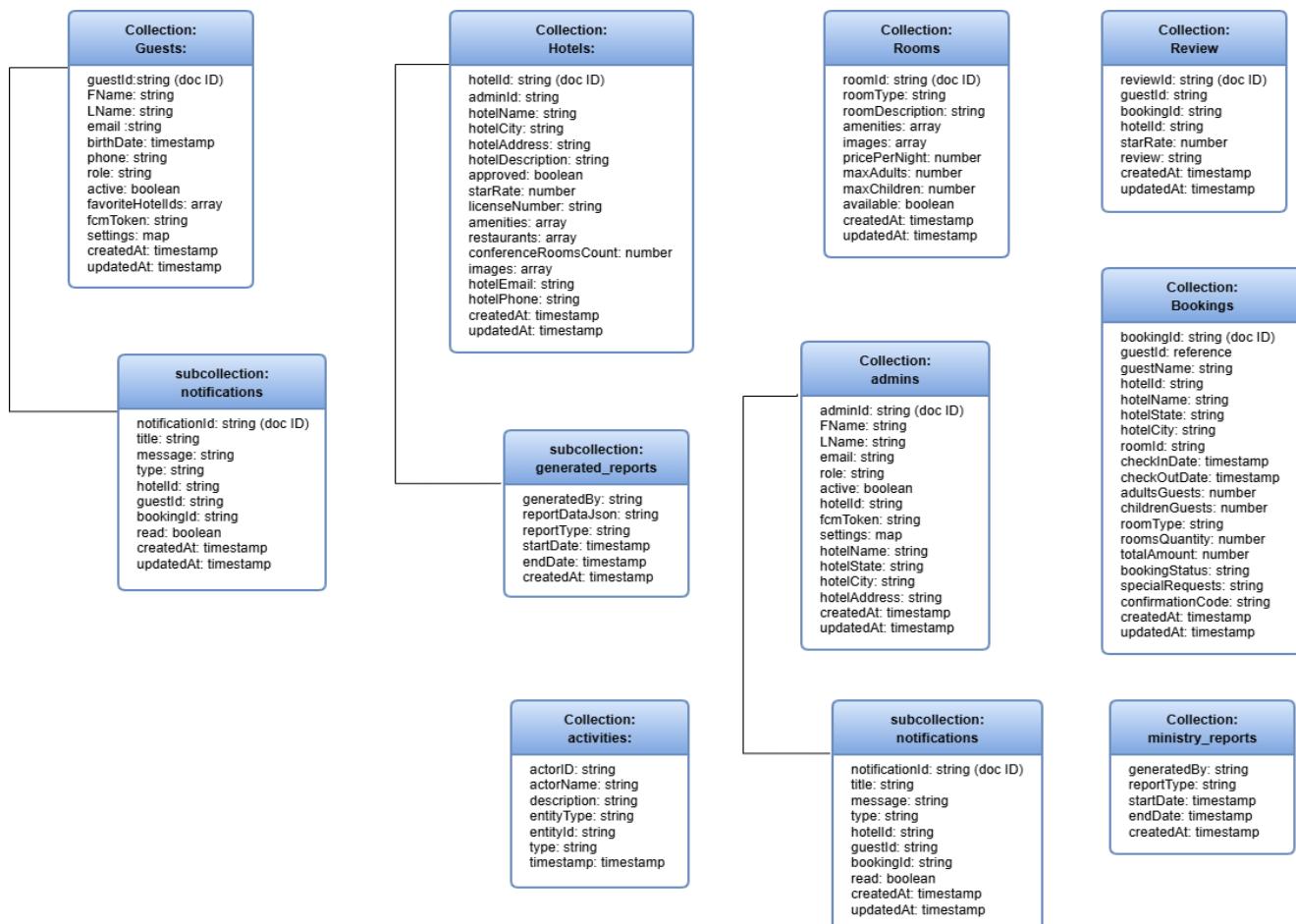


Figure 4.5 firestore diagram

#### 4.2.2 Data Dictionary (Firestore Field Documentation)

A Data Dictionary is a detailed description of the structure of a database. It lists every table or collection, its fields, the data type of each field, constraints (like primary keys, unique values, or required fields), and a description of what each field represents. The data dictionary serves as a guide for developers, database administrators, and analysts to ensure consistency in how data is stored, retrieved, and interpreted.

The Firestore Field Documentation serves as a reference for all fields used across collections and documents. It outlines the field names, data types, constraints, and descriptions.

#### Collection: guests

Field Name	Type	Valid values	Description
guestId	string	-	Document ID for guest
FName	string	-	Guest's first name
LName	string	-	Guest's last name
email	string	Valid email format	Guest's email address
birthDate	timestamp	-	Guest's Date of Birth
phone	string	-	Guest's contact number
role	string	“Guest”	The user’s role in the system
active	boolean	“true”, “false”	The user’s account status
favoriteHotelIds	array	-	Guest’s favorite hotels
fcmToken	string	-	Firebase Cloud Messaging generated Token, an ID for sending push notifications to guests
settings	map	{“emailNotifications”: “true/false”, “pushNotification”: “true/false”, “smsNotifications”: “true/false”, “updatedAt”: “timestamp” }	Guest’s account settings, including notifications enabling and last time updated
createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

Table 4.1 guests table

### Subcollection: notifications

Field Name	Type	Valid values	Description
notificationId	string	-	Document ID for notification
title	string	-	Notification's title
message	string	-	Notification's body
type	string	“Registration”, “booking status”, “review”	Notification's content type
hotelId	string	Ref → hotels/{hotelId}	Related hotel ID reference
bookingId	string	Ref → bookings/{bookingId}	Related booking ID reference
guestId	string	Ref → guests/{guestId}	Notification receiver ID reference
read	boolean	“true”, “false”	Message read status
createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

Table 4.2 notifications table

### Collection: hotels

Field Name	Type	Valid values	Description
hotelId	string	-	Document ID for Hotel
adminId	string	Ref → admins/{adminId}	The admin assigned to the hotel ID reference
hotelName	string	-	Hotel's name
hotelState	string	-	Hotel's state
hotelCity	string	-	Hotel's city
hotelAddress	string	-	Hotel's location
hotelDescription	string	-	Hotel's description

approved	boolean	“true”, “false”	Hotel approval status
starRate	number	1, 2, 3, 4, 5	Hotel’s average rating
licenseNumber	string	-	Hotel’s regulatory license ID
amenities	array	Predefined list (e.g. WiFi, AC)	Hotel amenities
restaurants	array	{“number”: “location”: , “name”:}	The restaurants the hotel owns
conferenceRoomsCount	number	1,2,3, etc.	Number of the conference rooms in the hotel
images	array	Only image URLs	Hotel, rooms images
hotelEmail	string	Valid email format	Hotelt's contact email address
hotelPhone	string	-	Hotel's contact number
createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

**Table 4.3 hotels table**

#### Subcollection: generated\_reports

Field Name	Type	Valid values	Description
generatedBy	string	-	Document ID of the admin generated the report
reportDataJson	string	-	The content of the report
reportType	string	“Booking Summary”, “Revenue Report”	Type of the report
startDate	timestamp	-	Start date of the wanted period for the report
endDate	timestamp	-	End date of the wanted period for the report
createdAt	timestamp	-	Report generation time

**Table 4.4 generated\_reports**

### Collection: rooms

Field Name	Type	Valid values	Description
roomId	string	-	Document ID for room
roomType	string	“Single”, “double”, “suite”	booked room type date
roomDescription	string	-	Brief description of the room
amenities	array	Predefined list (“WiFi”, “AC”, “TV”, etc.)	Room amenities
images	array	Only image URLs	Images of the room
pricePerNight	number	-	Room Cost per night
maxAdults	number	1, 2, 3, etc.	Maximum capacity for adult guests
maxChildren	number	1, 2, 3, etc.	Maximum capacity for children guests
available	boolean	“true”, “false”	Room availability for booking
createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

**Table 4.5 rooms table**

### Collection: reviews

Field Name	Type	Valid values	Description
reviewId	string	-	Document ID for review
guestId	string	Ref → guests/{guestId}	Reviewer guest ID reference
bookingId	string	Ref → bookings/{bookingId}	Related booking ID reference
hotelId	string	Ref → hotels/{hotelId}	Reviewed Hotel ID reference
starRate	number	1, 2, 3, 4, 5	Review rating score
review	string	-	Guest feedback for the stayed booking
createdAt	timestamp	-	Record creation time

updatedAt	timestamp	-	Last update time
-----------	-----------	---	------------------

**Table 4.6 reviews table**

### Collection: bookings

Field Name	Type	Valid values	Description
bookingId	string	-	Document ID for booking
guestId	reference	Ref → guests/{guestId}	Guest made the booking ID reference
guestName	string	-	Guest's name denormalized from guests collection
hotelId	string	Ref → hotels/{hotelId}	Related hotel ID reference
hotelName	string	-	Hotel's name denormalized from hotels collection
hotelState	string	-	Hotel's state denormalized from hotels collection
hotelCity	string	-	Hotel's city denormalized from hotels collection
roomId	string	Ref → rooms/{roomId}	Booked room ID reference
checkInDate	timestamp	-	Booking's check-in date
checkOutDate	timestamp	-	Booking's check-out date
adultsGuests	number	1, 2, 3, etc.	Number of adults guests
childrenGuests	number	1, 2, 3, etc.	Number of children guests
roomType	string	"Single", "double", "suite"	Type of the booked room
roomsQuantity	number	1, 2, 3, etc.	Number of booked rooms
totalAmount	number	-	Booking's total amount
bookingStatus	string	"confirmed", "cancelled", "pending", "completed"	Booking status
specialRequests	string	-	Guest's notes or extra requests
confirmationCode	string	-	Unique code for booking confirmation

createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

**Table 4.7 bookings table**

### Collection: admins

Field Name	Type	Valid values	Description
adminId	string	-	Document ID for Admin
FName	string	-	Admin's first name
LName	string	-	Admint's last name
email	string	Valid email format	Admint's email address
role	string	“hotel admin”, “ministry admin”	The user’s role in the system
active	boolean	“true”, “false”	The user’s account status
hotelId	string	Ref → hotels/{hotelId}	Hotel assigned ID reference
fcmToken	string	-	Firebase Cloud Messaging generated Token, an ID for sending push notifications to admins
settings	map	{“emailNotifications”: “true/false”, “pushNotification”: “true/false”, “smsNotifications”: “true/false”, “updatedAt”: “timestamp” }	Guest’s account settings, including notifications enabling and last time updated
hotelName	string	-	Hotel’s name denormalized from hotels collection
hotelState	string	-	Hotel’s state denormalized from hotels collection
hotelCity	string	-	Hotel’s city denormalized from hotels collection

hotelAddress	string	-	Hotel's location denormalized from hotels collection
createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

**Table 4.8 admins table**

### Subcollection: notifications

Field Name	Type	Valid values	Description
notificationId	string	-	Document ID for notification
title	string	-	Notification title
message	string	-	Notification body
type	string	“New user registration”, “new booking”, “new user status”	Notification’s content type
hotelId	string	Ref → hotels/{hotelId}	Related hotel ID reference
guestId	string	Ref → guests/{guestId}	Related Guest ID reference
bookingId	string	Ref → bookings/{bookingId}	Related booking ID reference
read	boolean	“true”, “false”	Read status
createdAt	timestamp	-	Record creation time
updatedAt	timestamp	-	Last update time

**Table 4.9 notifications table**

### Collection: ministry\_reports

Field Name	Type	Valid values	Description
generatedBy	string	-	Document ID of the admin generated the report
reportDataJson	string	-	The content of the report

reportType	string	“Platform Revenue Summary”, “Hotel Summary”, “Platform Booking Activity”, “Users”	Type of the report
startDate	timestamp	-	Start date of the wanted period for the report
endDate	timestamp	-	End date of the wanted period for the report
createdAt	timestamp	-	Report generation time

**Table 4.10 ministry\_reports**

### Collection: activities

Field Name	Type	Valid values	Description
actorID	string	Ref → guests/{GuestId} Ref → admins/{adminId}	ID reference of the user made the activity
actorName	string	-	Name of the user made the activity
description	string	-	description of the activity's context
entityType	string	“User”, “Guest”, “Hotel”, “Booking”	Entity type of the activity context
entityId	string	Ref → guests/{guestId} Ref → admins/{adminId} Ref → hotels/{hotelId} Ref → bookings/{bookingId}	ID reference of the entity's type of the activity
type	string	“User Status Update”, ”Hotel Status Update”, “New Booking”, “Booking Cancellation”, “New User Registration”	Type of the activity
timestamp	timestamp	-	Last update time

**Table 4.11 activities table**

# **CHAPTER FIVE**

# **RESULTS AND DISCUSSION**

# CHAPTER FIVE

## RESULTS AND DISCUSSION

### 5.1 User Prototype Interface

#### Guest mobile app screens

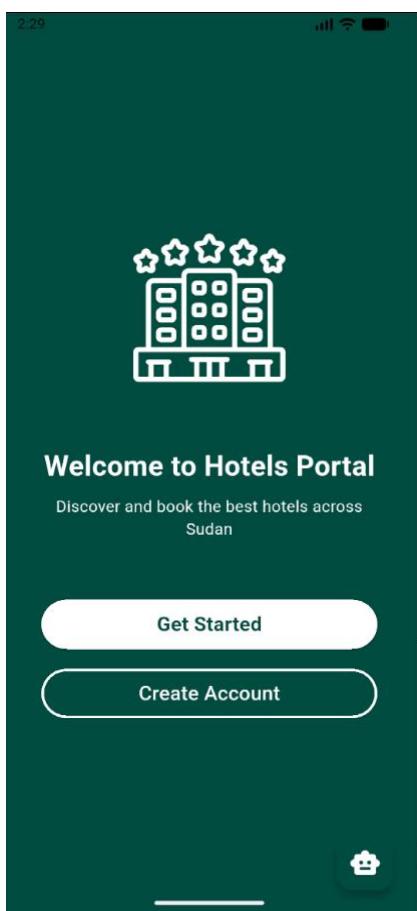


Figure 5.1: Guest-Welcome Screen

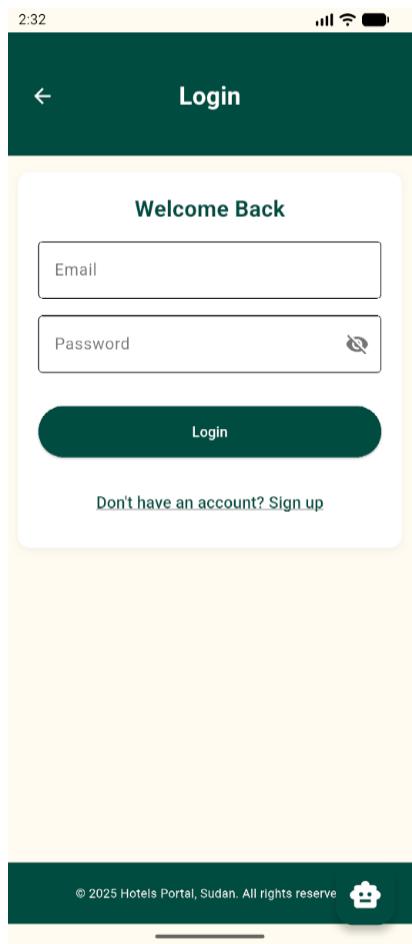


Figure 5.2: Guest-Login Screen

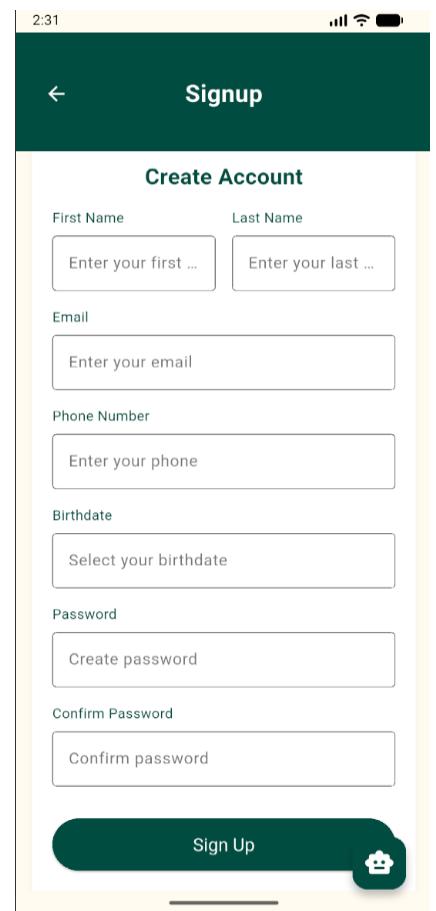


Figure 5.3: Signup Screen

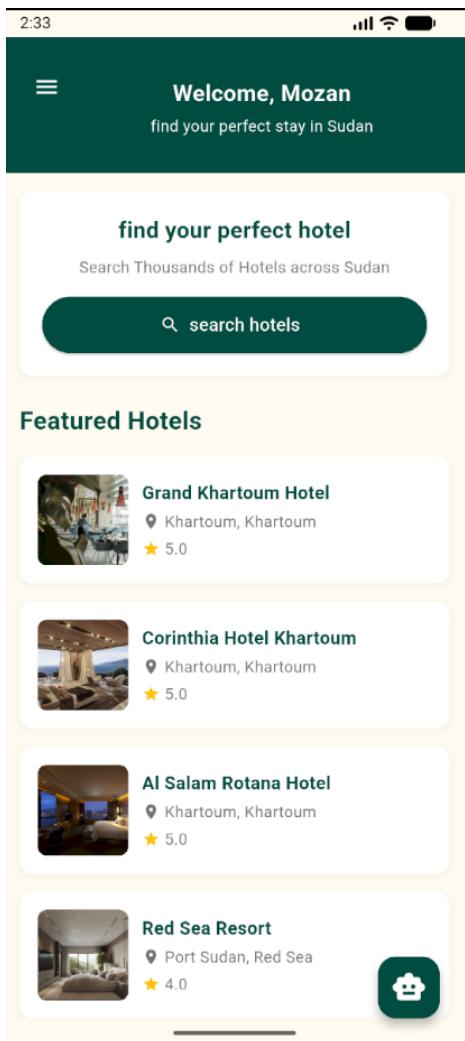


Figure 5.4: Home Screen (1)

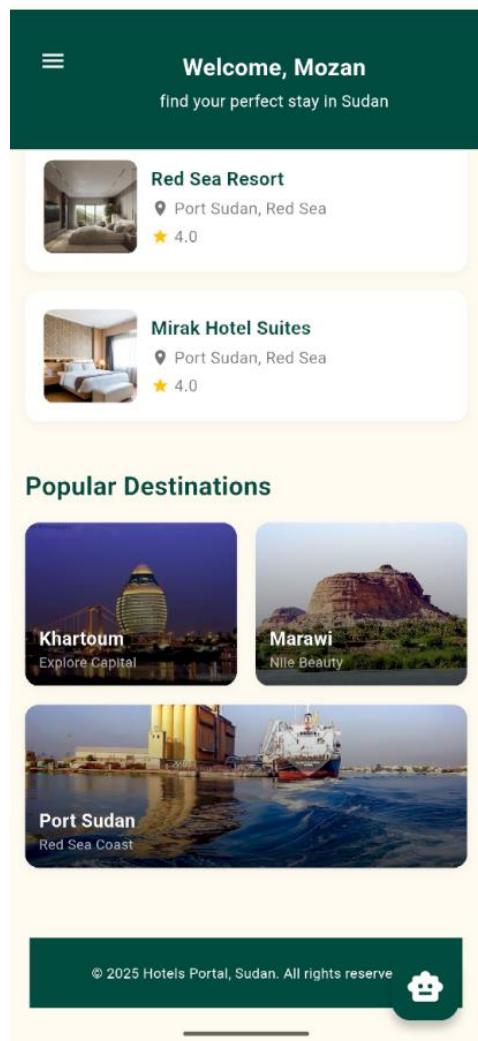


Figure 5.5: Home Screen (2)

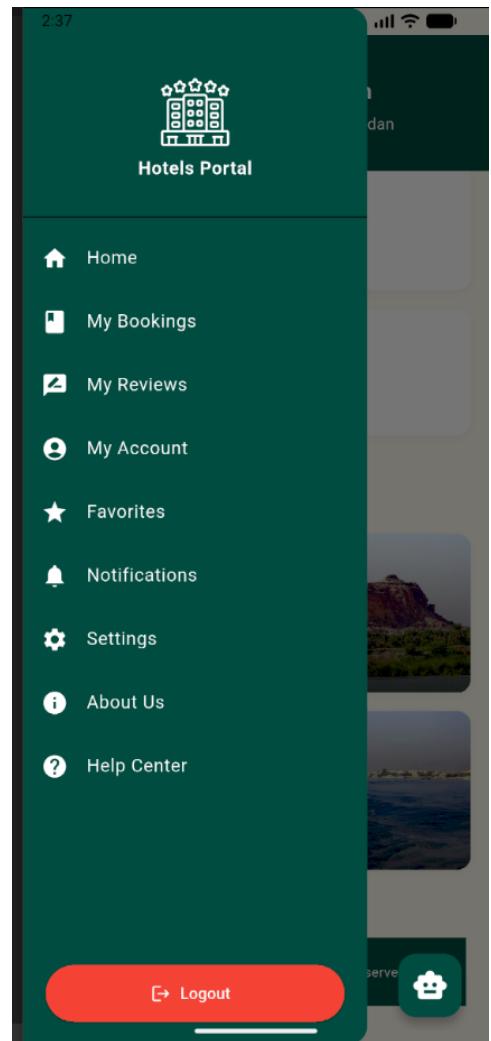


Figure 5.6: Sidebar Screen

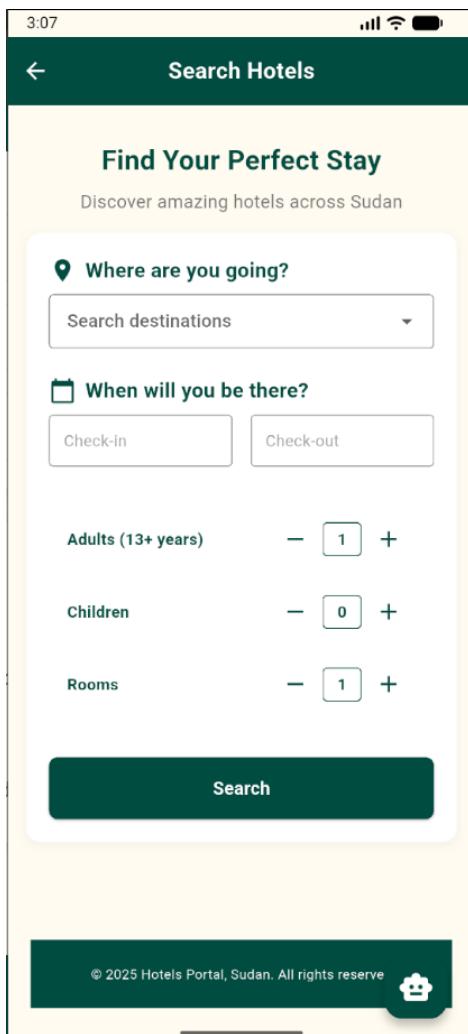


Figure 5.7: Search Hotels Screen

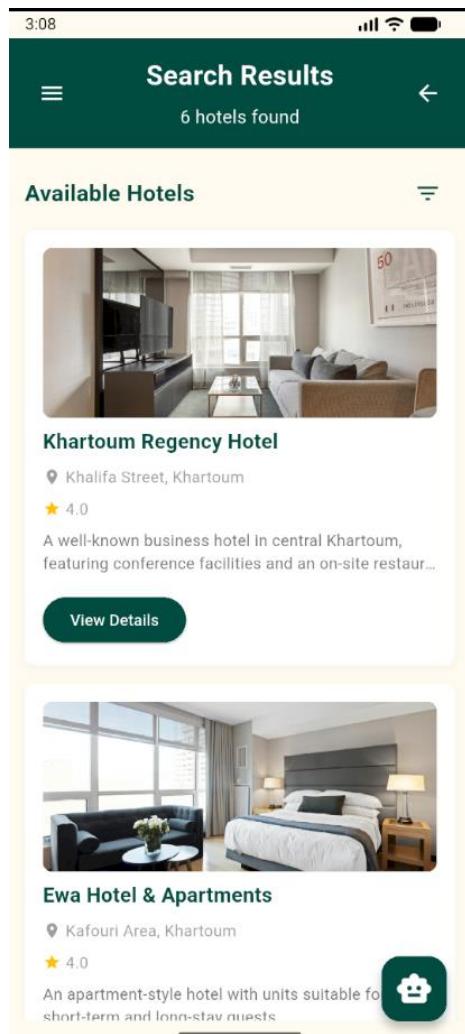


Figure 5.8: Search Results Screen

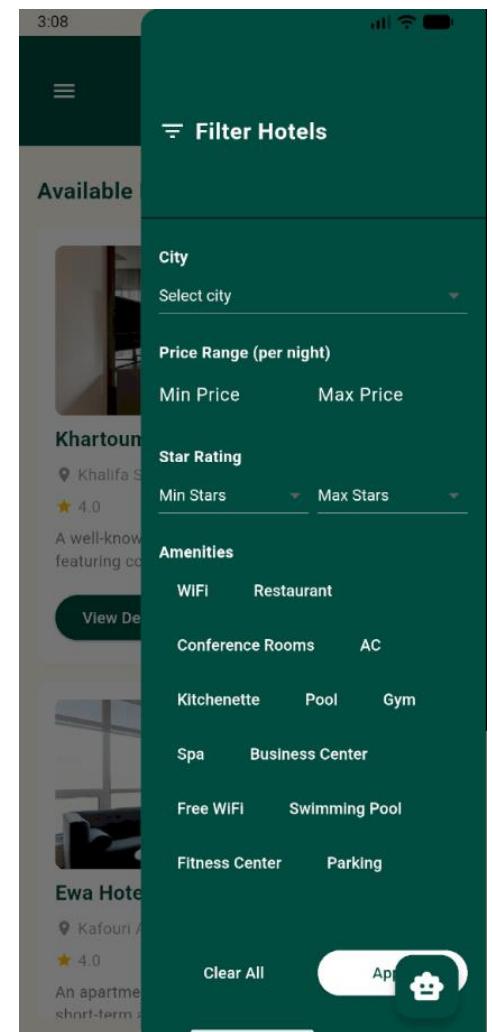


Figure 5.9: Filter Hotels Screen

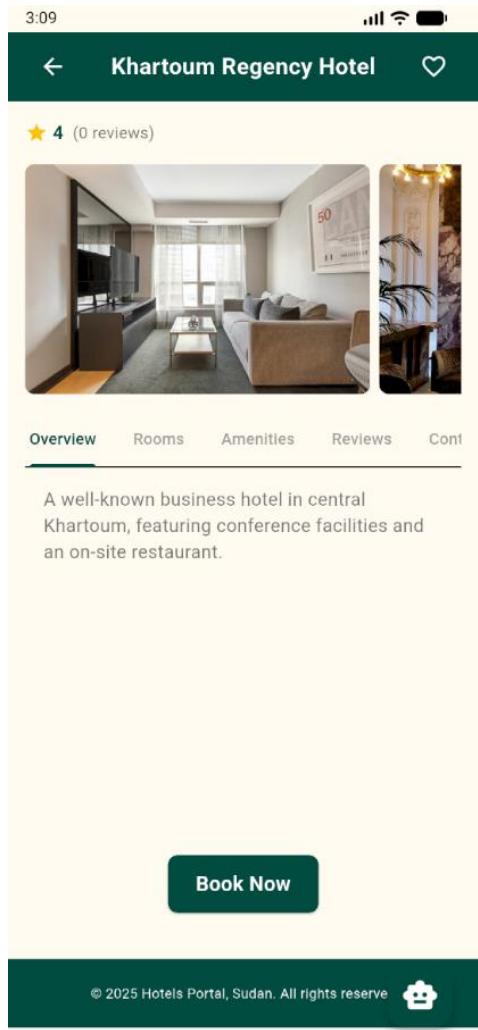


Figure 5.10: Hotels Info Screen

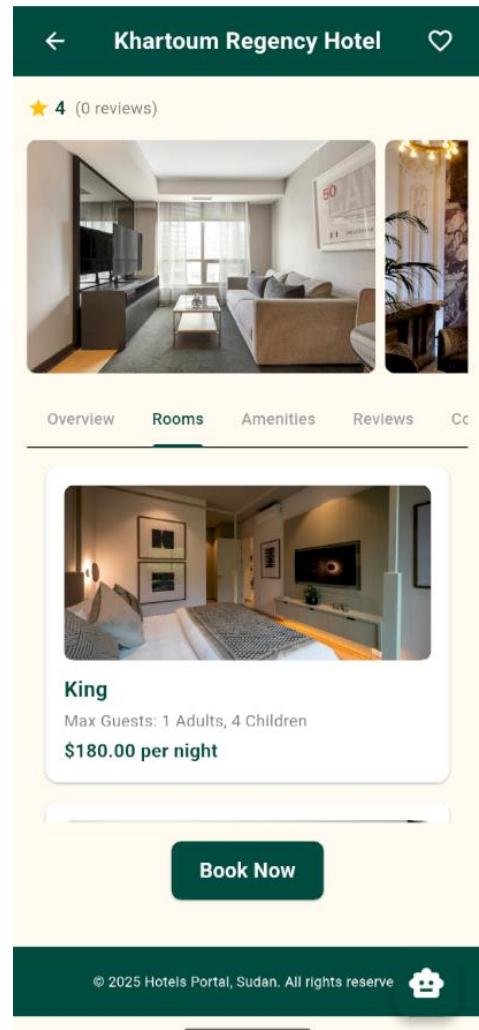


Figure 5.11: Hotel Rooms Screen

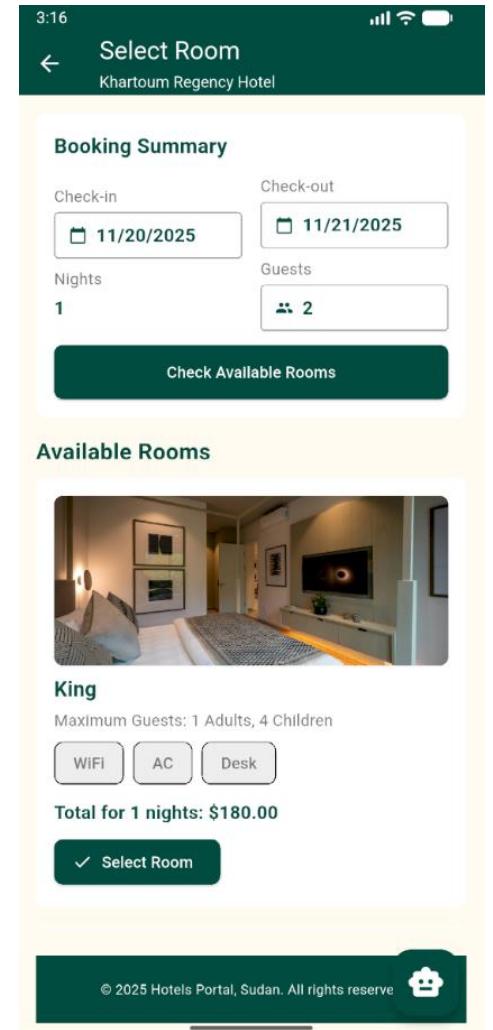


Figure 5.12: Select Rooms Screen

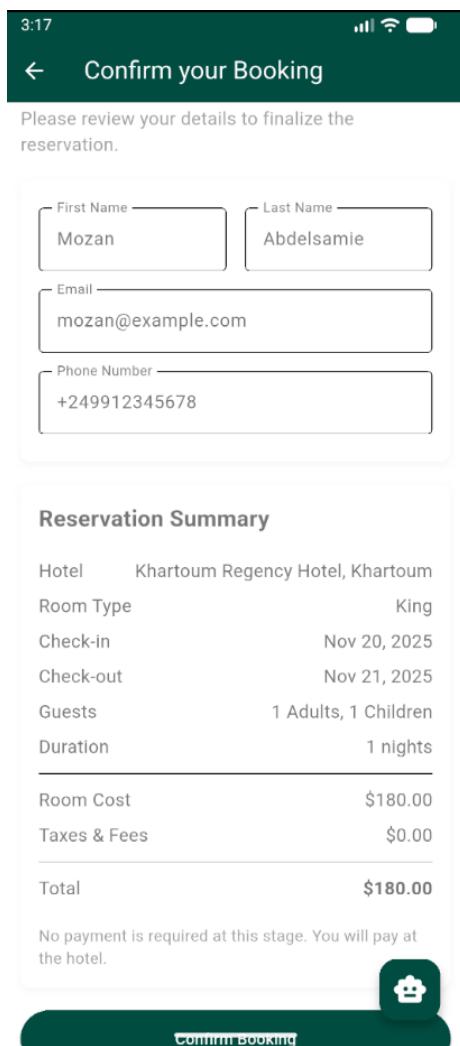


Figure 5.13: Confirm Booking Screen

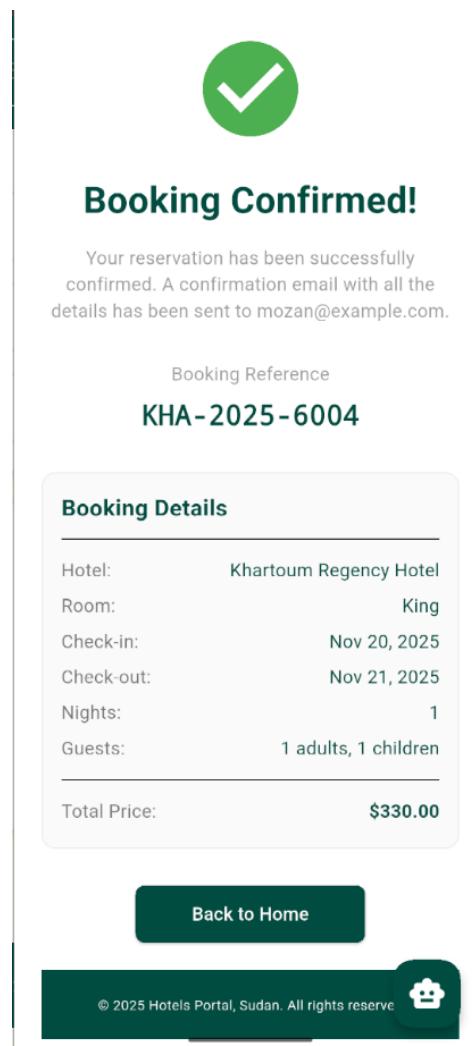


Figure 5.14: Booking Confirmed Screen

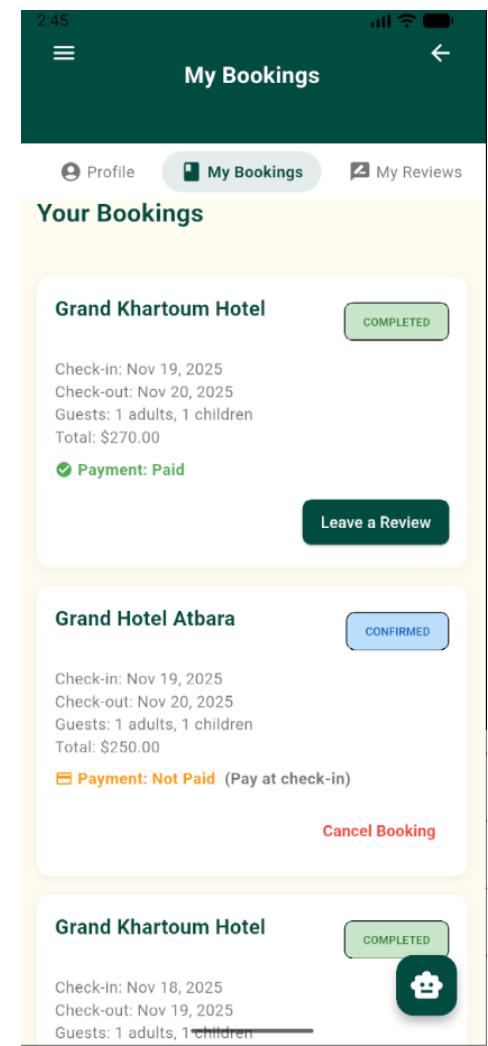


Figure 5.15: My Bookings Screen

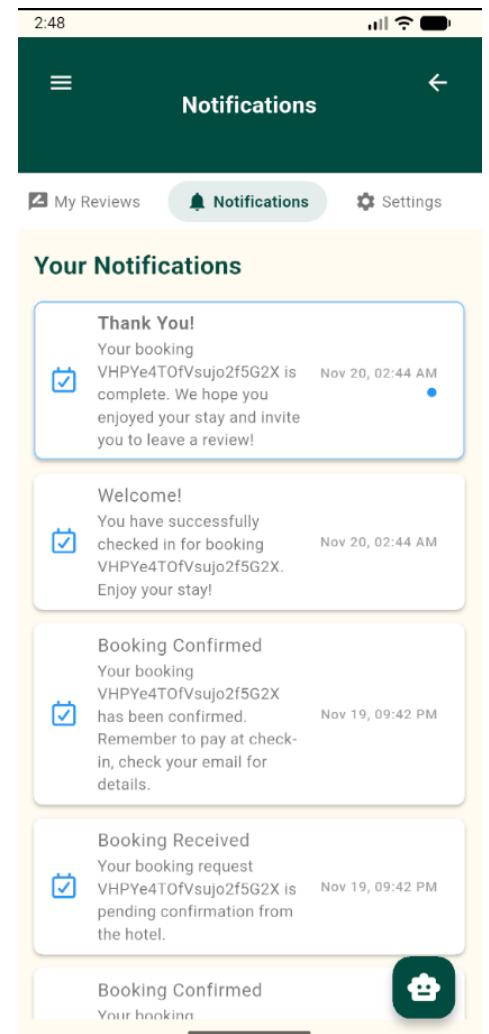
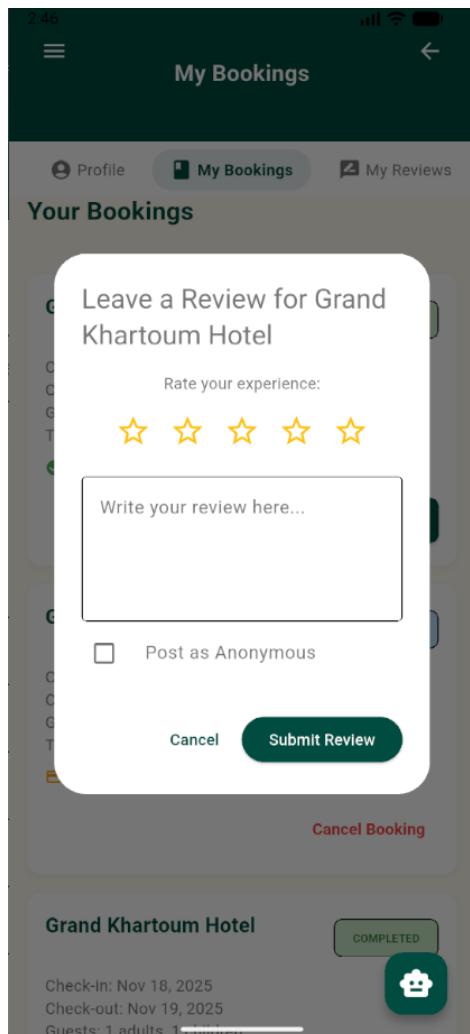


Figure 5.16: Leave Review Screen

Figure 5.17: My Reviews Screen

Figure 5.18: Guest-Notifications Screen

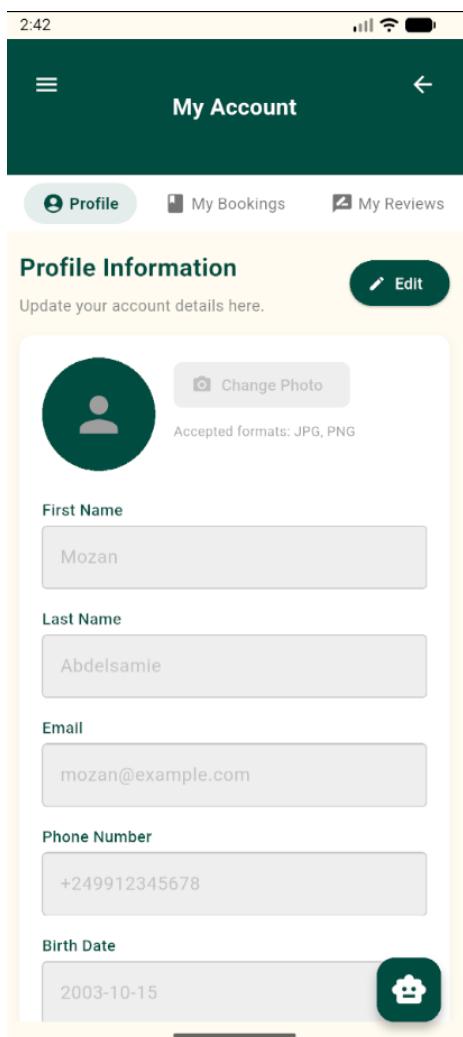


Figure 5.19: Guest-Profile Screen

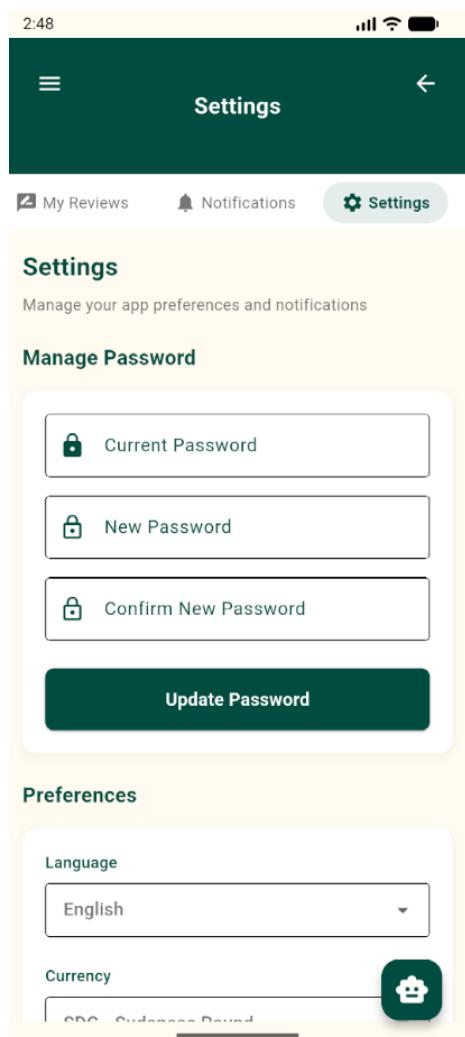


Figure 5.20: Guest-Settings Screen (1)

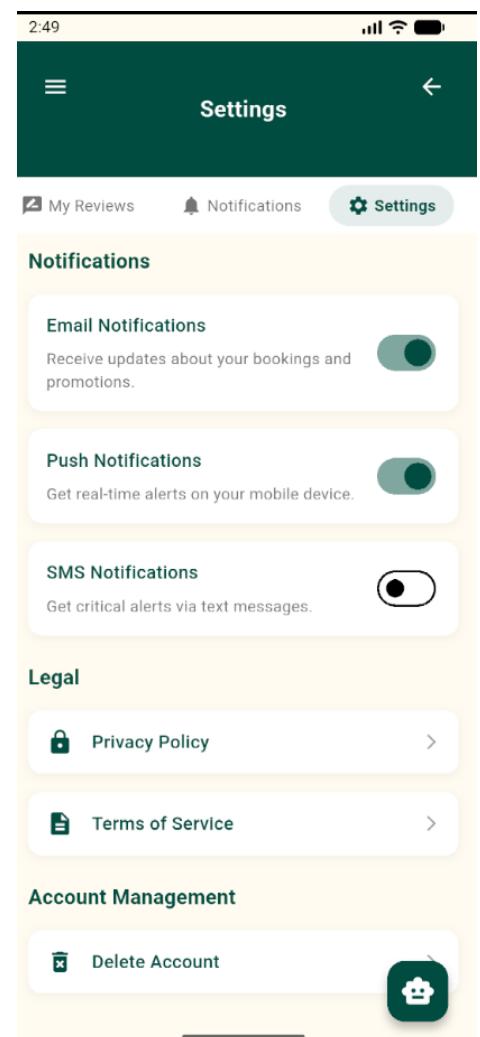


Figure 5.21: Guest-Settings Screen (2)

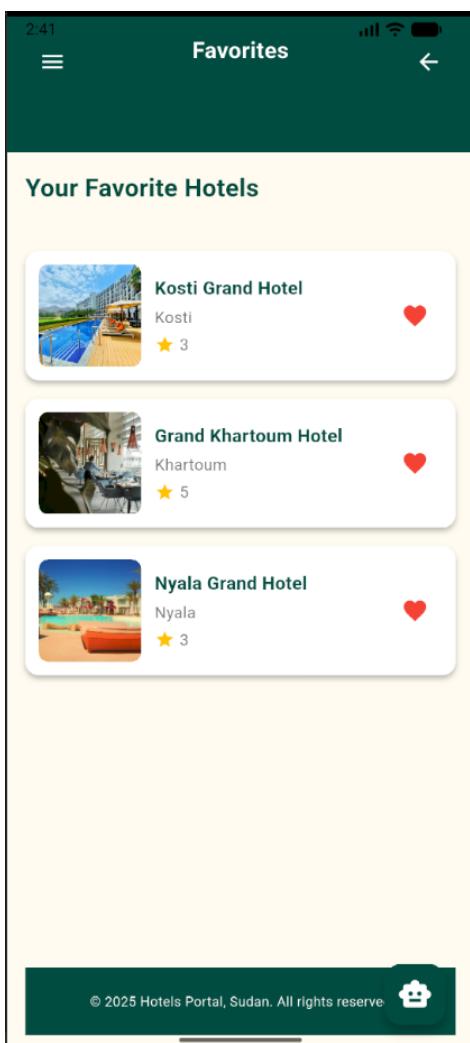


Figure 5.22: Favorites Screen

## About Us

### Hotels Portal

Your trusted gateway to discovering and booking hotels across Sudan.

#### Our Story

Founded in 2024, Hotels Portal started with a simple idea: to create a centralized, trustworthy platform for booking hotels specifically within Sudan. We saw a need for a modern, user-friendly system that connects travelers with the best accommodations available in the country.

From luxury resorts in Khartoum to cozy guesthouses in Port Sudan, our platform is designed to cater to every traveler's needs, ensuring a comfortable and memorable stay.

#### Our Mission

To provide a seamless, secure, and comprehensive hotel booking experience for travelers in Sudan.

#### Our Vision

To be the leading and most trusted hotel booking platform in Sudan.

Figure 5.23: About Us Screen (1)

platform for booking hotels specifically within Sudan. We saw a need for a modern, user-friendly system that connects travelers with the best accommodations available in the country.

hotel booking experience for travelers in Sudan.

**Our Vision**  
To be the leading and most trusted hotel booking platform in Sudan.

From luxury resorts in Khartoum to cozy guesthouses in Port Sudan, our platform is designed to cater to every traveler's needs, ensuring a comfortable and memorable stay.

**Developed by:**

Mohammed Jamal Abdalla (202003073)  
Mozan Abdelsamie Mohamed (202003115)

Supervisor: Dr. Emmalyn Capuno  
Faculty of Information Technology  
Future University – Sudan, 2025

© 2025 Hotels Portal, Sudan. All rights reserved.

Figure 5.24: About Us Screen (2)

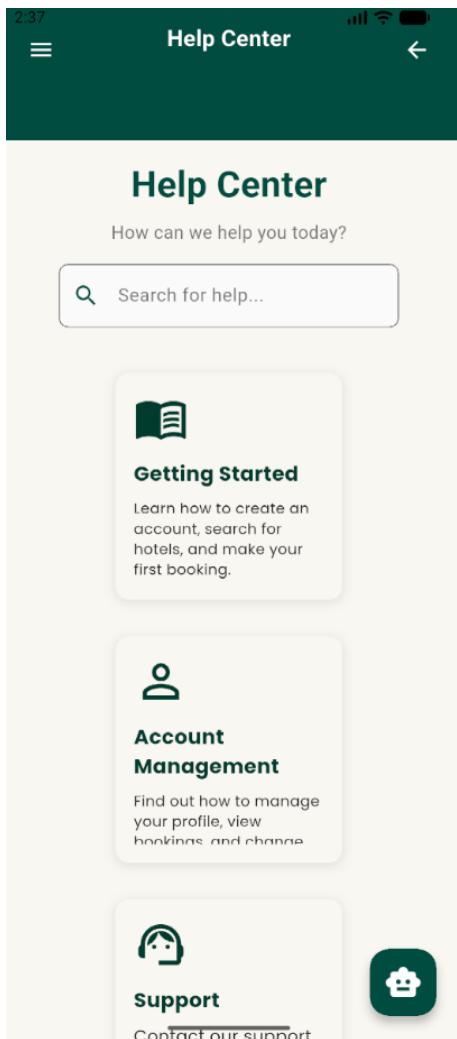


Figure 5.25: Help Center Screen (1)

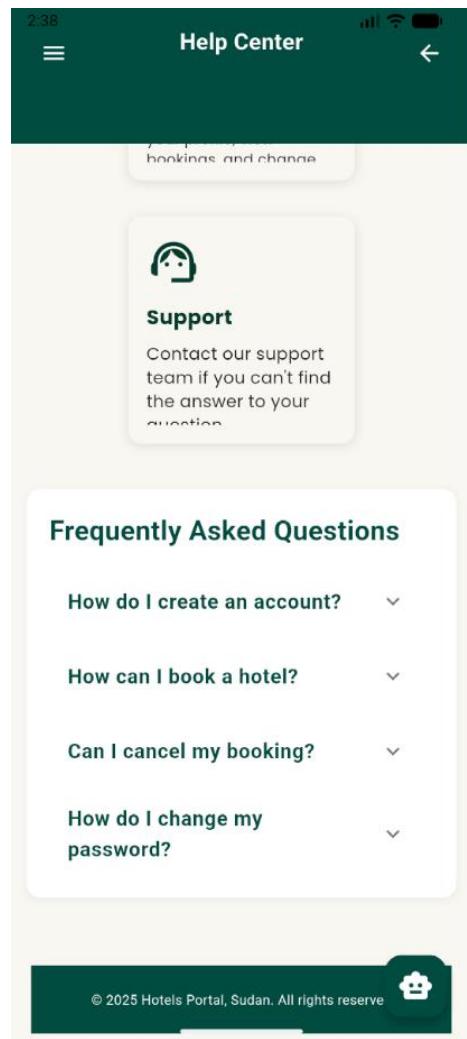


Figure 5.26: Help Center Screen (2)

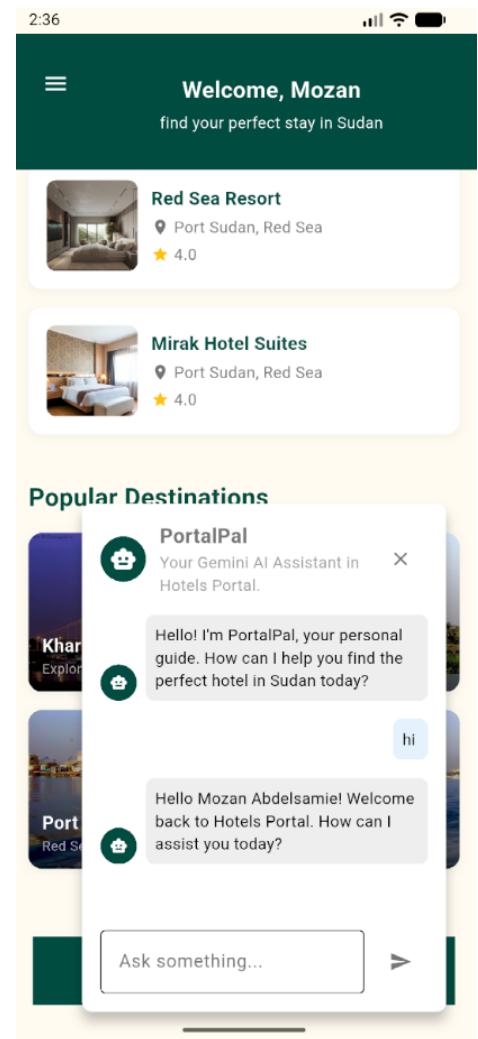


Figure 5.27: AI Assistant Screen

## Admins dashboard web app

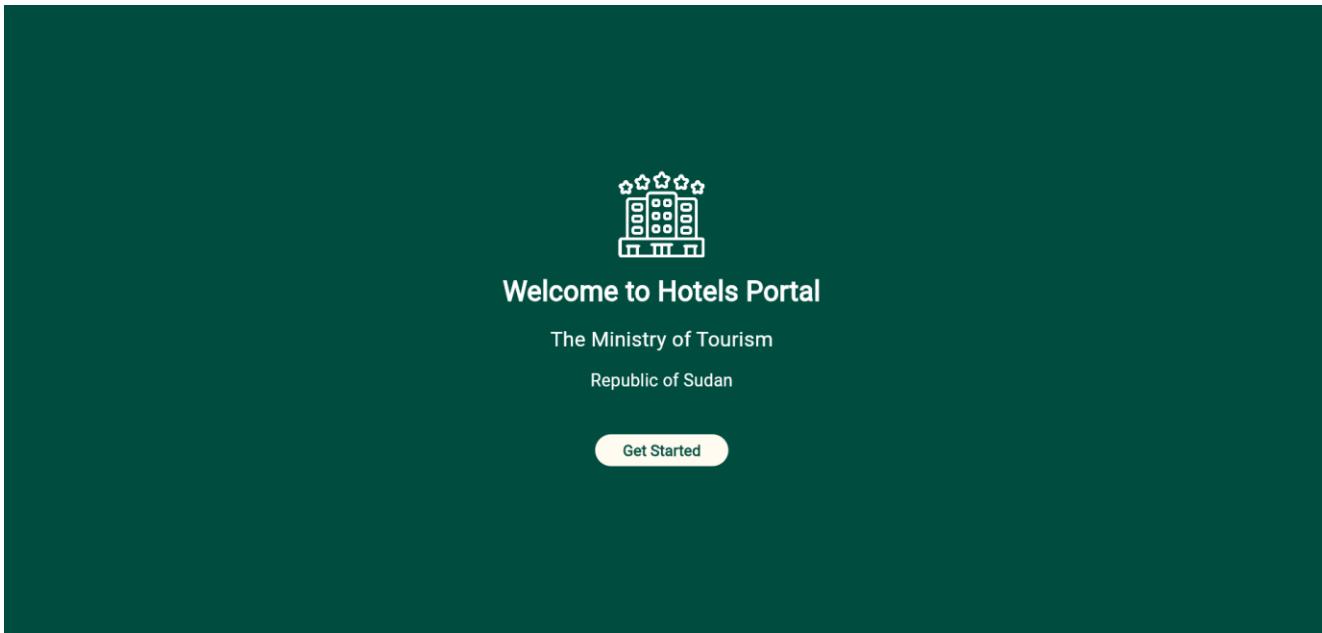


Figure 5.28: Admins-Welcome Page

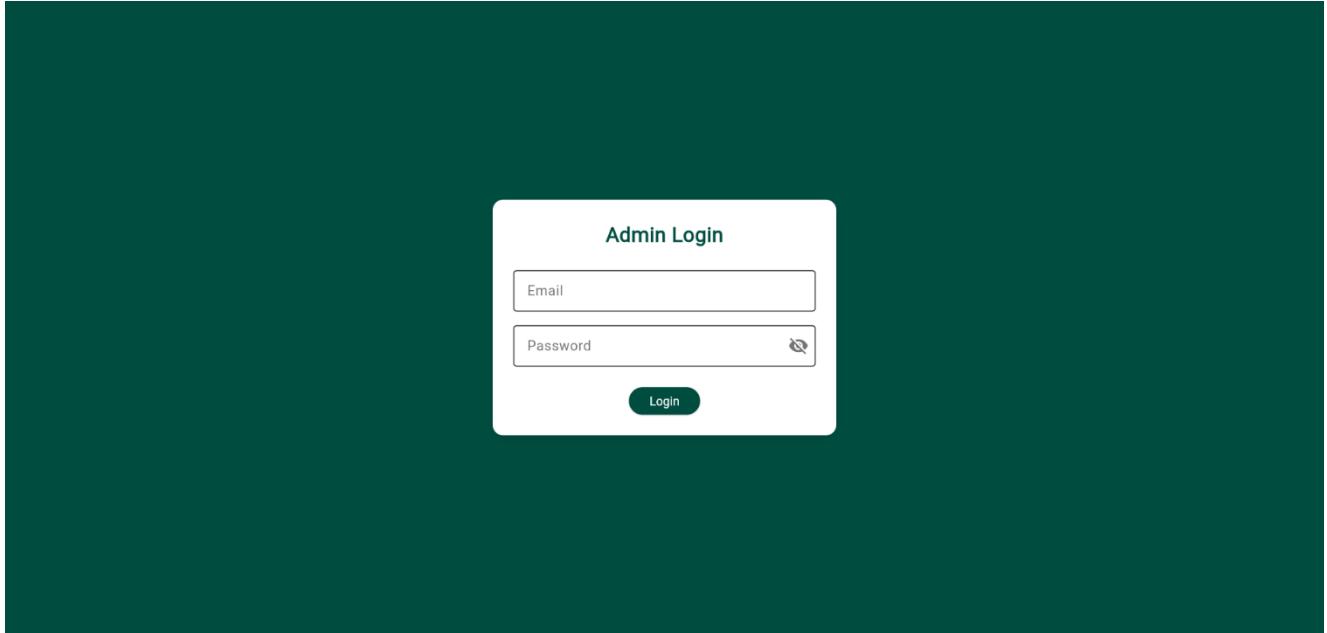


Figure 5.29: Admins-Login Page

## Ministry admin dashboard

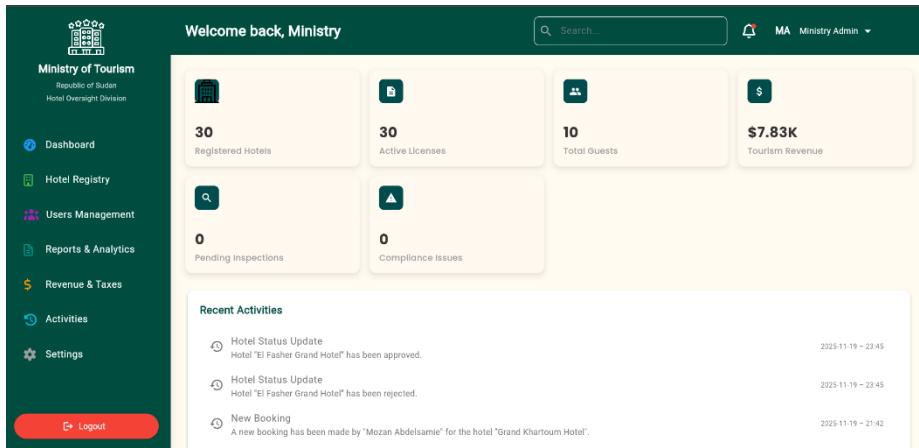


Figure 5.30: Ministry Dashboard Page (1)

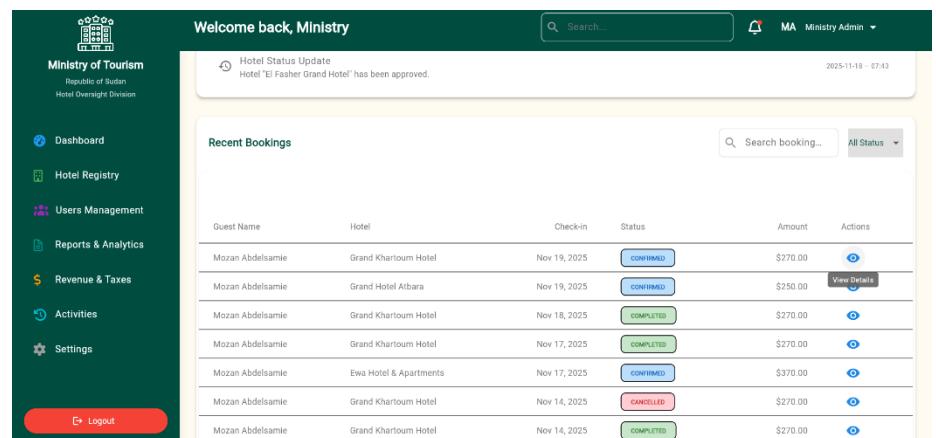


Figure 5.31: Ministry Dashboard Page (2)

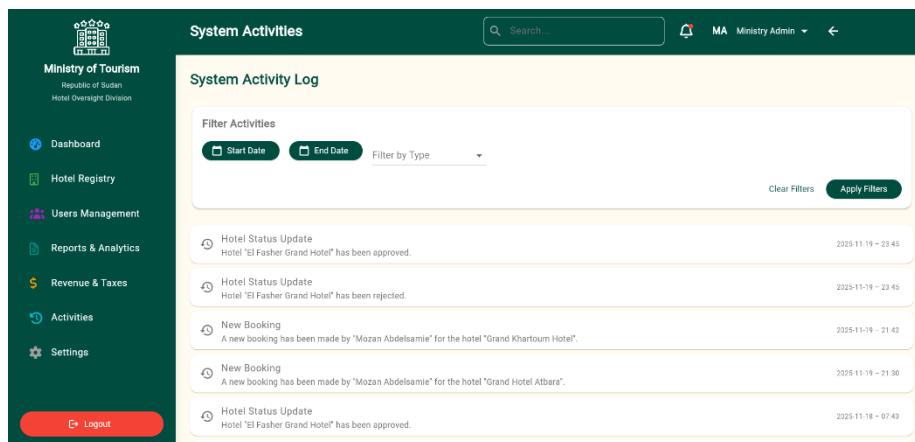


Figure 5.32: System Activities Page

**Users Management**

Search... MA Ministry Admin

Name	Email	Role	Status	Actions
5M Admin	admin.5m@hotelportal.com	hotel admin	Active	[Edit]
Abdelrahman Badreldin	abdo@mail.com	Guest	Active	[Edit]
Admin Rabak	admin.rabak@hotelportal.com	hotel admin	Active	[Edit]
AlSalam Director	director.alsalam@hotelportal.com	hotel admin	Active	[Edit]
All Osman	all.osman64@example.com	Guest	Active	[Edit]
Atbara Admin	admin.atbara@hotelportal.com	hotel admin	Active	[Edit]
Baasher Manager	manager.baasher@hotelportal.com	hotel admin	Active	[Edit]
Corinthia GM	gm.corinthia@hotelportal.com	hotel admin	Active	[Edit]
Damazin Manager	manager.damazin@hotelportal.com	hotel admin	Active	[Edit]
EdDeein Clerk	clerk.eddeein@hotelportal.com	hotel admin	Active	[Edit]

Figure 5.33: Users Management Page

**Hotel Registry**

Search... MA Ministry Admin

Name	Location	Rating	Status
El Fasher Grand Hotel	Al Fashir, North Darfur	★ 3.0	ACTIVE
Nyala Grand Hotel	Nyala, South Darfur	★ 3.0	ACTIVE
Imperial Hotel Wad Madani	Wad Madani, Gezira	★ 3.0	ACTIVE
Ewe Hotel & Apartments	Khartoum, Khartoum	★ 4.0	ACTIVE
Grand Hotel Atbara	Atbara, River Nile	★ 5.0	ACTIVE
Ed Daein Hotel	Ed Daein, East Darfur	★ 3.0	ACTIVE
Khartoum Regency Hotel	Khartoum, Khartoum	★ 4.0	ACTIVE
Sennar Hotel	Sennar, Sennar	★ 4.0	ACTIVE

Figure 5.34: Registered Hotels Page

**Register New Hotel**

Hotel Details

Hotel Name:

License Number:

Hotel Email:

State: Select a State:

City:

Hotel Address:

Admin Account Details

Admin First Name:

Admin Last Name:

Admin Login Email:

Admin Initial Password:

Cancel Register

Ministry Admin

El Fasher Grand Hotel	Al Fashir, North Darfur	★ 3.0	ACTIVE
Nyala Grand Hotel	Nyala, South Darfur	★ 3.0	ACTIVE
Imperial Hotel Wad Madani	Wad Madani, Gezira	★ 3.0	ACTIVE
Ewe Hotel & Apartments	Khartoum, Khartoum	★ 4.0	ACTIVE
Grand Hotel Atbara	Atbara, River Nile	★ 5.0	ACTIVE
Ed Daein Hotel	Ed Daein, East Darfur	★ 3.0	ACTIVE
Khartoum Regency Hotel	Khartoum, Khartoum	★ 4.0	ACTIVE
Sennar Hotel	Sennar, Sennar	★ 4.0	ACTIVE

Figure 5.35: Register New Hotel Page

The screenshot shows the 'Hotel Information & Status' section for 'El Fasher Grand Hotel'. On the left, a sidebar menu for the 'Ministry of Tourism' is visible, featuring icons for Dashboard, Hotel Registry, Users Management, Reports & Analytics, Revenue & Taxes, Activities, and Settings, along with a Logout button. The main content area displays the hotel's approval status ('Hotel Approved'), rating (3 stars), contact info (Phone: +249 555 777 222, Email: info@elfashergrand.com), address ('Main Street, Al-Fashir, North Darfur, Sudan'), and a central note about its frequent use by NGOs and regional travelers. A large 'Edit' button is located in the top right corner of this section.

Figure 5.36: Hotel Information Page (1)

This screenshot shows a different view of the hotel information page for 'El Fasher Grand Hotel'. It highlights the 'Facilities' and 'Restaurants' sections. The 'Facilities' section includes icons for WiFi, AC, and Restaurant. The 'Restaurants' section lists 'The Meeting Point' and 'Lobby'. The overall layout is similar to Figure 5.36, with the Ministry of Tourism sidebar on the left and a central content area.

Figure 5.37: Hotel Information Page (2)

The screenshot shows the 'Profile Settings' page for the 'Ministry Admin'. The left sidebar is identical to Figure 5.36. The main content area contains three sections: 'Admin Profile' (with fields for First Name: 'Ministry' and Last Name: 'Admin'), 'Change Password' (with fields for Current Password, New Password, and Confirm New Password), and 'Notification Settings' (which is currently empty). An 'Edit' button is located at the top right of the Admin Profile section.

Figure 5.38: Ministry-Profile Page

Ministry of Tourism  
Republic of Sudan  
Hotel Oversight Division

Reports & Analytics

Search... MA Ministry Admin

Reports Dashboard

+ New Custom Report

Recently Generated Reports

- Users  
Generated on: Nov 20, 2025  
Period: Nov 1, 2025 - Nov 19, 2025
- Hotel Summary  
Generated on: Nov 20, 2025  
Period: Nov 1, 2025 - Nov 19, 2025
- Platform Booking Activity  
Generated on: Nov 20, 2025  
Period: Nov 1, 2025 - Nov 19, 2025
- Platform Revenue Summary  
Generated on: Nov 20, 2025  
Period: Nov 1, 2025 - Nov 19, 2025
- Platform Revenue Summary  
Generated on: Nov 19, 2025  
Period: Nov 1, 2025 - Nov 19, 2025
- Platform Booking Activity

Figure 5.39: Ministry-Reports Dashboard Page

Ministry of Tourism  
Republic of Sudan  
Hotel Oversight Division

Custom Reports

Search... MA Ministry Admin

Generate a New Report

Report Type: Platform Revenue Summary

Start Date      End Date

Hotel (Optional): All Hotels

Generate Report

© 2025 Hotels Portal, Sudan. All rights reserved

Figure 5.40: Ministry-Custom New Reports Page

## Hotel admin dashboard

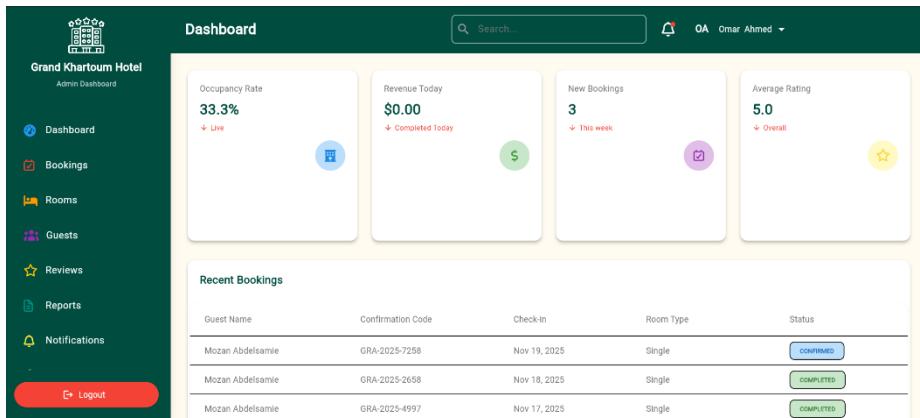


Figure 5.41: Hotel Dashboard Page (1)

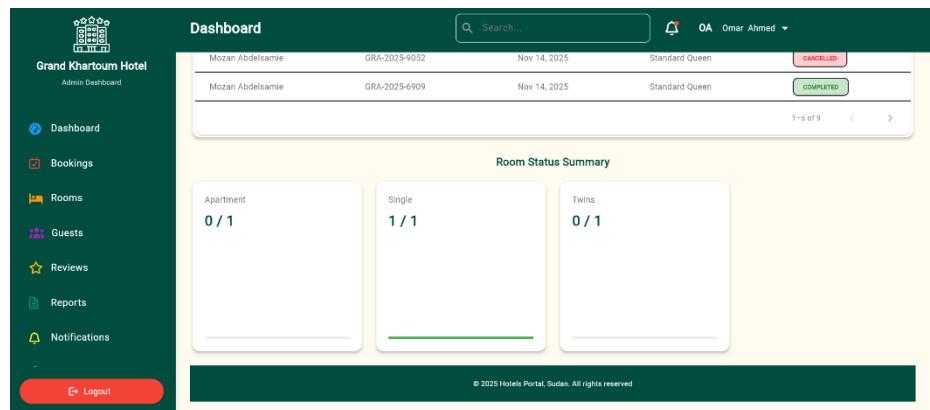


Figure 5.42: Hotel Dashboard Page (2)

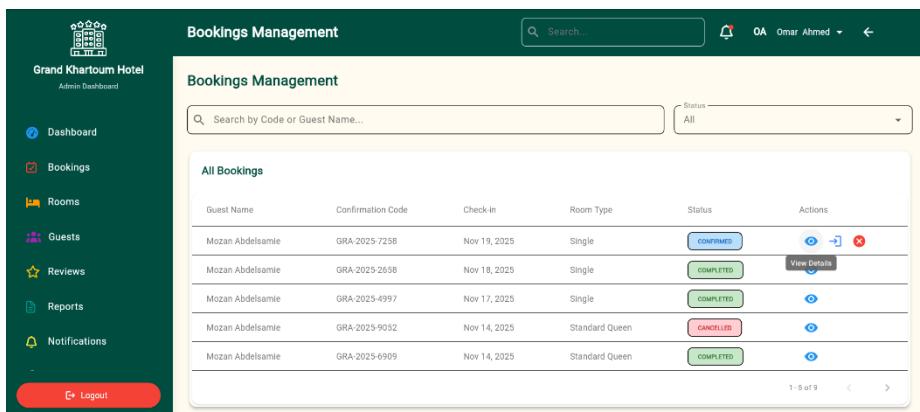


Figure 5.43: Hotel Bookings Management Page

**Booking Details**

Confirmation Code: GRA-2025-7258

**Guest Information**

Name:	Mozan Abdelsamie
Email:	mozan@example.com
Phone:	+249912345678

**Payment Summary**

Total Amount:	\$270.00
Payment Status:	Not Paid

**Booking Information**

Check-In Date:	Nov 19, 2025
Check-out Date:	Nov 20, 2025
Number of Rooms:	1
Guests:	1 Adults, 1 Children
Special Requests:	None

**Logout**

Figure 5.44: Booking Details Page (1)

**Booking Details**

**Booking Information**

Check-In Date:	Nov 19, 2025
Check-out Date:	Nov 20, 2025
Number of Rooms:	1
Guests:	1 Adults, 1 Children
Special Requests:	None

**Room Details**

Room Type:	Single
Price Per Night:	\$120.00
Max Guests:	4 Adults, 3 Children
Amenities:	WiFi, TV, AC

© 2025 Hotels Portal, Sudan. All rights reserved

Figure 5.45: Booking Details Page (2)

**Reviews**

**Customer Reviews**

<b>Mozan Abdelsamie</b>	Nov 18, 2025
★★★★★	nice

<b>Mozan Abdelsamie</b>	Nov 17, 2025
★★★★★	nice staff

<b>Mozan Abdelsamie</b>	Nov 14, 2025
★★★★★	good service

<b>Mozan Abdelsamie</b>	Nov 13, 2025
-------------------------	--------------

6 Reviews

**Logout**

Figure 5.46: Guests Reviews Page

Name	Email	Phone	Total Bookings	Last Visit
Mozan Abdelsamie	mozan@example.com	+249912345678	8	Nov 19, 2025
first last	user@example.com	123456789	1	Nov 05, 2025

Figure 5.47: Guests History Page

Room Type	Status
Twins	Unavailable
Single	Available
Apartment	Available

Figure 5.48: Room Management Page

Figure 5.49: Add New Room Page

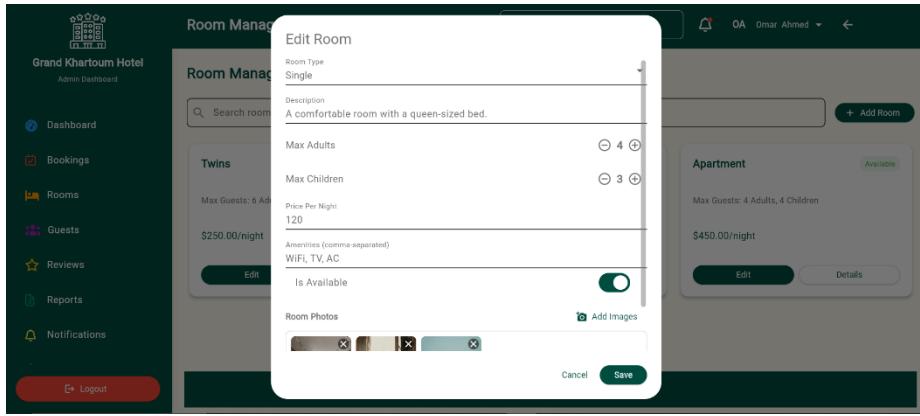


Figure 5.50: Edit Room Page

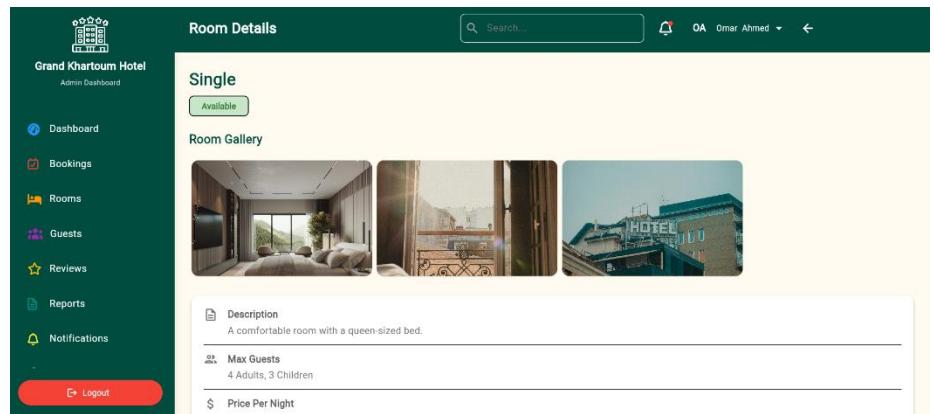


Figure 5.51: Room Details Page (1)

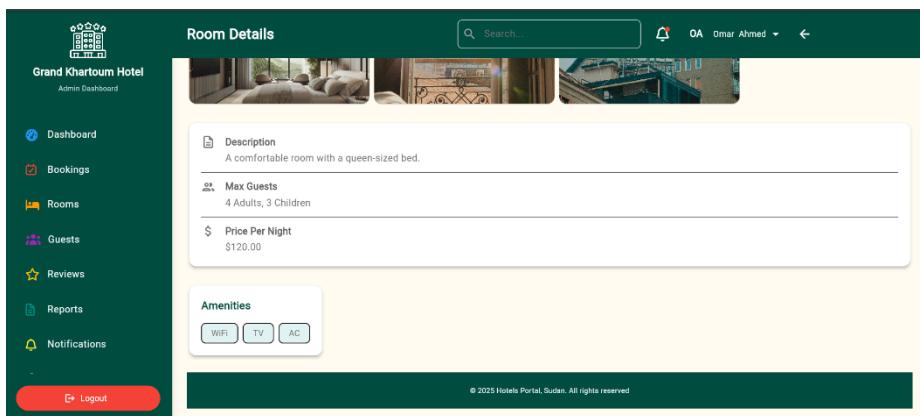


Figure 5.52: Room Details Page (2)

The screenshot shows the 'Profile Settings' page for the 'Grand Khartoum Hotel'. The left sidebar includes links for Admin Dashboard, Bookings, Rooms, Guests, Reviews, Reports, Notifications, and Settings, with 'Logout' at the bottom. The main content area has a header 'Profile Settings' with a search bar and user info (OA Omar Ahmed). It contains two sections: 'Admin Profile' (First Name: Omar, Last Name: Ahmed, Email: manager.khartoum@hotelportal.com) and 'Change Password' (Current Password, New Password, Confirm New Password).

Figure 5.53: Hotel-Profile Page (1)

The screenshot shows the 'Profile Settings' page for the 'Grand Khartoum Hotel'. The left sidebar includes links for Admin Dashboard, Bookings, Rooms, Guests, Reviews, Reports, Notifications, and Settings, with 'Logout' at the bottom. The main content area has a header 'Profile Settings' with a search bar and user info (OA Omar Ahmed). It contains three sections: 'Change Password' (Current Password, New Password, Confirm New Password), 'Notification Settings' (Email Notifications and Push Notifications, both toggled on), and a footer with a 'Save Changes' button.

Figure 5.54: Hotel-Profile Page (2)

The screenshot shows the 'Notifications' page for the 'Grand Khartoum Hotel'. The left sidebar includes links for Admin Dashboard, Bookings, Rooms, Guests, Reviews, Reports, Notifications, and Settings, with 'Logout' at the bottom. The main content area has a header 'Notifications' with a search bar and user info (OA Omar Ahmed). It displays a list of notifications under 'Your Notifications': 'New Booking Received' (Nov 19, 09:42 PM), 'New Booking Received' (Nov 18, 03:53 AM), 'New Booking Received' (Nov 17, 03:33 AM), 'Booking Cancelled' (Nov 14, 12:38 AM), 'New Booking Received' (Nov 14, 12:11 AM), and 'New Booking Received' (Nov 13, 03:40 AM).

Figure 5.55: Hotel-Notifications Page

**Grand Khartoum Hotel**

**Hotel Settings**

**Hotel Information**

- Hotel Name: Grand Khartoum Hotel
- Phone Number: +249 123 456 789
- Email Address: contact@grandkhartoum.com
- Address: 123 Nile Street, Khartoum

**Hotel Description**

A luxurious 5-star hotel offering world-class amenities and breathtaking views of the Nile.

**Profile** **Logout** **Edit**

Figure 5.56: Hotel-Settings Page (1)

**Grand Khartoum Hotel**

**Hotel Settings**

**Hotel Rating** ★★★★☆ 5.0 / 5.0

**Hotel Photos**

**Legal**

- Hotel Policy
- Privacy Policy
- Terms of Service

**Logout**

Figure 5.57: Hotel-Settings Page (2)

**Grand Khartoum Hotel**

**Reports & Analytics**

**Generate a New Report**

Report Type: Booking Summary

Start Date: [ ] End Date: [ ]

Generate Report

**Past Reports**

- Booking Summary: Generated on: Nov 20, 2025 Period: Nov 1, 2025 - Nov 19, 2025
- Revenue Report: Generated on: Nov 20, 2025 Period: Nov 1, 2025 - Nov 19, 2025
- Booking Summary: Generated on: Nov 20, 2025

Download as CSV **Download as PDF**

**Logout**

Figure 5.58: Hotel-Reports Page

## **5.2 Testing and Implementation**

This section presents the final evaluation and deployment stages of the Hotels Portal Reservation and Management System. It includes structured testing phases Alpha and Beta followed by system implementation and then Acceptability test. These stages ensure the system is functionally complete, user-ready, and aligned with project objectives.

### **5.2.1. Alpha and Beta Test Results**

Alpha testing is conducted internally near the end of development to identify and resolve bugs before external release. It focuses on verifying design integrity, functional accuracy, and system stability. This phase builds confidence in the product's readiness for user acceptance.

Beta testing is the second stage, where selected real users interact with the system under real-world conditions. Their feedback helps refine usability, performance, and compatibility. The goal is to ensure the system performs reliably across diverse environments and meets user expectations.

#### **5.2.1.1. Alpha Test Results**

The alpha test was performed by the internal development team using controlled scenarios. Key outcomes included:

- Verified core functions such as login, hotel search, booking, and admin dashboards.
- Identified and resolved major bugs affecting booking logic and UI responsiveness.
- Assessed app stability across Android devices using Flutter's emulator and physical testing.
- Validated registration workflows and role-based access for guests, hotel admins, and ministry officials.
- Evaluated performance metrics including load time, database sync, and resource usage.
- Collected feedback on design clarity and navigation flow for iterative UI improvements.

#### **5.2.1.2. Beta Test Results**

Beta testing involved external participants including students, hotel staff, and ministry reviewers. The system was tested in real-world conditions. Key outcomes included:

- Tested app functionality across various devices and network conditions.
- Collected user feedback on booking experience, dashboard usability, and navigation.
- Discovered additional bugs related to booking cancellation and review submission.
- Confirmed system reliability and data consistency under concurrent usage.
- Enhanced user interface based on feedback regarding layout and accessibility.
- Finalized fixes and optimizations for public launch readiness.

### **5.2.2. Implementation**

After successful testing and user acceptance, the implementation phase transitioned the system from development to deployment. This phase involved installing the mobile application on user devices and providing training for effective usage.

Users were guided through the following:

- How to install and navigate the mobile application.
- How to create secure accounts and input personal or hotel data.
- How to search for hotels, view profiles and hotel details, and make bookings.
- How hotel admins manage listings, update hotel information and room details, and respond to bookings.
- How ministry admins monitor system-wide activity and generate booking activities and revenue reports to aid in tourism analytics.

This structured rollout ensured smooth adoption, minimized onboarding challenges, and promoted efficient use of the platform across all user roles.

### **5.3 Acceptability Test**

Acceptability testing determines whether the system meets stakeholder expectations and project objectives. It evaluates user satisfaction, functional coverage, and readiness for deployment.

This phase assessed the system's alignment with project goals and user expectations.

#	Question	Strongly agree	Agree	Undecided	Disagree	Strongly Disagree
1	I think I would like to use this app frequently					
2	I found the app unnecessarily complicated					
3	I thought the app was easy to use					
4	I found the various functions in the app well integrated					
5	I thought there was too much inconsistency in the app					
6	I would imagine that most people would learn to use this app very quickly					
7	I felt very confident using the system					
8	I found the app very cumbersome to use					
9	I needed to learn a lot of things before I could get going with the app					

**Table 5.1 Acceptability Test**

## 5.4 Sample Generated Reports

This section presents examples of automated reports generated by the system's admin dashboards. Reports include 'Platform Revenue Summary', 'Hotel Summary', 'Platform Booking Activity', 'Users' for ministry admins, and 'Booking Summary', 'Revenue Report' for hotel admins.

### Ministry Admins Reports:

GRA-2025-69 09	Grand Khartoum Hotel	Mozan Abdelsamie	2025-11 -14	complet ed	\$270.0 0
GRA-2025-42 91	Grand Khartoum Hotel	Mozan Mohammed	2025-11 -13	complet ed	\$1050. 00
IMP-2025-269 2	Imperial Hotel Wad Madani	malaz ahmed	2025-11 -05	complet ed	\$250.0 0
KAS-2025-829 3	Kassala Hotel	Abdelrahman Badreldin	2025-11 -05	complet ed	\$265.0 0
GRA-2025-91 49	Grand Khartoum Hotel	mozan Abdelsamei	2025-11 -04	complet ed	\$270.0 0
Total:					\$2645

Prepared by: Mohammed Ali on 2025-11-20  
Email: admin@ministry.com

**Figure 5.59: Ministry- Platform Booking Activity Report**

Hotel Name	State	City	Star Rate	Status
Nyala Grand Hotel	South Darfur	Nyala	3	Approved
Rabak Hotel	White Nile	Rabak	3	Approved
Red Sea Resort	Red Sea	Port Sudan	4	Approved
Sennar Hotel	Sennar	Sennar	3	Approved
Zalingei Hotel	Central Darfur	Zalingei	3	Approved
Zanobia Hotel	North Kordofan	El Obeid	2	Approved

Prepared by: Mohammed Ali on 2025-11-20  
Email: admin@ministry.com

**Figure 5.60: Ministry-Hotel Summary Report**

### Ministry of Tourism - Republic of Sudan

Report Scope: Platform-Wide Report

#### Platform Revenue Summary

Description: A summary of platform revenue summary for period from: Nov 01, 2025 - To: Nov 19, 2025.

#### Summary:

Total Revenue: \$3245  
Total Bookings: 8  
Average Booking Value: \$405.625

#### Revenue by Room Type:

Standard Twin: \$265.00  
Standard: \$250.00  
Nile View Suite: \$1650.00  
Standard Queen: \$540.00  
Single: \$540.00

Prepared by: Mohammed Ali on 2025-11-20  
Email: admin@ministry.com

**Figure 5.61: Ministry-Platform Revenue Summary Report**

Name	Email	Role	Status	Created At
Regency Manager	manager.regency@hotelportal.com	Hotel Admin	Active	2025-10-29 02:56:59.557441
Resort Director	director.portsudan@hotelportal.com	Hotel Admin	Active	2025-10-29 02:24:03.728578
Sahil Manager	manager.sahil@hotelportal.com	Hotel Admin	Active	2025-10-29 02:38:12.021381
Sennar Admin	admin.sennar@hotelportal.com	Hotel Admin	Active	2025-10-29 02:56:09.483070
Tourism Director	director.merowe@hotelportal.com	Hotel Admin	Active	2025-10-29 02:34:58.788929
Zanobia Manager	manager.zanobia@hotelportal.com	Hotel Admin	Active	2025-10-29 02:44:03.562693
first last	user@example.com	Guest	Active	2025-11-05 01:37:34.392
malaz ahmed	malaz@mail.com	Guest	Active	2025-11-05 07:27:40.345

Prepared by: Mohammed Ali on 2025-11-20  
Email: admin@ministry.com

**Figure 5.62: Ministry-Users Report**

## Hotel Admins Reports:

### Ministry of Tourism - Republic of Sudan

Hotel Name: Grand Khartoum Hotel

#### Booking Summary

Description: A summary of booking summary for period from: Nov 01, 2025 - To: Nov 19, 2025.

Confirmation Code	Guest Name	Room Type	Check-in Date	Check-out Date	Status	Total Amount
GRA-2025-7 258	Mozan Abdelsamie	Single	2025-11-19	2025-11-20	confirmed	\$270.00
GRA-2025-2 658	Mozan Abdelsamie	Single	2025-11-18	2025-11-19	completed	\$270.00
GRA-2025-4 997	Mozan Abdelsamie	Single	2025-11-17	2025-11-18	completed	\$270.00
GRA-2025-9 052	Mozan Abdelsamie	Standard Queen	2025-11-14	2025-11-15	canceled	\$270.00
GRA-2025-6 909	Mozan Abdelsamie	Standard Queen	2025-11-14	2025-11-15	completed	\$270.00
GRA-2025-4	Mozan	Nile View	2025-11-	2025-11-1	completed	\$1050.00

**Figure 5.63: Hotel-Booking Summary (1) Report**

GRA-2025-6 909	Mozan Abdelsamie	Standard Queen	2025-11-14	2025-11-15	completed	\$270.00
GRA-2025-4 291	Mozan Mohammed	Nile View Suite	2025-11-13	2025-11-15	completed	\$1050.00
GRA-2025-8 997	first last	Nile View Suite	2025-11-05	2025-11-06	checked_in	\$600.00
GRA-2025-9 149	mozan Abdelsamei	Standard Queen	2025-11-04	2025-11-05	completed	\$270.00

Total: \$3270

Prepared by: Omar Ahmed on 2025-11-20  
Email: manager.khartoum@hotelportal.com

**Figure 5.64: Hotel-Booking Summary Report (2)**

### Ministry of Tourism - Republic of Sudan

Hotel Name: Grand Khartoum Hotel

#### Revenue Report

Description: A summary of revenue report for period from: Nov 01, 2025 - To: Nov 19, 2025.

#### Summary:

Total Revenue: \$2730  
Total Bookings: 6  
Average Booking Value: \$455

#### Revenue by Room Type:

Nile View Suite: \$1650.00  
Standard Queen: \$540.00  
Single: \$540.00

Prepared by: Omar Ahmed on 2025-11-20  
Email: manager.khartoum@hotelportal.com

**Figure 5.65: Hotel-Revenue Report**

# **CHAPTER SIX**

# **CONCLUSION AND**

# **RECOMMENDATIONS**

# CHAPTER SIX

## CONCLUSION AND RECOMMENDATIONS

### 6.1 Conclusion

This project aimed to develop a **Hotels Portal Reservation and Management System for the Ministry of Tourism, Antiquities, and Wildlife in Sudan**. The system was designed to support the recovery and modernization of Sudan's hospitality sector by delivering a unified digital platform that connects travelers, locals, hoteliers, and ministry administrators. Through the implementation of both mobile and web modules, the project aimed to improve hotel accessibility, streamline booking processes, and enhance digital infrastructure in preparation for Sudan's economic and tourism revival.

The following points are the conclusions which were derived based on the findings of the developed system:

- Developed a **mobile application** for secure user accounts (guest/admin), hotel search, detailed hotel profiles, real-time booking, and account/booking management.
- Built a **web-based ministry admin dashboard** for hotel approvals, monitoring bookings, user activity, and generating tourism and revenue reports.
- Built **web-based hotel admin dashboard** to manage hotel details, rooms, images, and booking requests efficiently.
- Integrated **reporting modules** for both the Ministry and hoteliers to track trends, bookings, and revenue.
- Successfully achieved all objectives, delivering a **functional system** that supports post-war tourism recovery in the hotels department that enhances Sudan's hospitality digital infrastructure.

### 6.2 Recommendations

The following recommendations are suggested by the researchers.

1. **For the project site:** It's recommended that the Ministry of Tourism should implement and use the system and also launch it to their clients, to meet client's needs and be more productive.
2. **For future researches:** It's recommended to empower new features such as:
  - a. **Implement secure online payment integration:** Add a secure online payment system that protects customers' card details (or a mobile money option common in the region) to allow confirmed, paid bookings.
  - b. **Add Arabic localization (and multilingual support):** Translate UI strings and align layouts for right-to-left rendering to widen adoption among local users.
  - c. **Plan iOS support:** Extend the Flutter app testing and package for iOS to serve a broader user base.

- d. **Integrate optional hotel management software connectivity:** Provide APIs or lightweight connectors so larger hotels can sync inventory and pricing from their own systems, reducing the risk of double-booking.
- e. **Implement the functions and screen for placeholders in the mobile app:** Some of the mobile app sections currently contain placeholder interfaces. These should be developed into fully functional screens to ensure completeness and smooth user navigation.
- f. **Add tours recommendations & programs feature:** Introduce a “Tours & Recommendations” module that displays tourist destinations, cultural attractions, and nearby services.

# **BIBLIOGRAPHY**

# BIBLIOGRAPHY

## 7.1 References:

1. <https://www.sapientcodelabs.com/blogs/evolution-of-web-development-journey-from-90s-to-now>  
Obtained on 15/4/2025
2. <https://www.geeksforgeeks.org/blogs/history-and-evolution-of-web-development/>  
Obtained on 14/5/2025
3. <https://www.appsrhino.com/blogs/evolution-of-mobile-apps> Obtained on 15/4/2025
4. <https://evincedev.com/blog/impact-of-the-mobile-apps-on-travel-and-tourism-industry/>  
Obtained on 15/4/2025
5. <https://innow8apps.com/blog/mobile-apps-transforming-hospitality/> Obtained on 15/4/2025
6. <https://www.tekrevol.com/blogs/mobile-apps-transforming-the-hospitality-industry/>  
Obtained on 17/4/2025
7. [https://www.researchgate.net/publication/274079295\\_Online\\_Hotel\\_Reservation\\_System](https://www.researchgate.net/publication/274079295_Online_Hotel_Reservation_System)  
Obtained on 17/4/2025
8. <https://web.acips.org/sudans-war-and-its-devastating-impact-on-infrastructure/>  
Obtained on 17/4/2025
9. <https://www.idea.int/sites/default/files/2023-09/war-in-sudan-15-april-2023-background-analysis-and-scenarios.pdf> Obtained on 18/4/2025
10. <https://www.policycenter.ma/publications/ongoing-war-sudan-and-its-implications-security-and-stability-horn-africa-and-beyond> Obtained on 18/4/2025
11. [https://sudan.un.org/sites/default/files/2019-10/180520\\_WH\\_sites\\_Final\\_en\\_0.pdf](https://sudan.un.org/sites/default/files/2019-10/180520_WH_sites_Final_en_0.pdf)  
Obtained on 20/4/2025
12. <https://dspace.spbu.ru/bitstreams/507c502f-67a0-463f-a5ea-ab68e6169f0a/download>  
Obtained on 20/4/2025
13. [https://link.springer.com/chapter/10.1007/978-3-030-10871-7\\_11](https://link.springer.com/chapter/10.1007/978-3-030-10871-7_11) Obtained on 23/4/2025
14. <https://www.ijiset.com/> Obtained on 23/4/2025
15. <https://doi.org/10.31033/ijemr.12.5.27> Obtained on 14/5/2025
16. [https://www.researchgate.net/publication/365151133\\_Research\\_on\\_Hotel\\_Management\\_System](https://www.researchgate.net/publication/365151133_Research_on_Hotel_Management_System)  
Obtained on 23/5/2025
17. <https://developers.amadeus.com/blog/flutter-hotel-booking> Obtained on 23/5/2025
18. <https://dart.dev/multiplatform-apps> Obtained on 23/5/2025

## **7.2 Attachments:**

- [www.flutter.dev](http://www.flutter.dev)
- [www.dart.dev](http://www.dart.dev)
- [www.firebaseio.google.com](http://www.firebaseio.google.com)
- [www.github.com](http://www.github.com)
- [www.figma.com](http://www.figma.com)
- [www.drawio.com](http://www.drawio.com)
- [www.stackoverflow.com](http://www.stackoverflow.com)
- [www.coursera.org](http://www.coursera.org)
- [www.youtube.com](http://www.youtube.com)

## **7.3 E-Books:**

**Author,year, book title, edition, publishing information**

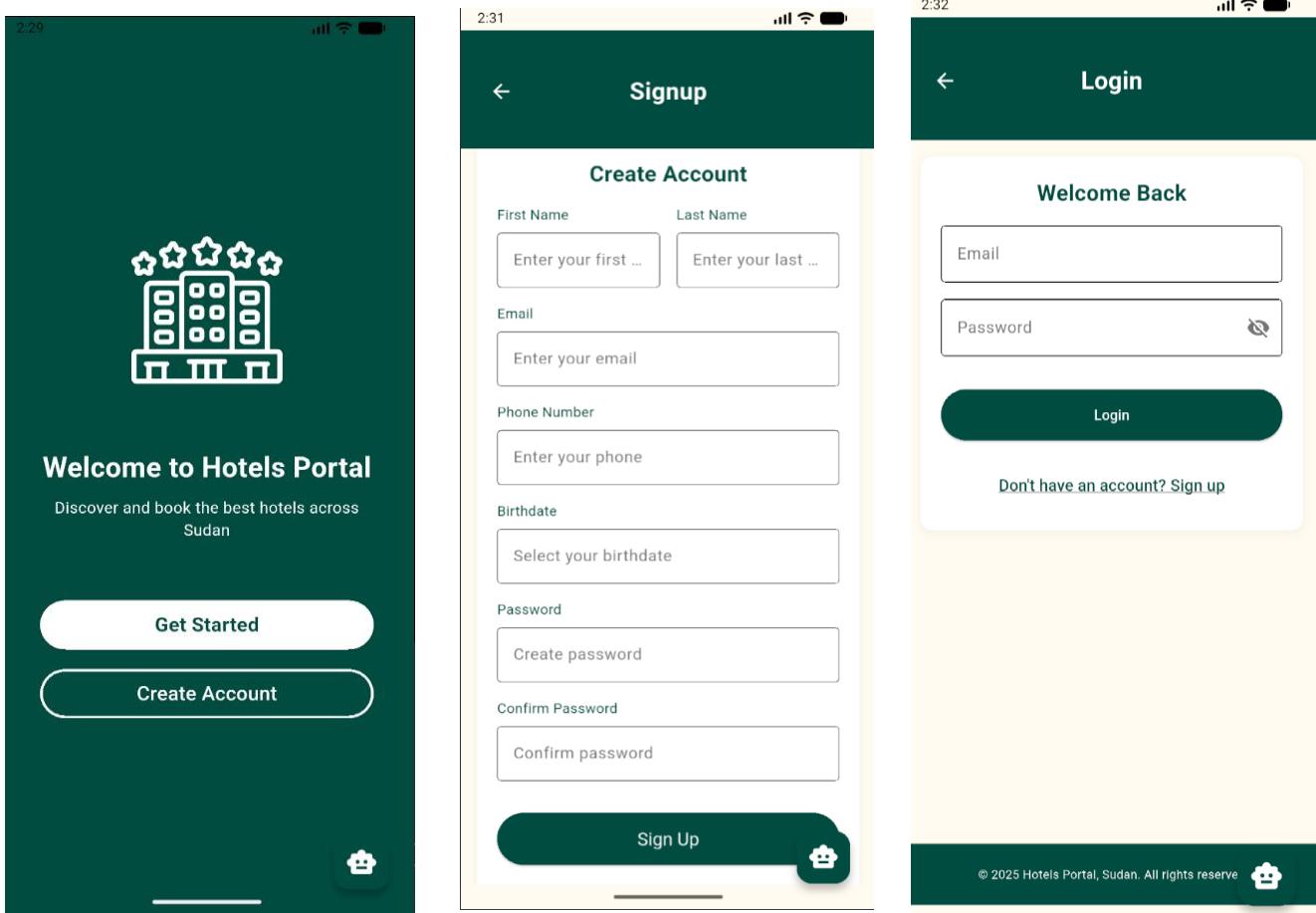
- Michael Blaha, James Rumbaugh, 2005, Object-Oriented Modeling and Design with UML, Second Edition, Pearson Education.
- Thomas Bailey, 2023, Alessandro Biessek, Flutter for Beginners, First Edition, Packt Publishing.
- Chopra Rajiv, 2016, Database Management System (DBMS): A Practical Approach, 5th Edition, S. Chand Publishing.
- Ashok Kumar S, 2021, Firebase Essentials – Android Edition, First Edition, Packt Publishing.
- K. N. Krishnaswamy, 2012, Hotel Management and Operations, Second Edition, Wiley India.

# **APPENDACIES**

# APPENDACIES

## 8.1 USER MANUAL

The following imagery is describing the interface that the user should expect while using the system.



### 1/ Guest-Welcome Screen:

- 1.1 open welcome screen.
- 1.2 click create account button.
- 1.3 go to SignUp.
- 1.4 Enter your credentials.
- 1.5 Click SignUp button.
- 1.6 Go to home Screen.

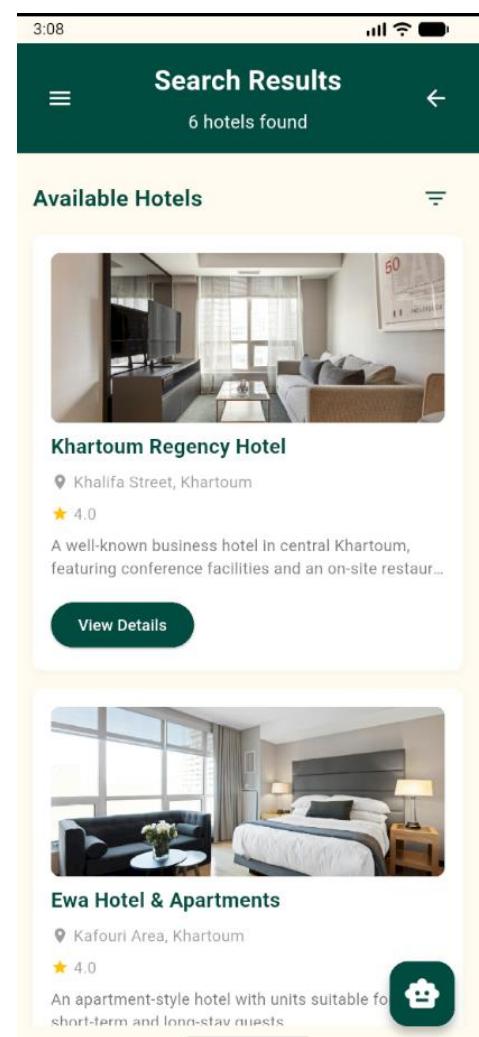
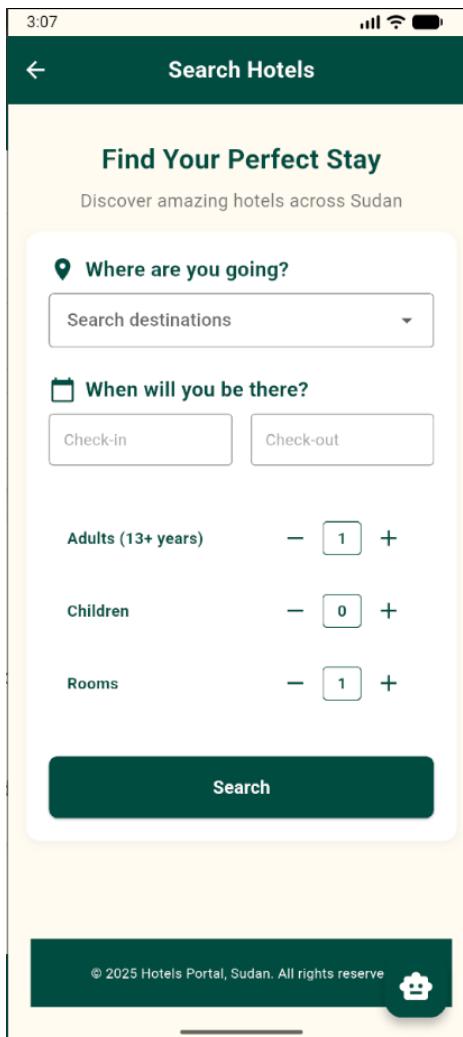
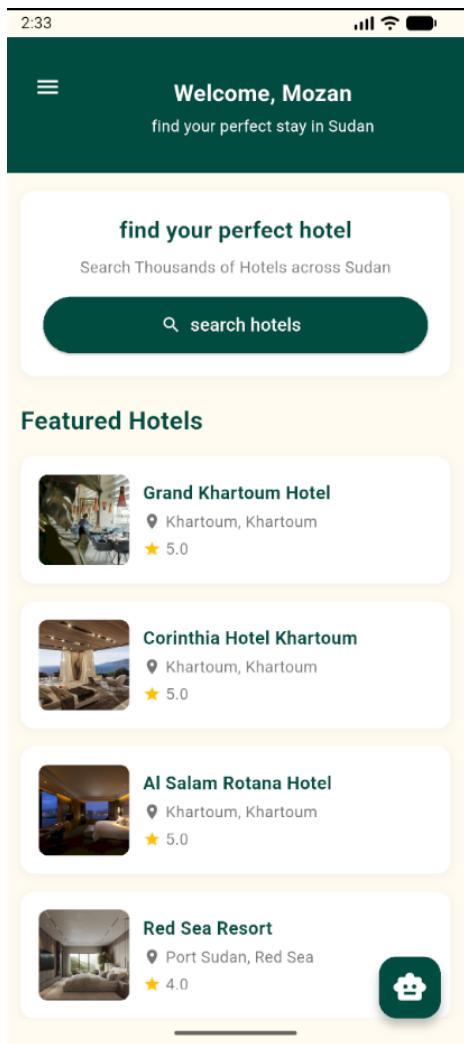
### 2/ SignUp Screen:

- 2.1 click create account button from welcome screen for new account.
- 2.2 enter you first name, last name.
- 2.3 enter your email.
- 2.4 enter your phone number.
- 2.5 enter your birthdate.
- 2.6 enter your password.
- 2.7 enter your password again to confirm.
- 2.8 click SignUp button.
- 2.9 go to home screen.

### 3/ Guest-Login Screen:

- 3.1 click get started from welcome screen if already have account.
- 3.2 enter your email.
- 3.3 enter your password.
- 3.4 click Login button.
- 3.5 go to home screen.

## Booking Process:



### 4/ Home Screen:

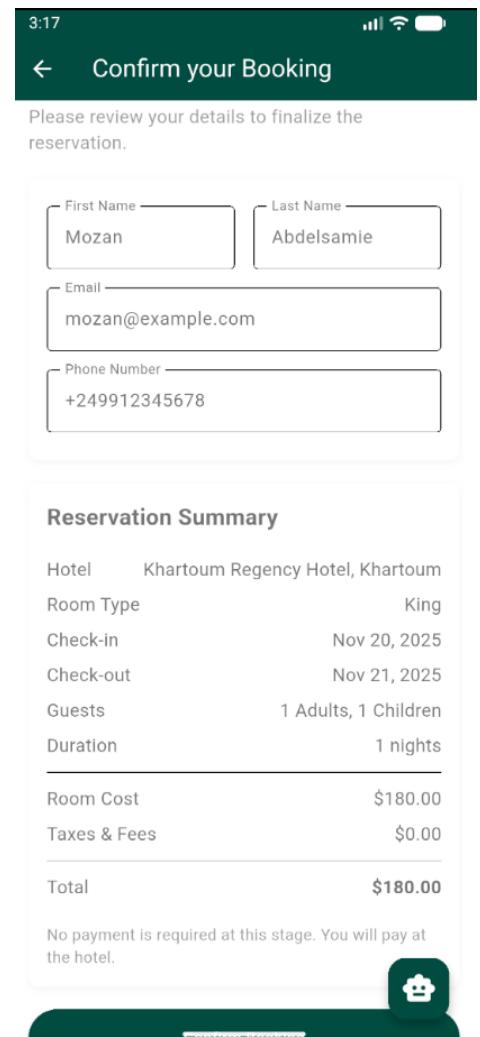
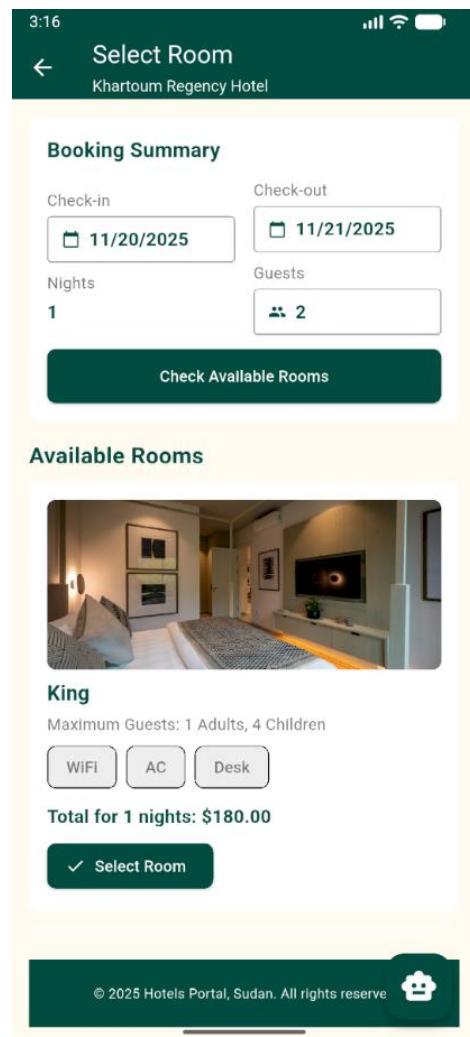
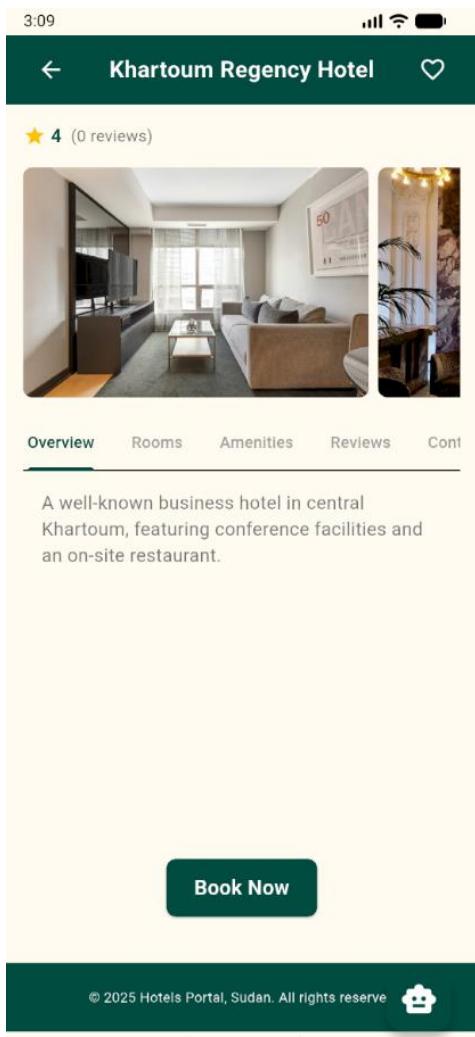
- 4.1 open home screen.
- 4.2 click search button,
- 4.3 go to search hotels screen.

### 5/ Search Hotels Screen:

- 5.1 choose your destination state.
- 5.2 choose your check-In date.
- 5.3 choose your check-Out date.
- 5.4 count how many adults guests.
- 5.5 count how many children guests.
- 5.6 count how many rooms.
- 5.7 click search button.
- 5.8 go to search results screen.

### 6/ Search Results Screen:

- 6.1 click filter menu if needed.
- 6.2 explore and compare available hotels based on your input information.
- 6.3 choose a hotel to stay in.
- 6.4 click view details from the chosen hotel.
- 6.5 go to hotel info screen.



#### 7/ Hotels Info Screen:

- 7.1 explore in hotel info screen sections (Overview, Rooms, Amenities, Reviews and Contact).
- 7.2 open Rooms section to explore all rooms, select one and go to confirm booking screen.
- 7.3 or click Book Now button from the screen.
- 7.4 go to select rooms.

#### 8/ Select Rooms Screen:

- 8.1 enter check-In and check-Out dates.
- 8.2 enter count of adults and children guests.
- 8.3 click check available rooms button.
- 8.4 explore available rooms.
- 8.5 click on select room from the chosen room.
- 8.6 go to confirm booking screen.

#### 9/ Confirm Booking Screen:

- 9.1 review your personal information.
- 9.2 edit your personal information if needed.
- 9.3 confirm your reservation summary.
- 9.4 read no payment note.
- 9.5 click on confirm booking button.
- 9.6 go to confirmed booking screen.

## Review Process:



**Booking Confirmed!**

Your reservation has been successfully confirmed. A confirmation email with all the details has been sent to mozan@example.com.

Booking Reference  
**KHA-2025-6004**

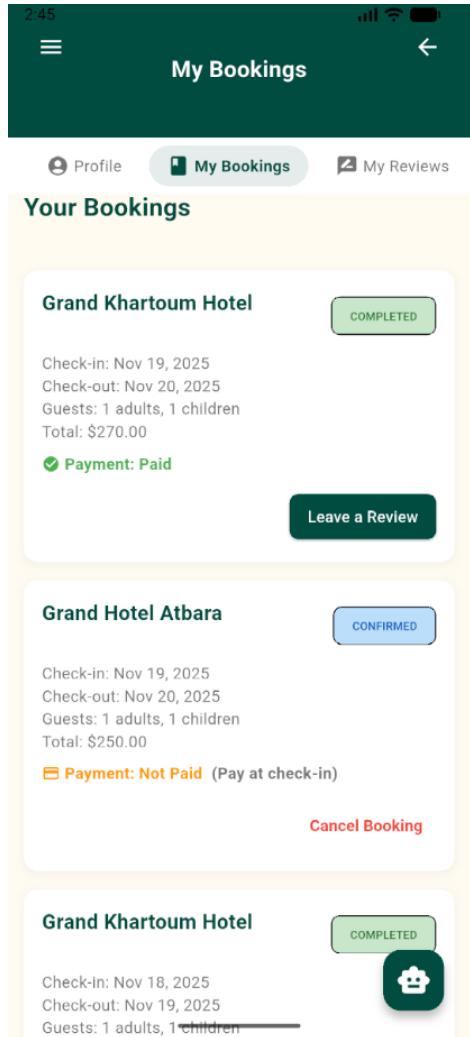
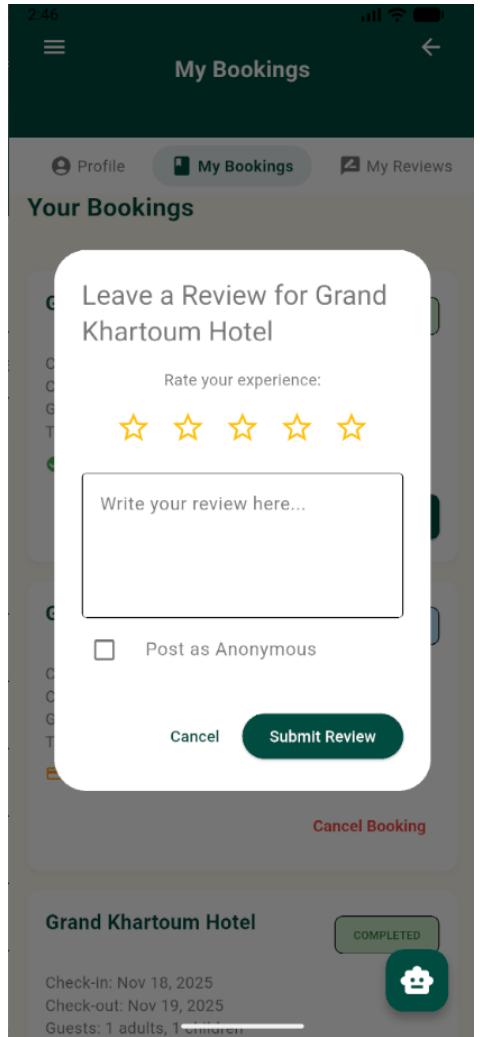
**Booking Details**

Hotel:	Khartoum Regency Hotel
Room:	King
Check-in:	Nov 20, 2025
Check-out:	Nov 21, 2025
Nights:	1
Guests:	1 adults, 1 children
Total Price:	\$330.00

**Back to Home**

© 2025 Hotels Portal, Sudan. All rights reserved.

- 10/ Booking Confirmed Screen:
- 10.1 take a screenshot or remember booking reference Id.
  - 10.2 receive a confirmation push notification if enabled.
  - 10.3 click back to home button.
  - 10.4 go to notifications section from my account screen.
  - 10.5 go to my bookings section in my account screen.

11/ My Bookings Scree:

- 11.1 open my account screen from sidebar.
- 11.2 explore sections (Profile, My Bookings, My Reviews, Notifications, Settings).
- 11.3 check your notifications in Notifications section.
- 11.4 check your old and current bookings in My Bookings section.
- 11.5 view your bookings status.
- 11.6 cancel your booking if wanted, by clicking cancel button and confirm cancellation by clicking Ok button.

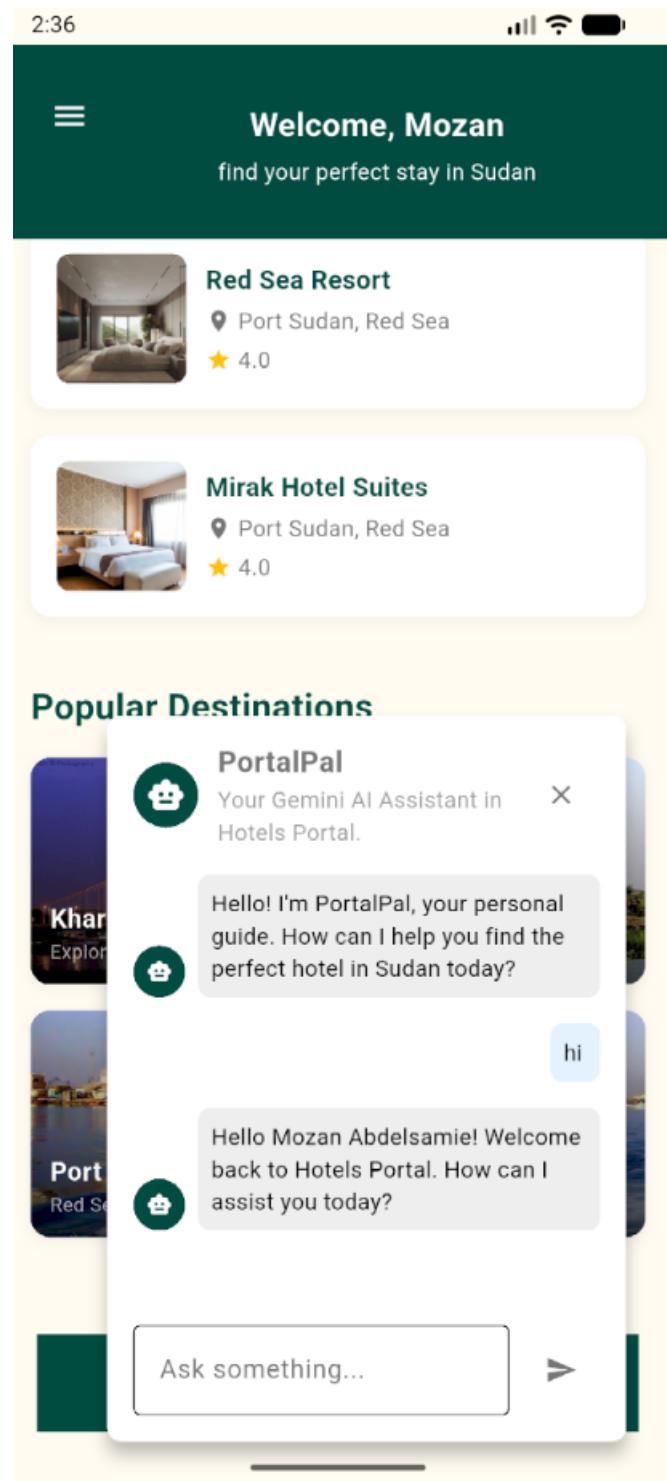
After completed bookings leave a review if wanted, by clicking leave a review button.

12/ Leave Review Screen:

- 12.1 open window by clicking leave a review button from my bookings section in my account screen.
- 12.2 choose how many stars to rate your staying.
- 12.3 type your review about your completed booking and the stayed hotel in the text field.
- 12.4 click submit review button to send your review or cancel button to cancel it.
- 12.5 check your notifications from sidebar or my account screen.

**AI Assistant:**

1. An AI assistant fixed in every screen in the app, a green slightly scalable button with robot icon in the right bottom of the app.
2. click on the green button in the right bottom and PortalPal your gemini AI Assistant in our app will be your guidance through all your activities.
3. Ask him about hotels in Sudan Only, and how to find your perfect hotels in your destination
4. he can't help in complicated processes or complaints, admins will contact you via email shortly.



## 8.2 PROGRAM CODES

**Entry point:**

- lib/main.dart

```
import 'package:firebase_messaging/firebase_messaging.dart';
import 'package:flutter/foundation.dart';
import 'package:flutter/material.dart';
import 'package:firebase_core/firebase_core.dart';
import 'package:hotel_booking_app/models/booking.dart';
import 'package:hotel_booking_app/models/hotel.dart';
import 'package:hotel_booking_app/models/room.dart';
import 'package:hotel_booking_app/providers/auth_provider.dart';
import 'package:hotel_booking_app/providers/analytics_provider.dart';
import 'package:hotel_booking_app/providers/booking_provider.dart';
import 'package:hotel_booking_app/providers/guest_provider.dart';
import 'package:hotel_booking_app/providers/hotel_provider.dart';
import 'package:hotel_booking_app/providers/search_provider.dart';
import 'package:hotel_booking_app/providers/theme_provider.dart';
import 'package:hotel_booking_app/providers/user_provider.dart';
import 'package:hotel_booking_app/screens/Guest/notifications_screen.dart';
import 'package:hotel_booking_app/utils/index.dart';
import 'package:hotel_booking_app/utils/globals.dart'; // Import globals
import 'package:provider/provider.dart';
import 'firebase_options.dart';
import 'package:flutter_local_notifications/flutter_local_notifications.dart';
import 'dart:io' show Platform;

// Guest App Screens
import 'screens/Guest/welcome_screen.dart';
import 'screens/Guest/login_screen.dart';
import 'screens/Guest/signup_screen.dart';
import 'screens/Guest/home_screen.dart';
import 'screens/Guest/my_bookings_screen.dart';
import 'screens/Guest/my_reviews_screen.dart';
import 'screens/Guest/my_account_screen.dart';
import 'screens/Guest/favorites_screen.dart';
import 'screens/Guest/settings_screen.dart';
import 'screens/Guest/search_hotels_screen.dart';
import 'screens/Guest/search_results_screen.dart';
```

```

import 'screens/Guest/hotel_info_screen.dart';
import 'screens/Guest/about_us_screen.dart';
import 'screens/Guest/help_center_screen.dart';
import 'screens/Guest/select_rooms_screen.dart';
import 'screens/Guest/confirm_booking_screen.dart';
import 'screens/Guest/booking_confirmed_screen.dart';

// Admin App Screens & Wrappers
import 'screens/admins_dashboard/admin_auth_wrapper.dart';
import 'screens/admins_dashboard/auth/login_screen.dart' as admin_login;
import 'screens/admins_dashboard/hotel_admin/dashboard_screen.dart'
    as hotel_dashboard;
import 'screens/admins_dashboard/ministry_dashboard_screen.dart'
    as ministry_dashboard;
import 'screens/admins_dashboard/hotel_admin/account_profile_screen.dart'
    as hotel_profile;
import 'screens/admins_dashboard/hotel_admin/hotel_admin_profile_screen.dart'
    as hotel_admin_profile;
import 'screens/admins_dashboard/hotel_admin/bookings_management_screen.dart'
    as hotel_bookings;
import 'screens/admins_dashboard/hotel_admin/guest_management_screen.dart'
    as hotel_guests;
import 'screens/admins_dashboard/hotel_admin/room_management_screen.dart'
    as hotel_rooms;
import 'screens/admins_dashboard/hotel_admin/reports_analytics_screen.dart'
    as hotel_reports;
import 'screens/admins_dashboard/hotel_admin/hotel_admin_settings_screen.dart'
    as hotel_settings;
import 'screens/admins_dashboard/hotel_admin/notifications_screen.dart'
    as hotel_notifications;
import 'screens/admins_dashboard/ministry_admin/hotel_registry_screen.dart'
    as ministry_registry;
import 'screens/admins_dashboard/ministry_admin/users_management_screen.dart'
    as ministry_users;
import 'screens/admins_dashboard/ministry_admin/reports_screen.dart'
    as ministry_reports;
import 'screens/admins_dashboard/ministry_admin/revenue_taxes_screen.dart'
    as ministry_revenue;
import 'screens/admins_dashboard/hotel_admin/reviews_screen.dart'
    as hotel_reviews;

```

```

import 'screens/admins_dashboard/hotel_admin/booking_analytics_screen.dart'
    as hotel_booking_analytics;
import 'screens/admins_dashboard/hotel_admin/booking_details_screen.dart'
    as hotel_booking_details;
import 'screens/admins_dashboard/hotel_admin/room_details_screen.dart'
    as hotel_room_details;
import 'screens/admins_dashboard/ministry_admin/ministry_admin_profile_screen.dart'
    as ministry_admin_profile;
import 'screens/admins_dashboard/ministry_admin/custom_reports_screen.dart'
    as ministry_custom_reports;
import 'screens/admins_dashboard/ministry_admin/notifications_screen.dart'
    as ministry_notifications;
import 'screens/admins_dashboard/ministry_admin/ministry_booking_details_screen.dart'
    as ministry_booking_details;
import 'package:hotel_booking_app/screens/admins_dashboard/ministry_admin/activities_screen.dart';
import 'widgets/ai_assistant.dart';

Future<void> _firebaseMessagingBackgroundHandler(RemoteMessage message) async {
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);
  print('Handling background message: ${message.messageId}');
}

void main() async {
  WidgetsFlutterBinding.ensureInitialized();
  await Firebase.initializeApp(options: DefaultFirebaseOptions.currentPlatform);

  // Android local notifications setup
  if (!kIsWeb && Platform.isAndroid) {
    const AndroidInitializationSettings androidSettings =
        AndroidInitializationSettings('@mipmap/ic_launcher');

    final InitializationSettings initializationSettings =
        InitializationSettings(android: androidSettings);

    await flutterLocalNotificationsPlugin.initialize(initializationSettings);
    await FirebaseMessaging.instance.requestPermission(
      alert: true,
      announcement: false,
      badge: true,
      carPlay: false,
    );
  }
}

```

```

criticalAlert: false,
provisional: false,
sound: true,
);
}

if (kIsWeb) {
  // Request permission for Web notifications
  await FirebaseMessaging.instance.requestPermission();
}

// Handle background FCM messages
FirebaseMessaging.onBackgroundMessage(_firebaseMessagingBackgroundHandler);

// Handle foreground FCM messages
FirebaseMessaging.onMessage.listen((RemoteMessage message) {
  print('Received message: ${message.notification?.title}');
  RemoteNotification? notification = message.notification;

  if (notification != null) {
    showNotification(
      notification.title ?? '',
      notification.body ?? '',
      flutterLocalNotificationsPlugin,
    );
  }
});

runApp(
  MultiProvider(
    providers: [
      ChangeNotifierProvider(create: (_) => AuthProvider()),
      ChangeNotifierProvider(create: (_) => ThemeProvider()),
      ChangeNotifierProvider(create: (_) => HotelProvider()),
      ChangeNotifierProvider(create: (_) => AnalyticsProvider()),
      ChangeNotifierProvider(create: (_) => BookingProvider()),
      ChangeNotifierProvider(create: (_) => UserProvider()),
      ChangeNotifierProvider(create: (_) => GuestProvider()),
      ChangeNotifierProvider(create: (_) => SearchProvider()),
    ],
  ),
);

```

```

        child: const App(),
    ),
);
}
}

class App extends StatelessWidget {
const App({super.key});

@Override
Widget build(BuildContext context) {
if (kIsWeb) {
    return const AdminApp();
} else if (!kIsWeb && Platform.isAndroid) {
    return const GuestApp();
} else {
    return const UnsupportedScreen();
}
}
}

// ----- Guest (Mobile) Application -----
class GuestApp extends StatelessWidget {
const GuestApp({super.key});

@Override
Widget build(BuildContext context) {
return Consumer<ThemeProvider>(
    builder: (context, themeProvider, child) {
        return MaterialApp(
            title: 'Hotels Portal',
            theme: themeProvider.currentTheme,
            home: const AuthWrapper(),
            debugShowCheckedModeBanner: false,
            builder: (context, child) {
                return Stack(children: [child!, const AiAssistant()]);
            },
            routes: {
                '/login': (context) => const LoginScreen(),
                '/signup': (context) => const SignUpScreen(),
                '/home': (context) => const HomeScreen(),
            }
        );
    }
);
}
}

```

```

'/my_bookings': (context) => const MyBookingsScreen(),
'/my_reviews': (context) => const MyReviewsScreen(),
'/my_account': (context) => const MyAccountScreen(),
'/favorites': (context) => const FavoritesScreen(),
'/settings': (context) => const SettingsScreen(),
'/notifications': (context) => const NotificationsScreen(),
'/search': (context) => const SearchHotelsScreen(),
'/hotel_info': (context) => HotelInfoScreen(
  hotel: ModalRoute.of(context)!.settings.arguments as Hotel,
),
'/search_results': (context) => SearchResultsScreen(),
'/about_us': (context) => const AboutUsPage(),
'/help_center': (context) => const HelpCenterScreen(),
'/select_rooms': _buildSelectRoomsScreen,
'/confirm_booking': (context) => const ConfirmBookingScreen(),
'/booking_confirmed': (context) => const BookingConfirmedScreen(),
},
);
},
);
}
}

// ----- Admin (Web/Windows) Application -----
class AdminApp extends StatelessWidget {
const AdminApp({super.key});


```

```

@override
Widget build(BuildContext context) {
return Consumer<ThemeProvider>(
  builder: (context, themeProvider, child) {
    return MaterialApp(
      title: 'Hotels Portal - Admin',
      theme: themeProvider.currentTheme,
      home: const AdminAuthWrapper(),
      debugShowCheckedModeBanner: false,
      routes: {
        '/login': (context) => const admin_login.LoginScreen(),
        '/hotel-dashboard': (context) =>
          const hotel_dashboard.HotelDashboardScreen(),
      }
    );
  }
);
}


```

```

'/ministry-dashboard': (context) =>
  const ministry_dashboard.MinistryDashboardScreen(),
'/hotel-account-profile': (context) =>
  const hotel_profile.AccountProfileScreen(),
'/hotel-admin-profile': (context) =>
  const hotel_admin_profile.HotelAdminProfileScreen(),
'/hotel-guest-management': (context) =>
  const hotel_guests.GuestManagementScreen(),
'/hotel-room-management': (context) =>
  const hotel_rooms.RoomManagementScreen(),
'/hotel-booking-analytics': (context) =>
  const hotel_booking_analytics.BookingAnalyticsScreen(),
'/hotel-bookings-management': (context) =>
  const hotel_bookings.BookingsManagementScreen(),
'/hotel-booking-details': (context) {
  final booking =
    ModalRoute.of(context)!.settings.arguments as Booking;
  return hotel_booking_details.BookingDetailsScreen(
    booking: booking,
  );
},
'/ministry-booking-details': (context) {
  final booking =
    ModalRoute.of(context)!.settings.arguments as Booking;
  return ministry_booking_details.MinistryBookingDetailsScreen(
    booking: booking,
  );
},
'/hotel-room-details': (context) {
  final room = ModalRoute.of(context)!.settings.arguments as Room;
  return hotel_room_details.RoomDetailsScreen(room: room);
},
'/hotel-reports-analytics': (context) =>
  const hotel_reports.ReportsAnalyticsScreen(),
'/hotel-admin-settings': (context) =>
  const hotel_settings.HotelAdminSettingsScreen(),
'/hotel-notifications': (context) =>
  const hotel_notifications.NotificationsScreen(),
'/hotel-registry': (context) =>
  const ministry_registry.HotelRegistryScreen(),

```

```

'/ministry-users': (context) =>
  const ministry_users.UsersManagementScreen(),
'/ministry-reports': (context) =>
  const ministry_reports.ReportsScreen(),
'/ministry-revenue': (context) =>
  const ministry_revenue.RevenueTaxesScreen(),
'/ministry-licensing': (context) =>
  const ministry_dashboard.MinistryDashboardScreen(),
'/ministry-inspections': (context) =>
  const ministry_dashboard.MinistryDashboardScreen(),
'/ministry-tourism-stats': (context) =>
  const ministry_dashboard.MinistryDashboardScreen(),
'/ministry-compliance': (context) =>
  const ministry_dashboard.MinistryDashboardScreen(),
'/hotel-reviews': (context) => const hotel_reviews.ReviewsScreen(),
'/ministry-admin-profile': (context) =>
  const ministry_admin_profile.MinistryAdminProfileScreen(),
'/ministry-custom-reports': (context) =>
  const ministry_custom_reports.CustomReportsScreen(),
'/ministry-notifications': (context) =>
  const ministry_notifications.MinistryNotificationsScreen(),
'/ministry-activities': (context) => const ActivitiesScreen(),
},
);
},
);
}
}

// ----- Auth Wrapper -----

```

```

class AuthWrapper extends StatelessWidget {
  const AuthWrapper({super.key});

  @override
  Widget build(BuildContext context) {
    final authProvider = Provider.of<AuthProvider>(context);
    if (authProvider.isLoggedIn) {
      return const HomeScreen();
    } else {
      return const WelcomeScreen();
    }
  }
}
```

```

        }
    }
}

// ----- Select Rooms Helper -----
Widget _buildSelectRoomsScreen(BuildContext context) {
final args =
    ModalRoute.of(context)!.settings.arguments as Map<String, dynamic>? ?? {};
final hotelJson = args['hotel'] as Map<String, dynamic>? ?? {};
final hotel = Hotel.fromJson(hotelJson);
final checkInDate = args['checkInDate'] as DateTime? ?? DateTime.now();
final checkOutDate =
    args['checkOutDate'] as DateTime? ??
    DateTime.now().add(const Duration(days: 1));
final numberOfGuests = args['numberOfGuests'] as int? ?? 1;
return SelectRoomsScreen(
    hotel: hotel,
    initialCheckInDate: checkInDate,
    initialCheckOutDate: checkOutDate,
    initialNumberOfGuests: numberOfGuests,
);
}

// ----- Unsupported Platform Screen -----
class UnsupportedScreen extends StatelessWidget {
const UnsupportedScreen({super.key});

@Override
Widget build(BuildContext context) {
    return Scaffold(
        appBar: AppBar(title: const Text('Unsupported Platform')),
        body: const Center(
            child: Text(
                'Sorry, this platform is not supported.',
                style: TextStyle(fontSize: 18),
                textAlign: TextAlign.center,
            ),
        ),
    );
}
}

```

```
}
```

## Guest Screens:

### - lib/screens/Guest/welcome\_screen.dart

```
```
```

```
Navigator.pushNamed(context, '/login');  
Navigator.pushNamed(context, '/signup');  
```
```

### - lib/screens/Guest/login\_screen.dart

```
```
```

```
final authProvider = Provider.of<AuthProvider>(context, listen: false);  
final error = await authProvider.signIn(email, password, isAdminLogin: false);  
if (error == null) Navigator.pushReplacementNamed(context, '/home');  
```
```

### - lib/screens/Guest/signup\_screen.dart

```
```
```

```
final authProvider = Provider.of<AuthProvider>(context, listen: false);  
final error = await authProvider.signUp(  
    email: email,  
    password: password,  
    fName: firstName,  
    lName: lastName,  
    phone: phone,  
    birthDate: birthdateController.text.isNotEmpty ? DateTime.parse(birthdateController.text) : null,  
);  
if (error == null) Navigator.pushNamed(context, '/home');  
```
```

### - lib/screens/Guest/home\_screen.dart

```
```
```

```
Provider.of<HotelProvider>(context, listen: false).fetchFeaturedHotels();  
final guest = Provider.of<AuthProvider>(context, listen: false).currentGuest;  
if (guest != null) {  
    Provider.of<BookingProvider>(context, listen: false).fetchBookingsForUser(guest.guestId);  
}
```

```
Provider.of<SearchProvider>(context, listen: false).clear();
```

```
Navigator.pushNamed(context, '/search');
Navigator.pushNamed(context, '/hotel_info', arguments: hotel);
```

```

#### - lib/screens/Guest/search\_hotels\_screen.dart

```
```
final searchProvider = Provider.of<SearchProvider>(context, listen: false);
// read provider values into local fields
Provider.of<SearchProvider>(context, listen: false).updateCriteria(
  state: _selectedState,
  checkIn: _checkInDate,
  checkOut: _checkOutDate,
  adults: _adults,
  children: _children,
  rooms: _rooms,
);
Navigator.pushNamed(context, '/search_results');
```

```

#### - lib/screens/Guest/search\_results\_screen.dart

```
```
final searchService = SearchService();
final searchProvider = Provider.of<SearchProvider>(context, listen: false);
final filters = SearchFilters(
  location: searchProvider.selectedState,
  city: _selectedCity,
  checkInDate: searchProvider.checkInDate,
  checkOutDate: searchProvider.checkOutDate,
  minPrice: _minPrice,
  maxPrice: _maxPrice,
  minStarRating: _minRating,
  maxStarRating: _maxRating,
  amenities: _selectedAmenities.isNotEmpty ? _selectedAmenities : null,
);
final hotels = await searchService.searchHotels(filters: filters);
setState(() => _hotels = hotels);
Navigator.pushNamed(context, '/hotel_info', arguments: hotel);
```

```

#### - lib/screens/Guest/hotel\_info\_screen.dart

```

```
Provider.of<HotelProvider>(context, listen: false).loadRooms(widget.hotel.hotelId);
Provider.of<HotelProvider>(context, listen: false).loadReviews(widget.hotel.hotelId);

final isCurrentlyFavorite = authProvider.currentGuest?.favoriteHotelIds.contains(widget.hotel.hotelId) ??
false;
authProvider.toggleFavorite(widget.hotel.hotelId);

Navigator.pushNamed(
  context,
  '/select_rooms',
  arguments: {
    'hotel': widget.hotel.toJson(),
    'checkInDate': DateTime.now(),
    'checkOutDate': DateTime.now().add(Duration(days: 1)),
    'numberOfNights': 1,
    'numberOfGuests': 1,
  },
);
```

```

#### **- lib/screens/Guest/select\_rooms\_screen.dart**

```

```
final hotelProvider = Provider.of<HotelProvider>(context, listen: false);
await hotelProvider.loadRooms(widget.hotel.hotelId);
final allHotelRooms = hotelProvider.rooms;

final availableRoomCounts = await BookingService().getAvailableRoomCounts(
  widget.hotel.hotelId,
  checkInDate,
  checkOutDate,
);

for (final room in allHotelRooms) {
  final availableCount = availableRoomCounts[room.roomType] ?? 0;
  final hasCapacity = room.maxAdults >= totalAdults && room.maxChildren >= totalChildren;
  if (availableCount > 0 && hasCapacity) { /* add room */ }
}

Navigator.pushNamed(

```

```
context,
'/confirm_booking',
arguments: {
  'hotel': widget.hotel.toJson(),
  'selectedRoom': room.toMap(),
  'checkInDate': checkInDate,
  'checkOutDate': checkOutDate,
  'adults': totalAdults,
  'children': totalChildren,
},
);
```

```

#### - lib/screens/Guest/confirm\_booking\_screen.dart

```
```

```

```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
if (authProvider.currentGuest == null) return; // require login

```

```
if (guestUpdates.isNotEmpty) {
  await GuestService().updateGuest(guest.guestId, guestUpdates);
  await authProvider.refreshGuestData();
}

```

```
final bookingId = FirebaseFirestore.instance.collection('bookings').doc().id;
final newBooking = Booking(...);
final createdBookingId = await hotelProvider.createBooking(newBooking, guestId, guestName);
if (createdBookingId != null) {
  showNotification('Booking Confirmed', 'Your booking has been confirmed successfully.',
  flutterLocalNotificationsPlugin);
  Navigator.pushReplacementNamed(context, '/booking_confirmed', arguments: {...});
}
```

```

#### - lib/screens/Guest/booking\_confirmed\_screen.dart

```
```

```

```
final args = ModalRoute.of(context)!.settings.arguments as Map<String, dynamic>? ?? {};
final bookingRef = args['bookingRef'] as String? ?? 'N/A';
Navigator.of(context).pushNamedAndRemoveUntil('/home', (route) => false);
```

```

### **- lib/screens/Guest/my\_bookings\_screen.dart**

```
```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
final guestId = authProvider.currentGuest?.guestId;
final bookings = await GuestService().getGuestBookingHistory(guestId);
final reviews = await GuestService().getGuestReviews(guestId);

await BookingService().cancelBooking(booking.bookingId, guestId, guestName);
await _fetchData();

final newReview = Review(...);
await GuestService().addReview(newReview);
```
```

### **- lib/screens/Guest/my\_reviews\_screen.dart**

```
```
final guestId = Provider.of<AuthProvider>(context, listen: false).currentGuest?.guestId;
final reviews = await GuestService().getGuestReviews(guestId);
setState(() => _reviews = reviews);
```
```

### **- lib/screens/Guest/notifications\_screen.dart**

```
```
final guestId = Provider.of<AuthProvider>(context, listen: false).currentGuest?.guestId;
_notificationsStream = GuestService().getGuestNotificationsStream(guestId);
await GuestService().markNotificationAsRead(guestId, notification.notificationId);
// StreamBuilder(stream: _notificationsStream, ...)
```
```

### **- lib/screens/Guest/settings\_screen.dart**

```
```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
final settings = authProvider.currentGuest?.settings;
await GuestService().updateGuestSettings(guest.guestId, newSettings.toMap());
await authProvider.refreshGuestData();
final error = await authProvider.updateUserPassword(currentPassword: currentPw, newPassword: newPw);
final error = await authProvider.deleteAccount(password);
if (error == null) Navigator.pushAndRemoveUntil(context, MaterialPageRoute(builder: (_) => WelcomeScreen()), (r)=>false);
```
```

**- lib/screens/Guest/my\_account\_screen.dart**

```
```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
final currentUser = authProvider.currentGuest;
(firstNameController.text = currentUser?.fName ?? "");
await GuestService().updateGuest(authProvider.currentGuest!.guestId, updates);
await authProvider.refreshGuestData();
```

```

**- lib/screens/Guest/favorites\_screen.dart**

```
```
final favoriteIds = Provider.of<AuthProvider>(context, listen: false).currentGuest?.favoriteHotelIds ?? [];
if (favoriteIds.isNotEmpty) Provider.of<HotelProvider>(context, listen: false).fetchFavoriteHotels(favoriteIds);
authProvider.toggleFavorite(hotel.hotelId);
```

```

**Admin dashboard screens:**

**Ministry admins:**

**- lib/screens/admins\_dashboard/admin\_auth\_wrapper.dart**

```
```
final authProvider = Provider.of<AuthProvider>(context);
if (authProvider.isLoggedIn && authProvider.role == 'admin') {
  if (authProvider.isMinistryAdmin) return MinistryDashboardScreen();
  else return HotelDashboardScreen();
} else return WelcomeScreen();
```

```

**- lib/screens/admins\_dashboard/welcome\_screen.dart**

```
```
Navigator.pushNamed(context, '/login');
```

```

**- lib/screens/admins\_dashboard/ministry\_dashboard\_screen.dart**

```
```
Provider.of<AnalyticsProvider>(context, listen: false).fetchDashboardStats();
StreamBuilder<List<Activity>>(stream: ActivityService().getRecentActivitiesStream(), builder: ...);
```

```

```

- lib/screens/admins_dashboard/ministry_admin/reports_screen.dart
```


_pastReportsStream = AnalyticsService().getPastMinistryReportsStream();  

Navigator.pushNamed(context, '/ministry-custom-reports');



// Download PDF core flow: decode data, build pdf Document, final bytes = await pdf.save();  

// create web blob & anchor to trigger download



// Download CSV core flow: convert rows via ListToCsvConverter().convert(rows); create web blob &  

anchor to download



StreamBuilder<List<GeneratedReport>>(stream: _pastReportsStream, builder: (context, snapshot) {  

final reports = snapshot.data ?? [];  

// View: Navigator.push(context, MaterialPageRoute(builder: (_)> CustomReportsScreen(pastReport:  

report)));  

// Download popup: if 'CSV' -> _downloadCsv(report); if 'PDF' -> _downloadPdf(report)  

});  

```


```

```

- lib/screens/admins_dashboard/ministry_admin/notifications_screen.dart
```


final adminId = Provider.of<AuthProvider>(context, listen: false).currentAdmin?.adminId;  

_notificationsStream = NotificationService().getAdminNotificationsStream(adminId);  

await NotificationService().markAdminNotificationAsRead(adminId, notification.notificationId);  

// StreamBuilder to show notifications  

```



```

- lib/screens/admins_dashboard/ministry_admin/users_management_screen.dart
```


Provider.of<UserProvider>(context, listen: false).fetchAllUsers();  

// filtering & search: filter userProvider.users by role and search query  

userProvider.toggleUserStatus(user, currentUserId!, currentUserName);  

```



```

- lib/screens/admins_dashboard/ministry_admin/ministry_booking_details_screen.dart
```


// fetch guest and room details for a booking  

final guestData = await GuestService().getGuest(widget.booking.guestId.id);  

final roomDoc = await widget.booking.roomId.get();


```


```


```

```

if (roomDoc.exists) _room = Room.fromMap(roomDoc.data() as Map<String, dynamic>);
```

- lib/screens/admins_dashboard/ministry_admin/ministry_admin_profile_screen.dart
```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
_fNameController.text = authProvider.currentAdmin?.fName ?? "";
await _adminService.updateAdminSettings(admin.adminId, newSettings.toMap());
final profileError = await authProvider.updateAdminProfile(fName: _fNameController.text, lName: _lNameController.text);
final passwordError = await authProvider.updateUserPassword(currentPassword: _currentPasswordController.text, newPassword: _newPasswordController.text);
await authProvider.refreshGuestData();
```

- lib/screens/admins_dashboard/ministry_admin/hotel_registry_screen.dart
```
// stream registered hotels
stream: AdminService().getAllHotels();

// open hotel info details
Navigator.push(context, MaterialPageRoute(builder: (_) => HotelInfoAmenitiesScreen(hotel: hotel)));

// register hotel + admin (uses secondary firebase app and HotelService)
final userCredential = await tempAuth.createUserWithEmailAndPassword(email: adminEmail, password: adminPassword);
final hotelId = FirebaseFirestore.instance.collection('hotels').doc().id;
await HotelService().createHotel(newHotel, currentAdminId, currentAdminName);
```

- lib/screens/admins_dashboard/ministry_admin/hotel_info_amenities_screen.dart
```
await _hotelService.updateHotel(widget.hotel.hotelId, {'approved': newStatus});
await _activityService.createActivity(Activity(...));
```

- lib/screens/admins_dashboard/ministry_admin/custom_reports_screen.dart
```
final hotelsResult = await HotelService().getAllHotels(limit: 1000);

```

```

final data = await AnalyticsService().getRevenueReportData(hotelId: _selectedHotelId ?? "", startDate: _startDate!, endDate: _endDate!);
final report = GeneratedReport(id: "", reportType: _selectedReportType, createdAt: DateTime.now(),
startDate: _startDate ?? DateTime.now(), endDate: _endDate ?? DateTime.now(), generatedBy: adminId,
reportDataJson: jsonEncode(data));
await AnalyticsService().saveMinistryReport(report);
// trigger CSV/PDF download flows using web Blob or pdf package
```

```

#### **- lib/screens/admins\_dashboard/ministry\_admin/activities\_screen.dart**

```
```

```

```

final activities = await ActivityService().getAllActivities(filterByType: _selectedActivityType == 'All
Types' ? null : _selectedActivityType, startDate: _startDate, endDate: _endDate);
setState(() => _activities = activities);
```

```

#### **Hotel admins:**

#### **- lib/screens/admins\_dashboard/hotel\_admin/dashboard\_screen.dart**

```
```

```

```

final authProvider = Provider.of<AuthProvider>(context, listen: false);
if (authProvider.hotelId != null) {
  Provider.of<AnalyticsProvider>(context, listen: false).fetchHotelDashboardStats(authProvider.hotelId!);
  Provider.of<AnalyticsProvider>(context, listen: false).fetchRoomStatusSummary(authProvider.hotelId!);
}
// Consumer<AnalyticsProvider> to read stats
```

```

#### **- lib/screens/admins\_dashboard/hotel\_admin/room\_management\_screen.dart**

```
```

```

```

final authProvider = Provider.of<AuthProvider>(context, listen: false);
if (authProvider.hotelId != null) {
  Provider.of<HotelProvider>(context, listen: false).loadRooms(authProvider.hotelId!);
}
_showAddRoomDialog(authProvider.hotelId!);
_showEditRoomDialog(room);
Navigator.pushNamed(context, '/hotel-room-details', arguments: room);

// _AddRoomDialog save core flow:
await HotelService().deleteHotelImage(url);

```

```
final url = await HotelService().uploadRoomImage(widget.hotelId, roomId, bytes, file.name);
final roomData = Room(...);
if (isEditing) await provider.updateRoom(roomData); else await provider.addRoom(roomData);
```

```

#### - lib/screens/admins\_dashboard/hotel\_admin/booking\_details\_screen.dart

```
```
final guestData = await GuestService().getGuest(widget.booking.guestId.id);
final roomDoc = await widget.booking.roomId.get();
if (roomDoc.exists) _room = Room.fromMap(roomDoc.data() as Map<String, dynamic>);
```

```

#### - lib/screens/admins\_dashboard/hotel\_admin/bookings\_management\_screen.dart

```
```
// table with filters and search
HotelBookingsTable(showActions: true, searchQuery: _searchQuery, statusFilter: _selectedStatus);
```

```

#### - lib/screens/admins\_dashboard/hotel\_admin/booking\_analytics\_screen.dart

```
```
// UI-driven analytics: select filter and render FlChart/BarChart datasets
selectedFilter = 'This Month'; // use to switch series
```

```

#### - lib/screens/admins\_dashboard/hotel\_admin/notifications\_screen.dart

```
```
final adminId = Provider.of<AuthProvider>(context, listen: false).currentAdmin?.adminId;
_notificationsStream = NotificationService().getAdminNotificationsStream(adminId);
await NotificationService().markAdminNotificationAsRead(adminId, notification.notificationId);
```

```

#### - lib/screens/admins\_dashboard/hotel\_admin/hotel\_admin\_profile\_screen.dart

```
```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
_fNameController.text = authProvider.currentAdmin?.fName ?? "";
await _adminService.updateAdminSettings(admin.adminId, newSettings.toMap());
final profileError = await authProvider.updateAdminProfile(...);
final passwordError = await authProvider.updateUserPassword(...);
await authProvider.refreshGuestData();
```

```

**- lib/screens/admins\_dashboard/hotel\_admin/guest\_management\_screen.dart**

```
```
final authProvider = Provider.of<AuthProvider>(context, listen: false);
Provider.of<UserProvider>(context, listen: false).fetchGuestsForHotel(authProvider.hotelId!);
final filteredGuests = userProvider.hotelGuests.where(...).toList();
```
```

**- lib/screens/admins\_dashboard/hotel\_admin/reviews\_screen.dart**

```
```
final hotelId = Provider.of<AuthProvider>(context, listen: false).hotelId;
final reviewsData = await GuestService().getHotelReviews(hotelId);
setState(() => _reviews = reviewsData);
```
```

**- lib/screens/admins\_dashboard/hotel\_admin/room\_details\_screen.dart**

```
```
// Display room details already loaded into widget.room
// image gallery: widget.room.images.map((url) => Image.network(url))
```
```

**- lib/screens/admins\_dashboard/hotel\_admin/reports\_analytics\_screen.dart**

```
```
_pastReportsStream = AnalyticsService().getPastReportsStream(hotelId);
final data = await AnalyticsService().getBookingReportData(hotelId: hotelId, startDate: _startDate!, endDate: _endDate!);
final newReport = GeneratedReport(...);
await AnalyticsService().saveGeneratedReport(hotelId: hotelId, report: newReport);
// Download CSV/PDF with ListToCsvConverter or pdf package and web Blob anchor
```
```

## 8.3 SYSTEM TESTING EVALUATION TOOL

This section presents the evaluation instruments used to assess the functionality, usability, and reliability of the Hotels Portal Reservation and Management System. The testing tools were designed to align with the Agile methodology and cover all user roles—guests, hotel administrators, and ministry officials. The evaluation focused on verifying system behavior under controlled and real-world conditions.

### Sample Evaluation Criteria

1. Does the system meet the set of objectives?

2. Evaluate the system in terms of speed, accuracy of data result and system function?
3. Does the system provide an intuitive and user-friendly interface for both regular users and administrators?
4. Evaluate the system in terms of performance (loading speed, response time), accuracy of data results, and stability.
5. Does the system provide accurate reports and outputs?
6. Evaluate the system in terms of creating a reservation and providing sufficient information?

## **8.4 PROPOSAL TEAM CURRICULUM VITAE**

### **proponent #1**

#### **Personal information**

Name: Mozan Abdelsamie Mohamed Ahmed.

Date and place of birth: 23rd of October 2000 - Khartoum, Sudan.

E-mail : [mozanfu@gmail.com](mailto:mozanfu@gmail.com)

Mobile: +249901979867

Nationality: Sudanese.

Address: Doha, Qatar.

#### **Education**

#### **University:**

- 2017-2020: University of Bahri, Information Technology.
- 2021-present (transferred): Future University, Information Technology.

#### **Courses:**

- Microsoft office
- C++

- HTML
- CSS
- PHP
- SQL
- Javascript
- Python fundamental
- Database essentials
- Flutter, dart - coursera
- Software engineering - coursera
- Networking essentials - Cisco netacad.com

## Skills

- English-Math-computer knowledge.
- Programming Languages, Database, and Networking essential knowledge.
- Communication skills.
- Organization and time management skills.
- Work under pressure.

## Languages

- Arabic (mother tongue)
- English (very good)

## Proponent #2

### Personal information

Name: Mohammed Jamal Abdalla Ali.

Date and place of birth: 11th of October 1997 - Omdurman, Sudan.

E-mail : [moeeltoam@gmail.com](mailto:moeeltoam@gmail.com)

Mobile: +249966397914

Nationality: Sudanese.

Address: Giza, Egypt.

### Education

#### University:

- 2015-2020: University of Khartoum, Faculty Of Mathematical Sciences.
- 2021-present (transferred): Future University, Information Technology.

#### Courses:

- Microsoft office
- C++
- HTML
- CSS
- PHP
- SQL
- Python fundamental
- Database essentials
- Flutter, dart - coursera
- Software engineering - coursera
- Google's Data Analysis Professional Certificate - coursera

## Skills

- English, Math, and computer knowledge.
- Programming Languages, Database, and Networking essential knowledge.
- Communication skills.
- Organization and time management skills.
- Work great under pressure and in a team.

## Languages

- Arabic (mother tongue)
- English (excellent)