Hotel Reservation System

Hoteleum

Name of the Institution: Presidential University
- **Name of Department: Software Engineering**
- **Name of Authors: Rustam Kadirov - 210151,**
  Zarshedjon Nasimov - 210058,
  Adil Raziev - 210103,
  Ganisher Boboyorov - 210100.
- **Date of submission: 01.02.2024**

Hoteleum

Table of Contents

1. Introduction

Our Hotel Management System is a mobile application designed to streamline the hotel reservation process. Users can log in, view hotel listings, select rooms, make payments, and manage their reservations.

It’s not a secret that from ancient times our ancestors traveled all over the world to expand their businesses, to make money or just to see the world from different perspectives. However, even nowadays these trends don't lose their popularity, rather they become more popular than ever before.

From our point of view, the aforementioned occasion led business owners and creative minds to brainstorm on the topic “How to increase profit and pursue people to travel?”. Then they come up with the idea of creating affordable housing for some period. And called this type of houses or apartments as “Hotel” or “Motel”. But in the 21st century in the era of digitalization of everything, it's quite difficult to catch up new trends especially in business, but with our application it will be too easy. At some point we thought: “For how much longer will we take a room when we arrive at a hotel or book a room without seeing what is inside?”. That idea led us to develop a web application based on Flutter called “Hoteleum ".

Hoteleum is an application where you can **see** what facilities will be in the room users are booking and book beforehand so it will be convenient for both businessmen who plan his business trips 5-6 months ahead and for those who just take their luggage in the early morning and decide to hike the country. If you don’t like a room on arrival, users can request to change their rooms to another one.

Our main aim lies in providing convenient and fast hotel reservation systems all around the world. So I can list them in order to better explain:

- We developed a database that efficiently manages and organises hotel reservations.
- Ensures quick and accurate data retrieval for staff members.
- User friendly interface

Efficient Reservation Management: One of the key features of the system is real time tracking which is convenient simultaneously for staff members and clients to make reservations easily.

Real-time Availability Tracking: Implement real time updates of rooms available thus we can prevent overbooking and ensure accurate reservation details.

Secure Data Storage: Prioritise the security of sensitive customer information by implementing robust data encryption and access controls.

Scalability: Design the database system to be scalable, accommodating the growth of the hotel and an increasing number of reservations.

Above is the list of our problem statements which clearly articulate the problem that we are trying to solve.

Firstly, as mentioned above, in the era of digitalization we are approaching automation of manual processes, so our project is not an exception.

- **Automation of Processes**: Automate routine tasks, such as confirmation emails, payment processing, and room assignment, to streamline operations and reduce manual effort.

  We thought that people who use our application deserve better and then planned to add feedback and review systems.

- **Feedback and Review System**: Is about implementation of a system for collecting customer feedback and reviews to continuously improve the reservation process and overall guest experience. So that customers are more aware where they should go and where they shouldn't.

  Multi-Language and Currency Support: Service to an international clientele by supporting multiple languages and currencies in the reservation system. Multi language and currency support is one of the ways to show respect to other nations and it will increase the reputation of our application.

2. *Requirements Catalogue*

2.1 Functional Requirements

1. **Authentication and User Registration:** A secure authentication login system should enable users to register.

2. **Lookup and Booking**: Users should be able to filter results for available rooms based on a variety of criteria, such as the number of guests, kind of lodging, and date. The system must allow users to make reservations for rooms.

3. **Control of the Room**: Room information should be addable, editable, and removable by administrators. The system ought to keep track of the availability of rooms.

4. **Confirmation of Reservation**: Reservation confirmations should be sent to users via email or through the system.

5. **Retraction and Adjustment:** Reservations should be able to be changed or cancelled by users within certain bounds.

6. **Processing of Payments**: Processing payments securely for reservation costs.

7. **User Profiles**: Users ought to have accounts where they may monitor past reservations and control personal data.

2.2 Non-Functional Requirements

1. Performance: System should not face any desirable performance decrease or crash when a huge number of concurrent users are served by the system.

2. Reliability: The system must be always accessible, with a little downtime required for maintenance.

3. Scalability: As the number of users and transactions increases, the system must be scalable.

4. Security: Data security requires secure user authorization management and authentication, as well as the encryption of sensitive data, particularly during payment transactions.

5. Usability: An intuitive and user-friendly user interface is essential.

6. Availability: Because of the system's high availability, users may rely on it at all times.

7. Data Integrity: Guaranteeing the accuracy and consistency of data across the system.

8. **Reaction Time:** The system has to react fast when a user makes or cancels a reservation.

9. **Audit Trail:** Features that keep track of and evaluate user behaviour to ensure accountability and security**.**

10. **Backup and Recovery:** Frequent data backups and a robust procedure in place for retrieving data in the case of a system breakdown.

11. **Documentation:** Comprehensive guidelines for users, administrators, and developers.

3. UML Activity Diagram

The UML Activity Diagram provides a visual representation of the workflow within the Hotel Management System. It illustrates the sequence of activities and the flow of control between these activities, offering a high-level overview of the application's functionality.

Key Components:

User Authentication:

- The diagram initiates with the User Authentication activity, depicting the process of user login or registration. Arrows indicate the decision-making points based on successful or unsuccessful authentication.

Dashboard Navigation:

- Upon successful authentication, the diagram illustrates the transition to the Dashboard. The Dashboard serves as a central hub, providing links to reservation, profile, and help functionalities.

Hotel Listing:

- From the Dashboard, the flow extends to the Hotel Listing activity. The diagram visually represents the retrieval and display of hotel information, including photos, names, and locations.

Hotel Details:

- When a user selects a specific hotel, the diagram captures the transition to the Hotel Details activity. Here, users can view room types and corresponding prices.

Room Selection:

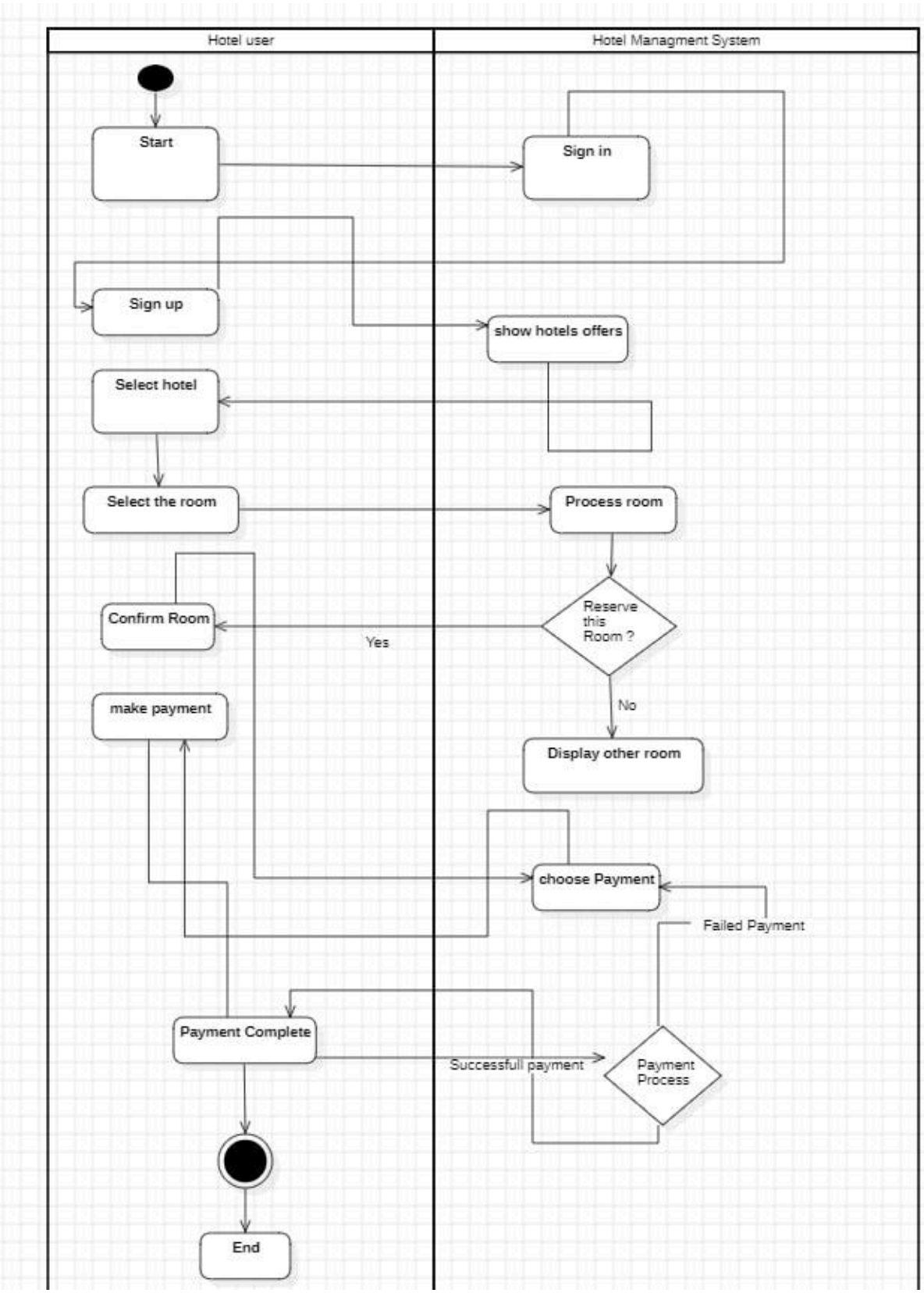
- The Room Selection activity allows users to choose a preferred room type and view the associated price. Arrows depict the decision-making process of selecting a room.

Payment Process:

- The Payment Process is a crucial step, represented by a distinct activity in the diagram. It includes a secure payment window where users enter the payment amount. The diagram visually indicates the outcomes of successful or unsuccessful payments.

Data Persistence:

- Post a successful payment, the diagram showcases the Data Persistence activity, signifying the storage of relevant information such as prices, images, and selected room types.


Hotel user
Hotel Managment System
Start
Sign in
Sign up
show hotels offers
Select hotel
Select the room
Process room
Reserve this Room ?
Yes
Confirm Room
No
Display other room
make payment
choose Payment
Failed Payment
Payment Complete
Successfull payment
Payment Process
End

4. Mobile Application Development Lifecycle

The development lifecycle of our mobile application follows these key stages:

1. Requirement Analysis:

- Define and analyze functional and non-functional requirements. Collaborate with stakeholders to gather insights and ensure a comprehensive understanding of project goals.

2. Design:

- Create user interface designs and system architecture. Utilize design tools such as Figma or Sketch to visually conceptualize the application's layout and structure. Develop a robust system architecture that aligns with the identified requirements.

3. Implementation:

- Code the application based on the design specifications. Front-end development tasks will be handled by Rustam Kodirov, full-stack development by Zarshedjon Nasimov, and back-end development by Raziev Adil. Utilize relevant programming languages and frameworks, such as Flutter for front-end and Node.js for back-end.

4. Testing:

- Conduct unit tests, widget tests, and integration tests to ensure functionality. Boboyorov Ganisher, our dedicated tester, will systematically validate each component of the application.

5. Maintenance:

- Address bugs, implement updates, and maintain the application. Regularly monitor user feedback and promptly address any reported bugs. Implement updates to enhance features and adapt to evolving user needs. Maintenance activities will be coordinated by the development team to ensure the application's ongoing reliability.

5. Application Portability

Our Hotel Reservation System, built with Flutter, ensures smooth operation across various devices and platforms.

Utilise Flutter's Cross-Platform Support:

- Flutter enables a single codebase for Android and iOS, reducing development efforts.
- Its widget-based architecture ensures consistent UI across diverse screen sizes and resolutions.
- Hot Reload allows instant testing and debugging on multiple devices.

Implement Responsive Screen Layouts:

- Utilize Flutter's media queries for dynamic layout adjustments based on screen size.
- Leverage flexible widgets like `Expanded` and `Flexible` for scalable UIs.
- Design layouts to be aware of device orientation for seamless transitions.

External Insights:

- Refer to the official Flutter documentation for in-depth guidance and best practices.
- Engage in Flutter community forums to learn from experiences and stay updated.

6. Basic Hardware Requirements:

Our Hotel Reservation System is designed to operate efficiently on both Android and iOS platforms with minimal hardware requirements:

Min SDK Version:

- Android API Level 21 (Android 7.5, Nougat) and above for Android devices.
- iOS 10 and above for iOS devices.

Processor:

- Standard processor found in contemporary smartphones.

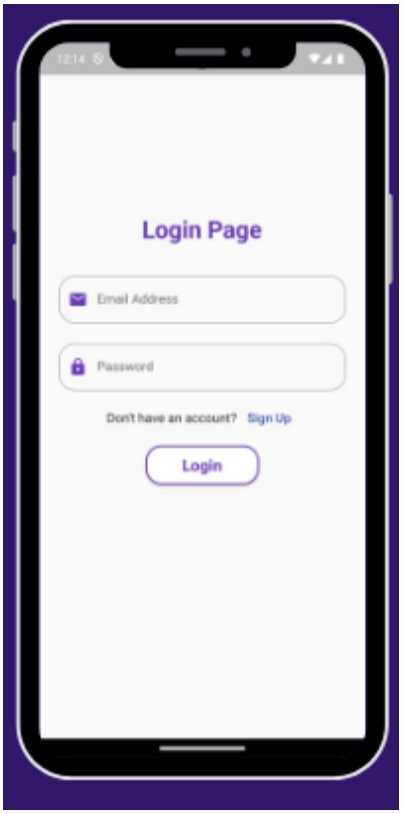
Memory (RAM):

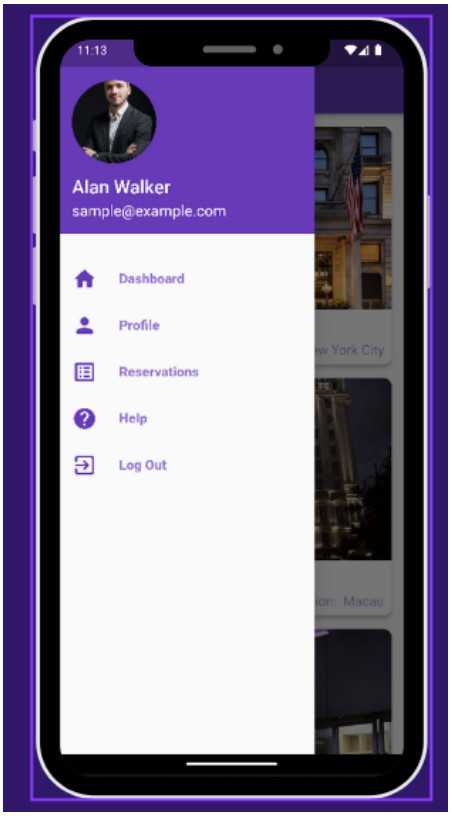
- 2 GB RAM or higher.

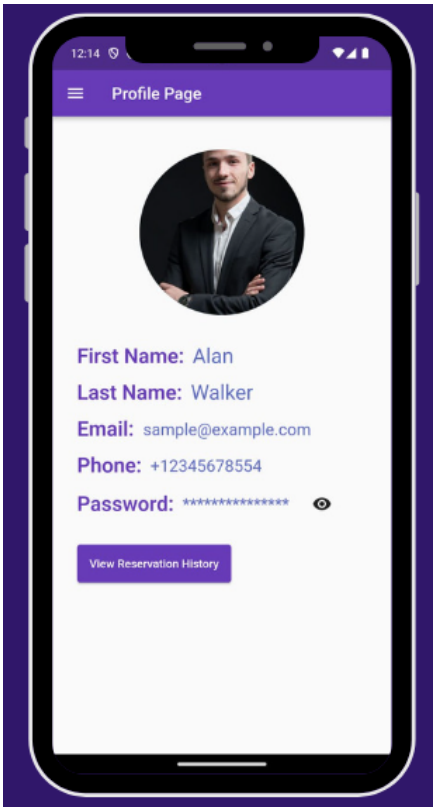
Storage:

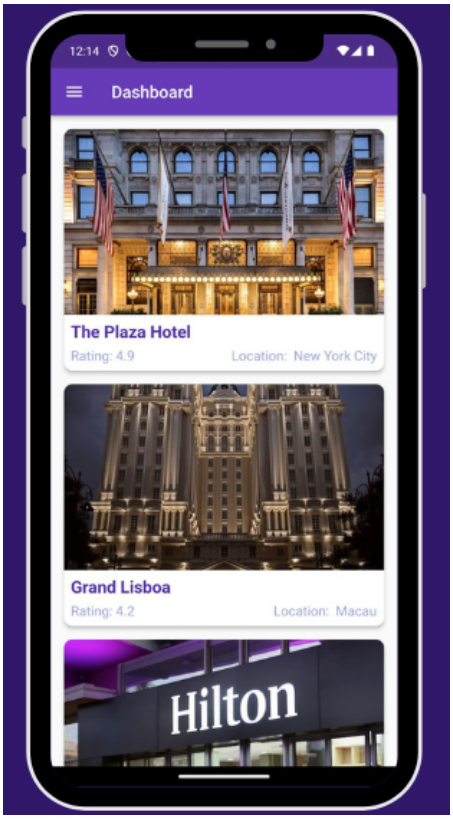
- 16 GB of internal storage.

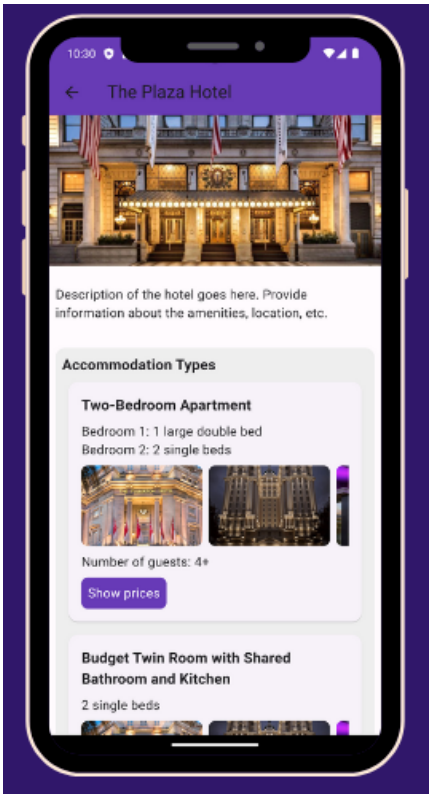
7. Screens:


Login Page
Email Address
Password
Don't have an account? Sign Up
Login


11:13
Alan Walker
sample@example.com
Dashboard
Profile
Reservations
Help
Log Out


12:14
Profile Page
First Name: Alan
Last Name: Walker
Email: sample@example.com
Phone: +12345678554
Password: **********
View Reservation History


12:14
Dashboard
The Plaza Hotel
Rating: 4.9
Location: New York City
Grand Lisboa
Rating: 4.2
Location: Macau
Hilton


10:30
The Plaza Hotel
Description of the hotel goes here. Provide information about the amenities, location, etc.
Accommodation Types
Two-Bedroom Apartment
Bedroom 1: 1 large double bed
Bedroom 2: 2 single beds
Number of guests: 4+
Show prices
Budget Twin Room with Shared Bathroom and Kitchen
2 single beds