American International University-Bangladesh (AIUB)

Department of Computer Science

Faculty of Science & Technology (FST)

Paying Guest Management System

A Software Requirement Engineering Project Submitted
By

Semester: SPRING_23_24		**Section: C**	**Group Number: 05**	
SN	Student Name	Student ID	Contribution in % (CO+CO2)	Individual Marks
21	MAHASHETAB HASSAN	20-44361-3	50%	
32	ATUNU SAHA	22-46357-1	50%	

The project will be Evaluated for the following Course Outcomes

EVALUATION CRITERIA	Total Marks (50)	
Introduction, Format, Submission, Defense	[10 Marks]	
System Overall Description & Functional Requirements	[10 Marks]	
System Quality Attributes and Project Requirements	[10 Marks]	
UML and E-R Diagram with Data Dictionary	[10 Marks]	
UI/UX Prototyping	[10 Marks]	

Software Requirements Specification

for

<Paying Guest Management System>

Version 1.1 approved

Prepared by

<MAHASHETAB HASSAN>

<ATUNU SAHA>

<AMERICAN INTERNATIONAL UNIVERSITY-BANGLADESH>

<20 March, 2024>

Contents

Revision History

Name	Date	Reason for Changes	Version
Mahashetab Hasan	10/05/2024	Purpose Correction	1.1

1. Introduction

1.1 Purpose

The aim of a Software Requirements Specification (SRS) document is multi-dimensional, seeking to offer a thorough comprehension of the software project to all stakeholders involved. The SRS acts as an official contract between the client and the development team, precisely defining the objectives of the software.

Software Requirement Specification (SRS) is a complete specification and description of requirements of the software that need to be fulfilled for the successful development of the software system. An SRS document's scope definition aids in the customer's comprehension of the objectives and value of the program. It also includes information on the estimated cost and duration of creation, making the project's boundaries evident. It conveys both the functional and non-functional requirements of the software to ensure everyone is in alignment on what needs to be developed. Test plans are generated by testing group based on the external behavior and support staff need it to understand what the software product is supposed to do.

The SRS aids in the identification of possible risks and uncertainties. It helps stakeholders to foresee problems and adjust mitigation efforts by outlining requirements and expectations precisely. Functional requirements specify the expected behavior of the system-which outputs should be produced from the given inputs. Stakeholders can make sure that the software solution not only satisfies short-term business requirements but also fits in with the organization's long-term strategic goals by providing further detail on these criteria within the SRS.

This goal captures the main objective of the SRS, which is to direct the creation of a software solution that addresses the unique requirements and difficulties encountered by the company while also being in line with the larger corporate goals and objectives.

1.2 Document Conventions

Convention	Description
Font	Use a standard font, such as Times New Roman or Arial, and keep the font size consistent throughout the document. Use a larger font size for headings and subheadings to differentiate them from the body text.
Bold	Bold words emphasize key terms or phrases, such as requirements or important concepts.
Bullet points and Numbering	bullet points and numbering are used to break down lists or requirements into manageable sections. This can help make the document more scannable and easier to read.
Tables	Tables are used to organize data. This can help make the information more structured and easier to understand.

1.3 Intended Audience and Reading Suggestions

The document is intended for a wide range of readers with various project-related tasks and responsibilities. The document is meant for the following various reader types:

Developers: Developers are software engineers and programmers who are in charge of planning, developing, and executing the project. They want precise technical knowledge and specs in order to comprehend how to integrate and expand the system into their development environment. The SRS provides a detailed description of the software system to be developed, including its functional and non-functional requirements. This information is essential for developers to understand what the system is supposed to do and how it should perform.

Project Managers: Project managers are in charge of managing the whole project and guaranteeing its effective completion. They need a high-level grasp of the system's capabilities, features, and possible influence on the timing and resources of the project. The SRS helps project managers to plan and manage the development of the software system. It provides information on the system's scope, schedule, budget, and risks.

Marketing Staff: To develop successful marketing campaigns, the marketing team must comprehend the product's core selling points, target market, and distinctive qualities. The SRS can be used by marketing staff to develop marketing materials and sales pitches for the software system. It provides information on the system's features, benefits, and target market.

Users: End-users or consumers who will use the product are included in this category. The SRS can be used by users to learn about the software system and how to use it. It provides information on the system's user interface, navigation, and features.

Testers: Testers are essential to assuring the quality of the system and locating any problems or defects. They have to concentrate on the passages that include thorough test instructions, test cases, anticipated results, and any unique testing requirements or concerns. The SRS can be used by testers to develop test cases and verify that the software system meets its requirements. It provides information on the system's functional and non-functional requirements.

Documentation Writers: This group is in charge of producing and updating project documentation. They must have an intimate knowledge of the system in order to provide correct and up-to-date documentation. The SRS can be used by documentation writers to create user manuals, training materials, and other documentation for the software system. It provides information on the system's features, benefits, and how to use it.

Here is a suggested sequence for reading the document based on the reader types:

1. Purpose section: The purpose section provides a brief overview of the software system, including its purpose, scope.

2. Functional requirements: The functional requirements section describes the features and functionality of the software system.

3. Non-functional requirements: The non-functional requirements section describes the system's performance, security, and other non-functional requirements.

4. Design: The design section describes the system's architecture, diagrams, and interfaces.

5. Implementation constraints: The implementation constraints section describes restrictions of the system.

6. User classes and characteristics: The user classes and characteristics section describe about user’s roles in the software.

7. Environment: The operating environment section describes in which platforms the software will operate.

8. Conclusion: The conclusion provides a summary about the whole software system

1.4 References

[1] Draw.io - free flowchart maker and diagrams online. [Online]. Available: https://app.diagrams.net/?src=about# [Accessed: 01-May-2024].

[2] Murahari Prithvi Yash, Chinmay Choudhary, Akanksha Lakra, Swati Dewangan “RentoAxis: Android App for Paying Guest Management,” researchgate.net. [Online]. Available: (PDF) RentoAxis: Android App for Paying Guest Management (researchgate.net)

2. Overall Description

2.1 Product Perspective

The problem background for developing paying guest management system originates from several key challenges faced by both hoteliers and guests in the hospitality industry.
These challenges include:

Manual Booking Processes: Traditional methods of booking rooms often involve manual paperwork or phone calls, leading to inefficiencies, errors, and delays in the reservation process.
Room Availability Management: Without a centralized system, hotels struggle to manage room inventory effectively, leading to discrepancies between online booking platforms and actual availability.

The paying guest management system being specified in this Software Requirements Specification (SRS) is a new, self-contained product developed to address the aforementioned challenges in the hospitality industry. It is not a follow-on member of a product family nor a replacement for existing systems, but rather a standalone solution designed to streamline hotel operations and enhance guest experiences.

The primary business objectives of developing this system include:

Increase Booking Conversions: Streamlining the booking process to make it easier and more convenient for guests, thereby increasing booking conversions and capturing a larger market share.
Optimize Revenue Generation: Through real-time inventory management, dynamic pricing strategies, and efficient payment processing, maximize room utilization and revenue generation for hotels.
Enhance Guest Satisfaction: Personalize guest experiences based on preferences and booking history to foster loyalty and positive reviews, leading to repeat bookings and referrals.
Improve Operational Efficiency: Automate administrative tasks, reduce manual errors, and improve staff productivity to lower operational costs and enhance service quality.

2.2 Product Functions

**Major Functions
of product:**

User:

1. Sign-up
2. Login
3. Post Room
4. Check Vacancy
5. Select Room
6. Make payment
7. Confirm Checkout
8. Update Booking Status
9. Logout

Room:

1. Sign-up
2. Login
3. Preferred room type
4. Request booking status
5. Service History
6. Logout

Host-Registration:

1. Sign-up
2. Login
3. Request for post room
4. Request service status
5. Service History
6. Logout

Guest-Registration:

1. Sign-up
2. Login
3. Check room status
4. Cancel booking
5. Confirm booking
6. Make payment
7. Logout

Admin:

1. Sign-up
2. Login
3. Check room status
4. Cancel booking
5. Confirm booking
6. View payment record
7. Maintain booking status
8. Make payment
9. Logout

Booking:

1. Sign-up
2. Login
3. Booking approval
4. Maintain booking status
5. Logout

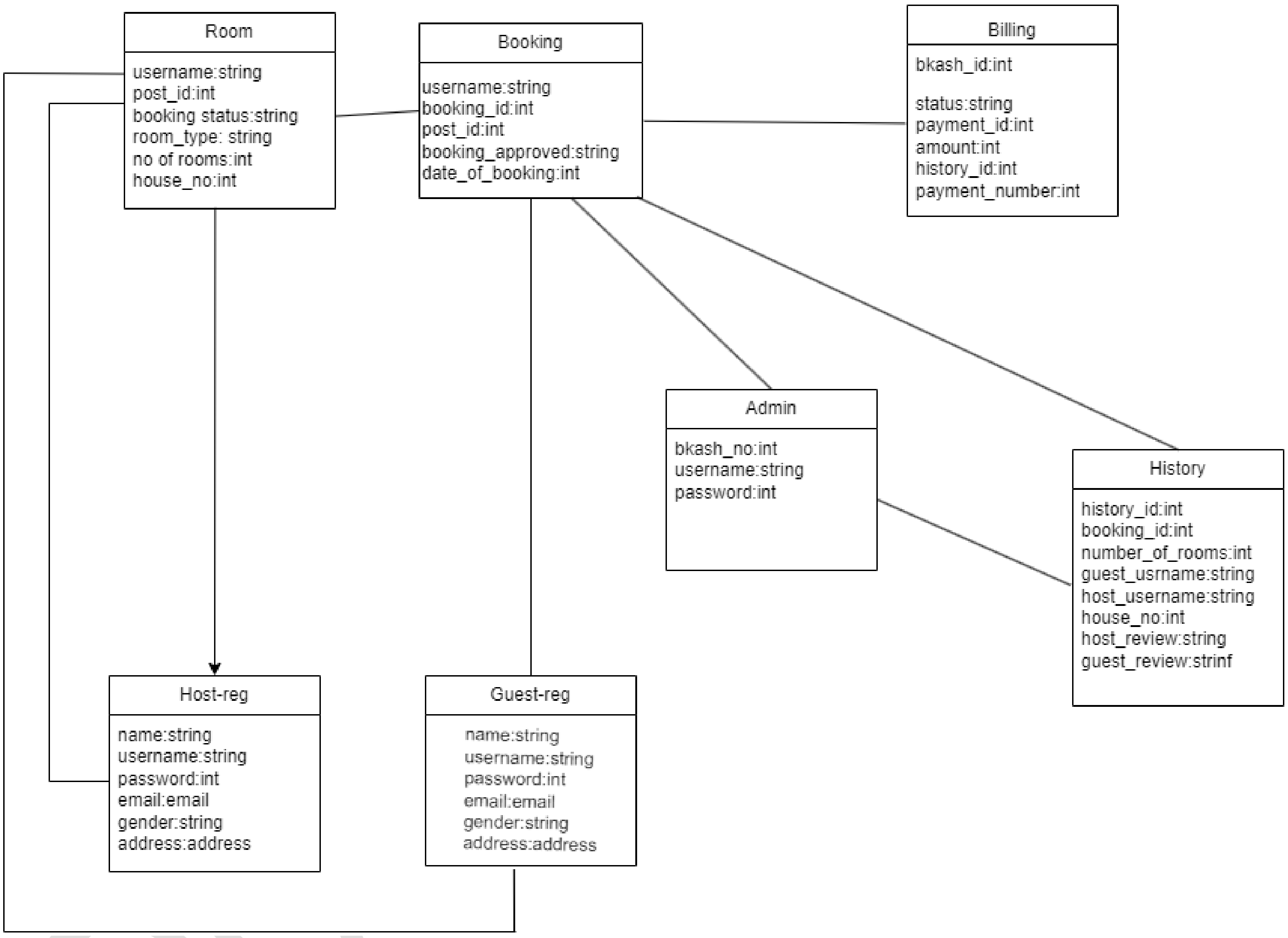
Billing:

1. Sign-up
2. Login
3. Payment status
4. Make payment
5. Logout

History:

1. Sign-up
2. Login
3. Review the guest and host.
4. Check-in and checkout history
5. Logout

Class Diagram



2.3 User Classes and Characteristics

Here are some of the pertinent characteristics of each user class for a paying guest management system, and some requirements that may be specific to each user class:

Admin:

- May have limited technical expertise.
- May use the system frequently.
- May have varying levels of experience with similar systems.
- Requirements: The system should provide advanced features for checking room status, booking confirmation and cancelation. Also maintain the booking system and make payment.
- Experience is mandatory.

Room(post):

- May be responsible for maintaining the paying guest management system.
- May require advanced technical expertise.
- May require access to advanced features such as user management and system monitoring.
- Requirements: The system should provide advanced administration features such as booking status, number of rooms, username, date of post.
- Experience is mandatory.

Host-Registration:

- May have limited technical expertise.
- May use the system frequently.
- May have varying levels of experience with similar systems.
- Requirements: The system should be easy to use and understand, with clear instructions and intuitive interfaces. Help documentation should be readily available, and the system should provide easy-to-use features for setting up and managing guest schedules.
- Experience should be mandatory.

Guest-Registration:

- May have limited technical expertise.
- May use the system infrequently.
- May not varying levels of experience with similar systems.
- Experience shouldn’t be mandatory.

Booking:

- May be responsible for providing assistance to users who have booking with this system.
- May have limited technical expertise.
- May use the system frequently.
- Requirements: The system should provide detailed help documentation and give the approval of the booking.

Billing:

- May have limited technical expertise.
- May be responsible for providing assistance to users who have billing with this system.
- May use the system frequently.
- Requirements: The system should be scalable and able to make payment

2.4 Hardware and Operating Environment

It will operate on any smartphone with android or IOS or Tizen or Windows, PC with Windows or Linux or Mac or other OS by using our application or any browser. Every device with latest or recent OS can run our application.

2.5 Design and Implementation Constraints

Hardware Limitations: A our software is for every device including mobile phones, for this our software needs to be lite. Because of this some options are limited for the developers.

Interfaces to other applications: The software needs to interface with other mobile banking applications. This can limit the options available to developers, as they will need to use technologies that are compatible with the other applications.

Budget and timeline constraints: The available options for development may be limited by the project's budget and timeline constraints. For example, a custom development approach may be too expensive or time-consuming to implement within the project's constraints.

Specific technologies, tools, and databases to be used: The company may have specific technologies, tools, and databases that they want to use. This can limit the options available to developers, as they will need to use the technologies that the company has specified.

Security considerations: The software needs to meet certain security requirements. This can limit the options available to developers, as they will need to use technologies that support the required security features.

Design conventions or programming standards: The company may prefer to utilize particular design conventions or programming standards. As a result, developers' options may be limited since they must adhere to the company's requirements.

2.6 User Documentation

Here are some user documentation components that could be delivered along with the software in a smart irrigation system:

- **User manual**: A detailed guide that explains how to install, configure, and use the smart irrigation system. This manual can be in a printable format, such as a PDF document.
- **Online help**: An interactive help system that provides users with assistance and guidance on how to use the system. This can be in the form of a web-based help system, accessed through a browser, or built-in to the user interface of the software.
- **Tutorials**: Step-by-step tutorials that guide users through common tasks, such as setting up an irrigation schedule, configuring alerts, and generating reports.
- **Release notes**: Documentation that describes the changes and new features that have been introduced in each software release.

3. System Requirements

3.1 System Features

1. Room Management

Functional Requirements (FRs)

1.1 Hosts shall be able to post rooms with details such as room type, amenities, price, and availability.
1.2 Hosts shall have the capability to edit, activate, or deactivate posted rooms as needed.
1.3 The system shall update the booking status of rooms to "booked" when a guest successfully books a room.
1.4 If a room is booked, the system shall notify the admin and update the booking status accordingly.
1.5 If a room is vacant, guests shall be able to view and book it through the system.

Priority Level: High
Precondition: Host is logged in and provides accurate room details.

Benefit: 8 (out of 10)

Penalty: 3 (out of 10)

Cost: 5 (out of 10)

Risk: 4 (out of 10)

2. Booking Management

Functional Requirements (FRs)

2.1 The system shall send booking requests to the admin for review upon guest booking.
2.2 Admins shall have the ability to either confirm or reject booking requests.
2.3 Upon admin confirmation, booking room details shall appear on the check-in page for the guest.
2.4 If the admin rejects a booking request, the booking status of the room shall remain empty.
2.5 Upon guest check-out, the booking status of the room shall be updated to empty.

Priority Level: High
Precondition: Guest successfully books a room and admin reviews the booking request.

Benefit: 7

Penalty: 4

Cost: 6

Risk: 5

3. Payment Processing

Functional Requirements (FRs)

3.1 Guests shall be able to pay room rent using a bkash number and transaction ID.
3.2 Payment requests shall be sent to the admin for verification.
3.3 Admins shall confirm payment details for accuracy.
3.4 Upon admin confirmation, payment information shall be stored and displayed in the history of both the host and guest dashboard.
3.5 If payment details are incorrect, the system shall prompt the guest to resubmit payment information.

Priority Level: High
Precondition: Guest must be logged in and provides accurate payment details.

Benefit: 9

Penalty: 3

Cost: 7

Risk: 6

4. Review and Feedback

Functional Requirements (FRs)

4.1 After check-out, guests and hosts shall have the option to share reviews with each other.
4.2 Reviews shared between guests and hosts shall be stored and accessible for future reference.
4.3 Reviews shall contribute to improving the system and user experience.

Priority Level: Medium
Precondition: Guest and host have completed their stay and have access to the review feature.

Benefit: 6

Penalty: 2

Cost: 4

Risk: 3

5. Payment Withdrawal

Functional Requirements (FRs)

5.1 Hosts shall be able to request payment withdrawal from the admin after guest payment confirmation.
5.2 Admins shall process payment withdrawal requests from hosts.
5.3 Upon admin approval, the desired payment shall be transferred to the host.

Priority Level: High
Precondition: Host requests payment withdrawal and admin approves the request.

Benefit: 8

Penalty: 4

Cost: 5

Risk: 5

3.2 Non-Functional/Quality Requirements

QA1: Performance: The system shall be able to handle concurrent requests from multiple users without significant degradation in response time, ensuring that each user's experience remains responsive.

Priority Level: High
Precondition: N/A
Cross-references: QA3, QA5

QA2: Reliability: The system shall maintain an uptime of at least 99.9%, ensuring that it is consistently available for users to access and perform necessary tasks.

Priority Level: High
Precondition: N/A
Cross-references: QA6, QA9

QA3: Scalability: The system architecture shall be designed to scale horizontally and vertically to accommodate an increasing number of users and data volume over time without sacrificing performance.

Priority Level: High
Precondition: N/A
Cross-references: QA1, QA5

QA4: Security: The system shall implement robust security measures, including data encryption, access control, and authentication mechanisms, to protect sensitive user information and prevent unauthorized access or data breaches.

Priority Level: High
Precondition: N/A
Cross-references: QA7, QA10

QA5: Availability: The system shall provide 24/7 availability, ensuring that users can access the system and perform necessary tasks at any time without disruption.

Priority Level: High
Precondition: N/A
Cross-references: QA1, QA3

QA6: Maintainability: The system shall be designed with modular and well-documented code, facilitating ease of maintenance, updates, and enhancements by system administrators and developers.

Priority Level: Medium
Precondition: N/A
Cross-references: QA2, QA9

QA7: Compliance: The system shall comply with relevant industry standards, regulations, and legal requirements related to data privacy, security, and financial transactions, ensuring legal compliance and mitigating regulatory risks.

Priority Level: High
Precondition: N/A
Cross-references: QA4, QA10

QA8: Interoperability: The system shall be compatible with various operating systems, web browsers, and devices, ensuring seamless integration and accessibility for users across different platforms.

Priority Level: Medium
Precondition: N/A
Cross-references: QA3, QA6

QA9: Usability: The system interface shall be intuitive and user-friendly, allowing users to navigate through the application and perform tasks with minimal training or guidance.

Priority Level: High
Precondition: N/A
Cross-references: QA4, QA8

QA10: Performance Efficiency: The system shall optimize resource utilization, minimize response time, and maximize throughput to ensure efficient use of computing resources and provide a satisfactory user experience.

Priority Level: High
Precondition: N/A
Cross-references: QA1, QA4

QA11: Accessibility: The system shall comply with accessibility standards, ensuring that users with disabilities can effectively use the application through keyboard navigation, screen readers, and other assistive technologies.

Priority Level: Medium
Precondition: N/A
Cross-references: QA9

3.3 Project Requirements

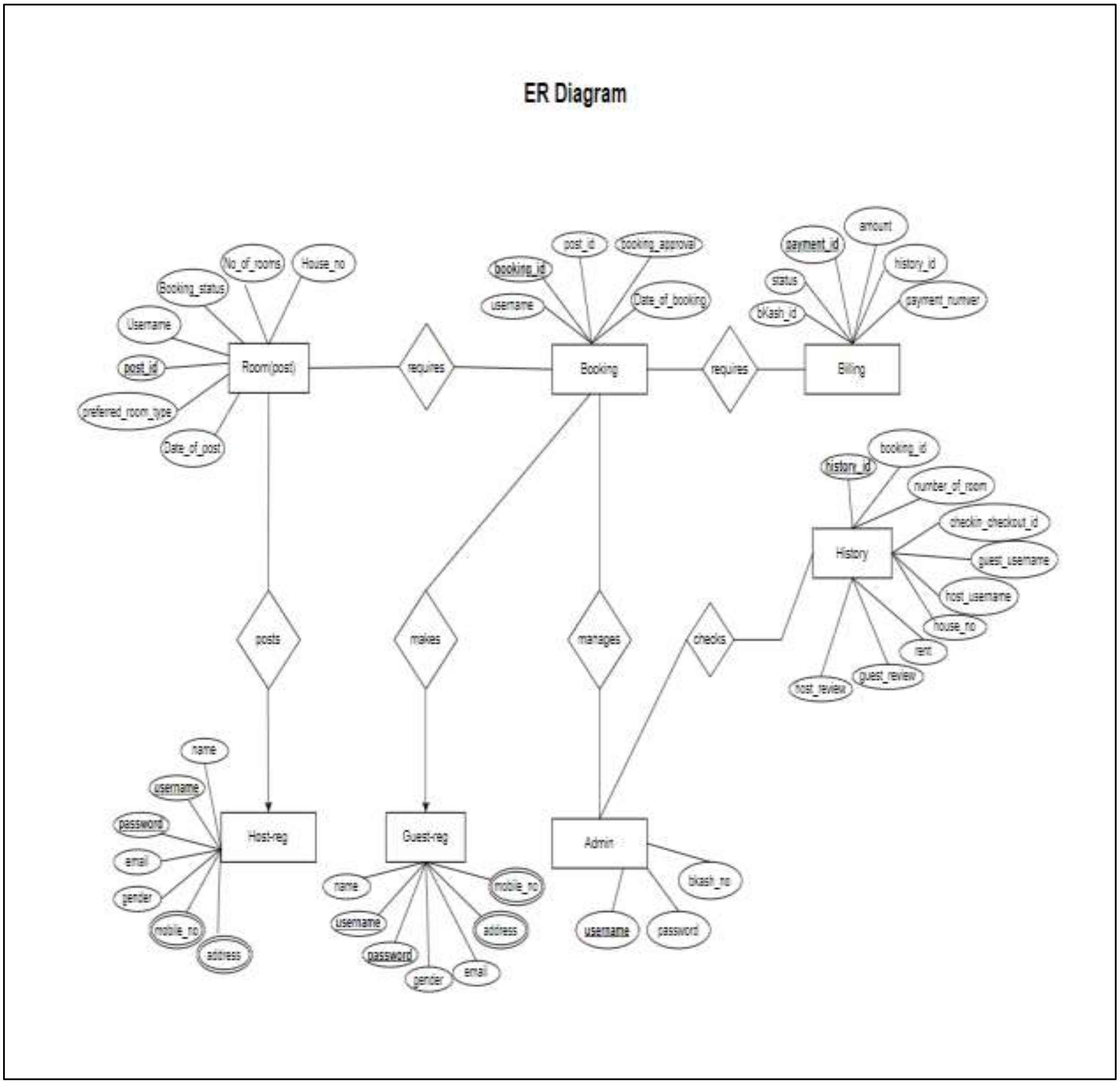
Tools:

- The system needs enough data storage capacity in the database to keep all the user's information.
- The system needs backup processes to manage data corruption, or a human-caused event, such as a malicious attack (virus or malware), or accidental deletion of data.
- The system developer needs selenium tools in perform testing activities in week 6.

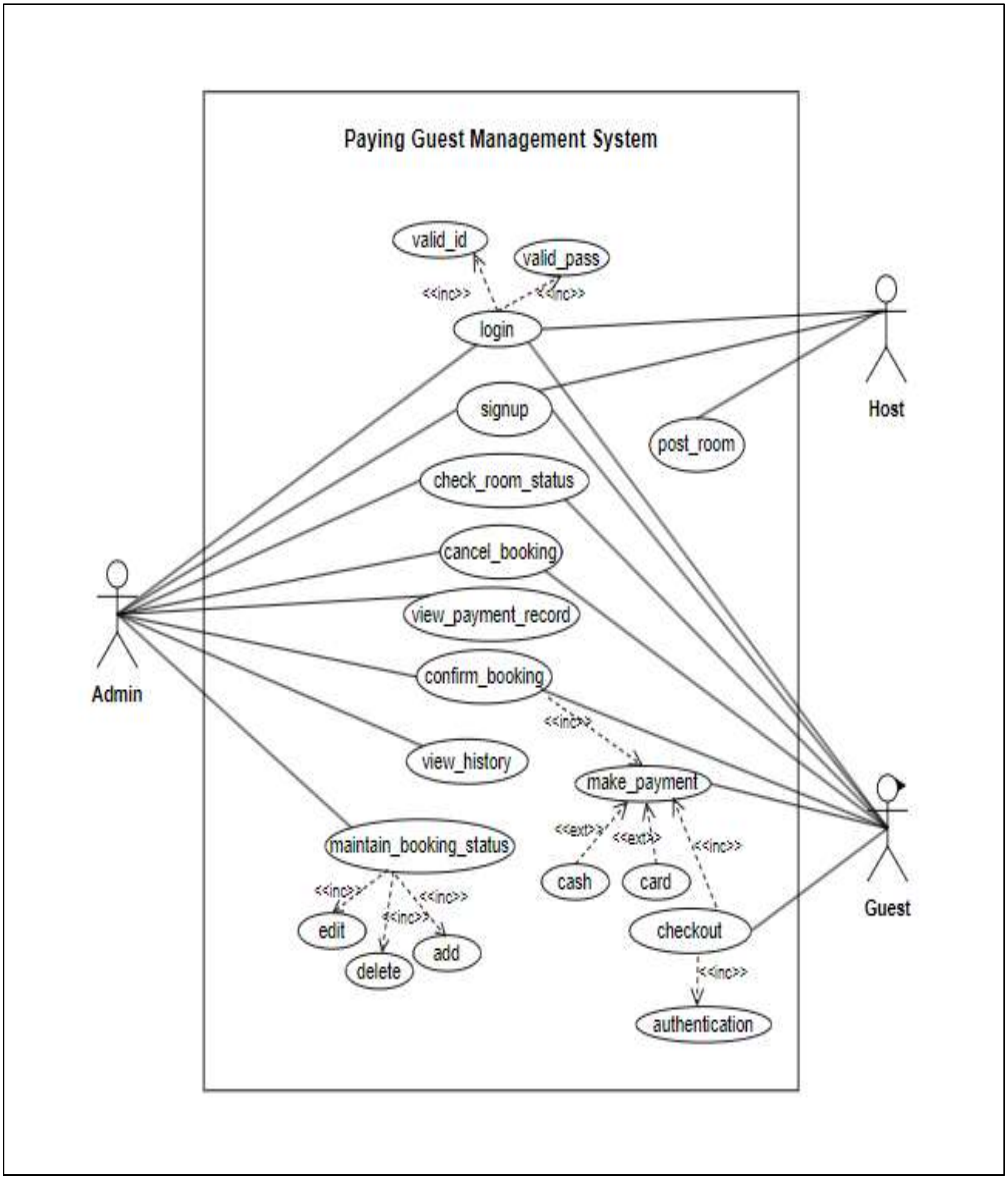
4. Design and Interface Requirements

4.1 UML Diagrams

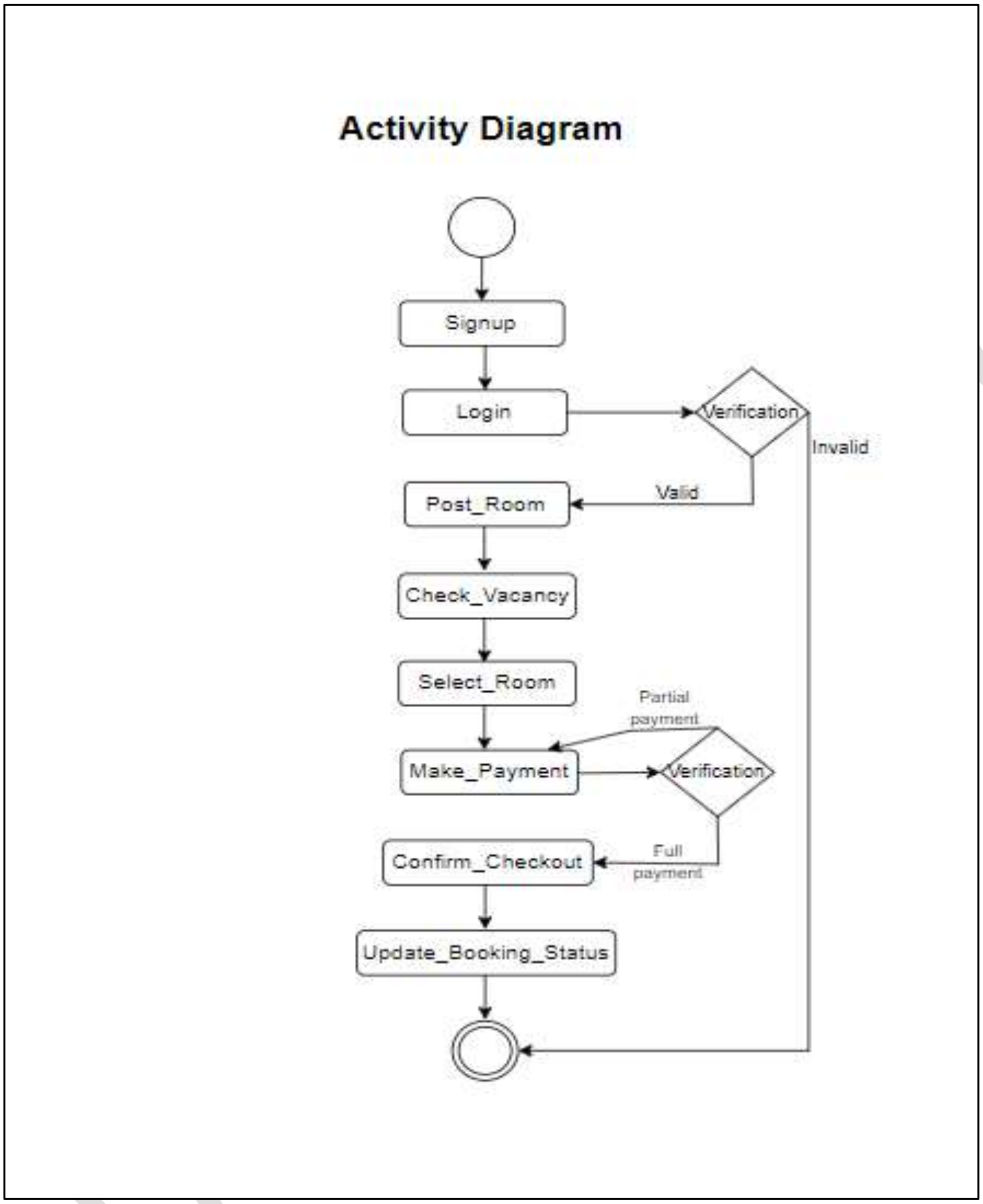
- E-R Diagram:



- Use Case Diagram:



- Activity Diagram:



4.2 Data Dictionary

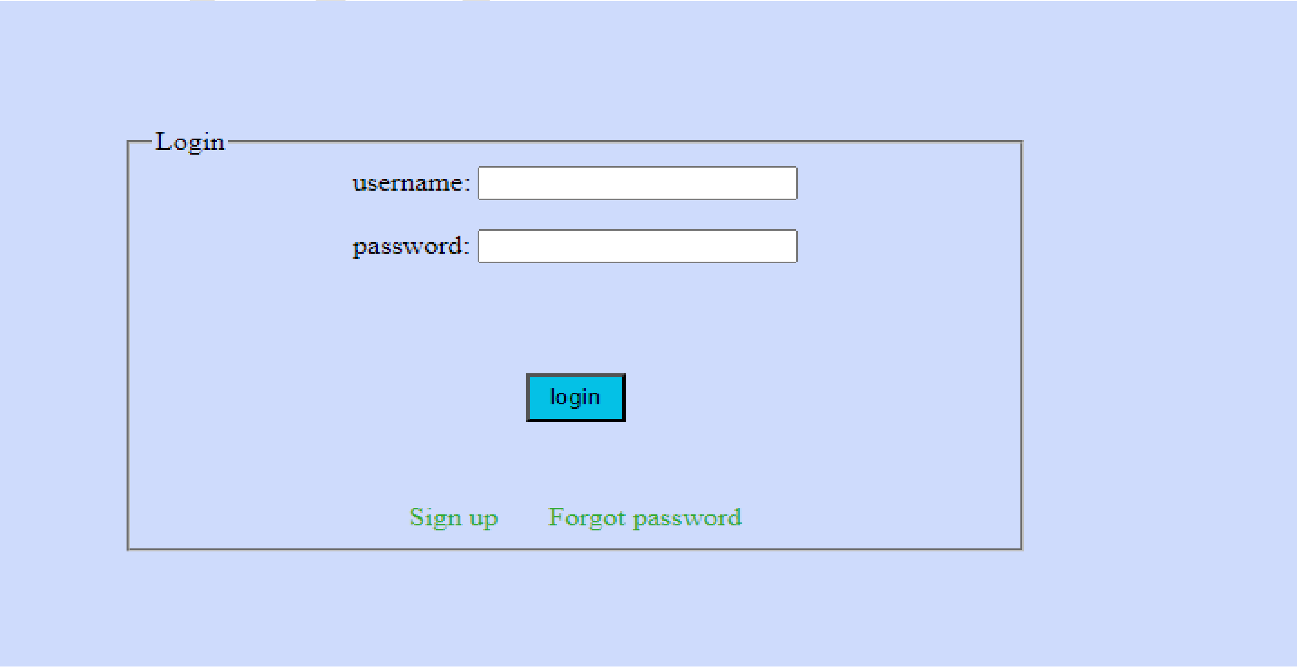
Entity	Attribute	Type/Size	Validation	Key
Host-reg	name	VARCHAR2(50)	Required	
Host-reg	email	VARCHAR2(50)	Required	
Host-reg	Phone	NUMBER(11)	Required	
Host-reg	address	VARCHAR2(50)	Required	
Host-reg	gender	VARCHAR2(10)	Required	
Host-reg	username	VARCHAR2(20)	Required	
Host-reg	password	VARCHAR2(50)	Required	
checkin_checkout	booking_id	NUMBER(38)	0000000001-9999999999	Primary
checkin_checkout	checkin_status	VARCHAR2(60)	Required	
checkin_checkout	checkin_date	DATE	Required	
checkin_checkout	checkout_status	VARCHAR2(60)	Required	
checkin_checkout	checkout_date	DATE	Required	
booking	booking_id	NUMBER(38)	0000000001-9999999999	Primary
booking	post_id	NUMBER(38)	0000000001-9999999999	Primary
booking	username	VARCHAR2(100)	Required	
booking	booking_approval	VARCHAR2(60)	Required	
booking	date_of_booking	DATE	Required	
post	post_id	NUMBER(38)	0000000001-9999999999	Primary
post	House_Number	VARCHAR2(30)	Required	
post	Num_of_Rooms	VARCHAR2(20)	Required	
post	preferred_room_types	VARCHAR2(100)	Required	
post	Rent	VARCHAR2(120)	Required	
post	Address	VARCHAR2(100)	Required	
post	username	VARCHAR2(40)	Required	
post	Booking_status	VARCHAR2(20)	Required	
post	Date_of_post	DATE	Required	
guest_registration	name	VARCHAR2(20)	Required	
guest_registration	email	VARCHAR2(50)	Required	

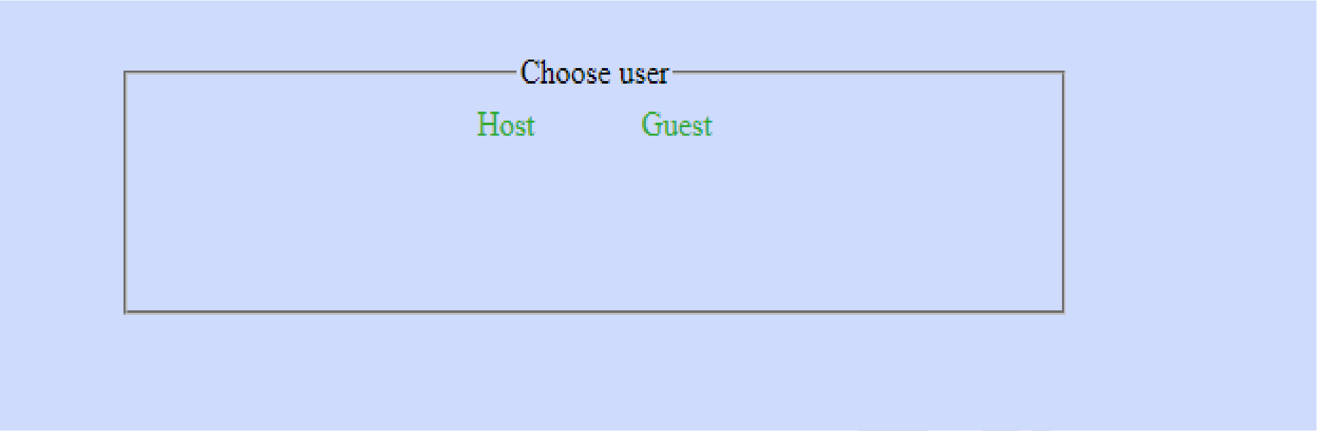
guest_registration	phone	NUMBER(11)	Required	
guest_registration	address	VARCHAR2(40)	Required	
guest_registration	gender	VARCHAR2(10)	Required	
guest_registration	username	VARCHAR2(30)	Required	
guest_registration	password	VARCHAR2(50)	Required	
history	booking_id	NUMBER(38)	0000000001-9999999999	Primary
history	checkin_checkout_id	NUMBER(40)	0000000001-9999999999	
history	guest_username	VARCHAR2(50)	Required	
history	host_username	VARCHAR2(60)	Required	
history	house_number	VARCHAR2(30)	Required	
history	number_of_rooms	NUMBER(20)	Required	
history	rent	NUMBER(38)	Required	
history	address	VARCHAR2(50)	Required	
history	host_review	VARCHAR2(100)	Required	
history	guest_review	VARCHAR2(100)	Required	

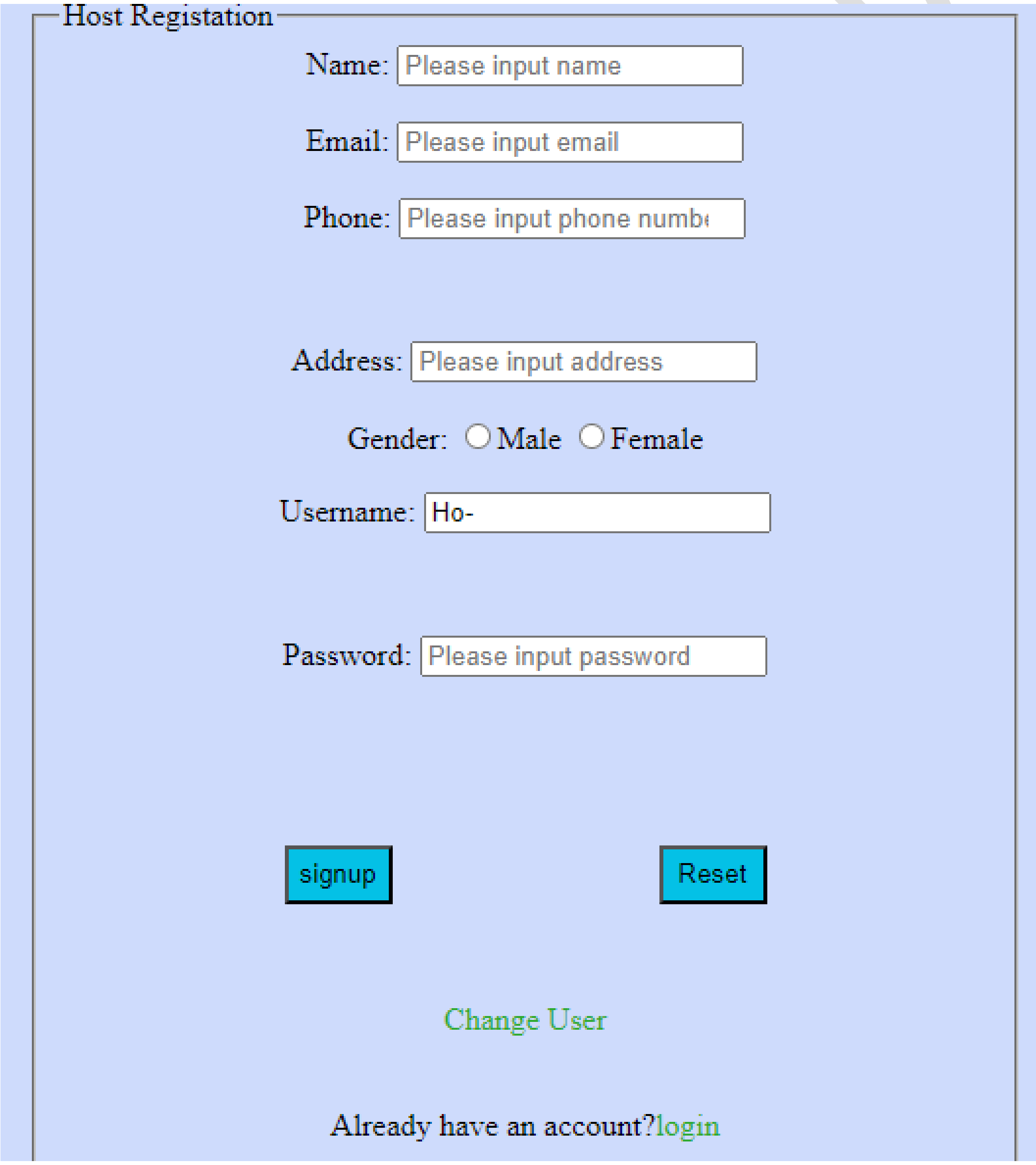
4.3 UI/UX Design Specification

- Tools for prototyping: Visual Studio

Login:






Host Registation

Name: Please input name

Email: Please input email

Phone: Please input phone numbe

Address: Please input address

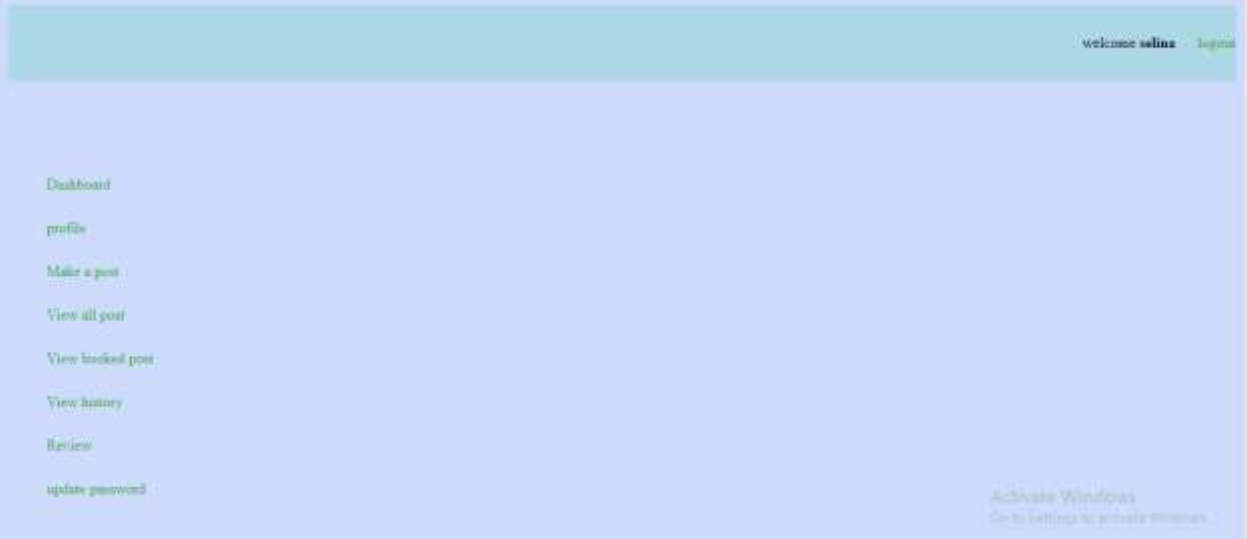
Gender: ○ Male ○ Female

Username: Ho-

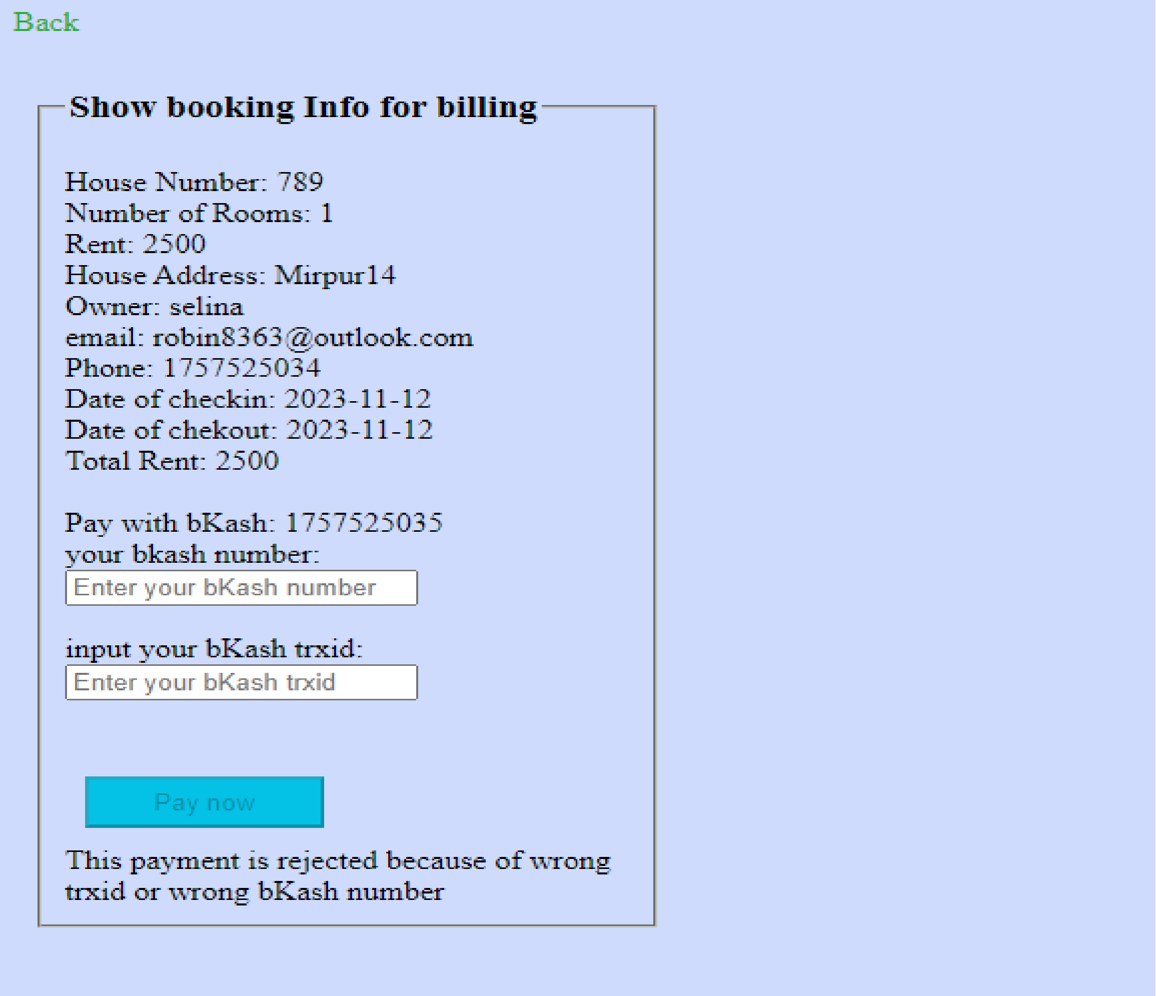
Password: Please input password

Already have an account?login

Host Dashboard:



Billing:



Show booking Info for billing

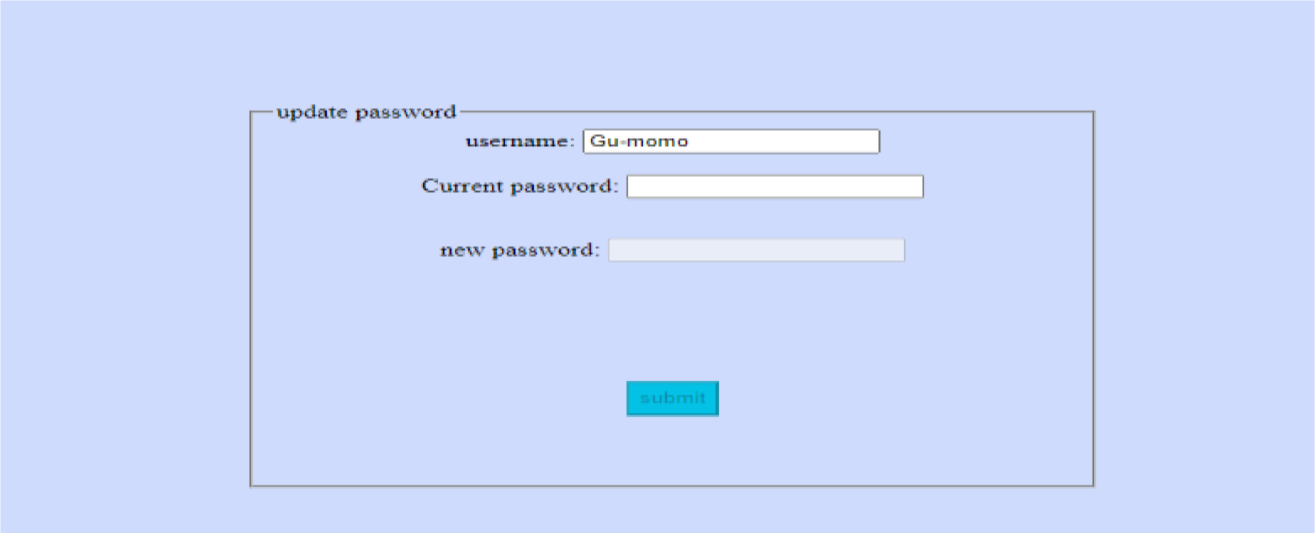
House Number: 789
Number of Rooms: 1
Rent: 2500
House Address: Mirpur14
Owner: selina
email: robin8363@outlook.com
Phone: 1757525034
Date of checkin: 2023-11-12
Date of chekout: 2023-11-12
Total Rent: 2500

Pay with bKash: 1757525035
your bkash number:
Enter your bKash number

input your bKash trxid:
Enter your bKash trxid

This payment is rejected because of wrong trxid or wrong bKash number

Update Password:



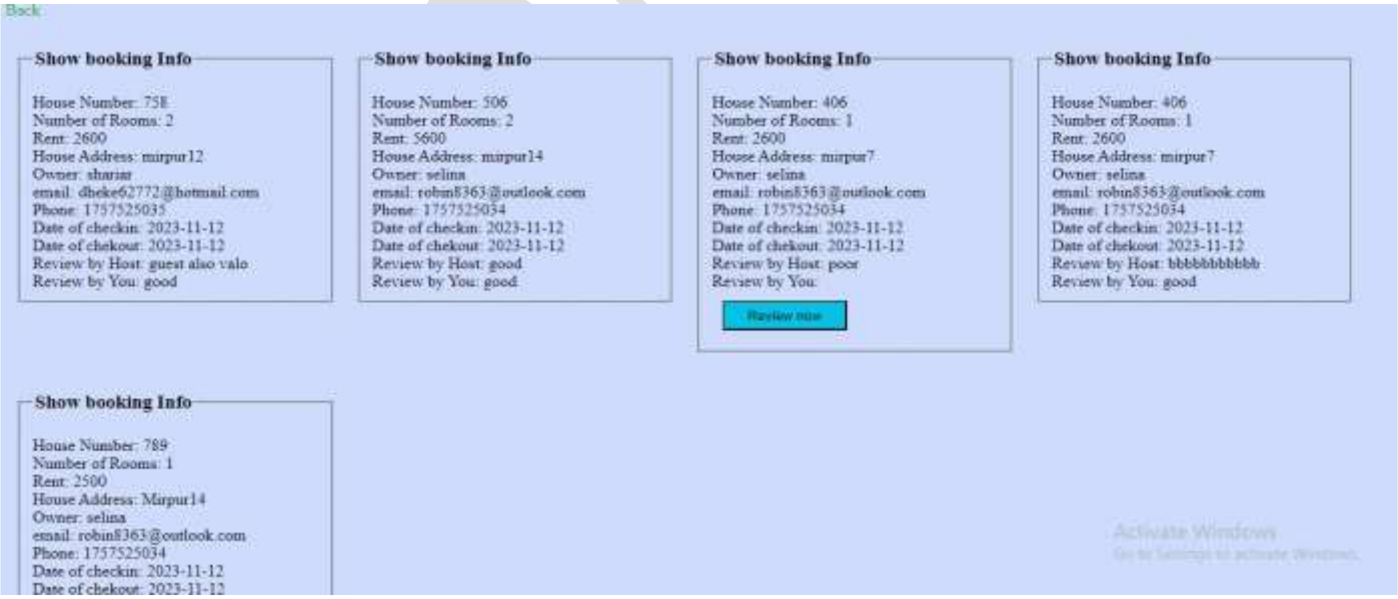
update password

username: Gu-momo

Current password:

new password:

Review:



Show booking Info

House Number: 758
Number of Rooms: 2
Rent: 2600
House Address: mirpur12
Owner: shariar
email: dheke62772@hotmail.com
Phone: 1757525035
Date of checkin: 2023-11-12
Date of chekout: 2023-11-12
Review by Host: guest also valo
Review by You: good

Show booking Info

House Number: 506
Number of Rooms: 2
Rent: 5600
House Address: mirpur14
Owner: selina
email: robin8363@outlook.com
Phone: 1757525034
Date of checkin: 2023-11-12
Date of chekout: 2023-11-12
Review by Host: good
Review by You: good

Show booking Info

House Number: 406
Number of Rooms: 1
Rent: 2600
House Address: mirpur7
Owner: selina
email: robin8363@outlook.com
Phone: 1757525034
Date of checkin: 2023-11-12
Date of chekout: 2023-11-12
Review by Host: poor
Review by You:

Show booking Info

House Number: 406
Number of Rooms: 1
Rent: 2600
House Address: mirpur7
Owner: selina
email: robin8363@outlook.com
Phone: 1757525034
Date of checkin: 2023-11-12
Date of chekout: 2023-11-12
Review by Host: bbbbbbbbbbb
Review by You: good

Show booking Info

House Number: 789
Number of Rooms: 1
Rent: 2500
House Address: Mirpur14
Owner: selina
email: robin8363@outlook.com
Phone: 1757525034
Date of checkin: 2023-11-12
Date of chekout: 2023-11-12