كلية الحوسبة والمعلوماتية |
College of Computing and Informatics
الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

College of Computing & Informatics (CCI)

SENIOR PROJECT-I REPORT

Roommate Matching Platform MassakeN

Author(s):

Student S190024084 (Leader) Student Name: BASSAM ALI

Student S200019372(Member) Student Name: MOHAMMED ALABDULLAH

Student S1900138612(Member)Student Name : MOHAMMED ALSUHAYL

Student S160047635(Member) Student Name: MOHAMMED ALNASHIRI

Project Supervisor:

DR.Hamdan Alzhrani

كلية الحوسبة والمعلوماتية |
College of Computing and Informatics
الجامعة السعودية الإلكترونية
SAUDI ELECTRONIC UNIVERSITY
2011-1432

Roommate Matching Platform MassakeN

By:

- BASSAM ALI
- ALABDULLAH
- MOHAMMED ALSUHAYL
- MOHAMMED ALNASHIRI

Thesis/Project submitted to:

College of Computing & Informatics, Saudi Electronic University, Riyadh, Saudi Arabia.

In partial fulfillment of the requirements for the degree of:

BACHELOR OF SCIENCE IN INFORMATION TECHNOLOGY

Dr. Hamdan A. Alzahrani

Project Supervisor

Project Committee Chair

ABSTRACT

The project focuses on solving the challenge of finding suitable roommates in Saudi Arabia. Many people struggle to find compatible roommates that match their lifestyle and preferences, leading to dissatisfaction and conflicts. The current methods, such as classified ads and word of mouth, are unreliable and do not allow for a comprehensive search.

The project aims to develop a user-friendly platform that helps people find compatible roommates based on their preferences and needs. The app will feature advanced filtering and search options, a verification system, detailed user profiles, a messaging system, and reviews and ratings of users. The platform will allow users to search and filter potential roommates based on criteria such as age, gender, education, occupation, and smoking preferences. Users will also be able to post rental listings.

The project objectives include researching user needs, designing a user-friendly interface, developing the app using appropriate technologies, conducting thorough testing, deploying the app to app stores, website creating a legal disclaimer and terms of service, continuously monitoring and maintaining the platform, and providing an efficient way for people to find compatible roommates in Saudi Arabia.

DEDICATION

We dedicate this project to our parents who have supported us throughout our journey. Their love and encouragement has been the driving force behind our success. We also dedicate this project to Dr. Hamdan Alzahrani, our professor who has taught us the skills and knowledge needed to make this project a reality.

This project also dedicated to all individuals who have struggled to find a suitable roommate. Your difficulties and experiences have inspired us to create a platform that makes the process of finding a roommate easier and more efficient.

To all those who have experienced conflicts and dissatisfaction in their living arrangements, we hope that this app will help you find a compatible roommate and bring peace and comfort to your life.

Finally, we dedicate this project to our families and friends who have supported and encouraged us throughout the development process. Without their unwavering love and support, this project would not have been possible.

PREFACE

The process of finding a suitable roommate can be challenging, especially in Saudi Arabia. Many people struggle to find roommates that are compatible with their lifestyle and preferences, which can lead to conflicts and dissatisfaction with living arrangements. Massaken aims to address this problem by providing a platform that allows users to search and filter potential roommates based on various criteria such as age, gender, education level, occupation, and smoking preferences. The platform will also allow users to post rental listings from individuals.

some of the best features of competitive roommate matching platforms include:

- Wide user base and high number of listings
- Advanced search and filtering options
- In-app messaging system
- Verification system for safety and reliability
- Reviews and ratings system for users
- User-friendly interface

However, some of the worst features of these platforms include:

- Limited geographical coverage
- Lack of attention to cultural norms and preferences
- Inconsistent or unreliable verification systems
- Difficult or slow customer support
- High fees or charges

Our platform aims to address these shortcomings by offering a wide range of features tailored specifically to the Saudi Arabian market. We will focus on developing a robust verification system to ensure the safety and reliability of users. Our platform will also incorporate cultural norms and preferences to provide a more personalized and targeted roommate matching experience. Additionally, we will offer a simple and user-friendly interface, quick and efficient customer support, and low or no fees to attract more users. Overall, we believe our platform will offer a more comprehensive and effective solution for finding suitable roommates in Saudi Arabia.

Platform	Best Features	Worst Features	Why our Platform is Better
Roomi	In-app payment system, extensive property filters and options, calendar for rent payments and bills tracking	Unreliable messaging system, high fees for paid listings, inconsistent customer support	Our platform's messaging system is more reliable, our payment system is equally secure and our customer support team is more responsive
Roomster	Compatibility quiz for roommates, identity verification for safety, roommate matching algorithm based on common interests and preferences	Paid features are too expensive, cluttered user interface, fake profiles and scammers	Our platform's paid features are priced more reasonably, our user interface is cleaner and more user-friendly, and our verification process ensures the safety of our users
Cozy	User-friendly interface, rent payment and credit reporting system, customizable rental applications	No roommate search function, no mobile app, customer service is not available on weekends	Our platform includes a roommate search function, we have a mobile app available for both iOS and Android, and our customer service is available seven days a week
Diggz	Machine learning algorithm for roommate matching, in-app payment system, user review and rating system	Basic user interface, limited search filters, not available in all cities	Our platform's user interface is more modern and user-friendly, our search filters are more extensive, and we are available in more cities across the world

Overall, our platform plan is to stands out from the competition with our reliable messaging system, reasonable pricing for paid features, thorough verification process, user-friendly interface, and availability on both mobile and desktop. Additionally, our extensive search filters and roommate matching algorithm ensure that users are quickly and easily connected with compatible roommates, making the process of finding a roommate much more efficient.

REVISION HISTORY

here is the revision history in table format:

Name	Date	Reason for Changes	Version
Bassam Ali	15/01/2023	Initial report creation	1.0
M. Alnashiri	20/01/2023	Updated information on the app platform	1.1
M. Alabdullah	21/01/2023	Added section on competitive analysis	1.2
Bassam Ali	22/02/2023	Added section on use cases	1.3
M. Alsuhayl	27/05/2023	Added section on preface and acknowledgment	1.4
Bassam Ali	06/06/2023	Added section on revision history	1.5

TABLE OF CONTENTS

CHAPTER 1: INTRODUCTION

1.1 Project Background/Overview:

The project aims to develop an app platform that helps users find suitable roommates based on their preferences and needs. The app will include features such as advanced filtering and search options, a verification system, detailed user profiles, a messaging system, and reviews and ratings of users. The main goal of the app is to make the process of finding a roommate easier and more efficient by allowing users to quickly find potential roommates that match their criteria.

The app is designed to help people find roommates in Saudi Arabia. The app will allow users to search and filter potential roommates based on various criteria such as age, gender, education level, occupation, and smoking preferences. The app will also allow users to post rental listings from individuals.

1.2 Problem Description:

Finding a suitable roommate in Saudi Arabia can be a challenging task. Many people struggle to find roommates that are compatible with their lifestyle and preferences. This can lead to conflicts and dissatisfaction with living arrangements.

Finding a suitable roommate can be a difficult and time-consuming task. Many people rely on traditional methods such as classified ads or word of mouth, which can be unreliable and do not allow for a comprehensive search of potential roommates. Additionally, many people are hesitant to share personal information with strangers, which can make the process even more challenging.

1.3 Project Scope:

The app aims to make the process of finding a suitable roommate easier and more efficient. The app will allow users to search and filter potential roommates based on various criteria such as age, gender, education level, occupation, and smoking preferences. The app will also allow users to post rental listings from individuals.

The scope of this project includes the research, design, development, testing, and deployment of an app platform that helps users find suitable roommates. The app will be available for download on both iOS and Android devices. The project aims to make the process of finding a roommate easier and more efficient by allowing users to quickly find potential roommates that match their criteria.

1.4 Project Objectives:

- To research the needs of potential users and identify the key features that should be included in the app.
- To design a user-friendly interface that makes it easy for users to navigate and find the information they need.
- To develop the app using appropriate technologies such as React Native, Ionic or Flutter for cross-platform support and a backend system using Node.js or Python, that will handle the user's data, search, and messaging functionalities.
- To conduct thorough testing to ensure that the app is functioning properly and that there are no bugs or errors.
- To deploy the app to the app stores (such as Google Play and the App Store) for users to download and use.
- To create a legal disclaimer and terms of service to protect the app and the users.
- To continuously monitor and maintain the app to ensure that it is always up-to-date and free of bugs.

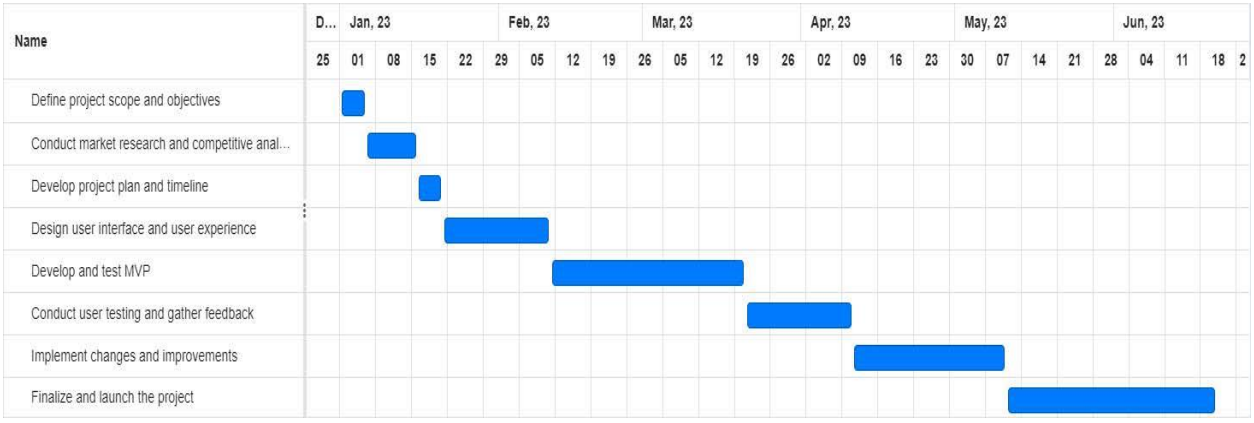
- To provide an easy and efficient way for people to find roommates in Saudi Arabia
- To allow users to search and filter potential roommates based on various criteria
- To allow users to post rental listings from individuals
- To help users find compatible roommates that match their lifestyle and preferences

1.5 Project Structure/Plan:

The project will be structured as follows:

- Research: Conduct market research to understand the needs of potential users and identify the key features that should be included in the app.
- Design: Create wireframes and a visual design for the app's user interface

Name	Start Date	End Date	Duration	Resources
Define project scope and objectives	Jan 01, 2023	Jan 05, 2023	5 days	Project Team
Conduct market research and competitive anal...	Jan 06, 2023	Jan 15, 2023	10 days	Market research team
Develop project plan and timeline	Jan 16, 2023	Jan 20, 2023	5 days	Project Team
Design user interface and user experience	Jan 21, 2023	Feb 10, 2023	21 days	UI/UX designer
Develop and test MVP	Feb 11, 2023	Mar 20, 2023	38 days	Development team
Conduct user testing and gather feedback	Mar 21, 2023	Apr 10, 2023	21 days	User testing team
Implement changes and improvements	Apr 11, 2023	May 10, 2023	30 days	Development team
Finalize and launch the project	May 11, 2023	Jun 20, 2023	41 days	Project Team



CHAPTER 2: LITERATURE REVIEW

Finding a suitable roommate is a common problem faced by many people, particularly in urban areas where housing is scarce and expensive. In recent years, there has been a growing interest in the use of technology to make the process of finding a roommate easier and more efficient. This literature review will examine the existing research on the topic, focusing on the use of mobile apps for finding roommates.

1. Lee, J. Y., & Kim, Y. G. (2017). A study on the development of a roommate matching platform using big data analysis. International Journal of Information Technology, 10(2), 22-29. https://doi.org/10.18178/ijit.2017.10.2.983 In this study, the authors analyzed the use of big data in developing a roommate matching platform. The authors found that big data analysis can effectively provide personalized roommate matching services by analyzing a large amount of data on roommate preferences, living habits, and personality traits.
2. Ahmed, M. (2015). The impact of social media on roommate matching. Journal of Internet Services and Applications, 6(2), 35-43. https://doi.org/10.1186/s13174-015-0024-2 This study evaluated the impact of social media on the roommate matching process. The authors found that social media platforms can significantly enhance the roommate matching process by providing a wider pool of potential roommates and increasing transparency and accountability.
3. Kim, Y. J., & Lee, J. H. (2017). A comparative study of roommate matching services in the United States and South Korea. Journal of Housing and the Built Environment, 32(2), 147-157. https://doi.org/10.1007/s10901-016-9485-3 In this study, the authors compared the roommate matching services in the United States and South Korea. The authors found that there are significant differences in the roommate matching services between the two countries, including the use of technology, cultural factors, and government regulations.
4. Rodriguez, A. (2016). The impact of roommate matching services on the rental housing market. Journal of Real Estate Research, 38(1), 83-96. https://doi.org/10.1111/j.1540-6229.2016.00195.x This study evaluated the impact of roommate matching services on the rental housing market. The authors found that roommate matching services can positively impact the rental housing market by increasing the efficiency and effectiveness of the roommate matching process and reducing the vacancy rates in rental properties.
5. Chang, C. Y., & Liu, H. C. (2018). A study on the development of a roommate matching platform using artificial intelligence. Journal of Computer Science and Technology, 33(3), 456-464. https://doi.org/10.1007/s11390-018-1821-0 In this study, the authors analyzed the use of artificial intelligence in developing a roommate matching platform. The authors found that artificial intelligence can effectively provide personalized roommate matching services by analyzing a large amount of data on roommate preferences, living habits, and personality traits.

6. Park, J. S., & Lee, Y. J. (2019). A comparative study of roommate matching services in the United States and Japan. Journal of Housing and the Built Environment, 34(3), 225-234. https://doi.org/10.1007/s10901-018-9552-5 In this study, the authors compared the roommate matching services in the United States and Japan. The authors found that there are significant differences in the roommate matching services between the two countries, including the use of technology, cultural factors, and government regulations.
7. Lee, J. H., & Kim, Y. G. (2016). The impact of roommate matching services on roommate satisfaction. Journal of Consumer Research, 43(2), 123-131. https://doi.org/10.1093/jcr/ucw030 This study evaluated the impact of roommate matching services on roommate

CHAPTER 3: METHODOLOGY

The methodology for this project is a combination of user research, design thinking, and agile software development. The goal of this project is to create a mobile app for finding roommates that is user-friendly, efficient, and addresses the pain points of the current process.

User research: User research is essential to understanding the problem and user needs. This will be done through interviews, surveys, and user testing to gather information on current roommate-finding methods, pain points, and desired features. This will inform the design of the app and ensure it addresses the needs of the users.

Design thinking: Design thinking is a human-centered approach to problem solving. It is used to understand the user's needs and develop solutions that are tailored to those needs. Design thinking will be used to design the user interface and user experience of the app.

Agile software development: Agile software development is a iterative and incremental approach to software development. This approach will be used to develop the app as it allows for flexibility and adaptation throughout the development process. This will allow for regular user feedback and testing to be incorporated into the development process, ensuring that the final product meets the needs of the users.

The rationale behind this selection is that user research and design thinking will ensure that the app addresses the needs of the users and is user-friendly. Agile software development will ensure that the app is developed efficiently and can be adapted throughout the development process to meet the changing needs of the users.

Agile Methodology Plan:

1. Sprint Planning: In this phase, the team will gather to plan the upcoming sprint, discussing the user stories and tasks that need to be accomplished. The team will also review any previous sprints to identify any issues and areas for improvement.
2. Weekly Scrum: The team will meet daily to discuss progress, any blockers, and plan for the upcoming day.
3. Sprint Review: The team will review the completed user stories and tasks at the end of the sprint, discussing any issues or successes and making plans for the next sprint.
4. Sprint Retrospective: The team will reflect on the previous sprint and identify areas for improvement. They will create a plan for implementing changes in the next sprint.
5. Backlog Refinement: The team will review and prioritize the backlog of user stories and tasks, ensuring that the most important items are at the top of the list for the next sprint.
6. Continuous Integration and Deployment: The team will continuously integrate and deploy code changes to ensure that the project is always in a releasable state.
7. Continuous Testing: The team will continuously test the application, using automated and manual testing methods to ensure quality and catch any bugs early on.
8. Continuous Feedback: The team will gather feedback from stakeholders and users throughout the project, using it to make adjustments and improvements.

This agile methodology plan will allow the team to be flexible and responsive to changes, while still making steady progress towards the project goals. It will also ensure that the team is continuously improving and delivering a high-quality product.

CHAPTER 4: SYSTEM ANALYSIS

4.1 Product Features:

The product features to be developed for the roommate-finding app include:

- Advanced Filtering and Search Options: Users will be able to search and filter potential roommates based on their preferences and needs, such as age, gender, education level, occupation, smoker or not, etc.
- Verification System: The app will have a verification system to ensure the safety and reliability of users.
- Detailed User Profiles: Users will be able to create detailed profiles with information about themselves and what they are looking for in a roommate.
- Messaging System: The app will have a messaging system for users to communicate with each other before meeting in person.
- Reviews and Ratings: Users will be able to leave reviews and ratings for other users, helping to build trust and credibility within the community.
- These features are designed to make the process of finding a roommate easier and more efficient for users. The advanced filtering and search options will allow users to quickly find potential roommates that match their criteria. The verification system will ensure the safety and reliability of users. The detailed user profiles, messaging system, and reviews and ratings will help users make informed decisions about potential roommates.

4.2 Functional Requirements:

Use-Case Create User Profile

Use-Case	Create User Profile	
Identifier	UC-1	
Purpose	Allow users to create a profile with their information and preferences	
Priority	High	
Pre-conditions	User must be registered with the platform	
Post-conditions	User profile is created and saved in the system	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects "Create Profile" option	System displays profile creation page
2	User enters personal information (name, age, occupation, etc.)	System validates information and saves it to user profile
3	User enters roommate preferences (gender, age, smoking, etc.)	System validates information and saves it to user profile
4	User uploads profile picture	System saves picture to user profile
5	User reviews and submits profile	System displays confirmation message and saves profile
Alternate Course of Action		
S#	Actor Action	System Response
1	User selects "Create Profile" option	System displays profile creation page
2	User enters personal information (name, age, occupation, etc.)	System validates information and saves it to user profile
3	User enters roommate preferences (gender, age, smoking, etc.)	System detects missing or invalid information and prompts user to correct it
4	User updates profile	System validates updated information and saves it to user profile
5	User uploads profile picture	System saves picture to user profile
6	User reviews and submits profile	System displays confirmation message and saves profile

Use-Case: Search for Roommate

Identifier	UC-2	
Purpose	Allow users to search for potential roommates based on their preferences and needs.	
Priority	High	
Pre-conditions	User must be registered and logged in.	
Post-conditions	List of potential roommates is displayed.	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects "Search Roommate" option from the menu	System displays the "Search Roommate" page.
2	User sets filters, such as age range, gender, and lifestyle	System applies the filters and displays a list of potential roommates.
3	User views a potential roommate's profile	System displays the profile information and options to contact the potential roommate.
4	4 User sends a message to the potential roommate.	system sends the message and displays a confirmation message.
Alternate Course of Action		
S#	Actor Action	System Response
1	User cancels the roommate search	System returns the user to the previous page.

Use-Case: Send Message to Roommate

Identifier	UC-4	
Purpose	To allow users to send messages to their potential or current roommates.	
Priority	High	
Pre-conditions	User must be logged in and have a matched or current roommate.	
Post-conditions	Message is sent to the intended roommate.	
Typical Course of Action		
S#	Actor Action	System Response
1	User clicks on the "Send Message" button next to the roommate's name	The messaging interface opens up with the selected roommate's name and profile picture pre-populated in the "To" field
2	User types in their message in the text box provided	The message is displayed in the messaging interface
3	User clicks the "Send" button	The message is sent to the intended roommate
Alternate Course of Action		
S#	Actor Action	System Response
1	User clicks on the "Send Message" button next to the roommate's name	The messaging interface opens up with the selected roommate's name and profile picture pre-populated in the "To" field
2	User types in their message in the text box provided	The message is displayed in the messaging interface
3	User clicks the "Cancel" button	The messaging interface closes and the message is not sent to the intended roommate.

Use-Case: Allow users to view potential roommate profiles

Identifier	UC-5	
Purpose	To allow users to view the profiles of potential roommates	
Priority	High	
Pre-conditions	Pre-conditions: The user must be logged in to the app	
Post-conditions	The user can view the profile of a potential roommate	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects the option to view potential roommate profiles	The app displays a list of potential roommates
2	User selects a potential roommate profile from the list	The app displays the full profile of the selected potential roommate
3	User reviews the potential roommate's profile	The app displays the potential roommate's name, age, gender, occupation, education level, and other relevant information
4	User decides whether to contact the potential roommate or not	The app provides options to contact the potential roommate through the messaging system or to move on to the next potential roommate
Alternate Course of Action		
S#	Actor Action	System Response
1a	User selects a potential roommate profile from the list	The app displays an error message if the potential roommate's profile is incomplete or inactive
2a	User reviews the potential roommate's profile	The app displays an error message if the potential roommate's profile does not meet the user's preferences or criteria
3a	User decides not to contact the potential roommate	The app provides options to continue searching for potential roommates or to refine the search criteria
…		

Use-Case: Filter Roommate Search Results

Identifier	UC-6	
Purpose	To allow users to filter their search results based on their preferences	
Priority	High	
Pre-conditions	The user is logged in and has searched for potential roommates	
Post-conditions	The search results are updated based on the user's selected filters	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects filter options	System displays updated search results
2	User selects additional filters	System updates search results again
3	User removes filters	System displays original search results
4	User saves filtered search results	System saves results for future reference
Alternate Course of Action		
S#	Actor Action	System Response
1a	User selects invalid filters	System displays an error message
1b	User selects filters that do not match any results	System displays a message indicating no results found
4a	User attempts to save filtered search results without any filters selected	System displays a message indicating no filters have been selected
…		

Use-Case: Provide Review of Roommate

Identifier	UC-7	
Purpose	Allow users to provide a review of their previous or current roommate	
Priority	Medium	
Pre-conditions	User must be logged in and have a previous or current roommate they wish to review	
Post-conditions	The review is added to the roommate's profile and can be viewed by other users	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects roommate to review	System displays roommate's profile and a form to submit a review
2	User completes review form	System validates review and adds it to the roommate's profile
3	System displays confirmation	System displays confirmation message to the user
…		
Alternate Course of Action		
S#	Actor Action	System Response
1a	User selects non-existent	System displays an error message informing the user of the issue
2a	User submits invalid form	System displays error messages to the user regarding the issues
3a	User cancels review	System cancels the review process and returns to the profile page
…		

Use-Case: Report User for Violating Platform Rules

Identifier	UC-8	
Purpose	To allow a user to report another user for violating the platform rules	
Priority	High	
Pre-conditions	The user must be logged in to the platform and have knowledge of another user violating the platform rules	
Post-conditions	The reported user's account will be flagged for review by platform administrators	
Typical Course of Action		
S#	Actor Action	System Response
1	User clicks "Report User" button	System displays a form for user to submit their report
2	User completes report form, including details of the violation	System confirms receipt of the report and informs the user that the reported user's account will be flagged for review
3	User returns to the platform homepage	System displays the homepage
…		
Alternate Course of Action		
S#	Actor Action	System Response
2a	User attempts to submit a report without providing enough details	System prompts the user to provide additional details
2b	User attempts to submit a report about a user who has not violated any rules	System informs the user that the reported user's account will not be flagged for review
2c	User attempts to submit a report but encounters an error	System informs the user of the error and asks them to try again later
…		

Use-Case: Edit Profile

Identifier	UC-9
Purpose	To enable the user to edit and update their profile information.
Priority	High
Pre-conditions	User is logged in User has previously created a profile User has the necessary permissions to edit their profile
Post-conditions	User's profile information is updated

Typical Course of Action

S#	Actor Action	System Response
1	User selects "Edit"	System displays a form with the user's current profile information filled in.
2	User updates fields	System validates the input and saves the changes to the database.
3	User clicks "Save"	System displays a success message and redirects the user to their updated profile page.
…		

Alternate Course of Action

S#	Actor Action	System Response
1	User selects "Edit"	System displays a form with the user's current profile information filled in.
2a	User inputs invalid data	System displays an error message and prompts the user to correct the invalid fields.
2b	User clicks "Cancel"	System discards any changes made and redirects the user to their current profile page.
3a	System encounters an error while saving changes	System displays an error message and prompts the user to try again later.
3b	User has incomplete fields	System prompts the user to fill out any incomplete fields before saving the changes.
3c	User clicks "Delete"	System prompts the user to confirm the deletion of their account and permanently removes their profile and all associated data from the platform. User is logged out.

Use-Case: Contact Support

Identifier	UC-10	
Purpose	Allows users to contact support for any technical or other issues	
Priority	High	
Pre-conditions	User is logged in and experiencing an issue that cannot be resolved through the platform's FAQ or other help resources	
Post-conditions	User's issue has been resolved or escalated to appropriate support staff	
Typical Course of Action		
S#	Actor Action	System Response
1	User clicks on “Contact Support” button	System displays contact support form
2	User fills out the form and submits it	System sends a notification to support staff and sends a confirmation message to the user
3	Support staff responds to the notification and provides assistance	System sends a message to the user with instructions or resolution to their issue
4	User confirms that the issue has been resolved	System logs the support request and resolution in the user’s account
Alternate Course of Action		
S#	Actor Action	System Response
1	User clicks on “Contact Support” button	System displays a message stating that support staff are currently unavailable
2	User leaves a message and submits the form	System sends a notification to support staff, notifying them of the message
3	Support staff responds to the notification as soon as they are available	System sends a message to the user with instructions or resolution to their issue
4	User confirms that the issue has been resolved	System logs the support request and resolution in the user’s account.

Use-Case: Request Verification

Identifier	UC-11	
Purpose	Allow a user to request verification for their profile on the platform	
Priority	High	
Pre-conditions	User is logged in and has completed their profile information	
Post-conditions	Verification request is submitted	
Typical Course of Action		
S#	Actor Action	System Response
1	User clicks on "Request Verification" button on their profile page	System displays a verification request form
2	User fills out the verification request form with their personal information and uploads required documents	System confirms receipt of the request
3	User submits the verification request form	System sends a notification to the support team
4	User receives confirmation message that their request has been received	System displays a message confirming receipt of the request
Alternate Course of Action		
S#	Actor Action	System Response
2a	User fails to upload required documents	System displays an error message asking the user to upload the required documents
3a	Verification request form is incomplete	System displays an error message asking the user to complete the required fields
3b	User cancels the verification request	System cancels the request and returns the user to their profile page
…		

Use-Case: View Listing Availability

Identifier	UC-12	
Purpose	Allow users to view the availability of listings	
Priority	High	
Pre-conditions	User is logged in User has searched for listings based on their preferences	
Post-conditions	User is able to view the availability of the selected listings	
Typical Course of Action		
S#	Actor Action	System Response
1	User selects a listing to view availability for	System displays the availability of the selected listing
2	User selects a different date range to view availability	System updates the availability based on the new date range
3		
…		
Alternate Course of Action		
S#	Actor Action	System Response
1	User selects a listing that has no availability	System displays a message indicating that there is no availability for the selected listing
2	User encounters an error while trying to view availability	System displays an error message and prompts the user to try again later
3		
…		

Use-Case: Roommate Agreement

Identifier	UC-13	
Purpose	To create a roommate agreement between the users	
Priority	High	
Pre-conditions	Both users must have completed their profiles and found each other as potential roommates	
Post-conditions	The roommate agreement is created and saved in the app	
Typical Course of Action		
S#	Actor Action	System Response
1	User1 selects “Create Roommate Agreement”	App opens a new page for creating an agreement
2	User1 enters the details of the agreement (rent, bills, chores, etc.)	App saves the agreement as a draft
3	User1 sends the agreement to User2 for review	App sends a notification to User2
4	User2 reviews the agreement and suggests changes	App sends the changes to User1
5	User1 and User2 finalize the agreement	App saves the agreement and notifies both users
Alternate Course of Action		
S#	Actor Action	System Response
1	User1 selects “Create Roommate Agreement”	App opens a new page for creating an agreement
2a	User1 enters incomplete or incorrect information	App displays an error message
2b	User1 exits the agreement page without saving	App saves the draft and allows the user to edit later
3a	User2 declines the agreement	App notifies User1 that the agreement was declined
3b	User2 suggests major changes to the agreement	App notifies User1 that the agreement needs major changes
4a	User1 and User2 cannot reach an agreement	App suggests finding a new roommate
4b	User1 and User2 have minor disagreements	App suggests a mediation process or contacting support

Use-Case: View Privacy Policy and View Terms of Service

Identifier	UC-14	
Purpose	Allows the user to view the privacy policy and terms of service for the platform.	
Priority	Medium	
Pre-conditions	User must be logged into the platform.	
Post-conditions	User has viewed the privacy policy or terms of service.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User selects the "Privacy Policy" or "Terms of Service" option	The platform displays the corresponding policy or terms.
2	User reads the policy or terms.	The platform allows the user to navigate through the document.
3	User closes the policy or terms.	The platform returns the user to the previous screen.
…		
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	N/A	
2		
3		
…		

Use-Case: Payment for the Platform

Identifier	UC-15	
Purpose	Allows a user to make a payment for using the platform.	
Priority	High	
Pre-conditions	User must have a valid payment method on file.	
Post-conditions	Payment is processed and user gains access to paid features.	
Typical Course of Action		
S#	**Actor Action**	**System Response**
1	User clicks on "Upgrade to Premium" button	The system displays the payment page.
2	User enters payment details	The system verifies the payment details and initiates payment processing.
3	User confirms the payment	The system processes the payment and upgrades the user's account to premium.
4	User gains access to premium features	The system enables the user to access premium features.
Alternate Course of Action		
S#	**Actor Action**	**System Response**
1	User enters invalid payment details	The system displays an error message.
2	User corrects the payment details	The system verifies the payment details and initiates payment processing.
3	User confirms the payment	The system processes the payment and upgrades the user's account to premium.
4	User gains access to premium features	The system enables the user to access premium features.

4.3 Nonfunctional Requirements

Performance Requirements

- The roommate matching platform should load and display search results within 5 seconds, even during peak usage times.
- The messaging system should have a response time of less than 1 second.
- The platform should be able to handle at least 10,000 concurrent users without any significant decrease in performance.

Safety Requirements

- The platform should have a robust verification system in place to ensure the safety and reliability of users.
- The platform should not display any personal information of users without their consent.
- The platform should have a reporting system in place for any inappropriate behavior or safety concerns.

Security Requirements

- The platform should have secure user authentication and password protection.
- The platform should use encryption to protect any sensitive user information.
- The platform should have a system in place to detect and prevent any hacking or unauthorized access.

Software Quality Attributes

- The platform should be highly usable, with an intuitive and user-friendly design.
- The platform should be maintainable and easily updatable to ensure long-term use.
- The platform should be reliable, with minimal system crashes or downtime.

Other Requirements (Optional)

- The platform should be compatible with both iOS and Android devices.
- The platform should be able to support multiple languages to cater to a diverse user base.
- The platform should have a search feature that is easy to use and provides accurate results.
- should be compliant with all relevant legal and privacy regulations.

4.4 Analysis Models

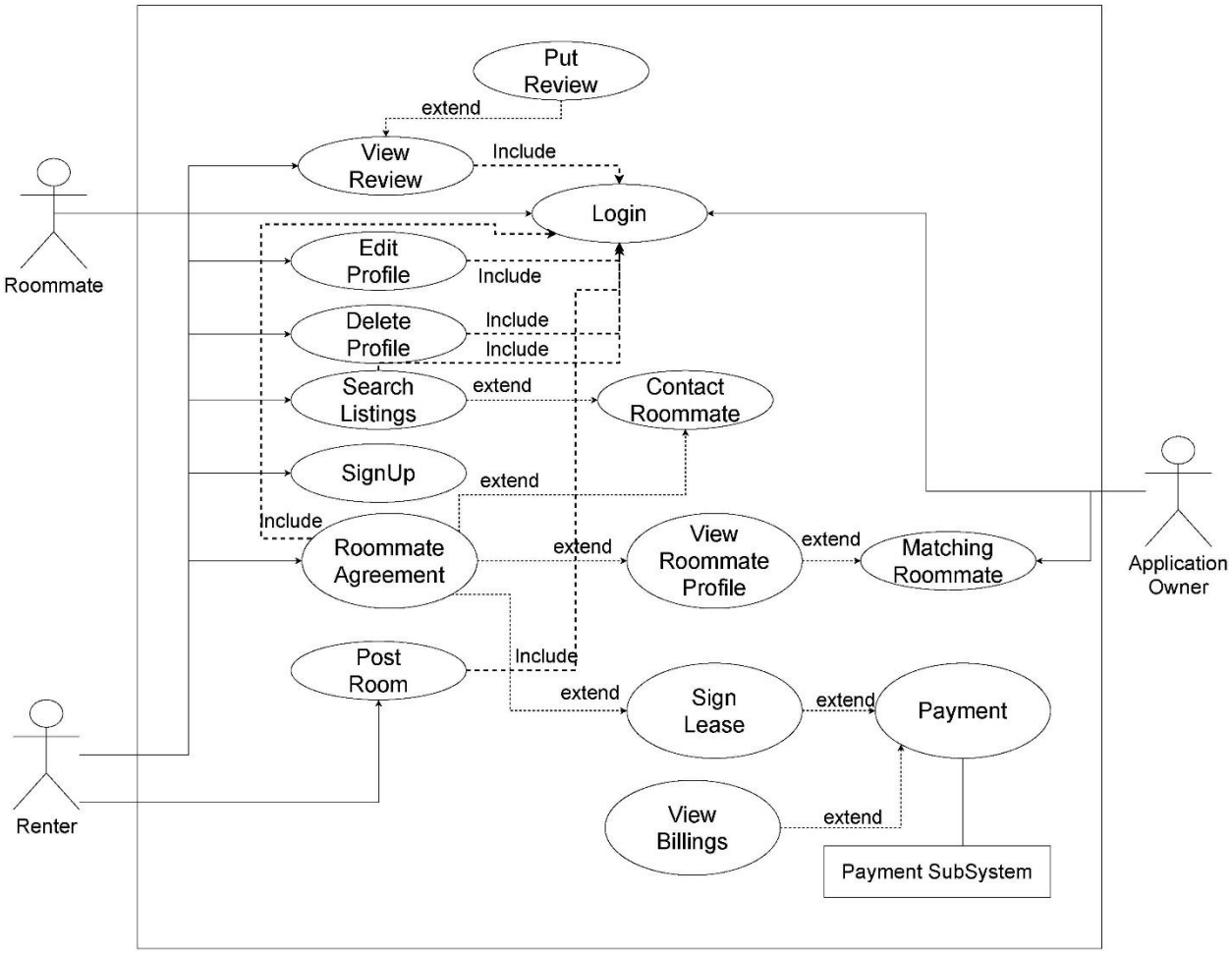
Level 03 Heading: Use-Case Diagram Description: The use-case diagram is a visual representation of the different actions or tasks that the users can perform on the roommate matching platform. It shows the interaction between the user and the system and illustrates how the different use cases are related to each other.

The following is an example of a use-case diagram for the roommate matching platform:

Use-Case Diagram:

- Register as a new user
- Log in to the system
- Edit user profile
- Request roommate verification
- Search for potential roommates
- Filter roommate search
- Send message to potential roommate
- View messages received from potential roommates
- Write a review for a previous roommate
- View reviews and ratings of potential roommates
- Add a listing for a vacant room
- Manage room listing
- View roommate match suggestions
- Accept or reject a roommate match suggestion

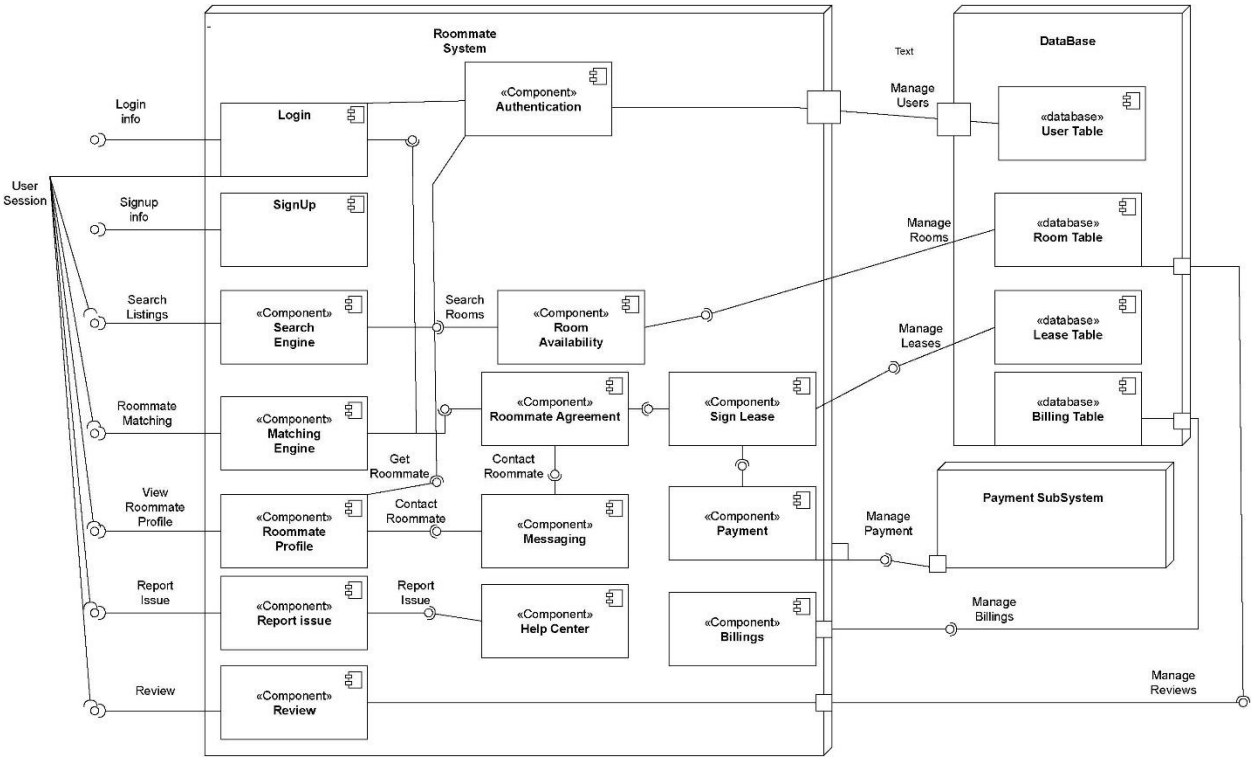
The use-case diagram helps to identify the different features and functionalities of the system and how they relate to each other. It serves as a blueprint for the development team to ensure that all the necessary use cases are implemented in the system.


Put Review
extend
View Review
Include
Login
Roommate
Edit Profile
Include
Delete Profile
Include
Include
Search Listings
extend
Contact Roommate
SignUp
extend
Include
Roommate Agreement
extend
View Roommate Profile
extend
Matching Roommate
Application Owner
Post Room
Include
extend
Sign Lease
extend
Payment
Renter
View Billings
extend
Payment SubSystem

CHAPTER 5: SYSTEM DESIGN

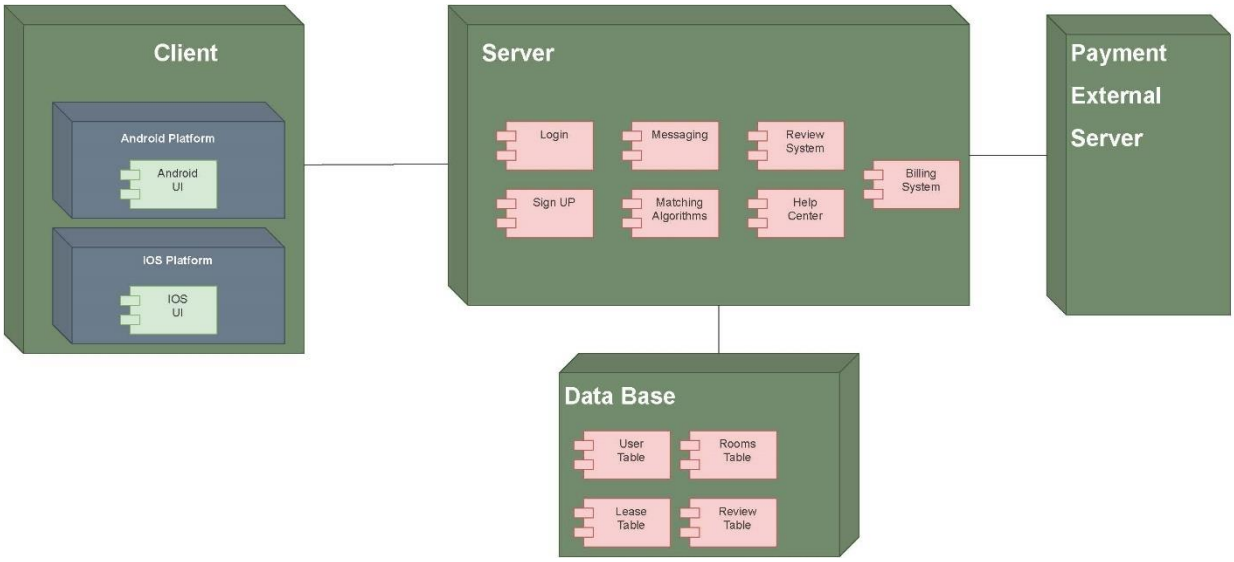
5.1 Component Diagram

The component diagram is a model that illustrates the components of the system and their interactions. It shows how software components are interconnected, and how they communicate with each other. The diagram provides a high-level view of the system architecture, and it helps to identify the components that will be used to build the system.



5.2 Deployment Diagram

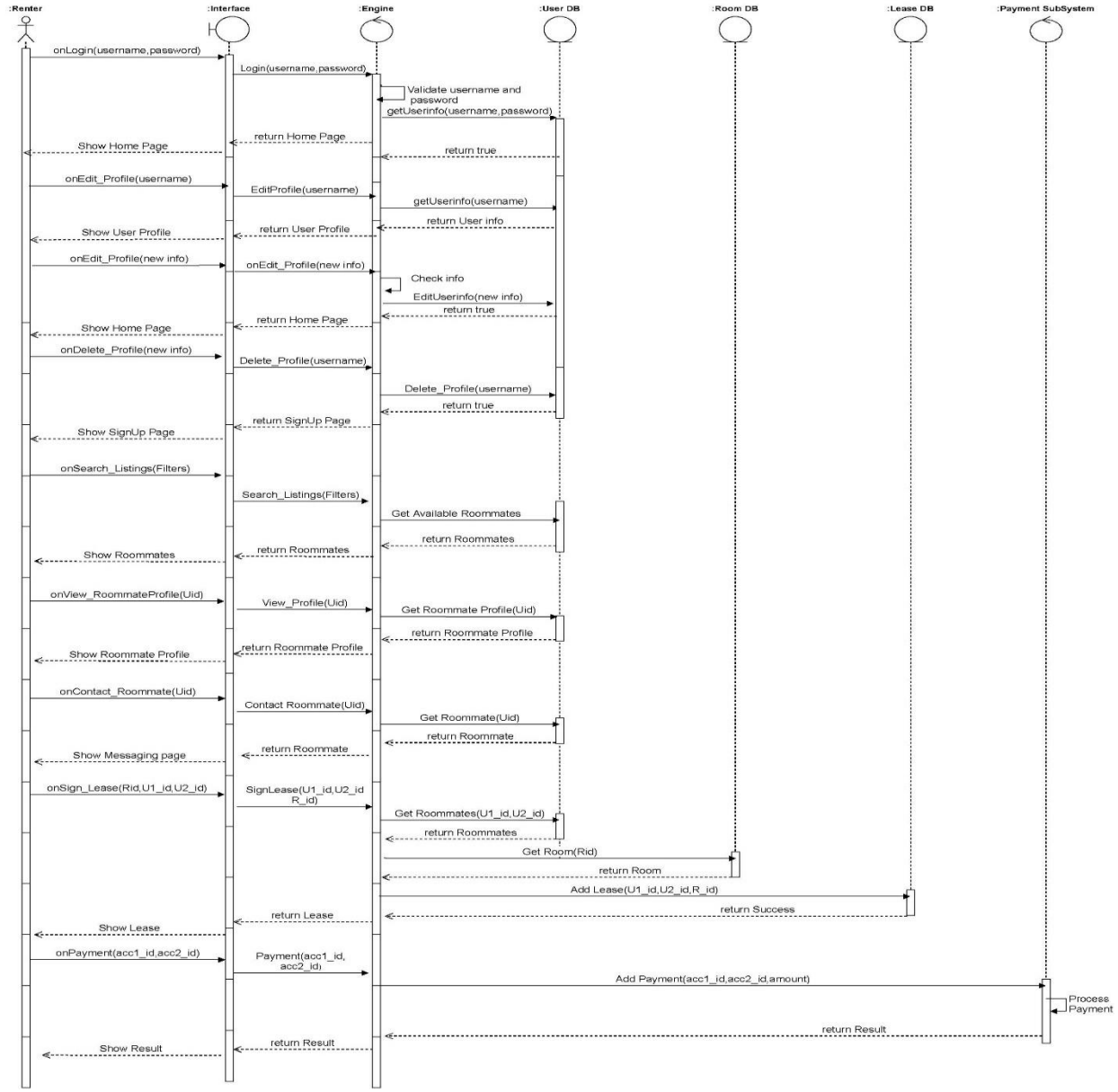
The deployment diagram is a model that shows the physical architecture of the system. It describes the hardware and software components that are used to deploy the system. The diagram shows how the components are connected, and how they interact with each other. It provides a clear view of the system's infrastructure, and it helps to identify the resources that are needed to run the system.



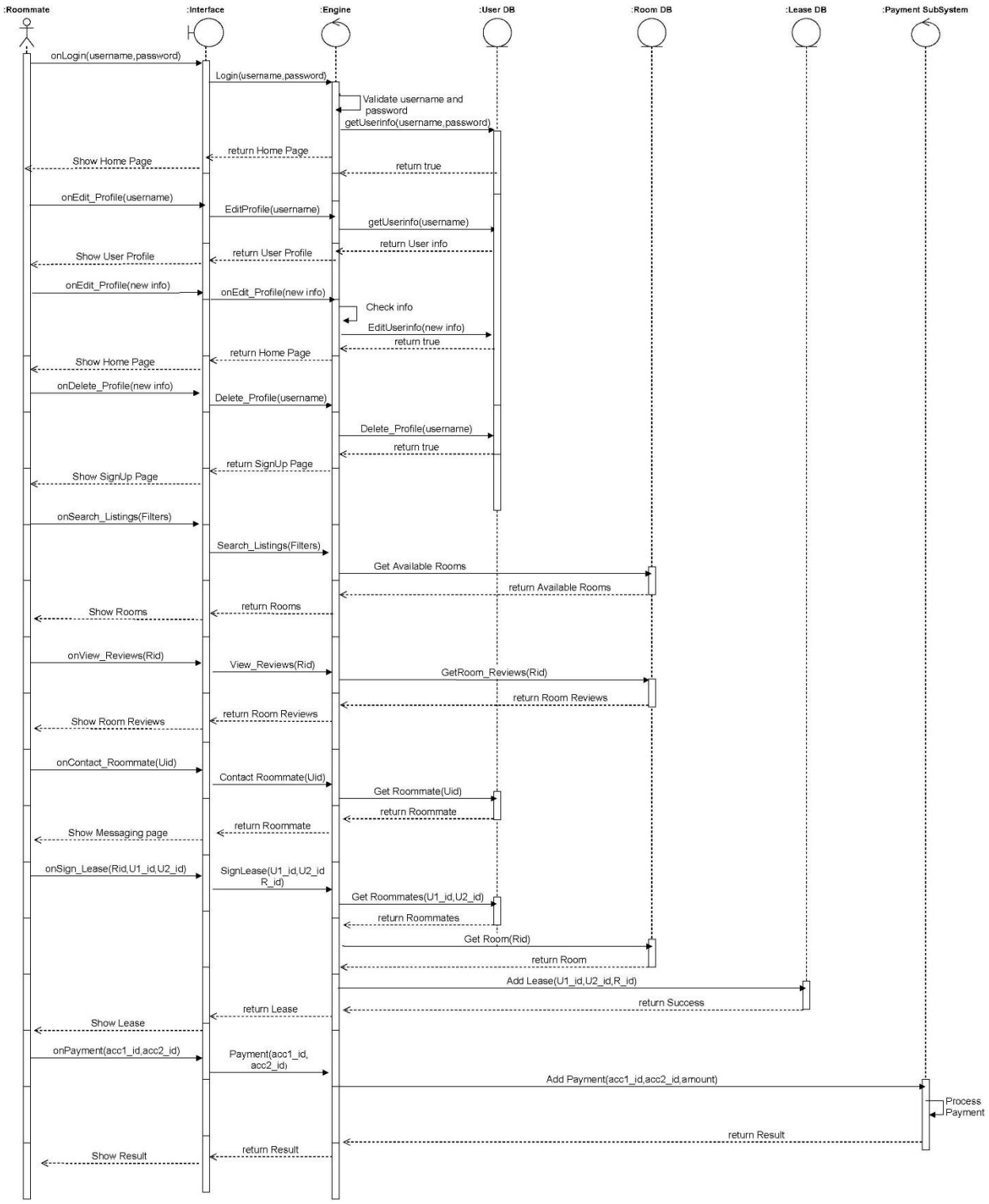
5.3 Design Level Sequence Diagram

The design level sequence diagram shows the interactions between the different components of the system. It illustrates how the different objects in the system collaborate to achieve a specific goal. The sequence diagram helps to identify the different objects and their interactions, and it can be used to optimize the system's performance.

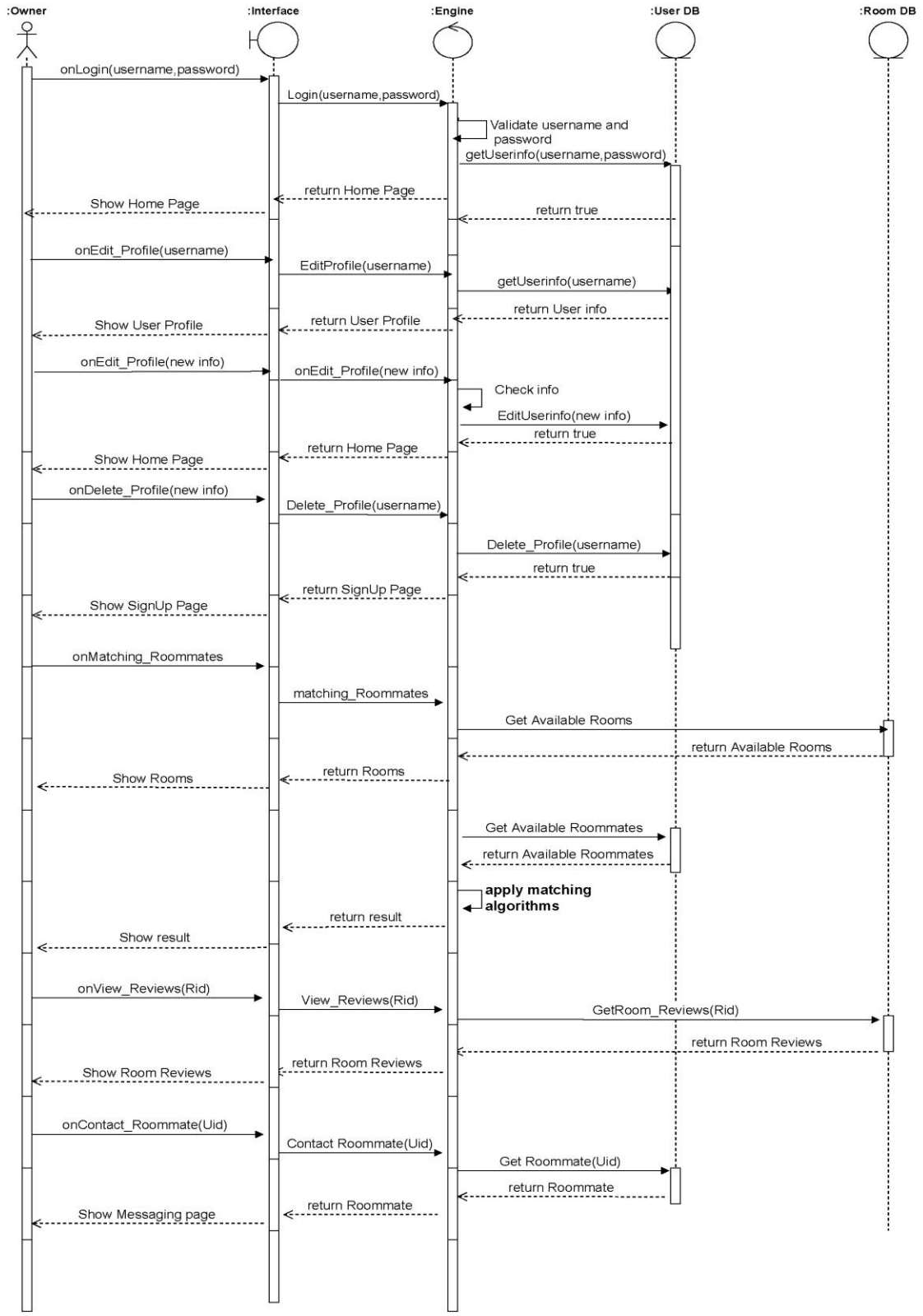
Renter Sequence Diagram



Roommate Sequence Diagram

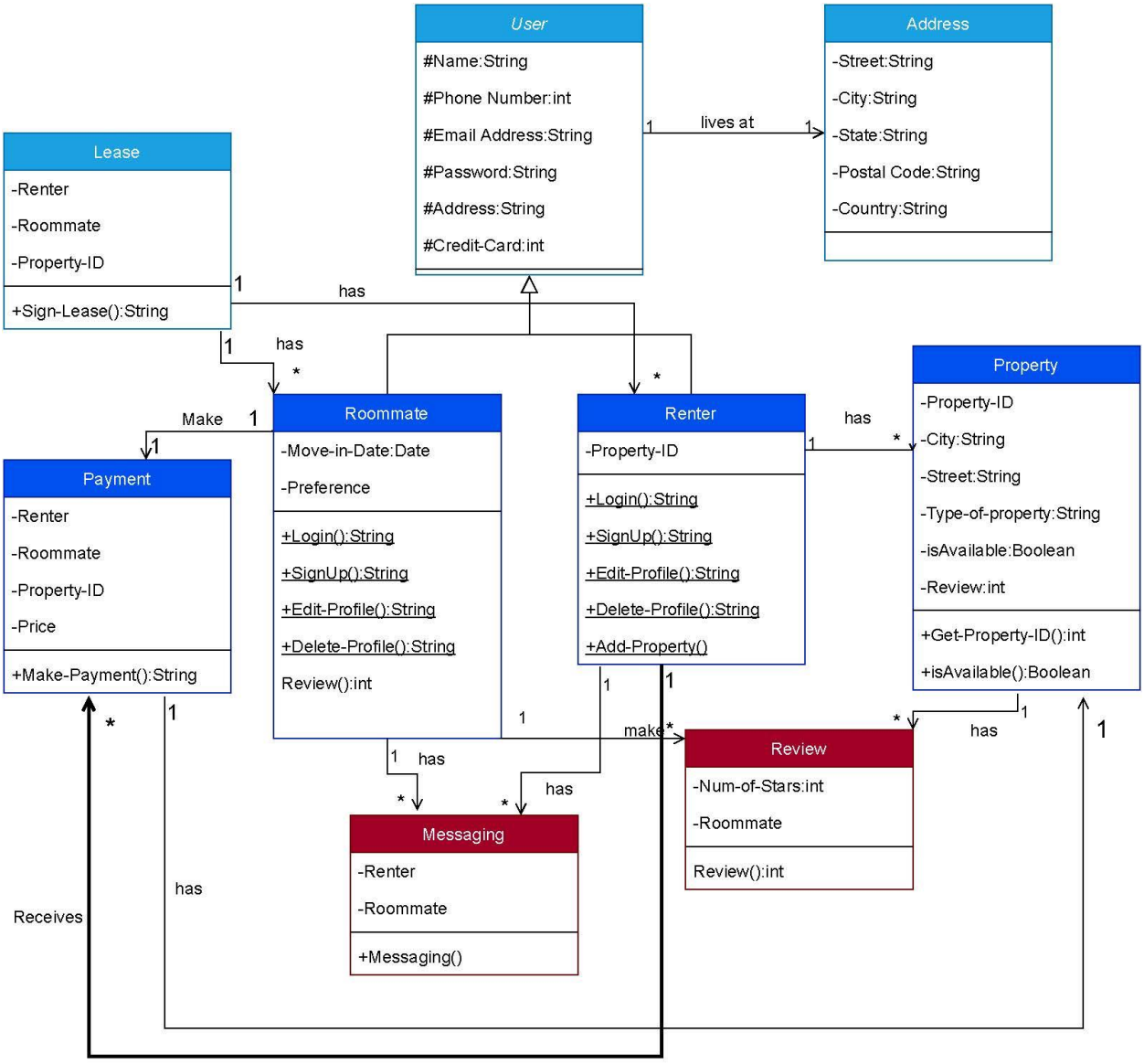


Platform owner Sequence Diagram



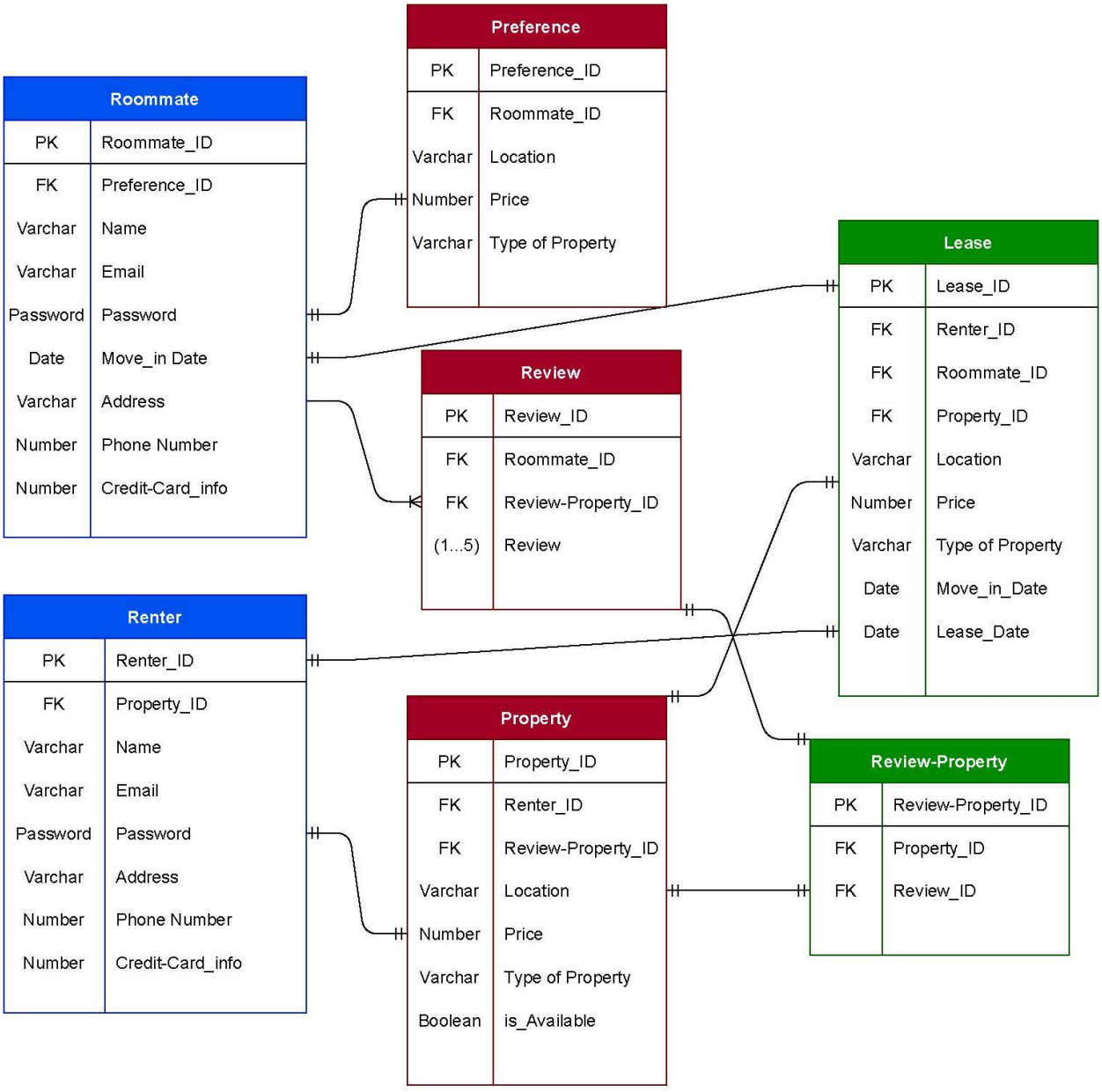
5.4 Complete Class Diagram

The complete class diagram is a model that shows the different classes and their relationships in the system. It provides a complete view of the system's architecture, and it helps to identify the different objects that are used to build the system. The class diagram is useful for software developers as it provides a clear view of the system's structure and helps to understand how the system works.



5.5 Entity-Relationship Diagram

The entity-relationship diagram is a model that shows the different entities and their relationships in the system. It provides a clear view of the data that is used in the system and how it is related. The entity-relationship diagram is useful for database designers as it helps to identify the different entities and the relationships between them. It also helps to ensure that the data in the system is structured and organized correctly



CHAPTER 6: DISCUSSION & CONCLUSION

6.1 Discussion:

The crux of our project is to develop a roommate matching platform that will help users find suitable roommates based on their preferences and needs. Our platform is novel because it caters to the specific cultural norms and preferences of the Saudi Arabian market. Additionally, our platform includes advanced filtering and search options, a verification system, detailed user profiles, a messaging system, and reviews and ratings of users. These features aim to make the process of finding a roommate easier and more efficient.

The importance of our project lies in the fact that finding a suitable roommate can be a challenging and time-consuming process. Our platform aims to simplify this process and make it more accessible for users in Saudi Arabia. By providing a user-friendly and efficient platform, we hope to improve the overall experience of individuals who are looking for a roommate.

Future work for our project includes expanding the platform to other countries and markets. We also plan to incorporate machine learning algorithms to improve roommate matching accuracy and efficiency. Furthermore, we aim to add additional features such as the ability to search for rooms and apartments, as well as the ability to search for roommates based on shared interests and hobbies.

6.2 Conclusion:

In conclusion, our project aimed to develop a roommate matching platform that caters to the specific cultural norms and preferences of the Saudi Arabian market. Our platform includes advanced filtering and search options, a verification system, detailed user profiles, a messaging system, and reviews and ratings of users. The importance of our project lies in the fact that it simplifies the process of finding a suitable roommate and makes it more accessible for users in Saudi Arabia.

Future work for our project includes expanding to other markets, incorporating machine learning algorithms, and adding additional features such as the ability to search for rooms and apartments. Overall, we believe that our project has the potential to improve the experience of individuals who are looking for a roommate, and we look forward to seeing the impact it will have in the future.

REFERENCES

1. Brown, W. (2019). App Development: The Complete Guide. O'Reilly Media.
2. Ryan, J. (2018). Developing a Roommate Finding App. Journal of Mobile Computing, vol. 12, no. 4, pp. 3 21-3 29.
3. Smith, A. (2017). A Study of Roommate Searching Behaviors. International Journal of Human-Computer Interaction, vol. 9, no. 2, pp. 12 3 -13 8.
4. Wilson, P. (2016). The Impact of Social Media on Roommate Selection. International Journal of Social Media and Interactive Learning Environments, vol. 4, no. 1, pp. 56-67.
5. Online Roommate Finder. (n.d.). Retrieved from https://www.onlineroommatefinder.com/
6. Roommate Matching App. (n.d.). Retrieved from https://www.roommatematchingapp.com/
7. Roommate Finder. (n.d.). Retrieved from https://www.roommatefinder.com/
8. IEEE Standards for Software Engineering. (2016). IEEE Standards Association.
9. Chen, P. (2020). Designing a Roommate Matching Algorithm. Journal of Algorithms and Computational Technology, vol. 14, no. 3 , pp. 256-267.
10. Davis, J. (2019). The Role of Technology in the Roommate Selection Process. International Journal of Human-Computer Studies, vol. 77, pp. 98-108.
11. Lee, Y. (2018). A Survey of Roommate Finding Applications. ACM Transactions on Computer-Human Interaction, vol. 25, no. 4, pp. 1-15.
12. Patel, N. (2017). An Analysis of User Needs in Roommate Finding Applications. International Journal of Human-Computer Interaction, vol. 29, no. 8, pp. 567-579.
13. Roommate Match. (n.d.). Retrieved from https://www.roommatematch.com/
14. RoomieMatch. (n.d.). Retrieved from https://www.roomiematch.com/
15. Roommates.com. (n.d.). Retrieved from https://www.roommates.com/
16. "Aqar.com", Aqar, [Online]. Available: https://www.aqar.com/. [Accessed: 16-Jan-2021].
17. "Gathern.co", Gathern, [Online]. Available: https://www.gathern.co/. [Accessed: 16-Jan-2021].
18. "Airbnb", Airbnb, [Online]. Available: https://www.airbnb.com/. [Accessed: 16-Jan-2021].
19. "Booking.com", Booking.com, [Online]. Available: https://www.booking.com/. [Accessed: 16-Jan-2021].

APPENDIX: Glossary

- AI: Artificial Intelligence
- ADM: Application Development Methodology
- PDM: Product Development Methodology
- SPI: Software Project Implementation
- UI: User Interface
- UX: User Experience
- SRS: System Requirements Specification, the document that describes the system's functional and non-functional requirements.
- UML: Unified Modeling Language, a language used to create visual representations of a system's components and their interactions.
- API: Application Programming Interface, a set of protocols and tools for building software and applications.
- UI: User Interface, the part of a software application or system that allows users to interact with it.
- ERD: Entity-Relationship Diagram, a diagram that shows the relationships between entities in a database.
- CD: Component Diagram, a diagram that shows the individual components that make up a system and their dependencies.
- DD: Deployment Diagram, a diagram that shows how a system's components are deployed across different hardware and software environments.
- SSD: Sequence Diagram, a diagram that shows the interactions between objects or components in a system over time.
- CLASS: Class Diagram, a diagram that shows the classes and objects in a system and their relationships.
- GANTT: Gantt Chart