Palestine Technical University (Kadoorie)

Faculty of Engineering and Technology

Computer Systems Engineering

Course name:

Software Engineering

Project title:

University Housing Management Software

By:

Akram Sous	202110198
Islam Mosmar	202110106
Osama Draidi	202110173
Thamer Draidi	202110049

Supervisor:

Dr.Osama Hamed

Tulkarm, Palestine

2024/5/27

Acknowledgement

Praise be to Allah, and prayer and peace be upon the Messenger of Allah -may Allah's peace and blessings be upon him -Sayyid Al-Anam. Despite the siege, the occupation and the difficult circumstances surrounding us, innovation and challenge have continued and will continue in our schools and universities, foremost among them the Palestinian Technical University-Kadoorie. This warm lap has ensured for years that we have faced many difficulties, but in the end, we will reap the fruits of tiredness and staying up late. This is our modest work, which finally sees the light. We cannot help but express our sincere thanks to our dear university, represented by the distinguished administrative board and its professors in charge of its affairs. Special in appreciation and gratitude to dear Doctor: Osama Hamed who has provided us with assistance and guidance. His continued dedication and effort have had a positive and effective impact. He has deep thanks and gratitude for his preference and supervision of this research.

Abstract:

The University Housing Management Software is a comprehensive project aimed at simplifying and streamlining the housing process for university students. This software is designed to address the various challenges faced by students when searching for university housing, such as availability, affordability, location, and amenities. The university housing management system integrates various core functionalities including online room application and selection, automated room assignment, billing and payment processing, and robust communication tools for both students and housing staff. Utilizing modern web technologies, the system ensures a user-friendly interface and seamless access across multiple devices. The findings highlight the critical role of technology in enhancing university housing services, suggesting that such systems can significantly reduce manual workload, improve response times for maintenance issues, and foster better communication between students and housing administrators.

Table of contents

Chapter 1: Introduction

1.1 Problem Statements:

Nowadays, a majority of students in Tulkarm suffering from the lack of finding a place to settle in while they are in the university lifetime, this problem faces the students that came from other cities in the Westbank, as we all know the distances between the cities in Palestine is not too large, but because of the occupation obstacles and the probability of getting harmed by settlers, also this problem became more efficient because our university receives an ascending number of students from other cities than Tulkarm.

1.2 Aims and objectives:

This project aims to make the university housing process more efficient, transparent, and user-friendly for students. By leveraging technology, we can create a better housing experience for university students. This software project is not just a tool, but a step towards a more organized and stress-free university life.

Also, this project aims to organize the renting process for the students in our university with the owners of the rental places and to reduce the human efforts.

Maybe this project will be developed in the future to include a broader category.

1.3 Key Features:

1. **Housing Database**: A comprehensive database of all available university housing options, including on-campus dormitories, off-campus apartments, and shared housing options.
2. **Search and Filter**: Students can search for housing based on various parameters such as location, price, type of accommodation, and more. The advanced filter options allow students to narrow down their choices and find the perfect housing option that suits their needs.
3. **Booking and Payment**: The software includes a secure booking and payment system. Students can book their chosen housing and make payments directly through the platform.
4. **Reviews and Ratings**: Students can leave reviews and ratings for their housing, providing valuable feedback for future students and helping them make informed decisions.

5. **Maintenance Requests**: A feature for students to report any issues or maintenance requests for their housing. These requests are sent directly to the university's maintenance department for quick resolution.
6. **Notifications**: Students receive notifications about housing deadlines, payment due dates, and any updates or changes to their housing situation.

Chapter 2: Requirements Gathering Approach

Regarding the approach that was followed in collecting the user and system requirements, more than one approach was taken due to what was mentioned that none of the current team members had rent apartments before. The most important method that was followed is to divide the tasks among the members of the team so that it pervades Interest and proper discussion to obtain the best requirements to system.

The first thing that was started is the idea of our project, which is to meet the users. In the beginning they were university students, and then we expanded to include everyone by asking them what they need in this system and the most important things that anyone would need when he want for rent an apartment, and we consulted With some of us until we got all the requirements for the users, including what has a higher priority than the others, therefore some requirements were deprecated, and some of them were also inconsistent with other requirements that were deprecated because their priority is less.

Then we go to meet the second user of the system, the owners of apartments, and we asked them about the best way through the system that we can provide for them in order to facilitate their interests and has an easy system for them to use because different age user have different ideas will use the system for both the tenant and the owner, and then their requirements were written after checked that doesn't have any inconsistent.

And thirdly, we went to meet some people to ask them about what is related to renting apartments, such as accountants and lawyers, with regard to the system so that it is complete and perfect, but after going to the owners, some of them saw that making all these matters to control money and payment on the site is not safe, and some of them opposed the idea of using it if everything was done on the system, so we decided that at the present time the financial matters will not be implemented and will be considered for the development of the project in the future (after we get full confidence from all owners).

At the end of the meetings with the users is that we have reviewed all the requirements and made sure that they are what they want and make sure that they are not inconsistent and that they are clear, understandable and comprehensive.

Then we decided to go to the web pages and visit some sites such as the Booking.com page which has been greatly helpful in the topic of the presentation and the most important things that must be provided, to ensure that system as much as will get excellent feedback is obtained from the users.

And the over Goal for our team that the system is starting here in Palestine and the End in the whole world.

Chapter 3: User requirements definition

3.1 Functional Requirements

1) The system will have Log in page and be password-protected. The system will be a multi-user system where every user must log in (but for user who wants to reviews system only he can pass this without make an account).

2) To make it easier for users, they can search for the city they want by writing the city name.

3) An option is added in the login menu with the ability to know the password in case you forget it that a message containing the password is sent to the e-mail entered by the user in his account on the system.

4) For the visitor user and user has account:

a. Allowing the user, in the event of viewing the contents of the system without doing anything, to see all apartments and filter them in the required manner.

b. In the event that the user wants to make a request for a specific apartment or make a group of preferred apartments in order to compare them, in this case the user is requested to log in.

c. In the case of a login request, the user must have an account, and if he does not have an account, he can easily create a new account.

d. For the user who has an account, he can make a list of favorite apartments in order to compare them and be able to apply for the apartment he needs.

5) For the client:

a. An option is provided for him in order to give an appraisal of the apartment.

b. A list is provided for him regarding the rented apartment and displaying all the rental details of the period and amount.

c. Providing the application renewal service in case the rental time is nearing the end.

6) For the owner:

a. Allow owner to add new apartment and delete non-rental apartment, accept or refuse a request from a tenant and modify details of the non-rental apartment because it’s not acceptable to do that for renting an apartment because that not accepted according to the law.

b. Allow the owner to get data about everything at any time (how many apartments he added, how many of these are rented, for whom, and the time).

c. He has a page for the offers made by users for the non-rented apartments and through it he can obtain the applicant's information, including also his phone number, which will be a requirement. The user can only be allowed to place it and he can accept or reject the request.

d. The owner is prevented from deleting the rented apartment and modifying it, for example, such as changing the rent.

7) For Housing Apartment:

a. Each apartment is provided with its own appraisal amount.

b. Full information of the apartment is available from the location, number of rooms, the amount of rent and all the details requested from the users.

c. The 3 most recent apartments that have been added and are still not rented are added to the system's home page.

3.2 Non-Functional Requirements

1) Requirements on usability, reliability, performance, supportability, security, recovery, interface, implementation, operation, and portability.

2) The system will be a screen-based application with interface that help reach to anything in easy way.

3) The GUI chosen should be simple and effective such that any user will be able to navigate his way through the application easily. Since the age groups used for the systems are different, the site was so easy to use even an older person does not need help most of the time. (usability)

4) The administrator is responsible for the user’s accessibility into the database. Password verification is used to ensure that there is no manipulation of the system. (security)

5) The Backup will be daily to prevent any data lost or when system is break down (reliability)

6) Support will be provided by team members 24 hours a day so that a number will be added to contact him in the event of any defect in the system or inquiries about something specific. (supportability)

7) Care is taken to use the latest programming languages such as React and Java Script, in order to ensure the speed of execution of programming sentences and their abbreviations in order to ensure the speed of the site. (speed)

The system is designed to work on the web, the browser is through the Internet, thus it will work on various systems such as Android, Windows, Linux, etc., as a result of this, the current system will not require any costs to transfer it from one work environment to another work environment. (Probability)

Chapter 4: System architecture

Understanding system architecture is crucial because it lays the foundation for how the system will operate, including aspects like performance, fault tolerance, scalability, and reliability.

When selecting an architecture, it's essential to consider how it will manage these factors. Figures 1 and 2 illustrate this process by providing an initial overview of a system designed to link its users with a stored database.

The key functionalities of the system, as shown in Figure 1, include adding, deleting, and modifying entries in the database. These actions are performed directly on the database. Additionally, the system is designed with security in mind, ensuring that user access and visibility are managed according to assigned permissions.

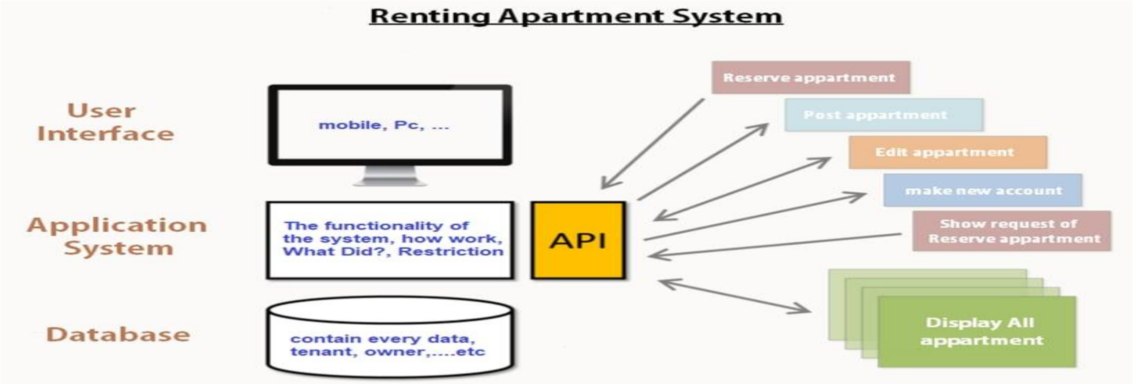


Figure 4.1: First System architecture for Renting Apartment System

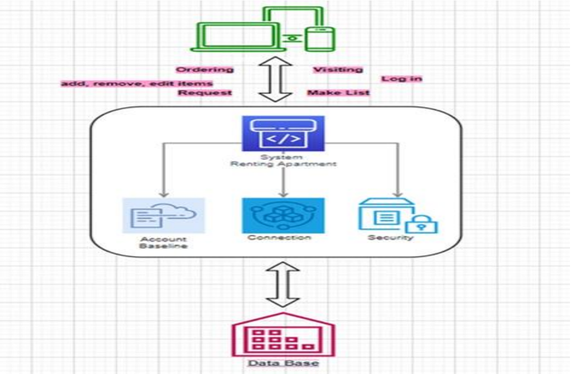


Figure 4.2: Second System architecture for Renting Apartment System

Chapter 5: System requirements specification

For Usability: the system provides a help and support menu in all interfaces for the user to interact with the system. The user can use the system by reading help and support.

For Security: the system provides username and password to prevent the system from unauthorized.

For Availability: the system should always be available for access at 24 hours, 7 days a week. Also, in the occurrence of any major system malfunctioning, the system should be available in 1 to 2working days, so that business process is not severely affected.

For Error handling: Error should be considerably minimized and an appropriate error message that guides the user to recover from an error should be provided. Validation of user's input is highly essential.

For Ease of use: Considered the level of knowledge possessed by the different ages users of this system, a simple but quality user interface should be developed to make it easy to understand and required less training.

Chapter 6: System models

6.1 Use-case diagram

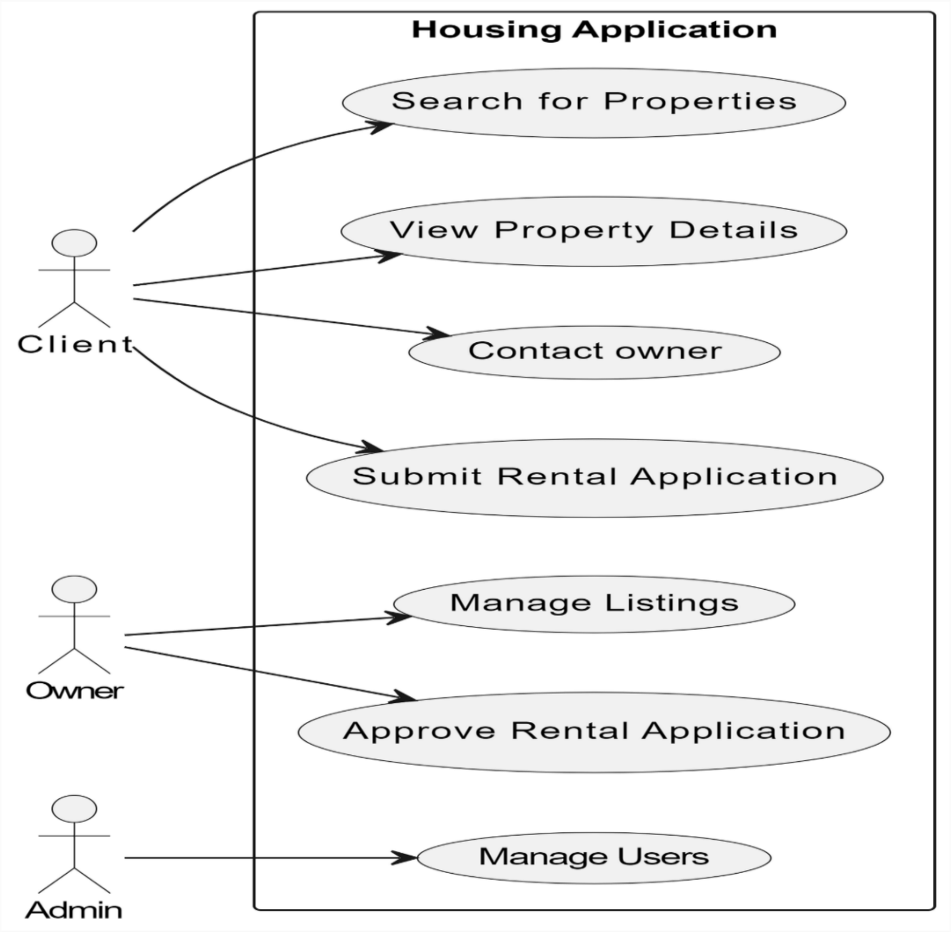
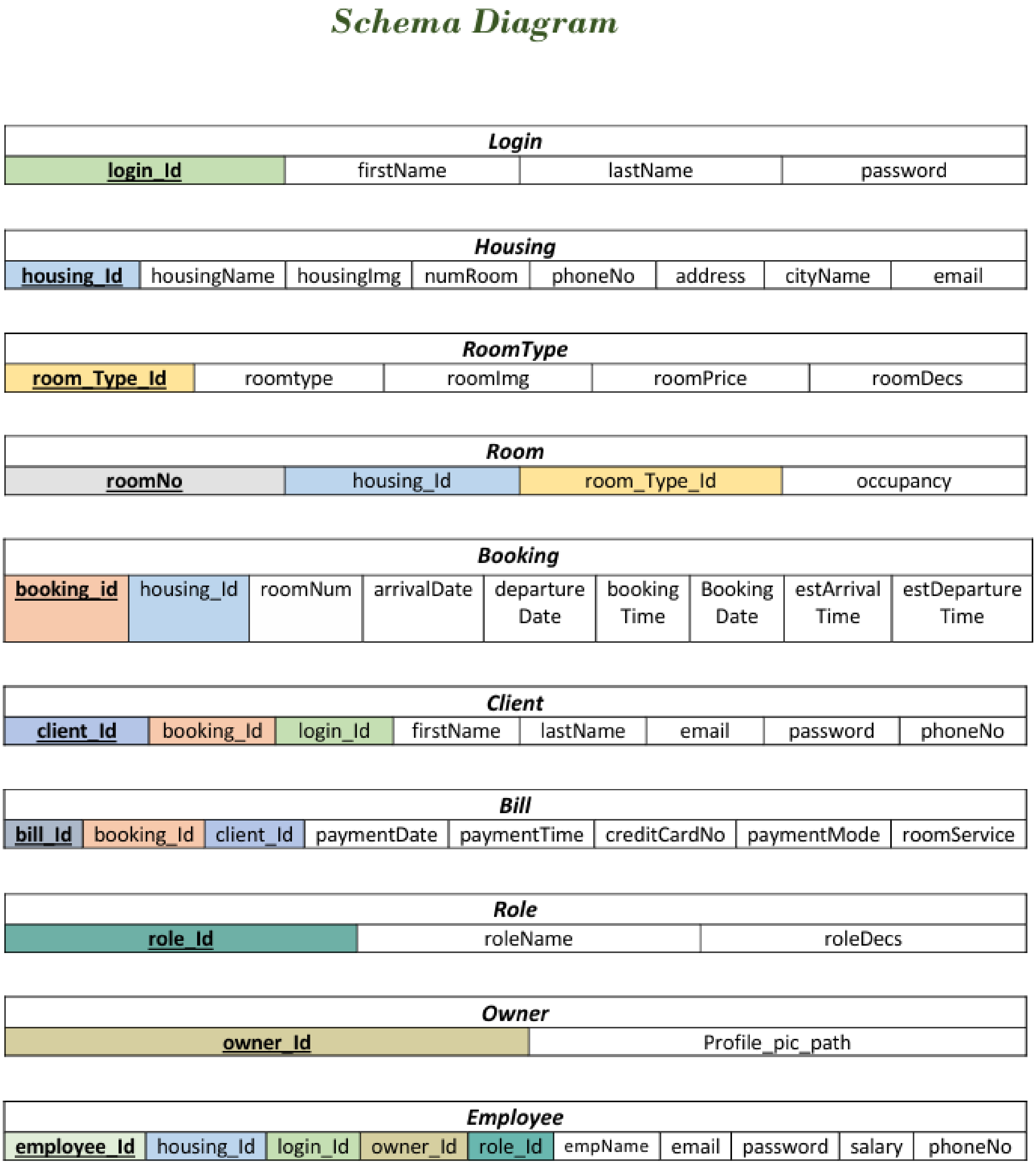


Figure 6.1.1: Use-case model for Renting Apartment System

6.2 Schema Diagram



Schema Diagram

Login

login_Id	firstName	lastName	password

Housing

housing_Id	housingName	housingImg	numRoom	phoneNo	address	cityName	email

RoomType

room_Type_Id	roomtype	roomImg	roomPrice	roomDecs

Room

roomNo	housing_Id	room_Type_Id	occupancy

Booking

booking_id	housing_Id	roomNum	arrivalDate	departure Date	booking Time	Booking Date	estArrival Time	estDeparture Time

Client

client_Id	booking_Id	login_Id	firstName	lastName	email	password	phoneNo

Bill

bill_Id	booking_Id	client_Id	paymentDate	paymentTime	creditCardNo	paymentMode	roomService

Role

role_Id	roleName	roleDecs

Owner

owner_Id	Profile_pic_path

Employee

employee_Id	housing_Id	login_Id	owner_Id	role_Id	empName	email	password	salary	phoneNo

6.3 ER-Diagram

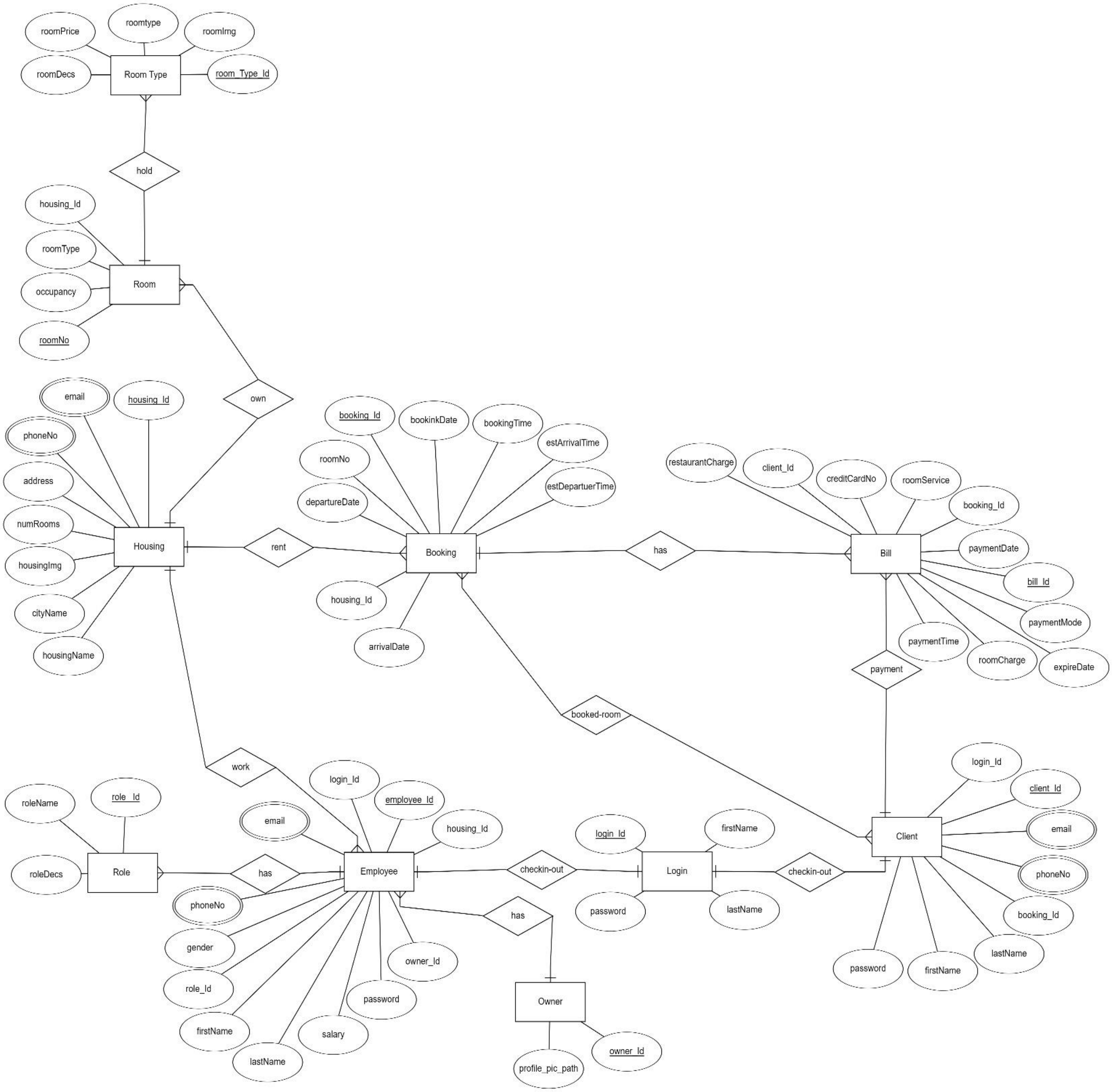


Figure 6.2.1 ER-Diagram

6.4 Class Diagram

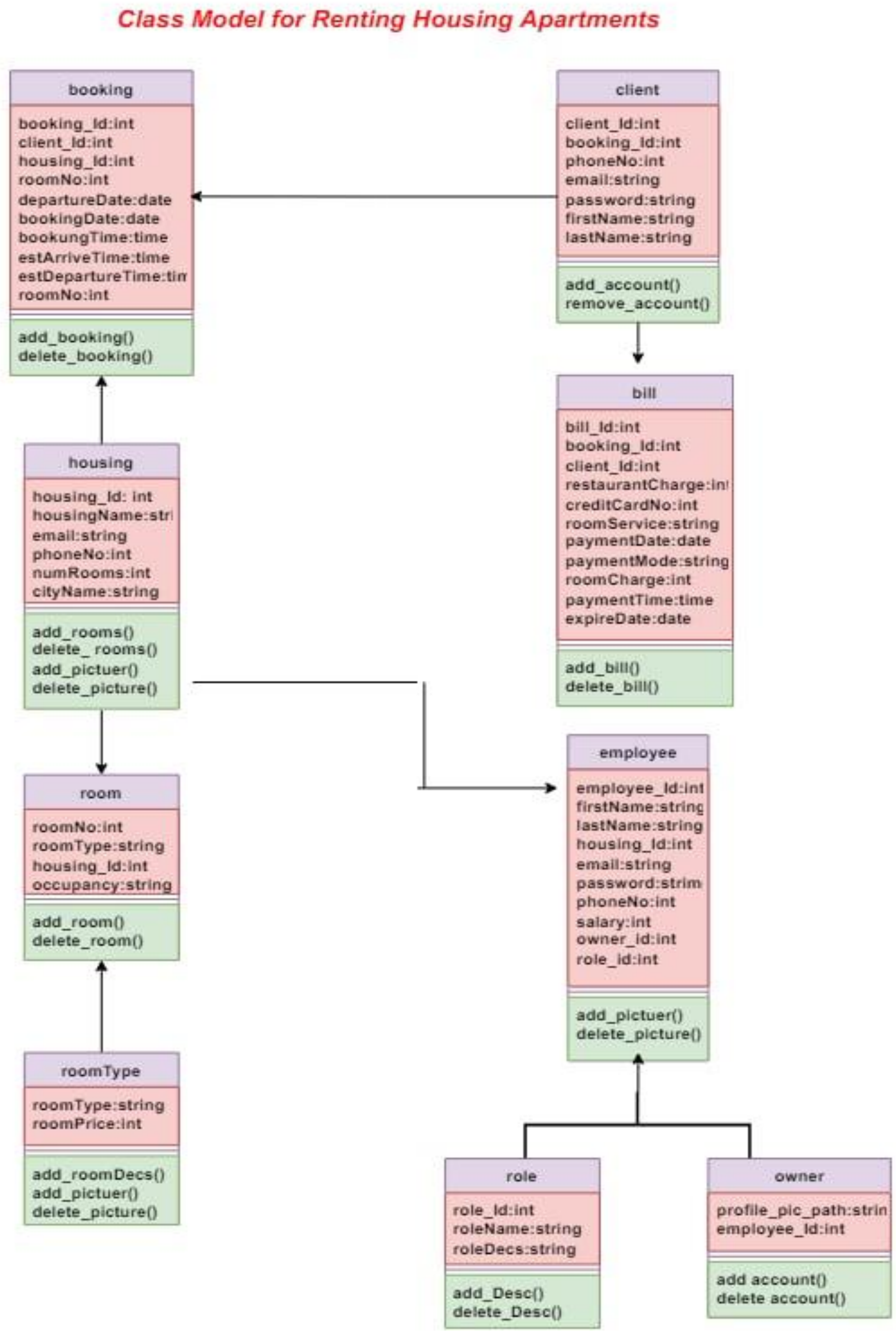


Figure 6.3.1: Class model for Renting Apartment System

6.5 Activity-Diagram

6.5.1 Client Activity-Diagram

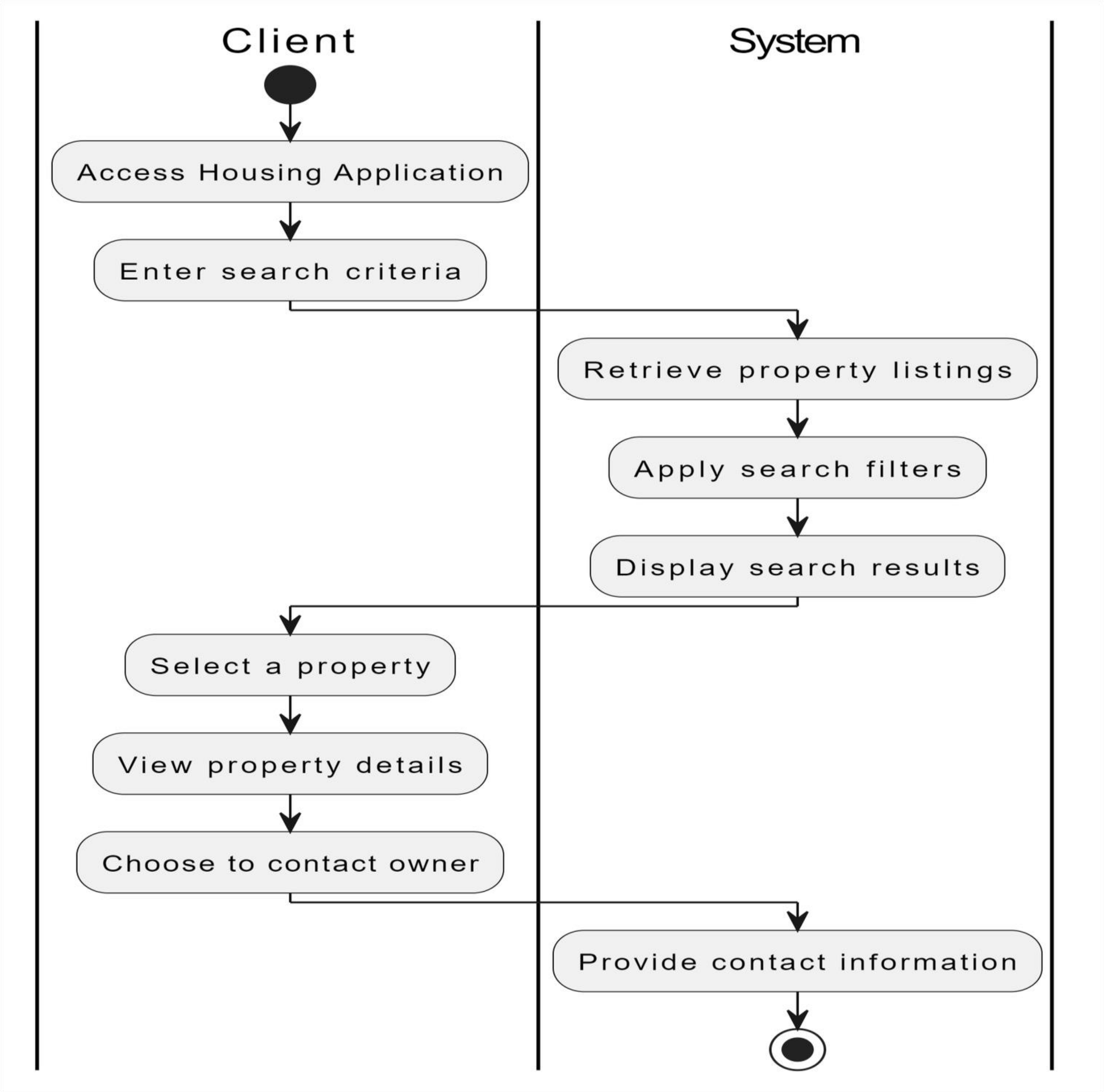


Figure 6.5.1.1 Client Activity-Diagram

6.5.2 Owner Activity-Diagram

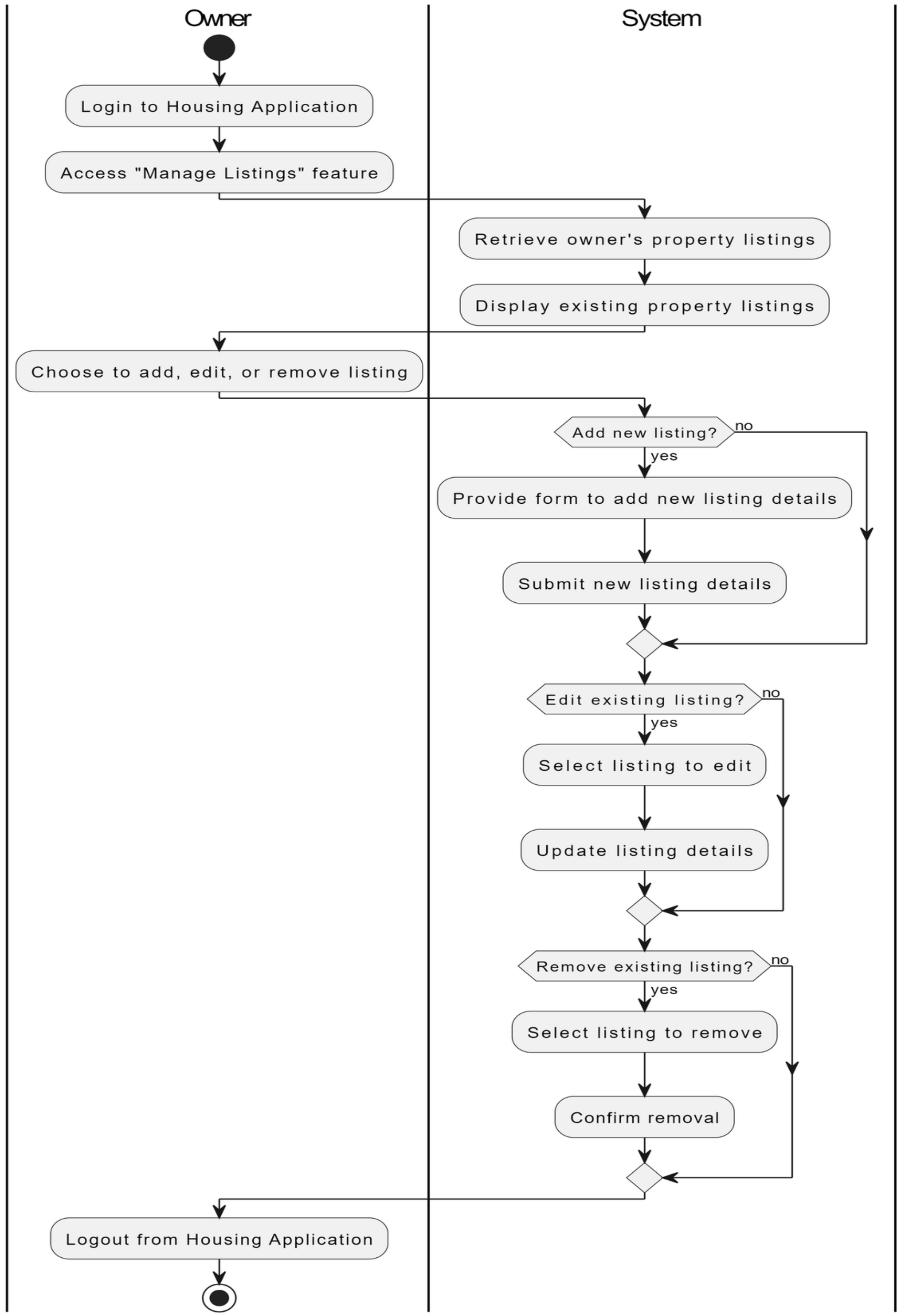

Owner
System
Login to Housing Application
Access "Manage Listings" feature
Retrieve owner's property listings
Display existing property listings
Choose to add, edit, or remove listing
Add new listing?
no
yes
Provide form to add new listing details
Submit new listing details
Edit existing listing?
no
yes
Select listing to edit
Update listing details
Remove existing listing?
no
yes
Select listing to remove
Confirm removal
Logout from Housing Application

Figure 6.5.2.1 Owner Activity-Diagram

6.5.3 Admin Activity-Diagram

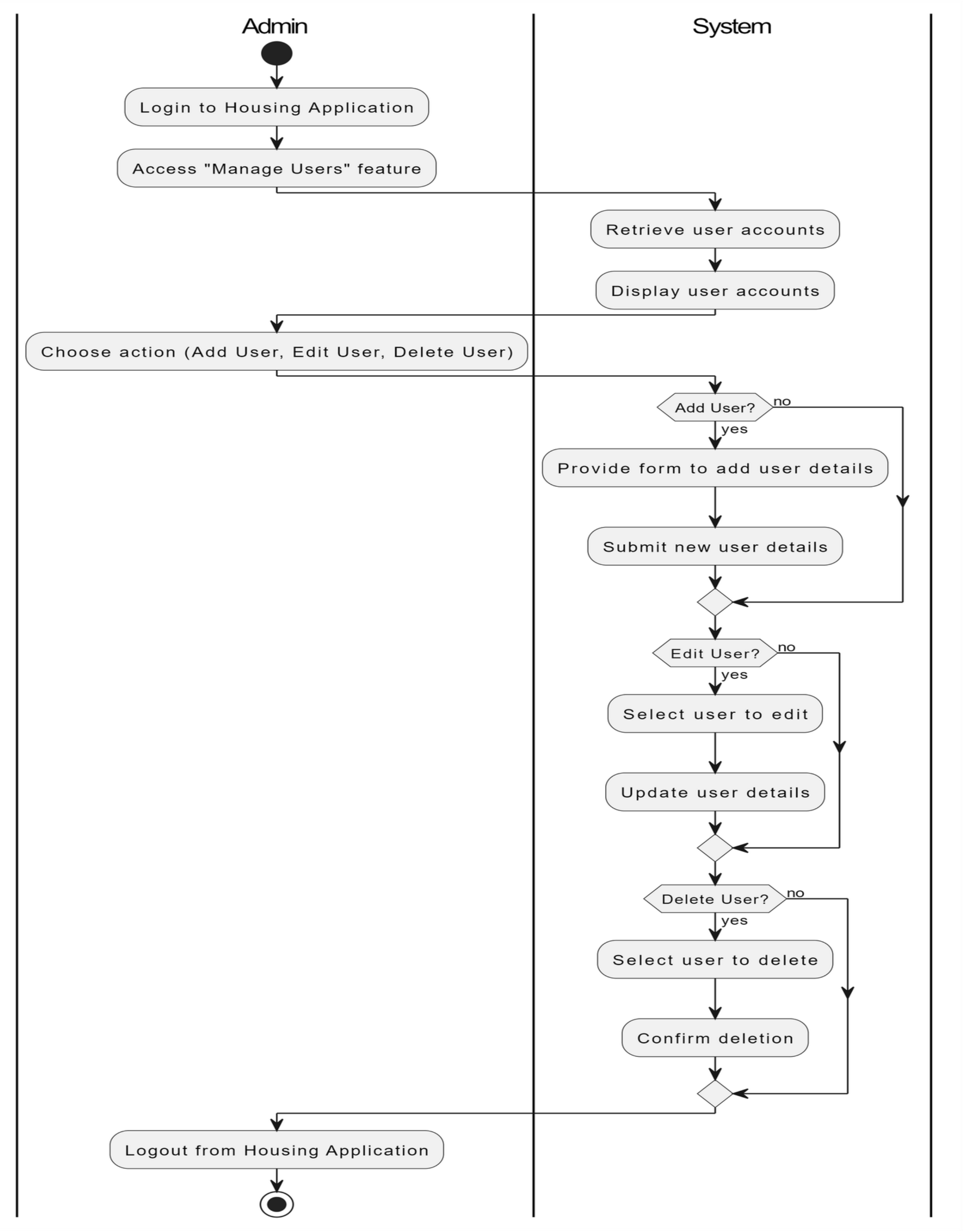


Figure 6.5.3.1 Admin Activity-Diagram

6.6 Sequence Diagram

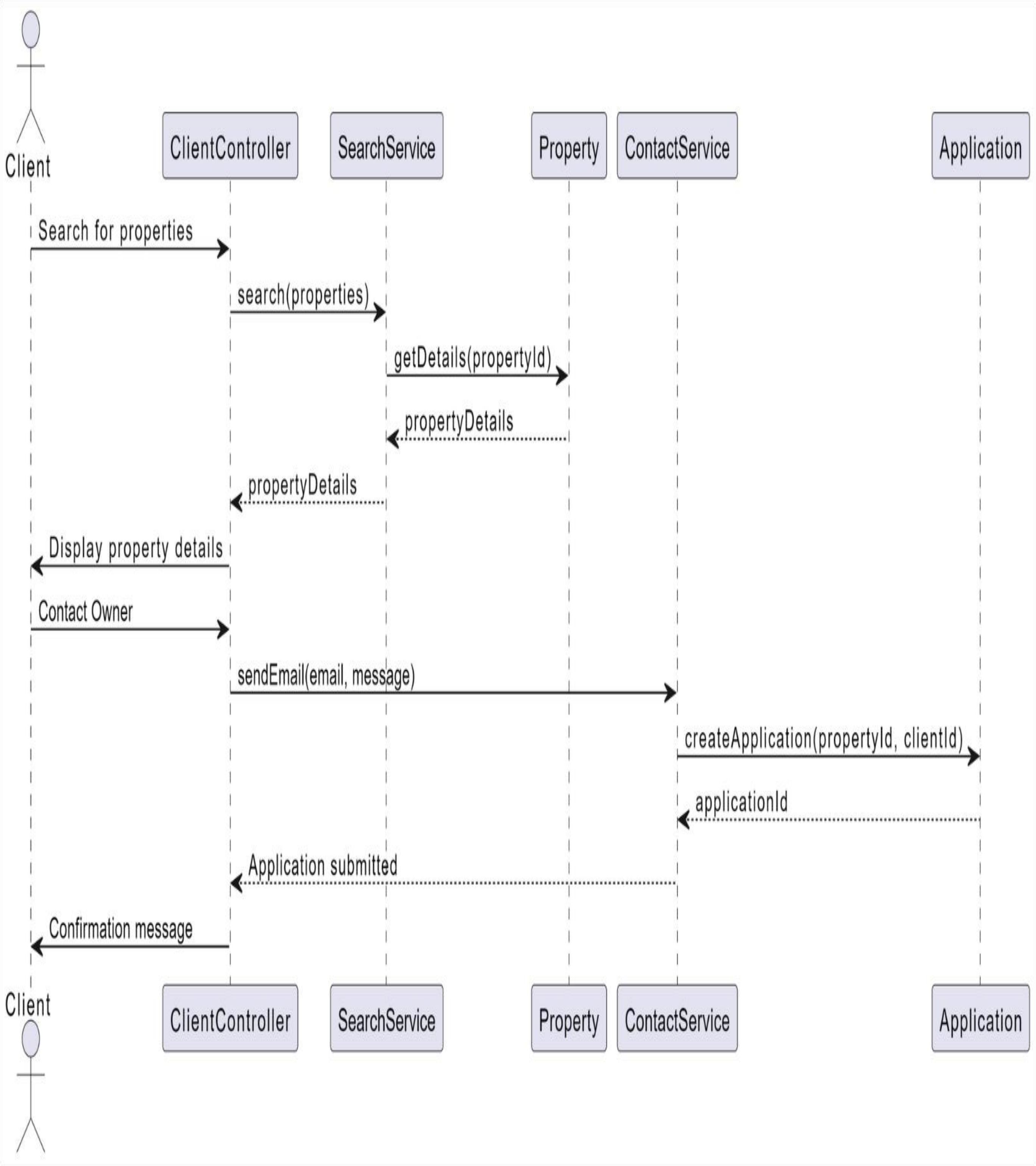


Figure 6.6.1 Client Sequence Diagram

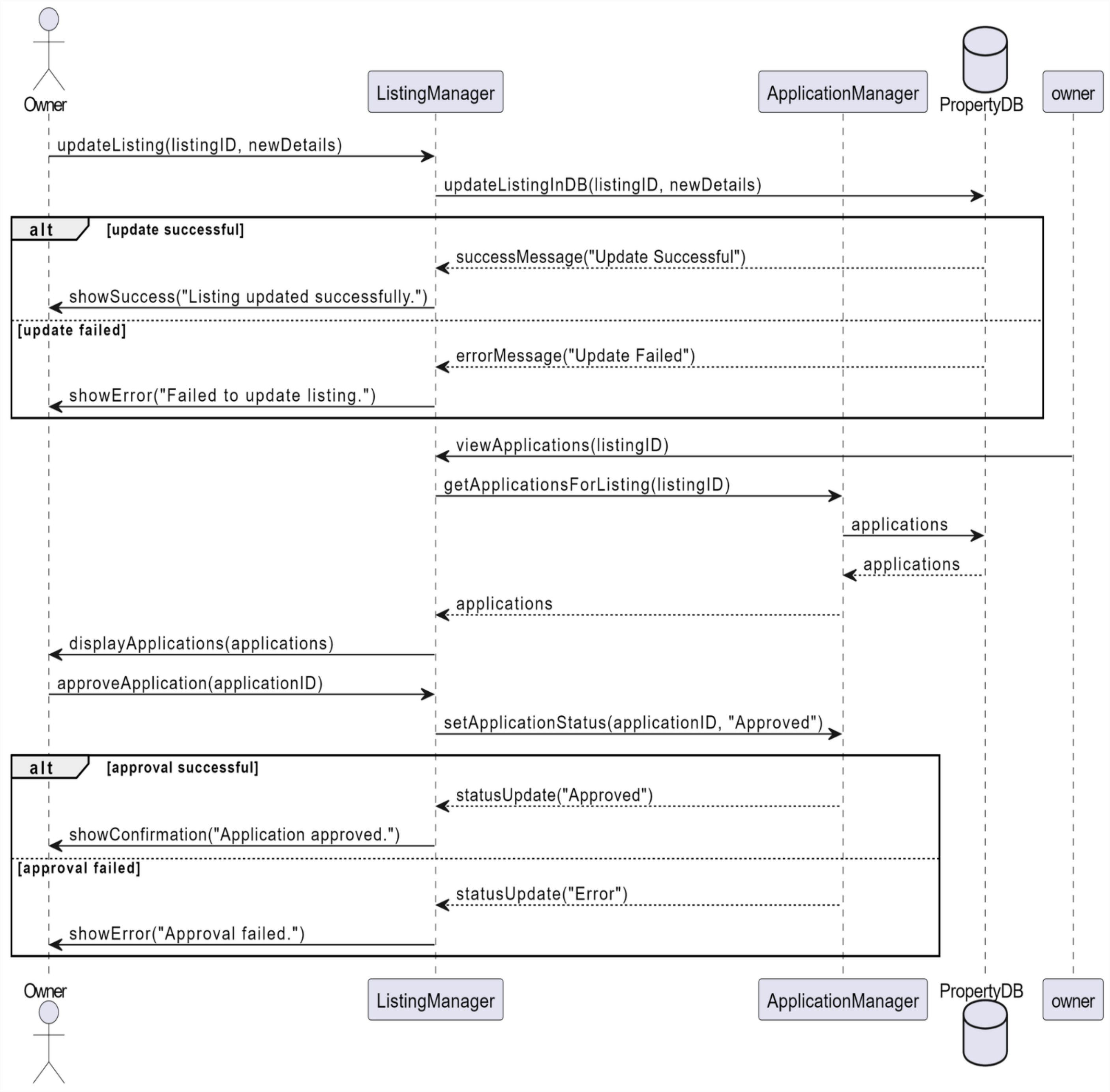

Owner
ListingManager
ApplicationManager
PropertyDB
owner
updateListing(listingID, newDetails)
updateListingInDB(listingID, newDetails)
alt
[update successful]
successMessage("Update Successful")
showSuccess("Listing updated successfully.")
[update failed]
errorMessage("Update Failed")
showError("Failed to update listing.")
viewApplications(listingID)
getApplicationsForListing(listingID)
applications
applications
applications
displayApplications(applications)
approveApplication(applicationID)
setApplicationStatus(applicationID, "Approved")
alt
[approval successful]
statusUpdate("Approved")
showConfirmation("Application approved.")
[approval failed]
statusUpdate("Error")
showError("Approval failed.")
Owner
ListingManager
ApplicationManager
PropertyDB
owner

Figure 6.6.2 Owner Sequence Diagram

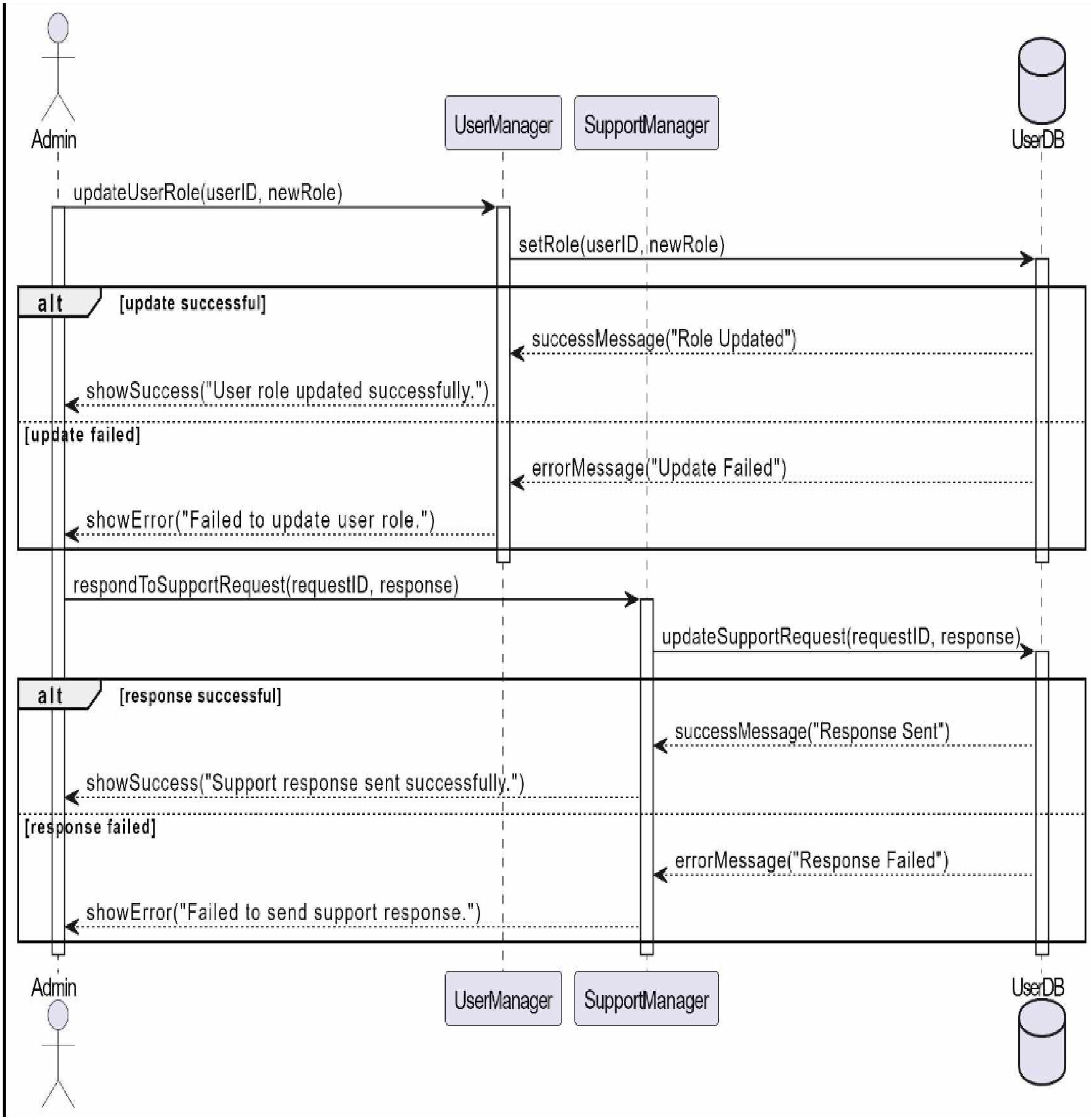

Admin
UserManager
SupportManager
UserDB
updateUserRole(userID, newRole)
setRole(userID, newRole)
alt [update successful]
successMessage("Role Updated")
showSuccess("User role updated successfully.")
[update failed]
errorMessage("Update Failed")
showError("Failed to update user role.")
respondToSupportRequest(requestID, response)
updateSupportRequest(requestID, response)
alt [response successful]
successMessage("Response Sent")
showSuccess("Support response sent successfully.")
[response failed]
errorMessage("Response Failed")
showError("Failed to send support response.")
Admin
UserManager
SupportManager
UserDB

Figure 6.6.3 Admin Sequence Diagram

Chapter 7: System evolution

This document was developed between all four group members communicating via Zoom and Discord because meetings face-face wasn't easy through the current situation. There was much debate over the existence of certain classes. One of the biggest discussions within the group was regarding how to lease terminations should be handled. After we back to the owners, we found most of them didn't trust to make an end of regarding and pay the treatment through the system.

Therefore, the most important things that will be developed and added to the system are as follows:

- Payment and rent payments are provided through the system itself, in Er-diagram we have put all the information's about ways of payments, but we didn't provide this feature in the web page, so we hope we add this feature to the page in the future.

- Important services will be added to the system, such as a request to change an apartment, a request for modifications and repairs to the apartment, and financial matters related to these repairs will be handled through the system itself.

- All legal matters will be added to the system, such as in the event of a request to cancel the contract, move out of the apartment or expire the period and demand a reduction or increase in the rent.

- A full page will be devoted to the owner himself for his mathematical matters with all the details, telling him, for example, the amount of profit from rents during a particular week or month, and knowing the amount of costs paid for repairs and all of these things that the owners deal with on paper, which are cumbersome in the event that the owner has a lot of Apartments

- For the tenant, he will be provided with the option to pay the way he wants through: Pal Pay, Visa Card, ...

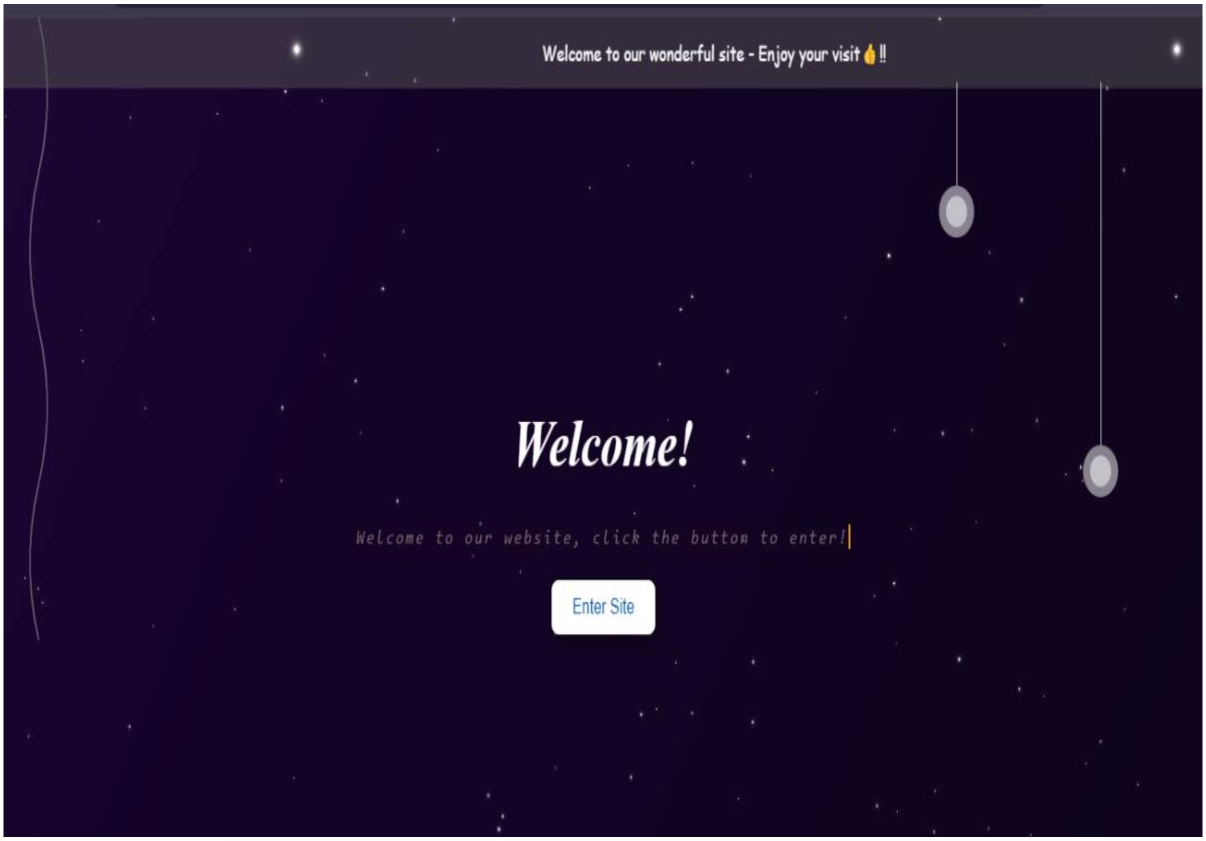
Chapter 8: Web Explanation

8.1 Links for our project on GitHub

- https://github.com/islammosmar1/software_project
- https://islammosmar1.github.io/software_project/

8.2 Images of the web page functions

1) The main interface page through which one enters the site.



Welcome to our wonderful site - Enjoy your visit 👍!!

Welcome!

Welcome to our website, click the button to enter!

Figure 9.1.1

2)Start your search, which is the interface through which your information and specifications of the university housing you prefer are placed.

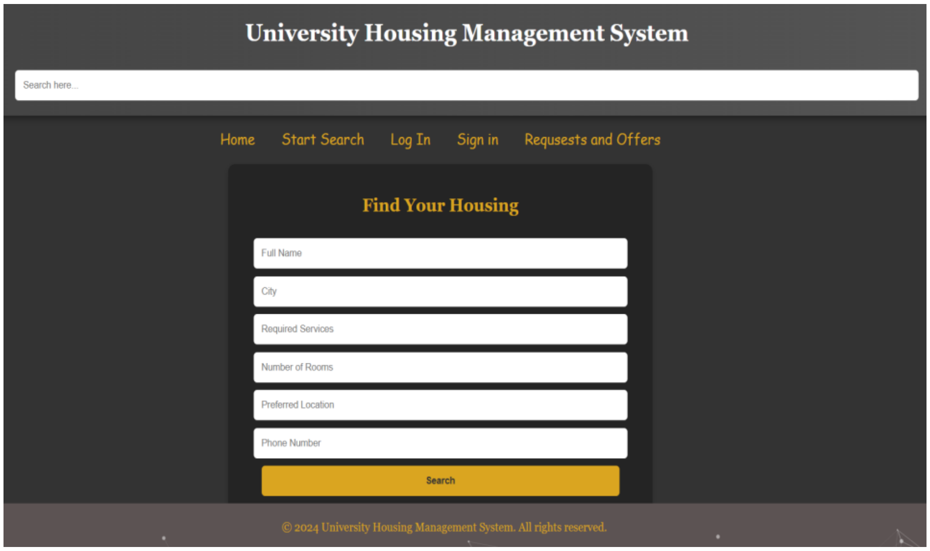


Figure 9.2.1

3) login is the page responsible for logging in for users so that they can enjoy all the features of the site.

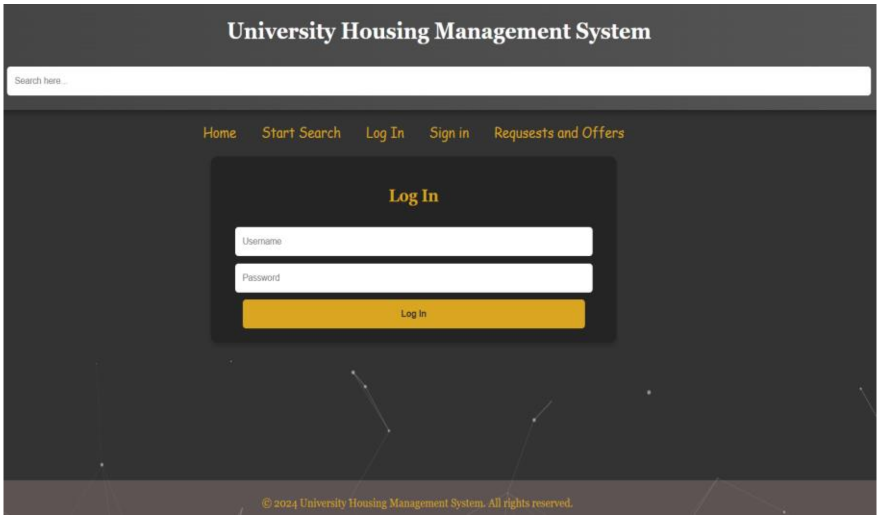
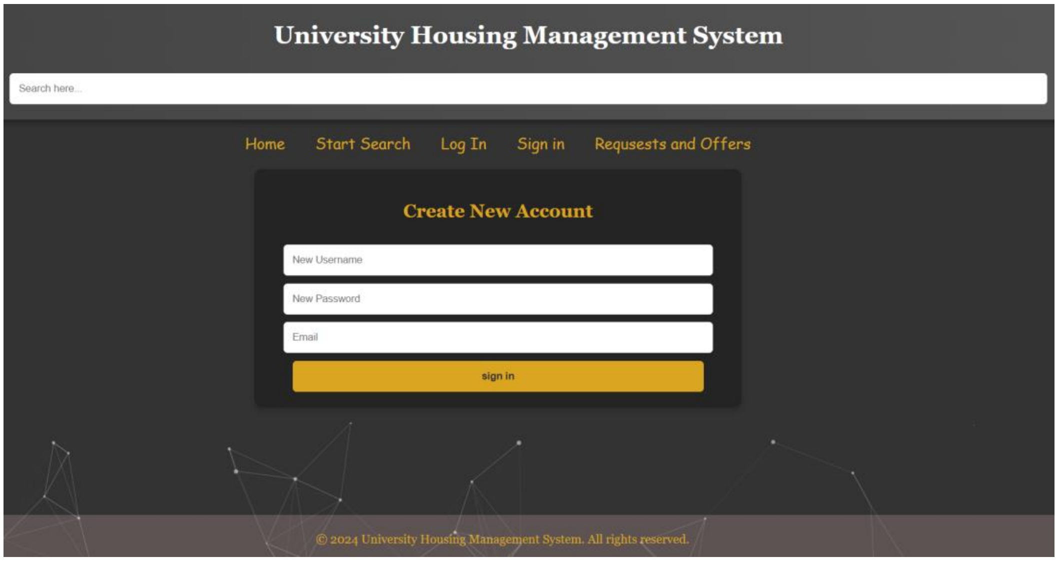

University Housing Management System
Search here...
Home
Start Search
Log In
Sign in
Requsests and Offers
Log In
Username
Password
Log In
© 2024 University Housing Management System. All rights reserved.

Figure 9.3.1

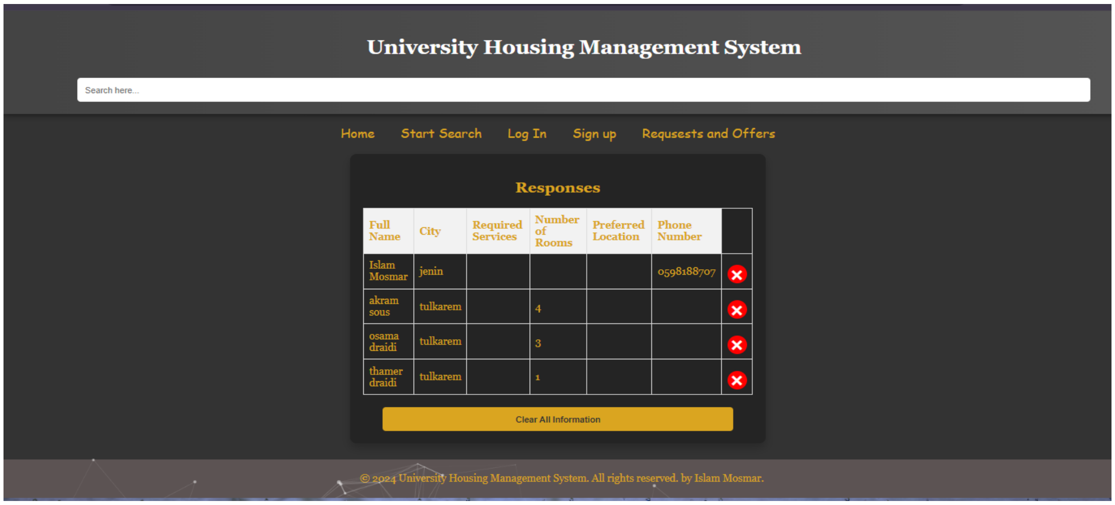
4) The sign-up page enables the user to register a new account if he does not have a previous account.



© 2024 University Housing Management System. All rights reserved.

Figure 9.4.1

5) Finally, the list of requests and offers, which is the list in which all searches performed by users are purified, and the request that you sent from the user who made the request can. only be deleted.



Full Name	City	Required Services	Number of Rooms	Preferred Location	Phone Number	
Islam Mosmar	jenin				0598188707	☒
akram sous	tulkarem		4			☒
osama draidi	tulkarem		3			☒
thamer draidi	tulkarem		1			☒

© 2024 University Housing Management System. All rights reserved. by Islam Mosmar.

Figure 9.5.1

Chapter 9: Conclusion

After the completion of this project, we felt we have learned a significant amount about the work and processes that go into handling apartment management. There are many different users in the process, many of whom we've never actively thought about. This includes not only the owners and Tenants but also supervisors, employees, and where they all fit into managing an apartment unit. Our class diagram design involved a lot of collaborative thinking to figure out where the users fit and what roles do, they accomplish.

The job of creating a requirements document for the system taught us much about how UML is structured. There are a lot of aspects of it that are still confusing such as some of the different objects in a sequence diagram. Thinking about the sequence diagram and the class diagram, were our biggest challenges. Some of the aspects of the system, such as request renting, are very complicated and required a lot of critical thinking on our part to be able to identify all the different scenarios and sequence steps for proper identification. The project in general definitely did a lot to expand our skills in UML and modeling and hopefully, we will be able to use these skills in the future.

We have learned how to develop use cases from the domain problem, also we have learned to design the static vision of the system developing the domain class diagram, and how to use controller classes, and how to develop a sequence diagram from the use case description.

So finally, the general need for apartment owners who own a large number of them, it will become difficult to carry paper with him wherever he goes, therefore through this system he will know at any time his tenants and know how long the time for rent is and how much is left and what is the cost of rent at any time he needs this information because if he wanted to obtain all This information on paper will take a long time to obtain.

Chapter 10: TEAM WORK

Name	Work
Akram Sous	Class diagram, schema, ER-diagram, literature review, functional requirements
Islam Mosmar	System architecture, ER-diagram, literature review, web explanation, non-functional requirements
Osama Draidi	Use case diagram, sequence diagram, Activity diagram, functional requirements
Thamer Draidi	Use case diagram, sequence diagram, Activity diagram, non-functional requirements

The rest of the titles and parts of the project that are not present in the table above, we have cooperated in completing them, including the web page and its code.

We reviewed these titles together and interacted with each other.

The titles in the table represent what each of us did alone as we reviewed each other's work.

Chapter 11: List Of References

References at this point will be drawn from personal experiences and widely available resources on the internet and regulations regarding residential rental units.

We used the Internet to learn and take advantage of the way of designing models and the way of writing requirements, then we used the draw.io site to draw these models (use-case, activity, sequence, and class models) and ERDPlus to draw the ER diagram, this led to make us experience in the field of UML models. We designed the web by visual studio, and we wrote the SRS on Microsoft Word.

We used many sites to learn how to make this system:

- zumper.com
- apartments.com
- homes.com