

**SOFTWARE REQUIREMENTS AND  
DESIGN DOCUMENT  
INTERNAL DOCUMENT  
FOR  
HOSTEL MANAGEMENT SYSTEM**

---

**PREPARED BY:**  
**HASSAN REHMAN 22I-2592**  
**HAFSA WAQAR 22I-2625**  
**SHAJIA TAHIR 22I-2695**

**FAST UNIVERSITY NUCES ISLAMABAD**  
**06-06-2024**

---

.....	0
1. INTRODUCTION.....	7
1.1 Purpose.....	7
1.2 Product Scope.....	7
1.3 Title.....	7
1.4 Objectives .....	7
1.5 Problem Statement.....	8
2. Overall Description .....	9
2.1 Product Perspective.....	9
2.2 Product Functions .....	10
2.3 List of Use Cases .....	11
2.4 Extended Use Cases.....	11
2.4.1. Booking a Hostel .....	11
2.4.2. MANAGING HOSTEL.....	13
2.4.3. Managing Owner .....	14
2.4.4. SEARCHING HOSTEL.....	16
2.4.5. Managing Student Information.....	18
2.4.6. Advertising Hostel .....	19
2.4.7. Comparing Hostels.....	21
2.4.8. Managing Complaints and Maintenance Requests .....	23
2.4.9. SUBMIT FEEDBACK AND REVIEWS .....	24
2.4.10. Registering Student.....	26
2.4.11. Managing Fee Details .....	28
2.4.12. Availing Transportation Services .....	30
2.4.13. Allocating Rooms .....	31
2.4.14. Messaging and Communication.....	33
2.4.15. Managing Leave requests .....	34
2.4.16. MANAGING FOOD DETAILS.....	36
2.5 Use Case Diagram .....	38
3. Other Nonfunctional Requirements .....	39
3.1 Performance Requirements.....	39
3.2 Safety Requirements .....	39
3.3 Security Requirements.....	39
3.4 Software Quality Attributes .....	39
3.5 Business Rules.....	40

---

3.6 Operating Environment.....	40
3.7 User Interfaces .....	41
3.7.1 Login.....	41
3.7.2 Login Error Message.....	42
3.7.3 Hostel Owner Dashboard .....	42
3.7.4 Resident Dashboard .....	43
3.7.5 Search Hostel .....	44
3.7.6 Select Hostel.....	45
3.7.7 Confirm Details.....	46
3.7.8 Success Message .....	47
3.7.9 Search Hostel Results.....	47
3.7.10. Comparison Result.....	48
3.7.11. Warning Message .....	48
3.7.12 Compare Hostels.....	49
3.7.13. Application Cancelled.....	49
3.7.14. Leave Request Application From .....	50
3.7.15. Error message.....	50
3.7.16. Feedback Form.....	51
3.7.17. Complaint Request .....	52
3.7.18. Compose new complaint .....	53
3.7.19. Complaint Recorded .....	54
3.7.20. Confirmation Dialog.....	54
3.7.21. Advertise Hostel.....	55
3.7.22. Manage Hostel .....	56
3.7.23. Validation Error .....	56
3.7.24. Leave Request Management .....	57
3.7.25. Room Allocation .....	58
3.7.26. Complaints Management .....	59
3.7.27. Room Allocation .....	60
3.7.27. Composing response to complaints.....	61
4. Domain Model .....	62
5. System Sequence Diagram .....	63
5.1 Book Hostel.....	63
5.2 Manage hostel .....	64
5.2 Alternate scenario .....	65

---

5.3 Manage owner.....	66
5.4 search hostel .....	67
5.5 Manage student information.....	68
5.6 advertise hostel .....	69
5.7 Compare hostel.....	70
5.7 alternate scenario .....	71
5.8 Manage Complaints and Maintenance Requests.....	72
5.9 submit feedback and reviews.....	73
5.9 alternate scenario .....	74
5.10 register student.....	75
5.10 alternate scenario.....	76
5.11 manage fee details .....	77
5.11 alternate scenario.....	78
5.12 avail Transportation services .....	79
5.13 Allocate rooms .....	80
5.14 message and communication.....	81
5.15 manage leave request.....	82
5.16 manage food details .....	83
6. Sequence Diagram .....	84
6.1 Book a Hostel .....	84
6.1.1 Search Hostel .....	84
6.1.2 Select Hostel.....	85
6.1.3 Confirm Booking .....	85
6.2 Manage Hostel.....	86
6.2.1 View and Manage Staff Details .....	86
6.2.2 Task Assignment .....	86
6.2.3 Update Room Availability.....	87
6.2.4 Update Database .....	87
6.3 Manage Owner .....	88
6.3.1 Update Hostel Details .....	88
6.3.2 Upload Hostel Pictures .....	89
6.4. Searching Hostel .....	90
6.4.1 Search Hostel .....	90
6.4.2 Select Hostel.....	91
6.5. Managing Resident Information.....	92

---

6.5.1 Select Resident .....	92
6.5.3 Change Resident Information.....	93
6.6. Advertise Hostel .....	94
6.6.1 Create Advertisement .....	94
6.6.2 Store Advertisement.....	94
6.7. Comparing Hostels.....	95
6.7.1 Compare Criteria.....	95
6.7.2 Select Hostels.....	96
6.7.3 Take Actions.....	96
6.8. Managing Complaints and Maintenance Requests .....	97
6.8.1. Retrieve Forms .....	97
6.8.2 Form Submission .....	97
6.8.3 Confirm Submission.....	98
6.9. Submit feedback and reviews.....	98
6.9.1 Retrieve Feedback Form .....	98
6.9.2 Submit Form .....	99
6.9.3 Send Notification.....	99
6.10. Register User .....	100
6.10.1 Get Registration Form .....	100
6.10.2 Fill Registration Form.....	100
6.10.3 Store User Profile .....	101
6.11. Manage Fee Details .....	102
6.11.1 Access and view fee details.....	102
6.11.2 Update fee amount .....	103
6.12. Avail Transportation Services .....	103
6.12.1 Access transportation module .....	103
6.12.2 Select desired date and time.....	104
6.12.3 Select transportation .....	104
6.12.4 Confirm Booking .....	105
6.13. Allocate Rooms .....	105
6.13.1 Access room allocation.....	105
6.13.2 Select room for allocation .....	106
6.13.3 Select room for changing occupancy status.....	106
6.14. Message and Communication .....	107
6.14.1 Compose Message .....	107

---

6.14.2 View Message.....	107
6.15. Manage Leave Requests.....	108
6.15.1 Access leave management section .....	108
6.15.2 Initiate leave request.....	108
6.15.3 Send notification .....	109
6.16. Manage Food Details.....	109
6.16.1 Access food details.....	109
6.16.2 Modify Food menu .....	110
7. updated artificats .....	111
7.1 Advertise Hostel .....	111
7.1.1 System sequence Diagram.....	111
7.1.2 Sequence Diagram: .....	112
a. Create Advertisement .....	112
b. Store Advertisement.....	113
7.2 Compare Hostel.....	114
7.2.1 Sequence Diagram.....	114
7.2.2 Sequence Diagrams .....	115
a. Compare Criteria.....	115
b. Select Hostel.....	116
c. Take Action .....	117
7.3 Manage Hostel.....	118
7.3.1 System Sequence Diagram .....	118
7.3.1 Sequence Diagrams .....	119
a. Open Hostel Manage Page.....	119
b. Update Hostel .....	120
c. Delete Hostel .....	120
7.4 Search Hostel.....	121
7.4.1 System Sequence Diagram .....	121
7.4.2 Sequence Diagrams .....	122
a. Select Hostel .....	122
b. Search Hostel .....	123
7.5 Manage Leave Request .....	124
7.5.1 System Sequence Diagram .....	124
7.5.2 Sequence Diagrams .....	125
a. Access Leave .....	125

---

b. Apply Leave Request.....	125
7.6 Allocate Room.....	126
7.6.1 System Sequence Diagram .....	126
7.6.2 Sequence Diagram.....	127
a. Access Room Allocation.....	127
b. Add Room .....	128
7.7. Complaint and Maintenance.....	129
7.7.1 System Sequence Diagram .....	129
7.7.2 Sequence Diagram.....	130
a. View Complaint .....	130
b. Register Complaint .....	130
7.8 Booking A Hostel .....	131
7.8.1 System Sequence Diagram .....	131
7.8.2 Sequence Diagrams .....	132
a. Select Hostel .....	132
b. Show Details .....	133
c. Book Room .....	134
7.9 Submit Feedback and Reviews.....	135
7.9.1 System Sequence Diagram .....	135
7.9.2 Sequence Diagrams .....	136
a. Give Feedback .....	136
b. Submit Form.....	136
8. Class Diagram.....	137
9. High-level Architecture.....	138
9.1 PACKAGE DIAGRAM.....	139
9.2 Deployment Diagram.....	140
9.3 Component Diagram .....	141
10. Work Division Table.....	141

---

## 1. INTRODUCTION

### 1.1 PURPOSE

"Hostel Hunt" is an innovative application designed to meet the needs of hostel residents, managers, and owners. Our platform aims to streamline the hostel experience by addressing common challenges faced by residents, such as dealing with strict rules and staying updated in the digital age. Additionally, Hostel Hunt goes beyond traditional hostel services by offering solutions for accessing healthy food, maintaining hygiene, and even providing transportation options. With our comprehensive approach, we strive to enhance the overall hostel living experience for everyone involved. Moreover, Hostel Hunt also serves as a platform for hostel advertising, allowing users to compare different hostels and make informed decisions about their accommodation.

### 1.2 PRODUCT SCOPE

Hostel Hunt is an HBMA which streamlines the process of searching and booking hostels at a reasonable price. Hostel Hunt is designed to address issues faced by hostel residents, managers and owners. Residents including students and job professionals often find difficulties to find a hostel according to their need. On the other hand, hostel owners find it challenging to properly advertise and manage their hostels.

### 1.3 TITLE

Hostel Management System

### 1.4 OBJECTIVES

Objectives include:

**For Users:**

- **Navigation and Comparison:** Users can navigate the application to compare various hostels based on their needs. This includes filtering by location, price, facilities, and reviews.

- 
- **Simplified Booking Process:** The system facilitates a user-friendly booking process, allowing users to reserve the desired hostel with ease.
  - **Inclusive Feature Access:** Users can explore and utilize a wide range of features within the application. This includes booking meals, arranging transportation, and initiating communication with the hostel management for any inquiries or requests.

#### **For Hostel Owners:**

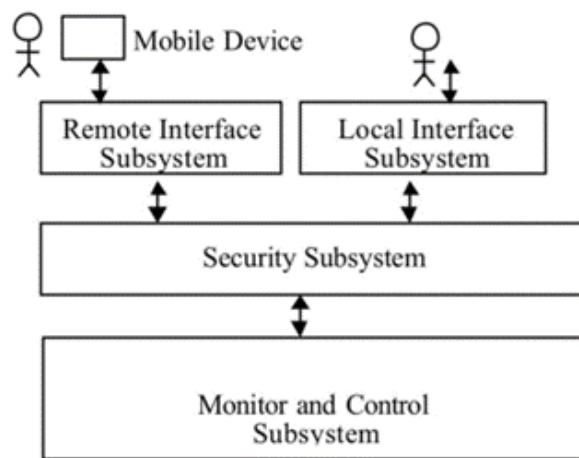
- **Effective Hostel Advertising:** Hostel owners can leverage the platform to advertise their properties, showcasing their unique offerings and attracting potential residents.
- **Efficient Resident Management:** The system offers tools to manage resident information efficiently. This includes tracking bookings, resident details, and facilitating communication with guests.

## 1.5 PROBLEM STATEMENT

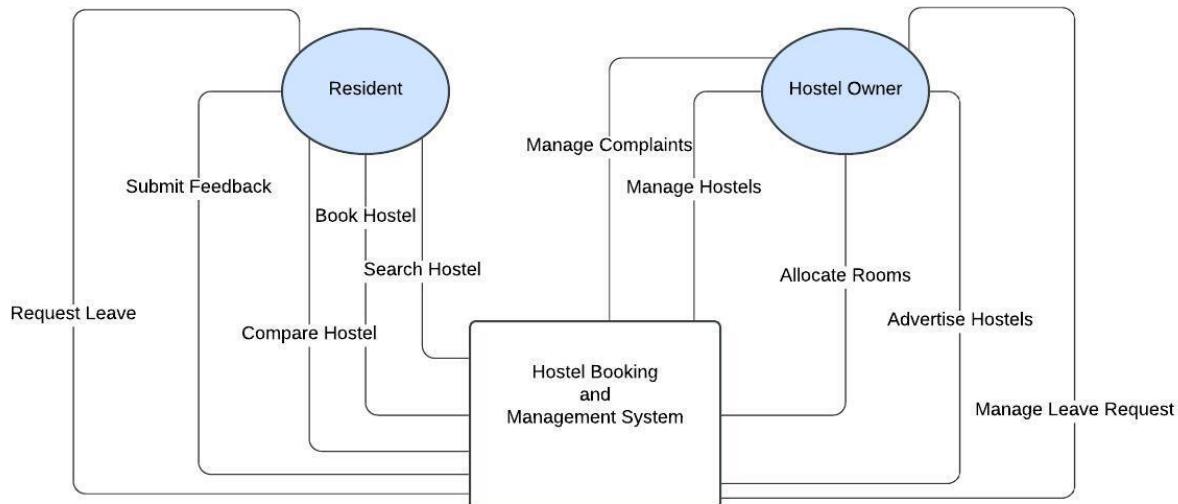
Problem	Description
The problem of	students, job professionals and hostel owners to book and advertise hostels is the main issue.
Affects	students and working professionals to find suitable hostels. It is difficult for hostel operators to successfully market and run their hotels.
The impact of which	decreased availability of appropriate hostel options for job professionals and students also it reduced occupancy rates and lost profits for hostel owners because of poor advertising.
a successful solution would be	HBMA which provides a user-friendly interface for hostel residents so that they can easily find a hostel and book it. For owners it would be easier to advertise and manage hostel issues and accommodation fee transactions.

## 2. OVERALL DESCRIPTION

### 2.1 PRODUCT PERSPECTIVE



## 2.2 PRODUCT FUNCTIONS



### Booking

- Users can search for and book available hostel rooms.
- Search results prioritize available rooms and are sorted by relevance and availability.

### Advertising

- Owners can post advertisements for available rooms.
- Advertisements must adhere to hostel policies and standards.

### Comparing

- Users can compare different hostels based on various criteria such as room availability, amenities, and price.

### Room Allocation

- Owners can allocate rooms to residents.
- Rooms can only be allocated if they are available.

### Leave Request

- Residents can submit leave requests, which must be done at least one week in advance.
- Owners must respond to leave requests within 3 business days.

### Complaints

- Residents can file complaints about hostel services or conditions.

- Complaints must be acknowledged by owners within 24 hours and addressed within 48 hours.

### **Manage Hostel**

- Owners can manage hostel details, including room availability and advertisements.
- Role-based access control ensures that only authorized personnel can perform specific management functions.

## **2.3 LIST OF USE CASES**

- 2.3.1. Register User
- 2.3.2. Search Hostel
- 2.3.3. Book Hostel
- 2.3.4. Allocate Room
- 2.3.5. Advertise Hostel
- 2.3.6. Manage Hostel
- 2.3.7. Manage Owner
- 2.3.8. Manage Resident Information
- 2.3.9. Manage Fee Details
- 2.3.10. Compare Hostel
- 2.3.11. Manage Food Details
- 2.3.12. Manage Leave requests
- 2.3.13. Submit Feedback & Reviews
- 2.3.14. Manage Complaints and Maintenance
- 2.3.15. Messaging and communication

## **2.4 EXTENDED USE CASES**

### **2.4.1. BOOKING A HOSTEL**

Use Case 1	
Use Case ID:	1
Name	Booking a Hostel

<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Owner/Manager	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Students and Job Holders: Want to find and book hostel accommodations that meet their preferences.</li> <li>Hostel Owners and Managers: Need to manage the hostel and ensure customer satisfaction.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>Hostel management system is accessible.</li> <li>Users have valid accounts registered with the system.</li> </ul>	
<b>Postconditions</b>	Users can securely log in to their accounts using appropriate credentials.	
<b>Main Success Scenario</b>	Actor Action	System Responsibility
	1. User searches for hostel based on their preferences  3. User selects a hostel according to his/her needs and wants  5. User confirms the booking	2. System retrieves the hostel information that matches the search of user  4. System reserves the selected hostel for booking  6. Update the booking and notify the user for confirmation
<b>Extensions</b>	<ul style="list-style-type: none"> <li>If the entered credentials are incorrect, the system displays an error message and prompts the user to retry.</li> <li>In case of forgotten passwords, users can request password reset links via email.</li> </ul>	
<b>Special Requirements</b>	Ability for users to view and compare hostel details, reviews, and ratings before making a booking decision.	
<b>Variations in Technology and Data</b>	Use of location to display hostels based on the user's current location or specified area.	
<b>Frequency of Occurrence</b>	Hostel booking occurs regularly as students and job holders plan their accommodations.	
<b>Miscellaneous</b>	Regular updates to the hostel database to ensure accurate and up-to-date information for users.	

## 2.4.2. MANAGING HOSTEL

<b>Use Case 2</b>		
<b>Use Case ID:</b>	2	
<b>Name</b>	Managing Hostel	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Owner/Manager	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Hostel Owners: Interested in efficiently managing hostel operations.</li> <li>• Hostel Managers: Responsible for day-to-day hostel management tasks.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The hostel owner/manager is logged into the system.</li> <li>• Hostel owners and managers have valid accounts registered with the system.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Hostel owners and managers can effectively manage various aspects of hostel operations.</li> <li>• Changes made by owners and managers are made in the system.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Action</b>	<b>System Responsibility</b>
	1. User accesses the management dashboard  3. Manager/Owner navigates to the room allocation option  5. Owner/Manager can update room availability by marking rooms as occupied or vacant based on bookings.	2. System displays all available management options  4. System provides access to the room allocation module  6. System displays room availability status

	<p>7. Owners/Managers assigns tasks to the staff members</p> <p>10. Owner/manager can view and manage staff details like personal information, contact information and roles/responsibilities</p>	<p>8. System offers functionality to user to assign tasks to staff members.</p> <p>9. System provides access to staff management features, allowing viewing and editing of staff details.</p> <p>11. System updates the database with changes to room allocation, staff work schedules, and updates about staff tasks and assignments</p>
<b>Extensions</b>	If there are no available rooms for allocation, the system alerts the owner/manager and suggests alternative solutions.	
<b>Special Requirements</b>	Secure authentication for hostel owner/manager login.	
<b>Variations in Technology and Data</b>	Implementation of recovery mechanisms to prevent data loss and ensure system reliability.	
<b>Frequency of Occurrence</b>	Hostel management tasks occur regularly as owners and managers manage the hostel daily.	
<b>Miscellaneous</b>	Regular updates and maintenance of the system.	

### 2.4.3. MANAGING OWNER

<b>Use Case 3</b>	
<b>Use Case ID:</b>	3

<b>Name</b>	Managing Owner	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Owner/Manager	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Hostel Owners: Interested in efficiently managing their hostel profiles.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Hostel Owner is registered with the hostel management system.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Hostel owners can effectively manage their profiles and update information.</li> <li>• Changes made by owners are accurately reflected in the system.</li> </ul>	
<b>Main Success Scenario</b>	<ul style="list-style-type: none"> <li>• Actor Action</li> </ul>	System Responsibility
	1. Owner/Manager accesses the owner management dashboard  3. User views and updates their personal and hostel info from personal management section  5. Owner edits personal/ hostel information such as hostel name, address, contact information etc. 6. Owner changes profile pictures or uploads pictures of their hostel that they wish to upload  8. Owner then saves the changes made	2. Systems displays all the available owner management operations options  4. System fetches the information and displays in onto the screen for user to update  7. System provides access to user to change/update pictures

		9. System saves the updated changes in the database
<b>Extensions</b>	If there are errors or missing information in the profile, the system prompts the owner to provide the required details.	
<b>Special Requirements</b>	Implementation of a user-friendly and clear instructions for profile and advertisement management.	
<b>Variations in Technology and Data</b>	-	
<b>Frequency of Occurrence</b>	Owner management tasks occur periodically as owners update their profiles or manage advertisements promotional needs.	
<b>Miscellaneous</b>	Regular monitoring and maintenance of the system to address any technical issues	

#### 2.4.4. SEARCHING HOSTEL

<b>Use Case 4</b>		
<b>Use Case ID:</b>	4	
<b>Name</b>	Searching Hostel	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Students and job holders	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Students and Job Holders: Interested in finding suitable hostel accommodations based on their preferences and requirements.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>Hostel management system is accessible</li> <li>Students and job holders have registered accounts with valid credentials.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>Students and job holders can successfully search for hostels that match their preferences and requirements within the system.</li> </ul>	
<b>Main Success Scenario</b>	Actor Action	System Responsibility

	<p>1. Students/Job holders selects the hostel search feature</p> <p>3. User searches for a specific hostel according to his/her needs and wants by using filters</p> <p>5. User then submits the search</p> <p>7. User selects a hostel that matches his/her interests the most</p>	<p>2. System provides access to search feature</p> <p>4. Provides search and filter options to user</p> <p>6. System retrieves all available hostels that matches the users search and displays them all for the user</p> <p>8. System will then display detailed information about the selected hostel for the user</p>
<b>Extensions</b>	If no hostels match the specified search criteria, the system notifies the user.	
<b>Special Requirements</b>	<ul style="list-style-type: none"> <li>Implementation of a user-friendly search interface with intuitive controls and clear instructions for specifying search criteria.</li> <li>Support for filtering and sorting options to enable users to customize search results based on their preferences.</li> </ul>	
<b>Variations in Technology and Data</b>	Implementation of responsive design principles.	
<b>Frequency of Occurrence</b>	Hostel search activities occur frequently as students and job holders explore accommodation options based on their changing needs and preferences.	
<b>Miscellaneous</b>	Regular monitoring and maintenance of the system to address any technical issues	

## 2.4.5. MANAGING STUDENT INFORMATION

<b>Use Case 5</b>		
<b>Use Case ID:</b>	5	
<b>Name</b>	Managing Student Information	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Owner/Manager	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Hostel Owners: Interested in accessing and managing information about the people residing in their hostel.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• Hostel management system is accessible.</li> <li>• Students and job holders have registered accounts with valid credentials.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Hostel owners can successfully access and view information about the residents in their hostel, facilitating effective management and communication.</li> <li>• Student information is accurately displayed.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Action</b>	<b>System Responsibility</b>
	1. Owner navigates to the student information option  3. Owner selects a specific student from the list of all students/residents in the hostel	2. System provides access to the user to student information feature  4. System displays the selected student's personal and contact information, fee details and all other information

	5. Owner may update/change any detail/information	6. Gives access to user to make changes in student information.
<b>Extensions</b>	<ul style="list-style-type: none"> <li>If the requested student information is not found in the system, the owner is notified</li> <li>In case of technical issues or errors during information retrieval, the system provides error messages</li> </ul>	
<b>Special Requirements</b>	<ul style="list-style-type: none"> <li>Support for easy navigation and intuitive user interface design to facilitate efficient access and management of student information.</li> </ul>	
<b>Variations in Technology and Data</b>	Implementation of responsive design principles.	
<b>Frequency of Occurrence</b>	Hostel owners access student information regularly to track resident status, manage accommodation allocations, and ensure effective communication with residents.	
<b>Miscellaneous</b>	Continuous monitoring and updating of resident information to reflect any changes or updates.	

## 2.4.6. ADVERTISING HOSTEL

<b>Use Case 6</b>	
<b>Use Case ID:</b>	6
<b>Name</b>	Advertising Hostel
<b>Scope</b>	Hostel Management System
<b>Level</b>	User Goal
<b>Primary Actor</b>	Hostel Owner/Manager
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Hostel Owner/Manager wants to promote their hostel to bring more customers.</li> </ul>

	<ul style="list-style-type: none"> <li>• Hostel Residents seek information about available hostels.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The Hostel Owner must be logged in.</li> <li>• The Hostel Owner has access to the advertisement creation interface.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The advertisement is successfully posted.</li> <li>• The residents can view the advertised hostel.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Actions</b> 1. The hostel Owner selects "Create New Advertisement." 2. Hostel Owner fills out the advertisement details (e.g., hostel name, location, facilities, pricing, contact information). 3. The hostel Owner uploads images or videos to enhance the advertisement. 4. The hostel Owner reviews and confirms the advertisement.  7. Potential residents can view the advertisement and contact the hostel owner for inquiries.	<b>System Responsibilities</b>  5. The system validates the advertisement details. 6. Advertisement is posted on the platform.
<b>Extensions</b>		
<b>Special Requirements</b>	The advertisement creation interface should support multimedia content upload (images, videos).	
<b>Variations in Technology and Data</b>	A variety of image and video formats need to be supported by the advertisement system.	
<b>Frequency of Occurrence</b>	Owners of hostels may publish advertisements often, particularly during periods of high demand for accommodation.	
<b>Miscellaneous</b>	Residents could rate and review hostels that are advertised by hostel owners.	

## 2.4.7. COMPARING HOSTELS

<b>Use Case 7</b>		
<b>Use Case ID:</b>	7	
<b>Name</b>	Comparing Hostels	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	User Goal	
<b>Primary Actor</b>	Students/Residents	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Student: To make better decisions about where to stay, they want to compare hostels according to their facilities and features.</li> <li>Hostel Owners: Provide information about their hostels to attract potential residents.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>The Student/Resident must be logged in.</li> <li>The Hostel Owner has access to the advertisement creation interface.</li> </ul>	
<b>Postconditions</b>	The student compares hostels successfully and comes to a well-informed conclusion.	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>
	<ol style="list-style-type: none"> <li>1. The student navigates to the hostel comparison section.</li> <li>2. The student selects the criteria for comparison (e.g., location, price, facilities, amenities).</li> <li>3. The student specifies any preferences or filters (e.g., within a certain distance from campus, budget constraints).</li> <li>5. The student selects two or more hostels to compare.</li> </ol>	<ol style="list-style-type: none"> <li>4. The system retrieves a list of hostels matching the specified criteria.</li> </ol>

	<p>7. Students review the comparison results, paying attention to factors important to their accommodation needs.</p> <p>8. The student makes an informed decision based on the comparison and takes appropriate action (e.g., contacting the preferred hostel for booking).</p>	<p>6. The system displays a side-by-side comparison of the selected hostels, highlighting their respective facilities and amenities.</p>
<b>Extensions</b>	<ul style="list-style-type: none"> <li>The system notifies the student and proposes extending the search parameters if no hostels meet the set requirements.</li> </ul>	
<b>Special Requirements</b>	<ul style="list-style-type: none"> <li>Many comparison factors, such as location, cost, amenities, features, and reviews/ratings, should be supported by the comparison feature.</li> <li>The system should ensure accurate and up-to-date information about hostels to provide reliable comparison results.</li> </ul>	
<b>Variations in Technology and Data</b>	Efficient search techniques should be used by the system to immediately retrieve and compare hostel information.	
<b>Frequency of Occurrence</b>	Students may use the hostel comparison feature regularly, especially during peak enrollment periods or when seeking new accommodation options.	
<b>Miscellaneous</b>	Integration with mapping services or transportation information could enhance the comparison process by considering accessibility to campus or public transportation routes.	

## 2.4.8. MANAGING COMPLAINTS AND MAINTENANCE REQUESTS

<b>Use Case 8</b>		
<b>Use Case ID:</b>	8	
<b>Name</b>	Managing Complaints and Maintenance Requests	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	User Goal	
<b>Primary Actor</b>	Students/Residents	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Student/Resident: want to report problems or make maintenance requests to better their living conditions as a student or employee.</li> <li>Hostel Owner/Manager: To guarantee resident happiness, must respond quickly to complaints and maintenance requests.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>The Student/Job Holder is logged into the hostel management system.</li> <li>The Student/Job Holder can easily access the complaints and maintenance submission interface.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>The complaint or maintenance request is successfully submitted and stored in the system.</li> <li>Hostel Owner/Manager receives notification of complaints and maintenance requests.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>
	1. Student/Job Holder navigates to the complaints and maintenance section. 2. Student/Job Holder fills out the complaint/maintenance request form, including details such as Type of issue (e.g., cleanliness, facilities, utilities) and Description of the problem. 3. Student/Job Holder attaches any relevant photos or documents to support the	

	<p>complaint or maintenance request.</p> <p>4. Student/Job Holder reviews and confirms the submission.</p> <p>6. The hostel Owner/Manager receives notification about the new submission.</p>	<p>5. The system validates the submission and stores the complaint/maintenance request in the database.</p>
<b>Extensions</b>	<ul style="list-style-type: none"> <li>• If required fields are missing or invalid, the system prompts the Student/Job Holder to correct them.</li> <li>• If the submission fails validation, the system displays an error message and allows the Student/Job Holder to correct the issues.</li> </ul>	
<b>Special Requirements</b>	<ul style="list-style-type: none"> <li>• To prioritize responses, distinguish between urgent requests and those that are not.</li> <li>• The system assists students and job holders in tracking the progress of their applications by offering a feedback tool.</li> </ul>	
<b>Variations in Technology and Data</b>	The system should support the attachment of various file formats (e.g., images and documents) to the complaint/maintenance request.	
<b>Frequency of Occurrence</b>	Complaints and maintenance requests may be submitted regularly, depending on the condition and maintenance of the hostel facilities.	
<b>Miscellaneous</b>	Establish a tracking system to keep an eye on the progress and outcome of complaints and maintenance requests that have been made.	

## 2.4.9. SUBMIT FEEDBACK AND REVIEWS

<b>Use Case 9</b>	
-------------------	--

<b>Use Case ID:</b>	9	
<b>Name</b>	Submit feedback and reviews	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	User Goal	
<b>Primary Actor</b>	Students/Residents	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Student/Job Holder: Wants to provide feedback or review to share their experience and suggest improvements.</li> <li>• Hostel Owner/Manager: Look for reviews and feedback to figure out how satisfied the hostel's residents are and identify areas that need improvements.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The Student/Job Holder is logged into the hostel management system.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• The feedback or review is successfully submitted and stored in the system.</li> <li>• The hostel Owner/Manager receives notification of the submission and considers it for improvements.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Actions</b> <ol style="list-style-type: none"> <li>1. Student/Job Holder navigates to the feedback and reviews section.</li> <li>3. Student/Job Holder fills out the feedback/review form, including details such as Rating (e.g., on a scale of 1 to 5 stars), Title of the feedback/review, and Description of the experience.</li> <li>4. Student/Job Holder reviews and confirms the submission.</li> </ol>	<b>System Responsibilities</b> <ol style="list-style-type: none"> <li>2. The system opens a form for submitting new feedback or reviews.</li> <li>5. The system validates the submission and stores the feedback/review in the database.</li> </ol>

	<p>6. The hostel Owner/Manager receives a notification about the new feedback/review.</p>	<p>7. The system sends a notification to the Hostel Owner/Manager with details of the feedback or review.</p> <p>8. The system displays a confirmation message to the Student/Job Holder.</p>
<b>Extensions</b>	If required fields are missing or invalid, the system prompts the Student/Job Holder to correct them.	
<b>Special Requirements</b>	The system should provide an option for Student/Job Holders to submit feedback and reviews anonymously.	
<b>Variations in Technology and Data</b>	The system uses the notification system to notify the Hostel Owner/ Manager about feedback.	
<b>Frequency of Occurrence</b>	Residents may submit reviews and feedback regularly.	
<b>Miscellaneous</b>	Provide a system that residents can easily show their reviews and feedback.	

## 2.4.10. REGISTERING STUDENT

<b>Use Case 11</b>	
<b>Use Case ID:</b>	11
<b>Name</b>	Registering Student
<b>Scope</b>	Hostel Management System
<b>Level</b>	User Goal
<b>Primary Actor</b>	Students/Residents
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Student: Would like to set up a profile on the app to book a hostel.</li> <li>• Hostel Administrator: To oversee hostel accommodations, correct student information is required.</li> </ul>
<b>Preconditions</b>	The student has downloaded and installed the hostel management app.

<b>Postconditions</b>	The student's profile is successfully registered in the system.	
<b>Main Success Scenario</b>	<b>Actor Actions</b> <ol style="list-style-type: none"> <li>1. Student fills out the registration form with personal and contact information, such as: (Full name, Date of birth, Gender, Contact number, and Email address)</li> <li>2. The student reviews and confirms the registration details.</li> </ol>	<b>System Responsibilities</b> <ol style="list-style-type: none"> <li>3. The system validates the registration details.</li> <li>4. The system stores the student's profile in the database.</li> <li>5. The app saves the registration information in the system.</li> <li>6. The app displays a confirmation message to the student.</li> </ol>

<b>Extensions</b>	If the registration fails validation, the app displays an error message and allows the student to correct the issues.
<b>Special Requirements</b>	User-friendly interface for an easy registration process.
<b>Variations in Technology and Data</b>	Integration with student databases to streamline the registration process for existing students.
<b>Frequency of Occurrence</b>	Student registration may occur regularly, especially at the beginning of academic semesters or when new students join the hostel.
<b>Miscellaneous</b>	Implement a system to monitor and manage student registration status.

## 2.4.11. MANAGING FEE DETAILS

<b>Use Case 11</b>		
<b>Use Case ID:</b>	11	
<b>Name</b>	Manage Fee Details	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Students/Residents	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Hostel Owner/Manager: Wants to efficiently manage fee details for hostellites.</li> <li>• Hostel Residents: Want accurate and transparent fee information.</li> <li>• System Administrator: Needs to ensure the proper functioning of the fee management system.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>• The hostel owner/manager is logged into the system.</li> <li>• Fee details module is accessible and functional.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Fee details for hostel residents are accurately updated in the system.</li> <li>• Hostel residents can view their fee details.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>

	<p>1. Hostel owner/manager accesses the fee details module</p> <p>3. Owner selects a hostel resident to manage their fee details</p> <p>4. Owner updates the fee amount</p> <p>9. Hostel resident can view the updated fee details.</p>	<p>2. System presents interface to manage fee details</p> <p>5. System retrieves resident information and fee details</p> <p>6. System updates fee amount for the selected resident</p> <p>7. System confirms successful update</p> <p>8. System presents updated fee details to the resident</p>
<b>Extensions</b>	1. If the owner encounters an error while updating fee details, an error message is displayed. 2. If the owner wants to add new fee categories or modify existing ones, they can do so through the system's settings.	
<b>Special Requirements</b>	1. Secure authentication for hostel owner/manager login. 2. The system should provide options for different types of facilities and corresponding fee structures.	
<b>Variations in Technology and Data</b>	Various data formats for fee details (e.g., currency, date).	
<b>Frequency of Occurrence</b>	Daily, as hostel owners/managers need to manage fee details regularly	

<b>Miscellaneous</b>	Feedback mechanism for hostel residents to report discrepancies in fee details.
----------------------	---

## 2.4.12. AVAILING TRANSPORTATION SERVICES

<b>Use Case 12</b>		
<b>Use Case ID:</b>	12	
<b>Name</b>	Avail Transportation Services	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Students/Job Holders	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Student/Job Holder: Wants to avail transportation services conveniently based on their schedule.</li> <li>• Hostel Administrator: Needs to manage transportation schedules and ensure availability.</li> <li>• Transport Service Provider: Responsible for operating transportation services efficiently.</li> </ul>	
<b>Preconditions</b>	<ol style="list-style-type: none"> <li>1. Student/Job Holder is registered with the hostel management system.</li> <li>2. Transportation services are available and operational.</li> </ol>	
<b>Postconditions</b>	<ol style="list-style-type: none"> <li>1. Student/Job Holder successfully avails transportation services according to their schedule.</li> <li>2. Transportation service provider receives necessary information about passengers and schedules.</li> </ol>	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>
	1. They navigate to the transportation services section.  3. They select the desired date and time for transportation.	2. Hostel management system presents transportation module

	<p>5. User views available transportation options 6. They select a transportation option.  8. They confirm the booking and provides necessary details.  10. They board the transportation at the designated time and location.</p>	<p>4. System retrieves available transportation options  7. System records the selected transportation booking  9. System presents booking confirmation and details  11. System ensures transportation availability and readiness</p>
<b>Extensions</b>	If the desired transportation option is unavailable, the system notifies the user of unavailability.	
<b>Special Requirements</b>	Ability to handle peak demand periods and accommodate last-minute bookings.	
<b>Variations in Technology and Data</b>	Data formats for storing passenger information, schedules, and route details.	
<b>Frequency of Occurrence</b>	Daily, as students/job holders need transportation services regularly based on their schedule.	
<b>Miscellaneous</b>	Feedback mechanism for users to provide suggestions or report issues related to transportation services	

## 2.4.13. ALLOCATING ROOMS

<b>Use Case 13</b>	
<b>Use Case ID:</b>	13
<b>Name</b>	Allocate Rooms

<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Owner/Administrator	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Hostel Owner/Administrator: Wants to efficiently allocate rooms.</li> <li>• Hostel Residents: Expect fair and suitable room allocations according to their preferences and budget.</li> <li>• Hostel Staff: Ensuring an easy stay at the hostel.</li> </ul>	
<b>Preconditions</b>	Rooms are available for allocation.	
<b>Postconditions</b>	Rooms are allocated to hostel residents based on their preferences and availability.	
<b>Main Success Scenario</b>	<b>Actor Actions</b> 1. Hostel owner/administrator accesses the room allocation section of the hostel management system.  3. They view a list of available rooms along with their prices, types and current occupancy status. 4. Hostel owner/administrator selects a suitable room for allocation.  6. They confirm the room allocation and updates the room's occupancy status.	<b>System Responsibilities</b>  2. System retrieves and displays available room.  5. System marks the selected room as allocated  7. System updates the room's occupancy status 5. System notifies the resident of the allocated room details

<b>Extensions</b>	In case of changes in room availability, the system updates the rooms data accordingly.
<b>Special Requirements</b>	Updating of room availability and occupancy status to avoid double bookings
<b>Variations in Technology and Data</b>	Use of database systems to store room information, including types, prices, and availability.
<b>Frequency of Occurrence</b>	Room allocations occur whenever new residents join the hostel or existing residents request changes.
<b>Miscellaneous</b>	Feedback mechanism for residents to provide input on room allocation processes and suggest improvements.

## 2.4.14. MESSAGING AND COMMUNICATION

<b>Use Case 14</b>		
<b>Use Case ID:</b>	14	
<b>Name</b>	Messaging and Communication	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Owner/Administrator, Students	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Hostel Owner/Administrator: Wants to communicate important announcements, updates, and notices to hostel residents efficiently.</li> <li>Students: Want to receive information and communicate with hostel management regarding various issues or concerns.</li> </ul>	
<b>Preconditions</b>	Hostel management system is accessible to both hostel owners/administrators and students.	
<b>Postconditions</b>	Effective communication between hostel management and students is facilitated.	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>
	1. Hostel owner/administrator accesses the messaging section of the hostel management system. 2. They compose a message.	

	<p>3. They send the message through the system.</p> <p>5. Students receive the message in the hostel management app and can view the message and respond.</p> <p>6. Hostel owner/administrator can view responses and engage in further communication if required.</p>	<p>4. System sends the message to the intended recipients</p> <p>7. System allows students to view and respond to the message</p> <p>8. System enables the owner/administrator to view responses and engage further if needed.</p>
<b>Extensions</b>	In case of technical issues users are informed of the disruption.	
<b>Special Requirements</b>	Integration with notification systems to notify users of new messages.	
<b>Variations in Technology and Data</b>	Storage and retrieval of message history for reference.	
<b>Frequency of Occurrence</b>	Can occur regularly between hostel owners/administrators and students, depending on the need for announcements and updates.	
<b>Miscellaneous</b>	Feedback mechanisms for users to provide input on the messaging system's usability and functionality.	

## 2.4.15. MANAGING LEAVE REQUESTS

<b>Use Case 15</b>	
<b>Use Case ID:</b>	15
<b>Name</b>	Managing Leave requests

<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Students, Hostel Managers/Owners	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>• Students: Want to request permission to leave the hostel.</li> <li>• Hostel Managers/Owners: Responsible for managing and approving leave requests.</li> </ul>	
<b>Preconditions</b>	Hostel management system is accessible to both students and hostel managers/owners.	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>• Students can submit leave requests with reasons.</li> <li>• Hostel managers/owners can review and manage leave requests efficiently.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>
	<ol style="list-style-type: none"> <li>1. Student accesses the leave management section of the hostel management system.</li> <li>2. They initiate a leave request, providing necessary details.</li> <li>3. Student submits the leave request through the system.</li>   <li>5. Hostel manager/owner receives the request and accesses the system.</li> <li>6. Manager/owner approves or denies the leave request through the system.</li>   <li>8. Student receives notification of the decision on their leave request. If approved, student is allowed to leave the hostel. If denied, student is informed of the decision</li> </ol>	<ol style="list-style-type: none"> <li>4. System records and forwards the leave request</li>   <li>7. System processes the decision and notifies the student</li> </ol>

	and may resubmit the request if necessary.	
<b>Extensions</b>	In case of technical issues users are informed of the disruption.	
<b>Special Requirements</b>	Leave request forms should include fields for students to provide detailed reasons for their leave	
<b>Variations in Technology and Data</b>	Integration with notification systems to inform users of leave request status updates.	
<b>Frequency of Occurrence</b>	Leave requests may occur regularly, depending on students' individual needs and circumstances.	
<b>Miscellaneous</b>	Providing students with visibility into the status of their leave requests.	

## 2.4.16. MANAGING FOOD DETAILS

<b>Use Case 16</b>		
<b>Use Case ID:</b>	16	
<b>Name</b>	Managing Food Details	
<b>Scope</b>	Hostel Management System	
<b>Level</b>	Subfunction	
<b>Primary Actor</b>	Hostel Manager	
<b>Stake Holders and Interests</b>	<ul style="list-style-type: none"> <li>Hostel Manager: Wants to manage and update food menus efficiently.</li> <li>Hostel Residents: Interested in knowing the daily or weekly food menus.</li> </ul>	
<b>Preconditions</b>	<ul style="list-style-type: none"> <li>The hostel management system is accessible to the hostel manager.</li> <li>Initial food menus are defined and available in the system.</li> </ul>	
<b>Postconditions</b>	<ul style="list-style-type: none"> <li>Hostel Manager can update and manage food menus effectively.</li> <li>Hostel residents can view the daily or weekly food menus through the system.</li> </ul>	
<b>Main Success Scenario</b>	<b>Actor Actions</b>	<b>System Responsibilities</b>

	<p>1. Hostel manager accesses the food details management section of the hostel management system.</p> <p>2. Manager initiates the process to update the food menu.</p> <p>3. Manager modifies or adds food items to the menu.</p> <p>5. Manager saves the updated food menu in the system.</p>	<p>4. System records the changes to the food menu</p> <p>6. System updates the food menu in the database and displays the updated food menu.</p>
<b>Extensions</b>	If there are special dietary requirements or requests from hostel residents, manager may accommodate them by adjusting the menu as needed	
<b>Special Requirements</b>	User-friendly interface for hostel managers to easily navigate and update food menus.	
<b>Variations in Technology and Data</b>	Use of calendar integration to schedule and display menus for specific dates.	
<b>Frequency of Occurrence</b>	Food menus may need to be updated daily or weekly, depending on the meal planning schedule of the hostel.	
<b>Miscellaneous</b>	Providing residents with the ability to provide feedback or suggestions on food menus through the system.	

## 2.5 USE CASE DIAGRAM



Figure 1 Use Case Diagram

## 3. OTHER NONFUNCTIONAL REQUIREMENTS

### 3.1 PERFORMANCE REQUIREMENTS

- **System Response Time:** All system operations, including room allocation, leave request management, maintenance and complaint handling, advertisement posting, and hostel searching, must be completed within 2-3 seconds.
- **Scalability:** The system should handle up to 10,000 concurrent users without significant performance degradation.
- **Database Transactions:** All database transactions, including insertions, updates, and deletions, must be completed within 1 second to ensure data integrity and consistency.
- **Search Functionality:** Search results should be returned within 2 seconds, even with a large dataset of hostels.

### 3.2 SAFETY REQUIREMENTS

- **Data Backup:** All critical data, including room allocations, leave requests, complaints, and advertisements, must be backed up daily to prevent data loss.
- **Data Recovery:** Implement a robust data recovery mechanism to restore data within 4 hours in case of system failure.
- **Failover Mechanism:** The system should have a failover mechanism to ensure continuous operation in case of server failure.

### 3.3 SECURITY REQUIREMENTS

- **User Authentication:** Implement multi-factor authentication for all users, including residents and owners, to ensure secure access.
- **Authorization:** Role-based access control must be implemented to restrict functionalities based on user roles (e.g., only owners can allocate rooms and post advertisements).
- **Data Encryption:** All sensitive data, including personal information and complaints, must be encrypted both in transit and at rest.
- **Audit Logs:** Maintain detailed audit logs of all system activities, including logins, data modifications, and user actions, for at least one year.
- **Vulnerability Management:** Regularly scan the system for vulnerabilities and apply necessary patches and updates promptly.

### 3.4 SOFTWARE QUALITY ATTRIBUTES

- **Reliability:** The system must operate reliably 24/7, with a minimum uptime of 99.9%.

- **Usability:** The user interface must be intuitive and easy to navigate for both residents and owners, with clear instructions and feedback for all operations.
- **Maintainability:** The system must be designed for easy maintenance, with clear documentation and modular code to facilitate updates and bug fixes.
- **Portability:** The system should be accessible from various devices, including desktops, laptops, tablets, and smartphones, and should be compatible with major operating systems and browsers.
- **Performance Efficiency:** The system should efficiently utilize resources, minimizing CPU, memory, and network usage.

## 3.5 BUSINESS RULES

- **Room Allocation:** Rooms can only be allocated if they are available. Only owners are authorized to allocate rooms.
- **Leave Request Management:** Residents must submit leave requests at least one week in advance. Owners must respond to leave requests within 3 business days.
- **Complaint Handling:** Complaints must be addressed within 48 hours. Owners must acknowledge receipt of complaints within 24 hours.
- **Advertisement Posting:** Advertisements must comply with hostel policies and standards. Only authorized personnel can create and edit advertisements.
- **Hostel Search:** Search results should prioritize available rooms and should be sorted by relevance and availability by default. Only residents can search for hostels.

## 3.6 OPERATING ENVIRONMENT

- Integrated Development Environment (IDE): Eclipse
- Database Management System (DBMS): MySQL
- Database Connector: MySQL Connector
- Icon Library: FontAwesome
- User Interface Library: JavaFX

## 3.7 USER INTERFACES

### 3.7.1 Login

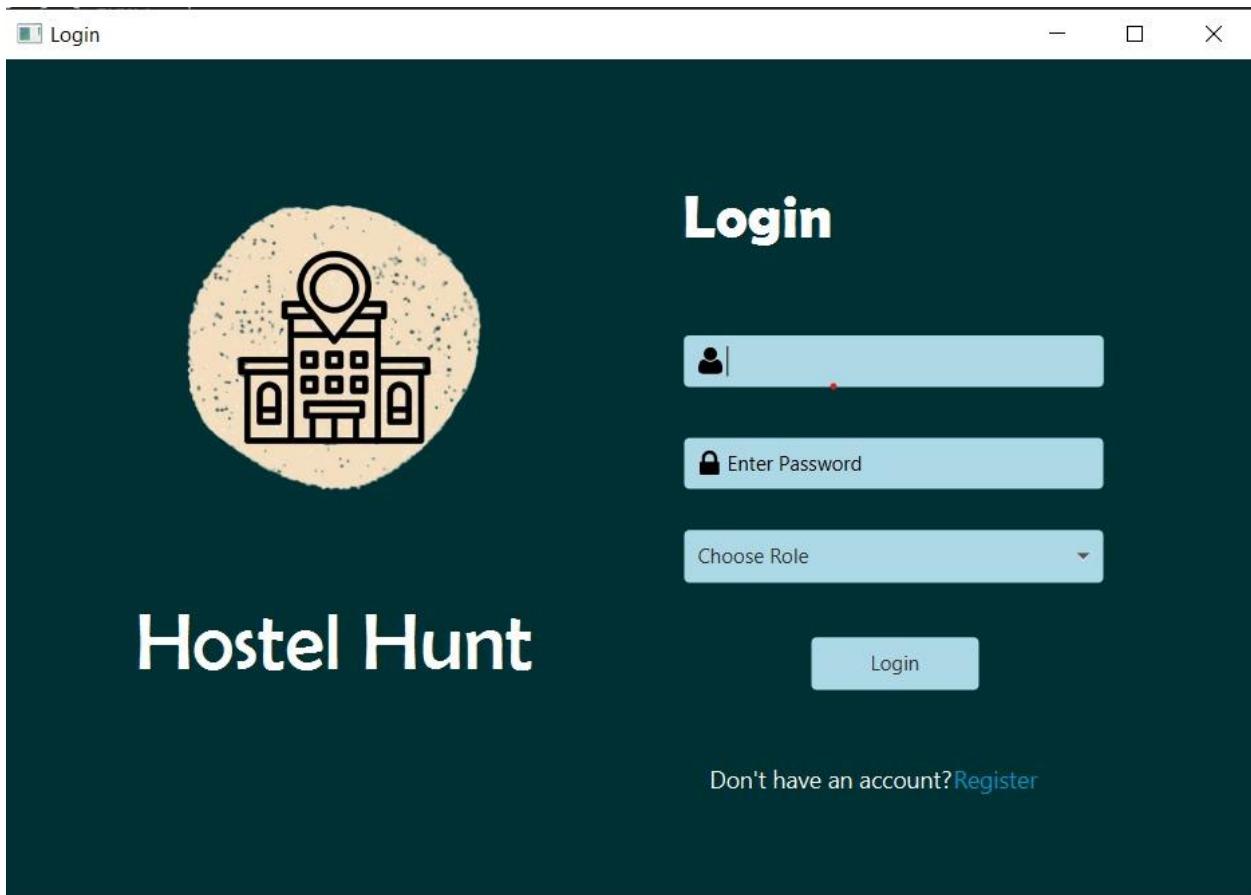
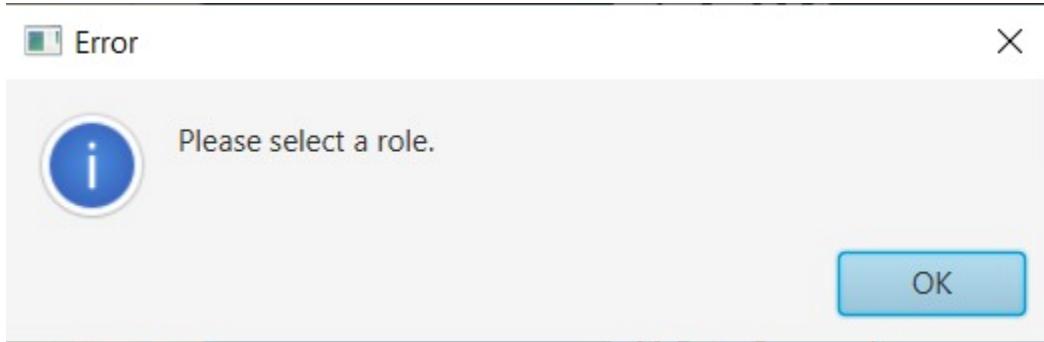


Figure 2 Hostel Hunt -Login

### 3.7.2 Login Error Message

In case of not selecting a role this error message will pop up.



### 3.7.3 Hostel Owner Dashboard

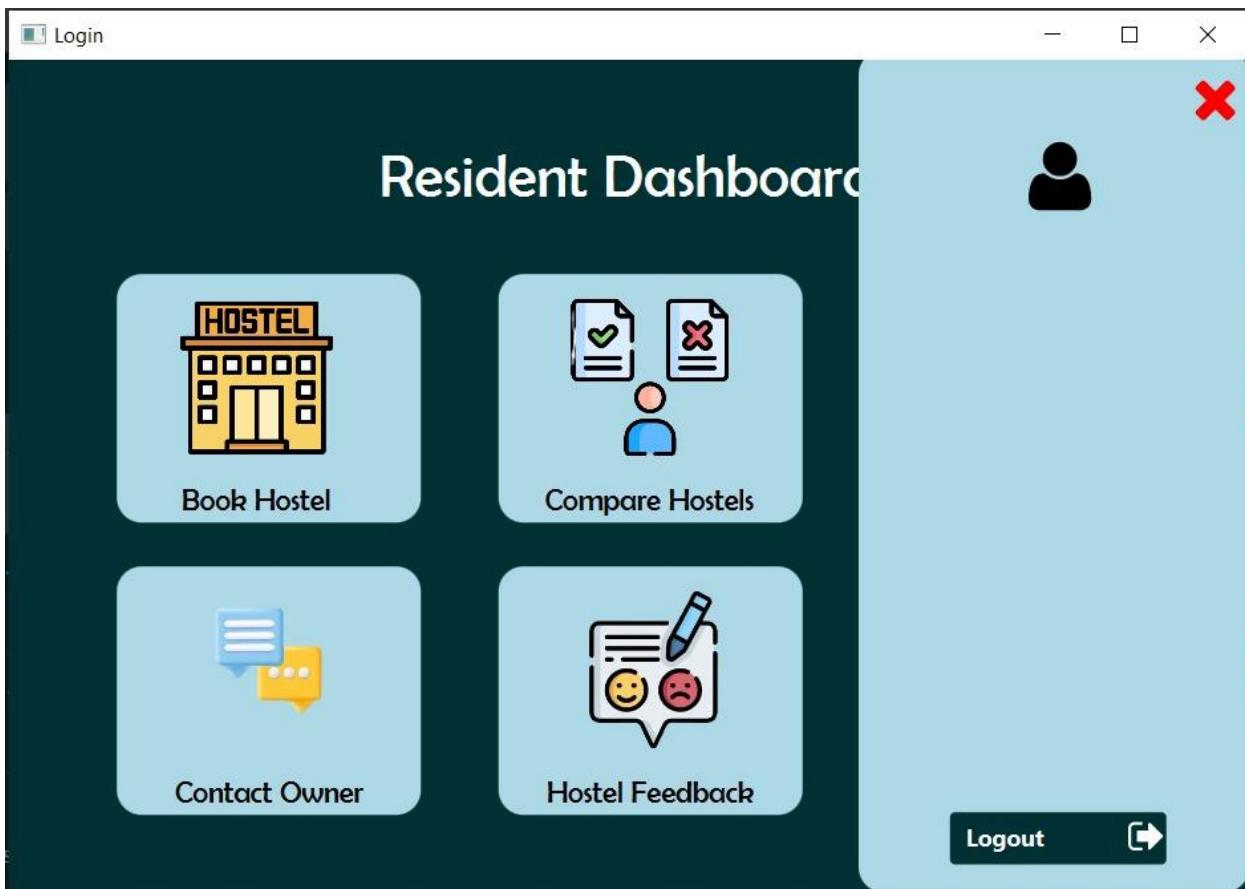
Hostel Owner is navigated to Hostel Owner Dashboard. They can always logout.



Figure 3 Hostel Hunt- Management Dashboard

### 3.7.4 Resident Dashboard

Resident is navigated to Resident Dashboard. They can always logout.



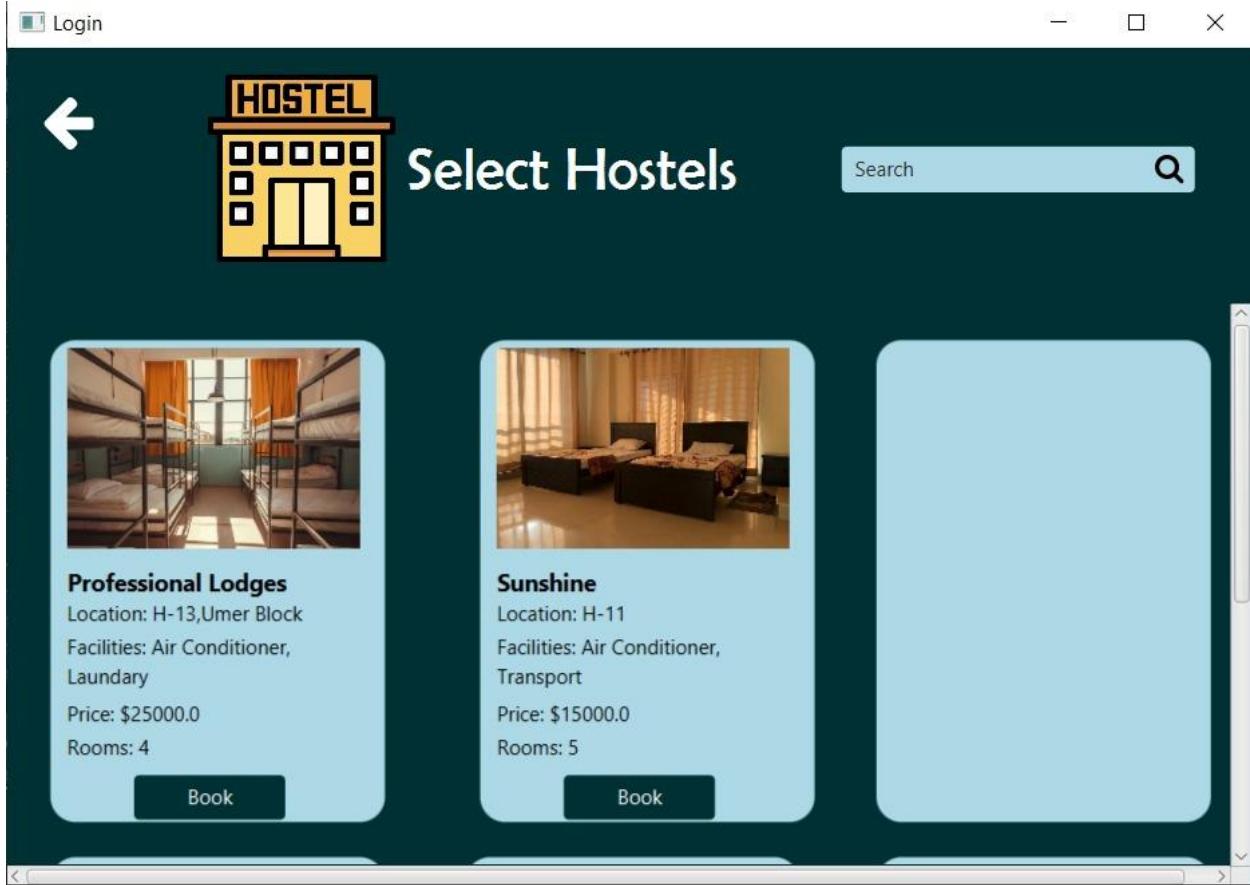
### 3.7.5 Search Hostel

Search hostel interface.

The screenshot shows a user interface titled "Search For Hostel". At the top left is a back arrow icon. At the top right are window control buttons for minimize, maximize, and close. Below the title is a "Login" button. The main area contains five input fields with icons: "Location" (location pin), "Price Range" (price tag), "Air Conditionor" (air conditioner), "Laundry" (laundry basket), and "Transport" (bus). A "Submit" button is located at the bottom center of the form.

### 3.7.6 Select Hostel

For booking process you can select Hostel.



### 3.7.7 Confirm Details

For confirming the booking details.

The screenshot shows a user interface titled "Confirm Details". At the top left is a "Login" button and a back arrow icon. To the right of the title is a blue envelope icon containing a white document with a green checkmark. The main area contains five input fields: "Full Name" (with a person icon), "Age" (with a calendar icon), "Email" (with an envelope icon), "Contact No." (with a phone receiver icon), and a dropdown menu labeled "Room Type". Below these fields is a large blue "Book" button.

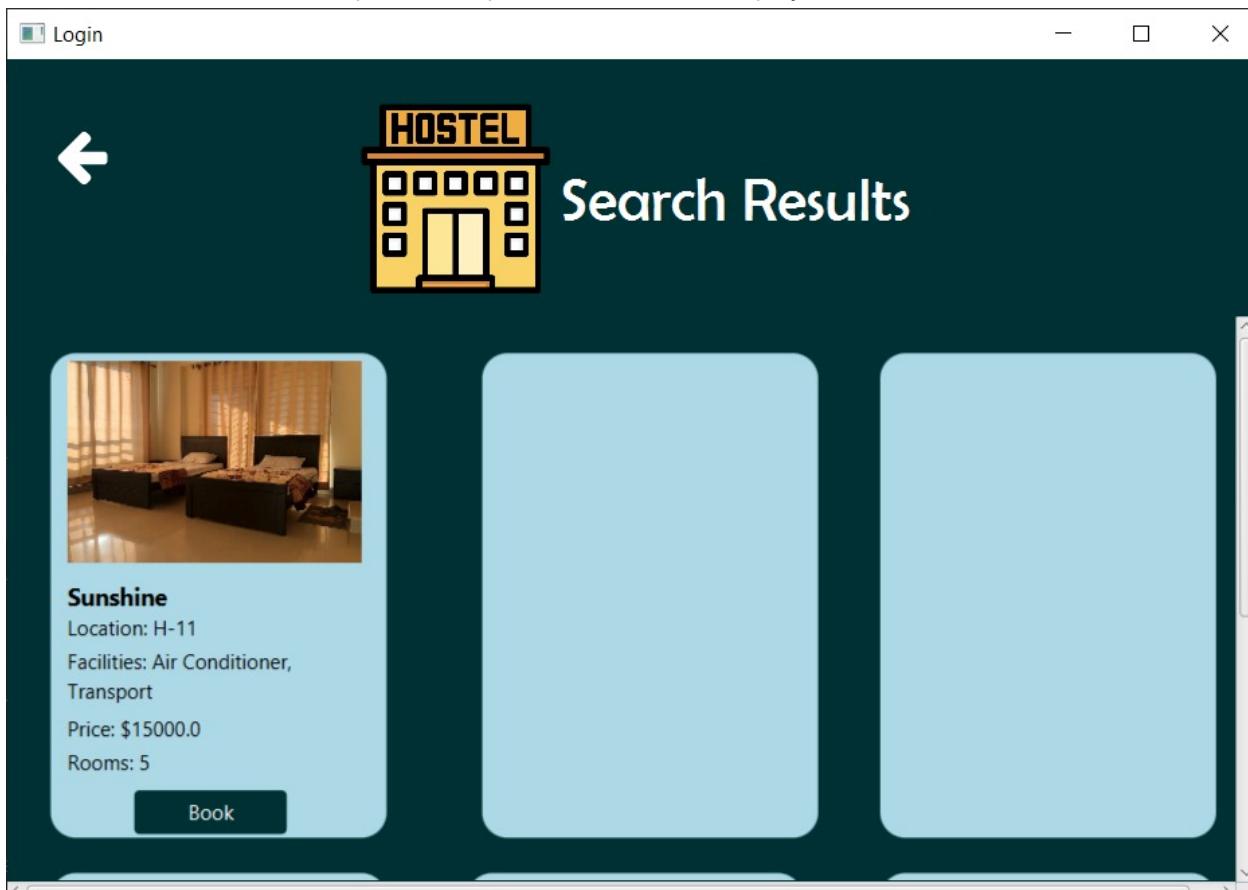
### 3.7.8 Success Message

Success message is shown for successful processes.



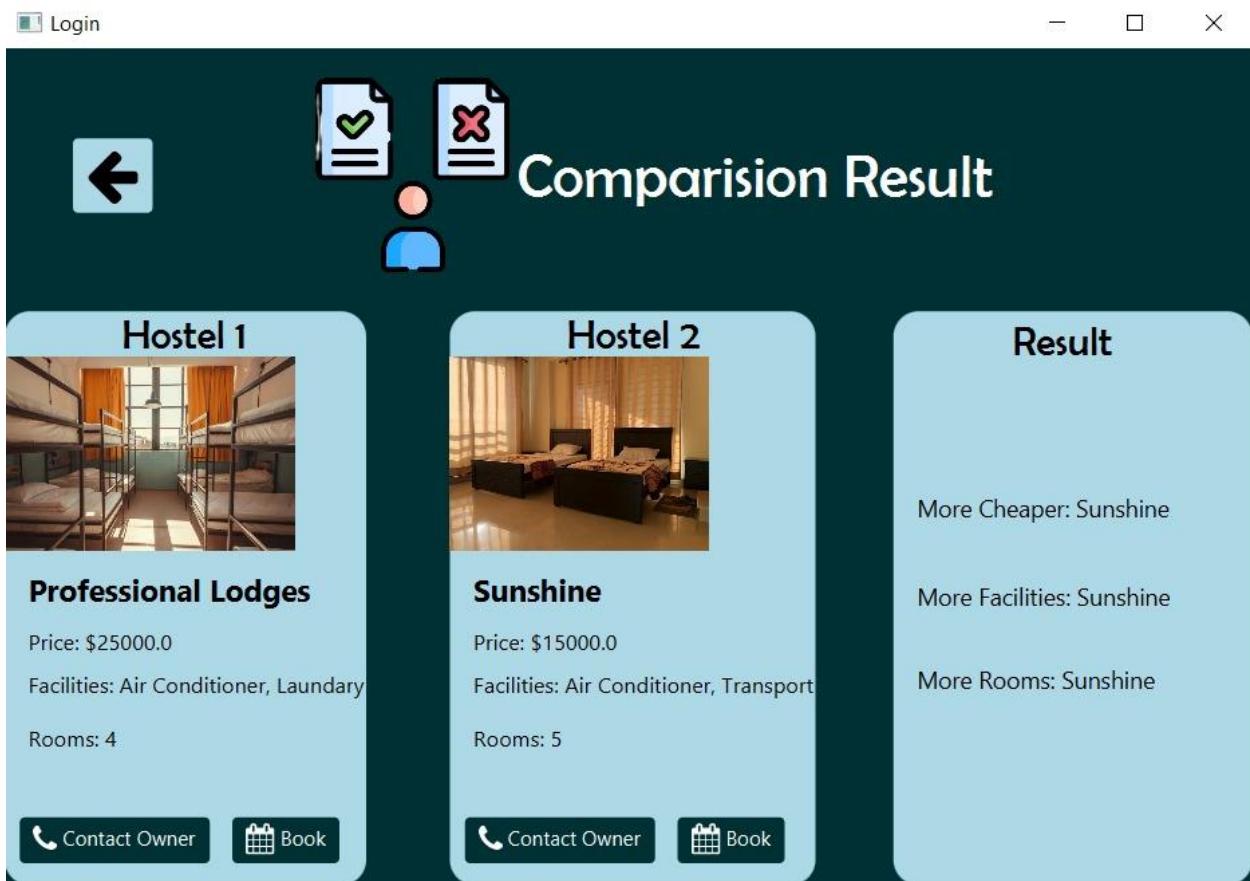
### 3.7.9 Search Hostel Results

When hostel is searched the specified requirements hostel is displayed.



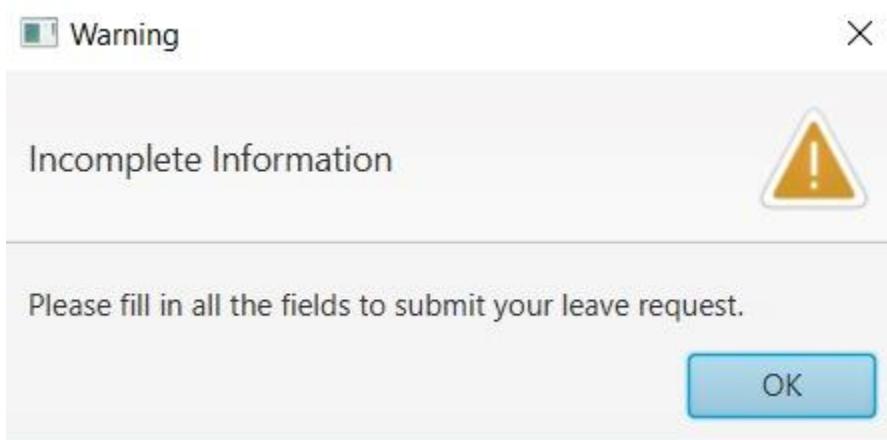
### 3.7.10. Comparison Result

When hostel comparison is done this is an example interface for search results.



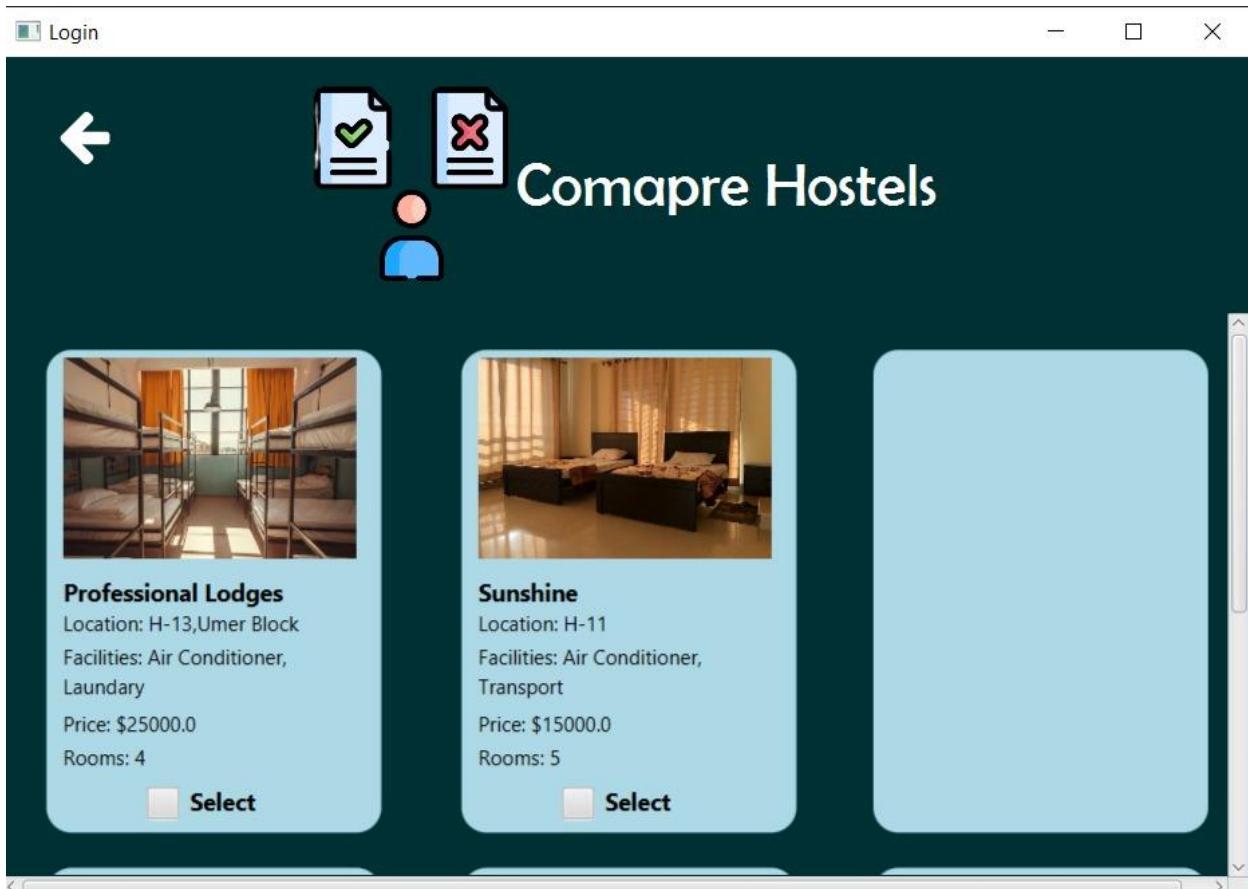
### 3.7.11. Warning Message

Incase of incomplete requirements there is warning message.



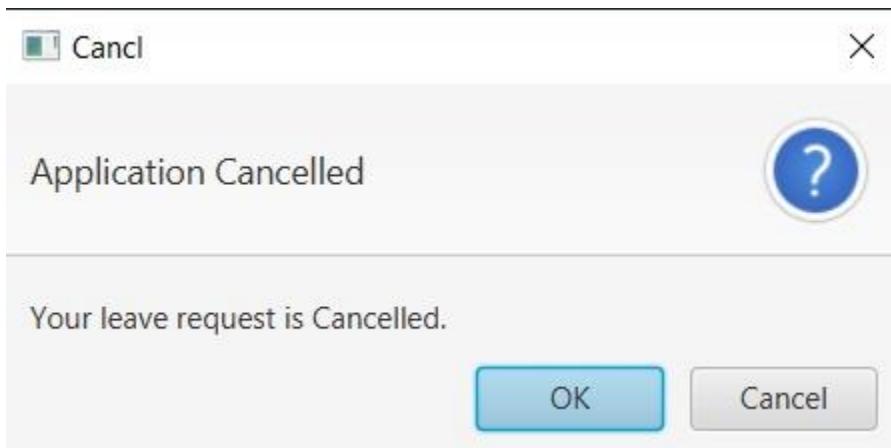
### 3.7.12 Compare Hostels

Compare Hostels Interface



### 3.7.13. Application Cancelled

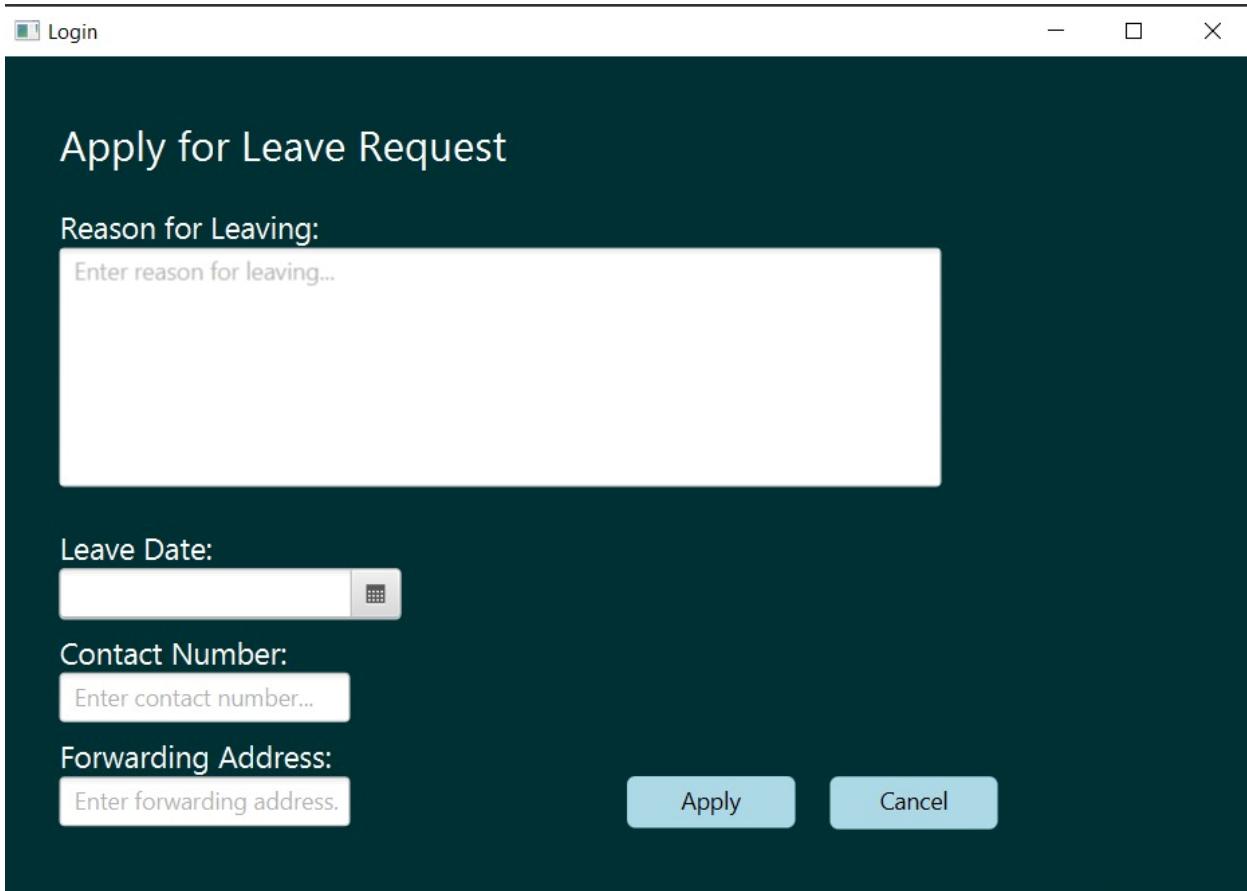
Incase application is cancelled.



### **3.7.14. Leave Request Application From**

The form where residents apply for leave.

---



Apply for Leave Request

Reason for Leaving:

Leave Date:

Contact Number:

Forwarding Address:

Enter reason for leaving...

Enter contact number...

Enter forwarding address...

Apply

Cancel

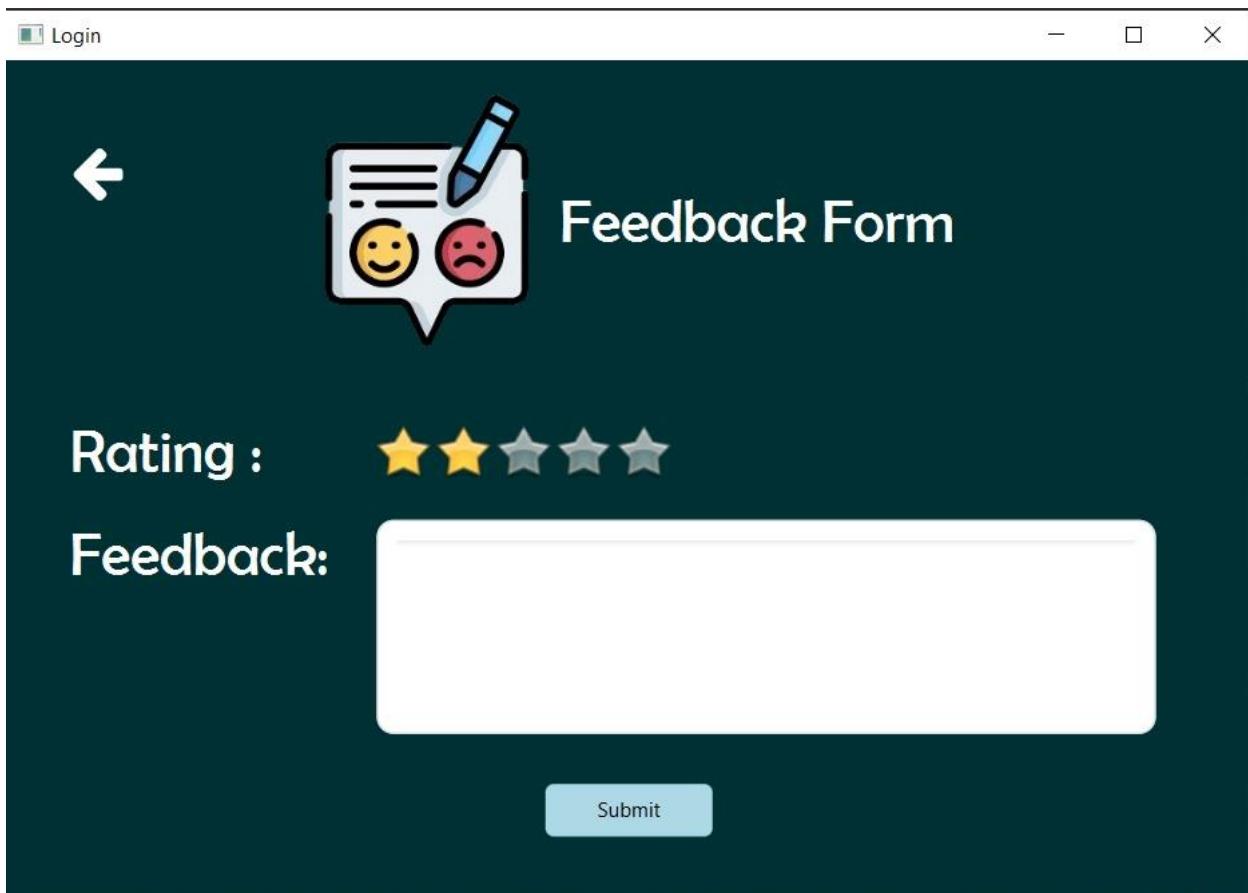
### **3.7.15. Error message**

Error message for no booked room.



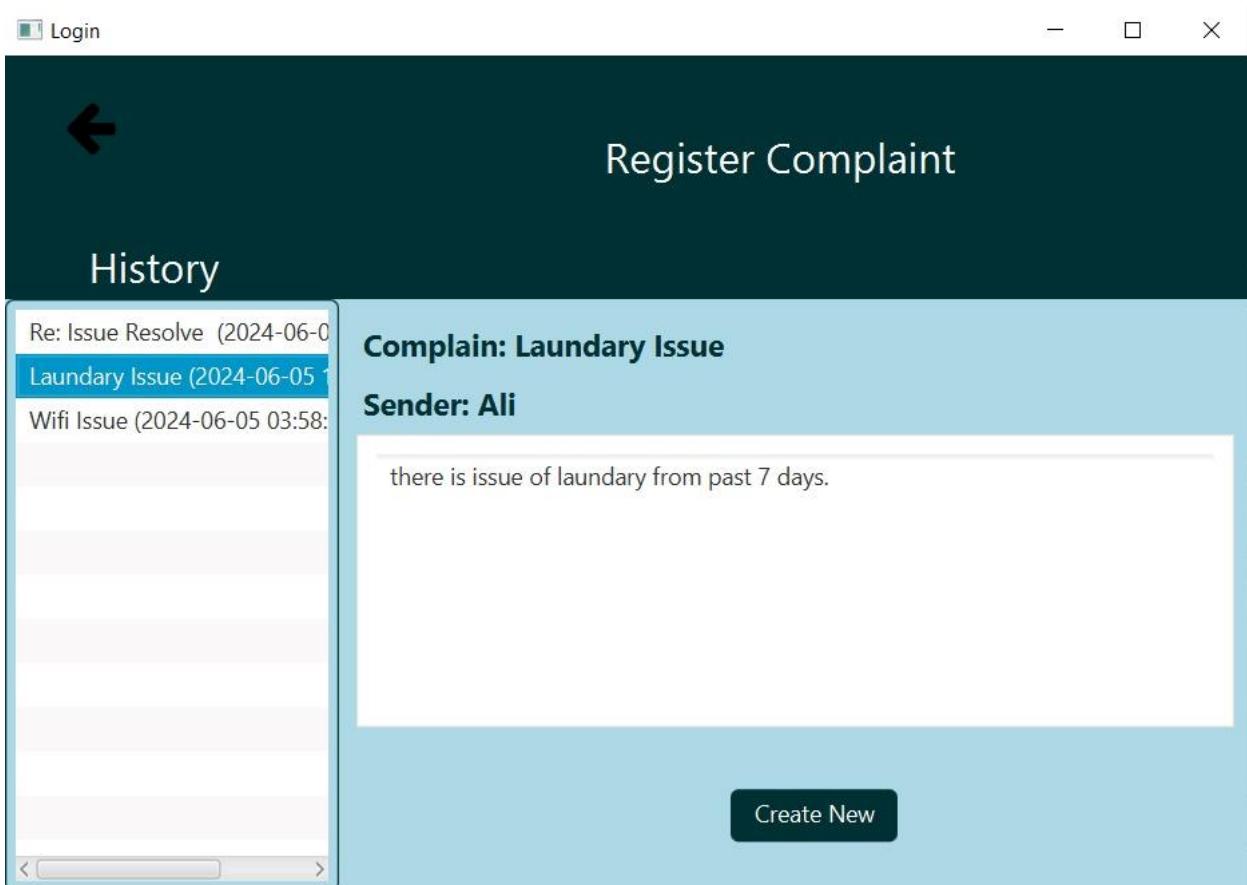
### **3.7.16. Feedback Form**

Feedback form interface.



### 3.7.17. Complaint Request

Here residents file complaints



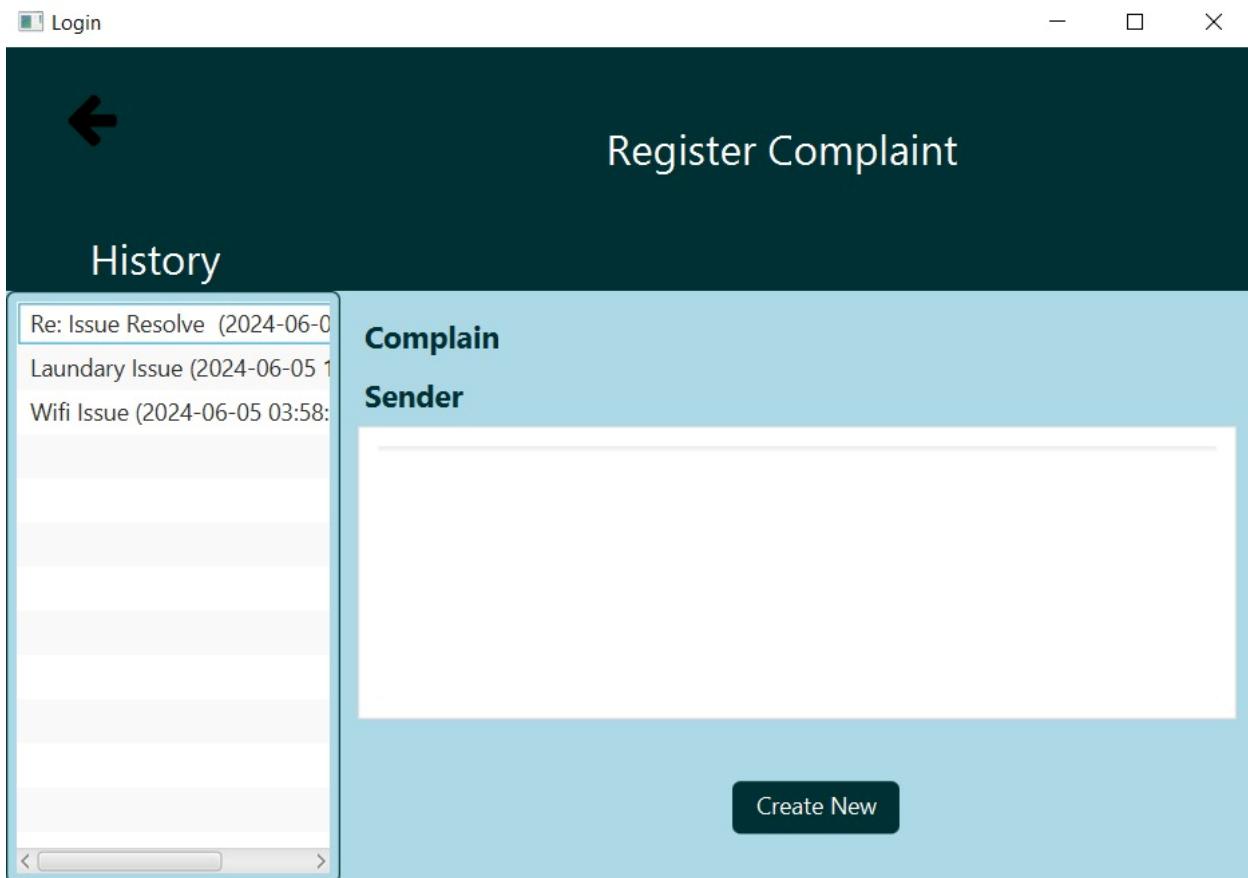
### ***3.7.18. Compose new complaint***

Residents file new complaint

The screenshot shows a window titled "Compose New Complain". The window has a light blue header bar with standard window controls (minimize, maximize, close) on the right. The main area contains three input fields: "Recipient" (with a placeholder icon), "Complain" (with a placeholder icon), and "Description" (a large text area). Below these fields is a dark blue "Send" button.

### 3.7.19. Complaint Recorded

The complaint that is sent can be seen in records.



### 3.7.20. Confirmation Dialog

Add advertisement confirmation dialog.



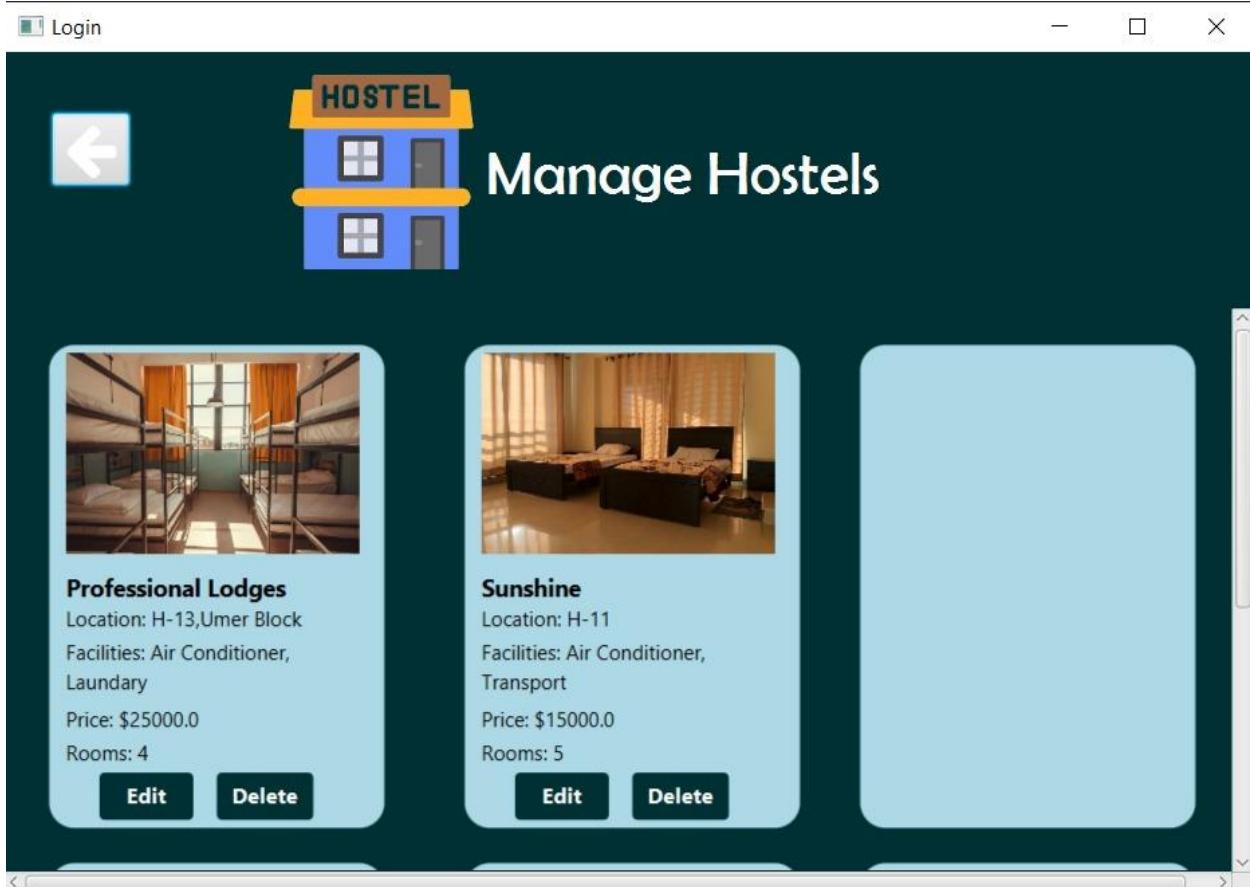
### 3.7.21. Advertise Hostel

Advertise Hostel Form.

The screenshot shows a Windows application window titled "Advertise Hostels". The window has a dark teal background. At the top left is a "Login" button and a back arrow icon. In the center is a yellow icon of a building labeled "HOSTEL". To the right of the icon is the title "Advertise Hostels". Below the title are two input fields: "Hostel Name" with a house icon and "Location" with a location pin icon. Underneath these are two dropdown menus: one showing "5" and another with a price symbol. To the left is a section titled "Facilities" with three checkboxes: "Air Conditioner" (checked), "Laundry" (unchecked), and "Transport" (unchecked). To the right is a large blue rectangular area labeled "Upload Images". At the bottom center is a blue "Submit" button.

### 3.7.22. Manage Hostel

Manage hostel interface where you can edit or delete advertisements.



### 3.7.23. Validation Error

Validation Error shown incase of incomplete requirements.



### 3.7.24. Leave Request Management

Leave requests are shown like this to the owner where he can reject or accept.

The screenshot shows a software application window titled "Leave Requests". At the top left is a "Login" button and at the top right are standard window control buttons (minimize, maximize, close). A back arrow icon is on the left side of the title bar. The main content area displays three leave requests:

Id = 1 - Resident Id = 1 - Reason = i am shifting to new city - Status = Approved
Id = 2 - Resident Id = 1 - Reason = Moving abroad for further studies - Status = Pending
Id = 3 - Resident Id = 1 - Reason = checking system - Status = Pending

Below this, under "Leave Request Details", are the following fields:

Resident ID: 1  
Reason: Moving abroad for further studies  
Leave Date: 2025-07-12  
Contact Number: 09876544567  
Forwarding Address: not confirmed  
Status: Pending

Approval Comment:

Approve      Reject

### 3.7.25. Room Allocation

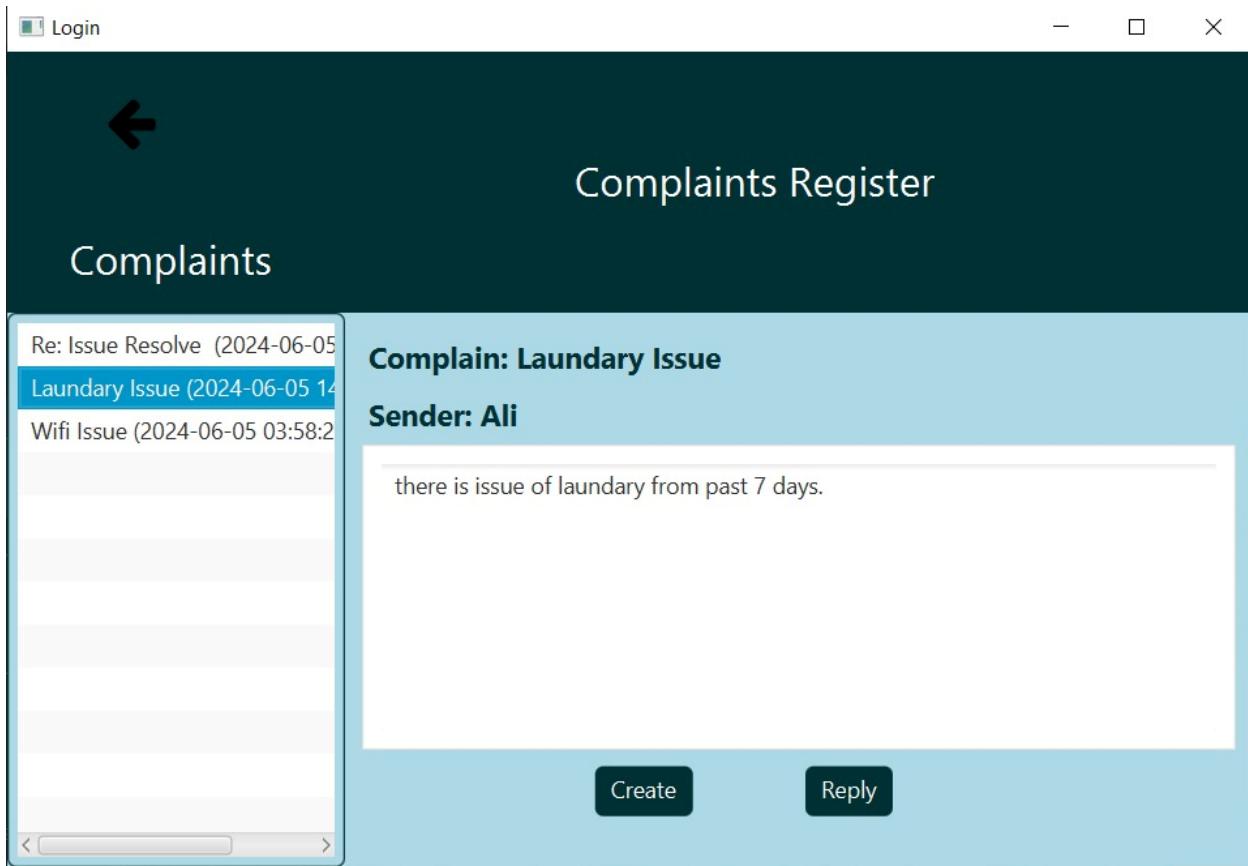
Room allocation for management

The screenshot shows a software application window titled "Room Allocation". At the top left is a back arrow icon. The main area contains a table with four columns: "Room Type", "Available Rooms", and "Availability" (which is highlighted in grey). To the left of the table are four dropdown menus: "Select Hostel", "Room Type", "No. of Rooms", and "Availability", each with a downward arrow indicating they are dropdowns. Below these dropdowns is a button labeled "Add". The table body is light blue and displays the message "No content in table". The top right corner of the window has standard window control buttons for minimize, maximize, and close.

Room Type	Available Rooms	Availability
No content in table		

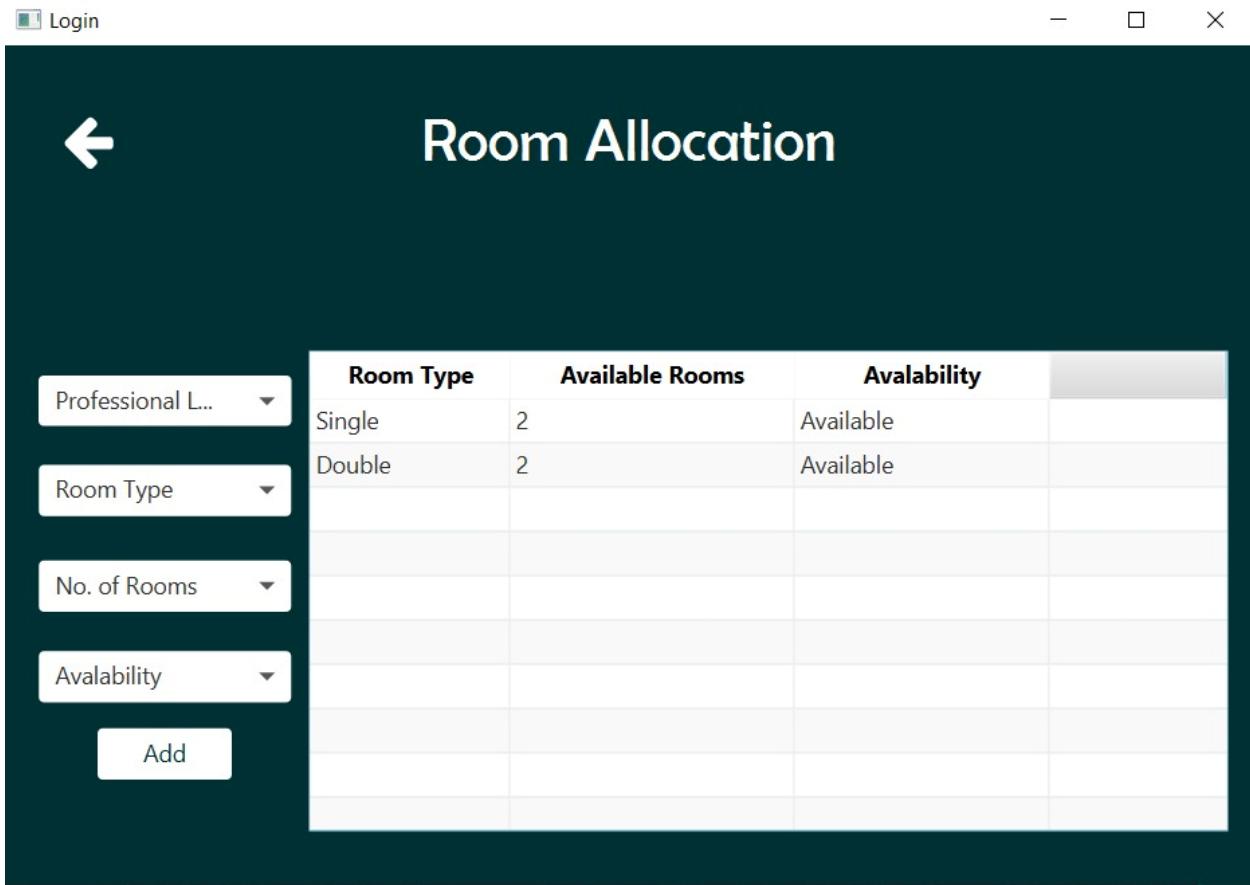
### 3.7.26. Complaints Management

Manager can view the complaints and read the complaint message.



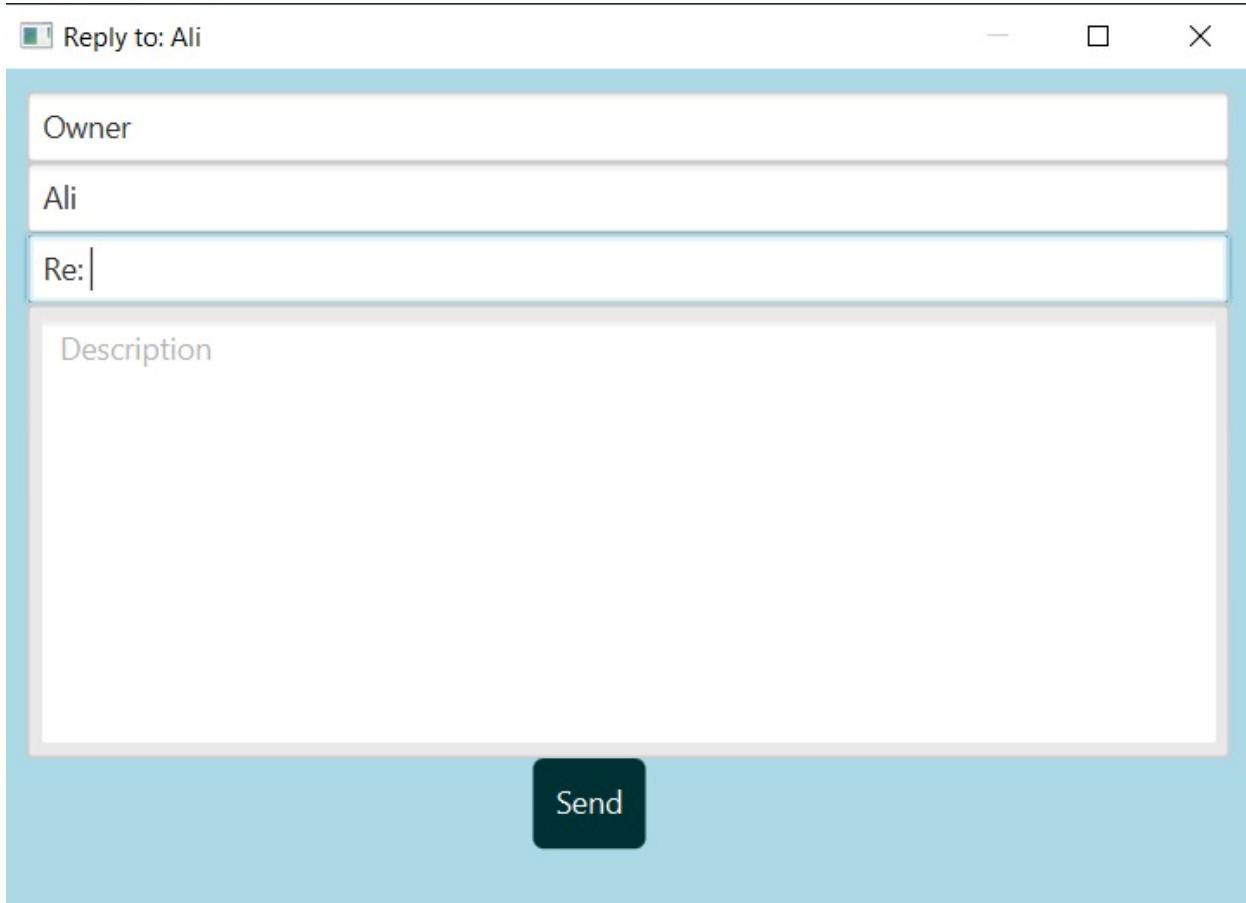
### **3.7.27. Room Allocation**

Rooms added to the records.



### ***3.7.27. Composing response to complaints***

Manager can respond to complaints from residents.



## 4. DOMAIN MODEL

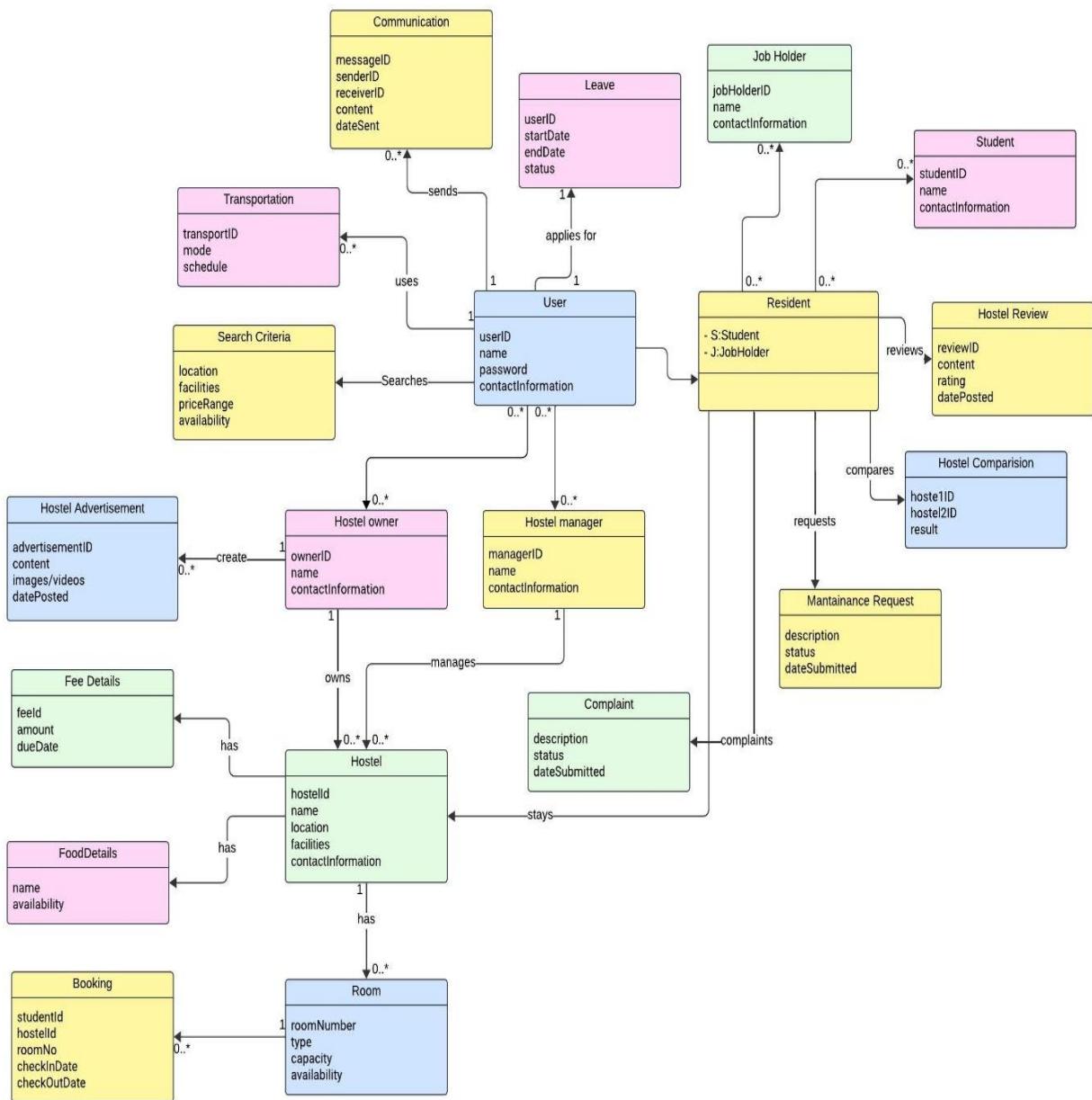


Figure 4 Domain Model

## 5. SYSTEM SEQUENCE DIAGRAM

### 5.1 BOOK HOSTEL

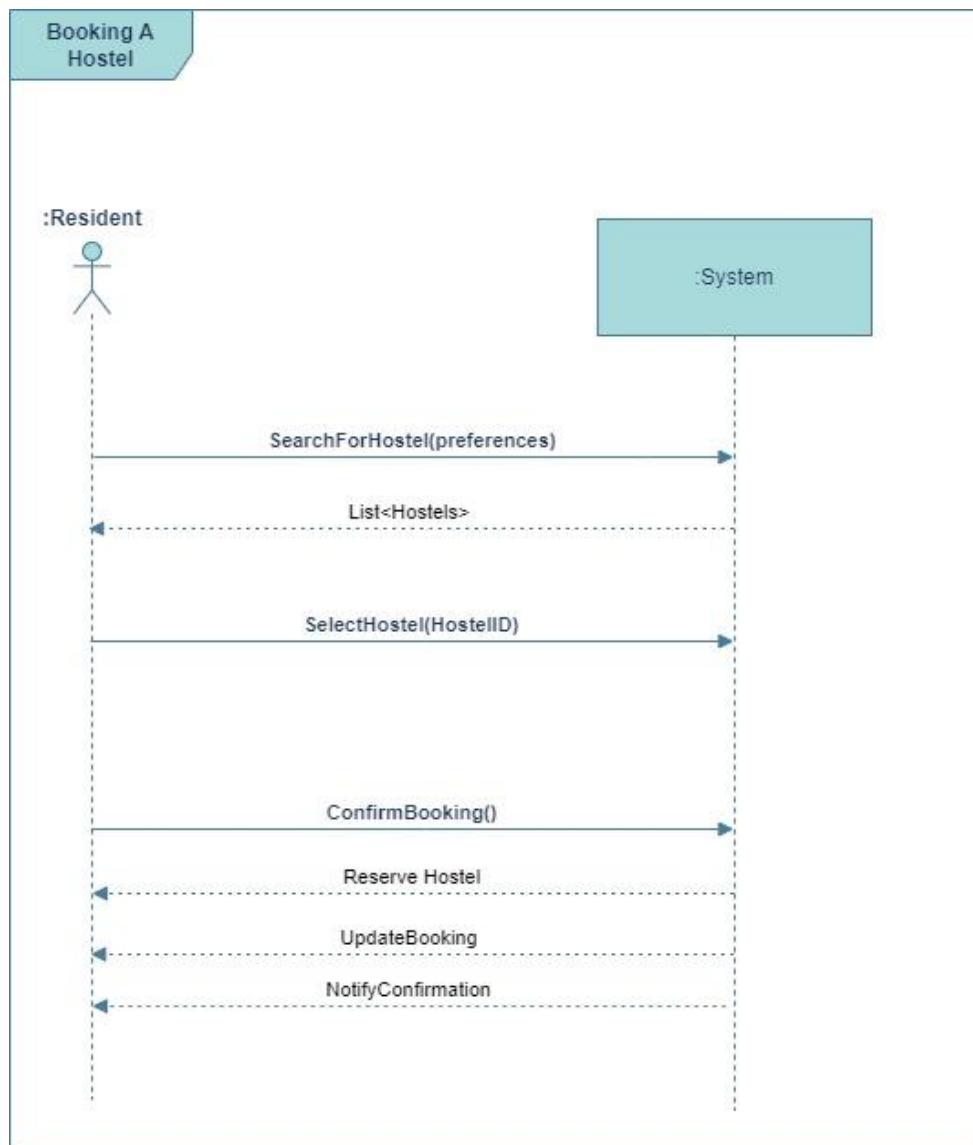


Figure 5 SSD- Book Hostel

## 5.2 MANAGE HOSTEL

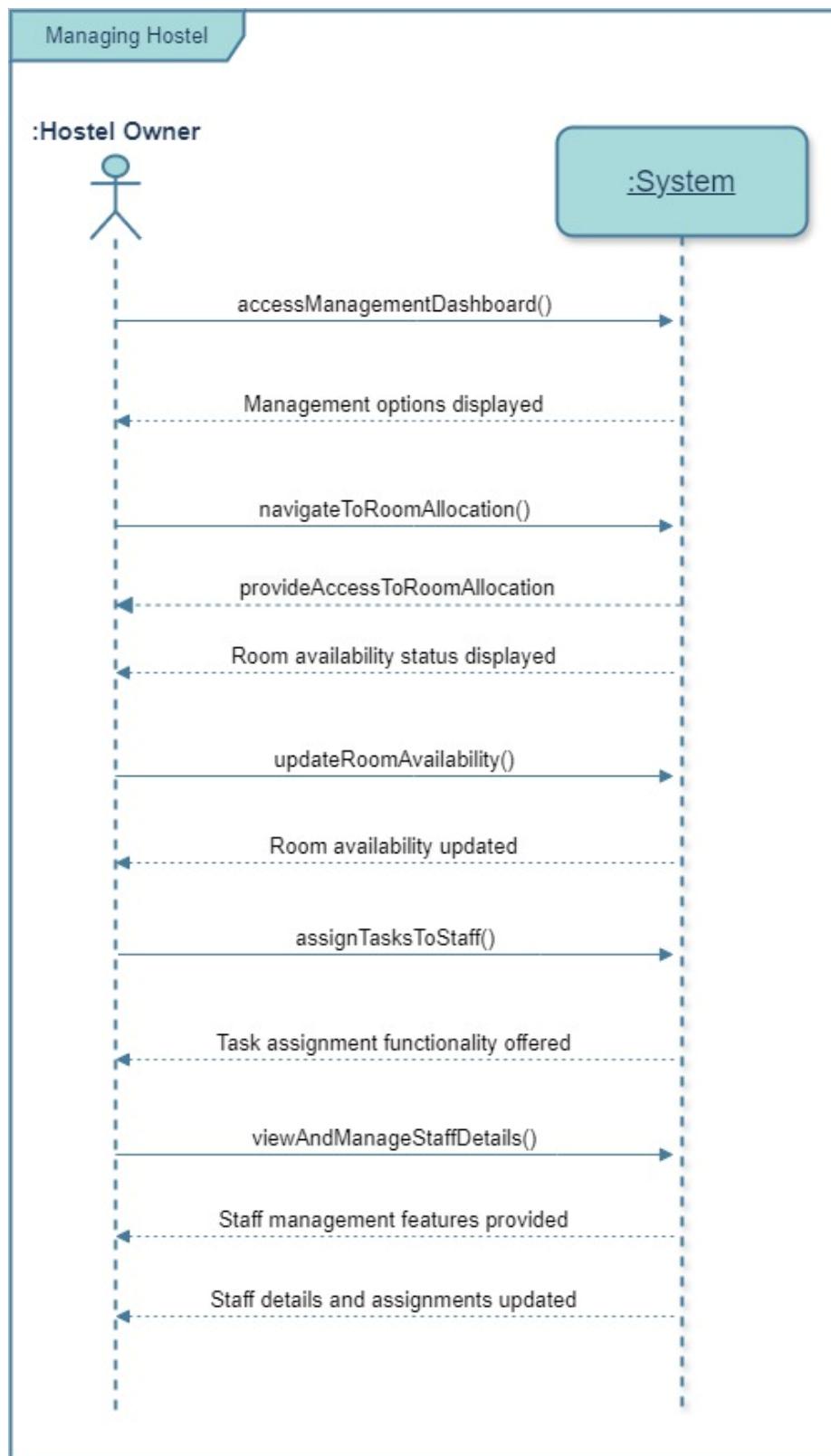


Figure 6 SSD- Manage Hostel

## 5.2 ALTERNATE SCENARIO

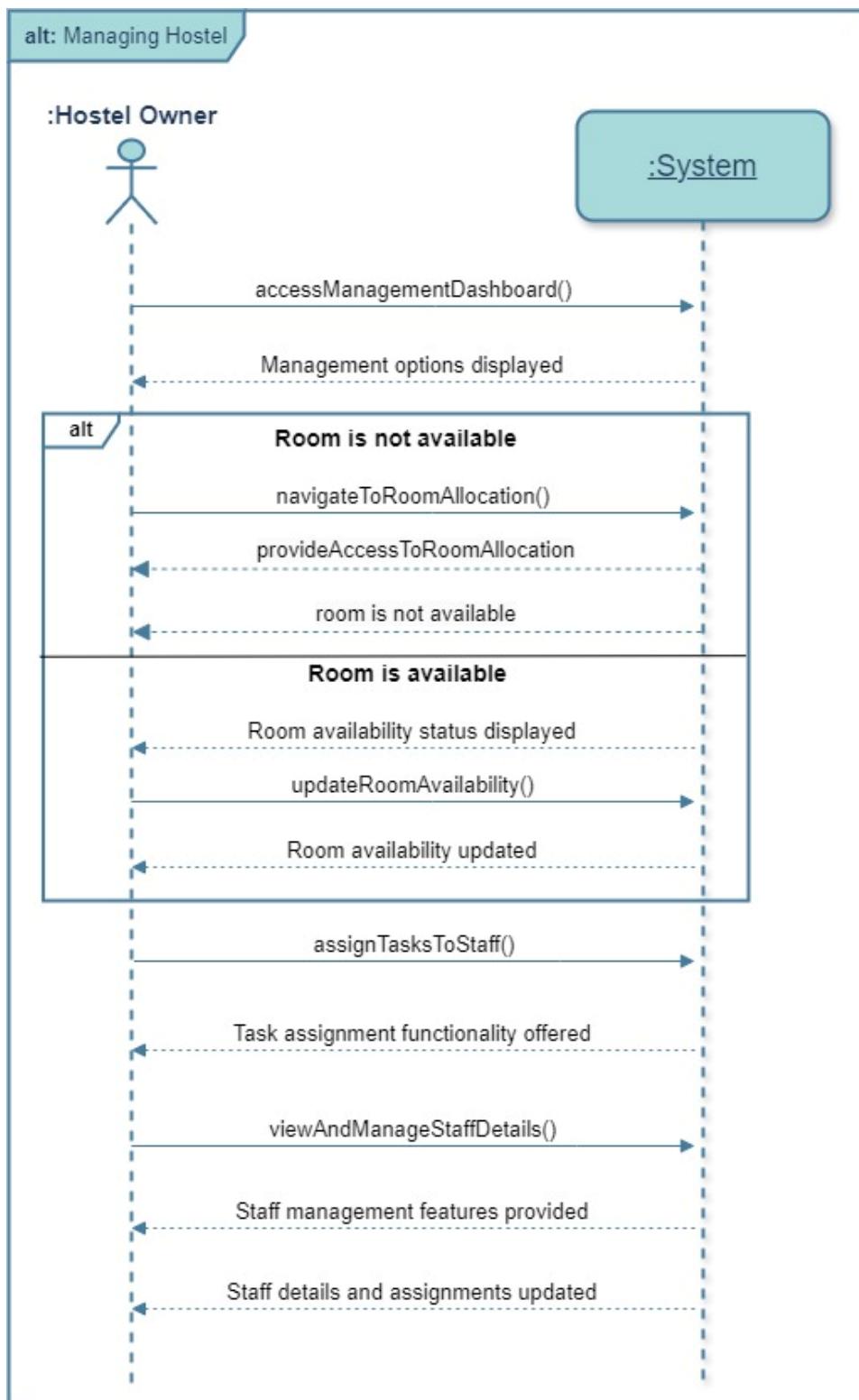


Figure 7 SSD-Manage Hostel Alt

## 5.3 MANAGE OWNER

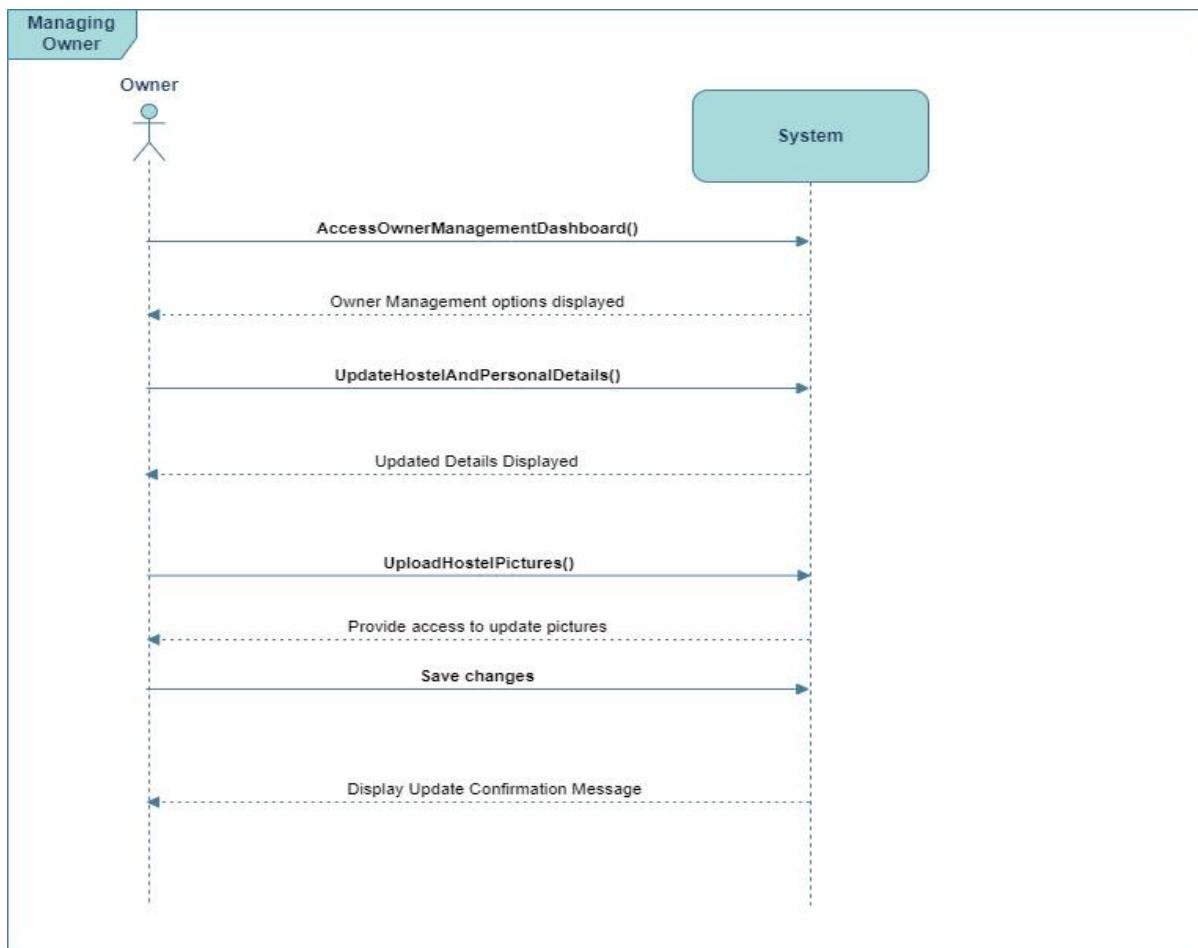


Figure 8 SSD-Manage Owner

## 5.4 SEARCH HOSTEL

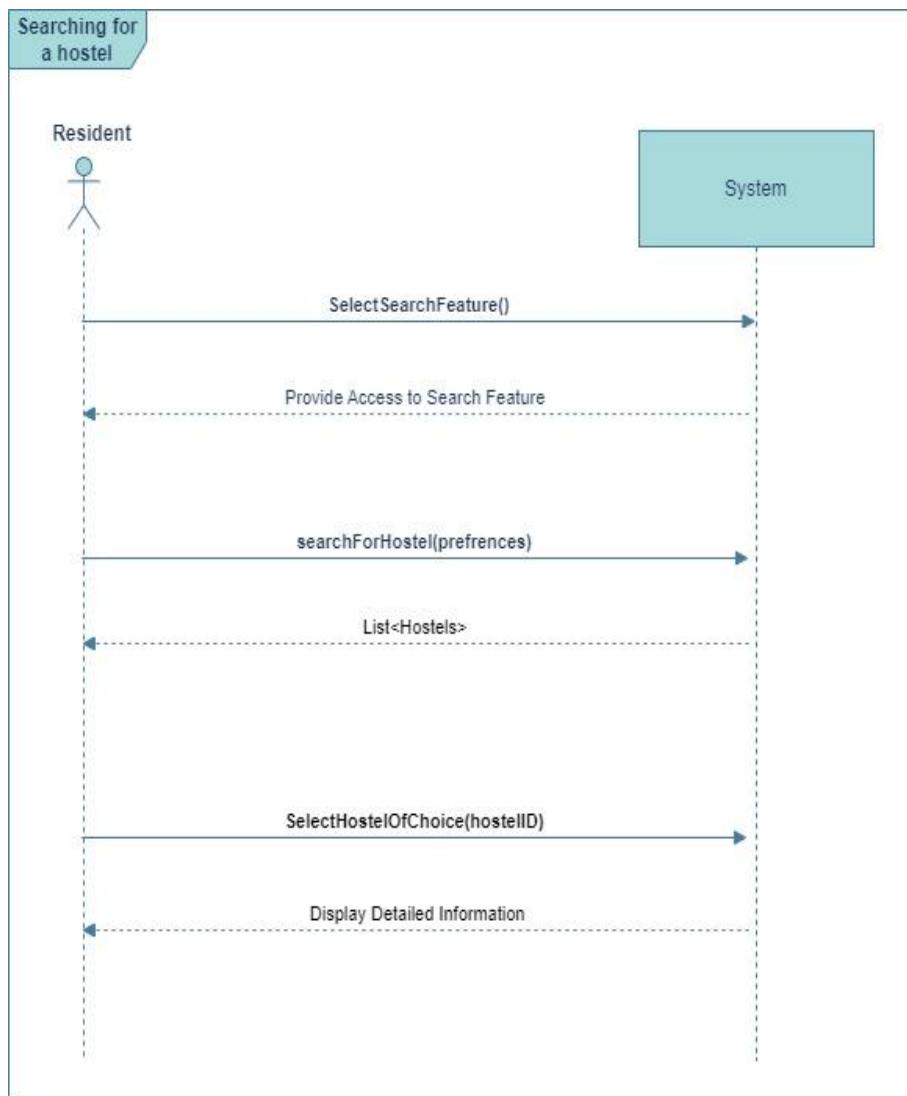


Figure 9 SSD- Search Hostel

## 5.5 MANAGE STUDENT INFORMATION

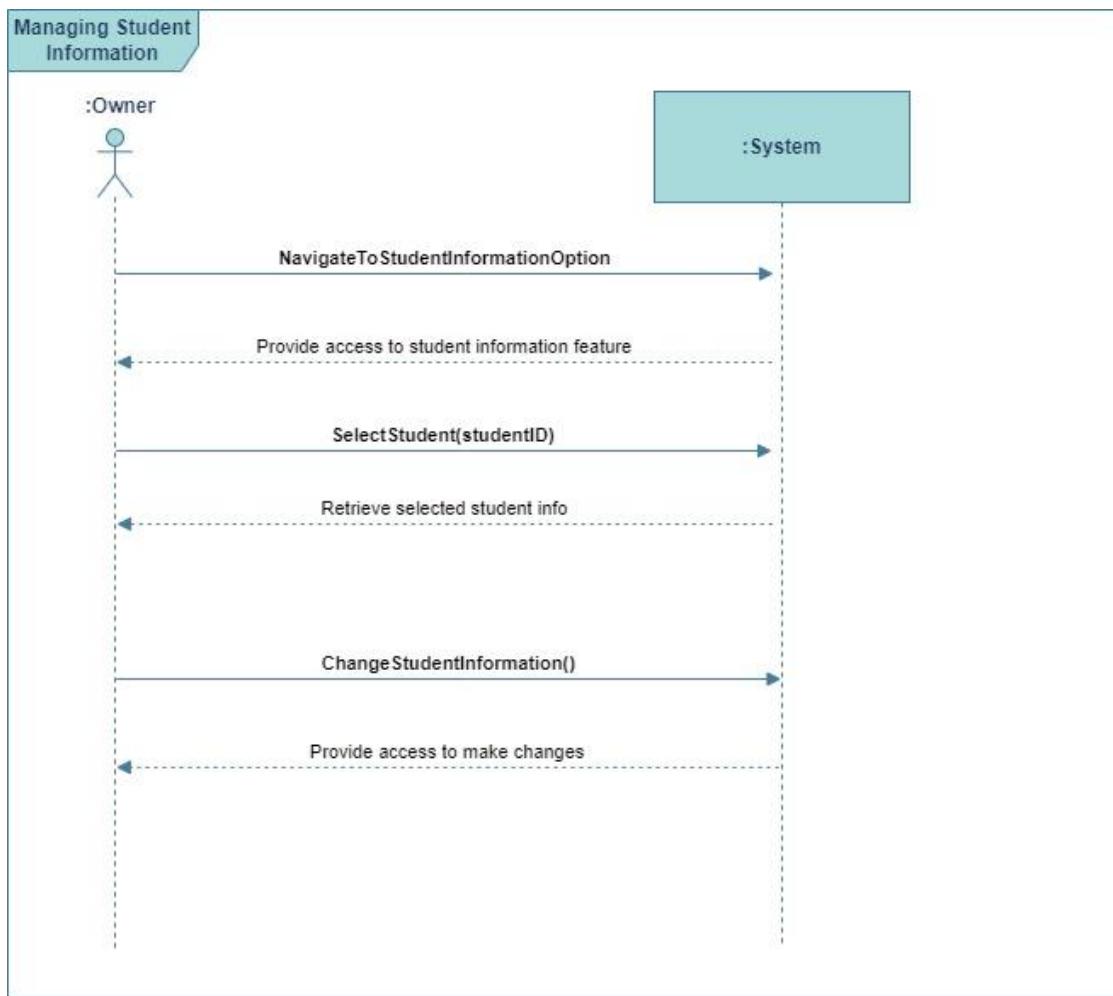


Figure 10 SSD- Manage student information

## 5.6 ADVERTISE HOSTEL

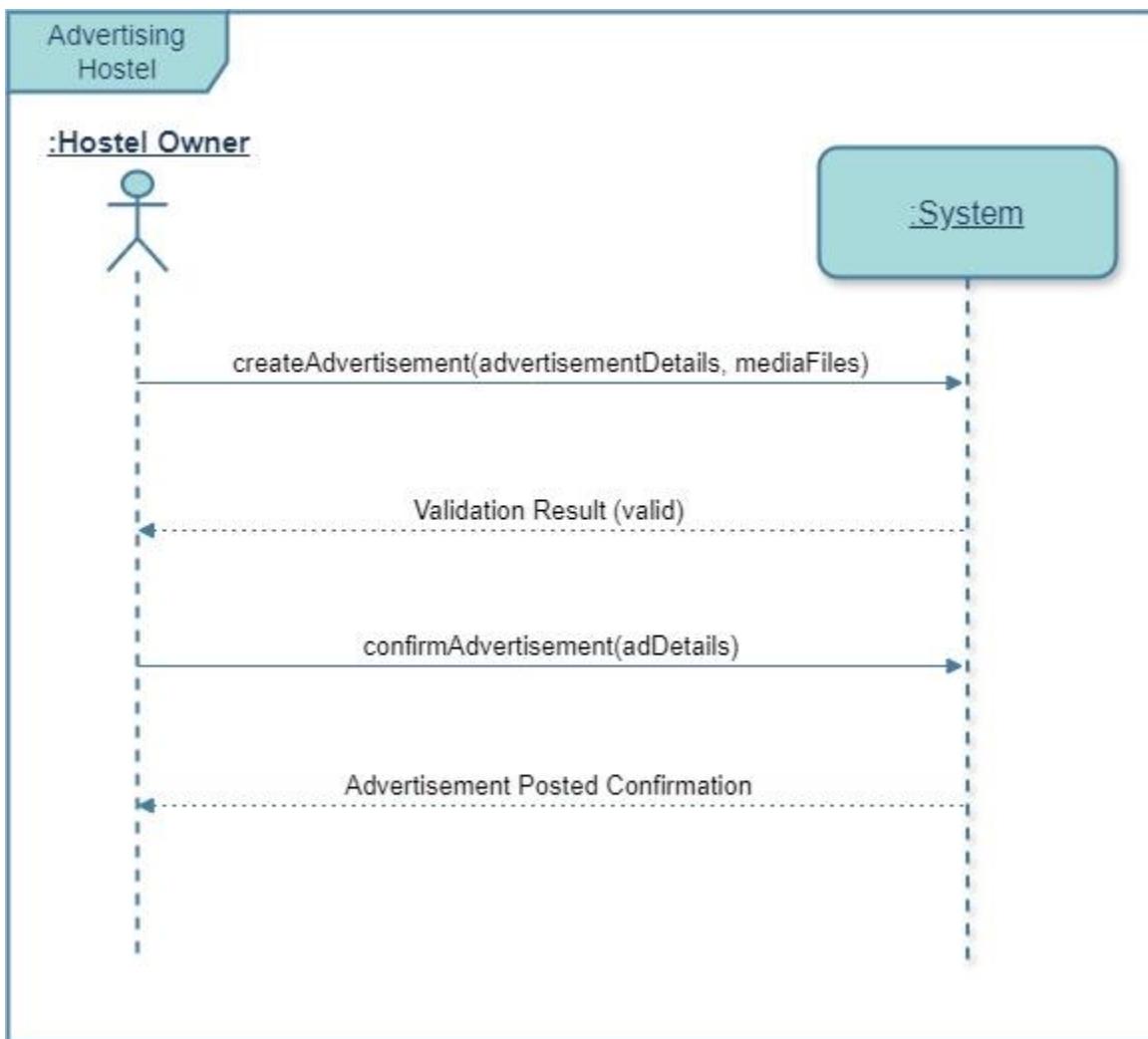


Figure 11 SSD- Advertise Hostel

## 5.7 COMPARE HOSTEL

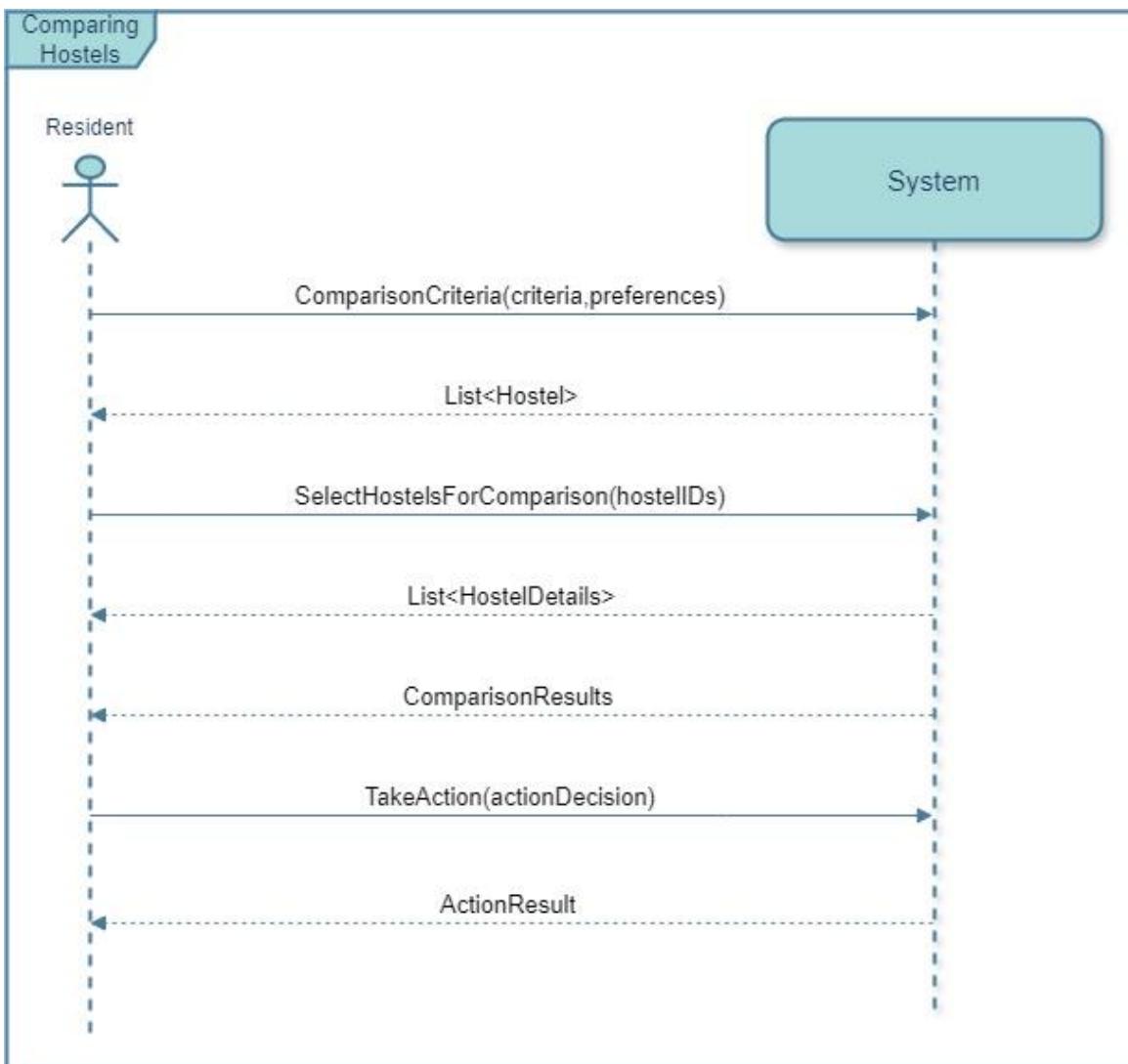


Figure 12 SSD- Compare Hostel

## 5.7 ALTERNATE SCENARIO

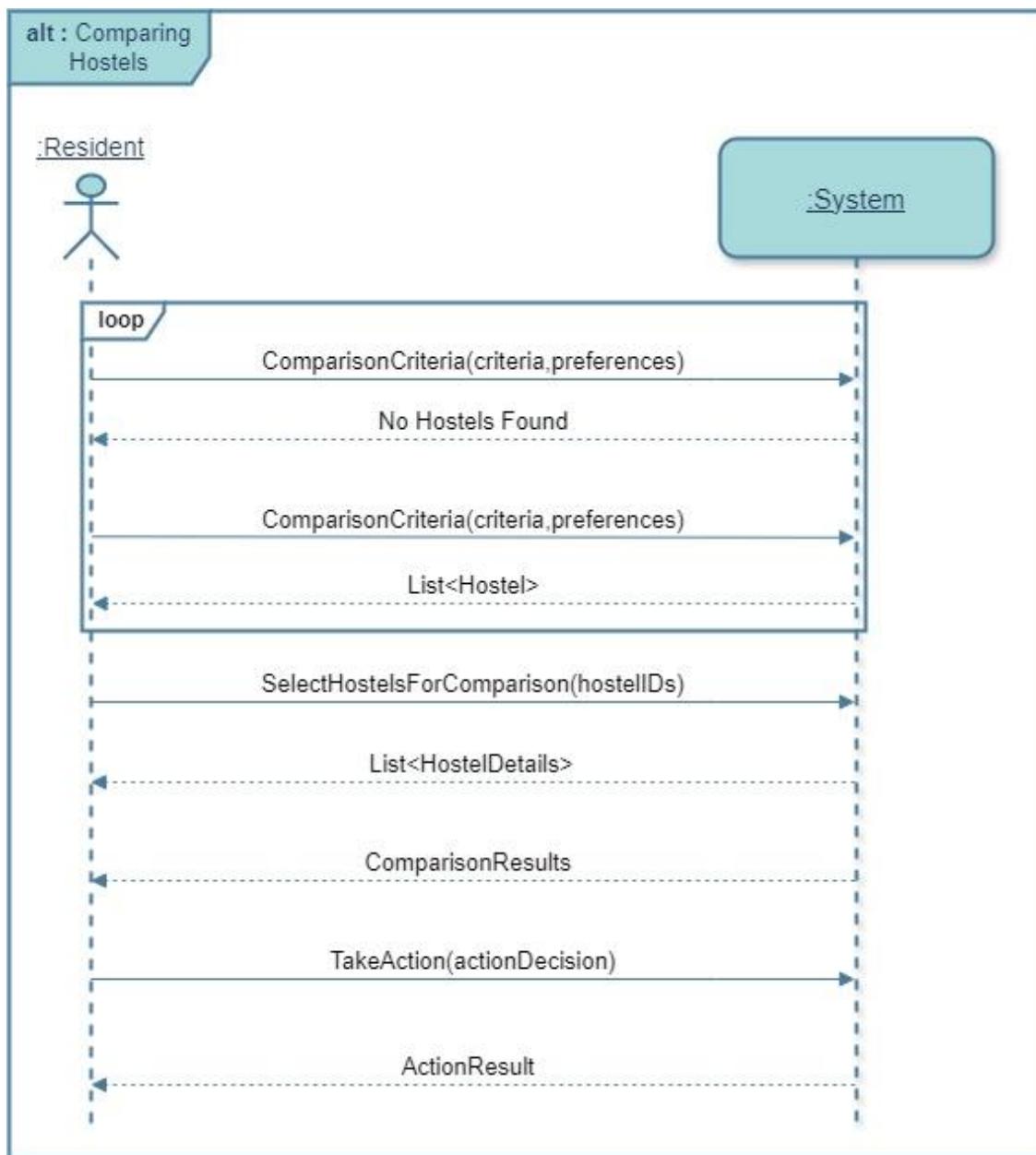


Figure 13 SSD- Compare Hostel Alt

## 5.8 MANAGE COMPLAINTS AND MAINTENANCE REQUESTS

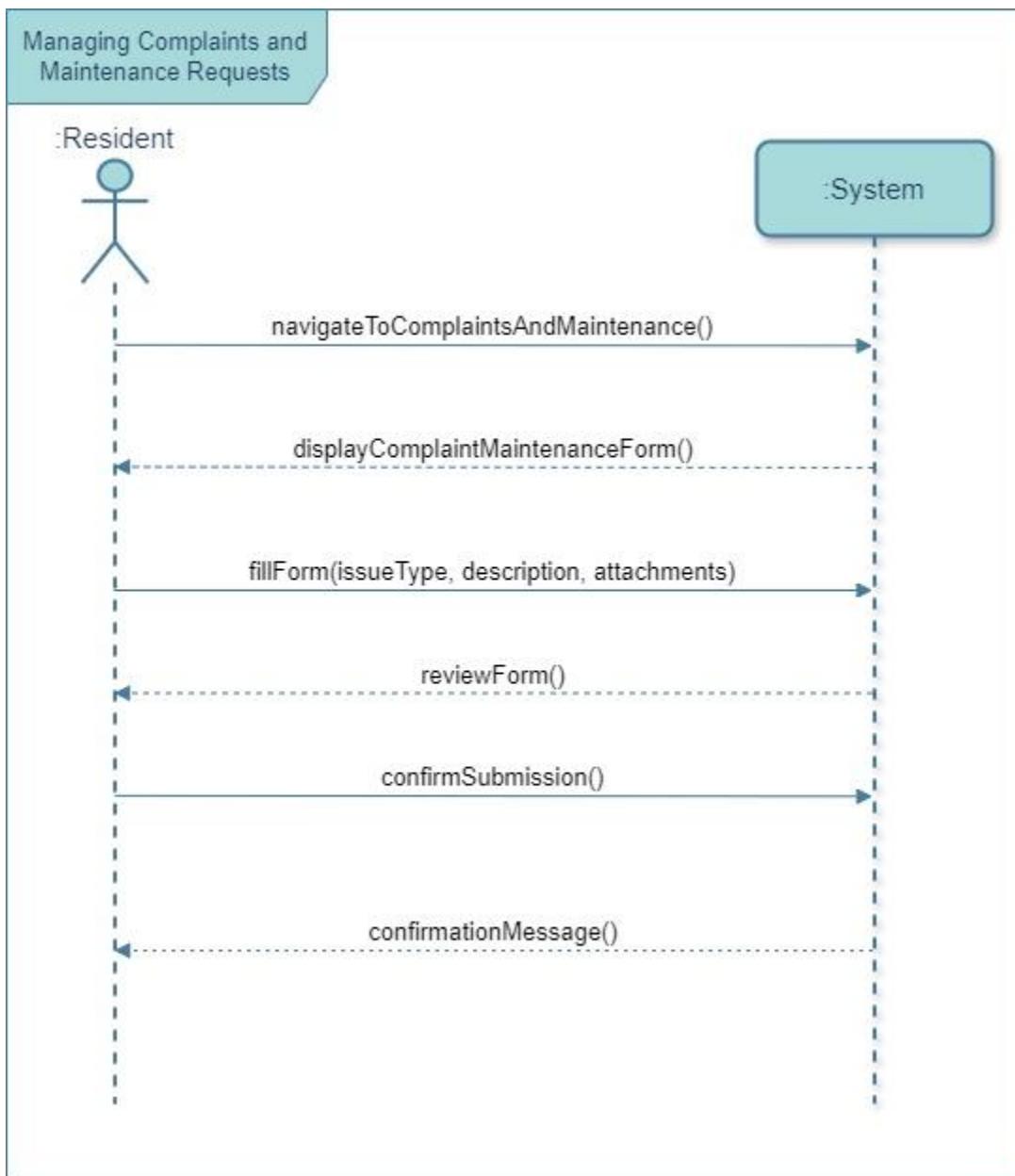


Figure 14 SSD- Manage Complaint and Maintenance Request

## 5.9 SUBMIT FEEDBACK AND REVIEWS

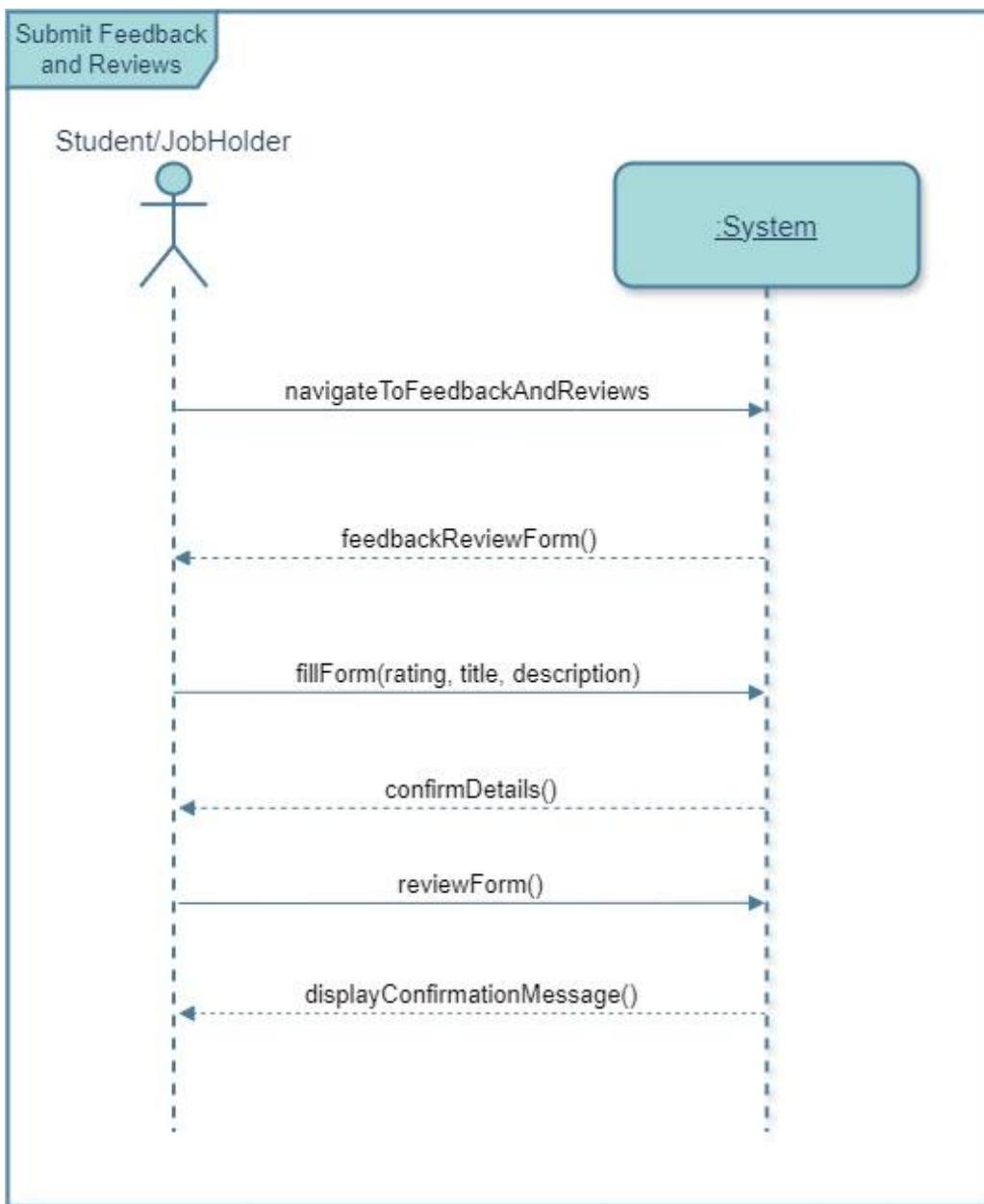


Figure 15 SSD- Submit Feedback and Review

## 5.9 ALTERNATE SCENARIO

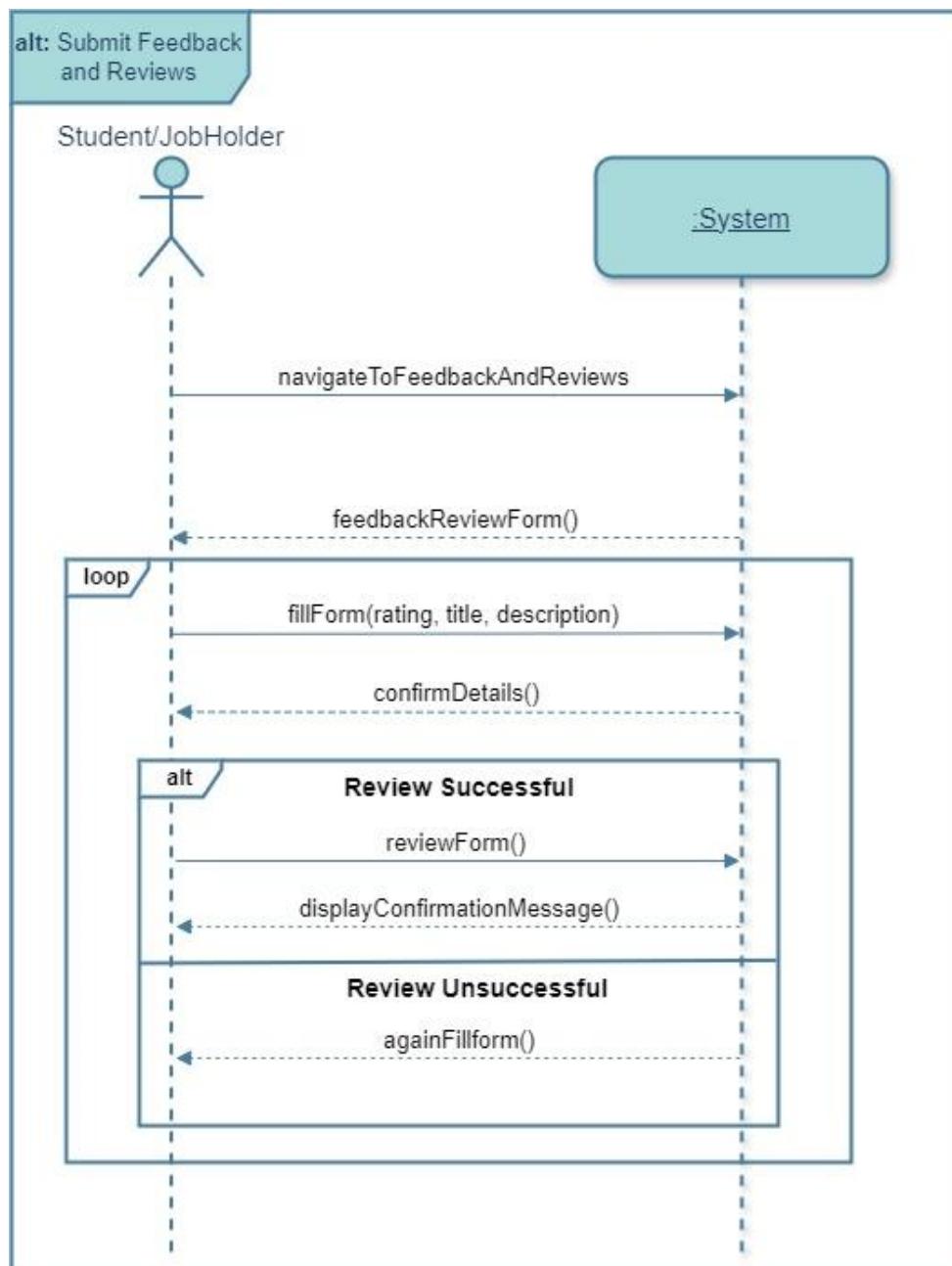


Figure 16 SSD- Submit Feedback and Review Alt

## 5.10 REGISTER STUDENT

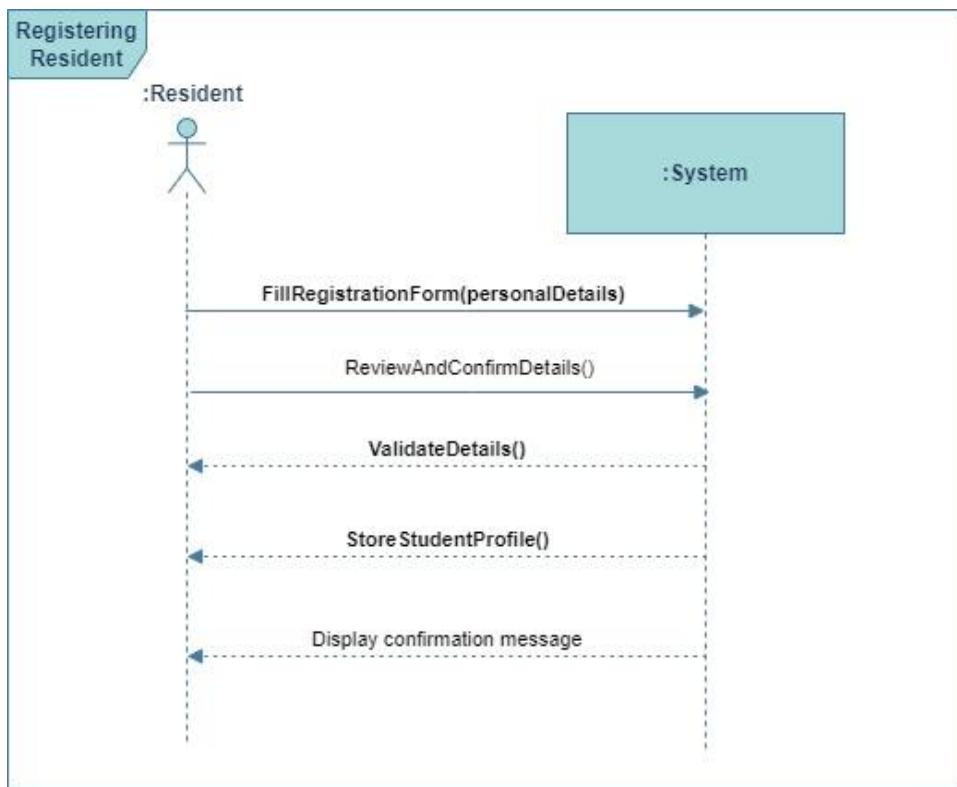


Figure 17 SSD- Register Student

## 5.10 ALTERNATE SCENARIO

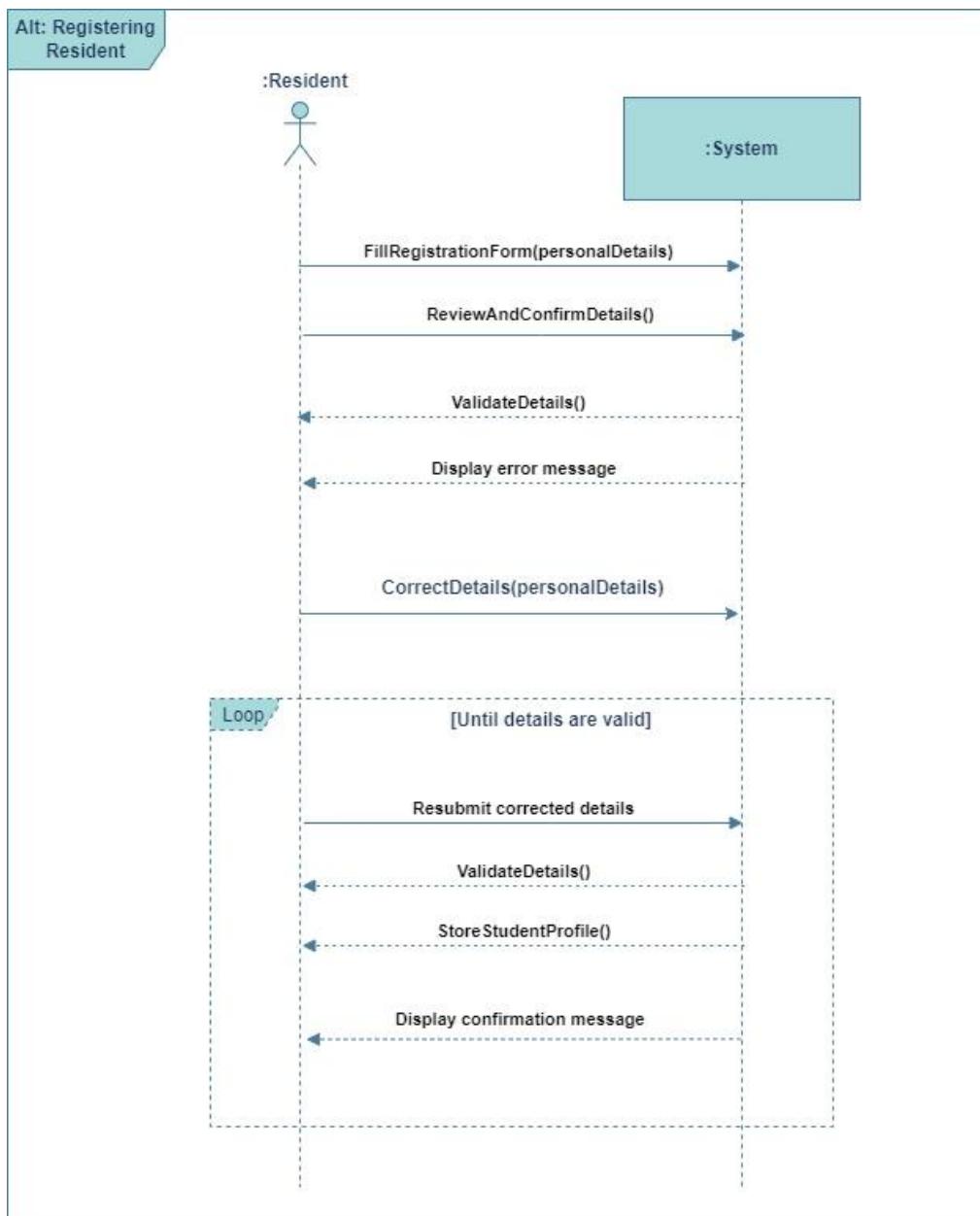


Figure 18 SSD- Register Student Alt

## 5.11 MANAGE FEE DETAILS

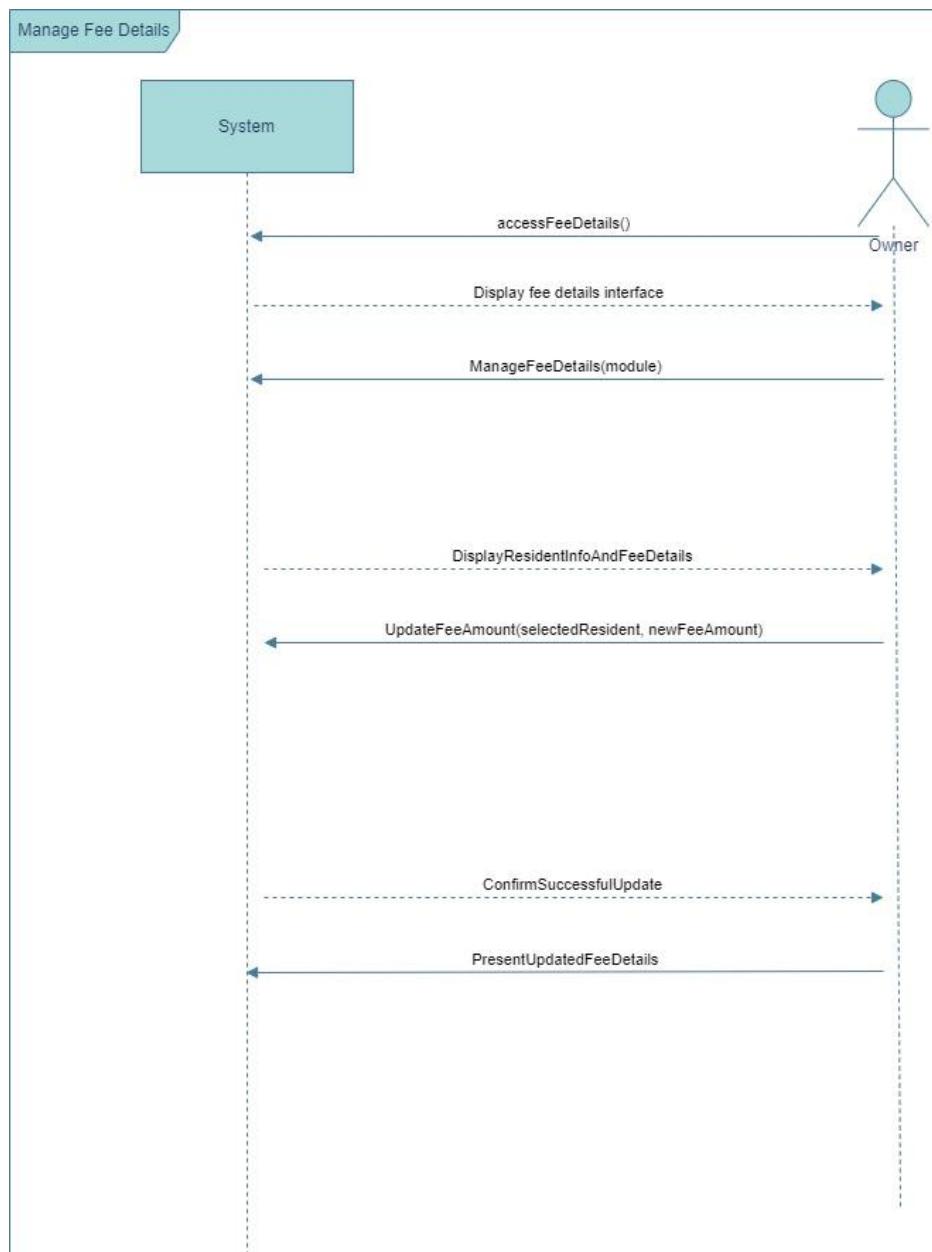


Figure 19 SSD- Manage Fee Details

## 5.11 ALTERNATE SCENARIO

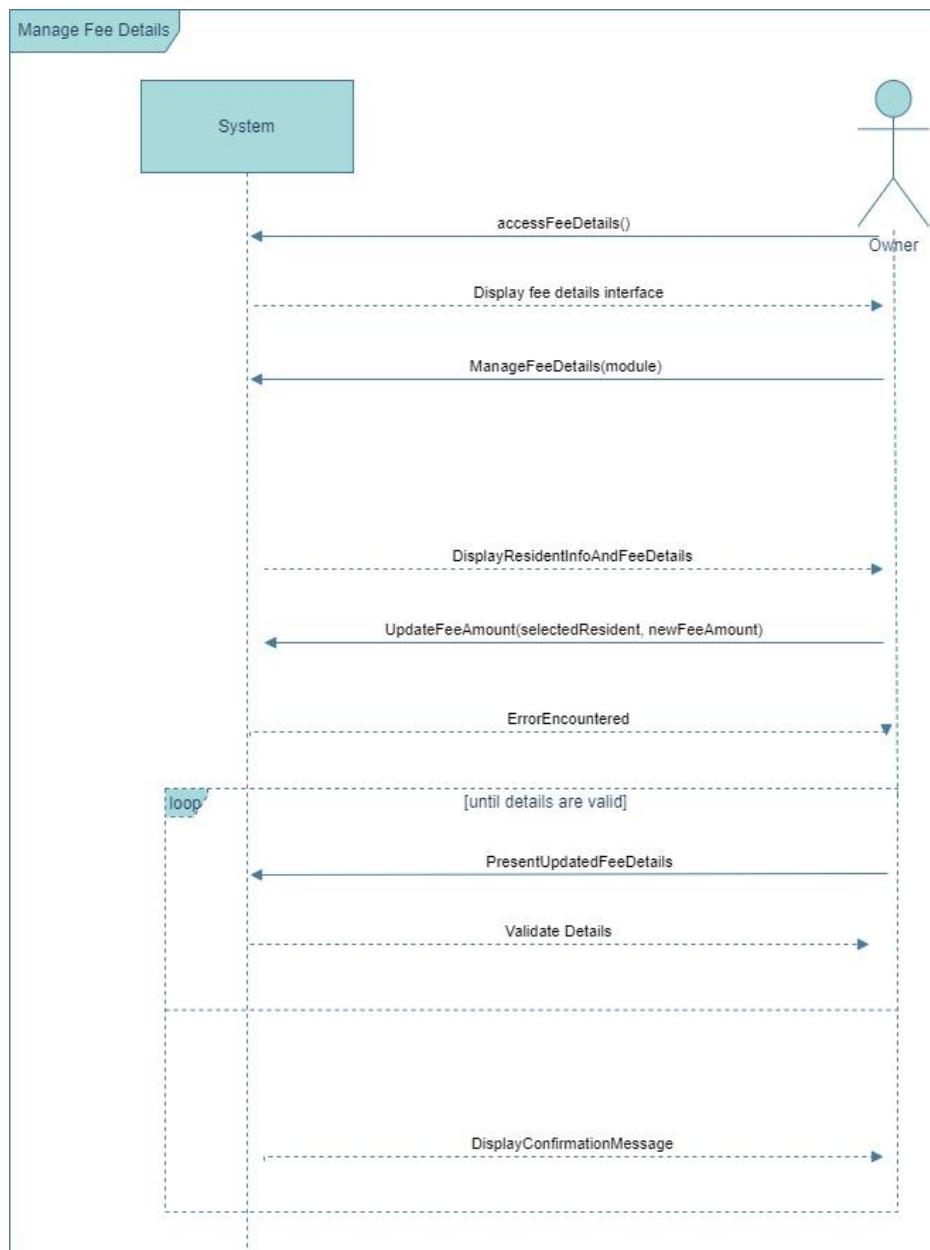


Figure 20 SSD- Manage Fee Details Alt

## 5.12 AVAIL TRANSPORTATION SERVICES

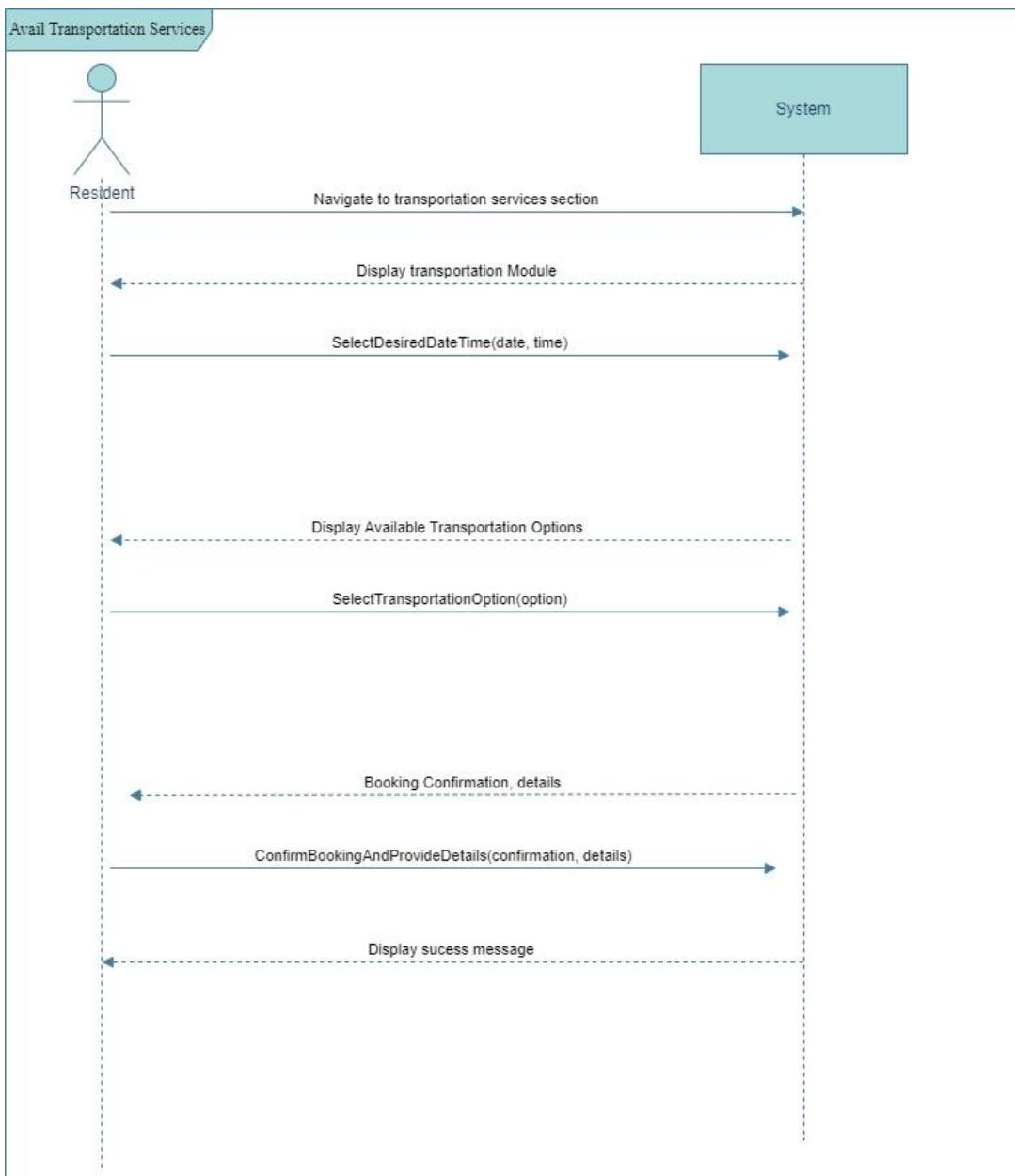


Figure 21 SSD- Avail Transportation Services

## 5.13 ALLOCATE ROOMS

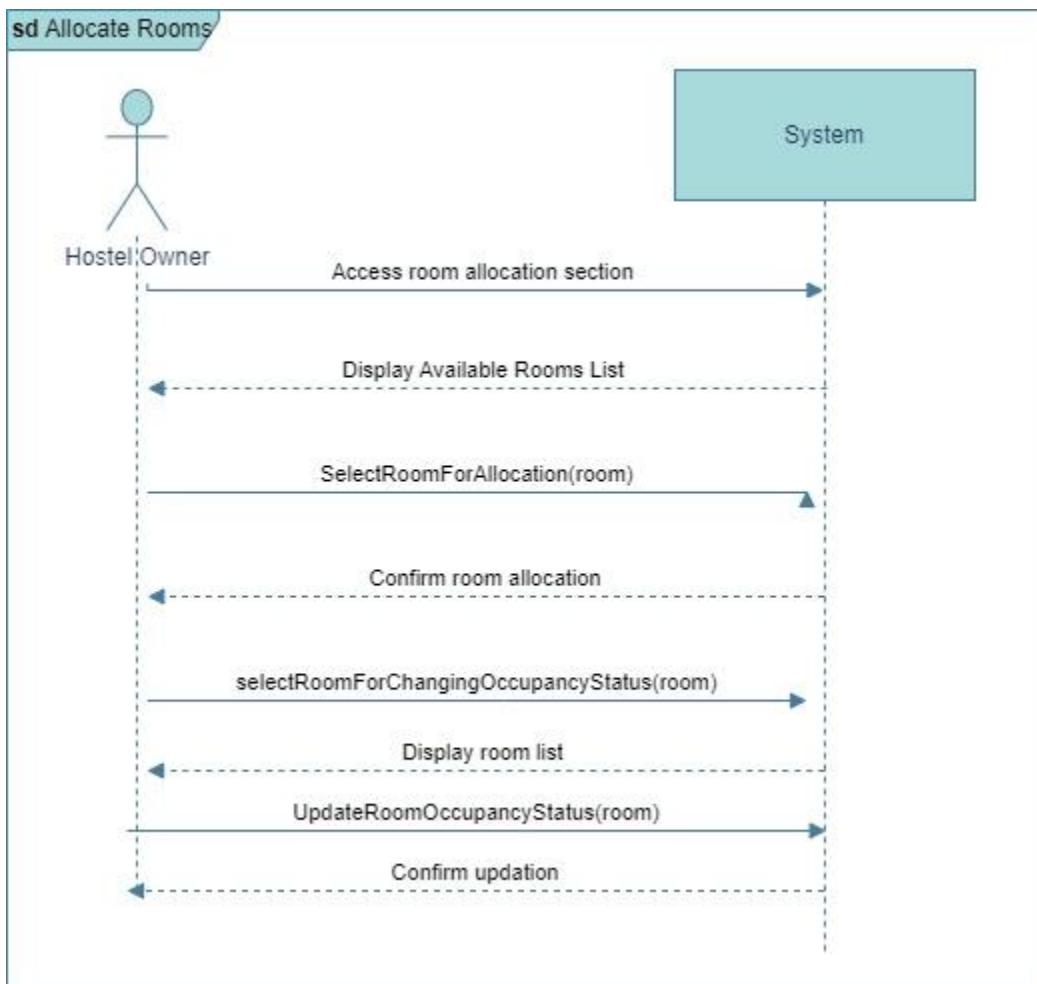


Figure 22 SSD- Allocate Rooms

## 5.14 MESSAGE AND COMMUNICATION

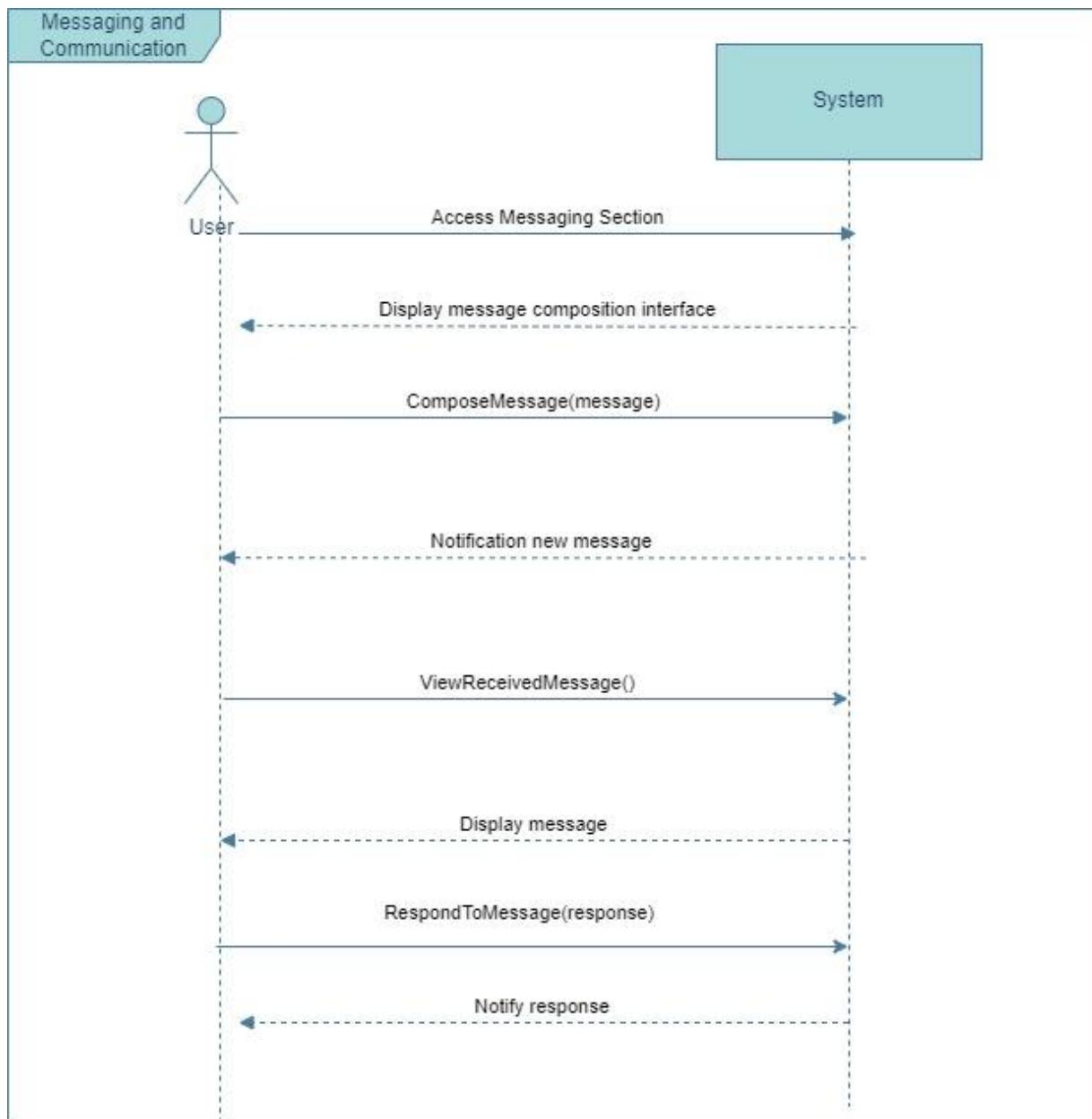


Figure 23 SSD- Message and Communication

## 5.15 MANAGE LEAVE REQUEST

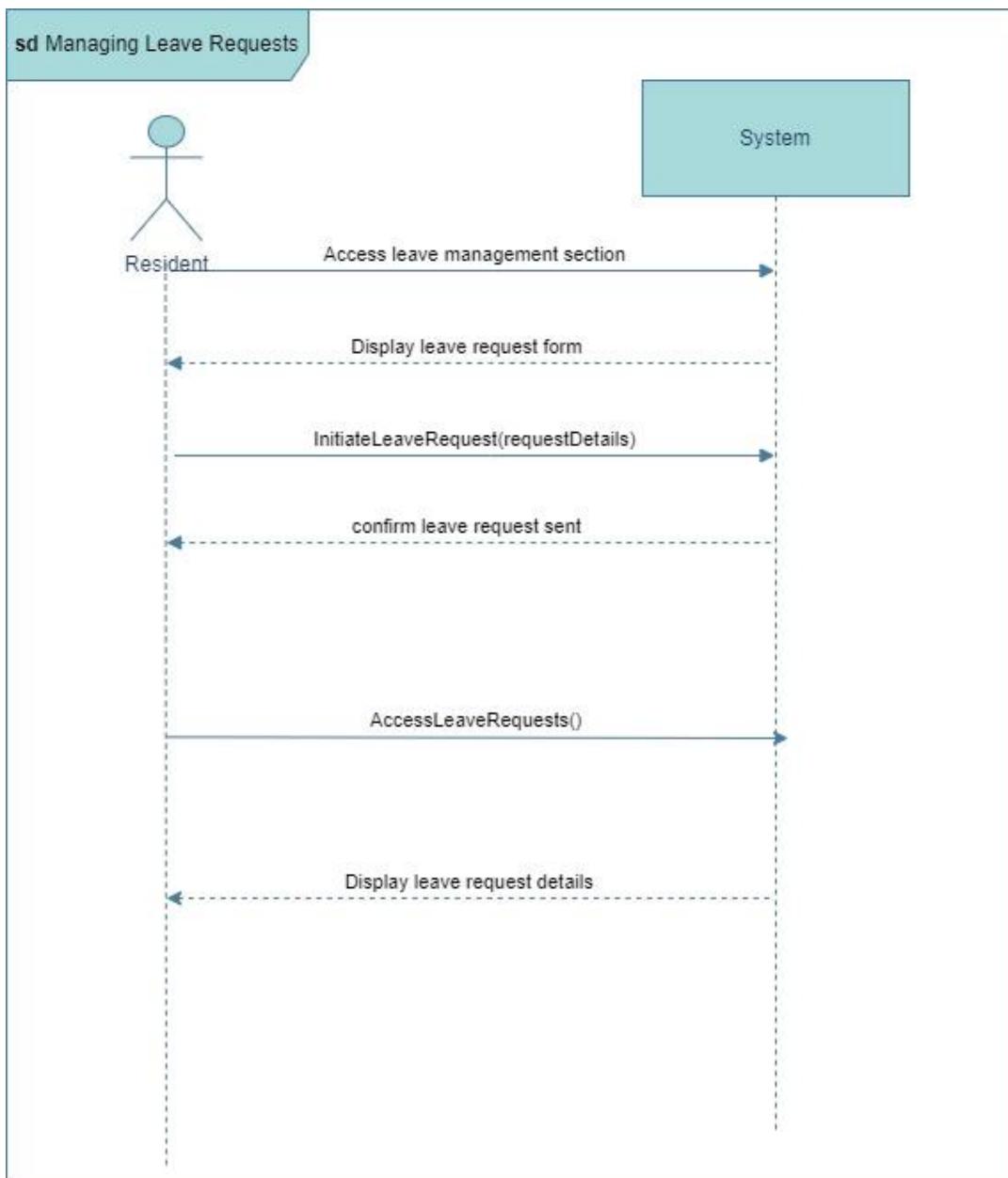


Figure 24 SSD- Manage Leave Request

## 5.16 MANAGE FOOD DETAILS

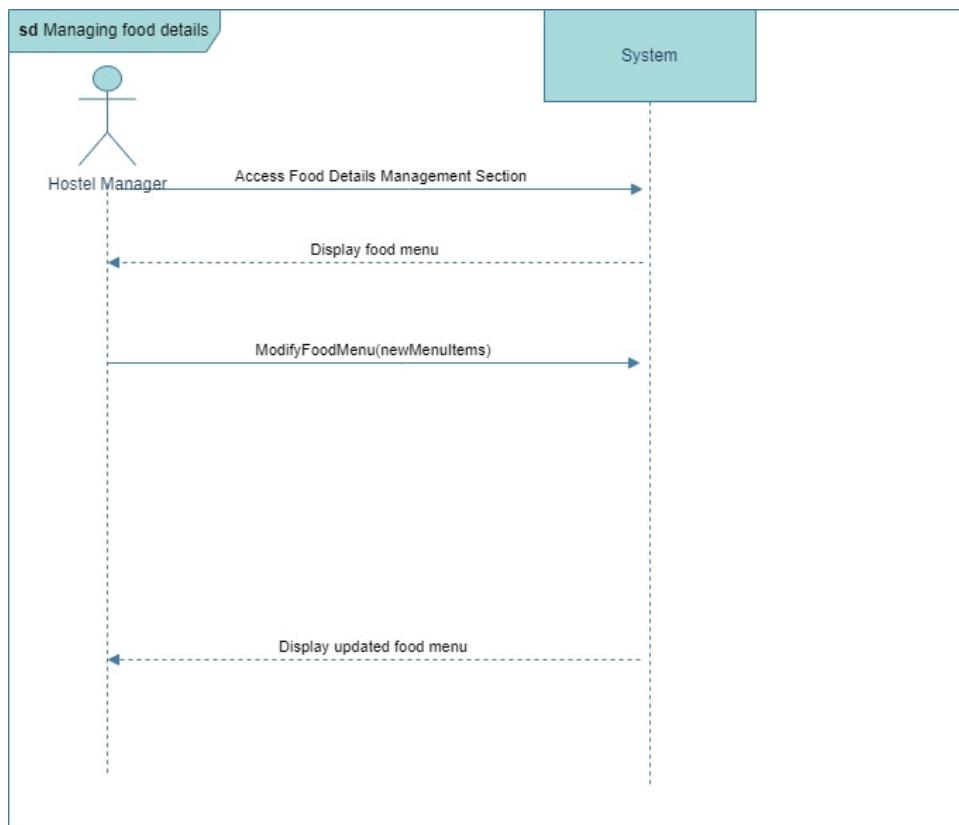


Figure 25 SSD- Manage Food Details

## 6. SEQUENCE DIAGRAM

### 6.1 Book a Hostel

#### 6.1.1 Search Hostel

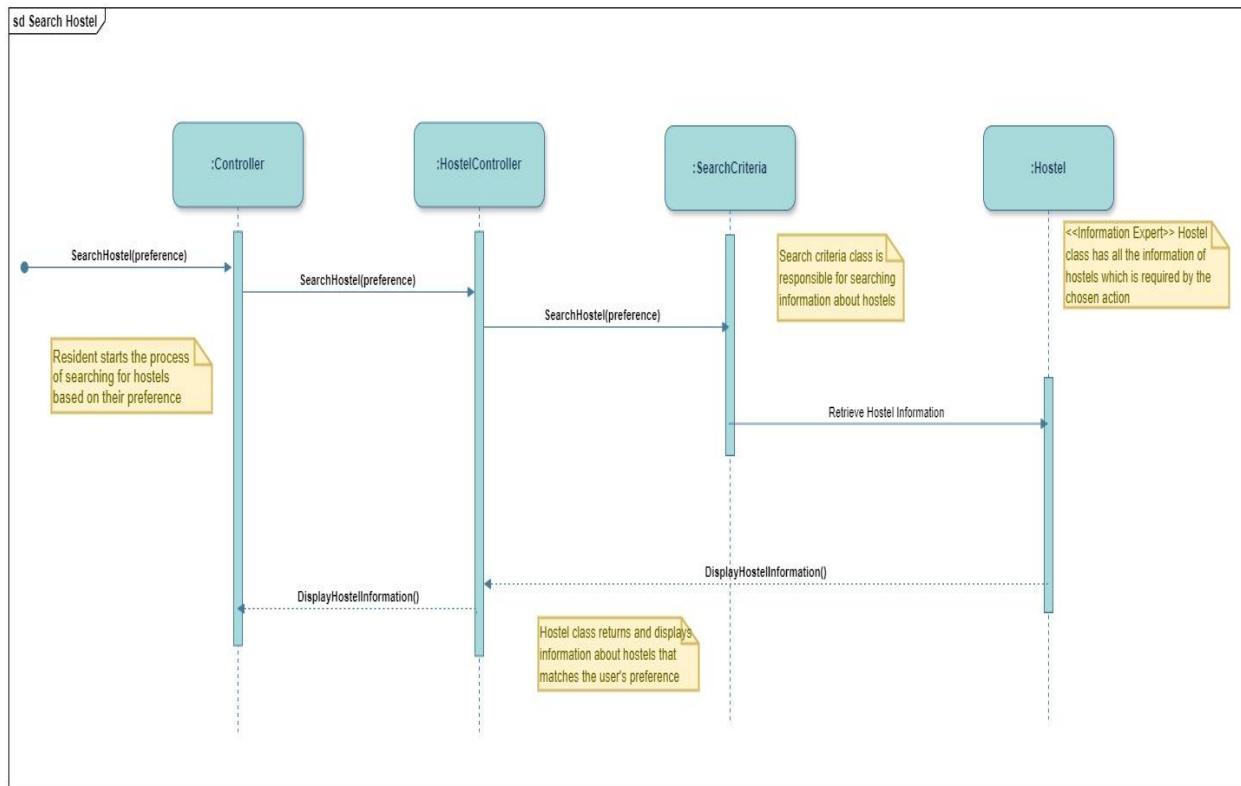


Figure 26 SD- Search Hostel

### 6.1.2 Select Hostel

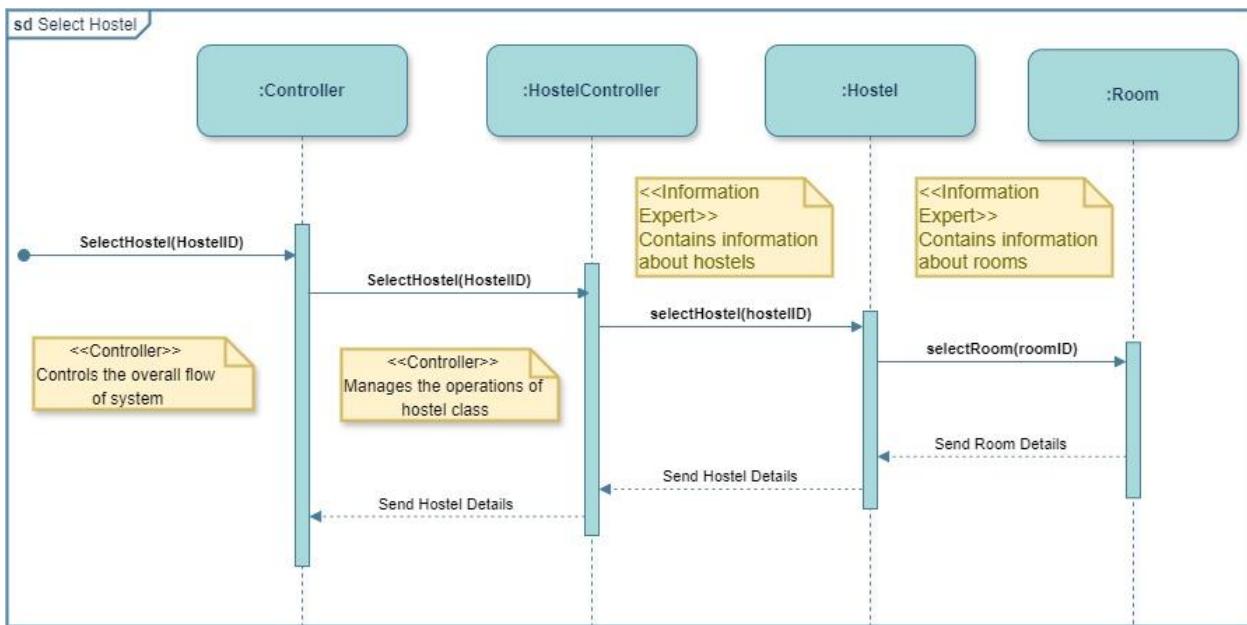


Figure 27 SD- Select Hostel

### 6.1.3 Confirm Booking

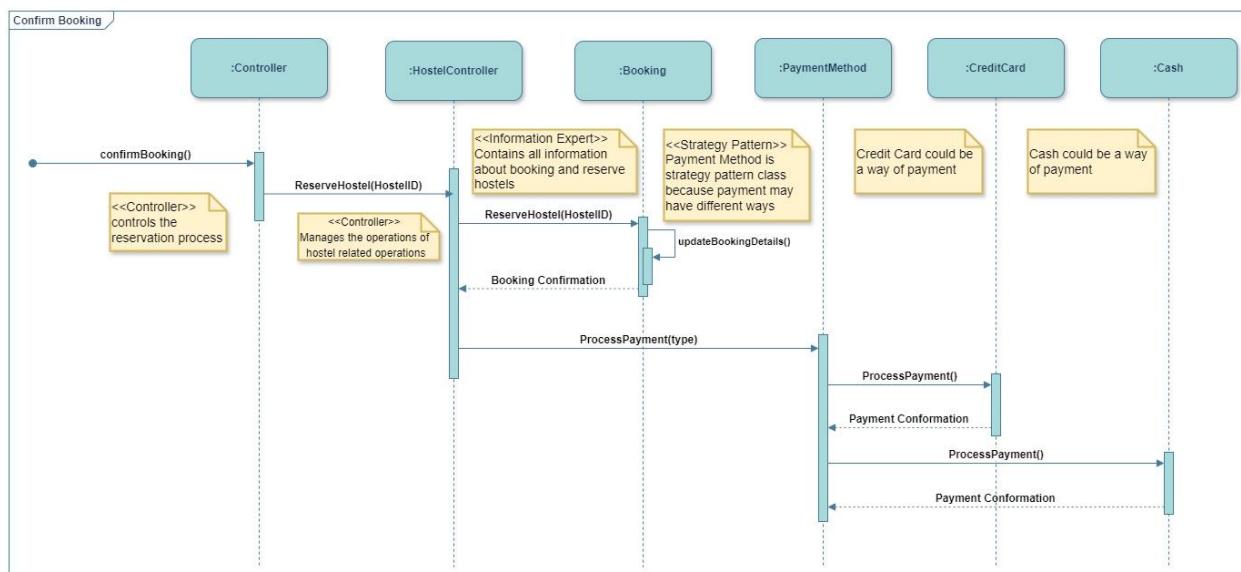


Figure 28 SD-Confirm Booking

## 6.2 Manage Hostel

### 6.2.1 View and Manage Staff Details

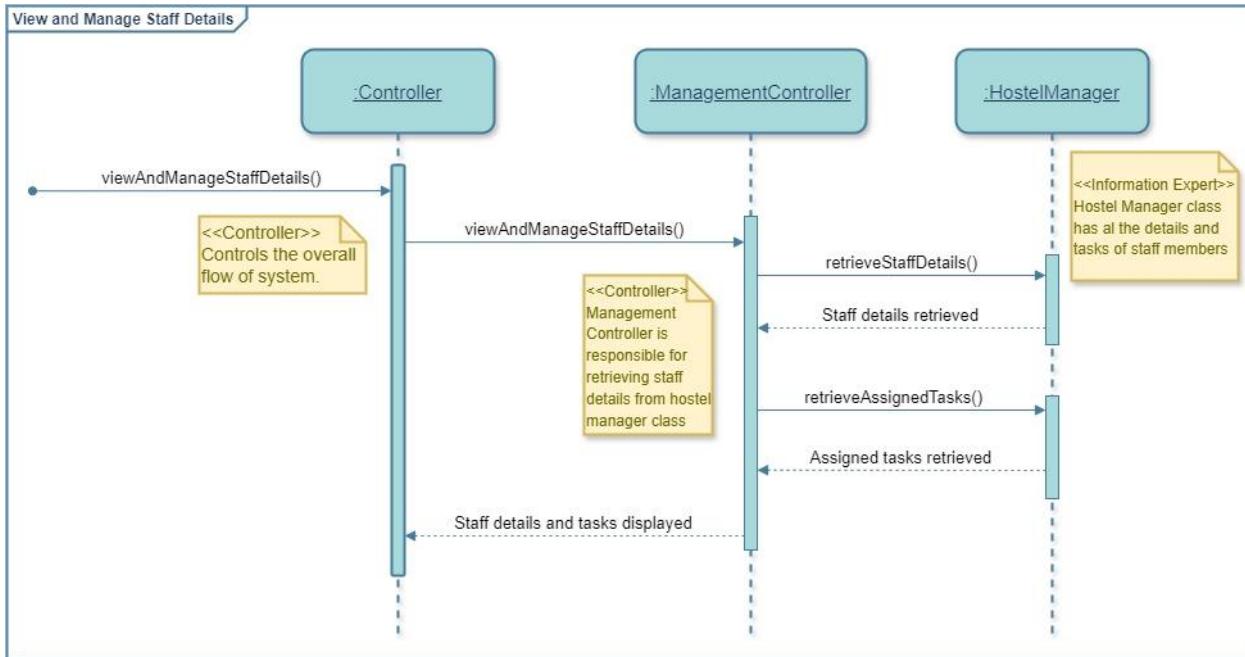


Figure 29 SD- View and Manage Staff Details

### 6.2.2 Task Assignment

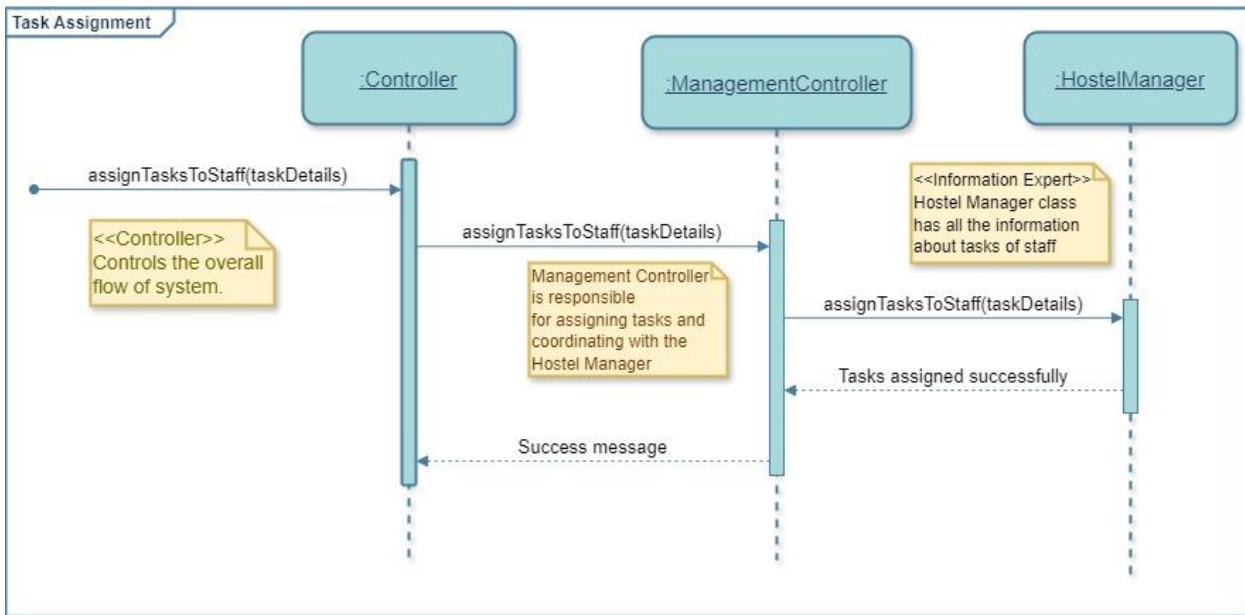


Figure 30 SD- Task Assignment

### 6.2.3 Update Room Availability

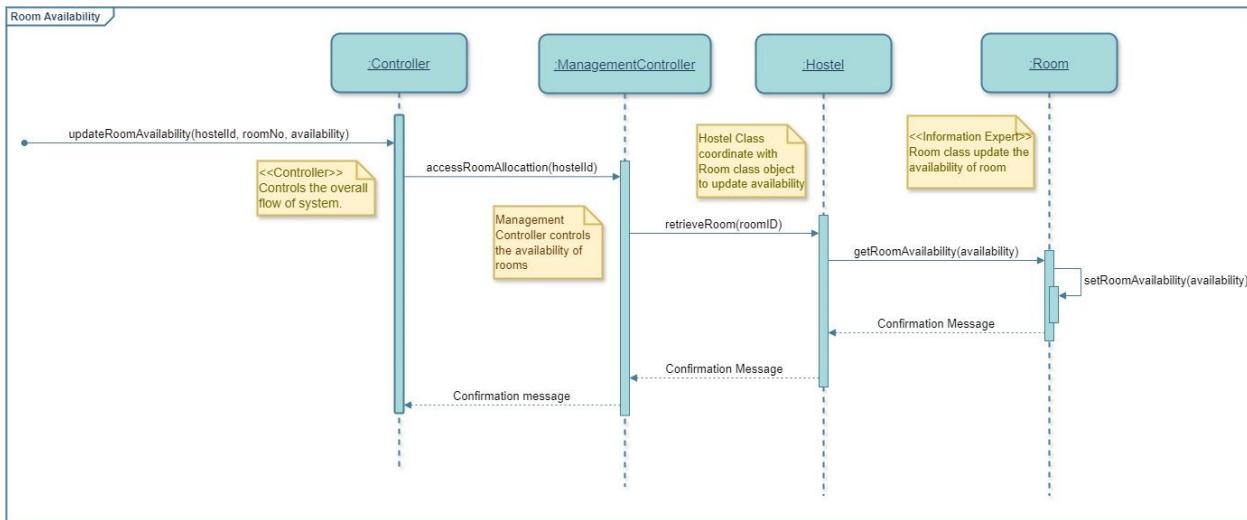


Figure 31 SD- Update Room Availability

### 6.2.4 Update Database

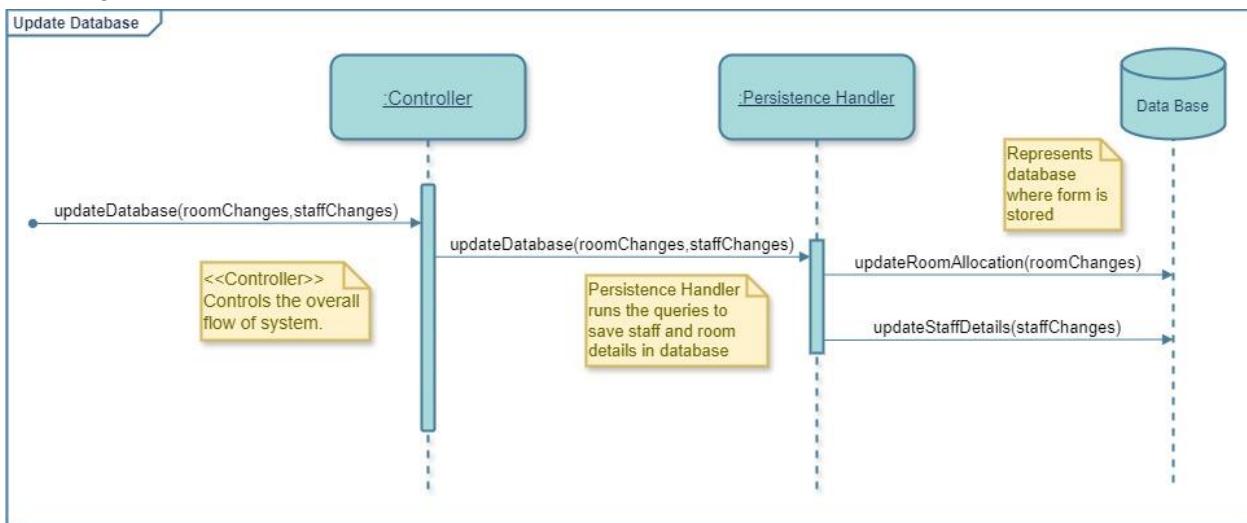


Figure 32 SD- Update Database

## 6.3 Manage Owner

### 6.3.1 Update Hostel Details

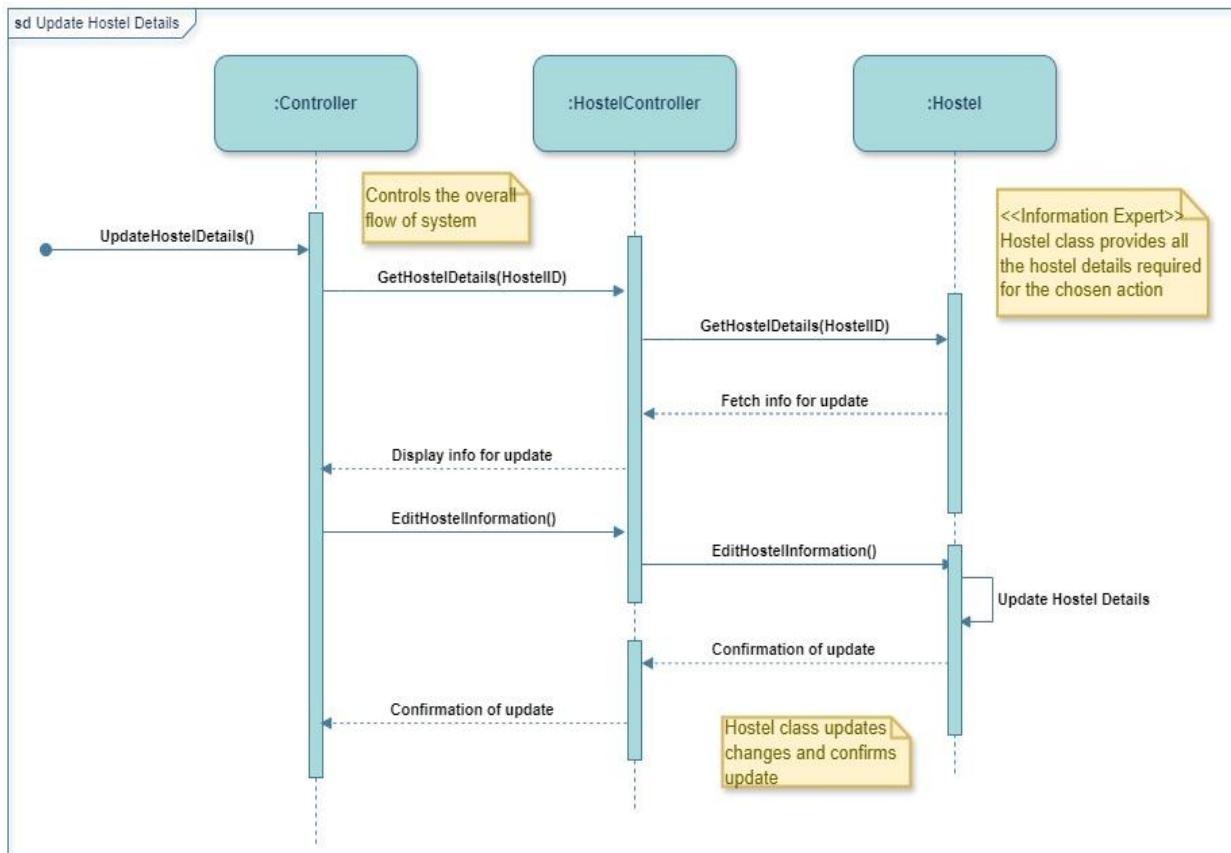


Figure 33 SD- Update Hostel Details

### 6.3.2 Upload Hostel Pictures

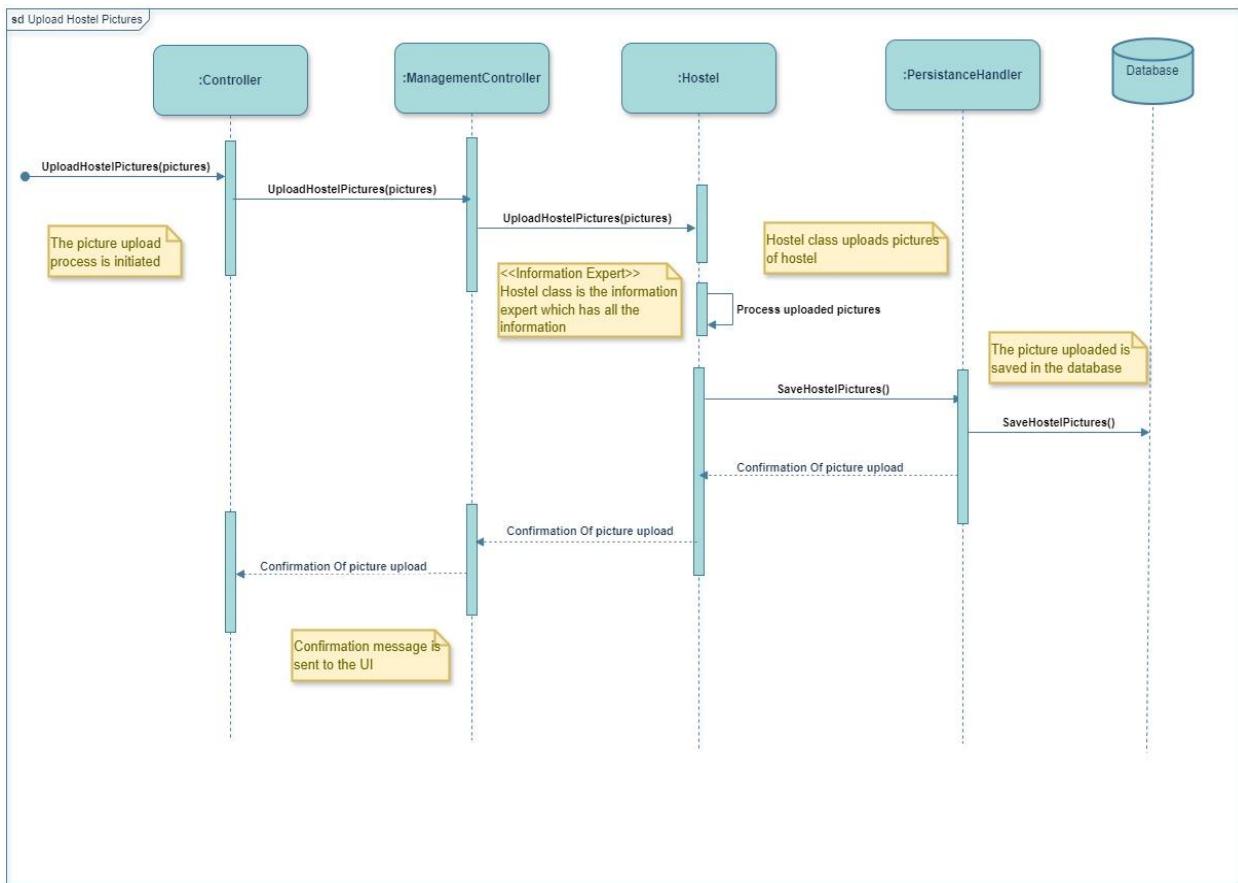


Figure 34 SD- Upload Hostel Pictures

## 6.4. Searching Hostel

### 6.4.1 Search Hostel

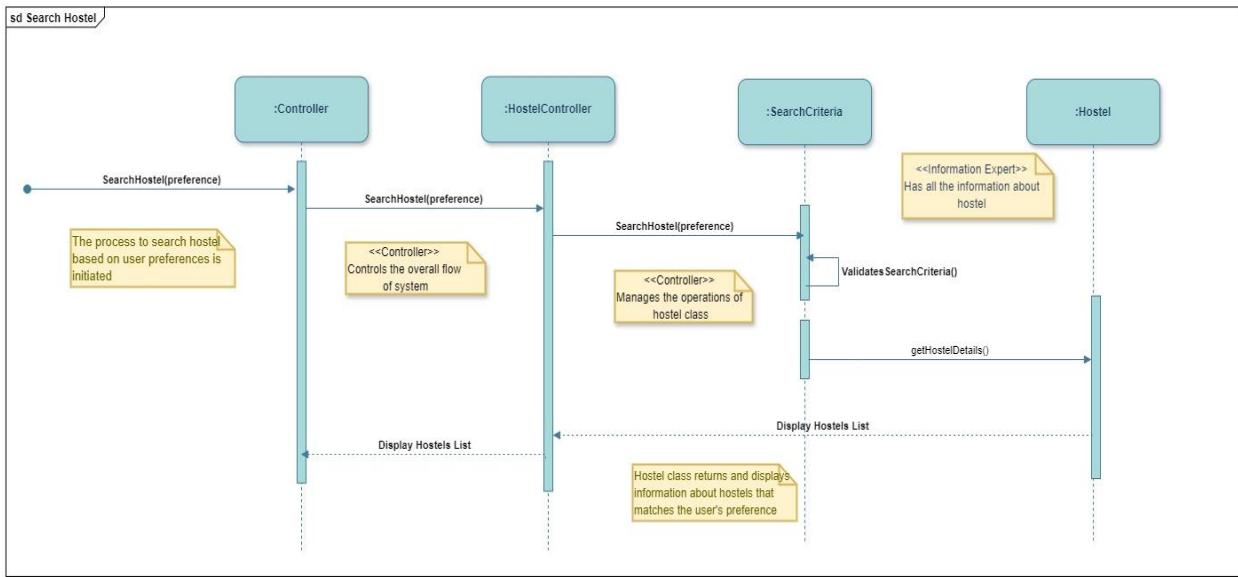


Figure 35 SD- Search Hostel

### 6.4.2 Select Hostel

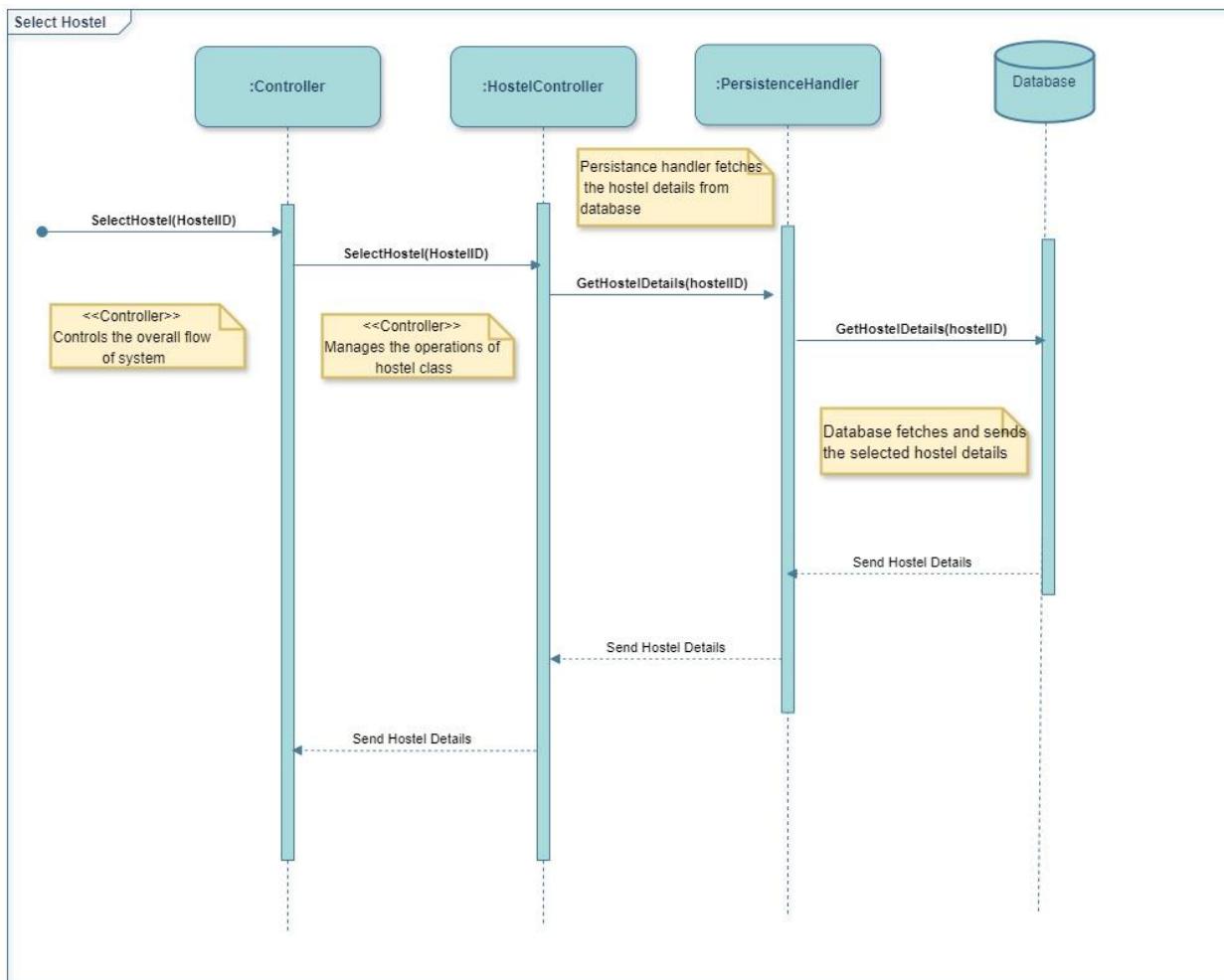


Figure 36 SD- Select Hostel

## 6.5. Managing Resident Information

### 6.5.1 Select Resident

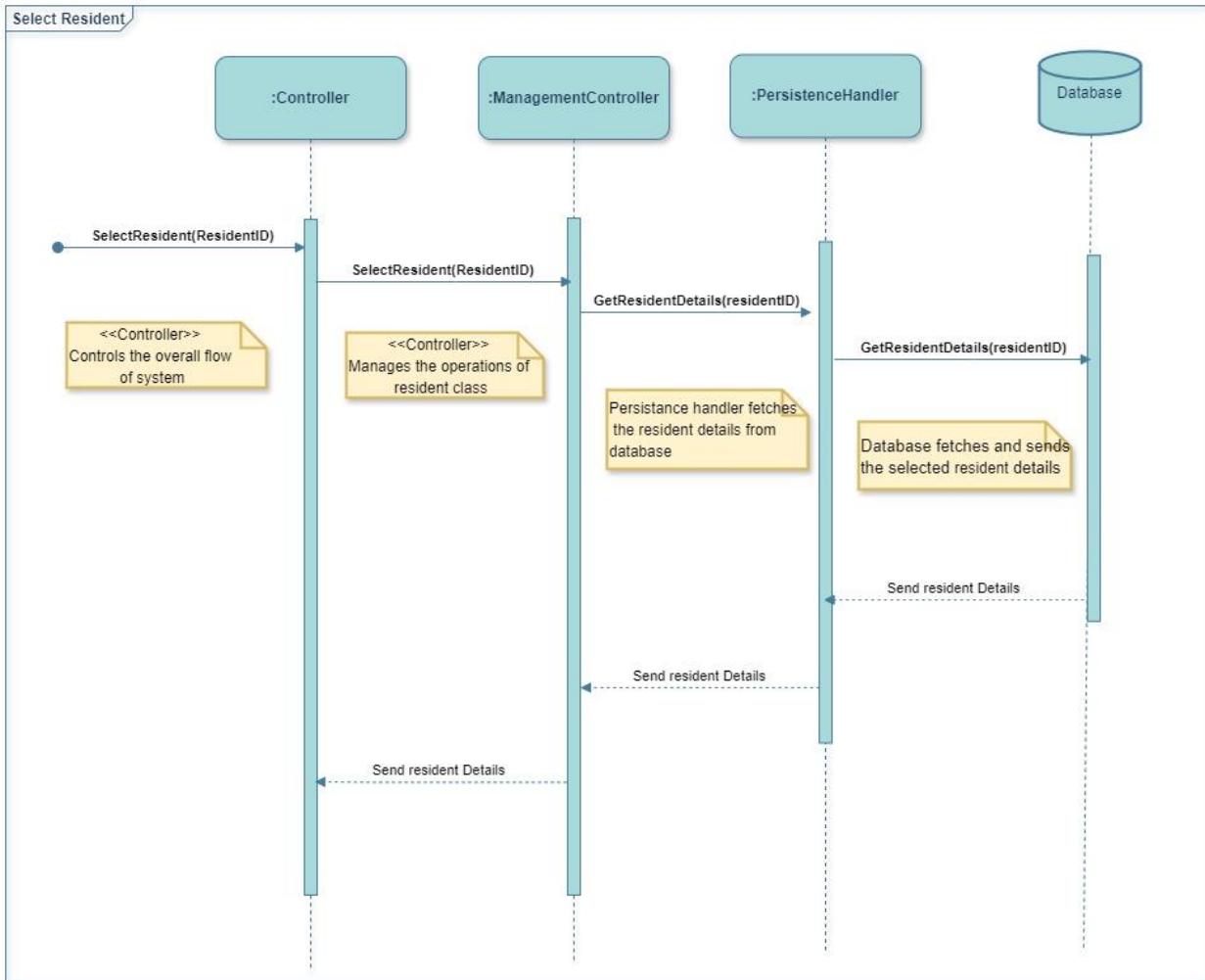


Figure 37 SD- Select Resident

### 6.5.3 Change Resident Information

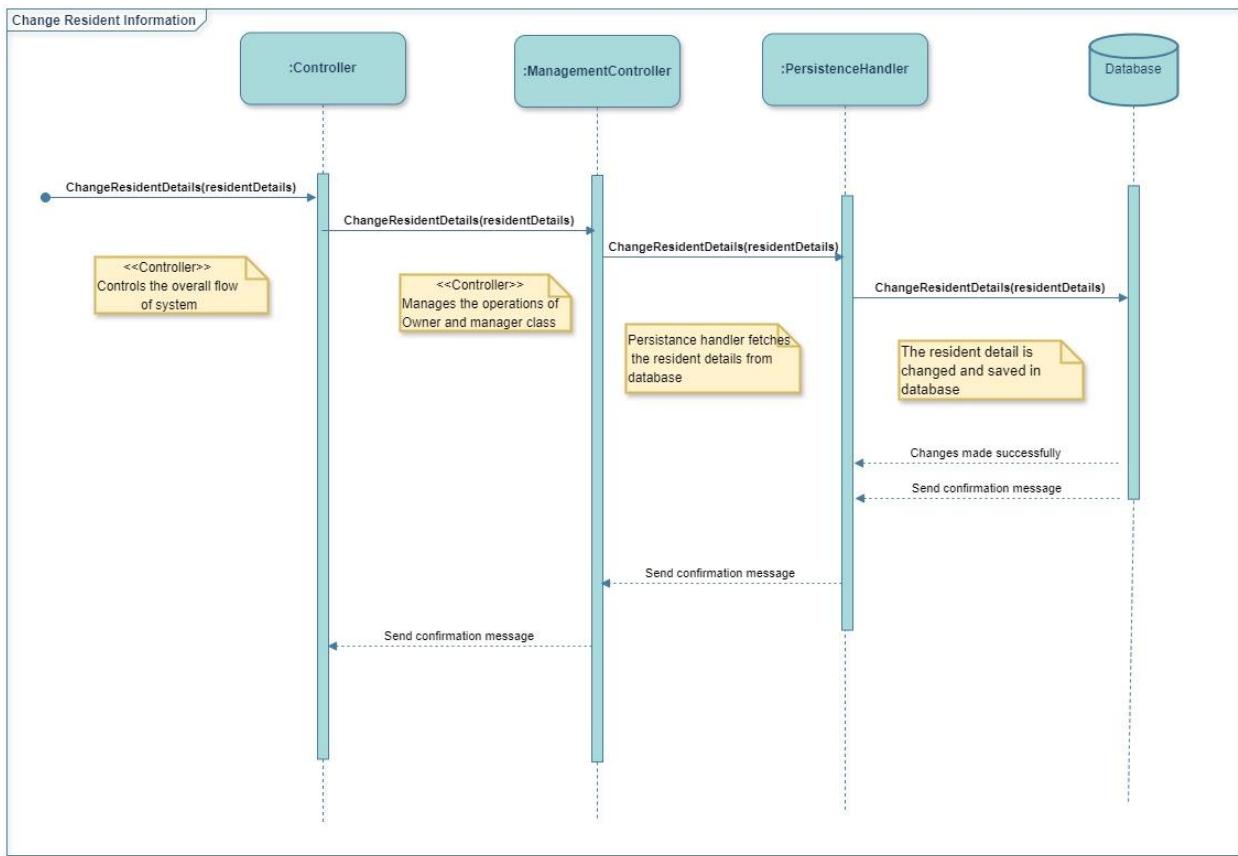


Figure 38 SD- Change Resident Information

## 6.6. Advertise Hostel

### 6.6.1 Create Advertisement

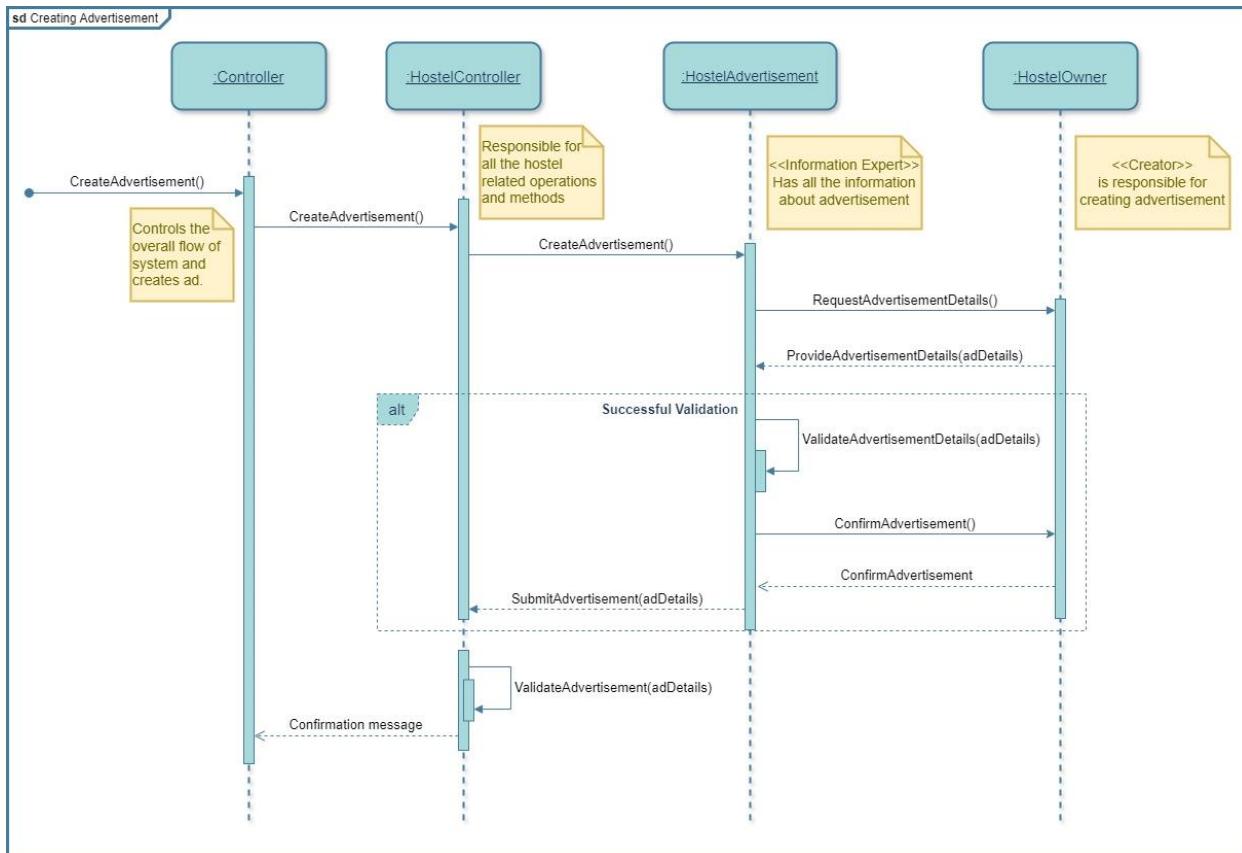


Figure 39 SD-Create Advertisement

### 6.6.2 Store Advertisement

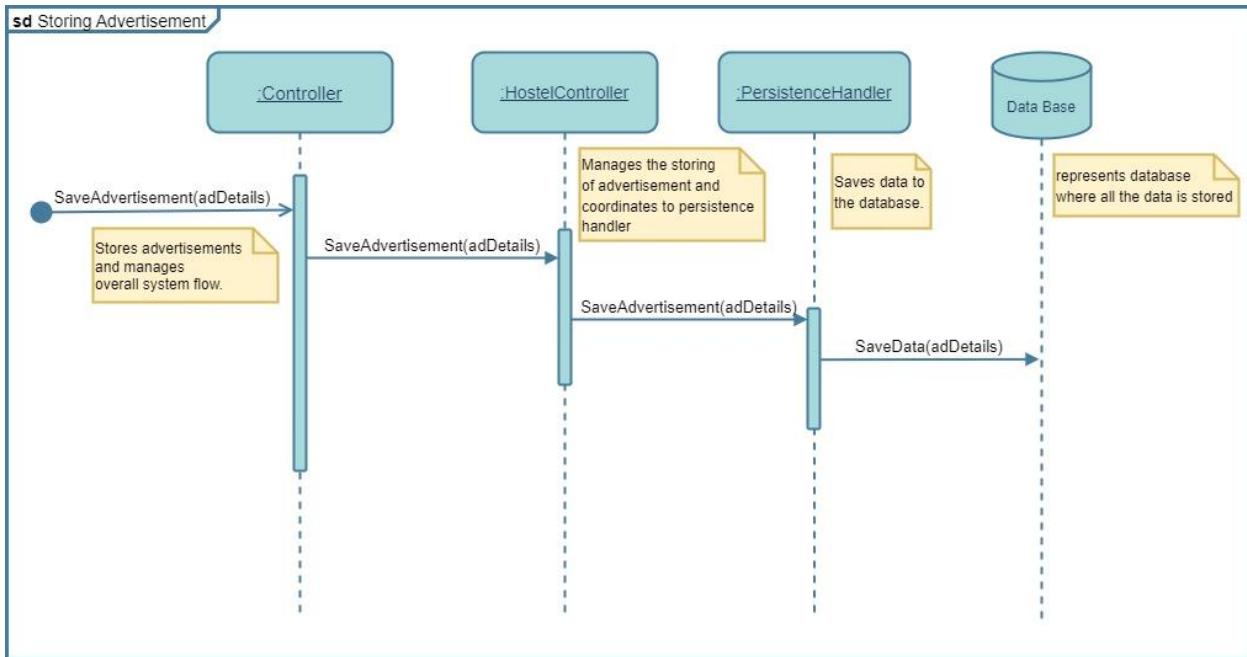


Figure 40 SD- Store Advertisement

## 6.7. Comparing Hostels

### 6.7.1 Compare Criteria

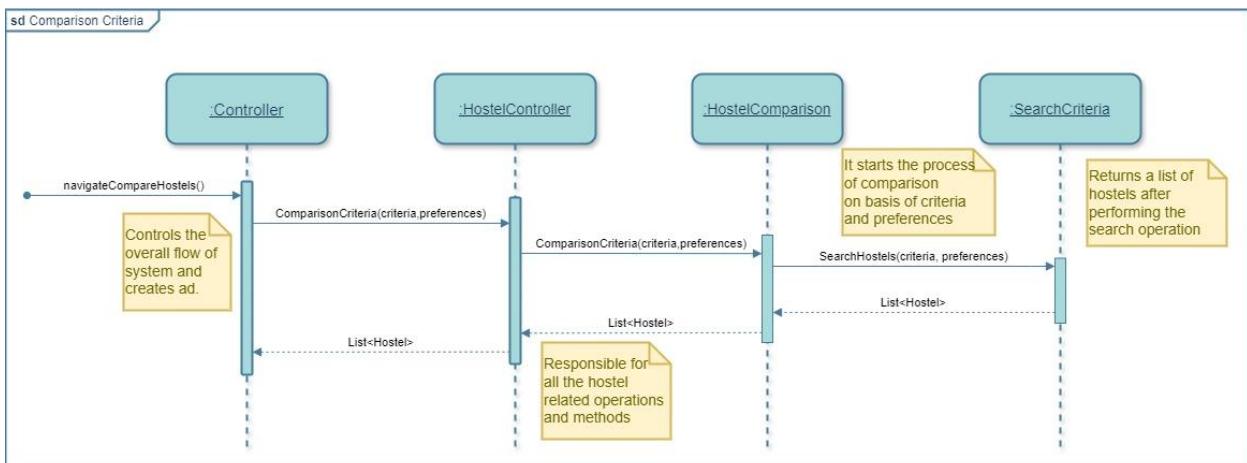


Figure 41 SD- Compare Criteria

### 6.7.2 Select Hostels

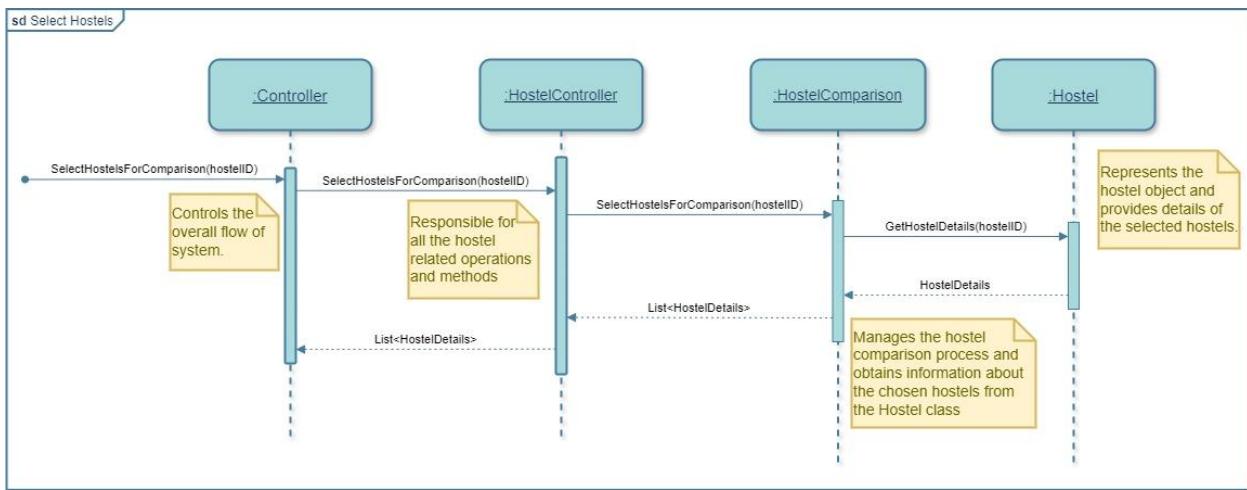


Figure 42 SD-Select Hotels

### 6.7.3 Take Actions

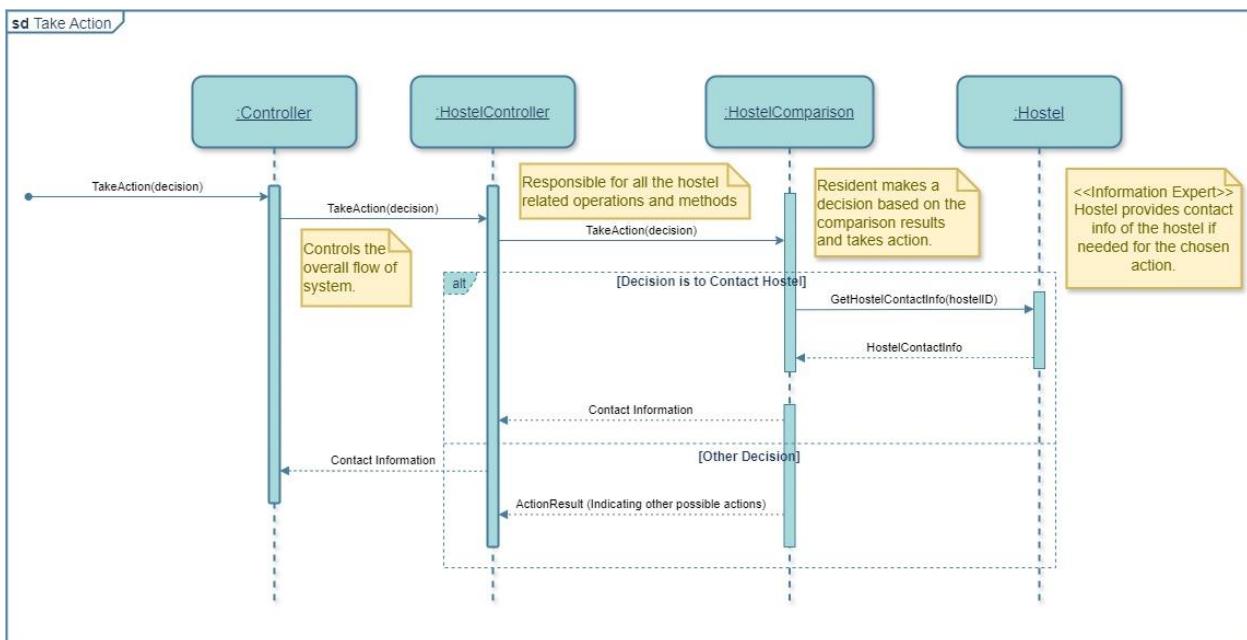


Figure 43 SD- Take Actions

## 6.8. Managing Complaints and Maintenance Requests

### 6.8.1. Retrieve Forms

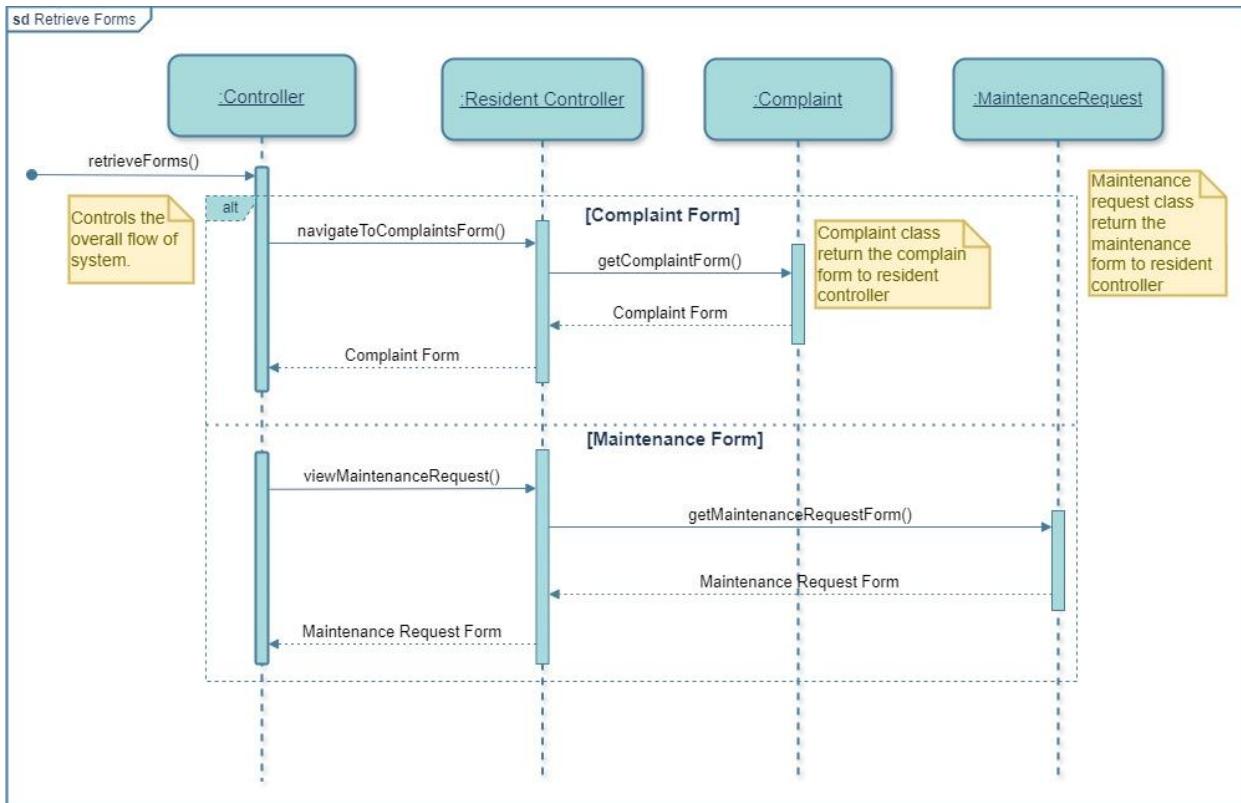


Figure 44 SD- Retrieve Forms

### 6.8.2 Form Submission

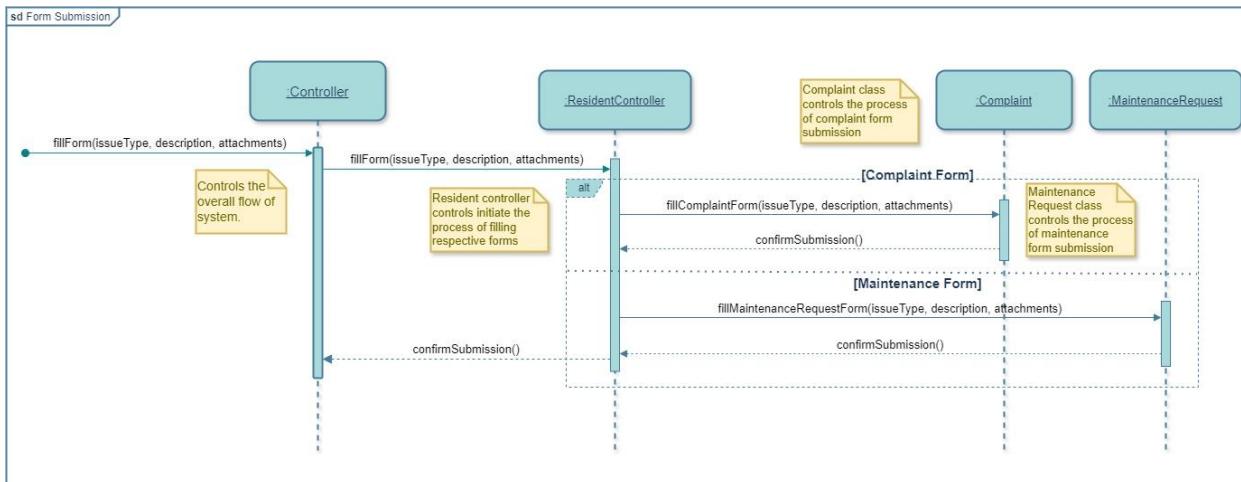


Figure 45 SD- Form Submission

### 6.8.3 Confirm Submission

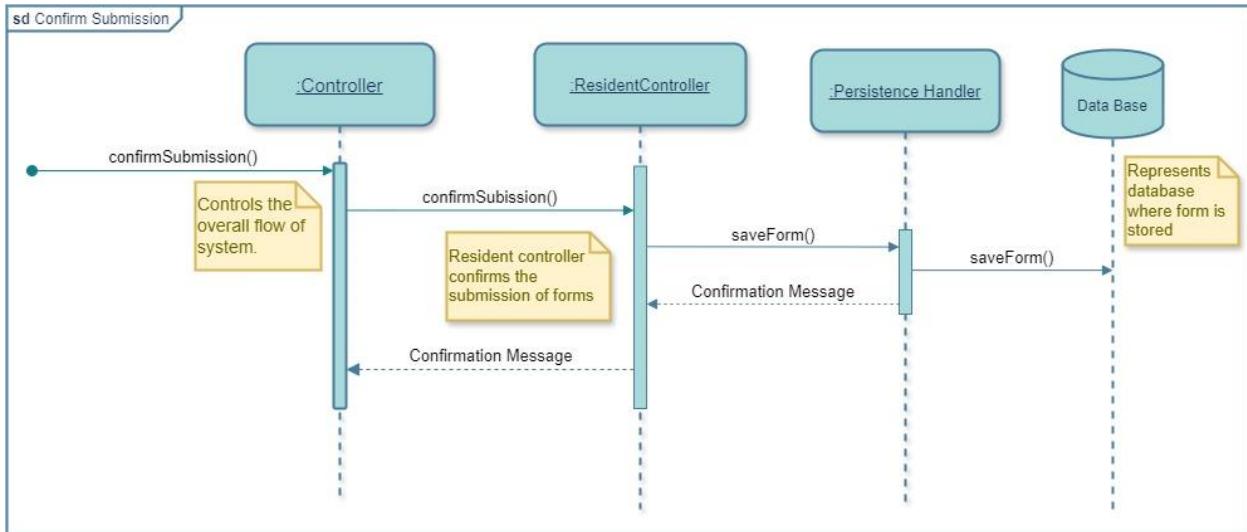


Figure 46 SD- Confirm Submission

## 6.9. Submit feedback and reviews

### 6.9.1 Retrieve Feedback Form

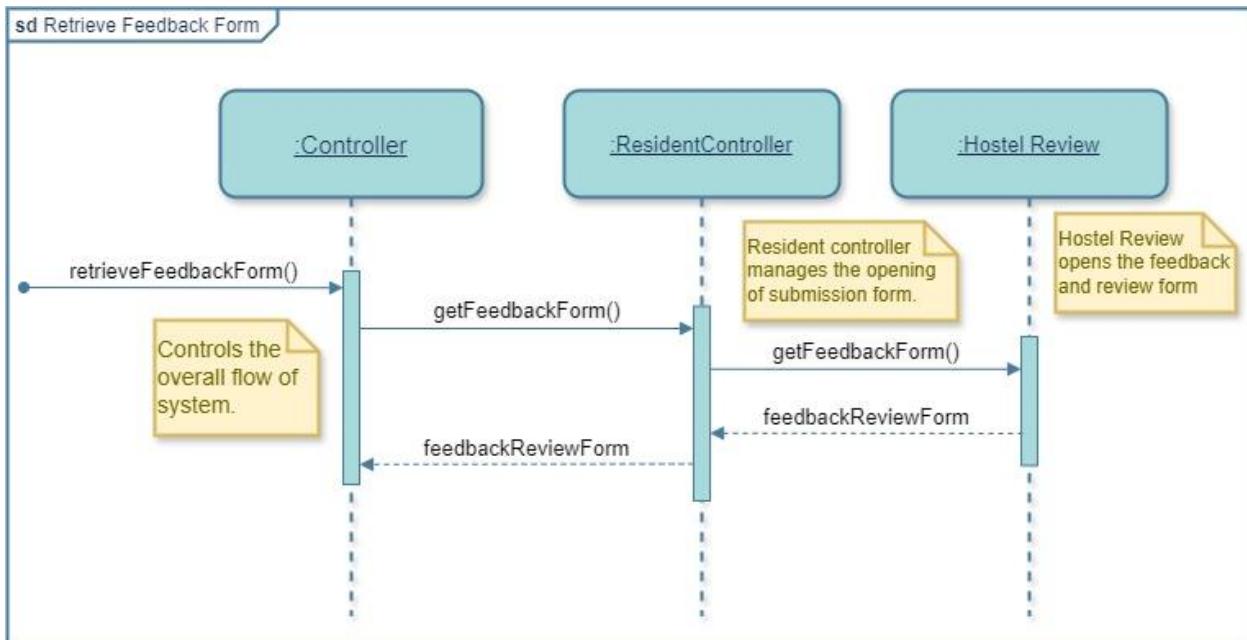


Figure 47 SD-Retrieve Feedback Form

### 6.9.2 Submit Form

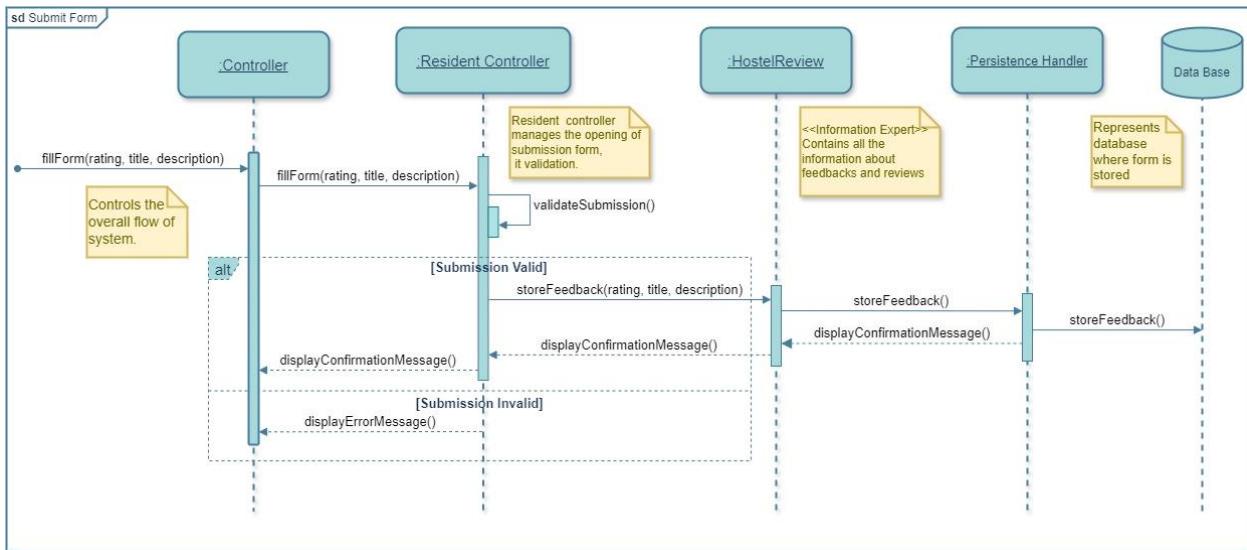


Figure 48 SD- Submit Form

### 6.9.3 Send Notification

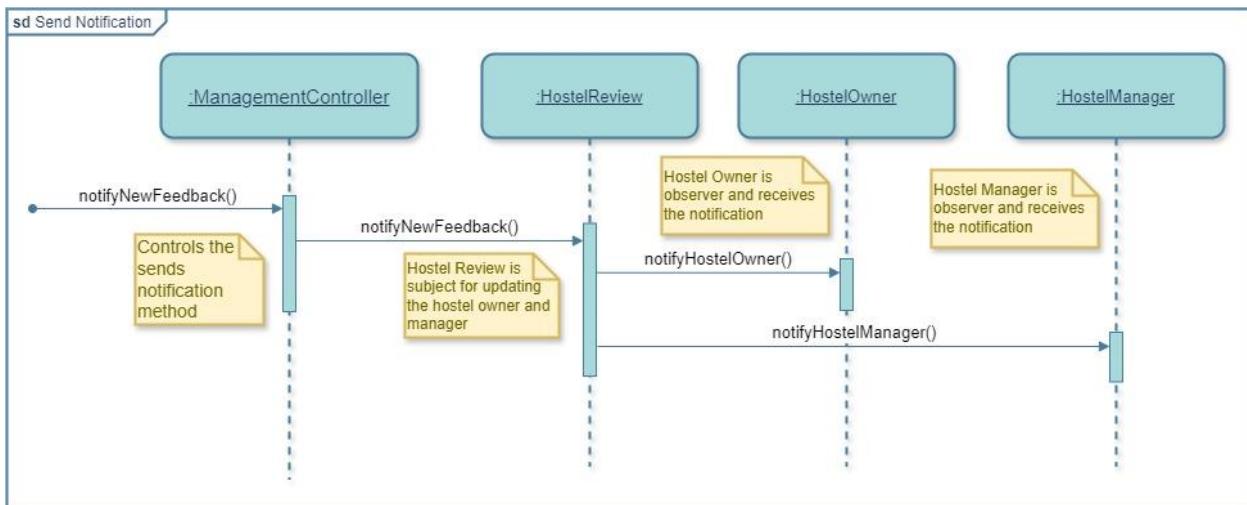


Figure 49 SD- Send Notification

## 6.10. Register User

### 6.10.1 Get Registration Form

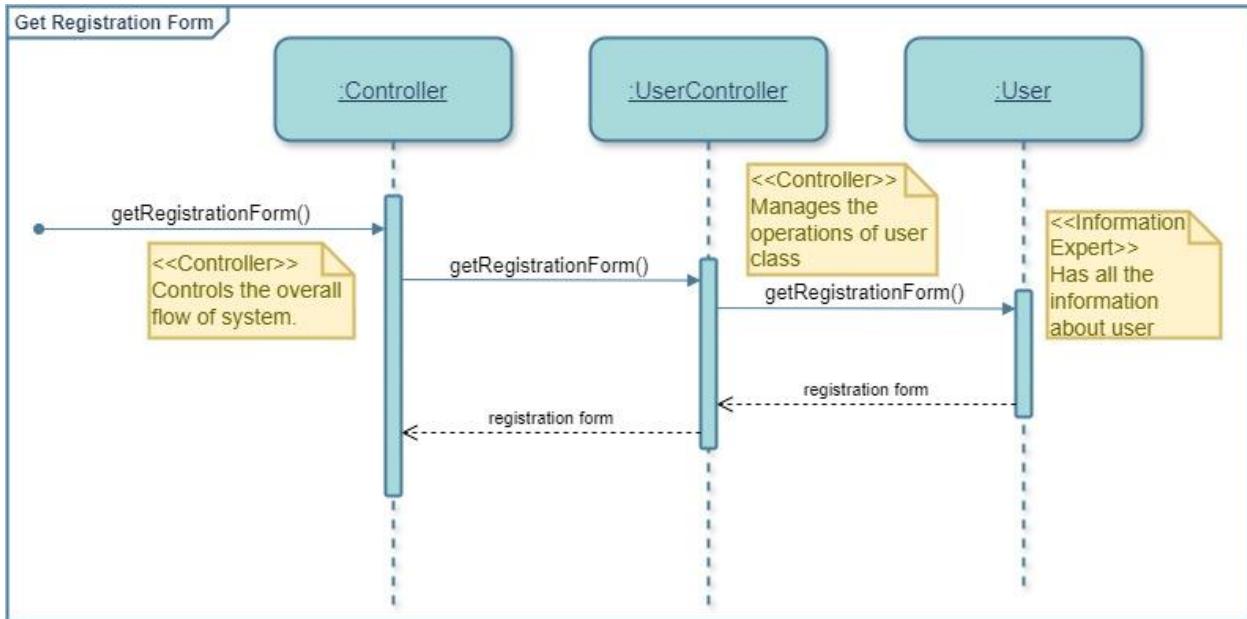


Figure 50 SD- Get Registration Form

### 6.10.2 Fill Registration Form

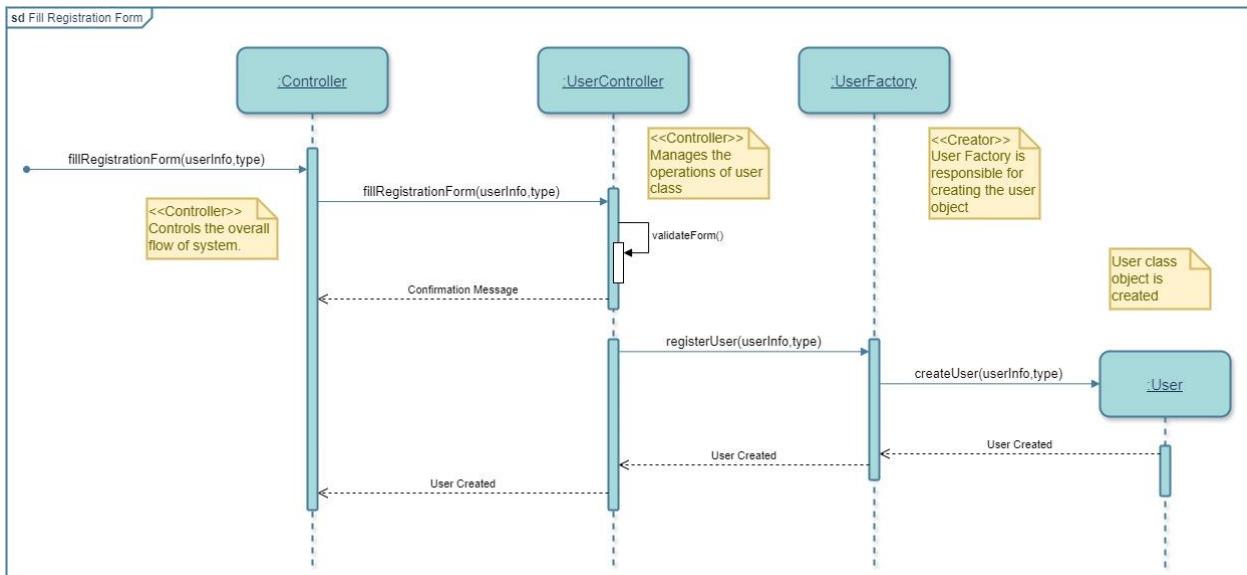


Figure 51 SD-Fill Registration Form

### 6.10.3 Store User Profile

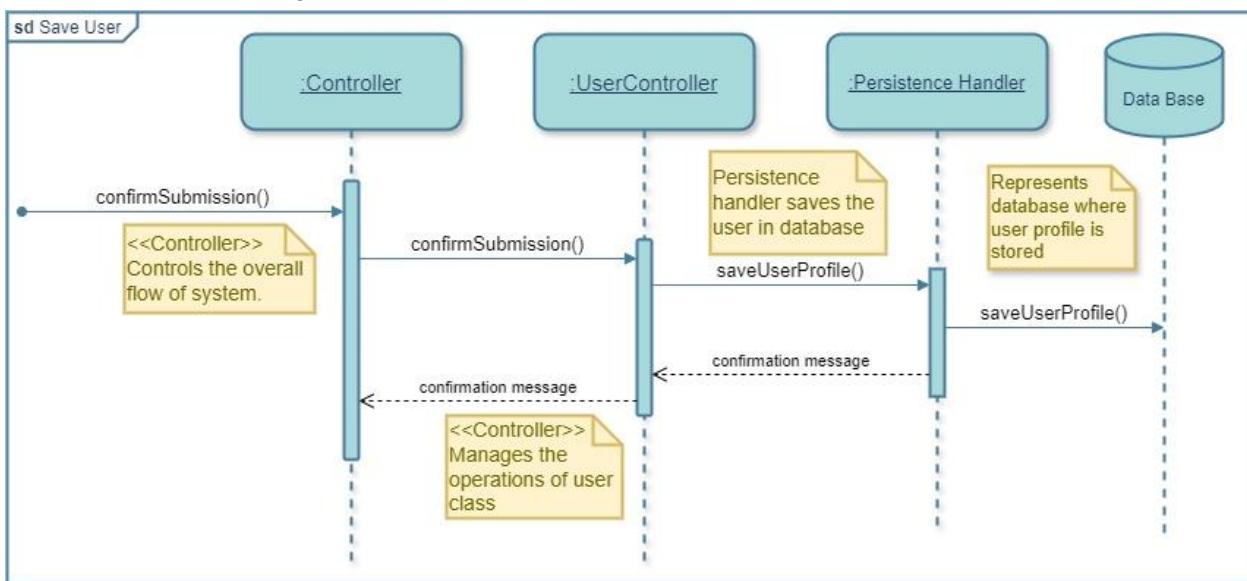


Figure 52 SD- Store User Profile

## 6.11. Manage Fee Details

### 6.11.1 Access and view fee details

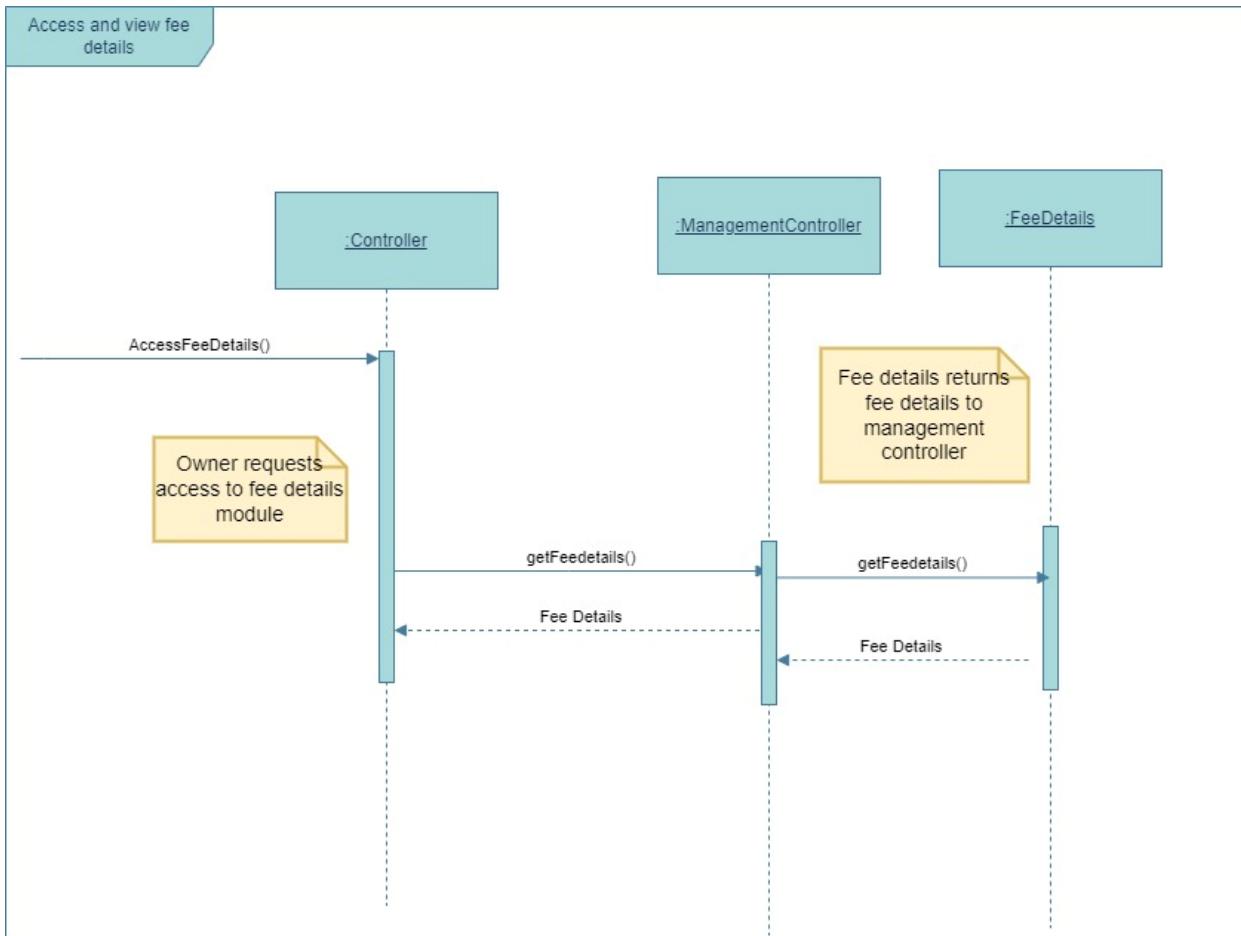


Figure 53 SD- Access and View Fee Details

### 6.11.2 Update fee amount

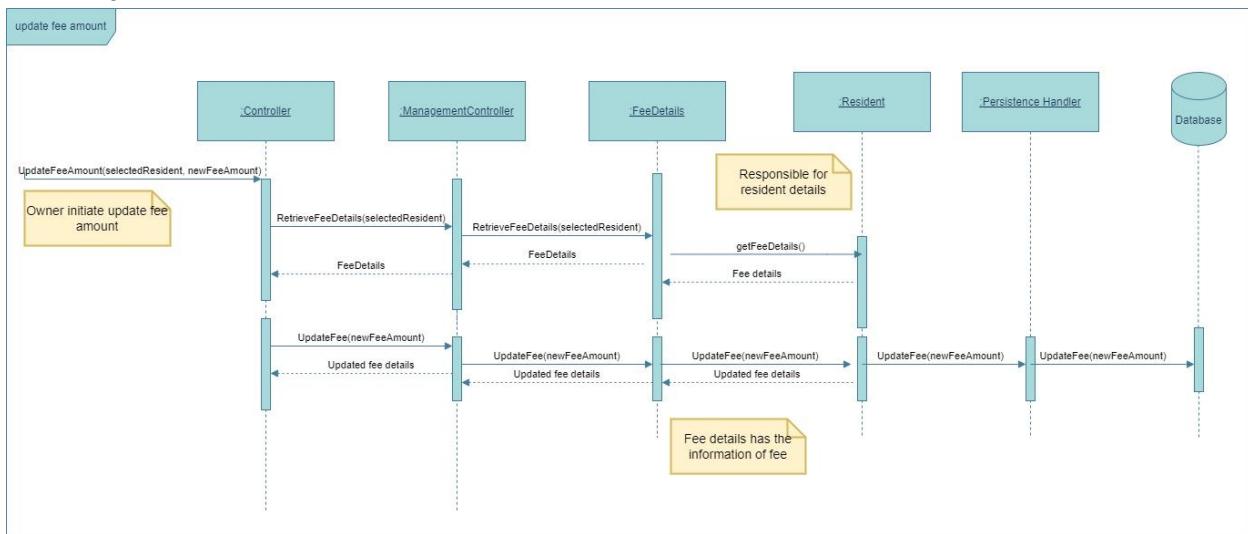


Figure 54 SD- Update Fee Amount

### 6.12. Avail Transportation Services

#### 6.12.1 Access transportation module

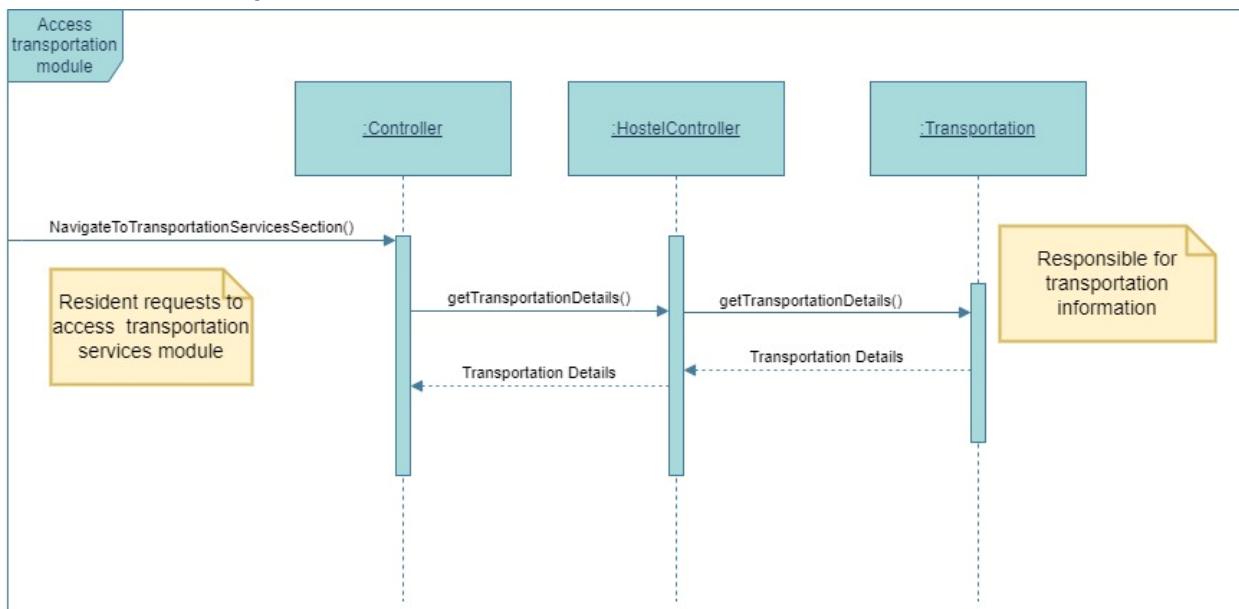


Figure 55 SD- Access Transportation Module

### 6.12.2 Select desired date and time

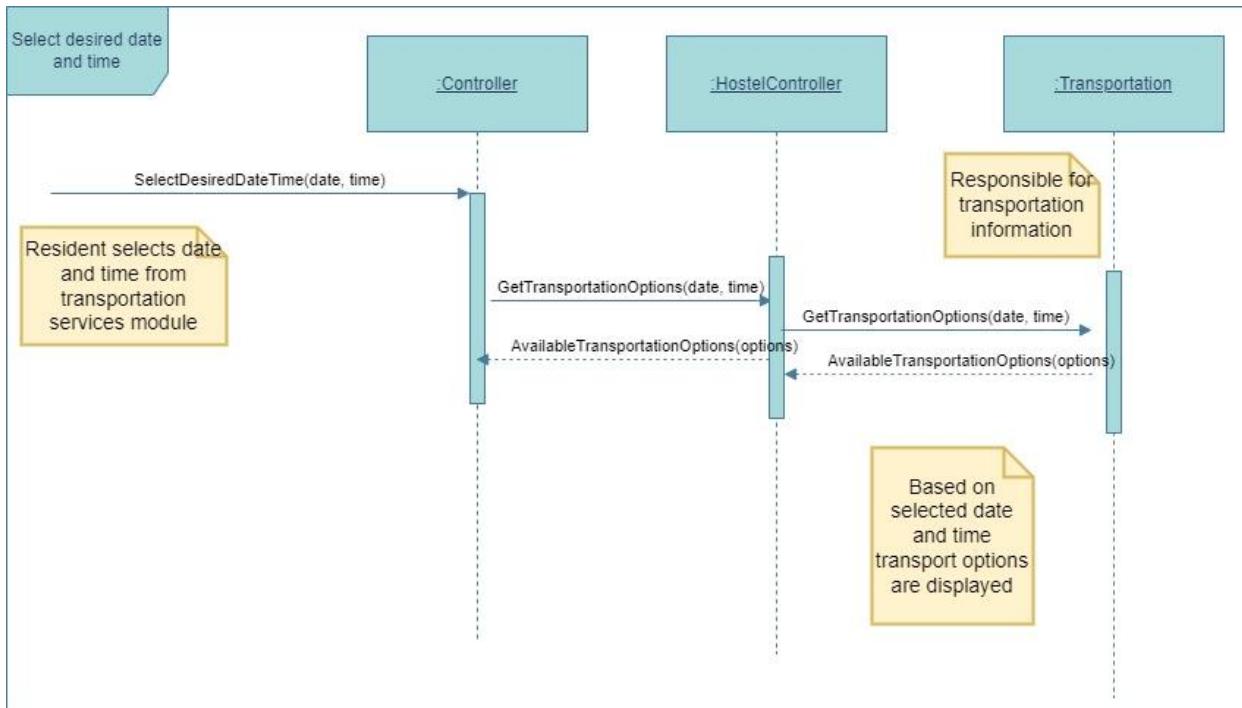


Figure 56 SD- Select Desired Date and Time

### 6.12.3 Select transportation

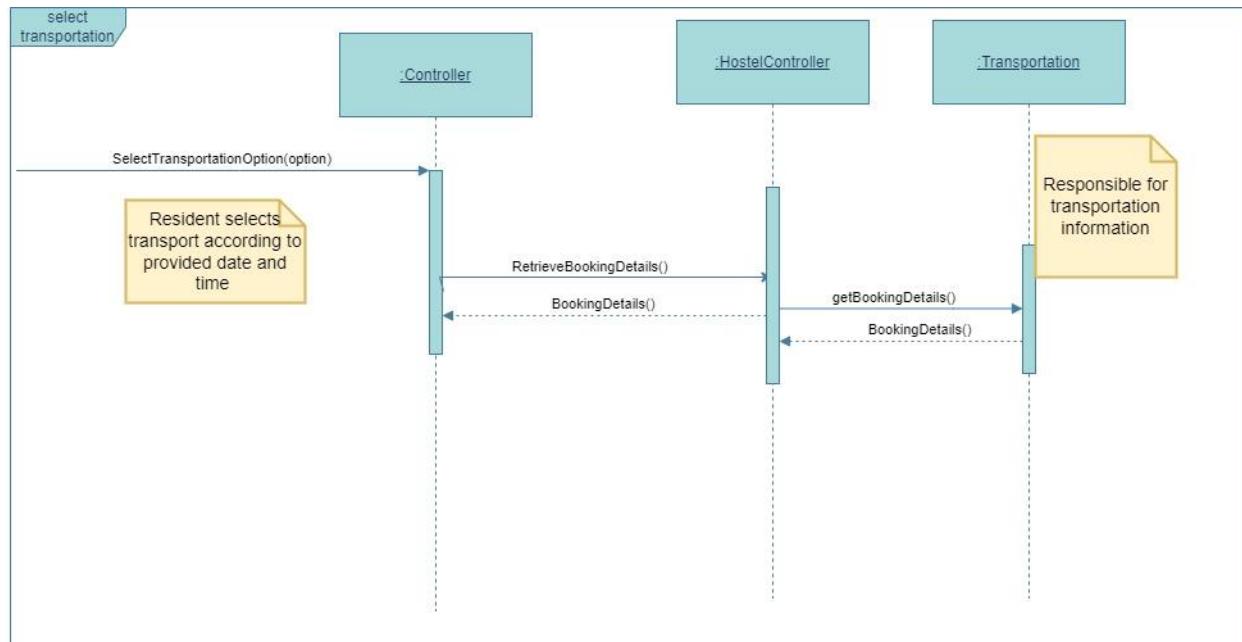


Figure 57 SD- Select Transportation

#### 6.12.4 Confirm Booking

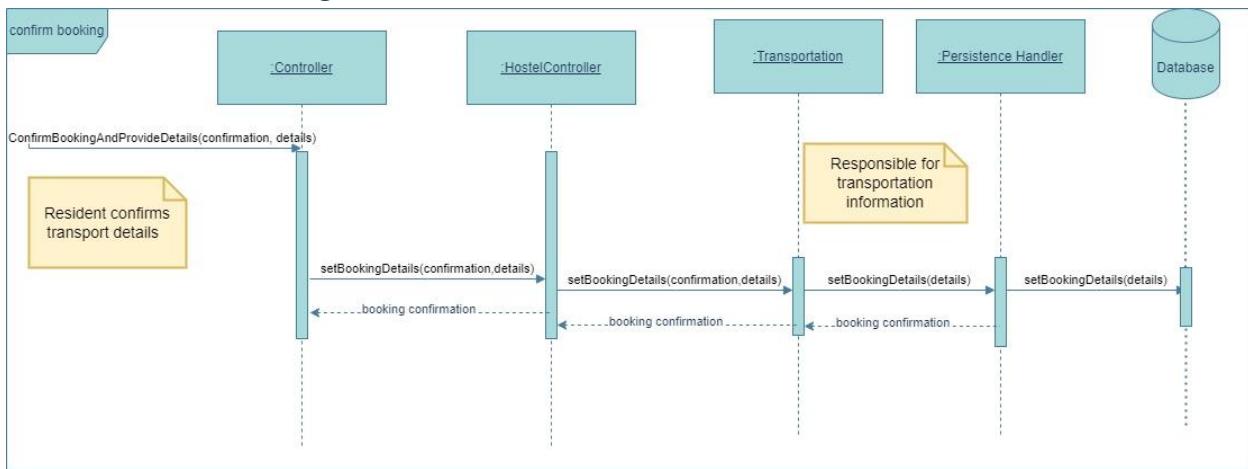


Figure 58 SD- Confirm Booking

#### 6.13. Allocate Rooms

##### 6.13.1 Access room allocation

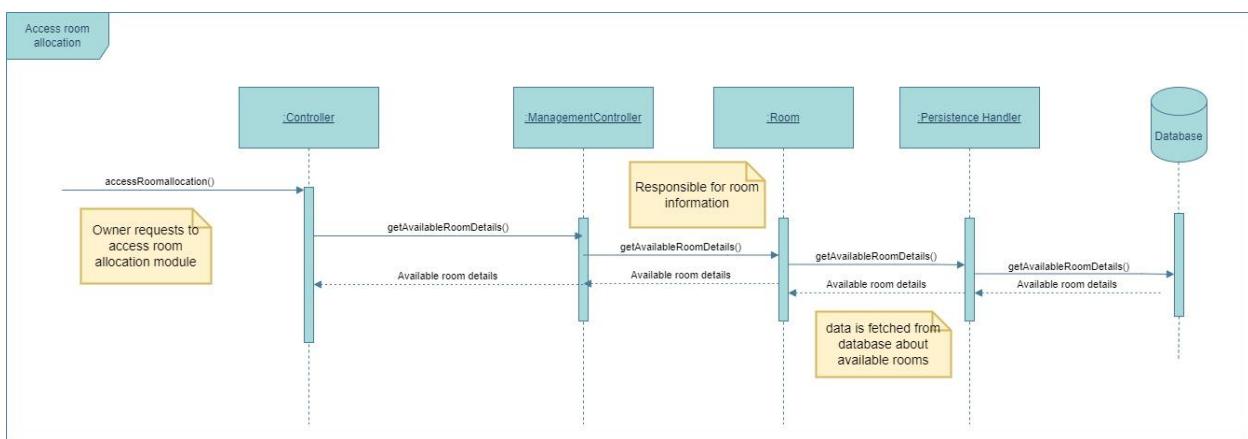


Figure 59 SD- Access Room Allocation

### 6.13.2 Select room for allocation

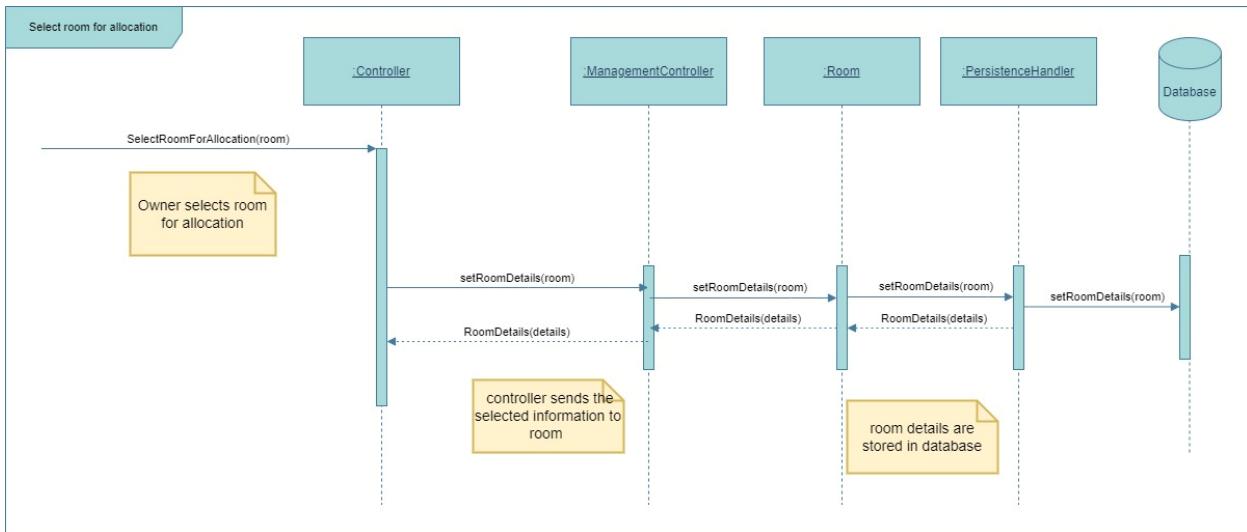


Figure 60 SD- Select Room for Allocation

### 6.13.3 Select room for changing occupancy status

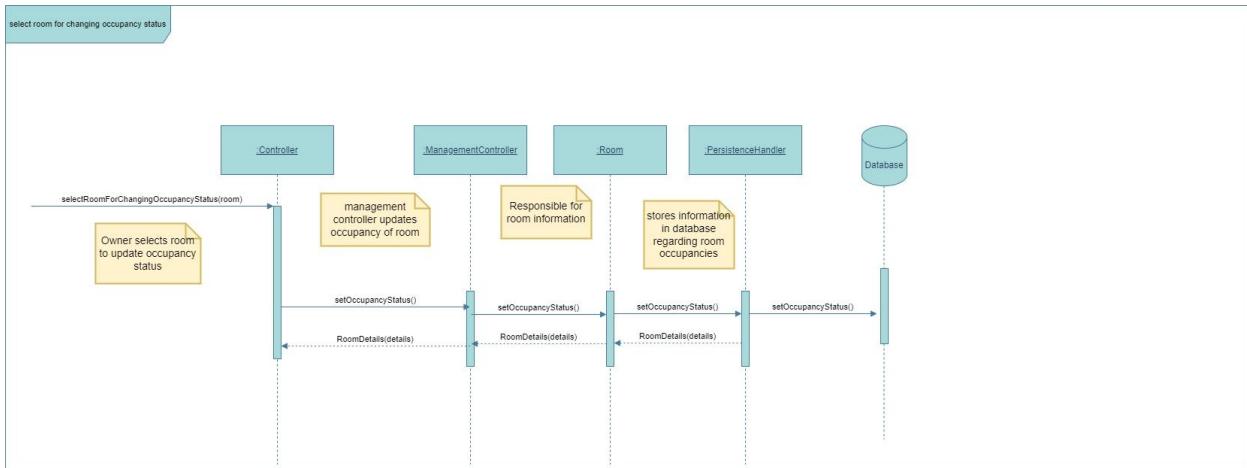


Figure 61 SD- Select Room for Changing Occupancy

## 6.14. Message and Communication

### 6.14.1 Compose Message

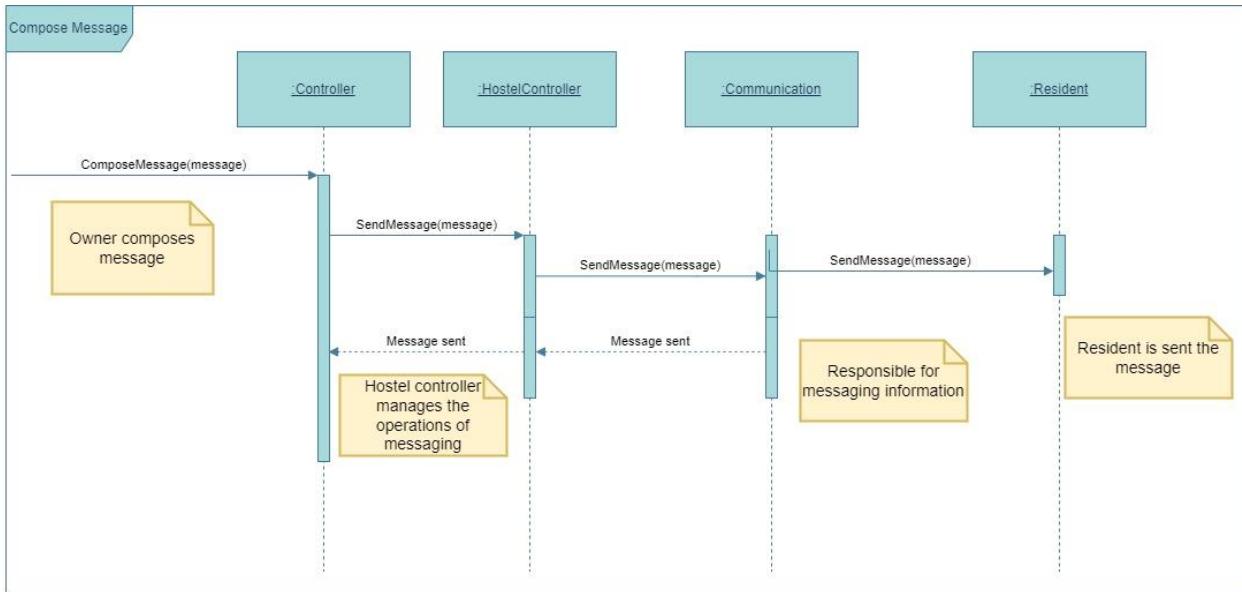


Figure 62 SD- Compose Message

### 6.14.2 View Message

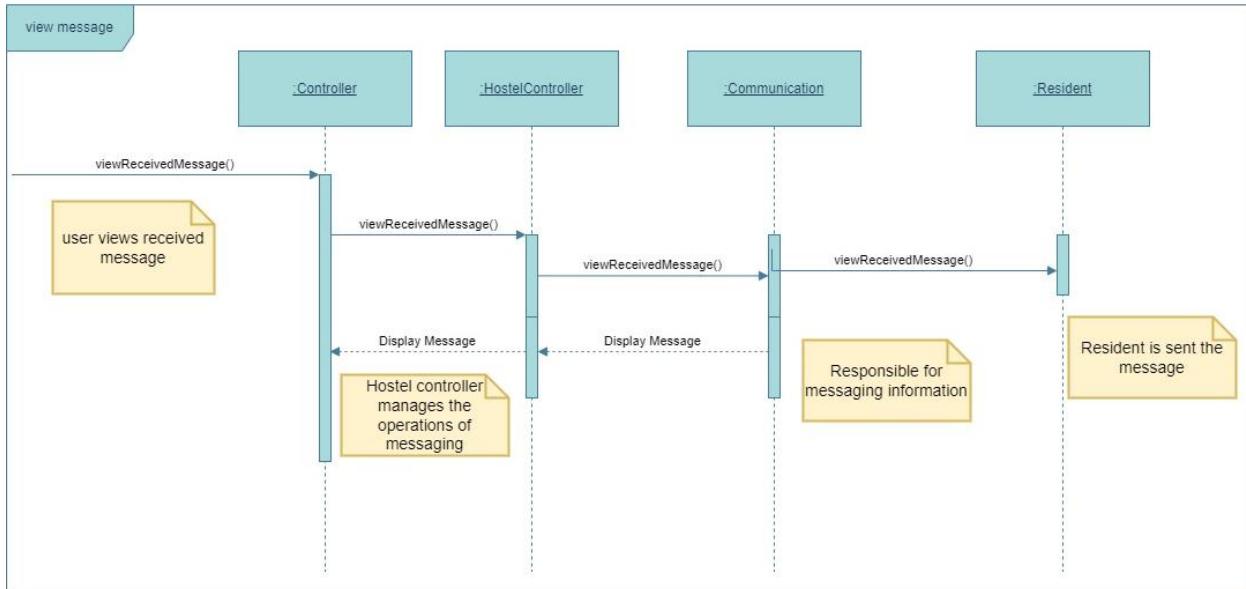


Figure 63 SD-View Message

## 6.15. Manage Leave Requests

### 6.15.1 Access leave management section

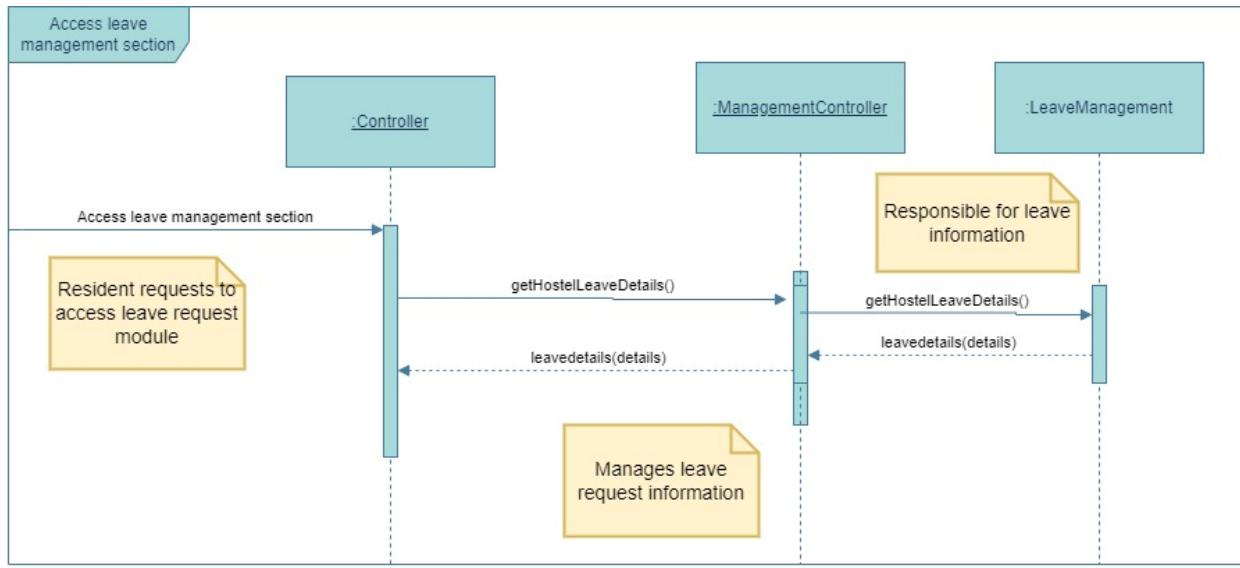


Figure 64 SD-Access Leave Management Section

### 6.15.2 Initiate leave request

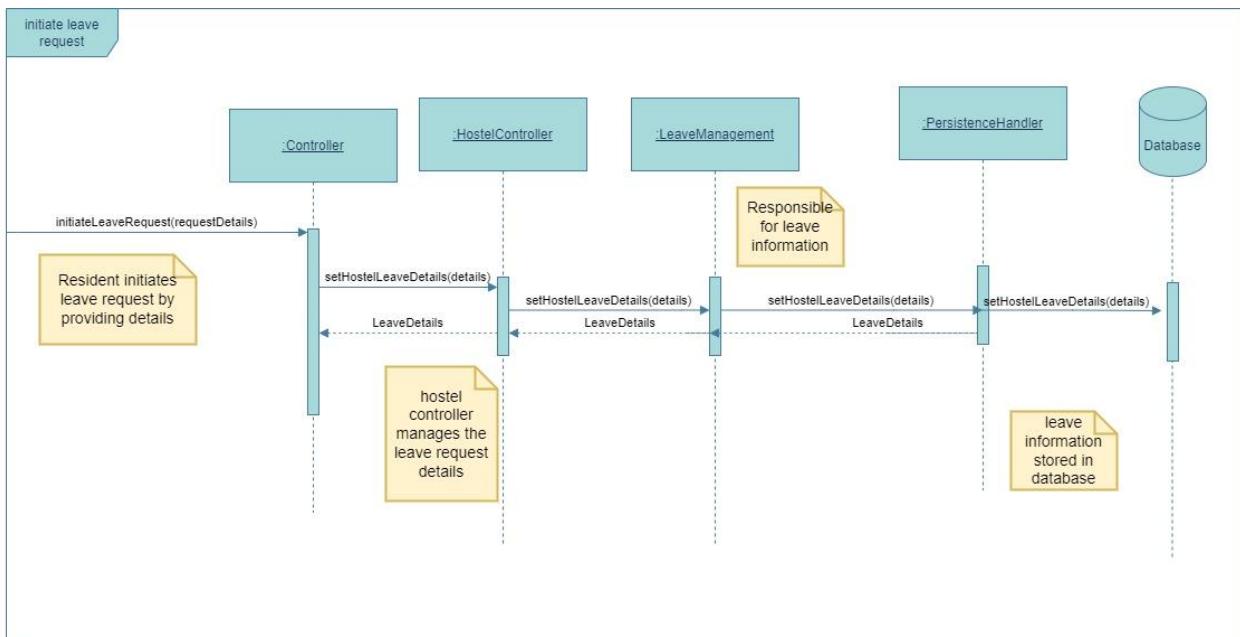


Figure 65 SD-Initiate Leave Request

### 6.15.3 Send notification

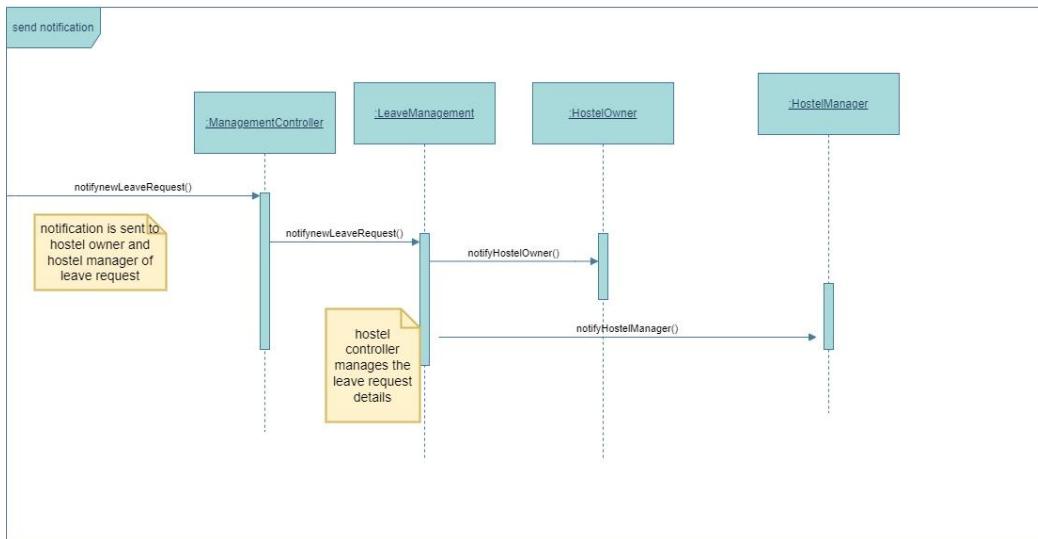


Figure 66 SD-Send Notification

## 6.16. Manage Food Details

### 6.16.1 Access food details

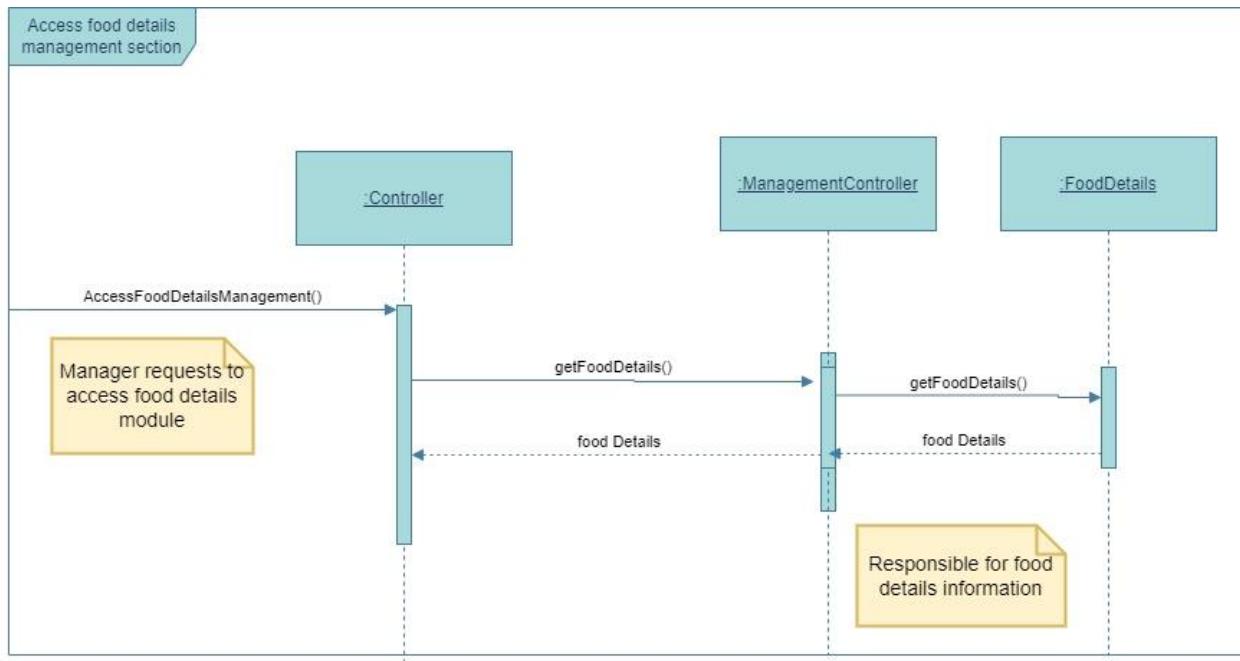


Figure 67 SD-Access Food Details

### 6.16.2 Modify Food menu

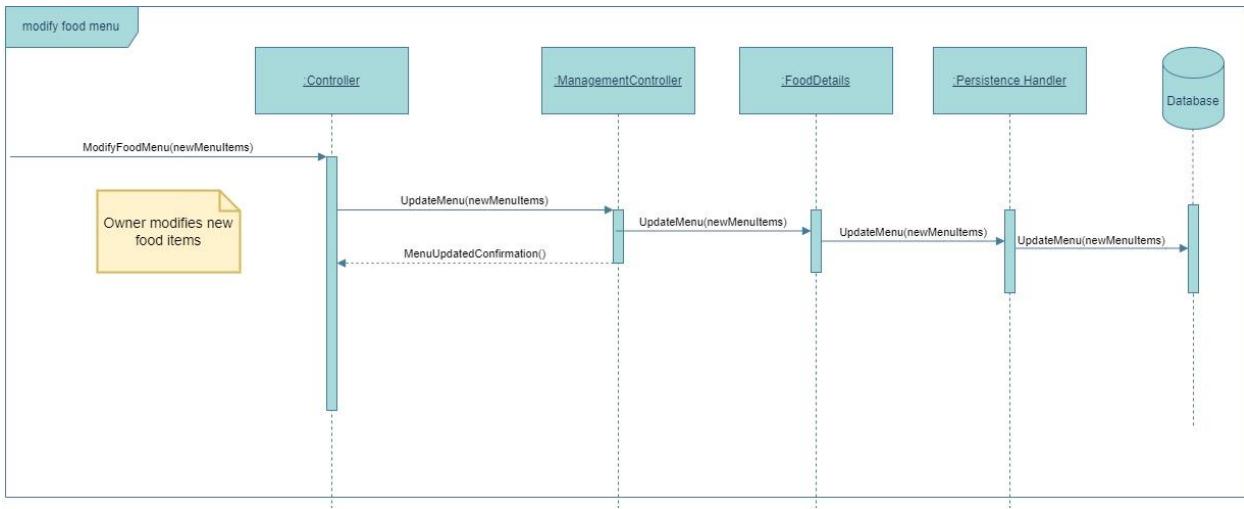


Figure 68 SD- Modify Food Menu

## 7. UPDATED ARTIFCATS

### 7.1 Advertise Hostel

#### 7.1.1 System sequence Diagram

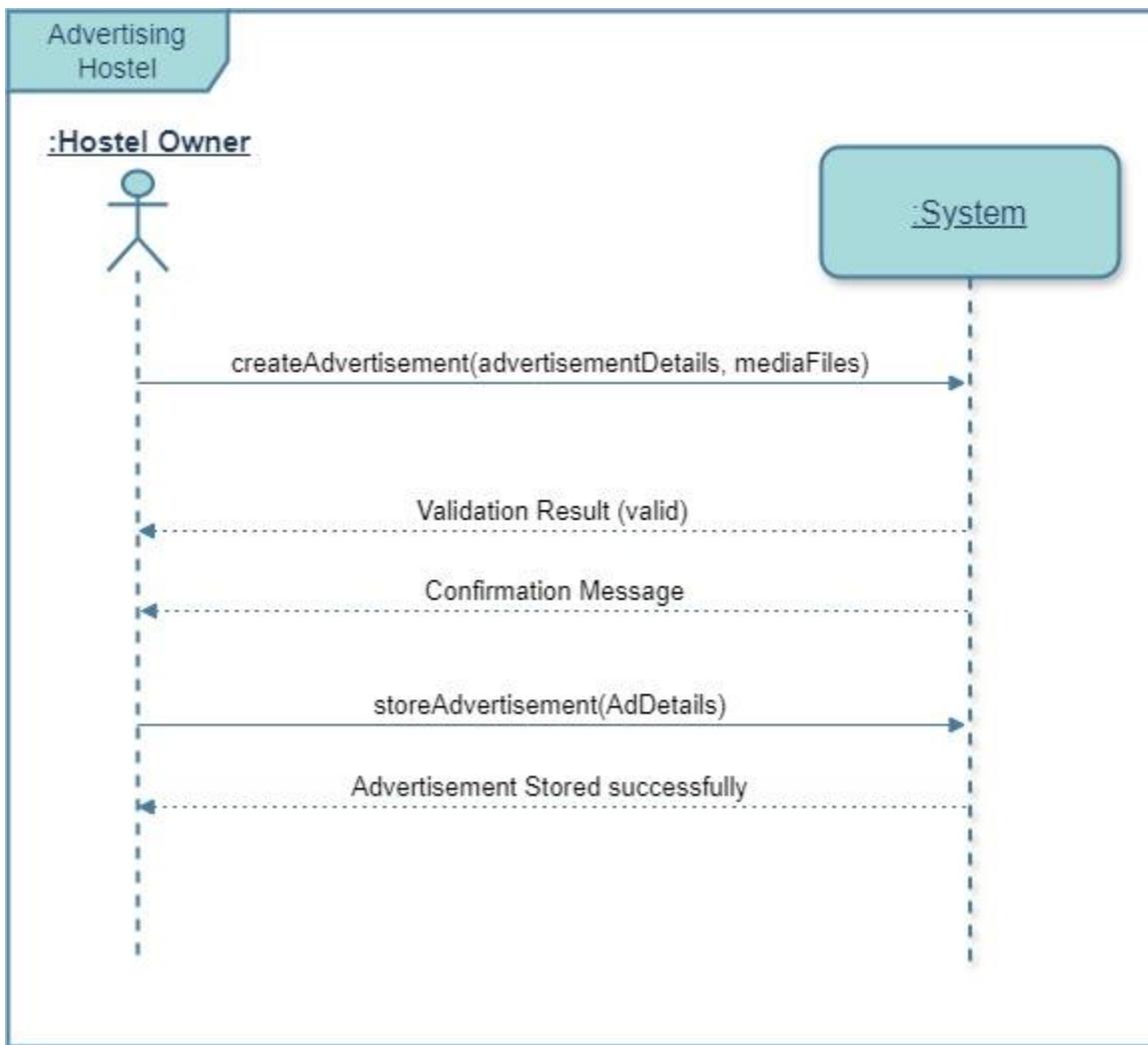


Figure 69 SSD- Advertise Hostel (Updated)

### 7.1.2 Sequence Diagram:

#### a. Create Advertisement

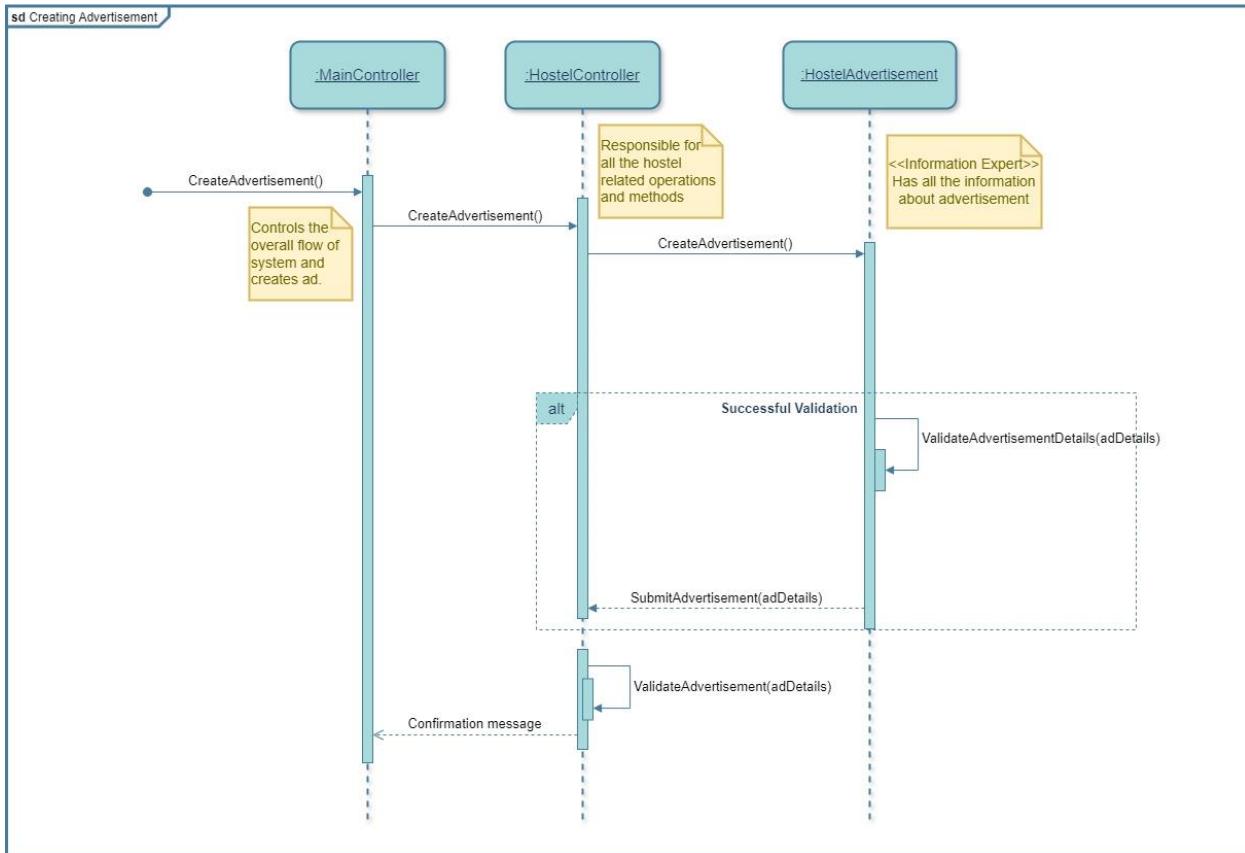


Figure 70 SD- Create Advertisement (updated)

**b. Store Advertisement**

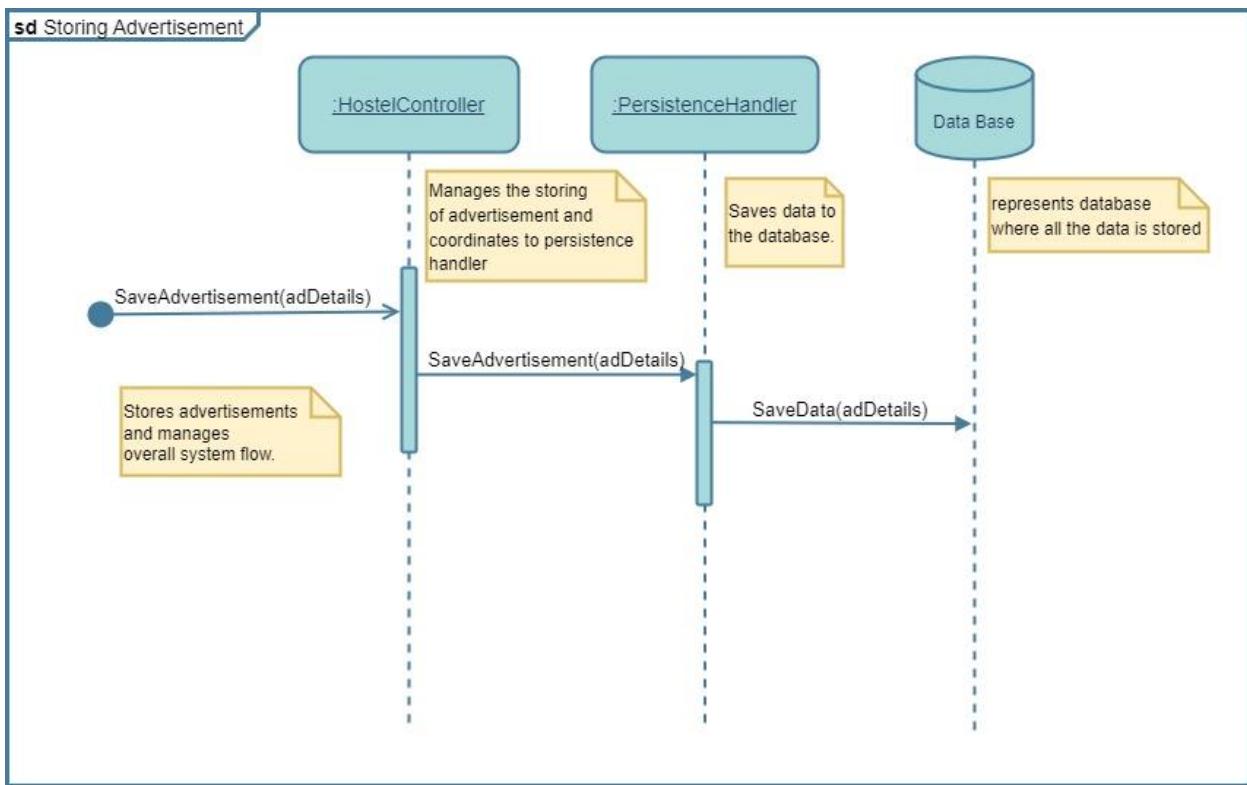


Figure 71 SD- Store Advertisement (updated)

## 7.2 Compare Hostel

### 7.2.1 Sequence Diagram

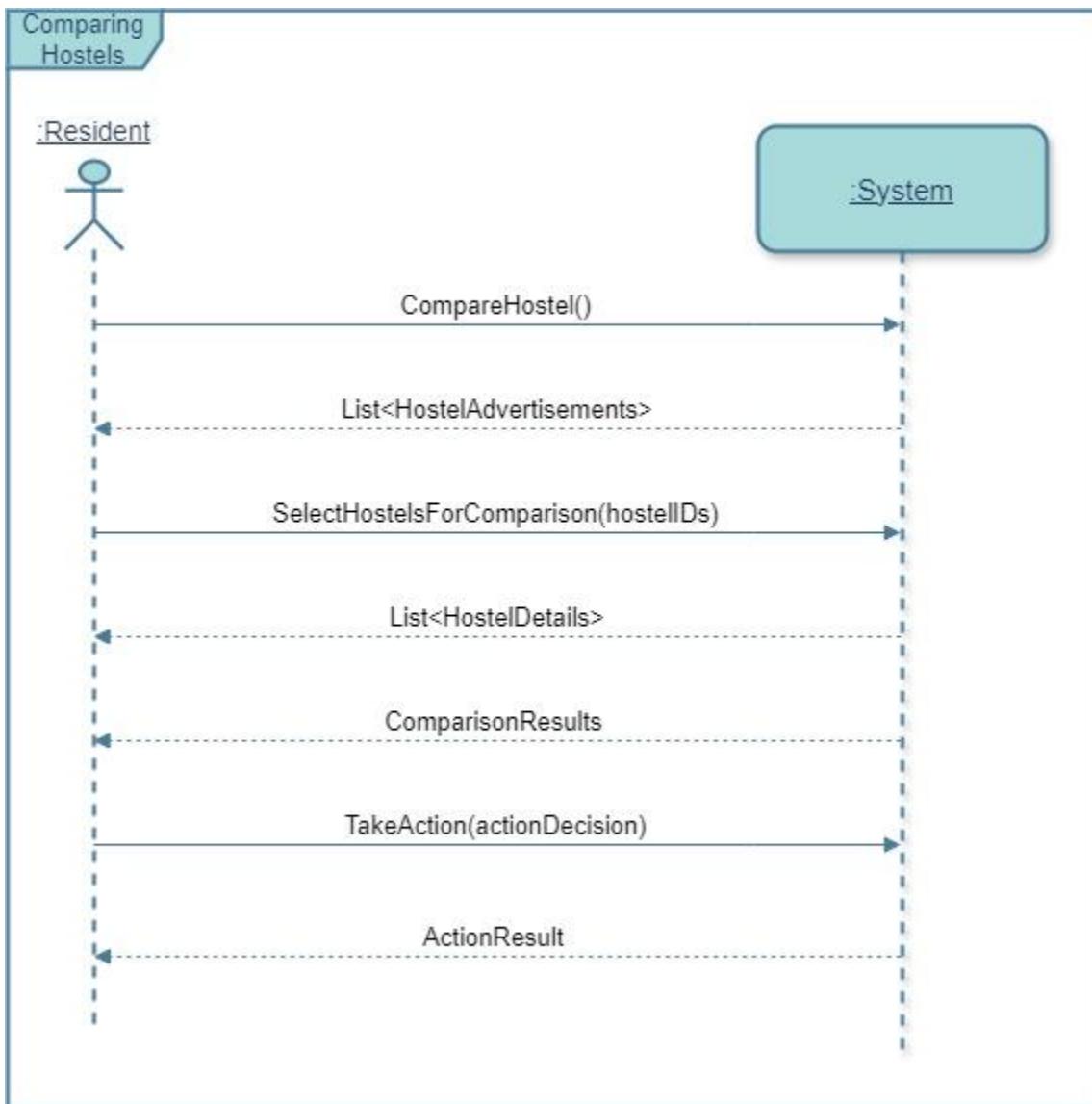


Figure 72 SSD- Compare Hostel (updated)

## 7.2.2 Sequence Diagrams

### a. Compare Criteria

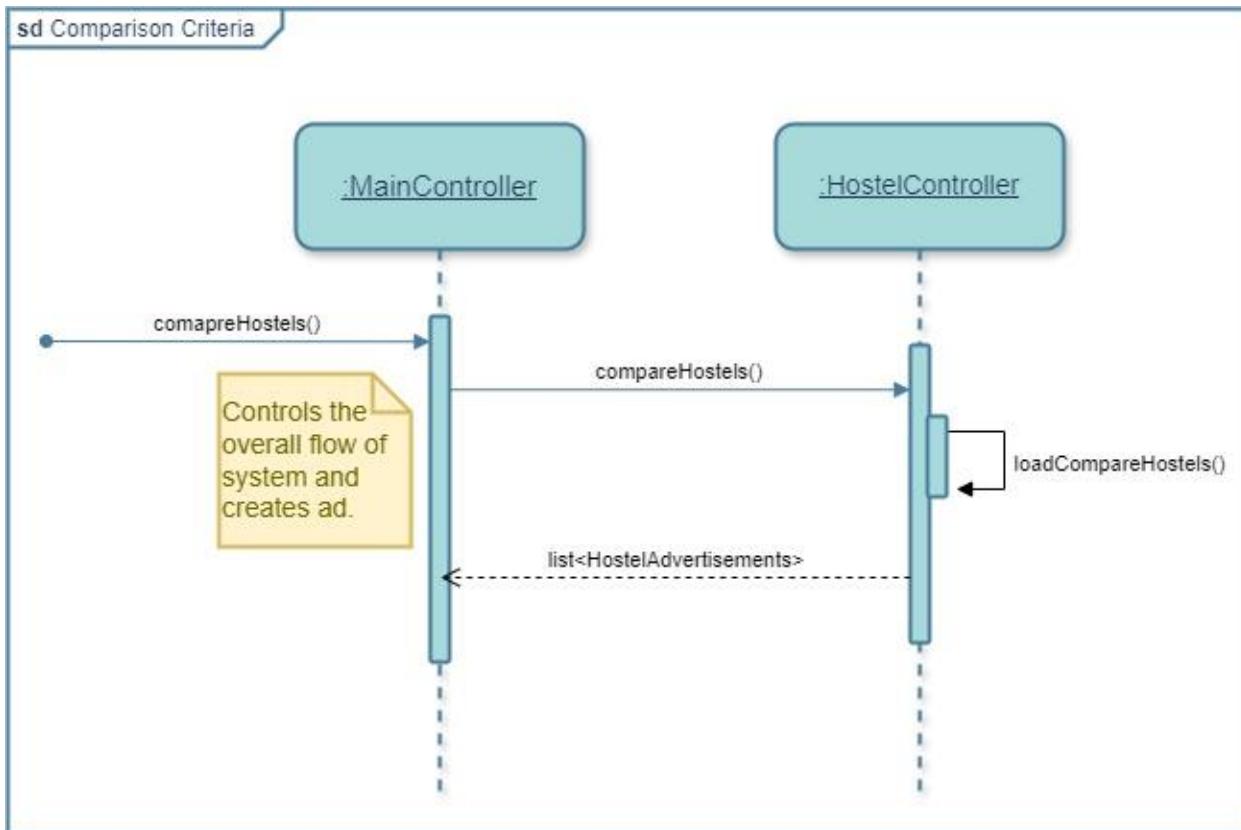


Figure 73 SD- Compare Criteria

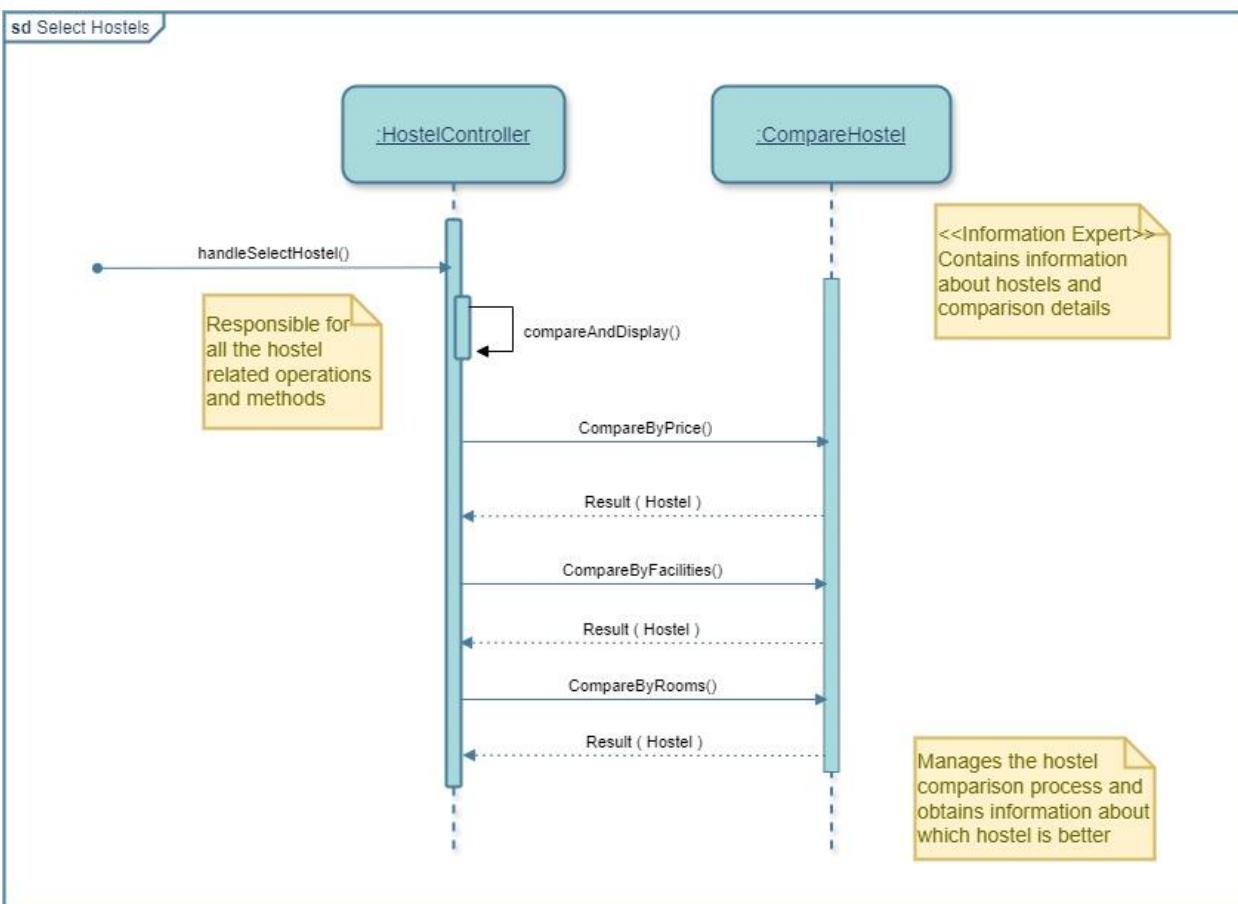
**b. Select Hostel**

Figure 74 SD- Select Hostel

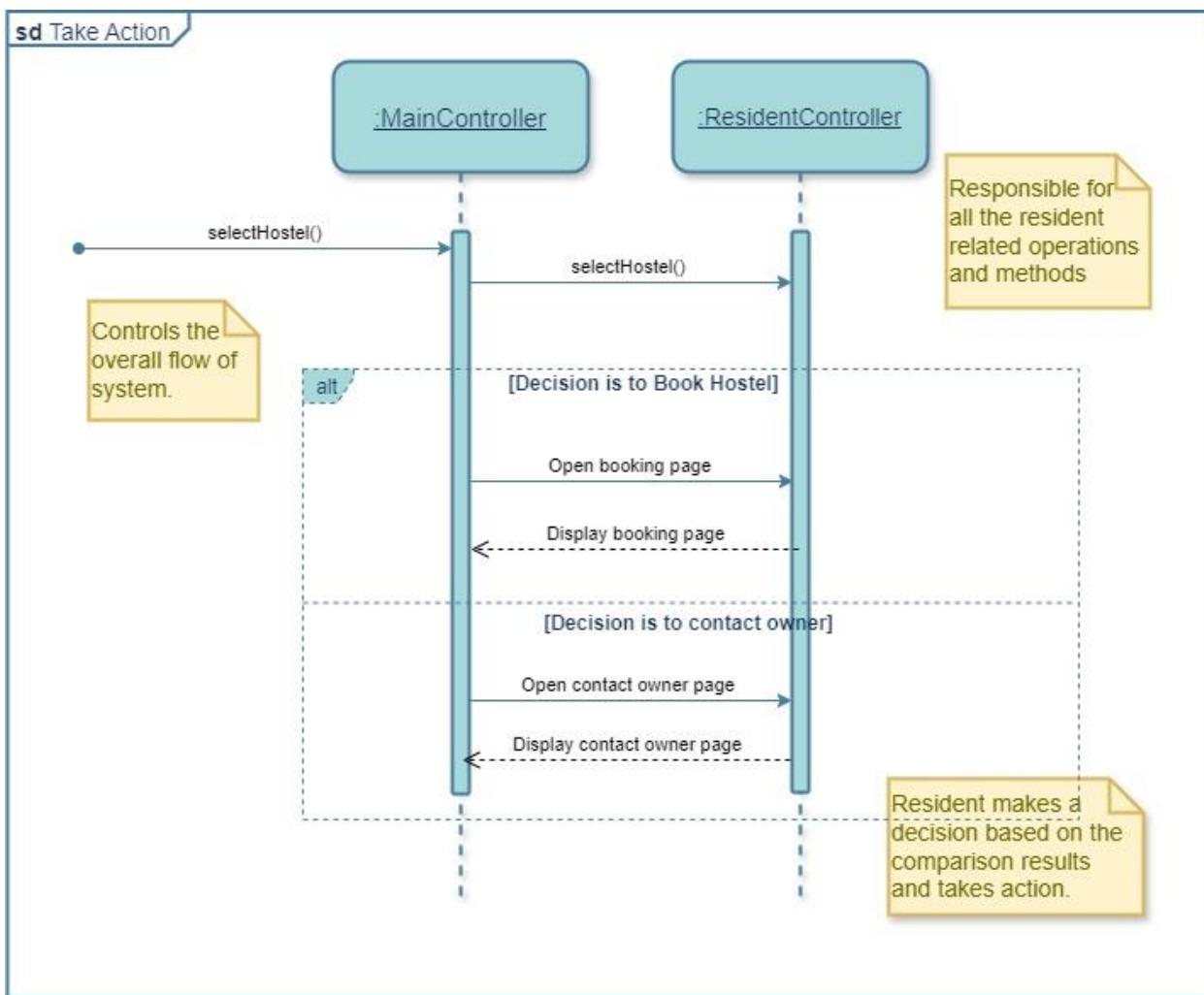
*c. Take Action*

Figure 75 SD- Take Action

## 7.3 Manage Hostel

### 7.3.1 System Sequence Diagram

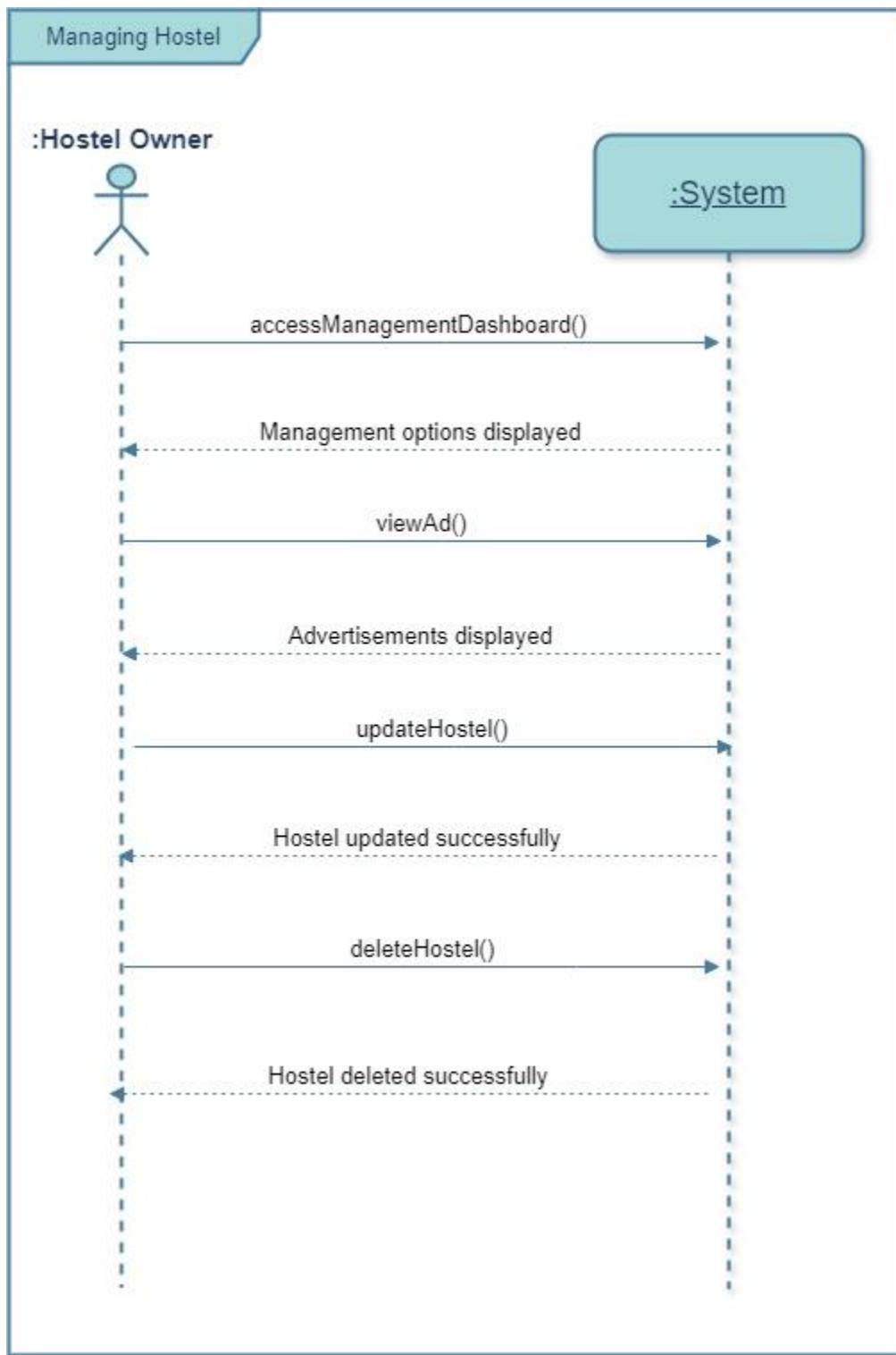


Figure 76 SSD- Manage Hostel

### 7.3.1 Sequence Diagrams

#### a. Open Hostel Manage Page

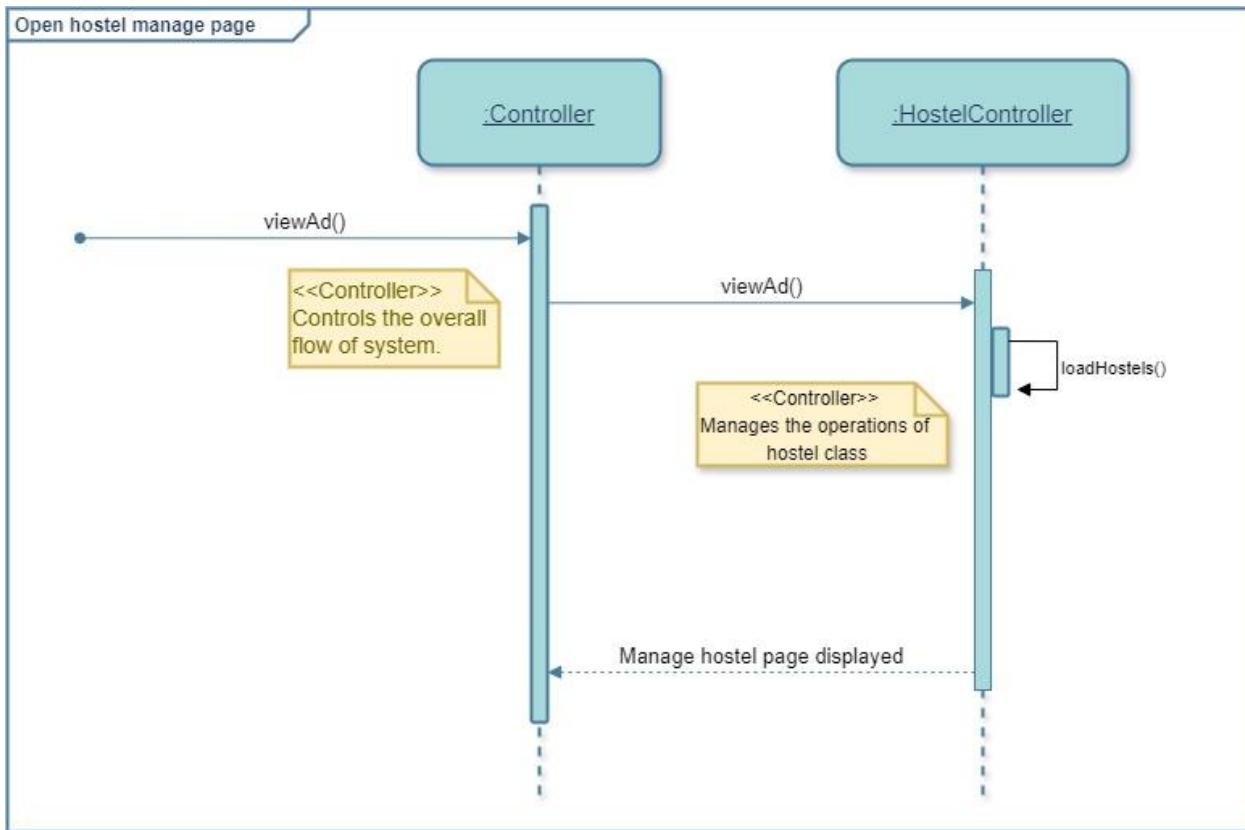


Figure 77 SD- Open Hostel Manage Page

### b. Update Hostel

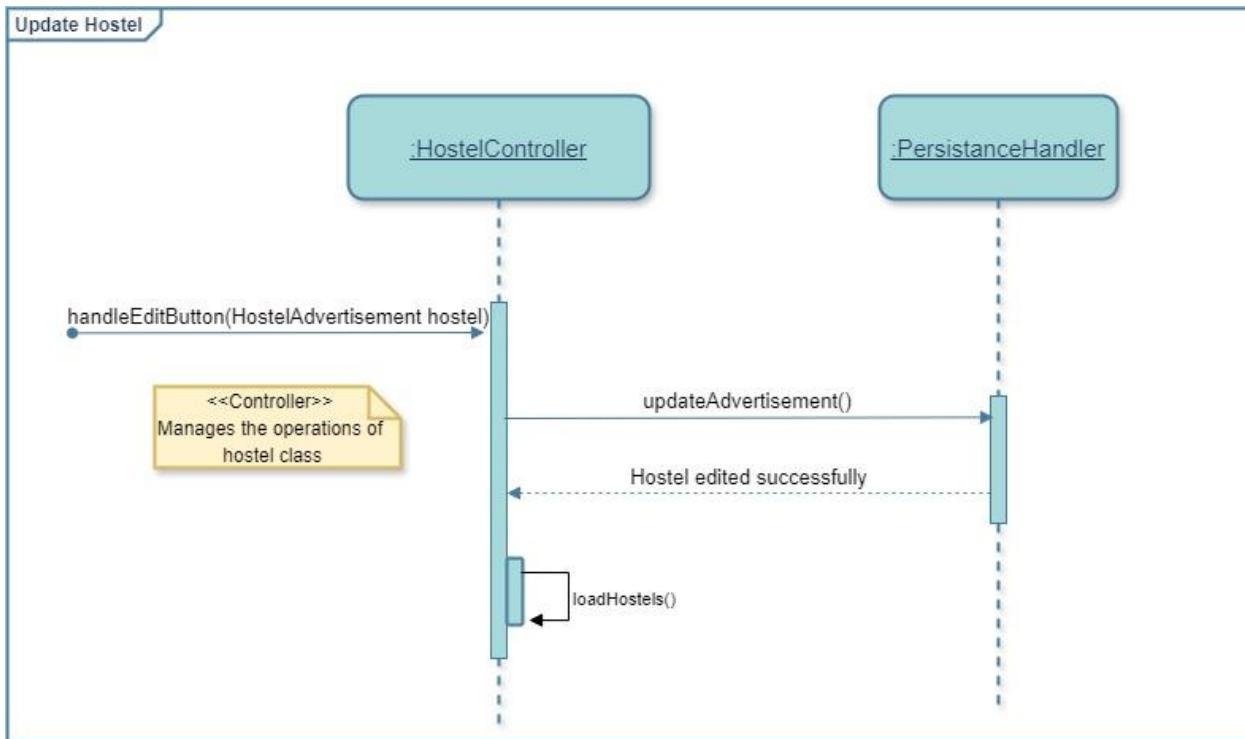


Figure 78 SD- Update Hostel

### c. Delete Hostel

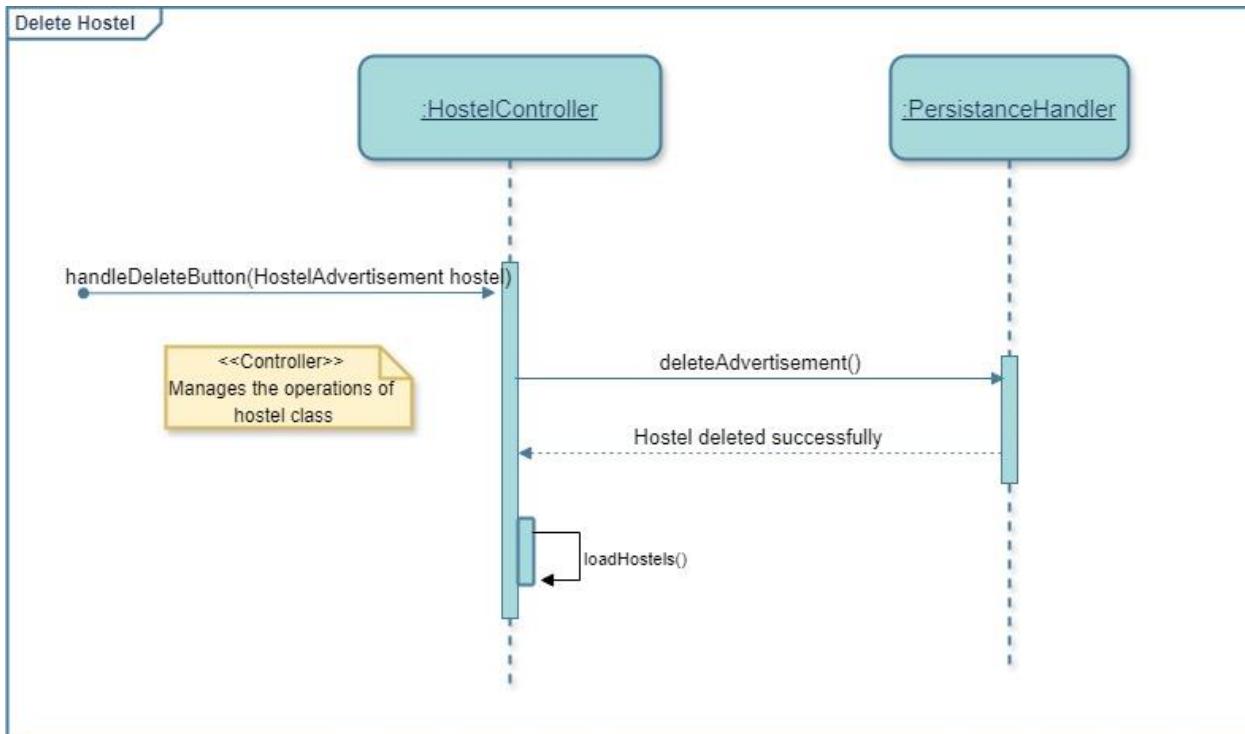


Figure 79 SD- Delete Hostel

## 7.4 Search Hostel

### 7.4.1 System Sequence Diagram

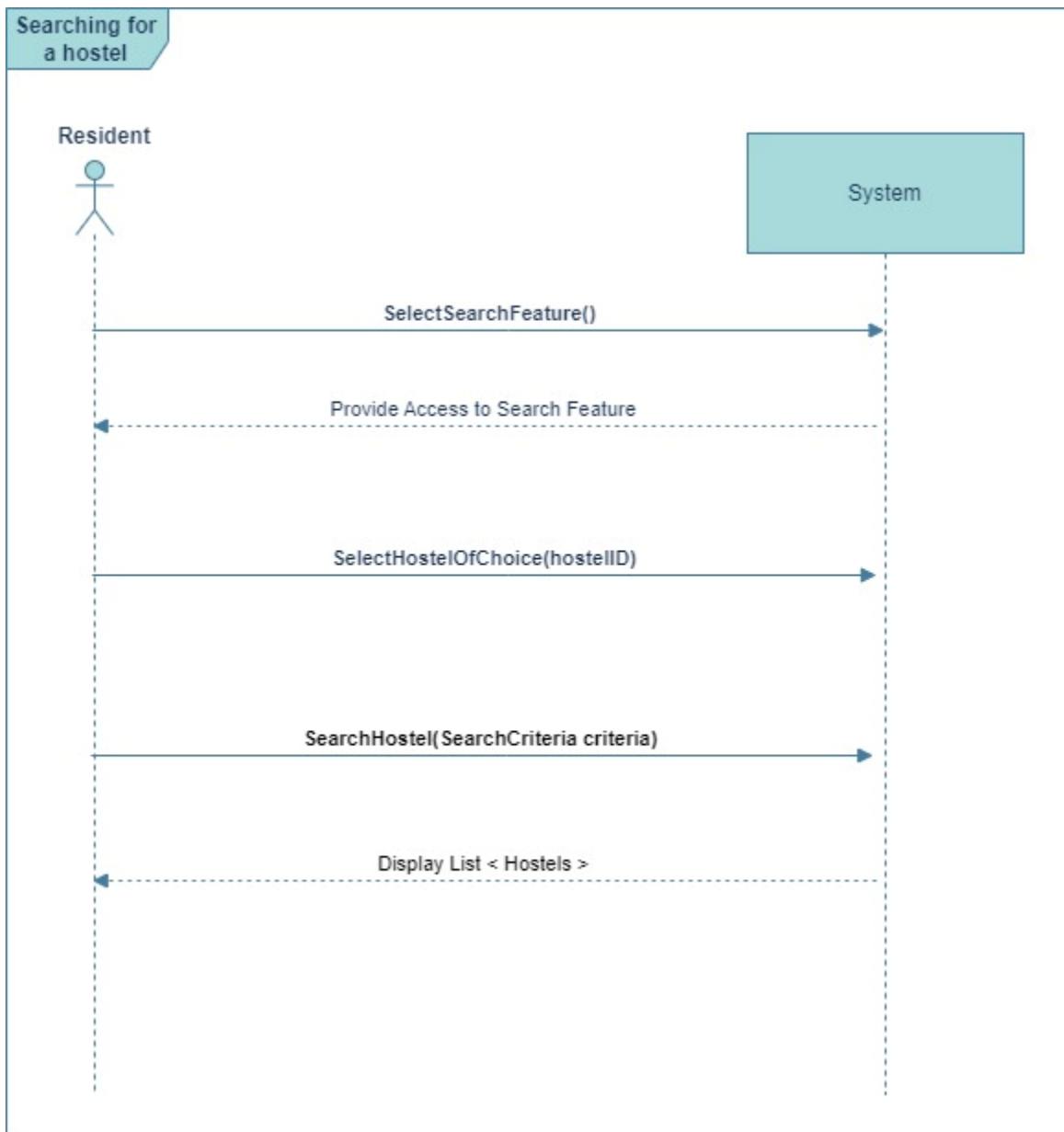


Figure 80 SSD- Search Hostel

### 7.4.2 Sequence Diagrams

#### a. Select Hostel

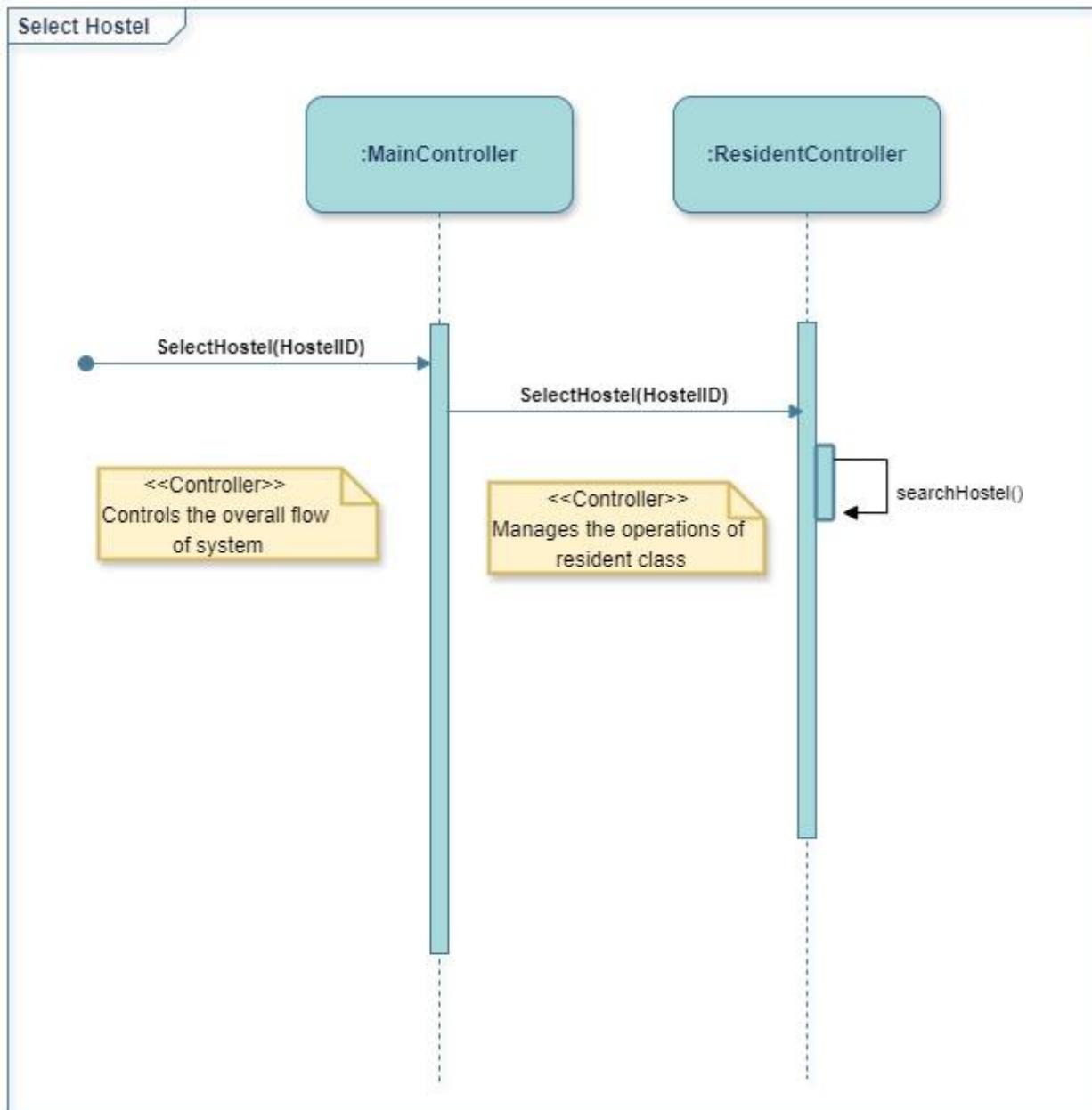


Figure 81 SD- Select Hostel

### b. Search Hostel

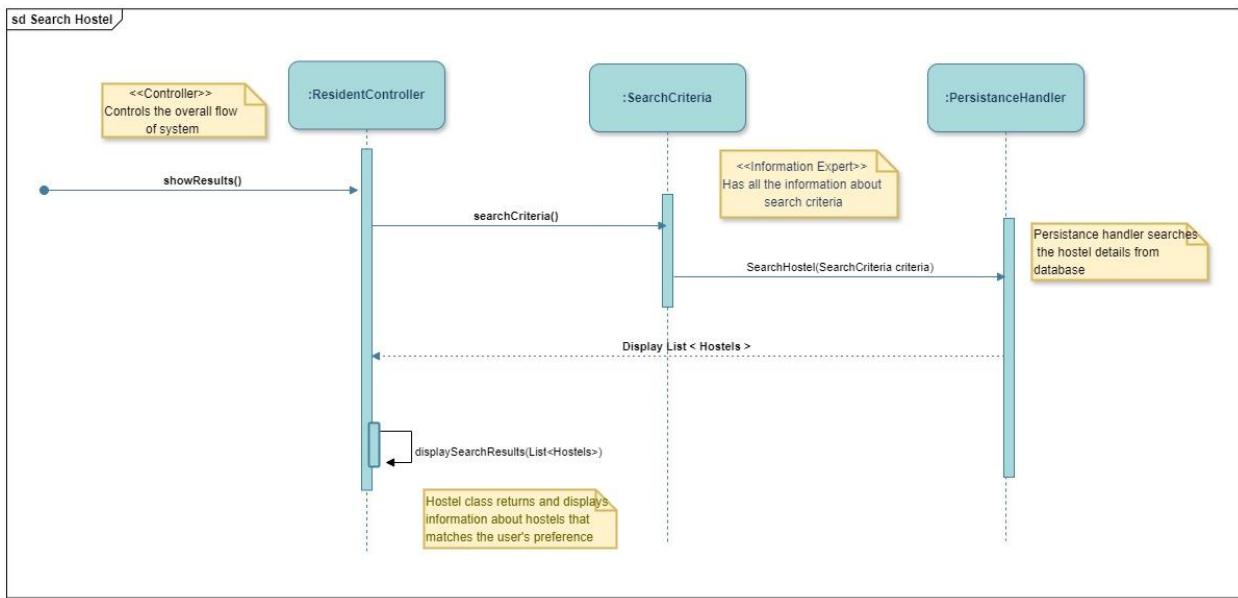


Figure 82 SD- Search Hostel

## 7.5 Manage Leave Request

### 7.5.1 System Sequence Diagram

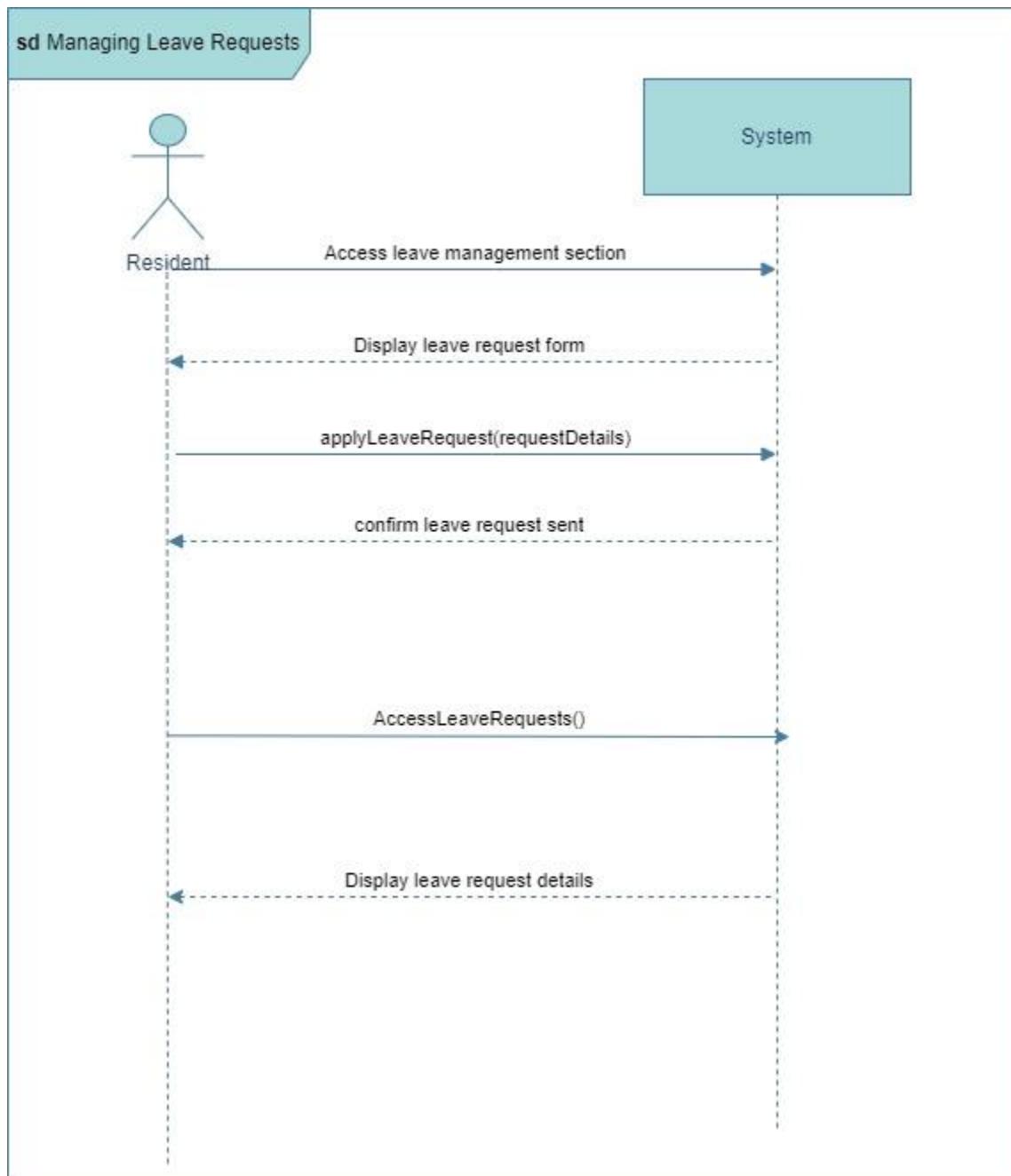


Figure 83 SSD- Manage Leave Request (updated)

## 7.5.2 Sequence Diagrams

### a. Access Leave

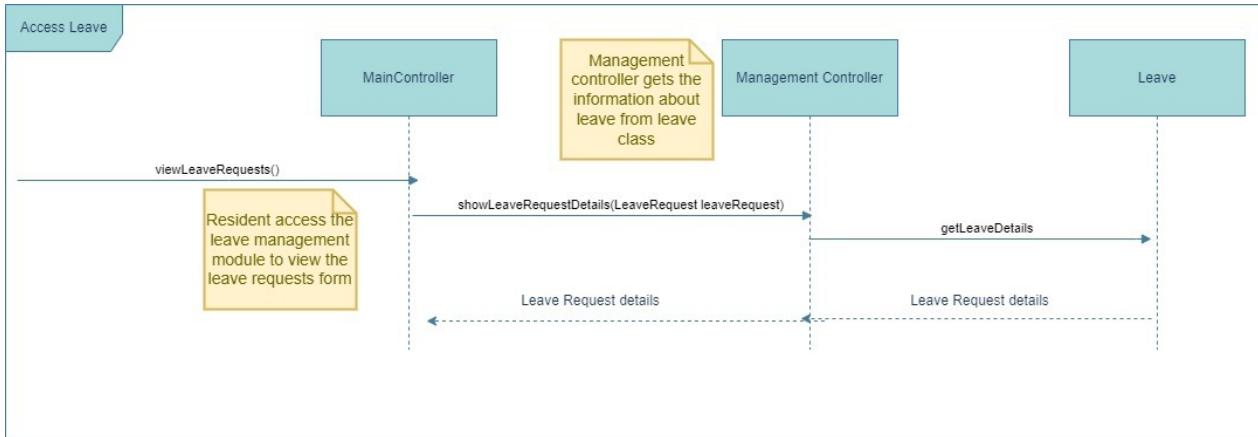


Figure 84 SD- Access Leave

### b. Apply Leave Request

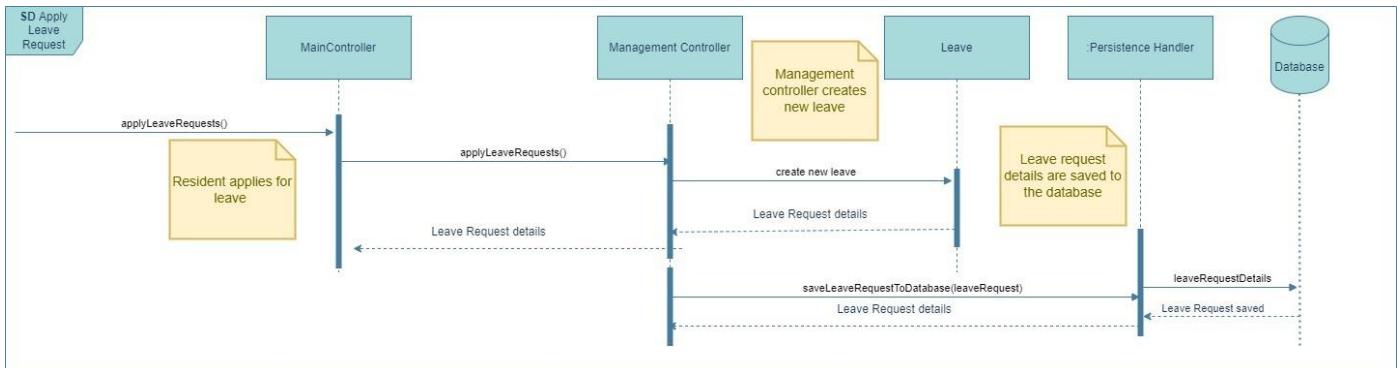


Figure 85 SD- Apply Leave Request

## 7.6 Allocate Room

### 7.6.1 System Sequence Diagram

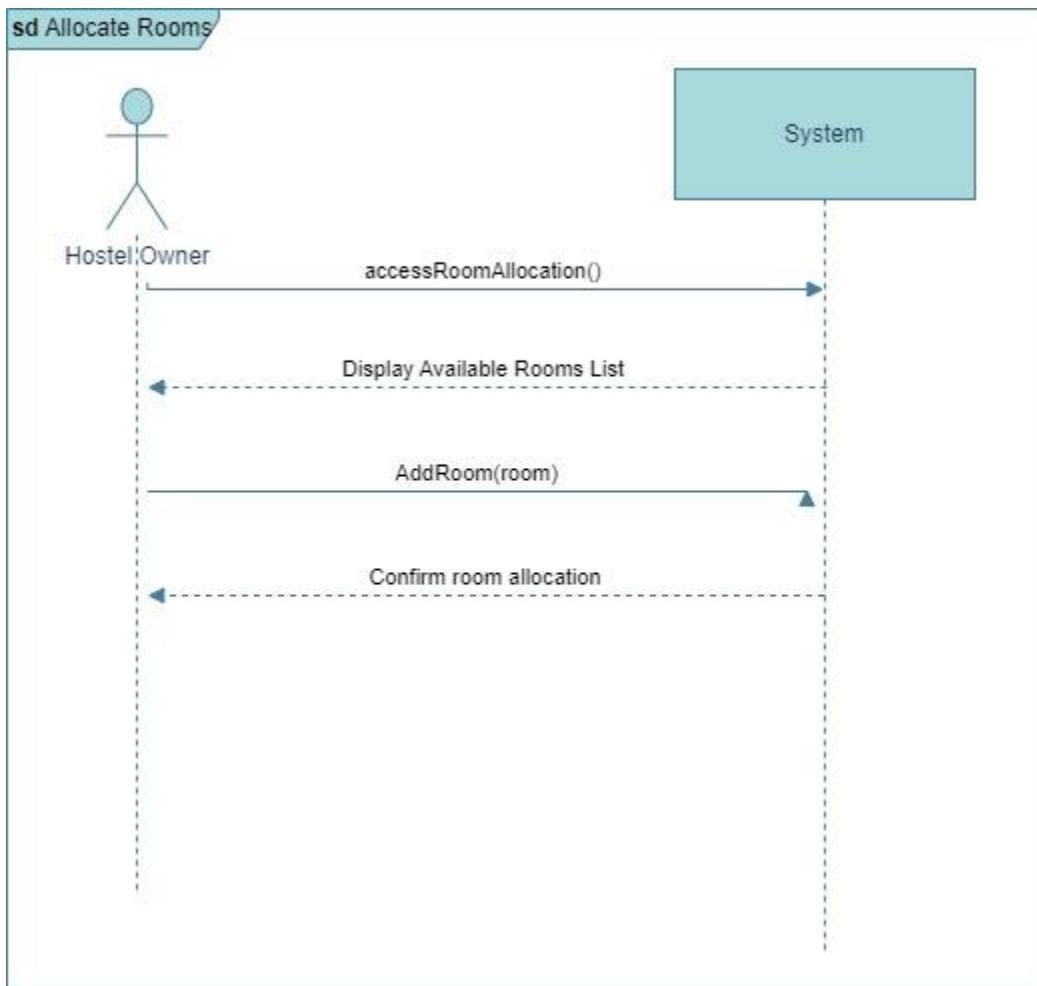


Figure 86 SSD- Allocate Room (updated)

## 7.6.2 Sequence Diagram

### a. Access Room Allocation

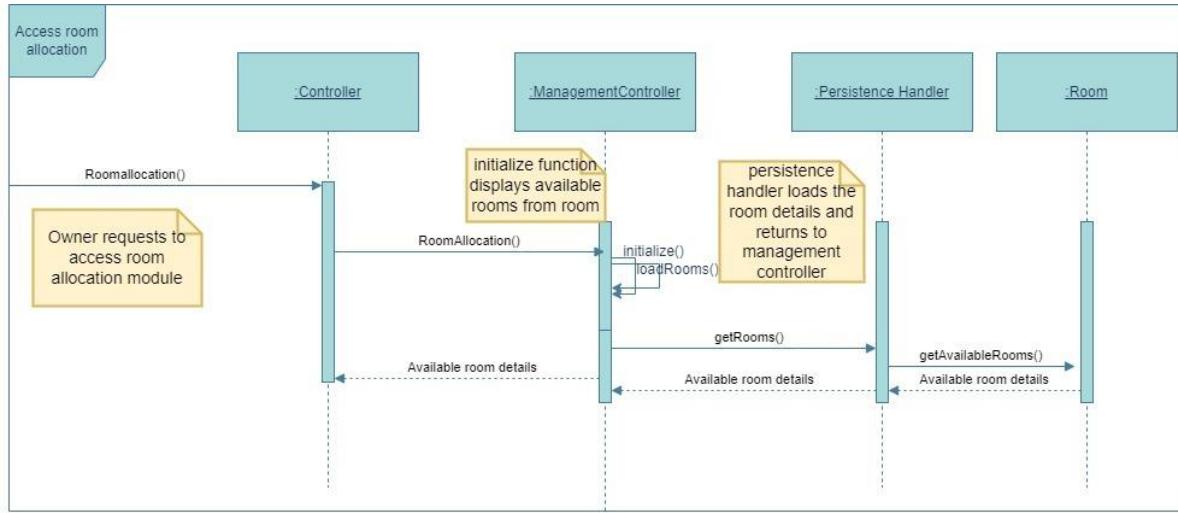


Figure 87 SD-Access Room Allocation

### b. Add Room

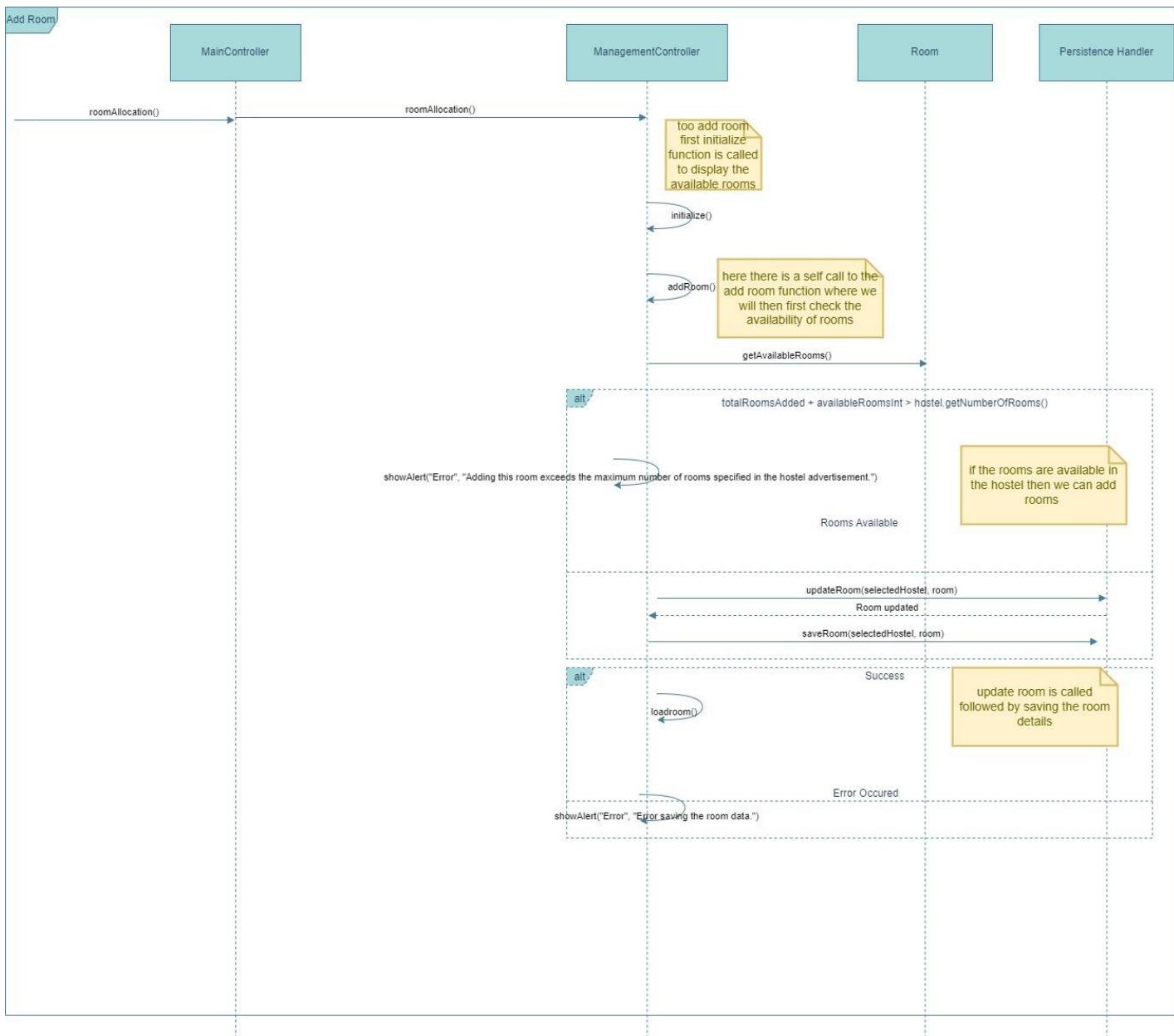
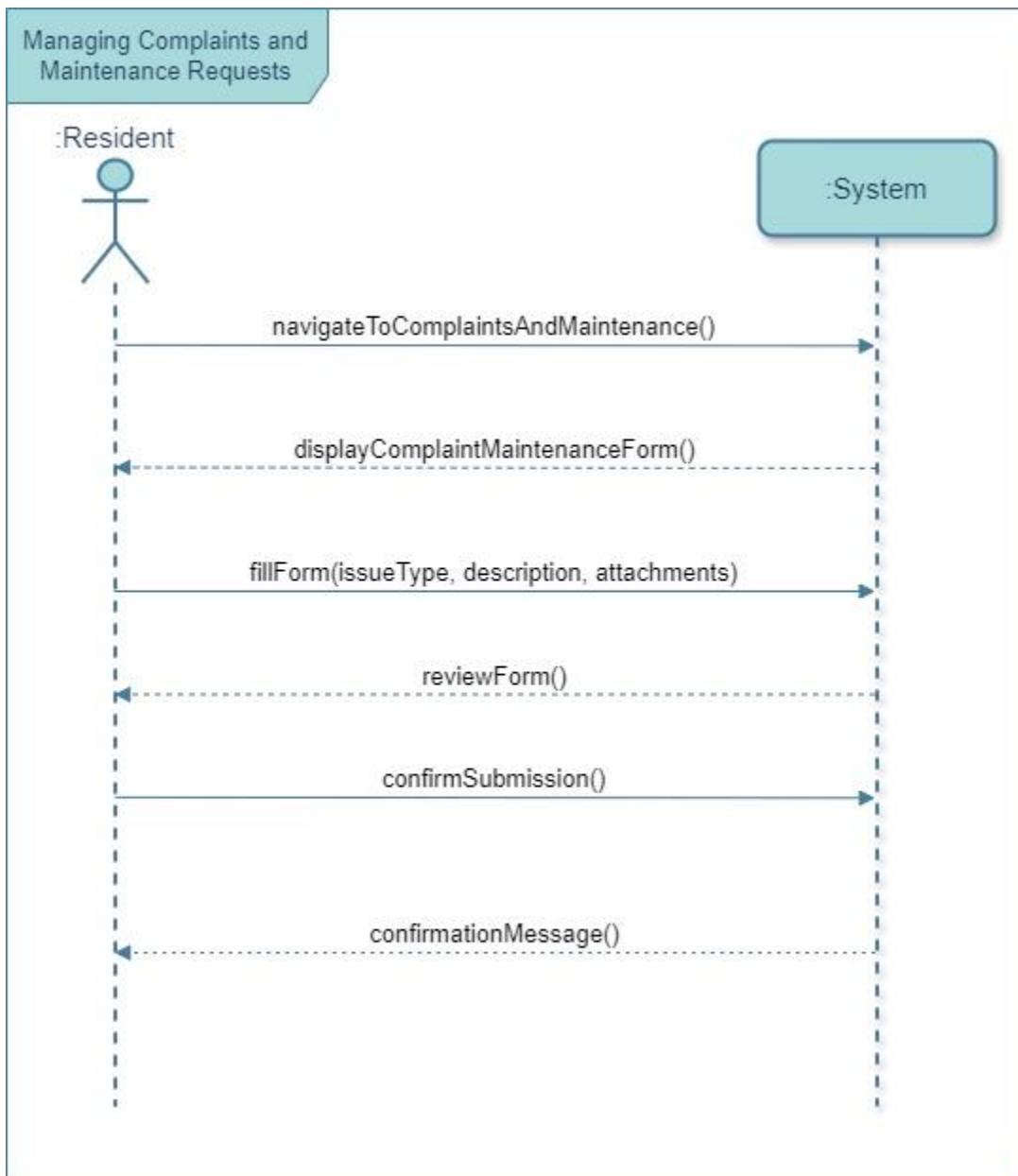


Figure 88 SD- Add Room

## 7.7. Complaint and Maintenance

### 7.7.1 System Sequence Diagram



## 7.7.2 Sequence Diagram

### a. View Complaint

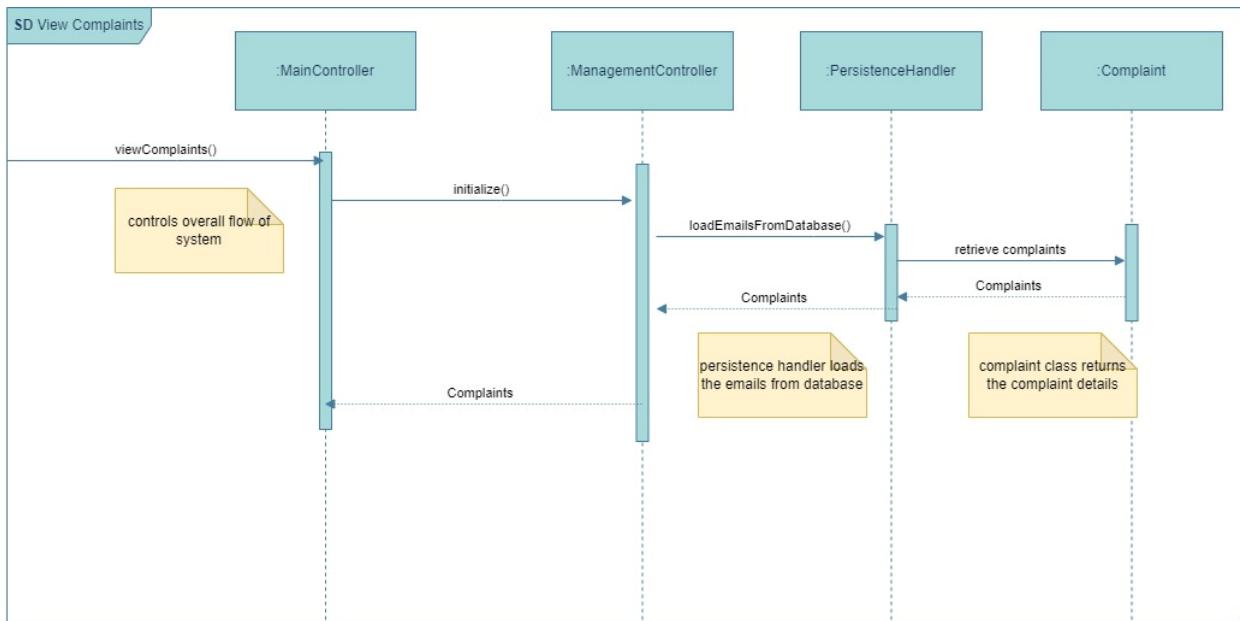


Figure 89 SD- View Complaint

### b. Register Complaint

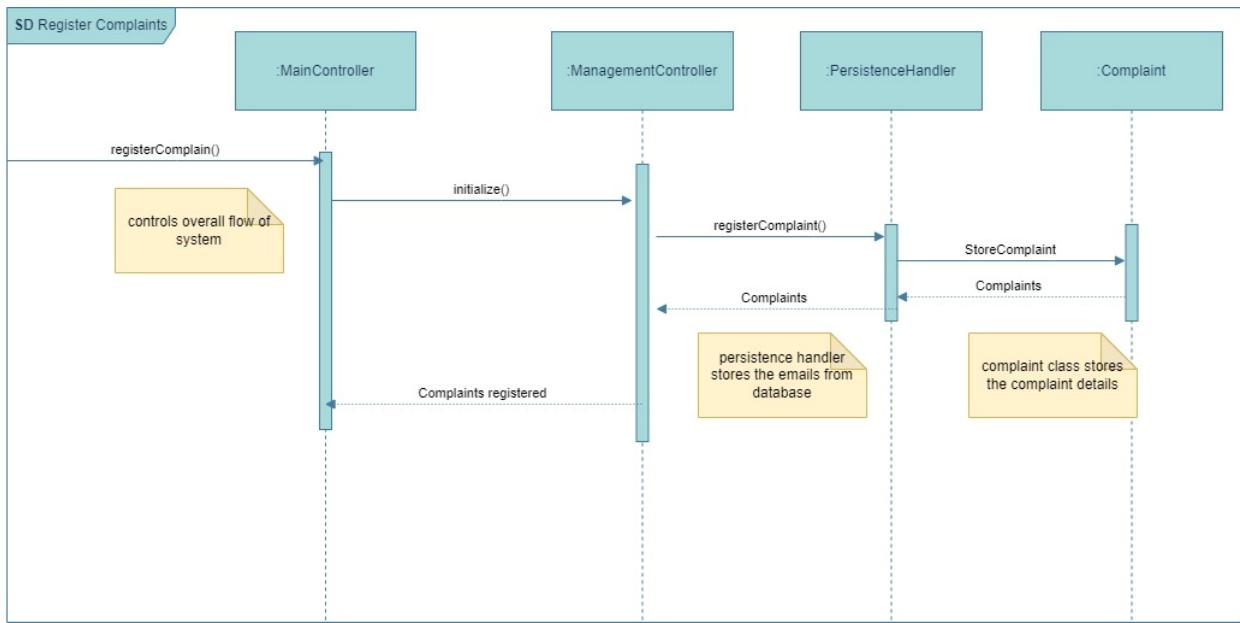


Figure 90 SD- Register Complaints

## 7.8 Booking A Hostel

### 7.8.1 System Sequence Diagram

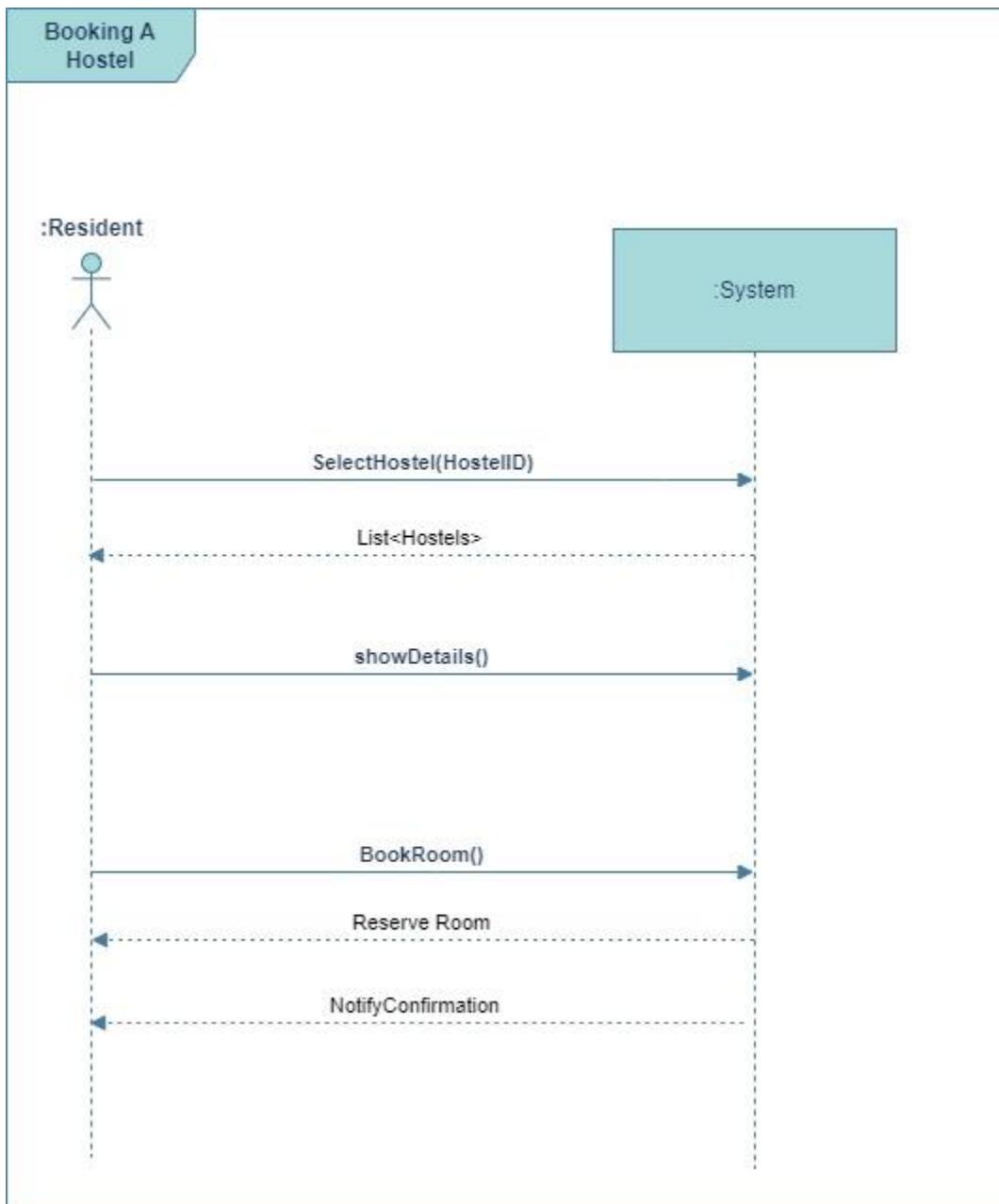


Figure 91 SSD- Booking A Hostel(Updated)

### 7.8.2 Sequence Diagrams

#### a. Select Hostel

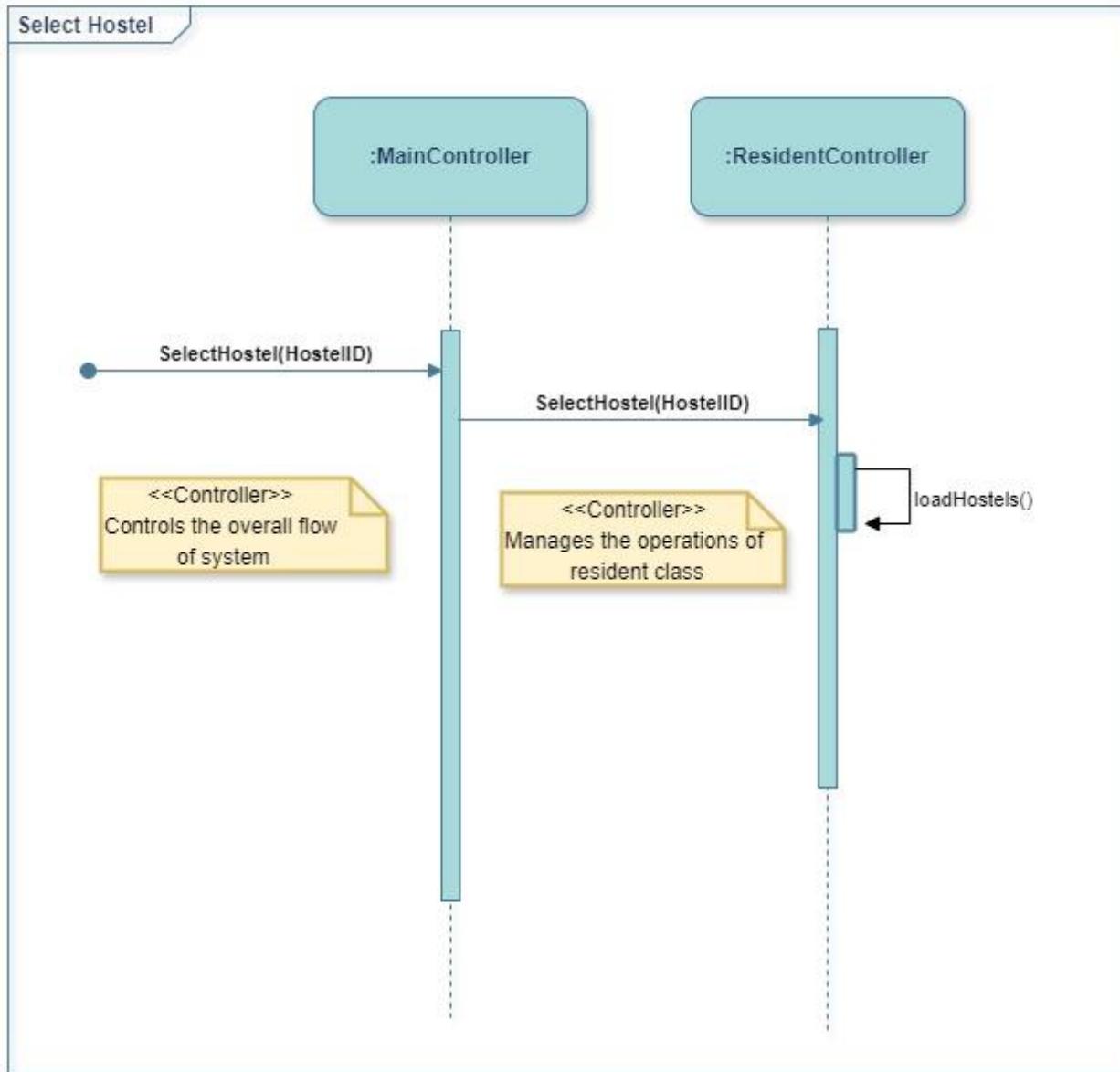


Figure 92 SD- Select Hostel

**b. Show Details**

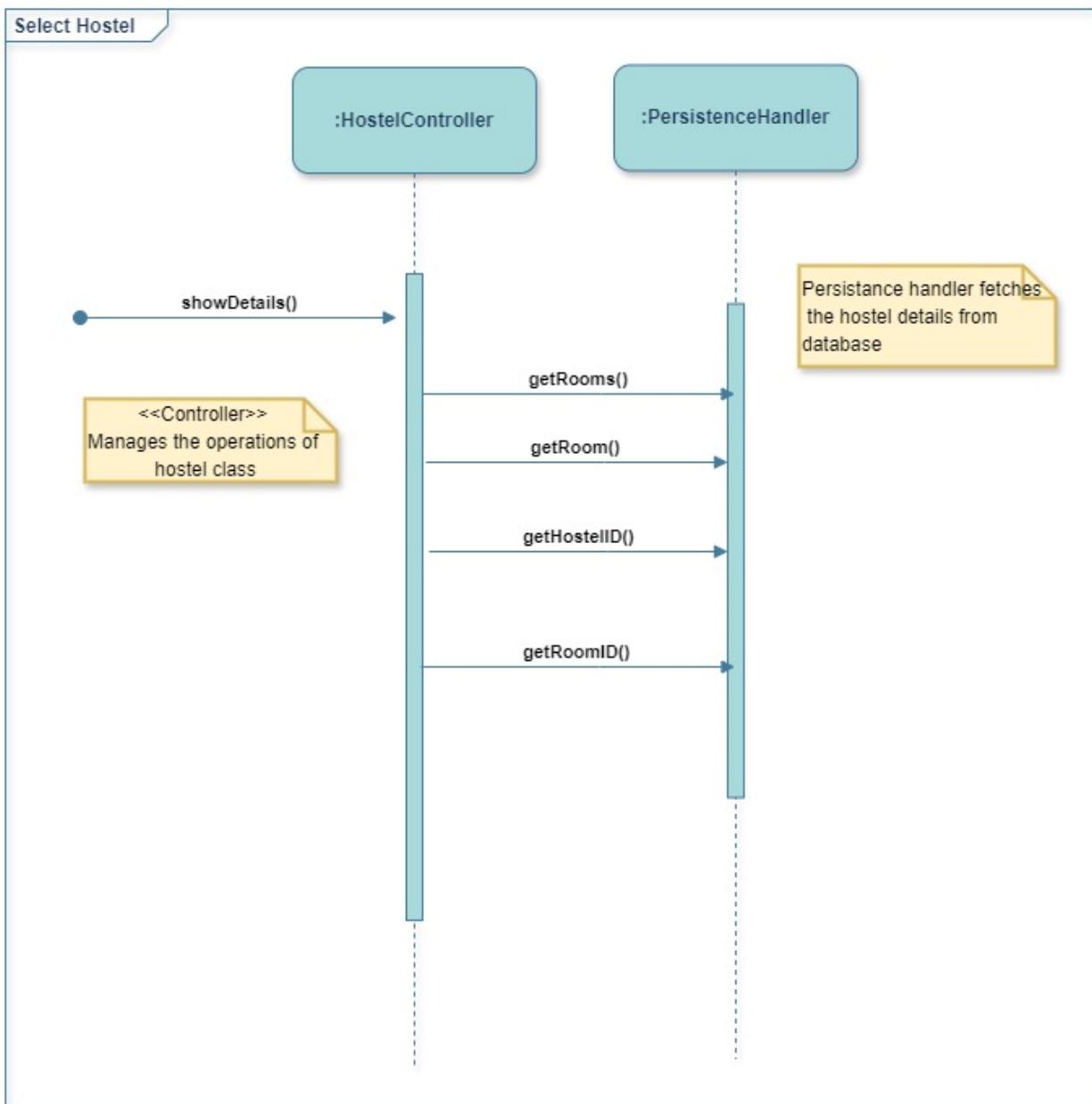


Figure 93 SD- Show Details

### c. Book Room

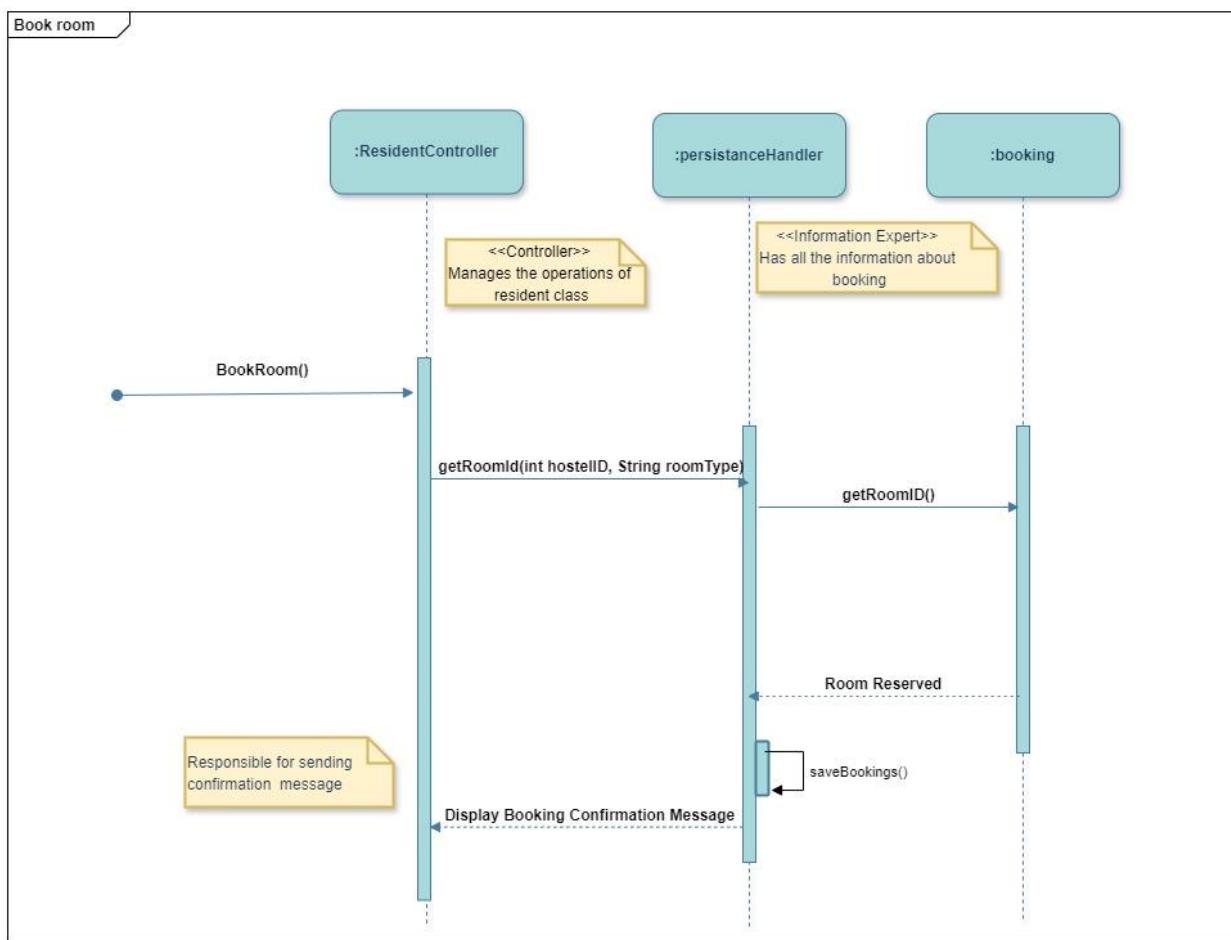


Figure 94 SD-Book Room

## 7.9 Submit Feedback and Reviews

### 7.9.1 System Sequence Diagram

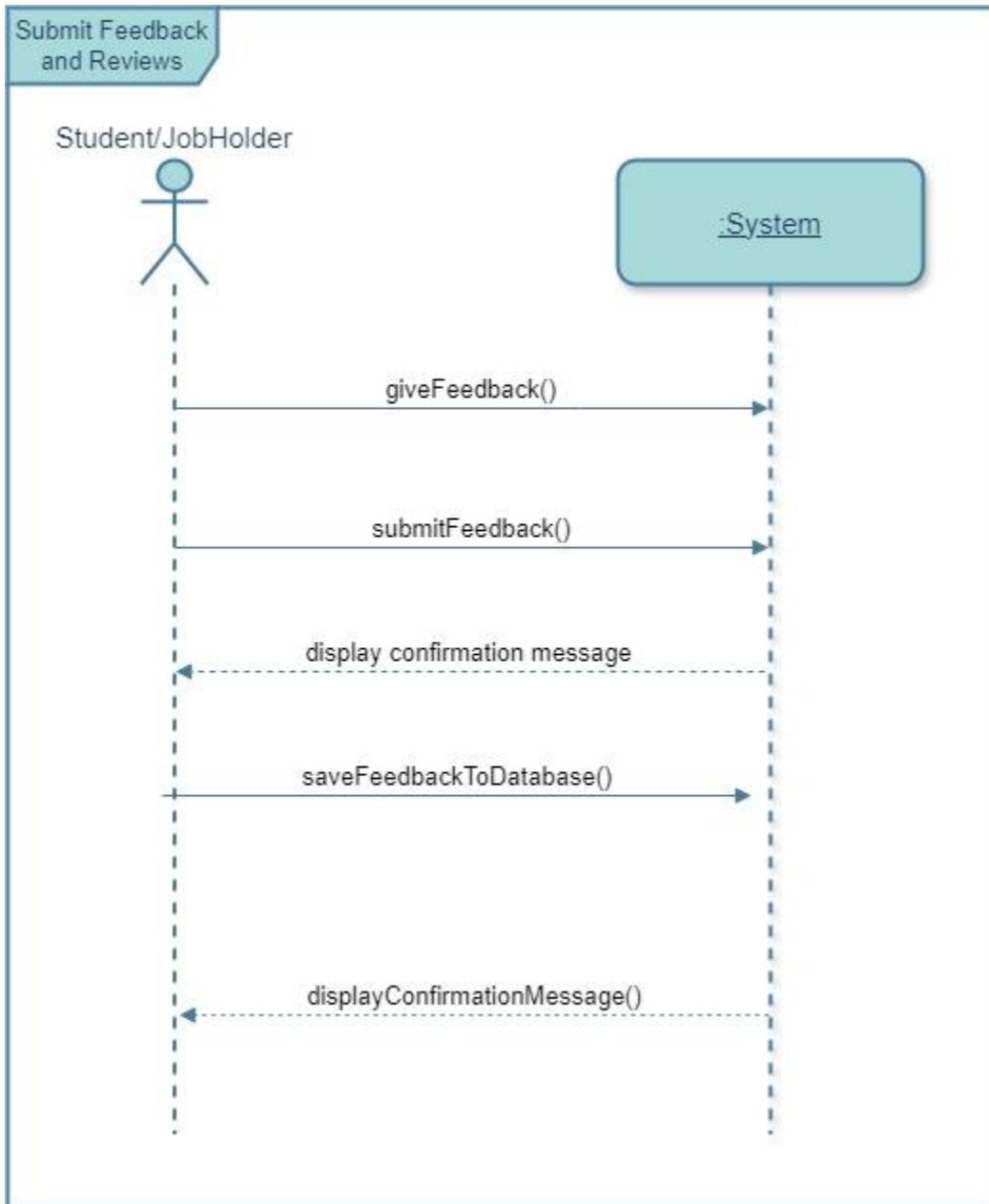


Figure 95 SSD- Submit Feedback And Reviews

## 7.9.2 Sequence Diagrams

### a. Give Feedback

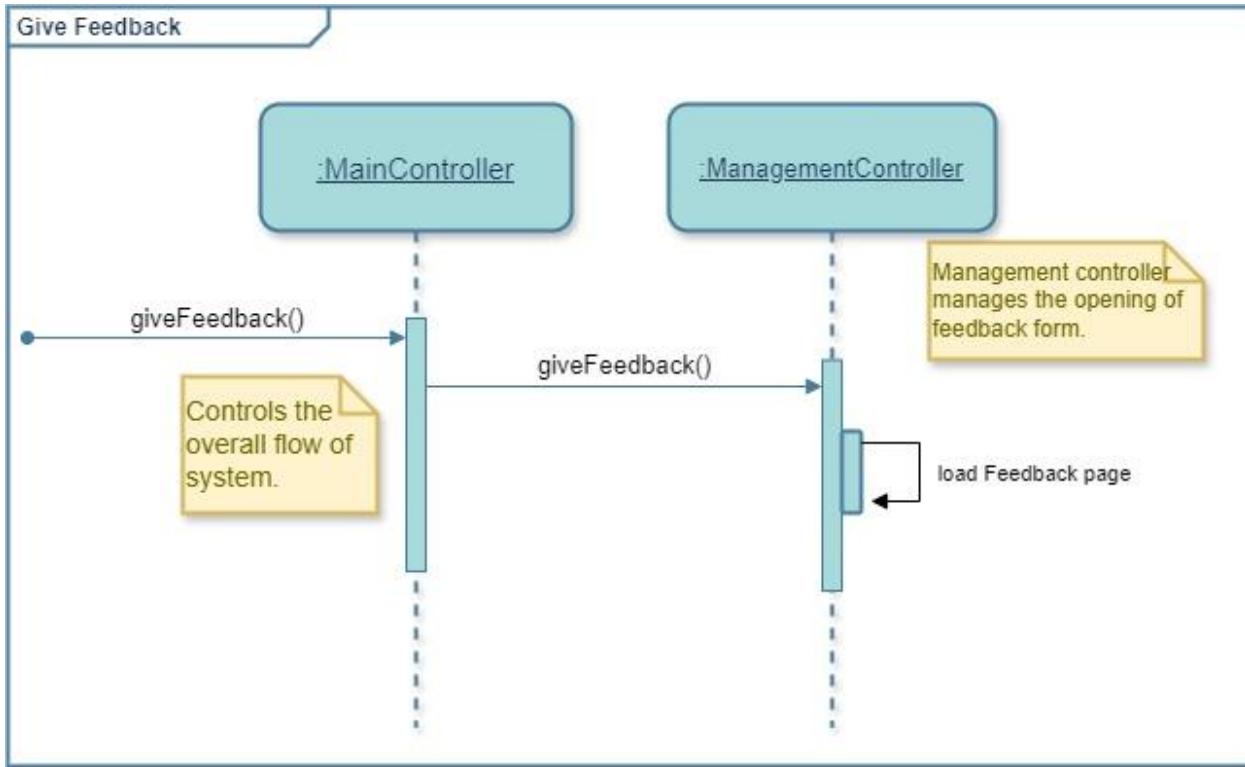


Figure 96 SD-Give Feedback

### b. Submit Form

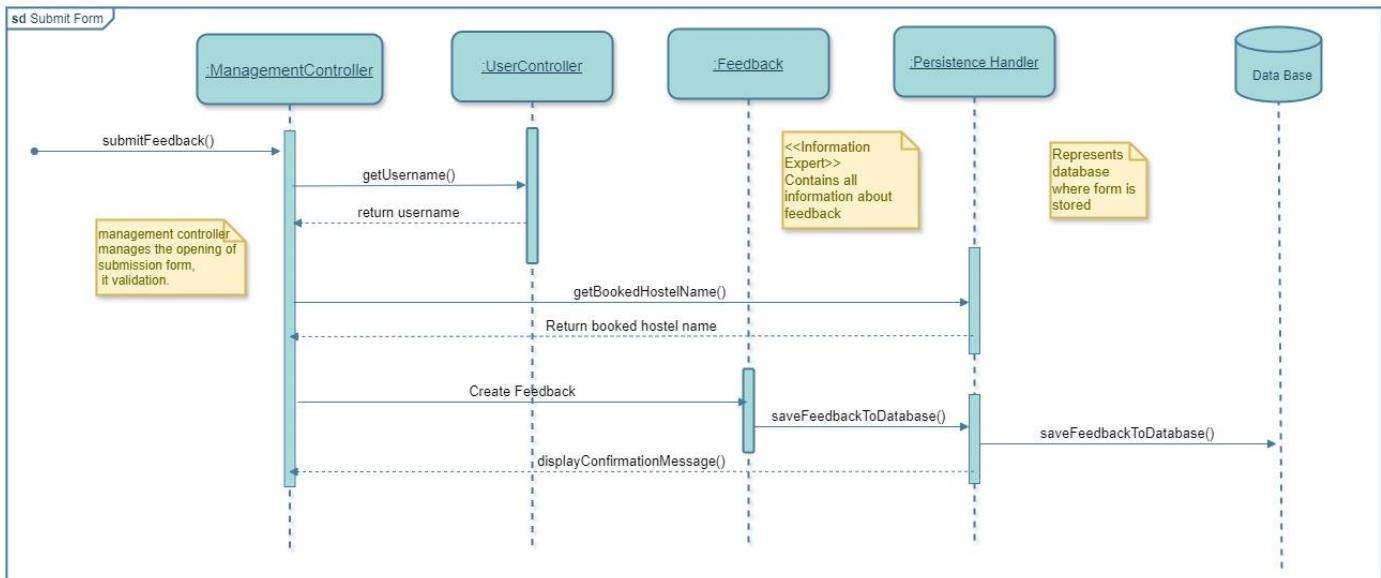
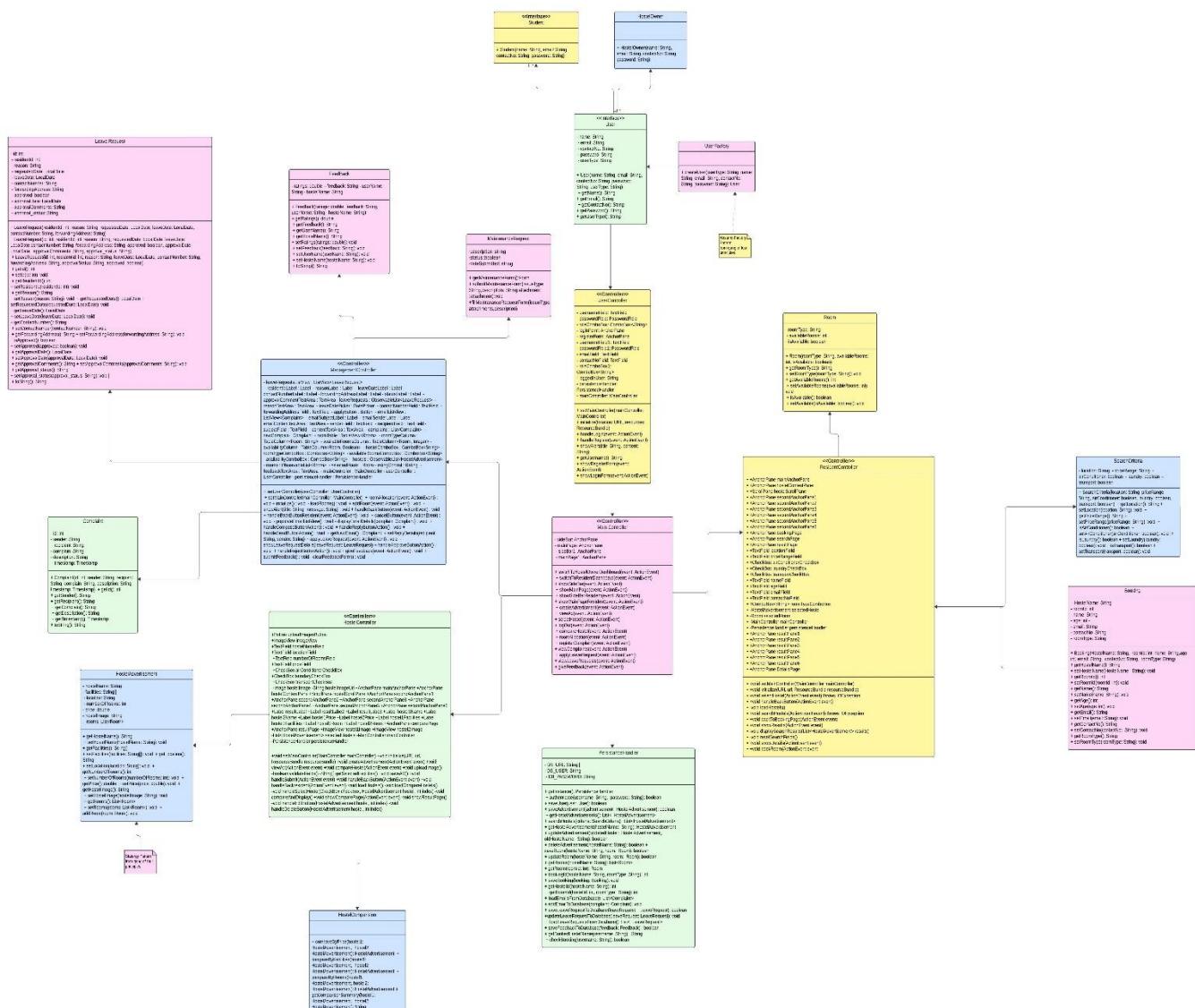


Figure 97 SD-Submit Form

## 8. CLASS DIAGRAM



## 9. HIGH-LEVEL ARCHITECTURE

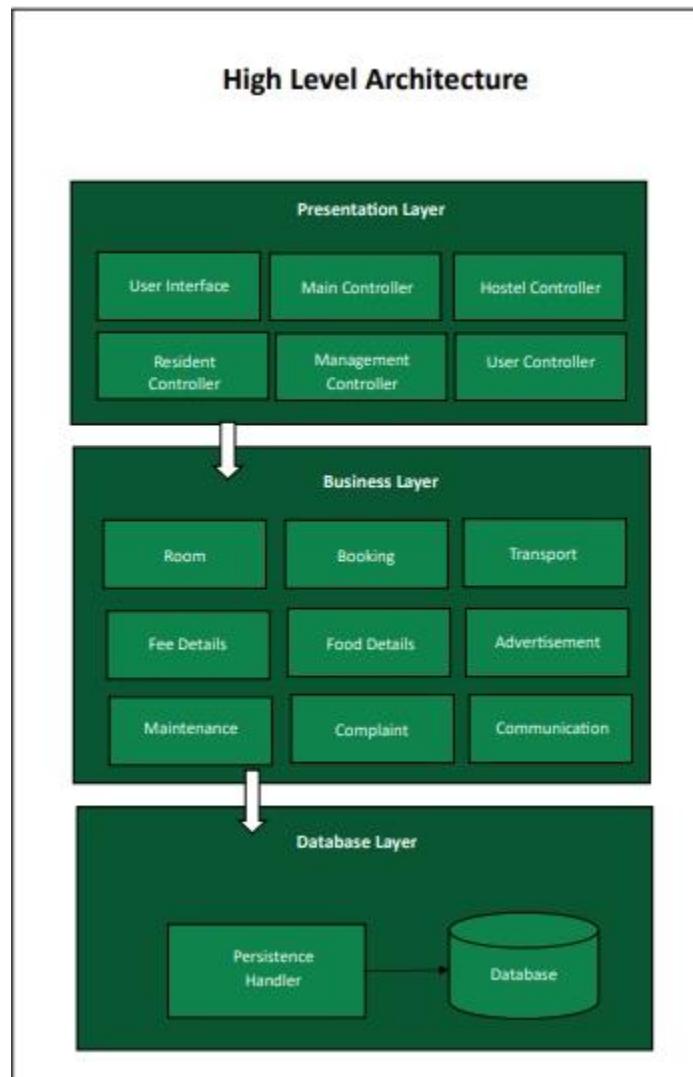


Figure 98 High Level Architecture

## 9.1 PACKAGE DIAGRAM

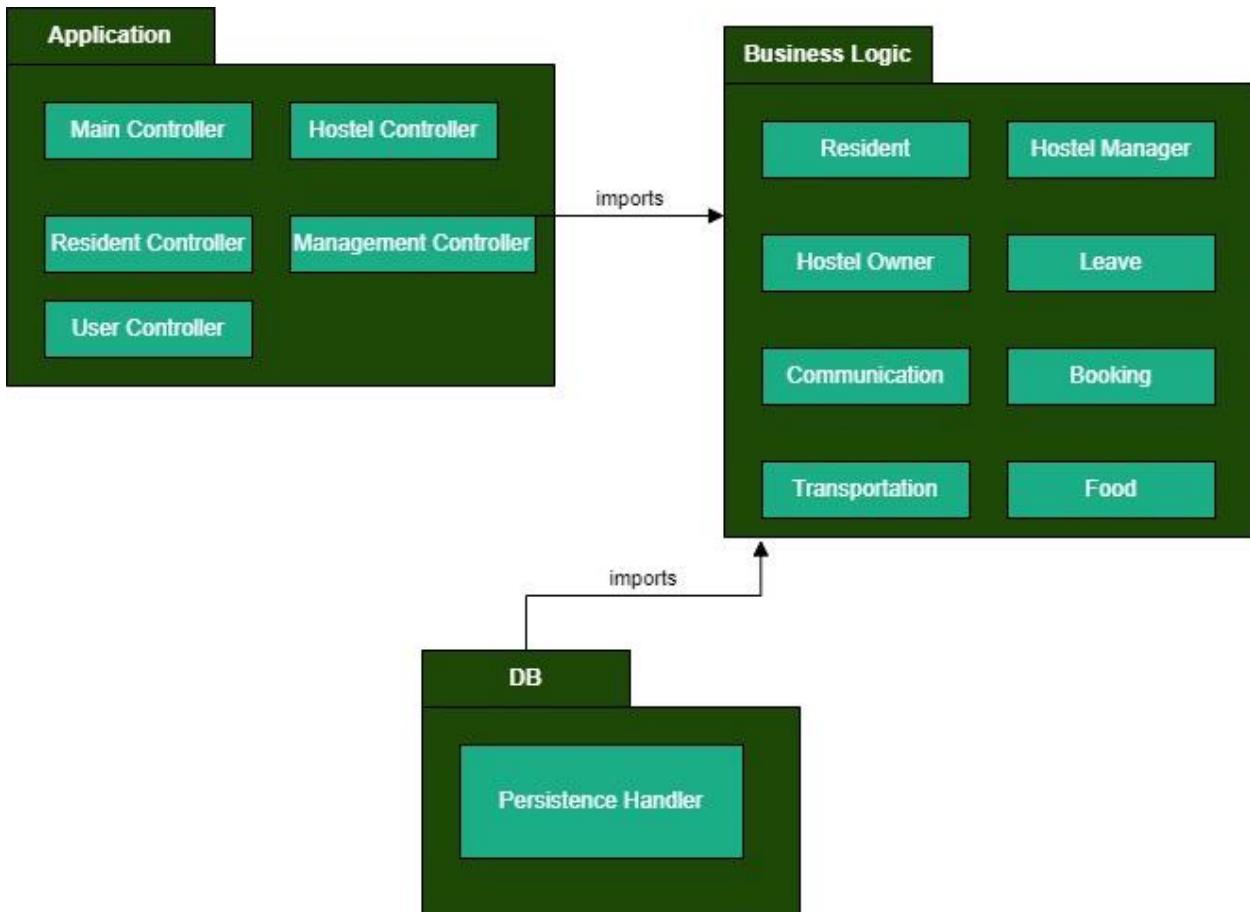


Figure 99 Package Diagram

## 9.2 DEPLOYMENT DIAGRAM

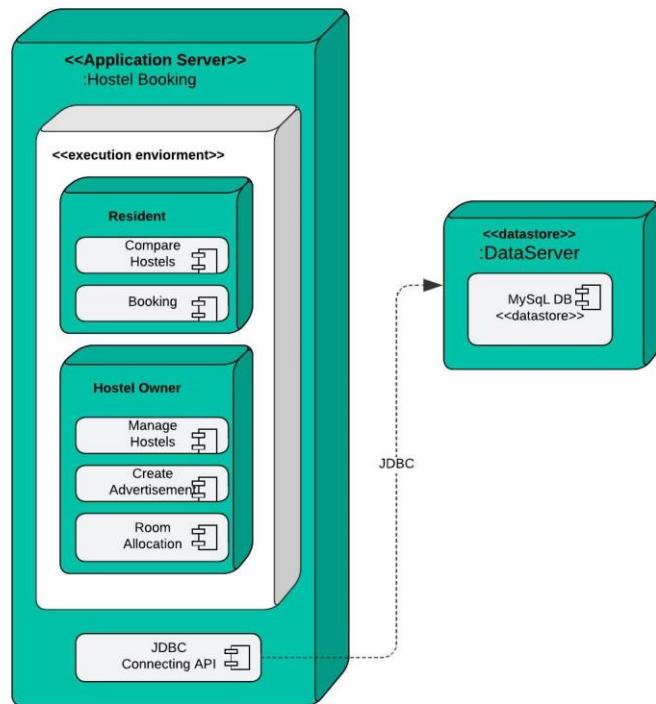


Figure 100 Deployment Diagram

## 9.3 COMPONENT DIAGRAM

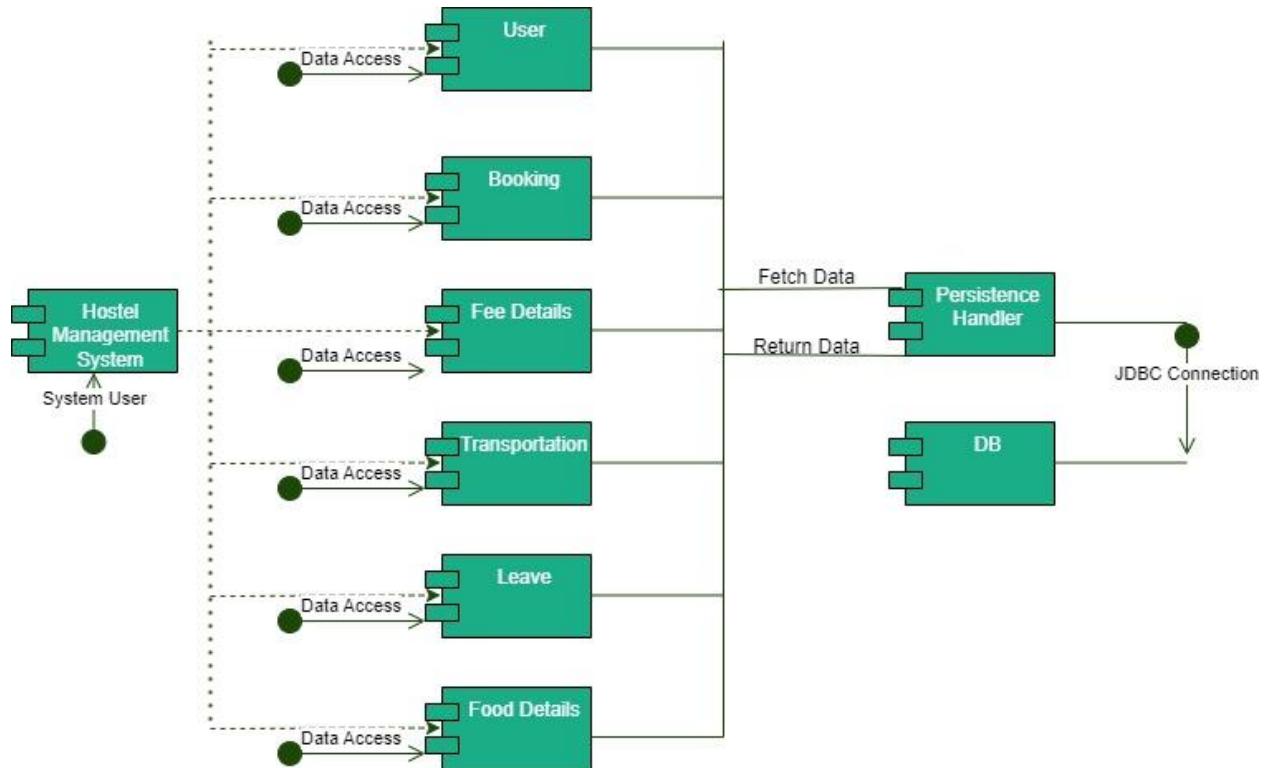


Figure 101 Component Diagram

## 10. WORK DIVISION TABLE

- Hassan
  - Implemented 3 functionalities:
    - Book Hostel
    - Compare hostel
    - Advertise hostel
  - Sequence diagrams of these functionalities
  - System sequence diagrams
  - Use case diagram
  - Domain model
  - Class diagram
  - Made Final report
  - Deployment diagram

- Hafsa
  - Implemented 3 functionalities:
    - a. Feedback
    - b. Manage hostel
    - c. Search hostel
  - Sequence diagrams of these functionalities
  - System sequence diagrams
  - Use case diagram
  - Domain model
  - Class diagram
  - Made Final report
  - High level architecture diagram
- Shajia
  - Implemented 3 functionalities:
    - a. Room allocation
    - b. Leave Request
    - c. Manage complaints
  - Sequence diagrams of these functionalities
  - System sequence diagrams
  - Use case diagram
  - Domain model
  - Class diagram
  - Made Final report
  - Package diagram
  - Component diagram