



TRƯỜNG ĐẠI HỌC FPT

MINISTRY OF EDUCATION AND TRAINING

FPT UNIVERSITY

Capstone Project Document

FU House Finder

SEP490_G1	
Group Members	Nguyễn Thế Giang HE153046 Bùi Ngọc Huyền HE150346 Nguyễn Thu An HE150432 Phùng Quang Thông HE150340 Nguyễn Trí Kiên HE150160
Supervisor	Nguyễn Tất Trung
Ext Supervisor	
Capstone Project code	FHF

- Hanoi, December 2022 -

Table of Contents

Acknowledgement	7
Definition and Acronyms	8
I. Project Introduction	9
1. Overview	9
1.1 Project Information.....	9
1.2 Project Team	9
2. Product Background.....	9
3. Existing Systems	10
3.1 FPT Can Tho.....	10
3.2 Nhatot System.....	10
3.3 BatDongSan System	10
3.4 Blog	11
4. Business Opportunity.....	11
5. Software Product Vision	11
6. Project Scope & Limitations	12
6.1 Major Features.....	12
6.2 Limitations & Exclusions	13
II. Project Management Plan	14
1. Overview	14
1.1 Scope & Estimation	14
1.2 Project Objectives	18
1.3 Project Risks	19
2. Management Approach	20
2.1 Project Process.....	20
2.2 Quality Management	21
2.3 Training Plan.....	21
3. Project Deliverables	23
4. Responsibility Assignments.....	24
5. Project Communications.....	24
6. Configuration Management.....	26
6.1 Document Management	26
6.2 Source Code Management.....	26
6.3 Tools & Infrastructures	26
7. Team Structure	27

III. Software Requirement Specification	28
1. Product Overview	28
2. User Requirements	28
2.1 Actors	28
2.2 Use Cases	30
3. Functional Requirements.....	33
3.1 System Functional Overview.....	33
3.2 Home page	41
3.3 Login/Signup	42
3.4 House Detail.....	43
3.5 Room Detail.....	43
3.6. Send Report [Student]	44
3.7 Register [Landlord].....	45
3.8 Dashboard [Landlord]	47
3.9 Upload House Information [Landlord]	47
3.10 House Detail [Landlord]	48
3.11 Update House [Landlord]	49
3.12 Delete House [Landlord]	49
3.13 Update Room [Landlord]	50
3.14 Add Room [Landlord]	50
3.15 Delete Room [Landlord].....	51
3.16 Dashboard [Staff]	52
3.17 List Of Landlords [Staff].....	52
3.18 Landlord Detail [Staff]	53
3.19 House Detail [Staff]	54
3.20 Room Detail [Staff].....	55
3.21 List Of Houses [Staff].....	56
3.22 List Of Reports [Staff]	57
3.21 List Of Landlords Signup Request [Staff].....	59
3.22 List Of Orders [Staff]	59
3.22 List of Accounts [Admin]	60
3.23. Add Account [Admin].....	60
3.24. Update Account [Admin]	61
3.25. Delete Account [Admin].....	61
3.26. Create Order [Student]	62
4. Non-Functional Requirements.....	63

4.1 External Interfaces	63
4.2 Quality Attributes.....	63
5. Requirement Appendix	64
5.1 Business Rules	64
5.2 Common Requirements	Error! Bookmark not defined.
5.3 Application Messages List	65
IV. Software Design Description	66
1. System Design	66
1.1 System Architecture.....	66
1.2 System Architecture Explanation.....	67
1.2 Package Diagram.....	69
2. Database Design.....	71
3. Database Tables	73
3.1. Campuses	73
3.2. Districts	73
3.3. Communes	73
3.4. Villages	73
3.5. Addresses	74
3.6. UserRoles	74
3.7. UserStatuses	74
3.8. Users	74
3.9. Houses.....	75
3.10. ImagesOfHouse	75
3.11. Issues.....	76
3.12. Rates.....	76
3.13. RoomStatuses	76
3.14. RoomTypes.....	76
3.15. Rooms	77
3.16. ImagesOfRoom.....	77
3.17. RoomHistories.....	78
3.18. ReportStatuses.....	78
3.19. Reports.....	78
3.20. OrderStatuses	78
3.21. Orders	79
4. Detailed Design	80
4.1 Home Page	80

4.2 House Detail.....	84
4.3 Room Detail.....	89
4.4. Landlord – Create House.....	93
4.5. Landlord – List Room.....	95
4.6 Landlord – Create Room	98
4.7. Landlord – Update Room	102
4.8 Landlord – Delete Room	106
4.9 Login	109
4.10 Landlord – Upload House Information	111
4.11 Staff – Dashboard.....	114
4.12 Staff – List Landlord.....	119
4.13. Staff - Landlord Detail	121
4.14. Staff - House Detail	125
4.15. Staff - Room Detail.....	129
4.16 Staff – List House.....	132
4.17 Staff – List Report.....	134
4.18. Staff – List Order	137
4.19. Staff – Change Password.....	139
4.20. Landlord – Update House	141
4.21. Landlord – Delete House.....	143
4.22 Admin – List Staff Account	145
4.23 Admin – Add Staff Account	147
4.24 Admin – Edit Staff Account	149
4.25 Admin – Delete Staff Account.....	151
V. Software Testing Documentation.....	153
1. Purpose	153
1.1. Scope of Testing	153
2. Test Plan.....	154
2.1. Testing tools and environment	154
2.2. Human Resources and Responsibilities	156
2.3. Test Strategy	157
2.4. Features to be tested	159
3. Test Case	160
3.1. Automation Testing with Unit Testing	160
3.2. API Testing.....	163
3.3. Integration Testing.....	165

3.4. System Testing	166
3.5. Acceptance Testing	166
3.6. Defect Log	167
4. Test Report.....	168
4.1. Automation test case report.....	168
4.2. Automation test report.....	169
4.3. Integration test case report	170
4.4. Integration test report	171
VI. Release Package & User Guides.....	172
1. Deliverable Package	172
2. Installation Guides	173
2.1 Environment for Development	173
2.2 Environment for Deployment	175
3. User Manual.....	177
3.1 Overview	177
3.2 Web application.....	177

Acknowledgement

We would like to sent a big thank you and express sincere gratitude towards our capstone project supervisor **Mr. Nguyen Tat Trung** for his wholehearted support throughout the past four months. We fully appreciate the time Mr.Trung spent sharing with us the philosophy of keeping the right attitude towards problems. We believe that was the most important factor that led to the success of this project.

It is also our pleasure to receive assistance from the staffs of **The University Admission Office** who provides us with valuable up-to-date information concerning fundamental requirements, admission rate and contact information of landlords in the area. We cannot obtain accurate data without their help.

Last but not least, we would like to gratefully thank all of our **Teachers** for being there to guide us, inspiring us with a love for learning. We hope you will find this project as a reflection of the knowledge and experiences you have given us during this period of three years.

Definition and Acronyms

Acronym	Definition
FHF	FU House Finder
AWS	Amazon Web Services
BA	Business Analysis
BR	Business Rule
ERD	Entity Relationship Diagram
GUI	Graphical User Interface
PM	Project Manager
SDD	Software Design Description
SPMP	Software Project Management Plan
SRS	Software Requirement Specification
UAT	User Acceptance Test
UC	Use Case
API	Application Program Interface
UI	User Interface

I. Project Introduction

1. Overview

1.1. Project Information

- Project name: FU House Finder
- Project code: FHF
- Group name: SWP490_G1
- Software type: Web Application

1.2. Project Team

Full Name	Role	Email	Mobile
Nguyen Tat Trung	Lecturer	Trungnt77@fe.edu.vn	0904399139
Nguyen The Giang	Leader	GiangNTHE153046@fpt.edu.vn	0944961228
Phung Quang Thong	Member	ThongPQHE150340 @fpt.edu.vn	0987206969
Bui Ngoc Huyen	Member	HuyenBNHE150346@fpt.edu.vn	0346034217
Nguyen Thu An	Member	AnNTHE150432@fpt.edu.vn	0815709131
Nguyen Tri Kien	Member	KienNTHE150160 @fpt.edu.vn	0987999975

Table 1-1: Team member's information

2. Product Background

First of all, the number of students attending to FPT University increases dramatically each year. From nearly 3000 students in 2019, it rose over 2000 students for the next 2 years, which lead to a consequence of the lack of accommodations. There are just 5 dormitories in the university campus which are able to provide approximately 1500 accommodations for students. As a result, it is very difficult for the students to look for a suitable place.

Admission Office staffs has been attempting to resolve these problems by creating a Facebook group named “[FU-Hòa Lạc] Hội review nhà trọ có tâm” which can help the landlords and students can update information easier. Besides, a lot of problems are still remaining, which can be listed as:

- The data input is still provided manually, which is time-consuming and inefficient. Consequently, these data could not be specific to the room unit.
- The infrequent update of the rented accommodation status could lead to a consequence of inaccurately reflected actual information. Therefore, the results of searching process are of poor quality.
- Students can write a lot of posts to find an accommodation which can make the landlords' posts become vanished.

To respond to the pronounced acceleration of students' rental demand, FU House Finder, with the advantage of qualified information of rentals by the admission department, is developed by the ambition of solving these biggest problems.

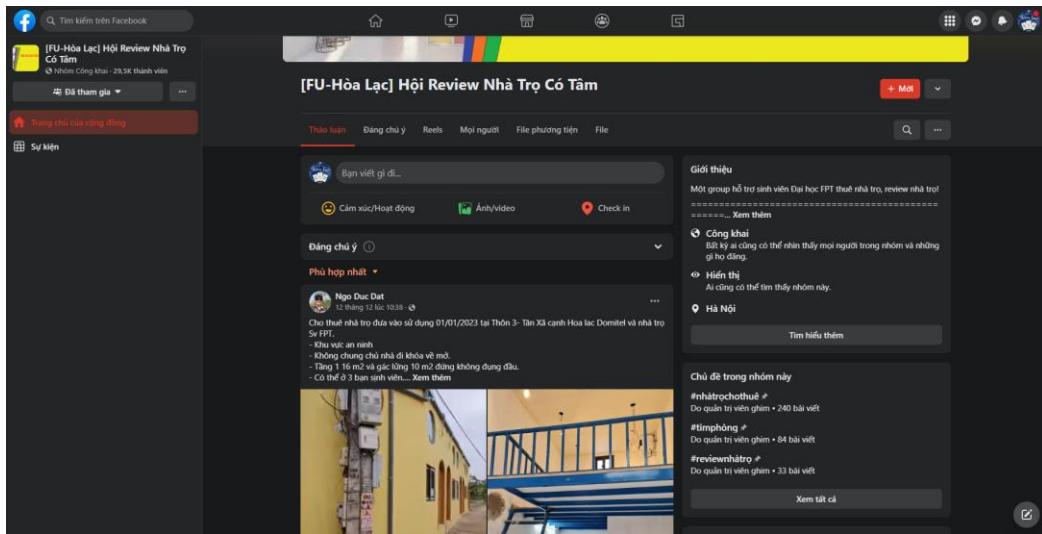


Figure 1-1: [FU-Hòa Lạc] Hội review nhà trọ có tâm Facebook Page

3. Existing Systems

3.1. FPT Can Tho

- Descriptions of the system: Helping students at FPT University Can Tho find accommodation
- Link: <https://dorm.fptcantho.vn/>
- System actors: guest, user, admin
- Features: Search for accommodation in Can Tho
- Pros:
 - There is a lot of information about the inn posted
 - Show full information about the accommodation people want to post
- Cons:
 - Pagination is still very weak
 - Difficult to attract users because the interface is not beautiful

3.2. Nhatot System

- Descriptions of the system: Helping people find accommodation
- Link: <https://www.nhatot.com/thue-phong-tro>
- System actors: guest, user, admin
- Features: Search for accommodation in Vietnam
- Pros:
 - Various hostels in many provinces
- Cons:
 - There are many ads inserted in eye-catching places that make it difficult for users to find the information they want to search

3.3. BatDongSan System

- Descriptions of the system: Helping people find accommodation

- Link: <https://batdongsan.com.vn/cho-thue-nha-tro-phong-tro-bac-tu-liem/gia-tu-1-trieu-den-3-trieu-dt-duoi-30m2>
- System actors: guest, user, admin
- Features: Search for accommodation in Vietnam
- Pros:
 - GUI is good
- Cons:
 - Can't show which rooms are available at the inn

3.4. Blog

- Descriptions of the system: Helping readers know how to convince customers
- Link: <https://timescityminhkhai.com/bi-quyet-luon-luon-lap-day-phong-trong.html>
- System actors: guest, user, admin

4. Business Opportunity

The number of students who enters FPT University is increasing extraordinarily. On the other hand, the number of rentals does not satisfy the demand due to the infrequent update of these rentals' status and inaccessibility of the information. We will address these problems by improving the experience of not only the students but also the landlords. We will develop a helpful system which the landlords could visit and update the status of their rental in an easy way to reflect the reality. The regular updates on these rooms could help students know which houses having available room for them to rent. Consequently, this system will help the landlords from making more money. Solving this problem will be good for our school because if the landlords find this is an advantageous system, they will be ready and willing to pay for it. Moreover, building up our reputation could attract more and more students' attention and reinforce the belief of students' parents in our school.

Additionally, other existing systems provide a large amount of information, usually nationwide but users still cannot find an appropriate rental in the countryside. FU House Finder will focus on a specific area around the university campus which can improve the experience of finding suitable accommodation.

5. Software Product Vision

For students who need to find an appropriate rented accommodation, FU House Finder system is an information system that drives the optimization the process of finding rental. Our team expected to coordinate with the admission department to collect the real data on each specific room to push to google sheet. Unlike the existing websites, our product will provide the most proper information with detailed description, images, etc of every room for the users. After that, we are to run the feature of import excel files to import these massive number of collected data into the database to serving the searching feature, which is expected to be done before October 20th, 2022. We expect to implement this system into the real life with the simplest features at the end of November, 2022 at the Hoa Lac Campus. If this is a successful system implementation, it will be carried out in the other four campuses.

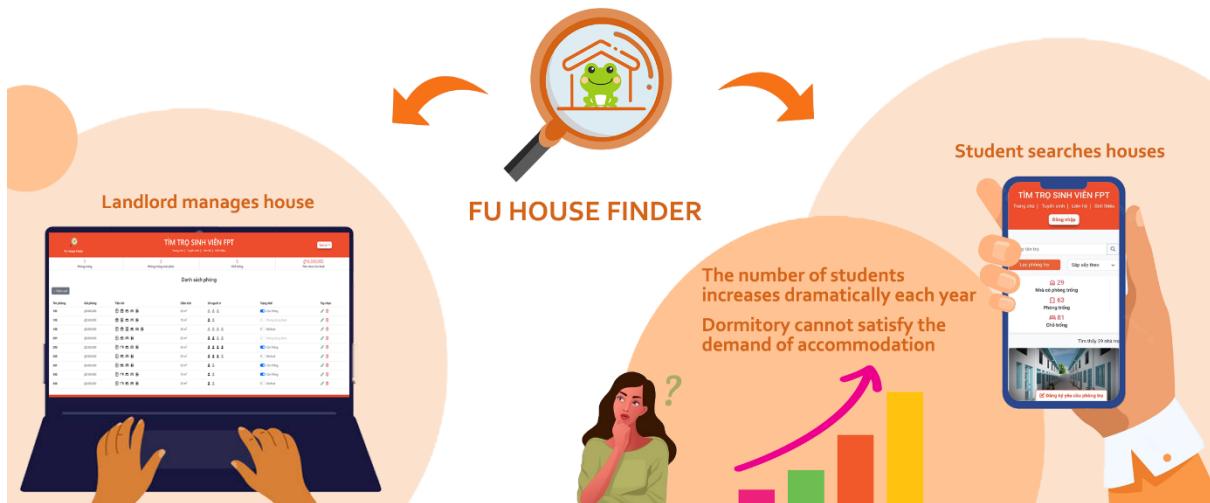
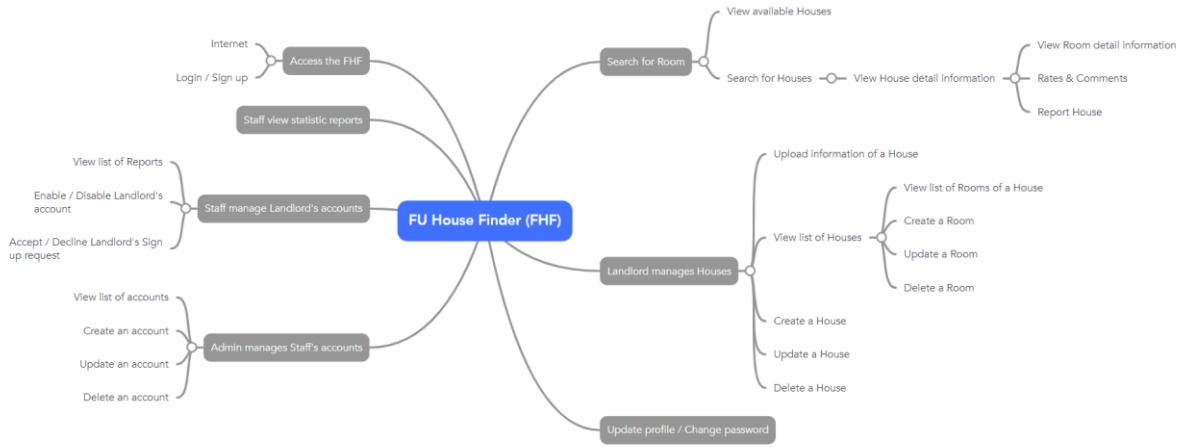


Figure 1-2: Our idea

6. Project Scope & Limitations

6.1. Major Features

- FE-01: Log in to the system using Google, Facebook or Email and Password
- FE-02: View list of available houses in an area around a school campus, search for a house by different criteria, filter houses
- FE-03: View the detail information of a house, view detail information of a room in a house
- FE-04: A student can give rates and comments to review a house, report it if it is violated
- FE-05: A landlord can upload information of a house, including many rooms
- FE-06: A landlord can view, create, update, delete his houses and each room in a house
- FE-07: A landlord can view and update his profile
- FE-08: A staff can view statistics and accept or decline landlord's sign up request
- FE-09: A staff can view and update his profile
- FE-10: A staff can view reports to enable or disable accounts of landlords
- FE-11: An admin can view, create, update, delete all staff's accounts
- FE-12: An admin change his password



6.2. Limitations & Exclusions

Due to the limitation of time, we will not implement some of the intended functions, although we are aware that they are also important and helpful for user of the system:

- LI-1: Students cannot compare 2 houses or 2 rooms by its information
- LI-2: Students cannot interact directly with the landlords via messaging in the system, they will have to make contact through phone or Facebook
- LI-3: Students will not have a wish list to add their favorite houses into
- LI-4: The system will not have the recommendation functionalities to recommend suitable houses to a student based on his history of views

II. Project Management Plan

1. Overview

1.1. Scope & Estimation

#	WBS Item	Complexity	Est. Effort (man-days)
1	<i>Login</i>		29
1.1	(ADMIN / STAFF) Login with email & password	Medium	3
1.2	(LANDLORD / STUDENT) Login with Facebook	Complex	13
1.3	(LANDLORD / STUDENT) Login with Google	Complex	13
2	<i>LANDLORD - Sign Up</i>		21
2.1	Sign Up with Facebook	Complex	9
2.2	Sign Up with Google	Complex	9
2.3	Upload ID Card	Simple	2
2.4	Provide Basic Information	Simple	1
3	<i>GUEST - List of Houses</i>		30
3.1	View House - Room Statistic	Simple	1
3.2	View List of Available Houses	Medium	7
3.2.1	View House Information	Medium	2
3.2.2	Paginate List of Houses	Complex	5
3.3	Filter House	Complex	16
3.3.1	Filter by Name	Simple	2
3.3.2	Filter by Campus	Simple	2
3.3.3	Filter by Region	Simple	2
3.3.4	Filter by Distance	Simple	2
3.3.5	Filter by Price	Simple	2
3.3.6	Filter by Room Type	Simple	2
3.3.7	Filter by House Utility	Simple	1
3.3.8	Filter by Room Utility	Simple	1
3.3.9	Filter by Rate	Simple	2
3.4	Order Houses	Medium	6

3.4.1	Order by Price	Simple	2
3.4.2	Order by Distance	Simple	2
3.4.3	Order by Rate	Simple	2
4	<i>STUDENT – Create Order</i>		3
4.1	Create Order	Simple	3
5	<i>GUEST - House Detail</i>		23
5.1	House detail information	Simple	3
5.2	Landlord Information	Simple	1
5.3	Google Map position	Complex	6
5.4	List of Rate & Comment	Simple	3
5.5	Create Rate & Comment	Medium	4
5.6	List of Available Rooms	Medium	4
5.7	View Room - Capacity Statistics	Simple	2
6	<i>GUEST - Room Detail</i>		2
6.1	View Room Detail Information	Simple	2
7	<i>STUDENT – Send Report</i>		3
7.1	Send Report violation of House	Simple	3
8	<i>LANDLORD - Manage Houses</i>		37
8.1	View List of Houses	Medium	4
8.1.1	View Information of Houses	Simple	2
8.1.2	House – Room Statistics	Simple	2
8.2	Add new House	Complex	16
8.2.1	Provide basic information	Simple	1
8.2.2	Upload house images	Medium	4
8.2.3	Choose Google Map location	Complex	10
8.3	Update House	Complex	12
8.3.1	Update basic information	Simple	1
8.3.2	Update house images	Medium	4
8.3.3	Update Google Map location	Complex	6
8.4	Delete House	Simple	2

8.5	View House's Rates & Comments	Simple	2
8.6	Reply to Rates & Comments	Medium	3
9	<i>LANDLORD - Manage Rooms of House</i>		42
9.1	View List of Rooms	Simple	2
9.2	View Room – Capacity Statistics	Simple	2
9.3	Change Room status	Medium	3
9.4	Add Single Room	Medium	5
9.4.1	Provide basic information	Simple	1
9.4.2	Upload room images	Medium	3
9.5	Add Multiple Room	Complex	21
9.5.1	Download Templates	Medium	2
9.5.2	Import Data	Complex	10
9.5.3	Upload room images	Complex	9
9.6	Update Room	Medium	5
9.6.1	Update basic information	Simple	1
9.6.2	Update room images	Medium	3
9.7	Delete Room	Simple	2
10	<i>LANDLORD - Profile</i>		5
10.1	View Profile	Medium	2
10.2	Update Profile	Medium	3
11	<i>STAFF - Dashboard</i>		13
10.1	View Numerical Statistics	Medium	5
10.2	View Graphical Statistics	Complex	8
12	<i>STAFF - Manage Landlords</i>		30
12.1	View List of Landlords	Medium	8
12.1.1	View Landlord Information	Simple	2
12.1.2	Search Landlord by Name	Simple	2
12.1.3	Paginate List of Landlords	Medium	4
12.2	View House – Room Statistics	Simple	2
12.3	Change Landlord Active Status	Simple	3

12.4	Landlord Details	Complex	17
12.4.1	View Landlord's Details information	Simple	3
12.4.2	View Landlord's List of Houses	Medium	5
12.4.3	View House Detail information	Simple	3
12.4.4	View House's List of Rooms	Medium	4
12.4.5	View Room Detail information	Simple	2
13	STAFF – List of Houses		13
13.1	View List of Houses	Medium	8
13.1.1	View House Information	Simple	2
13.1.2	Search House by Name	Simple	2
13.1.3	Paginate List of Houses	Medium	3
13.2	View House – Room Statistics	Simple	1
13.3	View House Details	Medium	5
13.3.1	View House Detail information	Simple	2
13.3.2	View House's List of Rooms	Simple	2
13.3.3	View Room Detail information	Simple	1
14	STAFF – List of Reports		37
14.1	View List of all Reports	Complex	18
14.1.1	View List of Reports	Simple	2
14.1.2	Filter List of Reports	Complex	7
14.1.3	Paginate List of Reports	Medium	4
14.1.4	View Report Detail	Medium	3
14.1.5	Change Status of Report	Medium	2
14.2	View Reports by House	Complex	19
14.2.1	View List of Reported Houses	Simple	3
14.2.2	Filter List of Reported Houses	Complex	7
14.2.3	Paginate List of Reported Houses	Medium	4
14.2.4	View List Reports of a House	Medium	3
14.2.5	Change Status of House	Medium	2
15	STAFF - Landlord Sign Up Requests		6

15.1	View List of Sign Up Request		Simple	2
15.2	Approve/Reject Sign Up Request		Medium	4
16	STAFF – Manage Orders			25
16.1	View Numerical Statistics		Medium	4
16.2	View Graphical Statistics		Medium	5
16.3	View List of Orders		Complex	17
16.3.1	View List of Orders		Simple	2
16.3.2	Filter List of Orders		Complex	6
16.3.3	Paginate List of Orders		Medium	3
16.3.4	View Orders Detail		Medium	3
16.3.5	Change Status of Orders		Medium	2
17	STAFF - Profile			5
17.1	View Profile		Medium	2
17.2	Update Profile		Medium	3
18	STAFF – Change Password			3
18.1	Change Password		Medium	3
19	ADMIN - Manage Staff Accounts			10
19.1	View List of Staffs		Simple	2
19.2	Add new Staff		Simple	3
19.3	Update Staff		Simple	3
19.4	Delete Staff		Simple	2

Total Estimated Effort (man-days) **337**

Table 2-1: Project scope & estimation

1.2. Project Objectives

#	Testing Stage	Test Coverage	No. of Defects	% of Defect	Notes
1	Reviewing	100%	20	45%	
2	Unit Test	100%	9	20.5%	
3	Integration Test	100%	9	20.5%	

4	System Test	100%	3	7%	
5	Acceptance Test	100%	3	7%	

Table 2-2: Project objectives

Milestone Timelines (%): 100

Allocated Effort (man-days): 337

1.3. Project Risks

#	Risk Description	Avoidance plan	Contingency plan	Status
1	Failure to meet deadline	<ul style="list-style-type: none"> - Plan and develop schedule carefully - Assign tasks carefully 	<ul style="list-style-type: none"> - Find the root cause of the problem - Reassign tasks - Change project scope 	Closed
2	Change in requirements	<ul style="list-style-type: none"> - The supervisor and the entire team must review any new updates to requirements 	<ul style="list-style-type: none"> - All changes in requirements will be announced in the next daily team meeting 	Closed
3	Misunderstanding of requirements	<ul style="list-style-type: none"> - Discuss requirements carefully with the customer - Any ambiguity in understanding requirements of team members will be recorded and handed to supervisor to clarify with customer 	<ul style="list-style-type: none"> - Update code and documentation to adapt with actual requirements 	Closed
4	Illness or absence of team members	<ul style="list-style-type: none"> - Provide meeting schedules in advance - Team member must announce absence in advance 	<ul style="list-style-type: none"> - All meetings with supervisor will be recorded for absent members - Assign the tasks of absent member to other members - Work overtime if necessary 	Closed
5	Conflict between team members	<ul style="list-style-type: none"> - Everything must be documented - Every team member has to express clearly and carefully 	<ul style="list-style-type: none"> - Make sure any miscommunication will be resolved 	Closed
6	Data loss	<ul style="list-style-type: none"> - Use GitHub for version control 	<ul style="list-style-type: none"> - Restore backup data from GitHub 	Closed

		<ul style="list-style-type: none"> - Train team members on Git usage and conflict resolution 		
7	Internet connection issue in Capstone project defense	<ul style="list-style-type: none"> - Prepare personal wireless internet connection 	<ul style="list-style-type: none"> - Demo project on localhost - Record demo video before the Capstone project defense 	Closed
8	Server failure	<ul style="list-style-type: none"> - Use paid and certified servers 	<ul style="list-style-type: none"> - Use a different server 	Closed

Table 2-3: Project risks

2. Management Approach

2.1. Project Process

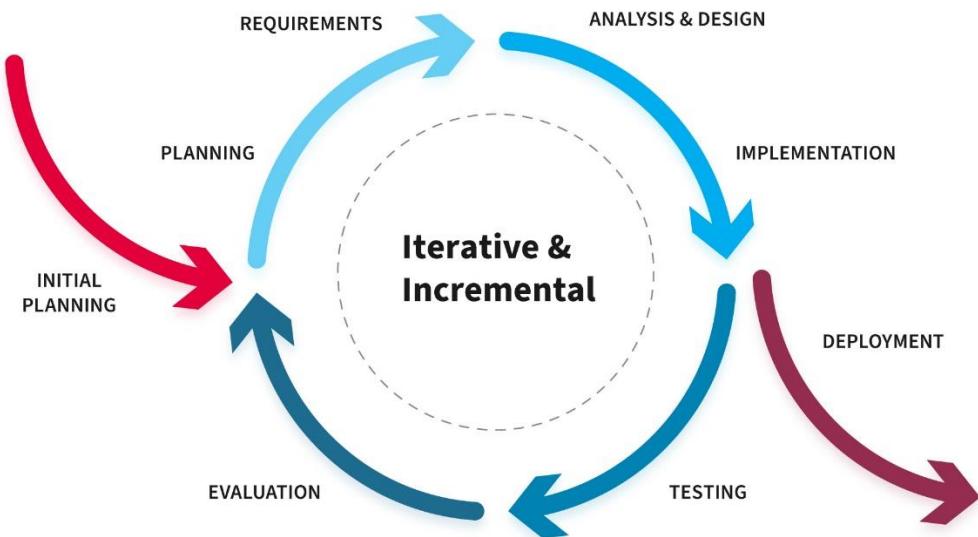


Figure 2-1: Iterative and Incremental Software Process Model

Based on recommendation of supervisor and actual situation of the project, the FHF Project team chooses to use the Iterative and Incremental Software Process Model. In an Iterative and Incremental model, initially, a partial implementation of a total system is constructed so that it will be in a deliverable state. The Iterative and Incremental model is mostly used when the scope of the project is big, the major requirements are defined clearly, some more details will be added later in software development, which is the case of this project. By using this software process model, we break down the developing system task into a series of smaller tasks which will be completed separately, allowing us to take advantage of what was learned during the development of earlier parts of the system.

The reasons for the project to choose this model are:

- You can develop some main functions that prioritize requirements first.
- Requirements changes can be easily accommodated.
- Testing and debugging during smaller iterations is easy.

- Client can give input to every item increment, accordingly maintaining a strategic distance from differences toward finish of improvement.

2.2. Quality Management

2.2.1 Code Review

- Any contribution of team members will be reviewed by team leader before merging into main branch.
- If team leader finds any defect or ambiguity in code or document, he will discuss directly with the author to clarify problems and give instruction on how to fix them.
- Team leader also checks if the added code follows code convention. If not, the author will be assigned to fix it.

2.2.2 Unit Testing

- As each team member will be not only a developer but also a tester, each person will be testing for what they code.
- Any defects found will be recorded on GitHub Issues as a bug tracking software, with details and images for evidence.
- That team member will then be responsible for repairing those defects.

2.2.3 Integration Testing

- All Integration test cases will be recorded into a spread sheet, divided by feature. The team member who developed the feature will perform Integration test for that feature.
- Any defects found will also be recorded on GitHub Issues and will be fixed by that team member.

2.2.4 System Testing

- In the team meeting at the end of each Iteration, all team will discuss to write some System test cases that can cover all features.
- Any defects found will also be recorded on GitHub Issues and will be fixed by the team member who is responsible for the feature to which it belongs.

2.2.5 Acceptance Testing

- At the end of each Iteration, system will be deployed on the Internet and team members will contact directly with actual landlords and students to use.
- Any feedbacks and comments from the client will be recorded and bring to discuss with whole team and supervisor to decide what should be done to improve the system.

2.3. Training Plan

Training Area	Participants	When, Duration	Detail
Use Angular 12	All Members	11/09/2022 - 17/09/2022	Member self-study
Use Git, Github	All Members	09/09/2022	Member self-study

Install SQL Server 2019	All Members	12/09/2022	Trainer: GiangNT
Install MkCert	All Members	12/10/2022	Trainer: AnNT
Run localhost on Mobile	All Members	05/11/2022	Trainer: GiangNT
Unit Test with NUnit	All Members	15/10/2022	Trainer: GiangNT Record: https://youtu.be/0yO0tn5J8Rw https://youtu.be/8ArLBM3dFCs

Table 2-4: Training plan

My files > SEP490 > Environment_Setup ↗

Name	Modified	Modified By	File size	Sharing
\Unit	November 20	Giang Nguyễn Thế	2 items	↗ Shared
1 số hướng dẫn về Authorization & Upload...	November 23	Giang Nguyễn Thế	138 KB	↗ Shared
appsettings.json	December 2	Giang Nguyễn Thế	834 bytes	↗ Shared
appsettings_deploy.json	Yesterday at 5:05 PM	Giang Nguyễn Thế	905 bytes	↗ Shared
Hướng dẫn cài SQL Server 2019.docx	A few seconds ago	Giang Nguyễn Thế	122 KB	↗ Shared
Hướng dẫn MkCert.docx	About a minute ago	Giang Nguyễn Thế	829 KB	↗ Shared
Hướng dẫn vào Localhost trên Mobile.docx	November 5	Giang Nguyễn Thế	518 KB	↗ Shared

Figure 2-3: Training Guide – Document

The screenshot shows a messaging application interface with a sidebar containing a 'FU House Finder' section and a search bar for '# cài đặt môi trường-project'. The main area displays a conversation thread:

- Message 1: 'Hướng dẫn sử dụng Jira' (with a link to a document titled 'Hướng dẫn cài đặt môi trường Code và các Tool liên quan').
- Message 2: 'WEB' (with a list of steps for setting up Jira).
- Message 3: 'APP MOBILE' (with a list of steps for setting up mobile applications).
- Message 4: 'Figma: Link Update' (with a link to a Figma document titled 'Hướng dẫn cài đặt SQL Server 2019').
- Message 5: 'Hướng dẫn cài đặt mkcert để chạy Project Angular bằng' (with a link to a document titled 'Hướng dẫn chạy Project Angular trên Điện thoại')).
- Message 6: 'Hướng dẫn Unit Test với NUnit' (with a link to a video record titled 'Hướng dẫn Unit Test với NUnit').

Figure 2-4: Training Guide – Document (2)

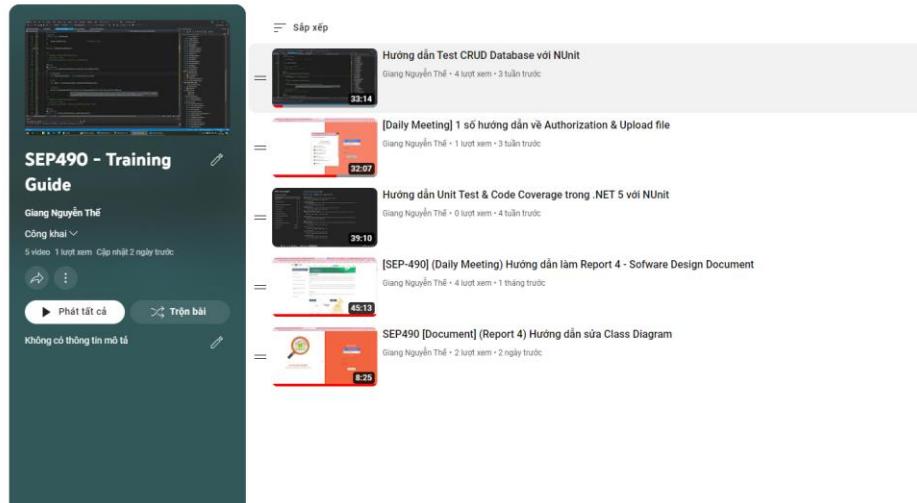


Figure 2-5: Training Guide - Record

3. Project Deliverables

#	Deliverable	Due Date	Notes
1	Product Vision Basic Use case System actors Report 1	10/09/2022	Initiate Project
2	Project Scope Requirements Business Rules	15/09/2022	Customer Meeting Functional Requirements Non-functional Requirements
3	Plan and Schedule Report 2	08/10/2022	Project Schedule Project Management Plan
4	System Design Report 3 Report 4	15/10/2022	Screen Design Architectural Design & Detail Design Database Design
5	Code & Implement Report 7 Iteration 1	30/10/2022	Source Code Unit Test cases & report Integration Test cases & report
6	Code & Implement Report 7 Iteration 2	18/12/2022	Source Code Unit Test cases & report Integration Test cases & report System Test cases & report User Acceptance Test
10	Final Report Final Product User Guides Presentation	19/12/2022	Final Report User Guides Presentation Slides

Table 2-5: Project deliverable

4. Responsibility Assignments

D~Do; R~Review; S~Support; I~Informed; <blank>- Omitted

Responsibility	GiangNTHE153046	HuyenBNHE150346	ThongPQHE150340	AnNTHE150432	KienNTHE150160
Collect Requirements	R,D	D	D	D	D
Prepare Project Introduction Document	R,D	I	D	I	I
Prepare Project Management Plan	R	D	I	D	R
Prepare SRS Document (User Requirements)	R,D	I	I	I	D
Create Screen Mockups	R,D	D	D	D	D
Design Database	R,D	D	D	D	D
Collect Administrative Unit Data	D	I	I	I	I
Draw Use Case Diagram	R	D	I	I	I
Code Function Login	R	I	I	D	I
Draw Entity Relationship Diagram	R	D	I	I	I
Draw Screen Flow Diagrams	D	I	I	I	I
Code function: Home Page	R,D	I	I	S	I
Code functions of Staff	R	D	I	I	D
Code functions of Landlord	R	I	D	D	I
Code functions of Admin	R	I	I	D	I
Prepare other Documents	R,D	D	D	D	D

Table 2-6: Responsibility assignments

5. Project Communications

Communication Item	Who/ Target	Purpose	When, Frequency	Type, Tool, Method(s)
Daily Meeting	All team members	- Checking progress of assigned tasks - Finding solutions for difficult problems	9 p.m everyday	Discord
Assign Tasks	All team members	- Project Manager assigns tasks to other team members	Everyday	Jira

Weekly Meeting With Supervisor	All team members, supervisor	- Checking progress of project - Plan upcoming tasks - Update requirements	Once a week	Offline
Meeting with Supervisor and University Staffs	All team members, supervisor, Staffs	- Clarify user requirements, roles & project scope	16/09/2022 13/09/2022	Offline
Meeting with Landlords to collect data	All team members, supervisor, Landlords	- Clarify user requirements	12/12/2022	Offline

Table 2-7: Project communications



Figure 2-6: Meeting with Supervisor and Staffs of the University (13/09/2022)



Figure 2-7, 2-8: Meeting at the University's Enrollment day (16/09/2022)



Figure 2-9, 2-10: Meeting with Landlords to collect data (12/12/2022)

6. Configuration Management

6.1. Document Management

- Management Tools:
 - Google Drive
 - OneDrive
 - GitHub
- Team leader assigns and describes tasks through Facebook & Discords. All Documents will be submitted to management tools to keep track of changes. Team leader then collects them all to submit to the mentor.

6.2. Source Code Management

- Management Tools:
 - GitHub
- Team leader assigns and describes tasks through Facebook & Discords. All source code will be pushed to Github for version control. Team leader then pull it back to run on deployment environment.

6.3. Tools & Infrastructures

Category	Tools / Infrastructure
Technology	Angular 12 (Front-end); ASP.NET 5 (Back-end)
Database	Microsoft SQL Server 2019
IDEs/Editors	Visual Studio Code; Visual Studio
Diagramming	DrawIO; Mindmeister
Documentation	Google Docs; Microsoft Office
Version Control	GitHub (Source Codes); Google Drive (Documents); OneDrive (Documents)
Deployment server	Amazon S3; Vercel

Project management	Jira (Schedule, Tasks, Defects)
UI/UX Design	Figma; Adobe XD; Adobe Photoshop
Development tools	MkCert; MobaXterm
Communication tools	Discord; Facebook; Messenger; Google Meet; Zalo
Test tools	NUnit; Postman

Table 2-8: Tools

7. Team Structure

In FHF Project Team, everyone can do and will do everything, so that each person can have all the skills needed to become a full-stack developer.

Team Member	Role
GiangNT	Project Manager, Quality Assurance Manager, Test Leader, Developer, Designer, Tester, Business Analyst
HuyenBN	Developer, Designer, Tester, Business Analyst
ThongPQ	Developer, Designer, Tester, Business Analyst
KienNT	Developer, Designer, Tester, Business Analyst
AnNT	Developer, Designer, Tester, Business Analyst, Technical Leader

Table 2-9: Project Team Member

III. Software Requirement Specification

1. Product Overview

The FU House Finder System is a new software system that helps the students and the landlords find each other through the Internet. The context diagram below illustrates the external entities and system interfaces for release 1.0. The system is expected to evolve over several releases, ultimately having the ability to recommend suitable houses for students based on view history and also earn money for the host.

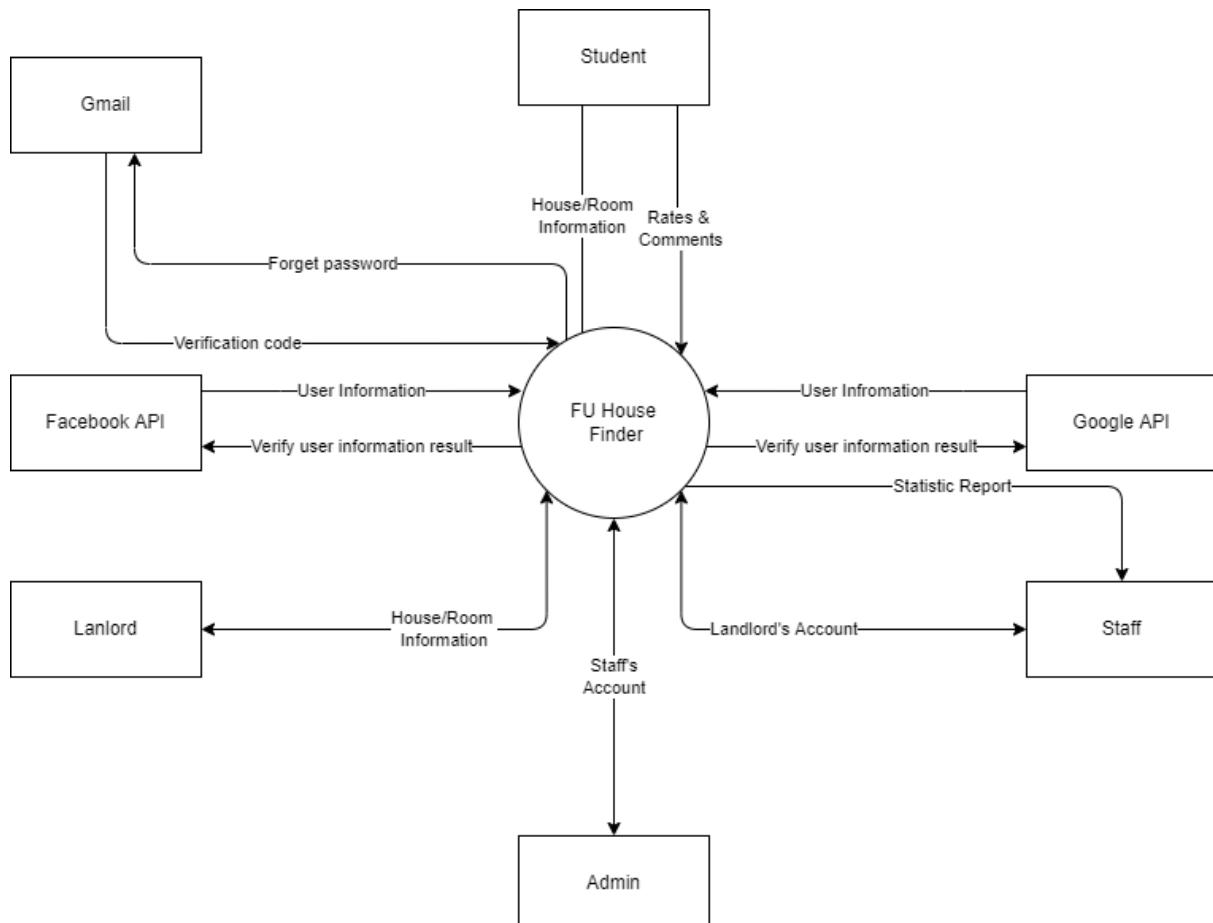


Figure 3-1: Context diagram

2. User Requirements

2.1. Actors

#	Actor	Description
1	Administrator	Administrator of the system; Manage accounts of Staffs
2	Staff	Staffs of the offices of the University; Manage accounts of Landlords and view statistics reports
3	Landlord	People having houses for rent; Manage their houses and rooms and their information

4	Student	People finding for houses to rent; Can search for available houses in the system
---	---------	--

Table 3-1: Actor description

2.2. Use Cases

2.2.1 Diagram

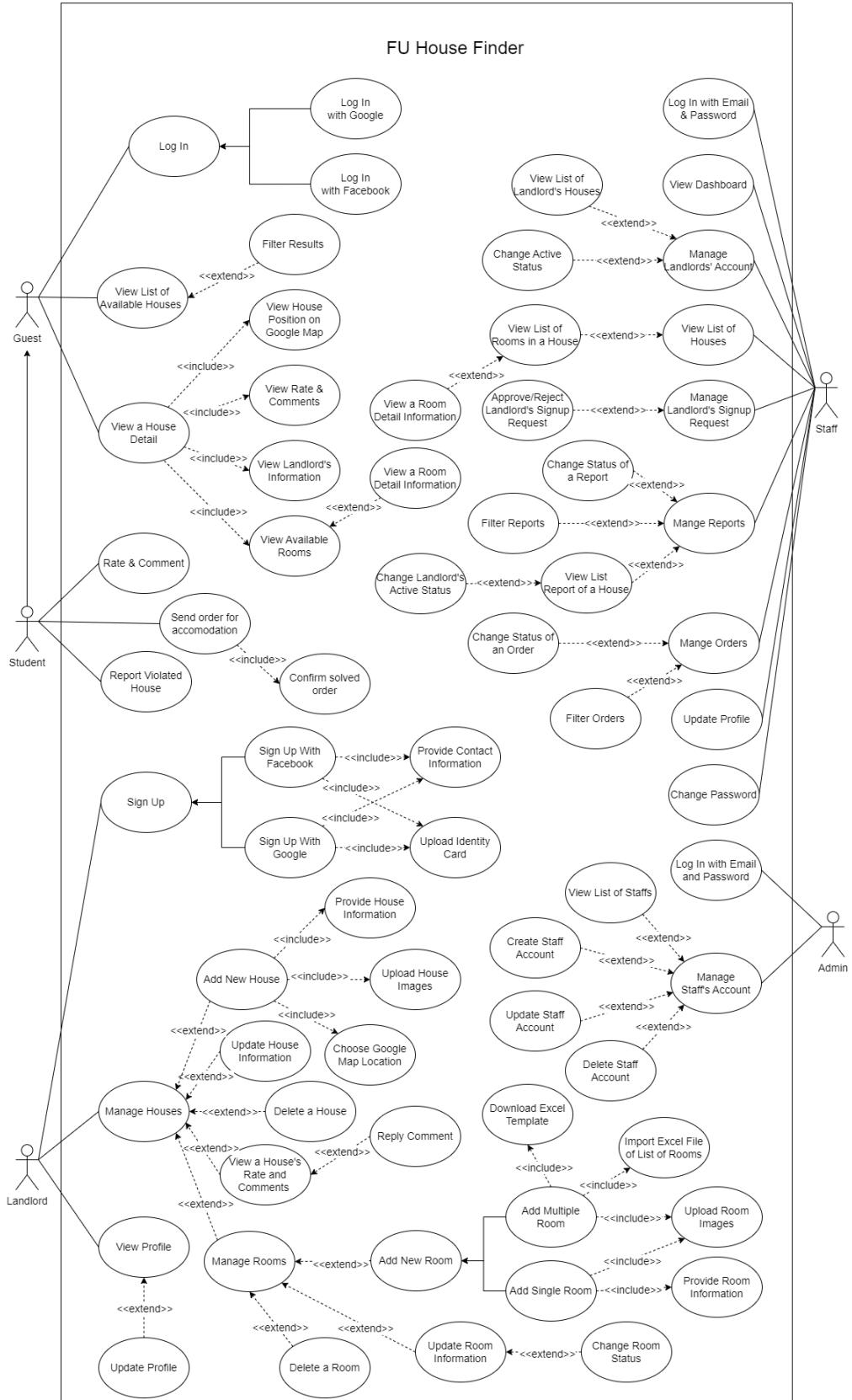


Figure 3-2: Use Case Diagram

2.2.2 Descriptions

ID	Use Case	Actors	Use Case Description
01	Log In	Guest	Guest logs into the system
02	View List Of Available Houses	Guest	Guest views list of houses
03	Filter Available Houses	Guest	Guest searches for houses by name or filters it by category or by map
04	View a House Detail	Guest	Guest views the detail information of house
05	View House Position on Google Map	Guest	Guest views specific location of a house on Google Map
06	View Rate and Comments	Guest	Guest views other user rate and reviews of a house
07	View Landlord's Information	Guest	Guest views landlord's information of a house
08	View Available Rooms	Guest	Guest views list of available rooms in a house
09	View a Room Detail Information	Guest	Guest views a room's detail information
10	Rate and Comment	Student	Student rates and comments the house
11	Send Order for Accommodation	Student	Student fills out his contact information and desired demand to Staff
12	Report Violated House	Student	Student reports the violated house
13	Log In with Email and Password	Admin	Admin logs into the system
14	Manage Staff's Account	Admin	Admin views the list of all staff accounts and can be change active status of the staff
15	Change Password	Admin	Admin changes login password
16	Log In with Email and Password	Staff	Staff logs into the system
17	View Dashboard	Staff	Staff views list of available houses and list of landlord's request. Staff can be search for houses & statistics by name or filter it by category at the same time can also accept or decline landlord's request
18	Manage Landlords' Account	Staff	Staff views list of landlords' accounts
19	View List of Landlord's Houses	Staff	Staff views list of all houses of a landlord
20	Change Active Status	Staff	Staff changes active status of a landlord then houses of this landlord does not appear for students
21	View List of Houses	Staff	Staff views list of all houses in the system
22	View List Rooms of a House	Staff	Staff views list of all rooms of a house

23	View a Room Detail Information	Staff	Staff views detail information of a room
24	Manage Landlords' Signup Request	Staff	Staff views a list of all landlords' signup requests
25	Approve/Reject Landlord's Signup Request	Staff	Staff approves or rejects a landlord's signup request
26	Manage Reports	Staff	Staff views a list of reports send by students
27	Change Status of a Report	Staff	Staff changes resolution status of a report
28	Filter Reports	Staff	Staff filters reports by reported date, resolution status
29	View List of Reported House	Staff	Staff views a list of reported houses
30	View List Report of a House	Staff	Staff views a list of reports send by students of a house
31	Manage Orders	Staff	Staff views list of students' accommodation orders
32	Change Status of an Order		Staff changes the resolution status of an order
33	Filter Orders	Staff	Staff filters orders by ordered date, resolution status, etc
34	Update Profile	Staff	Staff views and updates the profile
35	Change Password	Staff	Staff changes password to log in the system
36	Sign Up	Landlord	Landlord registers for an account
37	Provide Contact Information	Landlord	Landlord provides his contact information for register account
38	Upload Identity Card	Landlord	Landlord uploads his identity card imgae for register account
39	Log In	Landlord	Landlord logs into the system
40	Manage Houses	Landlord	Landlord views list of rooms, detail information of room at the same time can also add new, update, delete, change status the room
41	Add New House	Landlord	Landlord provides information of a new house to add it to the system
42	Provide House Information	Landlord	Landlord imports information of the house by excel file
43	Upload House Images	Landlord	Landlord uploads images of a house
44	Choose Google Map Location	Landlord	Landlord chooses a location on Google Map for a house
45	Update House Information	Landlord	Landlord updates detail information of a house
46	Delete a House	Landlord	Landlord deletes a house
47	View House's Rate and Comments	Landlord	Landlord views students' rate and comments on one of their house
48	Reply Comment	Landlord	Landlord replies students' comments
49	Add New Room	Landlord	Landlord adds new room to a house
50	Add Single Room	Landlord	Landlord fills out information in a form to add a new room to a house

51	Add Multiple Room	Landlord	Landlord uploads templated excel file to upload a list of rooms to a house
52	Download Excel Template	Landlord	Landlord downloads an excel template to fill out information of rooms
53	Import Excel File of List Rooms	Landlord	Landlord imports an excel file to add a list of rooms to a house
54	Upload Room Images	Landlord	Landlord uploads a folder of images of rooms named as convention
55	Provide Room Information	Landlord	Landlord provides detail information of a room
56	Update Room Information	Landlord	Landlord updates detail information of a room
57	Change Room Status	Landlord	Landlord changes the availability status of a room
58	Delete a Room	Landlord	Landlord deletes a room
59	View Profile	Landlord	Landlord views the profile
60	Update Profile	Landlord	Landlord updates the profile

Table 3-2: Use case Description

3. Functional Requirements

3.1. System Functional Overview

3.1.1 Screens Flow

Student

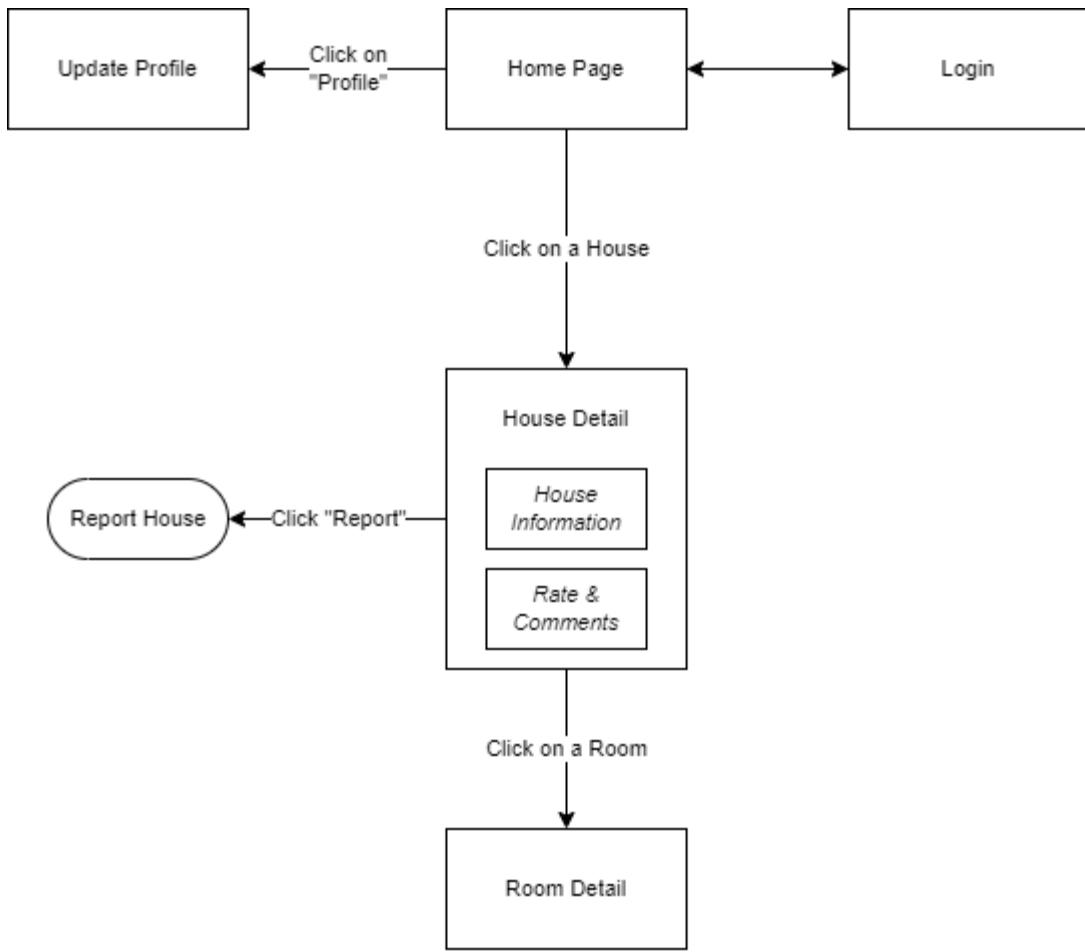


Figure 3-3: Screen Flow for Student

Landlord

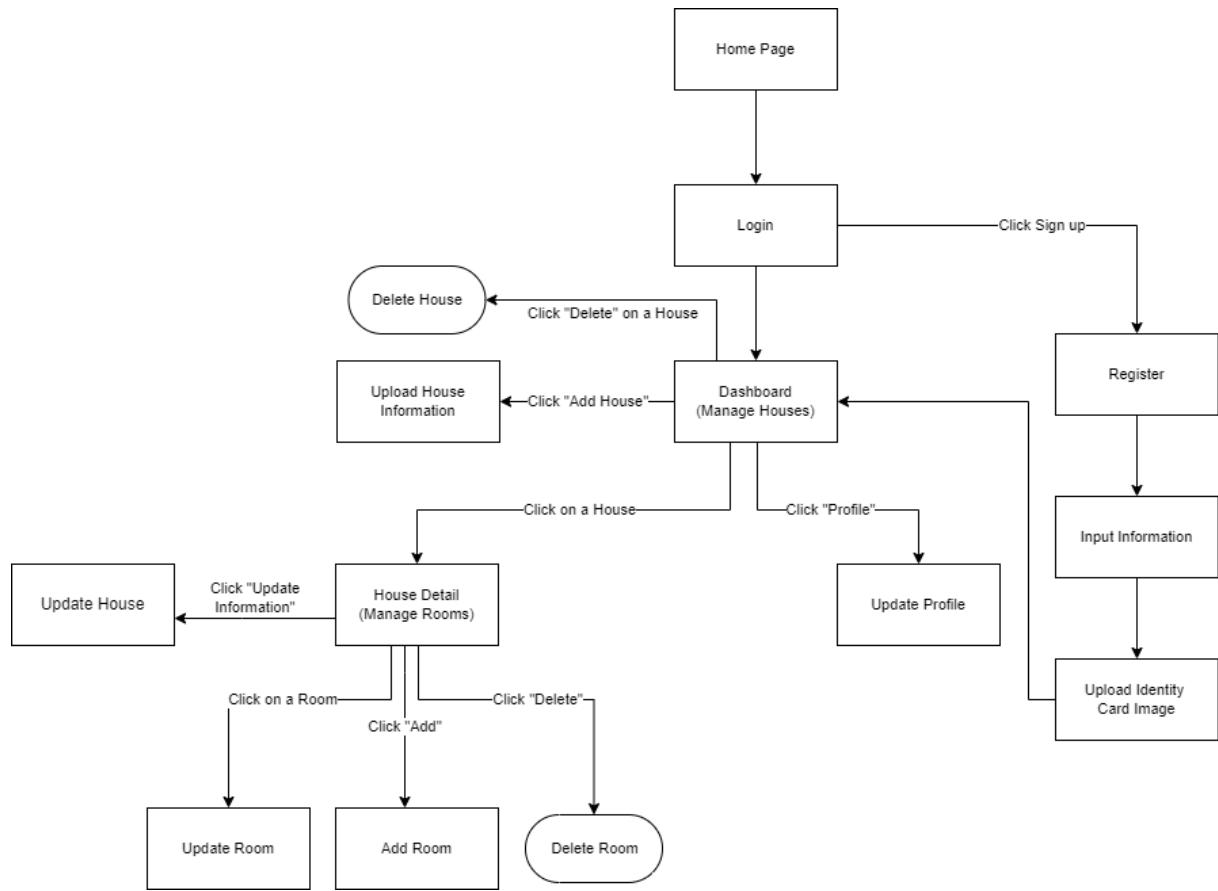


Figure 3-4: Screen Flow for Landlord

Staff

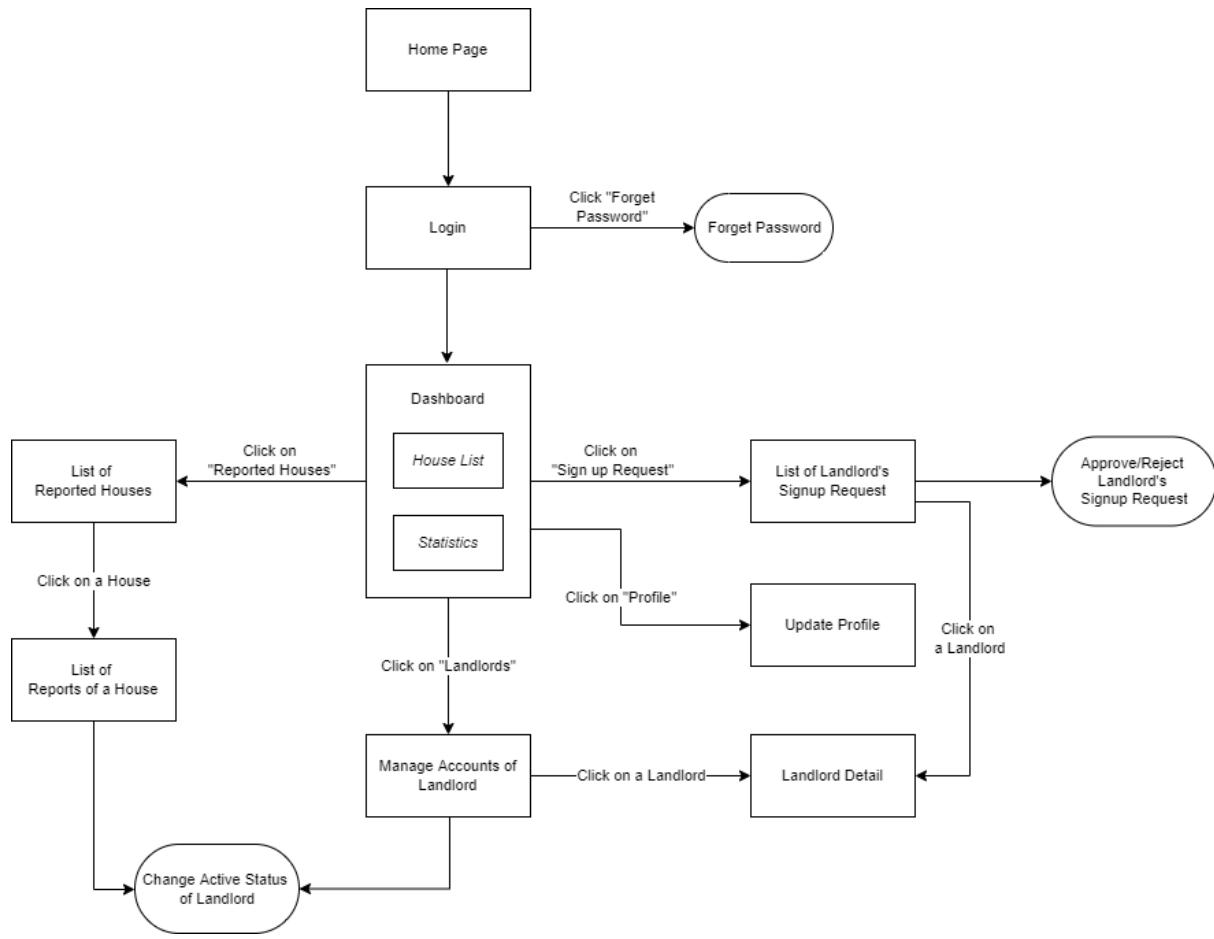


Figure 3-5: Screen Flow for Staff

Admin

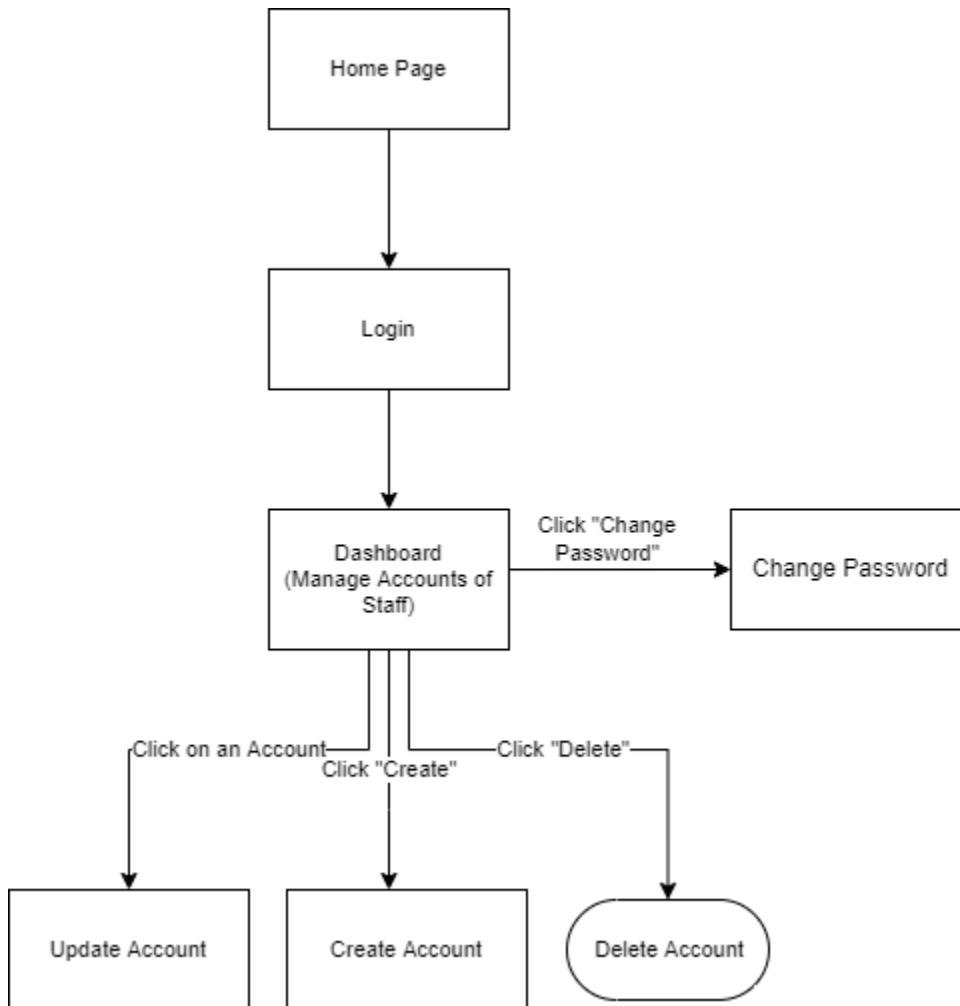


Figure 3-6: Screen Flow for Admin

3.1.2 Screen Descriptions

#	Feature	Screen	Description
1	View List Houses	Home Page	View list of houses
2	Search & Filter House	Home Page	Search for houses by name or filter it by category
3	Interact with map	Home Page	Use the map to find the location of the house
4	View House Detail	House Details	Display detail information of house
5	View Room Detail	Room Details	Display detail information of room
6	Report House	Report House	User reports the house
7	Login	User Login	User logs into the system
8	Register	User Register	User registers for an account
9	Register	Register [Landlord]	Landlord registers for an account

10	Verify Information Of Landlord	Input Information [Landlord]	Get information of landlord
11	Verify Identity Card Of Landlord	Upload Identity Card Image [Landlord]	Get identity card of landlord
12	View List Houses	Dashboard [Landlord]	Landlord views list of houses
13	Search & Filter House	Dashboard [Landlord]	Landlord searché for houses by name or filters it by category
14	Upload House Information	Upload House Information [Landlord]	Landlord imports information of the house
15	Manage Rooms	House Detail [Landlord]	Display detail information of house
16	Search & Filter Room	House Detail [Landlord]	Landlord searché for rooms by name or filters it by category
17	Update House	Update House [Landlord]	Landlord updates the house
18	Delete House	Delete House [Landlord]	Landlord deletes the house
19	Update Room	Update Room [Landlord]	Landlord updates the room
20	Add Room	Add Room [Landlord]	Landlord adds new the room
21	Delete Room	Delete Room [Landlord]	Landlord deletes the room
22	Update Proflie	Update Proflie [Landlord]	Landlord updates the proflie
23	View List Houses & Statistics	Dashboard [Staff]	Staff view list of available houses
24	Search & Filter House & Statistic	Dashboard [Staff]	Staff searches for houses & statistics by name or filter it by category
25	Accept Or Decline Landlord's Signup Request	List Of Landlord's Signup Request [Staff]	Staff views list of landlord's request and accepts or decline
26	Update Proflie	Update Proflie [Staff]	Staff updates the proflie
27	View List Accounts Of Landlords	Landlords List [Staff]	Staff views list of landlords

28	View Landlord Detail	Landlord Detail [Staff]	Staff views detail information of landlord
29	View List Of Reported Houses	List Of Reported Houses [Staff]	Staff views list of reported houses
30	View List Accounts Of Staffs	Dashboard [Admin]	Admin views the list of all staff accounts
31	Search & Filter Staff	Dashboard [Admin]	Admin searches for staffs by name or filter it by category
32	Change Password	Change Password [Admin]	Admin changes login password
33	Create Staff Account	Create Account [Admin]	Admin creates account for staff
34	Update Staff Account	Update Account [Admin]	Admin updates account of staff
35	Delete Staff Account	Delete Account [Admin]	Admin deletes account of staff

Table 3-3: Screen description

3.1.3 Screen Authorization

Screen	Guest	Student	Landlord	Staff	Admin
Home Page	X	X	X	X	X
House Details	X	X	X	X	X
Room Details	X	X	X	X	X
Report House		X			
User Login		X	X	X	X
User Register	X				
Register [Landlord]			X		
Input Information [Landlord]			X		
Upload Identity Card Image [Landlord]			X		
Dashboard [Landlord]			X		
Upload House Information [Landlord]			X		
House Detail [Landlord]			X		
Update House [Landlord]			X		
Delete House [Landlord]			X		
Update Room [Landlord]			X		
Add Room [Landlord]			X		
Delete Room [Landlord]			X		
Update Profile [Landlord]			X		
Dashboard [Staff]				X	

List Of Landlord's Signup Request [Staff]				X	
Update Profle [Staff]				X	
Landlords List [Staff]				X	
Landlord Detail [Staff]				X	
List Of Reported Houses [Staff]				X	
Dashboard [Admin]					X
Change Password [Admin]					X
Create Account [Admin]					X
Update Account [Admin]					X
Detele Account [Admin]					X

Table 3-4: Screen authorization

3.1.4 Non-Screen Functions

#	Feature	System Function	Description
1	Login with Facebook	Login	User logs in the system with his Facebook account
2	Login with Google	Login	User logs in the system with his Google account
3	Import File Excel	Sign up & Manage Houses	Convert data from Excel file into house & room information
4	Forget Password	Login	System send verification code to user Email to reset password

Table 3-5: Non-screen functions

3.1.5 Entity Relationship Diagram

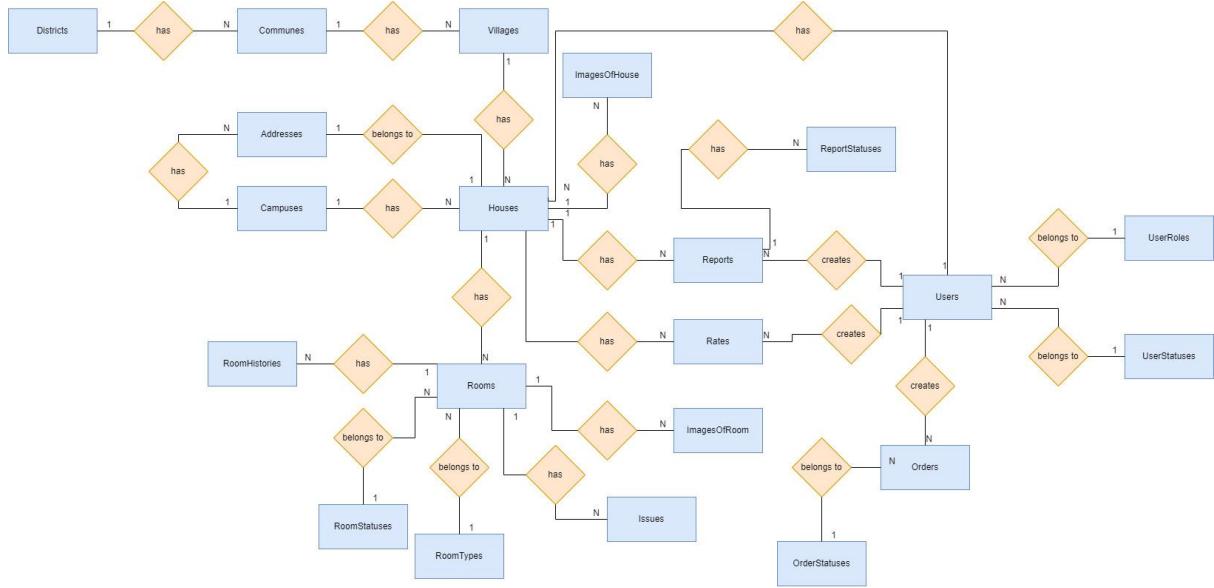


Figure 3-7: Entity Relationship Diagram

Entities Description

#	Entity	Description
1	Addresses	Used to store the address of User, House & Campus
2	Campuses	Used to store Campuses of FPT university
3	UserRoles	Used to store the role of each User
4	Users	Used to store Users in the system
5	Districts	Used to store Districts around the school
6	Communes	Used to store Communes around the school
7	Villages	Used to store Villages around the school
8	Houses	Used to store Houses
9	Statuses	Used to store Statuses of Room
10	RoomTypes	Used to store Types of Room
11	Rooms	Used to store Rooms
12	Rates	Used to store Rating & Comments of User
13	ImagesOfHouse	Used to store Images of House
14	ImagesOfRoom	Used to store Images of Room
15	Reports	Used to store student's Reports for House
16	RoomHistories	Used to store Histories of people staying in the House, for the landlord to voluntarily add if there is a need to manage & monitor

Table 3-6: Entity description

3.2. Home page

- Function trigger: User accesses the website
- Function description:
 - Role: Guest, Student
 - Purpose: User could view list of available houses and filter the results

- Screen layout:

The screenshot shows the homepage of the FU House Finder website. At the top, there's a navigation bar with links to 'Trang chủ', 'Cơ sở', 'Liên hệ', and 'Giới thiệu'. On the right, there's a 'Đăng nhập' button. Below the navigation is a search bar with placeholder text 'Nhập tên trọ, từ khóa tìm kiếm' and a magnifying glass icon. To the left, there's a sidebar titled 'LỌC PHÒNG TRỌ' with dropdown menus for 'Cơ sở', 'Bán kính: 0 - 10km', 'Khoảng giá: 0 - 20.000.000đ', 'Khu vực: Huyện: [dropdown], Xã: [dropdown], Thủ: [dropdown]', and a checkbox section for 'Tiện ích khác' (Wifi, Điều hòa, Nóng lạnh, Khóa vân tay, Camera an ninh). A 'Lọc' (Filter) button is at the bottom. The main content area displays a grid of apartment listings. Each listing includes a thumbnail image, the name of the place (e.g., Hòa Lạc Apartment, Hòa Lạc Link), price range (e.g., 2.000.000 - 5.000.000), location (e.g., Thủ 8, Thạch Hòa, Thạch Thất), distance (e.g., 3.5km), and contact number (e.g., 0896899996). There are also buttons for 'Sắp xếp theo' (Sort by) and another 'Lọc' button. The footer features the FPT Education logo and the text 'FPT EDUCATION GLOBAL'.

3.3. Login/Signup

- Function trigger: User clicks “Đăng nhập” from Home Page
- Function description:
 - Role: Guest
 - Purpose: Guest could sign up for an account or log in to access more functions
- Screen layout:

The screenshot shows the login/signup screen for the FU House Finder application. The background is orange. At the top right, there's a large button labeled 'ĐĂNG NHẬP'. Below it are two buttons: one for 'Đăng nhập bằng Facebook' with a Google 'G' icon and one for 'Đăng nhập bằng Google' with a Facebook 'f' icon. At the bottom right, there's a link 'Đăng ký tài khoản?'. On the left side, there's a decorative graphic featuring a magnifying glass over a house with a frog inside, and the text 'FU HOUSE FINDER' and 'Ứng dụng tìm trọ dành cho sinh viên Đại học FPT'.

3.4. House Detail

- Function trigger: User clicks on one house from Home page
- Function description:
 - Role: Guest, student
 - Purpose: View a house detail information (landlord's information, available rooms, etc.)
- Screen layout:

The screenshot shows a web application interface for finding dormitory rooms. At the top, there is a logo for 'FU House Finder' featuring a magnifying glass over a house icon, and a navigation bar with links for 'Trang chủ', 'Cơ sở', 'Liên hệ', and 'Giới thiệu'. On the right, there is a 'Đăng nhập' button. The main content area displays a large image of a modern apartment interior with a double bed, a white door, and a small map. To the right of the image is a sidebar titled 'Thông tin chủ trọ' (Landlord information) which includes a placeholder profile picture and details for 'Ông: Nguyễn Văn A', 'Facebook:', 'SĐT:', and 'Địa chỉ:'. Below the image, the title 'Hòa Lạc Apartment' is displayed, followed by its address: 'Địa chỉ: Số nhà 123, thôn 3, Tân Xã, Thạch Thất, Hà Nội'. Below the address are four price points: 'Giá điện: 3.500/số', 'Giá nước: 2.000/khối', and 'Phí dịch vụ khác: 150.000/phòng'. At the bottom, there is a section titled 'Danh sách phòng trống' (List of available rooms) with a table showing three room options: 201, 203, and 301.

Phòng	Giá phòng	Loại phòng	Đang ở	Ghi chú
201	2.300.000	Khép kín	👤👤👤	
203	2.300.000	Khép kín	👤👤👤	
301	2.500.000	Chung cư mini	👤👤👤	

3.5. Room Detail

- Function trigger: User clicks on one Room from a House Detail
- Function description:
 - Role: Guest, student
 - Purpose: View a room detail information (images, price, etc.)

- Screen layout:

The screenshot shows a room detail page for a house. At the top, there's a navigation bar with a magnifying glass icon, the text "FU HOUSE FINDER", and a "User" button. Below the navigation bar is a large image of a modern room with a desk, chair, and windows. To the right of the image is a box titled "Thông tin mô tả" containing text about the room's features. Below the image is another box titled "Thông tin chủ trọ" with contact information for the landlord. Further down is a section for reviews ("Người dùng đánh giá") with a comment input field, a rating scale from 1 to 5 stars, and a "Đăng" (Post) button. At the bottom of the page is a footer with the FPT Education logo and the text "FPT EDUCATION GLOBAL".

3.6. Send Report [Student]

- Function trigger: Student clicks “Báo cáo” in one House Detail
- Function description:
 - Role: Student
 - Purpose: Student uses to report an unusual house
- Screen layout:

The screenshot shows a modal dialog box titled "Báo cáo nhà trọ" (Report Room). It contains a text input field labeled "Lý do:" (Reason) and a red "Báo cáo" (Report) button. The background of the page shows the same room detail interface as the previous screenshot.

3.7. Register [Landlord]

- Function trigger: User click the landlord's register button
- Function description:
 - Role: Guest
 - Purpose: Landlord register for an account
- Screen layout:



CHỦ TRỌ ĐĂNG KÝ TÀI KHOẢN

1 Tạo tài khoản 2 Điền thông tin liên hệ 3 Chụp ảnh CMND

Số điện thoại

Link Facebook



CHỦ TRỌ ĐĂNG KÝ TÀI KHOẢN

1 Tạo tài khoản 2 Điền thông tin liên hệ 3 Chụp ảnh CMND

Số điện thoại

Link Facebook





CHỦ TRỌ ĐĂNG KÝ TÀI KHOẢN

1

Tạo tài khoản

2

Điền thông tin liên hệ

3

Chụp ảnh CMND

Kéo thả ảnh ở đây

CMND mặt trước

Kéo thả ảnh ở đây

CMND mặt sau

FPT Education
FPT EDUCATION GLOBAL



CHỦ TRỌ ĐĂNG KÝ TÀI KHOẢN

1

Tạo tài khoản

2

Điền thông tin liên hệ

3

Chụp ảnh CMND



CMND mặt trước

Kéo thả ảnh ở đây

CMND mặt sau

FPT Education
FPT EDUCATION GLOBAL



Đăng ký tài khoản thành công!
Hãy bắt đầu đăng ký thông tin nhà trọ của bạn!



3.8. Dashboard [Landlord]

- Function trigger: Landlord accesses the website
- Function description:
 - Role: Landlord
 - Purpose: Landlord view list of houses and search for houses by name or filter it by category
- Screen layout:

The dashboard screen shows the following layout:

- Header:** FU HOUSE FINDER, Host button.
- Search Bar:** Nhập tên trọ, từ khóa tìm kiếm, magnifying glass icon.
- Filter Sidebar:** LỌC PHÒNG TRỌ
 - Bán kính: 0 - 2km (slider)
 - Khu vực:
 - Huyện: dropdown menu
 - Xã: dropdown menu
 - Thôn: dropdown menu
- Statistics:** Tổng số tòa: 3 tòa, Số phòng trống/Tổng số phòng: 14/500 phòng.
- Thumbnail Preview:** Three small thumbnail cards for "Hòa Lạc Apartment" rooms 100, 101, 102, 103.
- Footer:** FPT Education GLOBAL.

3.9. Upload House Information [Landlord]

- Function trigger: Landlord clicks the import button
- Function description:

- Role: Landlord
- Purpose: Landlord import information of the house
- Screen layout:

FU HOUSE FINDER

Host

Thêm 1 phòng Thêm nhiều phòng

Tên phòng

Giá tiền

Thông tin

Số người ở

Diện tích

Tòa

Tầng

Trạng thái

Ảnh:
Choose file No file choose

Thêm mới

FPT Education

FPT EDUCATION GLOBAL

FU HOUSE FINDER

Host

Thêm 1 phòng Thêm nhiều phòng

Download template

Upload template No file choose

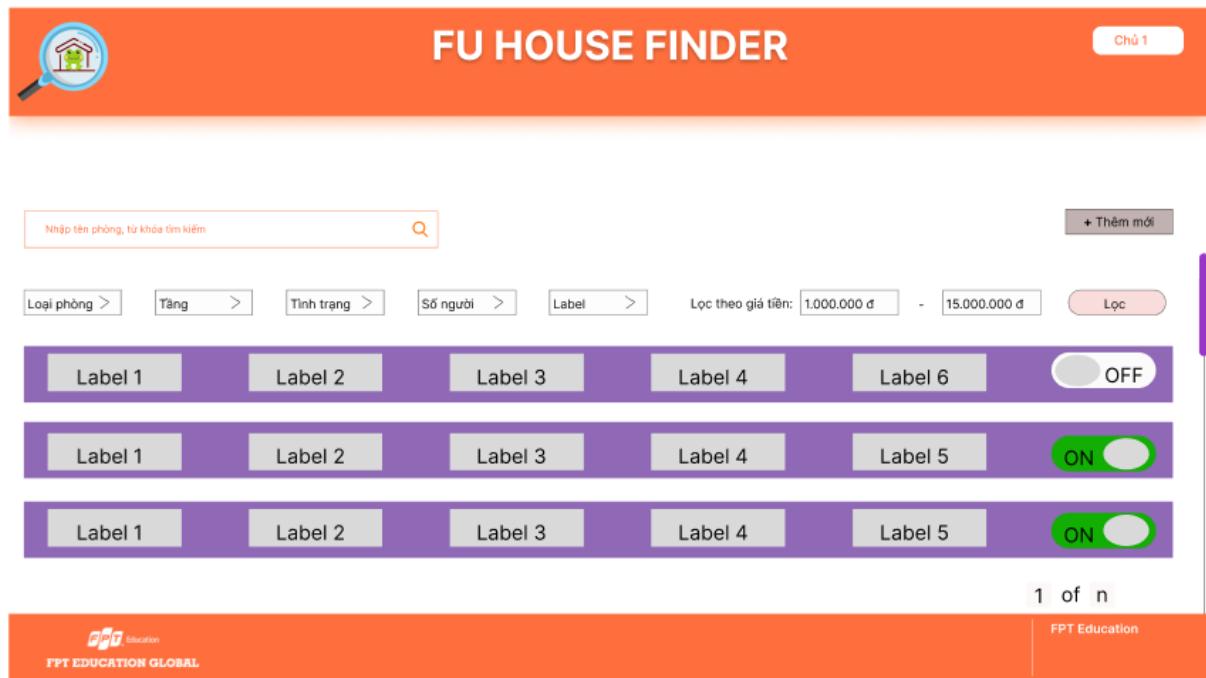
Thêm mới

FPT Education

FPT EDUCATION GLOBAL

3.10. House Detail [Landlord]

- Function trigger: Landlord clicks each item of list houses
- Function description:
 - Role: Landlord
 - Purpose: Display detail information of house and search for rooms by name or filter it by category
- Screen layout:



3.11. Update House [Landlord]

- Function trigger: Landlord clicks the update button
- Function description:
 - Role: Landlord
 - Purpose: Landlord update the house
- Screen layout:

3.12. Delete House [Landlord]

- Function trigger: Landlord clicks the delete button
- Function description:
 - Role: Landlord

- Purpose: Landlord delete the house
- Screen layout:



3.13. Update Room [Landlord]

- Function trigger: Landlord clicks the update button of each room in the list
- Function description:
 - Role: Landlord
 - Purpose: Landlord update the room
- Screen layout:

Tên phòng	Diện tích
Giá tiền	Tòa
Thông tin	Tầng
Số người ở	Trạng thái

Cập nhật



3.14. Add Room [Landlord]

- Function trigger: Landlord clicks the add button

- Function description:
- Role: Landlord
- Purpose: Landlord add the new room
- Screen layout:



The screenshot shows the 'FU HOUSE FINDER' application interface. At the top, there is a logo of a magnifying glass over a house icon and the text 'FU HOUSE FINDER'. On the right side, there is a small button labeled 'Host'. Below the header, there is a form with the following fields:

Tên phòng	Diện tích
Giá tiền	Tòa
Thông tin	Tầng
Số người ở	Trạng thái

Below the form, there is a file upload input field with the placeholder 'Choose file No file choose' and a button labeled 'Thêm mới'.

3.15. Delete Room [Landlord]

- Function trigger: Landlord clicks the delete button of each room in the list
- Function description:
 - Role: Landlord
 - Purpose: Landlord delete the room
- Screen layout:

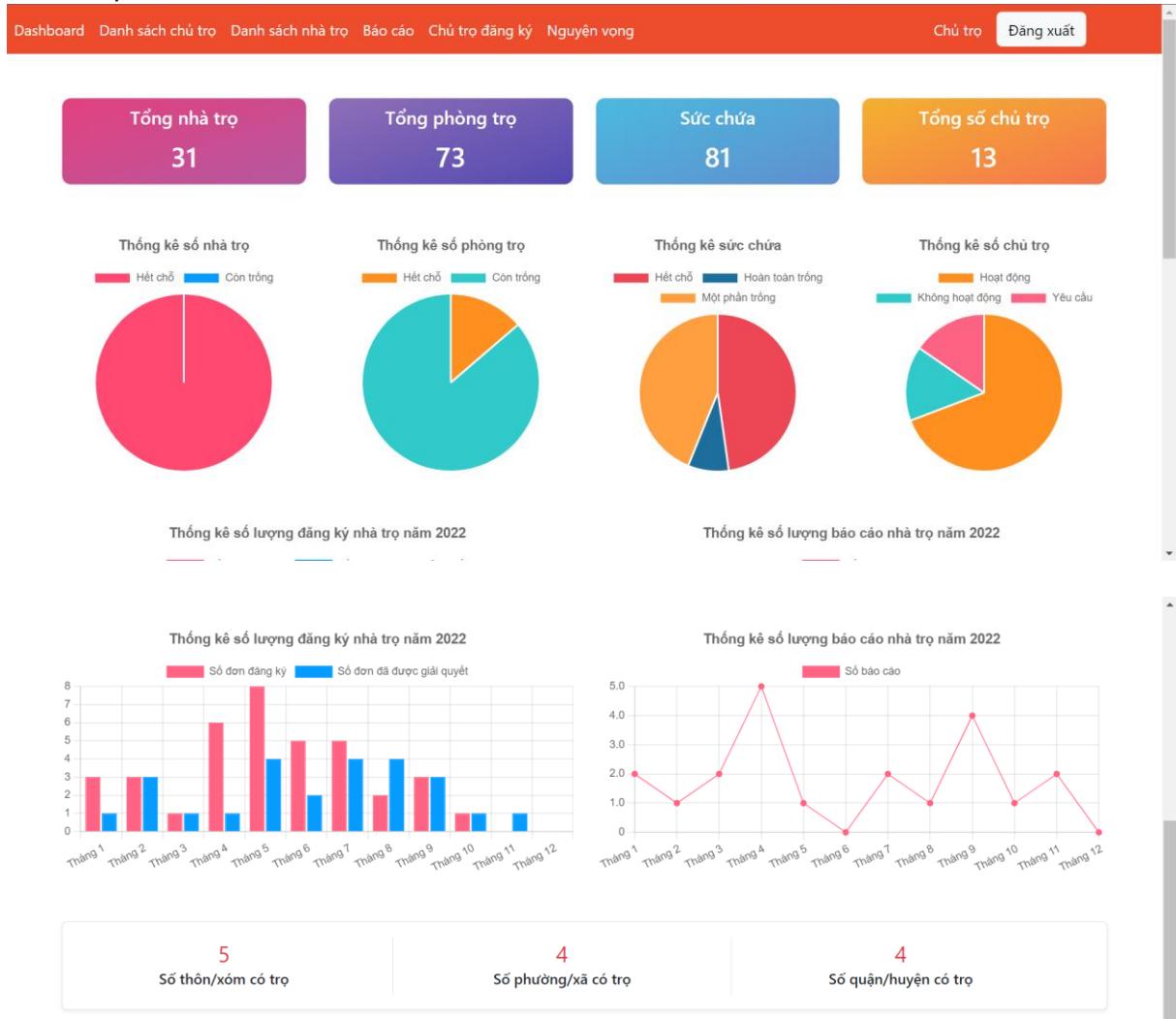


The screenshot shows a confirmation dialog box titled 'Xác nhận xóa phòng' (Confirm deletion of room). The dialog contains two buttons: 'Xóa' (Delete) on the left and 'Hủy' (Cancel) on the right.



3.16. Dashboard [Staff]

- Function trigger: Staff accesses the website
- Function description:
 - Role: Staff
 - Purpose: Staff view houses and rooms statistics
- Screen layout:



3.17. List Of Landlords [Staff]

- Function trigger: Staff clicks the list landlords button in the navbar
- Function description:
 - Role: Staff
 - Purpose: Staff view houses statistics, list of landlords and filter the results
- Screen layout:



Tổng nhà	Nhà trống	Tổng phòng	Phòng trống	Tổng chỗ	Chỗ trống
31	12	73	63	155	81

Danh sách chủ trọ

Nhập từ khóa tìm kiếm

STT	Id	Họ Tên	Số Điện Thoại	Facebook	Tổng nhà	Tổng phòng	Phòng trống	Trạng thái	Tùy chọn
1	LA000001	Tâm Lê	0987654321	Tâm Lê	4	16	11	<input checked="" type="checkbox"/> Hoạt động	i
2	LA000002	Dũng Nhung	0982298681	Dũng Nhung	4	10	9	<input checked="" type="checkbox"/> Hoạt động	i
3	LA000003	Nga Hương	0984530814	Nga Hương	7	13	9	<input checked="" type="checkbox"/> Hoạt động	i
4	LA000004	Bình Yên	0973866690	Bình Yên	4	6	6	<input checked="" type="checkbox"/> Hoạt động	i
5	LA000005	Tiến Phương	0961602245	Tiến Phương	5	7	7	<input checked="" type="checkbox"/> Hoạt động	i
6	LA000006	Ngọc Hương	0981914814	Ngọc Hương	1	3	3	<input checked="" type="checkbox"/> Hoạt động	i

10	LA000010	Thái Hà	0961602245	Thái Hà	1	3	3	<input checked="" type="checkbox"/> Hoạt động	i
11	LA000011	Viet Dũng	0363266546	Viet Dũng	1	3	3	<input checked="" type="checkbox"/> Hoạt động	i
12	LA000012	Tuấn Cường	0363266546	Tuấn Cường	1	3	3	<input checked="" type="checkbox"/> Hoạt động	i
13	LA000013	Phương Duy	0365928073	Phương Duy	0	0	0	<input checked="" type="checkbox"/> Hoạt động	i

« 1 2 3 »

3.18. Landlord Detail [Staff]

- Function trigger: Staff clicks the detail button of each landlord in the list
- Function description:
 - Role: Staff
 - Purpose: Staff view houses statistics, detail information of landlord, list houses of this landlord and filter the results
- Screen layout:

Dashboard Danh sách chủ trọ Danh sách nhà trọ Báo cáo Chủ trọ đăng ký Nguyên vọng

Chủ trọ Đăng xuất

Tổng nhà	Nhà trống	Tổng phòng	Phòng trống	Tổng chỗ	Chỗ trống
 31	12	73	63	155	81
	Họ và tên: Tâm Lê Facebook: facebook.com/tamle12 Số điện thoại: 0987654321 Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế				Tổng số nhà trọ: 4 nhà Tổng số phòng: 16 phòng Tổng số phòng trống: 11 phòng

Danh sách nhà trọ

Nhập từ khóa tìm kiếm

STT	ID	Tên nhà trọ	Giá điện	Giá nước	Giới tính	Số phòng	Loại phòng	Tiện ích	Tùy chọn
1	1	Trọ Tâm Lê	đ3,700	đ1,200	1	3	5	∅	
2	13	Trọ Linh Lê	đ3,700	đ1,200	0	3	3	∅	
3	14	Trọ Tâm Nguyễn	đ3,700	đ1,200	0	3	3	□	

STT	ID	Tên nhà trọ	Giá điện	Giá nước	Giới tính	Số phòng	Loại phòng	Tiện ích	Tùy chọn
1	1	Trọ Tâm Lê	đ3,700	đ1,200	1	3	5	∅	
2	13	Trọ Linh Lê	đ3,700	đ1,200	0	3	3	∅	
3	14	Trọ Tâm Nguyễn	đ3,700	đ1,200	0	3	3	□	
4	20	Trọ Campus 1	đ3,400	đ1,500	0	1	1	P	

« 1 2 3 »



FPT Education
Hoa Lạc High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

 
Sitemap website
Copyright © 2017 FPT Education

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyên đất & người
FPT Edu - Tin tức chung
Bàn chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế
Ngân hạn
Dài hạn

3.19. House Detail [Staff]

- Function trigger: Staff clicks the detail button of each house in the list
- Function description:
 - Role: Staff
 - Purpose: Staff view houses statistics, detail information of house, list rooms of this house and filter the results
- Screen layout:

Dashboard						Chủ trọ	Đang xuất
Tổng nhà	Nhà trống	Tổng phòng	Phòng trống	Tổng chỗ	Chỗ trống		
31	12	73	63	155	81		



Tên nhà trọ: Trọ Tâm Lê
Giá điện: 3700/kWh
Giá nước: 1200/m3
Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế
Thông tin khác: Rất đẹp

Tổng số phòng trống: 9 phòng
Tổng số chỗ trống: 0 phòng

Danh sách phòng

Nhập từ khóa tìm kiếm 

STT	Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn
1	101	đ3,000,000		20 m ²		Còn Trống	
2	102	đ3,000,000		15 m ²			
6	203	đ2,000,000		25 m ²		Đã thuê	
7	301	đ3,000,000		20 m ²		Còn Trống	
8	302	đ2,500,000		15 m ²		Còn Trống	
9	303	đ2,000,000		25 m ²		Đã thuê	

« 1 2 3 »



FPT Education
Hoa Lạc High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

Sitemap website
Copyright © 2017 FPT Education

Tin tức

Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyên đất & người
FPT Edu - Tin tức chung
Bàn chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh

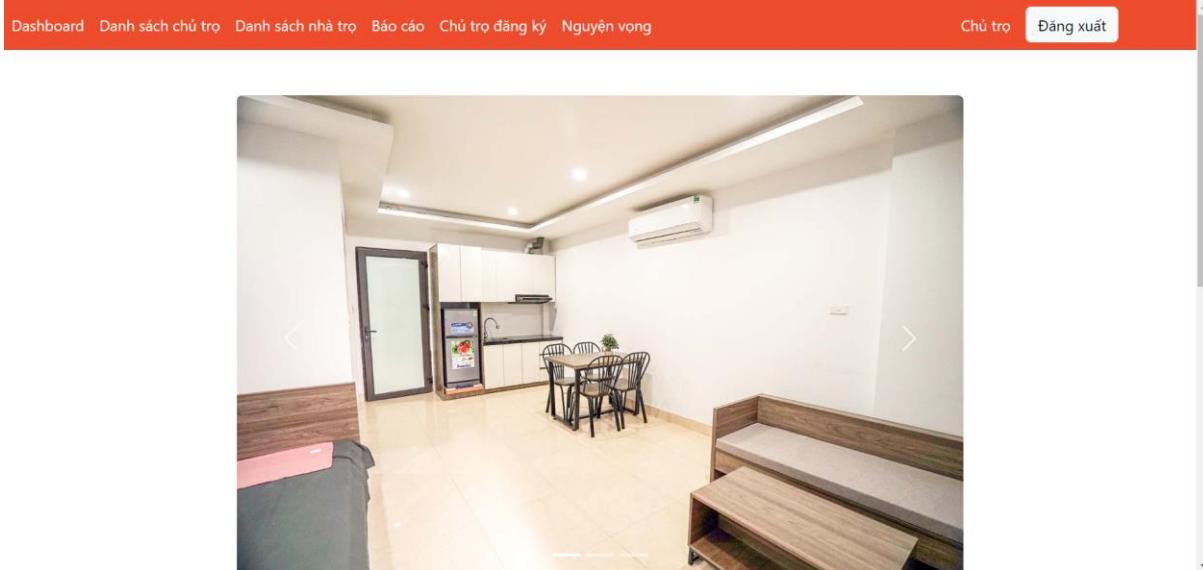
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế

Ngắn hạn
Dài hạn

3.20. Room Detail [Staff]

- Function trigger: Staff clicks the detail button of each room in the list
- Function description:
 - Role: Staff
 - Purpose: Staff view detail information of this room
- Screen layout:



Thông tin mô tả

<input checked="" type="checkbox"/> Tủ lạnh: Có	<input checked="" type="checkbox"/> Bếp: Có
<input checked="" type="checkbox"/> Máy giặt: Không	<input type="checkbox"/> Bàn học: Không
<input checked="" type="checkbox"/> Vệ sinh khép kín: Có	<input checked="" type="checkbox"/> Giường: Có
<input checked="" type="checkbox"/> Không chung chủ: Có	
Gạch sàn nhà có họa tiết hình con cá	



FPT Education
Hoa Lac High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

[Sitemap website](#)
Copyright © 2017 FPT Education

Tin tức

- Học tập khác biệt
- Môi trường giàu trải nghiệm
- Đa dạng màu sắc quốc tế
- Thành công sau tốt nghiệp
- FPT Edu - Chuyên đất & người
- FPT Edu - Tin tức chung
- Bàn chuyên giáo dục
- Phục vụ cộng đồng

Tuyển sinh

- Đại học FPT
- Greenwich Việt Nam
- Swinburne Việt Nam
- Cao đẳng FPT Polytechnic
- Viện Đào tạo Quốc tế FPT
- Melbourne Polytechnic Việt Nam

Sinh viên quốc tế

- Ngân hạn
- Dài hạn

3.21. List Of Houses [Staff]

- Function trigger: Staff clicks the list houses button in the navbar
- Function description:
 - Role: Staff
 - Purpose: Staff view houses statistics, list of houses and filter the results
- Screen layout:

Dashboard									Danh sách chủ trọ	Danh sách nhà trọ	Báo cáo	Chủ trọ đăng ký	Nguyễn vong	Chủ trọ	Đang xuất
31		12		73		63		155		81					
Tổng nhà	Nhà trống	Tổng phòng	Phòng trống	Tổng chỗ	Chỗ trống										
Danh sách nhà trọ															
Nhập từ khóa tìm kiếm															
STT	ID	Tên nhà trọ	Giá điện	Giá nước	Giường	Giường	WC	WC	Tiện ích	Tùy chọn					
1	5	Nhà trọ Tiên Phương	đ3,600	đ1,200	0	3	3	3	<input type="checkbox"/>	i					
2	6	Nhà trọ Phương Duy	đ3,800	đ1,300	0	3	3	3	<input type="checkbox"/>	i					
3	7	HOLA Campus	đ3,200	đ1,500	0	3	3	3	<input type="checkbox"/> P	i					
4	8	Trọ Hoàng Nam	đ3,400	đ1,250	0	3	3	3	<input type="circle"/>	i					
5	9	Kí túc xá Ông bà	đ3,600	đ1,400	0	3	3	3	<input type="circle"/>	i					
6	10	Nhà trọ Thái Hà	đ3,500	đ1,500	0	3	3	3	P	i					
9	24	Trọ Campus 5	đ3,400	đ1,500	0	1	1	1	P	i					
10	27	Trọ RoomType 3	đ3,400	đ1,500	0	1	1	1	P	i					
11	30	Trọ District 3	đ3,400	đ1,500	0	1	1	1	P	i					
12	31	Trọ Hồ Chí Minh	đ3,400	đ1,500	0	1	1	1	P	i					

 <p>FPT Education FPT EDUCATION GLOBAL</p>	FPT Education Hoa Lac High Tech Park, Hanoi, vietnam Email: fpteducation@fe.edu.vn	Tin tức Học tập khác biệt Môi trường giàu trải nghiệm Đa dạng màu sắc quốc tế Thành công sau tốt nghiệp FPT Edu - Chuyên đất & người FPT Edu - Tin tức chung Bàn chuyên giáo dục Phục vụ cộng đồng	Tuyển sinh Đại học FPT Greenwich Việt Nam Swinburne Việt Nam Cao đẳng FPT Polytechnic Viện Đào tạo Quốc tế FPT Melbourne Polytechnic Việt Nam
	  Sitemap website Copyright © 2017 FPT Education		Sinh viên quốc tế Ngắn hạn Dài hạn

3.22. List Of Reports [Staff]

- Function trigger: Staff clicks the list reports button in the navbar
- Function description:
 - Role: Staff
 - Purpose: Staff view list of reports, report of each house and filter the results
- Screen layout:

Danh sách báo cáo					
Nhập từ khóa tìm kiếm					
Từ ngày		Đến ngày		Sắp xếp	
11/09/2022		11/07/2022		Chọn sắp xếp	
Danh sách báo cáo					
STT	Tên sinh viên	Nhà trọ	Ngày báo cáo	Nội dung	
1	Nguyen The Giang	Trọ Tâm Lê	01/01/0001	Chủ trọ tăng giá phòng trái với hợp đồng	
2	Nguyen The Giang	Trọ Hòa Lạc Yên Lạc Viên	01/01/0001	Chủ trọ tự ý vào phòng của bạn và tháo bóng đèn trong nhà vệ sinh của bạn	
3	Nguyen The	Nhà trọ Bình Yên	01/01/0001	Chủ trọ tăng giá phòng trái với hợp đồng	

Danh sách nhà bị báo cáo					
Trang thái		Sắp xếp			
STT	Nhà trọ	Chủ trọ	Số lượt báo cáo	Trạng thái chủ trọ	Tùy chọn
1	Trọ Tâm Lê	Tâm Lê	5	<input checked="" type="checkbox"/>	i
2	Trọ Tâm Thảo	Dũng Nhung	1	<input checked="" type="checkbox"/>	i
3	Trọ Hòa Lạc Yên Lạc Viên	Nga Hương	2	<input checked="" type="checkbox"/>	i
4	Nhà trọ Bình Yên	Bình Yên	2	<input checked="" type="checkbox"/>	i

Thông tin chi tiết					
Thông tin chủ trọ					
Tên chủ trọ: Tâm Lê Số điện thoại: 0987654321 Link facebook: facebook.com/tamle12					
Danh sách báo cáo					
STT	Mã sinh viên	Tên sinh viên	Nội dung	Ngày báo cáo	
1	HE153046	Nguyen The Giang	Chủ trọ tăng giá phòng trái với hợp đồng	01/01/0001	
2	HE153046	Nguyen The Giang	Chủ trọ xâm phạm quyền riêng tư	01/01/0001	
3	HE153046	Nguyen The Giang	Chủ trọ tăng giá phòng trái với hợp đồng	01/01/0001	

3.21. List Of Landlords Signup Request [Staff]

- Function trigger: Staff clicks the list requests button in the navbar
- Function description:
 - Role: Staff
 - Purpose: Staff view list of request and filter the results
- Screen layout:

3.22. List Of Orders [Staff]

- Function trigger: Staff clicks the list orders button in the navbar
- Function description:
 - Role: Staff
 - Purpose: Staff view list of orders, detail of order and filter the results
- Screen layout:

STT	Mã sinh viên	Số điện thoại	Email	Ngày tạo	Ngày giải quyết	Trạng thái	Tùy chọn
1	HE150160	0346034217	huyenbnhe150346@fpt.edu.vn	03/05/2022	10/07/2022	Đã giải quyết	(i)
2	HE150432	0346034217	huyenbnhe150346@fpt.edu.vn	03/01/2022	10/01/2022	Đã giải quyết	(i)
3	HE150600	0346034217	huyenbnhe150346@fpt.edu.vn	03/06/2022	10/07/2022	Đã giải quyết	(i)
4	HE153222	0346034217	huyenbnhe150346@fpt.edu.vn	03/08/2022	18/11/2022	Đã giải quyết	(i)
5	HE150600	0346034217	huyenbnhe150346@fpt.edu.vn	03/01/2022	10/02/2022	Đã giải quyết	(i)

STT	Mã sinh viên	Số điện thoại	Email	Ngày tạo	Ngày giải quyết	Trạng thái	Tùy chọn
1	HE150160	0346034217	huyenbnhe150346@fpt.edu.vn	03/05/2022	10/07/2022	Đã giải quyết	(i)
2	HE150432	0346034217	huyenbnhe150346@fpt.edu.vn	03/01/2022	10/01/2022	Đã giải quyết	(i)
3	HE150600	0346034217	huyenbnhe150346@fpt.edu.vn	03/06/2022	10/07/2022	Đã giải quyết	(i)
4	HE153222	0346034217	huyenbnhe150346@fpt.edu.vn	03/08/2022	18/11/2022	Đã giải quyết	(i)
5	HE150600	0346034217	huyenbnhe150346@fpt.edu.vn	03/01/2022	10/02/2022	Đã giải quyết	(i)

7	HE153046	0346034217	huyenbnhe150346@fpt.edu.vn	03/02/2022	10/02/2022	Chưa giải quyết	
8	HE150340	0346034217	huyenbnhe150346@fpt.edu.vn	03/02/2022	10/02/2022	Đã giải quyết	
9	HE150160	0346034217	huyenbnhe150346@fpt.edu.vn	03/02/2022	10/03/2022	Đã giải quyết	
10	HE153299	0346034217	huyenbnhe150346@fpt.edu.vn	31/03/2022		Chưa giải quyết	

« 1 2 3 4 »

FPT Education
Hoa Lac High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyên đất & người
FPT Edu - Tin tức chung
Bản chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế
Ngắn hạn
Dài hạn

3.22. List of Accounts [Admin]

- Function trigger: User logged in as Admin
- Function description:
 - Role: Admin
 - Purpose: Admin view list of Accounts
- Screen layout:

STT	Họ và tên	Email	Vị trí	Hành động
1	Lê Thành	thanhle@gmail.com	Trưởng phòng tuyển sinh	
2	Staff 1	staff1@gmail.com	Cán bộ phòng tuyển sinh	
3	Staff 2	staff2@gmail.com	Cán bộ phòng tuyển sinh	
4	Staff 3	staff3@gmail.com	Nhân viên phòng phục vụ sinh viên	

3.23. Add Account [Admin]

- Function trigger: User clicks on “Thêm mới” button
- Function description:
 - Role: Admin
 - Purpose: Admin add new Account
- Screen layout:

TÌM TRỌNG VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Admin

Tạo tài khoản tuyển sinh

Họ và tên
Nhập họ và tên

Email
Nhập email

Mật khẩu
Nhập mật khẩu

Vị trí

Tạo mới

FPT Education
Hoa Lạc High Tech Park, Hanoi, Vietnam
Email: fpteducation@fe.edu.vn

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyển đổi & người
FPT Edu - Tin tức chung
Bản chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế
Ngân hàng
Dài hạn

3.24. Update Account [Admin]

- Function trigger: User clicks on Update button
- Function description:
 - Role: Admin
 - Purpose: Admin update existing Account
- Screen layout:

TÌM TRỌNG VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Admin

Cập nhật tuyển sinh

Họ và tên
Lê Thành

Email
lthanhle@gmail.com

Vị trí
Head of Admission Department

Thay đổi

FPT Education
Hoa Lạc High Tech Park, Hanoi, Vietnam
Email: fpteducation@fe.edu.vn

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyển đổi & người
FPT Edu - Tin tức chung
Bản chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế
Ngân hàng
Dài hạn

3.25. Delete Account [Admin]

- Function trigger: User clicks on Delete button
- Function description:
 - Role: Admin
 - Purpose: Admin delete existing Account

- Screen layout:

The screenshot shows a list of staff members with their names and emails. A modal dialog box is centered over the list, displaying a large exclamation mark icon and the text "Xóa tài khoản nhân viên". Below this, a question asks if the user wants to delete the account. Two buttons, "OK" and "Cancel", are at the bottom of the dialog.

STT	Họ và tên	Email	Hành động
1	Lê Thành	thanhle@gmail.com	
2	Staff 1	staff1@gmail.com	
3	Staff 2	staff2@gmail.com	
4	Staff 3	staff3@gmail.com	

At the bottom of the page, there is a footer with links to FPT Education, Sitemap, Copyright information, and various university links like Hoa Lac High Tech Park, Swinburne Vietnam, and Melbourne Polytechnic.

3.26. Create Order [Student]

- Function trigger: User clicks on “Đăng ký yêu cầu phòng trọ” on Home Page
- Function description:
 - Role: Student
 - Purpose: Send an Order for House
- Screen layout:

The screenshot shows a modal dialog titled "Đăng ký yêu cầu phòng trọ". It contains fields for "Tên" (Name), "Số điện thoại" (Phone number), "Email", and "Nguyên vọng" (Desires). At the bottom right of the dialog are two buttons: "Hủy" (Cancel) and "Gửi yêu cầu" (Send request).

On the left side of the main page, there is a sidebar with filtering options for location ("LỌC KẾT QUẢ"), price ("Giá tiền"), and room type ("Loại phòng").

The main content area displays a grid of accommodation listings. Each listing includes a thumbnail image, the name of the place (e.g., Trọ Tâm Nguyễn, Trọ Linh Lê, Trọ Tâm Lê), a distance from the user (e.g., 0.1 km, 0.9 km, 4.5 km), price range (e.g., ₫2,900,000 - ₫2,900,000, ₫2,750,000 - ₫2,750,000), address (e.g., Nhà số..., Đường... Đồi điện công sau Đại học F...), ratings (e.g., 3/5, 4/5, 4.5/5), and a "Đăng ký yêu cầu phòng trọ" button.

4. Non-Functional Requirements

4.1. External Interfaces

4.1.1 User Interfaces

- UI-1: All of the FU House Finder System's screens will be responsive to display information clearly and well organized in any screen size.
- UI-2: The website is designed with the feature of using Angular framework to provide a smooth user experience without having to reload the website many times.

4.1.2 Software Interfaces

- SI-1: FHF Front-end sends requests to Facebook API to perform authentication.
- SI-2: FHF Front-end sends requests to Google Cloud API to perform authentication, FHF Back-end verifies authentication token. FHF Front-end also sends request to use Google Map APIs (Distance Matrix API and Maps JavaScript API).
- SI-3: FHF Front-end sends a request to Amazon S3 as Image Server to retrieve images.
- SI-4: FHF Back-end sends a request to Amazon S3 as Image Server to store images and get image URL.

4.1.3 Hardware Interfaces

No hardware interfaces have been identified.

4.1.4 Communication Interfaces

- CI-1: Connection between FHF Front-end and FHF Back-end will be HTTPS for security.
- CI-2: Connection between External Interfaces to FHF System will be HTTPS for security.
- CI-3: Connection between FHF Back-end and Amazon RDS Database will be TCP.

4.2. Quality Attributes

4.2.1 Availability

- AVL-1: The FU House Finder website shall be available at least 98% of the time between 5:00 A.M. and midnight local time and at least 90% of the time between midnight and 5:00 A.M. local time, excluding scheduled maintenance windows.

4.2.2 Usability

- USB-1: The maximum number of mouse click any user role has to perform to get to any place they want in the system is 4.

- USB-2: The theme color of the website is Orange, following the theme color of other websites in the FPT Education system.

- USB-3: All ambiguous icons will have explanatory tooltips to help user understand.

- USB-4: Organization of information in a screen does not become disordered in different screen sizes.

- USB-5: Font size on all pages of the system is always at least 11pt when displayed on any screen size.

- USB-6: All components will always have at least 10px distance from each other.

USB-7: There are no use of slang, acronyms and abbreviations.

USB-8: Actions that require choosing between options will have no more than 7 choices, following Hick's Law.

USB-9: All components that belongs to one group will be styled with different color or with border from others, following the Law of Common Region.

USB-10: All components that belongs to one group will have similar size and shape with each other, following the Law of Similarity.

USB-11: Actions that require multiple steps will have no more than 5 steps to ensure that user can remember them all, following the Miller's Law.

4.2.3 Localization

LCL-1: The date format must be as follows: DD/MM/YYYY.

4.2.4 Performance

PE-1: The website must provide 60 seconds or less respond time in a Chrome browser in peak usage condition.

PE-2: Any pages in the system shall fully load in at maximum of 30 seconds in normal condition.

4.2.5 Security

SE-1: Only admin shall be able to create a new staff's account and only staff shall be able to approve/reject landlords' signup request.

SE-2: Landlords must provide their identity card image to be able to sign up a landlord account.

SE-3: Pages of authenticated roles will have authorization for extra protection.

5. Requirement Appendix

5.1. Business Rules

ID	Rule Definition
BR-01	Only Staff can approve Landlord's Sign up request and Deactive Landlord's account.
BR-02	Only Admin can create, modify, or deactivate Staff's account.
BR-03	Landlord can only upload House information after sign up request being approved by Staff.
BR-04	All Passwords require 256-bit encryption.
BR-05	Student's Search functionality requires at least searching by distance from campus, price and utility.
BR-06	Landlord's House information upload must adhere to the rules of the template

Table 3-7: Business rules

5.2. Application Messages List

#	Message code	Message Type	Context	Content
1	MSG01	In the body	There is not any search result	Rất tiếc, chúng tôi không tìm thấy nhà trọ bạn mong muốn
2	MSG02	Toast message	User not logged into the system	Vui lòng đăng nhập để sử dụng tính năng này!
3	MSG03	Toast message	Only student can use this feature	Chỉ học sinh có thể sử dụng tính năng này!
4	MSG04	Toast message	Send order successfully	Gửi đơn thành công
5	MSG05	Toast message	Send report successfully	Gửi báo cáo thành công
6	MSG06	Toast message	Comment successfully	Bình luận thành công
7	MSG07	Toast message	Server error	Lỗi Server
8	MSG08	Toast message	Ask if you want more rooms?	Bạn có muốn thêm phòng cho nhà trọ?
9	MSG09	Toast message	Upload room information successfully	Bạn đã upload thành công thông tin nhà trọ!
10	MSG10	Toast message	Confirm deletion of house	Bạn chắc chắn muốn xóa nhà trọ này?
11	MSG11	Toast message	Update successfully	Cập nhật thành công
12	MSG12	Toast message	Error	Lỗi! Thử lại sau vài giây!
13	MSG13	Toast message	Update room successfully	Cập nhật phòng thành công!
14	MSG14	Toast message	Add room successfully	Thêm mới phòng thành công!
15	MSG15	Toast message	Delete room successfully	Xóa phòng thành công!
16	MSG16	Toast message	Invalid login	Email hoặc Password không hợp lệ
17	MSG17	Toast message	Confirm deletion of account	Bạn có muốn xóa tài khoản này?

Table 3-8: Application Messages list

IV. Software Design Description

1. System Design

1.1. System Architecture

1.1.1 Diagram

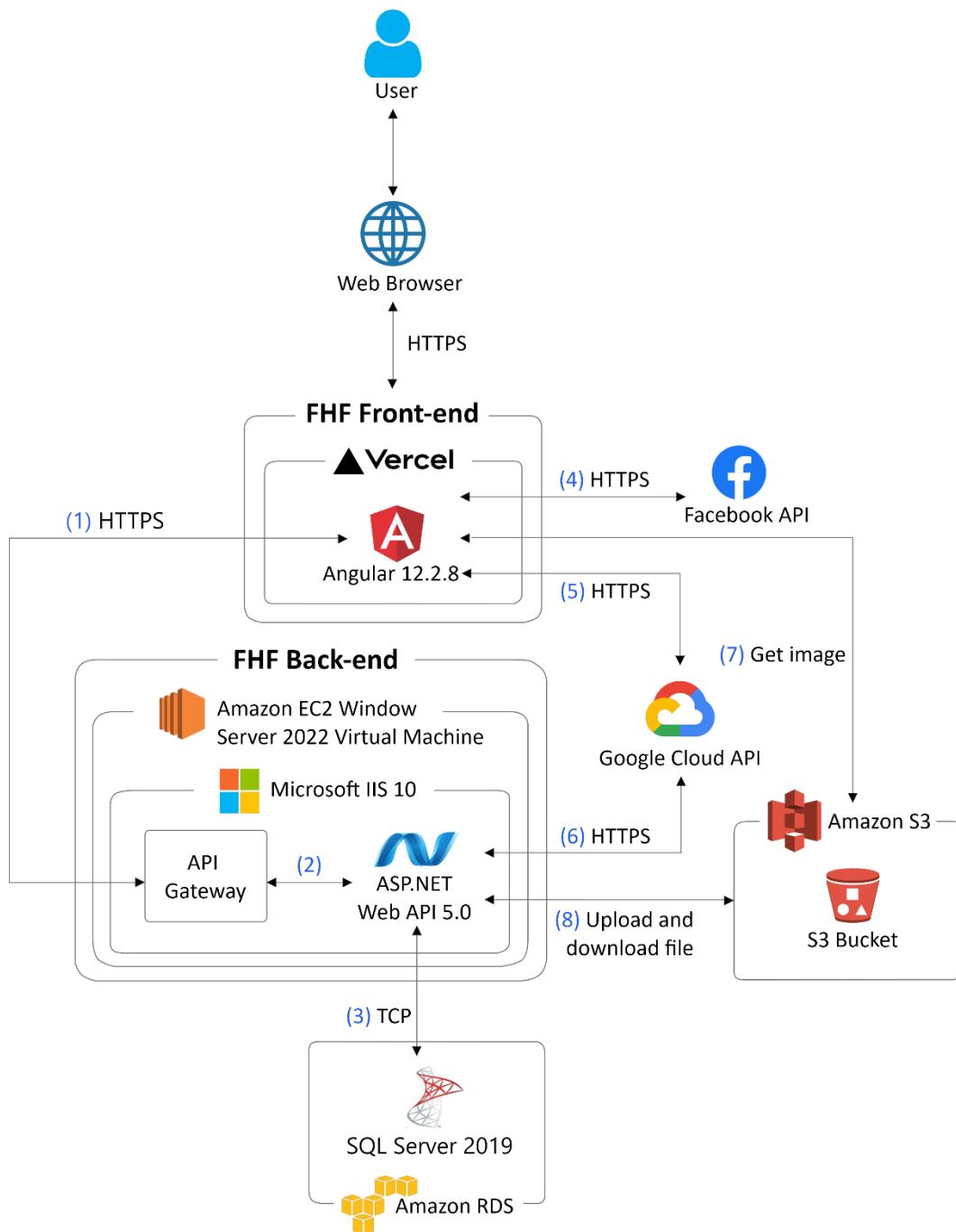


Figure 4-1: System Architecture Diagram

1.1.2 Explanation

- (1), (2) FHF Front-end sends a request to FHF Back-end through API Gateway.
- (3) FHF Back-end sends a request to Database.
- (4) FHF Front-end sends a request to Facebook API to perform authentication.
- (5), (6) FHF Front-end sends a request to Google Cloud API to perform authentication, FHF Back-end verifies authentication token. FHF Front-end also sends request to use Google Map APIs (Distance Matrix API and Maps JavaScript API)
- (7) FHF Front-end sends a request to Image Server to retrieve images
- (8) FHF Back-end sends a request to Image Server to store images and get image URL

1.2. System Architecture Explanation

The FHF Back-end system will be deployed on Amazon EC2 Linux Virtual Machine while FHF Front-end will be deployed on Vercel. We aim to deliver a secured and responsive system with high usability. In the following section, we will explain the function and mechanism of each unit in the system architecture design.

1.2.1 Angular



Figure 4-2: Angular

Angular is a platform and framework for building single-page client applications using HTML and TypeScript. Angular is written in TypeScript. It implements core and optional functionality as a set of TypeScript libraries that you import into your applications. **We use it to build entire website view instead of plain HTML/JavaScript.**

1.2.2 Facebook API



Figure 4-2: Angular

Facebook API is a bundle of solutions used as a primary way to get data in and out of the platform. It enables developers and app users to access the functionality of this network: user information, photos and videos, messages and more. **We use it to perform authentication by Facebook.**

1.2.3 Google Cloud API



Figure 4-3: Google Cloud API

Google Cloud APIs are programmatic interfaces to Google Cloud Platform services. They are a key part of Google Cloud Platform, allowing you to easily add the power of everything from computing to networking to storage to machine-learning-based data analysis to your applications. **We use it to perform authentication by Google and to use Google Map APIs.**

1.2.4 Microsoft IIS

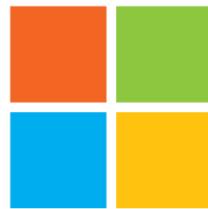


Figure 4-4: Microsoft IIS

Internet Information Services (IIS) for Windows Server is a flexible, secure and manageable Web server for hosting anything on the Web. From media streaming to web applications, IIS's scalable and open architecture is ready to handle the most demanding tasks. **We use it to host our API website.**

1.2.5 SQL Server



Figure 4-5: SQL Server

Microsoft SQL Server is a relational database management system developed by Microsoft. **We use it to manage FHF operational data.**

1.2.6 Amazon S3



Figure 4-6: Amazon S3

Amazon S3 is cloud object storage with industry-leading scalability, data availability, security, and performance. **We use it as a Server for Image Storage.**

1.2. Package Diagram

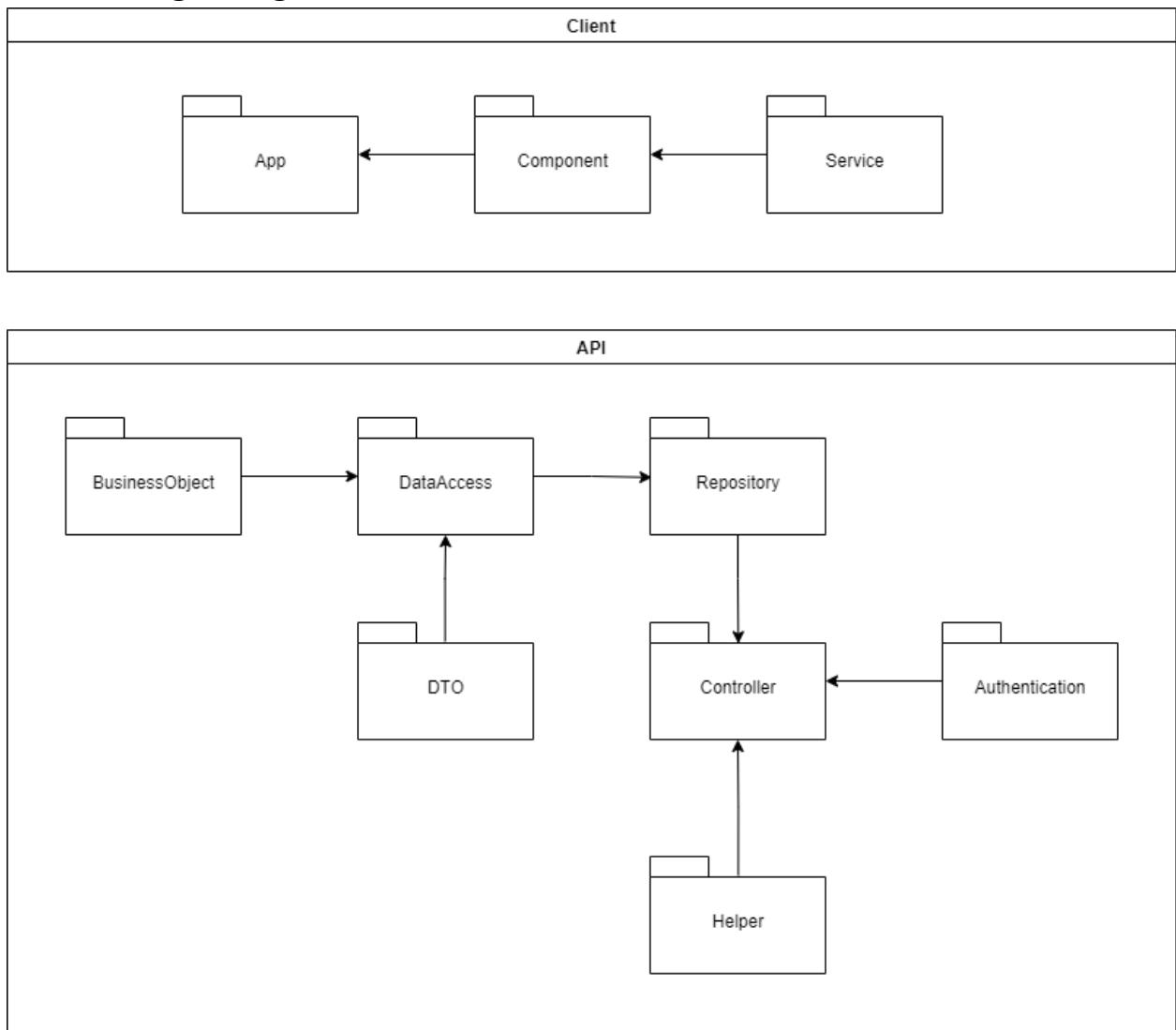


Figure 4-7: Package Diagram

Package Descriptions

No	Package	Description
01	BusinessObject	Define the logic of data extracted from Database
02	DataAccess	Provide the connection to the Database; Extract data for the system
03	Repository	Create an abstraction layer between the data access layer and the business logic layer of an application
04	DTO	Model with predefined validation in place for HTTP responses and requests
05	Authentication	Provide JWT Authentication methods for using in Controller
06	Helper	Provide helping methods for using in Controller
05	Controller	The controller takes the result of the model's processing (if any) and returns either the proper view and its associated view data or the result of the API call
06	App	Root component
07	Component	Components are the main building block for Angular applications. Each component consists of: An HTML template that declares what renders on the page. A TypeScript class that defines behavior
08	Service	Define code or functionalities that are then accessible and reusable in many other components in Angular project

Table 4-1: Package Description

2. Database Design

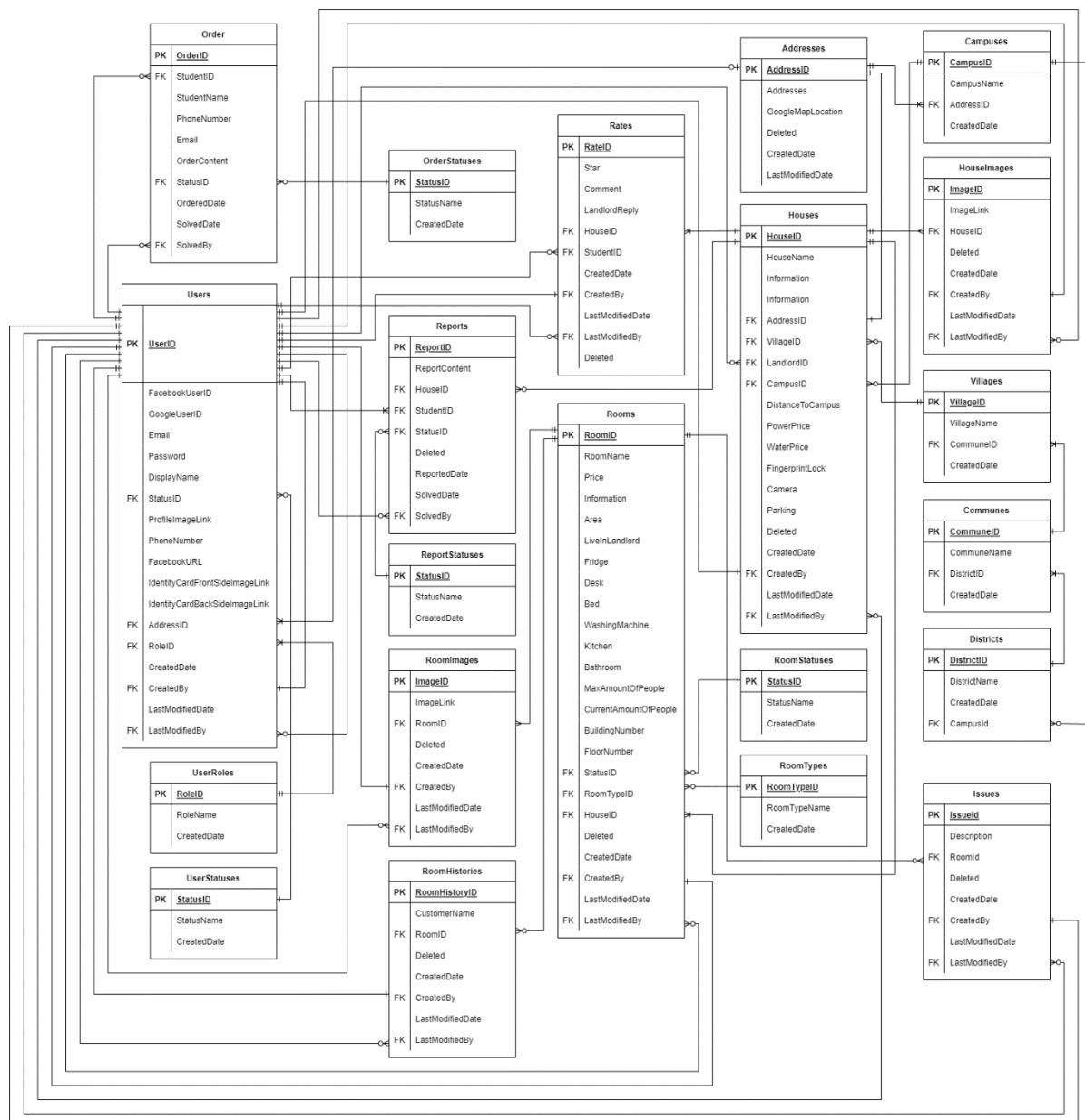


Figure 4-8: Database Diagram

Table Descriptions

No	Table	Description
01	Addresses	<p>Store the information of addresses for Users (Landlords) and Houses</p> <ul style="list-style-type: none"> Primary key: AddressId
02	Campuses	<p>Store the information of all campus for users</p> <ul style="list-style-type: none"> Primary key: CampusId Foreign keys: AddressId
03	Communes	<p>Store the information of all communes (for identifying house address)</p> <ul style="list-style-type: none"> Primary key: CommuneId Foreign keys: DistrictId

04	<i>Districts</i>	<i>Store the information of all districts (for identifying house address)</i> <ul style="list-style-type: none"> – Primary key: DistrictId – Foreign keys: CampusId
05	<i>Houses</i>	<i>Store the information of all houses</i> <ul style="list-style-type: none"> – Primary key: HouseId – Foreign keys: AddressId, LandlordId, VillageId, CampusId, CreatedBy, LastModifiedBy
06	<i>ImagesOfHouse</i>	<i>Store the information of all pictures of house</i> <ul style="list-style-type: none"> – Primary key: ImageId – Foreign keys: HouseId, CreatedBy, LastModifiedBy
07	<i>ImagesOfRoom</i>	<i>Store the information of all pictures of room</i> <ul style="list-style-type: none"> – Primary key: ImageId – Foreign keys: RoomId, CreatedBy, LastModifiedBy
08	<i>Issues</i>	<i>Store the information of all records of Issues usage for Landlords to manage if they want to</i> <ul style="list-style-type: none"> – Primary key: IssueId – Foreign keys: RoomId, CreatedBy, LastModifiedBy
09	<i>Order</i>	<i>Store the information of all records of all Orders</i> <ul style="list-style-type: none"> – Primary key: OrderId – Foreign keys: StudentId, StatusId
10	<i>OrderStatuses</i>	<i>Store the information of all status of an Order</i> <ul style="list-style-type: none"> – Primary key: StatusId
11	<i>Rates</i>	<i>Store the information of all rates and comments of users</i> <ul style="list-style-type: none"> – Primary key: RateId – Foreign keys: HouseId, StudentId, CreatedBy, LastModifiedBy
12	<i>ReportStatuses</i>	<i>Store the information of all status of a Report</i> <ul style="list-style-type: none"> – Primary key: StatusId
13	<i>Reports</i>	<i>Store the information of all Reports of Students to Houses</i> <ul style="list-style-type: none"> – Primary key: ReportId – Foreign keys: HouseId, StudentId, StatusId, SolvedBy
14	<i>RoomHistories</i>	<i>Store the information of all records of histories of a room</i> <ul style="list-style-type: none"> – Primary key: RoomHistoryId – Foreign keys: RoomId, CreatedBy, LastModifiedBy
15	<i>Rooms</i>	<i>Store the information of all rooms of houses</i> <ul style="list-style-type: none"> – Primary key: RoomId – Foreign keys: StatusId, RoomTypeId, HouseId, CreatedBy, LastModifiedBy
16	<i>RoomStatuses</i>	<i>Store the information of all status of a Room</i> <ul style="list-style-type: none"> – Primary key: StatusId
17	<i>RoomTypes</i>	<i>Store the information of all types of a Room</i> <ul style="list-style-type: none"> – Primary key: RoomTypeId
18	<i>UserRoles</i>	<i>Store the information of all roles of users</i> <ul style="list-style-type: none"> – Primary key: RoleId
19	<i>UserStatuses</i>	<i>Store the information of all status of an User</i> Primary key: StatusId
20	<i>Users</i>	<i>Store the information of all users</i> <ul style="list-style-type: none"> – Primary key: UserId

		<ul style="list-style-type: none"> - Foreign keys: RoleId, StatusId, AddressId, CreatedBy, LastModifiedBy
21	Villages	<p><i>Store the information of all villages (for identifying house address)</i></p> <ul style="list-style-type: none"> - Primary key: Villageld - Foreign keys: Communeld

Table 4-2: Database description

3. Database Tables

3.1. Campuses

Store the information of all campus for users

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	CampusId	int			Yes	PK	
2	CampusName	nvarchar	100		Yes		
3	AddressId	int			Yes		To Table Addresses
4	CreatedDate	datetime			Yes		

3.2. Districts

Store the information of all districts (for identifying house address)

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	DistrictId	int			Yes	PK	
2	DistrictName	nvarchar	100		Yes		
3	CampusId	int			Yes	FK	To Table Campuses
4	CreatedDate	datetime			Yes		

3.3. Communes

Store the information of all communes (for identifying house address)

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	CommunelId	int			Yes	PK	
2	CommunetName	nvarchar	100		Yes		
3	DistrictId	int			Yes	FK	To Table Districts
4	CreatedDate	datetime			Yes		

3.4. Villages

Store the information of all villages (for identifying house address)

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	Villageld	int			Yes	PK	
2	VillageName	nvarchar	100		Yes		
3	CommunelId	int			Yes	FK	To Table Communes
4	CreatedDate	datetime			Yes		

3.5. Addresses

Store the information of addresses for Users (Landlords) and Houses

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	AddressId	int			Yes	PK	
2	Addresses	nvarchar	1000		Yes		
3	GoogleMapLocation	nvarchar	MAX				Location on Google Map
4	Deleted	bit			Yes		
5	CreatedDate	datetime			Yes		
6	LastModifiedDate	datetime					

3.6. UserRoles

Store the information of all roles of users

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	RoleId	int			Yes	PK	
2	RoleName	nvarchar	100		Yes		
3	CreatedDate	datetime			Yes		

3.7. UserStatuses

Store the information of all status of an User

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	StatusId	int			Yes	PK	
2	StatusName	nvarchar	300		Yes		
3	CreatedDate	datetime			Yes		

3.8. Users

Store the information of all users

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	UserId	nchar	30		Yes	PK	Include student id, host id and admission id; Has specific convention to make
2	FacebookUserId	nchar	300				For people login with Facebook
3	GoogleUserId	nchar	300				For people login with Google
4	Email	nvarchar	100				
5	Password	nvarchar	100				Hashed with PBKDF2 with SHA-512
6	DisplayName	nvarchar	500				For Landlord and Staff
7	StatusId	int			Yes	FK	To Table <i>UserStatuses</i>
9	ProfileImageLink	nvarchar	500				For Landlord and Staff

10	PhoneNumber	nvarchar	50				Only for Landlord
11	FacebookURL	nvarchar	300				Only for Landlord
12	IdentityCardFrontSideImageLink	nvarchar	500				Only for Landlord
13	IdentityCardBackSideImageLink	nvarchar	500				Only for Landlord
14	AddressId	int					Only for Landlord; To Table Addresses
15	RoleId	int			Yes	FK	To Table Roles
16	CreatedDate	datetime			Yes		
17	LastModifiedDate	datetime					
18	CreatedBy	nchar	30		Yes	FK	To Table Users
19	LastModifiedBy	nchar	30			FK	To Table Users

3.9. Houses

Store the information of all houses

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	Houseld	int			Yes	PK	
2	HouseName	nvarchar	100		Yes		
3	Information	nvarchar	MAX				
4	View	int					
5	AddressId	int			Yes		To Table Addresses
6	Villageld	int			Yes	FK	To Table Villages
7	LandlordId	nchar	30		Yes	FK	To Table Users
8	CampusId	int			Yes	FK	To Table Campuses
9	DistanceToCampus	float			Yes		
10	PowerPrice	money			Yes		
11	WaterPrice	money			Yes		
12	FingerprintLock	bit					
13	Camera	bit					
14	Parking	bit					
15	Deleted	bit			Yes		
16	CreatedDate	datetime			Yes		
17	LastModifiedDate	datetime					
18	CreatedBy	nchar	30		Yes	FK	To Table Users
19	LastModifiedBy	nchar	30			FK	To Table Users

3.10. ImagesOfHouse

Store the information of all pictures of house

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ImageId	int			Yes	PK	
2	ImageLink	nvarchar	500		Yes		
3	Houseld	int			Yes	FK	To Table Houses
4	CreatedDate	datetime			Yes		
5	LastModifiedDate	datetime					

6	CreatedBy	nchar	30		Yes	FK	To Table Users
7	LastModifiedBy	nchar	30			FK	To Table Users
8	Deleted	bit			Yes		

3.11. Issues

Store the information of all records of Room usage for Landlords to manage if they want to

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	IssueId	int			Yes	PK	
2	Description	nvarchar	800		Yes		
3	RoomId	int			Yes	FK	To Table Rooms
4	Deleted	bit			Yes		
5	CreatedDate	datetime			Yes		
6	LastModifiedDate	datetime					
7	CreatedBy	nchar	30		Yes	FK	To Table Users
8	LastModifiedBy	nchar	30			FK	To Table Users

3.12. Rates

Store the information of all rates and comments of users

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	RateId	int			Yes	PK	
2	Star	int			Yes		
3	Comment	nvarchar	MAX				
4	LandlordReply	nvarchar	MAX				
5	HouseId	int			Yes	FK	To Table Houses
6	StudentId	nchar	30		Yes	FK	To Table Users
7	CreatedDate	datetime			Yes		
8	LastModifiedDate	datetime					
9	CreatedBy	nchar	30		Yes	FK	To Table Users
10	LastModifiedBy	nchar	30			FK	To Table Users
11	Deleted	bit			Yes		

3.13. RoomStatuses

Store the information of all status of a Room

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	StatusId	int			Yes	PK	
2	StatusName	nvarchar	300		Yes		
3	CreatedDate	datetime			Yes		

3.14. RoomTypes

Store the information of all types of a Room

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes

1	RoomTypeId	int			Yes	PK	
2	RoomTypeName	nvarchar	300		Yes		
3	CreatedDate	datetime			Yes		

3.15. Rooms

Store the information of all rooms of houses

#	Field name	Type	Size	Unique	Not Null	PK/ FK	Notes
1	RoomId	int			Yes	PK	
2	RoomName	nvarchar	50				
3	PricePerMonth	money			Yes		
6	Information	nvarchar	MAX				
7	AreaByMeters	float					
8	Fridge	bit					
9	Bed	bit					
10	WashingMachine	bit					
11	Desk	bit					
12	Kitchen	bit					
13	ClosedToilet	bit					
14	NoLiveWithLandlord	bit					
15	MaxAmountOfPeople	int					
16	CurrentAmountOfPeople	int					
17	BuildingNumber	int					
18	FloorNumber	int					
19	StatusId	int			Yes	FK	To Table Statuses
20	RoomTypeId	int			Yes	FK	To Table RoomTypes
21	Houseld	int			Yes	FK	To Table Houses
22	CampusId	int			Yes	FK	To Table Campuses
23	CreatedDate	datetime			Yes		
24	LastModifiedDate	datetime					
25	CreatedBy	nchar	30		Yes	FK	To Table Users
26	LastModifiedBy	nchar	30			FK	To Table Users
27	Deleted	bit			Yes		

3.16. ImagesOfRoom

Store the information of all pictures of room

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ID	int			Yes	PK	
2	ImageLink	nvarchar	500		Yes		
3	RoomId	int			Yes	FK	To Table Rooms
4	CreatedDate	datetime			Yes		
5	LastModifiedDate	datetime					
6	CreatedBy	nchar	30		Yes	FK	To Table Users
7	LastModifiedBy	nchar	30			FK	To Table Users
8	Deleted	bit			Yes		

3.17. RoomHistories

Store the information of all records of Room usage for Landlords to manage if they want to

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	RoomHistoryId	int			Yes	PK	
2	CustomerName	nvarchar	800		Yes		Name of the person stayed in the Room
3	RoomId	int			Yes	FK	To Table Rooms
4	CreatedDate	datetime			Yes		
5	LastModifiedDate	datetime					
6	CreatedBy	nchar	30		Yes	FK	To Table Users
7	LastModifiedBy	nchar	30			FK	To Table Users
8	Deleted	bit			Yes		

3.18. ReportStatuses

Store the information of all status of a Report

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	StatusId	int			Yes	PK	
2	StatusName	nvarchar	300		Yes		
3	CreatedDate	datetime			Yes		

3.19. Reports

Store the information of all Reports of Students to Houses

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	ReportId	int			Yes	PK	
2	ReportContent	nvarchar	MAX		Yes		
3	StudentId	nchar	30		Yes	FK	To Table Users
4	HouseId	int			Yes	FK	To Table Houses
5	ReportedDate	datetime			Yes		
6	SolvedDate	datetime					
7	SolvedBy	nchar	30			FK	To Table Users
8	Deleted	bit			Yes		

3.20. OrderStatuses

Store the information of all status of an Order

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	StatusId	int			Yes	PK	
2	StatusName	nvarchar	300		Yes		
3	CreatedDate	datetime			Yes		

3.21. Orders

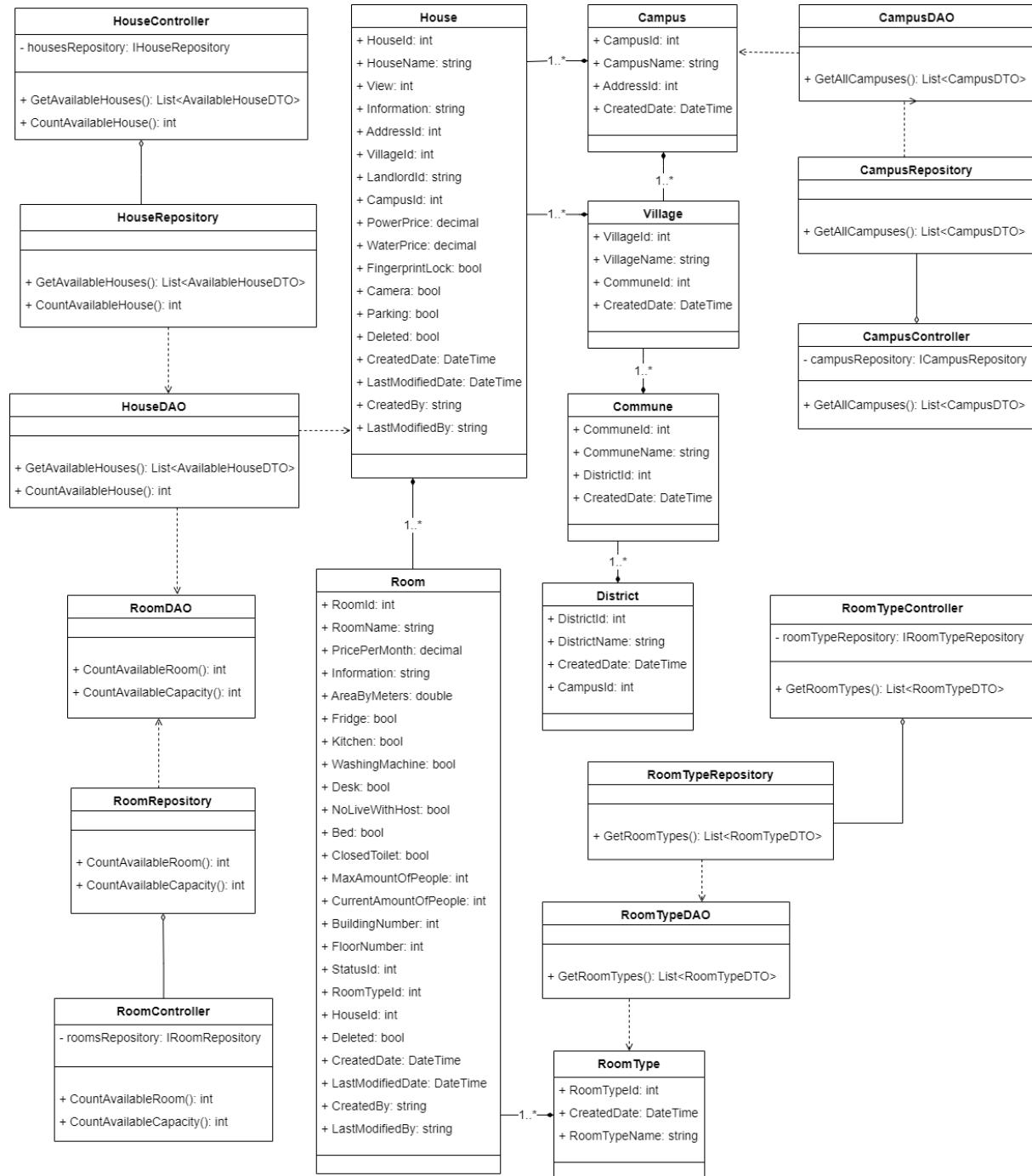
Store the information of all records of Order usage for Staffs to manage if they want to

#	Field name	Type	Size	Unique	Not Null	PK/FK	Notes
1	OrderId	int			Yes	PK	
2	StudentId	nchar	30		Yes	FK	To Table <i>Users</i>
3	StudentName	nvarchar	100		Yes		
4	PhoneNumber	nvarchar	50		Yes		
5	Email	nvarchar	100		Yes		
6	OrderContent	nvarchar					
7	StatusId	int			Yes	FK	To Table <i>OrderStatuses</i>
8	OrderedDate	datetime					
9	SolvedDate	datetime					
10	SolvedBy	nchar	30			FK	To Table <i>Users</i>

4. Detailed Design

4.1. Home Page

4.1.1 Class Diagram



4.1.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

CampusDAO Class

This class performs action to data (CRUD) in Campus Table in Database

Class Methods

No	Method	Description
01	GetAllCampuses()	Return list of all Campuses in the Database
02	GetCampusByName()	Find a Campus by its Name

CampusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllCampuses()	Return list of all Campuses in the Database
02	GetCampusByName()	Find a Campus by its Name

DistrictDAO Class

This class performs action to data (CRUD) in District Table in Database

Class Methods

No	Method	Description
01	GetAllDistricts()	Return list of all Districts in the Database

DistrictController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllDistricts()	Return list of all Districts in the Database

RoomTypeDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseId()	Get all roomTypes that this house has

RoomTypeController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseId()	Get all roomTypes that this house has

RoomDAO Class

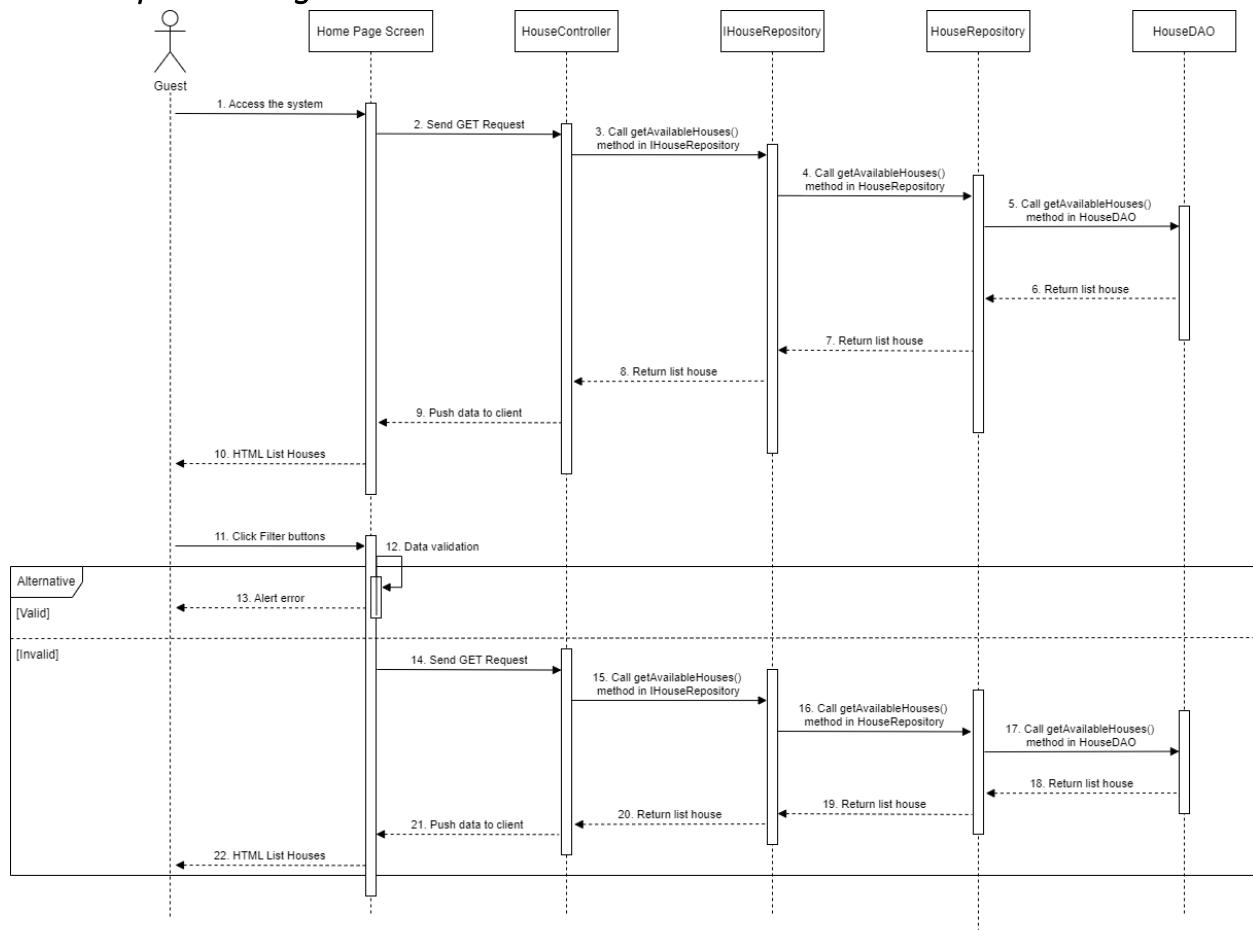
This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database

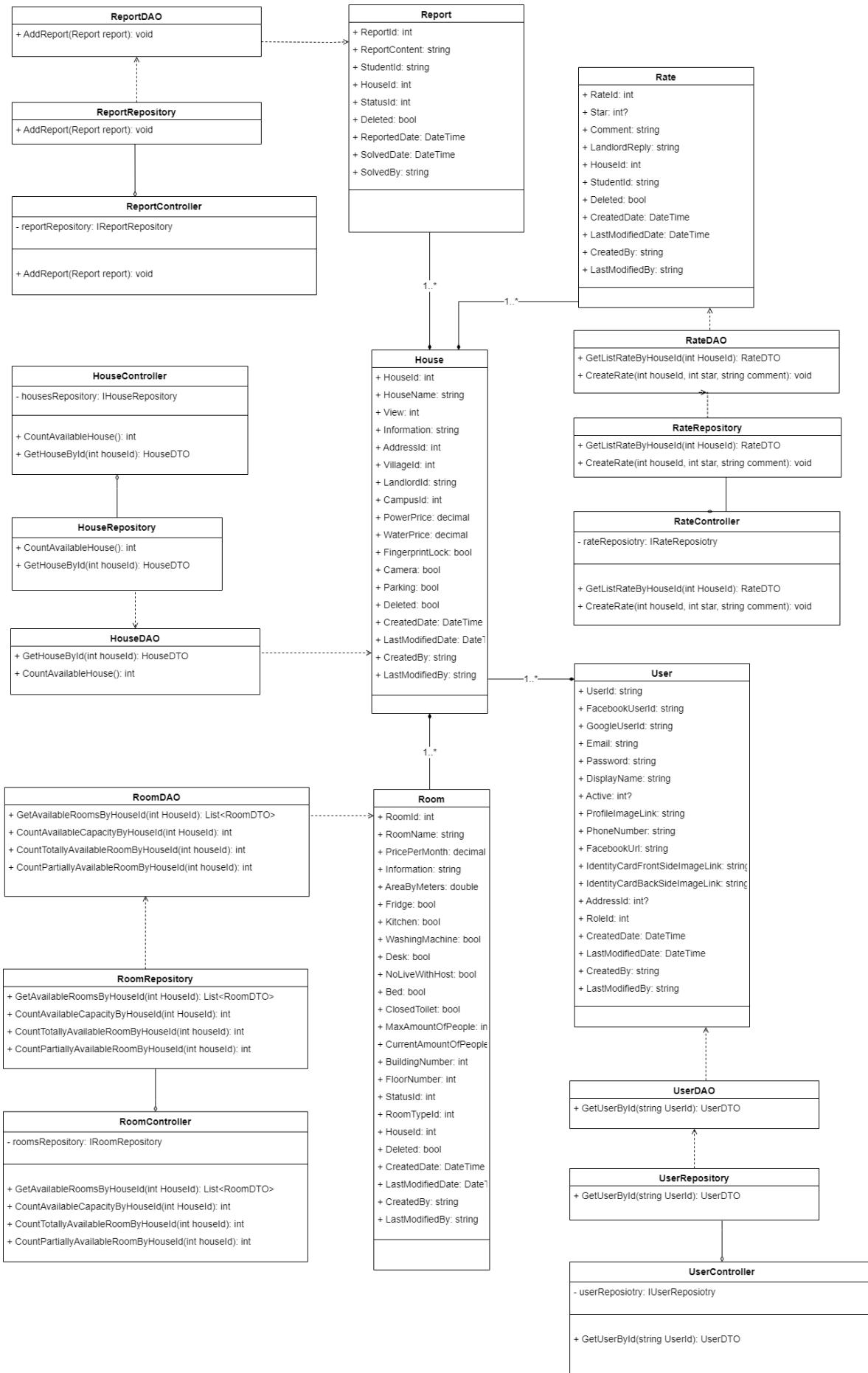
No	Method	Description
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

4.1.3 Sequence Diagram



4.2. House Detail

4.2.1 *Class Diagram*



4.2.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system

12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

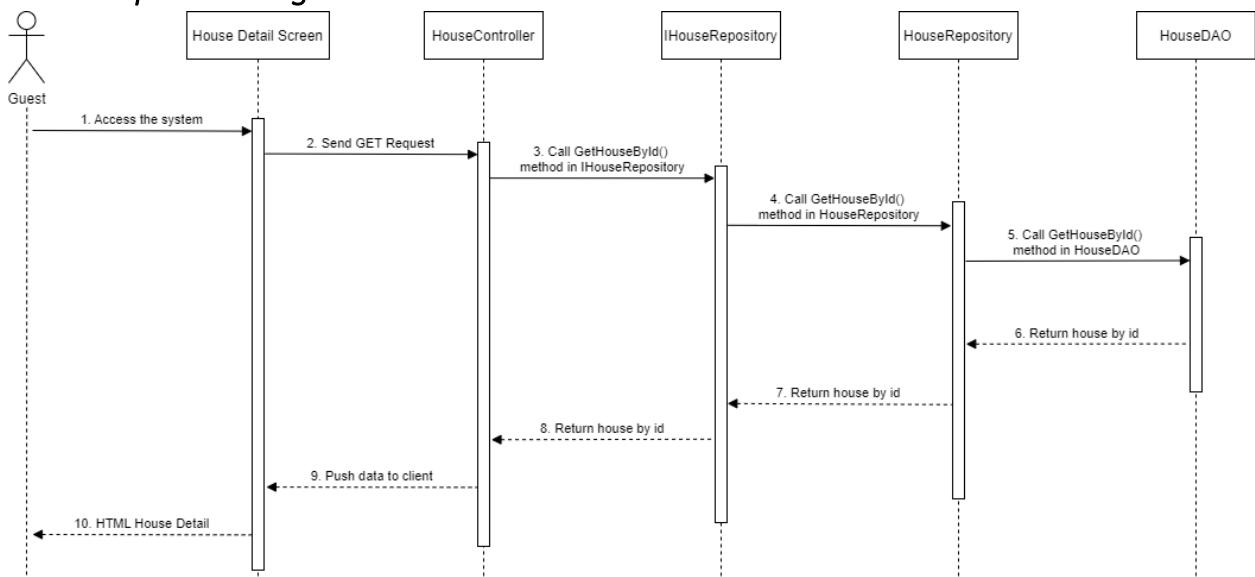
UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

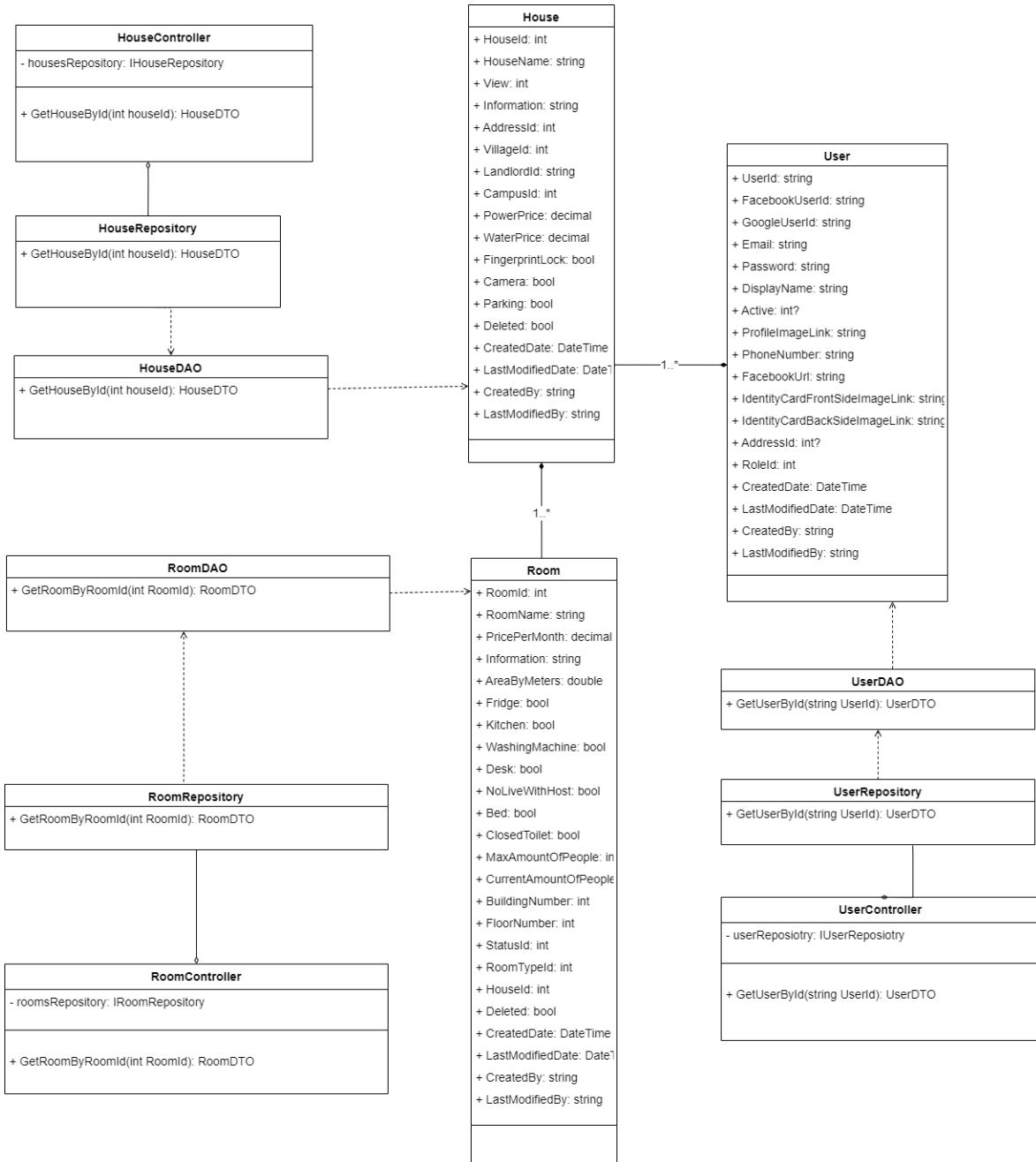
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

4.2.3 Sequence Diagram



4.3. Room Detail

4.3.1 Class Diagram



4.3.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
----	--------	-------------

01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House

13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Find detail information of list Rooms by house Id
02	GetAvailableRoomsByHouseId ()	Find detail information of list available Rooms by house Id
03	GetRoomsByRoomId ()	Find detail information of list Room by its Id
04	CreateRoom ()	Add a new Room into the Database
05	UpdateRoomByRoomId ()	Update a Room into the Database
06	DeleteRoom ()	Delete a Room
07	CountAvailableRoom ()	Count number of available Rooms in the system
08	CountTotalRoom()	Count number of total Rooms in the system
09	CountAvailableCapacity()	Count number of capacity of available Rooms in the system
10	CountTotalCapacity()	Count number of capacity of Rooms in the system
11	CountTotallyAvailableCapacity()	Count number of total capacity of available Rooms in the system
12	CountTotallyAvailableRoomByHouseId()	Count number of total capacity of available Rooms in the system by house id
13	CountPartiallyAvailableRoomByHouseId()	Count number of total partially of available Rooms in the system by house id
14	CountAvailableCapacityByHouseId()	Count number of total available capacity of Rooms in the system by house id
15	ChangeStatusRoom()	Update a Room status into the Database

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

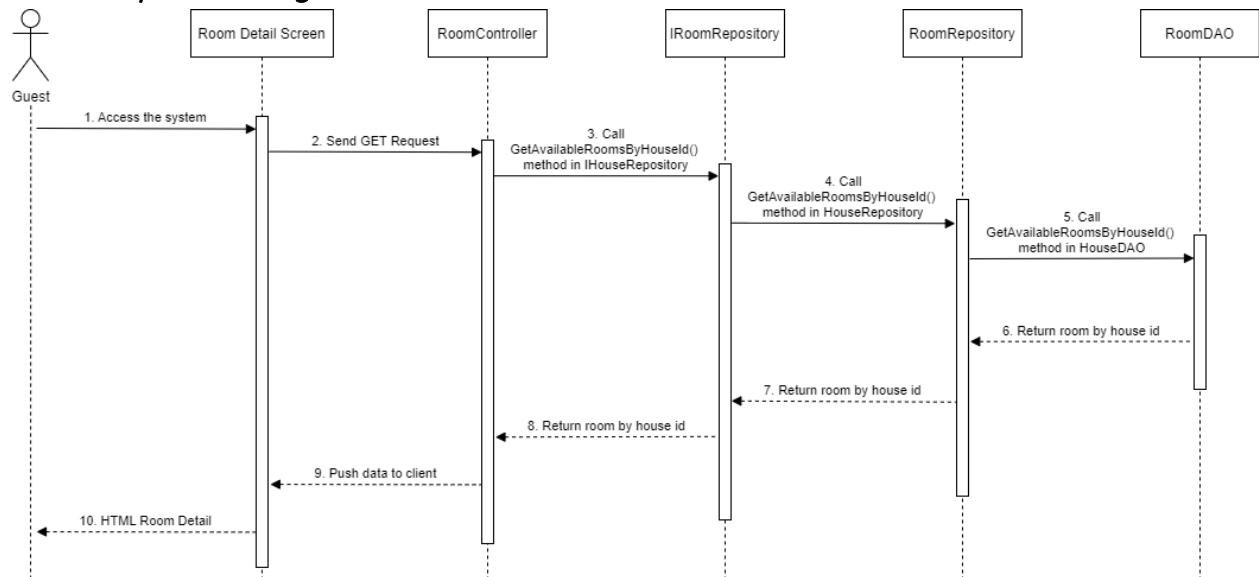
UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

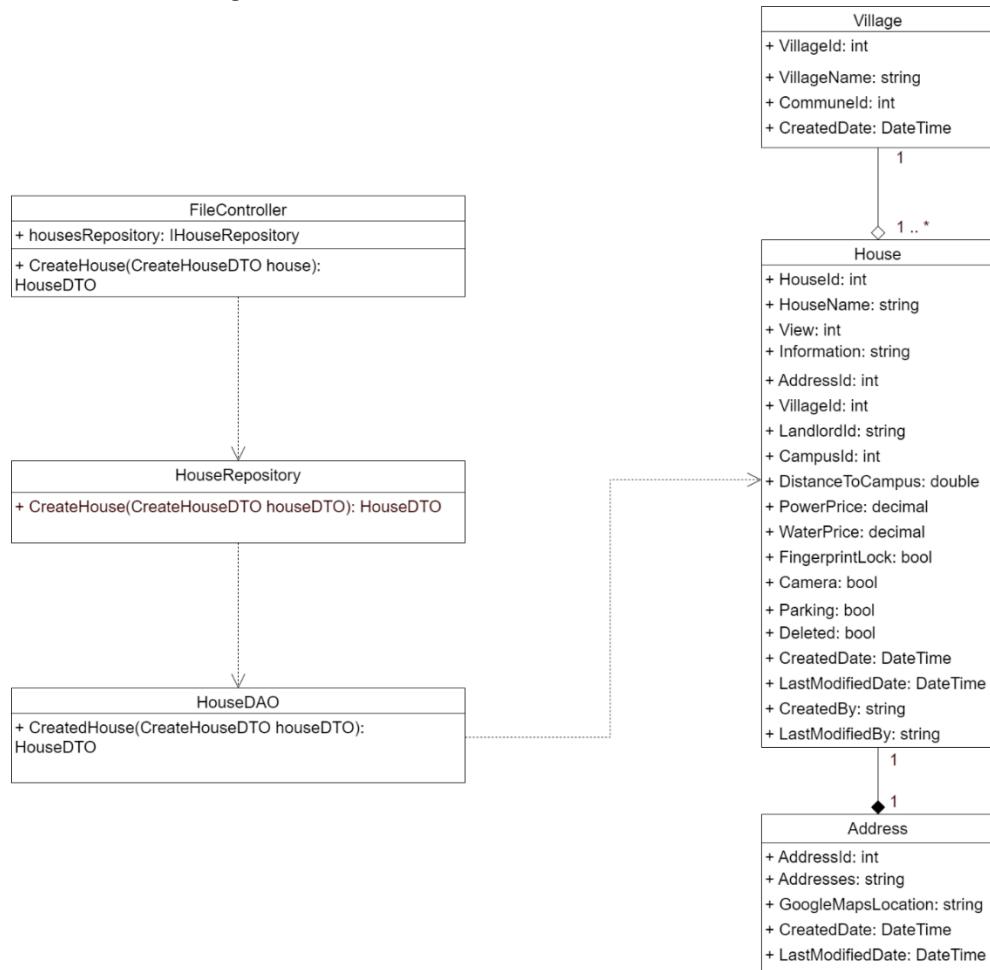
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

4.3.3 Sequence Diagram



4.4. Landlord – Create House

4.3.1 Class Diagram



4.3.2 Class Specifications

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

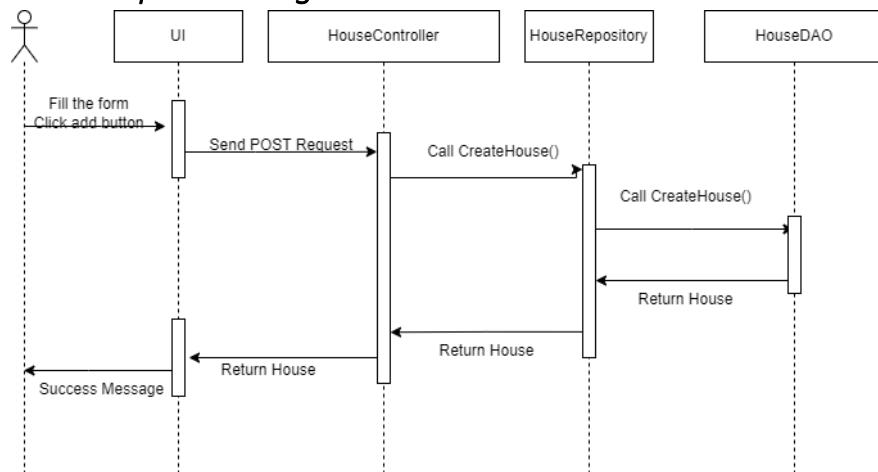
HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

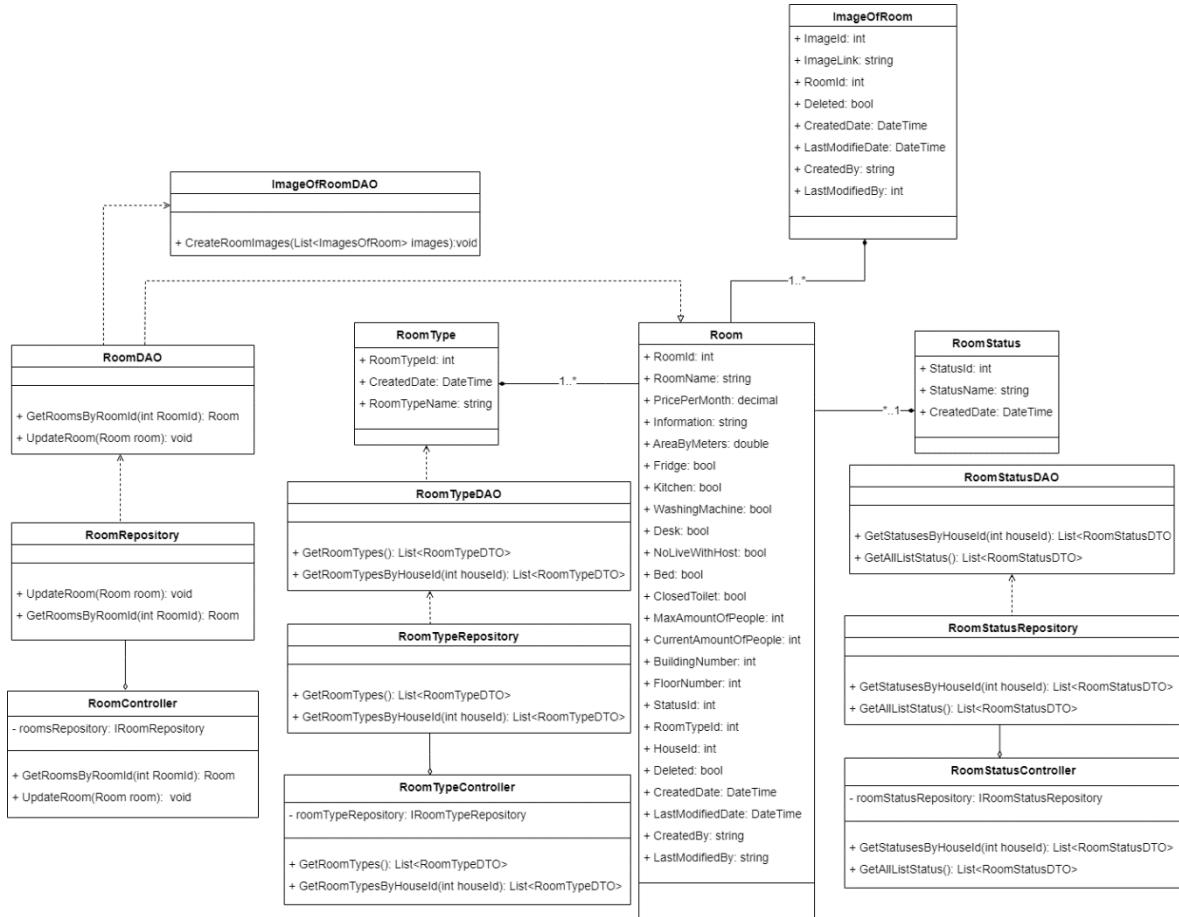
No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

4.3.3 Sequence Diagram



4.5. Landlord – List Room

4.5.1 Class Diagram



4.5.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
----	--------	-------------

01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House

03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomTypeDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseId()	Get all roomTypes that this house has

RoomTypeController Class

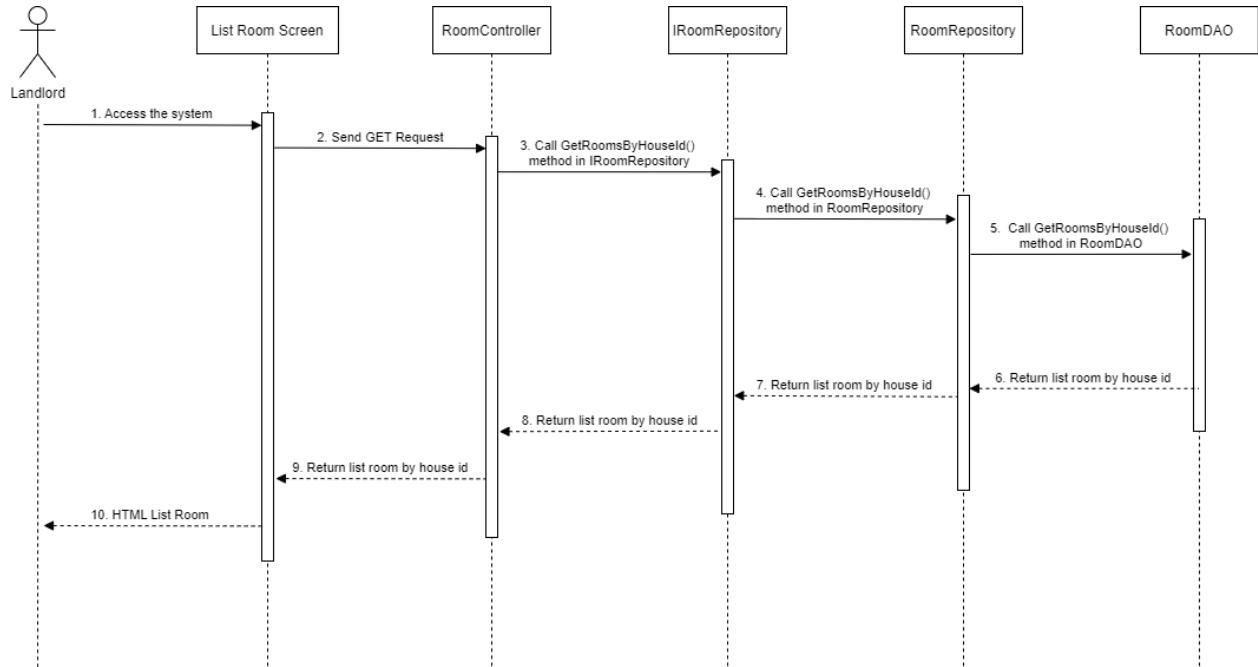
This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
----	--------	-------------

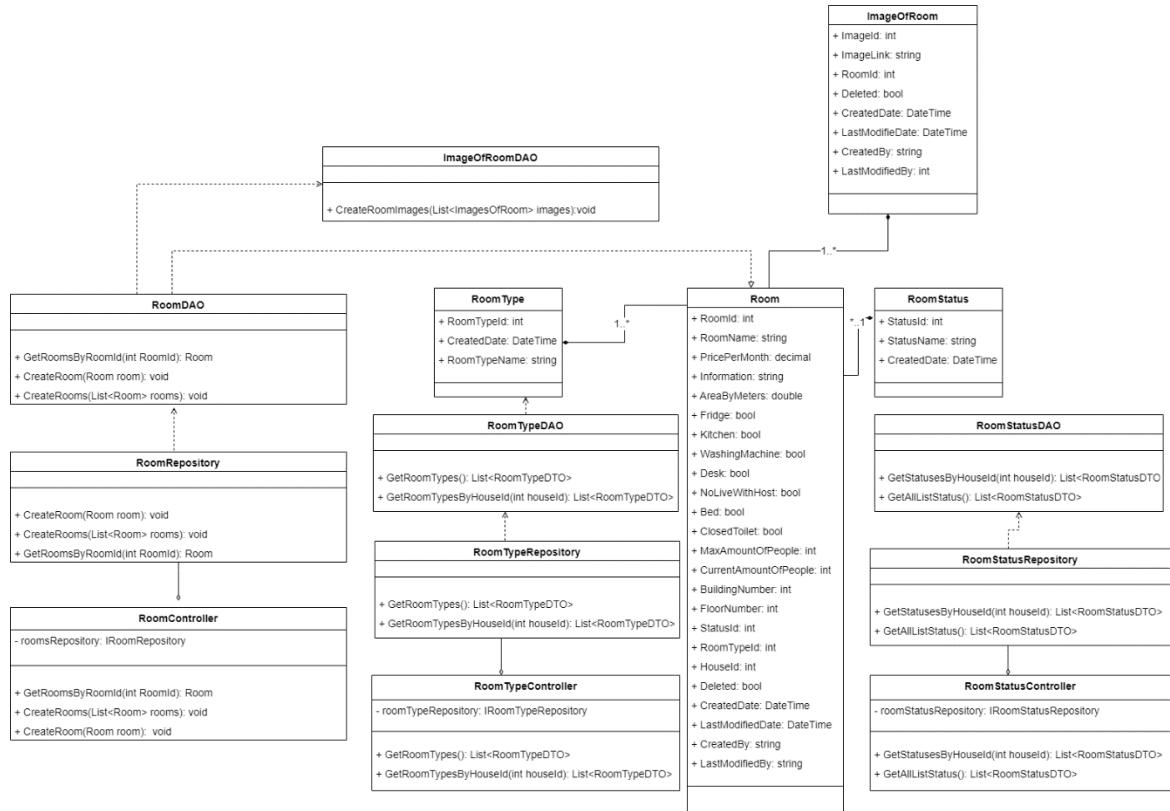
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseId()	Get all roomTypes that this house has

4.5.3 Sequence Diagram



4.6. Landlord – Create Room

4.6.1 Class Diagram



4.6.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system

10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database

03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomTypeDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseId()	Get all roomTypes that this house has

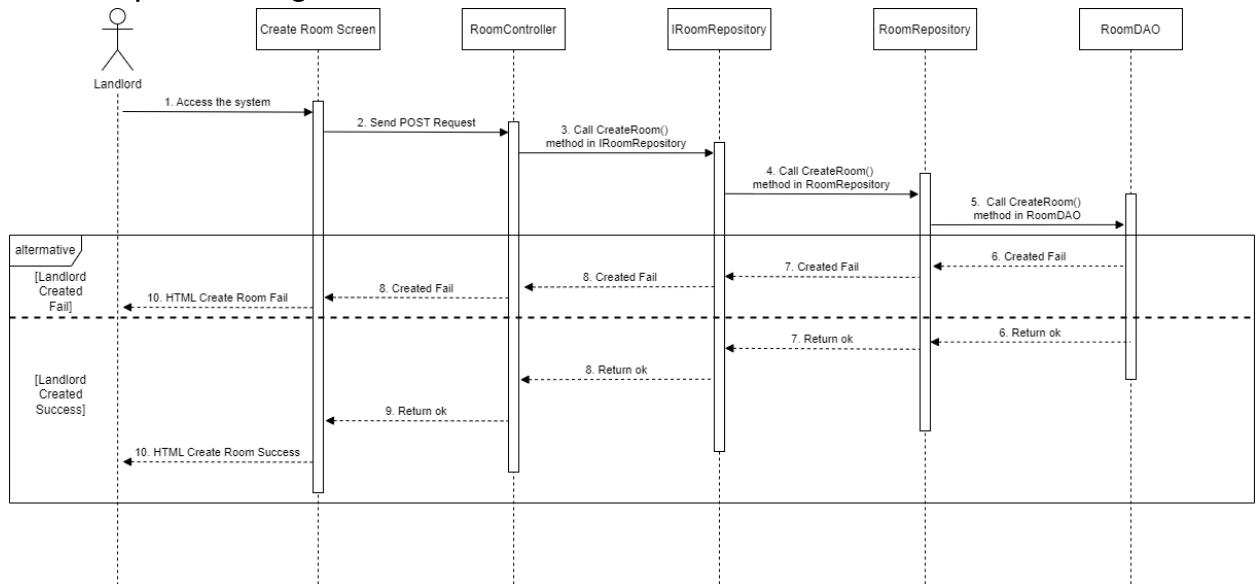
RoomTypeController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

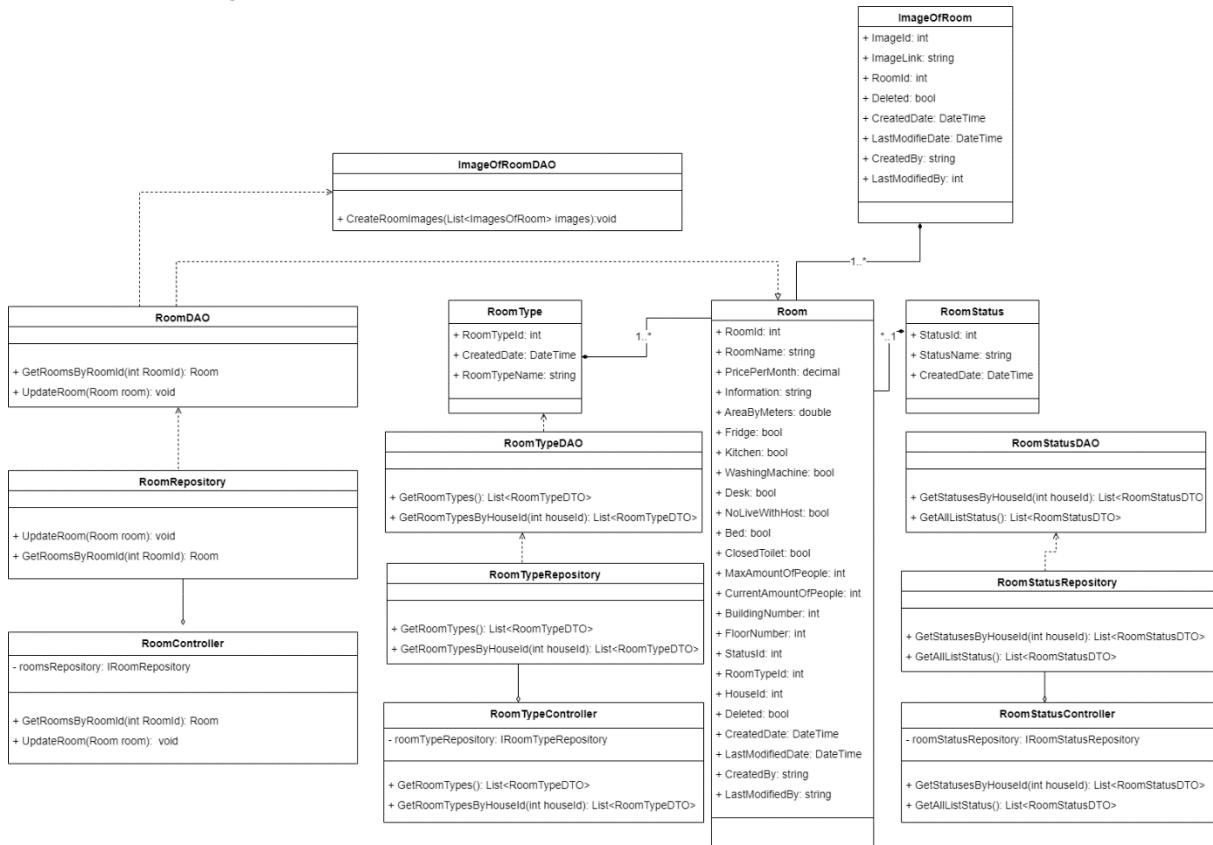
No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseId()	Get all roomTypes that this house has

4.6.3 Sequence Diagram



4.7. Landlord – Update Room

4.7.1 Class Diagram



4.7.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system

10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database

03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetUserByld ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseld()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseld()	Get all RoomStatus that this house has

RoomTypeDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseld()	Get all roomTypes that this house has

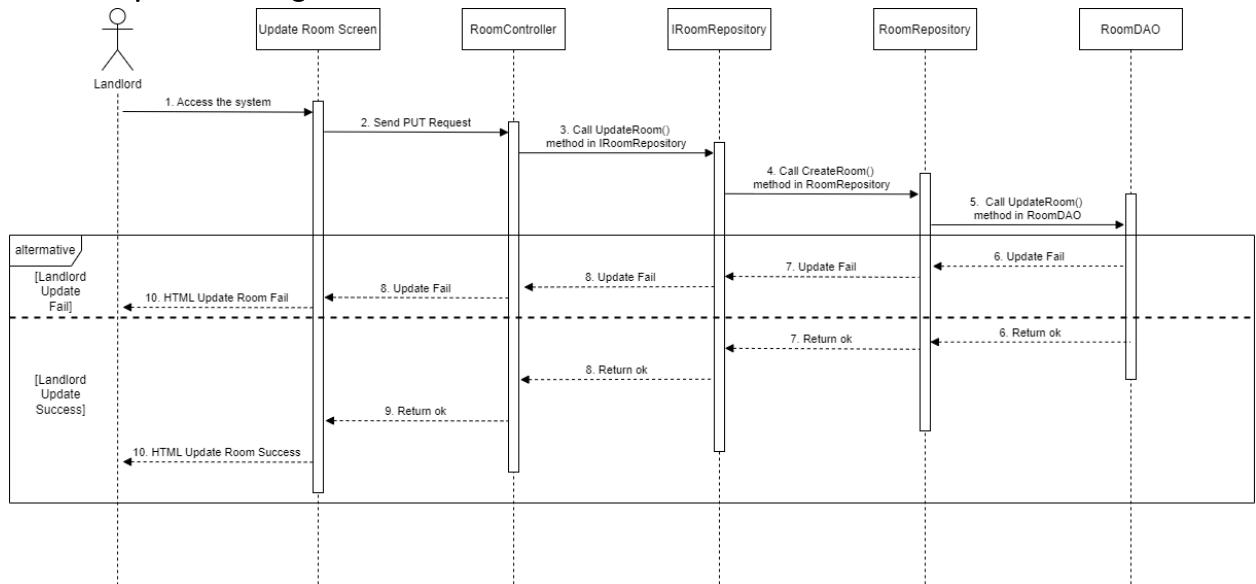
RoomTypeController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

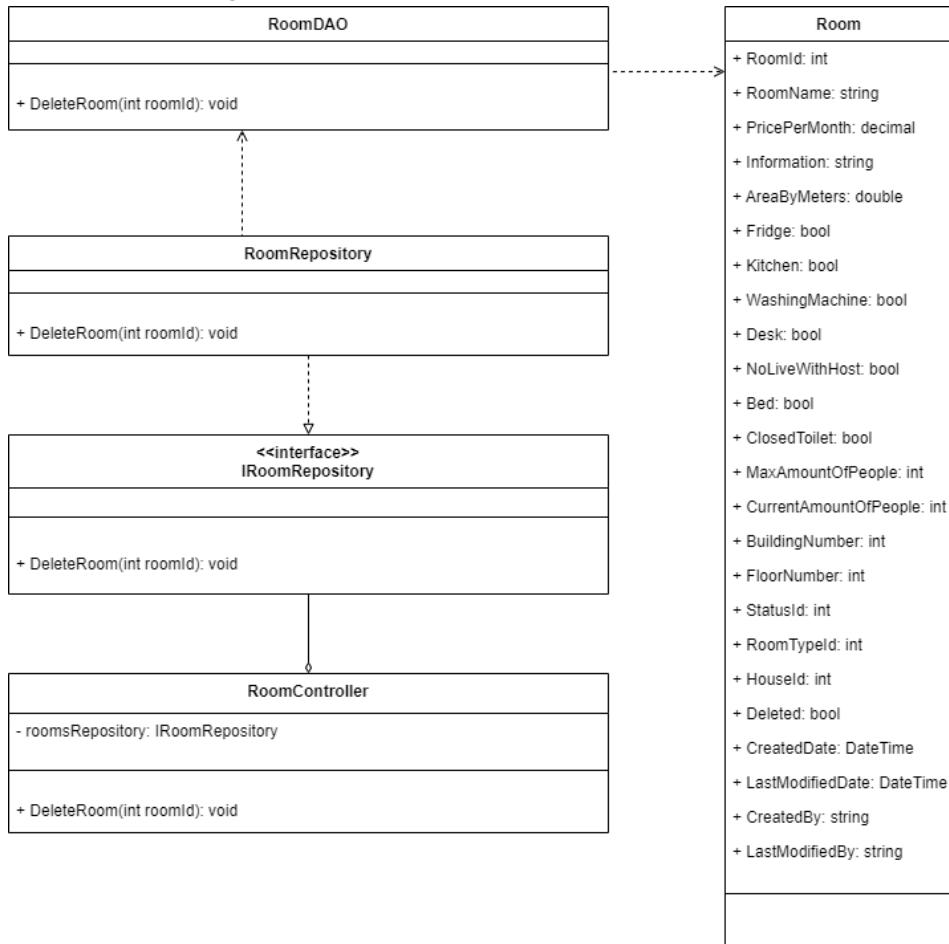
No	Method	Description
01	GetRoomTypes()	Return list of all RoomTypes in the Database
02	GetRoomTypesByHouseld()	Get all roomTypes that this house has

4.7.3 Sequence Diagram



4.8. Landlord – Delete Room

4.8.1 Class Diagram



4.8.2 Class Specifications

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

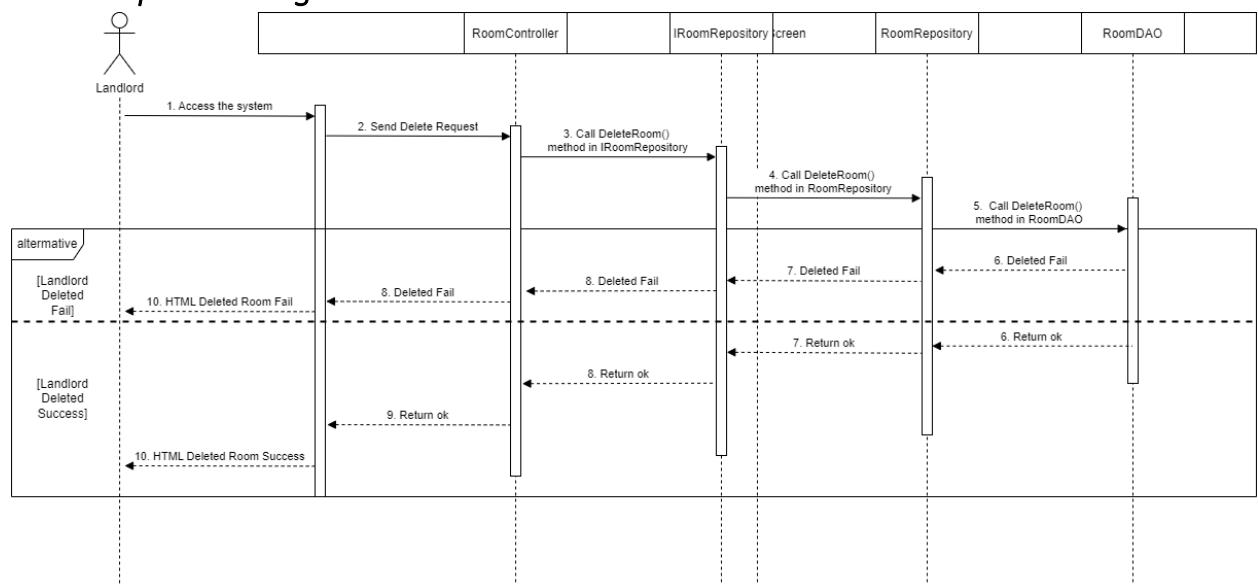
This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system

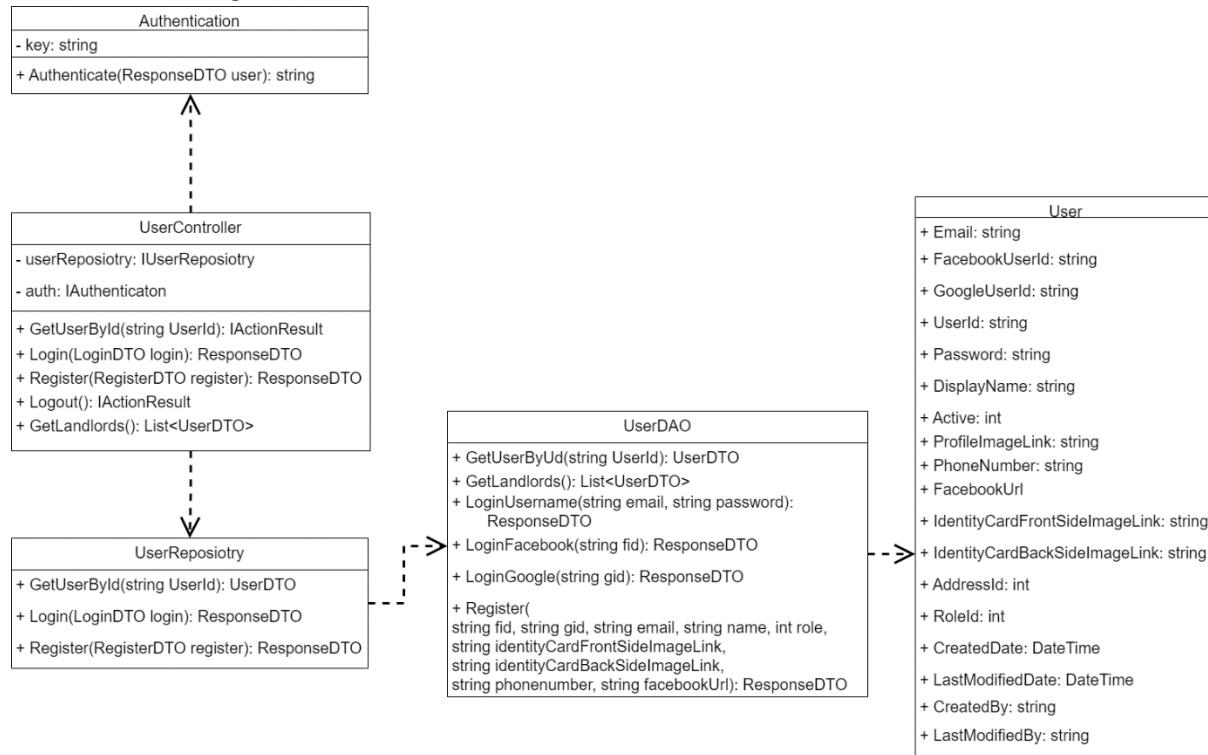
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

4.8.3 Sequence Diagram



4.9. Login

4.9.1 Class Diagram



4.9.2 Class Specifications

UserController Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

UserRepository Class

This class performs logic and get data from DAO to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	Login()	Get user with login data provided
06	Register()	User create an account

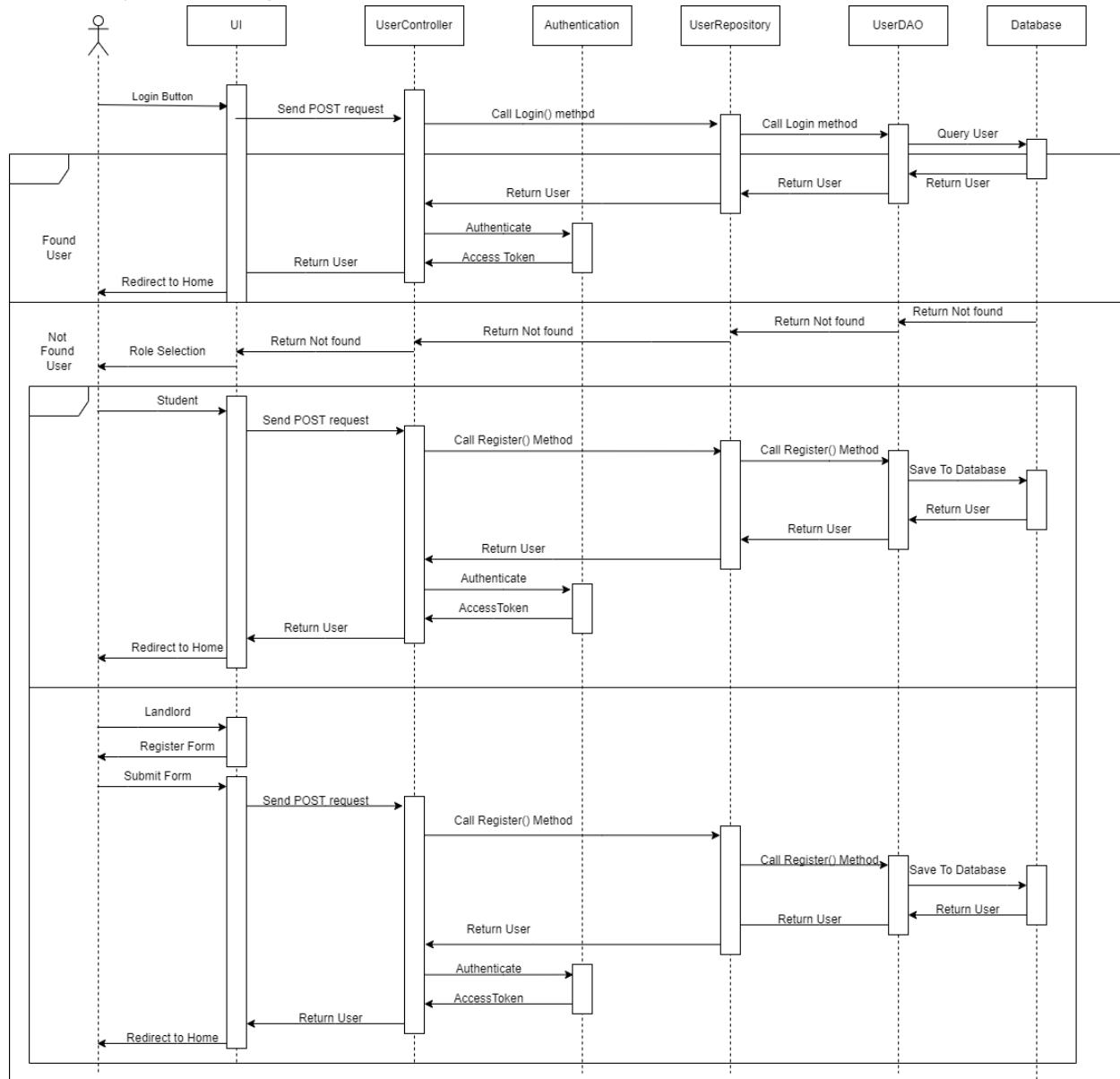
Authentication Class

This class create AccessToken for User

Class Methods

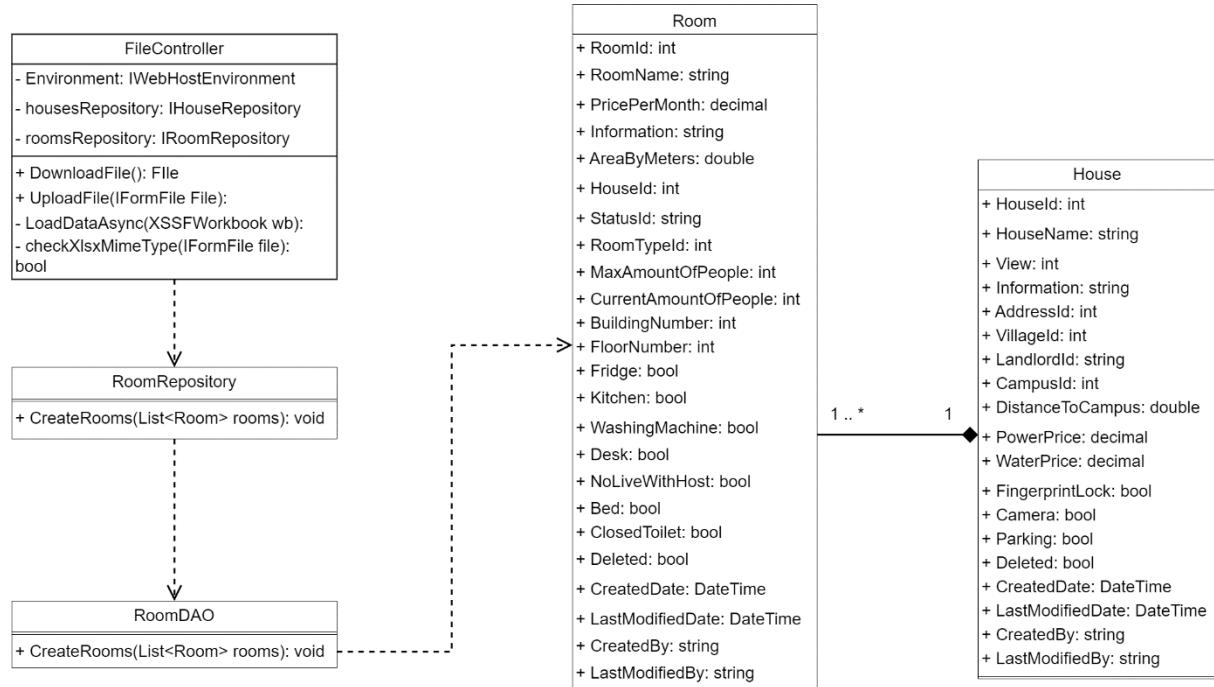
No	Method	Description
01	Authenticate()	Find detail information of User by its Id

4.9.3 Sequence Diagram



4.10. Landlord – Upload House Information

4.10.1 Class Diagram



4.10.2 Class Specifications

FileController Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	DownloadFile()	Download Template file to User device
02	UploadFile()	Get data File from request
03	LoadDataAsync()	Create house, room data with Data from file
04	checkXlsxMimeType()	Check upload file mime type

RoomDAO Class

This class class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system

10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

AddressDAO Class

This class performs action to data (CRUD) in Address Table in Database

Class Methods

No	Method	Description
01	CreateAddress()	Create Address record and save to database

CampusDAO Class

This class performs action to data (Read) in Campus Table in Database

Class Methods

No	Method	Description
01	GetAllCampuses()	Get List of All Campuses
02	GetCampusByName()	Get Campus by Campus Name
03	GetCampusById()	Get Campus by Campus Id

VillageDAO Class

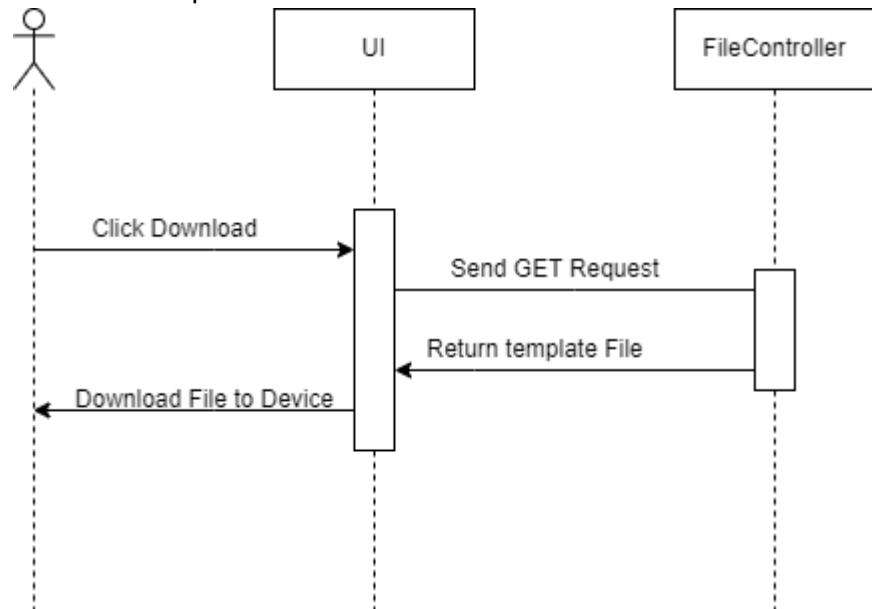
This class performs action to data (Read) in Village Table in Database

Class Methods

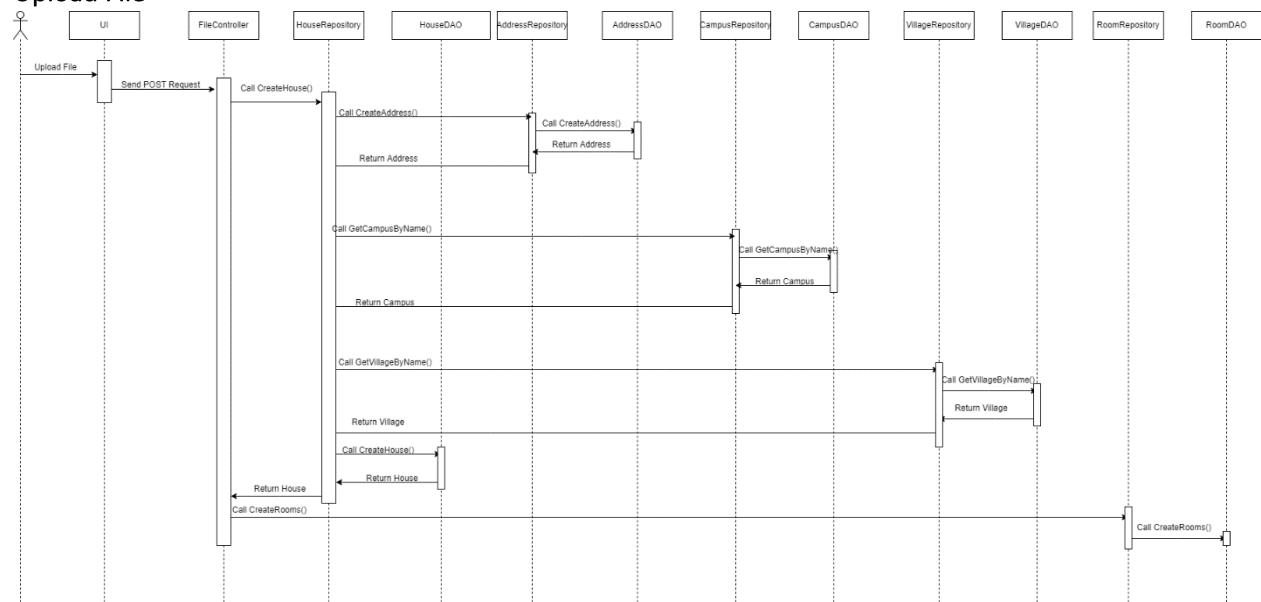
No	Method	Description
01	GetVillageByName()	Get Village by Village Name
02	GetVillageById()	Get Village by Village Id

4.10.3 Sequence Diagram

Download Template

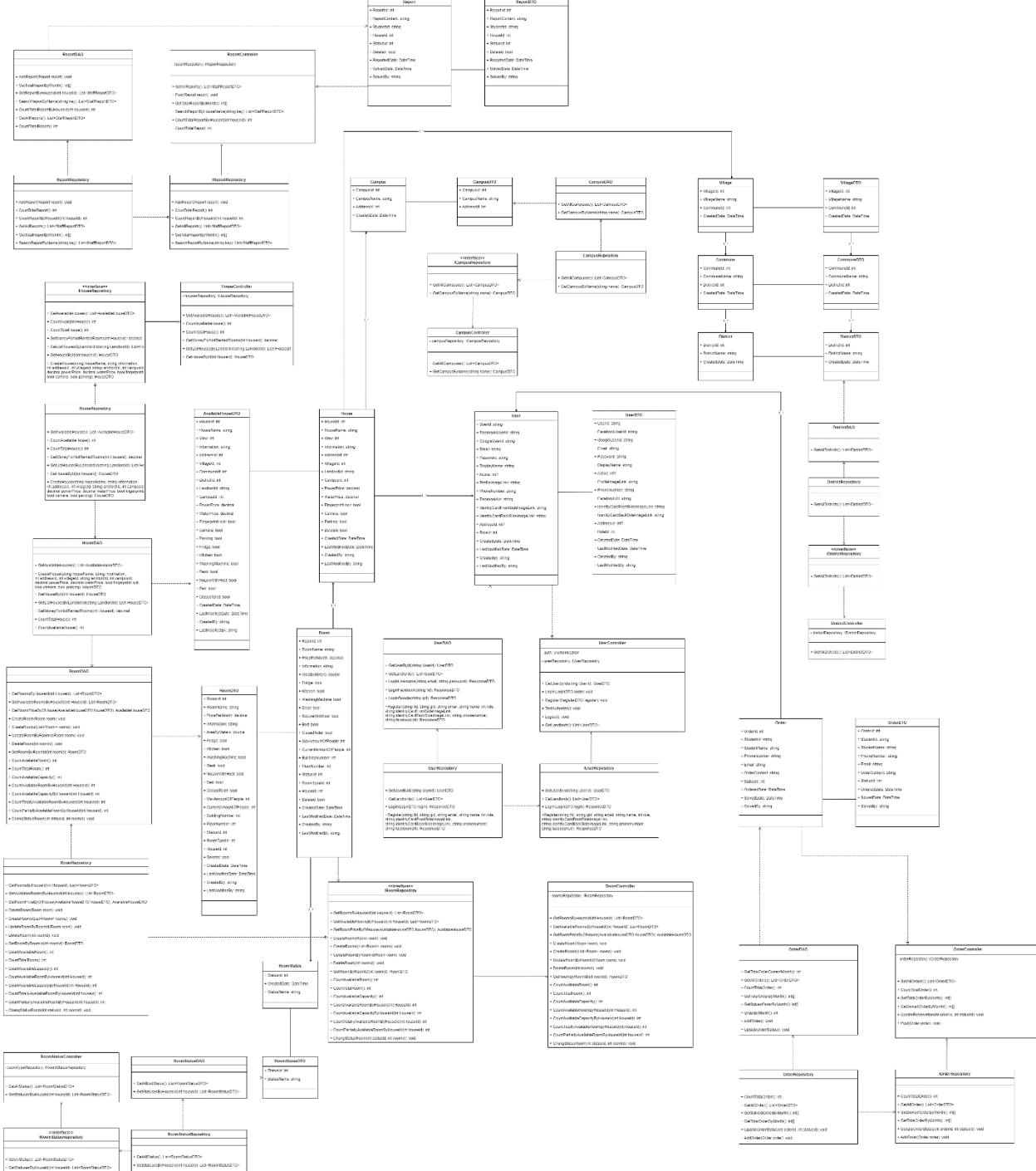


Upload File



4.11. Staff – Dashboard

4.11.1 Class Diagram



4.11.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
----	--------	-------------

01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House

13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database

02	GetStatusesByHouseId()	Get all RoomStatus that this house has
----	------------------------	--

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

OrderDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetTotalOrderCurrentMonth() ()	Get total order of the current month
02	GetAllOrders()	Get list of Orders in the database
03	CountTotalOrder()	Count total Orders
04	GetTotalOrderByMonth()	Get total Orders of the month
05	GetSolvedOrderByMonth()	Calculate number of solved Order by month
06	OrderByMonth()	Count Orders by month

OrderController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetTotalOrderCurrentMonth() ()	Get total order of the current month
02	GetAllOrders()	Get list of Orders in the database
03	CountTotalOrder()	Count total Orders
04	GetTotalOrderByMonth()	Get total Orders of the month
05	GetSolvedOrderByMonth()	Calculate number of solved Order by month
06	OrderByMonth()	Count Orders by month

ReportDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	AddReport()	Add new Report to the database
02	GetTotalReportByMonth ()	Count the total of Report by month
03	GetReportByHouseId()	Get list of Reports by house id
04	SearchReportByName()	Search Report by house's name
05	CountTotalReportByHouseId() ()	Count total Report by house id
06	GetAllReports()	Get all Reports
07	CountTotalReport()	Count total Report

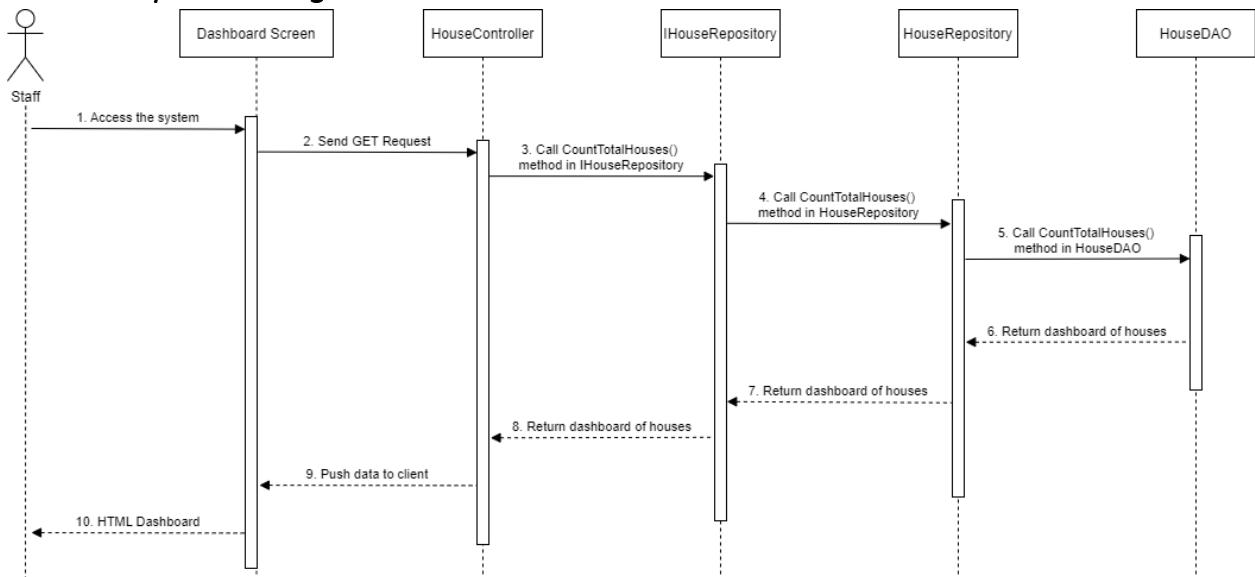
ReportController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

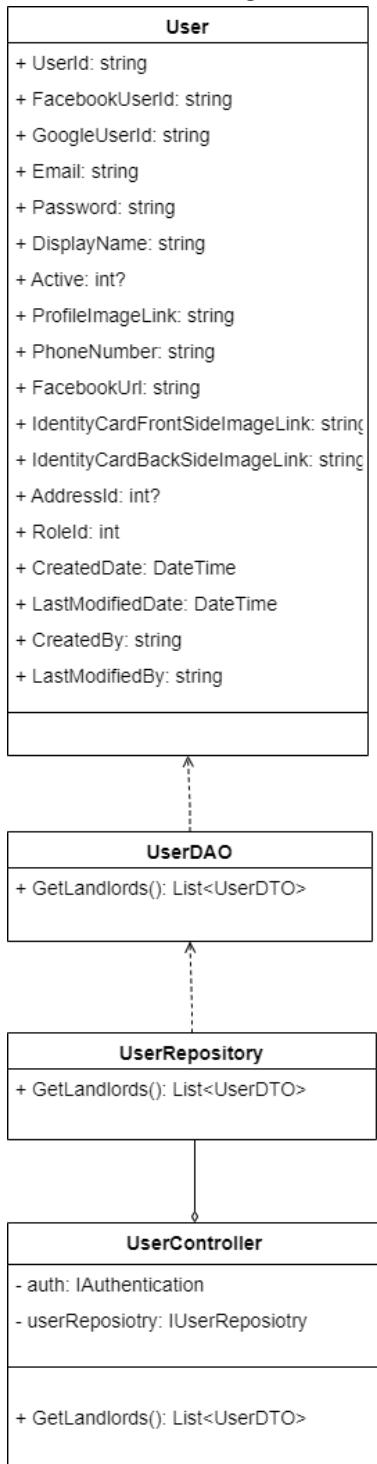
No	Method	Description
01	AddReport()	Add new Report to the database
02	GetTotalReportByMonth ()	Count the total of Report by month
03	GetReportByHouseId()	Get list of Reports by house id
04	SearchReportByName()	Search Report by house's name
05	CountTotalReportByHouseId() ()	Count total Report by house id
06	GetAllReports()	Get all Reports
07	CountTotalReport()	Count total Report

4.11.3 Sequence Diagram



4.12. Staff – List Landlord

4.12.1 Class Diagram



4.12.2 Class Specifications

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

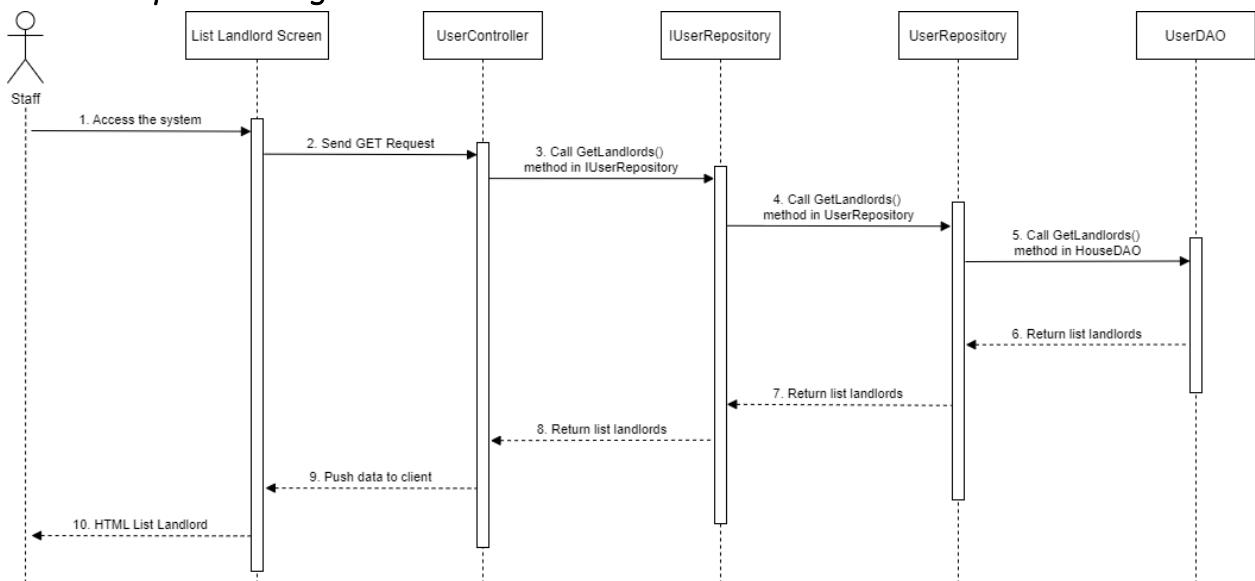
UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

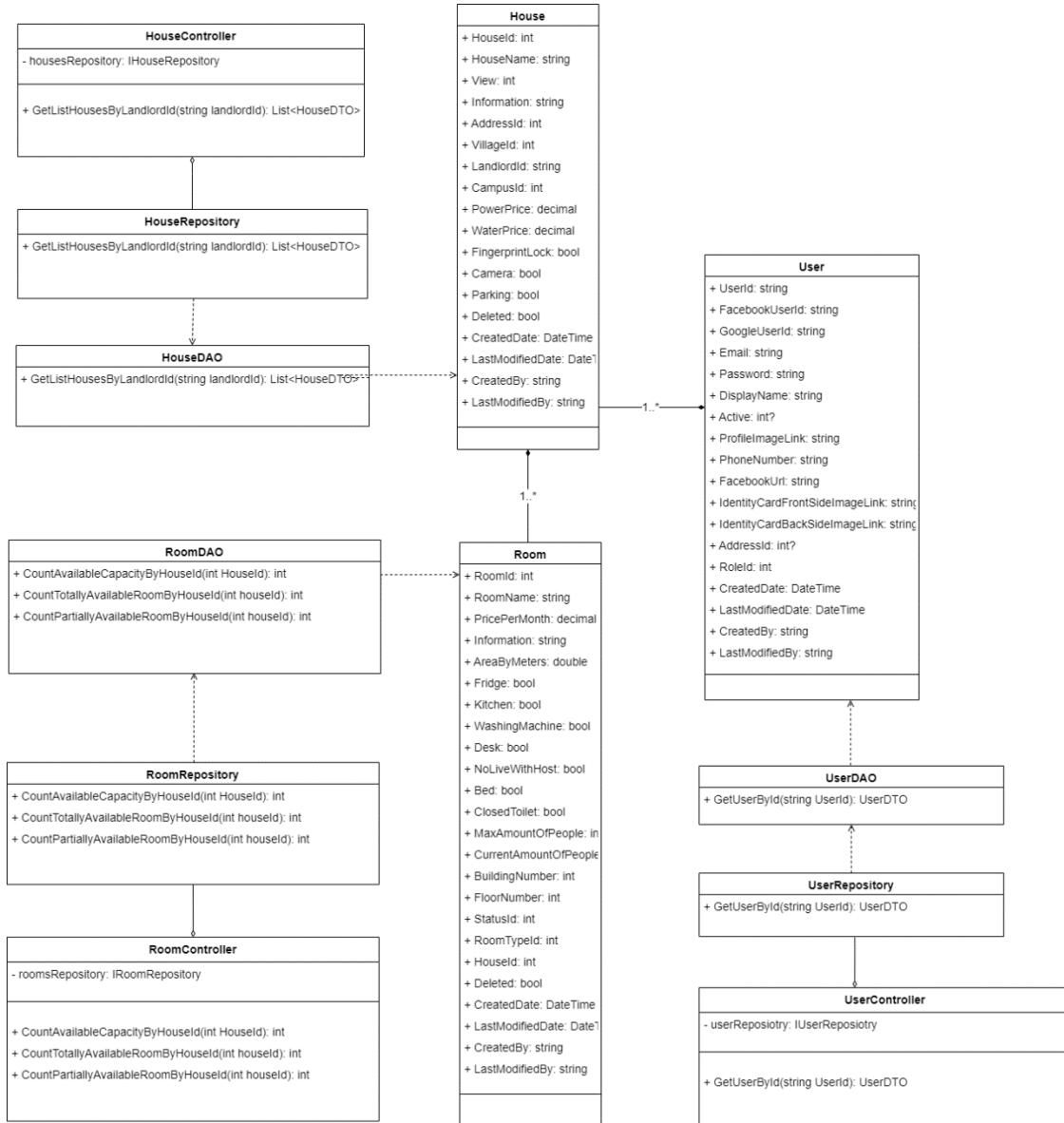
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

4.12.3 Sequence Diagram



4.13. Staff - Landlord Detail

4.13.1 Class Diagram



4.13.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)

06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id

09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description

01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

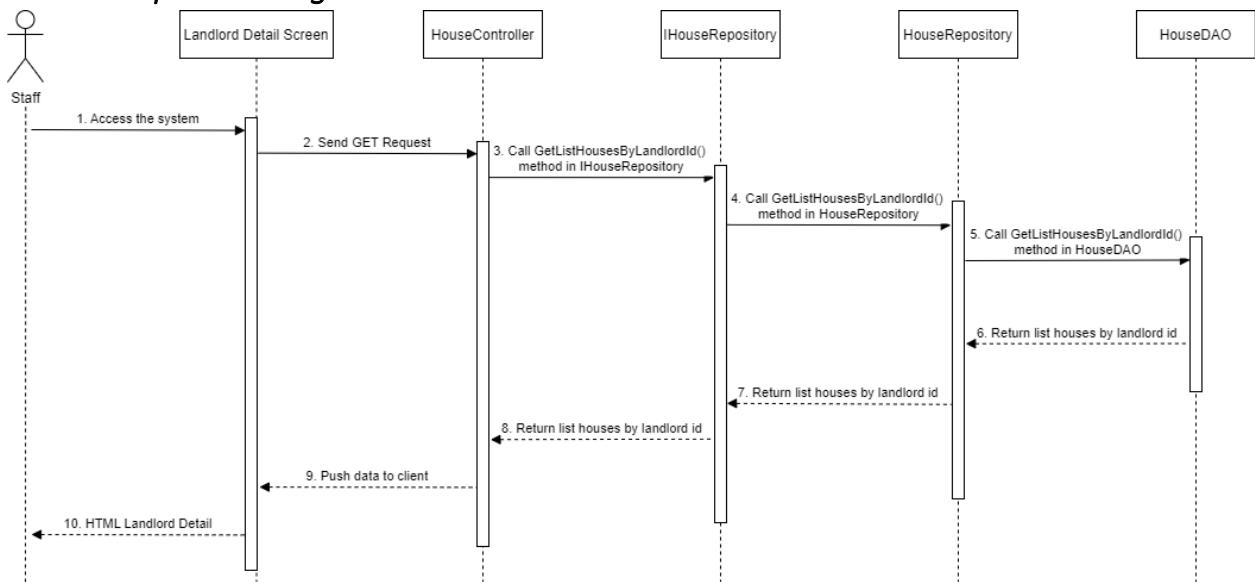
UserController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

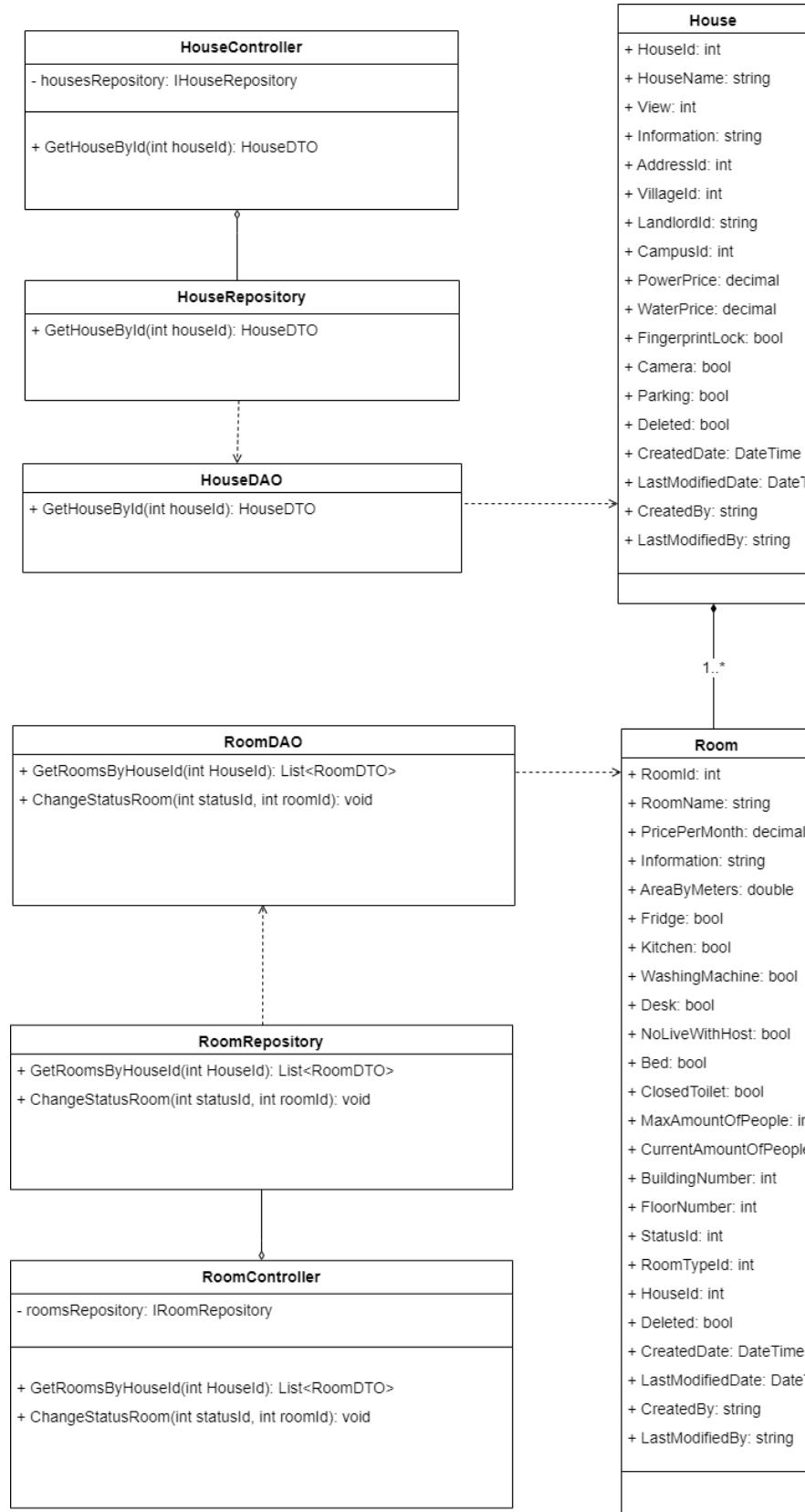
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

4.13.3 Sequence Diagram



4.14. Staff - House Detail

4.14.1 Class Diagram



4.14.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

RoomController Class

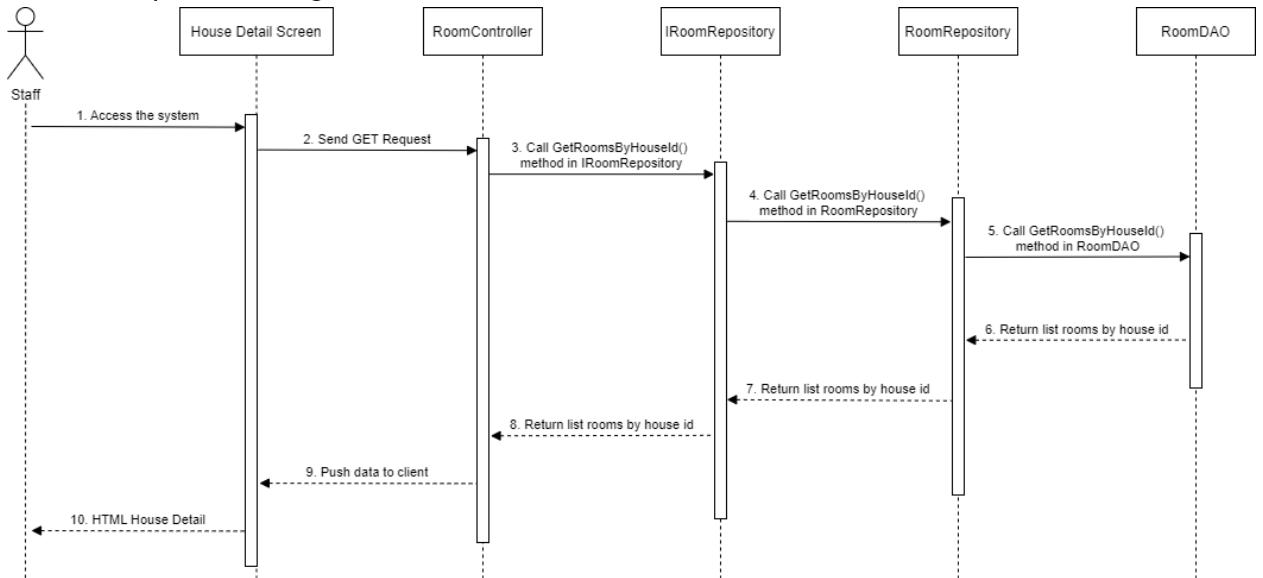
This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByOfHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system

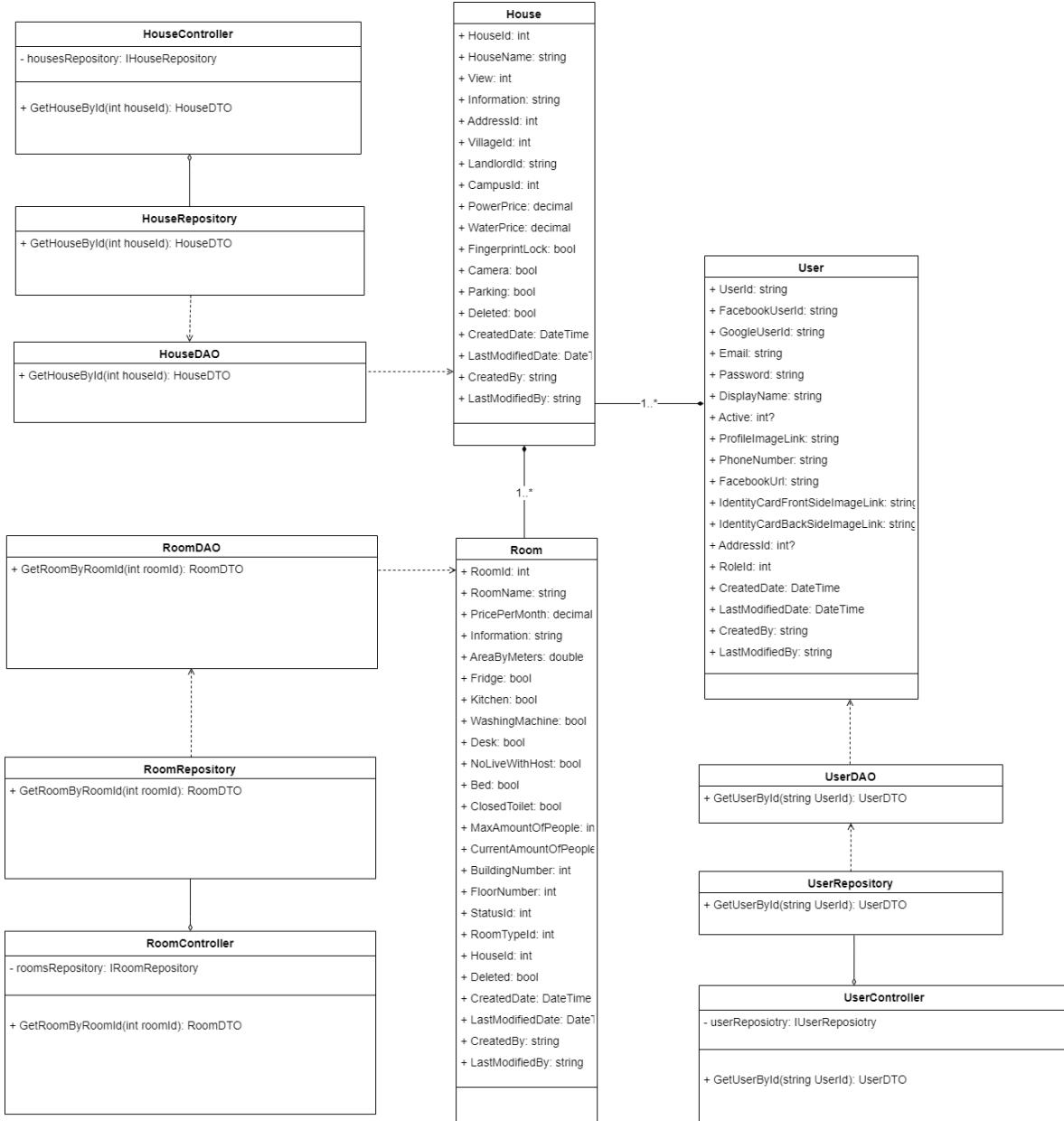
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

4.14.3 Sequence Diagram



4.15. Staff - Room Detail

4.15.1 Class Diagram



4.15.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage

05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

RoomStatusDAO Class

This class performs action to data (CRUD) in RoomType Table in Database

Class Methods

No	Method	Description
01	GetAllListStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomStatusController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAllStatus()	Return list of all RoomStatus in the Database
02	GetStatusesByHouseId()	Get all RoomStatus that this house has

RoomDAO Class

This class performs action to data (CRUD) in Room Table in Database

Class Methods

No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database

08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

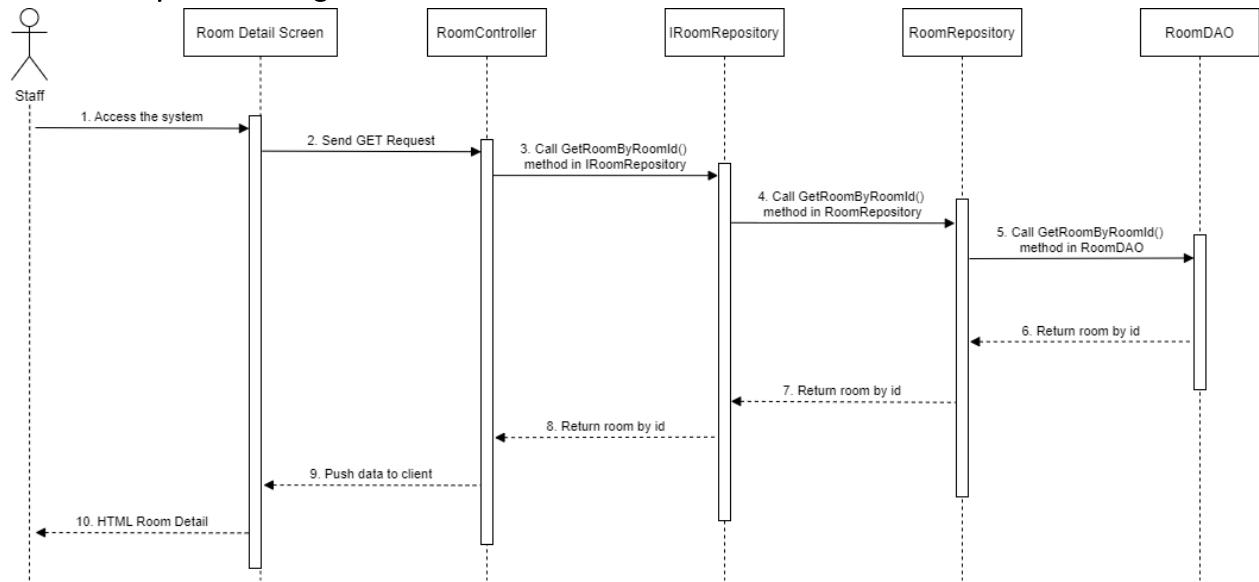
RoomController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

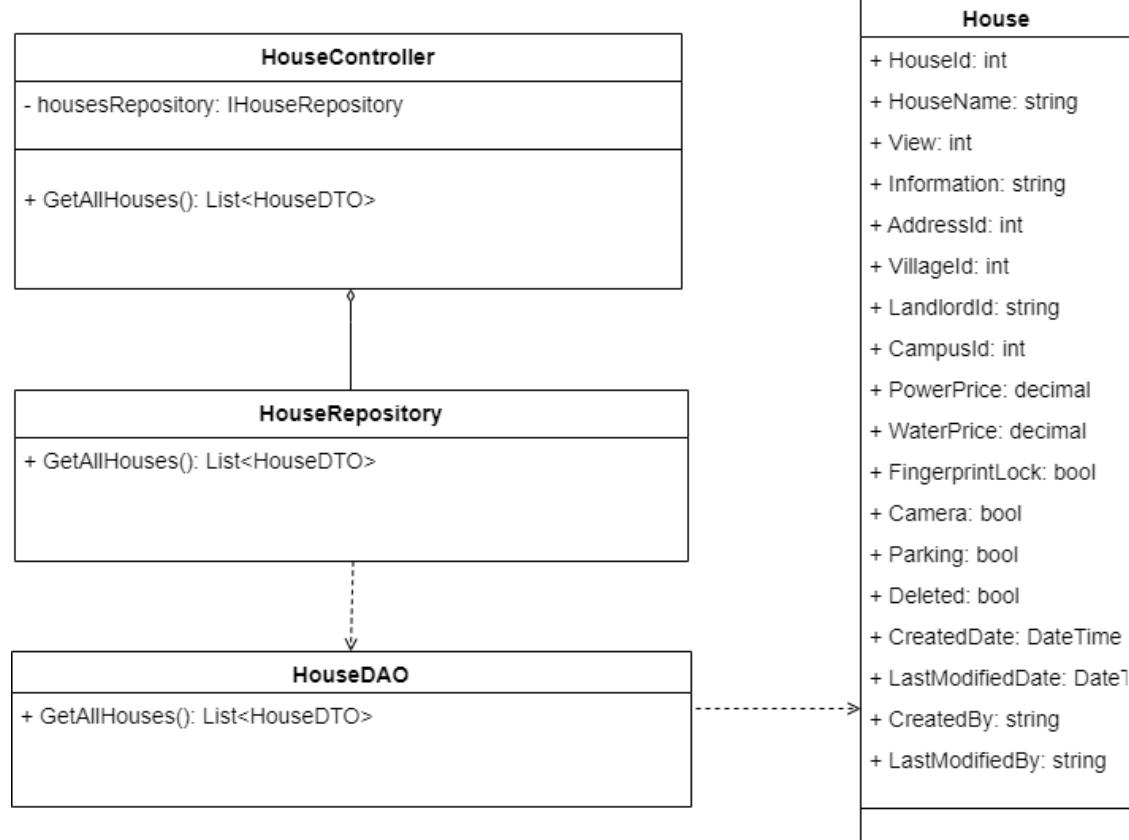
No	Method	Description
01	GetRoomsByHouseId()	Return list of Rooms of 1 House
02	GetAvailableRoomsByHouseId()	Get list of available Rooms (partially or totally) of 1 House
03	GetRoomPriceByHouse()	Get Lowest room price and Highest room price for of 1 House (used in HouseDAO)
04	CreateRoom()	Add a new Room to the Database
05	CreateRooms()	Add a list of Rooms of 1 House to the Database
06	UpdateRoomByRoomId()	Update detail information of 1 Room
07	DeleteRoom()	Delete a Room from a system without removing its data from the Database
08	GetRoomByRoomId()	Find a Room by its Id
09	CountAvailableRoom()	Count the sum of all available Rooms (partially or totally) of all Houses in the system
10	CountTotalRoom()	Count the sum of all Rooms (available or not) of all Houses in the system
11	CountAvailableCapacity()	Count the sum of Capacity of all Rooms of all Houses in the system
12	CountAvailableRoomByHouseId()	Count the sum of all available Rooms (partially or totally) of 1 House
13	CountAvailableCapacityByHouseId()	Count the sum of Capacity of all Rooms of 1 House
14	CountTotallyAvailableRoomByHouseId()	Count the sum of all available Rooms (totally only) of 1 House
15	CountPartiallyAvailableRoomByHouseId()	Count the sum of all available Rooms (partially only) of 1 House
16	ChangeStatusRoom()	Change Status (Available or Occupied) of 1 Room

4.15.3 Sequence Diagram



4.16. Staff – List House

4.16.1 Class Diagram



4.16.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

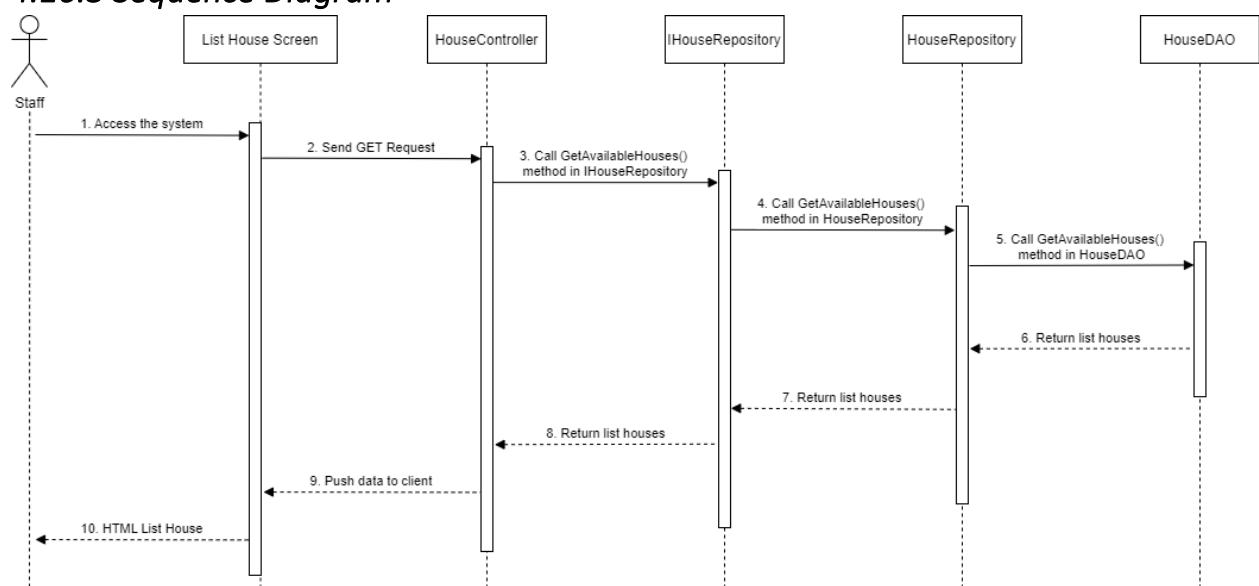
HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

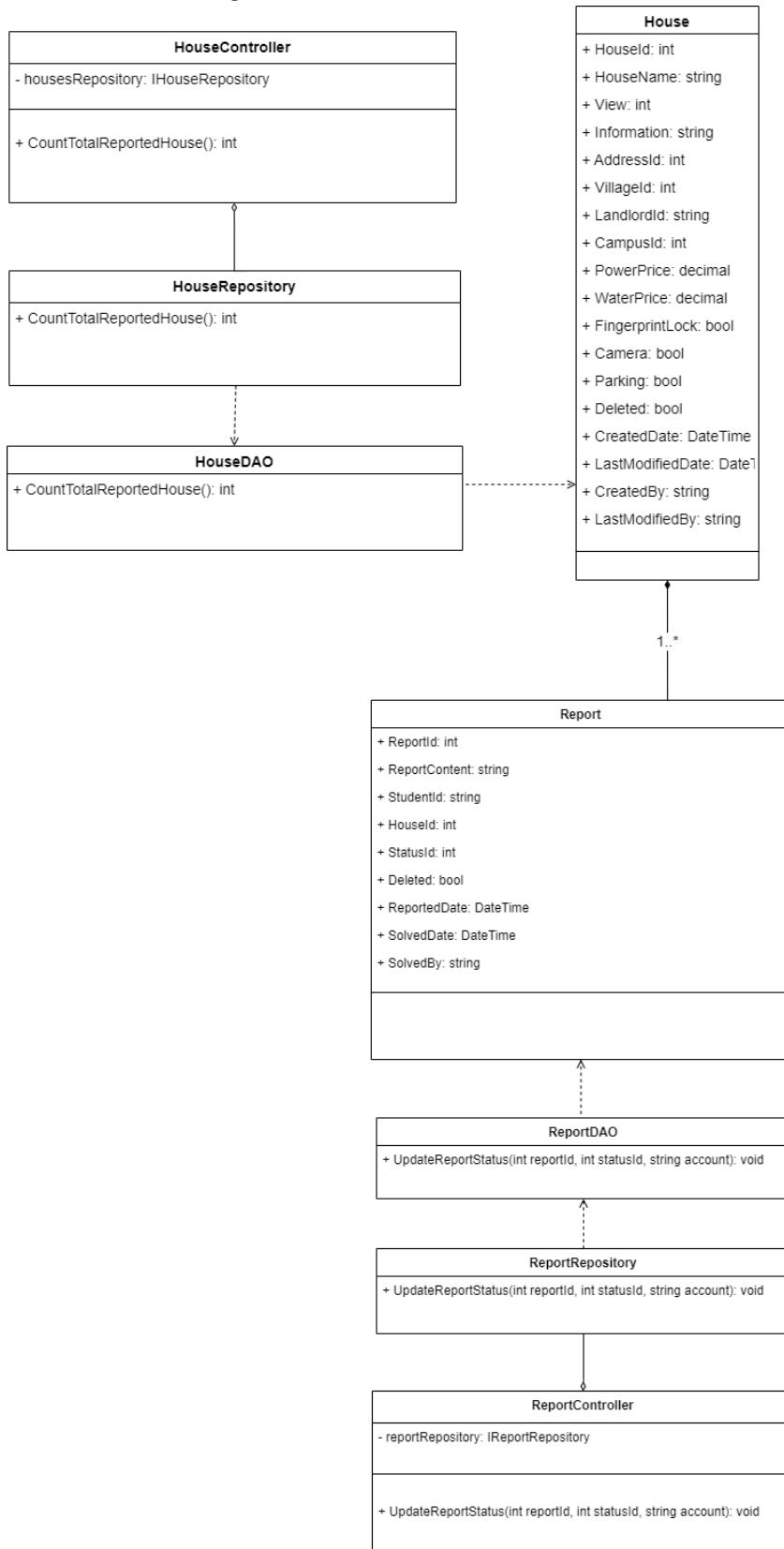
No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

4.16.3 Sequence Diagram



4.17. Staff – List Report

4.17.1 Class Diagram



4.17.2 Class Specifications

HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system

ReportDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	AddReport()	Add new Report to the database
02	GetTotalReportByMonth ()	Count the total of Report by month
03	GetReportByHouseId()	Get list of Reports by house id
04	SearchReportByName()	Search Report by house's name
05	CountTotalReportByHouseId() ()	Count total Report by house id
06	GetAllReports()	Get all Reports
07	CountTotalReport()	Count total Report

ReportController Class

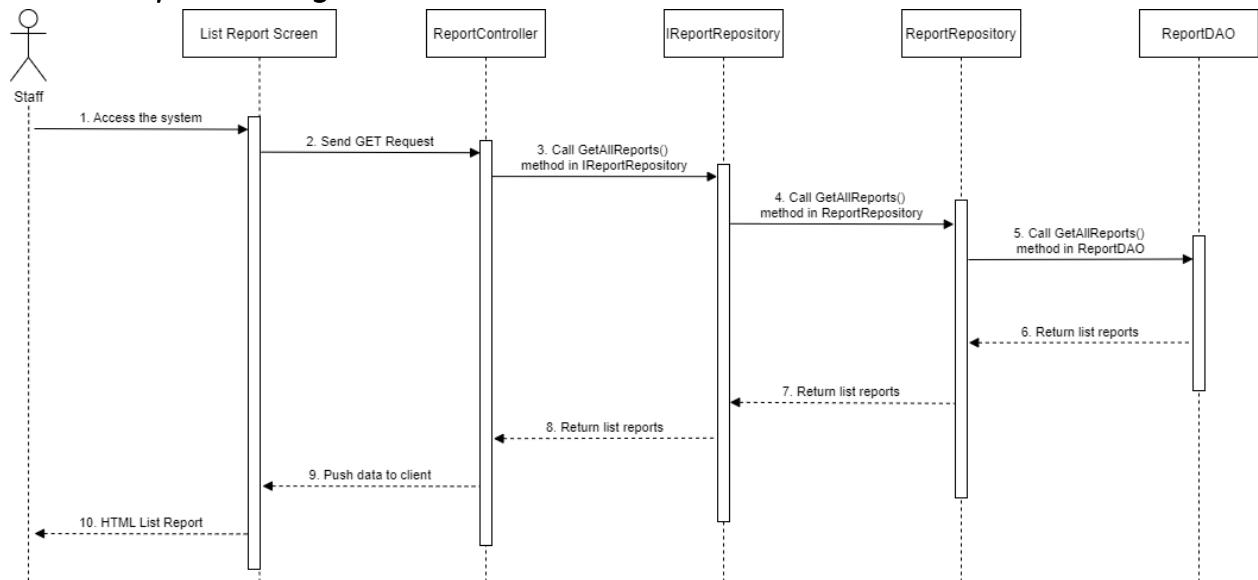
This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description

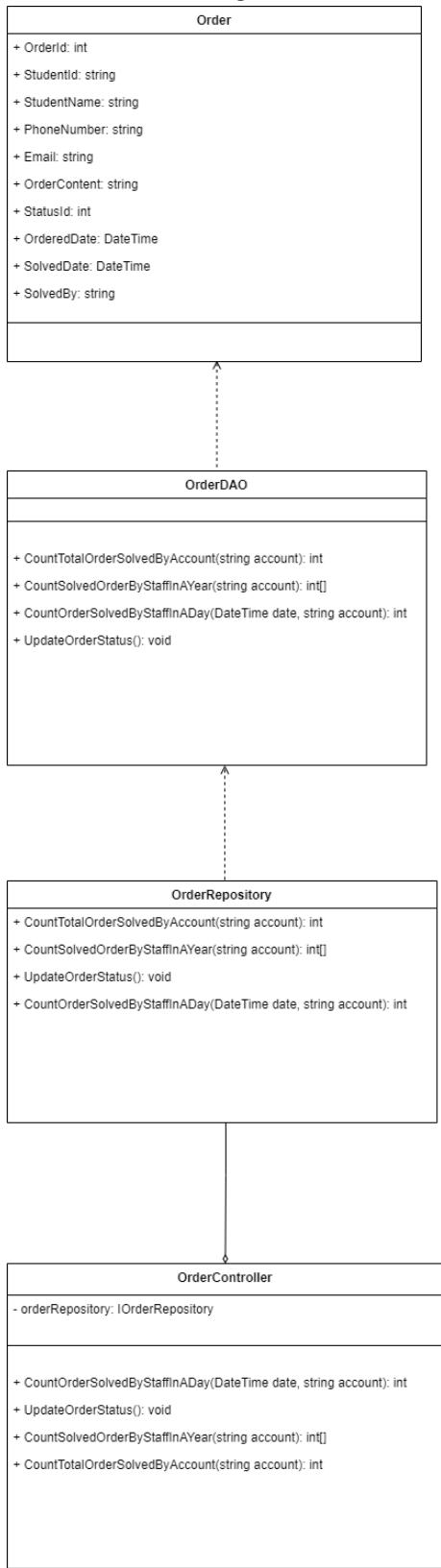
01	AddReport()	Add new Report to the database
02	GetTotalReportByMonth ()	Count the total of Report by month
03	GetReportByHouseId()	Get list of Reports by house id
04	SearchReportByName()	Search Report by house's name
05	CountTotalReportByHouseId() ()	Count total Report by house id
06	GetAllReports()	Get all Reports
07	CountTotalReport()	Count total Report

4.17.3 Sequence Diagram



4.18. Staff – List Order

4.18.1 Class Diagram



4.18.2 Class Specifications

OrderDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetTotalOrderCurrentMonth() ()	Get total order of the current month
02	GetAllOrders()	Get list of Orders in the database
03	CountTotalOrder()	Count total Orders
04	GetTotalOrderByMonth()	Get total Orders of the month
05	GetSolvedOrderByMonth()	Calculate number of solved Order by month
06	OrderByMonth()	Count Orders by month

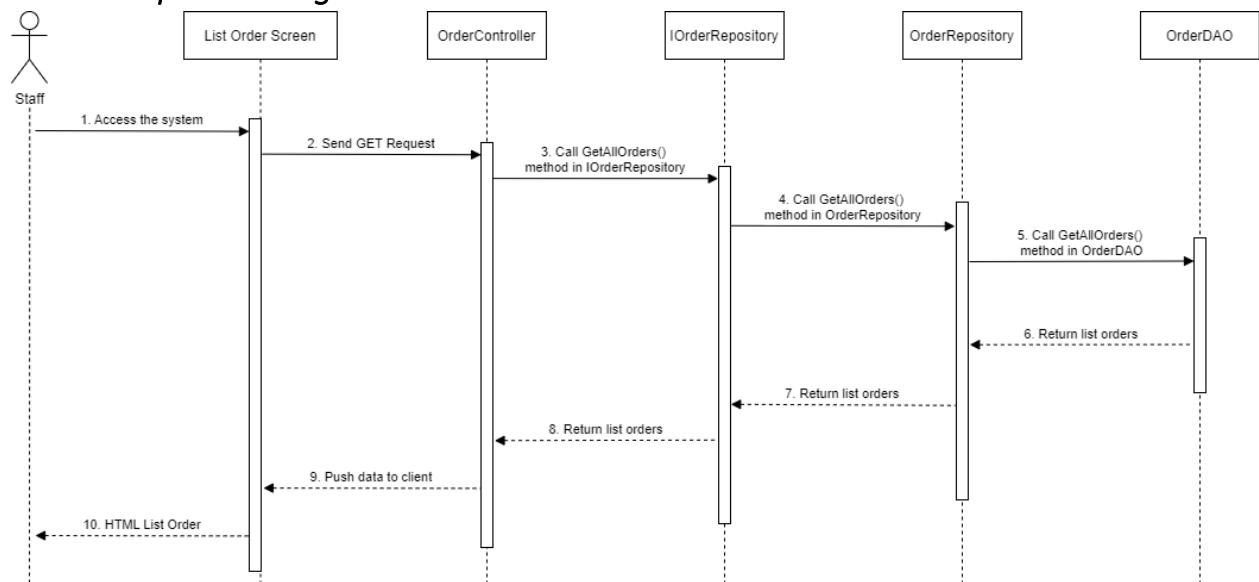
OrderController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

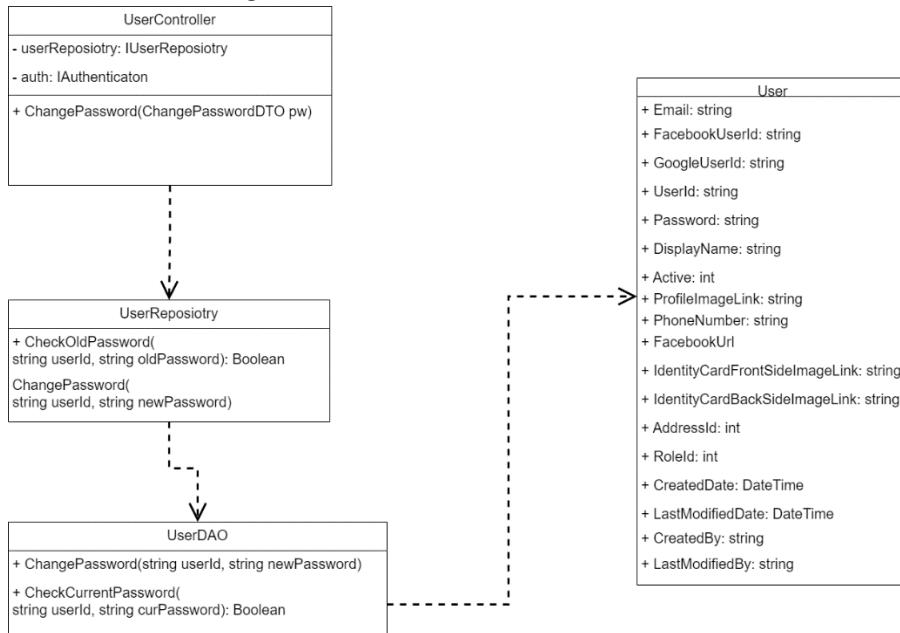
No	Method	Description
01	GetTotalOrderCurrentMonth() ()	Get total order of the current month
02	GetAllOrders()	Get list of Orders in the database
03	CountTotalOrder()	Count total Orders
04	GetTotalOrderByMonth()	Get total Orders of the month
05	GetSolvedOrderByMonth()	Calculate number of solved Order by month
06	OrderByMonth()	Count Orders by month

4.18.3 Sequence Diagram



4.19. Staff – Change Password

4.19.1 Class Diagram



4.19.2 Class Specifications

UserController Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account
07	ChangePassword()	Staff change password

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account
07	ChangePassword()	Staff change password

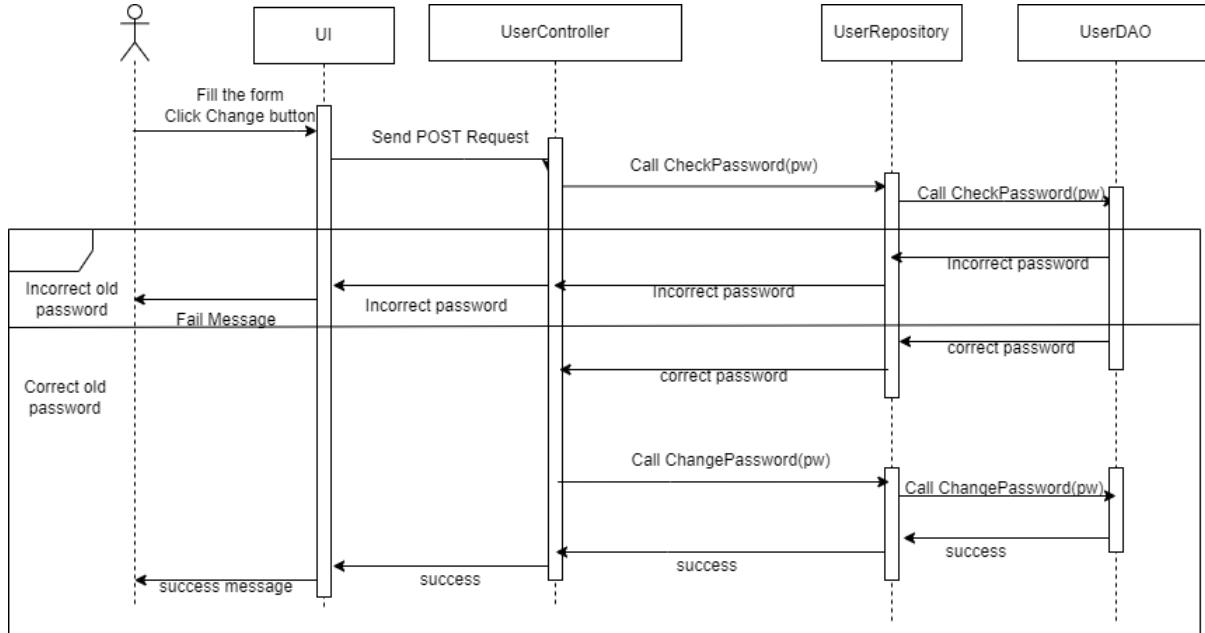
UserRepository Class

This class performs logic and get data from DAO to data (CRUD) in User Table in Database

Class Methods

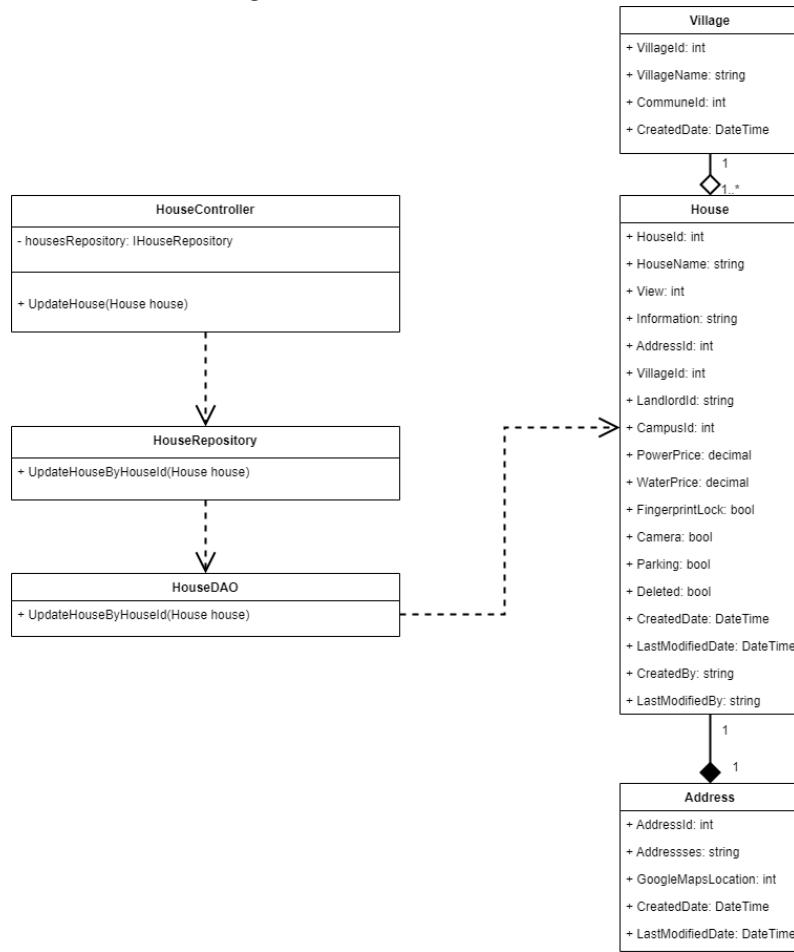
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	Login()	Get user with login data provided
06	Register()	User create an account
07	ChangePassword()	Staff change password

4.19.3 Sequence Diagram



4.20. Landlord – Update House

4.20.1 Class Diagram



4.20.2 Class Specifications

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system
08	UpdateHouse()	Update House data

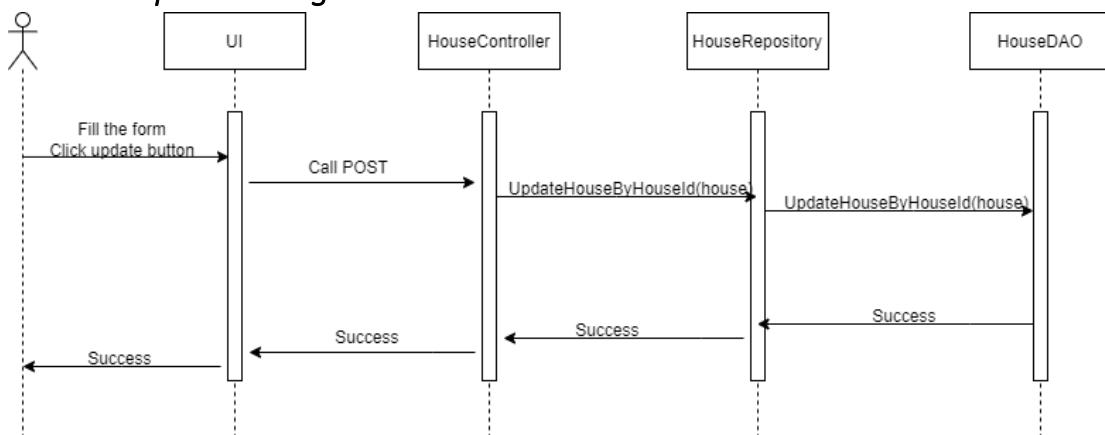
HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

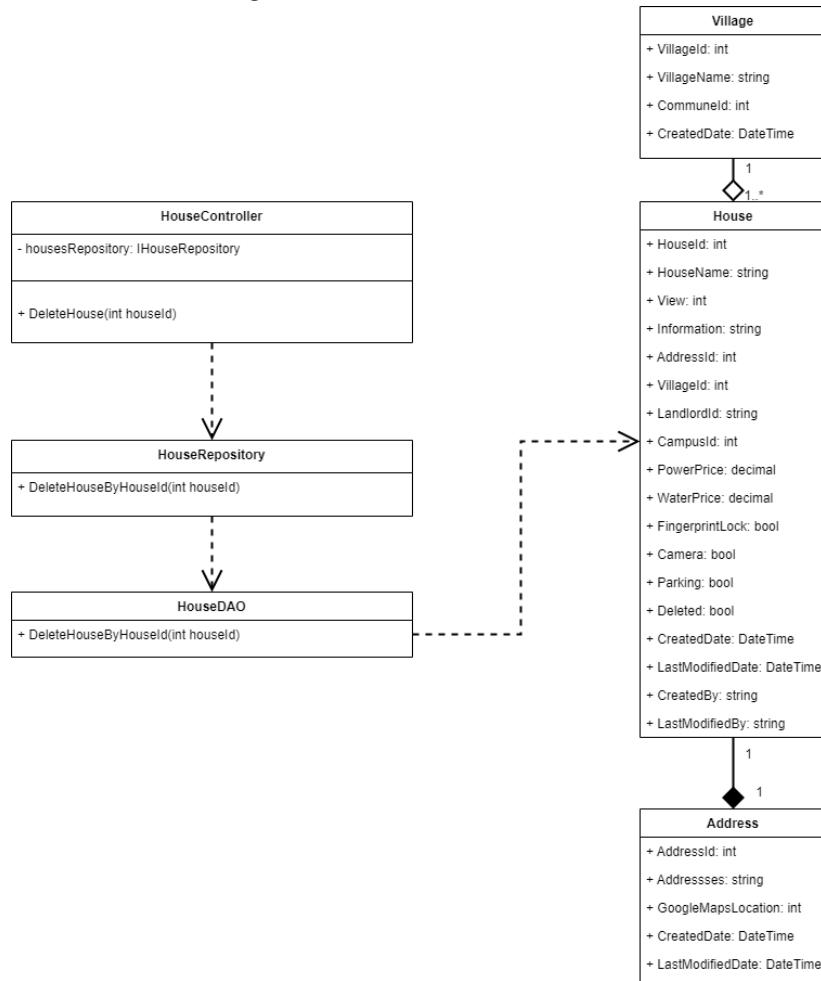
No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system
08	UpdateHouseByHouseId()	Update house data to database

4.20.3 Sequence Diagram



4.21. Landlord – Delete House

4.21.1 Class Diagram



4.21.2 Class Specifications

HouseController Class

This class takes data from DAO (Repository) and push it to API

Class Methods

No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system
08	UpdateHouse()	Update House data

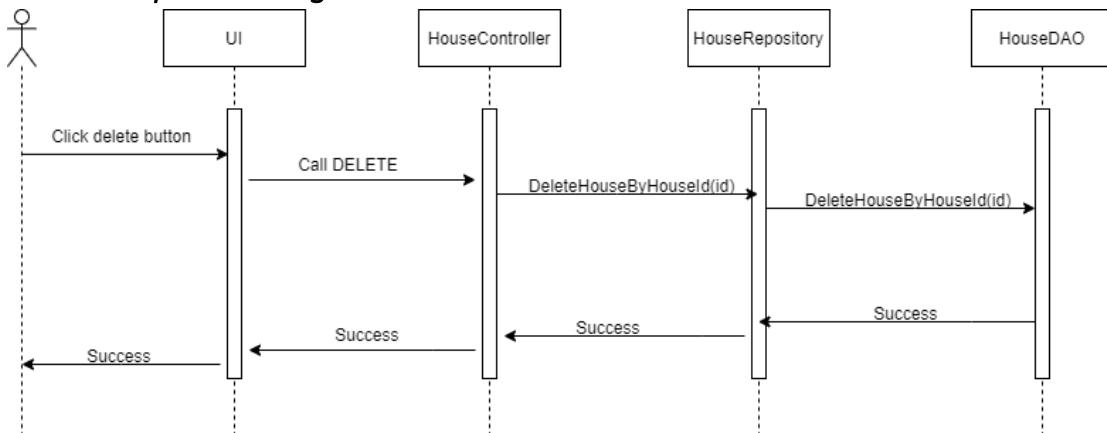
HouseDAO Class

This class performs action to data (CRUD) in House Table in Database

Class Methods

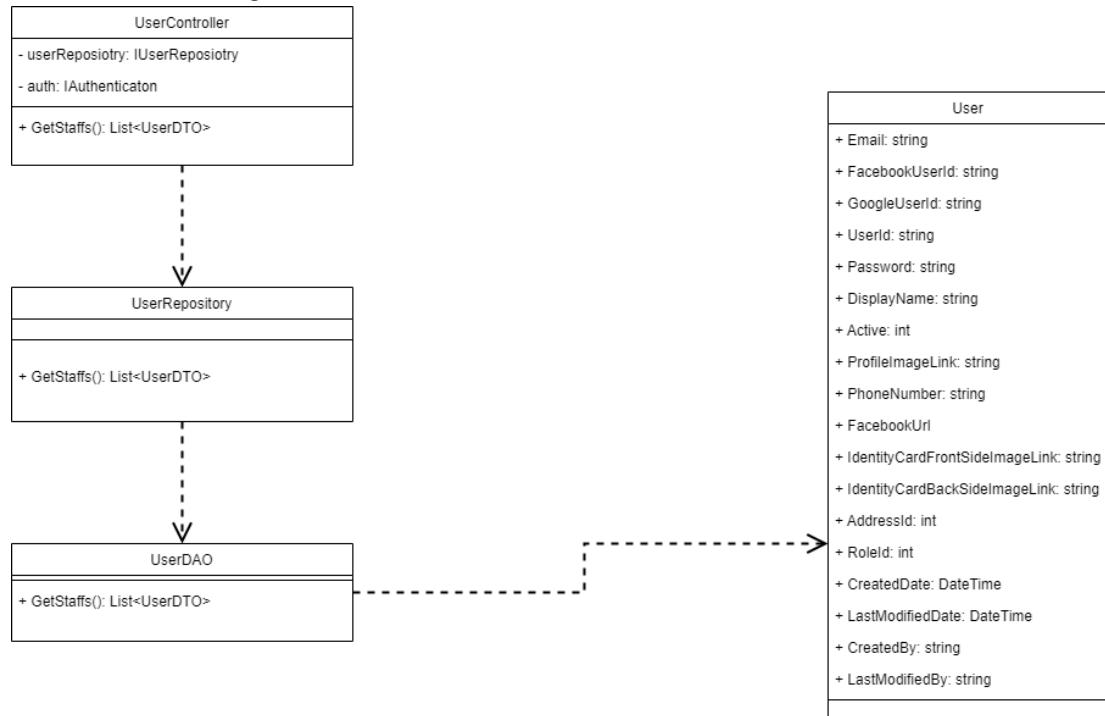
No	Method	Description
01	GetAvailableHouses()	Return list of all available Houses in the system (Houses that have available Rooms) with additional information to support filtering
02	CreateHouse()	Add a new House into the Database
03	GetHouseById()	Find detail information of a House by its Id
04	GetListHousesByLandlordId()	Get list of Houses of 1 Landlord for him to manage
05	GetMoneyForNotRentedRooms()	Get total amount of money of rooms that has not been rented (of 1 House)
06	CountTotalHouse()	Count total number of Houses in the system
07	CountAvailableHouse()	Count number of available Houses in the system
08	UpdateHouseByHouseId()	Update house data to database

4.21.3 Sequence Diagram



4.22. Admin – List Staff Account

4.22.1 Class Diagram



4.22.2 Class Specification

UserController Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account
07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account

07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

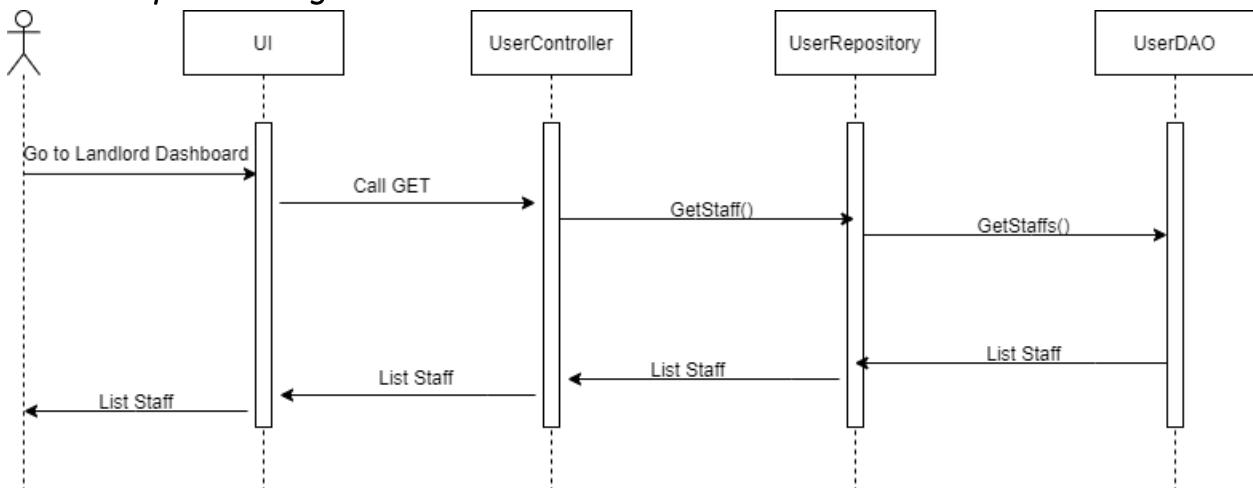
UserRepository Class

This class performs logic and get data from DAO to data (CRUD) in User Table in Database

Class Methods

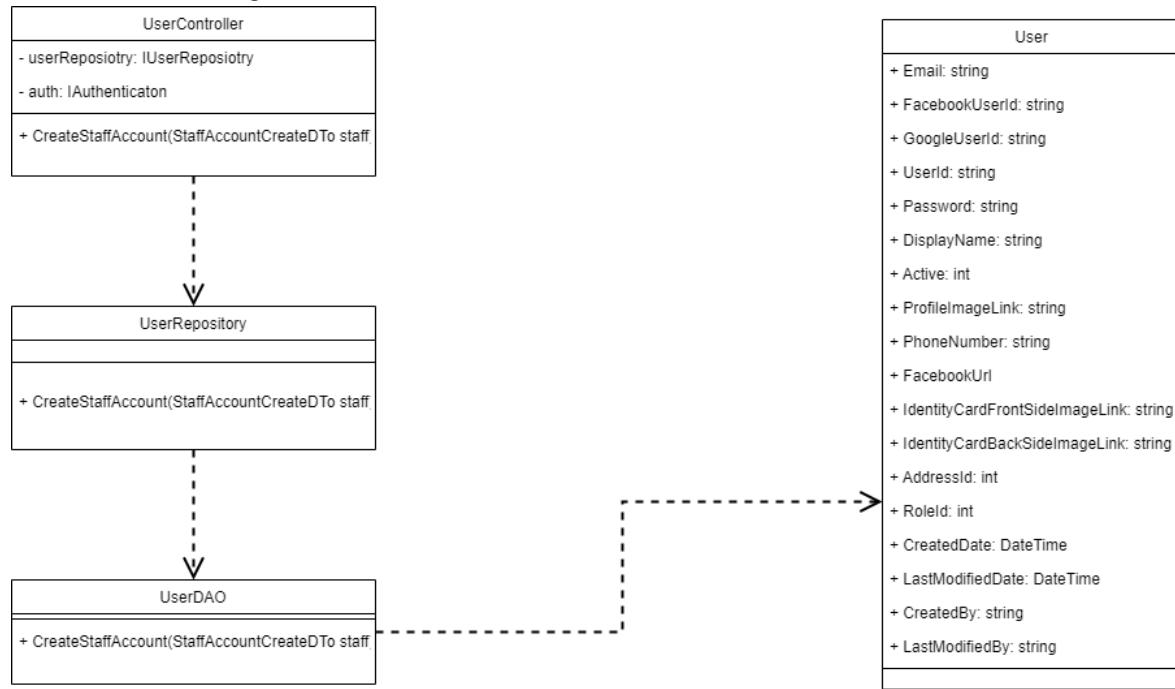
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	Login()	Get user with login data provided
03	Register()	User create an account
04	GetStaffs()	Get Staff List
05	CreateStaffAccount()	Admin add staff account to database
06	UpdateStaffAccount()	Admin Update Staff account
07	DeleteStaffAccount()	Admin remove staff account

4.22.3 Sequence Diagram



4.23. Admin – Add Staff Account

4.23.1 Class Diagram



4.23.2 Class Specification

UserController Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	GetUserById()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle()	User login with google in the system
06	Register()	User create an account
07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle()	User login with google in the system

06	Register()	User create an account
07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

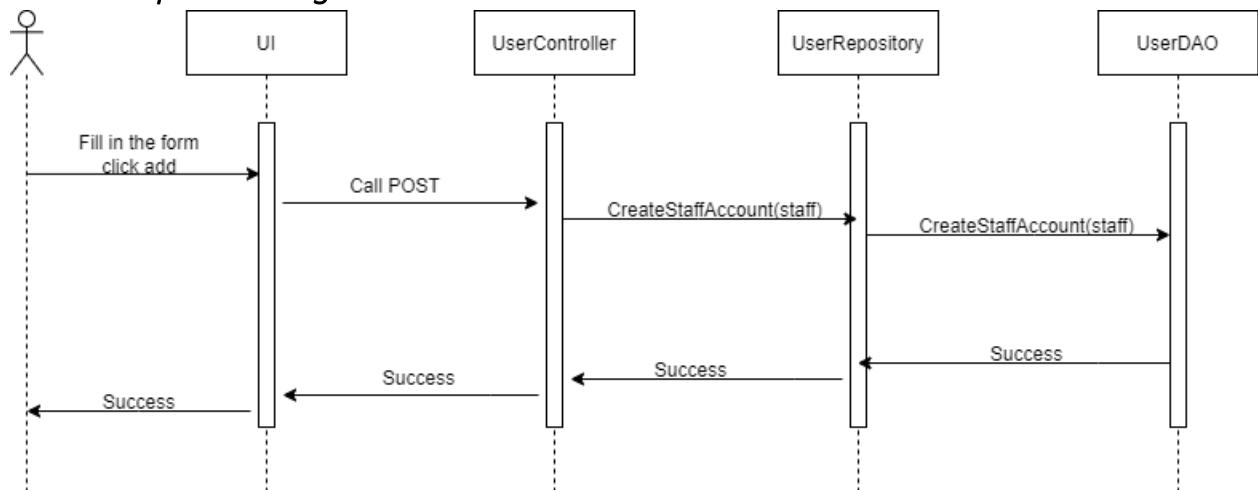
UserRepository Class

This class performs logic and get data from DAO to data (CRUD) in User Table in Database

Class Methods

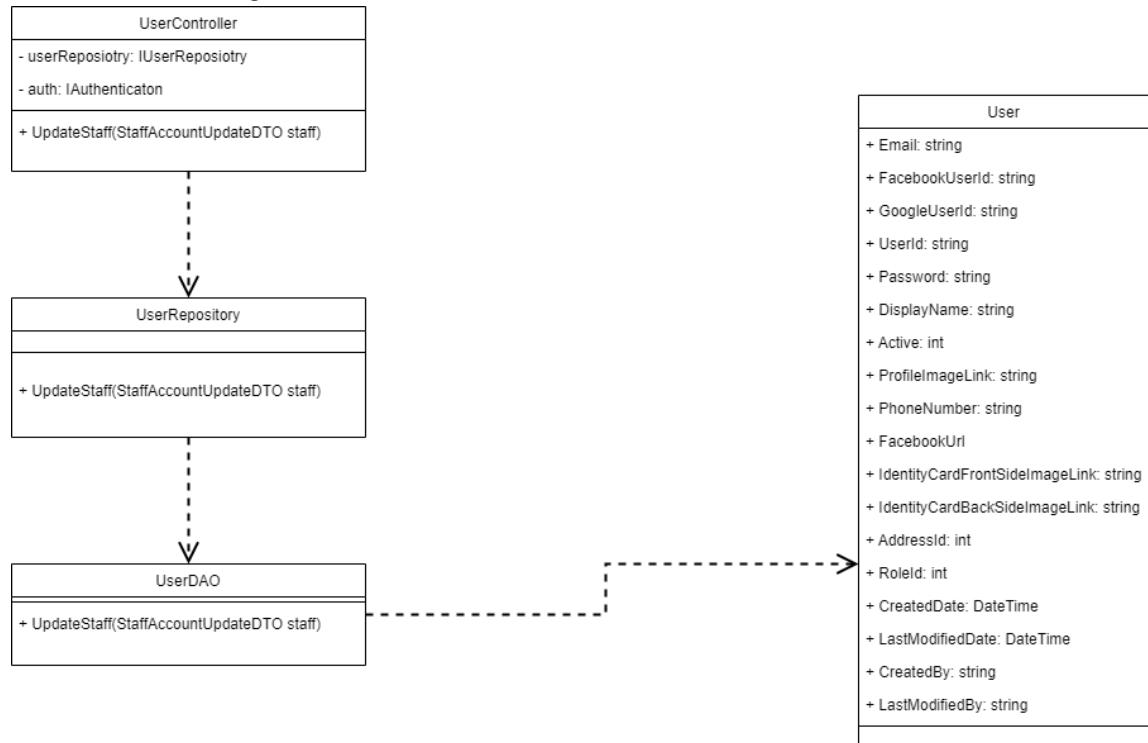
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	Login()	Get user with login data provided
03	Register()	User create an account
04	GetStaffs()	Get Staff List
05	CreateStaffAccount()	Admin add staff account to database
06	UpdateStaffAccount()	Admin Update Staff account
07	DeleteStaffAccount()	Admin remove staff account

4.23.3 Sequence Diagram



4.24. Admin – Edit Staff Account

4.24.1 Class Diagram



4.24.2 Class Specification

`UserController` Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	<code> GetUserById ()</code>	Find detail information of User by its Id
02	<code> GetLandlords()</code>	Get list of Landlords in the database
03	<code> LoginUsername()</code>	User login in the system
04	<code> LoginFacebook()</code>	User login with facebook in the system
05	<code> LoginGoogle ()</code>	User login with google in the system
06	<code> Register()</code>	User create an account
07	<code> GetStaffs()</code>	Get Staff List
08	<code> CreateStaffAccount()</code>	Admin add staff account to database
09	<code> UpdateStaffAccount()</code>	Admin Update Staff account
10	<code> DeleteStaffAccount()</code>	Admin remove staff account

`UserDAO` Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	<code> GetUserById ()</code>	Find detail information of User by its Id
02	<code> GetLandlords()</code>	Get list of Landlords in the database
03	<code> LoginUsername()</code>	User login in the system
04	<code> LoginFacebook()</code>	User login with facebook in the system

05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account
07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

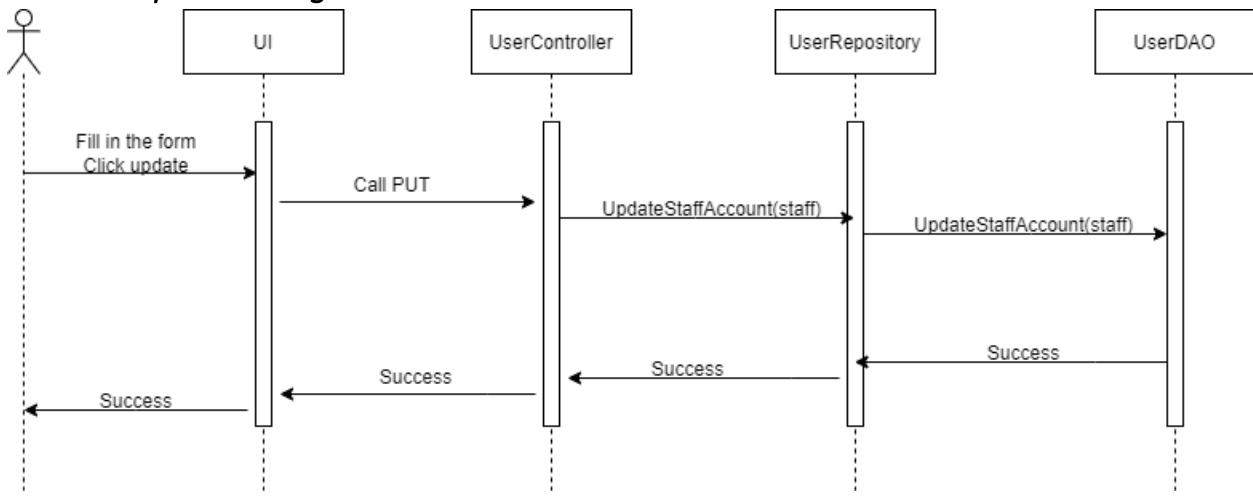
UserRepository Class

This class performs logic and get data from DAO to data (CRUD) in User Table in Database

Class Methods

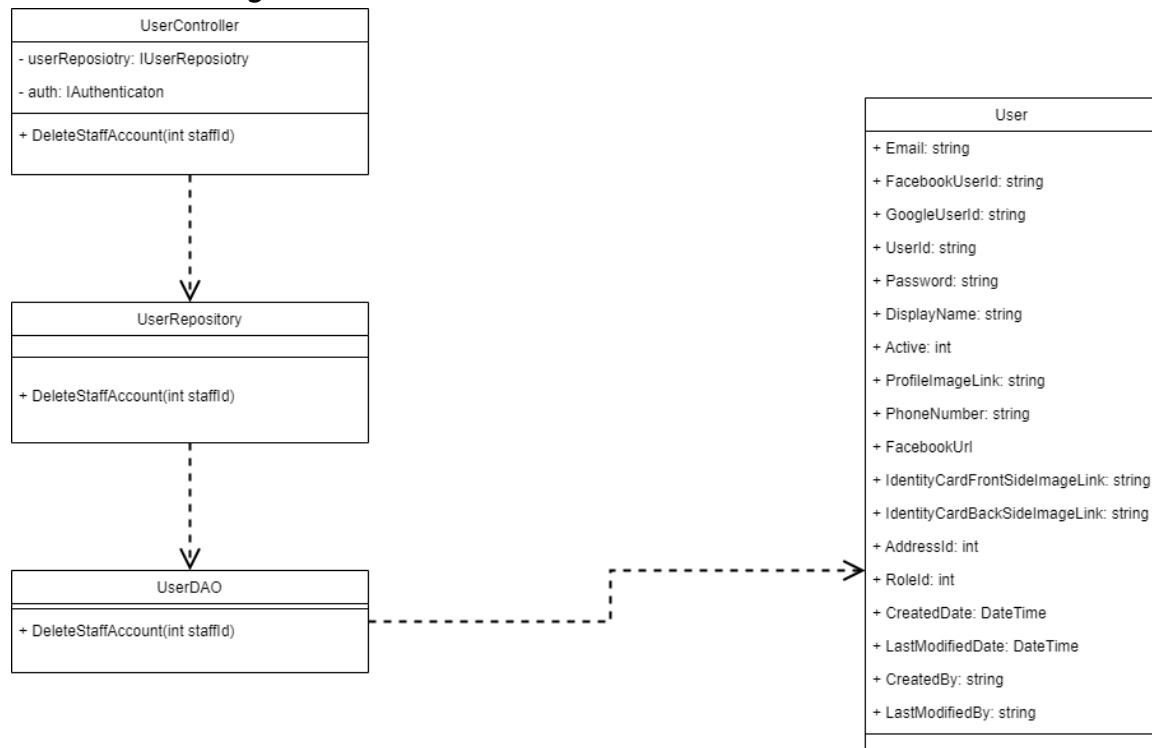
No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	Login()	Get user with login data provided
03	Register()	User create an account
04	GetStaffs()	Get Staff List
05	CreateStaffAccount()	Admin add staff account to database
06	UpdateStaffAccount()	Admin Update Staff account
07	DeleteStaffAccount()	Admin remove staff account

4.24.3 Sequence Diagram



4.25. Admin – Delete Staff Account

4.25.1 Class Diagram



4.25.2 Class Specification

UserController Class

This class takes request, get data from Repository and return Response

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system
06	Register()	User create an account
07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

UserDAO Class

This class performs action to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	GetLandlords()	Get list of Landlords in the database
03	LoginUsername()	User login in the system
04	LoginFacebook()	User login with facebook in the system
05	LoginGoogle ()	User login with google in the system

06	Register()	User create an account
07	GetStaffs()	Get Staff List
08	CreateStaffAccount()	Admin add staff account to database
09	UpdateStaffAccount()	Admin Update Staff account
10	DeleteStaffAccount()	Admin remove staff account

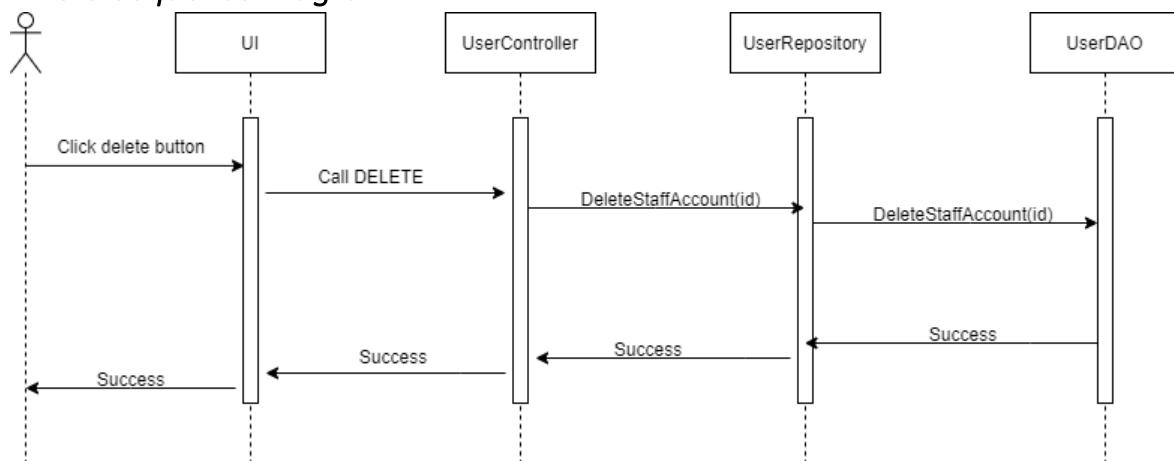
UserRepository Class

This class performs logic and get data from DAO to data (CRUD) in User Table in Database

Class Methods

No	Method	Description
01	GetUserById ()	Find detail information of User by its Id
02	Login()	Get user with login data provided
03	Register()	User create an account
04	GetStaffs()	Get Staff List
05	CreateStaffAccount()	Admin add staff account to database
06	UpdateStaffAccount()	Admin Update Staff account
07	DeleteStaffAccount()	Admin remove staff account

4.25.3 Sequence Diagram



V. Software Testing Documentation

1. Purpose

This chapter's main goal is to identify and prevent defects that developers might introduce during software development that could cause software failures. To ensure that the final product satisfies user requirements and the business, this chapter also aims to provide information on the level of quality and increase confidence in the system. The following sections are included:

- Scope of Testing
- Test Plan
- Test Cases
- Test Report

1.1. Scope of Testing

1.1.1. Testing Phases

There are 4 phases in Testing Process: Unit Testing, Integration Testing, System Testing and Acceptance Testing.

No.	Testing Phase	Description
1	Unit Testing	The main characteristic of this phase is that the software components are tested individually and isolated from all other software components of the system. The isolation is necessary to prevent external influences on component. In this phase, a unit is refer to a function, and a White-box Testing method is used. One of the biggest benefits of this testing phase is that it can be run every time a piece of code is changed, allowing issues to be resolved as quickly as possible.
2	Integration Testing	The purpose of this phase is to make sure all components collaborate correctly. Thus, the goal of Integration test is to expose faults in the interface and in the interaction between integrated components. This is particularly beneficial because it determines how efficiently the units are running together.
3	System Testing	System testing is the first level in which the complete application is tested as a whole. The goal at this level is to check whether the system has complied with all of the specified functional and non-functional requirements and to see that it meets Quality Standards.
4	Acceptance Testing	To ascertain whether the system is ready for release, Acceptance testing, also known as User Acceptance Testing, is carried out. To determine whether the application satisfies the business' needs, the tester and a few alpha test users will test the system during this phase.

Figure 5-1: Testing phases

1.1.2. Testing Types

- Unit Testing
- Regression Testing
- API Testing
- Integration Testing
- System Testing
- Acceptance Testing

1.1.3. Testing Range

Team performs all functions defined in the SRS based on the approved version.

2. Test Plan

2.1. Testing tools and environment

2.1.1. Testing tools

- **Chrome DevTools:** To inspect elements, view logs, network traffic and storage
 - Google Chrome Version: 108.0.5359.125



Figure 5-1: Chrome Developer Tools

- **Postman:** To manage list of all APIs and manually test API
 - Version: 10.6.0



POSTMAN

Figure 5-2: Postman

- **NUnit:** To perform Unit Testing for .NET code
 - Version: 3.13.1



Figure 5-3: NUnit

- **FluentAssertions:** Is a assertion library for NUnit, to help assertion result in a test
 - Version: 5.10.3



Figure 5-4: FluentAssertions

- **Swagger:** Is a suite of tools for testing APIs, comes as a default with ASP.NET Web API Project
 - Version: 5.6.3



Figure 5-5: Swagger

- **Microsoft Excel:** To record and manage test cases
 - Office Version: Microsoft Office Professional Plus 2016



Figure 5-6: Microsoft Excel

- **GitHub Issues:** To log defects and assign fixer

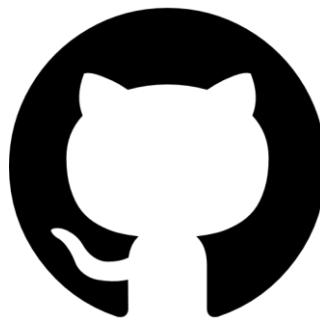


Figure 5-7: GitHub

2.1.2. Testing environment

Type of testing	Software	Hardware
Integration Test, System Test and Acceptance Test	<ul style="list-style-type: none">- Microsoft Excel 2016- Microsoft Word 2016- Google Chrome version 108.0.5359.125- GitHub Issues	Personal computer for developing with the minimum configuration:
Unit Test and API Test	<ul style="list-style-type: none">- Postman version 10.6.0- NUnit version 3.13.1- FluentAssertions version 5.10.3- Coverlet.collection version 3.0.2- Swagger version 5.6.3- GitHub Issues	<ul style="list-style-type: none">- OS: Windows 10 Home 64-bit- CPU: Intel® Core™ i5- RAM: 8.00GB

Table 5-2: Testing environment

2.2. Human Resources and Responsibilities

No.	Worker	Role	Responsibilities
1	Nguyễn Thế Giang	Project Manager Test Leader	<ul style="list-style-type: none">• Create Test Plan• Control testing activities• Review Test Cases and Reports
2	Nguyễn Thế Giang Bùi Ngọc Huyền Phùng Quang Thông Nguyễn Trí Kiên Nguyễn Thu An	Tester	<ul style="list-style-type: none">• Create and perform Integration Test and System Test• Create Test Report• Log defects
3	Nguyễn Thế Giang Bùi Ngọc Huyền Phùng Quang Thông Nguyễn Trí Kiên Nguyễn Thu An	Developer	<ul style="list-style-type: none">• Create and perform Unit Test• Fix bugs

Table 5-3: Human resources & responsibilities

2.3. Test Strategy

2.3.1. Testing Model

Overall, FHF project uses the Iterative and Incremental Software Process Model, FHF is divided into 2 sub-systems: FHF Back-end and FHF Front-end. With each sub-system, we use specific testing types to fit the requirement, the characteristic of the system and the human resource of the project team.

FHF Back-end has 2 levels of test:

- Unit testing: Automation tests that cover logic of Controller files.
- API testing: Manual tests that involve testing APIs directly (in isolation) to determine whether APIs return the correct response (in the expected format) for a broad range of feasible requests, react properly to edge cases such as failures and unexpected/extreme inputs.

FHF Front-end works mostly with GUI instead of logic and it depends on FHF Back-end, so that FHF Front-end implements Integration testing and System testing which covers the whole system.

2.3.2. Testing Types

Testing FHF Project will be carried out at the end of each Iteration as defined in project plan and will depend on internal delivery by development side. Each type of FHF Project Testing will be designated to cover a level of project, from component level to system level with Unit testing combined with API testing, Integration testing, System testing and Acceptance testing. On the other hand, FHF will use several testing types to prevent defects from code modifications and latent bugs.

The different types of testing that will be carried out this project are:

- **Unit Testing:**
 - Testing all individual implemented methods, functions of Controller files.
 - Unit test also includes database testing to verify constraint, transaction, default value, data types, data format which are mentioned in database design and software requirement.
 - Test case will have to cover all logic branch that function or method could execute with difference data input. Another alternative logic branch should be covered if not, that logic branch should be detected at API testing level.
 - Implemented function's error message and database error message will be included in this test.
 - It is a kind of white-box testing.
- **Regression Testing:**
 - This testing type is used to confirm that a recent code change has not adversely affected existing features. This testing is done to ensure that new code changes do not have side effects on the existing functionalities.
 - Regression Testing in FHF Back-end is executed as automation test of Unit test cases. Whenever a developer makes a change, he will run the automation test project to make sure all test cases still pass.
 - It is a kind of white-box testing.
- **API Testing:**

- Involves testing APIs directly to determine if they meet expectations for functionality, reliability, performance and security. API testing will test all of individual implemented API of FHF Back-end.
 - Testing will be performed by simulating FHF Front-end's call to Back-end by using Postman.
 - Test case will verify constraint of data which be mention in Business rule.
 - It is a kind of white-box testing.
- **Integration Testing:**
 - Integration Testing is defined as a type of testing where software modules are integrated logically and tested as a group. The purpose of this level of testing is to expose defects in the interaction between these software modules when they are integrated. Integration Testing focuses on checking data communication amongst these modules.
 - Integration Testing is carried out by performing test manually through interacting with the Front-end. Test cases will then be recorded, arranged and stored for further reference.
 - It is a kind of white-box testing.
- **System Testing:**
 - This testing type validates the complete and fully integrated software product. The purpose of it is to evaluate the end-to-end system specifications.
 - System Testing is performed manually by going through all the system, testing interaction with the User Interface like an actual user. Similar to Integration Testing, Test cases will be recorded and stored for further reference.
 - It is a kind of black-box testing.
- **Acceptance Testing:**
 - This test type will be executed by tester with designed test cases. Acceptance Test is a test type conducted to determine if the requirements of a specification or contract are met.
 - It also includes Alpha Testing. Alpha Testing takes place at close relation user's site and are free test to detect bug and strange behavior. By that, development team will improve UX and UI of system
 - It is a kind of black-box testing.

2.3.3. Testing Phases

Table below are the stages in which common tests are executed:

Testing Type	Testing Phase			
	Unit Testing	Integration Testing	System Testing	Acceptance Testing
Unit Testing	x			
Regression Testing	x	x	x	
API Testing	x	x		
Integration Testing		x	x	
System Testing			x	x
Acceptance Testing				x

Table 5-4: Testing Phase

2.3.4. Test Schedule

Table below is the Test Schedule for FHF Project:

Test Schedule	Start Date	End Date
<i>Phase 1: Authentication, Guest's and Landlord's related features</i>		
Unit Testing	26/09/2022	30/10/2022
Regression Testing	10/10/2022	28/10/2022
API Testing	17/10/2022	30/10/2022
Integration Testing	26/09/2022	30/10/2022
<i>Phase 2: Staff's and Admin's related features</i>		
Unit Testing	31/10/2022	18/12/2022
Regression Testing	06/11/2022	10/12/2022
API Testing	31/10/2022	12/12/2022
Integration Testing	12/12/2022	12/12/2022
System Testing	13/12/2022	15/12/2022
Acceptance Testing	15/12/2022	18/12/2022

Table 5-5: Test Schedule

2.3.5. Deliverables

Table below is the Deliverables for FHF Project:

Deliverables	Responsibilities	Complete date
<i>Phase 1</i>		
Test Plan	Project Manager	26/09/2022
Test Cases	All members	22/10/2022
Test case review	Project Manager	26/10/2022
Defect report	All members	29/10/2022
<i>Phase 2</i>		
Test Plan	Project Manager	31/10/2022
Test Cases	All members	08/12/2022
Test case review	Project Manager	15/12/2022
Defect report	All members	17/12/2022
Final test Summary report	Project Manager	18/12/2022

Table 5-6: Deliverables schedule

2.4. Features to be tested

Actor	Name
Guest	Login
	Home Page
	House Detail
	Room Detail
Student	Create Order
	Send Report

Actor	Name
Landlord	Upload House Info
	Manage Houses
	Manage Rooms
Staff	Dashboard
	Manage Landlords
	List Houses
	Manage Reports
	Manage Orders
	Manage Sign up Requests
	Change Password
	Update Profile
	Admin
	Manage Accounts

Table 5-7: Features to be tested

3. Test Case

3.1. Automation Testing with Unit Testing

Unit Testing will be done by the developers and approved by team leader. The FHF development team embrace this feature to gain the following advantages:

- Reduce the level of bugs in production code.
- Save development time.
- Automation tests can be run as frequently as required.
- Can easily form a document from the tests.
- Easier to maintain than GUI tests which are difficult to maintain with the short release cycles and frequent changes and with a complex system
- Reduce cost of resource to corresponding GUI testing

3.1.1. Automation Testing Framework

- For Unit Testing, we use NUnit Testing Framework combined with FluentAssertion library and Coverlet for code coverage observation
- Testing Framework and related libraries will be added as a Test Project to FHF Back-end Solution

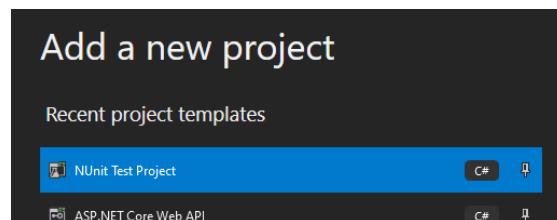


Figure 5-8: Create Test Project

- All automation test scripts will be created manually and saved to this Project as classes. Result of the test is stored in the **TestResults** directory and Code Coverage Report is stored in **coverageReport** directory.

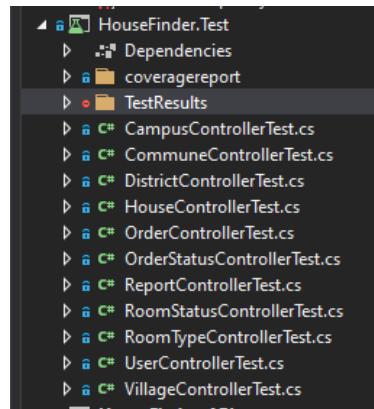


Figure 5-9: Test Project's directory structure

- Unit tests focus on individual functions in class and will be created as below:

```
/*
 * Method: GetAvailableHouses()
 * Scenario: None
 * Expected behavior: Returns ActionResult
 */
[Test]
public void GetAvailableHouses_Returns_ActionResult()
{
    //ARRANGE
    var houseController = new HouseController();

    //ACT
    var data = houseController.GetAvailableHouses();

    //ASSERT
    Assert.IsInstanceOf<ActionResult<IEnumerable<AvailableHouseDTO>>>(data);
}
```

Figure 5-10: Unit Test case sample

This `GetAvailableHouses()` function is used to retrieve all houses that are considered available. Test case will check if this function returns the correct data type.

- Test cases are executed automatically by clicking on “Run” in Test Explorer window in Visual Studio:

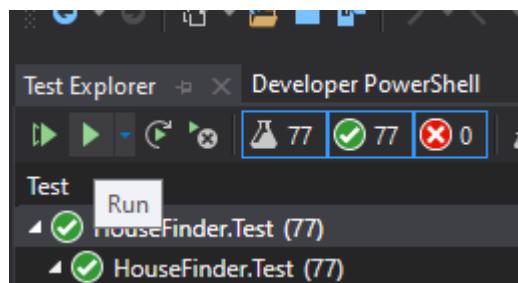


Figure 5-11: Run automation test in Visual Studio

- After running Test cases, test report will appear on the Test Explorer window:

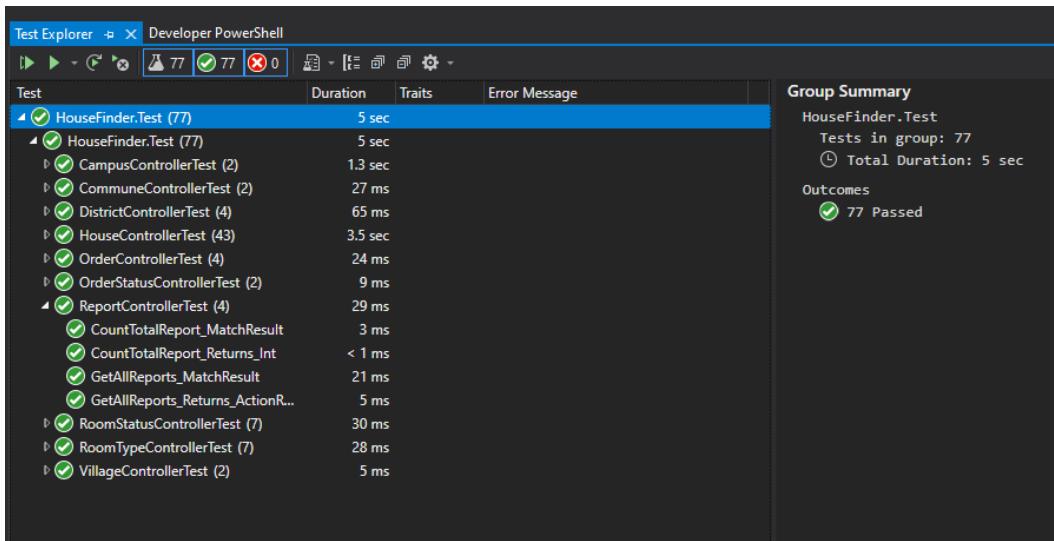


Figure 5-12: Sample Unit Test report

- By using coverlet library, Code Coverage report can be created and exported into static HTML or XML, using Terminal command:

```
Test Explorer Developer PowerShell x
+ Developer PowerShell - ⌂ ⌂ ⌂
*****
** Visual Studio 2019 Developer PowerShell v16.11.21
** Copyright (c) 2021 Microsoft Corporation
*****
PS D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder.Test> dotnet test --collect:"XPlat Code Coverage"
Determining projects to restore...
All projects are up-to-date for restore.
BusinessObjects -> D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\BusinessObjects\bin\Debug\net5.0\BusinessObjects.dll
DataAccess -> D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\DataAccess\bin\Debug\net5.0\DataAccess.dll
Repositories -> D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\Repositories\bin\Debug\net5.0\Repositories.dll
HouseFinder_API -> D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder_API\bin\Debug\net5.0\HouseFinder_API.dll
HouseFinder.Test -> D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder.Test\bin\Debug\net5.0\HouseFinder.Test.dll
Test run for D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder.Test\bin\Debug\net5.0\HouseFinder.Test.dll (.NETCoreApp, Version=v5.0)
Microsoft (R) Test Execution Command Line Tool Version 16.11.0
Copyright (c) Microsoft Corporation. All rights reserved.

Starting test execution, please wait...
A total of 1 test files matched the specified pattern.

Attachments:
D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder.Test\TestResults\c94c30ac-c198-4d12-9e84-18d3f628a1d9\coverage.cobertura.xml
Passed! - Failed: 0, Passed: 77, Skipped: 0, Total: 77, Duration: 4 s - HouseFinder.Test.dll (net5.0)
```

Figure 5-13: Using Terminal to perform Code Coverage and export to XML

```
PS D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder.Test> reportgenerator -reports:"TestResults\c94c30ac-c198-4d12-9e84-18d3f628a1d9\coverage.cobertura.xml" -targetdir:"coverageReport" -reporttypes:HTML  
2022-12-18T04:31:27: Arguments  
2022-12-18T04:31:27: -reports:TestResults\c94c30ac-c198-4d12-9e84-18d3f628a1d9\coverage.cobertura.xml  
2022-12-18T04:31:27: -targetdir:coverageReport  
2022-12-18T04:31:27: -reporttypes:HTML  
2022-12-18T04:31:30: Writing report file 'coverageReport\index.html'  
PS D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder.Test>
```

Figure 5-14: Using Terminal to export Code Coverage file to HTML

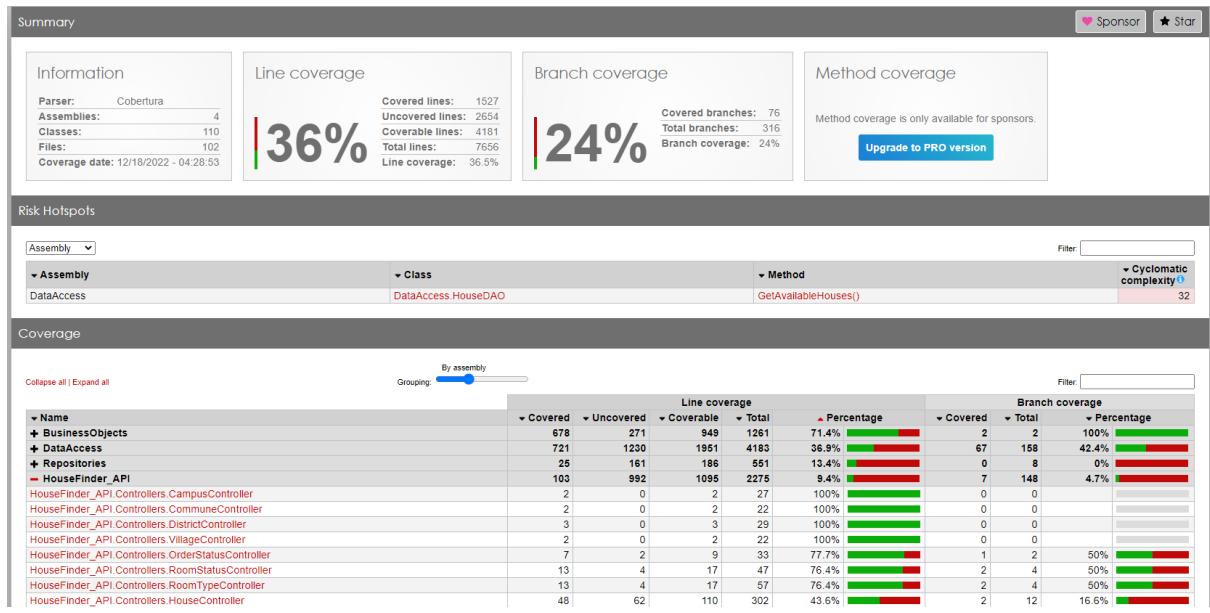


Figure 5-15: Sample Code Coverage report as static HTML

3.2. API Testing

API Testing will be done by the developers independently. The FHF development team embrace this feature to gain the following advantages:

- Make sure the API functions correctly before calling them in Front-end.
- To be aware of the data format to retrieve them accurately in Front-end.

3.2.1. API Testing with Swagger

API Testing can be performed by using the Swagger UI that becomes available while running the ASP.NET Web API Project with the Swagger library included (as default)

- Run Project to use Swagger:

```

Developer PowerShell ✘
+ Developer PowerShell + ⌂ ⓘ ⓘ ⓘ
*****
** Visual Studio 2019 Developer PowerShell v16.11.21
** Copyright (c) 2021 Microsoft Corporation
*****
PS D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder_API> dotnet run
Building...
info: Microsoft.Hosting.Lifetime[0]
Now listening on: https://localhost:5001
info: Microsoft.Hosting.Lifetime[0]
Now listening on: http://localhost:5000
info: Microsoft.Hosting.Lifetime[0]
Application started. Press Ctrl+C to shut down.
info: Microsoft.Hosting.Lifetime[0]
Hosting environment: Development
info: Microsoft.Hosting.Lifetime[0]
Content root path: D:\Study\Semester9\SEP490\fu-house-finder\HouseFinder_API\HouseFinder_API

```

Figure 5-16: Run Project

- Use the Swagger UI by accessing the URL that the project runs on:

Figure 5-17: Swagger UI

```

curl -X 'GET' \
  'https://localhost:5001/api/Campus' \
  -H 'accept: application/json;odata=minimal;odata.streaming=true'
Request URL
https://localhost:5001/api/campus
Server response
Code Details
200 Response body
[ {
  "campusId": 1,
  "name": "Hà Nội",
  "districts": [
    {
      "districtId": 1,
      "name": "Huyện Thanh Thủy",
      "communes": [
        {
          "communeId": 1,
          "name": "Thị trấn Liên Quan",
          "villages": [
            {
              "villageId": 1,
              "name": "Xã Quan 2",
              "commune": 1,
              "createdDate": "2022-12-18T04:19:44.577"
            }
          ]
        }
      ]
    }
  ]
}
  
```

Figure 5-18: Use the Swagger UI to test an API

3.2.2. API Testing with Postman

API Testing can also be performed by simulating FHF Front-end's call to Back-end by using Postman.

The screenshot shows the Postman application interface. On the left, there's a sidebar with sections for Home, Workspaces, API Network, and Explore. Below these are collections, APIs, environments, mock servers, monitors, flows, and history. The main area displays a collection named 'test'. Under this collection, there are several sub-folders: 'FU_House_Finder', 'Campus' (containing 'Get List Campus'), 'House' (containing 'GET Home Page Odata'), 'Room' (containing 'GET GetByHouse', 'GET GetAvailableRoom', 'GET CountCapacity', 'PUT ChangeStatus', 'DEL DeleteRoom'), 'File' (containing 'POST Upload File'), 'User' (containing 'POST Login', 'GET Logout'), 'Google Cloud API' (containing 'GET Google Map API', 'GET Google Drive API', 'GET Distance'), 'Deploy' (containing 'POST Test AWS Server', 'GET Test AWS Server 2', 'GET API Host'), and 'Order' (containing 'GET Order OData'). A specific endpoint, 'GET Home Page Odata', is selected and expanded. Its configuration shows a GET request to 'https://localhost:5001/api/House/availableHouses?\$skip=1&\$top=3&\$filter=contains(HouseName,'Tâm')'. The 'Params' tab is active, displaying three query parameters: '\$skip' (value 1), 'Stop' (value 3), and '\$filter' (value 'contains(HouseName,'Tâm')'). Below this, there are tabs for Body, Cookies, Headers (4), and Test Results, with 'Pretty' selected for the body view. The body content is a JSON object with numbered lines from 1 to 22, representing a house record.

Figure 5-19: Test API using Postman

3.3. Integration Testing

As a standard definition, FHF Project defines that a test case is:

A set of test data and test programs (test scripts) and their expected results. A test case validates one or more system requirements and generates a pass or fail.

A good test case should follow two basic aspects, the Contents and the Style. Test cases for functional testing are derived from the target of test's use cases. Test cases should be developed for each use case scenario. The use case scenarios are identified by describing the paths through the use case that traverse the basic flow and alternate flows start to finish through the use case.

The FHF Project team performs Integration Testing manually through interacting with the UI then record the test cases (including test scenario, test data, steps and results) into an Excel file:

	A	B	C	D	E	F	G	H
1	Module Name	Home Page			Pass	Fail	Untested	Number of Test cases
2	Created By	GiangNT			98	0	0	98
3	Creation Date	25/10/2022						
4	Reviewed By	HuyenBN						
5	Reviewed Date	29/10/2022						
6	Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions
7	TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_001	Enter valid house name & click Search button	1. Enter valid house name 2. Click on Search button	Valid URL Test Data	House name: T&m	User sees Houses with the name searched
8	TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_002	Enter invalid house name & click Search button	1. Enter invalid house name 2. Click on Search button	Valid URL Test Data	House name: qwerty	User sees no House
9	TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_003	Enter valid house name & press Enter	1. Enter valid house name 2. Press Enter key	Valid URL Test Data	House name: T&m	User sees Houses with the name searched
0	TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_004	Enter invalid house name & press Enter	1. Enter invalid house name 2. Press Enter key	Valid URL Test Data	House name: qwerty	User sees no House
1	TS_FHF_Home_002	Verify the Filter by	TC_FHF_Home_FilterByCamp	Choose a Campus	1. Select a Campus from the	Valid URL	Campus: FU - Hòa Lạc	User sees Houses of the Hoa

Figure 5-20: Sample Integration Test in [FHF]_IntegrationTestCases.xlsx

Detailed Test cases will be described in [FHF]_IntegrationTestCases.xlsx file.

3.4. System Testing

Each system test case is a complete flow of usage throughout the system for 1 user role.

The FHF Project team performs System Testing manually through interacting with the UI then record the test cases (including test scenario, test data, steps and results) into an Excel file:

A	B	C	D	E	F	G	H	I	J	K
1	Test Case ID	FHF_Guest_001	Test Case Description	Test the functionalities of role Guest in normal case						
2	Created By	GiangNT	Reviewed By	HuyenBN		Version	1			
3	Tester's Name	GiangNT	Date Tested	13/12/2022		Test Case (Pass/Fail/Not)	Pass			
4	No.	Prerequisites:		No.	Test Data					
5	1	Access to Chrome Browser		1	houseName: T&m					
6	Test Scenario	Verify functionalities of Guest's screens								
7	Step #	Step Details	Expected Results	Actual Results		Pass / Fail / Not executed / Suspended				
8	1	Navigate to https://fu-house-finder.vercel.app/home	Website is open, displays list of available houses	Website is open, displays list of available houses		Pass				
9	2	Enter houseName and click Search button	System displays Houses that name containing searched value	System displays Houses that name containing searched value		Pass				
10	3	Click on 1 House	System displays detail information of that House, along with its Landlord, Rate, Comments and available Rooms	System displays detail information of that House, along with its Landlord, Rate, Comments and available Rooms		Pass				
11	4	Click on House's Map	Opens a new Google Map tab that pins the location of the House	Opens a new Google Map tab that pins the location of the House		Pass				
12	5	Click on 1 Room	System displays detail information of that Room	System displays detail information of that Room		Pass				

Figure 5-21: Sample System Test in [FHF]_SystemTestCases.xlsx

Detailed Test cases will be described in [FHF]_SystemTestCases.xlsx file.

3.5. Acceptance Testing

UAT is a type of testing performed by the end user or the client to verify/accept the software system before moving the software application to the production environment. The main Purpose of UAT is to validate end to end business flow. It is kind of black-box testing where two or more end-users will be involved.

Our project will use the Checklist as a substitute for Acceptance Testing. The content of the Checklist is shown in the table below:

ID	Checklist	Yes	No
General			
CL-001	Text on all pages has no spelling and grammatical errors.	✓	
CL-002	All buttons function appropriately.	✓	
CL-003	All mandatory fields are validated.	✓	
CL-004	Validation error messages are displayed properly below the field.	✓	
CL-005	All error messages are displayed in red color.	✓	
CL-006	Delete functionality for any record on page are asked for confirmation.	✓	
CL-007	All numeric values are formatted properly.	✓	
CL-008	Unavailable pages are redirected to not found page.	✓	
CL-009	Add/Update/Delete actions display notification for successful or failed result.	✓	
GUI and Usability			
CL-010	Any user role will only need at most 4 times of mouse click to get to any place they want in the website.	✓	
CL-011	All ambiguous icons have explanatory tooltips to help user understand.	✓	
CL-012	Screens are designed follow project theme color Orange.	✓	
CL-013	Organization of information in a screen does not become disordered in different screen sizes.	✓	
CL-014	Font size on all pages of the system is always at least 11pt when displayed on any screen size.	✓	
CL-015	All components will always have at least 10px distance from each other.	✓	
CL-016	There are no use of slang, acronyms and abbreviations.	✓	
CL-017	All pages conform to Hick's Law.	✓	
CL-018	All pages conform to Law of Common Region.	✓	
CL-019	All pages conform to Law of Similarity.	✓	
CL-020	All pages conform to Miller's Law.	✓	
Performance			
CL-021	Page load time is within 30 seconds.	✓	
CL-022	Page still loads on slow connection.	✓	
Security			
CL-023	Unauthorized users cannot access authorized pages.	✓	

Table 5-8: Checklist table

3.6. Defect Log

FHF Project uses GitHub Issues to manage bugs and defects.

The GitHub Issues functionality in the Repository that is used as a version control of the project is also used to manage bugs and defects. Once a team member finds a bug, he will create a new Issue with its detail description, evidential images and assign a bug fixer.

The screenshot shows a GitHub Issues page with the search filter 'is:issue is:open'. There are 5 open issues listed:

- (Integration Test) {Kien} [Angular] Staff/staff-house-detail #559: Opened 5 days ago by huyenbnhe150346. Description: AvailableCapacity: Wrong displayed. Labels: bug.
- (Integration Test) {Kien} [Angular] House Detail - List Room, Room Detail #552: Opened last week by nguyenthegiang. Description: #552 opened last week by nguyenthegiang. Labels: bug.
- {An} [Angular] (Landlord/add-room) Fault when add single room #551: Opened last week by nguyenthegiang. Description: #551 opened last week by nguyenthegiang. Labels: bug.
- {An} [Angular] (Landlord/add-house) Fault: select region #550: Opened last week by nguyenthegiang. Description: #550 opened last week by nguyenthegiang. Labels: bug.
- {An} [API Deployment] SSL fault when calling from front-end #491: Opened 2 weeks ago by nguyenthegiang. Description: #491 opened 2 weeks ago by nguyenthegiang. Labels: bug.

Figure 5-9-1: Control bugs with GitHub Issues

The screenshot shows the details of issue #559: (Integration Test) {Kien} [Angular] Staff/staff-house-detail. The issue was opened by huyenbnhe150346 5 days ago. The description includes a screenshot showing a UI error: 'AvailableCapacity: Wrong displayed' and 'Tổng số phòng trống: 9 phòng' (Total empty rooms: 9 rooms). The issue has been labeled as 'bug'.

Comments:

- huyenbnhe150346 commented 5 days ago • edited by nguyenthegiang
- huyenbnhe150346 added the bug label 5 days ago
- huyenbnhe150346 assigned kienntfs1013 5 days ago
- kienntfs1013 commented 4 days ago

Assignees: kienntfs1013

Labels: bug

Projects: None yet

Milestone: No milestone

Development: Create a branch for this issue or link a pull request.

Notifications: Unsubscribe (You're receiving notifications because you're watching this repository.)

Participants: 2 participants (huyenbnhe150346, kienntfs1013)

Actions: Lock conversation, Pin issue

Figure 5-9-2: Control bugs with GitHub Issues

4. Test Report

4.1. Automation test case report

The contents of the Automation Test Case Report for Unit Test are shown in the table below:

Group of test case	Test Case	Pass	Fail	Not available	Number of Test Case
CampusController	Get all campuses	2	0	0	2
	Total	2	0	0	2
CommuneController	Count Commune having House	2	0	0	2

Group of test case	Test Case	Pass	Fail	Not available	Number of Test Case
	Total	2	0	0	2
DistrictController	Get all Districts	2	0	0	2
	Count District having House	2	0	0	2
	Total	4	0	0	4
HouseController	Get available Houses	2	0	0	2
	Get House by Id	3	0	0	3
	Update House	4	0	0	4
	Delete House	2	0	0	2
	Increase view	3	0	0	3
	Get list Houses by Landlord Id	3	0	0	3
	Get money for not rented Rooms	3	0	0	3
	Count total House	2	0	0	2
	Count available House	2	0	0	2
	Count total reported House	2	0	0	2
	Get reported Houses	2	0	0	2
	Total	28	0	0	28
OrderController	Get all Orders	2	0	0	2
	Count total Order	2	0	0	2
	Total	4	0	0	4
OrderStatusController	Get all OrderStatus	2	0	0	2
	Total	2	0	0	2
ReportController	Get all Reports	2	0	0	2
	Count total Report	2	0	0	2
	Total	4	0	0	4
RoomStatusController	Get RoomStatus by House Id	3	0	0	3
	Get all Status	2	0	0	2
	Total	5	0	0	5
RoomTypeController	Get RoomType by House Id	3	0	0	3
	Get RoomTypes	2	0	0	2
	Total	5	0	0	5
VillageController	Count Village having House	2	0	0	2
	Total	2	0	0	2
Total of test cases		58	0	0	58

Table 5-10: Automation test case report

4.2. Automation test report

Automation test is an integral part of development process. So that, there are 2 phases as 2 phases of Iterative and Incremental Software Process Model.

Test case	Phase 1		Phase 2		Final
	Pass	Fail	Pass	Fail	
CampusController	2	0	0	0	2

Test case	Phase 1		Phase 2		Final
	Pass	Fail	Pass	Fail	
CommuneController	2	0	2	0	2
DistrictController	2	0	2	0	4
HouseController	15	0	13	0	28
OrderController	0	0	4	0	4
OrderStatusController	0	0	2	0	2
ReportController	2	0	2	0	4
RoomStatusController	2	0	3	0	5
RoomTypeController	2	0	3	0	5
VillageController	0	0	2	0	2
Total of test cases	27	0	31	0	58

Table 5-11: Automation test report

Phase	Coverage			
	Statements	Branches	Functions	Lines
Phase 1	97.22%	100%	92.31%	98.25%
Phase 2	94%	90%	91.3%	96.25%

Table 5-12: Automation test coverage report

4.3. Integration test case report

The content of the Integration test case report are shown in the table below:

Module	Pass	Fail	Untested	Number of Test Cases
Home Page	98	0	0	98
House Detail	23	0	0	23
Room Detail	4	0	0	4
Login	23	0	0	23
(Landlord) Manage Houses	21	0	0	21
(Landlord) Upload House Info	12	0	0	12
(Landlord) Manage Rooms	16	0	0	16
(Staff) Dashboard	2	0	0	2
(Staff) Manage Landlords	17	0	0	17
(Staff) List Houses	16	0	0	16
(Staff) Manage Reports	7	0	0	7
(Staff) Manage Orders	21	0	0	21
(Staff) Manage Sign up Requests	4	0	0	4
(Staff) Change Password	8	0	0	8
(Admin) Manage Staffs	6	0	0	6
Total of Test Cases	278	0	0	278

Table 5-13: Integration Test case report

Test Case list:

Module Name	Home Page	Pass	Fail	Untested	Number of Test cases			
Created By	GiangNT	98	0	0	98			
Creation Date	25/10/2022							
Reviewed By	HuyenBN							
Reviewed Date	29/10/2022							
Test Scenario ID	Test Scenario Description	Test Case ID	Test Case Description	Test Steps	Preconditions	Test Data	Post Conditions	Expect
TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_001	Enter valid house name & click Search button	1. Enter valid house name 2. Click on Search button	Valid URL Test Data	House name: Tâm	User sees Houses with the name searched	System that the search
TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_002	Enter invalid house name & click Search button	1. Enter invalid house name 2. Click on Search button	Valid URL Test Data	House name: qwerty	User sees no House	System that the search
TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_003	Enter valid house name & press Enter	1. Enter valid house name 2. Press Enter key	Valid URL Test Data	House name: Tâm	User sees Houses with the name searched	System that the search
TS_FHF_Home_001	Verify the Search by Name functionality of Home Page	TC_FHF_Home_SearchName_004	Enter invalid house name & press Enter	1. Enter invalid house name 2. Press Enter key	Valid URL Test Data	House name: qwerty	User sees no House	System that the search
TS_FHF_Home_002	Verify the Filter by Campus functionality of Home Page	TC_FHF_Home_FilterByCampus_001	Choose a Campus	1. Select a Campus from the Dropdown	Valid URL Test Data	Campus: FU - Hòa Lạc	User sees Houses of the Hoa Lac Campus	System Hoa La
TS_FHF_Home_002	Verify the Filter by Campus functionality of Home Page	TC_FHF_Home_FilterByCampus_002	Choose a Campus	1. Select a Campus from the Dropdown	Valid URL Test Data	Campus: FU - Cần Thơ	User sees Houses of the Can Tho Campus	System Can Th

Figure 5-22: Test case list

For more details about all test suites, please look at file **[FHF]_IntegrationTestCases.xlsx**

4.4. Integration test report

We execute test in 2 phases of process model, to finish project. With the second phase, we test with 2 small tasks:

- The first one: re-test phase 1 and implemented function of phase 2
- The second one: test all test cases and re-test to confirm bug-free

The contents of the Test Report are shown in the table below:

Module	Phase 1		Phase 2		Final
	Pass	Fail	Pass	Fail	
Home Page	87	0	11	0	98
House Detail	20	0	3	0	23
Room Detail	4	0	0	0	4
Login	13	0	10	0	23
(Landlord) Manage Houses	12	0	9	0	21
(Landlord) Upload House Info	6	0	6	0	12
(Landlord) Manage Rooms	5	0	11	0	16
(Staff) Dashboard	0	0	2	0	2
(Staff) Manage Landlords	6	0	11	0	17
(Staff) List Houses	6	0	10	0	16
(Staff) Manage Reports	0	0	7	0	7
(Staff) Manage Orders	3	0	18	0	21
(Staff) Manage Sign up Requests	0	0	4	0	4
(Staff) Change Password	0	0	8	0	8
(Admin) Manage Staffs	0	0	6	0	6
Total of Test Cases	162	0	116	0	278

Table 5-14: Integration test report

VI. Release Package & User Guides

1. Deliverable Package

No.	Deliverable Item	Sub-items
<i>Source Codes</i>		
1	ASP.NET Core Web API	HouseFinder_API
2	Client	House-Finder-Client
<i>Database Script</i>		
1	Database	FHF_Database.sql
<i>Report Document</i>		
1	Project Introduction	[FHF]_Report 1_Project Introduction.docx
2	Project Management Plan	[FHF]_Report 2_Project Management Plan.docx
3	Software Requirement Specification	[FHF]_Report 3_Software Requirement Specification.docx
4	Software Design Document	[FHF]_Report 4_Software Design Document.docx
5	Software Test Documentation	[FHF]_Report 5_Test Documentation.docx
6	Software User Guide	[FHF]_Report6_Software User Guide.docx
7	Final Project Report	[FHF]_Report7_Final Project Report .docx
<i>Test Case Document</i>		
1	Integration Test Case	[FHF]_IntegrationTestCases.xlsx
2	System Test Case	[FHF]_SystemTestCases.xlsx
<i>Issues List</i>		
1	Issues	https://github.com/nguyenthegiang/fu-house-finder/issues
<i>Slide</i>		
1	Slide	[FHF]-Slide

Table 6-1: Deliverable packages

- Laptop with Microsoft Windows 10 Home
- Integrated Development Environment: Visual Studio 2019, Visual Studio Code 2022
- Database: Microsoft SQL Server 2019

2. Installation Guides

2.1. Environment for Development

- Laptop/PC using Windows 10 Home OS with minimum requirements:
 - 8GB of Memory
 - CPU Intel Core i5 or equivalent
- SQL Server: Version 2019 Express Edition
- SQL Server Management Tools: SSMS
- Angular Development IDE: Visual Studio Code 2022
- ASP.NET Development IDE: Visual Studio 2019
- Version Control Tool: Git, Github
- NodeJS v14

2.1.1 Setup Development environment

2.1.1.1 Install MSSQL

Go to <https://www.microsoft.com/en-in/sql-server/sql-server-downloads> And Download Express Edition installation file. Follow the instruction to install.

Go to <https://learn.microsoft.com/en-us/sql/ssms/download-sql-server-management-studio-ssms?view=sql-server-ver16>, click on link Free Download for SQL Server Management Studio (SSMS) 18.12.1 to start download Management Tool and follow the instruction to install

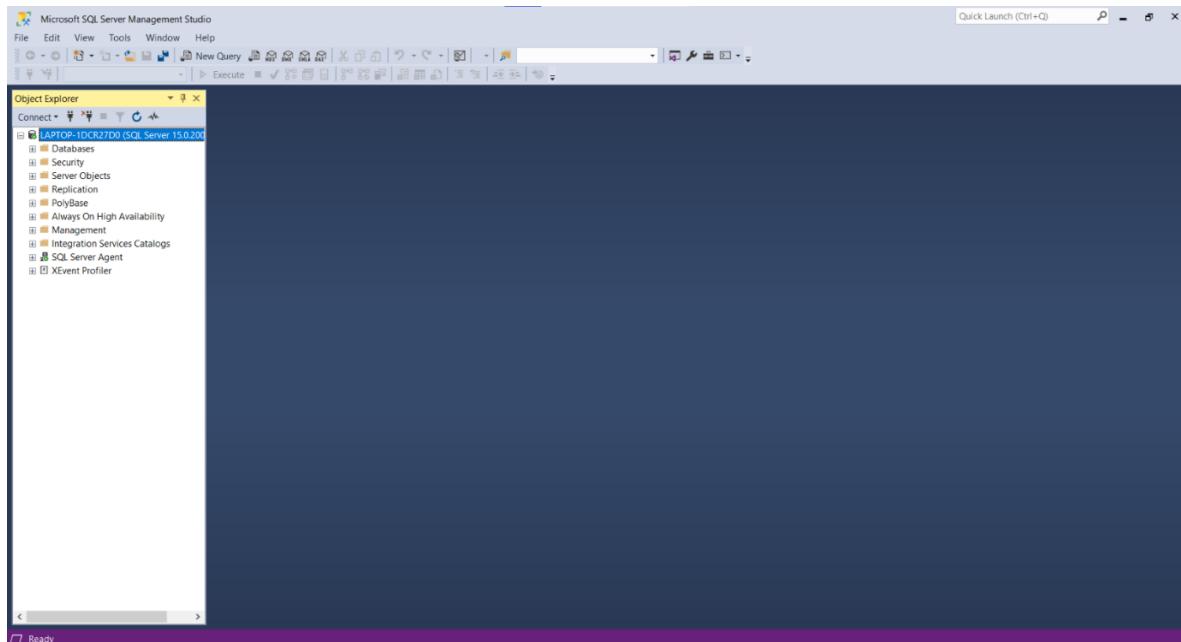


Figure 6-1: Microsoft SQL Server Management studio

2.1.1.2 Install Visual Studio Code

Go to <https://code.visualstudio.com/download> and select download for Windows to start download installation package and follow instruction to install

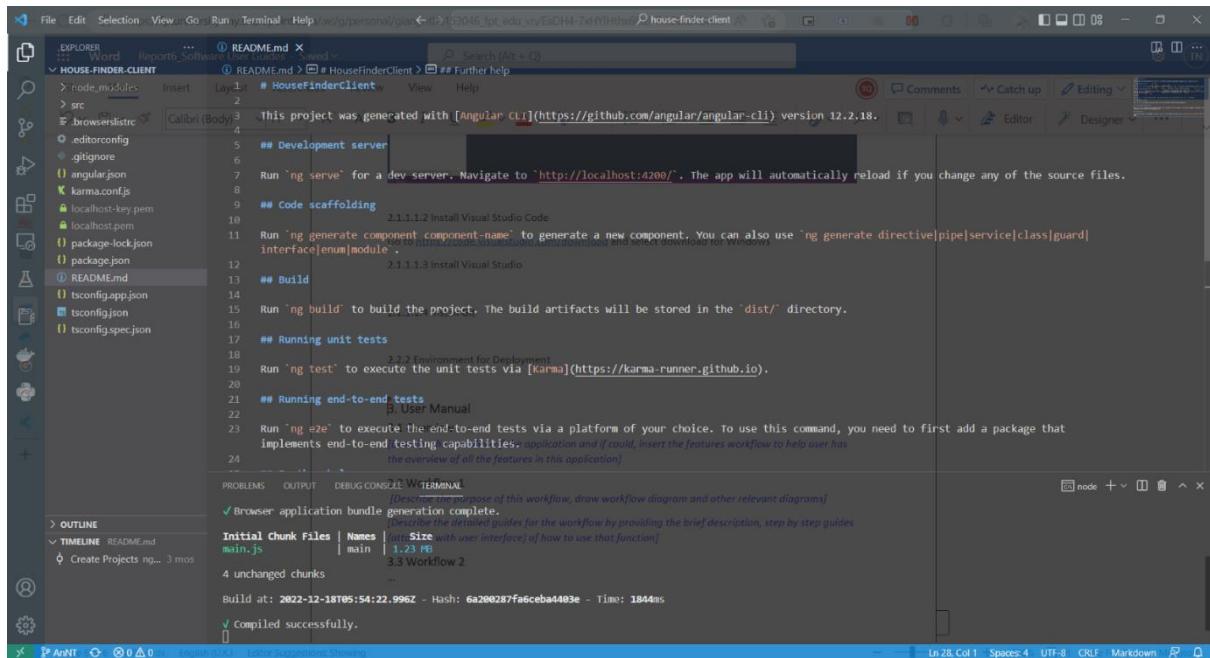


Figure 6-2: Visual Studio Code

2.1.1.3 Install Visual Studio

Goto <https://visualstudio.microsoft.com/vs/older-downloads/> Select Version 2019, Click download, sign in to Microsoft account and click Download button on Visual Studio 2019 Community Version to start download installation package. Follow the instruction to install

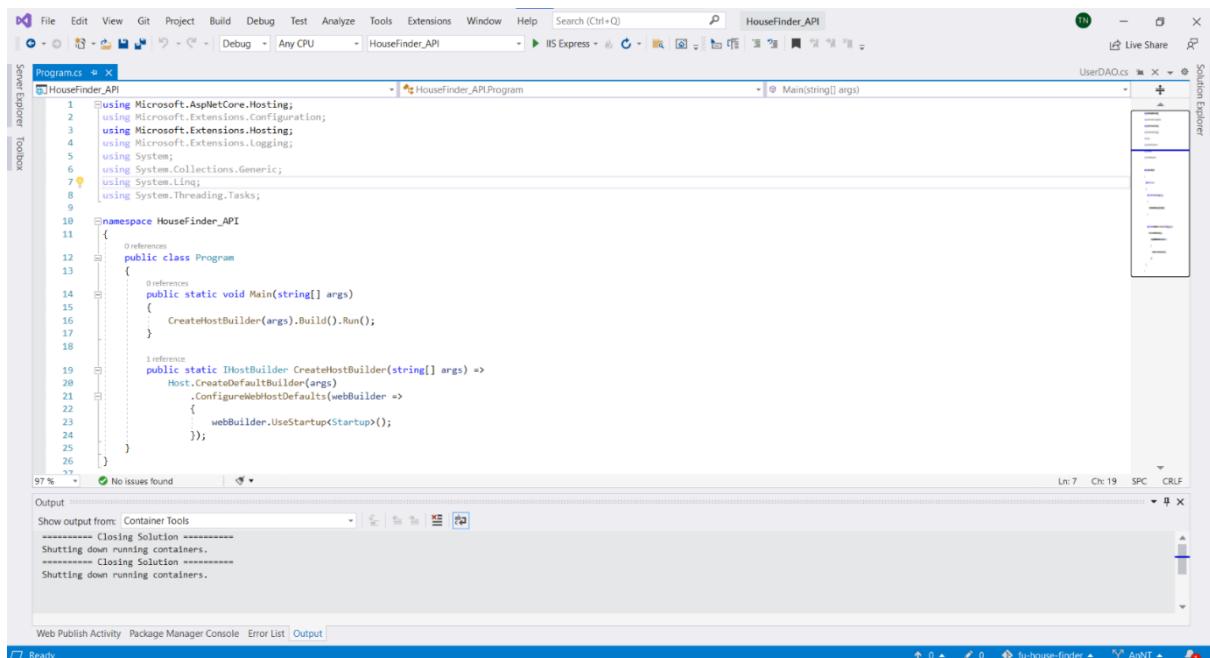


Figure 6-3: Visual Studio

2.1.1.4 Install Git

Go to <https://git-scm.com/downloads> And download version for windows. Follow the instruction to install.

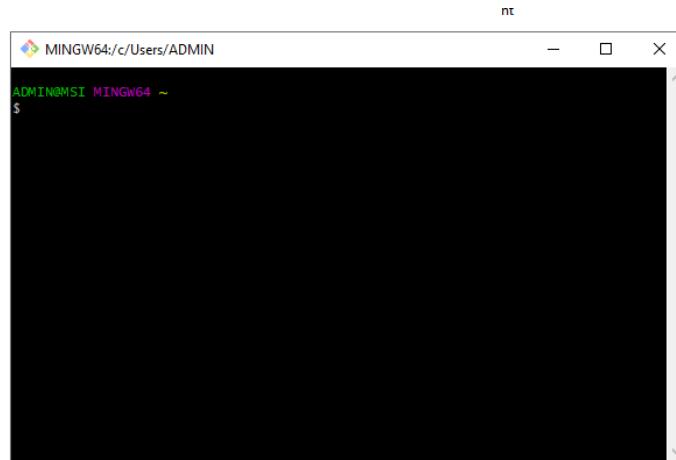


Figure 6-4: Git Bash

2.1.1.5 Install NodeJS

Go to <https://nodejs.org/en/blog/release/v14.17.3/> And Download version Window 64-bit Installer and follow instruction to install.

2.2. Environment for Deployment

- Window Server 2022 Base (provided by AWS EC2)
 - IIS 10 as Web Server
- SQL Server Service on Amazon RDS

2.2.1 Setup Deployment Environment

2.2.1.1 Prepare Window Server Virtual Machine

- Login an account on AWS Management Console at <https://aws.amazon.com/console/>
- From the AWS Management Console Search for EC2
- On EC2 Management Page select Instance from the left menu and click Launch Instance
- Select Windows Instance, AMI version Microsoft Windows Server 2022 Base
- Setup Server name, Storage, Login Key Pair (save key file to device to decrypt windows password)
- Setup Instance Type, in this project, we use t2.medium type comes with 2 vCPU and 4GiB Memory
- Setup Network Settings: Click Edit button
- For Inbound Settings: Setup 4 rules:
 - RDP: source type anywhere
 - HTTP: source type anywhere
 - HTTPS: source type anywhere
 - SSH: source type anywhere
- Click Launch instance and wait for about 5 minutes for the machine to be ready, result will be an instance in the instance list

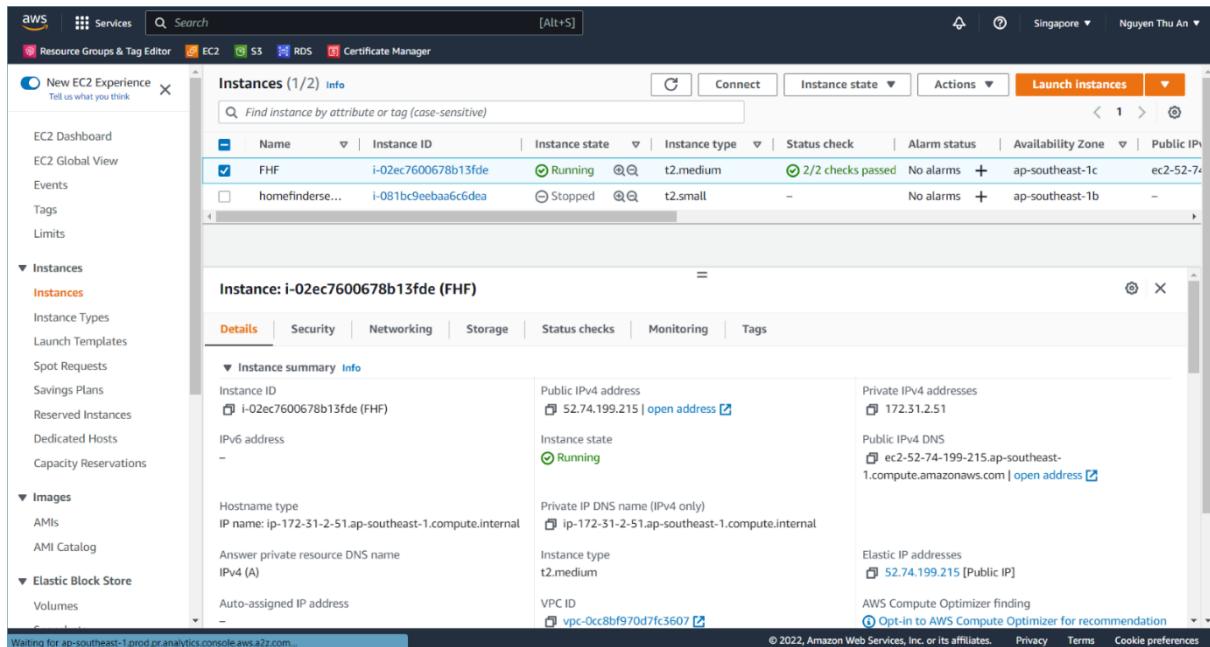


Figure 6-5: AWS Management Console

2.2.1.2 Setup Web Server

- Connect to the virtual machine:
 - Download IIS on Server Management App
 - Go to <https://dotnet.microsoft.com/en-us/download/dotnet/5.0> and download Hosting Bundle and install
- On IIS: Setup SSL Certificate for the website

2.2.1.3 Setup Front-end Server

- Access <https://vercel.com/> and create a new account
- Create a new Project and link to GitHub Repository
- Set up deployment environment (Angular) and deploy

Figure 6-6: Vercel

3. User Manual

3.1. Overview

This user guide will provide detailed instructions for user features according to each role (Guest, Student, Landlord, Staff, Admin) when accessing the FU House Finder website:

<https://fuhousefinder.com>

3.2. Web application

3.2.1 Authentication

3.2.1.1 Sign in

Role:(Student, Landlord, Staff, Admin) - Web user sign in to the system

Step 1: Open website

Step 2: Click on “Đăng nhập” button

The screenshot shows the homepage of the FU House Finder website. At the top, there is a red header bar with the title "TÌM TRỌ SINH VIÊN FPT" and a "Đăng nhập" button. Below the header, there is a search bar labeled "Nhập tên trọ" and a dropdown menu for sorting results. On the left, there is a sidebar titled "LỌC KẾT QUẢ" with dropdown menus for "Cơ sở", "Khu vực:", "Huyện/Quận", "Phường/Xã", and "Thôn/Xóm". There is also a field for "Khoảng cách đến trường" and a "Áp dụng" button. The main content area displays three dormitory options: "Nhà có phòng trống" (29 rooms), "Phòng trống" (63 rooms), and "Chỗ trống" (81 rooms). Each option includes a thumbnail image and the name of the dormitory: "Trọ Tâm Nguyễn", "Trọ Linh Lê", and "Trọ Linh Lê" respectively. A red button at the bottom right of the dormitory section says "Đang ký yêu cầu phòng trọ".

Step 3: Users fill all required information (username and password) or login with facebook and google into the form and click the “Đăng nhập” button.

The image shows two side-by-side screenshots of the FU HOUSE FINDER application. On the left is the home page, which features a magnifying glass icon with a frog inside a house, the text "FU HOUSE FINDER", and "Ứng dụng tìm trọ dành cho sinh viên Đại học FPT". On the right is the login page titled "ĐĂNG NHẬP" (Login). It includes social login options for Facebook and Google, a user input field for "Email" containing "thanhle@gmail.com", a password field with ".....", and a "Đăng nhập" (Login) button.

Step 4: Select role.

A modal dialog box is displayed over the login screen. It contains the text "Bạn muốn đăng ký tài khoản với vai trò là:" (You want to register an account with the role of:). Below this are two buttons: "Học sinh" (Student) in red and "Chủ trọ" (Landlord) in blue. The background of the dialog box is white, while the rest of the screen is orange.

a. System redirect to Home page if user selects student

The screenshot shows the homepage of the FU House Finder website. At the top, there is a navigation bar with the title "TÌM TRỌ SINH VIÊN FPT" and links for "Trang chủ", "Tuyển sinh", "Liên hệ", and "Giới thiệu". A dropdown menu "Lê Thành" is open. On the left, there is a sidebar titled "LỌC KẾT QUẢ" with filters for "Cơ sở", "Khu vực:", "Huyện/Quận", "Phường/Xã", "Thôn/Xóm", "Khoảng cách đến trường", and a "Áp dụng" button. In the center, there is a search bar with placeholder "Nhập tên trọ" and a magnifying glass icon. To the right of the search bar is a dropdown menu "Sắp xếp theo". Below these are three cards representing dormitory options:

- Nhà có phòng trống**: 29 units available. Includes a thumbnail image of a hallway with blue-framed doors.
- Phòng trống**: 63 units available. Includes a thumbnail image of a multi-story building with multiple entrances.
- Chỗ trống**: 81 units available. Includes a thumbnail image of a long hallway with doors on both sides.

A total count of 29 dormitories found is displayed on the right: "Tim thấy 29 nhà trọ". A red button at the bottom right says "Đăng ký yêu cầu phòng trọ" (Request room).

b. System redirect to Landlord page if user selects landlord and if the landlord is new, you must register an account for the landlord by following the steps below

Step 1: Input name and facebook profile url

This step involves entering basic contact details. It consists of three numbered fields:

- ① Điện Thông tin liên hệ: A field for entering a phone number.
- ② Chụp ảnh CMND: A field for entering a Facebook link.
- ③ Xác nhận thông tin: A red "Tiếp theo" (Next) button.

Step 2: Input ID card

This step involves uploading the front and back sides of an ID card. It consists of three numbered fields:

- ① Điện Thông tin liên hệ: Placeholder text "Điện Thông tin liên hệ" with a red crossed-out icon.
- ② Chụp ảnh CMND: Two fields for uploading the front and back sides of an ID card. Each field has a placeholder text "Kéo thả ảnh ở đây" and a "Quay lại" (Back) button.
- ③ Xác nhận thông tin: A red "Tiếp theo" (Next) button.

Step 3: Verify and register

1 Diền Thông tin liên hệ

2 Chụp ảnh CMND

3 Xác nhận thông tin

Xác nhận thông tin đăng ký

Số điện thoại: 0987654321
Facebook: <https://www.facebook.com/profile.php?id=123456789>
CMND mặt trước



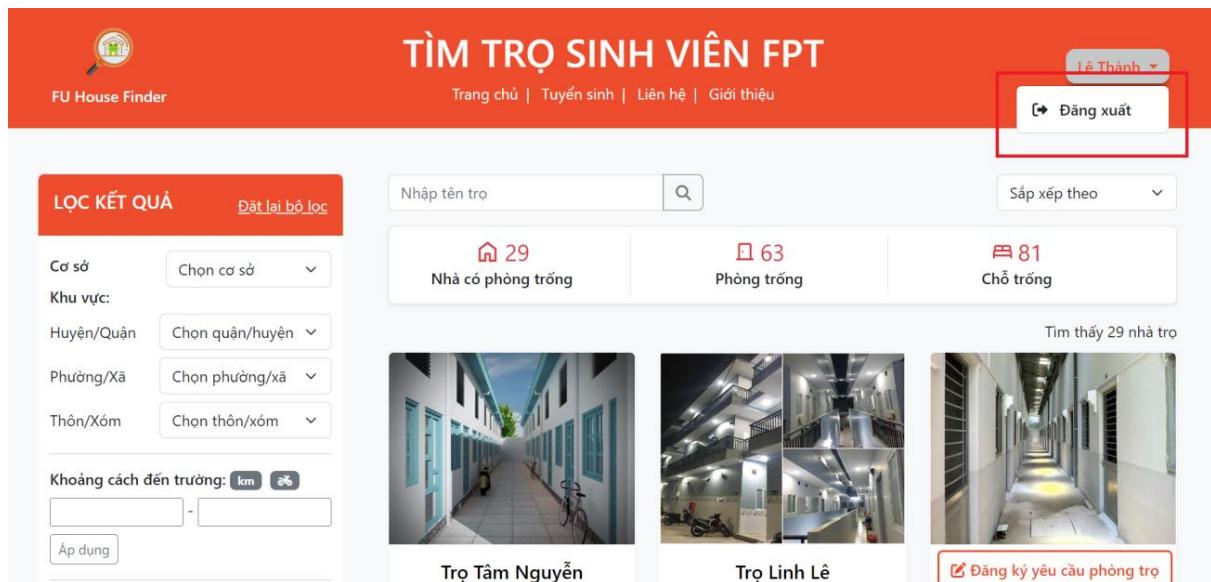
CMND mặt sau



Quay lại **Đăng kí**

3.2.1.2 Sign out

Step 1: On Header in Home page, click on “Đăng xuất” button



TÌM TRỌ SINH VIÊN FPT

Lê Thành

Đăng xuất

Nhập tên trọ

Sắp xếp theo

LỌC KẾT QUẢ

Đặt lại bộ lọc

Cơ sở: Chọn cơ sở

Khu vực:

Huyện/Quận: Chọn quận/huyện

Phường/Xã: Chọn phường/xã

Thôn/Xóm: Chọn thôn/xóm

Khoảng cách đến trường: km

Áp dụng

Nhà có phòng trống: 29

Phòng trống: 63

Chỗ trống: 81

Tim thấy 29 nhà trọ

Trọ Tâm Nguyễn

Trọ Linh Lê

Đăng ký yêu cầu phòng trọ

Step 2: Redirect to Home Page

The screenshot shows the homepage of the FU House Finder website. At the top, there is a logo with a magnifying glass icon and the text "FU House Finder". The main title "TÌM TRỌ SINH VIÊN FPT" is centered above a search bar with the placeholder "Nhập tên trọ" and a search icon. To the right of the search bar is a button "Đăng nhập" and a dropdown menu "Sắp xếp theo". Below the search area, there is a section titled "LỌC KẾT QUẢ" with a "Đặt lại bộ lọc" button. This section includes dropdown menus for "Cơ sở" (selected "Chọn cơ sở") and "Khu vực:" (selected "Chọn quận/huyện"). There are also dropdowns for "Huyện/Quận", "Phường/Xã", and "Thôn/Xóm". A distance filter section "Khoảng cách đến trường: km" is present, with two input fields and a "Áp dụng" button. To the right, there are three cards for dormitories: "Nhà có phòng trống" (29 units), "Phòng trống" (63 units), and "Chỗ trống" (81 units). Each card includes a thumbnail image, the name, and a "Đang ký yêu cầu phòng trọ" button.

3.2.2 Guest/Student

3.2.2.1 Filter house (name, price, distance, campus, district, commune, village, room type, house utility, room utility, rate)

Role: (Student/Guest)

Step 1: In Home page select filter

This screenshot is similar to the one above, showing the FU House Finder website. The "LỌC KẾT QUẢ" section on the left is highlighted with a red box. The rest of the interface is identical to the first screenshot, including the search bar, sorting options, and dormitory listing cards.

Giá tiền: VND/Tháng

-

Loại phòng:

- Khép kín
- Không khép kín
- Chung cư mini

Tiện ích khác:

- Khóa vân tay
- Camera an ninh
- Chỗ để xe
- Tủ lạnh
- Bếp
- Máy giặt
- Bàn học
- Không chung chủ
- Giường
- Vệ sinh khép kín

Trọ Thu Thảo

0.1 km

⌚ ₫2,900,000 - ₫2,900,000
📍 Nhà số..., Đường...; Đồi diệ...
🛏 3 🚧 3

80

Trọ Tâm Lê

0.9 km

⌚ ₫2,750,000 - ₫2,750,000
📍 Nhà số..., Đường...; Đồi diệ...
🛏 3 🚧 3

72

Trọ Tâm Thảo

4.5 km

⌚ ₫2,000,000 - ₫3,000,000
📍 Nhà số..., Đường...; Đồi diệ...
🛏 4 🚧 5

50

Tủ lạnh

Bếp

Máy giặt

Bàn học

Không chung chủ

Giường

Vệ sinh khép kín

Trọ Thu Thảo

0.5 km

⌚ ₫3,000,000 - ₫3,000,000
📍 Nhách 75, thôn 4, Thạch H...
🛏 3 🚧 3

31

Trọ Tâm Lê

1.2 km

⌚ ₫2,750,000 - ₫2,750,000
📍 Nhách 75, thôn 4, Thạch H...
🛏 3 🚧 3

87

Trọ Tâm Thảo

3.5 km

⌚ ₫3,000,000 - ₫4,000,000
📍 Nhách 75, thôn 4, Thạch H...
🛏 2 🚧 4

34

Tủ lạnh

Bếp

Máy giặt

Bàn học

Không chung chủ

Giường

Vệ sinh khép kín

Đánh giá:

★★★★★

★★★★★ trót lén

★★★★★ trót lén

★★★★★ trót lén

★★★★★ trót lén

Trọ RoomType 1

4.3 km

⌚ ₫3,000,000 - ₫3,000,000
📍 Nhách 75, thôn 4, Thạch H...
🛏 1 🚧 1

Trọ RoomType 2

3.3 km

⌚ ₫3,000,000 - ₫3,000,000
📍 Nhách 75, thôn 4, Thạch H...
🛏 1 🚧 1

Trọ Campus 5

4.4 km

⌚ ₫3,000,000 - ₫3,000,000
📍 Nhách 75, thôn 4, Thạch H...
☒ Đăng ký yêu cầu phòng trọ

Step 2: Show list houses found

LỌC KẾT QUẢ Đặt lại bộ lọc

Cơ sở: FU - Hòa Lạc ▼
 Khu vực:
 Huyện/Quận: Huyện Thạch Thất ▼
 Phường/Xã: Thị trấn Liên Quan ▼
 Thôn/Xóm: Đồng Cam ▼

Khoảng cách đến trường: km -
Áp dụng

Giá tiền: VND/Tháng -
Áp dụng

Lê Giá: Thấp đến Cao ▾

H 29	H 63	H 81
Nhà có phòng trống	Phòng trống	Chỗ trống
 Trọ Tâm Lê	 Trọ Linh Lê	 Trọ Tâm Lê
4.5 km ⚡ ₫2,000,000 - ₫3,000,000 Nhà số..., Đường...; Đối diện... 4 ⚡ 5	0.9 km ⚡ ₫2,750,000 - ₫2,750,000 Nhà số..., Đường...; Đối diện... 3 ⚡ 3	1.2 km ⚡ ₫2,750,000 - ₫2,750,000 Nhách 75, thôn 4, Thạch H... 3 ⚡ 3
<input checked="" type="checkbox"/> Đăng ký yêu cầu phòng trọ		

3.2.2.2 Create Order for a Room

Role: (Student)

Step 1: Sign in with role is Student



FU HOUSE FINDER
 Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
 trikienthachban@gmail.com 

hoặc sử dụng

Email *

Mật khẩu *

Step 2: In Home page click on “Đăng ký yêu cầu phòng trọ” on bottom

The screenshot shows the homepage of the FU House Finder website. At the top, there is a navigation bar with links to 'Trang chủ', 'Tuyển sinh', 'Liên hệ', and 'Giới thiệu'. On the right side of the header, there is a dropdown menu labeled 'Kien nguyen tri'. Below the header, there is a search bar with the placeholder 'Nhập tên trọ' and a search icon. To the right of the search bar is a dropdown menu labeled 'Sắp xếp theo'. On the left, there is a sidebar titled 'LỌC KẾT QUẢ' with several filtering options: 'Cơ sở' (selected 'Chọn cơ sở'), 'Khu vực:' (selected 'Chọn quận/huyện'), 'Huyện/Quận' (selected 'Chọn quận/huyện'), 'Phường/Xã' (selected 'Chọn phường/xã'), 'Thôn/Xóm' (selected 'Chọn thôn/xóm'), 'Khoảng cách đến trường: [km] - [miles]', and an 'Áp dụng' button. In the center, there are three cards for dormitory listings: 'Nhà có phòng trống' (House with room), 'Phòng trống' (Empty room), and 'Chỗ trống' (Empty place). Each card includes a thumbnail image, the name of the dormitory ('Trọ Tâm Nguyễn', 'Trọ Linh Lê', 'Trọ Tâm Lê'), and a red button labeled 'Đăng ký yêu cầu phòng trọ' (Request room). A note at the bottom right says 'Tim thấy 29 nhà trọ'.

Step 3: Input information in modal and click on “Gửi yêu cầu” button to send request

The screenshot shows a modal window titled 'Đăng ký yêu cầu phòng trọ'. It contains fields for 'Tên' (Name: Nguyễn Trí Kiên), 'Số điện thoại' (Phone number: 0987654321), 'Email' (Email: trikienthachban@gmail.com), and 'Nguyên vọng' (Desires: Cần tìm phòng dưới 3 triệu gần trường). At the bottom, there are 'Hủy' (Cancel) and 'Gửi yêu cầu' (Send request) buttons. The background shows a blurred view of the dormitory listing page from Step 3.

Step 4: Show message “Thay đổi thông tin thành công”

The screenshot shows a search interface for finding dormitory rooms (Trọ). On the left, there are several filters:

- Thôn/Xóm:** Chọn thôn/xóm
- Khoảng cách đến trường:** km (with input fields) - bicycle icon
- Giá tiền:** VND/Tháng (with input fields)
- Loại phòng:**
 - Khép kín
 - Không khép kín
 - Chung cư mini
- Tiện ích khác:**
 - Khóa vân tay
 - Camera an ninh
 - Chỗ để xe

 On the right, the results are displayed in a grid:

- Trọ Tâm Nguyễn**: 0.1 km, 2,900,000 - 2,900,000 VND, 3 bedrooms, 3 bathrooms, 79 reviews.
- Trọ Linh Lê**: 0.9 km, 2,750,000 - 2,750,000 VND, 3 bedrooms, 3 bathrooms, 72 reviews.
- Trọ Tâm Lê**: 4.5 km, 2,000,000 - 3,000,000 VND, 4 bedrooms, 5 bathrooms, 50 reviews.

 Below the results are two additional images: a 3D rendering of a modern building complex and a photo of a single-story house entrance. A red button at the bottom right says "Đăng ký yêu cầu phòng trọ" (Request room).

3.2.2.3 View Houses Detail

Role: (Student /Guest)

Step 1: In Home page click on house want to view detail house

This screenshot shows the search results page with three detailed listings:

- Trọ Tâm Nguyễn**: 0.1 km, 2,900,000 - 2,900,000 VND, 3 bedrooms, 3 bathrooms, 81 reviews.
- Trọ Linh Lê**: 0.9 km, 2,750,000 - 2,750,000 VND, 3 bedrooms, 3 bathrooms, 72 reviews.
- Trọ Tâm Lê**: 4.5 km, 2,000,000 - 3,000,000 VND, 4 bedrooms, 5 bathrooms, 50 reviews.

 Each listing includes a thumbnail image, address, price range, number of bedrooms and bathrooms, and review count. A red button at the bottom right of each listing says "Đăng ký yêu cầu phòng trọ" (Request room).

Step 2: Redirect to House detail page

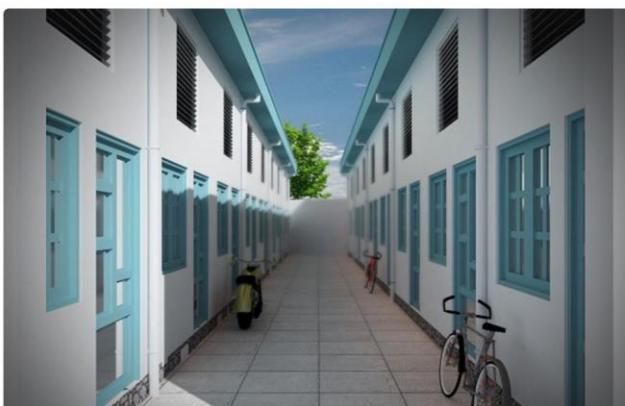


FU House Finder

TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Kien nguyen tri ▾

[Trang chủ](#) > Chi tiết nhà trọ

Thông tin chủ trọ



Họ và tên: Tâm Lê

Facebook: facebook.com/tamle12Số điện thoại: [Click để hiển thị](#)Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT;
Cạnh quán Bún bò Huế

Trọ Tâm Nguyễn

Giá điện: 3700/kWh

Giá nước: 1200/m³

Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế

Thông tin khác: Vệ sinh khép kín

Người dùng đánh giá

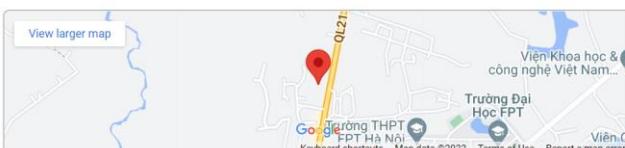
Bình luận:

Chưa có đánh giá

Viết bình luận:

Đánh giá: ☆☆☆☆☆

Đăng



Danh sách phòng trống

Tổng phòng trống hoàn toàn: 0 phòng

Tổng phòng trống một phần: 3 phòng



Tên phòng	Giá phòng	Tiện ích	Loại phòng	Diện tích	Số người ở
101	đ2,900,000		Không khép kín	3 m ²	
102	đ2,900,000		Không khép kín	3 m ²	
103	đ2,900,000		Không khép kín	3 m ²	

3.2.2.4 Rate and comment

Role: (Student)

Step 1: Sign in with role is User



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

hoặc sử dụng

Email *
thanhlle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Rate and comment the house



Trọ Tâm Nguyễn

Giá điện: 3700/kWh
Giá nước: 1200/m³
Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế
Thông tin khác: Vệ sinh khép kín

Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT;
Cạnh quán Bún bò Huế

Người dùng đánh giá

Bình luận:
Chưa có đánh giá

Viết bình luận:
Nhà trọ đẹp, dịch vụ tốt

Danh giá: ★★★★★

Đăng



Step 3: Click on “Đăng” button to send rate and comment



Địa chỉ: Nhà số..., Đường...; Đổi điện cổng sau Đại học FPT;
Cạnh quán Bún bò Huế

Trọ Tâm Nguyễn

Giá điện: 3700/kWh
Giá nước: 1200/m³
Địa chỉ: Nhà số..., Đường...; Đổi điện cổng sau Đại học FPT; Cạnh quán Bún bò Huế
Thông tin khác: Vệ sinh khép kín



Người dùng đánh giá

Bình luận:

Chưa có đánh giá

Viết bình luận:

Nhà trọ đẹp, dịch vụ tốt

Đánh giá: ★★★★★

Đăng



Step 4: Show message “Bình luận thành công”



Địa chỉ: Nhà số..., Đường...; Đổi điện cổng sau Đại học FPT;
Cạnh quán Bún bò Huế

Bình luận thành công

Người dùng đánh giá

Bình luận:
HE172885: Nhà trọ đẹp, dịch vụ tốt (★★★★★)

Viết bình luận:
Nhà trọ đẹp, dịch vụ tốt

Đánh giá: ★★★★★

Đăng



3.2.2.5 View Map

Role: (Student/Guest)

Step 1: In house detail, click on the map



Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT;
Cạnh quán Bún bò Huế

Trọ Tâm Nguyễn

Giá điện: 3700/kWh
Giá nước: 1200/m³
Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế
Thông tin khác: Vệ sinh khép kín

Người dùng đánh giá

Bình luận:

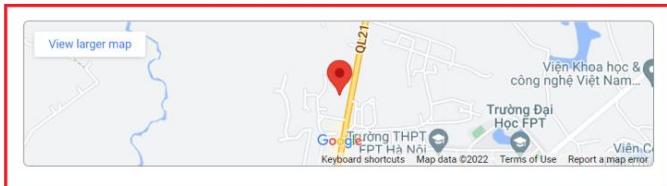
Chưa có đánh giá

Viết bình luận:

Nhà trọ đẹp, dịch vụ tốt

Dánh giá: ★★★★☆

Đăng



Step 2: Show map

21.0159519,105.5190104

21°00'57.4"N 105°31'08.4"E
21.015952, 105.519010

Directions Save Nearby Send to phone Share

2G89+FMC, Thạch Hòa, Thạch Thất, Hà Nội, Vietnam
2G89+9JJ Thạch Thất, Hanoi, Vietnam
Add a missing place
Add your business

Chợ Hoa Lạc, Nhà hàng Minh Nga Cố sô 1, Trường THPT FPT Hà Nội, Trạm Xăng Dầu Thạch Hòa 39, Nhà Hàng Gà Quang Thủ, Gà Ri Phú Bình 2, Vườn Xoài 68 Restaurant, Trường Đại Học FPT, Viện Khoa học & công nghệ Việt Nam, Viện Công nghiệp Phần mềm và Nội...
Nhà trọ Văn Giản, Nhà Thờ Phủ Hữu, Sân Bóng Hoàng Gia, Trung Tâm Vũ Trụ Việt Nam (VNSC), Định Quán Trần (Định Mục Uyên), FPT F-Ville 3 Hòa Lạc, Công Ty DT&C Vina, Khu Công Nghệ Cao Hòa Lạc, Trung tâm Internet Việt Nam - Bộ Thông tin &...
Ngân Hàng Quân Đội (Google3Bank) Hòa Lạc, FPT Hola Park 2, Khu Công Nghệ Cao Hòa Lạc, Hồ Tản Xã

3.2.2.6 Report house

Role: (Student)

Step 1: Sign in with role is Student



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Click the icon on bottom



Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế

Trọ Tâm Nguyễn

Giá điện: 3700/kWh
Giá nước: 1200/m³
Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế
Thông tin khác: Vệ sinh khép kín

Người dùng đánh giá

Bình luận:

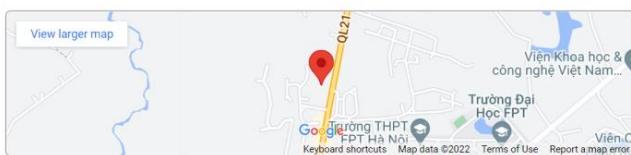
Chưa có đánh giá

Viết bình luận:

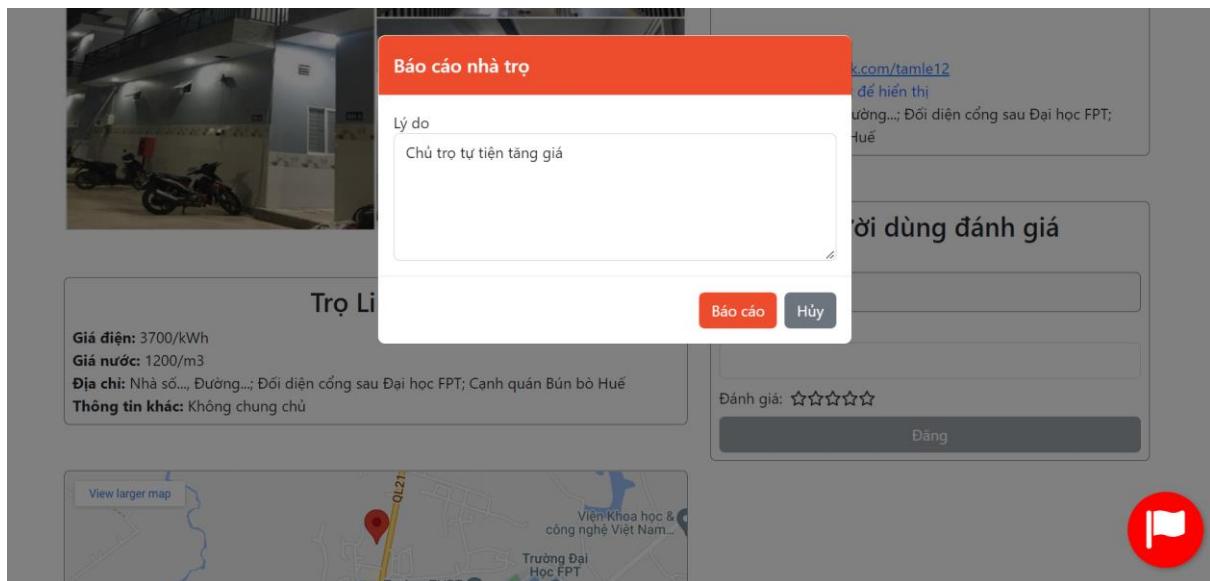
Nhà trọ đẹp, dịch vụ tốt

Đánh giá: ★★★★★

[Đăng](#)



Step 3: Input information in modal and click on “Gửi yêu cầu” button to send request



3.2.2.7 View room detail

Role: (Student/Guest)

Step 1: In House detail page click on room want to view detail room

Tên phòng	Giá phòng	Tiện ích	Loại phòng	Diện tích	Số người ở
101	đ2,900,000		Không khép kín	3 m ²	
102	đ2,900,000		Không khép kín	3 m ²	
103	đ2,900,000		Không khép kín	3 m ²	



Step 2: Redirect to Room detail page


FU House Finder

TÌM TRỌ SINH VIÊN FPT

[Kien nguyen tri ▾](#)

[Trang chủ](#) | [Tuyển sinh](#) | [Liên hệ](#) | [Giới thiệu](#)

[Trang chủ](#) > [Chi tiết nhà trọ](#) > Chi tiết phòng trọ



Thông tin mô tả

- | | |
|--|---|
| <input type="checkbox"/> Tủ lạnh: Có
<input checked="" type="checkbox"/> Máy giặt: Không
<input type="checkbox"/> Vệ sinh khép kín: Có
<input type="checkbox"/> Không chung chủ: Có | <input type="checkbox"/> Bếp: Không
<input checked="" type="checkbox"/> Bàn học: Có
<input type="checkbox"/> Giường: Có |
|--|---|

Loại phòng: Không khép kín

Không gian sinh hoạt riêng tư độc lập

Thông tin chủ trọ

Tên: Tâm Lê
Email: tamle@gmail.com
Số điện thoại: 0987654321
Facebook: facebook.com/tamle12



FPT Education
Hoa Lạc High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

 
[Sitemap website](#)
Copyright © 2017 FPT Education

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyên đất & người
FPT Edu - Tin tức chung
Bản chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế
Ngắn hạn
Dài hạn

3.2.3 Staff

3.2.3.1 View dashboard

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

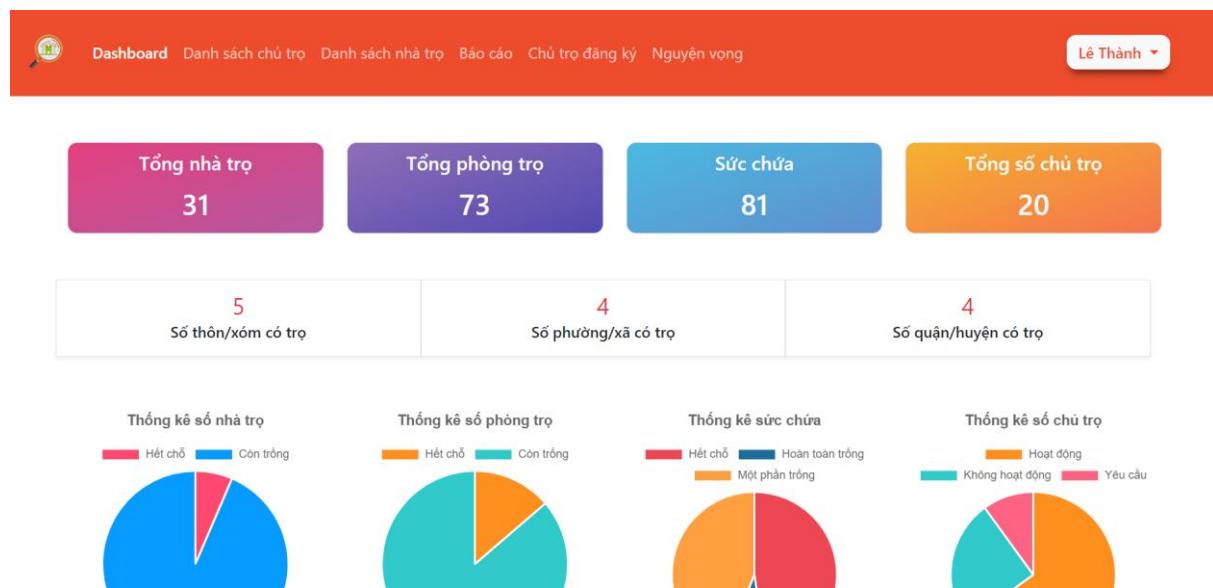
hoặc sử dụng

Email *

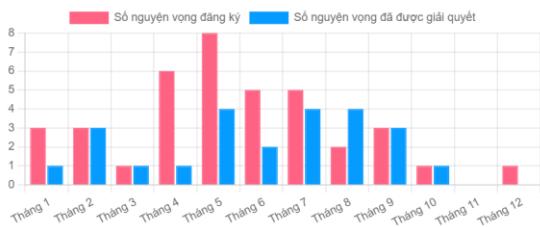
Mật khẩu *

[Đăng nhập](#)

Step 2: Redirect to Dashboard page



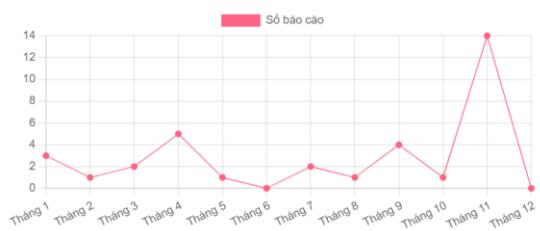
Thống kê số lượng nguyện vọng đăng ký nhà trọ từng tháng trong năm 2022



Thống kê tổng số lượng nguyện vọng đăng ký nhà trọ tới từng tháng trong năm 2022



Thống kê số lượng báo cáo nhà trọ năm 2022



3.2.3.2 View List Landlords

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

Continue with Facebook

Đăng nhập với tên Kien
trikienhachban@gmail.com

Google

hoặc sử dụng

Email *

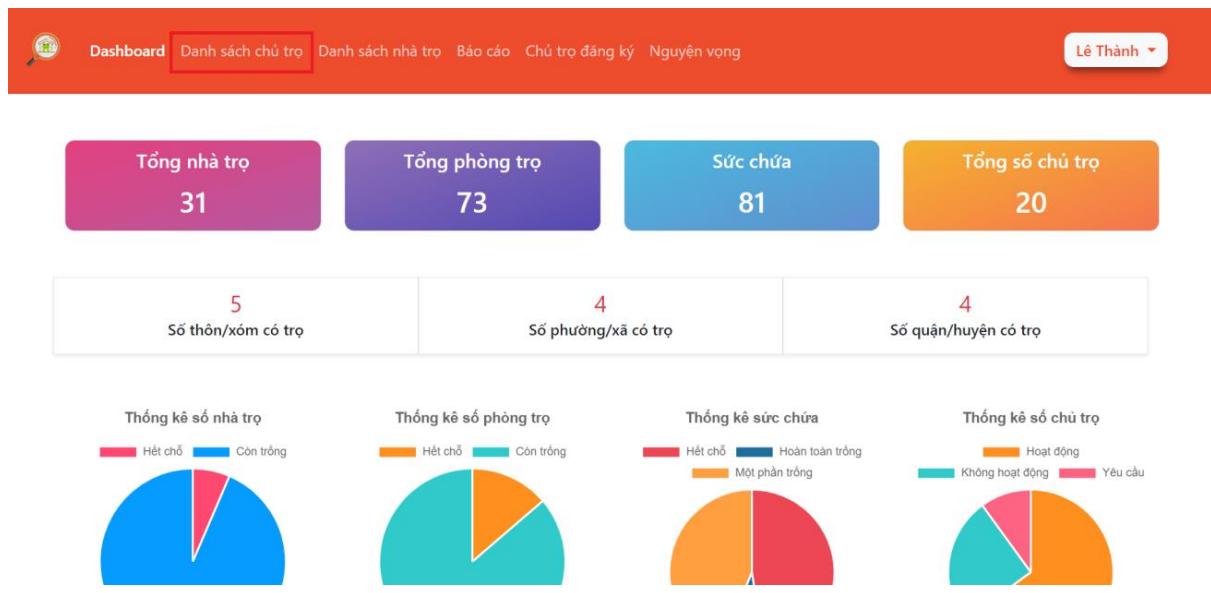
thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on “Danh sách chủ trọ” on header



Step 3: Redirect to List Landlord page

The list of landlords shows the following data:

Tổng nhà	Nhà trống	Tổng phòng	Phòng trống	Tổng chỗ	Chỗ trống
31	29	73	63	155	81

Bạn đang thấy 20 chủ trọ có 31 nhà trọ ở 5 thôn, 4 xã, 4 huyện

Danh sách chủ trọ

STT	Họ Tên	Số Điện Thoại	Tổng nhà	Tổng phòng	Phòng trống	Hành động
1	Bình Yên	0973866690	4	6	6	<input checked="" type="checkbox"/> Hoạt động
2	Dũng Nhung	0982298681	4	10	9	<input checked="" type="checkbox"/> Hoạt động
3	Hoàng Khánh	0989639985	1	3	3	<input checked="" type="checkbox"/> Hoạt động
4	Mai Linh	0846821118	1	3	3	<input checked="" type="checkbox"/> Hoạt động
5	Minh Hải	0365928071	0	0	0	<input type="checkbox"/> Dừng hoạt động
6	Nam Minh	0365928071	0	0	0	<input type="checkbox"/> Dừng hoạt động
7	Nga Hương	0984530814	7	13	9	<input checked="" type="checkbox"/> Hoạt động

3.2.3.3 View Landlord Detail

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

Continue with Facebook

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: click on landlord want to view detail

The dashboard shows the following statistics:

Tổng nhà	31	Nhà trống	29	Tổng phòng	73	Phòng trống	63	Tổng chỗ	155	Chỗ trống	81
----------	----	-----------	----	------------	----	-------------	----	----------	-----	-----------	----

Bạn đang thấy 20 chủ trọ có 31 nhà trọ ở 5 thôn, 4 xã, 4 huyện

Danh sách chủ trọ

STT	Họ Tên	Số Điện Thoại	Tổng nhà	Tổng phòng	Phòng trống	Hành động
1	Bình Yên	0973866690	4	6	6	<input checked="" type="checkbox"/> Hoạt động
2	Dũng Nhung	0982298681	4	10	9	<input checked="" type="checkbox"/> Hoạt động
3	Hoàng Khánh	0989639985	1	3	3	<input checked="" type="checkbox"/> Hoạt động
4	Mai Linh	0846821118	1	3	3	<input checked="" type="checkbox"/> Hoạt động
5	Minh Hải	0365928071	0	0	0	<input type="checkbox"/> Dừng hoạt động
6	Nam Minh	0365928071	0	0	0	<input type="checkbox"/> Dừng hoạt động
7	Nga Hương	0984530814	7	13	9	<input checked="" type="checkbox"/> Hoạt động

Step 3: Redirect to Landlord Detail page

[Danh sách chủ trọ](#) > Danh sách nhà trọ

Họ và tên: Bình Yên Facebook: facebook.com/binhyen56 Số điện thoại: 0973866690 Địa chỉ: Nhà số..., Đường...; Đổi diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế	Tổng số nhà trọ: 4 nhà Tổng số phòng: 6 phòng Tổng số phòng trống: 6 phòng
--	---

Danh sách nhà trọ

STT	Tên nhà trọ	Giá điện (VND)	Giá nước (VND)	♂	♀	WC	Tiện ích
1	Nhà trọ Bình Yên	3,500	1,300	0	3	3	∅
2	Trọ Campus 4	3,400	1,500	0	1	1	P
3	Trọ RoomType 2	3,400	1,500	0	1	1	P
4	Trọ District 2	3,400	1,500	0	1	1	P

3.2.3.4 View House Detail

Role: (Staff)

Step 1: Sign in with role is Staff



ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikinhthachban@gmail.com

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: click on house want to view detail

[Danh sách chủ trọ](#) > Danh sách nhà trọ

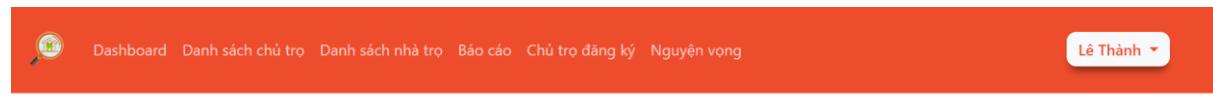
Họ và tên: Bình Yên
Facebook: facebook.com/binhyen56
Số điện thoại: 0973866690
Địa chỉ: Nhà số..., Đường...; Đối diện cổng sau Đại học FPT; Cạnh quán Bún bò Huế

Tổng số nhà trọ: 4 nhà
Tổng số phòng: 6 phòng
Tổng số phòng trống: 6 phòng

Danh sách nhà trọ

STT	Tên nhà trọ	Giá điện (VND)	Giá nước (VND)	♀ ♂	👤 🧑	🛏️	Tiện ích
1	Nhà trọ Bình Yên	3,500	1,300	0	3	3	∅
2	Trọ Campus 4	3,400	1,500	0	1	1	P
3	Trọ RoomType 2	3,400	1,500	0	1	1	P
4	Trọ District 2	3,400	1,500	0	1	1	P

Step 3: Redirect to House Detail page



Danh sách chủ trọ > [Danh sách nhà trọ](#) > Danh sách phòng trọ

	Tên nhà trọ: Nhà trọ Bình Yên Giá điện: 3500/kWh Giá nước: 1300/m ³ Địa chỉ: 7M3G+CR3, Kiên Sơn, Bình Xuyên, Vĩnh Phúc, Việt Nam Thông tin khác: Không chung chủ, giờ giấc thoải mái	Tổng số phòng trống: 9 phòng Tổng số chỗ trống: 0 chỗ
--	--	--

Danh sách phòng trọ

STT	Tên phòng	Giá phòng (VND)	Tiện ích	Diện tích	Số người ở	Trạng thái
1	101	3,100,000	▢ ▷ 🔍	4 m ²	👤 🧑	Còn Trống
2	102	2,400,000	▢ ▷ 🔍	4 m ²	👤 🧑	Còn Trống
3	103	2,200,000	▢ ▷ 🔍	4 m ²	👤 🧑	Còn Trống

3.2.3.5 View Room Detail

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: click on room want to view detail



Danh sách phòng trọ

STT	Tên phòng	Giá phòng (VND)	Tiện ích	Diện tích	Số người ở	Trạng thái
1	101	3,100,000		4 m ²		Còn Trống
2	102	2,400,000		4 m ²		Còn Trống
3	103	2,200,000		4 m ²		Còn Trống

Step 3: Redirect to Room Detail page

 Dashboard Danh sách chủ trọ Danh sách nhà trọ Báo cáo Chủ trọ đăng ký Nguyên vọng

Lê Thành ▾

[Danh sách chủ trọ](#) > [Danh sách nhà trọ](#) > [Danh sách phòng trọ](#) > Chi tiết nhà trọ



Thông tin mô tả

- Tủ lạnh: Có
- Máy giặt: Không
- Vệ sinh khép kín: Có
- Không chung chủ: Có

- Bếp: Không
- Bàn học: Có
- Giường: Không

Trật tự an ninh tốt



FPT Education
Hoa Lạc High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

 
[Sitemap website](#)
Copyright © 2017 FPT Education

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế
Thành công sau tốt nghiệp
FPT Edu - Chuyện đất & người
FPT Edu - Tin tức chung
Bản chuyên giáo dục
Phục vụ cộng đồng

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam
Cao đẳng FPT Polytechnic
Viện Đào tạo Quốc tế FPT
Melbourne Polytechnic Việt Nam

Sinh viên quốc tế

3.2.3.6 View List Houses

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

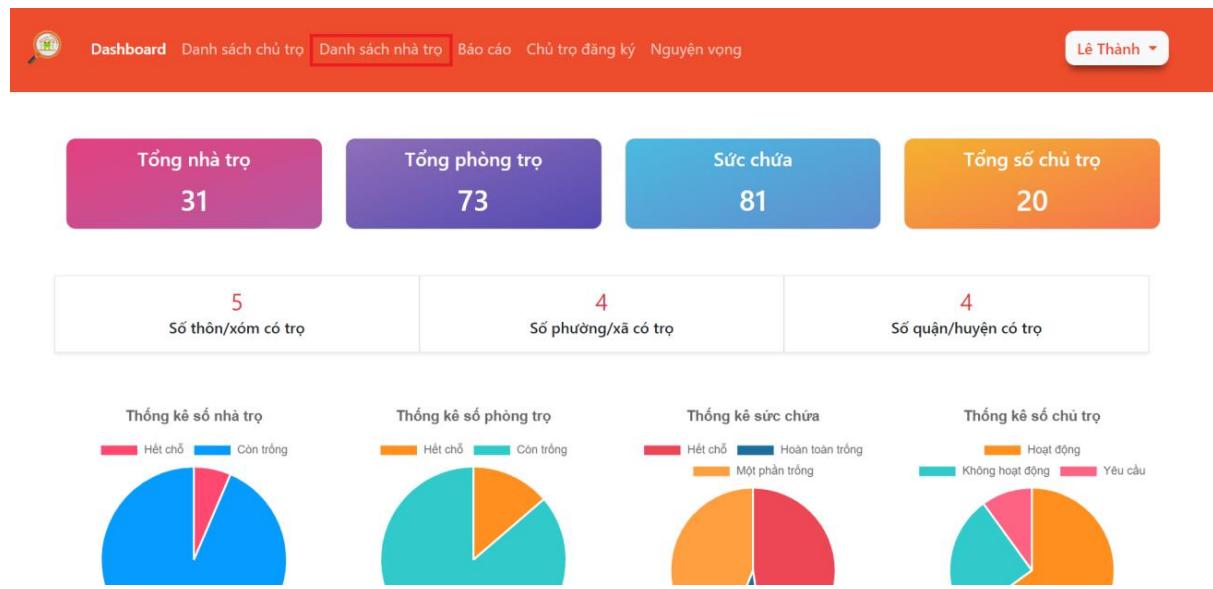
hoặc sử dụng

Email *

Mật khẩu *

Đăng nhập

Step 2: Click on “Danh sách nhà trọ” on header

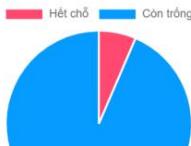


Danh sách chủ trọ Danh sách nhà trọ Báo cáo Chủ trọ đăng ký Nguyên vọng Lê Thành ▾

Tổng nhà trọ	Tổng phòng trọ	Sức chứa	Tổng số chủ trọ
31	73	81	20

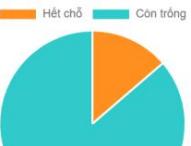
Số thôn/xóm có trọ	Số phường/xã có trọ	Số quận/huyện có trọ
5	4	4

Thống kê số nhà trọ



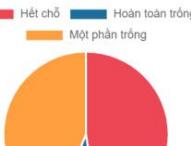
- Hết chỗ
- Còn trống

Thống kê số phòng trọ



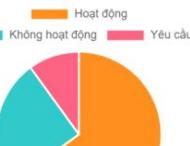
- Hết chỗ
- Còn trống

Thống kê sức chứa



- Hết chỗ
- Hoàn toàn trống
- Một phần trống

Thống kê số chủ trọ



- Hoạt động
- Không hoạt động
- Yêu cầu

Step 3: Redirect to List House page

Dashboard Danh sách chủ тро **Danh sách nhà trọ** Báo cáo Chủ тро đăng ký Nguyên vọng Lê Thành

31 Tổng nhà	29 Nhà trống	73 Tổng phòng	63 Phòng trống	155 Tổng chỗ	81 Chỗ trống
----------------	-----------------	------------------	-------------------	-----------------	-----------------

Bạn đang thấy 20 chủ trọ có 31 nhà trọ ở 5 thôn, 4 xã, 4 huyện

Danh sách nhà trọ

STT	Tên nhà trọ	Giá điện (VND)	Giá nước (VND)	👤	👤	🛏️	Tiện ích
1	Trọ Tâm Lê	3,700	1,200	1	3	5	∅
2	Trọ Tâm Thảo	3,500	1,300	2	0	4	▢
3	Trọ Hòa Lạc Yên Lạc Viên	3,400	1,500	0	0	0	∅
4	Nhà trọ Bình Yên	3,500	1,300	0	3	3	∅
5	Nhà trọ Tiến Phương	3,600	1,200	0	3	3	▢
6	Nhà trọ Phương Duy	3,800	1,300	0	3	3	▢

3.2.3.7 View List Reports

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

Continue with Facebook

Đăng nhập với tên Kien
trikienthachban@gmail.com

hoặc sử dụng

Email *

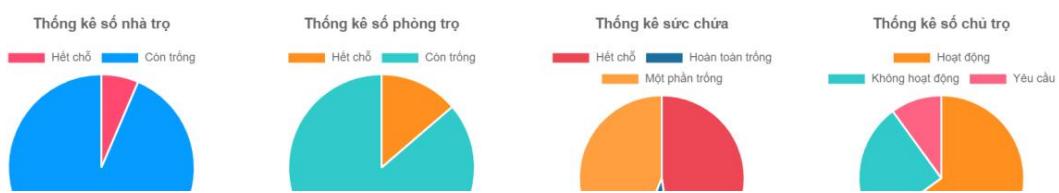
thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on “Báo cáo” on header



Step 3: Redirect to List Report page

Danh sách báo cáo

Báo cáo theo nhà

Danh sách báo cáo

Từ ngày	Đến ngày	Sắp xếp	Đặt lại bộ lọc						
mm/dd/yyyy	mm/dd/yyyy	Chọn sắp xếp	Đặt lại bộ lọc						
1 Nguyen The Giang	Nhà trọ Bình Yên	21/11/2022	Ngày báo cáo	22/11/2022	Ngày giải quyết	Staff 1	Người giải quyết	Chưa giải quyết	đèn TOEIC, viết tắt của Test of English for Intern...
2 Nguyen The Giang	Nhà trọ Bình Yên	21/11/2022	Ngày báo cáo	22/11/2022	Ngày giải quyết	Staff 1	Người giải quyết	Đang giải quyết	đèn, viết tắt của Test of English for International...
3 Nauyen The	HOLA Campus	21/11/2022	Ngày báo cáo		Ngày giải quyết		Người giải quyết	Chưa giải	Lorem ipsum

3.2.3.8 Filter Reports

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Filter report

Dashboard Danh sách chủ trọ Danh sách nhà trọ Báo cáo Chủ trọ đăng ký Nguyên vọng Lê Thành ▾

Danh sách báo cáo Báo cáo theo nhà

Danh sách báo cáo

Từ ngày	Đến ngày	Sắp xếp	Đặt lại bộ lọc							
mm/dd/yyyy	mm/dd/yyyy	Chọn sắp xếp								
21/11/2022	22/11/2022	Chọn sắp xếp								
1 Nguyen The Giang	Nhà trọ Bình Yên	21/11/2022	Ngày báo cáo	22/11/2022	Ngày giải quyết	Staff 1	Người giải quyết	Chưa giải quyết	Trạng thái	Nội dung
2 Nguyen The Giang	Nhà trọ Bình Yên	21/11/2022	Ngày báo cáo	22/11/2022	Ngày giải quyết	Staff 1	Người giải quyết	Đang giải quyết	Trạng thái	Nội dung
3 Nguyen The	HOT A Campus	21/11/2022	Ngày báo cáo	22/11/2022	Ngày giải quyết		Người giải quyết	Chưa giải	Trạng thái	Nội dung

Danh sách báo cáo Báo cáo theo nhà

Danh sách nhà bị báo cáo

Trang thái	Sắp xếp
Tất cả	Báo cáo nhiều nhất

STT	Nhà trọ	Chủ trọ	Số lượt báo cáo	Trạng thái chủ trọ
1	Nhà trọ Bình Yên	Bình Yên	8	<input checked="" type="checkbox"/> Hoạt động
2	HOLA Campus	Mai Linh	7	<input checked="" type="checkbox"/> Hoạt động
3	Trọ Tâm Lê	Tâm Lê	5	<input checked="" type="checkbox"/> Hoạt động
4	Nhà trọ Phương Duy	Ngọc Hương	4	<input checked="" type="checkbox"/> Hoạt động

Step 3: Show list reports found

Danh sách báo cáo Báo cáo theo nhà

Danh sách báo cáo

Từ ngày	Đến ngày	Sắp xếp
12/26/2021	12/16/2022	Ngày gần nhất

[Đặt lại bộ lọc](#)

STT	Tên sinh viên	Nhà trọ	Ngày báo cáo	Ngày giải quyết	Người giải quyết	Trạng thái	Nội dung
1	Nguyen The Giang	Nhà trọ Bình Yên	21/11/2022	22/11/2022	Staff 1	Chưa giải quyết	đèn TOEIC, viết tắt của Test of English for Intern...
2	Nguyen The Giang	Nhà trọ Bình Yên	21/11/2022	22/11/2022	Staff 1	Đang giải quyết	đèn, viết tắt của Test of English for International...
3	Nguyen The	HOLA Campus	21/11/2022			Chưa giải	Lorum ipsum

Danh sách báo cáo Báo cáo theo nhà

Danh sách nhà bị báo cáo

STT	Nhà trọ	Chủ trọ	Số lượt báo cáo	Trạng thái chủ trọ
1	Tro Tâm Thảo	Dũng Nhung	1	<input checked="" type="checkbox"/> Hoạt động
2	Kí túc xá Ông bà	Phương Oanh	1	<input checked="" type="checkbox"/> Hoạt động
3	Nhà trọ Thái Hà	Thái Hà	1	<input checked="" type="checkbox"/> Hoạt động
4	Tro Việt Dũng	Việt Dũng	1	<input checked="" type="checkbox"/> Hoạt động

3.2.3.9 View List Sign up Requests

Role: (Staff)

Step 1: Sign in with role is Staff



ĐĂNG NHẬP

[Continue with Facebook](#)

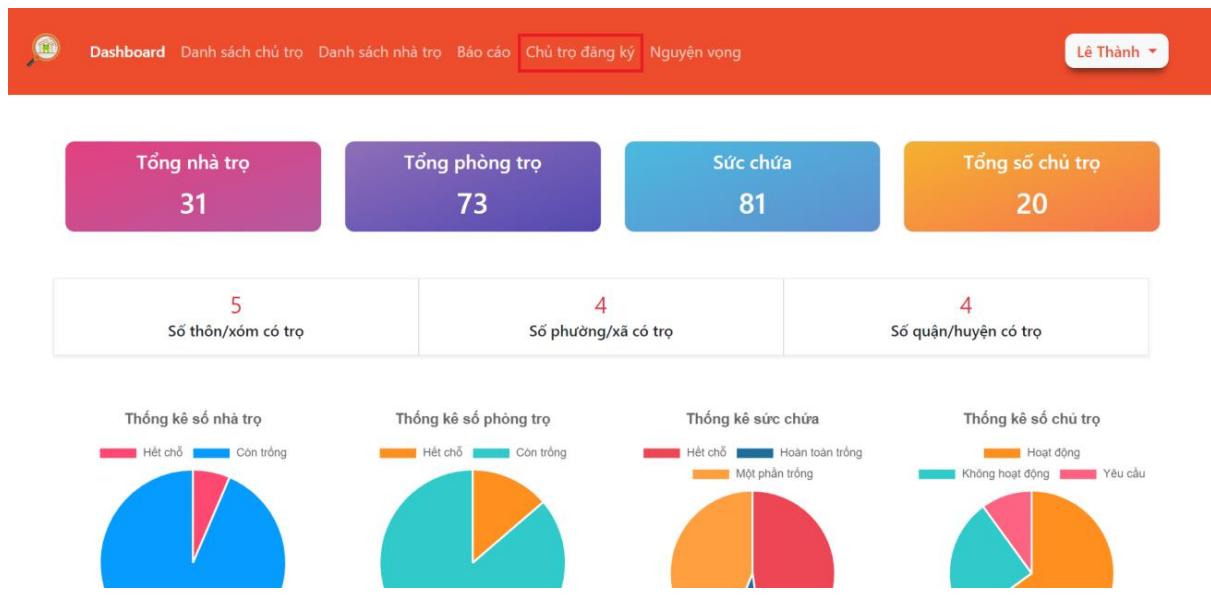
Đăng nhập với tên Kien
trikienthachban@gmail.com

hoặc sử dụng

Email *

Mật khẩu *

Step 2: Click on “Chủ trọ đăng ký” on header



Step 3: Redirect to List Request page

Danh sách yêu cầu chủ trọ

Danh sách chủ trọ bị từ chối

Họ Tên	Email	Số Điện Thoại	Ảnh Chứng Minh Thư	Ngày Đăng Ký	Hành động
Hải Nam	hainam@gmail.com	0365928071		12/12/2022	Chấp Nhận Từ Chối
Kim Anh	kimanh@gmail.com	0365928071		12/12/2022	Chấp Nhận Từ Chối
Minh Anh	minhanh@gmail.com	0365928071		12/12/2022	Chấp Nhận Từ Chối
Minh Vũ	minhv@gmail.com	0365928072		12/12/2022	Chấp Nhận Từ Chối
Phương Nguyễn	phuongnguyen@gmail.com	0365928071		12/12/2022	Chấp Nhận Từ Chối

3.2.3.10 View List Orders

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on “Nguyễn vọng” on header

The dashboard page includes the following data:

- Tổng nhà trọ: 31
- Tổng phòng trọ: 73
- Sức chứa: 81
- Tổng số chủ trọ: 20
- Số thôn/xóm có trọ: 5
- Số phường/xã có trọ: 4
- Số quận/huyện có trọ: 4

Four pie charts provide detailed breakdowns:

- Thống kê số nhà trọ:
 - Hết chỗ: 1
 - Còn trống: 30
- Thống kê số phòng trọ:
 - Hết chỗ: 10
 - Còn trống: 63
- Thống kê sức chứa:
 - Hết chỗ: 10
 - Hoàn toàn trống: 1
 - Một phần trống: 70
- Thống kê số chủ trọ:
 - Hoạt động: 15
 - Không hoạt động: 5
 - Yêu cầu: 0

Step 3: Redirect to List Order page

STT	Mã sinh viên	Tên sinh viên	Ngày tạo	Ngày giải quyết	Người giải quyết	Trạng thái
1	HE172885	Nguyễn Trí Kiên	15/12/2022			Chưa giải quyết
2	HE150432	Nguyễn Thu An	03/10/2022	22/10/2022	Staff 3	Đã giải quyết
3	HE153046	Bùi Ngọc Huyền	05/09/2022			Đang giải quyết
4	HE153590	Đinh Thế Thuận	03/09/2022	21/09/2022	Staff 3	Đã giải quyết
5	HE153590	Đinh Thế Thuận	03/09/2022	10/09/2022	Lê Thành	Đã giải quyết
6	HE153222	Trần Thị Nguyệt Hà	03/08/2022			Đang giải quyết
7	HE150600	Nguyễn Minh Hạnh	03/08/2022	10/08/2022	Lê Thành	Đã giải quyết
8	HE153046	Bùi Ngọc Huyền	05/07/2022			Chưa giải quyết
9	HE153222	Trần Thị Nguyệt Hà	03/07/2022			Chưa giải quyết
10	HE153590	Đinh Thế Thuận	03/07/2022	10/07/2022	Lê Thành	Đã giải quyết

FPT Education
FPT EDUCATION GLOBAL

FPT Education
Hoa Lạc High Tech Park, Hanoi, Vietnam
Email: fpteducation@fe.edu.vn

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm

Tuyển sinh
Đại học FPT
Greenwich Việt Nam

3.2.3.11 Filter Orders

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Filter order

Danh sách nguyện vọng						
Từ ngày	Đến ngày	Trạng thái	Sắp xếp			
mm/dd/yyyy	mm/dd/yyyy	Tất cả	Chọn sắp xếp	Đặt lại bộ lọc		
STT	Mã sinh viên	Tên sinh viên	Ngày tạo	Ngày giải quyết	Người giải quyết	Trạng thái
1	HE172885	Nguyễn Trí Kiên	15/12/2022			Chưa giải quyết
2	HE150432	Nguyễn Thu An	03/10/2022	22/10/2022	Staff 3	Đã giải quyết
3	HE153046	Bùi Ngọc Huyền	05/09/2022			Đang giải quyết
4	HE153590	Đinh Thế Thuận	03/09/2022	21/09/2022	Staff 3	Đã giải quyết
5	HE153590	Đinh Thế Thuận	03/09/2022	10/09/2022	Lê Thành	Đã giải quyết
6	HE153222	Trần Thị Nguyệt Hà	03/08/2022			Đang giải quyết
7	HE150600	Nguyễn Minh Hanh	03/08/2022	10/08/2022	Lê Thành	Đã giải quyết
8	HE153046	Bùi Ngọc Huyền	05/07/2022			Chưa giải quyết
9	HE153222	Trần Thị Nguyệt Hà	03/07/2022			Chưa giải quyết

Step 3: Show list orders found

Danh sách nguyện vọng						
Từ ngày	Đến ngày	Trạng thái	Sắp xếp			
01/16/2022	12/16/2022	Chưa giải quyết	Ngày gần nhất			Đặt lại bộ lọc
STT	Mã sinh viên	Tên sinh viên	Ngày tạo	Ngày giải quyết	Người giải quyết	Trạng thái
1	HE172885	Nguyễn Trí Kiên	15/12/2022			Chưa giải quyết
2	HE153046	Bùi Ngọc Huyền	05/07/2022			Chưa giải quyết
3	HE153222	Trần Thị Nguyệt Hà	03/07/2022			Chưa giải quyết
4	HE153299	Tống Trường Giang	16/05/2022			Chưa giải quyết
5	HE153299	Tống Trường Giang	30/04/2022			Chưa giải quyết
6	HE150691	Nguyễn Trần Hoàng	13/04/2022			Chưa giải quyết
7	HE153299	Tống Trường Giang	03/04/2022			Chưa giải quyết
8	HE153299	Tống Trường Giang	31/03/2022			Chưa giải quyết
9	HE153046	Bùi Ngọc Huyền	03/02/2022	10/02/2022	Lê Thành	Chưa giải quyết

3.2.3.12 Update profile

Role: (Staff)

Step 1: Sign in with role is Staff



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

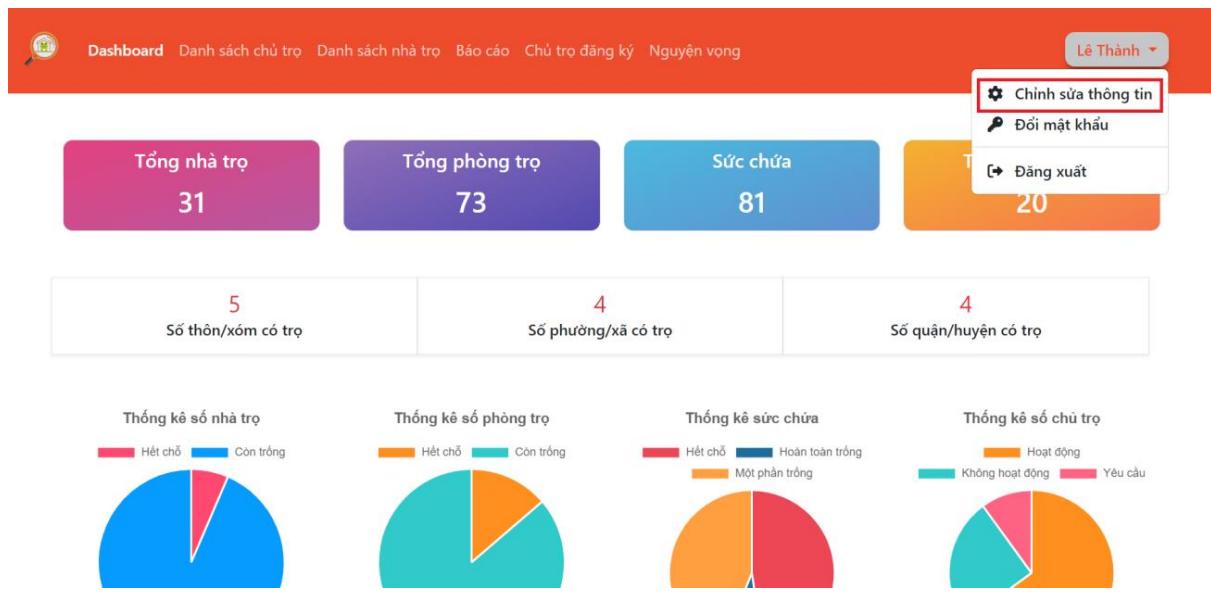
[Continue with Facebook](#)
[Đăng nhập với tên Kien trikienthachban@gmail.com](#)

hoặc sử dụng

Email *

Mật khẩu *

Step 2: Click on “Chỉnh sửa thông tin” on header



Step 3: Update name and email and click on “Thay đổi” button

The update form contains fields for:

- Họ và tên: Lê Thành
- Email: thanhle@gmail.com

A green 'Thay đổi' (Change) button is at the bottom.

At the bottom of the page, there is a footer with the following information:

- FPT Education Global**
- FPT Education**: Hoa Lac High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn
- Tin tức**: Học tập khác biệt
Môi trường giàu trải nghiệm
- Tuyển sinh**: Đại học FPT
Greenwich Việt Nam

Step 4: Show message “Cập nhật thành công”

Cập nhật thành công

OK

Cập nhật thông tin tuyển sinh

Họ và tên

Lê Thành

Email

thanhle@gmail.com

Thay đổi



3.2.3.13 Change password

Role: (Staff)

Step 1: Sign in with role is Staff

FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

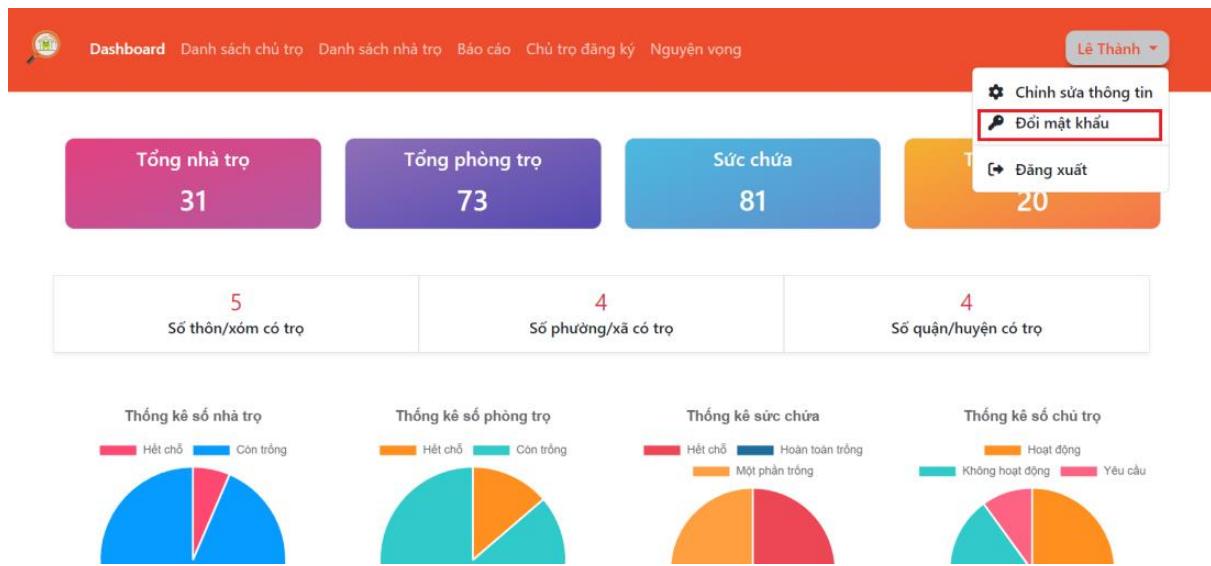
hoặc sử dụng

Email *
thanhle@gmail.com

Mật khẩu *
.....

Đăng nhập

Step 2: Click on “Đổi mật khẩu” on header



Step 3: Update password and click on “Thay đổi” button

The form is titled **Đổi mật khẩu** (Change Password) and contains the following fields:

- Nhập mật khẩu cũ: (Redacted)
- Nhập mật khẩu mới: (Redacted)
- Nhập lại mật khẩu mới: (Redacted)
- A green **Thay đổi** (Change) button.

3.2.4 Admin

3.2.4.1 View list staffs

Role: (Admin)

Step 1: Sign in with role is Admin



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Redirect to List Staff page

TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu Admin ▾

Danh sách tuyển sinh

+ Thêm mới

STT	Họ và tên	Email	Vị trí	Hành động
1	Lê Thành	thanhle@gmail.com	Trưởng phòng tuyển sinh	
2	Staff 1	staff1@gmail.com	Cán bộ phòng tuyển sinh	
3	Staff 2	staff2@gmail.com	Cán bộ phòng tuyển sinh	
4	Staff 3	staff3@gmail.com	Nhân viên phòng phục vụ sinh viên	

3.2.4.2 Add new staff

Role: (Admin)

Step 1: Sign in with role is Admin



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Click on “Thêm mới” button

TÌM TRỌ SINH VIÊN FPT

FU House Finder Admin ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Danh sách tuyển sinh

STT	Họ và tên	Email	Vị trí	Hành động
1	Lê Thành	thanhle@gmail.com	Trưởng phòng tuyển sinh	
2	Staff 1	staff1@gmail.com	Cán bộ phòng tuyển sinh	
3	Staff 2	staff2@gmail.com	Cán bộ phòng tuyển sinh	
4	Staff 3	staff3@gmail.com	Nhân viên phòng phục vụ sinh viên	

Step 3: Redirect to Create Account page, input information of staff and click on “Tạo mới” button

Tạo tài khoản tuyển sinh

Họ và tên
Thành Đinh

Email
thanhle@gmail.com

Mật khẩu
.....

Vị trí
Head of Admission Department

Tạo mới

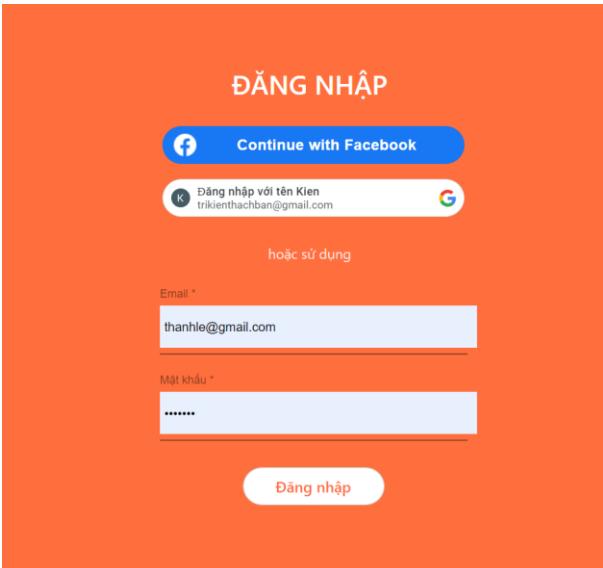
3.2.4.3 Update staff

Role: (Admin)

Step 1: Sign in with role is Admin



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT



The screenshot shows the login page for the FU House Finder application. The background is orange. At the top right, it says "ĐĂNG NHẬP". Below that is a blue button with "Continue with Facebook" and a Facebook icon. To the right of the button is a placeholder for a Google account: "Đăng nhập với tên Kien trikienthachban@gmail.com" next to a Google icon. Below these options is the text "hoặc sử dụng". There are two input fields: "Email *" containing "thanhle@gmail.com" and "Mật khẩu *" containing ".....". At the bottom right is a white button with the text "Đăng nhập" in red.

Step 2: Click on icon on table



TÌM TRỌ SINH VIÊN FPT

FU House Finder

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Admin ▾

Danh sách tuyển sinh

+ Thêm mới

STT	Họ và tên	Email	Vị trí	Hành động
1	Lê Thành	thanhle@gmail.com	Trưởng phòng tuyển sinh	 
2	Staff 1	staff1@gmail.com	Cán bộ phòng tuyển sinh	 
3	Staff 2	staff2@gmail.com	Cán bộ phòng tuyển sinh	 
4	Staff 3	staff3@gmail.com	Nhân viên phòng phục vụ sinh viên	 

Step 3: Redirect to Update Account page, input information of staff want to change and click on “Thay đổi” button

Cập nhật tuyển sinh

Họ và tên

Email

Vị trí

Thay đổi

3.2.4.4 Delete staff

Role: (Admin)

Step 1: Sign in with role is Admin



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on icon on table

TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu Admin ▾

Danh sách tuyển sinh

+ Thêm mới

STT	Họ và tên	Email	Vị trí	Hành động
1	Lê Thành	thanhle@gmail.com	Trưởng phòng tuyển sinh	
2	Staff 1	staff1@gmail.com	Cán bộ phòng tuyển sinh	
3	Staff 2	staff2@gmail.com	Cán bộ phòng tuyển sinh	
4	Staff 3	staff3@gmail.com	Nhân viên phòng phục vụ sinh viên	

3.2.5 Landlord

3.2.5.1 View dashboard

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Redirect to Dashboard page, landlord can view their list house

FU House Finder

TÌM TRỌ SINH VIÊN FPT

Dũng Nhung ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

4 Nhà	10 Phòng	9 Phòng trống					
Danh sách nhà trọ							
+ Thêm mới							
STT	Tên nhà trọ	Tiền chưa cho thuê	2	0	4	Tiện ích	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000				<input type="checkbox"/>	
2	Trọ Thủ Thảo	đ9,000,000	0	3	3	<input type="checkbox"/>	
3	Trọ Tâm Lê	đ8,250,000	0	3	3	<input type="checkbox"/>	
4	Trọ Campus 2	đ3,000,000	0	1	1	<input checked="" type="checkbox"/>	

3.2.5.2 Add new house

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on “Thêm mới” button

TÌM TRỌ SINH VIÊN FPT

Dũng Nhung ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

4 Nhà | 10 Phòng | 9 Phòng trống

Danh sách nhà trọ

+ Thêm mới

STT	Tên nhà trọ	Tiền chưa cho thuê	2	0	4	⋮	Tiện ích	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000	2	0	4	<input type="checkbox"/>		
2	Trọ Thủ Thảo	đ9,000,000	0	3	3	<input type="checkbox"/>		
3	Trọ Tâm Lê	đ8,250,000	0	3	3	<input type="checkbox"/>		
4	Trọ Campus 2	đ3,000,000	0	1	1	<input checked="" type="checkbox"/>		

Step 3: Redirect to Add House page, input information of new house and click on “Thêm mới” button



FU House Finder

TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

Thêm nhà trọ mới

Tên nhà trọ <input type="text" value="Nhập tên nhà trọ"/>	Ảnh nhà (1) <input type="file"/> Choose File No file chosen
Cơ sở <input type="text"/>	Ảnh nhà (2) <input type="file"/> Choose File No file chosen
Quận/Huyện <input type="text"/>	Ảnh nhà (3) <input type="file"/> Choose File No file chosen
Phường/Xã <input type="text"/>	Giá điện trên tháng (VNĐ) <input type="text" value="Nhập giá điện"/>
Thôn/Xóm <input type="text"/>	Giá nước trên tháng (VNĐ) <input type="text" value="Nhập giá nước"/>
Địa chỉ <input type="text" value="Nhập địa chỉ"/>	Tiện ích <input type="checkbox"/> Khóa vân tay <input type="checkbox"/> Camera giám sát <input type="checkbox"/> Chỗ để xe
Địa điểm Google Maps 	Thông tin mô tả <input type="text"/>

Thêm mới

3.2.5.3 Update house

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on update icon

TÌM TRỌ SINH VIÊN FPT

Dũng Nhung ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

4 Nhà | 10 Phòng | 9 Phòng trống

Danh sách nhà trọ

+ Thêm mới

STT	Tên nhà trọ	Tiền chưa cho thuê	Giới tính	Số phòng	Tiện ích	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000	2	0	4	
2	Trọ Thu Thảo	đ9,000,000	0	3	3	
3	Trọ Tâm Lê	đ8,250,000	0	3	3	
4	Trọ Campus 2	đ3,000,000	0	1	1	

Step 3: Redirect to Update House page, input information of house want to update and click on “Chỉnh sửa” button



FU House Finder

TÌM TRỌ SINH VIÊN FPT

[Trang chủ](#) | [Tuyển sinh](#) | [Liên hệ](#) | [Giới thiệu](#)

Dũng Nhung ▾

Chỉnh sửa nhà trọ

Tên nhà trọ

Trọ Tâm Thảo

Cơ sở

FU - Hòa Lạc

Quận/Huyện

Huyện Thạch Thất

Phường/Xã

Thị trấn Liên Quan

Thôn/Xóm

Đồng Cam

Địa chỉ

Nhách 75, thôn 4, Thạch Hoà, Thạch Thất, Hà Nội, Việt Nam

Giá điện / số (VNĐ)

3500

Giá nước / khói (VNĐ)

1300

Tiện ích Khóa vân tay Camera giám sát Chỗ để xe**Thông tin mô tả**

Rất đẹp

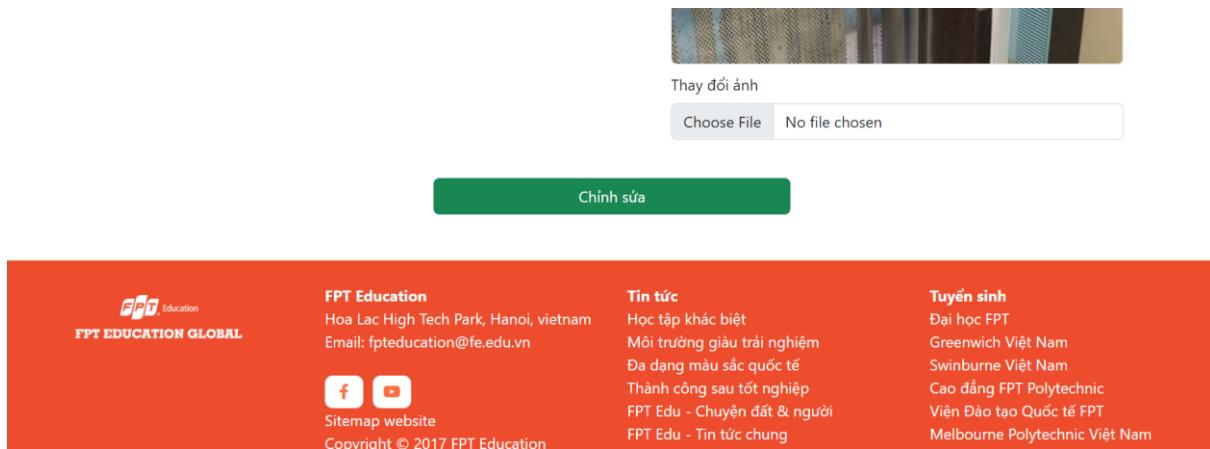
Ảnh nhà 1

Thay đổi ảnh

 No file chosen**Ảnh nhà 2**

Thay đổi ảnh

 No file chosen**Ảnh nhà 3**



3.2.5.4 Delete house

Role: (Landlord)

Step 1: Sign in with role is Landlord

The image contains two parts. On the left is the FU House Finder logo, which features a magnifying glass over a house with a frog inside. Below the logo is the text "FU HOUSE FINDER" and "Ứng dụng tìm trọ dành cho sinh viên Đại học FPT". On the right is a login form titled "ĐĂNG NHẬP". It includes social login buttons for Facebook and Google, an email input field with placeholder "thanhle@gmail.com", a password input field with placeholder ".....", and a "Đăng nhập" button.

Step 2: Click on delete icon

The screenshot shows the homepage of the FU House Finder website. At the top, there is a navigation bar with the title "TÌM TRỌ SINH VIÊN FPT" and a dropdown menu "Dũng Nhung". Below the navigation bar, there are three categories: "Nhà" (House) with value 4, "Phòng" (Room) with value 10, and "Phòng trống" (Empty Room) with value 9. The main content area is titled "Danh sách nhà trọ" (List of dormitories) and contains a table with four rows of data:

STT	Tên nhà trọ	Tiền chưa cho thuê	Người lớn	Giường	Tiện ích	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000	2	0	4	
2	Trọ Thu Thảo	đ9,000,000	0	3	3	
3	Trọ Tâm Lê	đ8,250,000	0	3	3	
4	Trọ Campus 2	đ3,000,000	0	1	1	

3.2.5.5 View rate

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

The screenshot shows the login page for the application. It features a large orange header with the text "ĐĂNG NHẬP". Below the header are two blue buttons for "Continue with Facebook" and "Đăng nhập với tên Kien trikienthachban@gmail.com" (using name Kien trikienthachban@gmail.com). There is also a "G" button for Google. Below these fields is the text "hoặc sử dụng". The next section contains two input fields: "Email *" with the placeholder "thanhle@gmail.com" and "Mật khẩu *" with a redacted password. A "Đăng nhập" (Login) button is located at the bottom right.

Step 2: Click on rate icon



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

4 Nhà	10 Phòng	9 Phòng trống
----------	-------------	------------------

Danh sách nhà trọ

+ Thêm mới

STT	Tên nhà trọ	Tiền chưa cho thuê	Người	Giới tính	Phòng	Tiện ích	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000	2	0	4	<input type="checkbox"/>	
2	Trọ Thu Thảo	đ9,000,000	0	3	3	<input type="checkbox"/>	
3	Trọ Tâm Lê	đ8,250,000	0	3	3	<input type="checkbox"/>	
4	Trọ Campus 2	đ3,000,000	0	1	1	<input checked="" type="checkbox"/>	

Step 3: Redirect to Rate House page



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

Người dùng đánh giá

Bình luận:

HE153046: Rất tuyệt vời, gần trường nữa 2 ()

→ Chủ trọ trả lời: Cảm ơn bạn 2

HE153046: Rất tuyệt vời, gần trường nữa 7 ()

→ Chủ trọ trả lời: Cảm ơn bạn 7

SA000001: Dịch vụ tốt ([phản hồi](#))



FPT Education
Hoa Lạc High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

Tin tức
Học tập khác biệt
Môi trường giàu trải nghiệm
Đa dạng màu sắc quốc tế

Tuyển sinh
Đại học FPT
Greenwich Việt Nam
Swinburne Việt Nam

3.2.5.6 Reply comment

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Click on “phản hồi”



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

Người dùng đánh giá

Bình luận:

HE153046: Rất tuyệt vời, gần trường nữa 2 ()
 → Chủ trọ trả lời: Cảm ơn bạn 2

HE153046: Rất tuyệt vời, gần trường nữa 7 ()
 → Chủ trọ trả lời: Cảm ơn bạn 7

SA000001: Dịch vụ tốt () [phản hồi](#)

 FPT Education Hoa Lac High Tech Park, Hanoi, vietnam Email: fpteducation@fe.edu.vn	Tin tức Học tập khác biệt Môi trường giàu trải nghiệm Đa dạng màu sắc quốc tế	Tuyển sinh Đại học FPT Greenwich Việt Nam Swinburne Việt Nam
---	---	--

Step 3: Input content to reply this comment and click on “Đăng” button



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

Người dùng đánh giá

Bình luận:

HE153046: Rất tuyệt vời, gần trường nữa 2 (★★★★★)
 → Chủ trọ trả lời: Cảm ơn bạn 2

HE153046: Rất tuyệt vời, gần trường nữa 7 (★★★★)
 → Chủ trọ trả lời: Cảm ơn bạn 7

SA000001: Dịch vụ tốt (★★★★★)

FPT Education

FPT Education
Hoa Lạc High Tech Park, Hanoi, Vietnam

Tin tức
Học tập khác biệt

Tuyển sinh
Đại học FPT

3.2.5.7 View house detail & list rooms

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)



hoặc sử dụng

Email *

Mật khẩu *

Step 2: Click on house want to view detail house

The screenshot shows the homepage of the FU House Finder website. At the top, there's a navigation bar with the title "TÌM TRỌ SINH VIÊN FPT" and a dropdown menu for "Dũng Nhung". Below the navigation bar, there are three categories: "Nhà" (House) with 4 results, "Phòng" (Room) with 10 results, and "Phòng trống" (Empty Room) with 9 results. The main content area is titled "Danh sách nhà trọ" (List of dormitories) and displays a table with four rows of dormitory information. Each row includes a red-bordered "Edit" icon.

STT	Tên nhà trọ	Tiền chưa cho thuê	Người lớn	Giới tính	Điều kiện	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000	2	0	4	
2	Trọ Thu Thảo	đ9,000,000	0	3	3	
3	Trọ Tâm Lê	đ8,250,000	0	3	3	
4	Trọ Campus 2	đ3,000,000	0	1	1	

Step 3: Redirect to House detail page, landlord can view list room of this house

The screenshot shows the "Danh sách phòng" (List of rooms) page for the dormitory "Trọ Tâm Thảo". The top navigation bar remains the same. The search filters are set to "Phòng trống" (2), "Phòng trống một phần" (0), "Chỗ trống" (4), and "Tiền chưa cho thuê" (đ10,500,000). The main content area is titled "Danh sách phòng" and displays a table with three rows of room information. Each row includes a red-bordered "Edit" icon.

Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn
101	đ4,000,000		6 m ²	2	Còn Trống	
102	đ3,500,000		6 m ²	2	Không dùng được	
103	đ3,000,000		6 m ²	2	Còn Trống	

3.2.5.8 Add new room

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER
Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Click on “Thêm mới” button



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

2 Phòng trống	0 Phòng trống một phần	4 Chỗ trống	đ10,500,000 Tiền chưa cho thuê																												
Danh sách phòng																															
+ Thêm mới <table border="1"> <thead> <tr> <th>Tên phòng</th> <th>Giá phòng</th> <th>Tiện ích</th> <th>Diện tích</th> <th>Số người ở</th> <th>Trạng thái</th> <th>Tùy chọn</th> </tr> </thead> <tbody> <tr> <td>101</td> <td>đ4,000,000</td> <td></td> <td>6 m²</td> <td>2</td> <td><input checked="" type="checkbox"/> Còn Trống</td> <td></td> </tr> <tr> <td>102</td> <td>đ3,500,000</td> <td></td> <td>6 m²</td> <td>2</td> <td><input type="checkbox"/> Không dùng được</td> <td></td> </tr> <tr> <td>103</td> <td>đ3,000,000</td> <td></td> <td>6 m²</td> <td>2</td> <td><input checked="" type="checkbox"/> Còn Trống</td> <td></td> </tr> </tbody> </table>				Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn	101	đ4,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống		102	đ3,500,000		6 m ²	2	<input type="checkbox"/> Không dùng được		103	đ3,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống	
Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn																									
101	đ4,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống																										
102	đ3,500,000		6 m ²	2	<input type="checkbox"/> Không dùng được																										
103	đ3,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống																										

FPT Education | Tin tức | Tuyển sinh

Step 3: Redirect to Add Room page, input information of new house and click on “Thêm mới” button



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

Thêm 1 phòng
[Thêm nhiều phòng](#)

Tòa nhà	<input type="text"/>	Loại phòng	<input type="text"/>
Tầng	<input type="text"/>	<input type="checkbox"/> Tủ lạnh	<input type="checkbox"/> Bếp
Tên phòng	<input type="text"/>	<input type="checkbox"/> Bàn	<input type="checkbox"/> Máy giặt
Giá	<input type="text"/>	<input type="checkbox"/> Vệ sinh khép kin	<input type="checkbox"/> Không chung chủ
Thông tin	<input type="text"/>	Ảnh 1	<input type="button" value="Choose File"/> No file chosen
Diện tích	<input type="text"/>	Ảnh 2	<input type="button" value="Choose File"/> No file chosen
Số người tối đa	<input type="text"/>	Ảnh 3	<input type="button" value="Choose File"/> No file chosen
Số người hiện tại	<input type="text"/>		

Thêm mới



FPT Education
Hoa Lạc High Tech Park, Hanoi, vietnam
Email: fpteducation@fe.edu.vn

 
[Sitemap website](#)
Copyright © 2017 FPT Education

Tin tức

- Học tập khác biệt
- Môi trường giàu trải nghiệm
- Đa dạng màu sắc quốc tế
- Thành công sau tốt nghiệp
- FPT Edu - Chuyên đất & người
- FPT Edu - Tin tức chung
- Bản chuyên giáo dục
- Phục vụ cộng đồng

Tuyển sinh

- Đại học FPT
- Greenwich Việt Nam
- Swinburne Việt Nam
- Cao đẳng FPT Polytechnic
- Viện Đào tạo Quốc tế FPT
- Melbourne Polytechnic Việt Nam

Sinh viên quốc tế

- Ngân hạn
- Dài hạn

3.2.5.9 Add multiple rooms

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

Đăng nhập với tên Kien
trikienthachban@gmail.com

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Click on “Thêm mới” button



TÌM TRỌ SINH VIÊN FPT

Dũng Nhung ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

2	0	4	đ10,500,000																												
Phòng trống	Phòng trống một phần	Chỗ trống	Tiền chưa cho thuê																												
Danh sách phòng																															
+ Thêm mới <table border="1"> <thead> <tr> <th>Tên phòng</th> <th>Giá phòng</th> <th>Tiện ích</th> <th>Diện tích</th> <th>Số người ở</th> <th>Trạng thái</th> <th>Tùy chọn</th> </tr> </thead> <tbody> <tr> <td>101</td> <td>đ4,000,000</td> <td> </td> <td>6 m²</td> <td> </td> <td><input checked="" type="checkbox"/> Còn Trống</td> <td> </td> </tr> <tr> <td>102</td> <td>đ3,500,000</td> <td> </td> <td>6 m²</td> <td></td> <td><input type="radio"/> Không dùng được</td> <td> </td> </tr> <tr> <td>103</td> <td>đ3,000,000</td> <td> </td> <td>6 m²</td> <td> </td> <td><input checked="" type="checkbox"/> Còn Trống</td> <td> </td> </tr> </tbody> </table>				Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn	101	đ4,000,000		6 m ²		<input checked="" type="checkbox"/> Còn Trống		102	đ3,500,000		6 m ²		<input type="radio"/> Không dùng được		103	đ3,000,000		6 m ²		<input checked="" type="checkbox"/> Còn Trống	
Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn																									
101	đ4,000,000		6 m ²		<input checked="" type="checkbox"/> Còn Trống																										
102	đ3,500,000		6 m ²		<input type="radio"/> Không dùng được																										
103	đ3,000,000		6 m ²		<input checked="" type="checkbox"/> Còn Trống																										

FPT Education | Tin tức | Tuyển sinh

Step 3: Redirect to Add Room page, click on “Thêm nhiều phòng”

TÌM TRỌ SINH VIÊN FPT

FU House Finder Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu Dũng Nhung ▾

Thêm 1 phòng **Thêm nhiều phòng**

Tòa nhà	Loại phòng
Tầng	<input type="checkbox"/> Tủ lạnh <input type="checkbox"/> Bếp <input type="checkbox"/> Máy giặt
Tên phòng	<input type="checkbox"/> Bàn <input type="checkbox"/> Giường <input type="checkbox"/> Không chung
Giá	<input type="checkbox"/> Vệ sinh khép kín
Thông tin	Ảnh 1 Choose File No file chosen
Diện tích	Ảnh 2 Choose File No file chosen
Số người tối đa	Ảnh 3 Choose File No file chosen

Step 4: Download excel template, input information of new room on this file, upload to this page and click on “Thêm mới” button

TÌM TRỌ SINH VIÊN FPT

FU House Finder Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu Dũng Nhung ▾

Thêm 1 phòng **Thêm nhiều phòng**

Download Template

Upload Template

Upload Image

3.2.5.10 Update status of room

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

Continue with Facebook

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on change status icon to change status of room

TÌM TRỌ SINH VIÊN FPT

Dũng Nhung ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

2 Phòng trống	0 Phòng trống một phần	4 Chỗ trống	đ10,500,000 Tiền chưa cho thuê
------------------	---------------------------	----------------	-----------------------------------

Danh sách phòng

+ Thêm mới

Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn	
101	đ4,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống		
102	đ3,500,000		6 m ²	2	<input type="radio"/> Không dùng được		
103	đ3,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống		

FPT Education | Tin tức | Tuyển sinh

The screenshot shows the homepage of the FU House Finder website. At the top, there's a navigation bar with the title "TÌM TRỌ SINH VIÊN FPT" and a search icon. Below the title, there are links for "Trang chủ", "Tuyển sinh", "Liên hệ", and "Giới thiệu". A dropdown menu for "Dũng Nhung" is visible. The main content area displays four categories: "Phòng trống" (2), "Phòng trống một phần" (0), "Chỗ trống" (4), and "Tiền chưa cho thuê" (đ10,500,000). Below these, a section titled "Danh sách phòng" lists three room options:

Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn
101	đ4,000,000		6 m ²	2	<input checked="" type="radio"/> Đã thuê	
102	đ3,500,000		6 m ²	2	<input type="radio"/> Không dùng được	
103	đ3,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống	

At the bottom of the page, there are links for "FPT Education", "Tin tức", and "Tuyển sinh".

3.2.5.11 Update room

Role: (Landlord)

Step 1: Sign in with role is Landlord

The image shows the FU House Finder logo, which features a magnifying glass focusing on a small house with a frog inside. Below the logo, the text "FU HOUSE FINDER" is written in orange, followed by the subtitle "Ứng dụng tìm trọ dành cho sinh viên Đại học FPT" in a smaller font.

The login page has a header "ĐĂNG NHẬP" and a "Continue with Facebook" button. It also features a "Đăng nhập với tên Kien trikienthachban@gmail.com" field with a Google "G" icon. Below these, there are fields for "Email *" containing "thanhle@gmail.com" and "Mật khẩu *" containing "*****". A "Đăng nhập" button is at the bottom right.

Step 2: Click on update icon

The screenshot shows the homepage of the FU House Finder website. At the top, there's a navigation bar with the title "TÌM TRỌ SINH VIÊN FPT". Below it are links for "Trang chủ", "Tuyển sinh", "Liên hệ", and "Giới thiệu". A dropdown menu for "Dũng Nhung" is also present. Below the navigation, there are four summary boxes: "Phòng trống" (2), "Phòng trống một phần" (0), "Chỗ trống" (4), and "Tiền chưa cho thuê" (đ10,500,000).

Danh sách phòng

+Thêm mới

Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn
101	đ4,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống	
102	đ3,500,000		6 m ²	1	<input type="checkbox"/> Không dùng được	
103	đ3,000,000		6 m ²	2	<input checked="" type="checkbox"/> Còn Trống	

Step 3: Input information to update on modal and click on “Lưu” button

The screenshot shows a modal window titled "Chỉnh sửa phòng" (Edit Room). It contains fields for room details: "Tên Phòng" (Room 101), "Diện tích" (6), "Giá tiền" (400000), "Tòa" (1), "Thông tin" (Vệ sinh khép kín), "Tầng" (1), "Trạng thái" (Available), and "Số người ở tối đa" (2). Below these are sections for "Số người ở" (0) and "Tiện ích" (checkboxes for Tủ lạnh, Máy giặt, Chung cư, Vệ sinh khép kín, Bếp, Bàn học, chุด, Giường). At the bottom are "Đóng" (Close) and "Lưu" (Save) buttons.

3.2.5.12 Delete room

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

Continue with Facebook

Đăng nhập với tên Kien
trikienthachban@gmail.com

G

hoặc sử dụng

Email *

thanhle@gmail.com

Mật khẩu *

.....

Đăng nhập

Step 2: Click on delete icon

TÌM TRỌ SINH VIÊN FPT

Dũng Nhung ▾

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

2	0	4	đ10,500,000			
Phòng trống	Phòng trống một phần	Chỗ trống	Tiền chưa cho thuê			
+ Thêm mới						
Tên phòng	Giá phòng	Tiện ích	Diện tích	Số người ở	Trạng thái	Tùy chọn
101	đ4,000,000	Wi-Fi, Điện thoại, Tivi, Bếp, Phòng tắm, WC riêng	6 m ²	2	Còn Trống	
102	đ3,500,000	Wi-Fi, Điện thoại, Tivi, Bếp, Phòng tắm, WC chung	6 m ²	1	Không dùng được	
103	đ3,000,000	Wi-Fi, Điện thoại, Tivi, Bếp, Phòng tắm, WC riêng	6 m ²	2	Còn Trống	

FPT Education

Tin tức

Tuyển sinh

3.2.5.13 Update profile

Role: (Landlord)

Step 1: Sign in with role is Landlord



FU HOUSE FINDER

Ứng dụng tìm trọ dành cho sinh viên Đại học FPT

ĐĂNG NHẬP

[Continue with Facebook](#)

hoặc sử dụng

Email *

Mật khẩu *

[Đăng nhập](#)

Step 2: Click on “Chỉnh sửa thông tin” on header



TÌM TRỌ SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Danh sách nhà trọ

STT	Tên nhà trọ	Tiền chưa cho thuê	Người lớn	Người nhỏ	Phòng	Tiện ích	Tùy chọn
1	Trọ Tâm Thảo	đ10,500,000	2	0	4	<input type="checkbox"/>	
2	Trọ Thu Thảo	đ9,000,000	0	3	3	<input type="checkbox"/>	
3	Trọ Tâm Lê	đ8,250,000	0	3	3	<input type="checkbox"/>	
4	Trọ Campus 2	đ3,000,000	0	1	1	<input checked="" type="checkbox"/>	

Step 3: Redirect to Update Profile page, input name and email want to update and click on “Thay đổi” button



FU House Finder

TÌM TRỌNG SINH VIÊN FPT

Trang chủ | Tuyển sinh | Liên hệ | Giới thiệu

Dũng Nhung ▾

Cập nhật thông tin chủ trọ

Họ và tên

Nhung Dũng

Email

nhungdung@gmail.com

Thay đổi



FPT Education
Hoa Lạc High Tech Park, Hanoi, Vietnam

Tin tức
Học tập khác biệt

Tuyển sinh
Đại học FPT