

وزارة التعليم العالي
جامعة الإمام عبد الرحمن بن فيصل
كلية المجتمع
قسم الحاسوب الآلي

MINISTRY OF EDUCATION
IMAM ABDULRAHMAN BIN FAISAL UNIVERISTY
COMMUNITY COLLEGE
COMPUTER DEPARTMENT



جامعة الإمام عبد الرحمن بن فيصل
IMAM ABDULRAHMAN BIN FAISAL UNIVERSITY



Dormitorio

A Graduation Project Report Submitted to Community College, Imam Abdulrahman Bin Faisal University in Partial Fulfillment of the Requirements for the Degree of Associate Diploma of computer department, track information system.

By:
Group (No.17)

Fay Yusif Alkazaim	2190005733
Ahad Faisal Althaqafi	2190004537
Amal Muhammad Alzahrani	2160007870
Rawan Jamal Alsadoon	2190001350
Sukainah Abdullah Al-Nasser	2190000389
Lama Mohammed Alshekhi	2190005768
Nouf Mohsen Alshamrani	2190004720

Supervised By:
Dr. Fatemah Alghamedy

Department of Computer (IS)
Community College
Imam Abdulrahman Bin Faisal University
KSA
1441–1442H

Declaration

This is to declare that the graduation project entitled "Dormitorio" has been prepared on the basis of my own work, in a partial requirement of the fulfillment for the degree" Associate Diploma of computer", at information system Department, Community college, Imam Abdulrahman Bin Faisal University .All the analysis, design and system development have been accomplished by the undersigned. Moreover, this project has not been submitted to any other college or university.

Graduation project's student(s):

- | | | |
|-------------------------------------|------------|------------------|
| 1. Group leader: Fay Yusif Alkazaim | 2190005733 | Signature: |
| 2. Ahad Faisal Althaqafi | 2190004537 | Signature: |
| 3. Amal Muhammad Alzahrani | 2160007870 | Signature: |
| 4. Rawan Jamal Alsadoon | 2190001350 | Signature: |
| 5. Sukainah Abdullah Al-Nasser | 2190000389 | Signature: |
| 6. Lama Mohammed Alshekhi | 2190005768 | Signature: |
| 7. Nouf Mohsen Alshmrani | 2190004720 | Signature: |

Abstract

After students are accepted in the university, students who will need to move or travels out of their hometown begin to think about an important problem which is where she can stay during the period of study. The question that arises here is that which is the best dorm choice: internal or external? which is confusing for many students. In addition, the cost of accommodation is one of the issues that students face. In order to facilitate students housing, we created a website to be a link between the owner of dorms and students to provide suitable housing for their needs with clear prices and features. Furthermore, the website provide a possibility of booking in a simple and clear manner.

Keywords: Dorm; Renter; Booking; University Students; Website

TABLE OF CONTENTS

List of Tables	viii
List of Figures	ix
1 Introduction	2
1.1 Introduction	3
1.2 Project Motivation and Problem Statement	3
1.2.1 Project Motivation	3
1.2.2 Problem Statement	3
1.3 Project Scope & Limitations	3
1.4 Project Software and Hardware Requirements	4
2 Background & Related Work	5
2.1 Related Work Survey	6
2.1.1 student.com	6
2.1.2 Uniplaces.com	6
2.1.3 Survey, Results and Analysis	7
2.1.3.1 University Dorms Survey Results	7
2.1.3.2 Apartments Survey Results	9
2.2 Proposed & Similar Systems Comparison	15
3 System Planning and Requirement	16
3.1 Functional Requirement	17
3.1.1 Functional Requirement (Renter)	17
3.1.2 Functional Requirement (Owner)	19
3.1.3 Functional Requirement (Administrator)	20
3.1.4 Functional Requirements (Guest)	20
3.2 Non-Functional Requirement	22
3.2.1 Appearance Requirement	22
3.2.2 Usability and Humanity Requirement	22
3.2.2.1 Ease of use requirements	22
3.2.2.2 Understand-ability and Politeness Requirements . . .	22
3.2.3 Performance	22

4 System Analysis & Design	23
4.1 DFD Approach	24
4.1.1 Context Diagram	24
4.1.2 Overview Diagram (Level 0)	25
4.1.3 Detailed DFDs	26
4.1.4 Data Dictionary of DFD	33
4.2 DATA Design	34
4.2.1 Entity Relationship Diagram (ERD)	34
4.2.1.1 Description of Entities	34
4.2.1.2 Description of Relations	34
4.2.1.3 Drawing ERD	36
4.2.1.4 Data Dictionary of ERD	37
4.2.2 Relation Database Schema	38
4.2.2.1 Attributes	39
4.2.2.2 Relations	43
4.3 Graphical User Interface (GUI) Design	44
5 Conclusion and Future Work	49
5.1 Lessons Learned	50
5.2 Future Work	50
6 System Implementation and Evaluation	51
6.1 Database Implementation	53
6.1.1 Admin	54
6.1.2 Booking	55
6.1.3 Dorm	56
6.1.4 Dorm images	58
6.1.5 Owner	58
6.1.6 Renter	59
6.2 Graphical User Interface Implementation	60
6.2.1 Gust User	63
6.2.1.1 Home Page	63
6.2.1.2 Register	64
6.2.1.3 Contact	65
6.2.1.4 FQA	66
6.2.2 Owner	67
6.2.2.1 Home Page	67
6.2.2.2 Register	68
6.2.2.3 Contact	69
6.2.2.4 FQA	70
6.2.2.5 Insert dorm	71
6.2.2.6 Your dorm	73
6.2.2.7 My dorms booking	74
6.2.2.8 Your profile	76
6.2.3 Renter	77

6.2.3.1	Home Page	77
6.2.3.2	Register	78
6.2.3.3	Contact	79
6.2.3.4	FQA	80
6.2.3.5	Dorm List	81
6.2.3.6	Dorm Details	82
6.2.3.7	Booking	83
6.2.4	Admin	85
6.2.4.1	Home Page	85
6.2.4.2	Customer Booking Page	86
6.2.4.3	Dorms Registration Requests	87
7	Conclusion and Future Work	89
7.1	Overall Weaknesses	89
7.2	Overall Strengths	90
7.3	Future Work	90
7.4	Useful Resources	90
Bibliography		92

LIST OF TABLES

1.1	Software and hardware with a short description.	4
2.1	Proposed and Similar Systems Comparison	15
4.1	Admin Table	39
4.2	Owner Table	39
4.3	Zone Table	39
4.4	Renter Table	40
4.5	Dorm Table	41
4.6	Contact Table	42
4.7	Dorms cities Table	42
4.8	Dorm-pictures	42
4.9	Dorm-reviews	42

LIST OF FIGURES

2.1	student.com website logo.	6
2.2	Uniplaces.com website logo.	6
2.3	University dorms survey results Q1	7
2.4	University dorms survey results Q2	8
2.5	University dorms survey results Q3	8
2.6	University dorms survey results Q4	8
2.7	University dorms survey results Q5	9
2.8	Apartments survey results Q1	9
2.9	Apartments survey results Q2	10
2.10	Apartments survey results Q3	10
2.11	Apartments survey results Q4	11
2.12	Apartments survey results Q5	11
2.13	Apartments survey results Q6	12
2.14	Apartments survey results Q7	12
2.15	Apartments survey results Q8	12
2.16	Apartments survey results Q9	13
2.17	Others' ideas for our project - part 1	13
2.18	Others' ideas for our project - part 2	14
2.19	Others' ideas for our project - part 3	14
4.1	Context Diagram	24
4.2	Overview Diagram (Level 0)	25
4.3	Level one (Provide home page)	26
4.4	Level one (Rent)	27
4.5	Level one (Provide account page)	28
4.6	Level one (Chat)	29
4.7	Level two (Choose city or collage)	30
4.8	Level two (Provide renter account)	31
4.9	Level two (Provide owner account)	31
4.10	Level two (Admin account)	32
4.11	Data Dictionary of DFD	33
4.12	ER Diagram	36
4.13	Data dictionary of ERD	37
4.14	Relation database schema	38
4.15	Relationships in ERD	43
4.16	Home Page	44
4.17	About us	45

4.18 Properties	46
4.19 About the dorm	47
4.20 Payment Page	48
6.1 XAMPP	53
6.2 Database Tables	54
6.3 Admin Table	55
6.4 Booking Table	55
6.5 Dorm Table 1	57
6.6 Dorm Table 2	57
6.7 Dorm Table 3	57
6.8 Dorm images Table	58
6.9 Owner Table	59
6.10 Renter Table	59
6.11 Home Page	63
6.12 Home Page PHP Code	63
6.13 Register	64
6.14 Register as owner PHP code	64
6.15 Register as renter PHP code	65
6.16 Contact	65
6.17 Contact PHP code	66
6.18 FQA	66
6.19 FQA code	67
6.20 Home Page-Owner	67
6.21 Owner home page PHP code	68
6.22 Register-Owner	68
6.23 Register as owner PHP code	69
6.24 Contact-Owner	69
6.25 Contact PHP code	70
6.26 FQA-Owner	70
6.27 FQA code	71
6.28 Insert Dorm Part1	71
6.29 Insert Dorm Part2	72
6.30 Insert dorm PHP code 1	72
6.31 Insert dorm PHP code 2	72
6.32 Insert dorm PHP code 3	73
6.33 Insert dorm PHP code 4	73
6.34 Your Dorm	74
6.35 Your dorm PHP code	74
6.36 My Booking Dorms	75
6.37 My dorm booking PHP code 1	75
6.38 My dorm booking PHP code 2	75
6.39 Your Profile-Owner	76
6.40 Your profile PHP code 1	76
6.41 Your profile PHP code 2	77

6.42 Your profile PHP code 3	77
6.43 Home Page-Renter	78
6.44 Renter home page PHP code	78
6.45 Register-Renter	79
6.46 Register as renter PHP code	79
6.47 Contact-Renter	80
6.48 Contact PHP code	80
6.49 FQA-Renter	81
6.50 FQA code	81
6.51 Dorm List	82
6.52 Dorm list PHP code	82
6.53 Dorm Details	83
6.54 Dorm details PHP code	83
6.55 Booking	84
6.56 Booking PHP code 1	84
6.57 Booking PHP code 2	84
6.58 Home Page-Admin	85
6.59 Admin homepage	85
6.60 Customer Booking Page	86
6.61 Customer booking code 1	86
6.62 Customer booking PHP code 2	87
6.63 Dorms Registration Requests	87
6.64 Dorms Registration Requests PHP code 1	88
6.65 Dorms Registration Requests PHP code 2	88

Chapter 1

Introduction

1.1	Introduction	3
1.2	Project Motivation and Problem Statement	3
1.2.1	Project Motivation	3
1.2.2	Problem Statement	3
1.3	Project Scope & Limitations	3
1.4	Project Software and Hardware Requirements	4

1.1 Introduction

The students encounter many problems during the search of appropriate dorms such as the distance between the campus and the dorm and the high demand costs. We will build a website that serves university female students in searching for suitable dorms whether inside or outside the university campus. The website will provide students with dorms suitable for their needs in terms of size, price, and services that available around the dorms such as restaurants, supermarkets, laundries, hospitals, and parking. In addition, the features of dorms will be easily explained to students on the website, whether it is vacant or not, and pictures of it. The students are allowed to book from the site besides communicating with the owner of the residence with an administrators' control of the chat to prevent any means of communication outside the site.

1.2 Project Motivation and Problem Statement

1.2.1 Project Motivation

We have two motivations to create this project. The first one is that many students suffer when they are searching for dorms if the university is far from their home. The second one is that students need to collect as much information about available services nearby to dorms.

1.2.2 Problem Statement

When students search for a dorm, they care about many aspects like the space of the room(s), location of the dorm, and the services nearby to this dorm. Students encounter problems during the search for appropriate dorms! Keep in mind that students will come without knowledge about the city that she will move to which makes this process take time and a lot of effort. For that, this website is important to make this process much easier for students.

1.3 Project Scope & Limitations

Any student who speaks either Arabic or English and wants to study in the eastern province and her home is far. In addition, any investor who wants his dorms to be

rented. Our system will be limit to Saudi Arabia only, and it supports only two languages and specific payment methods.

1.4 Project Software and Hardware Requirements

The system will use HTML and CSS to create user interfaces. The system will use JavaScript to make the website interact with users. The system will use PHP and SQL to deal with the server and databases. Furthermore, the system needs a network connection to create the system and use it, and the system needs to programming software such as netbeans to code it.

Software and Hardware	Short Description
HTML and CSS	HTML (Hypertext Markup Language) and CSS (Cascading Style Sheets) are two of the core technologies for building Web pages. HTML provides the structure of the page, and CSS provides the (visual and aural) layout, for a variety of devices.
JavaScript	JavaScript is a programming language for the web that can update and change both HTML and CSS. JavaScript can calculate, manipulate, and validate data.
PHP	PHP is a widely-used, open-source scripting language. PHP scripts are executed on the server.
jQuery	It is a popular library for JavaScript whose main purpose is to simplify the methods of dealing with the JavaScript language with the addition of ready-made features such as visual effects and are used in building web pages.
SQL	Database design program.
Adobe Dreamweaver	Program specialized in creating and managing applications and websites.
Adobe XD	Program for designing user interfaces (UI).
LaTeX	Alternative to the word. It's for documents format.
API Google maps	Tool for providing the maps into the web page.
Network Connection	Any network can be used.
Computer	Any good computer.

Table 1.1: Software and hardware with a short description.

Chapter 2

Background & Related Work

2.1	Related Work Survey	6
2.1.1	student.com	6
2.1.2	Uniplaces.com	6
2.1.3	Survey, Results and Analysis	7
2.1.3.1	University Dorms Survey Results	7
2.1.3.2	Apartments Survey Results	9
2.2	Proposed & Similar Systems Comparison	15

2.1 Related Work Survey

2.1.1 student.com

It is a website that provides dorms for students in an accessible way by providing housing by simple and clearly way. It covers more than 400 cities around the world.



Figure 2.1: student.com website logo.

2.1.2 Uniplaces.com

Uniplaces It is a site specialized in providing dorms students owns thousands of housing around world.



Figure 2.2: Uniplaces.com website logo.

2.1.3 Survey, Results and Analysis

This survey was written in order to help us in the project of providing students dorms electronically by knowing people's opinions about the idea and what are their basic needs and appropriate prices for them, as well as their suggestions for developing our ideas in creating the site. And based on they prefer university dorms or private dorms, two questionnaire forms were created to match both desires and we were able to obtain 143 responses from various points of view and opinions that we will present below:

Survey link: [click to view our survey](#)

2.1.3.1 University Dorms Survey Results

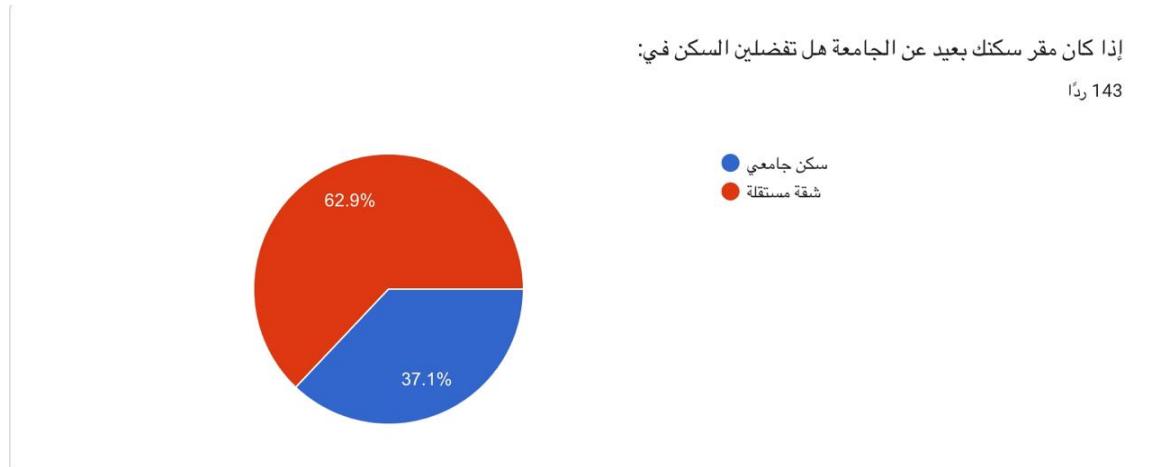


Figure 2.3: University dorms survey results Q1

هل جربتى السكن الجامعى ؟

ردًّا 53

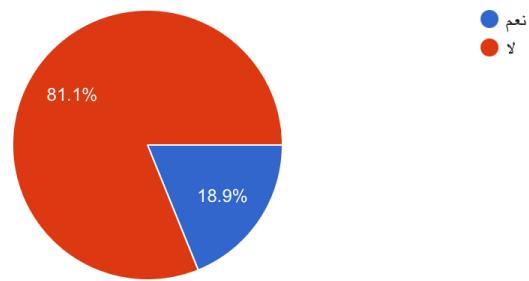


Figure 2.4: University dorms survey results Q2

هل واجهت مشاكل سابقاً عند البحث عن سكن أثناء دراستك الجامعية ؟

ردًّا 90

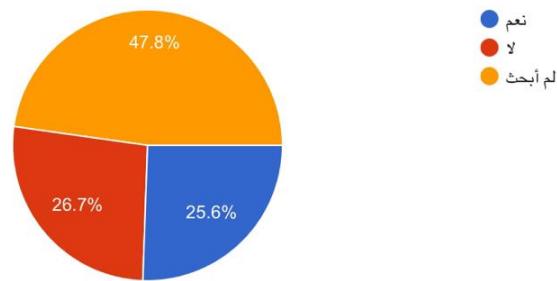


Figure 2.5: University dorms survey results Q3

هل لديك معلومات كافية عن سكن المطالبات في جامعتك ؟

ردًّا 53

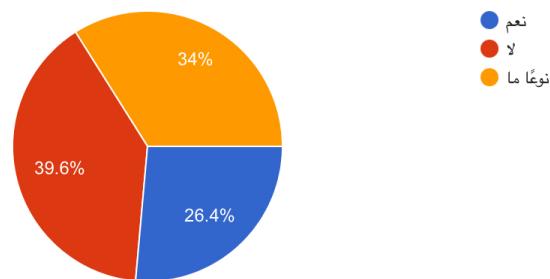


Figure 2.6: University dorms survey results Q4

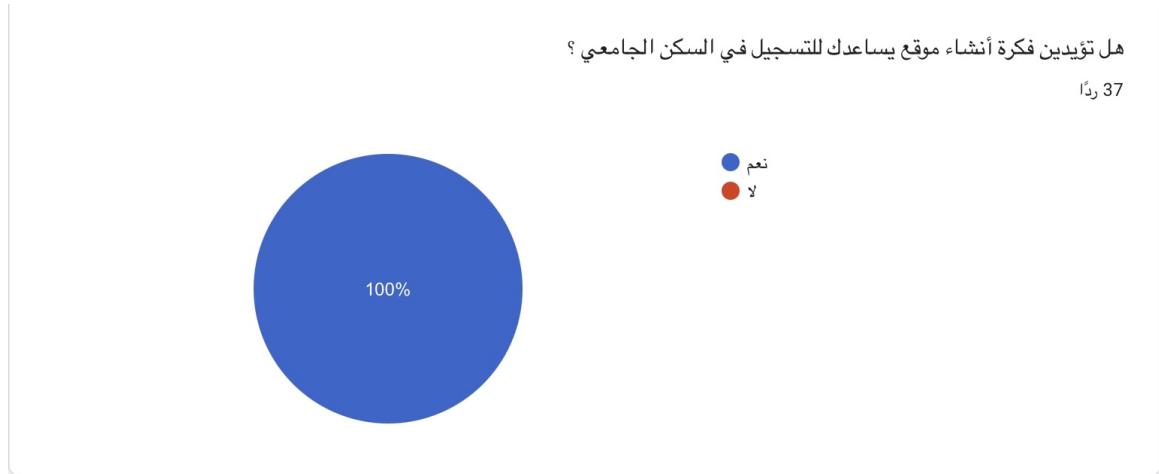


Figure 2.7: University dorms survey results Q5

2.1.3.2 Apartments Survey Results

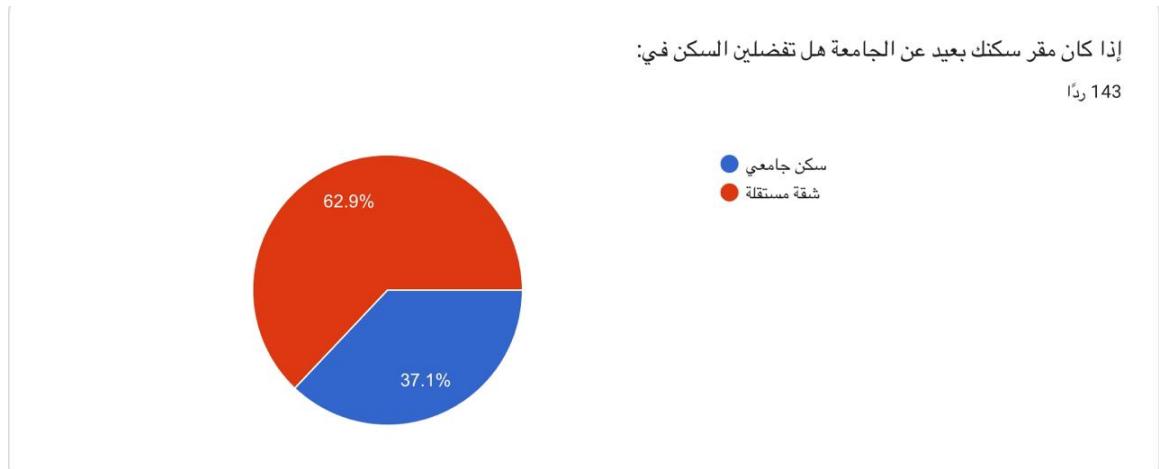


Figure 2.8: Apartments survey results Q1

هل جربت السكن في شقة مستقلة لدواعي الدراسة ؟

90 ردًّا

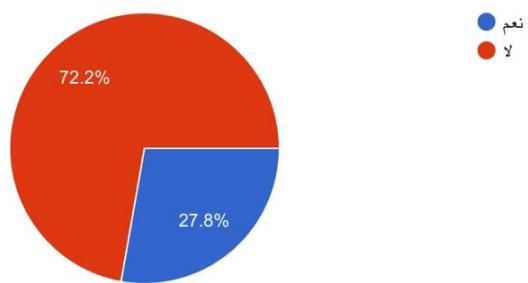


Figure 2.9: Apartments survey results Q2

هل واجهتك مشاكل سابقاً عند البحث عن سكن أثناء دراستك الجامعية ؟

90 ردًّا

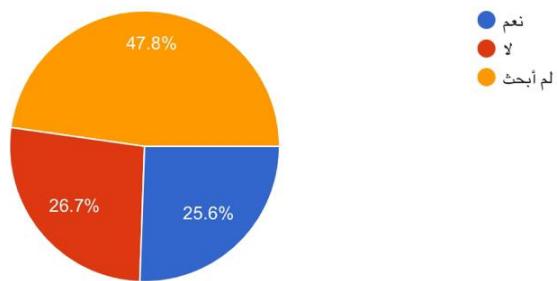


Figure 2.10: Apartments survey results Q3

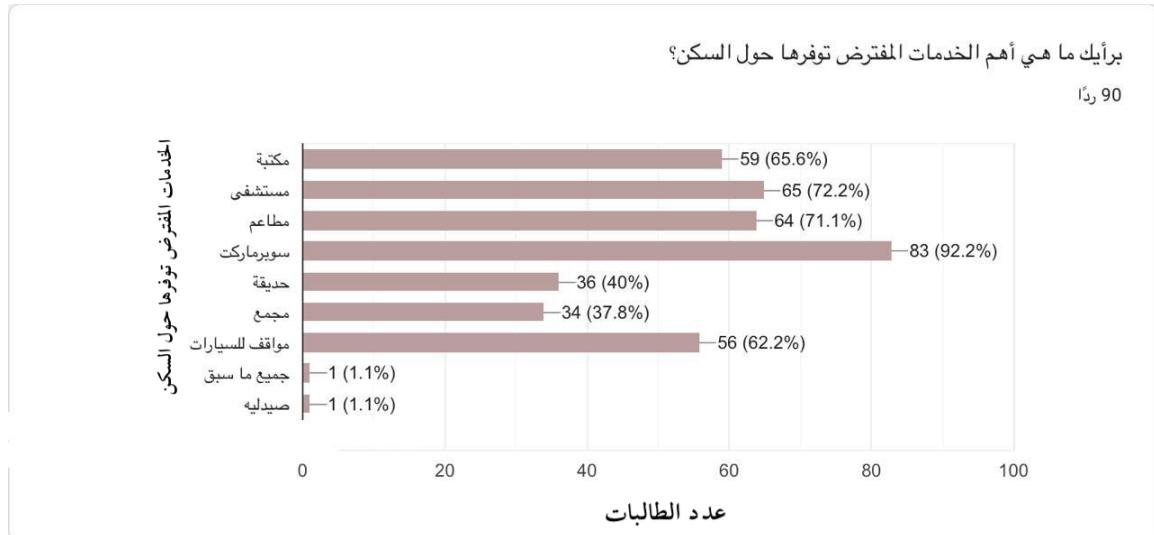


Figure 2.11: Apartments survey results Q4

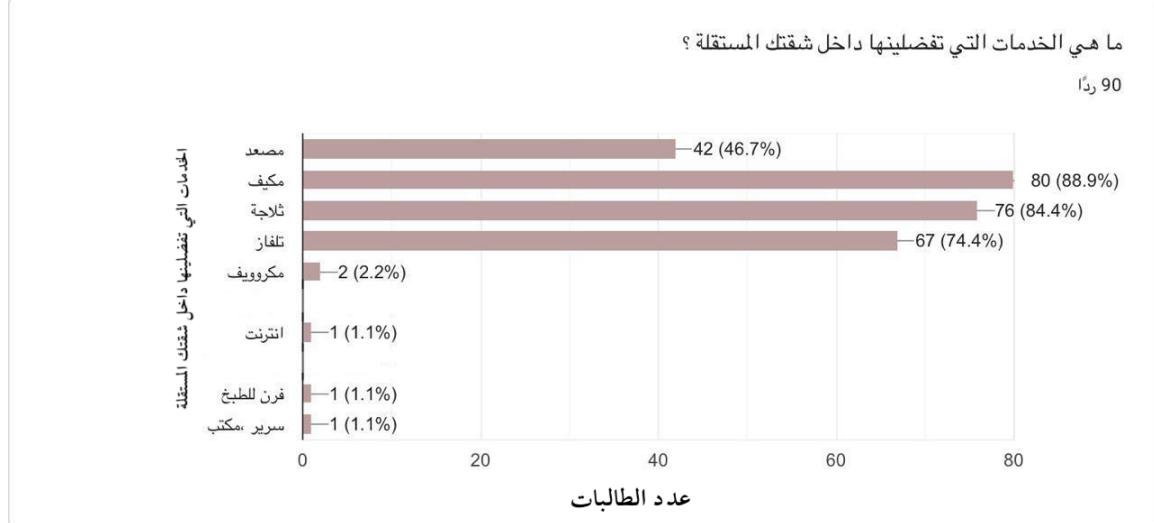


Figure 2.12: Apartments survey results Q5

هل تفضلين البقاء في غرفة :

ردًّا 90

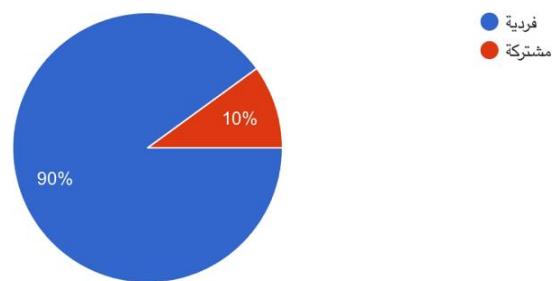


Figure 2.13: Apartments survey results Q6

كم عدد المشاركات المناسب معك ؟

ردًّا 9

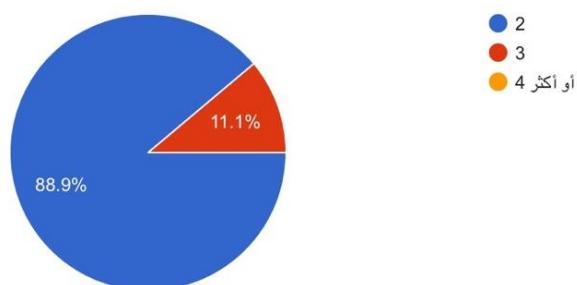


Figure 2.14: Apartments survey results Q7

برأيك ما هو السعر المناسب لك كطابة عند استئجار شقة مستقلة شهريًا ؟

ردًّا 81

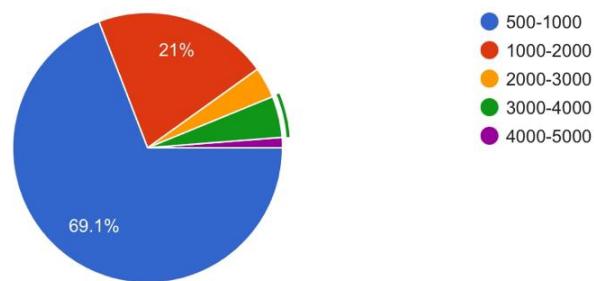


Figure 2.15: Apartments survey results Q8

هل تؤيدن فكرة إنشاء موقع يساعدك في البحث عن شقة مستقلة في حال عدم رغبتك في الانتقال للسكن الجامعي؟

15, 81

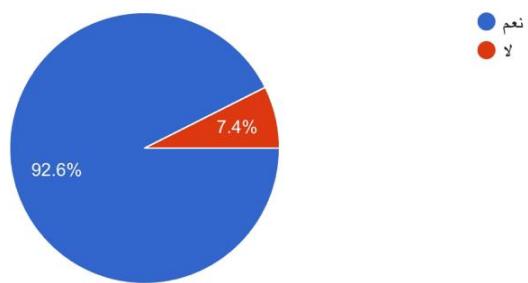


Figure 2.16: Apartments survey results Q9

ما هي الخدمات التي ترغبين بتواجدها في موقعنا؟

136

وصف السكن و سهوله الدفع

مجمع كامل متكمال بالاحتياجات الدراسية، الترفيه، الخ

صور للشقة + موقعها + سعرها

جمع الغف الشاغره في السكن

ان تكون الشقة مفروشة

فيديو 360 درجة لسكن المعرض بحيث تكون الصورة واضحة قدر المستطاع للطالبة - خيار لتحديد الاسعار المناسبة للطالبة بناء عليه يتم عرض السكن المناسب لإمكانياتها - خيار ما اذا كانت تحتاج معها رفيق في الغرفة او لا - ظهور خريطة الموقع عند اختيار سكن - توضيح ماهي المارافق او الخدمات القريبة من السكن - شرح مفصل عن السكن وما يحويه منه اختباره - ارقام واسباب للاتسقانة، والتواصل .

خدمة العملاء

تزويني بملوحة بالماكن القوية من الشقة

Figure 2.17: Others' ideas for our project - part 1



Figure 2.18: Others' ideas for our project - part 2



Figure 2.19: Others' ideas for our project - part 3

2.2 Proposed & Similar Systems Comparison

Name	Dormitorio	Student.com	Uniplaces
Type	Website	Website	Website and application
Scope	Kingdom of Saudi Arabia	It covers more than 100 countries.	It covers Europe only
Serves to	For female students only.	All students.	All students.
Services	-This site will deal with homeowners in different regions around the Kingdom of Saudi Arabia, and the housing information and its validity will also be verified to be displayed on the site to raise the quality standards on our site	-The site provides a reliable and safe online reservation service for students to search for suitable dorms.	-This site will be unique and will cater to many students looking for dorms

Table 2.1: Proposed and Similar Systems Comparison

Chapter 3

System Planning and Requirement

3.1	Functional Requirement	17
3.1.1	Functional Requirement (Renter)	17
3.1.2	Functional Requirement (Owner)	19
3.1.3	Functional Requirement (Administrator)	20
3.1.4	Functional Requirements (Guest)	20
3.2	Non-Functional Requirement	22
3.2.1	Appearance Requirement	22
3.2.2	Usability and Humanity Requirement	22
3.2.2.1	Ease of use requirements	22
3.2.2.2	Understand-ability and Politeness Requirements . . .	22
3.2.3	Performance	22

3.1 Functional Requirement

3.1.1 Functional Requirement (Renter)

- The system will allow the renter to register.
- The system will allow the renter to log In.
- The system will allow the renter to reset the password.
- The system will allow the renter to modify her information (phone number, password, email, and Name).
- The system will allow the renter to change the website's language.
- The system will allow the renter to search by city name or collage name.
- The system will allow the renter to book only one room.
- The system will allow the renter to view the dorms.
- The system will allow the renter to filter the search results
- The system will allow the renter to filter dorms based on price ranges.
- The system will allow the renter to filter dorms based on type (either single or shared).
- The system will allow the renter to filter dorms based on available rooms in the dorm.
- The system will allow the renter to filter dorms based on the distance between the dorm and college.
- The system will allow the renter to filter dorms based on the parking type (indoor, outdoor).
- The system will allow the renter to select a dorm.
- The system will allow the renter to view the features and details for a selected dorm.
- The system will allow the renter to view if the A/C is available or not in the dorm.

- The system will allow the renter to view if the kitchen is available or not in the dorm.
- The system will allow the renter to view if there are an oven, a fridge, and a microwave in the dorm.
- The system will allow the renter to view the buildings features (fire alarm, guard, parking, lock, elevator, monitoring camera, and mailbox).
- The system will allow the renter to view the price of the (single dorms) and (room in shared dorms).
- The system will allow the renter to view the distance between the dorm's building and the college.
- The system will allow the renter to view the dorm's pictures.
- The system will allow the renter to view the dorm's location.
- The system will allow the renter to view the required payment methods for the dorm.
- The system will allow the renter to view other renters reviews about the dorm.
- The system will allow the renter to contact with the dorm's owner (under admin control).
- The system will allow the renter to book the dorm.
- The system will allow the renter to pay electronically.
- The system will allow the renter to view the booking details (date of book, the price and owner contact information).
- The system will allow the renter to cancel her booking.
- The system will allow the renter to write a review for the dorm(s) that she rented.
- The system will allow the renter to contact with the website's IT by live chat or email provided by the site.
- The system will allow the renter to view the website's FAQ.

- The system will allow the renter to view a brief description of the website (about us).

3.1.2 Functional Requirement (Owner)

- The system will allow the owner to register.
- The system will allow the owner to log in.
- The system will allow the owner to reset password.
- The system will allow the owner to modify his information (email , phone number , password, and name).
- The system will allow the owner to send request to the admin to insert at least one dorm include (dorm location , features , and price).
- The system will allow the owner to check order statue of dorms insertion.
- The system will allow the owner to pay fees electronically.
- The system will allow the owner to insert dorms after admin approve.
- The system will allow the owner to insert the dorm's features (A/C, kitchen, oven, and microwave).
- The system will allow the owner to insert the buildings features (fire alarm, guard, parking, lock, elevator, monitoring camera, and mailbox).
- The system will allow the owner to insert the dorm's pictures.
- The system will allow the owner to insert the dorm's location.
- The system will allow the owner to update the dorm's details (price and features)
- The system will allow the owner to remove his own dorm(s) from the website.
- The system will allow the owner to contact with the renter (under system control).
- The system will allow the owner to receive the renter booking.
- The system will allow the owner to cancel the renter booking .

- The system will allow the owner to contact with website's IT via live chat or email form.
- The system will allow the owner to receive the renter's payment.

3.1.3 Functional Requirement (Administrator)

- The system will allow the admin to log in.
- The system will allow the admin to checks the renters information (ID ,email , and Phone number).
- The system will allow the admin to check the validate the owner ID and the renter ID through the Absher platform (future feature).
- The system will allow the admin to send to the users (owner , renters) a code to check the validate of her/his email and phone number.
- The system will allow the admin to checks the owners information (ID, email, and Phone number).
- The system will allow the admin to accept owners orders to insert dorms
- The system will allow the admin to reject owners orders to insert dorms.
- The system will allow the admin to receive the service fees from the owner.
- The system will allow the admin to checks dorms information.
- The system will allow the admin to contact with renters and owners (if needed).
- The system will allow the admin to add cities and colleges.

3.1.4 Functional Requirements (Guest)

- The system will allow the guest to sign up.
- The system will allow the guest to search by the city name.
- The system will allow the guest to search by the college name.
- The system will allow the guest to filter by features.
- The system will allow the guest to filter dorms based on price ranges.

- The system will allow the guest to filter dorms based on number of rooms in the available dorm.
- The system will allow the guest to filter dorms based on the dorm type (single or shared).
- The system will allow the guest to filter dorms based on the distance between the dorm and college.
- The system will allow the guest to filter dorms based on the parking type (indoor,outdoor).
- The system will allow the guest to select a dorm.
- The system will allow the guest to view the features and details for a selected dorm.
- The system will allow the guest to view if the A/C is available or not in the dorm.
- The system will allow the guest to view if the kitchen is available or not in the dorm.
- The system will allow the guest to view if there are an oven, a fridge, and a microwave in the dorm.
- The system will allow the guest to view the buildings features (fire alarm, guard, parking,lock, elevator, monitoring camera, and mailbox).
- The system will allow the guest to view the price of the dorm.
- The system will allow the guest to view the distance between the dorm's building and the college.
- The system will allow the guest to view the dorm's pictures.
- The system will allow the guest to view the dorm's location.
- The system will allow the guest to view the required payment methods for the dorm.
- The system will allow the guest to view other renters reviews about the dorm.

3.2 Non-Functional Requirement

3.2.1 Appearance Requirement

- The used languages in the website are Arabic and English.
- The system will provide texts, pictures, and maps to help renters and guests to view all needed information.

3.2.2 Usability and Humanity Requirement

3.2.2.1 Ease of use requirements

- The system shall be easy for users who don't have much knowledge of using websites.
- The system shall have the fewest possible errors. (2%).

3.2.2.2 Understand-ability and Politeness Requirements

- The system should be available at all times. A backup copy will be used for the maintenance situations.
- The system shall use simple words.

3.2.3 Performance

- The system will be fast (2 seconds) to do any operation such as download.
- The system will be secure (2-steps Login).
- The website will be small in size (around 490KB).

Chapter 4

System Analysis & Design

4.1	DFD Approach	24
4.1.1	Context Diagram	24
4.1.2	Overview Diagram (Level 0)	25
4.1.3	Detailed DFDs	26
4.1.4	Data Dictionary of DFD	33
4.2	DATA Design	34
4.2.1	Entity Relationship Diagram (ERD)	34
4.2.1.1	Description of Entities	34
4.2.1.2	Description of Relations	34
4.2.1.3	Drawing ERD	36
4.2.1.4	Data Dictionary of ERD	37
4.2.2	Relation Database Schema	38
4.2.2.1	Attributes	39
4.2.2.2	Relations	43
4.3	Graphical User Interface (GUI) Design	44

4.1 DFD Approach

4.1.1 Context Diagram

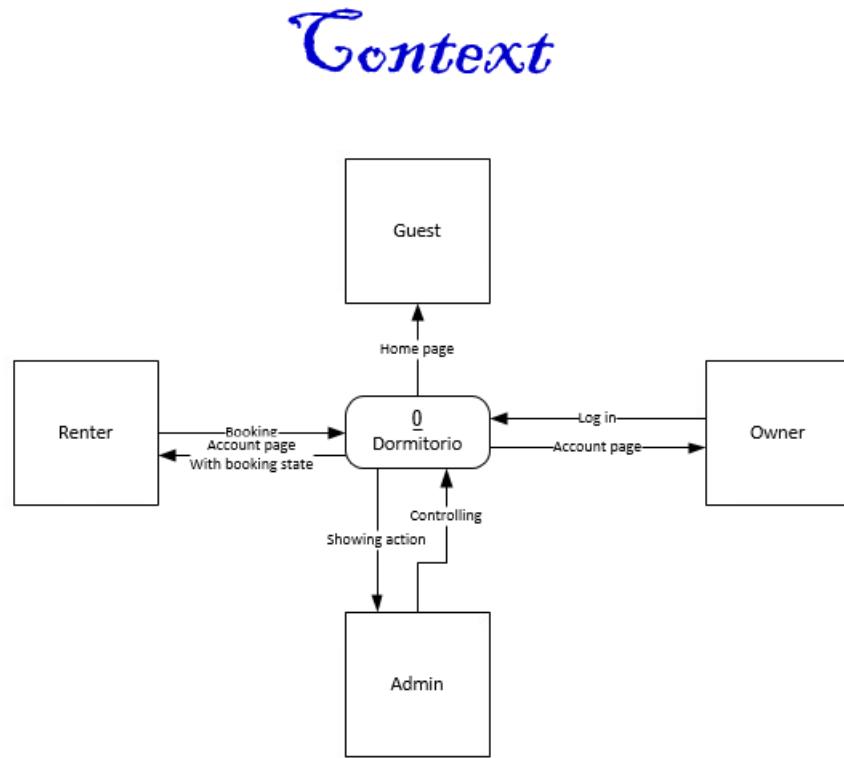


Figure 4.1: Context Diagram

4.1.2 Overview Diagram (Level 0)

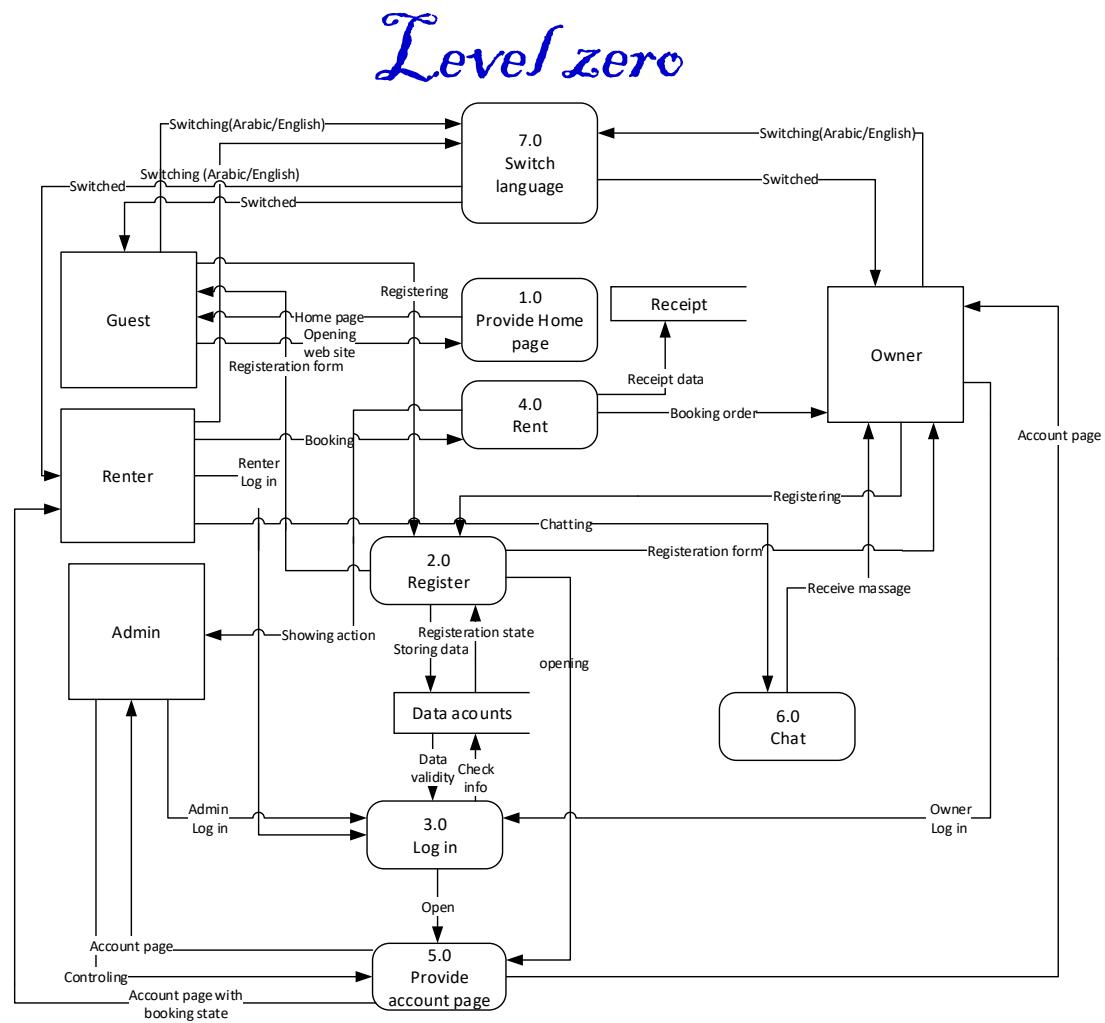


Figure 4.2: Overview Diagram (Level 0)

4.1.3 Detailed DFDs

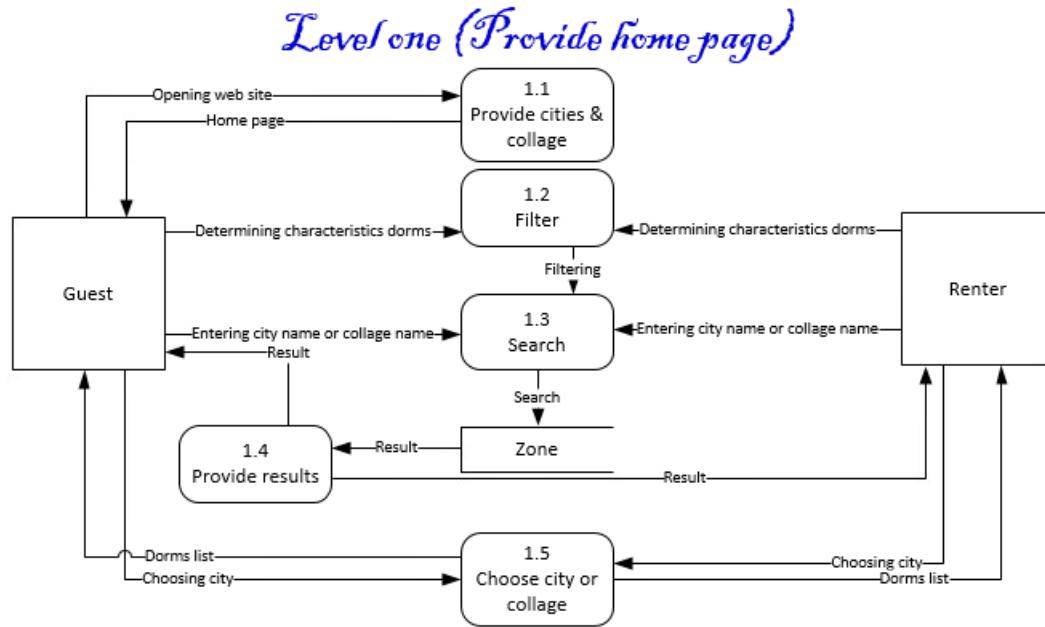


Figure 4.3: Level one (Provide home page)

Level one (Rent)

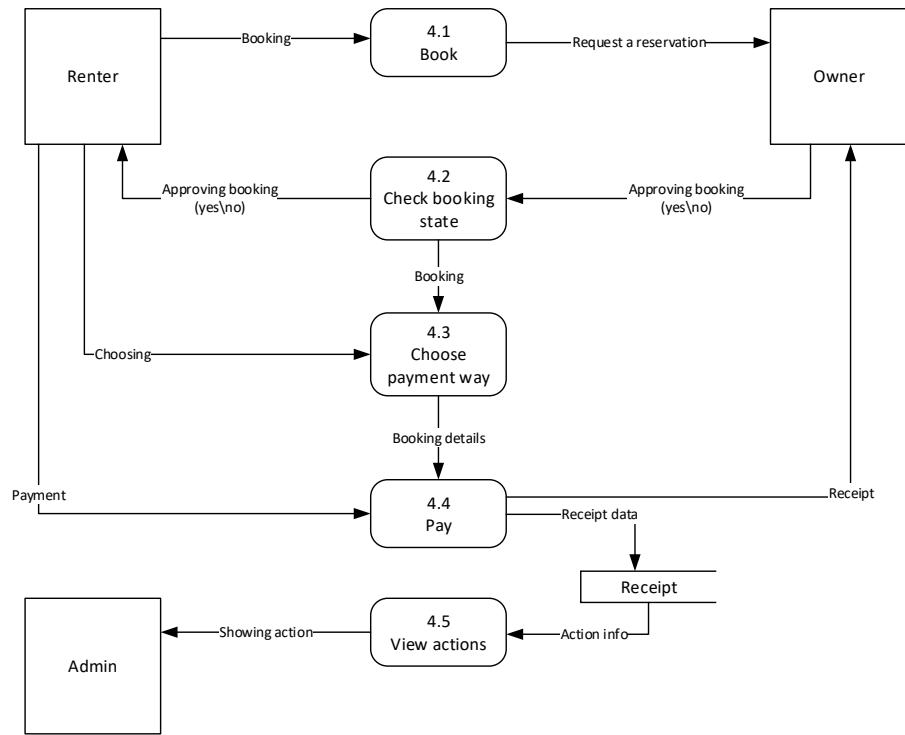


Figure 4.4: Level one (Rent)

Level one (provide account page)

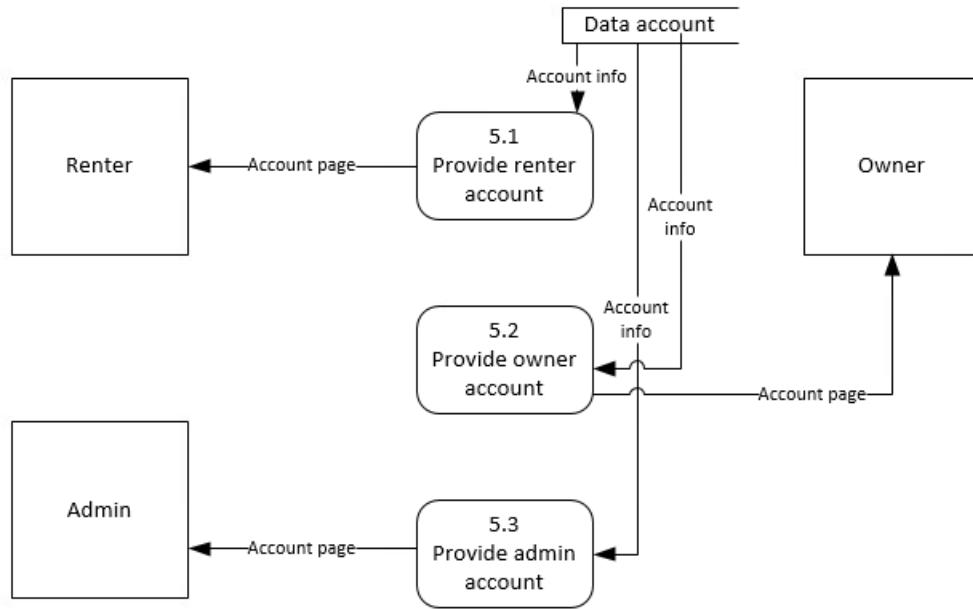


Figure 4.5: Level one (Provide account page)

Level one (chat)

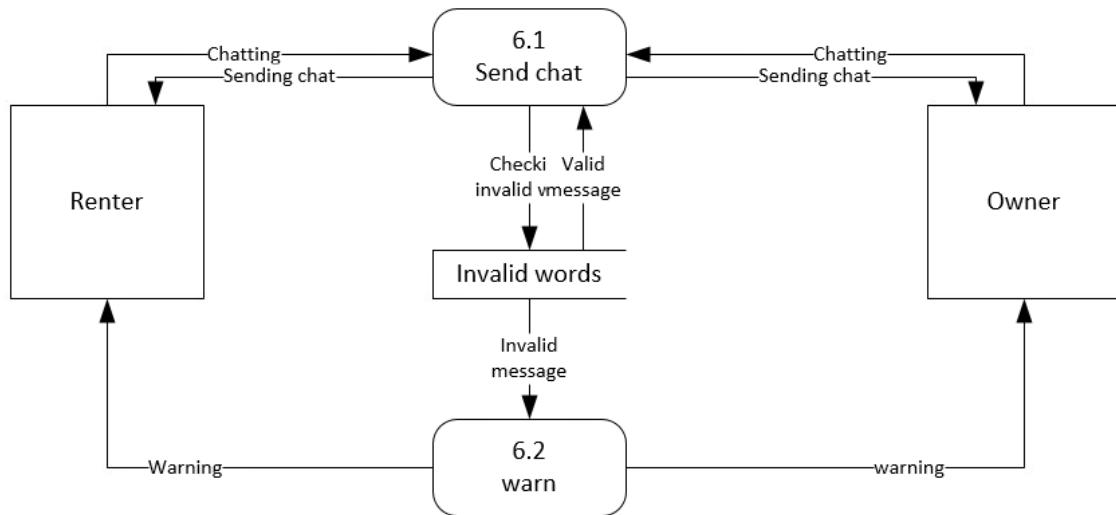


Figure 4.6: Level one (Chat)

Level two (choose city or collage)

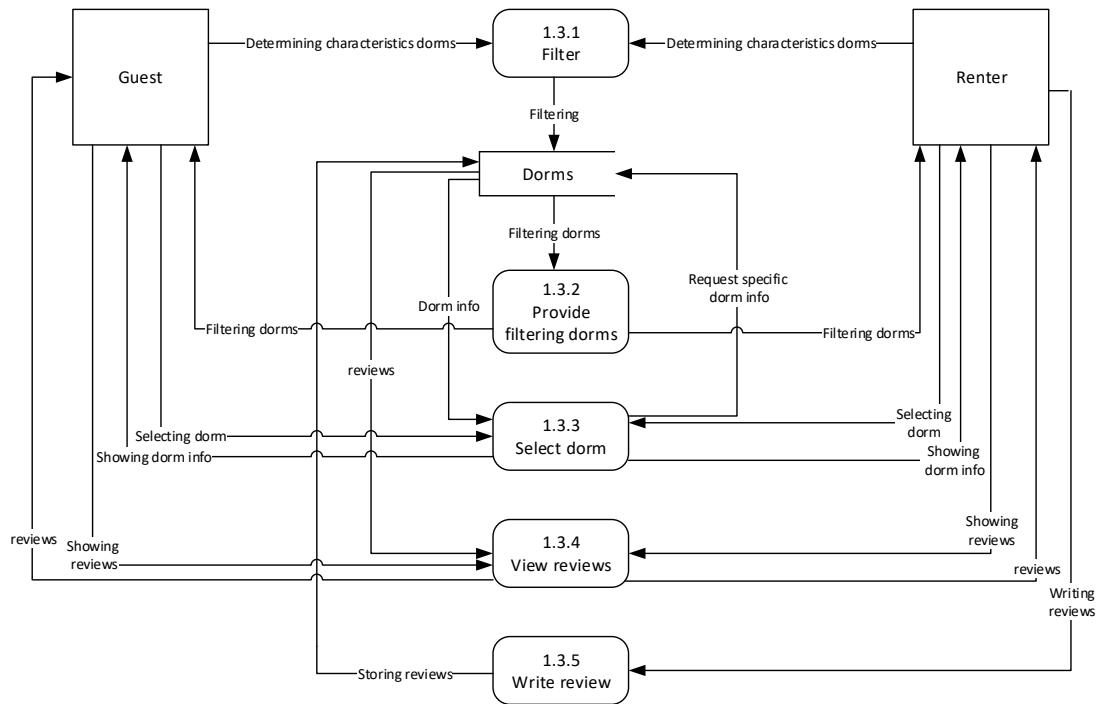


Figure 4.7: Level two (Choose city or collage)

Level two (Provide renter account)

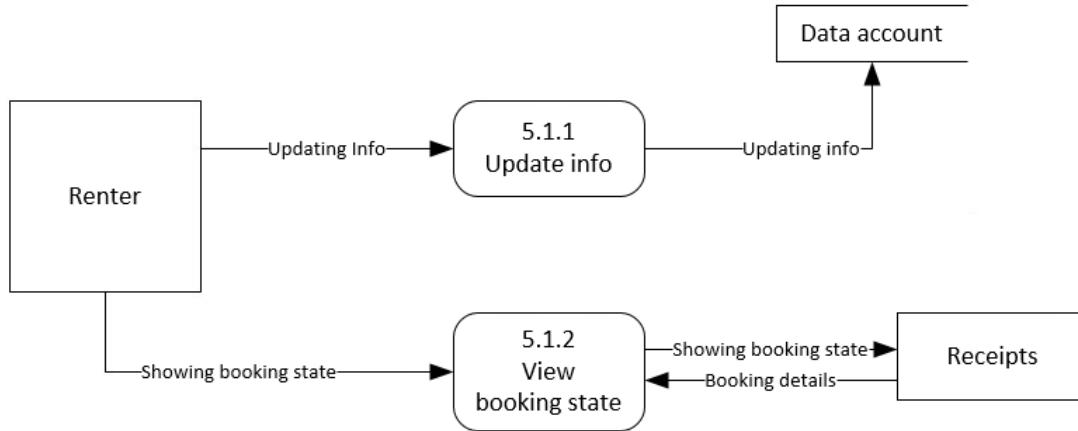


Figure 4.8: Level two (Provide renter account)

Level two (Provide owner account)

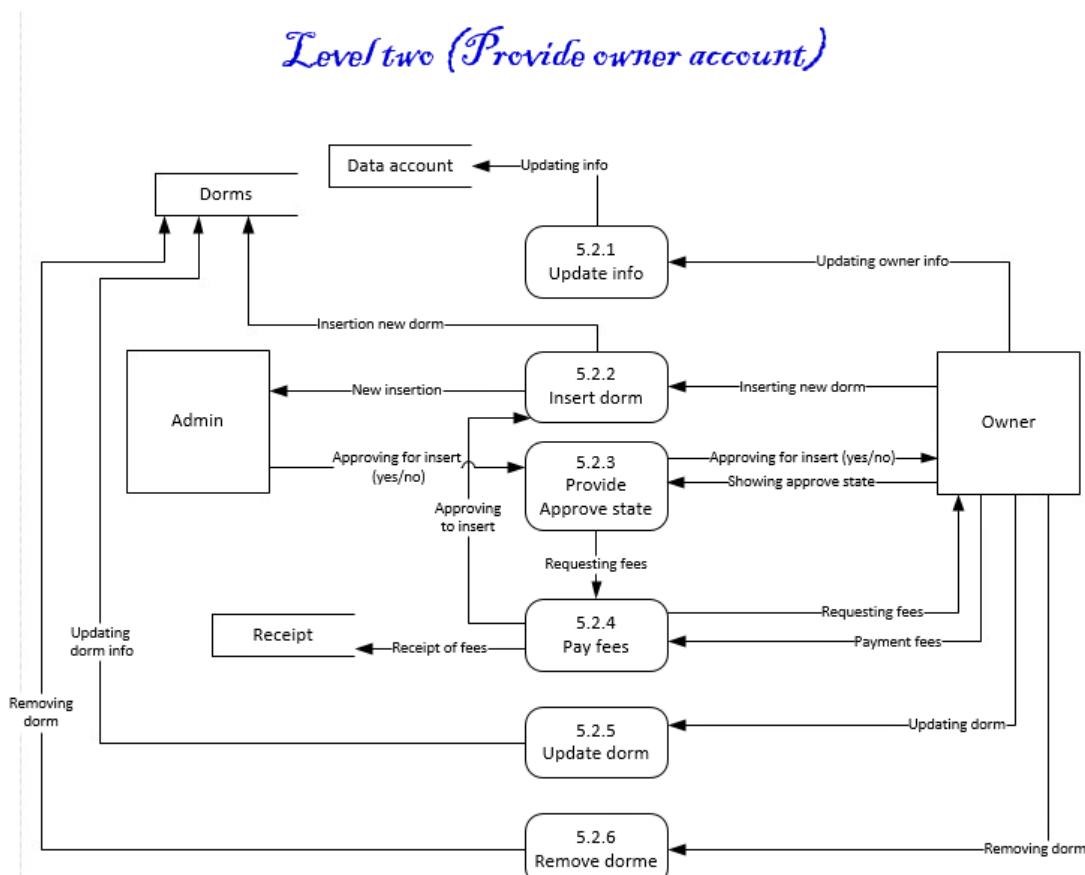


Figure 4.9: Level two (Provide owner account)

Level two (Admin account)

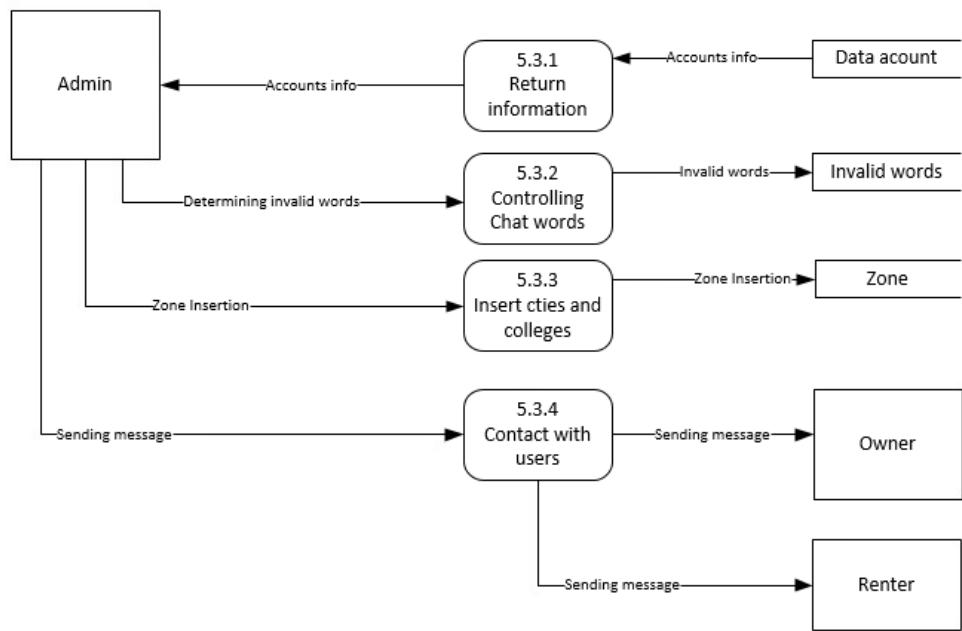


Figure 4.10: Level two (Admin account)

4.1.4 Data Dictionary of DFD

Shapes of Data Flow Diagram	Elements of Data Flow Diagram	Definition	Characteristic
	External Entity	An external entity is a person ,organization, organization unit, or system that is external to the system, but interacts with it	<ul style="list-style-type: none"> - Every external entity has a unique name that is a noun, and a description. - Every external entity has at least one input or output data flow.
	Process	A process is an activity or a function performed for some specific business reason	<ul style="list-style-type: none"> - Every process has a unique name that is an action-oriented verb phrase, a number, and a description. - Every process has at least one input and output data flow.
	Data Store	A data store is a collection of data that is stored in some way	<ul style="list-style-type: none"> - Every data store has a unique name that is a noun, and a description. - Every data store has at least one input and output data flow.
	Data Flow	A data flow is a single piece of data, or a logical collection of several pieces of information.	<ul style="list-style-type: none"> - Every data flow has a unique name that is a noun, and a description. - Every data flow connects to at least one process.

Figure 4.11: Data Dictionary of DFD

Our reference is [5].

4.2 DATA Design

4.2.1 Entity Relationship Diagram (ERD)

4.2.1.1 Description of Entities

Admin: He/she is responsible for managing the users' information and dorms' information. Admin contains Admin ID, Name, Password, Email and Phone Number.

Owner: He/she is responsible for posting the dorms on the website with their specifications, features and pictures ,in order to rent them to female students. Owners contains National ID/Residency Number, License Number, Name, Password, Email and Phone Number.

Renter: It represents the female student who wants to rent a dorm. Renter contains National ID/Residency Number, Name, Password, Email and Phone Number.

Dorm: It represents university dorms or apartments. Dorm contains Serial number, Location, Pictures, Number of all inserted room, Number of Available rooms, Price, Dorm type, properties (internal [A/C, Fridge, Microwave, Oven, kitchen] and external [elevator, fire alarm, parking]), Managed type, and Reviews.

Zone: It uses to filter dorms based on the city or university which is selected by the renter. Zone contains Zone ID, City, Colleges, and Language.

4.2.1.2 Description of Relations

Admin & Dorm & Admin (Insert): It is a ternary relationship. For simplification, we convert it into two binary relationship [8]:

1. Owner & Dorm (Insert): This relationship shows the relation between two entities (Owner) & (Dorm). The owner can insert one or more dorms. The dorm has one and only one owner. The attributes of the relationship are Date of insert and Fee payment.

2. Admin & Dorm (Insert (Approve)): This relationship shows the relation between two entities (Admin) & (Dorm). An Admin should approve the request of dorm insertion. The Admin can one or more request. The dorm will be approved by

one and only one admin. The attributes of the relationship is Admin decision.

Renter & Dorm (Book): This relationship shows the relation between two entities (Renter) & (Dorm). The renter can book one and only one room from the dorm. The dorm can be booked by more than one renter because the dorm may have more than one room. The attributes of the relationship are Payment information, Date of book, Date of payment, and Payment amount.

Owner & Renter (Contact): This relationship shows the relation between two entities (Owner) & (Renter). The owner can contact with many Renters and the renter can contact with many owners. The attributes of the relationship are Chat, Date of contact, and Exact time.

Zone & Dorm (Located in): This relationship shows the relation between two entities (Zone) & (Dorm). The dorm should locates in one city (two records in Zone: Arabic and English). Each zone has many choices of dorms.[5]

4.2.1.3 Drawing ERD

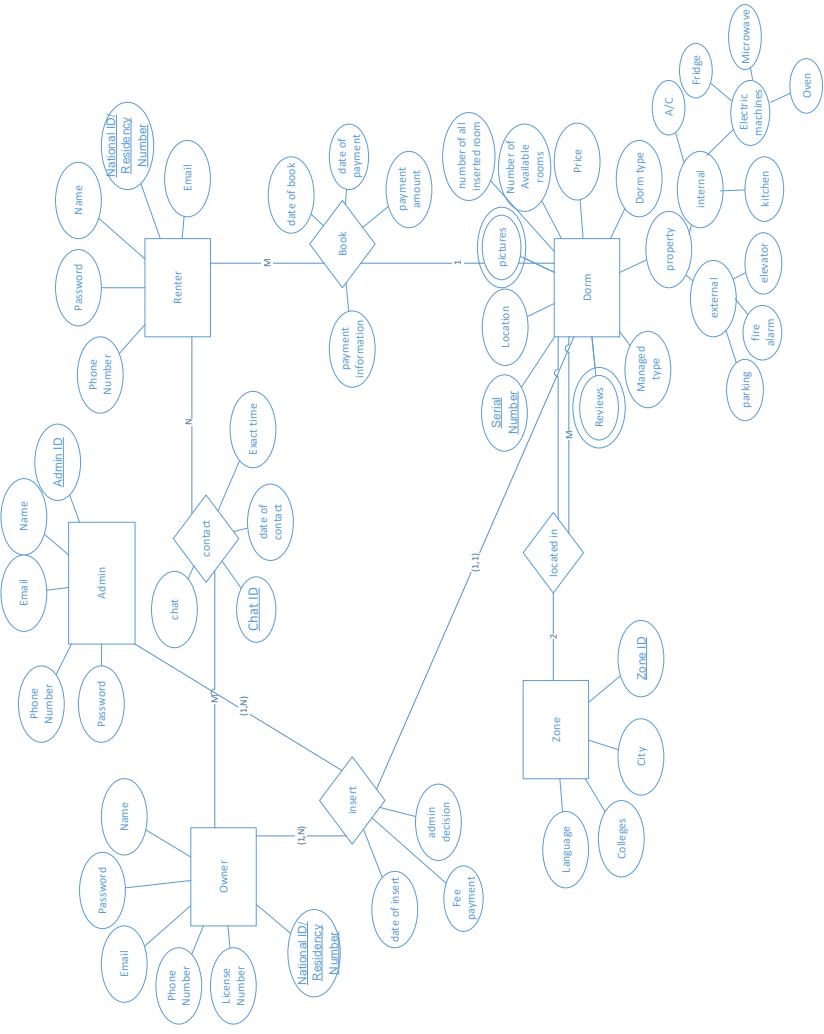


Figure 4.12: ER Diagram

4.2.1.4 Data Dictionary of ERD

	Entity is an object or component of data or a collection of related attributes of properties which you want to store information an entity is represented as rectangle in an ER diagram.
	Relationship is represented by diamond shape in ER diagram it shows the Relationship among entities and how entity share information and It also defined on basis of matching key columns.
	Attribute describes the property of an entity or the properties which define the entity type. An attribute is represented as Oval in an ER diagram.
	Key attribute can uniquely identify an entity from an entity set the text of key attribute is underlined.
	Is represented by an oval comprising of ovals An attribute consisting more than one value for a given entity. Also define as an attribute that is a combination of other attributes.
	Links entity set to attributes and entity set to a relationship represented as a line in an ER diagram.
	Multivalued attributes are those that are can take on more than one value.

Figure 4.13: Data dictionary of ERD

Our reference is [6].

4.2.2 Relation Database Schema

Admin
<u>Admin ID</u> A. Name A. Email A. Phone Number A. Password
Renter
<u>R. National ID/Residency Number</u> R. Name R. Email R. Phone Number R. Password
Owner
<u>O. National ID/Residency Number</u> License Number O. Email O. Phone Number O. Password O. Name
Zone
<u>Zone ID</u> City Colleges Language
Dorm
<u>Serial Number</u> Location Number of all inserted room Number of Available rooms Price Dorm type A/C Fridge Microwave Oven Kitchen elevator Fire alarm Parking Managed type <u>O. National ID/Residency Number</u> A. admin ID Data of insert Fee payment Admin decision
Dorm_Pictures
<u>D.Serial Number</u> Pictures
Dorm_Reviews
<u>D.Serial Number</u> Reviews
Booking
<u>R. National ID/Residency Number</u> <u>D.Serial Number</u> Date of book Date of payment Payment amount Payment information
Contact
<u>Chat ID</u> Chat <u>O. National ID/Residency Number</u> <u>R. National ID/Residency Number</u> Date of contact Exact time
Located in
<u>Z. Zone ID</u> <u>Serial Number</u>

Figure 4.14: Relation database schema

4.2.2.1 Attributes

Attributes	Data type
Admin ID	Integer
Name	String
Email	VARCHAR
Phone Number	Integer
Password	VARCHAR

Table 4.1: Admin Table

Attributes	Data type
National ID / Residency Number	Integer
Name	String
Password	VARCHAR
Email	VARCHAR
Phone number	Integer
License	VARCHAR

Table 4.2: Owner Table

Attributes	Data type
Zone ID	Integer
Language	String
Collage	String
City	String

Table 4.3: Zone Table

Attributes	Data type
National ID / Residency Number	Integer
Name	String
Phone number	Integer
Password	VARCHAR
Email	VARCHAR

Table 4.4: Renter Table

Attributes	Data type
Serial Number	Integer
O. National ID / Residency Number	Integer
A.Admin ID	Integer
Location	BLOB
Number of inserted room	Integer
Number of available rooms	Integer
Price	Float
Dorm type	String
Manged type	String
A/C	Boolean
Fridge	Boolean
Microwave	Boolean
Kitchen	Boolean
Oven	Boolean
Elevator	Boolean
Fire alarm	Boolean
Parking	Boolean
Date of Insert	Date
Fee payment	VARCHAR
Admin decision	VARCHAR

Table 4.5: Dorm Table

Attributes	Data type
Chat ID	Integer
Chat	VARCHAR
R. National ID / Residency Number	Integer
O. National ID / Residency Number	Integer
Date of contact	Date
Exact time	Time

Table 4.6: Contact Table

Attributes	Data type
Z.Zone ID	Integer
D.Serial number	Integer

Table 4.7: Dorms cities Table

Attributes	Data type
D.Serial number	Integer
Picture	BLOB

Table 4.8: Dorm-pictures

Attributes	Data type
D.Serial number	Integer
Review	VARCHAR

Table 4.9: Dorm-reviews

4.2.2.2 Relations

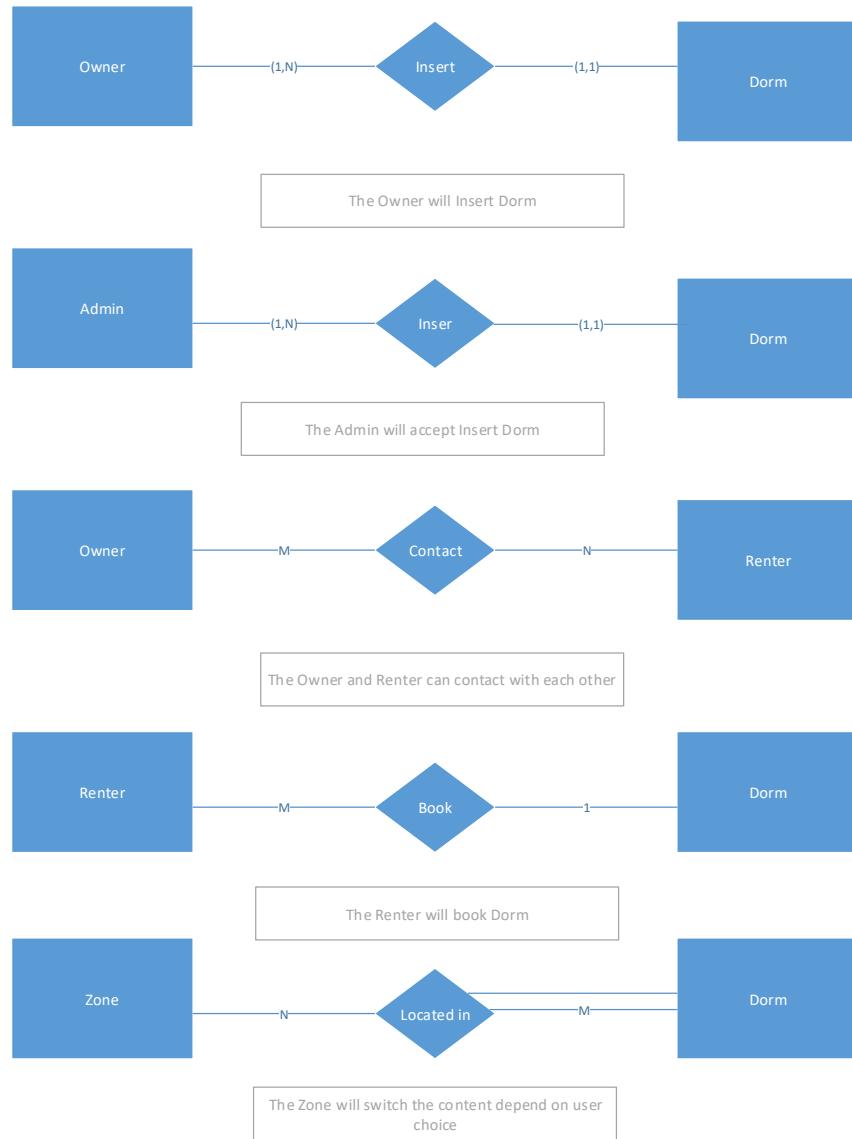


Figure 4.15: Relationships in ERD

4.3 Graphical User Interface (GUI) Design

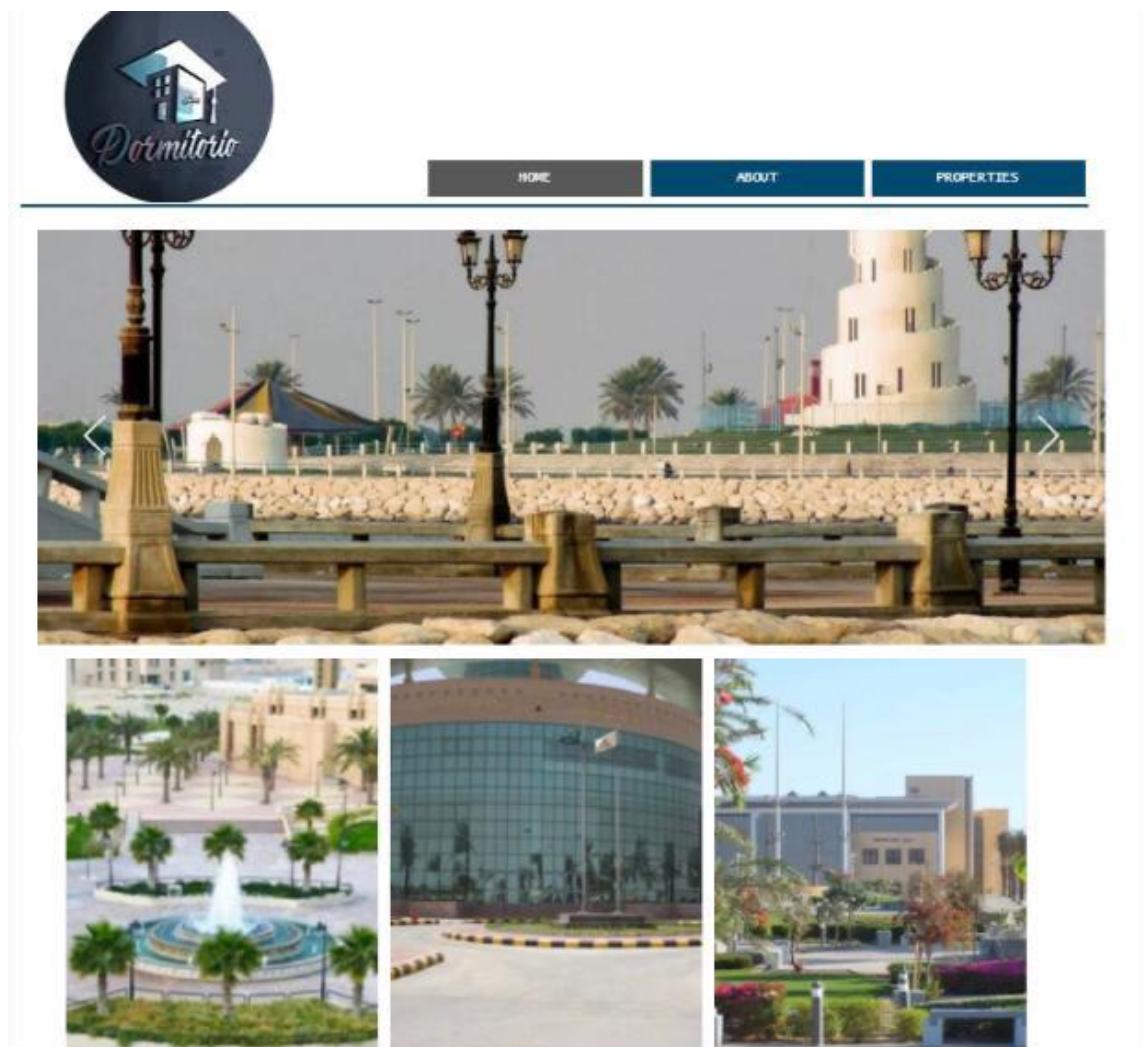


Figure 4.16: Home Page

The students encounter many problems during the search of appropriate dorms such as the distance between the campus and the dorm, also, the high demand costs. We build a website that serves university students (Females) in searching for suitable dorms whether inside or outside the university campus. The website will provide the students with apartments suitable for their needs in terms of size, price, and services available around the apartment as restaurants, supermarkets, laundries, hospitals, and parking. In addition, the advantages of housing will be easily explained to students on the website; whether it is vacant or not, and pictures of it. The students are allowed to book from the site besides communicating with the owner of the residence with administrators control for the chat to prevent any means of communication outside the site.

Figure 4.17: About us

Properties



Property 1

A college dormitory providing sleeping and residential quarters for female students only.

Price for **30 nights:**
SAR 1200

[View More](#)



Property 2

A single apartment to enjoy The privacy and comfort of a home.

Price for **30 nights:**
SAR 2000

Figure 4.18: Properties



Figure 4.19: About the dorm



HOME

ABOUT

PROPERTIES

Product Order Form

Place your order now for only \$15!

First Name

Last Name

Email *

Address

Phone Number

Buy Now



Thanks for your order!

Figure 4.20: Payment Page

Chapter 5

Conclusion and Future Work

5.1	Lessons Learned	50
5.2	Future Work	50

5.1 Lessons Learned

First, we learn how to be teamwork. Second, we got a chance to know how to use L^AT_EX. Plus to that, we learned to write requirements in a way that suits the system to be created. Also, we learned to think more deeply. In addition, we learned to analyze the problem logically. We learned how to apply DFD and ERD concepts to a real system. We learned to divide tasks evenly and communicate with the professor comfortably. We learned to accept other opinions. Finely we learned to accept our mistakes and work on them.

5.2 Future Work

Inshaa Allah We will Develop our website to cover areas beyond Saudi Arabia. In addition to that, it will be suitable for both male and female students. In future, more payment options will be added including cash option. We will try to make this website to cover some global universities if applicable. The database will be extended if need. We will work to increase the security procedures. This website will serve also employees, and will have other more languages. In other hand It's planed to add an Application to work beside this website.

Chapter 6

System Implementation and Evaluation

6.1	Database Implementation	53
6.1.1	Admin	54
6.1.2	Booking	55
6.1.3	Dorm	56
6.1.4	Dorm images	58
6.1.5	Owner	58
6.1.6	Renter	59
6.2	Graphical User Interface Implementation	60
6.2.1	Gust User	63
6.2.1.1	Home Page	63
6.2.1.2	Register	64
6.2.1.3	Contact	65
6.2.1.4	FQA	66
6.2.2	Owner	67

6.2.2.1	Home Page	67
6.2.2.2	Register	68
6.2.2.3	Contact	69
6.2.2.4	FQA	70
6.2.2.5	Insert dorm	71
6.2.2.6	Your dorm	73
6.2.2.7	My dorms booking	74
6.2.2.8	Your profile	76
6.2.3	Renter	77
6.2.3.1	Home Page	77
6.2.3.2	Register	78
6.2.3.3	Contact	79
6.2.3.4	FQA	80
6.2.3.5	Dorm List	81
6.2.3.6	Dorm Details	82
6.2.3.7	Booking	83
6.2.4	Admin	85
6.2.4.1	Home Page	85
6.2.4.2	Customer Booking Page	86
6.2.4.3	Dorms Registration Requests	87

6.1 Database Implementation

In this section, we explain the database implementation process. The database is an essential part of our system. It is crucial to store all our system's related data and transactions in a reliable database system. To implement the database system, we used MySQL which is a freely available open-source Relational Database Management System that uses Structured Query Language. SQL is the most popular language for adding, accessing, and managing content in a database. It is noted for its quick processing, proven reliability, ease, and flexibility of use. MySQL is an essential part of almost every open-source PHP application. Additionally, we selected MySQL because it is a fast, easy to use, and open source. Also, MySQL is very compatible with PHP, one of the most popular languages for web development, in which we based our website.[4]

We use XAMPP server which is free software and very easy to use and manage. Its also easy to download in all operation systems such as Windows, Macintosh, and Linux. We use XAMPP, as we see in Figure 6.1, to access MySQL database by MySQL Module, and provide the ability to use PHP codes by apache module.

the download link click here.



Figure 6.1: XAMPP

Afterwards, we proceeded to create our database. The database name is "dormitorio". The database design was based on the ERD system .

Table	Action	Rows	Type	Collation	Size	Overhead
admin	Browse Structure Search Insert Empty Drop	7	MyISAM	utf8_general_ci	2.4 Kib	-
booking	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	2.3 Kib	224 B
dorm	Browse Structure Search Insert Empty Drop	1	MyISAM	utf8_general_ci	4.2 Kib	2.0 Kib
dorm_images	Browse Structure Search Insert Empty Drop	32	MyISAM	utf8_general_ci	2.9 Kib	-
owner	Browse Structure Search Insert Empty Drop	3	MyISAM	utf8_general_ci	2.2 Kib	-
renter	Browse Structure Search Insert Empty Drop	4	MyISAM	utf8_general_ci	2.2 Kib	-
6 tables	Sum	48	InnoDB	utf8mb4_general_ci	16.3 Kib	2.3 Kib

Figure 6.2: Database Tables

In the database, we have a sex tables, Figure 6.2 , as following:

6.1.1 Admin

This table contains the information of admins in order to secure logged-in to the admin page that allows the admin to manage the owner and renter page including ID, Name, Email, and Password. In addition, the admin can accept or reject the inserted dorms. The primary key in the table is "ID". Figure 6.3 shows the Admin table in our database.

+ Options										
		←	→	▼	id	name	email	pass		
<input type="checkbox"/>		Edit		Copy		Delete	1	Amal	2160007870@iau.edu.sa	amal@123
<input type="checkbox"/>		Edit		Copy		Delete	2	Ahad	2190004537@iau.edu.sa	ahad@456
<input type="checkbox"/>		Edit		Copy		Delete	3	Rawan	2190001350@iau.edu.sa	rawan@789
<input type="checkbox"/>		Edit		Copy		Delete	4	Sukainah	2190000389@iau.edu.sa	sukainah@987
<input type="checkbox"/>		Edit		Copy		Delete	5	Fay	2190005733@iau.edu.sa	fay@333
<input type="checkbox"/>		Edit		Copy		Delete	6	Lama	2190005768@iau.edu.sa	lama@543
<input type="checkbox"/>		Edit		Copy		Delete	7	Nouf	2190004720@iau.edu.sa	nouf@989

Figure 6.3: Admin Table

6.1.2 Booking

The table "Booking" holds all information regarding the booking of the renter who wants to rent a dorm. It has 8 attributes: Id, Booking date, customer name, customer phone, customer email, Check in date, Check out date, and Dorm id. The website gives the ability to only the renter registered to the website, to enter their booking information, all the columns are required to fill out. The primary key in the table is "ID". Figure 6.4 shows the Booking table in our database.

SELECT * FROM "booking"

Profiling [\[Edit inline\]](#) [\[Edit\]](#)

<input type="checkbox"/> Show all	Number of rows:	25 <input type="button" value="▼"/>	Filter rows:	Search this table	Sort by key:	None <input type="button" value="▼"/>					
+ Options											
	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	<input type="checkbox"/> id	booking_date	customer_name	customer_phone	customer_email	check_in_date	check_out_date	dorm_id
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	4	06/04/2021	Arwa	566899887	Anwa dd@gmail.com	05/09/2022	01/09/2022	17
<input checked="" type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	3	06/04/2021	Sara	534545678	Sara_11@gmail.com	04/29/2021	12/21/2021	15
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	5	06/04/2021	Amal	567855543	name@gmail.com	04/06/2021	04/28/2021	18
<input type="checkbox"/>	<input type="button" value="Edit"/>	<input type="button" value="Copy"/>	<input type="button" value="Delete"/>	6	06/04/2021	Amal	567855543	name@gmail.com	04/14/2021	04/22/2021	18

Check all With selected:

Show all | Number of rows: 25 Filter rows: Search this table Sort by key: None

Query results operations

Figure 6.4: Booking Table

6.1.3 Dorm

The table "Dorm" holds all information about the dorms that inserted by the owners. It has 31 attributes ID, Name, Status, price, phone, picture, description, built in, city, rent type, pay type, min bed, min bath, min geo, min garage, AC , fire alarm, monetoring camera, kitchen, guard, oven, parking, has lock, microwave, elevator, mailbox, waterfront, waterfront desc, vedio 1, vedio 2, vedio 3, and owner id. The primary key in the table is "ID". Figures 6.5, 6.5, and 6.5 show the Dorm table in our database.

	Options														
	id	name	status	price	phone	picture	description	built_in	city	rent_type	pay_type	min_bed	min_bath	min_geo	min_garage
<input type="checkbox"/>	20	MY BEAUTIFUL DORM	For Rent	1000	566886545	IMG-5702.JPG	A simple dorm overlooking the park, only 4 minute...	2021	Dammam	every year	every six month	1	1	40.0	0
<input type="checkbox"/>	17	NEW DORM	For Rent	700	587634578	PHOTO-2021-06-08-11-28.jpg	A modern room characterized by attractive colors...	2020	Al-hasa	every six month	every month	1	1	65.0	3
<input type="checkbox"/>	18	TRLPLE ROOM	For Rent	950	587634578	PHOTO-2021-06-08-11-32.jpg	A very elegant room with its colors and furniture ...	2021	Jubail	every six month	every month	3	2	100.0	2
<input type="checkbox"/>	19	DORM	For Rent	800	566886545	IMG-5695.JPG	A distinctive individual residence located re...	2020	Qatif	every year	every six month	1	1	90.0	1
<input type="checkbox"/>	15	SINGLE DORM	For Rent	1200	55661234	dorm1.jpg	This room is located in the heart of Al Khobar wh...	2021	Khobar	every year	every six month	1	1	60.0	2
<input type="checkbox"/>	16	SHARED DORM	For Rent	2000	55661234	PHOTO-2021-06-08-11-31_1.jpg	This room is located in Dammam. You can share this...	2019	Dammam	every year	every six month	2	0	70.0	2

← Check all With selected: Edit Copy Delete Export

Figure 6.5: Dorm Table 1

min_geo	min_garage	A_C	fire_alarm	monitoring_camera	kitchen	guard	oven	parking	has_lock	microwave	elevator	mailbox	waterfront	waterfront_desc	vedio_1	vedio
40.0	0	1	1		1	1	0	1	1	0	0	0	0	1	0	NULL
65.0	3	1	1		1	1	1	1	1	0	0	0	1	1	1	NULL
100.0	2	1	0		1	1	0	0	1	1	0	0	0	0	0	NULL
90.0	1	1	1		1	1	1	0	1	0	1	0	1	0	0	NULL
60.0	2	1	1		1	1	1	0	1	0	0	0	0	1	0	NULL
70.0	2	1	1		1	1	0	1	1	0	1	0	0	0	0	NULL

Console

Figure 6.6: Dorm Table 2

vedio_1	vedio_2	vedio_3	owner_id	registration_date	is_accepted	is_booked
11	NULL				1	0
10	NULL				1	1
10	NULL				1	1
11	NULL				1	0
9	NULL				1	1
9	NULL				1	0

Figure 6.7: Dorm Table 3

6.1.4 Dorm images

The table "Dorm images" holds all pictures about the inserted dorm table. It has 3 attributes ID, image, and dorm id. The primary key in this table is "ID" as shown in Figure 6.8.

+ Options		id	image	dorm_id
<input type="checkbox"/>	Edit Copy Delete	8	29c6a4bc-9567-45d6-8c19-0bbffda68326.jpg	7
<input type="checkbox"/>	Edit Copy Delete	5	29c6a4bc-9567-45d6-8c19-0bbffda68326.jpg	6
<input type="checkbox"/>	Edit Copy Delete	6	banner.jpg	6
<input type="checkbox"/>	Edit Copy Delete	7	thumbnail_lmsor-aaf-f-fy-imgmaa_2mbj05i2.jpg	6
<input type="checkbox"/>	Edit Copy Delete	9	banner.jpg	7
<input type="checkbox"/>	Edit Copy Delete	10	thumbnail_lmsor-aaf-f-fy-imgmaa_2mbj05i2.jpg	7
<input type="checkbox"/>	Edit Copy Delete	11	29c6a4bc-9567-45d6-8c19-0bbffda68326.jpg	8
<input type="checkbox"/>	Edit Copy Delete	12	thumbnail_lmsor-aaf-f-fy-imgmaa_2mbj05i2.jpg	8
<input type="checkbox"/>	Edit Copy Delete	13	left-img.jpg	9
<input type="checkbox"/>	Edit Copy Delete	14	thumbnail_lmsor-aaf-f-fy-imgmaa_2mbj05i2.jpg	9
<input type="checkbox"/>	Edit Copy Delete	15		10
<input type="checkbox"/>	Edit Copy Delete	16		8
<input type="checkbox"/>	Edit Copy Delete	17		8
<input type="checkbox"/>	Edit Copy Delete	18		8
<input type="checkbox"/>	Edit Copy Delete	19		8
<input type="checkbox"/>	Edit Copy Delete	20		11
<input type="checkbox"/>	Edit Copy Delete	21		12
<input type="checkbox"/>	Edit Copy Delete	22		12

Figure 6.8: Dorm images Table

6.1.5 Owner

The table "Owner" holds all information about the owners. It has 13 attributes ID, Name, Email, Phone, Picture, Facebook, public email, twitter, website, Fax, License, Pass, and National id. The primary key in the table is "ID" as shown in Figure 6.9.

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

+ Options

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	12	DrNahla	566886545	Dr.Nahla@gmail.com	NULL	NULL	NULL	NULL	NULL	9443543	Nahla123123	1994466855	
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	11	DrFatima	566612345	Dr.Fatima_Alghamdy@gmail.com	NULL	NULL	NULL	NULL	NULL	6433543	Fatima123123	1238734666	
<input type="checkbox"/> Check all <input type="checkbox"/> With selected: <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete <input type="checkbox"/> Export																

Show all | Number of rows: 25 | Filter rows: Search this table | Sort by key: None

Query results operations

Print Copy to clipboard Export Display chart Create view

Figure 6.9: Owner Table

6.1.6 Renter

The table "Renter" holds all information about the renters. It has 13 attributes ID, Name, Email, Phone, Picture, Facebook, public email, twitter, website, Fax, License, Pass, and National id. The primary key in the table is "ID" as shown in Figure 6.10.

+ Options

	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	6	Sara	534545678	Sara_11@gmail.com	NULL	NULL	NULL	NULL	NULL	Sara12345	1000384727		
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	5	Anwa	566899987	Anwa.dd@gmail.com	NULL	NULL	NULL	NULL	NULL	Arwa123123	1947372848		
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	7	Asmaa	541212134	Asmaa_199@gmail.com	NULL	NULL	NULL	NULL	NULL	Asmaa11	1885256532		
	<input type="checkbox"/> Edit	<input type="checkbox"/> Copy	<input type="checkbox"/> Delete	8	soso	565585790	soso@gmail.com	NULL	NULL	NULL	NULL	NULL	1234	123456789		
<input type="checkbox"/> Check all <input type="checkbox"/> With selected: <input type="checkbox"/> Edit <input type="checkbox"/> Copy <input type="checkbox"/> Delete <input type="checkbox"/> Export																

Figure 6.10: Renter Table

6.2 Graphical User Interface Implementation

In this section, we describe the implementation of the webpages. The webpages are written using php, html and css. Here we mention the core php codes along with screenshots and a brief description. The following table shows an overview of the most important pages and a brief description.

Page User	Page Name	Description
Admin	Admin-page	The home page for the admin ,which gives general information about the customer booking and dorms registration requests.
	Customer booking	Customer Booking page, which allows the admin to see all information of each booking happened on the site.
	Dorms registration requests	Dorms registration requests page, allows the admin to see new dorms request and accept or reject the dorm.
Guest	Index	The home page which gives general information about the website and enables the website guest to navigate to other pages such as register, contact-us, log-in, view dorms, FAQ page, and view profile page if you want to register.
	Register	register and login page, enables guest to create an account in order or login, to benefit from the services.
	User-profile	user-profile page, view profile page if you want to register.
	Contact-2	Allows the guest to send complaints or suggestions for us.
	FAQ	Allows the guest to see the frequently asked questions about our website and show the answer.
Renter	Index.renter	The home page which gives general information to renter about the website and enables the renter to navigate other pages such as view dorms list, FAQ page, log-out, contact-us, and show renter profile if the renter want modify the information, also enables the renter to book a dorm.
	User-profile-renter	Allows the renter to view profile page if the renter want to modify the information.
	Contact-renter	Allows the renter to send complaints or suggestions for us.
	FAQ-renter	Allows the renter to see the frequently asked questions about our website and show the answer.
	Booking	view the booking detail for the renter to complete booking process.
Owner	Index.owner	The home page which gives general information to owner about the website and enables the owner to navigate other pages such as view your dorms, view your dorms booking, view other's dorms ,FAQ page, log-out, contact-us, and show owner profile if owner want modify the information , also enables the owner to insert dorms.

Owner	FAQ-owner	Allows the owner to see the frequently asked questions about our website and show the answer.
	User-profile-owner	Allow the owner to view profile page if owner want to modify the information.
	Contact-owner	Allows the owner to send complaints or suggestions for us.
	Insert	Allow to owner to insert new dorm with the features, pictures, video, description, and price.
	Functions	Allow to owner to delete own dorm from website.
	Submit-property	Allow to owner to edit on his dorms and edit the features, pictures, video, description, and price.
	User-properties	Allow to owner to edit and view own dorm from .website
	My_dorms_booking	Allow the owner to see booking on his dorms.
Renter-Owner	Properties	Allow to user to show all dorms.
	Property_details	Allow to user to show every dorm details.
Admin-Renter-Owner	Log-out	log out from your profile.

6.2.1 Gust User

6.2.1.1 Home Page

At the beginning of the page, you will see the navigation bar for all the services in the website such as, Home ,Contact ,Dorms ,Services, and button for log in and register. Also we put at the middle of the page the search bar to enable the gusts to search by collage or city. In the bottom of the page, we put a brief description (about us) , quick link , team member names and quick contact with us via email. Home Page and its code are shown in Figures 6.11 and 6.12.

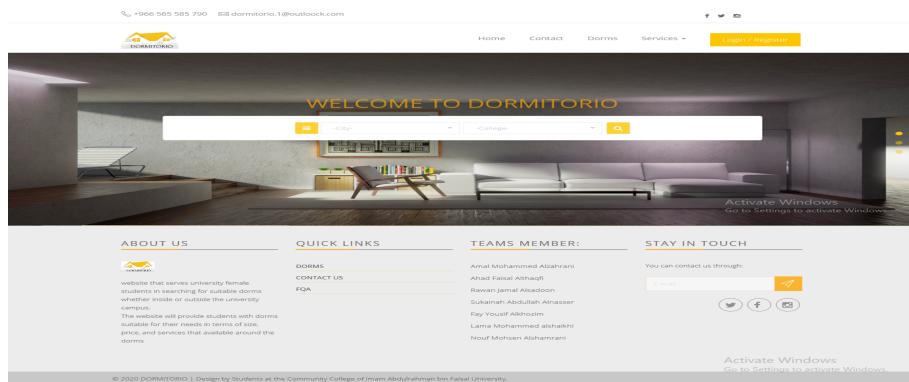


Figure 6.11: Home Page

```
<div class="button navbar-right">
<?php
    if(isset($_SESSION["user_id"])){
        echo '<button class="navbar-btn nav-button wow bounceInRight logout"
            onclick="window.open(\'logout.php\')"
            data-wow-delay="0.45s">Logout</button>';
    }
    else{
        echo '<button class="navbar-btn nav-button wow bounceInRight login"
            onclick="window.open(\'register.php\')"
            data-wow-delay="0.45s">Login /
            Register</button>';
    }
?
</div>
```

Figure 6.12: Home Page PHP Code

6.2.1.2 Register

In the top of the Register page, there is a navigation bar with one option (Home) which allows the user to go to the home page. In the left of the page, there is the register form; and in the right of the page, there is the log in form. In the bottom of the page there is a brief description (about us) , quick link , team member names, and quick contact with us via email. Register Page and its code are shown in Figures 6.13, 6.14, and 6.15.

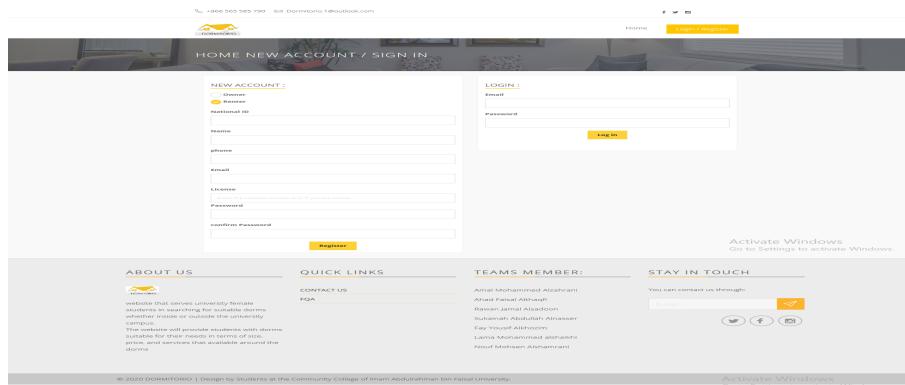


Figure 6.13: Register

```
<?php
$admin = "admin";
$owner = "owner";
$renter = "renter";

if(isset($_POST['Register'])){
    if ($_POST['user_type']==$owner)
    {
        if(empty($_POST['NID']) || empty($_POST['name']) || empty($_POST['phone']) || empty($_POST['email']) || empty($_POST['pass']))
        {
            //header("location:index.php?error=empty");
            echo "<script>alert('Please fill all the fields');</script>";
        }
        elseif($_POST['compass'] != $_POST['pass'])
        {
            echo "<script>alert('passwords don't match');</script>";
        }
        else
        {
            $sql="INSERT INTO owner (national_id , name , phone , email , license , pass) VALUES
            ('$_POST[NID]','$_POST[name]' , '$_POST[phone]' , '$_POST[email]' , '$_POST[license]' , '$_POST[pass]' )";
            $result=mysqli_query($conn , $sql) or die(mysqli_error($conn));
            //header("location:welcome1.php");
            echo "<script>alert('You are registered successfully');</script>";
        }
    }
}
```

Figure 6.14: Register as owner PHP code

```

if ($_POST['user_type']=='renter')
{
    if(empty($_POST['NID']) || empty($_POST['name']) || empty($_POST['phone']) || empty($_POST['email']) || empty($_POST['pass']))
    {
        //header("location:index.php?error=empty");
        echo "<script>alert('Pleas fill all the failds');</script>";

    }else if($_POST['conpass'] != $_POST['pass'])
    {
        echo "<script> alert('passwords don't match');</script>";
    }
    else
    {
        $sql="INSERT INTO renter (national_id , name , phone , email , license , pass) VALUES
        ('$_POST[NID]' , '$_POST[name]' , '$_POST[phone]' , '$_POST[email]' , '$_POST[license]' , '$_POST[pass]' )";
        $result=mysqli_query($conn , $sql) or die(mysqli_error($conn));
        //header("location:welcomel.php");
        echo "<script>alert('You are registered successfully');</script>";
    }
}
}

```

Figure 6.15: Register as renter PHP code

6.2.1.3 Contact

At the beginning of the page, you will see the navigation bar for all the services in the site such as, Home, Contact, Dorms, services, and button for log in and register. In the middle of the page, there is the contact form. In the bottom of the page, there is a map shows the current covered area in the site and there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.16 and 6.17.

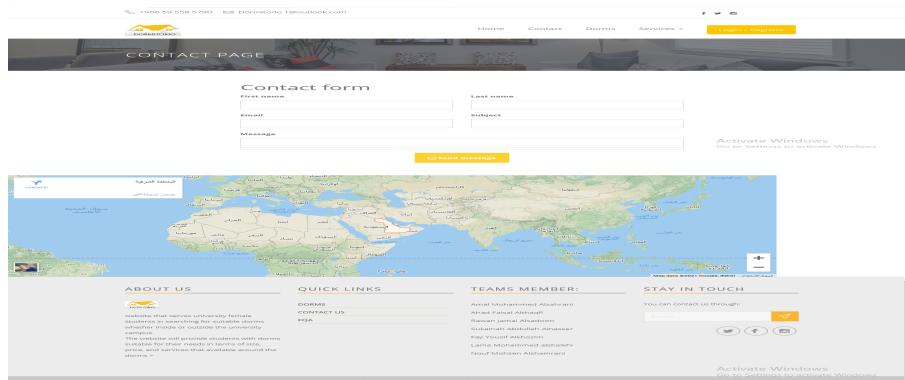


Figure 6.16: Contact

```

<?php
    if(isset($_SESSION["user_id"])){
        echo '<button class="navbar-btn nav-button wow bounceInRight logout"
        onclick="window.open(\'logout.php\')" data-wow-delay="0.45s">Logout</button>';
    }
    else{
        echo '<button class="navbar-btn nav-button wow bounceInRight login"
        onclick="window.open(\'register.php\')" data-wow-delay="0.45s">Login /
        Register</button>';
    }
?>

```

Figure 6.17: Contact PHP code

6.2.1.4 FQA

At the beginning of the page, you will see the navigation bar for all the services in the site such as, Home, Contact, Dorms, services and button for log in and register. In the middle of the page, there is some of the most frequent questions and the answers. In the bottom of the page, there is a brief description (about us) ,quick link ,team member names, and quick contact with us via email as shown in Figures 6.18 and 6.19.

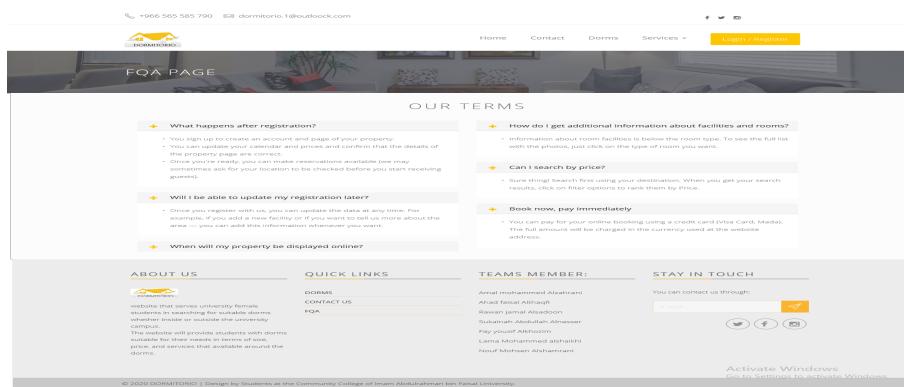


Figure 6.18: FQA

```

<div class="panel-body">
<ul>
<li> Information about room facilities is below the room type. To see the full list with the photos, just click on the type of room you want.</li>
</ul>
</div>

```

Figure 6.19: FQA code

6.2.2 Owner

6.2.2.1 Home Page

At the beginning of the page, you will see the navigation bar for all the services in the site such as, Home, Contact, Services and, button for log out. Also, we put in the middle of the page the search bar to enable the owner to search by collage or city. In the bottom of the page, we put brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.20 and 6.21.

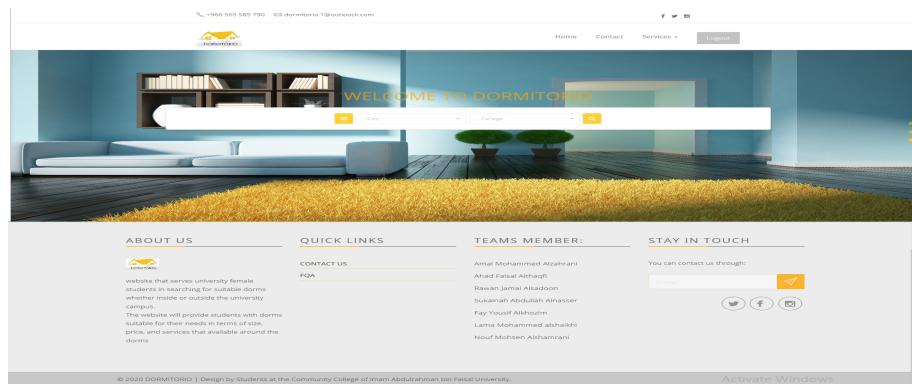


Figure 6.20: Home Page-Owner

```

<li><a href="register.php">Register | login</a> </li>
<?php
    if(isset($_SESSION['user_id'])) && $_SESSION['user_type'] =
    "owner" {
        echo'
            <li><a href="user-properties.php">your dorms</a> </li>
            <li><a href="My_dorms_bookings.php">My Dorms Bookings</a>
            </li>
            <li><a href="insert.php">insert dorms</a> </li>
        ';
    }
?>

```

Figure 6.21: Owner home page PHP code

6.2.2.2 Register

In the top of the page, there is a navigation bar with one option (Home) which allows the user to go to the home page. In the left of the page, there is the register form; and in the right of the page, there is log in form. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email. Register Page of the owner and its code are shown in Figures 6.22 and 6.23.

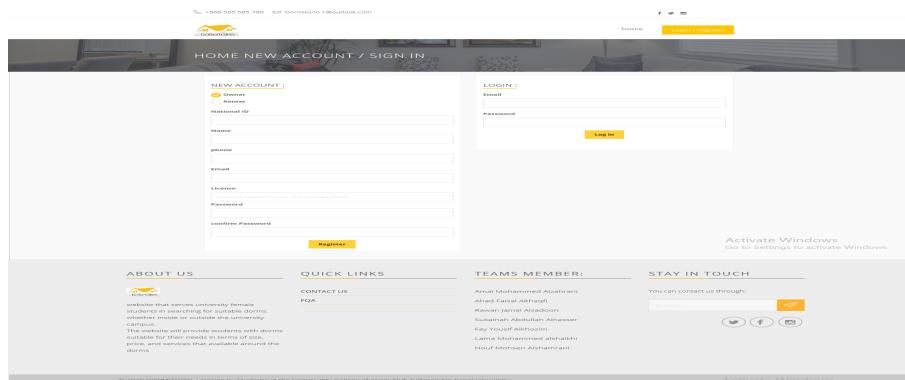


Figure 6.22: Register-Owner

```

▼ <?php
    $admin = "admin";
    $owner = "owner";
    $renter = "renter";
    if(isset($_POST['Register'])){
        if($_POST['user_type']=='$owner'){
            if(empty($_POST['NID']) || empty($_POST['name']) || empty($_POST['phone']) || empty($_POST['email']) || empty($_POST['pass'])){
                //header("location:index.php?error=empty");
                echo "<script>alert('Please fill all the fields');</script>";
            }elseif($_POST['cpassword'] != $_POST['pass']){
                echo "<script>alert('Passwords don't match');</script>";
            }else{
                $sql="INSERT INTO owner (national_id , name , phone , email , license , pass) VALUES
                ('$_POST[NID]' , '$_POST[name]' , '$_POST[phone]' , '$_POST[email]' , '$_POST[license]' , '$_POST[pass]' )";
                $result=mysqli_query($conn , $sql) or die(mysqli_error($conn));
                //header("location:welcome1.php");
                echo "<script>alert('You are registered successfully');</script>";
            }
        }
    }

```

Figure 6.23: Register as owner PHP code

6.2.2.3 Contact

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. in the middle of the page, there contact form. In the bottom of the page, there is a map shows the current covered area in the site and there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.24 and 6.25.

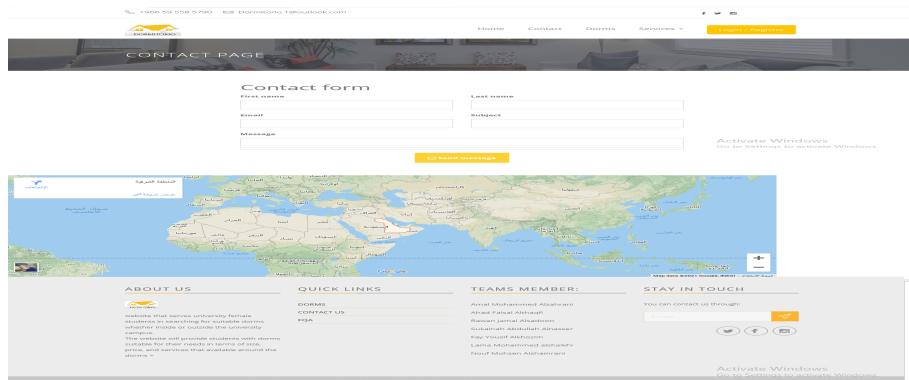


Figure 6.24: Contact-Owner

```

<?php

if(isset($_SESSION['user_id']) && $_SESSION['user_type'] == "owner"){

    echo'

        <li><a href="user-properties.php">your dorms</a> </li>
        <li><a href="My_dorms_bookings.php">My Dorms Bookings</a> </li>
        <li><a href="insert.php">insert dorms</a> </li>

    ';
}

?>

```

Figure 6.25: Contact PHP code

6.2.2.4 FQA

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. In the middle of the page, there is some of the most frequent questions and the answers. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.26 and 6.27.

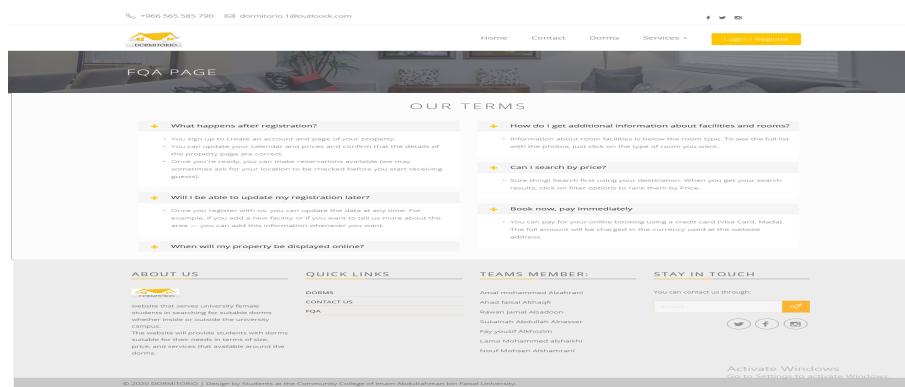


Figure 6.26: FQA-Owner

```

<div class="panel-body">
<ul>
<li> Information about room facilities is below the room type. To see the full list with the photos, just click on the type of room you want.</li>
</ul>
</div>

```

Figure 6.27: FQA code

6.2.2.5 Insert dorm

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. In the middle, there is insert form with four taps to fill the dorm information. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.28, 6.29, 6.30, 6.31, 6.32, and 6.33.

The screenshot shows a web form titled "INSERT YOUR DORM" with a progress bar at the top indicating "STEP 1", "STEP 2", "STEP 3", and "FINISHED". The main section is titled "Let's start with the basic information (with validation)". It contains several input fields:

- Dorm name proposal: [input field]
- Dorm price proposal: [input field]
- Telephone (only if you want to use default phone number): [input field]
- Built in: [input field]

A large gray rectangular area with the text "Click here" is positioned below the input fields. At the bottom right of the form is a yellow "NEXT" button.

Figure 6.28: Insert Dorm Part1

The figure consists of three vertically stacked screenshots of a web application titled "INSERT YOUR DORM".

- Step 1:** A form titled "Give us some images and videos?". It has two input fields: "Dorm image" and "Dorm video", both with "Browse..." buttons. Below the fields are checkboxes for "Select multiple images for your property" and "Select multiple videos for your property".
- Step 2:** A summary page titled "Finished and submit". It shows the user's name "Amal mohammed Alsharif" and a checkbox labeled "Accept terms and conditions".
- Step 3:** A summary page titled "About Us" and "Quick Links". It lists "TEAMS MEMBER:" with names like Amal mohammed Alsharif, Ahmad Faisal Alraifi, and others. It also includes links for "ABOUT US", "QUICK LINKS", and "STAY IN TOUCH".

Figure 6.29: Insert Dorm Part2

```
<?php
    require("config/connection.php");
    session_start();

    if(isset($_POST['finish'])){
        if(isset($_SESSION['user_id'])){

            $name = $_POST["name"];
            $status = $_POST["status"];
            $price = $_POST["price"];
            $phone = $_POST["phone"];
            $description = $_POST["description"];
            $built_in = $_POST["built_in"];
            $city = $_POST["city"];
            $rent_type = $_POST["rent_type"];
            $pay_type = $_POST["pay_type"];

            $min_bed = ($_POST["min_bed"] != "") ? $_POST["min_bed"] : 0;
            $min_bath = ($_POST["min_bath"] != "") ? $_POST["min_bath"] : 0;
            $min_geo = ($_POST["min_geo"] != "") ? $_POST["min_geo"] : 0;
            $min_garage = ($_POST["min_garage"] != "") ? $_POST["min_garage"] : 0;

            $AC = ($isset($_POST["AC"])) ? 1 : 0;
            $fire_alarm = ($isset($_POST["fire_alarm"])) ? 1 : 0;
            $monitoring_camera = ($isset($_POST["monitoring_camera"])) ? 1 : 0;
            $kitchen = ($isset($_POST["monitoring_camera"])) ? 1 : 0;
            $guard = ($isset($_POST["guard"])) ? 1 : 0;
            $oven = ($isset($_POST["oven"])) ? 1 : 0;
            $parking = ($isset($_POST["parking"])) ? 1 : 0;
            $has_lock = ($isset($_POST["has_lock"])) ? 1 : 0;
            $microwave = ($isset($_POST["microwave"])) ? 1 : 0;
            $elevator = ($isset($_POST["elevator"])) ? 1 : 0;
            $mailbox = ($isset($_POST["mailbox"])) ? 1 : 0;
            $waterfront = ($isset($_POST["waterfront"])) ? 1 : 0;
```

Figure 6.30: Insert dorm PHP code 1

```
$fire_alarm = ($isset($_POST["fire_alarm"])) ? 1 : 0;
$monitoring_camera = ($isset($_POST["monitoring_camera"])) ? 1 : 0;
$kitchen = ($isset($_POST["monitoring_camera"])) ? 1 : 0;
$guard = ($isset($_POST["guard"])) ? 1 : 0;
$oven = ($isset($_POST["oven"])) ? 1 : 0;
$parking = ($isset($_POST["parking"])) ? 1 : 0;
$has_lock = ($isset($_POST["has_lock"])) ? 1 : 0;
$microwave = ($isset($_POST["microwave"])) ? 1 : 0;
$elevator = ($isset($_POST["elevator"])) ? 1 : 0;
$mailbox = ($isset($_POST["mailbox"])) ? 1 : 0;
$waterfront = ($isset($_POST["waterfront"])) ? 1 : 0;

$video_1 = $_POST["video_1"];
$video_2 = $_POST["video_2"];
$video_3 = $_POST["video_3"];
$owner_id = $_SESSION['user_id'];

$new_dorm_id;
if($_FILES['picture']['name'] != ""){

    $picture = $_FILES['picture']['name'];
    $picture_tmp = $_FILES['picture']['tmp_name'];
    move_uploaded_file($picture_tmp, "images/".$picture);

    $sql = "INSERT INTO dorm (name , status , price , phone , picture , description , built_in , city , rent_type , pay_type , min_bed , min_bath , min_geo , min_garage , AC , fire_alarm , monitoring_camera , kitchen , guard , oven , parking , has_lock , microwave , elevator , mailbox , waterfront , video_1 , video_2 , video_3 , owner_id) VALUES ('$name' , '$status' , '$price' , '$phone' , '$picture' , '$description' , '$built_in' , '$city' , '$rent_type' , '$pay_type' , '$min_bed' , '$min_bath' , '$min_geo' , '$min_garage' , '$AC' , '$fire_alarm' , '$monitoring_camera' , '$kitchen' , '$guard' , '$oven' , '$parking' , '$has_lock' , '$microwave' , '$elevator' , '$mailbox' , '$waterfront' , '$video_1' , '$video_2' , '$video_3' , '$owner_id')";
```

Figure 6.31: Insert dorm PHP code 2

```

$SQL = "INSERT INTO dorm (name , status , price , phone , picture , description , built_in , city ,
rent_type , pay_type , min_bed , min_bath , min_geo , min_garage , A_C , fire_alarm , monitoring_camera ,
kitchen , guard , oven , parking , has_lock , microwave , elevator , mailbox , waterfront , vedio_1 ,
vedio_2 , vedio_3 , owner_id) VALUES ('$name' , '$status' , '$price' , '$phone' , '$picture' ,
'$description' , '$built_in' , '$city' , '$rent_type' , '$pay_type' , '$min_bed' , '$min_bath' ,
'$min_geo' , '$min_garage' , '$A_C' , '$fire_alarm' , '$monitoring_camera' , '$kitchen' , '$guard' ,
'$oven' , '$parking' , '$has_lock' , '$microwave' , '$elevator' , '$mailbox' , '$waterfront' ,
'$vedio_1' , '$vedio_2' , '$vedio_3' , '$owner_id')";

else{

$SQL = "INSERT INTO dorm (name , status , price , phone , description , built_in , city , rent_type ,
pay_type , min_bed , min_bath , min_geo , min_garage , A_C , fire_alarm , monitoring_camera , kitchen ,
guard , oven , parking , has_lock , microwave , elevator , mailbox , waterfront , vedio_1 , vedio_2 ,
vedio_3 , owner_id) VALUES ('$name' , '$status' , '$price' , '$phone' , '$description' , '$built_in' ,
'$city' , '$rent_type' , '$pay_type' , '$min_bed' , '$min_bath' , '$min_geo' , '$min_garage' , '$A_C' ,
'$fire_alarm' , '$monitoring_camera' , '$kitchen' , '$guard' , '$oven' , '$parking' , '$has_lock' ,
'$microwave' , '$elevator' , '$mailbox' , '$waterfront' , '$vedio_1' , '$vedio_2' , '$vedio_3' ,
'$owner_id')";

}

$result=mysqli_query($conn , $SQL) or die(mysqli_error($conn));
//header('location:welcome!.php');


```

Figure 6.32: Insert dorm PHP code 3

```

if(!mysqli_error($conn)){
echo "<script>alert('dorm inserted successfully');</script>";
if($_FILES['dorm_images']['name'] != ""){
$new_dorm_id = "select max(id) as new_id from dorm";
$result=mysqli_query($conn , $new_dorm_id) or die(mysqli_error($conn));
$row = mysqli_fetch_array($result);

$images_count = count($_FILES['dorm_images']['name']);
for($i=0; $i <$images_count; $i++){
$image = $_FILES['dorm_images']['name'][$i];
$image_tmp = $_FILES['dorm_images']['tmp_name'][$i];
move_uploaded_file($image_tmp,"images/".$image."");

$SQL = "insert into dorm_images (image , dorm_id) values ('$image' , '$row[new_id]')";
$res=mysqli_query($conn , $SQL) or die(mysqli_error($conn));
}
}
else{
echo "<script>alert(\".mysqli_error($conn).\" );</script>";
}
}
else{
echo "<script>alert('you are don't Logined , login to insert new dorms !!!');</script>";
}
}


```

Figure 6.33: Insert dorm PHP code 4

6.2.2.6 Your dorm

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms ,Services, and button for log out. In addition, there is a list of the owner dorms with dorms information and three option (edit, delete, and view).In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email shown in Figures 6.34 and 6.35.

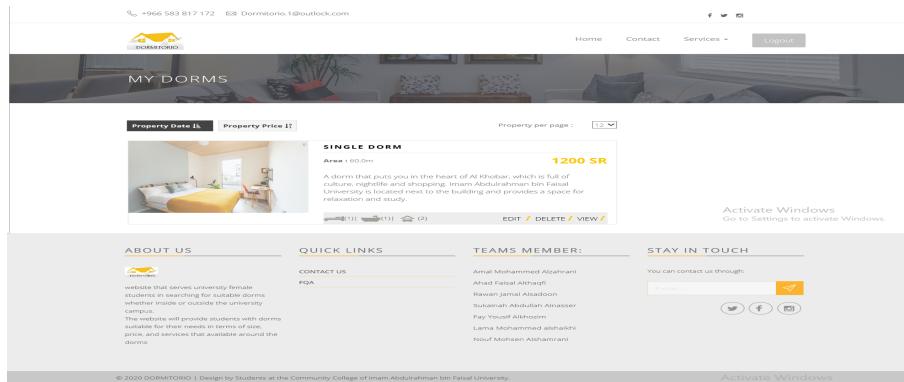


Figure 6.34: Your Dorm

```
<h2>User account</h2>
<ul>
<?php

if(isset($_SESSION['user_id']) && $_SESSION['user_type'] =
"owner") {

    echo '

<li><a href="user-properties.php">your dorms</a>  </li>
<li><a href="My_dorms_bookings.php">My Dorms Bookings</a>
</li>
<li><a href="insert.php">insert dorms</a>  </li>
';

}
?>
```

Figure 6.35: Your dorm PHP code

6.2.2.7 My dorms booking

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home ,Contact ,Dorms ,Services, and button for log out. Then there is list with booking information and the customer information. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.36, 6.37, and 6.38.

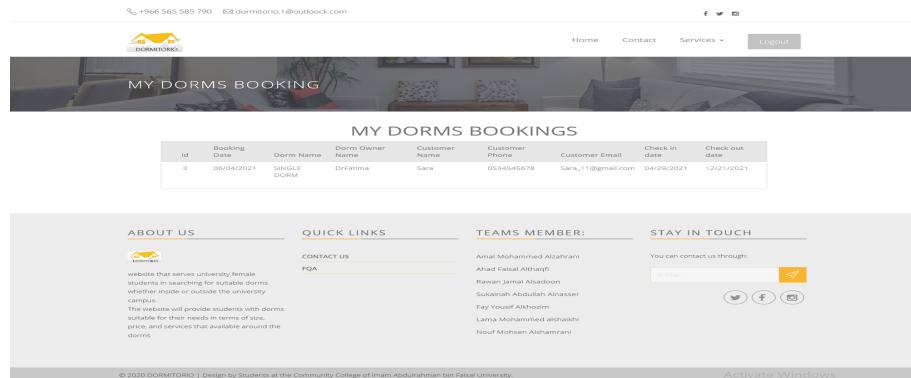


Figure 6.36: My Booking Dorms

```
<?php
if(isset($_SESSION['user_id'])){
    echo'
        <center>
        <div class="table-responsive patients_Table" style="width:100%; border:1px solid #ddd;">
            <table id="myTable" class="table table-hover">
                <thead style="background-color: #e6e6e6;">
                    <tr>
                        <th style="width:80px; text-align:center;">Id</th>
                        <th>Booking Date</th>
                        <th>Dorm Name</th>
                        <th>Owner Name</th>
                        <th>Customer Name</th>
                        <th>Customer Phone</th>
                        <th>Customer Email</th>
                        <th>Check in date</th>
                        <th>Check out date</th>
                    </tr>
                </thead>
                <tbody id="Accounts_table">
                    ';
                    $query = "select booking.id , booking.booking_date , dorm.name as dorm_name , owner.name as owner_name , booking.customer_name , booking.customer_phone , booking.customer_email , booking.check_in_date , booking.check_out_date , booking.dorm_id from owner , dorm , booking WHERE owner.id = dorm.owner_id and dorm.id = booking.dorm_id and owner.id = '".$_SESSION['user_id']."'";
```

Figure 6.37: My dorm booking PHP code 1

```
        from owner, dorm, booking WHERE owner.id = dorm.owner_id and
        dorm.id = booking.dorm_id and owner.id = ". $_SESSION['user_id'] .
        $result = mysqli_query($conn , $query);
        while($row = mysqli_fetch_array($result))
        {
            echo'
                <tr>
                    <td style="text-align:center;">'.$row['id'].'</td>
                    <td>'.$row['booking_date'].'</td>
                    <td>'.$row['dorm_name'].'</td>
                    <td>'.$row['owner_name'].'</td>
                    <td>'.$row['customer_name'].'</td>
                    <td>'.$row['customer_phone'].'</td>
                    <td>'.$row['customer_email'].'</td>
                    <td>'.$row['check_in_date'].'</td>
                    <td>'.$row['check_out_date'].'</td>
                </tr>
            ';
        }
        echo'
    </tbody>
</table>
';
}

```

Figure 6.38: My dorm booking PHP code 2

6.2.2.8 Your profile

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. The owner information is displayed with an ability to change and modify information. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.39, 6.40, 6.41, and 6.42.

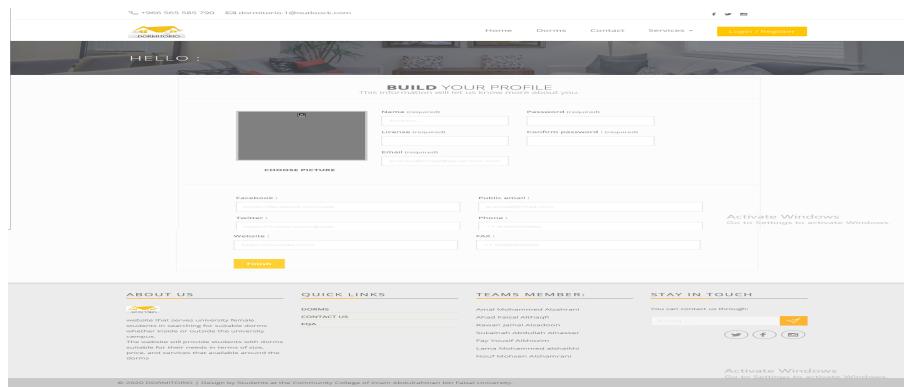


Figure 6.39: Your Profile-Owner

```
<?php
require("config/connection.php");
session_start();

if(isset($_REQUEST['show_profile']) && isset($_SESSION['user_id'])){

    if(isset($_SESSION["user_type"]) && $_SESSION["user_type"]=="owner"){

        $query = "select * from owner where id = ".$_SESSION['user_id']."";
    }
    else if(isset($_SESSION["user_type"]) && $_SESSION["user_type"]=="renter"){

        $query = "select * from renter where id = ".$_SESSION['user_id']."";
    }

    $result = mysqli_query($conn , $query);
    $profile_data = mysqli_fetch_array($result);

}

if(isset($_POST['finish'])){

    if($_FILES['personal_image_file']['name'] != ""){

        $image = $_FILES['personal_image_file']['name'];
        $image_tmp = $_FILES['personal_image_file']['tmp_name'];
        move_uploaded_file($image_tmp,"images/".$image,"");
    }
}
```

Figure 6.40: Your profile PHP code 1

```

if(isset($_SESSION["user_type"])) && $_SESSION["user_type"]=="owner") {
    $query = "update owner set name = '$_POST[name]', email = '$_POST[email]', phone = '$_POST[phone]',
    picture = '$image', facebook = '$_POST[facebook]', public_email = '$_POST[p_email]', twitter =
    '$_POST[twitter]', website = '$_POST[website]', fax = '$_POST[fax]', license = '$_POST[license]' , pass
    = '$_POST[pass]' where id = '$_SESSION[user_id]'";
}
else if(isset($_SESSION["user_type"])) && $_SESSION["user_type"]=="renter"){
    $query = "update renter set name = '$_POST[name]', email = '$_POST[email]', phone = '$_POST[phone]',
    picture = '$image', facebook = '$_POST[facebook]', public_email = '$_POST[p_email]', twitter =
    '$_POST[twitter]', website = '$_POST[website]', fax = '$_POST[fax]', license = '$_POST[license]' , pass
    = '$_POST[pass]' where id = '$_SESSION[user_id]'";
}
$result = mysqli_query($conn , $query);
echo "<script>alert('Your information has been updated.');//</script>";
echo "<script>window.location.href='user-profile.php?show_profile=true';</script>";
}
else{
    if(isset($_SESSION["user_type"])) && $_SESSION["user_type"]=="owner") {
        $query = "update owner set name = '$_POST[name]', email = '$_POST[email]', phone = '$_POST[phone]',
        facebook = '$_POST[facebook]', public_email = '$_POST[p_email]', twitter = '$_POST[twitter]', website =
        '$_POST[website]', fax = '$_POST[fax]', license = '$_POST[license]' , pass = '$_POST[pass]' where id =
        '$_SESSION[user_id]'";
    }
    else if(isset($_SESSION["user_type"])) && $_SESSION["user_type"]=="renter"){
        $query = "update renter set name = '$_POST[name]', email = '$_POST[email]', phone = '$_POST[phone]',
        facebook = '$_POST[facebook]', public_email = '$_POST[p_email]', twitter = '$_POST[twitter]', website =
        '$_POST[website]', fax = '$_POST[fax]', license = '$_POST[license]' , pass = '$_POST[pass]' where id =
        '$_SESSION[user_id]'";
    }
    $result = mysqli_query($conn , $query);
    echo "<script>alert('Profile Updated Successfully');//</script>";
    echo "<script>window.location.href='user-profile.php?show_profile=true';</script>";
}
}
?>

```

Figure 6.41: Your profile PHP code 2

```

$Query = "update owner set name = '$_POST[name]', email = '$_POST[email]', phone = '$_POST[phone]',
facebook = '$_POST[facebook]', public_email = '$_POST[p_email]', twitter = '$_POST[twitter]', website =
'$_POST[website]', fax = '$_POST[fax]', license = '$_POST[license]' , pass = '$_POST[pass]' where id =
'$_SESSION[user_id]'";
}
else if(isset($_SESSION["user_type"])) && $_SESSION["user_type"]=="renter"){
    $Query = "update renter set name = '$_POST[name]', email = '$_POST[email]', phone = '$_POST[phone]',
    facebook = '$_POST[facebook]', public_email = '$_POST[p_email]', twitter = '$_POST[twitter]', website =
    '$_POST[website]', fax = '$_POST[fax]', license = '$_POST[license]' , pass = '$_POST[pass]' where id =
    '$_SESSION[user_id]'";
}
$result = mysqli_query($conn , $Query);
echo "<script>alert('Profile Updated Successfully');//</script>";
echo "<script>window.location.href='user-profile.php?show_profile=true';</script>";
}
}
?>

```

Figure 6.42: Your profile PHP code 3

6.2.3 Renter

6.2.3.1 Home Page

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. Also we put in the middle of the page a search bar to enable the guest to search by collage or city. In the bottom of the page, we put a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.43

and 6.44.

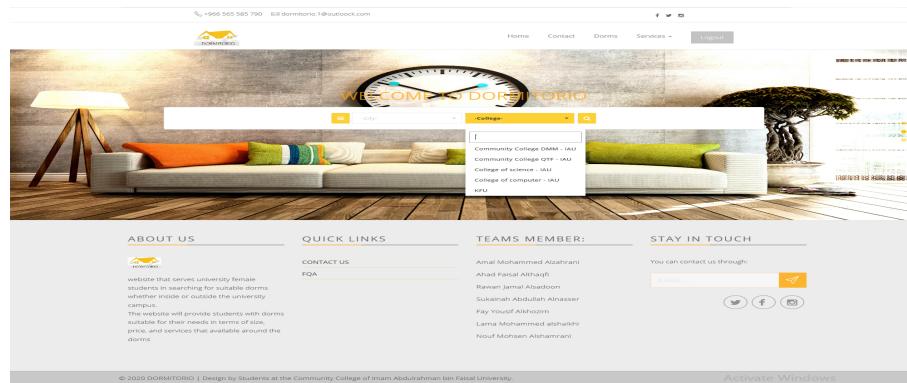


Figure 6.43: Home Page-Renter

```
<li><a href="register.php">Register | login</a>  </li>
<?php

    if(isset($_SESSION['user_id'])) && $_SESSION['user_type'] =
    "renter"){

    }

?>
```

Figure 6.44: Renter home page PHP code

6.2.3.2 Register

In the top of the page, there is navigation bar with one option (Home) which allows the user to go to the home page. In the left of the page, there is register form; and in the right of the page, there is log in form. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown Figures 6.45 and 6.46.

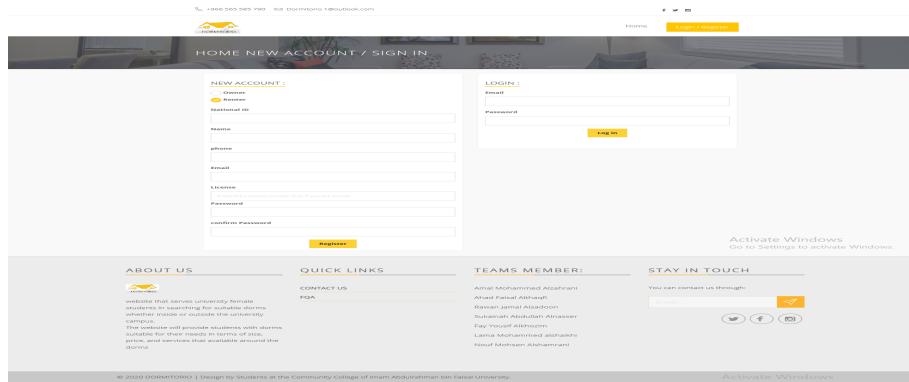


Figure 6.45: Register-Renter

```

if ($_POST['user_type']=='renter')
{
    if(empty($_POST['NID']) || empty($_POST['name']) || empty($_POST['phone']) || empty($_POST['email']) || empty($_POST['pass']))
    {
        //header("location:index.php?error=empty");
        echo "<script>alert('Please fill all the fields');</script>";
    }
    else if($_POST['conpass'] != $_POST['pass'])
    {
        echo "<script> alert('Passwords don't match');</script>";
    }
    else
    {
        $sql="INSERT INTO renter (national_id , name ,phone, email , license , pass) VALUES
        ('$_POST[NID]' ,'_POST[name]' ,'_POST[phone]' ,'_POST[email]' ,'_POST[license]' ,'_POST[pass]' )";
        $result=mysqli_query($conn , $sql) or die(mysqli_error($conn));
        //header('location:welcomem1.php');
        echo "<script>alert('You are registered successfully');</script>";
    }
}
}

```

Figure 6.46: Register as renter PHP code

6.2.3.3 Contact

At the beginning of the page you will see the navigation bar for all the services in the site such as , Home , Contact , Dorms ,services and button for log out. in the middle of the page there contact form. in the bottom of the page there is a map show the current covered area in the site and there is brief description (about us) , quick link , team member Names and quick contact with us via email shown in 6.47 and 6.48 .

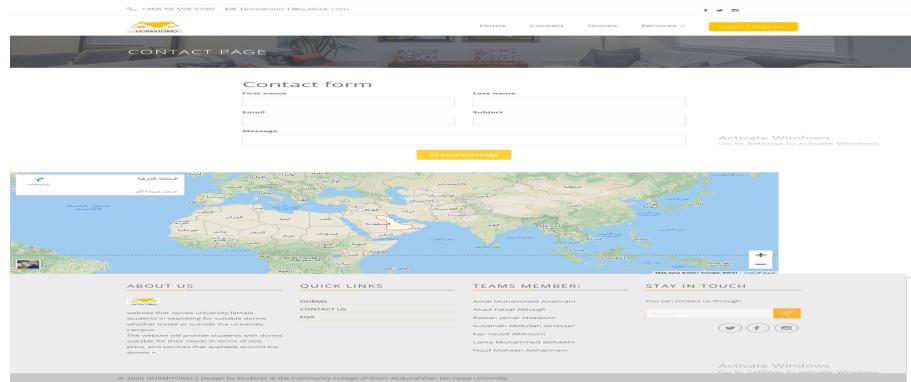


Figure 6.47: Contact-Renter

```
<?php
if(isset($_SESSION["user_id"])){
    echo '<button class="navbar-btn nav-button wow bounceInRight logout"
        onclick="window.open(\'logout.php\')" data-wow-delay="0.45s">Logout</button>';
}
else{
    echo '<button class="navbar-btn nav-button wow bounceInRight login"
        onclick="window.open(\'register.php\')" data-wow-delay="0.45s">Login /
    Register</button>';
}
?>
```

Figure 6.48: Contact PHP code

6.2.3.4 FQA

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. In the middle of the page, there is some of the most frequent questions and the answers. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.49 and 6.50.

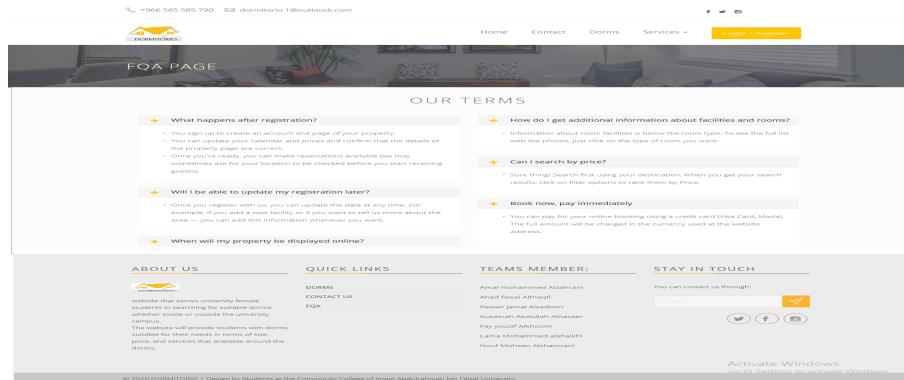


Figure 6.49: FQA-Renter

```
<div class="panel-body">
<ul>
<li> Information about room facilities is below the room type. To see the full list with the photos, just click on the type of room you want.</li>
</ul>
</div>
```

Figure 6.50: FQA code

6.2.3.5 Dorm List

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. Then in the top, there is two tools to filter the dorms. Then the dorm will display with a brief description. In the left side of the page, there is a smart search to filter the dorms. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact with us via email as shown in Figures 6.51 and 6.52.

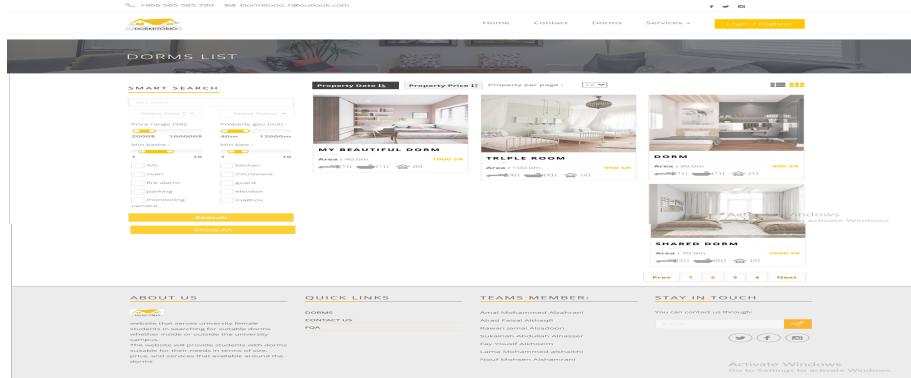


Figure 6.51: Dorm List

```
<?php
$result = mysqli_query($conn , $query);
while($dorm_data = mysqli_fetch_array($result)){
echo '
<div class="col-sm-6 col-md-4 p0">
<div class="box-two property-item">
<div class="item-thumb">
<a href="property_details.php?view=true&dorm_id='.$dorm_data['id'].'">
</a>
</div>
<div class="item-entry overflow">
<h3><a href="property_details.php?view=true&dorm_id='.$dorm_data['name'].'">'</a></h3>
<div class="dot-hr"></div>
<span class="pull-left"><b>Area :</b> '.$dorm_data['min_geo'].' m<br/><span class="property-price pull-right">' . $dorm_data['price'] . ' SR</span>
<p style="display: none;">' . $dorm_data['description'] . '</p>
<div class="property-icon">
 (' . $dorm_data['min_bed'] . ')
 (' . $dorm_data['min_bath'] . ')
 (' . $dorm_data['min_garage'] . ')
</div>
</div>
</div>
');
}
?>
```

Figure 6.52: Dorm list PHP code

6.2.3.6 Dorm Details

At the beginning of the page, you will see the navigation bar for all the services in the site such as Home, Contact, Dorms, Services, and button for log out. In the right side of the page, there is the owner information and Smart search tools. In the left side, there is the dorm pictures; and under the images, there are the features of the dorm with dorm price then there is a description of the dorm and additional details then you will see a button of booking. In the bottom of the page, there is a brief description (about us), quick link, team member names, and quick contact

with us via email as shown in Figures 6.53 and 6.54.

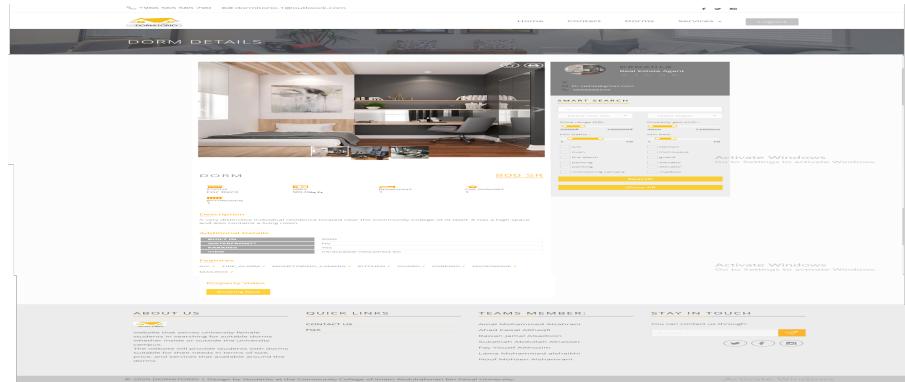


Figure 6.53: Dorm Details

```
1 <?php
2     $sq = "select * from dorm_images where dorm_id = '".$_GET['dorm_id']."'"; 
3     $res = mysqli_query($conn, $sq);
4     while($dorm_images = mysqli_fetch_array($res)){
5         echo '
6             <li data-thumb="images/'.$dorm_images['image'].'">'.
7                 '
8             </li>
9         ';
10    }
11    ?>
12    </ul>
13  </div>
14  </div>
15 </div>
16
17
18
19 <div class="single-property-wrapper">
20   <div class="single-property-header">
21     <h1 class="property-title pull-left"><?php echo $dorm_data['name']; ?></h1>
22     <span class="property-price pull-right"><?php echo $dorm_data['price']; ?> SR </span>
23   </div>
```

Figure 6.54: Dorm details PHP code

6.2.3.7 Booking

Booking page displays a small form for booking contain arrival and departure date and user information with two buttons: one for submit and the other one for reset as shown in Figures 6.55, 6.56, and 6.57.

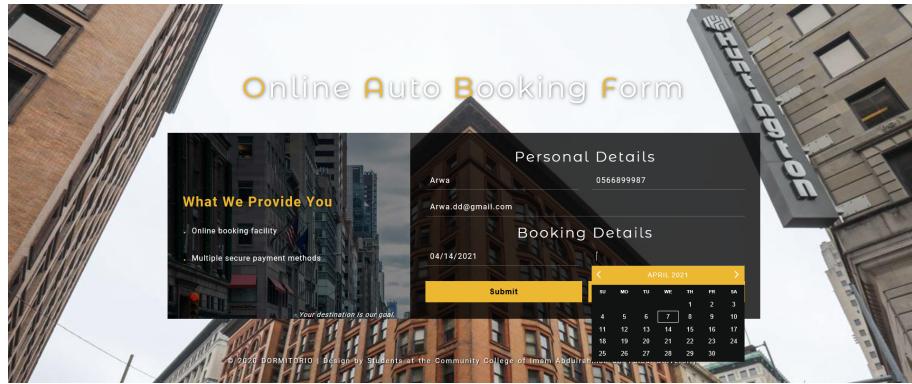


Figure 6.55: Booking

```
<?php
if(isset($_SESSION['user_id'])){
    echo'
        <center>
            <div class="table-responsive patients_Table" style="width:100%; border:1px solid #ddd;">
                <table id="myTable" class="table table-hover">
                    <thead style="background-color: #e6e6e6;">
                        <tr>
                            <th style="width:80px; text-align:center;">Id</th>
                            <th>Booking Date</th>
                            <th>Dorm Name</th>
                            <th>Dorm Owner Name</th>
                            <th>Customer Name</th>
                            <th>Customer Phone</th>
                            <th>Customer Email</th>
                            <th>Check in date</th>
                            <th>Check out date</th>
                        </tr>
                    </thead>
                    <tbody id="Accounts_table">
                        ';
                        $query = "select booking.id, booking.booking_date , dorm.name as dorm_name ,
                                owner.name as owner_name , booking.customer_name , booking.customer_phone ,
                                booking.customer_email , booking.check_in_date , booking.check_out_date ,
                                booking.dorm_id from owner, dorm, booking WHERE owner.id = dorm.owner_id and
                                dorm.id = booking.dorm_id";
                        $result = mysqli_query($conn , $query);
                        while($row = mysqli_fetch_array($result))
                        (

```

Figure 6.56: Booking PHP code 1

```
echo'
        <tr>
            <td style="text-align:center;">'.$row['id'].'</td>
            <td>'.$row['booking_date'].'</td>
            <td>'.$row['dorm_name'].'</td>
            <td>'.$row['owner_name'].'</td>
            <td>'.$row['customer_name'].'</td>
            <td>'.$row['customer_phone'].'</td>
            <td>'.$row['customer_email'].'</td>
            <td>'.$row['check_in_date'].'</td>
            <td>'.$row['check_out_date'].'</td>
        </tr>
    ';
}
echo'
</tbody>
</table>
</div>
</center>
';
}
?>
```

Figure 6.57: Booking PHP code 2

6.2.4 Admin

6.2.4.1 Home Page

At the beginning of the page you will see the navigation bar for two services that belong to the admin, customer booking, dorms registration requests and button for log out. At the bottom of the page we put a team members names and quick contact with us via email shown in 6.58 and 6.59 .

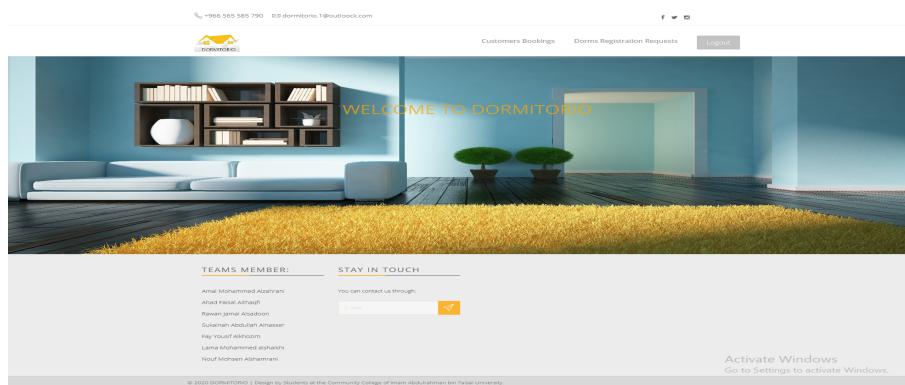


Figure 6.58: Home Page-Admin

```
<?php
    if(isset($_SESSION["user_id"])){
        echo '<button class="navbar-btn nav-button wow bounceInRight logout"
            onclick="window.open(\'logout.php\')" data-wow-delay="0.45s">Logout</button>';
    }
    else{
        echo '<button class="navbar-btn nav-button wow bounceInRight login"
            onclick="window.open(\'register.php\')" data-wow-delay="0.45s">Login /
            Register</button>';
    }
?>
</div>
```

Figure 6.59: Admin homepage

6.2.4.2 Customer Booking Page

At the beginning of the page, you will see the navigation bar for two services that belong to the admin, customer booking, dorms registration requests, and button for log out. At the middle of the page, you will see the details of customer booking requests. At the bottom of the page, we put a team members names and quick contact with us via email as shown in Figures 6.60, 6.61, and 6.62.

The screenshot shows a web application interface for 'DORMITORIO'. At the top, there is a header with a phone number (+966 565 585 790), a logo, and links for 'Customers Bookings', 'Dorms Registration Requests', and 'Logout'. Below the header is a section titled 'CUSTOMERS BOOKINGS' containing a table with two rows of booking data:

ID	Booking Date	Dorm Name	Dorm Owner Name	Customer Name	Customer Phone	Customer Email	Check in date	Check out date
4	06/04/2021	NEW DORM	Abdullah	Arwa	0566899987	Arwa.ds@gmail.com	05/09/2021	01/09/2022
3	06/04/2021	SINGLE DORM	DrFatima	Sara	0534545678	Sara_11@gmail.com	04/29/2021	12/21/2021

Below the bookings table is a 'TEAMS MEMBER:' section listing team members: Amal Mohammed Alzahrani, Ahad Faisal Althaqfi, Rawan Jemal Alhadron, Sukeinah Abdullah Alnasser, Fay Yousef Alkhoom, Laima Mohammed alshaikhi, and Nour Mohsen Alshamrani. To the right of this section is a 'STAY IN TOUCH' section with a 'Contact Us' form and social media icons for Facebook, Twitter, and YouTube. At the bottom right, there is a link to 'Activate Windows'.

Figure 6.60: Customer Booking Page

```
<?php
if(isset($_SESSION['user_id'])) {
    echo'
        <center>
        <div class="table-responsive patients_Table" style="width:100%; border:1px solid #ddd;">
            <table id="myTable" class="table table-hover">
                <thead style="background-color: #e6e6e6;">
                    <tr>
                        <th style="width:0px; text-align:center;">Id</th>
                        <th>Booking Date</th>
                        <th>Dorm Name</th>
                        <th>Dorm Owner Name</th>
                        <th>Customer Name</th>
                        <th>Customer Phone</th>
                        <th>Customer Email</th>
                        <th>Check in date</th>
                        <th>Check out date</th>
                    </tr>
                </thead>
                <tbody id="Accounts_table">
                    ';
                    $query = "select booking.id, booking.booking_date , dorm.name as dorm_name , owner.name as owner_name , booking.customer_name , booking.customer_phone , booking.booking_id , booking.booking_check_in_date , booking.booking_check_out_date , booking.dorm_id from owner, dorm, booking WHERE owner.id = dorm.owner_id and dorm.id = booking.dorm_id";
                    $result = mysqli_query($conn , $query);
                    while($row = mysqli_fetch_array($result))
                    {

```

Figure 6.61: Customer booking code 1

```

echo'
<tr>
    <td style="text-align:center;">'.$row['id'].'</td>
    <td>'.$row['booking_date'].'</td>
    <td>'.$row['dorm_name'].'</td>
    <td>'.$row['owner_name'].'</td>
    <td>'.$row['customer_name'].'</td>
    <td>'.$row['customer_phone'].'</td>
    <td>'.$row['customer_email'].'</td>
    <td>'.$row['check_in_date'].'</td>
    <td>'.$row['check_out_date'].'</td>

```

Figure 6.62: Customer booking PHP code 2

6.2.4.3 Dorms Registration Requests

At the beginning of the page, you will see the navigation bar for two services that belong to the admin, customer booking, dorms registration requests, and button for log out. At the middle of the page, you will see the dorms insertion requests and button for admin approval if it is accept or reject. At the bottom of the page, we put a team members names and quick contact with us via email as shown in Figures 6.63 , 6.64 and 6.65 .

The screenshot shows a web application interface for managing dorm registration requests. At the top, there's a navigation bar with links for 'Customers Bookings', 'Dorms Registration Requests', and 'Logout'. Below the navigation is a section titled 'DORMS REGISTRATION REQUESTS' containing a table with the following data:

ID	Date	Dorm Name	Dorm Status	Owner Name	Price	Description	Built In	City	Accepted	Operation
20	07/07/2021	MY BEAUTIFUL DORM	Rent	Drhahha	1000	A dorm room overlooking the park, only 5 minutes away from the College of Science. Do not miss the opportunity and book with us!	2021	Dammam	<input checked="" type="checkbox"/>	<button>Accept</button>
17	NEW DORM	For Rent	Akisullah	700	A modern dorm characterized by attractive colors, located near King Fahd University. It has a great view and provides a space for relaxation.	2020	Al-Jessa	<input checked="" type="checkbox"/>	<button>Accept</button>	
18	TRIPLE ROOM	For Rent	Akisullah	950	A very elegant room with its colors and furniture that you can share with your friends at an affordable price, located in Jeddah.	2021	Jeddah	<input checked="" type="checkbox"/>	<button>Accept</button>	
19	DORM	For Rent	Drhahha	800	A large dorm room located in the center of the Community College of Al-Qatif. It has a great view and also contains a swimming pool.	2020	Qatif	<input checked="" type="checkbox"/>	<button>Accept</button>	
15	SINGLE DORM	For Rent	Drhahha	1,200	A dorm that puts you in the heart of Al-Jubbar, which is full of greenery and nature. It is located next to the building and provides a space for relaxation.	2021	Khobar	<input checked="" type="checkbox"/>	<button>Accept</button>	
15	SINGLE DORM	For Rent	Drhahha	1,200	A dorm that puts you in the heart of Al-Jubbar, which is full of greenery and nature. It is located next to the building and provides a space for relaxation.	2021	Khobar	<input checked="" type="checkbox"/>	<button>Accept</button>	
16	SHARED DORM	For Rent	Drhahha	2000	This room is located in Dammam. You can share this room with your friend. How wonderful your friend is to study and also to live with you.	2019	Dammam	<input checked="" type="checkbox"/>	<button>Accept</button>	

Below the table, there are sections for 'TEAMS MEMBER:' and 'STAY IN TOUCH'. The 'TEAMS MEMBER:' section lists team members: Amal Muhammed Alzahrani, Ahmed Faisal Alshayghi, Rawan Jamal Alabdullah, Salwa Alabdullah Alghaser, Fay Yousef Alkhoomi, Laima Mohammad Alshabsheh, and Neof Momen Alshabani. The 'STAY IN TOUCH' section includes a contact form and social media links for Facebook, Twitter, and LinkedIn. At the bottom, there are links for 'Activate Windows' and 'Go to Settings to activate Windows'.

Figure 6.63: Dorms Registration Requests

```

<?php
if(isset($_SESSION['user_id'])){
echo'
<center>
<div class="table-responsive patients_Table" style="width:100%; border:1px solid #ddd;">
<table id="myTable" class="table table-hover">
<thead style="background-color: #e6e6fa;">
<tr>
<th style="width:80px; text-align:center;">Id</th>
<th>Date</th>
<th>Dorm Name</th>
<th>Dorm Status</th>
<th>Owner Name</th>
<th>Price</th>
<th>Description</th>
<th>Build In</th>
<th>City</th>
<th>Accepted</th>
<th>Operation</th>
</tr>
</thead>
<tbody id="Accounts_table">
',
$query = "select dorm.id , dorm.registration_date , dorm.name , dorm.status ,
owner.name as owner_name , dorm.price , dorm.description , dorm.built_in ,
dorm.city , dorm.accepted , dorm where owner.id = dorm.owner_id";
$result = mysqli_query($conn , $query);
while($row = mysqli_fetch_array($result))
{
echo'

```

Figure 6.64: Dorms Registration Requests PHP code 1

```

echo'
<tr>
<td style="text-align:center;">'. $row['id'] . '</td>
<td>'. $row['registration_date'] . '</td>
<td>'. $row['status'] . '</td>
<td>'. $row['owner_name'] . '</td>
<td>'. $row['price'] . '</td>
<td>'. $row['description'] . '</td>
<td>'. $row['built_in'] . '</td>
<td>'. $row['city'] . '</td>
<td style="text-align:center;"><input type="checkbox" name="accepted" value="0" /> ' . if($row['is_accepted'] == 1) {echo "checked"; } . echo ' />
</td>
<td>
<div class="dropdown" style="padding: 2px 8px 2px 8px; width: 126px;
height:50px; "
<button class="btn btn-primary dropdown-toggle operation_dropdown"
data-toggle="dropdown" data-expanded="false"
type="button">Operation</button>
<div class="dropdown-menu" role="menu" style="padding: 2px; ">
<a href="#" name="show_detail" class="dropdown-item" role="presentation"
 href="dorm_registration_requests.php?id=' . $row['id'] . '">
style="padding: 0px; border-bottom: 1px solid #eee; font-size: 14px; width: 100%;">Show Dorm Detail</a>
<a href="#" name="accept" class="dropdown-item" role="presentation"
 href="dorm_registration_requests.php?
operation=accept&id=' . $row['id'] . '" style="padding: 0px; border-bottom: 1px solid #eee; font-size: 14px; width: 100%;">Accept</a>
<a href="#" name="reject" class="dropdown-item" role="presentation"
 href="dorm_registration_requests.php?
operation=reject&id=' . $row['id'] . '" style="padding: 0px; border-bottom: 1px solid #eee; font-size: 14px; width: 100%;">Reject</a>
</div>
</td>

```

Figure 6.65: Dorms Registration Requests PHP code 2

Chapter 7

Conclusion and Future Work

In conclusion, we proposed, analyzed, designed, and implemented a website that facilitates providing student dorms. The system components including the database and web pages were implemented using PHP and MySQL. We tested our system using different scenarios and made sure most of the system requirements are met. In the following sections, we describe the system's weaknesses, strengths, and future work.

7.1 Overall Weaknesses

The first challenge that we faced was the lack of time with the many and complex features that we wanted to add to our site, such as the payment page, the use of two languages on the site (Arabic and English), as well as the addition of effective maps for the website. Nevertheless, we tried to cover the largest number possible requirements for the site to show it in the best way and certainly will be developed much more in the future.

7.2 Overall Strengths

One of the most important features of our website is the clarity of the pages, the ease of accessing them in addition to the attractive design of the pages, the clear, and beautiful presentation. We also tried to use different tools to display the site to make it suitable for all types of users. All interfaces are specially designed for each type of user in order to achieve the greatest possible benefit. In addition, storing the data in the database works perfectly and we are able to retrieve the needed information from the database efficiently.

7.3 Future Work

In the future, we hope that the scope of our project will expand to cover most countries of the world and benefit the largest possible number of students. We also aim to use the best and safest payment methods on our website and use more than one language on the site to benefit all its users. In the future, we also aim to use effective maps to display the locations of dorms and their proximity and distance from universities.

7.4 Useful Resources

Through our work on this project, we have learned from a lot of useful sources that we want to talk about in this section. We start with the first reference which is w3schools.com which is a website where we find different codes of different programming languages. It helped us figure out how to solve some of the problems we had while programming the site [15].

The second reference was flexcourses.com which is a website that offers paid

programming courses using HTML, CSS, JavaScript, PHP with a programmer with you and the courses have helped us to understand these programming languages and solve some of the site problems that we have encountered, taking experience from the programmer, and consulting him when facing difficulty [7].

Then, we use Stackoverflow.com in programming stage which is provide many codes in PHP, JAVASCRIPT, HTML, and CSS. We use it to help us in javascript code [14]. We also use Derek Banas Channel in YouTube.com in how to make query in MySQL data base [11]. To develop our skills in JavaScript, we use Web Dev Simplified Channel and FreeCodeCamp.org Channel in YouTube [9]. For PHP and MySQL, we use also The Bad Tutorials Channel and Ken Swartzout Channel in youtube to take some courses to develop our skills [1] [13]. For XAMPP, we use Ken Swartzout Channel and TECH WOLF channel in YouTube.com to learn more in how to use Xampp [2].

For writing our graduation report, we choose to use unique tool such as LaTeX, so we used LaTeXTutorial.com which is help us in using this tool by providing us some codes [12]. In addition, we use [3] research which provides us very useful information about \LaTeX and how we can use it in the best way. The other reference is The Math Student Channel in YouTube.com, which is help us in with guideline in how to insert images and how to create professional tables [10].

Bibliography

- [1] PHP & MySQL Tutorial 60: Executing a Query Using PHP. Php & mysql tutorial. <https://www.youtube.com/watch?v=WHw6vVoRTMs>, 2014.
- [2] XAMPP TUTORIAL FOR BEGINNERS. Xampp for beginners. <https://www.youtube.com/watch?v=sWJdeSMDDbQ>, 2020.
- [3] Florence Débarre. Introduction to latex. 2012.
- [4] Harvey M Deitel, Paul J Deitel, and Tem R Nieto. *Internet & world wide web: how to program*. Pearson/Prentice Hall, 2004.
- [5] Alan Dennis, Barbara Wixom, and David Tegarden. *Systems analysis and design: An object-oriented approach with UML*. John wiley & sons, 2015.
- [6] Ramez Elmasri and Shamkant Navathe. *Fundamentals of database systems sixth edition pearson education*. 2011.
- [7] flexcourses. CSS HTML web design. <https://www.flexcourses.com/courses/html-and-css/>, 2020. [Online; accessed 2021].
- [8] Jennifer Widom Hector Garcia-Molina, Jeffrey Ullman. *Database systems: The complete book*. Pearson, 2008.
- [9] JavaScript. Javascript promises in 10 minutes. <https://www.youtube.com/watch?v=DHvZLI7Db8E>, 2019.
- [10] captions LaTeX insert pictures and referencing. Latex. <https://www.youtube.com/watch?v=bQ8Gp1LSCmY>, 2012.
- [11] latex tutorial. learning latex. <https://www.youtube.com/watch?v=VhmkLr0jLsw>, 2019.

- [12] latex tutorial. latex-tutorial. <https://latex-tutorial.com/tutorials/tables/>, 2021. [Online; accessed 2021].
- [13] MySQ Using PHP. MySql select query using php. <https://www.youtube.com/watch?v=HonolrYBIQM>, 2017.
- [14] stackoverflow programming. stackoverflow php,javascript,html,css. <https://stackoverflow.com/>, 2020.
- [15] w3schools. HTML and CSS. <https://www.w3schools.com/>, 2020. [Online; accessed 2021].