DEPARTMENT OF COMPUTER SCIENCE

FACULTY OF SCIENCE

UNIVERSITY OF JAFFNA

Boarding Management System

Submitted in partial fulfillment of requirements for the course unit

CSC241SC2 2017

Team Members:

H.M.S.D.Herath - 2016/CSC/008
S.Lesslylavan - 2015/CSC/001
S.K.K.Anuka Sanjaya - 2016/CSC/021
Ajanthy Jayarajan - 2016/CSC/031
H.M.A.S.K.llangasinghe - 2016/CSC/040
R.Shehan Lahiru Jayasinghe - 2016/CSC/048

Group 1S

Lecturer: Mr.S.Suthakar

Assistant Lecturer : Mr.T.Laventhiran

Mentor: Miss. Tharuka Dilani

7 June 2019

Executive summary

The students of University of Jaffna which is situated at Sir.Pon Ramanathan Road, Thirunelvelly are facing many difficulties in finding the boarding houses in their second year and third year. The university welfare society is providing hostel facility for all faculty students in their first and final year.

Therefore, we decided to do boarding management system for this identified problem. Our project helps the students to search the boarding houses via website and it helps the owners of the houses to advertise their boarding houses.

Admin can view all owners' details and other admins' details separately, can remove a particular owner and his boarding in any circumstances, he/ she is able to block or unblock the user comments, review the complaints and remove the admin.

Owners can add his/ her own boarding houses, can remove & update their own boarding house, and deactivate their account.

Our system provides a website to the students to obtain the information about boarding details which are provided by the owners.

Actors: Student, Admin, Owner

Actions:

Owner –

- Owner can fill the registration form.
- Owner can add, update and remove their boarding house details.
- Owners can view the comments about their boarding houses.
- Owners can update their profile details and deactivate their account from the system.

Admin –

- Admin can view all details of the owners, other admins and boarding houses.
- Admin can remove a particular owner and remove a specific boarding house in any circumstances.
- Admin can view user comments, block or unblock the comments.
- Admin can add new admin to the system and also he is able to remove an admin from the system. (There can be more than one admin, Super admin is the one who cannot be removed from the system)
- Admin can review the complaints given by the students (user).
- Admin can send emails regarding actions which was taken for the complaints by reviewing and solving.

Student –

- Student can view all details of the boarding houses and their owners.
- Student can find his appropriate boarding house.
- Student can add user comments and view the comments of the boarding houses.
- Student can complain and rate the boarding.

Acknowledgement

We would like to thank all our lecturers for accepting our group to develop Boarding Management System. And also, have to extend our gratitude to our supervisor Mr.S.Suthakar for approving our project and guiding us in the right path. Further, our assistant lecturer Mr.T.Laventhiran and supervisor Miss. Tharuka Dilani were a great support for us in developing our system. Next, we have to thank Dr.Ramanan and Miss. J. Samantha Tharani who guided us in the proper way in all our troubles and Miss.R.Nirthika who helped us in developing our project. All the friends who helped us in our troubles regarding the system. As a whole, we are so grateful to all the people who supported us on various ways to achieve our target.

Contribution of each member

- H.M.S.D.Herath – 2016/CSC/008
    - The owner data application form was created and gathered the requirements from the owners.
    - Then, the project has been divided into parts and works were assigned to the group members.
    - The diagrams were created. (Context diagram, Structural diagram, ER Diagram)
    - The database was designed with tables.
    - Backend coding was developed and modified.

- S. Lesslylavan - 2015/CSC/001
    - Frontend and template designing (User interface) was done.
    - Help was given to combine all the pages and finally project final report was written.

- S.K.K. Anuka Sanjaya - 2016/CSC/021
    - The owner data application form was created.
    - The requirements from the owners were gathered.
    - Frontend and template designing (User interface) was done.
    - Help was provided to finish final project report.
- Ajanthy Jayarajan - 2016/CSC/031
    - The google form was created to collect the students' requirements.
    - Tamil data collection form was created.
    - The diagrams were created. (ER Diagram, Use Case Diagram)
    - The backend was created and modified.
    - Help was provided to finish final project report.

- H.M.A.S.K. Illangasinghe - 2016/CSC/040
    - The owner data application form was created and gathered the requirements from the owners.
    - Developed the email system for the project and other related parts to the system.
    - Help was provided to finish final project report.

- R. Shehan Lahiru Jayasinghe - 2016/CSC/048
    - The diagrams such as class diagram and sequence diagram were developed.
    - Help was provided to modify the backend coding and to finish final project report.

Consent of the client

Consent of the client

Comments from Supervisor

Table of Contents

List of Figure

Acronyms

1. ER diagram – Entity Relationship diagram
2. UI – User Interface
3. RS – Requirement Specification
4. POS – Point of Sale
5. SDLC – System Development Life Cycle
6. IT – Information Technology
7. BMS-Boarding Management System

1 Introduction

1.1 Purpose of the project

Our purpose of this project is to create a Boarding Management System with the details of boarding houses nearby the university premises.

The system will be more digitalized and quite friendly to the environment as less paper work is required. The developed system will reduce the effort of students to find their boarding places. It will allow to find appropriate and safest boarding for the students. Students will obtain proper details about a boarding house without travelling cost. Students' valuable time and money can be saved. (Effective Time management).The boarding owners will obtain an opportunity to advertise their boarding with full details to easily reach their target customers. Confidentiality and privacy will be protected.

1.2 Scope of the project

Currently, the university students are obtaining the information about the boarding houses through newspapers, seniors and from their friends. However, the new system will allow a more convenient and easy way to retrieve information about boarding houses in the university premises. The system allows an effective way of details retrieval. Further, the system has a multilevel user access functionality that allows users with different roles to perform while using the same environment.

Figure 1-1-Logo

1.3 Overview of the project

The Boarding management system will provide a website to obtain boarding house details near the university. Further, the system facilitates to provide all the students of university of Jaffna a good, safe, healthy and comfortable residence for their learning purposes. Additionally, this works as a platform to connect both student who need boarding house and boarding house owners.

1.4 Overview of the report

This report covers the sections such as introduction, requirement engineering, software design, implementation, testing, discussions and challenges.

- Chapter 1 gives the introduction by including the purposes and scope.
- Chapter 2 includes the detailed information about requirement engineering which includes how we gathered our requirements, functional and non- functional requirements and etc.
- Chapter 3 includes all the diagrams and the software design approach which we handled in our project development.
- Chapter 4 displays the user interfaces of our systems and special features.
- Chapter 5 depicts about the testing process.
- Chapter 6 is about the challenges that we faced in our project and our future plans.

2 Requirement Engineering

Before gathering the requirements from the owner we took a permission letter from the department to prove that we are getting their details to our project.

DCS
UNIVERSITY OF JAFFNA

10 September 2018

From: Dr. A. Ramanan
Head
Department of Computer Science
University of Jaffna

To whom it may concern,

Request to Help our Undergraduate Students in their Group Project

The Department of Computer Science is offering a course unit with the course code CSC241SC2: Group Project I, for the Level-2S students. In this regard, five to six students together form a group and are expected to develop a system in eight months duration in order to complete it before entering to their third year. I understand that the following group of students are interested to develop a web-based application to facilitate the undergraduate students in securing a private accommodation during their study programme at the University of Jaffna under the supervision of our staff Mr. S. Suthakar (Senior Lecturer in Computer Science).

Name of the Student	Reg. No.
Mr. H.M.S.D. Herath*	2016/CSC/008
Mr. S. Lesslylavan	2015/CSC/008
Mr. S.S.K. Anuka Sanjaya	2016/CSC/021
Ms. Ajanthy Jayarajan	2016/CSC/031
Ms. Abhimani Sachithra	2016/CSC/040
Mr. Shehan Lahiru Jayasinghe	2016/CSC/048

* Team leader of the Group

Please be good enough to support the above students to provide information regarding the accommodation available to the students in your property.

Thanking You.

Yours Sincerely,

HEAD
Department of Computer Science
University of Jaffna
Sri Lanka.

Figure 2-1: Permission Letter

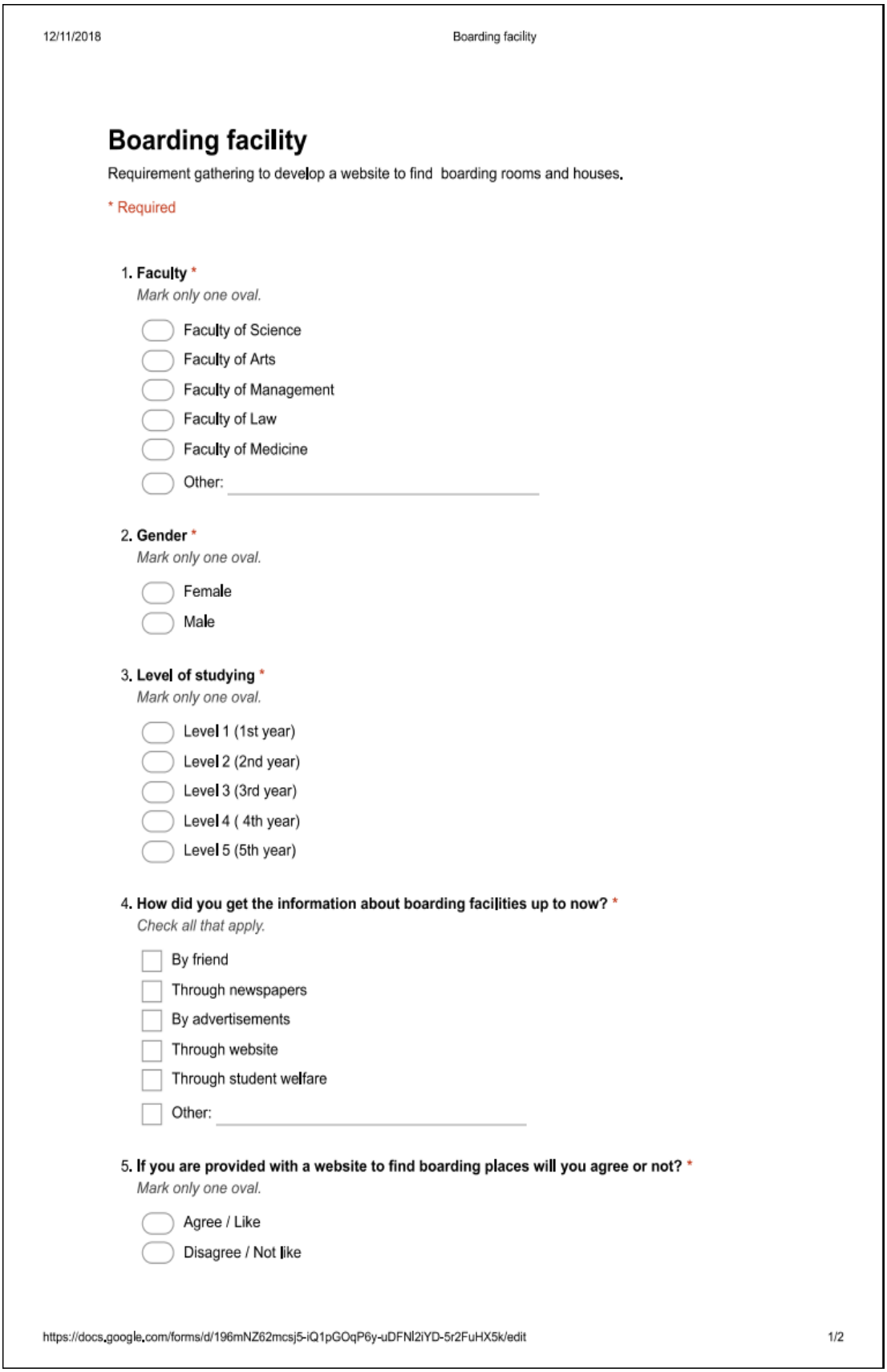
2.1 Techniques used for requirements gathering

It's difficult to build a solution if we don't know the exact requirements. Requirement gathering was done by google sheet which was shared among the students of all faculties in university of Jaffna. In the survey, students were requested about the requirements. Further, similar websites were analyzed and some ideas were obtained. By analyzing the data, we decided what the system should do.

We were needed to collect the student and owner requirements.

Student Requirements

The requirements which are expected by the user (student) were collected from the university students by google form which is attached below. And, 218 responses were received from the university students.



12/11/2018 Boarding facility

Boarding facility

Requirement gathering to develop a website to find boarding rooms and houses.

* Required

1. **Faculty** *
Mark only one oval.
- ◯ Faculty of Science
- ◯ Faculty of Arts
- ◯ Faculty of Management
- ◯ Faculty of Law
- ◯ Faculty of Medicine
- ◯ Other: _____

2. **Gender** *
Mark only one oval.
- ◯ Female
- ◯ Male

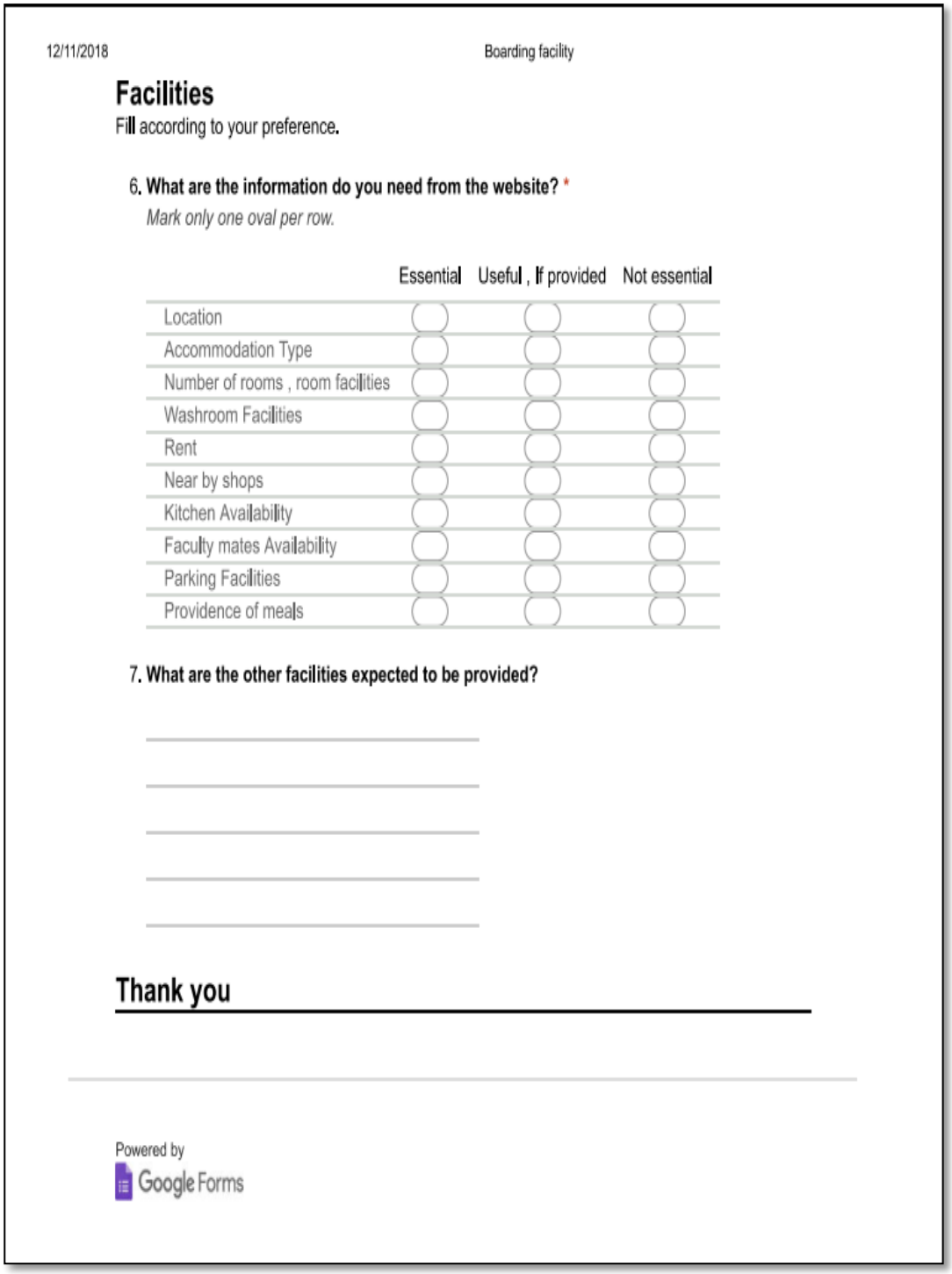
3. **Level of studying** *
Mark only one oval.
- ◯ Level 1 (1st year)
- ◯ Level 2 (2nd year)
- ◯ Level 3 (3rd year)
- ◯ Level 4 (4th year)
- ◯ Level 5 (5th year)

4. **How did you get the information about boarding facilities up to now?** *
Check all that apply.
- ☐ By friend
- ☐ Through newspapers
- ☐ By advertisements
- ☐ Through website
- ☐ Through student welfare
- ☐ Other: _____

5. **If you are provided with a website to find boarding places will you agree or not?** *
Mark only one oval.
- ◯ Agree / Like
- ◯ Disagree / Not like

https://docs.google.com/forms/d/196mNZ62mcsj5-iQ1pGOqP6y-uDFNl2iYD-5r2FuHX5k/edit 1/2

Figure 2-2: Google form page 1



12/11/2018 Boarding facility

Facilities

Fill according to your preference.

6. **What are the information do you need from the website?** *

Mark only one oval per row.

	Essential	Useful , If provided	Not essential
Location	◯	◯	◯
Accommodation Type	◯	◯	◯
Number of rooms , room facilities	◯	◯	◯
Washroom Facilities	◯	◯	◯
Rent	◯	◯	◯
Near by shops	◯	◯	◯
Kitchen Availability	◯	◯	◯
Faculty mates Availability	◯	◯	◯
Parking Facilities	◯	◯	◯
Providence of meals	◯	◯	◯

7. **What are the other facilities expected to be provided?**

Thank you

Powered by Google Forms

Figure 2-3: Google form page 2

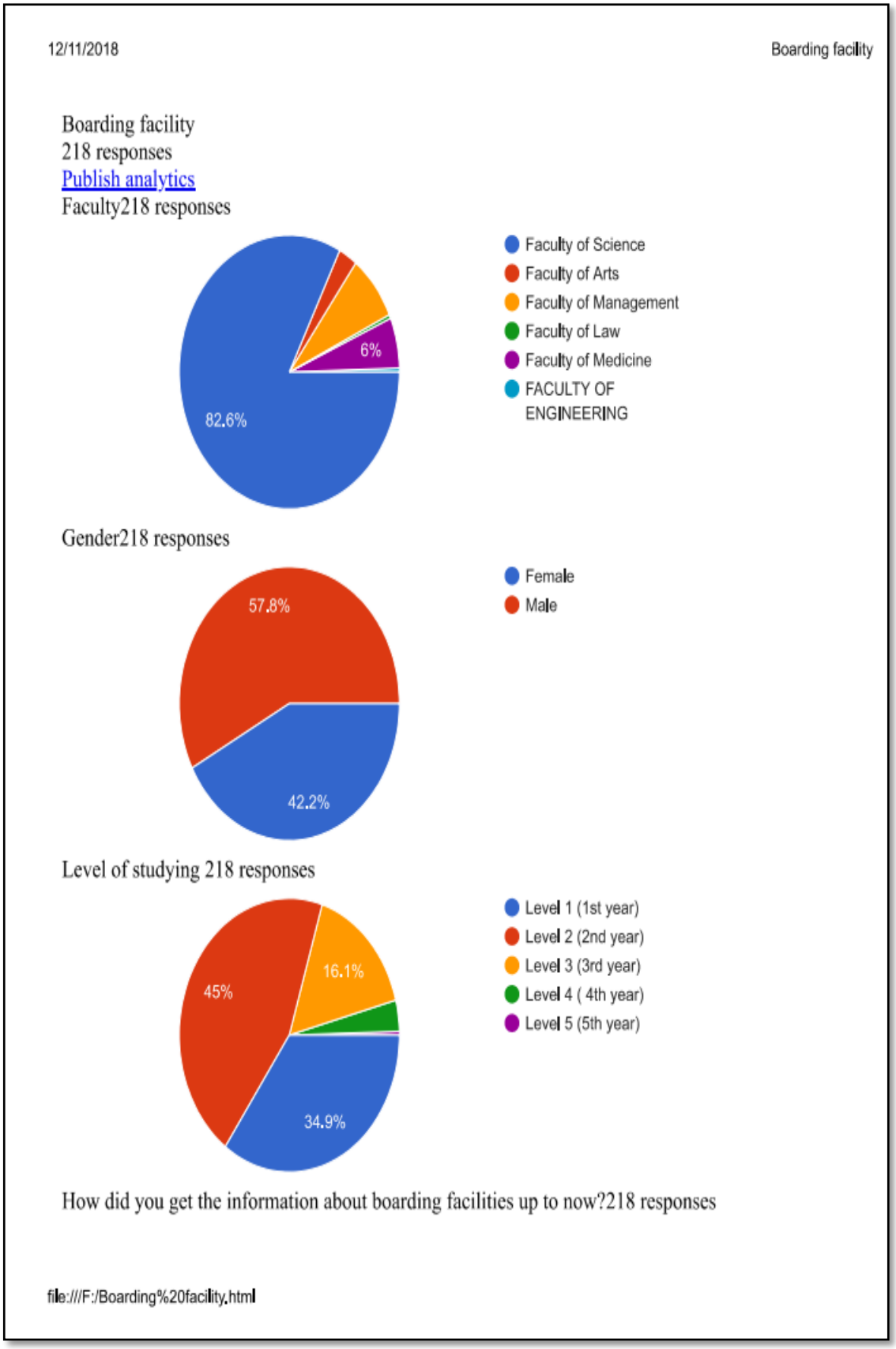


Figure 2-4: Google form Response 1

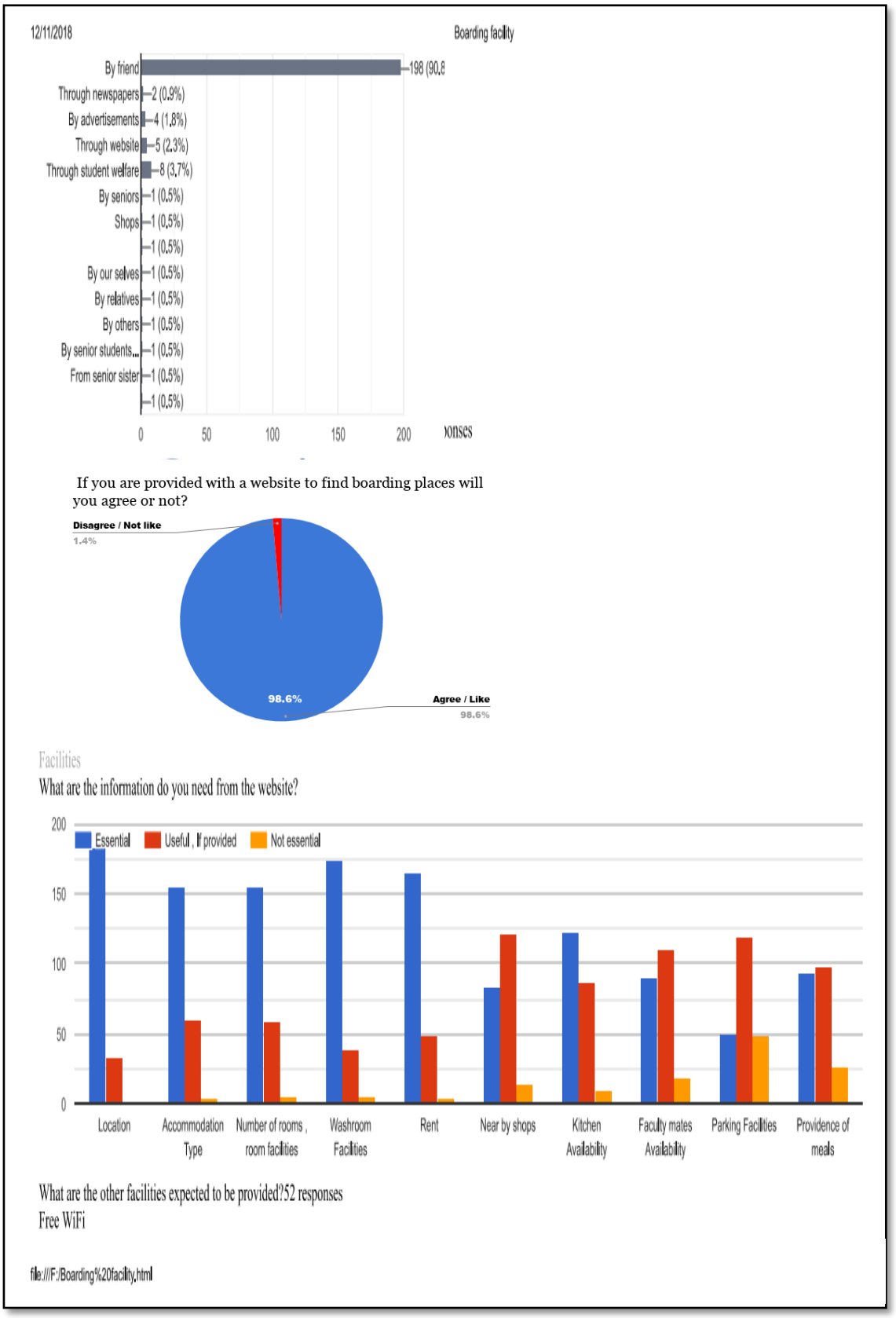


Figure 2-5: Google form response 2


12/11/2018 Boarding facility

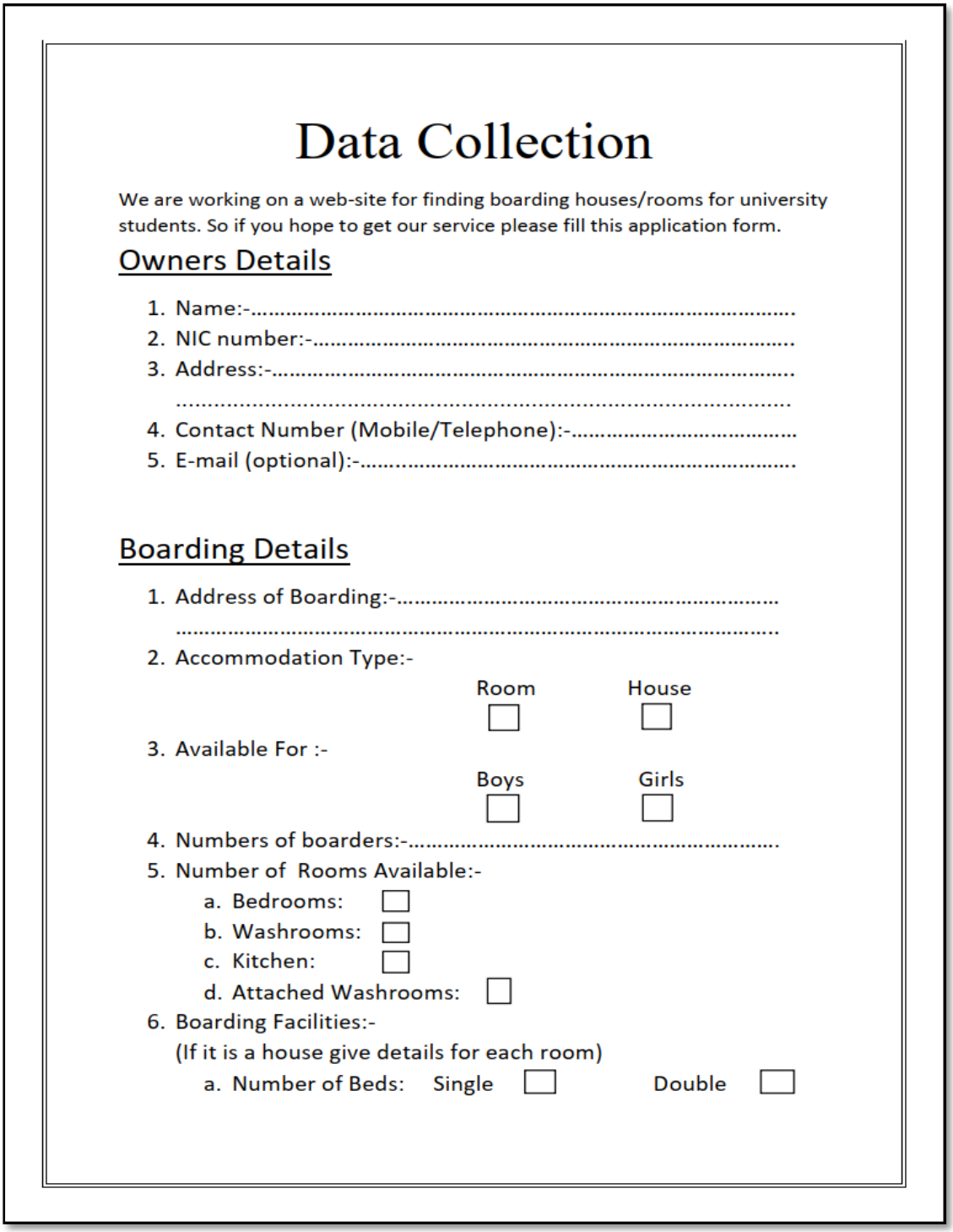
Security for girls boarding houses
Agreement details, whether the boarding will given on contract basis or oral talks
No
How far from campus to bording
It is enough
😀😀😀😀😀😀😀
If rent is shared, managing that using the same app
Safety
Distance from university
Details about Contact numbers and about the availability of a security(owners/security stay there or not).
How long(how many years) they will give the room for rent Do we need to pay the full amount monthly or have to pay some advance first Do we have any time restriction to come to the room
Distance from the university, current bill, water bill price
Transport
Distance to the university
Study hall
It will be useful if they specify the rent as per person or per room or per house. And also is it included the electricity bill and other payments or not.
Distance from University of Jaffna Other conditions of Owner(ex.Veg/Non Veg cooking) Review of the people who already stayed there
Furniture facilities and whether the rent includes the current bill or not
Near to town
Protection
Good for girls,advance,condition
Availability and functioning of Water supplying units
Furnitures, cooking goods
Distance to faculty
Wifi and other e resourses like moodle
Electricity charging
Drinking water-Essential
Wifi facilities
About the security of the surrounding. (especially if it is a girls' boarding house) About the contact details of the owner, previously stayed ones etc.
FIlter drinking water
Camera system available
Drinking water Availability
Near by university
Pure water for drinking
In jaffna university students one of the main problem is the water. Students have to spend lots of their money to buy water. In main campus premises there is a water filter. If there is water transport facilities provided it will be very helpful for students.
Distance to the campus
Current Bill and Water bill..
Working with a friendly manner
Security for boardings
Owner's contact number

file:///F:/Boarding%20facility.html

Figure 2-6 Google form Response 3

Owner requirements

The boarding details of specific owners has been collected via data collection form. Further, due to the language issue, the form was developed in both Tamil and English Languages.



Data Collection

We are working on a web-site for finding boarding houses/rooms for university students. So if you hope to get our service please fill this application form.

Owners Details

1. Name:-.....
2. NIC number:-.....
3. Address:-.....
   .....
4. Contact Number (Mobile/Telephone):-.....
5. E-mail (optional):-.....

Boarding Details

1. Address of Boarding:-.....
   .....
2. Accommodation Type:-
   Room ☐ House ☐
3. Available For :-
   Boys ☐ Girls ☐
4. Numbers of boarders:-.....
5. Number of Rooms Available:-
   a. Bedrooms: ☐
   b. Washrooms: ☐
   c. Kitchen: ☐
   d. Attached Washrooms: ☐
6. Boarding Facilities:-
   (If it is a house give details for each room)
   a. Number of Beds: Single ☐ Double ☐

Figure 2-7: Data Application Form 1

b. Wardrobe: ☐
c. Study Table: ☐
d. Chairs: ☐
e. Fans: ☐

7. Monthly boarding charges:-.....
   Include: Electricity ☐
   Water ☐
   Meals ☐
8. Conditions for boarders:-.....
   .....
   .....
   .....

I confirm that the details above are true.

Date:-..... Signature:-.....

Figure 2-8: Data Application Form

தரவுத் தொகுப்பு

வெளி விடுதி தொடர்பான தகவல்களை மாணவர்களுக்கு எடுத்து செல்லும் இணையத்தளம் ஒன்றை உருவாக்க உள்ளோம். அதில் உங்களுடைய விடுதிசார் சேவைகள் இடம்பெற விரும்பினால் இவ் விண்ணப்ப படிவத்தை நிரப்புங்கள்.

உரிமையாளர் விபரம்

1. பெயர்:-.....
2. அடையாள அட்டை இலக்கம்:-.....
3. முகவரி:-.....
   .....
4. தொடர்பிலக்கம்:-.....
5. மின்னஞ்சல்:-.....

விடுதி பற்றிய விபரங்கள்

1. விடுதியின் முகவரி:-.....
   .....
   .....
2. வதிவிட வகை:-

   அறை ☐ வீடு ☐

3. Available For :-

   பெண்கள் ☐ ஆண்கள் ☐

4. விடுதியில் வசிப்போரின் எண்ணிக்கை:-.....
5. அறைகளின் எண்ணிக்கை:-
   a. குளியலறை: ☐
   b. படுக்கையறை: ☐
   c. சமையலறை: ☐
   d. இணைக்கப்பட்ட குளியலறை: ☐

6. தங்குமிட வசதிகள்:-
   (வீடாயின் ஒவ்வொரு அறைக்குமான தகவல்களைத் தரவும்)
   a. படுக்கை வசதி: Single ☐ Double ☐

Figure 2-9: Data Application form (Tamil)

b. அலுமாரி: ☐
c. மேசை: ☐
d. கதிரை: ☐
e. விசிறிகள் ☐

7. மாத வாடகை:-.....

Include: Electricity ☐
Water ☐
உணவு ☐

8. விடுதியில் தங்குபவர்களுக்கான மேலதிக கட்டளைகள்:-

.....
.....
.....
.....
.....
.....

மேற்குறிப்பிட்ட தகவல்கள் அனைத்தும் உண்மையானவை என உறுதிப்படுத்துகிறேன்

....

திகதி கையொப்பம்

Figure 2-10: Data Application Form (Tamil)

2.2 Functional requirements

1. Admin signup:
   - Actor: Admin
   - Input: Admin have to provide all the necessary details present in the sign-up page
   - Output: All the details entered in the page will be verified and accepted by the system into database
2. Admin Login:
   - Actor: Admin
   - Input: Admin uses the user-name and password for login which he/she created at the time of signup
   - Output: The system will check the username and password with the database and accept into the system
3. Checking owners details:
   - Actor: Admin
   - Input: Admin uses the owner details section to obtain information about specific owners and their boarding house details.
   - Output: The system will check owner ID number with the database and retrieve the details of the owner from the database
4. Removing the Owner:
   - Actor: Admin
   - Input: Admin uses option available in the system to remove the details
   - Output: The system will check the owner ID number with the database and remove the details of the owner from the database
5. Removing the boarding details:
   - Actor: Admin
   - Input: Admin uses the option available in the system to remove the specific boarding belong to the owner
   - Output: The system will check the boarding ID number in the database and remove the details of the boarding from the database

6. Commented Users:
   - Actor: Admin
   - Input: Admin check the comments by the students for the appropriate boarding
   - Output: Admin can block or unblock the comments given by the students
7. Monitoring Complaints:
   - Actor: Admin
   - Input: Admin monitors the complaint and take appropriate actions regarding that
   - Output: The action will be updated by the system.
8. Owner signup:
   - Actor: Owner
   - Input: Owner have to provide all the necessary details present in the sign-up page
   - Output: All the details entered in the page will be verified and accepted by the system into database.
9. Owner Login:
   - Actor: Owner
   - Input: Owner uses the user-name and password for login which he/she created at the time of signup.
   - Output: The system will check the username and password with the database and accept into the system.
10. Owner fill up the boarding details :
    - Actor: Owner
    - Input: Owner have to provide all the necessary details present in the boarding details form
    - Output: All the details entered in the page will be verified and accepted by the system into database
11. Owner can update, remove the boarding details:
    - Actor: Owner

- Input: Owner check the details provided by him/her and make appropriate changes
- Output: The system will check the changes with the database and accept into the system

12. Owner can deactivate his/her account:
    - Actor: Owner
    - Input: Owner can deactivate his/her account if the boarding is not available
    - Output: The system will remove the details of the specific owner from the database.
13. Student can search for a boarding according to their wish:
    - Actor: Student
    - Input: Student can search for a boarding house according to their need (such as rent below 3000, only for girls etc.)
    - Output: The system will sort the details according to the students' wish
14. Student can comment and complaint about the boarding house:
    - Actor: Student
    - Input: Student can comment and complaint in the option available below the boarding details
    - Output: The details entered in the page will be verified and accepted by the system into database

2.3 Non-functional requirements

1. Performance Requirements

- All the queries shall take no longer than 10 seconds to load onto the screen after the user request.
- JPEG, JPG, PNG, and most of other image types are supported when uploading the boarding pictures, boarding profile picture and owner profile picture and the picture size less than 100 Mega Bytes can be uploaded.
- Pictures and all other formats shall not take more than 15 seconds to load onto the screen.

2. Security Requirements

- Admin and owner is required to login to the system to view the requests by his/her username and password.
- All the input fields are validated.
- The system verifies only registered admin and owner.
- The system contains high secured password with alpha numeric characters, symbolic characters.
- The responsibilities and access permissions have been divided between Super Admin (Admin who cannot be removed from the system), Admin, Owner, Student.
- The safety, wellbeing, continuity of the system is improved by provision of blocking the comments and blocking the user.
- Admin cannot be an owner. If an owner has to be an admin, first he has to remove all the boarding houses belongs to him and after that only he can be an admin of the system.

3. Software Quality Attributes

- Availability: The system is available to students, admin and owner at all the time on the internet.
- Maintainability: The system can be updated or upgraded according to the new user requirements without affecting the old version.
- Usability: The system is user- friendly. The user doesn't need any knowledge to handle this.
- Robustness: The query to be searched is always verified by the system. The inputs given by the user is always verified.
- Portability: Internet can be accessed in laptops, personal computers and mobile phones. Therefore, anyone at any place can obtain access to the system.

2.4 Domain requirements

The global objective of the domain requirements description activity is gathering needs of expectations of application stakeholders and at providing a complete description of the behavior of the application to be developed.

1. Login:

    - Description: This use case describes the scenario where the admin logs in to system.

    - Actor: Admin

    - Input: The admin logs into the system with the user-name and password which were provided by him while registering.

2. Signup and Login:

    - Description: In Sign up, the admin wants to provide the details which are in the sign-up page. This use case describes the scenario where the owner logs in to system.

    - Actor: Owner

    - Input: The owner logs into the system with the user-name and password which were provided by him while registering.

3. Update/delete the boarding details:

    - Description: This use case describes the scenario where the owner can update or delete the students in the system.

    - Actor: Owner

    - Input: While updating or deleting the details these details will be uploaded in the system.

4. View the boarding system:

    - Description: This use case describes scenario where the user (students) can view the sorted list details about boarding houses.
    - Actor: Student
    - Input: Student can view the boarding and can get the sorted boarding details.

2.4.1 System requirements

- A desktop computer or Laptop or Mobile Phone with Internet facility
- Windows 7 or higher or a Linux operating system.
- Internet

2.4.2 Stakeholders

- Students of University of Jaffna.

  It contains the faculties

  - Faculty of Science
  - Faculty of Management
  - Faculty of Medicine
  - Faculty of Arts.
  - Faculty of Law.

2.5 Challenges

- Language Problem

  In Jaffna mostly it is surrounded by Tamils so while us collecting details from the house owners they don't know English to communicate with us. So to solve this problem we have created an application form in Tamil.

- Time

  Limited time frame. Task is behind schedule.

3 Software Design

Project was designed using an object-oriented approach. While developing the system, several diagrams were come across to achieve the optimum quality of the software product. The design strategy is mainly focused on entities and its characteristics. Initially, the objects were identified and grouped into classes.

The classification and characteristics of entities was categorized using class diagram. Further, the ER diagram was created. Finally, with the help of ER diagram and class diagram database was created.

3.1 Software design approach

For the development of the software product, the best which helps to design, develop and test high quality software which meets or exceeds customer expectations, reaches completion within times and cost estimates have to be selected. The method which is used to develop the system is the Software Development Life Cycle (SDLC). The SDLC process includes project identification and selection, project introduction and planning analysis, design, implementation and maintenance.

Structure of a methodology: -

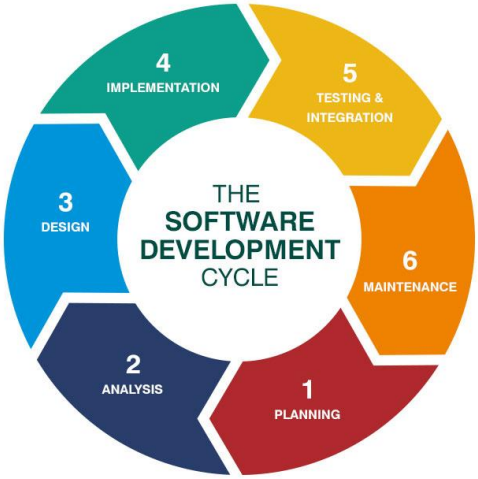


Figure 3-1: Software Development Life Cycle

Firstly, planning and requirement gathering was carried out. Secondly, analysis of the system was continued. Thirdly, analysis of the system is being translated into

design. Fourthly, frontend and Backend Code was produced during implementation which is driven by the design. Finally, testing was done to the implemented system.

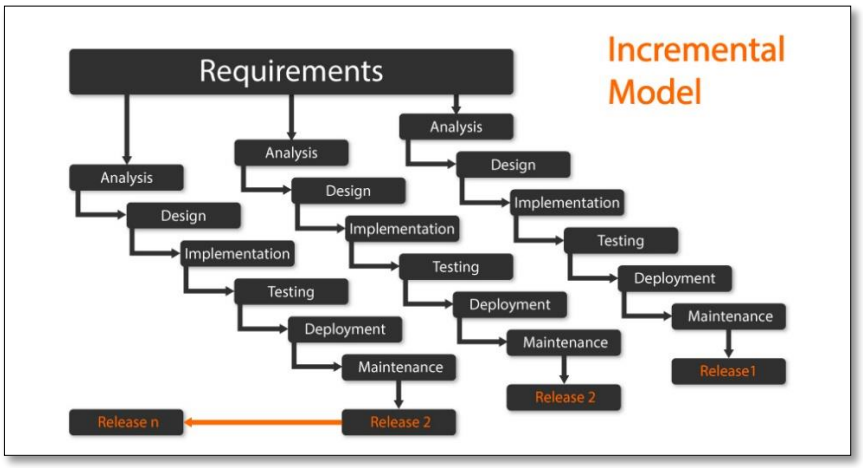


Figure 3-2: Incremental Model

We have chosen incremental model approach because of the below reasons.

- Requirements of the system are clearly understood.
- Early releases can be shown.
- The changes can be done in the stages.
- Customers' feedbacks can be received.
- Easy to identify the errors.

3.2 Context diagram

A context diagram in engineering is a diagram that defines the boundary between the system or a part of the system and its environment, showing the entities that interact with. This diagram is high level view of the system. It is similar to block diagram.

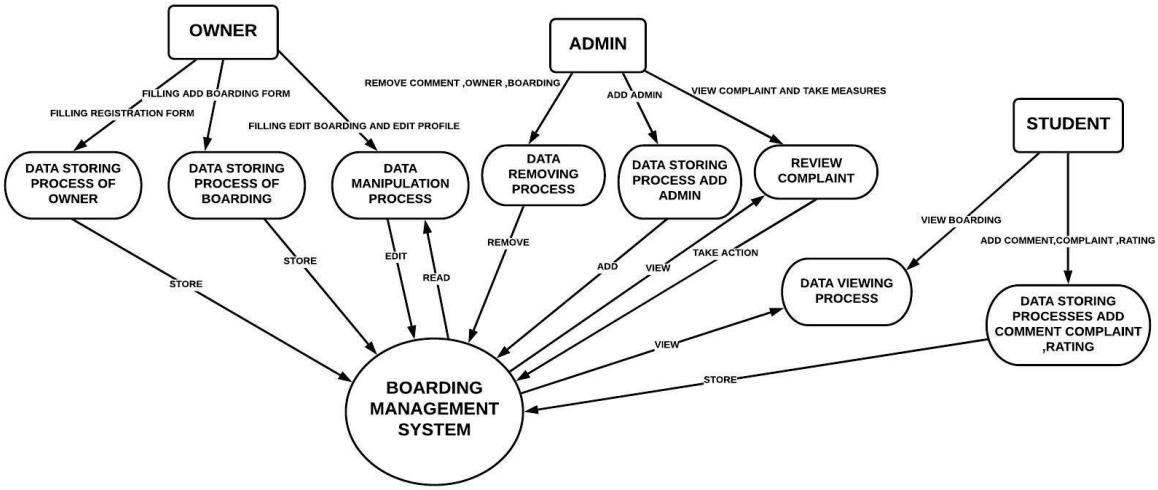


Figure 3-3: Context diagram

3.3 Structural diagrams

Structural diagram shows the hierarchy or structure of different components or modules of the system and shows how they connect and interact with each other.

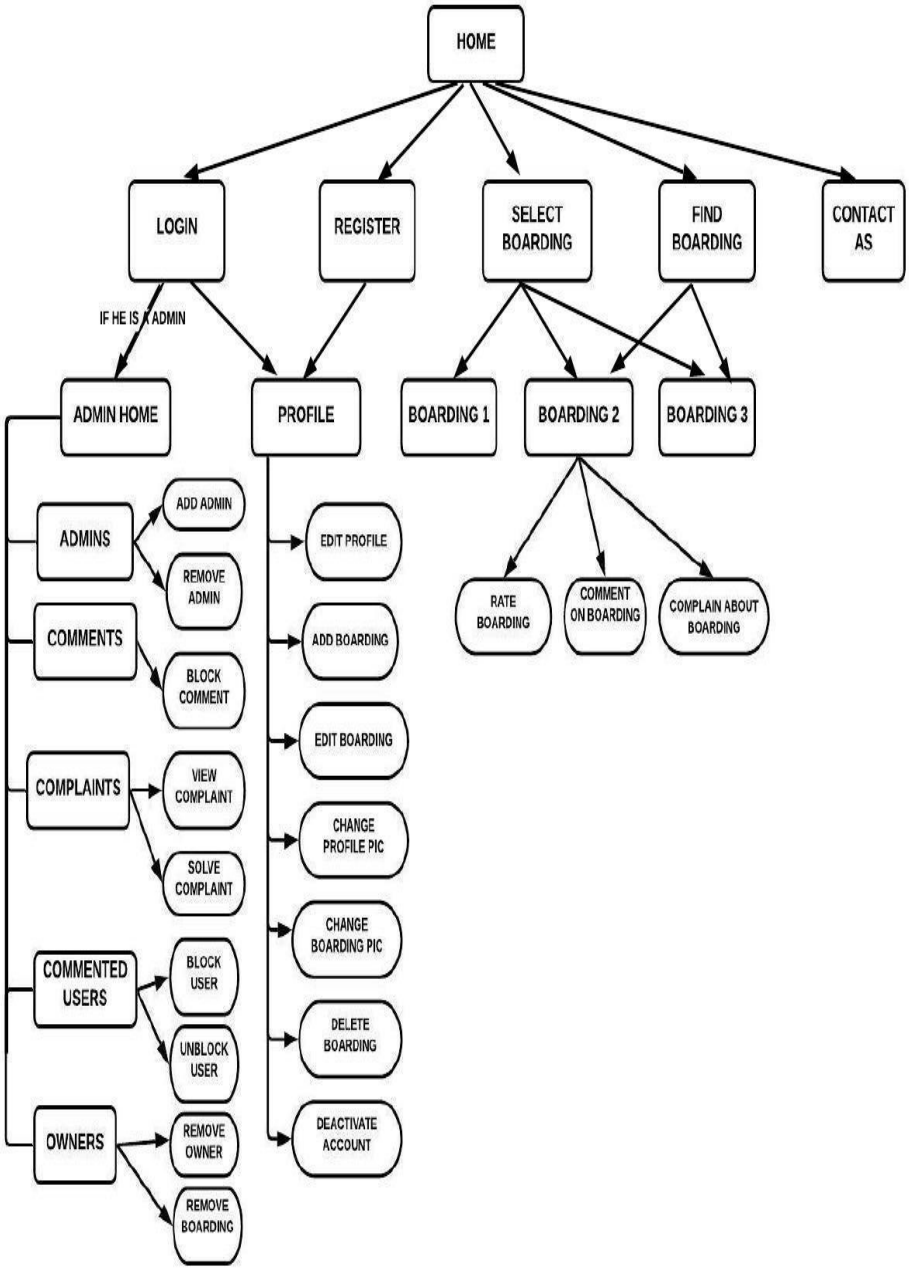


Figure 3-4: Structural diagram

3.4 Interaction diagrams

Interaction diagrams are modules that describe how the group of objects collaborate in a single use case. Sequence diagram is a better way to describe single use case.

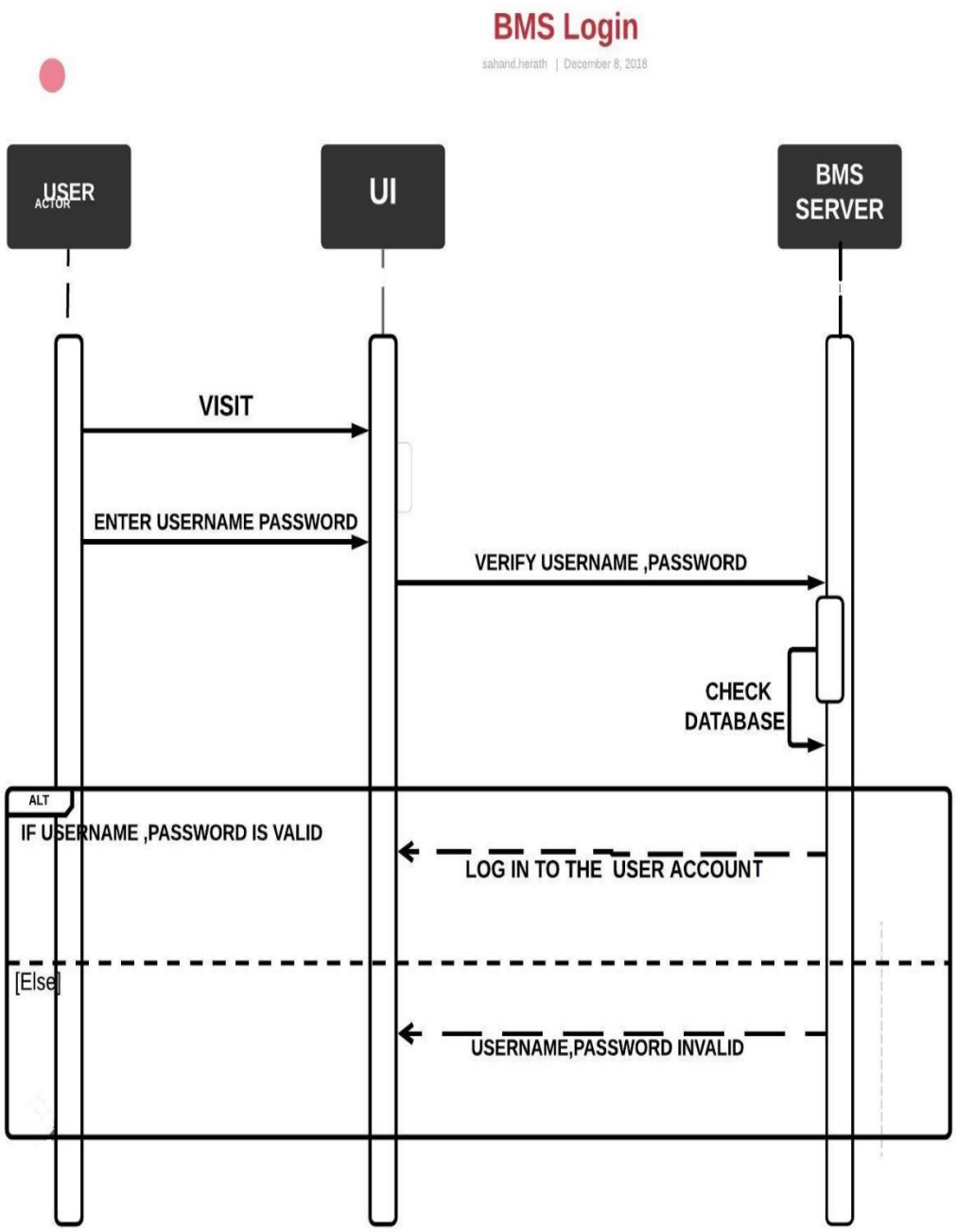


Figure 3-5: Sequence Diagram (Login)

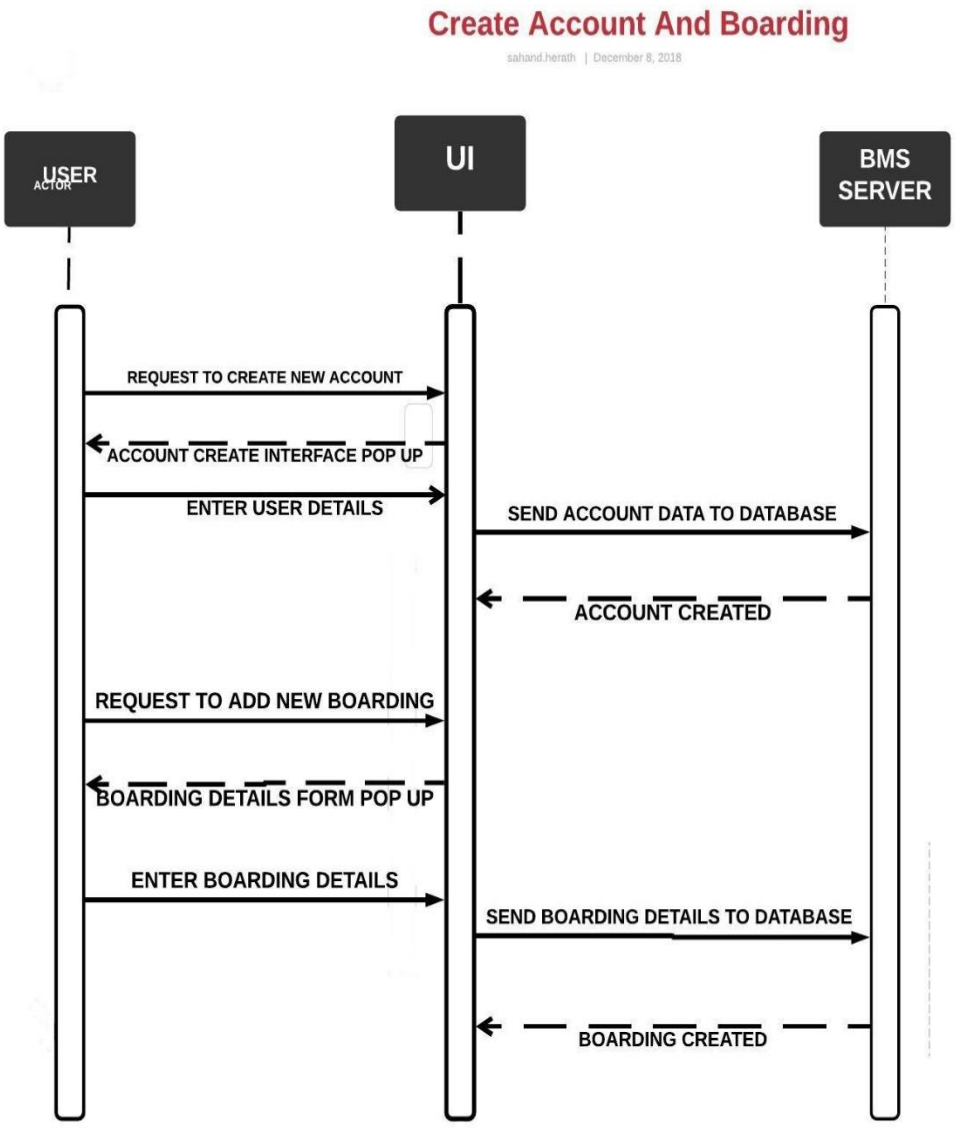


Figure 3-6: Interaction diagram (Create and Add boarding)

Complaining and Solving

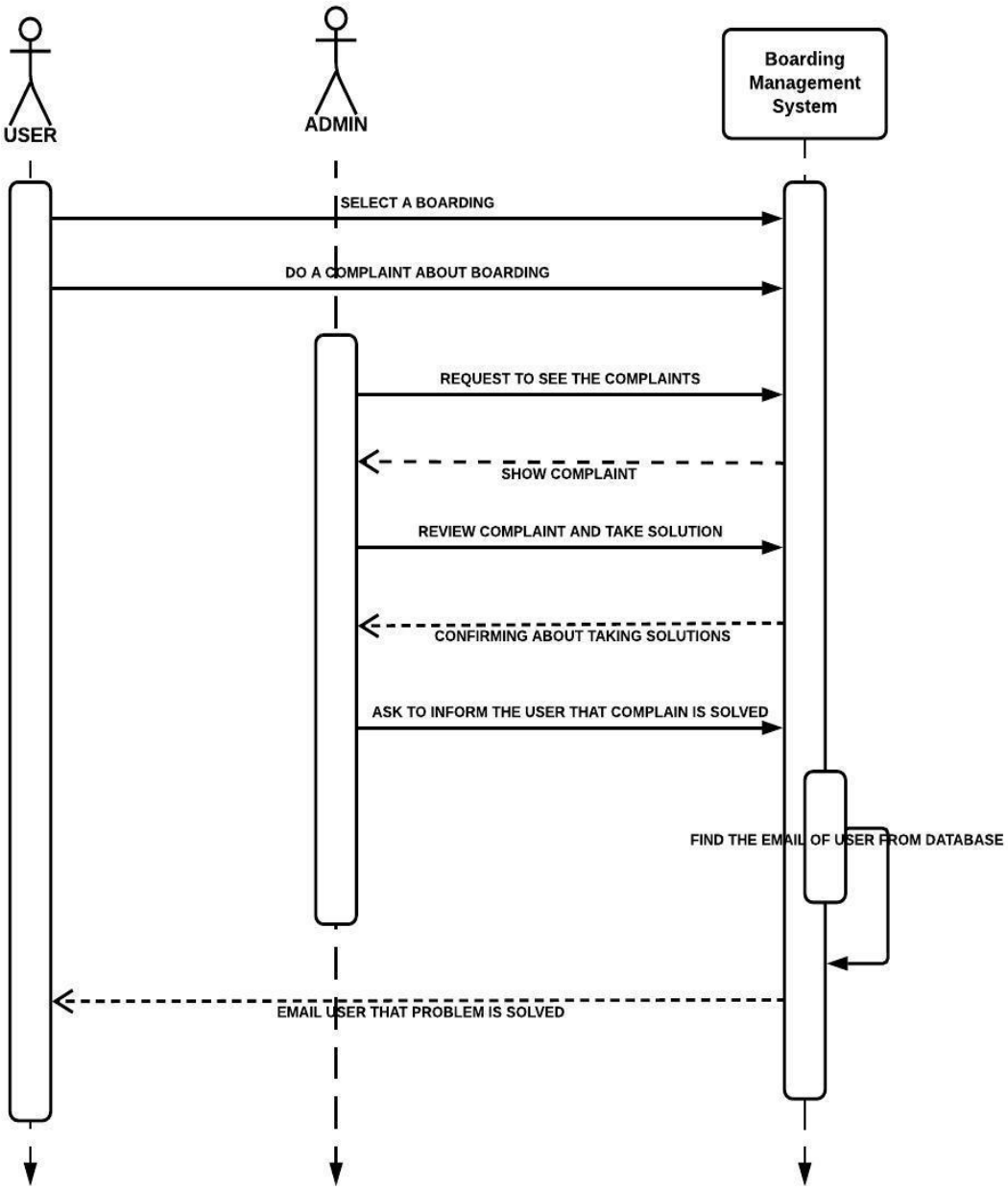


Figure 3-7: Sequence diagram (Complain)

Commenting and blocking comment

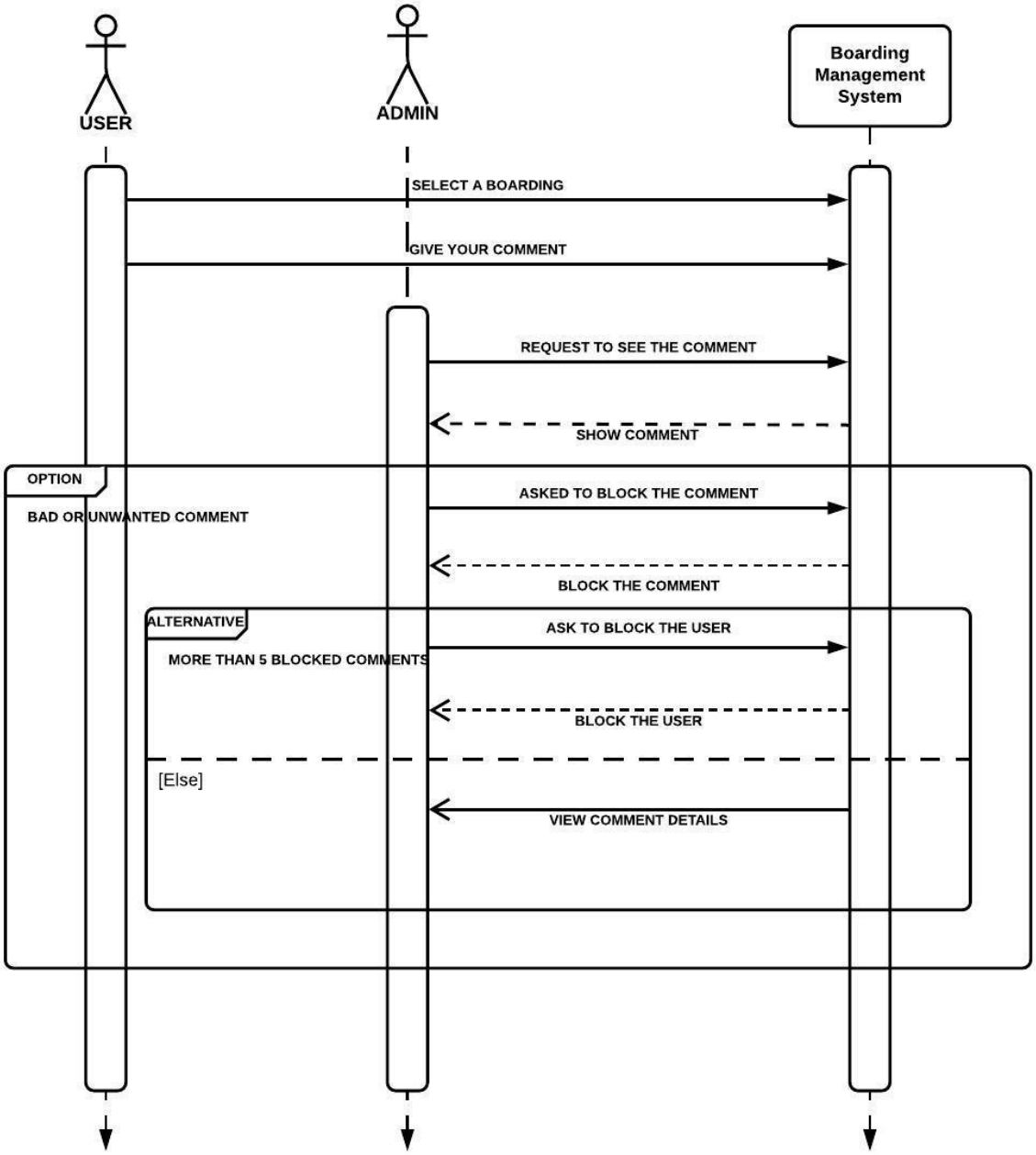


Figure 3-8: Sequence Diagram (Comment)

3.5 Behavioral diagrams

Behavioral diagrams depict the elements of the system that are dependent on time and that convey the dynamic concept of the system and how they are related to each other.

Use case diagrams, activity diagram, state-chart diagram are some examples for behavioral diagrams.

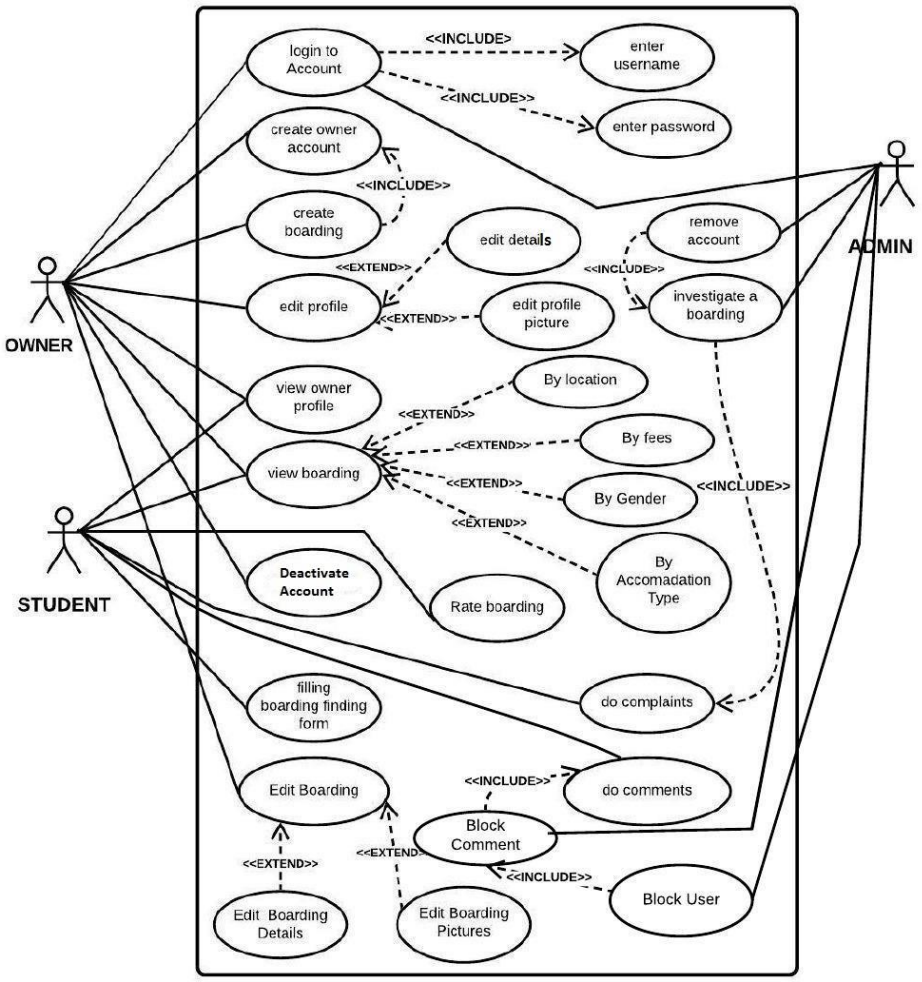


Figure 3-9: Sequence Diagram (Comment)

3.6 Data model

3.6.1 ER diagram

Entity Relationship Diagram: An entity relationship model describes interrelated things of interest in a specific domain of knowledge. The basic ER model is composed of entity types and specific relations that can exist between them.

Entity: These are objects or concepts. We can store data about them. In our project we have several entities.

1. User
2. Boarding
3. Charge
4. Picture
5. Room
6. Facility
7. Comment
8. Complaint
9. Rating

Relationship: A relationship that exist between entities are called relationships.

Attributes: Attributes are properties or characteristics of entities and ER attribute can be denoted as a primary key (which identifies a unique attribute) or a foreign key.

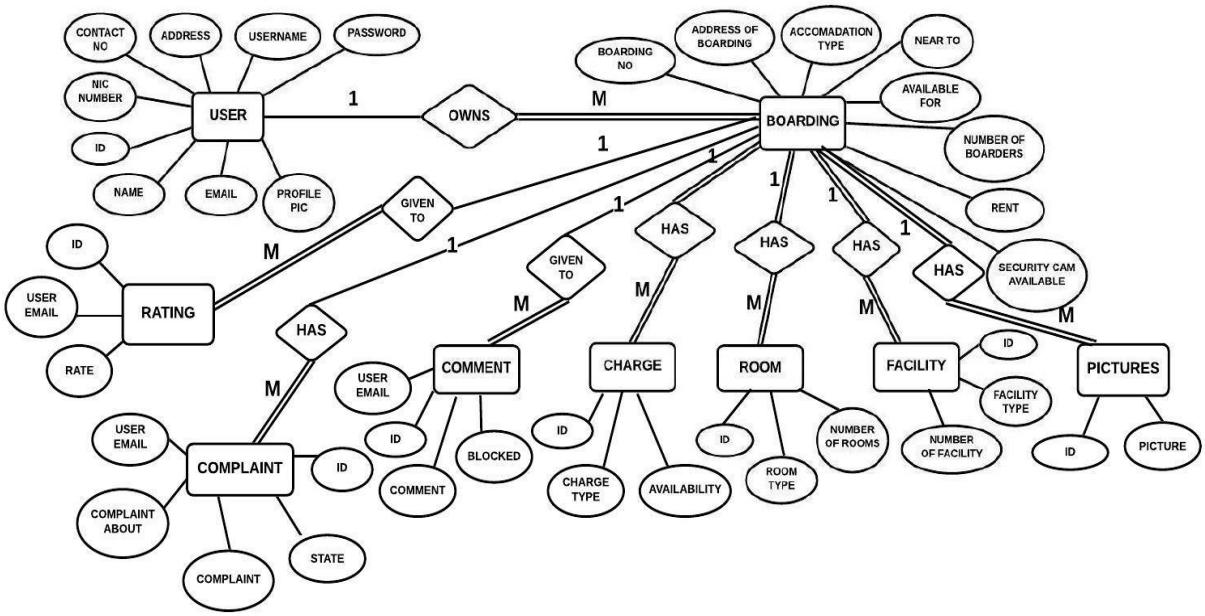


Figure 3-10: Entity relationship Diagram

- We use auto increment ID as primary keys.
- We store pictures' names in the database while keeping those pictures in the folder named public.

3.6.2 Class diagram

Class diagrams are structural diagram that describes the structure of a system by showing the system classes, their attributes, operations and relationships among objects.

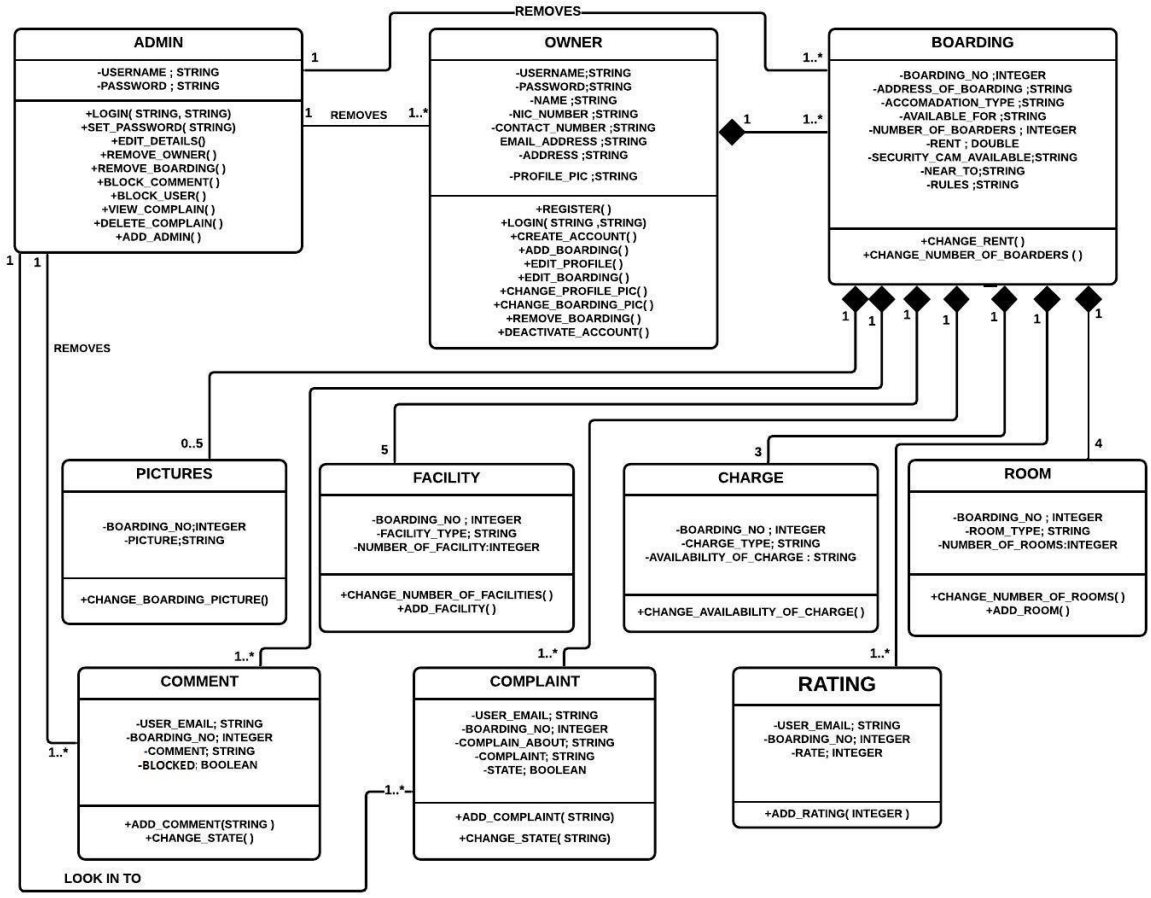


Figure 3-11: Class Diagram

4 Implementation

4.1 Tools and Techniques used and justification

- The database was created in MySQL using XAMPP as the server.
- For the user interface we used Google's Materialize CSS framework and customized it.
- We used php framework laravel 5.5 and some of its components.

❖ Home page

- In the home page we have options for login, register, select boarding, find boarding, about us, contact us.

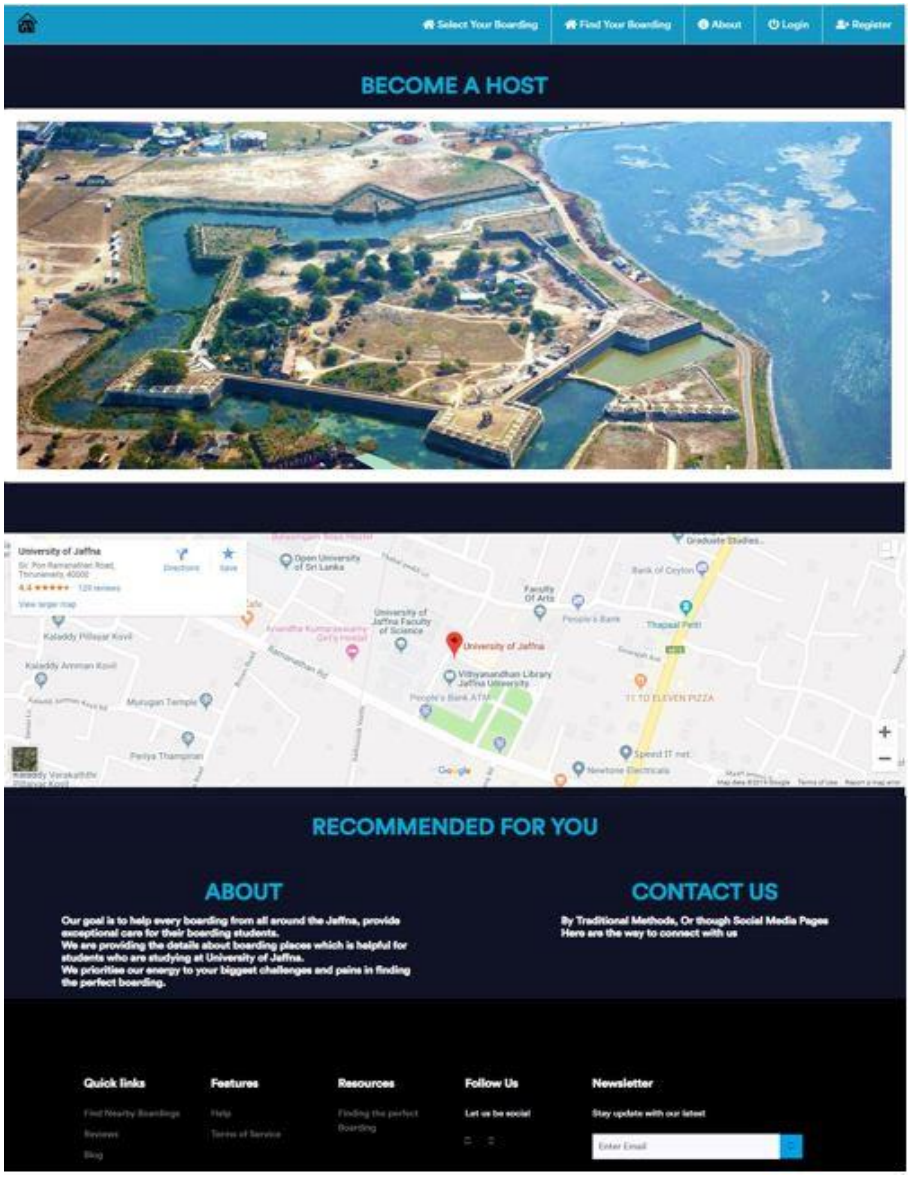


Figure 4-1: Home Page

❖ Register and login

- Admin and owner both have the same login page. Owner register through the given register form.

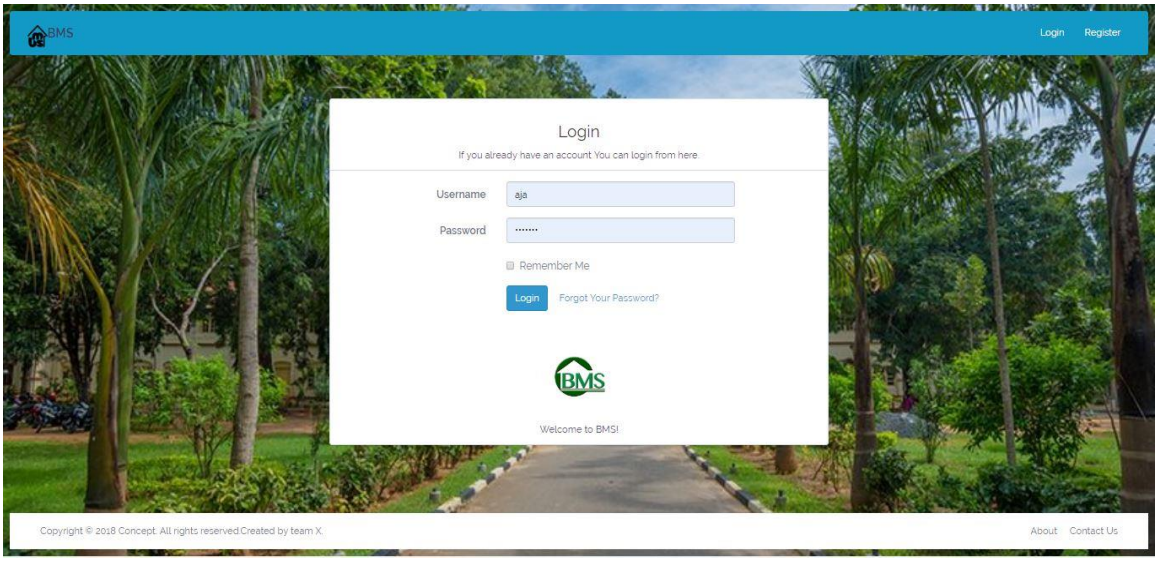


Figure 4-2: Login

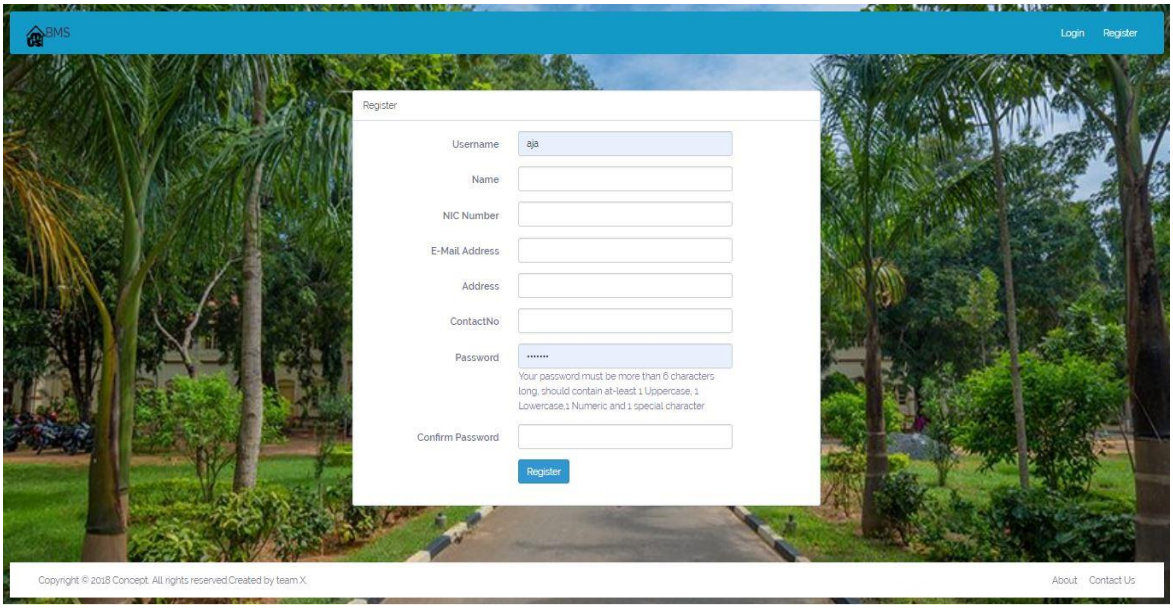


Figure 4-3- Register

❖ Admin home

- After login as an admin you will be directed to admin home. All the admin options will be available there.

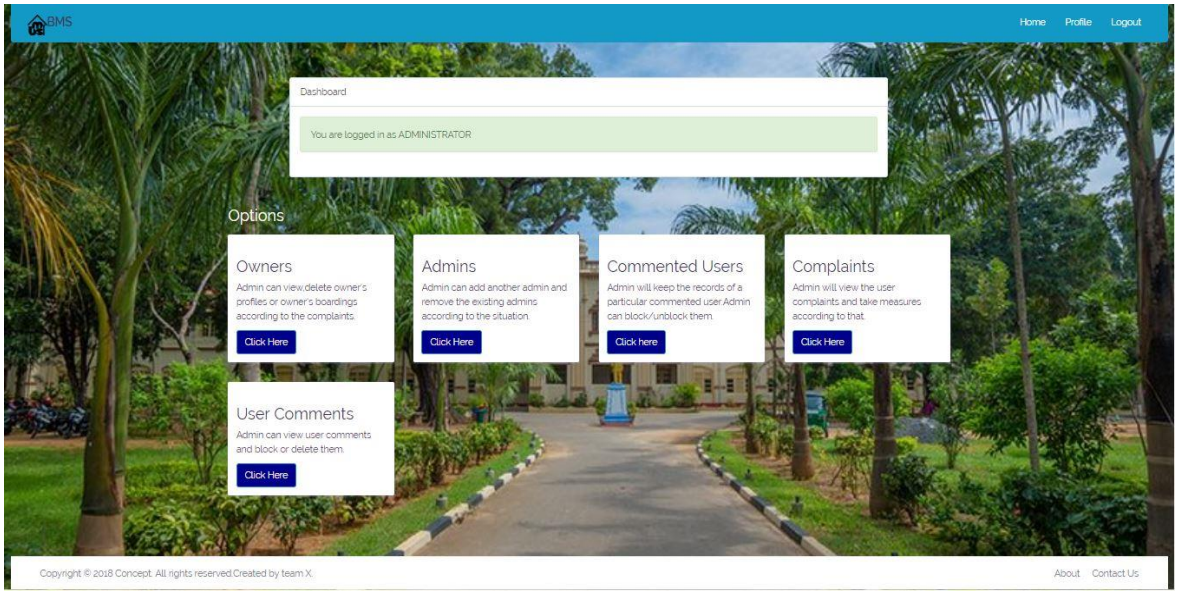


Figure 4-4- Admin home

❖ Admin details

- Details of all admins are available in admins home. Admin can add or remove an admin on this page.

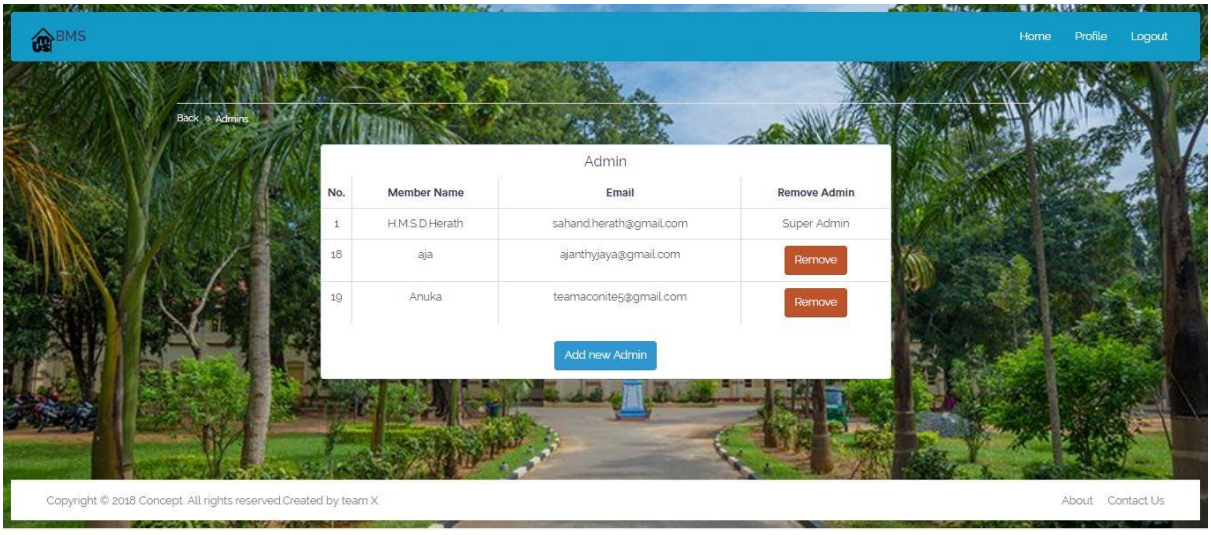


Figure 4-5- Admin page

❖ Owners' details

- Admin can view any owner detail and he is able to remove a boarding or owner according to the complaints that he get.

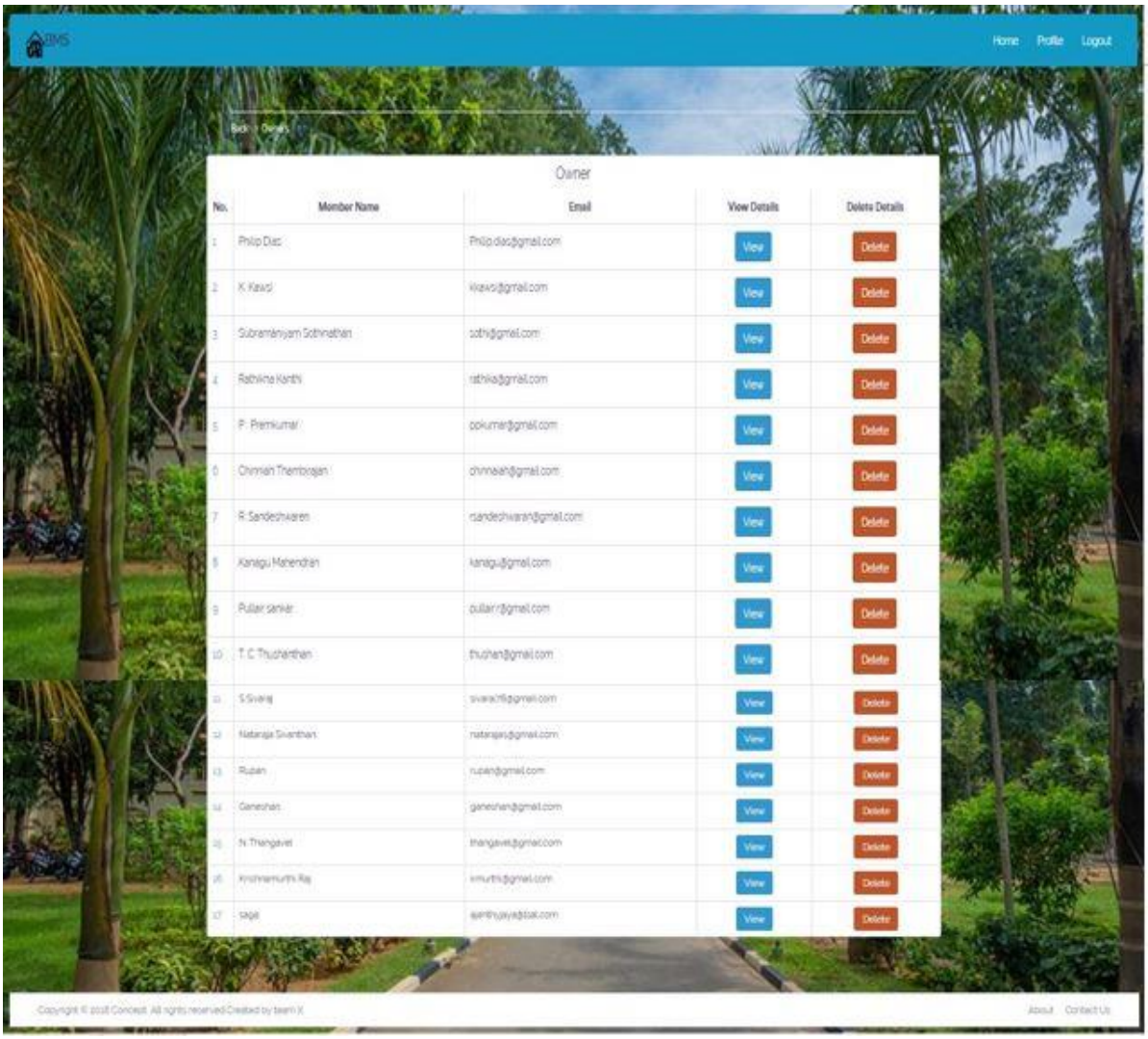


Figure 4-6- Owner details

❖ Complaint Details

- Admin can view all complaint details and he can mark the complaint as read, solved. After solving those complaints can be deleted.

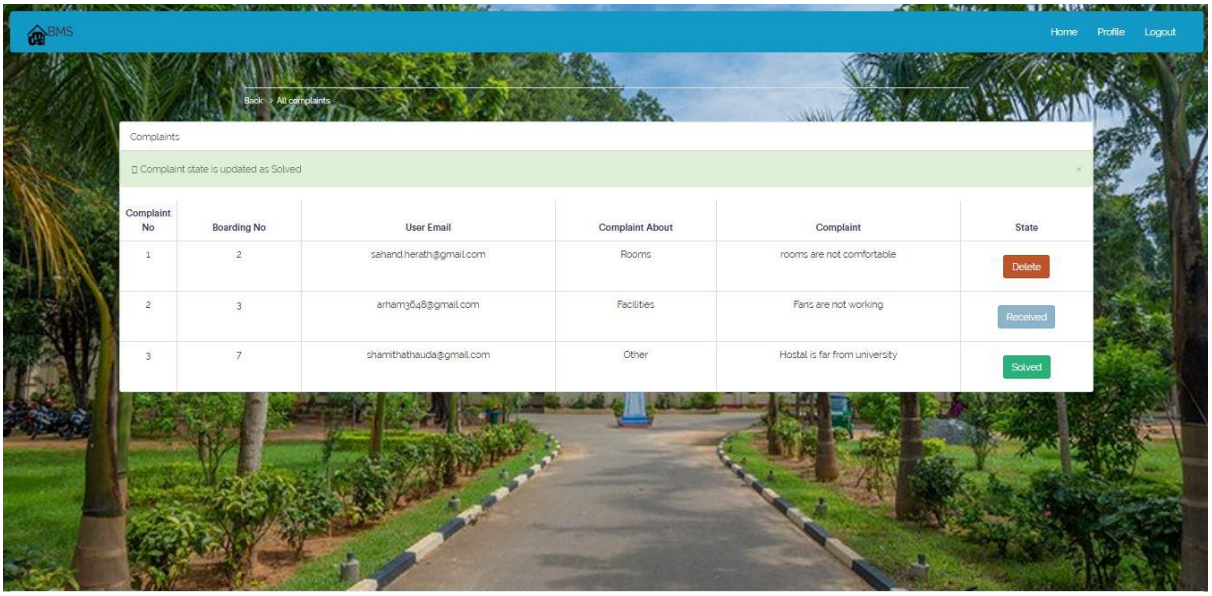


Figure 4-7- All complaints

❖ All Comments

- Admin can view all the comments and block any unwanted comments.

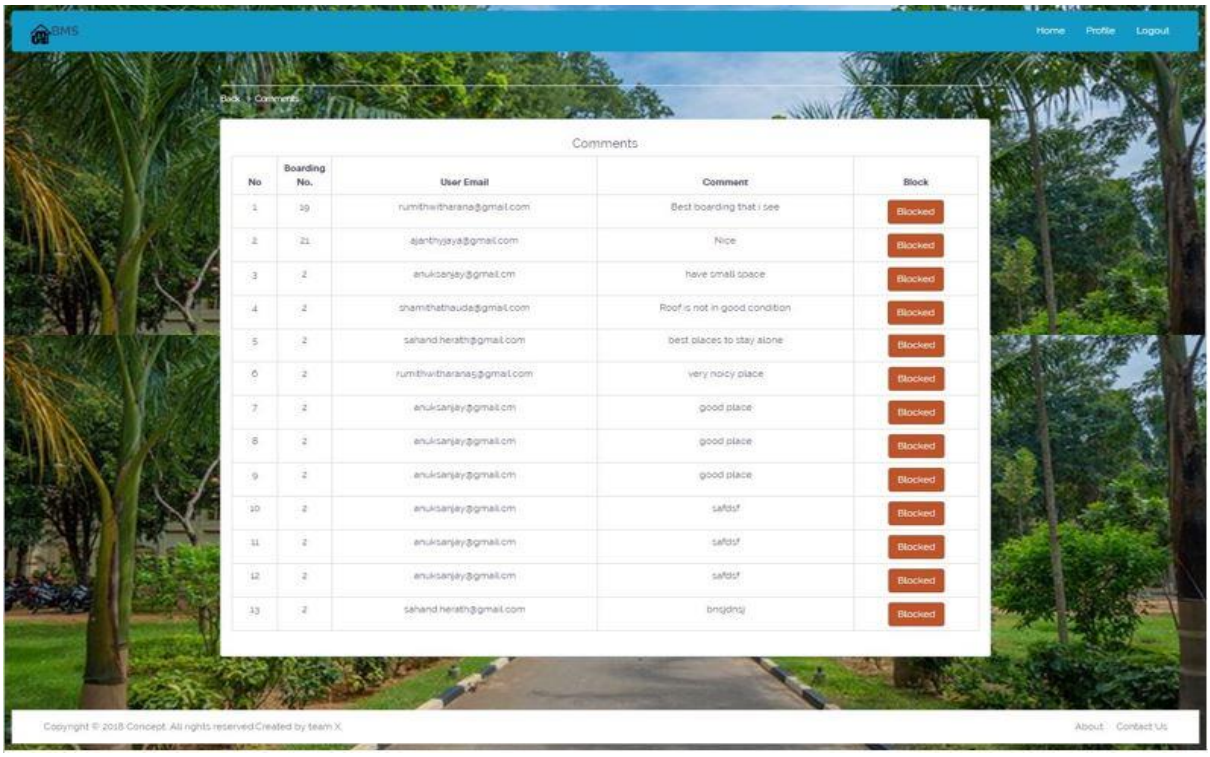


Figure 4-8- All comments

❖ Commented users

- All the commented users will be shown here. Admin can block/unblock any user according to his comments.

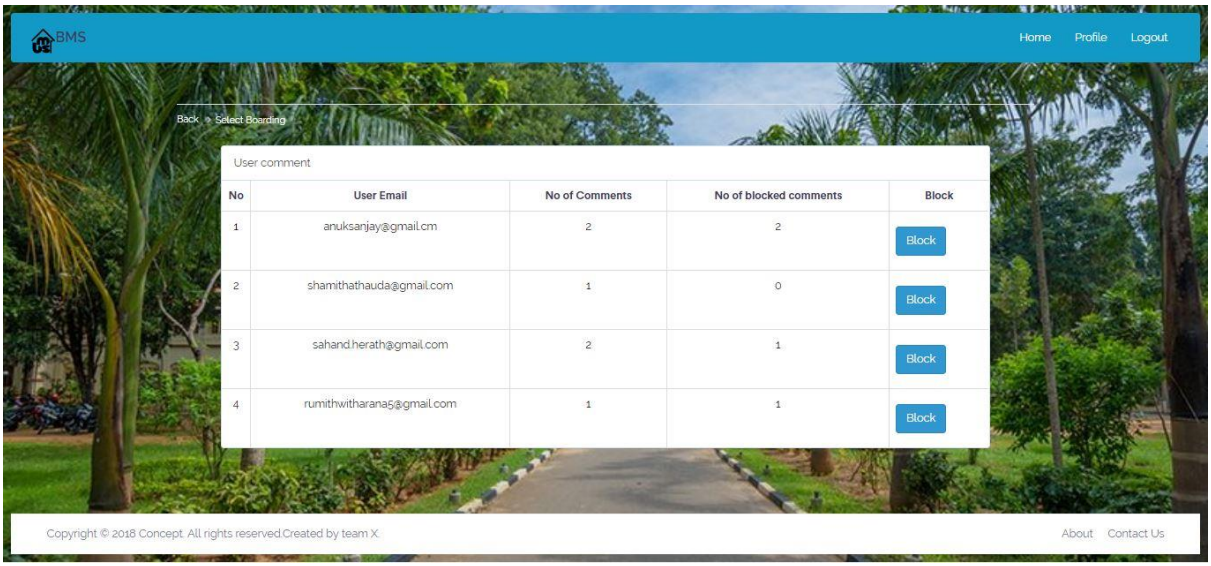


Figure 4-9- Commented users

❖ About us

- This include details of team members.

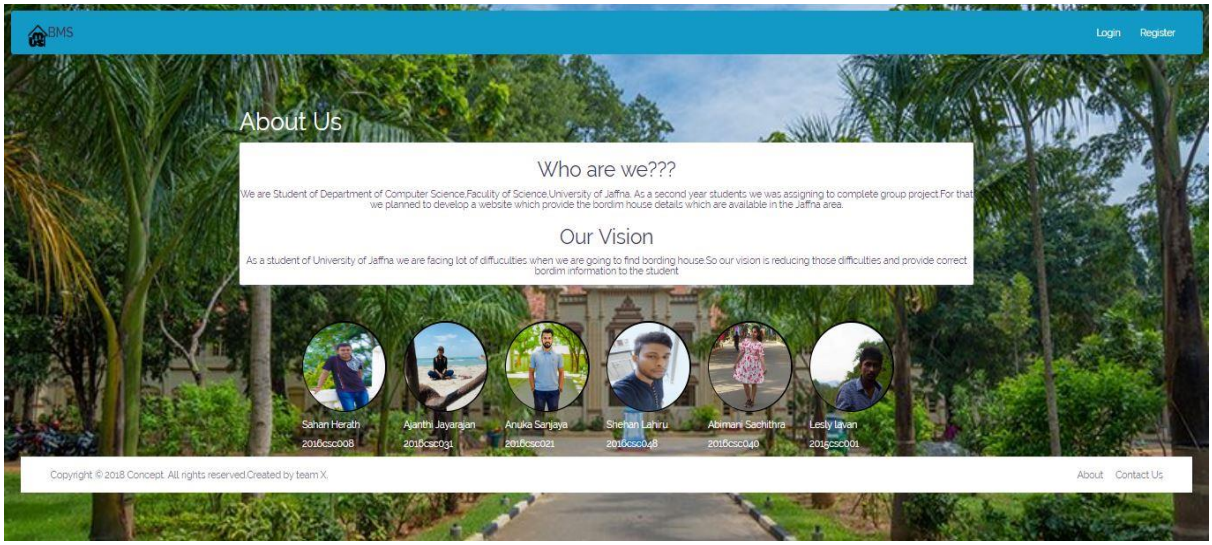


Figure 4-10- About us

❖ Admin profile and owner profile

- Admin has a profile. He can edit profile, change profile picture and deactivate account.

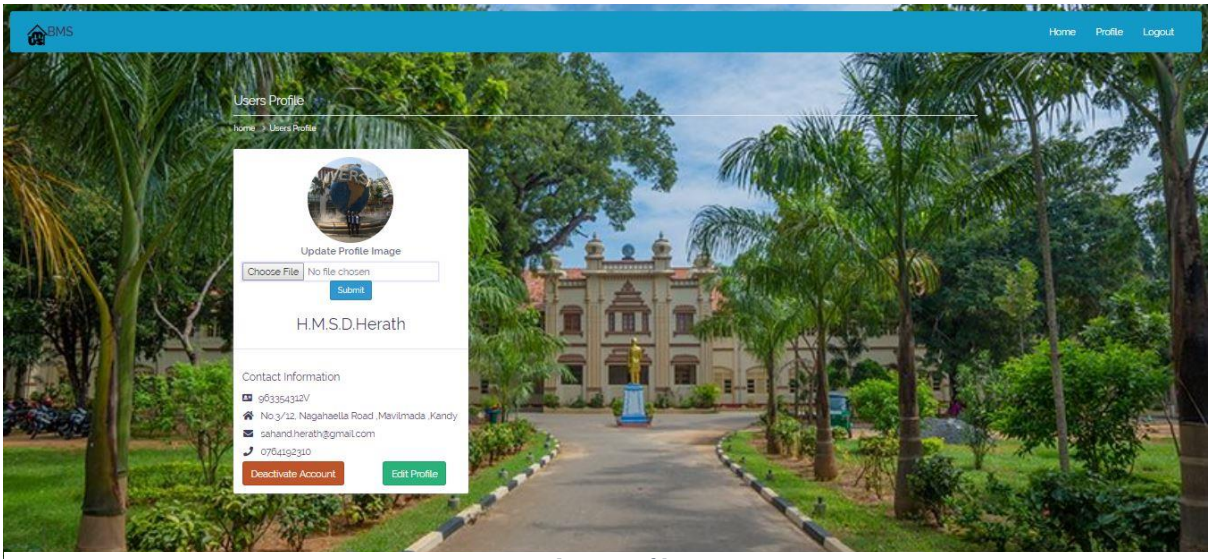


Figure 4-11- Admin profile

- Owner profile give many options such as add boarding, deactivate account, edit profile, edit boarding, view boarding, delete boarding, change profile picture, change boarding picture.

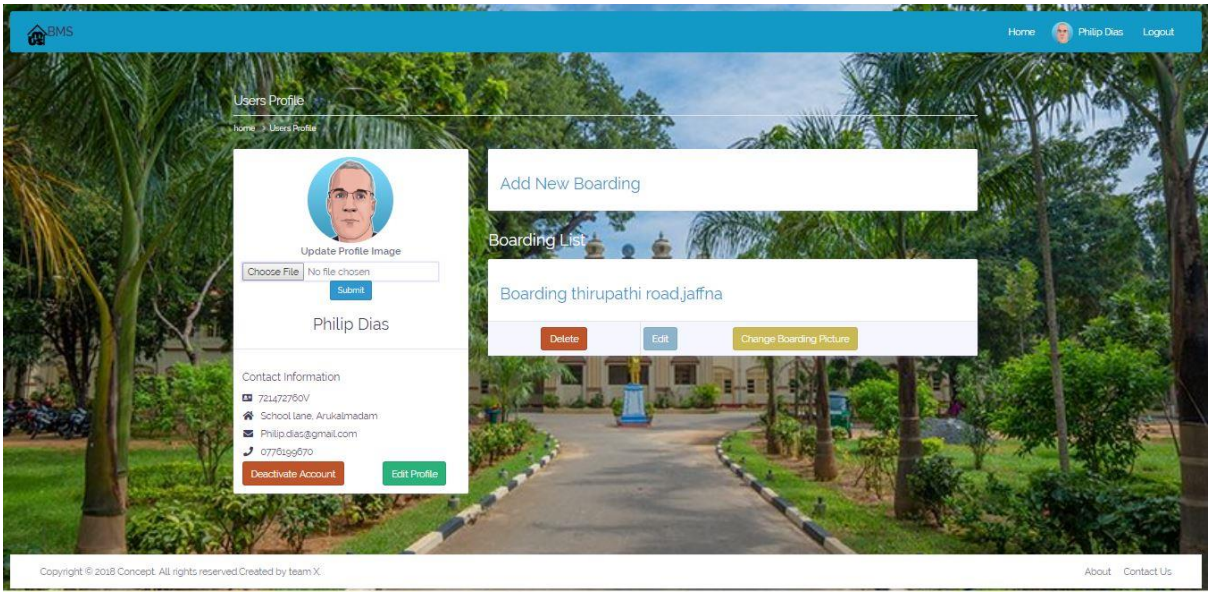


Figure 4-12- Owner profile

❖ The interface of the owner profile for admin

- Admin can view owner details and delete boarding by accessing through admin panel.

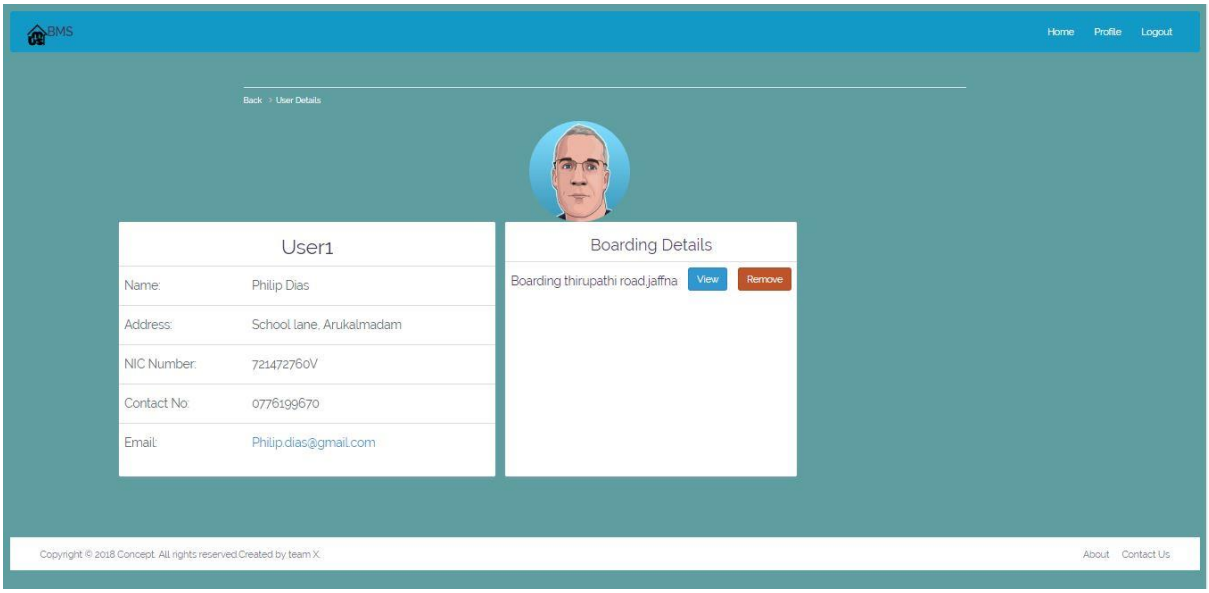


Figure 4-13- Owner Details admin view

❖ Owner's home page

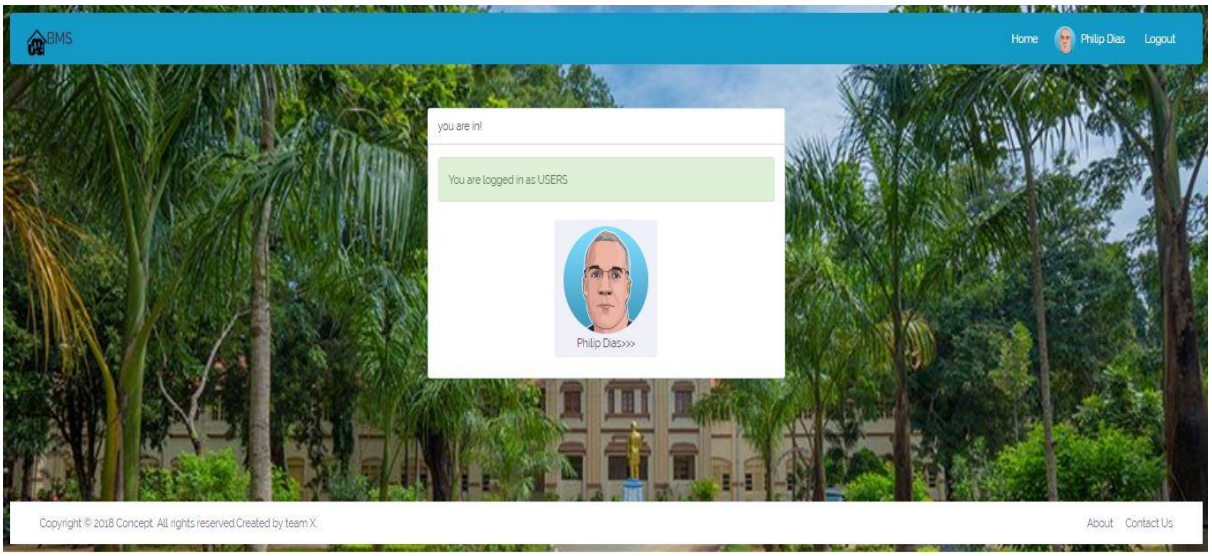


Figure 4-14- Owner home page

❖ Add boarding

- Owner can add boarding details using add boarding form. Several details related to boarding will be added here.

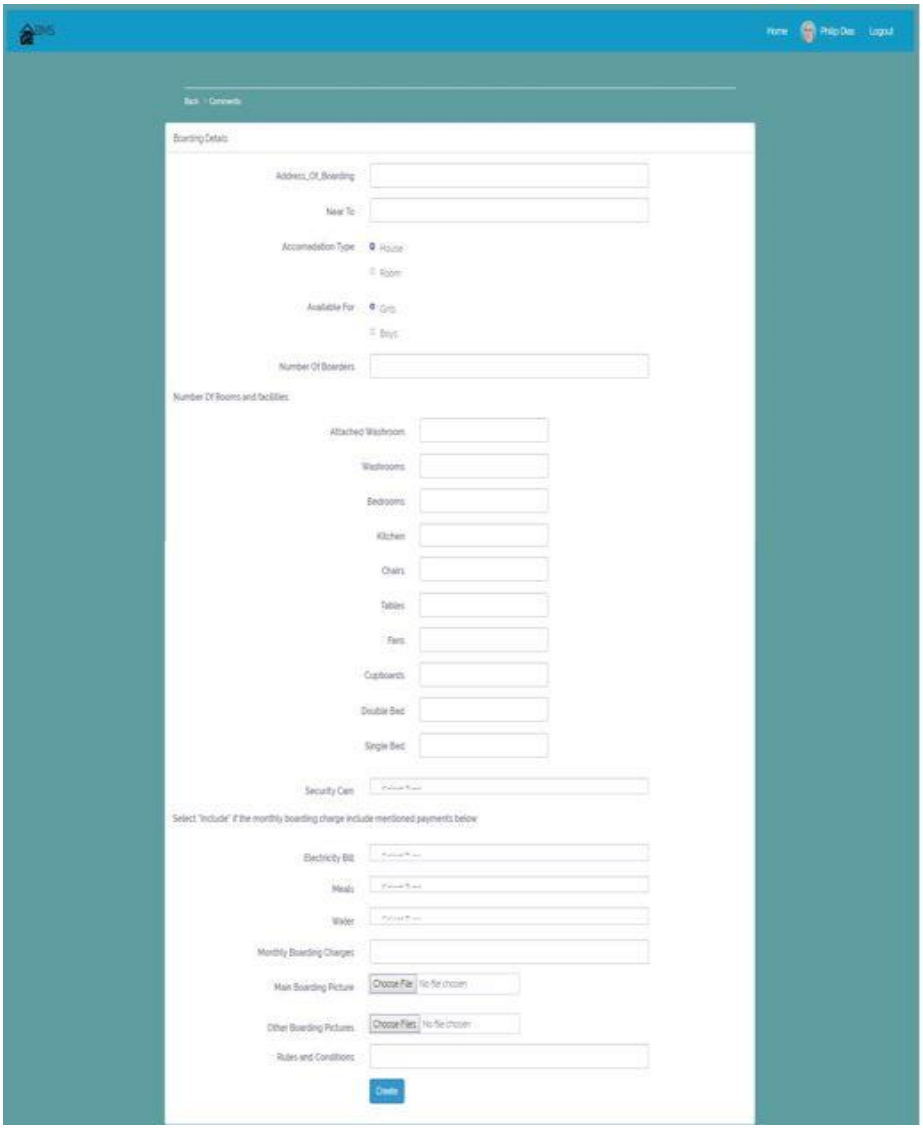


Figure 4-15- Add boarding

❖ **Edit profile and edit boarding**

- Owner can edit his/her profile details and also he can edit his/her boarding details.

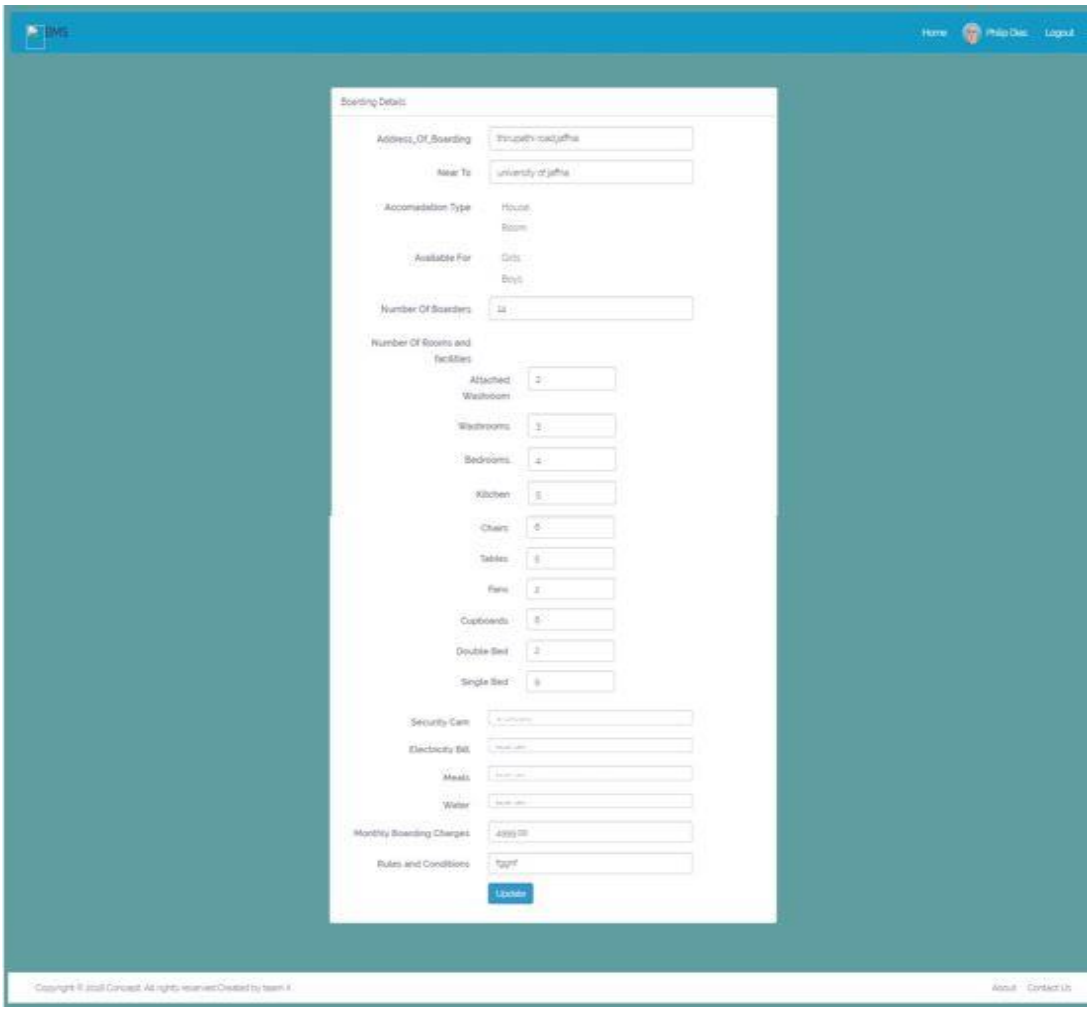


Figure 4-17- Edit boarding

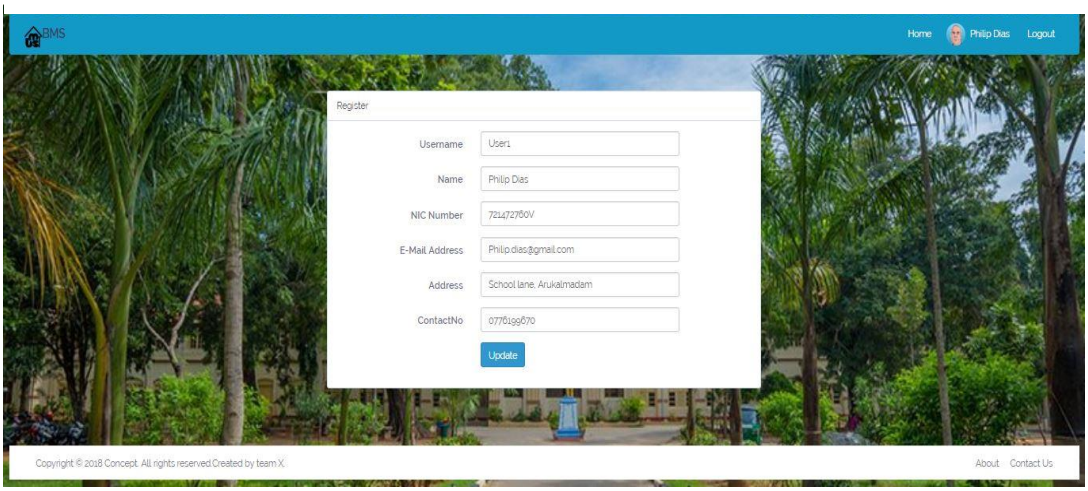


Figure 4-16- Edit profile

❖ Change boarding pictures

- Boarding owner can change boarding pictures of his/her boarding. He/she can change the main boarding picture and also other pictures.

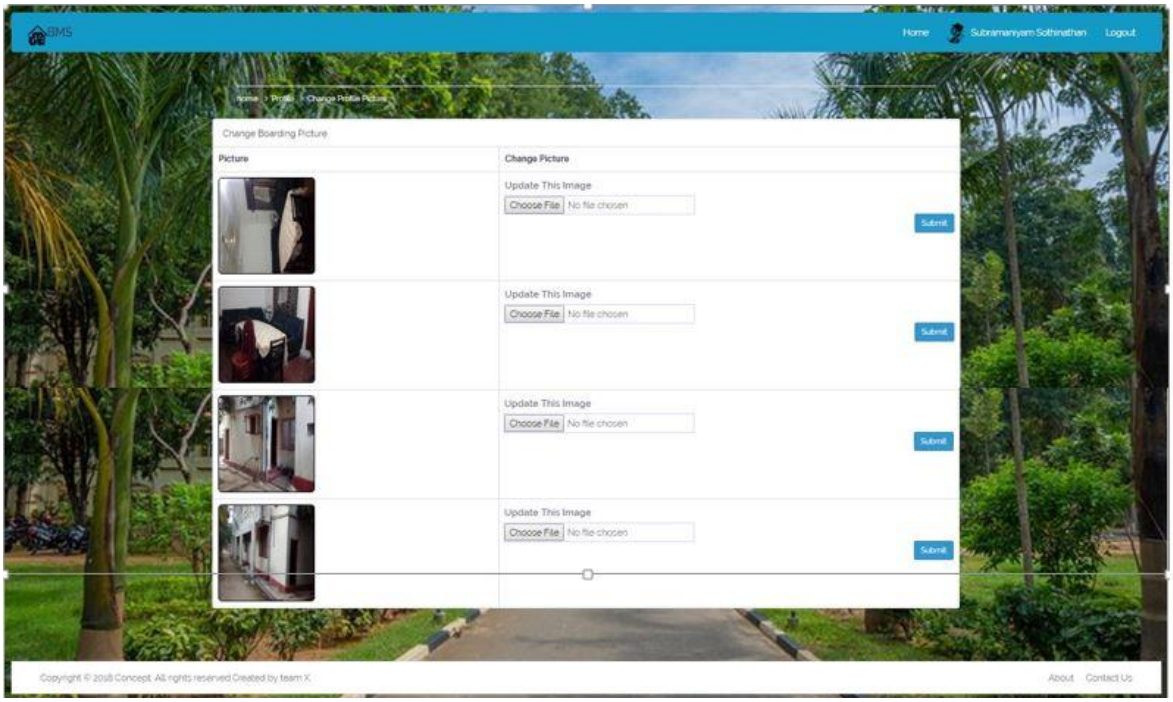


Figure 4-18- Change boarding pictures

❖ Contact us

- This gives information about how you can contact system developers.

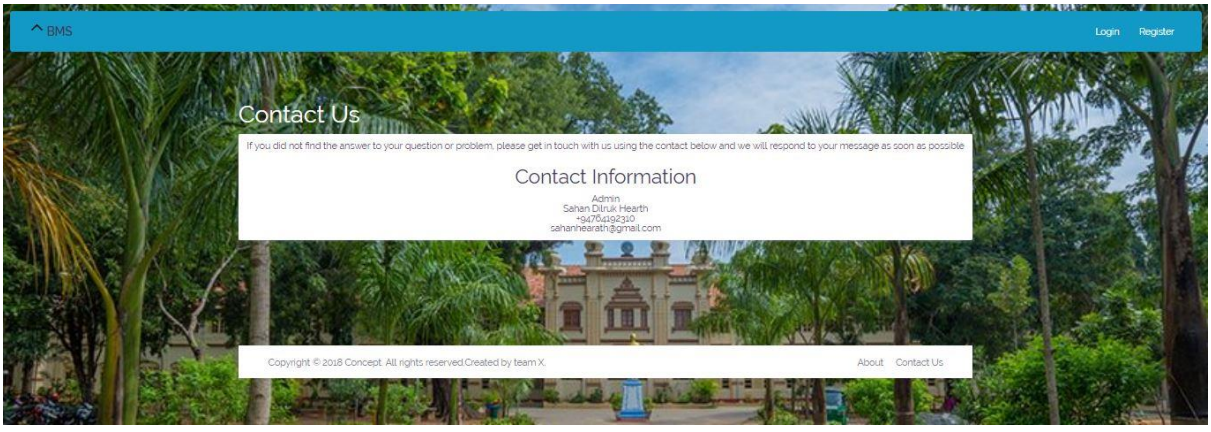


Figure 4-19-Contact us

❖ Select boarding

- All the boarding houses are displayed here with minimum details. Address of the boarding, boarding no, Accommodation type, Available for are displayed. Search bar is available for searching for a particular boarding.

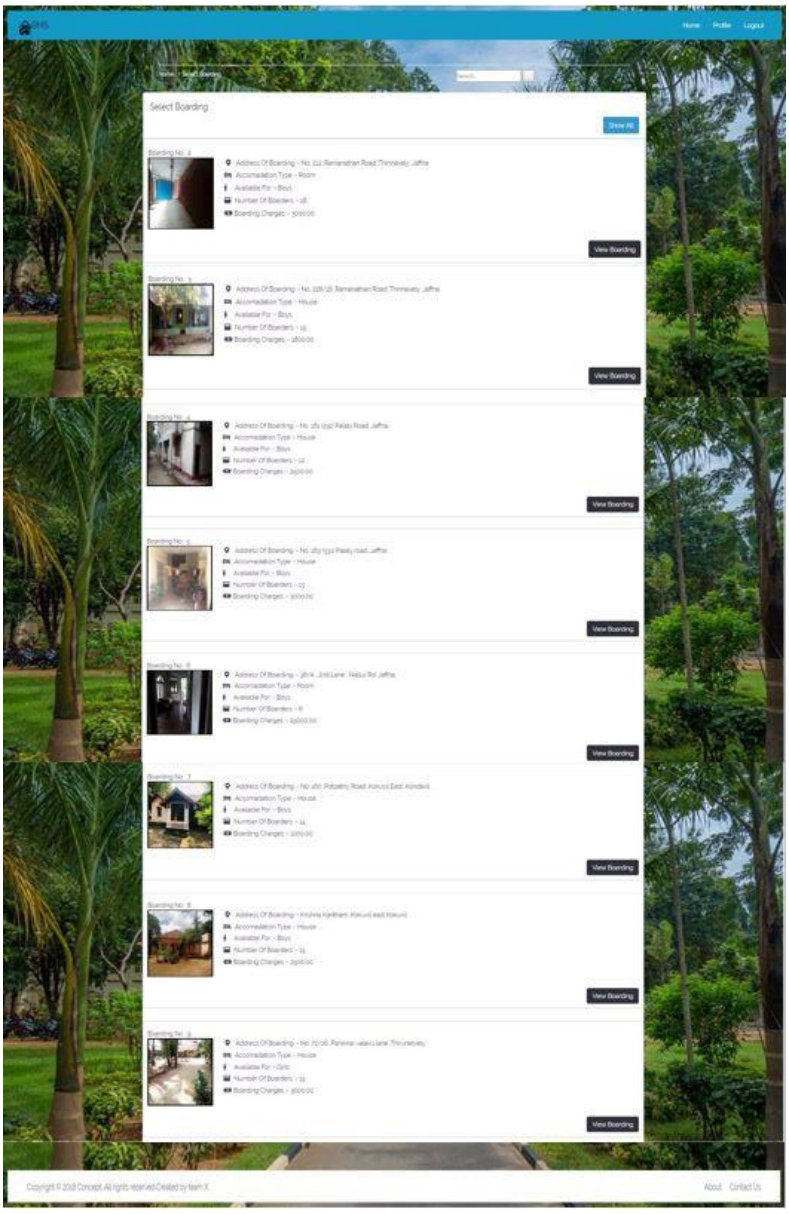


Figure 4-20- Select Boarding

❖ Boarding Details

- All the boarding details are displayed in here and user can comment on any boarding, user can complaint on any boarding and also he can rate them.

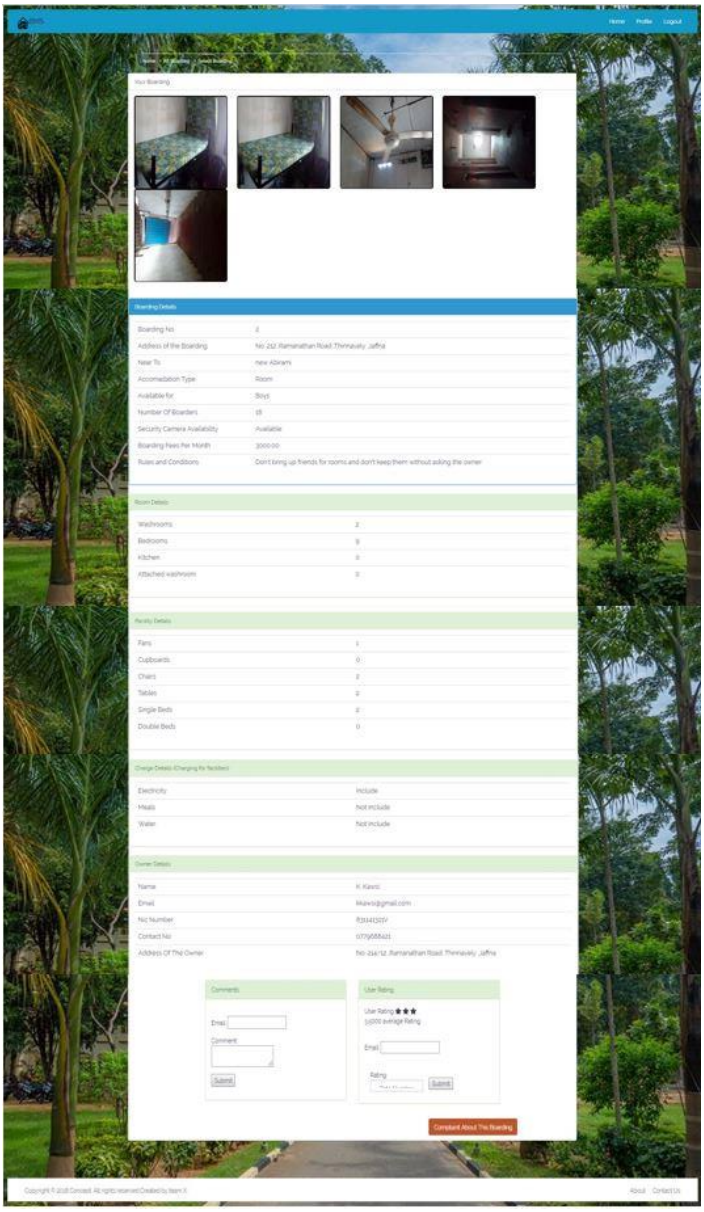


Figure 4-21- Boarding Details

❖ Find boarding

- This use to reduce the number of boarding according to your choice. If you have clear idea about what kind of boarding you are using then you can use this option.

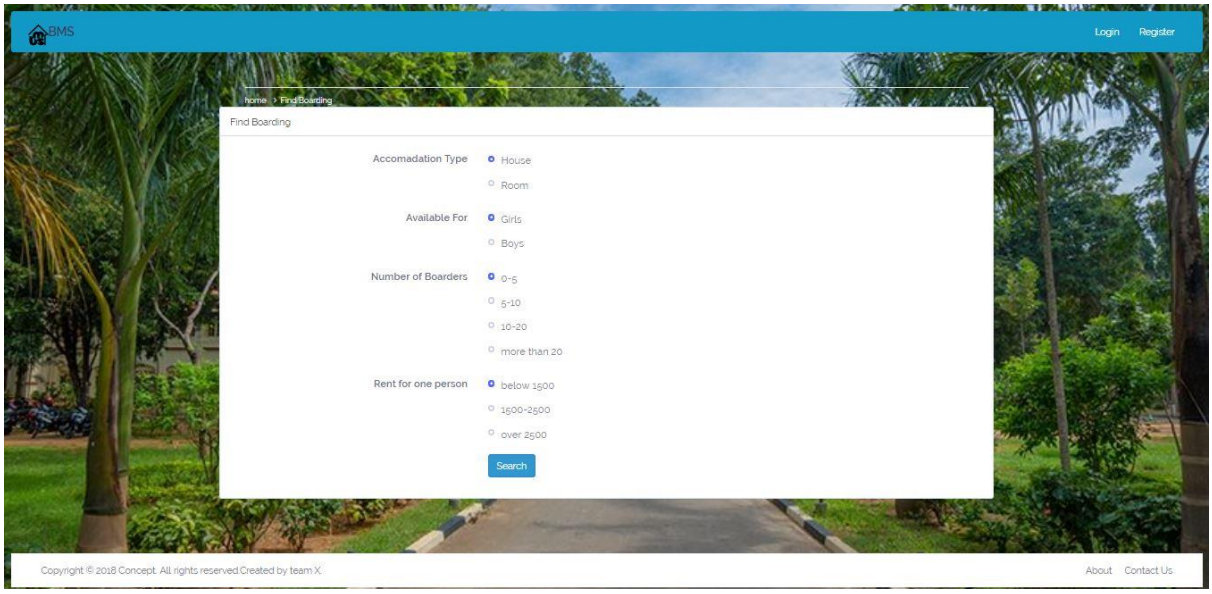


Figure 4-22- Find Boarding

❖ Complaint

- If user need to do a complaint he should use this form. User will be directed here through boarding details page.

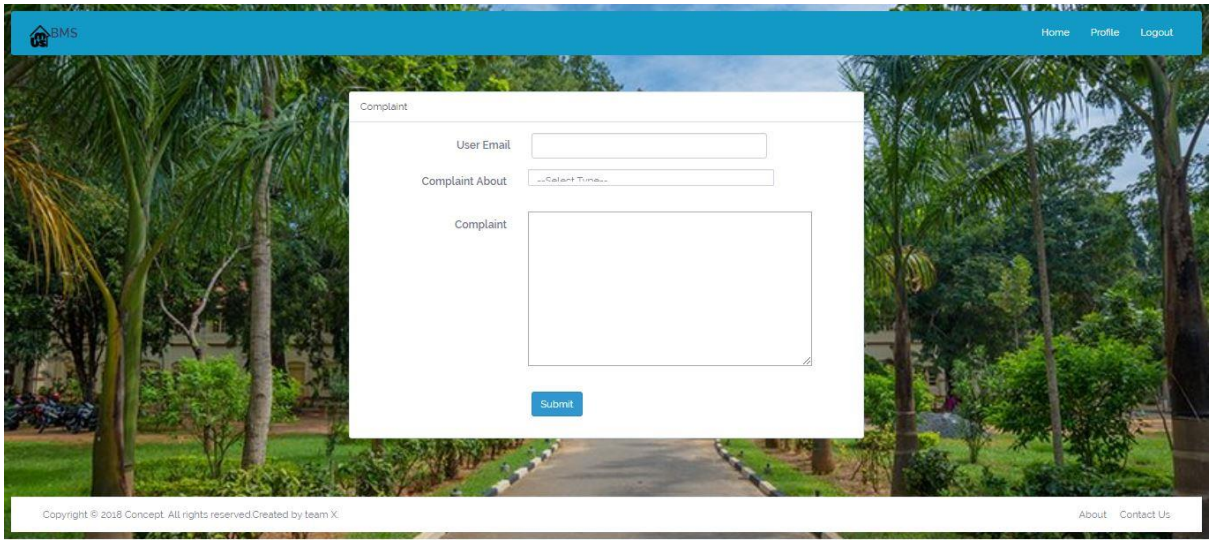


Figure 4-23- Complaint form

❖ **Confirmation and flash message**

- There are confirmation message to confirm the action and flash message to ensure that action is done.

Figure 4-24- Flash message

Figure 4-25- Confirmation message

❖ Email Responses

- Admin will review the complaints about boarding houses or owners and send Emails informing the complainer about reviewing the complaint and solving the complaint.
- Admin can block any user from commenting then those users will be informed by emails and those users can unblocked by the admin according to their replies.

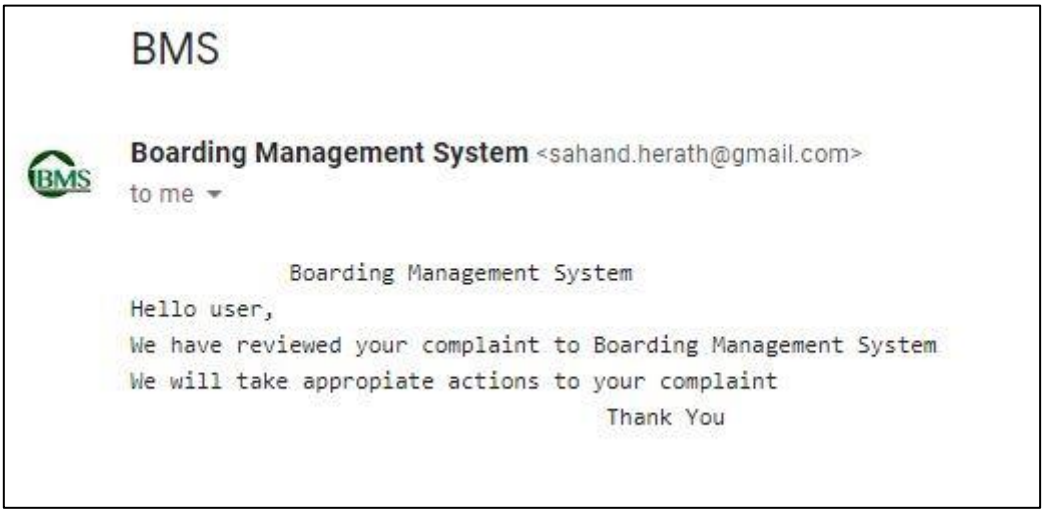


Figure 4-26- Email review

4.2 Special features

- Student can search specifically and filter boarding according to his/her choice. (e.g.: can search in a particular details)
- An admin can remove any owner, any boarding according to the complaints that he get.
- An admin can review the complaints and take the solutions for that.
- Admin can block user comments if those comments are include something unwanted.
- Admin can block any student if his comments are hateful or unwanted. Then on his/her comments won't be displayed in the system.
- Boarding owner can add boarding, change his profile picture, change boarding pictures, edit profile, edit boarding and deactivate his account.
- Student can rate the boarding and because of that they can identify the best boarding.
- User will get a mail when they blocked from commenting. He can send a reply email to admin to in order to get the commenting function back for him/her.
- Admin will be informed by a mail when he is removed from the system.
- System automatically counts the number of emails and number of blocked emails of particular user.

5 Testing

We assigned one special person for the testing purpose from our group and she was the responsible person for the testing. When we finished the first part of the system, she did the first testing process for check weather our system work properly or not. She gave some inputs and checked whether outputs are correct or not. After the first test we followed many testing methods for check our works.

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues. It was done by our developers themselves. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

Functional testing, a testing technique that is used to test the features/ functionality of the system or software, should cover all the scenarios including failure paths and boundary cases. Boarding details and owner details have been checked the system on create, edit, delete and update functions.

After completion of the fully system we did the **system testing** for evaluate our complete system. So after that we identified some bugs and errors and then we fixed those immediately.

Usability testing, a non-functional testing technique that is a measure of how easily the system can be used by end users. For that purpose, we selected a student and gave him the system for use. According to his feedback about user experience we did some modification to our system.

As end of the testing procedure we did the **acceptance testing**. Acceptance testing, a technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users. We have done the acceptance testing by showing the system to one of the owner and user (student). After all the testing we make some modification to our system and clear all the bugs and errors.

6 Discussion and Challenges

We faced lot of challenges while working with the project and as a team we were faced overcome with those challenges.

The major challenge occurred when we had gone to find a client. Actually we didn't have specific client for our project so we made university student as our client and made google form to collect requirements from them.

Next the biggest challenge occurred when we were went to collect details from house owners. Most of them had poor English knowledge. For that we made data collection form in Tamil language.

While we were working with our system we had to care about our academic activities to. So it was very difficult period. But we divided our works and that helped us to overcome it.

When we start our work we had less knowledge about Laravel. So we learned it from basic without any knowledge so it took long time.

End user's satisfaction and providing good user experience for the end user's is big challenge for the developers. So we worked hard for make a good user friendly system.

After finish the system we did some security checking. Laravel provide some default security features but we felt that it is not enough to provide a secured system. So we have add some additional security features.

Another big task was we had to manage our work within the given time period and to give maximum output.

Our project gives us an opportunity to work with team spirit. We were guided by the advices of our supervisor and mentors.

There are lots of experiences we get for a better outcome and those things help us to improve our knowledge.

6.1 Future Work

We are hoping to host our website in the near future. Mostly we cared about providing a good user experience to end user. We already have email system which send mails that when someone's account got blocked. Rather than that we are hoping to send confirmation email when new house owners registered to our system. This might be the future step to our email system.

6.2 Conclusion

In conclusion, as university student we have faced lot of difficulties when we have gone to find boarding houses. Our main goal is to avoid those difficulties from our **Boarding Management System**. Now student can easily visit our system and select house according to their needs and that owners can sell their houses without too much effort. There is no any benefit for us and this is done specially for our university students. So we hope this will be helpful for them.

7 References

Books: -

- **PHP Pandas : The PHP Programming Language for Everyone written by Dayle Rees.**

World Wide Web: -

- https://w3schools.com
- http://php.net
- https://www.tutorialspoint.com/mysql/index.htm
- https://dev.mysql.com/doc/refman/8.0/en/tutorial.html
- https://laravel.com/docs/5.5
- https://stackoverflow.com/
- https://www.quora.com/
- https://www.youtube.com/
- https://codeload.github.com/