Alaka Acharya

softwarica college of It and e-commerce

Hostel Finder

# Acknowledgement

I would like to thank all the wonderful people who have constantly provided me support and help throughout the project.

This project could never have been possible without our module leader, Mr. Sudip Lal Bajimaya. As a teacher and a mentor, he has taught me more than I could give him credit for. Your willingness to guide us through the bits of the project, supervise our work and provide feedbacks is highly appreciated.

Thank you to all the teachers, for being kind and patient with us while teaching us. I have been benefitted immensely by the things they have taught me and most of my knowledge have been used in shaping this project.

I would also like to express my gratitude to the chairperson, the principal, and all the founder members, for their continuous efforts has brought our college to such a reputed position in the education field and me and my friends, the students of this institute have got this amazing opportunity to be a part of this beautiful family where we get to learn new things every single day and strive to be the best students.

Thank you to IT club of Softwarica for conducting workshop on PHP framework and Mr. for guiding us throughout the workshop. It was during the workshop that I gathered so many concepts of PHP that were unknown to me before.

I wish to acknowledge the help provided by all of my friends, batch mates and class mates for sharing what they have learnt and providing me with criticisms and compliments continuously.

Finally, I wish to thank my family for their support and encouragement throughout my study.

# Abstract

Every year, there are thousands of student who move from their hometowns to a different city in search of better education and better opportunities. Many of those students choose to stay in a hostel rather than renting a room or an apartment. The process of finding a suitable hostels among so many of them found in every lanes of bigger cities is a very tedious job.

Considering this situation, ‘Hostel Finder’ is a website which helps you to browse through all the hostels in Nepal and view their details. This is also helpful for hostel owner who want to make their hostel visible to the students across Nepal. They can add their hostel in the system very easily.

The purpose of this document is to give an overview of the process I went through before actually developing the system. It contains details of analysis, design, implementation and testing phase which took place during the system development life cycle.

It is hoped that the system will be genuinely helpful to students seeking help and the document will be helpful to everyone who wants to understand the system thoroughly.

Contents

[Acknowledgement 1](#_Toc528226337)

[Abstract 2](#_Toc528226338)

[1. Introduction 7](#_Toc528226339)

[2. Analysis 8](#_Toc528226340)

[2.1 Introduction to Analysis 8](#_Toc528226341)

[2.2 Requirement Gathering Techniques 8](#_Toc528226342)

[Feasibility study 11](#_Toc528226343)

[2.4 Analysis methodology: Soft systems Methodology SSM): 12](#_Toc528226344)

[Rich picture 13](#_Toc528226345)

[Root definition 15](#_Toc528226346)

[Conceptual model 16](#_Toc528226347)

[2.5 System Requirement Specification 16](#_Toc528226348)

[2.5.1 What is SRS? 16](#_Toc528226349)

[2.5.2 Types of SRS 16](#_Toc528226350)

[2.5.3 Functional requirement specification 17](#_Toc528226351)

[2.5.4 Non-functional Requirements 18](#_Toc528226352)

[2.4 Use case diagram 19](#_Toc528226353)

[2.7 NLA 22](#_Toc528226354)

[2.8 Initial class diagram 23](#_Toc528226355)

[3. Design 24](#_Toc528226356)

[Structural Model 24](#_Toc528226357)

[Behavioral Model 26](#_Toc528226358)

[Database Model 30](#_Toc528226359)

[UI Design 31](#_Toc528226360)

[5. Testing 36](#_Toc528226361)

[6. Other project issues 39](#_Toc528226362)

[Issues 39](#_Toc528226363)

[Evaluation of own work 40](#_Toc528226364)

[Future work 40](#_Toc528226365)

[Limitations of my project 40](#_Toc528226366)

[User Manual 40](#_Toc528226367)

[Registration 41](#_Toc528226368)

[Logging In 42](#_Toc528226369)

[Adding your hostel 43](#_Toc528226370)

[Searching 46](#_Toc528226371)

[Booking 47](#_Toc528226372)

[Leave us a review 49](#_Toc528226373)

[FAQs 51](#_Toc528226374)

[7. Conclusion 52](#_Toc528226375)

[8. References and Bibliography 53](#_Toc528226376)

[9. Appendix 54](#_Toc528226377)

**Table of Figures**

[Figure 1: Rich picture diagram 14](#_Toc528240332)

[Figure 2: Conceptual Model 15](#_Toc528240333)

[Figure 3: Use case of Admin 20](#_Toc528240334)

[Figure 4: Use case of Hostelers 21](#_Toc528240335)

[Figure 5: Flowchart 26](#_Toc528240336)

[Figure 6: Activity Diagram 27](#_Toc528240337)

[Figure 7: Sequence diagram 28](#_Toc528240338)

[Figure 8: Sequence diagram 29](#_Toc528240339)

[Figure 9: UI Model-Home 32](#_Toc528240340)

[Figure 10: UI model: vacancies 33](#_Toc528240341)

[Figure 11: UI model-list 33](#_Toc528240342)

[Figure 12: UI model-add hostel 34](#_Toc528240343)

[Figure 13: UI model-book 35](#_Toc528240344)

[Figure 14: UI model-register 35](#_Toc528240345)

[Figure 15: addhostel.php 56](#_Toc528240346)

[Figure 16:details.php 58](#_Toc528240347)

[Figure 17:book.php 61](#_Toc528240348)

[Figure 18: demo.css 67](#_Toc528240349)

[Figure 19:error.php 68](#_Toc528240350)

[Figure 20:help.php 70](#_Toc528240351)

[Figure 21:cpindex.php 73](#_Toc528240352)

[Figure 22: loginn.php 74](#_Toc528240353)

[Figure 23: nav.php 76](#_Toc528240354)

[Figure 24:search.php 77](#_Toc528240355)

[Figure 25: searchtwo.php 77](#_Toc528240356)

[Figure 26: server.php 79](#_Toc528240357)

[Figure 27:serveradd.php 80](#_Toc528240358)

[Figure 28: serverbook.php 81](#_Toc528240359)

[Figure 29: signup.php 82](#_Toc528240360)

[Figure 30:suggestion.php 84](#_Toc528240361)

[Figure 31:vacancies.php 85](#_Toc528240362)

# Keywords and Abbreviations

ERD- Entity Relationship Diagram

SDLC- System Development Lifecycle

SSM- Soft systems Methodologies

|  |  |
| --- | --- |
| **Keyword** | **Meaning** |
| Analysis | Systematic examination of the system |
| Design | Blueprint of the system |
| Black-box testing | Testing output of system by providing input against the expected output |
| Feasibility | Compatibility of system with constraints |
| Vacancy | Rooms available in a hostel |

# Introduction

My project is titled ‘Hostel finder’ as it helps in searching hostels.

**Background of the system**

Finding a hostel in big cities is definitely a tedious job, it takes a lot of time and effort and still might not give us the best results. Hostel finder automates this work as all the hostels in specific area are brought together. The main objective behind this project is to solve the problem of students across Nepal.

**Overview of the design**

The system is designed in such a way that it proves to be helpful for students as well as hostel owners. While, students get to browse through a list of hostels and book them, owners get to add their hostel to the system so that as many students as possible can view them.

**Aims and objective**

The first and foremost aim of the system is to automate the tedious task of hostel hunting in cities. Along with that, there are many others things we are aiming at via this project which are listed below:

* To create an online portal for assisting students to search a suitable hostel for them.
* To create a platform for hostel owners to advertise and update students about their services.
* To provide information about all the hostels across the country in a single platform
* Create an easier way to inform clients about vacancies.
* To alert hostels owner to maintain their standard via regular reviewing

Academically, I am expecting to fulfill the following objectives:

* Implement the php knowledge learnt from previous modules
* Develop a problem solving spirit.

# Analysis

## 2.1 Introduction to Analysis

**What is Analysis?**

Analysis is the first phase of the system development lifecycle. Analysis is an interpretive process, breakdown of a system into its organized components to gain an insight about what works with what, what causes success or failure of something. This process focuses on critically viewing the system, defining the methodology used to build a system, taking feasibility study.

**Why analysis?**

System analysis is important in that it provides an avenue for solutions in the system through the various tasks involved in the analysis process.

* Better management and controls
* Risk Management
* Minimizing errors

[(FTIAS, 2018)](#requirement)

## 2.2 Requirement Gathering Techniques

These are the techniques to do market research to understand needs of customers and weakness of competitors.

* Questionnaires
* Survey
* Focus Group
* Brainstorming
* Observation
* Reverse Engineering
* Interview

[(Brighthub Project Management, 2018)](#requirement)

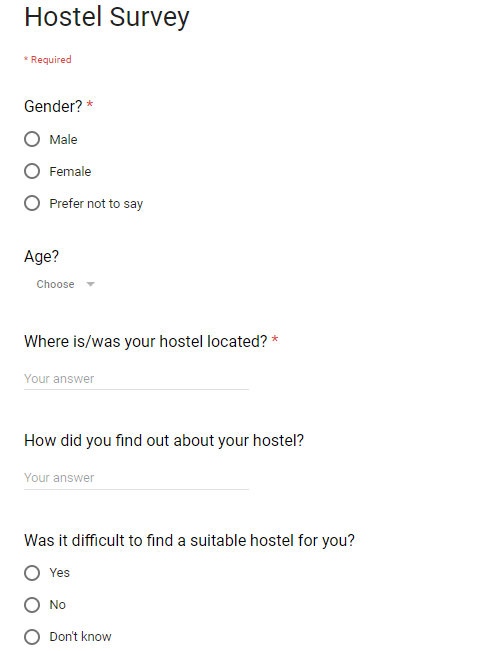
These are the techniques I focused on for requirement gathering.

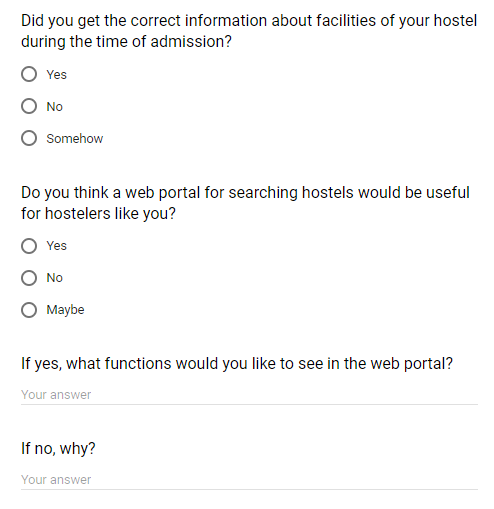
**Brainstorming**:

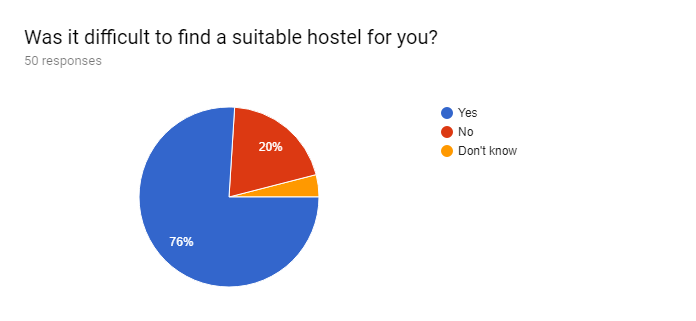
Brainstorming combines a relaxed, informal approach to problem solving with lateral thinking. Since this is an individual project, I started requirement gathering process by brainstorming, doing subtle research via internet and books. Individual brainstorming makes us think out of the box and discovers various discreet possibilities, generates creative solutions to the problem.

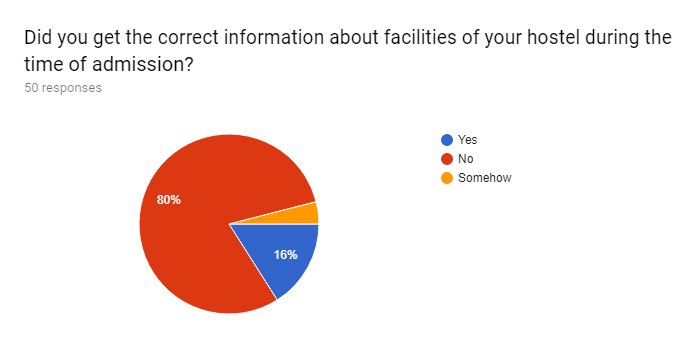
**Questionnaires:**

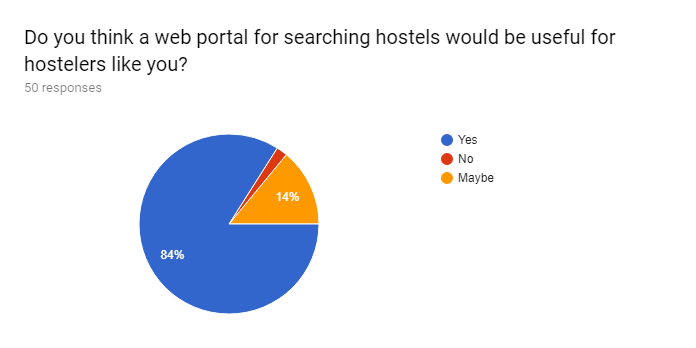
It is difficult to reach among individual audience and have one-to-one interviews with them. Questionnaires manage to gather answers from a larger audience at once and the responses can be sorted out to draw a simple conclusion from there.

I conducted a survey among 50 hostelers where they were asked some questions via google forms. I analyzed the feedbacks I got which helped me to make some important decisions. I have all the questions asked along with an overview of the analysis.









The charts prove that finding a hostel is indeed a tedious work to do and most of them agree that a web portal would be very easy for people like them.

## Feasibility study

Feasibility study is carried out to know whether or not a project is possible under certain circumstances. I performed the following studies to determine whether or not my project is feasible to be carried out.

* + 1. **Technical feasibility**: Technical feasibility is the procedure of determining whether the system can be developed using the available technical resources. Experiences available to use the technology, difficulty level is also determined. All these topics are covered under technical feasibility study.

This project is technically feasible as all the hardware and software necessities is already available to me. Also, a computer system with internet connection, which is vital for all the users is generally available very easily to most of the people now a days.

* + 1. **Schedule feasibility:** It is the likelihood of a project being completed on a provided time frame or within a deadline.

My project passes this feasibility test as it is bound to complete on time which is depicted via Gantt chart and WBS. This is an academic project which has to be completed within 3 months and there’s no possibility of exceptions. So, every steps of the system development will complete within provided time frame.

* + 1. **Economic feasibility:** Includes study of investments and revenues of a project to determine whether or not it is economically profiting project.

My project is made for academic purposes which doesn’t require a lot of financing and investing to be done at the moment. So, it is safe to say that it is economically feasible.

* + 1. **Cultural feasibility:** It includes validation of the system against the people and their cultures. If the system has negative impact on local and general culture, it is very likely to fail.

The project “Hostel finder” respects the sentiments of students and their parents and tries to solve the problems they face. It is not culturally offensive or questionable in any way.

* + 1. **Legal/ethical feasibility:** It defines legal implications and ethical considerations of the project. Determines whether the proposed system conflicts with legal requirements, e.g. a Data Processing system must comply with the local Data Protection Acts.

There is no such aspect in the system which conflicts with legal considerations. Similarly, the project is in compliance with ethical boundaries of the society and users. No such harm is intended.

[(Brighthub Project Management, 2018)](#feasible)

## 2.4 Analysis methodology: Soft systems Methodology (SSM):

SSM attempts to learn and appreciate the problem situations between the groups of stakeholders rather than set out to solve a problem that is pre-defined (Huaxia). SSM is a people focused approach to information system analysis that works in seven steps:

* Enter situation considered problematical
* Express the problem situation
* Formulate root definitions of relevant systems of purposeful activity
* Build conceptual models of the systems named in the root definitions
* Compare models with real world situations
* Define possible changes which are both possible and feasible
* Act to improve the problem situation. [(UKEssays, 2018)](#ssm)

. **Features and advantages of SSM are:**

* Human activity is modelled.
* Enables input of user knowledge and skills
* Offers flexibility in the approach
* Searches for solution which is more than just technical
* Open discussions
* Doesn’t need professional practicing.

**Disadvantages of SSM:**

* May not be appropriate for complex systems and large organization
* Can be difficult to manage

**Why is it appropriate for this project?**

* We’re dealing with a situation that involves human problems.
* It is a small-scale project done individually.
* User, social, cultural issues need to be considered.

### Rich picture

Rich Pictures are comprehensive, graphic (habitually hand-drawn) illustrations of what are stereotypically composite and ill-defined systems. It is an extremely simple way to conceptualize the scope/context of a system we are analyzing. A rich picture of how the system works originally is included below

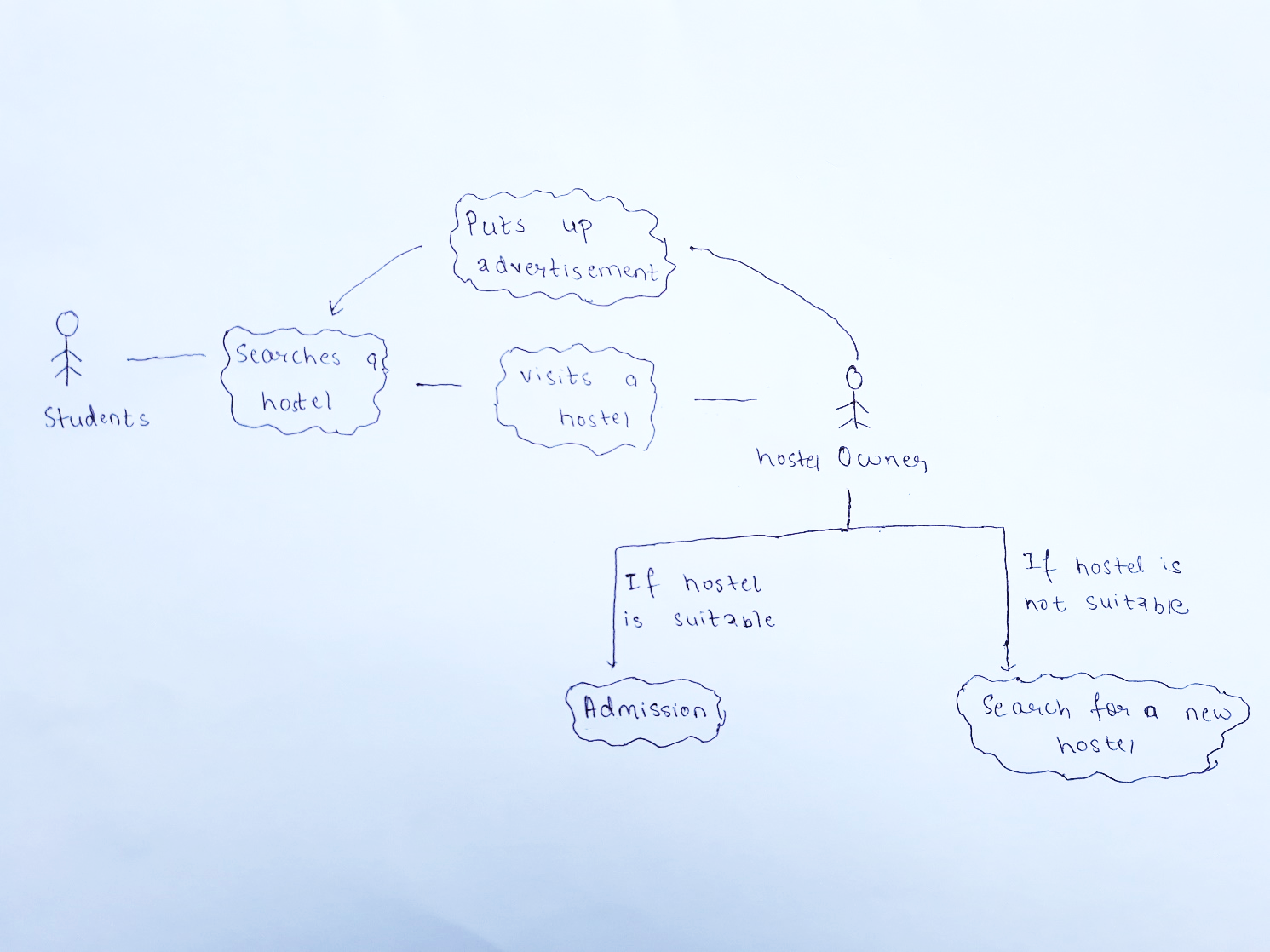


Figure 1: Rich picture diagram

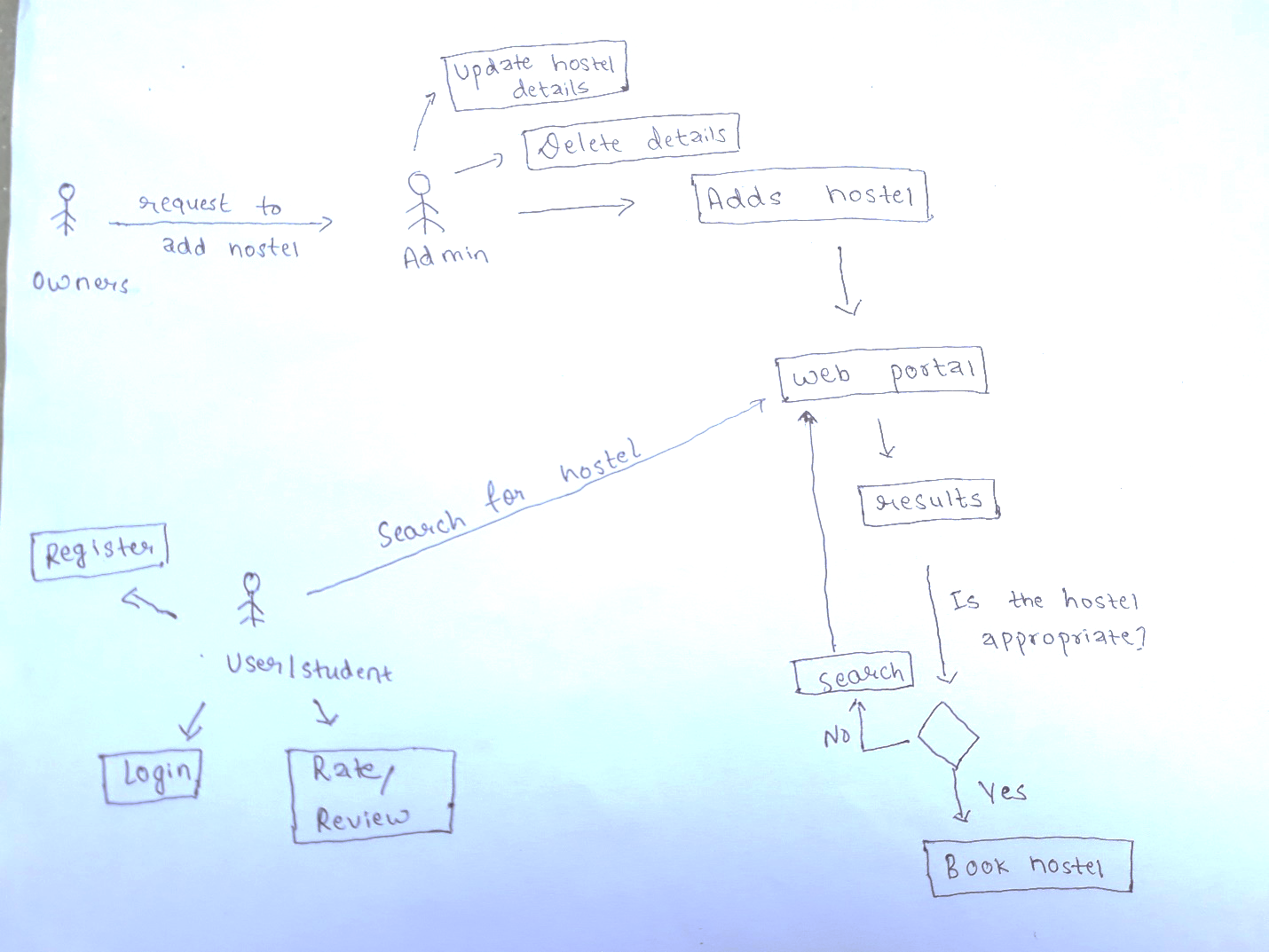


Figure 2: Conceptual Model

### Root definition

A Root Definition is a structured description of a system. It is a clear statement of activities which take place (or might take place) in the organization being studied.

Root definitions are powerful ways of building systems models which can help get to the heart of the problem issue. A powerful mnemonic – CATWOE – helps create root definitions which need to take account of and spell out: (https://businesschange.co.za/what-is-catwoe-analysis/)

|  |  |  |
| --- | --- | --- |
|  | Details | System overview |
| Clients | Who is the system operated for? | Hostelers/Students/Parents |
| Actors | Who is responsible for performing this transformation? | System admin |
| Transformation | What transformation will this system bring? | Making hostel searching process automatic and web based. |
| Worldview | What makes this transformation worthwhile? What is the bigger picture? | Reducing the mental turmoil migrant students face while searching for accommodation. |
| Owner | Who is authorized to make changes happen with the system? | System admin |
| Environment constraint | What constraints might have impact upon functioning of this system? | * No internet availability to many students * Less advertising * Financial constraints |

### Conceptual model

It is a composition of concepts involved in a system, used to represent a complex system in a simpler way.

Comparison between old and new system

|  |  |  |
| --- | --- | --- |
|  | Old system | New system |
| 1 | Students have to rely upon adverts or recommendation to search hostels | Students can browse the web portal to search hostels |
| 2 | Students have to visit hostel to find out about their features | Students will get information from the web portal |
| 3 | Student don’t get provision of advance booking or such. | There is a provision of advanced booking |
| 4 | It’s hectic to visit hostels one by one and choose the best one for you | It’s easier as there’s a ranking system. |

## 2.5 System Requirement Specification

### 2.5.1 What is SRS?

A software requirements specification (SRS) is a description of a software system to be developed. It labels functional and non-functional requirements, describes features and behaviors of the system. SRS allows developers to be clear on the goals of the software and on what they should focus on. Furthermore, it allows them to:

* Save time on communication
* Minimize development efforts
* Gives the customer feedback
* Eliminate task duplication
* Facilitate the transfer to new users or to new machines
* Breaks problems down into parts
* Serves as the main document to verify the validation and testing processes

### 2.5.2 Types of SRS

Functional requirement specification:

**Functional requirement specification:** Describes a particular behavior of function of the system when certain conditions are met.

**Non-functional requirement specification:** Describes how a system should behave and what limits there are on its functionality. It lists its quality and attributes.

[(Junaid Hassan's Technical blog, 2018)](#srs)

## 2.5.3 Functional requirement specification

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ID | Functions | Data | Descriptions | Purpose |
| FR1 | Register as a user | First Name, last Name,  Email,  hostel name | Users can create their accounts | To get hostel owners connected in the app |
| FR2 | Login as a user | Username/email, password | Registered users can login to the system | To fully access the app |
| FR3 | Add hostel | Hostel name, hostel location, prices | Owners can add details to their hostels | To add new details to the hostel |
| FR4 | Update hostel details | Hostel details | Details of the hostel can be updated by the owner. | To cope with the changes made in the hostel |
| FR5 | Add pictures | Pictures of the hostel | Pictures can be added | To see how hostel looks like |
| FR6 | Delete hostel details | - | Details can be deleted. | To delete the features of a hostel |
| FR7 | Rate a hostel | - | Students can rate a hostel they’ve lived in. | To compare hostels |
| FR8 | Leave a review on a hostel | Feedbacks and reviews of hostelers | Students can leave comments about the hostels they’ve lived in. | To get vies of hostelers on how good or bad is a hostel |
| FR9 | Search hostels by location | Location | Hostels can be searched based on the location. | To find hostels when you’re completely unknown about hostels in your area |
| FR10 | Search hostels by name | Hostel name | Hostel can be searched by typing their name | To find the hostels whose name you already know |
| FR11 | Notify about vacancies via SMS | - | Students seeking for vacancies will be notified via SMS when vacancy is available. | To be informed when the hostel you want to live in has vacancies |
| FR12 | Notification about vacancies in the app | - | Students seeking for vacancies will be notified in their account when vacancy is available. | To be informed when the hostel you want to live in has vacancies |
| FR13 | Display latest vacancies | Hostel details, vacancy amount | New vacancies will be displayed in the front page. | Provide information about vacancies |
| FR14 | Book hostels | Details of hostelers | Students willing to join a hostel can book when there is a vacancy. | Make it easier to students |
| FR15 | Remove booked vacancies | - | When vacancies are occupied, they are removed | To make vacancies information managed |
| FR16 | Hostel ranking | - | Hostels with highest ratings will be ranked as top 10 | To encourage hostels to be the best. |
| FR17 | Visitor count | - | Number of visitors who have visited the site | To review the popularity of the app |
| FR18 | Alert notifications | - | Inform hostel owners when their hostel if getting poor ratings | Encourage owners to constantly improve |
| FR19 | Delete user | - | Remove users when they are no longer part of the group | To remove users who are no longer part of the group |
| FR20 | User manual | FAQ, Help, Glossary | Guide to the user on how to use the system. | To make sure users know how to use the system. |

## 2.5.4 Non-functional Requirements

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Requirement | Details | Purpose |
| NF1 | Security | Protection of the system and user data | To secure data |
| NF2 | Reliability | Users should have the trust in the system even after using it for a long time. | To maintain user trust |
| NF3 | Availability | App is available and running every time | For easy access to the system every time. |
| NF4 | Usability | Navigation is easy to use for everyone | To make is easy to use for everyone |
| NF5 | Scalability |  |  |
| NF6 | Recoverability | Data can be recovered if ever erased | To recover valuable data |
| NF7 | Manageability |  |  |
| NF8 | Portability | Ability to be moved from one platform to another | To be able to use it in different devices |
| NF9 | Appearance | Aesthetics of the application | To make it visually pleasing |
| NF10 | Speed | Speed of loading and running | To save time |
| NF11 | Accuracy | Data should be accurate | To provide the right information |

**2.5.4 MoSCoW Prioritization**

Moscow prioritization is a technique that helps to understand and manage the priority of each task within the provided deadline. Here MoSCoW stands for:

|  |  |
| --- | --- |
| M-Must have | Minimum Usable Subset of requirement which project guarantees to deliver. |
| S-Should have | Important but not vital. Workaround is possible. |
| C-Could have | Valuable but no problem if dropped. |
| W-Won’t have | Won’t have this time |

[(Agile Business Consortium, 2018)](#moscow)

|  |  |  |  |
| --- | --- | --- | --- |
| ID | Functions | Priority | Dependency |
| FR1 | Register as a user | M |  |
| FR2 | Login as a user | M | FR3 |
| FR3 | Add hostel | M | FR2 |
| FR4 | Update hostel details | M | FR2, FR5 |
| FR5 | Add pictures | M | FR2, FR5 |
| FR6 | Delete hostel details | M | FR2, FR5 |
| FR7 | Rate a hostel | M | FR3, FR4, FR5 |
| FR8 | Leave a review on a hostel | C | FR3, FR4, FR5, FR9 |
| FR9 | Search hostels by location | S | FR3 |
| FR10 | Search hostels by name | S | FR3 |
| FR11 | Notify about vacancies via SMS | M | FR3, FR4 |
| FR12 | Notification about vacancies in the app | M | FR3, FR4, |
| FR13 | Display latest vacancies | M |  |
| FR14 | Book hostels | M |  |
| FR15 | Remove booked vacancies | S |  |
| FR16 | Hostel ranking | S |  |
| FR17 | Visitor count | W |  |
| FR18 | Alert notifications | S |  |
| FR19 | Delete user | S | FR1, FR2, FR3, FR4 |
| FR20 | User Manual | M |  |
|  |  |  |  |
| NF1 | Security | M | FR1, FR2, FR3, FR4 |
| NF2 | Reliability | M | FR1, FR2, FR3, FR4, FR11, FR12 |
| NF3 | Availability | M | FR22 |
| NF4 | Usability | S | ALL |
| NF5 | Scalability | W | ALL |
| NF6 | Recoverability | W | FR1, FR3 |
| NF7 | Manageability | C | ALL |
| NF8 | Portability | W | ALL |
| NF9 | Appearance | C | ALL |
| NF10 | Speed | S | ALL |
| NF11 | Accuracy | S | ALL |

## Use case diagram

Use case is a UML diagram whose purpose is to demonstrate the different ways that a user might interact with the system. Use case diagrams are ideal for:

* Representing the goals of system-user interactions
* Defining and organizing functional requirements in a system
* Specifying the context and requirements of a system
* Modeling the basic flow of events in a use case
* Demonstrating the scope of your system

(https://www.tutorialspoint.com/uml/uml\_use\_case\_diagram.htm)

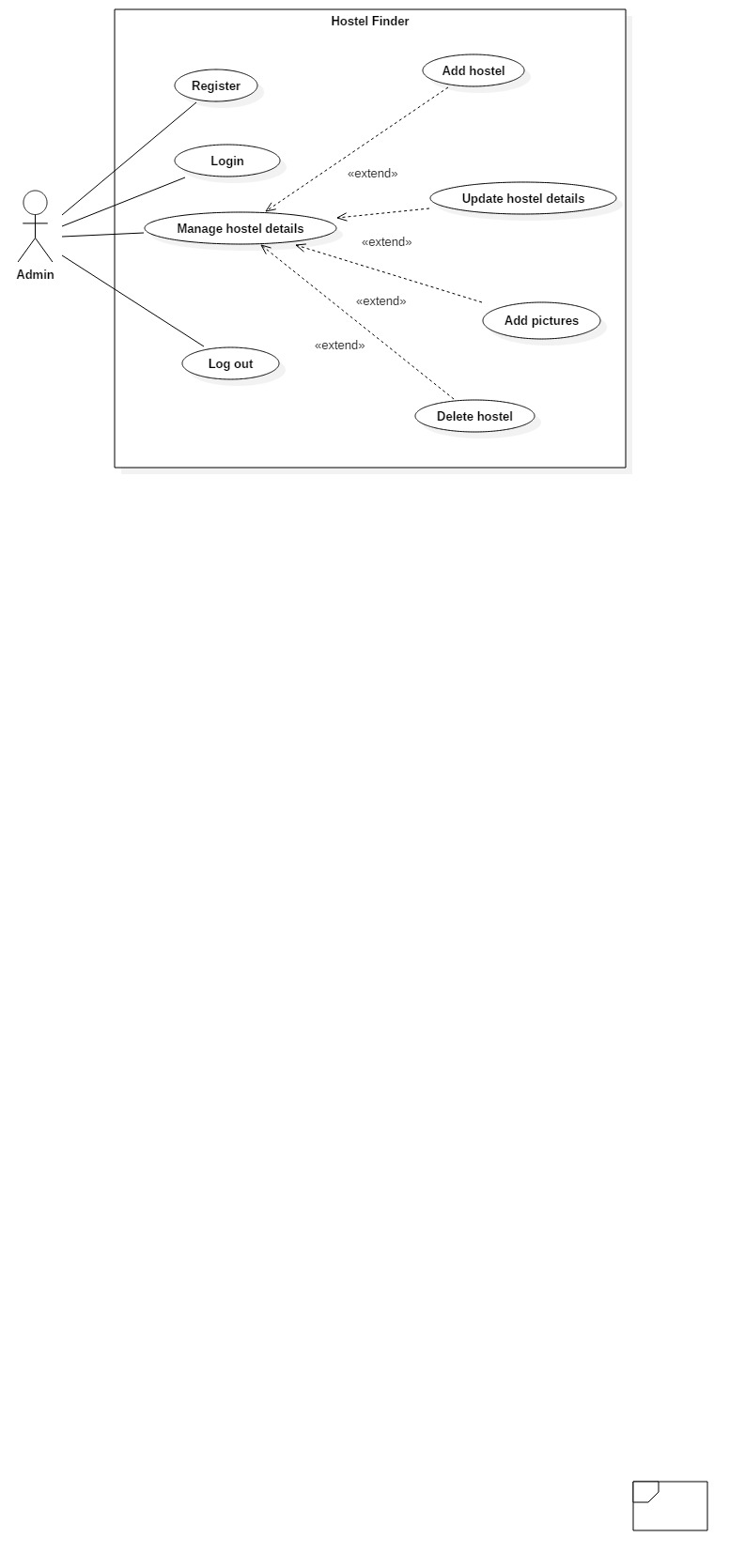


Figure 3: Use case of Admin

**Admin Login**

|  |  |
| --- | --- |
| Actors | Admin |
| Flow of events | Provides email and password |
| Alternative flow | A dialog box is displayed, if password and email don’t match  User will enter correct data. |
| Entry condition | Data entered must be valid |
| Exit condition | User will be displayed a success message. |
| Relationships | Includes: Email and password |

**Manage hostel details**

|  |  |
| --- | --- |
| Actors | Admin |
| Flow of events | Admin logs in to the system  Adds a new hostel/ Updates existing details/ deletes hostel  The data is stored in the database |
| Alternative flow |  |
| Entry condition | Data entered must be valid |
| Exit condition | Admin will be provided success message |
| Relationships | * Extends: Add hostel * Extends: Add pictures * Extends: Update details * Extends: Delete hostel |

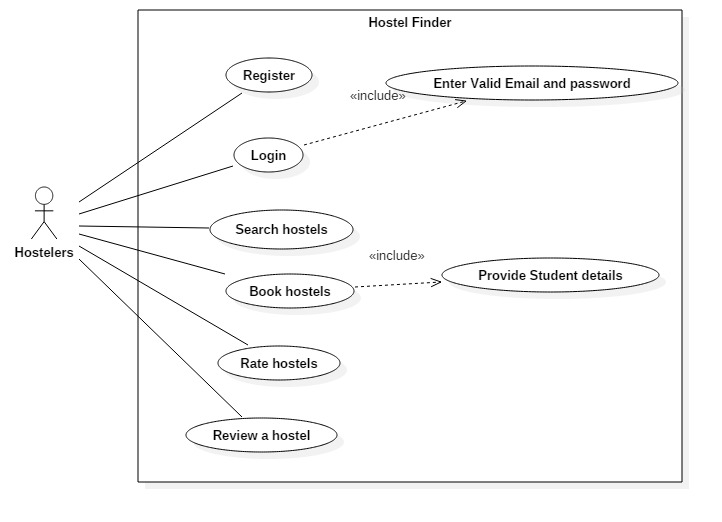


Figure 4: Use case of Hostelers

**User register**

|  |  |
| --- | --- |
| Actors | User |
| Flow of events | Clicks on register option  Provides Name, a unique email and password, Date of birth  Email is verified |
| Alternative flow | A dialog box is displayed, if data is not verified or authentic  User will enter correct data. |
| Entry condition | Data entered must be valid |
| Exit condition | User will be provided a success message |
| Relationships | Includes: Email and password |

**User login**

|  |  |
| --- | --- |
| Actors | User |
| Flow of events | Provides email and password |
| Alternative flow | A dialog box is displayed, if password and email don’t match  User will enter correct data. |
| Entry condition | Data entered must be valid |
| Exit condition | User will be provided a success message |
| Relationships | Includes: Email and password |

**Search hostel**

|  |  |
| --- | --- |
| Actors | User |
| Flow of events | User types in name of hostel, or preferred location  Search results are displayed |
| Alternative flow | If no result is found, a message is displayed |
| Entry condition | **-** |
| Exit condition | **-** |
| Relationships | **-** |

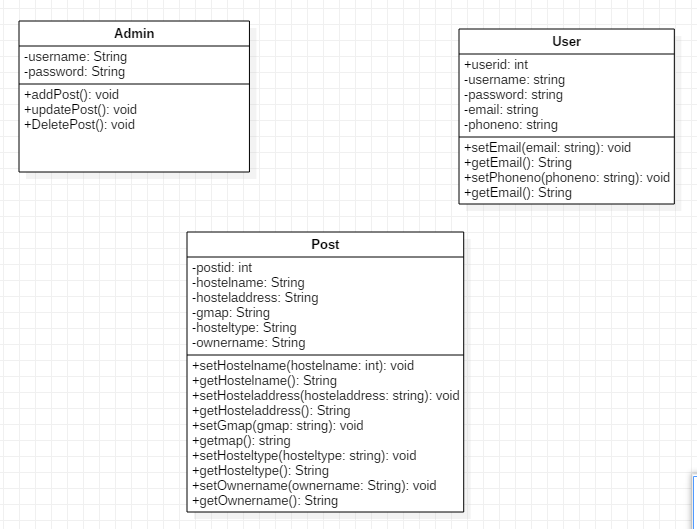
**Book hostel**

|  |  |
| --- | --- |
| Actors | Users |
| Flow of events | * User logs in to the system * Finds a suitable vacancy * Clicks on “book now” option * Fills in the form – Provides details |
| Alternative flow | * If any data is not invalid * A message box is displayed |
| Entry condition | Data entered must be valid |
| Exit condition | Admin will be provided success message |
| Relationships | * Extends: Add hostel * Extends: Add pictures * Extends: Update details * Extends: Delete hostel |

# 2.7 NLA

|  |  |
| --- | --- |
| **Nouns** | **Verbs** |
| User, Post, Admin, | Registers, Logs in, update post, add, post delete post |

# 2.8 Initial class diagram



# Design

**What is design?**

Design in software development lifecycle is the phase where process of thinking about various issues of the problem domain and finding out how you might solve the issue takes place. During the design phase, requirements of the project are further broken down to identify the workflow, estimation of time and resources, and other specific designs.

**Why is design needed?**

Design is important because it allows us to go through the problem domain early in the development phase prior to spending time on implementation phase. It helps us in the following ways:

* Vulnerabilities of the project are discovered.
* Problem solving happens before implementation phase.
* Easier to preserve and evolve the project
* Development becomes cheaper and more efficient.
* A standard is set and we can attempt to stick to it. [(Study.com, 2018)](#sdlc)

During design phase, most of the modelling of the project takes place which were done as shown below:

## Structural Model

Structural diagrams model the different parts of the system such as different objects and classes being used. It shows how the individual parts interact to create a complete system.

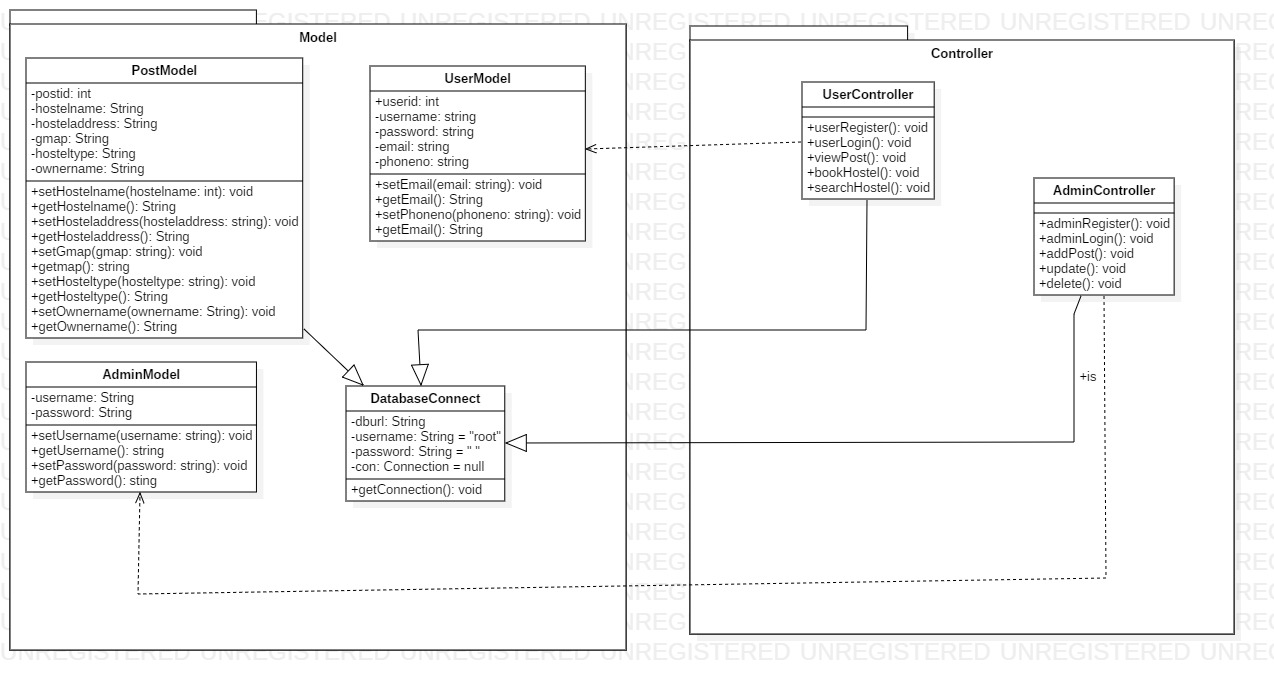
Similarly, structural diagrams visualize the function of the system, starting from the ‘input’ to ‘processes’ and to the ‘output’. Developers decide how the parts/interfaces of a system should be connected based on the representations of structural diagrams. There are different types of structural diagrams such as:

* Class diagram
* component diagram,
* composite structure diagram,
* Deployment diagram
* object diagram
* package diagram

Among them, I have chosen to use class diagram and flow chart.

***Class diagram***

Class diagram is generally used at the time of system construction as it visualizes the structure of that system by modeling its classes, attributes, operations and relationships between them.



***Flow chart***

Flow chart is a basic diagram, which, with the use of symbols and arrows, shows the flow of events in a complex system. Flowcharts help in decision making by emphasizing the significant processes and eliminating the unnecessary ones.

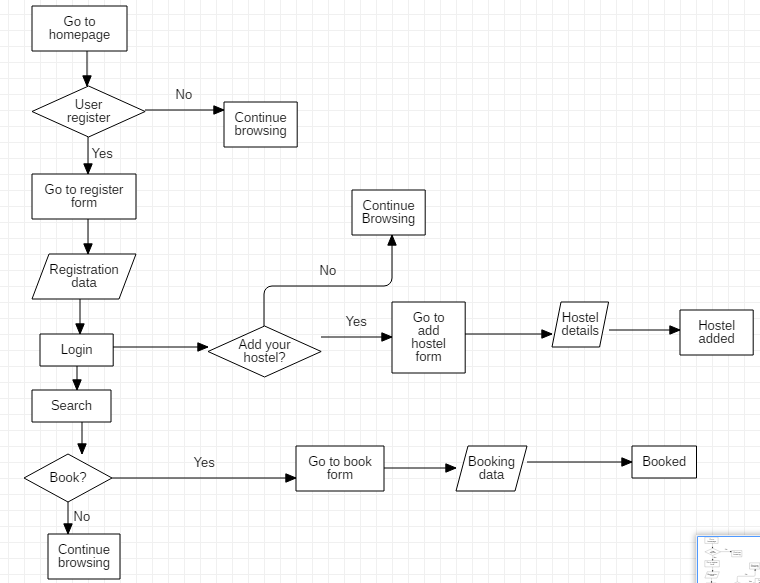


Figure 5: Flowchart

.

## Behavioral Model

Diagrams in behavioral model show how the system to be modeled is expected to behave.

* Activity diagram
* Communication diagram
* Sequence diagram
* Use case diagram
* Interaction overview diagram

***Activity Diagram***

It is a diagram which, with the use of various symbols and signs, represents and describes the flow of control in a system. It shows activities in a system and links between them. Various activity diagrams can be designed to capture the entire flow in a system. It gives us the idea of how a system will work after it is executed.

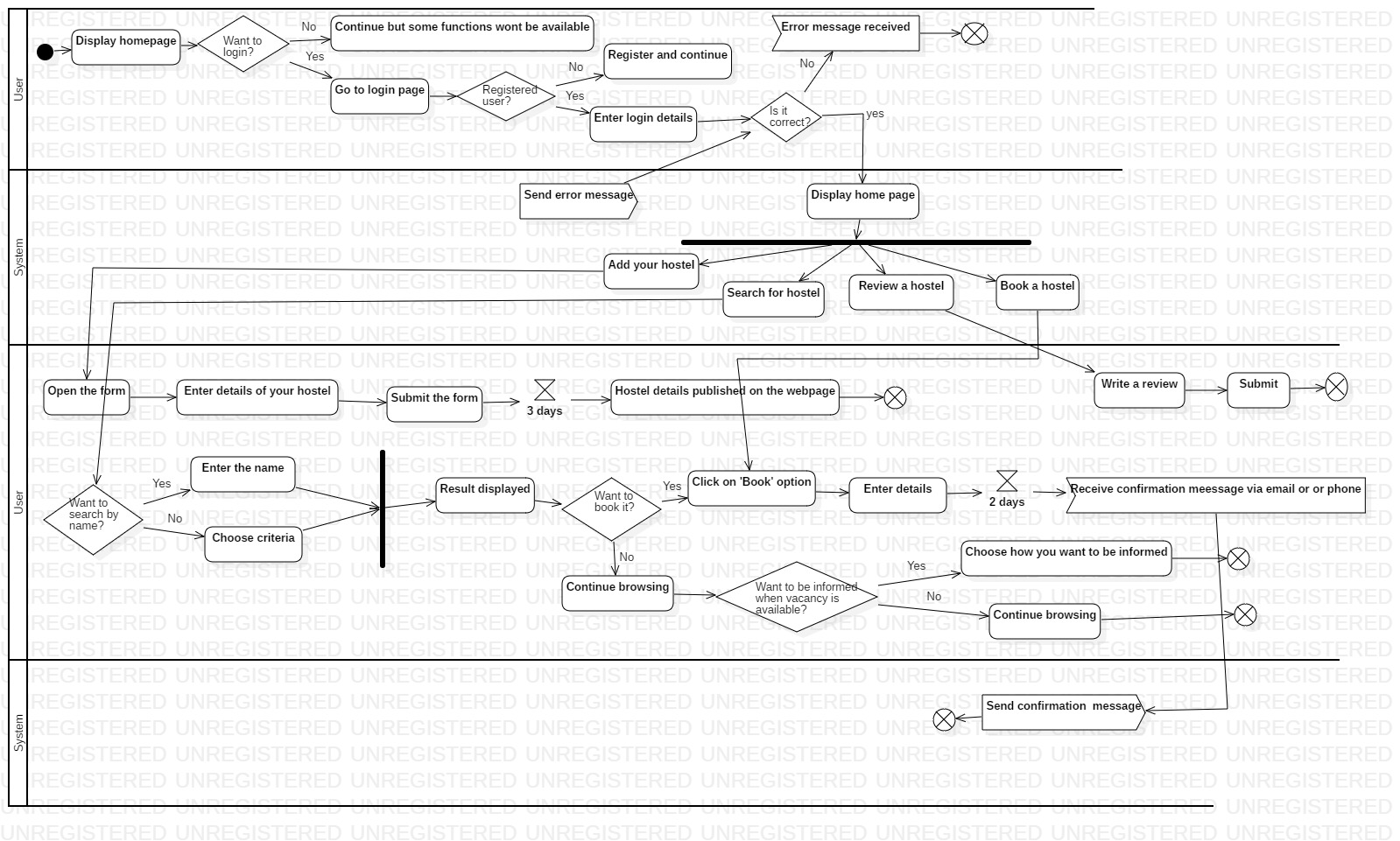


Figure 6: Activity Diagram

***Sequence Diagram***

Sequence diagrams show details of how the operations of the system are carried out along with the time constraint. They capture the interaction among objects in the perspective of a collaboration. Sequence diagrams are organized according to time and object which are placed at vertical and horizontal axis. (GeeksforGeeks, 2018)

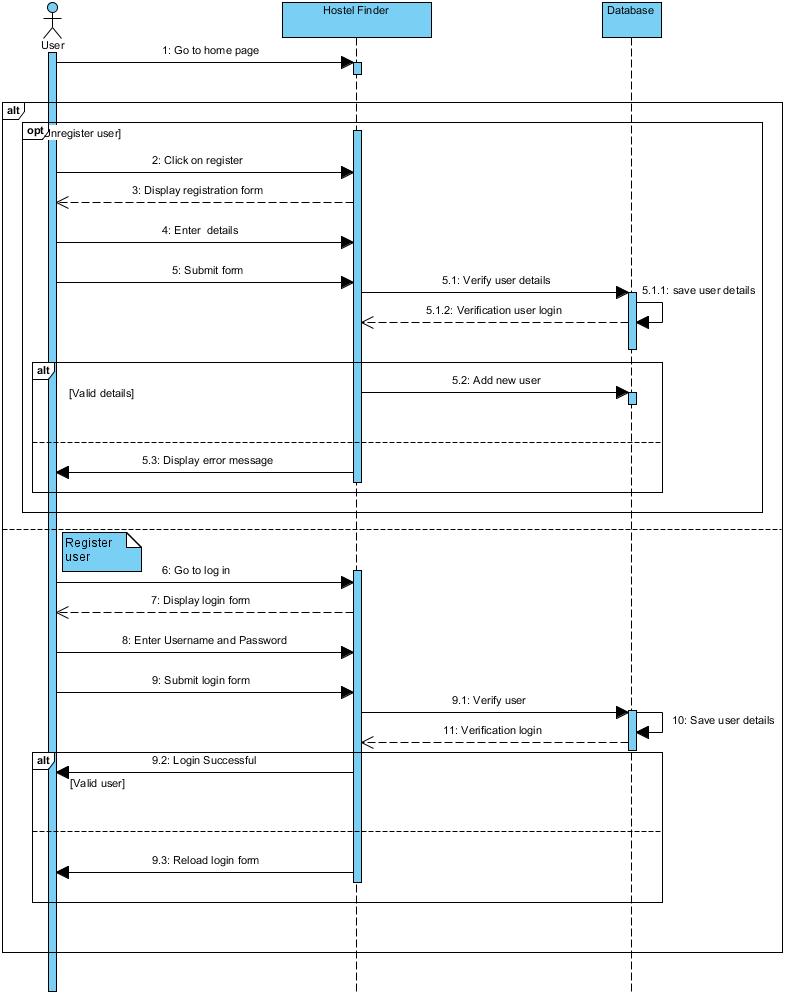


Figure 7: Sequence diagram

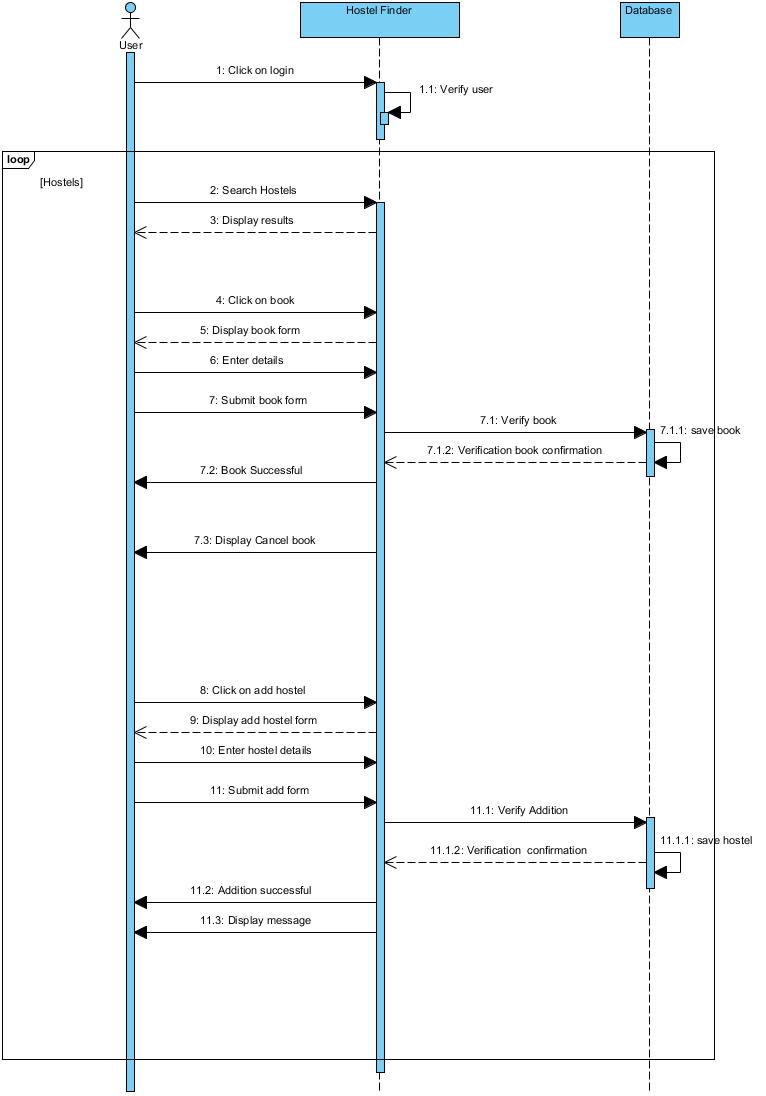
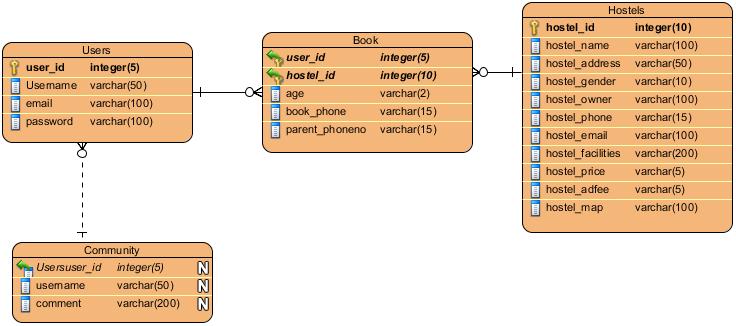


Figure 8: Sequence diagram

## Database Model

Database modelling is done via Entity Relationship Diagram which shows the entities of a system and the relationship between them. Data of the system will be saved under these entities later. My Entity relationship diagram is shown below:



**Users**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** |
| User\_id | varchar | 5 | Not null | Primary Key |
| Username | varchar | 50 | Not null |  |
| email | varchar | 100 | Not null |  |
| Password | varchar | 50 |  |  |

**Community**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** |
| Comment\_id | varchar | 5 | Not null | Primary Key |
| Username | varchar | 50 | Not null | Foreign Key |
| Comment | varchar | 200 | Not null |  |
| date | timestamp |  | Not null |  |

**Hostels**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** |
| Hostel\_id | varchar | 5 | Not null | Primary Key |
| Hostel\_name | varchar | 50 | Not null |  |
| Hostel\_address | varchar | 5 | Not null |  |
| Hostel\_owner | Varchar | 100 | Not null |  |
| Hostel\_gender | Varchar | 5 | Not null |  |
| Hostel\_phone | Varchar | 15 |  |  |
| Hostel\_facilities | Varchar | 200 |  |  |
| Hostel\_price | Integer | 5 |  |  |
| Hostel\_email | Varchar | 100 |  |  |
| Hostel\_adfee | Integer | 5 |  |  |
| Hostel\_map | Varchar | 200 |  |  |

**Booking**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Column Name** | **Data Type** | **Length** | **Null** | **Key** |
| User\_id | varchar | 5 | Not null | Foreign Key |
| Hostel\_id | varchar | 50 | Not null | Foreign Key |
| Book\_age | varchar | 5 | Not null |  |
| Book\_phone | varchar | 15 | Not null |  |
| Parent\_phoneno | varchar | 15 | Not null |  |

## UI Design

User Interface design was made via ‘balasmiq mockups’. Designing User Interface might not seem very important but in fact it can be very useful later in the development project as it gives understanding of the coding we are expected to do later on.

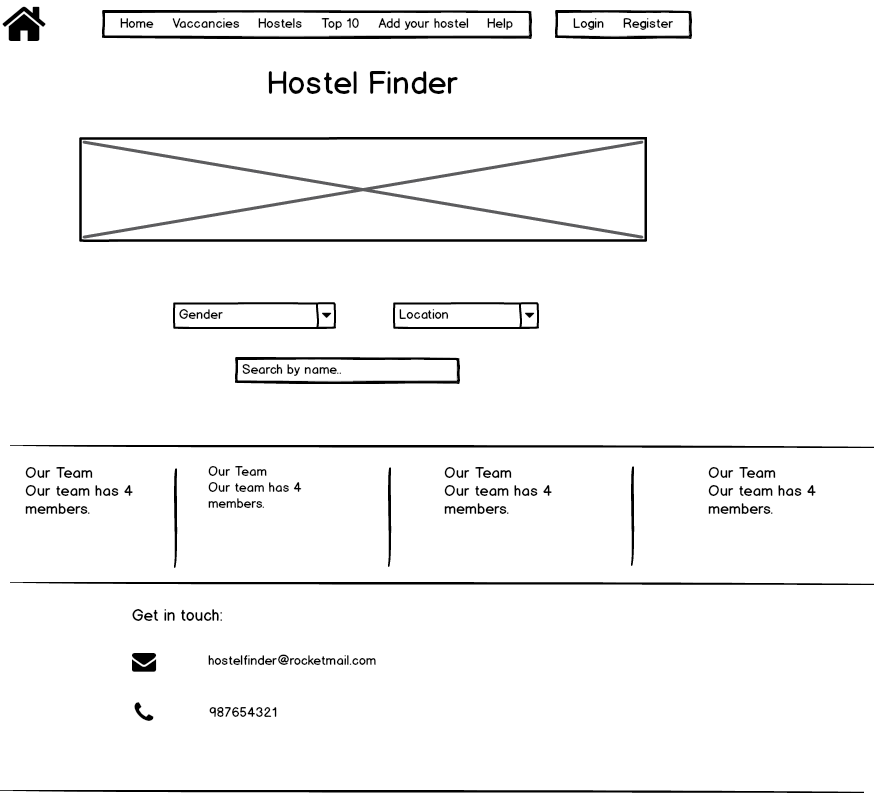


Figure 9: UI Model-Home

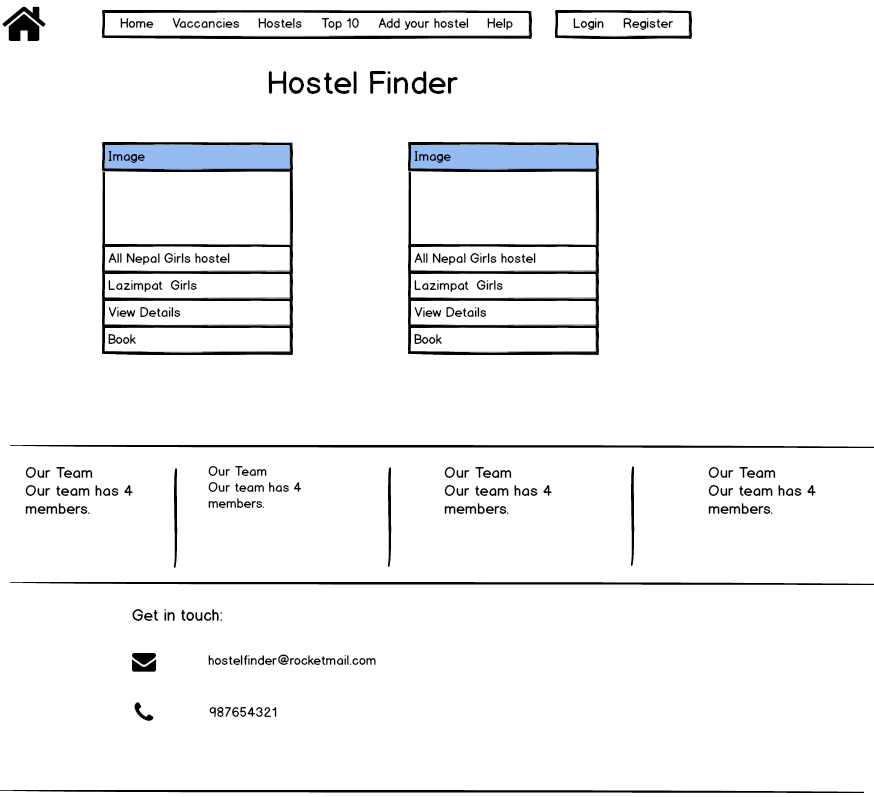


Figure 10: UI model: vacancies



Figure 11: UI model-list

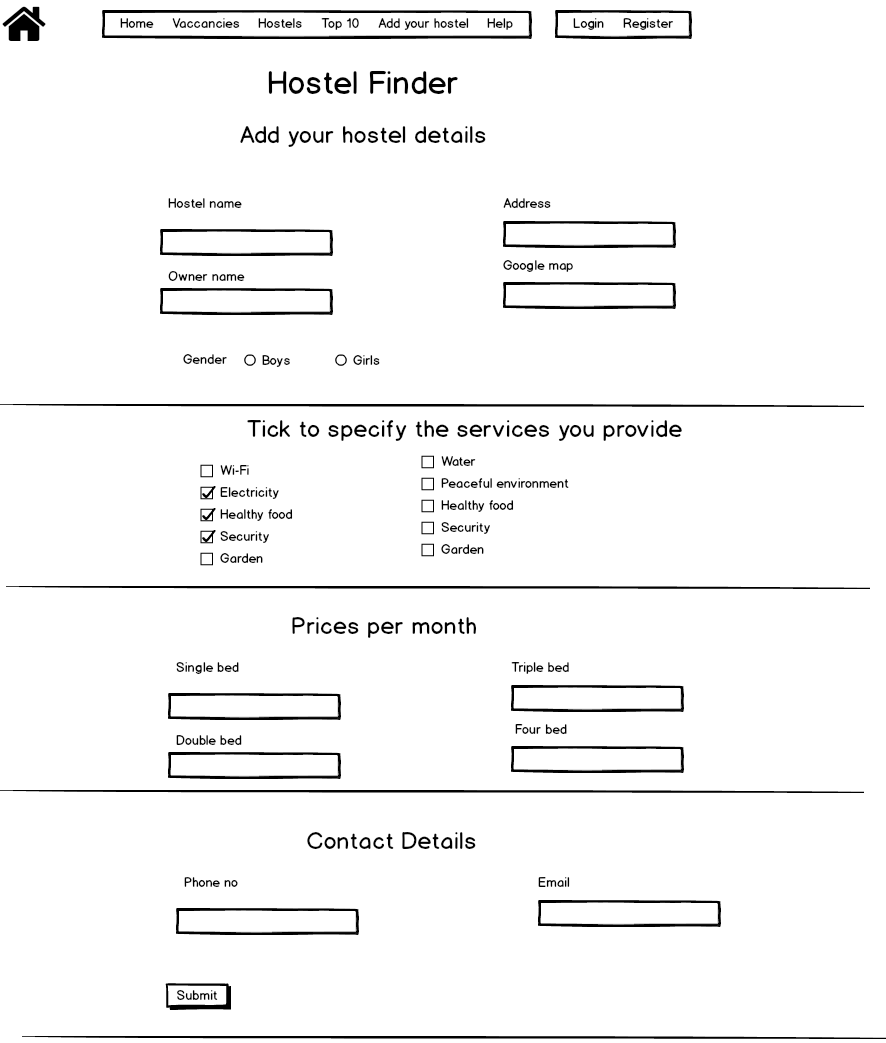


Figure 12: UI model-add hostel

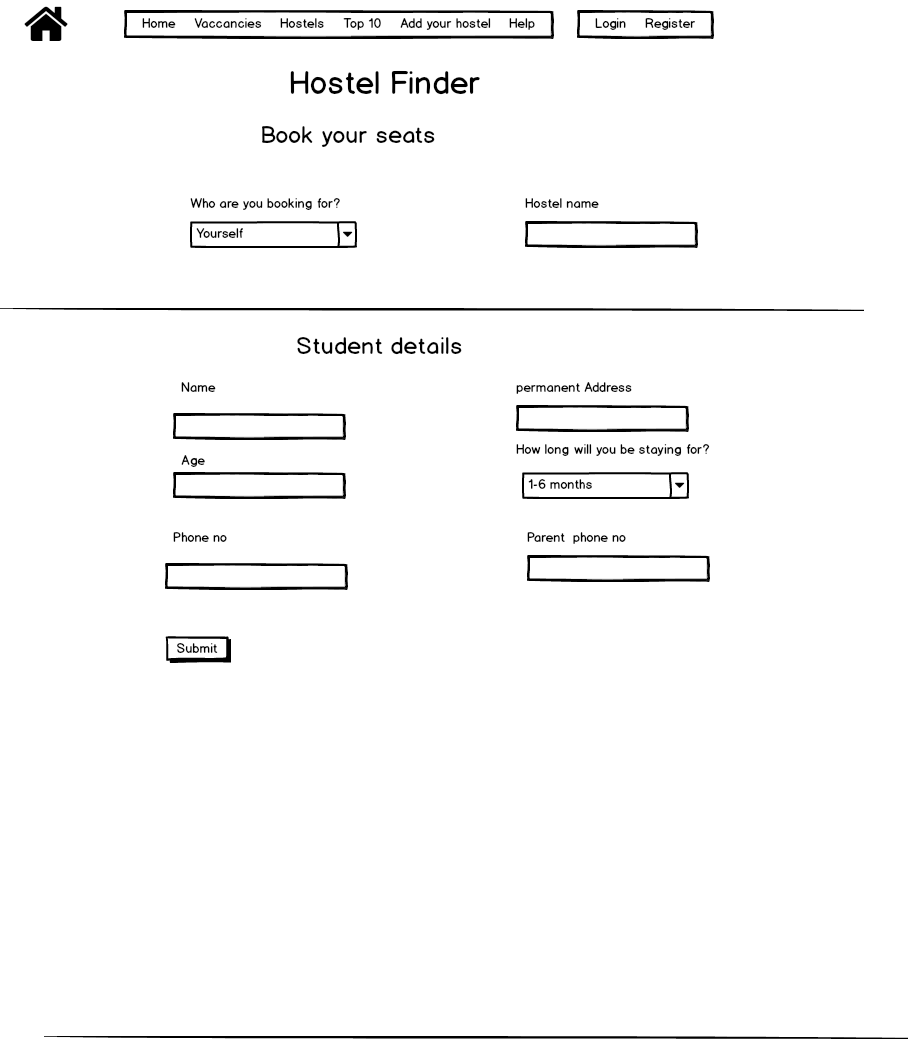


Figure 13: UI model-book

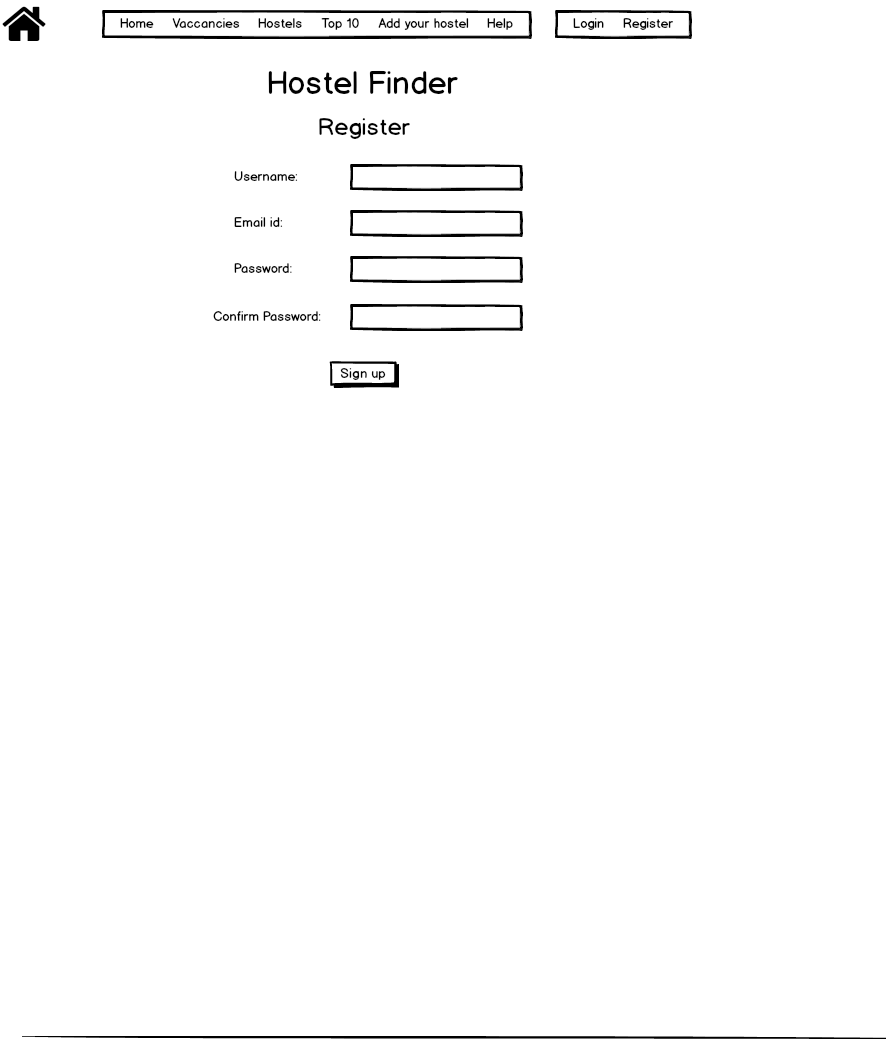


Figure 14: UI model-register

# Testing

**Testing**

Software testing is a process of evaluating the functionality of a software. Testing aims to find out any existing errors, shortcomings or unexpected outputs in a system. Testing can be of different type. Different types of testing can be done to evaluate a software. Black box testing, white box testing, unit testing, integration testing are some of them. Among all these types of testing, I have performed black box testing.

**Black-box testing**

Black box is a basic kind of software testing where functionality of software is checked without considering the internal structure of the code, details of the implementation and knowledge of internal structure of software. Basically, it is the process of testing outputs against the provided input. This testing is based on software requirements and software specifications.

**Registration**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Plan** | **Test Cases** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T01 | User registration | Username, email id, password | Registration was done with valid data | User should be register | As Expected | Pass |
| T02 | User register with invalid data | Username, email id, password | Passwords did not match | User should not be able to register | As Expected | Pass |

**Login Test**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Plan** | **Test Cases** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T03 | Check Customer Login with valid Data | Username, password | - | User should Login into application | As Expected | Pass |
| T04 | Check Customer Login with invalid Data | Username, password | Password was wrong | User shouldn’t login into the system and error message should be displayed | As Expected | Pass |

**Add hostel**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T05 | Add hostel with valid data | Hostel name, hostel address, contact details facilities, price, gender | All fields were filled | Hostel should be added and success message should be displayed. | As Expected | Pass |
| T06 | Add hostel with invalid data | Hostel name, hostel address, contact details facilities, price, gender | Hostel name field was left empty | Hostel should not be added and message should be displayed | As expected | Pass |

**Book Hostel**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T07 | Book hostel with valid data | Name, permanent address, phone no, staying time, parent phone no. | Hostel name, hostel address, contact details and other details. | Hostel should be added and success message should be displayed. | As Expected | Pass |
| T08 | Book hostel with invalid data | Name, permanent address, phone no, staying time, parent phone no. | Permanent address field was left empty | Booking should not be successful and message should be displayed | As expected | Pass |

**Search results**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Case** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T09 | Search via gender and location | Gender and location | Hostel gender was selected ‘girls’ and location was ‘Maitidevi’ | Girls hostel in Maitidevi should be displayed | As Expected | Pass |
| T10 | Search hostel via name | Hostel name | Hostel details | Hostel with such word in its name should be displayed | As expected | Pass |
| T11 | Search hostel via name | A random word | word | Nothing should be displayed | As expected | Pass |
| T12 | Search hostel via name | An alphabet | Hostel name field was less than 3. | Error message should be displayed | As expected | Pass |

**Suggestion**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Test Scenario | Test Case | Description | Expected Results | Actual Results | Pass/Fail |
| T13 | Provide feedback while logged out | Username, password | Feedback form was opened while logged out | Feedback form should not open | Feedback form opened | Fail |
| T14 | Insert suggestion while logged out | Username, password | Feedback form was opened while logged out | Feedback form should not open | As Expected | Pass |
| T15 | Display suggestion | Username, password | Hostel details | Feedback with username should be displayed | As expected | Pass |

**Log out**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Plan** | **Test Cases** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T16 | Download user manual | Username, email id, password | Download link was click on help page | User manual should be downloaded | As Expected | Pass |

**Log out**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Plan** | **Test Cases** | **Description** | **Expected Results** | **Actual Results** | **Test log** |
| T17 | User Log out | Username, email id, password | Log out button was clicked | User should be logged out from the system | As Expected | Pass |

# 6. Other project issues

## Issues

Every project has some issue which might be solved or remain unsolved. During, risk management, I had complied some risks likely to occur during the project.

|  |  |
| --- | --- |
| **Likelihood** | **Value** |
| Low | 1 |
| Medium | 2 |
| High | 3 |

|  |  |
| --- | --- |
| **Consequence** | **Value** |
| Very low | 1 |
| Low | 2 |
| Medium | 3 |
| High | 4 |
| Very high | 5 |

**Impact=Likelihood\*Consequence** is the formula used to calculate the impact a particular risk will cause upon the system.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **Risk Description** | **Likelihood** | **Consequences** | **Impact** | **Preventive measures** |
| 1 | **Scope creep-**unexpected changes and uncontrolled growth to a project’s scope which can result in additional costs, failure to meet time constraints, even project failure. | 1 | 3 | 3 | Make a definite plan and stick to it. |
| 2 | **Inaccurate project estimating**-The intangible nature of software makes it extremely difficult to estimate and schedule. | 2 | 3 | 6 | Use more than one method to create estimates and search for a midpoint. |
| 3 | **Unstable application** -browser incompatibility, incorrect memory usage, fatal production bugs, and more | 1 | 5 | 5 | Optimize SSL/TLS, Monitor live issues |
| 4 | **Poor Productivity-**  During the lifespan of project, there can be times where productivity of developers | 2 | 5 | 10 | Short iterations, coaching, motivation |
| 5 | **Software and hardware defects** | 3 | 4 | 12 | Scheduled maintenance, Proper backup |

## Evaluation of own work

Looking back at my work and evaluating it, I am pretty much satisfied with it. There have been some limitations but overall, I’ve dealt with the issues well. Risk management was done as per the plan. I dealt with the issues occurred as explained in the previous topic. Configuration management turned out to be as follows:

Also, the backup of the project in was kept in google drive, onedrive and github. The link to my github is: (https://github.com/alakaacharya/Hostel-Finder)

## Future work

I want to see my project grow to be a better and more useful version of itself in the future. I have planned to work on these aspects of the system in the future:

* Notifications of vacancies via SMS
* Rating a hostel
* Ranking of hostels based on their ratings
* GPS based search
* Development of a mobile app

## Limitations of my project

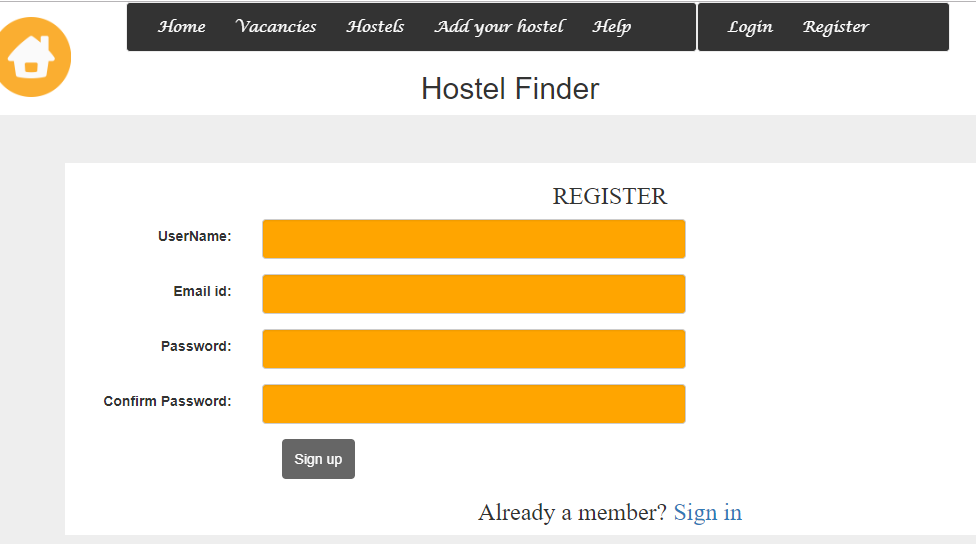
* Doesn’t contain the real / reliable data, instead, I have used self-reported data
* My website might not be visible uniformly on all kinds of devices.
* Some functionalities have been missed which are separated for future work.

# User Manual

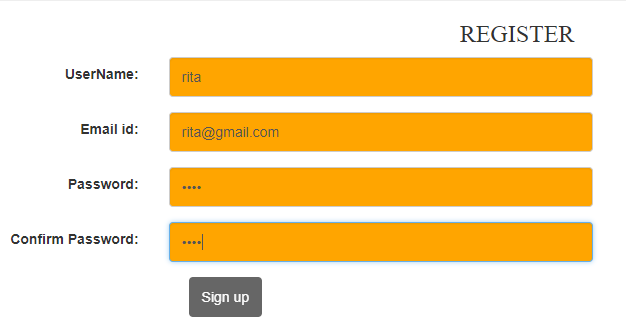
# Registration

Registration can be done in few simple steps.

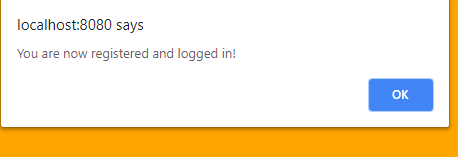
* Visit Register option



* Fill out the form



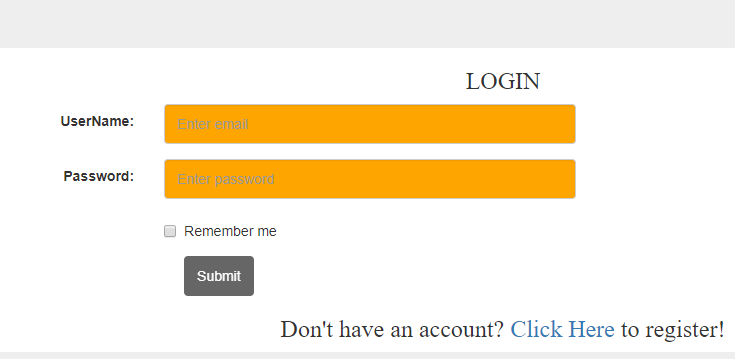
* Submit it



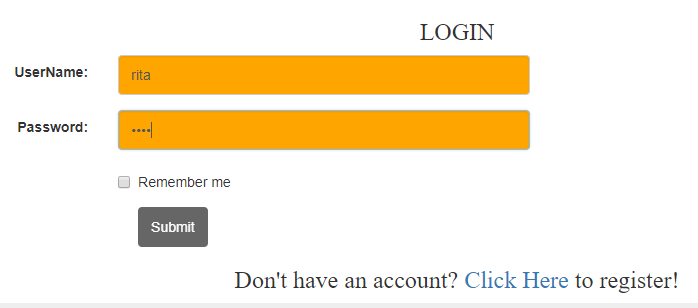
# Logging In

If you have already registered, follow the following steps

* Go to login page



* Enter your username and password



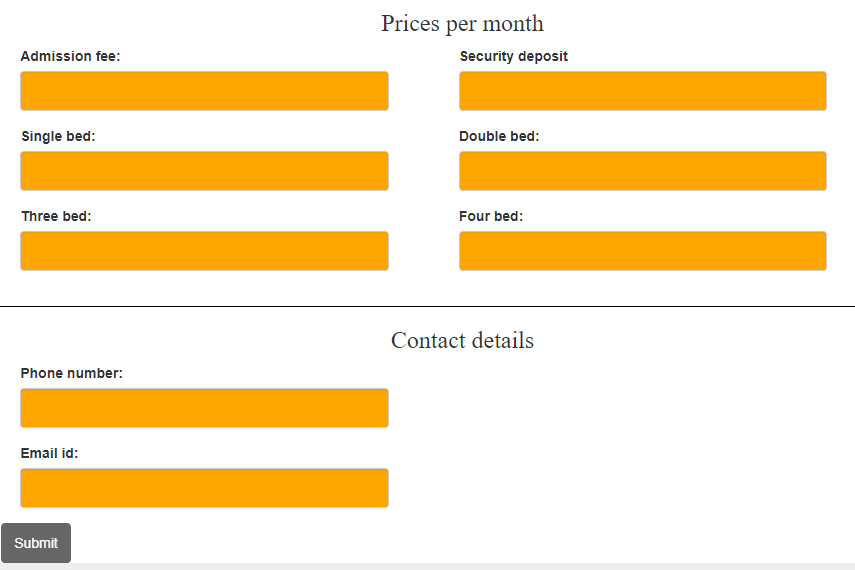
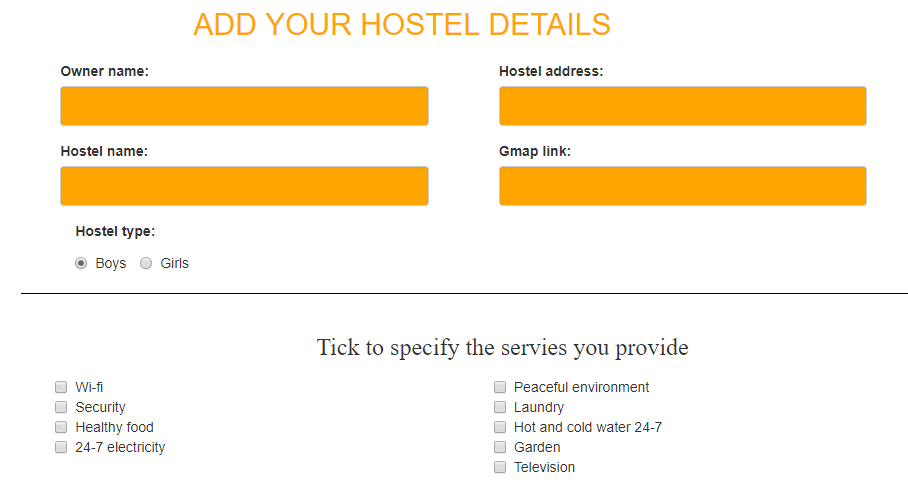
* Submit it



# Adding your hostel

If you are a hostel owner, and you want to add your hostel to our page, follow these steps:

* Go to ‘Add your hostel option’
* A form opens



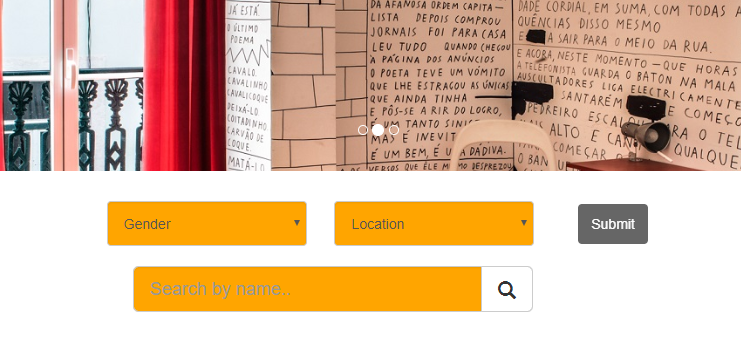
* Fill the form as asked
* Submit it



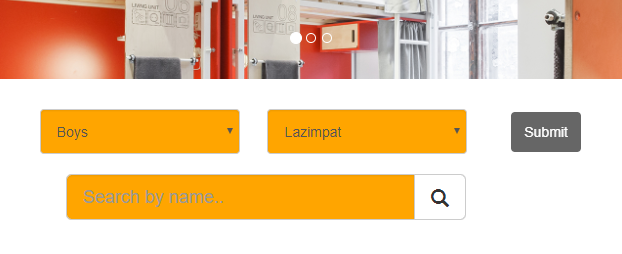
After we receive the data, we will contact you as soon as possible to verify the details before publishing it to the page.

# Searching

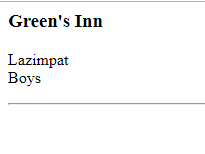
Searching a hostel is very easy with hostel finder. Our search areas are located right at the center of our home page.



* Choose the type of hostel you are searching for and your preferred location
* Click on Submit button

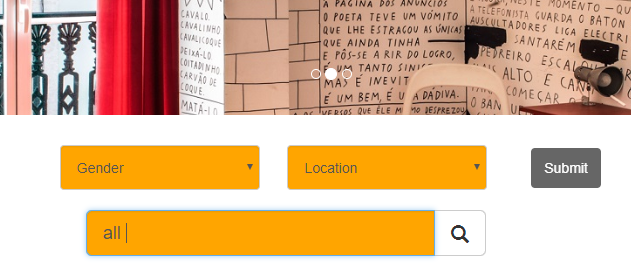


* All the results will be displayed to you.



This feature is for you if you already know the name of a hostel you want to search.

* Go to the text bar and type a name.
* Click on ‘find icon’

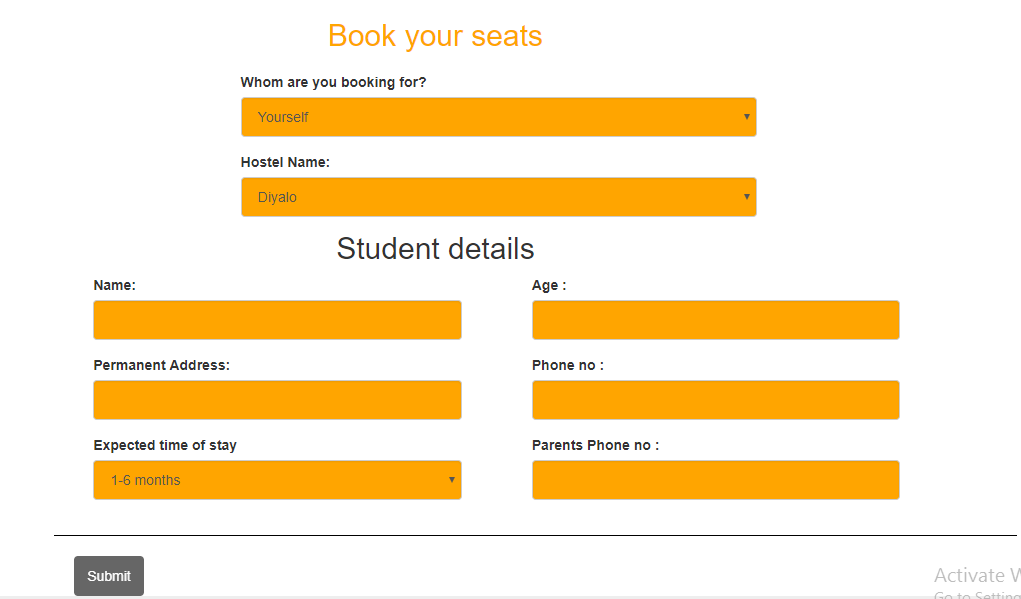
All the results will be displayed.



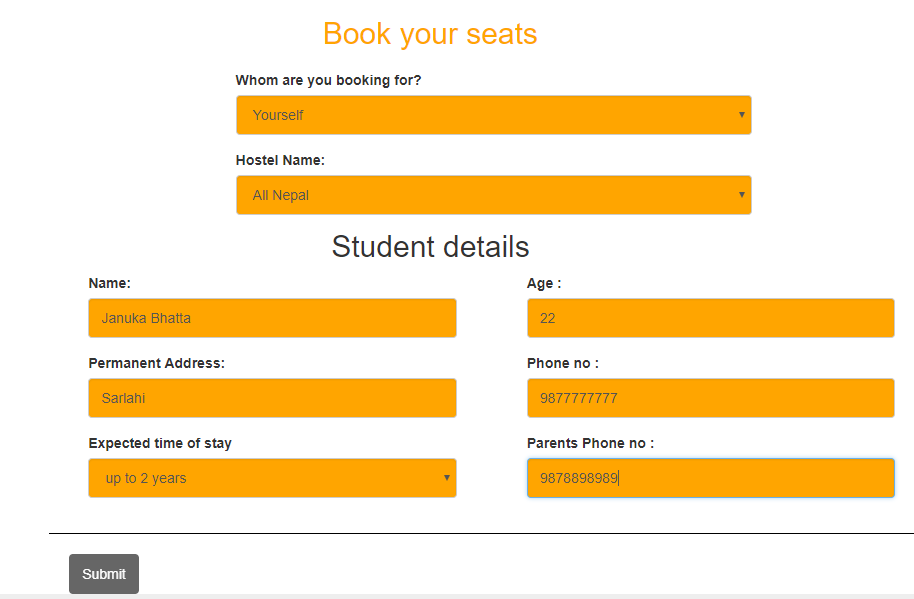
# Booking

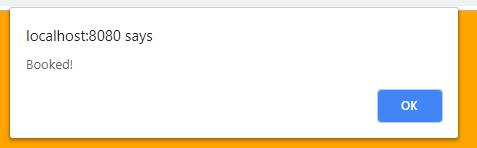
To book a hostel, you have to follow the following steps:

* Go to the ‘vacancies’ page to search for vacant hostels.
* Find a hostel and click on ‘Book’
* A form opens



* Fill the details as specified
* Click on submit form.
* Wait for success message

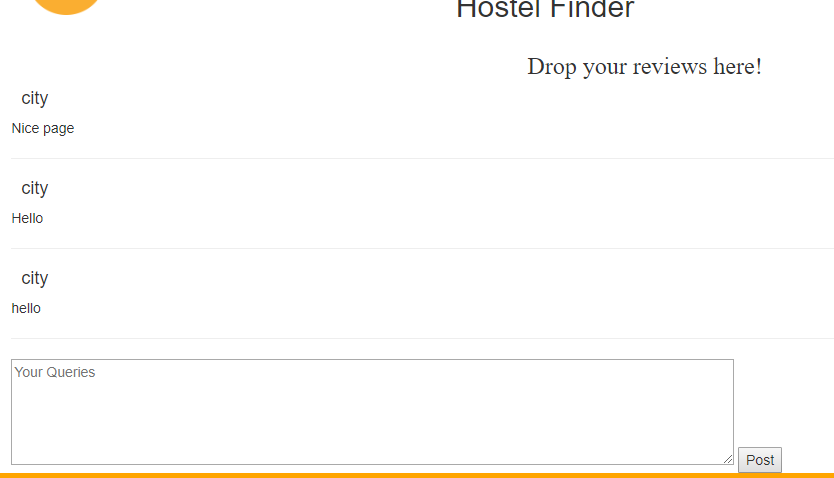




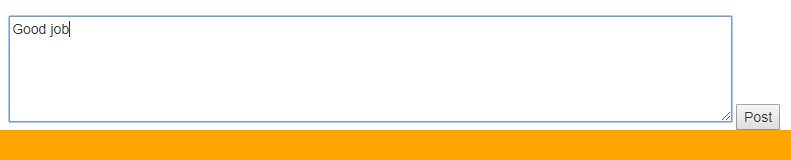
# Leave us a review

Before you leave, please provide us some feedbacks.

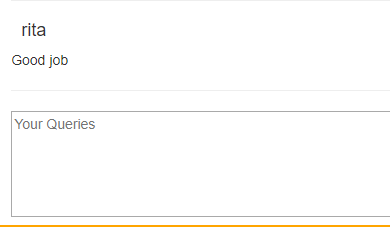
* Visit our help page
* Go to the review form



* Enter your feedbacks



* Click on submit button
* Your feedbacks will be then displayed



# FAQs

* What does the hostel finder do?

-Hostel Finder helps you in finding the right hostel for you!

* What can I do via hostel finder?

-You can search hostels in preferred locations, you can browse through the lists of hostels, and you can find the right hostel for you and book it!

* I am a hostel owner. How can I add my hostel in hostel finder?

-You can go to the hostel finder, fill the form with all the details for your hostels and submit it to us. We will contact you in few days for verification and your hostel will be added.

* What type of hostels are featured in hostel finder?

-Every boys and girls hostel in Kathmandu valley are featured in hostel finder.

* Can I search for hostels outside Kathmandu valley via hostel finder?

-We are sorry to say that for now, we are featuring the hostels within Kathmandu valley only. We are working towards expanding hostel finder.

* How can I book a hostel?

-Go to the booking form, fill out the details and submit it. Easy!

* What happens once I submit the book form?

-We will forward your details to the respective hostel owners and they will contact you as soon as possible.

* How do I announce vacancy in my hostel?

-Contact us and we will do that for you!

* How can I contact you?

-Go to the contacts page at the footer of homepage for our contact details.

# 7. Conclusion

In this way, the project was built successfully more or less according my initial plan and design. At first analysis of the project was done to find out if it was suitable to carry out the plan. After that, design of the project was made. Class diagrams, Activity diagram, Sequence diagram were made during this phase. After that code for the project was written using php. The code was tested consecutively. The project was documented.

# 8. References and Bibliography

www.tutorialspoint.com. (2018). UML - Modeling Types. [online] Available at: https://www.tutorialspoint.com/uml/uml\_modeling\_types.htm [Accessed 4 Sep. 2018].

GeeksforGeeks. (2018). Unified Modeling Language (UML) | Sequence Diagrams - GeeksforGeeks. [online] Available at: https://www.geeksforgeeks.org/unified-modeling-language-uml-sequence-diagrams/ [Accessed 8 Sep. 2018].

Junaid Hassan's Technical blog. (2018). What is SRS (Software Requirement Specification)? What is SRS means? Why it is necessary to create SRS? What are its features, advantages and uses? What is Software requirements specification template according to IEEE?. [online] Available at: https://junaidhassanalvi.wordpress.com/2015/12/26/what-is-srs-software-requirement-specification-what-is-srs-means-why-it-is-necessary-to-create-srs-what-are-its-features-advantages-and-uses-what-is-software-requirements-specification-template/ [Accessed 8 Oct. 2018].

UKEssays. (2018). Soft Systems Methodology. [online] Available at: https://www.ukessays.com/essays/information-systems/soft-systems-methodology.php [Accessed 13 Sep. 2018].

Study.com. (2018). Design Phase in SDLC - Video & Lesson Transcript | Study.com. [online] Available at: https://study.com/academy/lesson/design-phase-in-sdlc.html [Accessed 10 Sep. 2018].

Brighthub Project Management. (2018). *10 Essential Business Requirements Gathering Techniques*. [online] Available at: https://www.brighthubpm.com/project-planning/60264-techniques-used-in-business-requirements-gathering/ [Accessed 17 Aug. 2018].

FTIAS. (2018). The Benefits of System Analysis - FTIAS. [online] Available at: http://www.ftias.com/benefits-system-analysis/#.W9FPg2gzbIV [Accessed 20 Aug. 2018].

Brighthub Project Management. (2018). Explaining the Different Types of Feasibility Studies. [online] Available at: https://www.brighthubpm.com/project-planning/56372-types-of-feasibility-studies/ [Accessed 18 Aug. 2018].

Agile Business Consortium. (2018). DSDM Atern Handbook (2008). [online] Available at: https://www.agilebusiness.org/content/moscow-prioritisation-0 [Accessed 13 Sep. 2018].

# 9. Appendix





Figure 15: addhostel.php



Figure 16:details.php



Figure 17:book.php

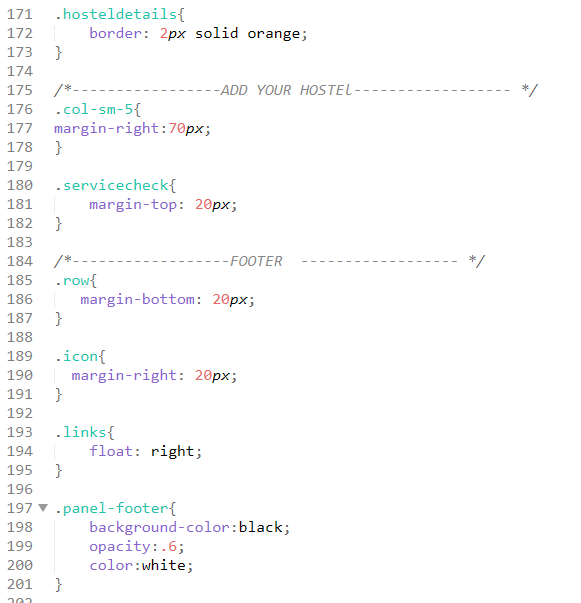
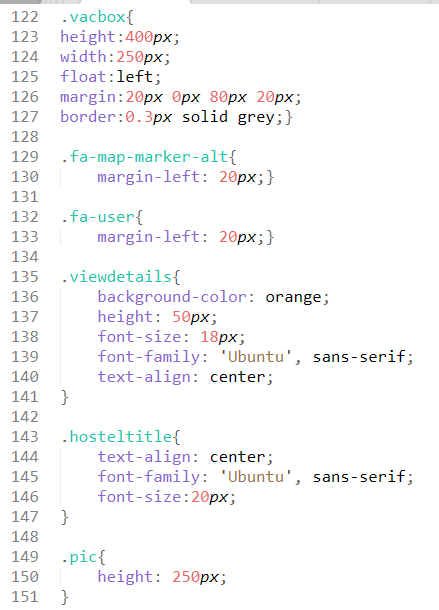
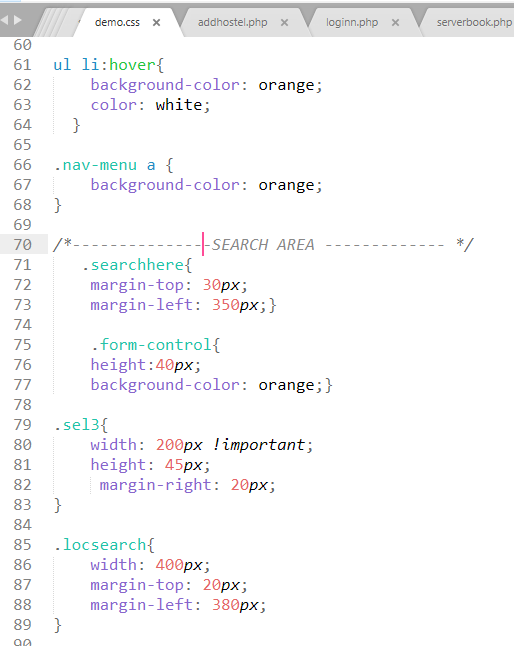
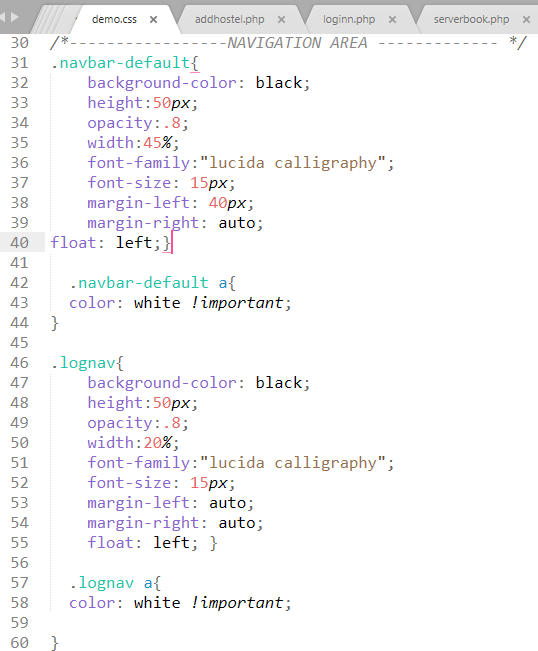


Figure 18: demo.css

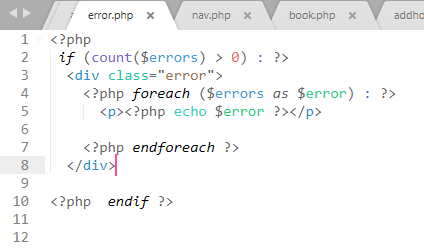


Figure 19:error.php

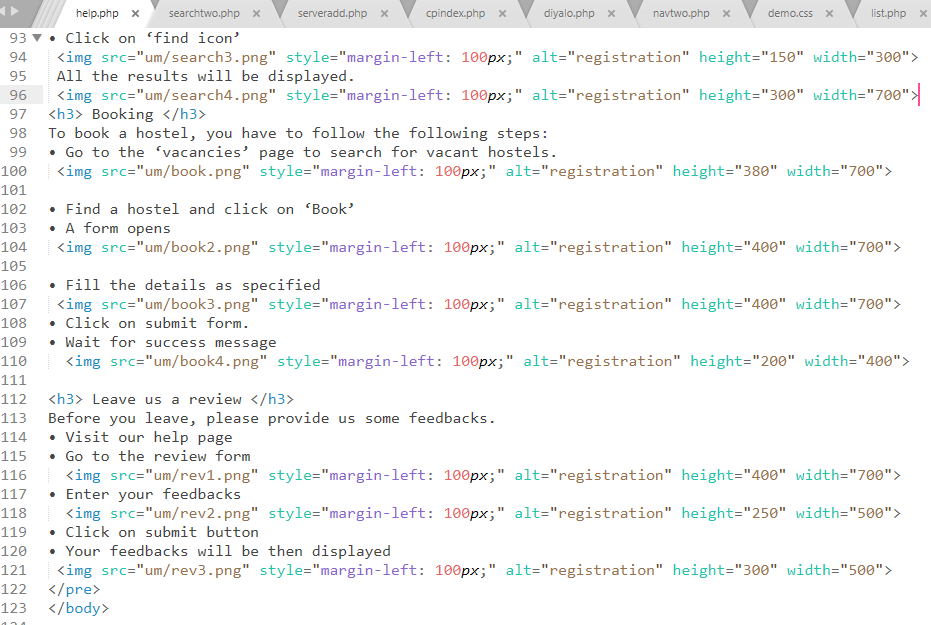
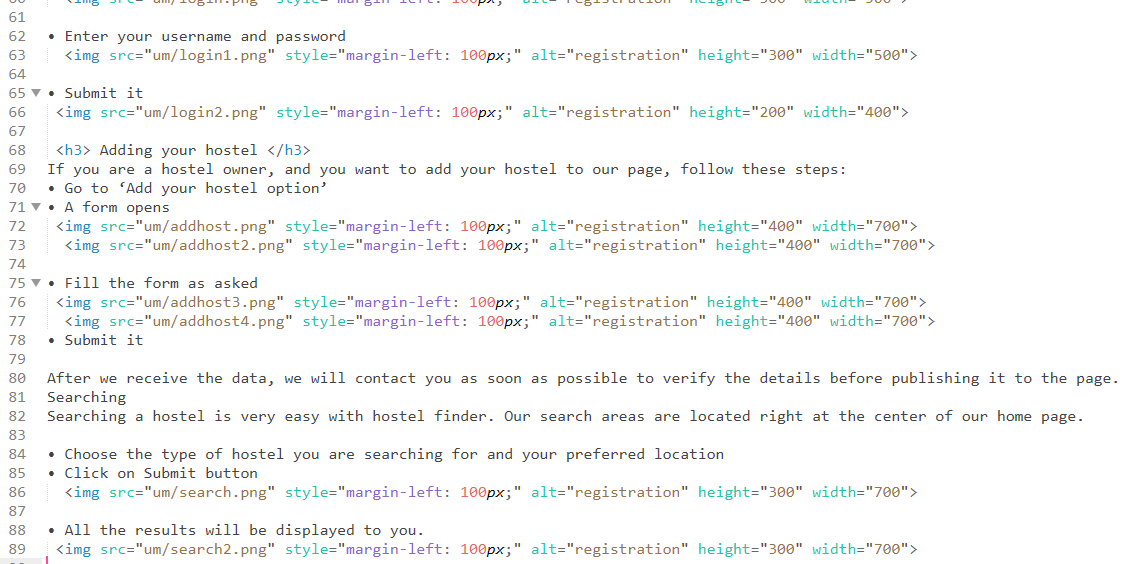
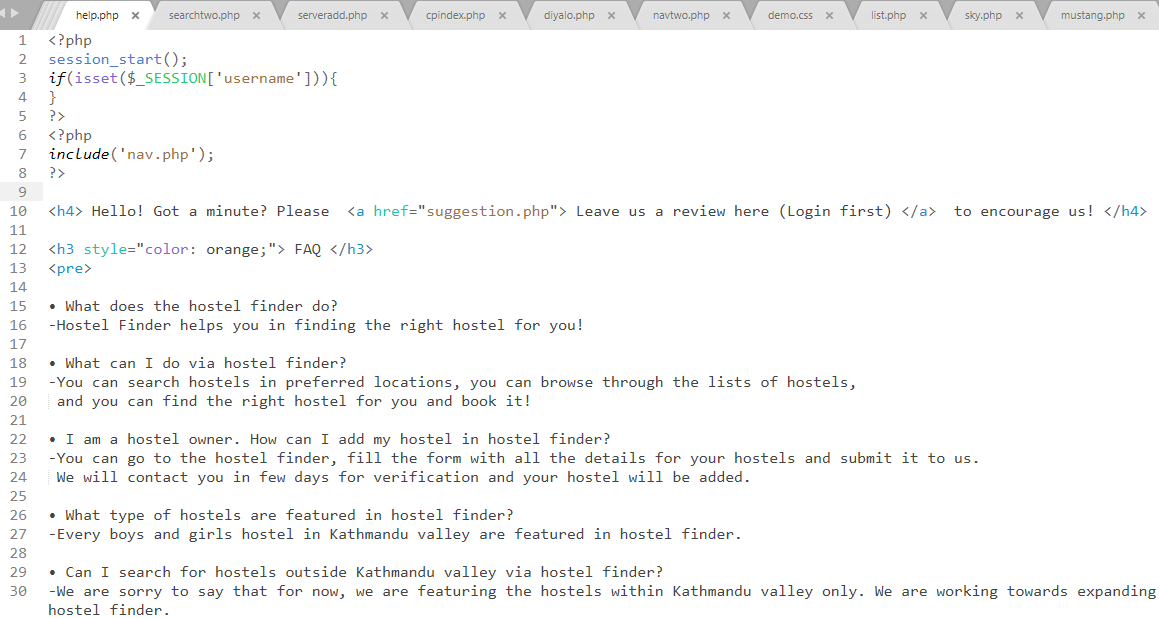


Figure 20:help.php

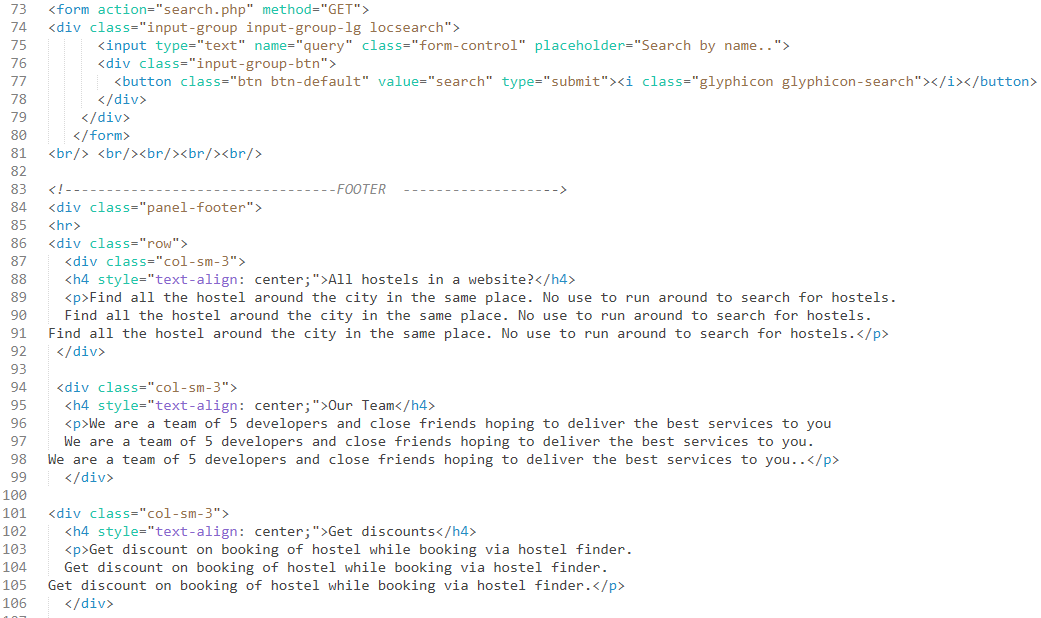


Figure 21:cpindex.php

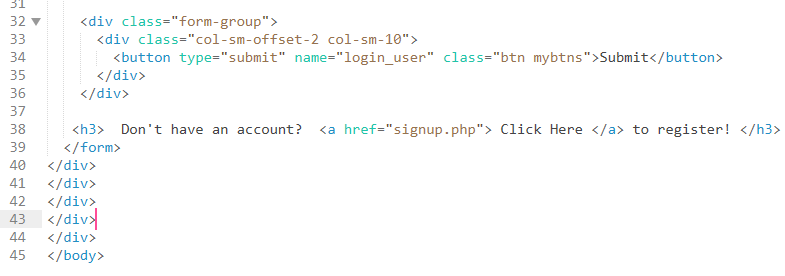


Figure 22: loginn.php

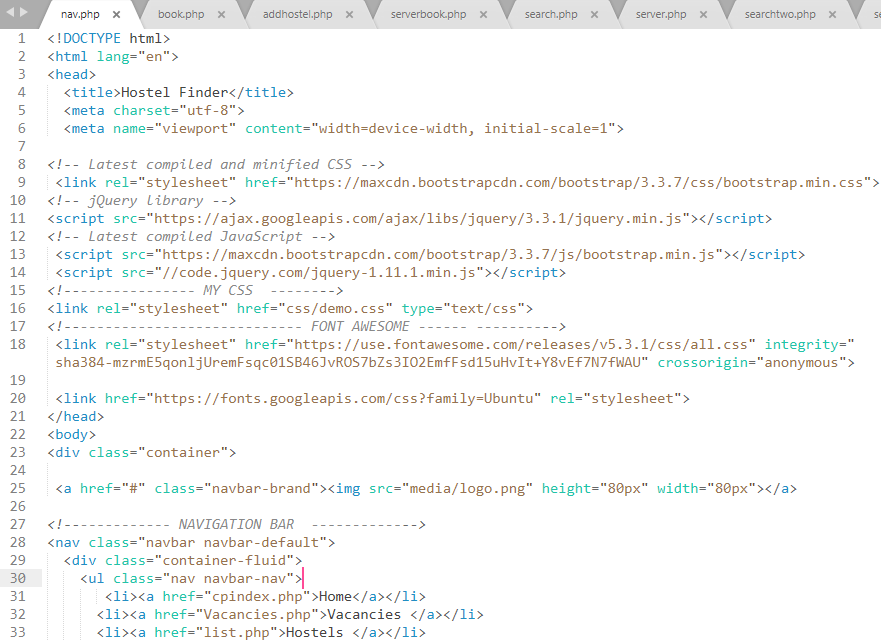


Figure 23: nav.php

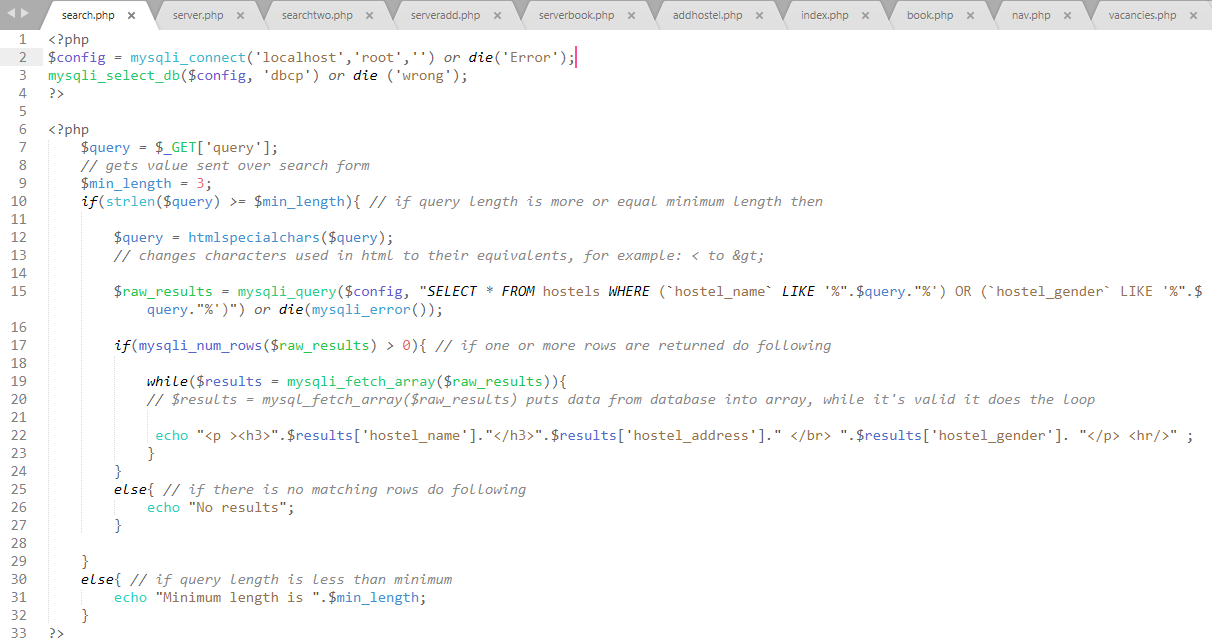


Figure 24:search.php

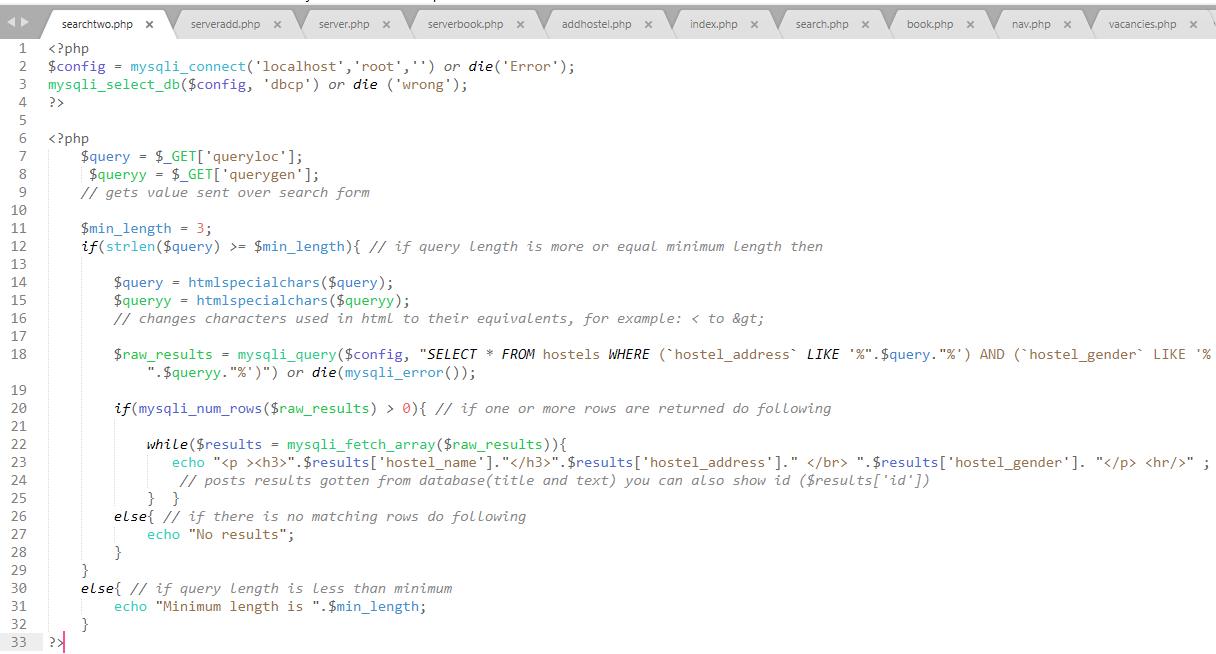


Figure 25: searchtwo.php

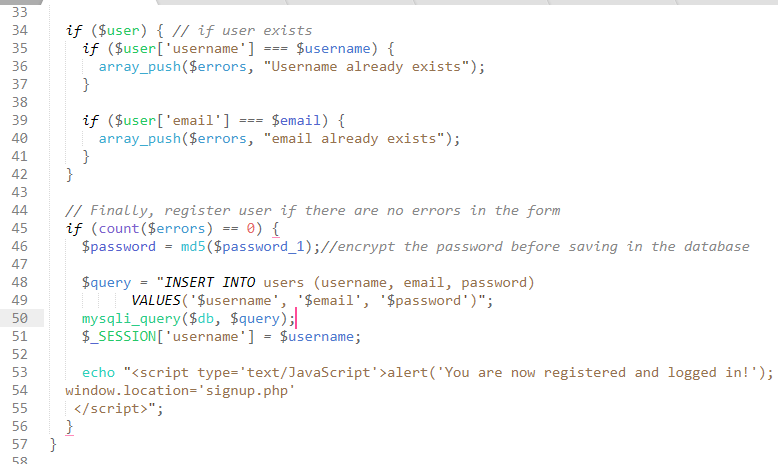




Figure 26: server.php

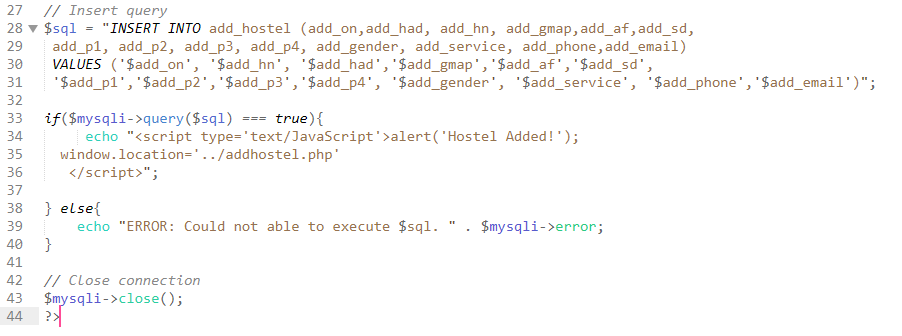
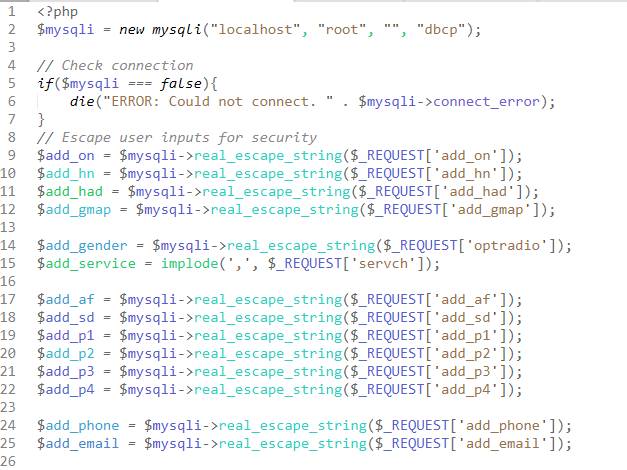


Figure 27:serveradd.php



Figure 28: serverbook.php



Figure 29: signup.php



Figure 30:suggestion.php



Figure 31:vacancies.php