

Bachelor of Science in Computer Science and Engineering

House Rental Management System

Md. Basedul islam

ID: 150101093

Chow. Wasif Hossain

ID: 150101052

July, 2019

Department of Computer Science & Engineering

Bangladesh Army University of Science and Technology

Saidpur(Cantonment), Nilphamari

House Rental Management System

This thesis is submitted in partial fulfillment of the requirement for the degree of Bachelor of Science in
Computer Science & Engineering.

By
Md. Basedul Islam
ID: 150101093
Chow. Wasif Hossain
ID: 150101052

Supervised by
Md. Al-Hasan
Lecturer
Department of Computer Science & Engineering(CSE)
Bangladesh Army University of Science & Technology

July, 2019
Department of Computer Science & Engineering
Bangladesh Army University of Science and Technology
Saidpur(Cantonment), Nilphamari

The thesis titled “**House Rental Management System**” submitted by Roll No. 150101052, 150101093, Session 2015-2019 has been accepted as satisfactory in fulfillment of the requirement for the degree of Bachelor of Science in Computer Science & Engineering (CSE) as B.Sc. Engineering to be awarded by the Bangladesh Army University of Science & Technology.

Board of Examiners

1_____

Chairman

Md. Al-Hasan

Lecturer

Department of Computer Science & Engineering

Bangladesh Army University of Science & Technology(BAUST).

2_____

Member

Md. Al-Hasan

Lecturer

Department of Computer Science & Engineering

Bangladesh Army University of Science & Technology(BAUST).

3_____

Member

Md. Al-Hasan

Lecturer

Department of Computer Science & Engineering

Bangladesh Army University of Science & Technology(BAUST).

BANGLADESH ARMY UNIVERSITY OF SCIENCE AND TECHNOLOGY

July 2019

Declaration

To the best of my knowledge and belief, I certify that this thesis does not:

- I. Incorporate any material previously submitted for a degree or diploma in any institution of higher education without acknowledgement.
- II. Contain any material previously published or written by other person except where due reference is made in the text.
- III. Contain any defamatory material.

I also grant permission to **Bangladesh Army University of Science and Technology** to make duplicate copies of my thesis as required.

Signature:

Date:

Abstract

The project House Rental Management System is implemented to reduce the manual work and enhances the accuracy of work for renting. This system has been made in a user friendly interface. This project is also designed with full consideration to help the users in an easy manner without any unnecessary wastage of time. It could be House, Hotel or Car. Searching a house for renting is a tough task in Bangladesh. To make it easier we are trying to make a website where people can rent their affordable house right sitting on their present home without rooming around for searching it. On the other hand, if someone is going for a trip to somewhere, they can rent their hotel room just like they did rent their house. They can also rent a car from our website. Moving home isn't an easy task but being prepared can help to make the process smoother, saving you time and unnecessary stress. This guide to renting aims to provide you with the advice you'll need to get the most out of renting as well as reassurance with answers to common problems and questions. Bangladeshi rental market has grown significantly over the past few years.

Key Words: House Rental Management System, PHP, HTML, CSS, JavaScript, Wamp server, MySQL, Database.

Contents

| | |
|---|----|
| 1 Introduction | 1 |
| 1.1 The problem | 1 |
| 1.2 Project objectives | 2 |
| 1.3 Project requirements analysis | 3 |
| 1.3.1 Requirements analysis | 3 |
| 1.4 Project deliverables | 3 |
| 2 Technological background | 5 |
| 2.1 Implementation process | 5 |
| 2.2 Tools and technologies | 5 |
| 2.2.1 MySQL database | 5 |
| 2.2.2 PHP | 5 |
| 2.2.3 HTML | 6 |
| 2.2.4 CSS | 7 |
| 2.2.5 JavaScript | 7 |
| 2.2.6 Apache server | 7 |
| 2.2.7 Visual Studio Code | 8 |
| 2.3 Chapter Summary | 9 |
| 3 System design | 10 |
| 3.1 Architectural design | 10 |
| 3.2 Database diagram | 12 |
| 3.2.1 Entity diagram | 12 |
| 3.2.2 E-R diagram | 14 |
| 3.2.3 Landlords | 15 |
| 3.2.4 Rent Seekers | 16 |
| 3.3 Incremental Model | 17 |
| 3.4 Database schema | 17 |
| 3.4.1 Database tables structures | 18 |
| 3.5 Chapter Summary | 20 |
| 4 Implementation & Testing | 21 |
| 4.1 Chapter overview | 21 |
| 4.2 User Module | 21 |
| 4.2.1 User Interface | 22 |
| 4.2.2 Results of Search feature | 22 |
| 4.2.3 Goggle Map | 23 |
| 4.3 Test Case | 25 |
| 4.4 Version Control | 26 |
| 4.5 Code Documentation | 27 |
| 4.6 Chapter summary | 27 |
| Conclusion | 28 |

| | |
|-----------------------|----|
| References | 29 |
| Acknowledgement | 31 |

List of Figures

| | |
|--|----|
| 3.1 Architecture of House Rental Management System | 11 |
| 3.2 Admin entity diagram | 13 |
| 3.3 Client entity diagram | 13 |
| 3.4 House add entity diagram | 14 |
| 3.5 Comments/Review entity diagram | 14 |
| 3.6 ER diagram of House Rental Management System | 15 |
| 3.7 Activity diagram of Landlords | 16 |
| 3.8 Activity diagram of Rent seekers | 16 |
| 3.9 Incremental Model for House Rental Management System | 17 |
| 4.1 User Interface | 22 |
| 4.2 Search User Interface | 23 |
| 4.3 Architecture of House Rental Management System | 24 |

List of Tables

| | |
|-----------------------------------|----|
| 3.1 Admin registration | 19 |
| 3.2 Registration for client | 20 |
| 3.3 Client order list | 20 |
| 4.1 Test Cases | 26 |

1 Introduction

Searching for a suitable house for rent has become a painful job especially in crowded cities like Dhaka. Both the landlords and rent seekers faces trouble to get the right thing. Now a days the Bangladesh government has also keep track of every people in every flat/house. But this is also becoming a tough task to keep the right record each time as they relocating quickly to a new flat/house for their job. We worked in such home rental system to build a more dynamic system than the existing ones. Our goal of this thesis is to make the system as user-friendly as possible. For this purpose, we have shrunk the whole in via an icon in the mobile phones as well so that the users can search for vacant houses and transport at any time. To minimize any kind of communication gap, we introduce a chatting system between agents and admin. Users always want to get a result in the shortest span of time. In many cases, may be users get the reply lately due to internal long manual processes. To solve that, we introduce a dynamic mail system to notify the users' search result in the least possible time. Therefore, the home rental system holds features like rental search, dynamic mail system, tenants can search for vacant houses randomly according to their wishes based on unique area or city, they can track for vehicles from their given location as well. Additionally, they can add their wish list if their desired home is not available. There will be a chat system in which admin and agent can chat directly. All the values which are used for processing result are dynamic in our home rental system. In our system, there is an option whether u want family house or bachelor. However, recently it has become a trivial issue for the bachelors because none of the house owners are ready to rent their houses to them. Therefore, we create such a safe and user-friendly platform for the bachelors so that they can rent house efficiently and, in a hassle, free procedure.

1.1 The problem

Roam around to rent a house has always been a hassle for people. Especially, on recent times, people have so many priorities based on which they have to rent their house. Some people want their house to be in the commercial space, or some want in a chaos free space. Some people prefer to choose the area of their house relating the religion they belong. Again, there are a lot of people who love pets; therefore, they want a house which has pet allowance. Basically, in this era of modernism people want to rent their house like online shopping. To rent a house in physical world has become less popular now a days [1] one wants to roam around here and there to search for a house. People would prefer a virtual system to rent a house.

To decode this situation and to represent a hassle-free environment to the people, a dynamic system can be implemented. That system would give the tenants the best service for renting houses without any kind of hassle. Government can make one unique system where people can rent house based on their priority instead of having so many rental systems. In that system, all the vacant houses of any district of Bangladesh will be listed there. One system will hold every details of every vacant house from any district, any are. To, make the system more liable, there should be a system by which tenants can verify the owner or agent. Also, to analysis the place they will rent for house they need to know the location of that. Hence, every information details which have minimum priority to rent a house will hold by the system. There a one special feature for the bachelors so that they can rent houses efficiently as now a day house owner does not want to rent their houses to the bachelors for safety issue.

1.2 Project objectives

According to the universal declaration of human rights by which we get to know that is belong to a human the right to have a proper standard of living. (United Nations, The Human Rights-article 25, 1948).[1] A standard living place is every citizen's basic right. Choosing that place is also their right. As technology is growing so fast every single day, it is necessary to make the system the most dynamic approach. By the need of this, a dynamic home rental system is needed where every citizen can choose their house according to their choice and also it has to be most dynamic as people will access this system at anytime from anywhere.

Targeting these issues in manual house rental management system, we are proposing an automated rental management system where the bellow three stakeholders will be benefited greatly-

- Landlords
- Rent Seeker
- The government

Our project is about Renting House. Searching a house for renting is a tough task in Bangladesh. To make it easier we are trying to make a website where people can rent their affordable house right sitting on their present home without rooming around for searching it.

Moving home isn't an easy task but being prepared can help to make the process smoother, saving you time and unnecessary stress.

This guide to renting aims to provide you with the advice you'll need to get the most out of renting as well as reassurance with answers to common problems and questions. Bangladeshi rental market has grown significantly over the past few years. Though once reserved only for students or those unable to purchase, renting has become an increasingly attractive alternative to buying by providing a number of unique benefits such as:

- **Flexibility**-when you rent you agree to stay in a property for a fixed length, usually 12 months, meaning that you are free to move elsewhere if you want.
- **Cost**-because you do not own the property you are not liable for property maintenance costs or replacing expensive items such as boilers.

Most popular in larger towns and cities, renting in the Bangladesh is often the first step for most after moving out of their family home. Though currently not as heavily regulated as the resale and buying market, many new rules have come into effect over the last couple of years providing greater protection for tenants. While the majority of tenants will never experience a problem with their tenancy or landlord, it's important to understand what's usual (and what's not) and what's expected from both parties.

1.3 Project requirements analysis

Project requirements analysis are important stage in the system development. It determines the functions of the whole application integrity and stability. Software requirements analysis is an ongoing process of understanding and progressive refinement. Through requirements analysis, design functions of the management system as below-

- a. **User management:** User can signup (client), signin and signout the application.
- b. **Adding a house:** admin can add new house information for showing to clients.
- c. **House's information browsing:** The house's are grouped by categories. Clients may browse the detailed information of a house.
- d. **house query:** admin can query house's according to price, name.
- e. **house comments:** clients can write comments on the house.

1.3.1 Requirements analysis

The in-front management system is the user visits home list and signup user as client. Only the admin can manage his/her searching potion about the specific home, comment and price. So in this part, specific functions are described as below:

- **Signin and Signout:** User can sign in into system and also sign out from the system.
- **Register:** If user have no account, user have to must create an account.
- **Modify personal information:** User can also modify his personal information.
- **Browse detailed information of a home:** User can browse details of home.
- **Comments:** User can post a comment for each home.

1.4 Project deliverables

Deliverables[2] are usually classified as internal deliverables and external deliverables.

Internal deliverables : Internal deliverables are usually deliverables that make a project run, but they are not a part of the product that the end users would like to see. They are deliverables which the project generates internally. Project Management, Configuration Management, Training and Testing are some examples of internal deliverables.

External Deliverables : External deliverables are usually those that the project delivers to the users or the client. An external deliverable could be an IT system and subsystems that make it up or the resulting organizational transition and benefits from a project to reduce the turnaround time of a process.

The main deliverable of this project is to build a simple and easy use of Asian restaurant management application following the specific software requirements as well as the programming languages.

- Functioning software application.
- List of (non) functional requirement.
- Schematic models (analysis+design).
- Evaluation findings.
- Dissertation report.

2 Technological background

2.1 Implementation process

In the thesis project, Window 10 as an operating system,MySQL as a database and Visual Studio Code as a IDE. This System is a collection of Apache server, MySQL database comprehensive programming and easy to use.

2.2 Tools and technologies

The tools used to accomplish in this project are MySQL[3], Apache Server[4] and PHP[5] is Object oriented language and HTML, CSS, JavaScript

2.2.1 MySQL database

MySQL[3] is an open source relational database management system. It is based on the structure query language (SQL), which is used for adding, removing, and modifying information in the database. Standard SQL commands, such as ADD, DROP, INSERT, and UPDATE can be used with MySQL.

A database is a data structure that stores organized information. Most databases contain multiple tables, which may each include several different fields. For example, a company database may include tables for products, employees, and financial records. Each of these tables would have different fields that are relevant to the information stored in the table. Today's relational databases allow users to access, update, and search information based on the relationship of data stored in different tables. Relational databases can also run queries that involve multiple databases. While early databases could only store text or such numeric data, modern databases also let users store other data types such as sound clips, pictures, and videos.

SQL (Structured Query Language) is a standardized programming language used for managing relational databases and performing various operations on the data in them. Initially created on 1970s, SQLs is regularly being used by database administrators, as well as by developers writing data integration scripts and data analysts looking to set up and run analytical queries. For example, books information, client information etc.

2.2.2 PHP

PHP[6] is a server-side scripting language, especially suited for the creation of dynamic web pages. This programming language offers web developers a large selection of instruments. PHP, which has become the basis for many web applications, allows easy insertion in HTML code and connection to MySQL and PgSQL Databases.

PHP was created in 1995 by Rasmus Lerdorf and means Personal Home Page language. Ever since its beginning it has been one of the most intuitive programming languages used for the creation of dynamic web pages. Later, when the PHP team was formed, the parser was rewritten and PHP version 3.0 saw the

light of day. The PHP 3 version was the first serious improvement of the PHP language, which fixed a lot of bugs and completely redesigned the PHP core. The meaning of the abbreviation was also changed to PHP Hypertext Preprocessor.

The launch of PHP4 marked serious progress in the usage of the programming language. This is the first version powered by the Zend Engine, which allows working with encoded files through the Zend Optimizer.

The latest, so far, version 5 introduced a completely reworked, advanced object-oriented programming support, as well as many improvements towards better productivity and security. PHP 5 is an attempt to break from the PHP 3 era, while still offering almost complete backwards compatibility. Unfortunately, this PHP version still does not offer native support for Unicode or multibyte strings.

NTC Hosting offers its clients an ultimate PHP hosting solution with the possibility to choose the older, but proven PHP4, or to easily upgrade to the newer PHP5. To change the PHP version, simply go to the PHP Settings menu in our Web Hosting Control Panel and select your preferred PHP revision.

2.2.3 HTML

HTML[5] (Hypertext Markup Language) is a text-based approach to describing how content contained within an HTML file is structured. This markup tells a web browser how to display the text, images and other forms of multimedia on a webpage.

The role of HTML is to inform a web browser about how the content contained within an HTML file is structured. Commonly used HTML tags include <H1>, which describes a top-level heading; <H2>, which describes a second-level heading; <p> to describe a paragraph; <table>, which describes tabular data; and , which describes an ordered list of information.

In the early days of the world wide web, marking up text-based documents using HTML syntax was more than sufficient to facilitate the sharing of academic documents and technical memos. However, as the internet expanded beyond the walls of academia and into the homes of the general population, greater demand was placed on webpages in terms of formatting and interactivity.

HTML 4.01 was released in 1999, at a time when the internet was not yet a household name, and HTML5 was not standardized until 2014. During this time, HTML markup drifted from the job of simply describing the structure of the content on a webpage into the role of also describing how content should look when a webpage displays it.

As a result, HTML4-based webpages often included information within a tag about what font to use when displaying text, what color should be used for the background and how content should be aligned.

Describing within an HTML tag how an HTML element should be formatted when rendered on a webpage is considered an HTML antipattern. HTML should describe how content is structured, not how it will be styled and rendered within a browser.

For rendering, the proper practice is to use cascading style sheets (CSS). An HTML file can link to a cascading style sheet, which will contain information about which colors to use, which fonts to use and other HTML element rendering information. Separating information about how a page is structured, which is the role of HTML, from the information about how a webpage looks when it is rendered in a browser, which is the role of a CSS file, is a software development pattern and best practice known as separation of concerns.

2.2.4 CSS

CSS[7] (Cascading Style Sheets) is used to style and layout web pages-for example, to alter the font, colour, size and spacing of your content, split it into multiple columns, or add animations and other decorative features. This module gets you started on the path to CSS mastery with the basics of how it works, including selectors and properties, writing CSS rules, applying CSS to HTML, how to specify length, colour, and other units in CSS, cascade and inheritance, and debugging CSS.

In this module we start off with a theoretical grounding, looking at what CSS is, how the browser turns HTML into a DOM, how CSS is applied to parts of the DOM, some very basic syntax examples, and what code is used to actually include our CSS in our web page.

2.2.5 JavaScript

JavaScript[8] works on the Client Side (on the user's computer).

The primary purpose of JavaScript is to provide a better experience for the user. It manipulates the objects within the HTML document. The objects and HTML elements collectively form what we call the Document Object Model (DOM). We will get into XHTML and XML later, but for now, we are going to use JavaScript with the HTML document and Cascading Styles Sheets (CSS).

A little bit of history on JavaScript. Netscape Communications Corporation invented JavaScript back in 1995. Many different languages (from C to Python) influenced the syntax and structure of JavaScript. JavaScript is arguably one of the most flexible languages ever created. However, JavaScript is one of those languages that people claim to know, but have absolutely no idea how complex JavaScript can get. We won't dive super far into JavaScript, but we will build a fundamental base for you.

2.2.6 Apache server

Apache[4] is an open-source and free web server software that powers around 46% of websites around the world. The official name is Apache HTTP Server, and it's maintained and developed by the Apache Software Foundation.

It allows website owners to serve content on the web-hence the name "web server". It's one of the oldest and most reliable web servers, with the first version released more than 20 years ago, in 1995.

When someone wants to visit a website, they enter a domain name into the address bar of their browser. Then, the web server delivers the requested files by acting as a virtual delivery man.

Here at Hostinger, our web hosting infrastructure uses Apache in parallel with NGINX, which is another popular web server software. This particular setup allows us to get the best of both worlds. It greatly improves server performance by compensating the weaker sides of one software with the strengths of another.

File servers, database servers, mail servers, and web servers use different kinds of server software. Each of these applications can access files stored on a physical server and use them for different purposes.

The job of a web server is to serve websites on the internet. To achieve that goal, it acts as a middleman between the server and client machines. It pulls content from the server on each user request and delivers it to the web.

The biggest challenge of a web server is to serve many different web users at the same time each of whom is requesting different pages. Web servers process files written in different programming languages such as PHP, Python, Java, and others.

They turn them to static HTML files and serve these files in the browser of web users. When you hear the word web server, think of it as the tool responsible for the proper server-client communication.

2.2.7 Visual Studio Code

Visual Studio Code[9] combines the simplicity of a source code editor with powerful developer tooling, like IntelliSense code completion and debugging.

First and foremost, it is an editor that gets out of your way. The delightfully frictionless edit-build-debug cycle means less time fiddling with your environment, and more time executing on your ideas.

Visual Studio Code supports macOS, Linux, and Windows - so you can hit the ground running, no matter the platform.

At its heart, Visual Studio Code features a lightning fast source code editor, perfect for day-to-day use. With support for hundreds of languages, VS Code helps you be instantly productive with syntax highlighting, bracket-matching, auto-indentation, box-selection, snippets, and more. Intuitive keyboard shortcuts, easy customization and community-contributed keyboard shortcut mappings let you navigate your code with ease.

For serious coding, you'll often benefit from tools with more code understanding than just blocks of text. Visual Studio Code includes built-in support for IntelliSense code completion, rich semantic code understanding and navigation, and code refactoring.

And when the coding gets tough, the tough get debugging. Debugging is often the one feature that developers miss most in a leaner coding experience, so we made it happen. Visual Studio Code includes an interactive debugger, so you can step through source code, inspect variables, view call stacks, and execute commands in the console.

VS Code also integrates with build and scripting tools to perform common tasks making everyday workflows faster. VS Code has support for Git so you can work with source control without leaving the editor including viewing pending changes diffs.

Customize every feature to your liking and install any number of third-party extensions. While most scenarios work "out of the box" with no configuration, VS Code also grows with you, and we encourage you to optimize your experience to suit your unique needs. VS Code is an open-source project so you can also contribute to the growing and vibrant community on GitHub.

VS Code includes enriched built-in support for Node.js development with JavaScript and TypeScript, powered by the same underlying technologies that drive Visual Studio. VS Code also includes great tooling for web technologies such as JSX/React, HTML, CSS, SCSS, Less, and JSON.

Architecturally, Visual Studio Code combines the best of web, native, and language-specific technologies. Using Electron, VS Code combines web technologies such as JavaScript and Node.js with the speed and flexibility of native apps. VS Code uses a newer, faster version of the same industrial-strength HTML-based editor that has powered the "Monaco" cloud editor, Internet Explorer's F12 Tools, and other projects. Additionally, VS Code uses a tools service architecture that enables it to integrate with many of the same technologies that power Visual Studio, including Roslyn for .NET, TypeScript, the

Visual Studio debugging engine, and more.

Visual Studio Code includes a public extensibility model that lets developers build and use extensions, and richly customize their edit-build-debug experience.

2.3 Chapter Summary

This chapter helps to understand basic idea of different tools and languages. Next Chapter gives us design concept of our House Rental Management System.

3 System design

System design [10] is the process of defining the elements of the system such as the architecture, modules and components, the different interfaces of those components and the data that goes through the system. It is meant to satisfy specific needs and requirements of a business or organization through the engineering of a coherent and well-running system. Systems design implies a systematic approach to the design of a system. It may take a bottom-up or top-down approach, but either way the process is systematic wherein it takes into account all related variables of the system that needs to be created from the architecture, to the required hardware and software, right down to the data and how it travels and transforms throughout its travel through the system. Systems design then overlaps with systems analysis, systems engineering and systems architecture. The systems design approach first appeared right before World War II, when engineers were trying to solve complex control and communications problems. They needed to be able to standardize their work into a formal discipline with proper methods, especially for new fields like information theory, operations research and computer science in general.

3.1 Architectural design

A system architecture [11] design is the conceptual model that defines the structure, behavior, and more views of a system. An architectural description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system. One can think of system architecture as a set of representations of an existing (or future) system. These representations initially describe a general, high-level functional organization, and are progressively refined to more detailed and concrete descriptions. System architecture conveys the informational content of the elements consisting of a system, the relationships among those elements, and the rules governing those relationships. The architectural components and set of relationships between these components that an architecture description may consist of hardware, software, documentation, facilities, manual procedures, or roles played by organizations or people. In this part, system block diagram details are given. Various organizations can define systems architecture in different ways, including:

- The fundamental organization of a system, embodied in its components, their relationships to each other and to the environment, and the principles governing its design and evolution.
- An allocated arrangement of physical elements which provides the design solution for a consumer product or life-cycle process intended to satisfy the requirements of the functional architecture and the requirements baseline.
- An architecture consists of the most important, pervasive, top-level, strategic inventions, decisions, and their associated rationales about the overall structure (i.e) essential elements and their relationships) and associated characteristics and behavior.
- A formal description of a system, or a detailed plan of the system at component level to guide its implementation.

- The composite of the design architectures for products and their life-cycle processes.
- The structure of components, their interrelationships, and the principles and guidelines governing their design and evolution over time.

According to requirements gather, the House Rental Management System will be designed as below:

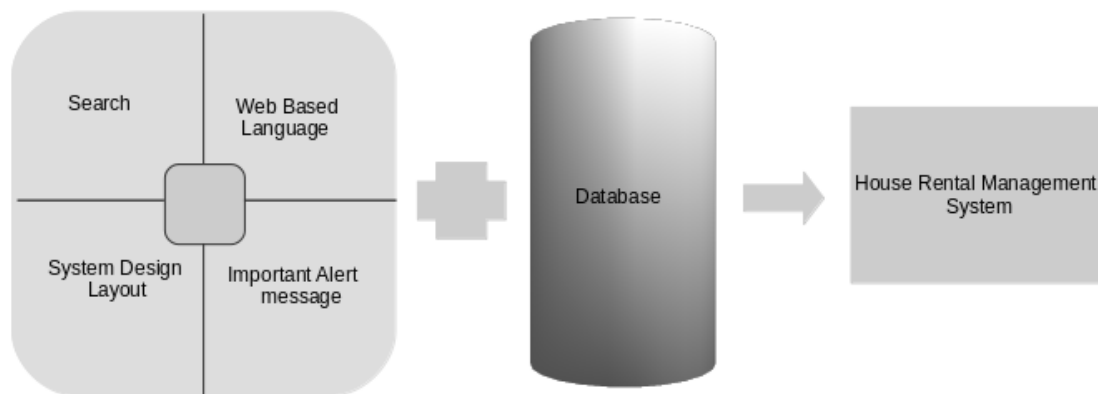


Fig-3.1 Architecture of House Rental Management System

As shown Fig-3.1, the tools has been used to built the architecture of the system is html (hypertext markup language), javascript, css (cascading style sheet), php and mysql.

We have tried to achieve the following features in our select problem domain of designing a modern dynamic home rental application-

- Rental search
- Rental posting
- Chat server
- Wish list
- Mail alert system
- Google map view
- A responsive application of the system wev view.

3.2 Database diagram

Within a database diagram [12], each relationship can appear with three distinct features: endpoints, a line style, and related tables.

Endpoints The endpoints of the line indicate whether the relationship is one-to-one or one-to-many. If a relationship has a key at one endpoint and a figure-eight at the other, it is a one-to-many relationship. If a relationship has a key at each endpoint, it is a one-to-one relationship.

Line Style The line itself (not its endpoints) indicates whether the Database Management System (DBMS) enforces referential integrity for the relationship when new data is added to the foreign-key table. If the line appears solid, the DBMS enforces referential integrity for the relationship when rows are added or modified in the foreign-key table. If the line appears dotted, the DBMS does not enforce referential integrity for the relationship when rows are added or modified in the foreign-key table.

Related Tables The relationship line indicates that a foreign-key relationship exists between one table and another. For a one-to-many relationship, the foreign-key table is the table near the line's figure-eight symbol. If both endpoints of the line attach to the same table, the relationship is a reflexive relationship.

3.2.1 Entity diagram

An Entity Relationship (ER) Diagram [13] is a type of flowchart that illustrates how “entities” such as people, objects or concepts relate to each other within a system. ER Diagrams are most often used to design or debug relational databases in the fields of software engineering, business information systems, education and research. Also known as ERDs or ER Models, they use a defined set of symbols such as rectangles, diamonds, ovals and connecting lines to depict the interconnectedness of entities, relationships and their attributes. They mirror grammatical structure, with entities as nouns and relationships as verbs. As shown in Fig-3.2 if admin wants to check the order list from an client and add new house, admin must first signup with name, user name, email and password. After registration, he will be able to check the order list after signin, else he can not even see the order list and he will not be able to add house to the house list .

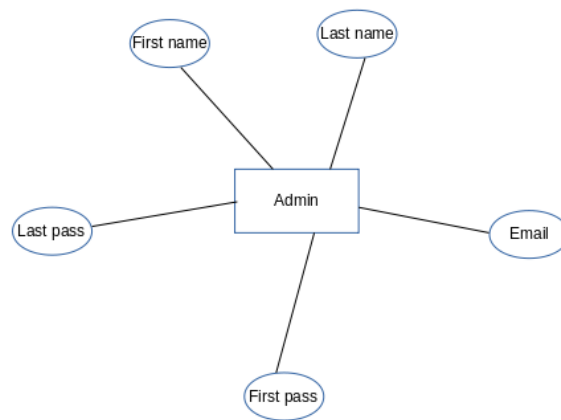


Fig-3.2 Admin entity diagram

As shown in Fig-3.3 client can browse about house name, price and details from house list. But if client wants to place an order, client must need to sign up first with client name, username, e-mail and password and then signin with his username and password. After logging in, he can order the house from the house category. After ordering, client can be able to signout.

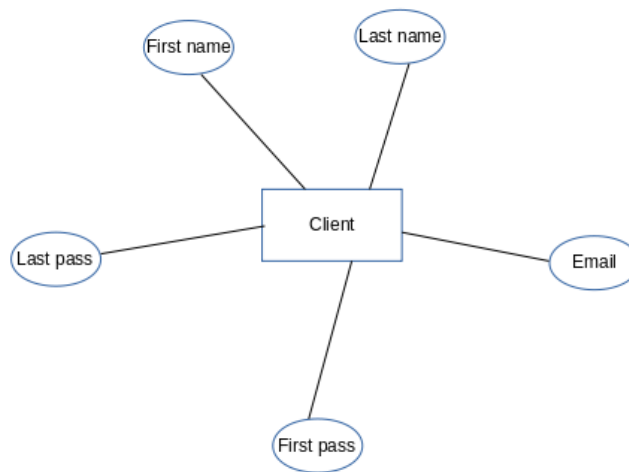


Fig-3.3 Client entity diagram

As shown in Fig-3.4 if admin wants to add new house items to a house list, admin must first signup with name, user name, email and password. After registration, he will be able to check the order list after signin, else he can not even see the order list and he will not be able to add house to the house list.

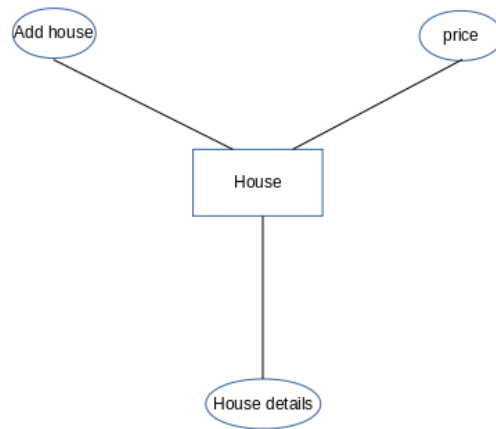


Fig-3.4 House add entity diagram

As shown in Fig-3.5 Client can comment on each house's and also can give reviews for a specific item of house.

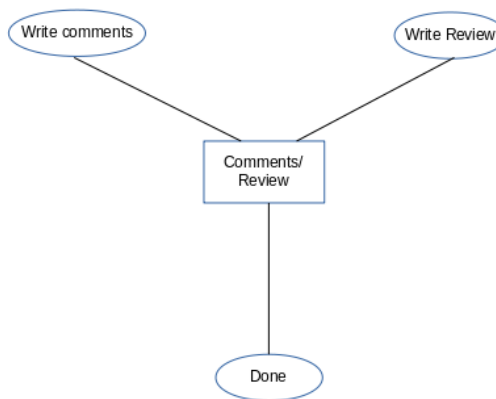


Fig-3.5 Comments/Review entity diagram

3.2.2 E-R diagram

An entity relationship diagram (ERD) [14], also known as an entity relationship model, is a graphical representation of an information system that depicts the relationships among people, objects, places, concepts or events within that system. An ERD is a data modeling technique that can help define business processes and be used as the foundation for a relational database.

Entity relationship diagrams provide a visual starting point for database design that can also be used to help determine information system requirements throughout an organization. After a relational database

is rolled out, an ERD can still serve as a referral point, should any debugging or business process re-engineering be needed later.

However, while an ERD can be useful for organizing data that can be represented by a relational structure, it can't sufficiently represent semi-structured or unstructured. It's also unlikely to be helpful on its own in integrating data into a preexisting information system.

we have described our website. Our website has two parts one for landlords and one for Renters. To upload their house a landlord, need to sign up in our website as landlord if someone doesn't sign up he/she can not upload their house or hotel. Once the house is rented to someone they can delete it from the website. On the other hand, Rent seekers also need to have a account on our website to take rent. They can not take any rent without singing up in our website. To match there's preferences they can search by their preferences like location, number of room, number of family member etc.

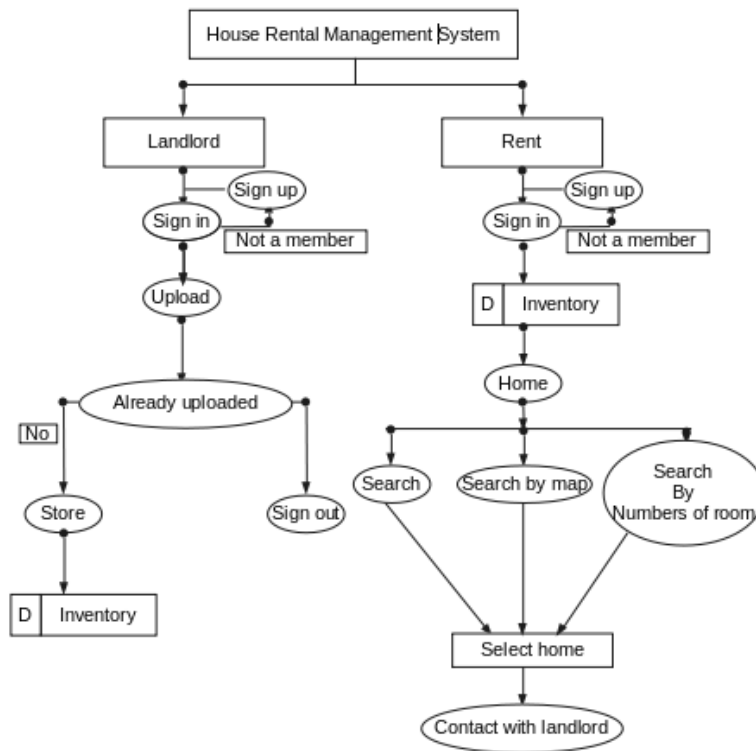


Fig-3.6 ER diagram of House Rental Management System

3.2.3 Landlords

In fig-3.7 we have described our activity diagram of landlord. A landlord need to sign up an account on our website to upload their house for rent. They also need to give a description of their house like where its located, how many rooms available, do they allow bachelors or not etc. If they get a renter which meets their desires they need to mark the house as unavailable for others.

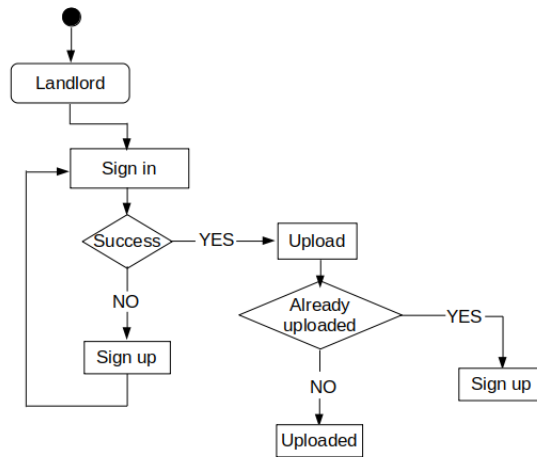


Fig-3.7 Activity diagram of Landlords

3.2.4 Rent Seekers

In fig-3.8 we have described the activity diagram of rent seekers. A rent seeker also should have an account on our website in term to rent any house from our website. After signing in a rent seeker can search randomly or they can search by their desire wishes like any specific location, by the number of rooms, bachelor or family etc. Once they find their desired house they can chat with the landlord of that house or they can also contact with the admin of our website.

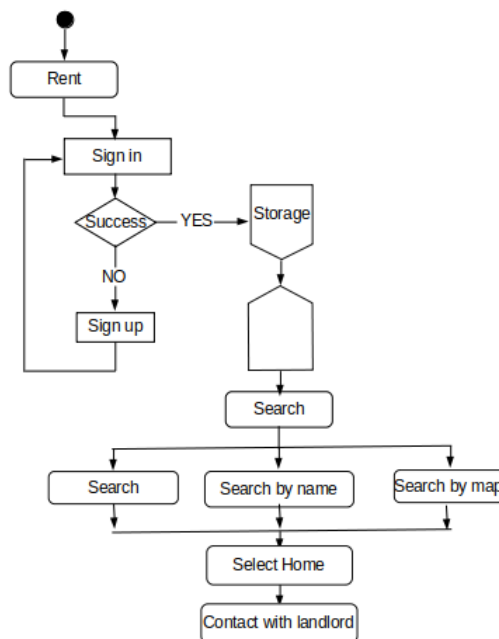


Fig-3.8 Activity diagram of Rent seekers

3.3 Incremental Model

In fig-3.9 Incremental model[15] the whole requirement is divided into various builds. Multiple development cycles take place here, making the life cycle a “multi-waterfall” cycle. Cycles are divided up into smaller, more easily managed modules. Incremental model is a type of software development model like V-model, Agile model etc.

In this model, each module passes through the requirements, design, implementation and testing phases. A working version of software is produced during the first module, so you have working software early on during the software life cycle. Each subsequent release of the module adds function to the previous

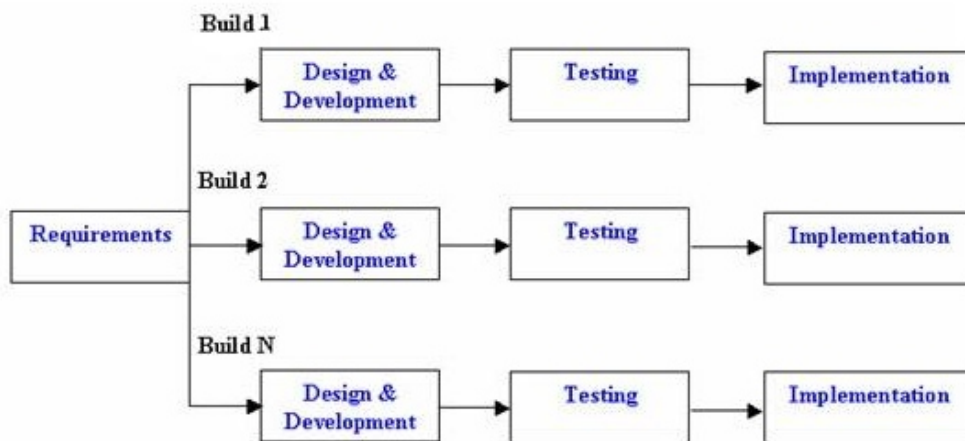


Fig-3.9 Incremental Model for House Rental Management System

3.4 Database schema

A Database schema [16] represents the logical configuration of all or part of a relational database. It can exist both as a visual representation and as a set of formulas known as integrity constraints that govern a database. These formulas are expressed in a data definition language, such as SQL. As part of a data dictionary, a database schema indicates how the entities that make up the database relate to one another, including tables, views, stored procedures, and more. Typically, a database designer creates a database schema to help programmers whose software will interact with the database. The process of creating a database schema is called data modeling. When following the three-schema approach to database design, this step would follow the creation of a conceptual schema. Conceptual schemas focus on an organization’s informational needs rather than the structure of a database.

- **Registration for admin(u-id, u-f, u-l, u-u, um, ufp, usp, status):** For admin registration have a database table which have eight columns, first column is used for unique id, second column is used for admin’s first name, third column use for admin’s last name, fourth column is used for admin’s user name, fifth column is used for admin’s email, sixth and seventh columns is used for admin’s password and last column is used for admin is active or not.

- **Registration(uid, uf, ul, uu, um, ufp, usp, status):** Registration is database table which have eight columns, first column is used for unique id, second column is used for client's first name, third column use for client's last name, fourth column is used for client's user name, fifth column is used for client's email, sixth and seventh columns is used for client's password and last column is used for client is active or not.
- **House upload(house name, house rent price, house details, house rent price):** For upload a house, we have a database table where given by details of house then save it.
- **Comments(house-id, comments):** For user comments we use a table, where store the house id and user comments.
- **Review(house-id, review):** For user review we use a table, where store the house id and user review.

3.4.1 Database tables structures

The database structure is the collection of record type and field type definitions that comprise your database. These define the type of entities or research objects you wish to capture (e.g. Person) Fields. These are the properties or attributes that describe your record types (e.g. Gender, Age, Height etc).

The Tab.3.1 is used for signup a user as a admin by giving first name, last name, email and password on the application. If admin already have an account then signin in the application then manipulate the order list or add house on house list by given house name, house price and details of house. In Tab.3.1 u-id means user unique id, which type is integer and default value is not null, u-f-name means user first name which type is varchar, size is 80 and default value is null, u-l-name means user last name which type is varchar, size is 80 and default value is null, u-mail means user email which type is varchar, size is 50 and default value is null, u-f-pass means first password which type is varchar, size is 20 and default value is null, u-l-pass means last password which type is varchar, size is 20 and default value is null,

Tab-3.1 Admin registration

| No | Field | Type | Null | Default | Extra | Description |
|----|----------|-------------|------|---------|----------------|---------------------------------|
| 1 | u-id | int(3) | No | None | Auto-increment | Stores the admin id |
| 2 | u-f-name | varchar(80) | Yes | None | | Stores the admin first name |
| 3 | u-l-name | varchar(80) | Yes | None | | Stores the admin last name |
| 5 | u-mail | varchar(50) | Yes | None | | Stores the admin email |
| 6 | u-f-p | varchar(50) | Yes | None | | Stores the admin first password |
| 7 | u-l-p | varchar(50) | Yes | None | | Stores the admin last password |

The Tab.3.2 is used for signup a user as a client by giving first name, last name, email and password on the application. If client already have an account then signin in the application then select the house from house list and before confirm order must signup by given personal information after signup client signin in the system by given client email and password. In table 3.2 u-id means user unique id, which type is integer and default value is not null, u-f-name means user first name which type is varchar, size is 80 and default value is null, u-l-name means user last name which type is varchar, size is 80 and default value is null, u-mail means user email which type is varchar, size is 50 and default value is null, u-f-pass means first password which type is varchar, size is 20 and default value is null, u-l-pass means last password which type is varchar, size is 20 and default value is null,

Tab-3.2 Registration for client

| No | Field | Type | Null | Default | Extra | Description |
|----|----------|-------------|------|---------|----------------|----------------------------------|
| 1 | u-id | int(3) | No | None | Auto-increment | Stores the client id |
| 2 | u-f-name | varchar(80) | Yes | None | | Stores the client first name |
| 3 | u-l-name | varchar(80) | Yes | None | | Stores the client last name |
| 5 | u-mail | varchar(50) | Yes | None | | Stores the client email |
| 6 | u-f-p | varchar(50) | Yes | None | | Stores the client first password |
| 7 | u-l-p | varchar(50) | Yes | None | | Stores the client last password |

The Tab.3.3 is used for hold the name of the list of house and total price of house choosing by client and name of client. Here col-id means list of house's unique id which type is integer and size is 3, col-name means name of house, which type is varchar and size is 255 and default value is null, col-price means price of house which type is float and user email means which client order the house and user email type is varchar, size is 255 and default value is null.

Tab-3.3 Client order list

| No | Field | Type | Null | Default | Extra | Description |
|----|-----------|--------------|------|---------|----------------|---|
| 1 | col-id | int(3) | No | None | Auto-increment | Stores the order id |
| 2 | col-name | varchar(255) | Yes | None | | Stores the name of house |
| 3 | col-price | float(10, 2) | Yes | None | | Stores the total price |
| 4 | username | varchar(255) | Yes | None | | Stores the client name who is order house |

3.5 Chapter Summary

This chapter has displayed many graphical representations of the design of the system. The implementation of the system is documented in the next chapter.

4 Implementation & Testing

4.1 Chapter overview

Implementation is the process in the project in which the existing design is changed into working system and is giving assurance confidence on the new system for the users that it will work correctly. It involves careful planning, checkout of the current system and its compulsion or constraints of design, implementation of methods to achieve the changeover, an evaluation of the changeover methods. Apart from it planning major tasks of arranging the implementation are educating and training of users. The more complex the system is being implemented the more involved will be the system analysis and the design efforts requires just for the implementation. The implementation process begins at preparing a plan for the implementation of the system.

The implementation[17] process (putting the network design into practice) is probably the most critical stage of the project, as it requires a huge commitment in terms of manpower and financial resources, and can be quite disruptive of the day to day operation of the organization. The main activities that you might reasonably expect to be undertaken during implementation are listed below:

- installation and testing of network cabling.
- installation of network equipment accommodation.
- installation, configuration and testing of network hardware.
- creation of the network file system.
- creation of a suitable domain structure.
- configuration and testing of network services.
- transfer of data from existing to new system.
- user training.
- final system testing.

4.2 User Module

The user module[18] allows users to register, sign in, and sign out. Users benefit from being able to sign on because this associates content they create with their account and allows various permissions to be set for their roles. The user module supports user roles, which can be set up with fine-grained permissions allowing each role to do only what the administrator permits. Each user is assigned one or more roles. By default there are three roles: anonymous (a user who has not logged in) and authenticated (a user who is registered), and administrator (a signed in user who will be assigned site administrator permissions). Users can use their own name or handle and can fine tune some personal configuration settings through their individual my account page. Registered users need to authenticate by supplying their username and password, or alternately an open id sign in.

4.2.1 User Interface

In fig-4.1 a user need to log in to use the application. After logging in he/she can use it for renting and they can also use to upload their house for rent.

If the user is not yet a registered member he/she will be able to register providing the details mentioned in the figure. The user will not be directed to the homepage unless he or she signs in.

Already signpued user will have to signin in the system by given user name and password in order to enter the system. If user name is not matched with the name and password which are hold on database by creating an account, then show an error that you are given wrong information. If user name matched but password not mathed then show an error message that enter correct information. The password field is encrypted by star character.

A new user will have to sign up in the system by providing essential details in order to be a valid user of the restaurant. The essential details are user name, first name, last name, email, password. If user name already signup then show an dialog message that you hava already an account.If user name is not already exist then successfully create an account.

The figure displays two screenshots of a web application's user interface. The top screenshot, titled 'Log in to your account', features a central white box on a light gray background. It contains an 'Email' input field, a 'Password' input field with a 'SHOW' button, a 'Forgot your password?' link, a blue 'SIGN IN' button, and a 'No account? Create one here' link at the bottom. The bottom screenshot, titled 'Create an account', also has a light gray background. It includes a header 'Already have an account? Log in instead!' and a 'Social title' section with radio buttons for 'Mr.' and 'Mrs.'. Below this are input fields for 'First name', 'Last name', and 'Email'. The 'Password' field has a 'SHOW' button. A 'Birthdate' field is shown with a date picker (MM/DD/YYYY) and a note '(E.g.: 05/31/1970)'. There are two checkboxes: 'Receive offers from our partners' and 'Sign up for our newsletter'. A small disclaimer text is present below the second checkbox. A blue 'SAVE' button is located at the bottom right of the form.

Fig-4.1 User Interface

4.2.2 Results of Search feature

A search term[19], or search query, is the word or phrase someone enters into a search engine to search for on the internet. These can be single words or names, like [Yeast], or a combination of words,

such as [buy robot vacuum cleaner], or even complete questions, like [how do I train my puppy?]. Note that each individual could phrase his or her query uniquely, so five people looking for the exact same thing may just use five completely different search terms. Moreover, search terms can contain spelling errors, numbers substituting words, or random word order, and still take someone where they need to be.

While everyone entering a search term into a search engine is trying to find something, their intent can certainly differ. It's important to be aware of this. Someone typing [how do I train my puppy?] into Google is looking for information, whereas someone who enters [Yoast] probably wants to navigate to Yoast.com. And someone looking for [buy robot vacuum cleaner] is, evidently, looking to buy a robot vacuum sometime in the near future.

If the page someone clicks on in the search results doesn't match their intent, they're likely to bounce back and look for another page that does align with their purpose. So, you'll understand that it's important, as a site owner, to make sure people end up on a page that's consistent with their expectations and what they're trying to achieve!

In fig-4.2 as mentioned in the previous chapter of implementation we explained how one can opt for a choice of priority. Based on the selected priority the results will vary in some extent.

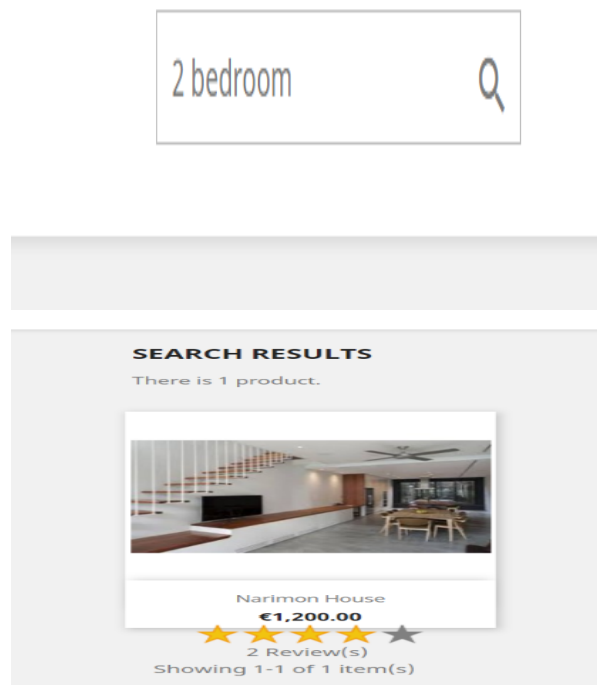


Fig-4.2Search User Interface

4.2.3 Goggle Map

Google Maps[20] is a Web-based service that provides detailed information about geographical regions and sites around the world. In addition to conventional road maps, Google Maps offers aerial and

satellite views of many places. In some cities, Google Maps offers street views comprising photographs taken from vehicles.

Google Maps offers several services as part of the larger Web application, as follows.

- A route planner offers directions for drivers, bikers, walkers, and users of public transportation who want to take a trip from one specific location to another.
- The Google Maps application program interface (API) makes it possible for Web site administrators to embed Google Maps into a proprietary site such as a real estate guide or community service page.
- Google Maps for Mobile offers a location service for motorists that utilizes the Global Positioning System (GPS) location of the mobile device (if available) along with data from wireless and cellular networks.
- Google Street View enables users to view and navigate through horizontal and vertical panoramic street level images of various cities around the world.
- Supplemental services offer images of the moon, Mars, and the heavens for hobby astronomers.

In fig-4.3 the area shows the location of the user the user has provided the system while searching for rentals. Other the pointers show the results of the search based on area focusing the area Niketon. Upon clicking the Details button attached with every rental shown in the box the user will be redirected to page where not only the user can see the details of the rental including pictures but the user can also search for the transport by manually providing the location as we have not integrated GPS system to automatically extract the user location yet.

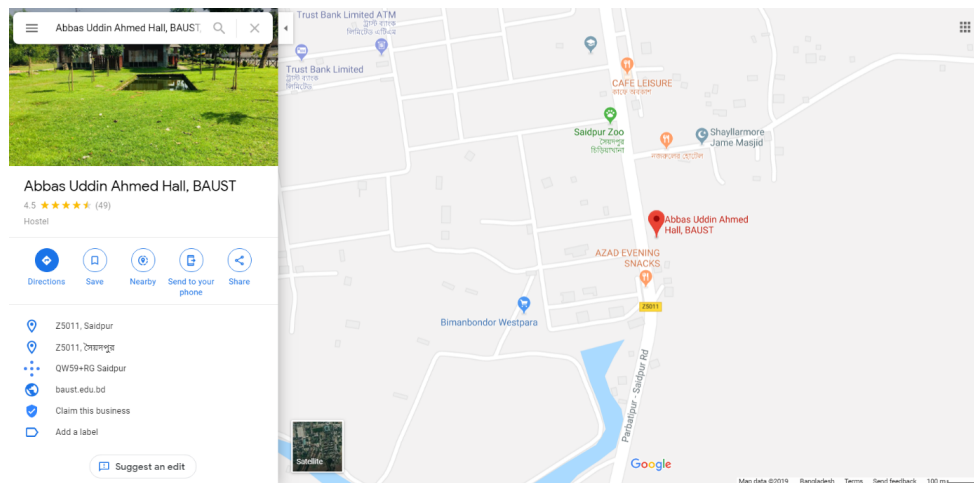


Fig-4.3 Architecture of House Rental Management System

4.3 Test Case

A Test case[21] is an automated test that addresses one objective of a test plan.

A test case:

- Drives the application from the initial state to the state you want to test.
- Verifies that the actual state matches the expected (correct) state. Your QA department might use the term baseline to refer to this expected state. This stage is the heart of the test case.
- Cleans up the application, in preparation for the next test case, by undoing the steps performed in the first stage.

In order for a test case to function properly, the application must be in a stable state when the test case begins to execute. This stable state is called the base state. The recovery system is responsible for maintaining the base state in the event the application fails or crashes, either during the execution of a test cases or between test cases.

Each test case is independent and should perform its own setup, driving the application to the state that you want to test, executing the test case, and then returning the application to the base state. The test case should not rely on the successful or unsuccessful completion of another test case, and the order in which the test case is executed should have no bearing on its outcome. If a test case relies on a prior test case to perform some setup actions, and an error causes the setup to fail or, worse yet, the application to crash, all subsequent test cases will fail because they cannot achieve the state where the test is designed to begin.

A test case has a single purpose: a single test case should verify a single aspect of the application. When a test case designed in this manner passes or fails, it is easy to determine specifically what aspect of the target application is either working or not working.

If a test case contains more than one objective, many outcomes are possible. Therefore, an exception may not point specifically to a single failure in the software under test but rather to several related function points. This makes debugging more difficult and time-consuming and leads to confusion in interpreting and quantifying results. The result is an overall lack of confidence in any statistics that might be generated. But there are techniques you can use to perform more than one verification in a test case.

A Test cases a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works correctly. The process of developing test cases can also help find problems in the requirements or design of an application.

Test cases were created to test adding, deleting, and editing both house items. Specifically these test cases make certain that houses are stored and retrieved from the database correctly. Test cases were also generated to perform boundary testing on how many entries could be successfully added or updated. In addition, test cases where created to verify the function of the compare class, which is used to validate input.

Tab-4.1 Test Cases

| Scenario | Test Cases | Expected Result | Test Result | Pass or Fail |
|------------------|--|---|---|--------------|
| Registration | Enter null in mandatory fields. | It should not do the registration and show error. | It will show message that please fill out this field. | Pass |
| | Enter incorrect data | It should not do the registration and show error. | Email: Please enter an email. Password: Password. do not match | Pass |
| | Enter correct data of all required field | It should lead to successfully registration | It will show the message of successfully registrattion | Pass |
| Login | Enter null email or password. | It should not do the login and show error. | It will show message that please fill out this field. | Pass |
| | Enter wrong data of email or password. | It should not do the login and show error. | It will show message that please enter an email or your email address or password are not correct.. | Pass |
| | Enter correct data of email or password. | It should let do the login. | It will redirect to your all purchased products | Pass |
| Update password | Enter null in mandatory fields. | It should not do the update and show error. | It will show message that please fill out this field. | Pass |
| | Enter incorrect data | It should not do the update and show error. | It will show the message that the old password do not match. | Pass |
| | Enter correct data of all required field. | It should let do registration. | It will show the message that password successfully updated. | Pass |
| Search House | Enter null or incorrect data in search fields. | It will not search house | It will show the message that there was no search results! | Pass |
| | Enter correct data in Search fields. | It should search houses depending the keywords. | It will display the houses regarding the keywords. | Pass |
| Comment on house | Enter non ASCII character. | It will not let to comment | It will show error as unsupported character. | Pass |
| | Enter ASCII character. | It will lead a successful comment. | It will view the succession of a comment. | Pass |

4.4 Version Control

Due to the type of development methodology used for this project, incremental backups of the system were required. Version control systems (also known as Revision control) such as Mercurial manage the changes to documents storing each backup in its own revision with the ability to restore back to a particular

version in the event of debugging.

4.5 Code Documentation

Code documentation is an important part of any software engineering project. Throughout the implementation, a PHPDoc tool was used to generate HTML API documentation of the project. PHPDocumentor two[22] is a tool that makes it possible to generate documentation directly from your PHP source code. With this you can provide your consumers with more information regarding the functionality embedded within your source and not just what is usable to them from your user interface. Documentation generated by phpDocumentor 2 does not aim to be a replacement for conventional documentation but is rather supplemental, or reference, documentation. This proves to be useful in the following example situations:

4.6 Chapter summary

This chapter has displayed many graphical representations of the design of the system. The implementation of the system is documented in the next chapter.

Conclusion

In this following chapter, we sum our work in brief and conclude the paper. We also represent a future works which we could not able to do because of some contradictions we face during our research. We will work more on this and we have mentioned in the future work what we tend to o next.

- **Conclusion:** In this paper, we analyzed how the system can be made more dynamic. As the systems' main users are the general people so it has to be more user-friendly and dynamic. This will make life easier for the bachelors. People do not need to roam around now for renting they can do it with some little clicks.
- **Future Work:** We would like make some changes in our searching. We would like to make it more dynamic. We will add payment system and we would also like to make more use of google map for location finding.

On reflection, even though the majority of the proposed features were completed and the project was deemed a huge success, the author felt that he could have been more disciplined in keeping to the plan. He also felt that the proposed features were slightly unrealistic and some even unnecessary. For the general project, the author felt that important aspects of research were not undertaken including interviews with restaurant owners and user questionnaires. This would have provided good insight into existing solutions. This project has helped the author to attain new skills as well as develop existing skills. The skills attained have been both technical and individual with the main individual skill being project manage- ment which required good time keeping and management of the workload.

This chapter has concluded the project report and provided an insight into possible future development.

References

- [1] S. R. Manalu, "Development and evaluation of mobile application for room rental information with chat and push notification," 2006. [Online]. Available: <https://ieeexplore.ieee.org/document/7930293/authors>
- [2] "Deliverables are usually classified as internal deliverables and external deliverables." [Online]. Available: <https://www.simplilearn.com/what-is-a-deliverable-article>
- [3] "Mysql." [Online]. Available: <https://www.siteground.com/tutorials/php-mysql/mysql>
- [4] "What is apache? an in-depth overview of apache web server." [Online]. Available: <https://www.hostinger.com/tutorials/what-is-apache#gref>
- [5] "Html." [Online]. Available: <https://www.theserverside.com/definition/HTML-Hypertext-Markup-Language>
- [6] "Php." [Online]. Available: <https://www.ntchosting.com/encyclopedia/scripting-and-programming/php/>
- [7] "Css." [Online]. Available: https://developer.mozilla.org/en-US/docs/Learn/CSS/Introduction_to_CSS
- [8] "Javascript." [Online]. Available: <https://www.afterhoursprogramming.com/tutorial/javascript/javascript-overview/>
- [9] "Visual studio code." [Online]. Available: <https://code.visualstudio.com/docs/editor/whyvscode>
- [10] "What is an system design?" [Online]. Available: <https://www.techopedia.com/definition/29998/system-design>
- [11] "Systems architecture." [Online]. Available: https://en.wikipedia.org/wiki/Systems_architecture
- [12] "Database diagram." [Online]. Available: <https://docs.microsoft.com/en-us/sql/ssms/visual-db-tools/design-database-diagrams-visual-database-tools?view=sql-server-2017>
- [13] "What is an er diagram?" [Online]. Available: <https://www.lucidchart.com/pages/er-diagrams?a=0>
- [14] "Entity relationship diagram." [Online]. Available: <https://searchdatamanagement.techtarget.com/definition/entity-relationship-diagram-ERD>
- [15] "Incremental model." [Online]. Available: <http://tryqa.com/what-is-incremental-model-advantages-disadvantages-and-when-to-use-it/>
- [16] "Database schema." [Online]. Available: <https://www.lucidchart.com/pages/database-diagram/database-schema>

- [17] “Implementation and testing.” [Online]. Available: <http://www.technologyuk.net/telecommunications/networks/implementation-and-testing.shtml>
- [18] “User module.” [Online]. Available: <https://www.drupal.org/docs/8/core/modules/user/overview>
- [19] “Search term.” [Online]. Available: <https://yoast.com/search-term/>
- [20] “Goggle map.” [Online]. Available: <https://whatis.techtarget.com/definition/Google-Maps>
- [21] “Test case.” [Online]. Available: <http://documentation.microfocus.com/help/index.jsp?topic=%2Fcom.borland.silktest.classic.doc%2FSTCLASSIC-753381C7-ABOUT-TESTCASES.html>
- [22] “Php documentation.” [Online]. Available: <https://www.phpdoc.org/>

Acknowledgement

Foremost, we would like to express my sincere gratitude to the almighty Allah for His blessings and protection throughout our undergraduate studies.

our sincere thanks to our supervisor, **Al-Hasan** of the Institute of Computer Science and Engineering at Bangladesh Army University of Science and Technology for the continuous support of our Bachelor's study, patience, motivation, enthusiasm, and immense knowledge. His guidance helped us in all the time of research and writing of this thesis. We could not have imagined having a better advisor and mentor. We also thank our fellow classmates, our friends and for all the fun we have had in the last four years. Last but not the least; We would like to thank our parents for their prayers, support, encouragement and their admonishment.