**“Appartment Visitors Management System”**

**A PROJECT REPORT**

**Submitted in partial fulfillment of the**

**Requirements for the Degree of**

**MASTER OF COMPUTER APPLICATION**

**By**

**ABHISHEK JHA**

**University Roll No 1900290140001**

**Under the supervision of**

**Ms. Neelam Rawat**

**KIET Group of Institutions**



**Submitted to**

**Department Of Computer Applications**

**KIET Group of Institutions, Ghaziabad**

**Uttar Pradesh-201206**

**JUNE, 2022**

**Declaration**

I hereby declare that the work presented in this report entitled “**Apartment visitors management System**”, was carried out by me. I have not submitted the matter embodied in this report for the award of any other degree or diploma of any other University or Institute.

I have given due credit to the original authors/sources for all the words, ideas, diagrams, graphics, computer programs, experiments, results, that are not my original contribution. I have used quotation marks to identify verbatim sentences and given credit to the original authors/sources.

I affirm that no portion of my work is plagiarized, and the experiments and results reported in the report are not manipulated. In the event of a complaint of plagiarism and the manipulation of the experiments and results, I shall be fully responsible and answerable.

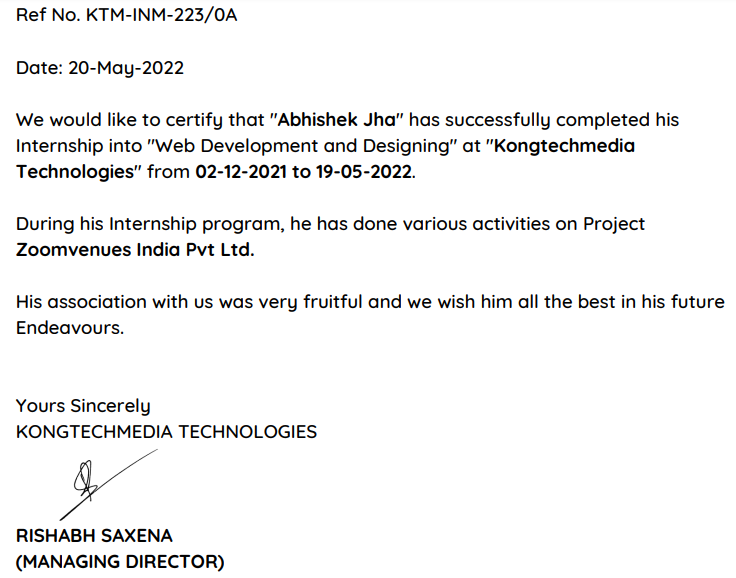
Name : Abhishek jha

R0ll no : 1900290140001

Branch : Masters of computer application

(Candidate signature)

56.PNG





**Abstract**

Apartment Visitor Management System deals with the security provided at society premises from the unauthorized or unwanted visitors. Nowadays, in most society visitor management consists of visitors scribbling their name in a paper book. Instead, Apartment Visitor Management System will assist you the professionalized way in which you welcome your visitors. This software is a complete Visitor Management service to improve the efficiency, productivity and security.

**ACKNOWLEDGEMENT**

I extend my deepest appreciation to Kongtech Media Technologies Pvt Ltdmy esteemed guide, for providing me with the possibility to complete this project with the right guidance and advice.

Special gratitude I give to my respected head of the division **Dr. Ajay Kr Srivastav, Head of Department, and all faculties Master of Computer Applications**, for allowing me to use the facilities available and also help me to coordinate my project

Furthermore, I would also like to acknowledge with much appreciation the crucial role of faculty members on this occasion.

Last but not least, I would like to thank friends who help me to assemble the parts and gave a suggestion about the project.

**TABLE OF CONTENT**

|  |  |  |
| --- | --- | --- |
|  | | |
|  | |  |
|  | | Declaration  Training Certificate |
|  | | Certificate |
|  | | Abstract |
|  | | Acknowledgement |
|  | | List of Figure |
| **CHAPTER 1-** | | **INTRODUCTION 9-11** |
| 1.1 | | INTRODUCTION 9 |
| 1.2 | | PURPOSE 10 |
| 1.3 | | SCOPE 11 |
| **CHAPTER 2-** | | **LITERATURE REVIEW 12-14** |
| 2.1 | | TECHNOLOGY USED 13 |
| 2.2 | | LANGUAGE USED 14 |
| **CHAPTER 3-** | | **SOFTWARE REQUIREMENT SPECIFICATION - 25-35** |
| 3.1 | | HARDWARE CONFIGURATION 25 |
| 3.2 | | SOFTWARE REQUIREMENT 26 |
| 3.3 | | FUCTIONAL REQUIREMENT 31 |
| 3.4  3.5  3.6  3.7  3.8  3.9 | | NON- FUNCTIONAL REQUIREMENT 31  SOFTWARE SYSTEM ATTRIBUTE 32  FEATURE 32  PRELIMINARY INVESTIGATION 33  APPROACH USED: AGILE APPROACHED 34  PRELIMINARY DESCRIPTION 35 |
| **CHAPTER 4-** | | **FEASIBILITY STUDY 36-42** |
| 4.1  4.2  4.3 | | ECONOMIC FEASIBILITY 36  TECHNICAL FEASIBILITY 36  OPERATINAL FEASIBILITY 36 |
| **CHAPTER 5-** | **PLANNING SCHEDULING AND FLOW 43-44** | | |  |
| 5.1  5.2 | GANT CHART 43  PERT CHART 44 | | |  |
| **CHAPTER 6- ANALYSING AND DESIGN 45-57**  6.1 ANALYSIS 45  6.2 DESIGN 47  6.3 UML DIAGRAM 49  6.4 USED CASE DIAGRAM 52  6.5 CLASS DIAGRAM 56  **CHAPTER 7- ER DIAGRAM AND DATA DICTIONARY 58-64**  7.1 ER NOTATION 59  7.2 ADMIN TABLE 62  7.3 VISITOR TABLE 64  **CHAPTER 8- IMPLEMENTATION AND SYSTEM TESTING 65-70**  8.1 SYSTEM TESTING 67  **CHAPTER 9- VALUATION 70-77** | | | |  |
| **CONCLUSION 78** | | | |  |
| **REFRENCES 80** | | | |  |
| **DATADICTIONARY** | | | |  |
| **CODING (DATA BASE)** | | | |  |

**List of Figures**

**Page no.**

Fig3.1AGILE APPROACH 34

Fig6.1 USER 50

Fig6.2 USED CASE 50

Fig6.3 USED CASE DIAGRAM OF ADMIN 54

Fig6.4CLASS DIAGRAM 56

Fig7.1 ER- DIAGRAM 64

Fig9.1 LOGIN PAGE 69

Fig9.2 FORGOT PASSWORD 70

Fig9.3 RESET PASSWORD 70

Fig9.4 DASH BOARD 71

Fig9.5 ADMIN PROFILE 71

Fig9.6 CHANGE PASSWORD 72

Fig9.7 ADD NEW VISITOR 73

Fig9.8 MANAGE VISITOR 73

Fig9.9 VISITOR DETAIL 74

Fig9.10 VISITOR DETAIL AFTER UPDATE 75

Fig9.11BETWEEN DATE REPORT 75

Fig9.12 ADMIN DETAIL 76

**Chapter 1**

**Introduction**

**1.1 Introduction:**

Apartment Visitor Management system is a web-based technology that will revolutionize the way your society manages visitors. Visitor Management system is more important to security guards or security society. This web application provides a way to effectively control record & track society visitor traffic.

In AVMS we use PHP and MySQL database. This is the project which keeps records of visitors who visited in the Society. **AVMS has one module i.e. admin**

* Dashboard: In this section, admin can briefly view how many visitors visited in a particular period.
* Visitors: In this section, admin adds new visitors by filling their information in add visitors sections and also view and manage visitor’s records. Admin also put visitors out time in the manage records section.
* Search: In this bar, admin can search a particular person by their name and phone number.
* Reports: In this section admin can generate visitor’s reports between two dates.

Admin can also update his profile, change password and recover password.

**1.2 Purpose:-**

The purpose of developing apartment visitor management system is to computerized the tradition way of visitors. Another purpose for developing this application is to generate the report automatically.

**1.3 Scope:-**

Apartment Visitor Management System project is developed as a web application and it will work over web.

The [visitor management software](https://queueme.io/visitor-management-system.html) is an extremely user-friendly system that helps you manage and track any visitors or guests to your apartment. It keeps track of anyone visiting your apartment, starting from guests and family members to maids and contract laborers. At times, we believe, our home is not at risk and do not pay a lot of attention to security systems. But security threats can arise at any time, and the [visitor management software](https://queueme.io/visitor-management-system.html) can help keep you and your loved ones safe and secure.

**Why is a Visitor Management System Beneficial for Apartments?**

An apartment complex is a gated community that receives plenty of visitors throughout the day. It can be difficult for the security to check each visitor, check their entry and exits manually. The security needs to call the apartment to verify the visitor to let them in. It not only requires more effort but is not deceit proof. So, it is important to have an automated system that can effectively track visitors to the gated apartment. The software will create a database of visitors that can be easily referred to in case of any problems or conflicts. Plus, the database will have all contact information of the visitors, which makes tracking more efficient, in case of any problem.

## User-Friendly Features of Visitor Management Software

The [visitor management system](https://queueme.io/visitor-management-system.html) is extremely efficient. It not only records the personal details of a visitor, but it also keeps track of entry and exit times along with the purpose of the visit. These records can prove to be very useful for identifying any visitor at a later date. It is particularly useful for catching culprits after a mishap.

The [visitor management software](https://queueme.io/visitor-management-system.html) is easy to use and can be used by any security personnel. It keeps track of every apartment owner and sends personalized notifications whenever a visitor arrives. The system also helps prevent the entry of strangers. In case of a threat, a pop-up alert or a warning message will be sent to the security staff within the premises. You can use the software to generate daily in-out passes for regular visitors or workers. The system also generates bulk passes during any occasion organized within the compound.

**Issues with Register based Visitor Management**

* The handwriting is often illegible making the information captured useless.
* Apartment Visitors often provide invalid phone numbers, which again renders the information useless.
* The Information captured in the Register is very difficult to search in case an Incident Investigation needs to be done.
* CCTV camera footages are only stored for 7-14 days. So in case, any criminal investigation is to be conducted beyond that, the manual register is the only source of information. Authentic visitors might be giving their correct details, but visitors with any wrong intentions would definitely write down incorrect information. During any investigation, this becomes a major roadblock.
* The information in the register is not interfaced with the Apartment Resident so they can raise an alert in case there is an anomaly (e.g., a Visitor daily enters taking the name of a locked flat) of register.
* Capturing the Visitor data in a Register with full history available right at the Gate makes the data vulnerable to misuse. This is also another reason that many Visitors do not provide their genuine details.
* Typical Gated Communities can have anywhere from 5 to 35 Registers! With the regular change in the Security Team personnel, there is high training effort to ensure the Guards know who to be entered in which Register.
* MC members or Estate Manager have no way to quickly audit if all Visitor Details are being captured, following the right process. They can only do so by physically sitting next to the Security Guards, at which time they will do their work well.
* With no surprise audit feasibility for Registers, gaping holes are left for mis-utilization related to material movement e.g., Water Tanker or Diesel Supply.
* While piles and piles of Visitor Registers are created, it is impossible to get any intelligence from the movement trends at the Gate which can be utilized towards Security optimization.
* While one Security Agency may place a certain set of Registers with certain processes to be followed, the handover of all these details becomes cumbersome to the subsequent Security Agency. As a result, the Management Committee may also be unaware of the specific reasons behind setting up of certain Registers and look for Best Practices in complexes with similar security requirements as theirs.

These are the some of the reasons why more and more Gated Communities are packing off their Registers and moving to an [App-Based Visitor Management system](http://addagatekeeper.io/).

**What is an Apartment Visitor Management System?**

[Visitor Management](https://addagatekeeper.io/) Systems are quite common in corporate setups and commercial buildings, where many have done away with the obsolete pen and dog-eared registers for gathering visitor information.

These smart systems do away with writing in a visitor book, calling employees to inform about visitor’s arrival and ensures that unattended visitors are not left in crowded receptions.

**Visitor Management Systems in Apartment Complexes**

In Apartment Complexes, the age-old practice of using manual registers for capturing visitors’ information still carries on.

Whereas in commercial premises, what security protects is physical assets like data, servers, etc. security in complexes is supposed to safe guard lives of family – the elderly, the children. Therefore, security in apartment complexes is much more important.

Thankfully, today apartment complex security is being taken seriously. Many forward-thinking apartment and villa communities are looking at transforming their security with the use of technology.

**Chapter 2**

**Literature review**

To understand the significance of Visitor Management in UNWHS, extensive literature was reviewed tostudy the use of Visitor Management plansasintroduced in UNWHS globally. The criteria of categorising the visitor management plan as successes and not a success story weredependent on the preservation and conservation status at the sites as well as satisfied visitor flow. Literature review put forward the good practices followed to save these sites as well as the vital role played by numerous proactive and reactive techniquesof Visitor Management and conservation and preservation practices that were implemented in collaboration with locals, stakeholders and government agencies.

Visitor expectation from a site on UNWHS is not always met since there have been many accidents due tolack of crowdmanagement as well as incidents like a massive copper and bronze chandelier fell off its hook, the heavy metalpinnacle of one of the four minarets crashed, large chunks of red sandstone fell off rusted clamps and so on (asianews.it, 2018; indiatoday.in, 2018; bbc.com 2018)

Research shows that for every 30 new tourists to a destination, one new job is created (WTO-ILO, 2014).Therefore, closing down tourismand visitationsfor the preservation and conservation of the Taj Mahal is not possible. A viable visitor management strategy for the monument is the need of the time. Also, most importantly, the site is the chief revenue earner among other historic and UNWHS–sites in India. According to Minister –State for tourism and culture, the monument earned over Rs. 200 crore (Jaiswal, 2019)through ticket sales for three years (years2016-19).

**2.1 Technology Used**

**PHP**

PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.

The **PHP Hypertext Preprocessor (PHP)** is a programming language that allows web developers to create dynamic content that interacts with databases. PHP is basically used for developing web based software applications. This tutorial helps you to build your base with PHP.

**PHP** is a MUST for students and working professionals to become a great Software Engineer specially when they are working in Web Development Domain. I will list down some of the key advantages of learning PHP:

* PHP is a recursive acronym for "PHP: Hypertext Preprocessor".
* PHP is a server side scripting language that is embedded in HTML. It is used to manage dynamic content databases, session tracking, even build entire e-commerce sites.
* It is integrated with a number of popular databases, including MySQL, PostgreSQL, Oracle, Sybase, Informix, and Microsoft SQL Server.
* PHP is pleasingly zippy in its execution, especially when compiled as an Apache module on the Unix side. The MySQL server, once started, executes even very complex queries with huge result sets in record-setting time.

**BENEFITS OF PHP**

As mentioned before, PHP is one of the most widely used language over the web. I'm going to list few of them here:

* PHP performs system functions, i.e. from files on a system it can create, open, read, write, and close them.
* PHP can handle forms, i.e. gather data from files, save data to a file, through email you can send data, return data to the user.
* You add, delete, modify elements within your database through PHP.
* Access cookies variables and set cookies.
* Using PHP, you can restrict users to access some pages of your website.
* It can encrypt data.

**SoftWare and Applications Used**

APPLICATION : Eclipse

OPERATING SYSTEM : WINDOWS 10

FRONT END : HTML, CSS, PHP

BACK END : MySQL,Firebase

**Back-end :**

* **Firebase: -**It is Realtime database developed by firebase and then acquired by Google in 2014 used for developing the high quality applications act as a storage of information of various data.
* **MYSQL:** MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Wideness’s daughter, and "SQL", the abbreviation for Structured Query Language

**Some Other Requirements**

Performance Requirements:

To achieve good performance the following requirements must be satisfied

* Scalability: The ease with which a system or component can be modified to fit the problem area.
* Portability: The ease with which a system or component can be transferred from one hardware or software environment to another.
* Security: It is the ideal state where all information can be communicated across the internet / company secure from unauthorized persons being able to read it and/or manipulate it..
* Maintainability: The ease with which a software system or component can be modified to correct faults, improve performance, or other attributes, or adapt to a changed environment.
* Reliability: The ability of a system or component to perform its required functions under stated conditions for a specified period of time.
* Reusability: The degree to which a software module or other work product can be used in more than one computing program or software system.

**Safety Requirements:**

In case scenarios where data integrity can be compromised, measures should be taken to ensure that all changes are made before system is shutdown. The user must have a registered account to use all facility of the web application.

**2.2 Language Used**

This project has been developed XML and Java.

**HTML**

The HyperText Markup Language or HTML is the standard markup language for documents designed to be displayed in a web browser. It can be assisted by technologies such as Cascading Style Sheets and scripting languages such as JavaScript.

* HTML stands for Hyper Text Markup Language
* HTML is the standard markup language for creating Web pages
* HTML describes the structure of a Web page
* HTML consists of a series of elements
* HTML elements tell the browser how to display the content
* HTML elements label pieces of content such as "this is a heading", "this is a paragraph", "this is a link", etc.

## A Simple HTML Document

### Example

<!DOCTYPE html>  
<html>  
<head>  
<title>Page Title</title>  
</head>  
<body>  
  
<h1>My First Heading</h1>  
<p>My first paragraph.</p>  
  
</body>  
</html>

* **CSS:**Cascading Style Sheets is a style sheet language used for describing the presentation of a document written in a markup language such as HTML. CSS is a cornerstone technology of the World Wide Web, alongside HTML and JavaScript.
* Inline CSS.

An inline CSS is **used to apply a unique style to a single HTML element**. An inline CSS uses the style attribute of an HTML element. We can apply CSS in a single element by inline CSS technique.The inline CSS is also a method to insert style sheets in HTML document. This method mitigates some advantages of style sheets so it is advised to use this method sparingly.

If you want to use inline CSS, you should use the style attribute to the relevant tag

Disadvantages of Inline CSS

* You cannot use quotations within inline CSS. If you use quotations the browser will interpret this as an end of your style value.
* These styles cannot be reused anywhere else.
* These styles are tough to be edited because they are not stored at a single place.
* It is not possible to style pseudo-codes and pseudo-classes with inline CSS.
* Inline CSS does not provide browser cache advantages.

.

* Internal or Embedded CSS.

Internal CSS is a form of [CSS](https://www.simplilearn.com/11-css-secrets-rar400-article) using which you can add CSS to [HTML](https://www.simplilearn.com/tutorials/html-tutorial/what-is-html) documents. It helps to design the layout of a single HTML web page and change the styles of a web page within HTML code.

Internal CSS is one of the most widely used CSS forms for changing, styling, and modifying the unique styles of a single web page. You can use the internal CSS by integrating the <style> element in the <head> section of a HTML web page.

Internal CSS can be applied to the whole web page but not on multiple web pages and you can style several web pages by using the same code on every page.

## How to Use Internal CSS?

Internal CSS is a way of adding the CSS codes in the <style> element of <head> section of the document.

All the changes done by the internal CSS can be applied only to a single web page. You can use internal CSS for one-page websites or projects where you do not want to use external CSS.

Internal CSS is only used for one-page HTML websites. You can add internal CSS to multiple HTML web pages; however, it is not a good practice in website development. Internal CSS is mostly used when creating templates for your web page.

Internal CSS is only beneficial for single-page HTML websites. If you use internal CSS in your HTML web page, you cannot reference this stylesheet to the <link> elements. It can make the modification and changes in a website more complex and challenging.

Internal CSS is not favorable for large projects where you are required to have more than one web page, therefore, the most common way to add CSS, is to keep the styles in external [CSS files.](https://www.simplilearn.com/tutorials/html-tutorial/link-css-files-to-html)

* External CSS.

The external style sheet is generally used when you want to make changes on multiple pages. It is ideal for this condition because it facilitates you to change the look of the entire web site by changing just one file.

It uses the <link> tag on every pages and the <link> tag should be put inside the head section.

The external style sheet may be written in any text editor but must be saved with a .css extension. This file should not contain HTML elements.

By applying consistent formatting to website pages, external style sheets help in bringing a uniform, global look and feel to a website. The external style sheet can be linked from HTML pages. When using an external style sheet, styles need to be set up only once for each element. The external style sheet contains only CSS syntax and also carries a “text/CSS” MIME type. One of the unique features associated with external style sheets is that they can be created in any text editor, but need to be saved with a .css extension. The file should never have any elements of HTML. There are two main techniques of calling an external style sheet into an HTML document. One method is by using the <link> tag within the HTML document head. Another method is with the help of a combination of external CSS functions along with embedded CSS.

There are benefits associated with external style sheets. The external style sheet can be applied to an unlimited number of web pages. An external style sheet can be applied instantly to apply the look to each and every web page. They can also help in bringing a uniform format to every page they are attached to.

* **PHP:**PHP is a general-purpose scripting language geared toward web development. It was originally created by Danish-Canadian programmer Rasmus Lerdorf in 1994. The PHP reference implementation is now produced by The PHP Group.
* **MySQL**

MySQL is an open-source relational database management system. Its name is a combination of "My", the name of co-founder Michael Widenius's daughter, and "SQL", the abbreviation for Structured Query Language.

MySQL is a database management system.

A database is a structured collection of data. It may be anything from a simple shopping list to a picture gallery or the vast amounts of information in a corporate network. To add, access, and process data stored in a computer database, you need a database management system such as MySQL Server. Since computers are very good at handling large amounts of data, database management systems play a central role in computing, as standalone utilities, or as parts of other applications.

**MySQL databases are relational.**

A relational database stores data in separate tables rather than putting all the data in one big storeroom. The database structures are organized into physical files optimized for speed. The logical model, with objects such as databases, tables, views, rows, and columns, offers a flexible programming environment. You set up rules governing the relationships between different data fields, such as one-to-one, one-to-many, unique, required or optional, and “pointers” between different tables. The database enforces these rules, so that with a well-designed database, your application never sees inconsistent, duplicate, orphan, out-of-date, or missing data.

The SQL part of “MySQL” stands for “Structured Query Language”. SQL is the most common standardized language used to access databases. Depending on your programming environment, you might enter SQL directly (for example, to generate reports), embed SQL statements into code written in another language, or use a language-specific API that hides the SQL syntax.

SQL is defined by the ANSI/ISO SQL Standard. The SQL standard has been evolving since 1986 and several versions exist. In this manual, “SQL-92” refers to the standard released in 1992, “SQL:1999” refers to the standard released in 1999, and “SQL:2003” refers to the current version of the standard. We use the phrase “the SQL standard” to mean the current version of the SQL Standard at any time.

**MySQL software is Open Source*.***

Open Source means that it is possible for anyone to use and modify the software. Anybody can download the MySQL software from the Internet and use it without paying anything. If you wish, you may study the source code and change it to suit your needs. The MySQL software uses the GPL (GNU General Public License), <http://www.fsf.org/licenses/>, to define what you may and may not do with the software in different situations. If you feel uncomfortable with the GPL or need to embed MySQL code into a commercial application, you can buy a commercially licensed version from us. See the MySQL Licensing Overview for more information (<http://www.mysql.com/company/legal/licensing/>).

**The MySQL Database Server is very fast, reliable, scalable, and easy to use.**

If that is what you are looking for, you should give it a try. MySQL Server can run comfortably on a desktop or laptop, alongside your other applications, web servers, and so on, requiring little or no attention. If you dedicate an entire machine to MySQL, you can adjust the settings to take advantage of all the memory, CPU power, and I/O capacity available. MySQL can also scale up to clusters of machines, networked together.

MySQL Server was originally developed to handle large databases much faster than existing solutions and has been successfully used in highly demanding production environments for several years. Although under constant development, MySQL Server today offers a rich and useful set of functions. Its connectivity, speed, and security make MySQL Server highly suited for accessing databases on the Internet.

**MySQL Server works in client/server or embedded systems.**

The MySQL Database Software is a client/server system that consists of a multithreaded SQL server that supports different back ends, several different client programs and libraries, administrative tools, and a wide range of application programming interfaces (APIs).

We also provide MySQL Server as an embedded multithreaded library that you can link into your application to get a smaller, faster, easier-to-manage standalone product.

**A large amount of contributed MySQL software is available.**

MySQL Server has a practical set of features developed in close cooperation with our users. It is very likely that your favorite application or language supports the MySQL Database Server.

The official way to pronounce “MySQL” is “My Ess Que Ell” (not “my sequel”), but we do not mind if you pronounce it as “my sequel” or in some other localized way.

### The Main Features of MySQL

This section describes some of the important characteristics of the MySQL Database Software. In most respects, the roadmap applies to all versions of MySQL. For information about features as they are introduced into MySQL on a series-specific basis, see the “In a Nutshell” section of the appropriate Manual:

#### Internals and Portability

* Written in C and C++.
* Tested with a broad range of different compilers.
* Works on many different platforms. See <https://www.mysql.com/support/supportedplatforms/database.html>.
* For portability, configured using **CMake**.
* Tested with Purify (a commercial memory leakage detector) as well as with Valgrind, a GPL tool (<http://developer.kde.org/~sewardj/>).
* Uses multi-layered server design with independent modules.
* Designed to be fully multithreaded using kernel threads, to easily use multiple CPUs if they are available.
* Provides transactional and nontransactional storage engines.
* Uses very fast B-tree disk tables (MyISAM) with index compression.
* Designed to make it relatively easy to add other storage engines. This is useful if you want to provide an SQL interface for an in-house database.
* Uses a very fast thread-based memory allocation system.
* Executes very fast joins using an optimized nested-loop join.
* Implements in-memory hash tables, which are used as temporary tables.
* Implements SQL functions using a highly optimized class library that should be as fast as possible. Usually there is no memory allocation at all after query initialization.
* Provides the server as a separate program for use in a client/server networked environment.

#### Data Types

* Many data types: signed/unsigned integers 1, 2, 3, 4, and 8 bytes long, [FLOAT](https://dev.mysql.com/doc/refman/8.0/en/floating-point-types.html), [DOUBLE](https://dev.mysql.com/doc/refman/8.0/en/floating-point-types.html), [CHAR](https://dev.mysql.com/doc/refman/8.0/en/char.html), [VARCHAR](https://dev.mysql.com/doc/refman/8.0/en/char.html), [BINARY](https://dev.mysql.com/doc/refman/8.0/en/binary-varbinary.html), [VARBINARY](https://dev.mysql.com/doc/refman/8.0/en/binary-varbinary.html), [TEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html), [BLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html), [DATE](https://dev.mysql.com/doc/refman/8.0/en/datetime.html), [TIME](https://dev.mysql.com/doc/refman/8.0/en/time.html), [DATETIME](https://dev.mysql.com/doc/refman/8.0/en/datetime.html), [TIMESTAMP](https://dev.mysql.com/doc/refman/8.0/en/datetime.html), [YEAR](https://dev.mysql.com/doc/refman/8.0/en/year.html), [SET](https://dev.mysql.com/doc/refman/8.0/en/set.html), [ENUM](https://dev.mysql.com/doc/refman/8.0/en/enum.html), and OpenGIS spatial types.
* Fixed-length and variable-length string types.

#### Statements and Functions

* Full operator and function support in the [SELECT](https://dev.mysql.com/doc/refman/8.0/en/select.html) list and WHERE clause of queries. For example:
* mysql> SELECT CONCAT(first\_name, ' ', last\_name)
* -> FROM citizen

-> WHERE income/dependents > 10000 AND age > 30;

* Full support for SQL GROUP BY and ORDER BY clauses. Support for group functions ([COUNT()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_count), [AVG()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_avg), [STD()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_std), [SUM()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_sum), [MAX()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_max), [MIN()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_min), and [GROUP\_CONCAT()](https://dev.mysql.com/doc/refman/8.0/en/aggregate-functions.html#function_group-concat)).
* Support for LEFT OUTER JOIN and RIGHT OUTER JOIN with both standard SQL and ODBC syntax.
* Support for aliases on tables and columns as required by standard SQL.
* Support for [DELETE](https://dev.mysql.com/doc/refman/8.0/en/delete.html), [INSERT](https://dev.mysql.com/doc/refman/8.0/en/insert.html), [REPLACE](https://dev.mysql.com/doc/refman/8.0/en/replace.html), and [UPDATE](https://dev.mysql.com/doc/refman/8.0/en/update.html) to return the number of rows that were changed (affected), or to return the number of rows matched instead by setting a flag when connecting to the server.
* Support for MySQL-specific [SHOW](https://dev.mysql.com/doc/refman/8.0/en/show.html) statements that retrieve information about databases, storage engines, tables, and indexes. Support for the INFORMATION\_SCHEMA database, implemented according to standard SQL.
* An [EXPLAIN](https://dev.mysql.com/doc/refman/8.0/en/explain.html) statement to show how the optimizer resolves a query.
* Independence of function names from table or column names. For example, ABS is a valid column name. The only restriction is that for a function call, no spaces are permitted between the function name and the “(” that follows it. See [Section 9.3, “Keywords and Reserved Words”](https://dev.mysql.com/doc/refman/8.0/en/keywords.html).
* You can refer to tables from different databases in the same statement.

#### Security

* A privilege and password system that is very flexible and secure, and that enables host-based verification.
* Password security by encryption of all password traffic when you connect to a server.

#### Scalability and Limits

* Support for large databases. We use MySQL Server with databases that contain 50 million records. We also know of users who use MySQL Server with 200,000 tables and about 5,000,000,000 rows.
* Support for up to 64 indexes per table. Each index may consist of 1 to 16 columns or parts of columns. The maximum index width for [InnoDB](https://dev.mysql.com/doc/refman/8.0/en/innodb-storage-engine.html" \o "Chapter 15 The InnoDB Storage Engine) tables is either 767 bytes or 3072 bytes. See [Section 15.22, “InnoDB Limits”](https://dev.mysql.com/doc/refman/8.0/en/innodb-limits.html). The maximum index width for [MyISAM](https://dev.mysql.com/doc/refman/8.0/en/myisam-storage-engine.html" \o "16.2 The MyISAM Storage Engine) tables is 1000 bytes. See [Section 16.2, “The MyISAM Storage Engine”](https://dev.mysql.com/doc/refman/8.0/en/myisam-storage-engine.html). An index may use a prefix of a column for [CHAR](https://dev.mysql.com/doc/refman/8.0/en/char.html), [VARCHAR](https://dev.mysql.com/doc/refman/8.0/en/char.html), [BLOB](https://dev.mysql.com/doc/refman/8.0/en/blob.html), or [TEXT](https://dev.mysql.com/doc/refman/8.0/en/blob.html) column types.

#### Connectivity

* Clients can connect to MySQL Server using several protocols:
* Clients can connect using TCP/IP sockets on any platform.
* On Windows systems, clients can connect using named pipes if the server is started with the [named\_pipe](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html" \l "sysvar_named_pipe) system variable enabled. Windows servers also support shared-memory connections if started with the [shared\_memory](https://dev.mysql.com/doc/refman/8.0/en/server-system-variables.html" \l "sysvar_shared_memory) system variable enabled. Clients can connect through shared memory by using the [--protocol=memory](https://dev.mysql.com/doc/refman/8.0/en/connection-options.html#option_general_protocol) option.
* On Unix systems, clients can connect using Unix domain socket files.
* MySQL client programs can be written in many languages. A client library written in C is available for clients written in C or C++, or for any language that provides C bindings.
* APIs for C, C++, Eiffel, Java, Perl, PHP, Python, Ruby, and Tcl are available, enabling MySQL clients to be written in many languages.
* The Connector/ODBC (MyODBC) interface provides MySQL support for client programs that use ODBC (Open Database Connectivity) connections. For example, you can use MS Access to connect to your MySQL server. Clients can be run on Windows or Unix. Connector/ODBC source is available. All ODBC 2.5 functions are supported, as are many others. See [MySQL Connector/ODBC Developer Guide](https://dev.mysql.com/doc/connector-odbc/en/).
* The Connector/J interface provides MySQL support for Java client programs that use JDBC connections. Clients can be run on Windows or Unix. Connector/J source is available. See [MySQL Connector/J 5.1 Developer Guide](https://dev.mysql.com/doc/connector-j/5.1/en/).
* MySQL Connector/NET enables developers to easily create .NET applications that require secure, high-performance data connectivity with MySQL. It implements the required ADO.NET interfaces and integrates into ADO.NET aware tools. Developers can build applications using their choice of .NET languages. MySQL Connector/NET is a fully managed ADO.NET driver written in 100% pure C#. See [MySQL Connector/NET Developer Guide](https://dev.mysql.com/doc/connector-net/en/).

#### Localization

* The server can provide error messages to clients in many languages. See [Section 10.12, “Setting the Error Message Language”](https://dev.mysql.com/doc/refman/8.0/en/error-message-language.html).
* Full support for several different character sets, including latin1 (cp1252), german, big5, ujis, several Unicode character sets, and more. For example, the Scandinavian characters “å”, “ä” and “ö” are permitted in table and column names.
* All data is saved in the chosen character set.
* Sorting and comparisons are done according to the default character set and collation. is possible to change this when the MySQL server is started (see [Section 10.3.2, “Server Character Set and Collation”](https://dev.mysql.com/doc/refman/8.0/en/charset-server.html)). To see an example of very advanced sorting, look at the Czech sorting code. MySQL Server supports many different character sets that can be specified at compile time and runtime.
* The server time zone can be changed dynamically, and individual clients can specify their own time zone. See [Section 5.1.15, “MySQL Server Time Zone Support”](https://dev.mysql.com/doc/refman/8.0/en/time-zone-support.html).

#### Clients and Tools

* MySQL includes several client and utility programs. These include both command-line programs such as **[mysqldump](https://dev.mysql.com/doc/refman/8.0/en/mysqldump.html" \o "4.5.4 mysqldump — A Database Backup Program)** and **[mysqladmin](https://dev.mysql.com/doc/refman/8.0/en/mysqladmin.html" \o "4.5.2 mysqladmin — A MySQL Server Administration Program)**, and graphical programs such as [MySQL Workbench](https://dev.mysql.com/doc/refman/8.0/en/workbench.html).
* MySQL Server has built-in support for SQL statements to check, optimize, and repair tables. These statements are available from the command line through the **[mysqlcheck](https://dev.mysql.com/doc/refman/8.0/en/mysqlcheck.html" \o "4.5.3 mysqlcheck — A Table Maintenance Program)** client. MySQL also includes **[myisamchk](https://dev.mysql.com/doc/refman/8.0/en/myisamchk.html" \o "4.6.4 myisamchk — MyISAM Table-Maintenance Utility)**, a very fast command-line utility for performing these operations on MyISAM tables. See [Chapter 4, MySQL Programs](https://dev.mysql.com/doc/refman/8.0/en/programs.html).
* MySQL programs can be invoked with the --help or -? option to obtain online assistance.

**Chapter 3**

**Software Requirement Specification**

**3.1 Hardware Configuration :**

**Client Side:**

|  |  |
| --- | --- |
| **RAM RAMfgdfRA R RAM** | 512 MB |
|  |  |
| **Hard disk** | 10 GB |
|  |  |
| **Processor** | 1.0 GHz |
|  |  |
|  |  |

**Server side:**

|  |  |
| --- | --- |
| **RAM** | **1 GB** |
| **Hard disk** | **20 GB** |
| **Processor** | **2.0 GHz** |

**3.2 Software Requirement:**

**Client Side:**

|  |  |
| --- | --- |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

**Server Side:**

|  |  |
| --- | --- |
| **Web Server** | APACHE |
| **Server side Language** | PHP5.6 or above version |
| **Database Server** | MYSQL |
| **Web Browser** | Google Chrome or any compatible browser |
| **Operating System** | Windows or any equivalent OS |

**APACHE**

The Apache HTTP Server Project is an effort to develop and maintain an open-source HTTP server for modern operating systems including UNIX and Windows. The goal of this project is to provide a secure, efficient and extensible server that provides HTTP services in sync with the current HTTP standards.

The Apache HTTP Server ("httpd") was launched in 1995 and it has been the most popular web server on the Internet since April 1996. It has celebrated its 20th birthday as a project in February 2015.

Apache is the most widely used webserver software and runs on 67% of all websites in the world. Developed and maintained by [Apache Software Foundation](https://www.apache.org/), Apache is [open source](https://www.wpbeginner.com/glossary/open-source/) software and available for free.

It’s fast, reliable, and secure. And Apache can be highly customized to meet the needs of many different environments by using extensions and modules.

Most WordPress hosting providers use Apache as their webserver software. However, WordPress can run on other webserver software as well.

#### What Is a Webserver?

A webserver is software run by your [website hosting provider](https://www.wpbeginner.com/showcase/best-free-website-hosting-compared/) so that visitors can view the web pages on your site. Many WordPress hosting providers use Apache.

The software performs a similar role to a restaurant host. When you arrive at a restaurant, the host greets you, checks your booking information, and takes you to your table. In a similar way, the webserver checks for the web page you have requested.

A webserver does more. It also acts like a restaurant server when it fetches the page and serves it for your viewing pleasure.

And it performs the role of the restaurant’s maitre d’ by handling communications with the website, handling your requests, and making sure that other modules are ready to serve you.

Finally, a webserver is also like the busboy clearing tables. It cleans the memory, cache and modules and clears them for new website visitors.

In summary, a webserver like Apache performs many tasks:

* It receives your request to access a web page.
* It runs a few security checks on your HTTP request and takes you to the web page.
* It may ask the server to run extra modules while generating the document to serve you.
* It then serves you the document you requested.
* If you wish to run WordPress [locally on your computer](https://www.wpbeginner.com/glossary/localhost/), then you will need to run the same software as your WordPress hosting provider. This includes Apache (the webserver software), [MySQL](https://www.wpbeginner.com/glossary/mysql/) (the database), and [PHP](https://www.wpbeginner.com/glossary/php/) (the programming language).
* Luckily, software stacks like [WAMP](https://www.wpbeginner.com/wp-tutorials/how-to-install-wordpress-on-your-windows-computer-using-wamp/) (for Windows), [MAMP](https://www.wpbeginner.com/wp-tutorials/how-to-install-wordpress-locally-on-mac-using-mamp/) (for Mac), LAMP (for Linux), and [XAMPP](https://www.wpbeginner.com/wp-tutorials/how-to-create-a-local-wordpress-site-using-xampp/) (cross-platform) make installing this software easy.
* [Apache](https://www.sumologic.com/application/apache/) HTTP Server is a free and open-source web server that delivers web content through the internet. It is commonly referred to as Apache and after development, it quickly became the most popular HTTP client on the web. It’s widely thought that Apache gets its name from its development history and process of improvement through applied patches and modules but that was corrected back in 2000. It was revealed that the name originated from the respect of the Native American tribe for its resiliency and durability.

**PHP**

* PHP stands for PHP: Hypertext Preprocessor.
* PHP is a server-side scripting language, like ASP.
* PHP scripts are executed on the server.
* PHP supports many databases (MYSQL, Informix, Oracle, Sybase, Solid, Generic ODBC, etc.).
* PHP is open source software.
* PHP is free to download and use.

**MYSQL**

* MYSQL is a database server
* MYSQL is ideal for both small and large applications
* MYSQL supports standard SQL
* MYSQL compiles on a number of platforms
* MYSQL is free to download and use

**3.3 Functional Requirements:**

**Internet Connectivity:**

As discussed that Application will work on Online mode so it need regular Internet Connectivity to signup and login.

**Facebook Account**

User can directly login to the facebook account to access this application they don’t need to signin in apps environment for which facebook account is mandatory.

**Email id and Mobile Number**

To access the application and to signin or login user must have email id and mobile number to fill the mandatory field in the form.

**3.4 Non-functional Requirements:**

Performance Requirements

* User friendly**:** The system should be user friendly so that it can easily be understand by the user without any difficulty.
* Ease of maintenance :- System should be easy to maintain and use.
* Less time consuming: The system should be less time consuming which could be achieved by good programming.
* Error free: The system should easily handle the user error in any case.
* Static: Application runs on stand alone machine i.e. Android mobile phone of API level 16 and onward. Support only single user.

**3.5 Software System Attributes:**

1. **Security**: The system should be secure from the unauthorized access and should be password protected so that no other user can access it.

If the user is new then he needs to Signup with required details and a can also login with the facebook.

1. **Portability**:- The system should be machine independent.
2. **Maintainability**: The system will be designed in a maintainable order. The system can be easily modified and renewed according to the need of the organization.

**3.6 Features**

* Security of data.
* Ensures data accuracy.
* Minimize manpower.
* Minimum time consumption.
* Greater efficiency.
* Fast
* Better services.
* User friendliness and interactive.
* Minimum time required.
* Easy to update
* User friendly
* Free for the user
* knowing about entry time and exit time

Apartment Visitor Management system is a web-based technology that will revolutionize the way your society manages visitors. Visitor Management system is more important to security guards or security society. This web application provides a way to effectively control record & track society visitor traffic.

In AVMS we use PHP and MySQL database. This is the project which keeps records of visitors who visited in the Society. **AVMS has one module i.e. admin**. The information displayed should be easy to customize. **Our suggestion is that the user could choose different ‘skins’ and that these skins would represent different types of stores and events**. Our vision is to encourage the visitors to visit the apartments without any hurdle in their journey. The user will always have the brief information about the apartment along with flat no. and floor they don’t need to pay extra amount to the security man of that apartment security will make their safe and hassle free entry.

**3.7 Preliminary investigation:**

Fact Finding:

After obtaining the background knowledge, we began to collect data on the existing system.

The tools that are used in information gathering are as follows:

* On-site observation.
* Questionnaire.
* Review of the peoples.

The model we have used is Waterfall Model. In this model, first of all the existing system is observed, then customer requirements are taken in consideration then planning, modelling, construction and finally deployment.

3.8Approach used: Agile Approach



Fig. 3.1

Agile is **an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches**. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments.

3.9 Preliminary Description:

The first step in the system development life cycle is the preliminary investigation to determine the feasibility of the system. The purpose of preliminary investigation is to evaluate project requests. It is not a design study nor does it include the collection of details to describe the system in all respect. Rather, it is the collecting of information that helps committee members to evaluate the merits of project request and make an informed judgement about the feasibility of the proposed project.

**Analyst working on the preliminary investigation should accomplish the following objectives:**

* Clarify and understand the project request.
* Determine the size of the project.
* Access costs and benefits of alternative approaches.
* Determine the technical and operational feasibility of alternative approaches.
* Report the findings to management with recommendations outlining the acceptance and rejection of the proposal.

**Chapter 4**

**Feasibility study**

After studying and analyzing all the existing and requires functionalities of the system, the next task is to do the feasibility study for the project. Feasibility study includes consideration of all the possible ways to provide a solution to a given problem. The proposed solution should satisfy all the user requirements and should be flexible enough so that future changes can be easily done based on the future upcoming requirements.

**4.1 Economical Feasibility:**

It will be freely available on the Google play store without having any cost.

* 1. **Technical feasibility:**

This included the study of function, performance and constraints that may affect the ability to achieve an acceptable system. For this feasibility study, we studied complete functionalities to be provided in the system, as described in the System Requirement Specification (SRS), and checked if everything was possible using different type of front end and back end platform.

* 1. **Operational Feasibility:**

No doubt the technically growing Bihar needs more enhancement in technology, this apps is very user friendly and all inputs to be taken all self-explanatory even to a layman.

Feasibility of the system in an important aspect, which is to be considered. The system needs to satisfy the,law of economic, which states that the maximum output should be yielded in minimum available resources. A feasibility analysis evaluates the project’s potential for success; therefore, perceived objectivity is an essential factor in the credibility of the study for potential investors and lending institutions. There are five types of feasibility study—separate areas that a feasibility study examines, described below.

Technical Feasibility

This assessment focuses on the technical resources available to the organization. It helps organizations determine whether the technical resources meet capacity and whether the technical team is capable of converting the ideas into working systems. Technical feasibility also involves the evaluation of the hardware, software, and other technical requirements of the proposed system. As an exaggerated example, an organization wouldn’t want to try to put Star Trek’s transporters in their building—currently, this project is not technically feasible.

Economic Feasibility

This assessment typically involves a cost/ benefits analysis of the project, helping organizations determine the viability, cost, and benefits associated with a project before financial resources are allocated. It also serves as an independent project assessment and enhances project credibility—helping decision-makers determine the positive economic benefits to the organization that the proposed project will provide.

Legal Feasibility

This assessment investigates whether any aspect of the proposed project conflicts with legal requirements like zoning laws, data protection acts or social media laws. Let’s say an organization wants to construct a new office building in a specific location. A feasibility study might reveal the organization’s ideal location isn’t zoned for that type of business. That organization has just saved considerable time and effort by learning that their project was not feasible right from the beginning.

Operational Feasibility

This assessment involves undertaking a study to analyze and determine whether—and how well—the organization’s needs can be met by completing the project. Operational feasibility studies also examine how a project plan satisfies the requirements identified in the requirements analysis phase of system development.

Scheduling Feasibility

This assessment is the most important for project success; after all, a project will fail if not completed on time. In scheduling feasibility, an organization estimates how much time the project will take to complete.When these areas have all been examined, the feasibility analysis helps identify any constraints the proposed project may face, including:

Internal Project Constraints: Technical, Technology, Budget,Resource, etc.

Internal Corporate Constraints: Financial, Marketing, Export, etc.

External Constraints: Logistics, Environment, Laws, and Regulations, etc

**MAIN ASPECTS**

There are three aspects of feasibility to be considered namely.

1. Technical
2. Operational
3. Economical

TECHNICAL:

In the technical aspects one may consider the hardware equipment for the installation of the software. The system being centralized will required very little hardware appliances. Hence this helps the system to work smoothly with limited amount of working capitals.

OPERATIONAL:

In the operational aspects may think of the benefits of the workload that many a personal may have to share. This is eased out and the required output may be retrieved in a very short time. Thus there is accuracy in the work on time is also saved there will be very little work that needs to be performed.

ECONOMICAL:

Economical system is definitely feasible because the hardware requirement is less and the operational working for the system requires less number of recruits. This help introduction over-staffing and wastage funds.

We studied on the position to evaluate solution. Most important factors in this study were tending to overlook the confusion inherent in system Development the constraints and the assumed studies. It can be started that it the feasibility study is to serve as a decision document it must answer three key questions.

1. Is there a new and better way to do the job that will benefit the user?
2. What are the costs and savings of the alternatives?
3. What is recommended?

On these questions it can be explained that feasibility study of the system includes following different angles.

**Technical feasibility:**

This centers on the existing computer system (hardware, software etc.) and to what extent it can support the proposed additional equipment .in this stage of study, we have collected information about technical tools available by which I could decide my system design as the technical requirements.

**Operational Feasibility:**

In this stage of study we have checked the staff availability. I concentrate on knowledge of end users that are going to use the system. This is also called as behavioral feasibility in which I have studied on following aspects; people are inherently resistant to change, and computers have been known to facilitate change .An estimate has been made to how strong a reaction the user staff is having toward the development of a computerized system. It is common knowledge that computer installations have something to do with turnover. I had explained that there is need to educate and train the staff on new ways of conducting business.

**Economical feasibility**:

Economical analysis is the most frequently used method for evaluating the effectiveness of candidate system. More commonly known as cost\benefit analysis, the procedure is to determine the benefits and savings that benefits outweigh costs. The decision was to design and implement system because it is for having chanced to be approved. This is an on going effort that improves the accuracy at each phase of the system life cycle.In developing cost estimates for a system I need to consider several cost elements. Among these is hardware personal facility. Operating and supply costs.

**BENEFITS**

Benefits of conducting a feasibility study:

* Improves project teams’ focus
* Identifies new opportunities
* Provides valuable information for a “go/no-go” decision
* Narrows the business alternatives
* Identifies a valid reason to undertake the project
* Enhances the success rate by evaluating multiple parameters
* Aids decision-making on the project
* Identifies reasons not to proceed

**SOFTWARE REQUIREMENT SPECIFICATION**

Any system can be designed after specifies the requirement of the user about that system. For this first of all gathered information from user by the preliminary investigation which is starting investigation about user requirement..

The data that the analysts collect during preliminary investigation are gathered through the various preliminary methods.

Documents Reviewing Organization

The analysts conducting the investigation first learn the organization involved in, or affected by the project. Analysts can get some details by examining organization charts and studying written operating procedures.

Collected data is usually of the current operating procedure:

* The information relating to clients, projects and students and the relationship between them was held manually.
* Managing of follow-ups was through manual forms.
* Complaints require another tedious work to maintain and solve.
* Payments details had to be maintained differently.

Gathering Information By Asking Questions

Interviewing is the most commonly used techniques in analysis. It is always necessary first to approach someone and ask them what their problems are, and later to discuss with them the result of your analysis.

Questionnaires

Questionnaires provide an alternative to interviews for finding out information about a system. Questionnaires are made up of questions about information sought by analyst. The questionnaire is then sent to the user, and the analyst analyzes the replies.

Electronic Data Gathering

Electronic communication systems are increasingly being used to gather information. Thus it is possible to use electronic mail to broadcast a question to a number of users in an organization to obtain their viewpoint on a particular issue.

In my project, with the help of KONGTECHMEDIA TECHNOLOGIES, I have send questionnaire through electronic mail to twenty employees of the company and retrieved the information regarding the problem faced by existing system.

Interviews

Interview allows the analysts to learn more about the nature of the project request and reason of submitting it. Interviews should provide details that further explain the project and show whether assistance is merited economically, operationally or technically.One of the most important points about interviewing is that what question you need to ask.It is often convenient to make a distinction between three kinds of question that is

* 1. Open questions
  2. Closed question
  3. Probes

Open questions are general question that establish a persons view point on a particular subject.

Closed questions are specific and usually require a specific answer.

**Chapter 5**

**Planning and Scheduling and Flow**

5.1 **Gantt chart**

A Gantt chart can be developed for the entire project or a separate chart can be developed for each function. A tabular form is maintained where rows indicate the task with milestones and columns indicate duration(weeks/months).

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Days  Process | 1-5 | | 6-25 | 26-30 | | | 30-80 | | 80-85 | | | 85-90 | | |
| Requirement  Gathering |  |  |  |  | | |  | |  | | |  | | |
| Design |  |  |  |  | | |  | |  | | |  | | |
| Test Cases |  | |  |  |  |  | |  | | |  | | |  |
| Coding |  | |  |  |  | |  | |  | | |  | | |
| Testing |  | |  |  | | |  | |  |  | |  |  | |
| Build |  | |  |  | | |  | |  | | |  |  | |

5.2 PERT CHART

Program evaluation and review technique (pert) is a project scheduling method that is applied to software development.

Pert provide quantitative tool that allow the software planner to-Determine the critical path-the chain of tasks that determines the duration of the project;Establish “most likely” time estimates for individual tasks by applying statistical models; andCalculate “boundary times” that defines a time “window” for a particular task.

Pert chart(program evolution review technique) for project-

Fig 5.1

Preliminary investigation

Dummy

Testing

(73,91)

Coding

Design

Analysis

Data collection

**Project report**

**Chapter 6**

**Analysis and Design**

**6.1 Analysis:**

In present all visitor work done on the paper. The whole year visitor is stored in the registers. We can’t generate reports as per our requirements because its take more time to calculate the visitors report.

**Disadvantage of present system:**

* **Not user friendly:** The present system not user friendly because data is not stored in structure and proper format.
* **Manual Control:** All report calculation is done manually so there is a chance of error.
* **Lots of paper work:** Visitors maintain in the register so lots of paper require storing details.
* **Time consuming**

**6.2 Design Introduction:**

Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization.

Once the software requirements have been analyzed and specified the software design involves three technical activities - design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity, decisions ultimately affecting the success of the software implementation and its ease of maintenance are made. These decisions have the final bearing upon reliability and maintainability of the system. Design is the only way to accurately translate the customer’s requirements into finished software or a system.

Design is the place where quality is fostered in development. Software design is a process through which requirements are translated into a representation of software. Software design is conducted in two steps. Preliminary design is concerned with the transformation of requirements into data

The objective of software project planning is to provide a framework that enables the manager to make reasonable estimates of resources, costs and schedule. These estimates are made within a limited time frame at the beginning of a software project and should be updated regularly as the project progresses. In addition, estimates should attempts to define “best case” and “worst case” scenarios so that project outcomes can be bounded.

The first activity in software project planning is the determination of software scope. Function and performance allocated to software during system engineering should be assessed to establish a project scope that is ambiguous and understandable at Presidency and technical levels. Software scope describes function, performance, constraints, interfaces and reliability.

During early stages of project planning, a microscopic schedule is developed. This type of schedule identifies all major software engineering activities and the product functions to which they are applied. As the project gets under way, each entry on the macroscopic schedule is refined into detailed schedule. Here specific software tasks are identified and scheduled.

Scheduling has following principles:

* + - 1. Compartmentalization: the project must be compartmentalized into a number of manageable activities and tasks.
      2. Interdependency: the interdependencies of each compartmentalized activity or tasks must be determined.
      3. Time allocation: each task to be scheduled must be allocated some number of work units.
      4. Effort validation: every project has a defined number of staff members.
      5. Defined responsibilities: every task that is scheduled should be assigned to a specific team member.
      6. Defined outcomes: every task that is scheduled should have a defined outcome.

The elements are like components which can be associated in different ways to make a complete UML picture, which is known as diagram. Thus, it is very important to understand the different diagrams to implement the knowledge in real-life systems.

Any complex system is best understood by making some kind of diagrams or pictures. These diagrams have a better impact on our understanding. If we look around, we will realize that the diagrams are not a new concept but it is used widely in different forms in different industries.

We prepare UML diagrams to understand the system in a better and simple way. A single diagram is not enough to cover all the aspects of the system. UML defines various kinds of diagrams to cover most of the aspects of a system.

You can also create your own set of diagrams to meet your requirements. Diagrams are generally made in an incremental and iterative way.

There are two broad categories of diagrams and they are again divided into subcategories −

* Structural Diagrams
* Behavioral Diagrams

## Structural Diagrams

The structural diagrams represent the static aspect of the system. These static aspects represent those parts of a diagram, which forms the main structure and are therefore stable.

These static parts are represented by classes, interfaces, objects, components, and nodes. The four structural diagrams are −

* Class diagram
* Object diagram
* Component diagram
* Deployment diagram

### Class Diagram

Class diagrams are the most common diagrams used in UML. Class diagram consists of classes, interfaces, associations, and collaboration. Class diagrams basically represent the object-oriented view of a system, which is static in nature.

Active class is used in a class diagram to represent the concurrency of the system.

Class diagram represents the object orientation of a system. Hence, it is generally used for development purpose. This is the most widely used diagram at the time of system construction.

### Object Diagram

Object diagrams can be described as an instance of class diagram. Thus, these diagrams are more close to real-life scenarios where we implement a system.

Object diagrams are a set of objects and their relationship is just like class diagrams. They also represent the static view of the system.

The usage of object diagrams is similar to class diagrams but they are used to build prototype of a system from a practical perspective.

### Component Diagram

Component diagrams represent a set of components and their relationships. These components consist of classes, interfaces, or collaborations. Component diagrams represent the implementation view of a system.

During the design phase, software artifacts (classes, interfaces, etc.) of a system are arranged in different groups depending upon their relationship. Now, these groups are known as components.

Finally, it can be said component diagrams are used to visualize the implementation.

### Deployment Diagram

Deployment diagrams are a set of nodes and their relationships. These nodes are physical entities where the components are deployed.

Deployment diagrams are used for visualizing the deployment view of a system. This is generally used by the deployment team.

**Note** − If the above descriptions and usages are observed carefully then it is very clear that all the diagrams have some relationship with one another. Component diagrams are dependent upon the classes, interfaces, etc. which are part of class/object diagram. Again, the deployment diagram is dependent upon the components, which are used to make component diagrams.

## Behavioral Diagrams

Any system can have two aspects, static and dynamic. So, a model is considered as complete when both the aspects are fully covered.

Behavioral diagrams basically capture the dynamic aspect of a system. Dynamic aspect can be further described as the changing/moving parts of a system.

UML has the following five types of behavioral diagrams −

* Use case diagram
* Sequence diagram
* Collaboration diagram
* Statechart diagram
* Activity diagram

### Use Case Diagram

Use case diagrams are a set of use cases, actors, and their relationships. They represent the use case view of a system.

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external controllers. These controllers are known as **actors**.

### Sequence Diagram

A sequence diagram is an interaction diagram. From the name, it is clear that the diagram deals with some sequences, which are the sequence of messages flowing from one object to another.

Interaction among the components of a system is very important from implementation and execution perspective. Sequence diagram is used to visualize the sequence of calls in a system to perform a specific functionality.

### Collaboration Diagram

Collaboration diagram is another form of interaction diagram. It represents the structural organization of a system and the messages sent/received. Structural organization consists of objects and links.

The purpose of collaboration diagram is similar to sequence diagram. However, the specific purpose of collaboration diagram is to visualize the organization of objects and their interaction.

### Statechart Diagram

Any real-time system is expected to be reacted by some kind of internal/external events. These events are responsible for state change of the system.

Statechart diagram is used to represent the event driven state change of a system. It basically describes the state change of a class, interface, etc.

State chart diagram is used to visualize the reaction of a system by internal/external factors.

### Activity Diagram

Activity diagram describes the flow of control in a system. It consists of activities and links. The flow can be sequential, concurrent, or branched.

Activities are nothing but the functions of a system. Numbers of activity diagrams are prepared to capture the entire flow in a system.

Activity diagrams are used to visualize the flow of controls in a system. This is prepared to have an idea of how the system will work when executed.

**Note** − Dynamic nature of a system is very difficult to capture. UML has provided features to capture the dynamics of a system from different angles. Sequence diagrams and collaboration diagrams are isomorphic, hence they can be converted from one another without losing any information. This is also true for Statechart and activity diagram.

Purpose of Activity Diagrams

The basic purposes of activity diagrams is similar to other four diagrams. It captures the dynamic behavior of the system. Other four diagrams are used to show the message flow from one object to another but activity diagram is used to show message flow from one activity to another.

Activity is a particular operation of the system. Activity diagrams are not only used for visualizing the dynamic nature of a system, but they are also used to construct the executable system by using forward and reverse engineering techniques. The only missing thing in the activity diagram is the message part.

It does not show any message flow from one activity to another. Activity diagram is sometimes considered as the flowchart. Although the diagrams look like a flowchart, they are not. It shows different flows such as parallel, branched, concurrent, and single.

The purpose of an activity diagram can be described as −

* Draw the activity flow of a system.
* Describe the sequence from one activity to another.
* Describe the parallel, branched and concurrent flow of the system.

How to Draw an Activity Diagram?

Activity diagrams are mainly used as a flowchart that consists of activities performed by the system. Activity diagrams are not exactly flowcharts as they have some additional capabilities. These additional capabilities include branching, parallel flow, swimlane, etc.

Before drawing an activity diagram, we must have a clear understanding about the elements used in activity diagram. The main element of an activity diagram is the activity itself. An activity is a function performed by the system. After identifying the activities, we need to understand how they are associated with constraints and conditions.

Before drawing an activity diagram, we should identify the following elements −

* Activities
* Association
* Conditions
* Constraints

Where to Use Activity Diagrams?

The basic usage of activity diagram is similar to other four UML diagrams. The specific usage is to model the control flow from one activity to another. This control flow does not include messages.

Activity diagram is suitable for modeling the activity flow of the system. An application can have multiple systems. Activity diagram also captures these systems and describes the flow from one system to another. This specific usage is not available in other diagrams. These systems can be database, external queues, or any other system.

We will now look into the practical applications of the activity diagram. From the above discussion, it is clear that an activity diagram is drawn from a very high level. So it gives high level view of a system. This high level view is mainly for business users or any other person who is not a technical person.

This diagram is used to model the activities which are nothing but business requirements. The diagram has more impact on business understanding rather than on implementation details.

Activity diagram can be used for −

* Modeling work flow by using activities.
* Modeling business requirements.
* High level understanding of the system's functionalities.
* Investigating business requirements at a later stage.

**6.3 UML Diagrams:**

Actor:  
 A coherent set of roles that users of use cases play when interacting with the use `cases.

Fig.6.1

Use case:A description of sequence of actions, including variants, that a system performs that yields an observable result of value of an actor.

Fig.6.2

UML stands for Unified Modeling Language. UML is a language for specifying, visualizing and documenting the system. This is the step while developing any product after analysis. The goal from this is to produce a model of the entities involved in the project which later need to be built. The representation of the entities that are to be used in the product being developed need to be designed.

**6.4 USECASE DIAGRAMS:**

Use case diagrams model behavior within a system and helps the developers understand of what the user require. The stick man represents what’s called an actor.

Use case diagram can be useful for getting an overall view of the system and clarifying who can do and more importantly what they can’t do.

Use case diagram consists of use cases and actors and shows the interaction between the use case and actors.

* The purpose is to show the interactions between the use case and actor.
* To represent the system requirements from user’s perspective.
* An actor could be the end-user of the system or an external system.

**USECASE DIAGRAM:**A Use case is a description of set of sequence of actions. Graphically it is rendered as an ellipse with solid line including only its name. Use case diagram is a behavioral diagram that shows a set of use cases and actors and their relationship. It is an association between the use cases and actors. An actor represents a real-world object. Primary Actor – Sender, Secondary Actor Receiver.

**Use Case Diagrams:**

**Admin**

Fig.6.3

**6.5 Class Diagram:**

A description of set of objects that share the same attributes operations, relationships, and semantics

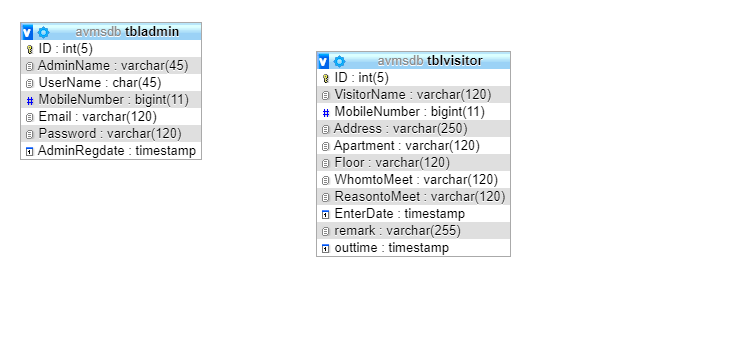


Fig.6.4

**Chapter 7**

**ER Diagram**

The Entity-Relationship (ER) model was originally proposed by Peter in 1976 [Chen76] as a way to unify the network and relational database views. Simply stated the ER model is a conceptual data model that views the real world as entities and relationships. A basic component of the model is the Entity-Relationship diagram which is used to visually represent data objects. Since Chen wrote his paper the model has been extended and today it is commonly used for database design for the database designer, the utility of the ER model is:

* It maps well to the relational model. The constructs used in the ER model can easily be transformed into relational tables.
* It is simple and easy to understand with a minimum of training. Therefore, the model can be used by the database designer to communicate the design to the end user.
* In addition, the model can be used as a design plan by the database developer to implement a data model in specific database management software.

**7.1 ER Notation**

There is no standard for representing data objects in ER diagrams. Each modeling methodology uses its own notation. The original notation used by Chen is widely used in academics texts and journals but rarely seen in either CASE tools or publications by non-academics. Today, there are a number of notations used; among the more common are Bachman, crow's foot, and IDEFIX.

All notational styles represent entities as rectangular boxes and relationships as lines connecting boxes. Each style uses a special set of symbols to represent the cardinality of a connection. The notation used in this document is from Martin. The symbols used for the basic ER constructs are:

* **Entities** are represented by labeled rectangles. The label is the name of the entity. Entity names should be singular nouns.
* **Relationships** are represented by a solid line connecting two entities. The name of the relationship is written above the line. Relationship names should be verbs
* **Attributes**, when included, are listed inside the entity rectangle. Attributes which are identifiers are underlined. Attribute names should be singular nouns.
* **Cardinality** of many is represented by a line ending in a crow's foot. If the crow's foot is omitted, the cardinality is one.

**Existence** is represented by placing a circle or a perpendicular bar on the line. Mandatory existence is shown by the bar (looks like a 1) next to the entity for an instance is required. Optional existence is shown by placing a circle next to the entity that is optional.

An Entity Relationship (ER) Diagram 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.

**Uses of entity relationship diagrams**

* Database design: ER diagrams are used to model and design relational databases, in terms of logic and business rules (in a logical data model) and in terms of the specific technology to be implemented (in a physical data model.) In software engineering, an ER diagram is often an initial step in determining requirements for an information systems project. It’s also later used to model a particular database or databases. A relational database has an equivalent relational table and can potentially be expressed that way as needed.
* Database troubleshooting: ER diagrams are used to analyze existing databases to find and resolve problems in logic or deployment. Drawing the diagram should reveal where it’s going wrong.
* Business information systems: The diagrams are used to design or analyze relational databases used in business processes. Any business process that uses fielded data involving entities, actions and interplay can potentially benefit from a relational database. It can streamline processes, uncover information more easily and improve results.
* Business process re-engineering (BPR): ER diagrams help in analyzing databases used in business process re-engineering and in modeling a new database setup.
* Education: Databases are today’s method of storing relational information for educational purposes and later retrieval, so ER Diagrams can be valuable in planning those data structures.
* Research: Since so much research focuses on structured data, ER diagrams can play a key role in setting up useful databases to analyze the data. ER Diagrams are composed of entities, relationships and attributes. They also depict cardinality, which defines relationships in terms of numbers. Here’s a glossary:

### Entity

* A definable thing—such as a person, object, concept or event—that can have data stored about it. Think of entities as nouns. Examples: a customer, student, car or product. Typically shown as a rectangle.
* **Entity type:**A group of definable things, such as students or athletes, whereas the entity would be the specific student or athlete. Other examples: customers, cars or products.
* **Entity set:** Same as an entity type, but defined at a particular point in time, such as students enrolled in a class on the first day. Other examples: Customers who purchased last month, cars currently registered in Florida. A related term is instance, in which the specific person or car would be an instance of the entity set.
* **Entity categories:** Entities are categorized as strong, weak or associative. A **strong entity** can be defined solely by its own attributes, while a **weak entity** cannot. An associative entity associates entities (or elements) within an entity set.
* **Entity keys:** Refers to an attribute that uniquely defines an entity in an entity set. Entity keys can be super, candidate or primary. **Super key:**A set of attributes (one or more) that together define an entity in an entity set. **Candidate key:**A minimal super key, meaning it has the least possible number of attributes to still be a super key. An entity set may have more than one candidate key. **Primary key:**A candidate key chosen by the database designer to uniquely identify the entity set. **Foreign key:**Identifies the relationship between entities.

### Relationship

* How entities act upon each other or are associated with each other. Think of relationships as verbs. For example, the named student might register for a course. The two entities would be the student and the course, and the relationship depicted is the act of enrolling, connecting the two entities in that way. Relationships are typically shown as diamonds or labels directly on the connecting lines.
* **Recursive relationship:**The same entity participates more than once in the relationship.

### Attribute

* A property or characteristic of an entity. Often shown as an oval or circle.
* **Descriptive attribute:**A property or characteristic of a relationship (versus of an entity.)
* **Attribute categories:**Attributes are categorized as simple, composite, derived, as well as single-value or multi-value. **Simple:** Means the attribute value is atomic and can’t be further divided, such as a phone number. **Composite:**Sub-attributes spring from an attribute. **Derived:**Attributed is calculated or otherwise derived from another attribute, such as age from a birthdate.
* **Multi-value:**More than one attribute value is denoted, such as multiple phone numbers for a person.
* **Single-value:** Just one attribute value. The types can be combined, such as: simple single-value attributes or composite multi-value attributes.

### **Cardinality**

* Defines the numerical attributes of the relationship between two entities or entity sets. The three main cardinal relationships are one-to-one, one-to-many, and many-many. A **one-to-one example** would be one student associated with one mailing address. A **one-to-many example (or many-to-one, depending on the relationship direction):** One student registers for multiple courses, but all those courses have a single line back to that one student. **Many-to-many example:**Students as a group are associated with multiple faculty members, and faculty members in turn are associated with multiple students.

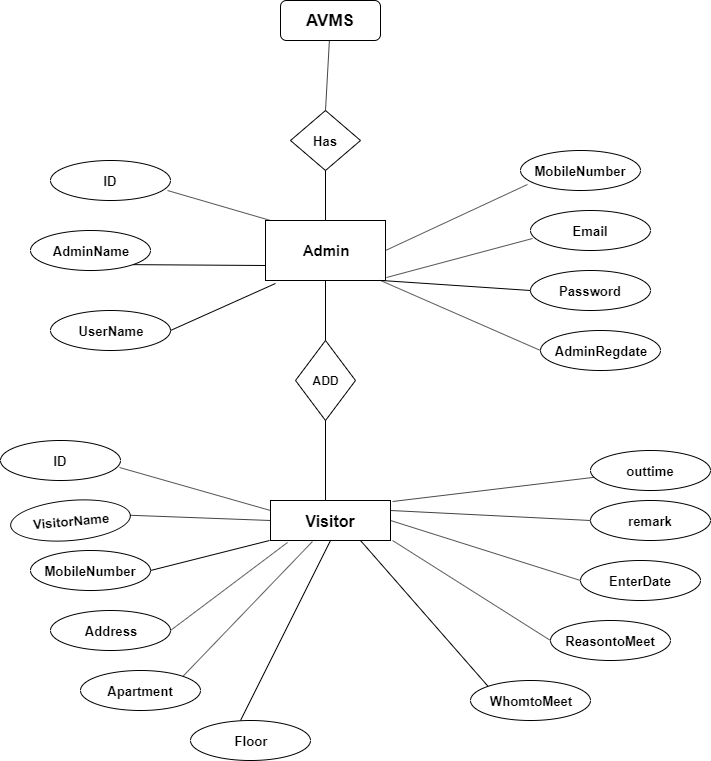
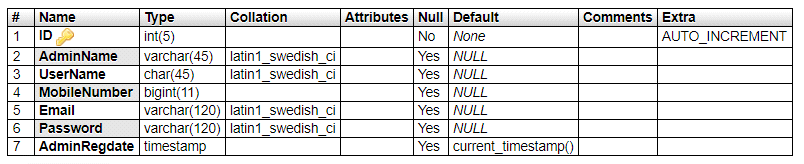


Fig.7.1

**MySQL Data Tables:**

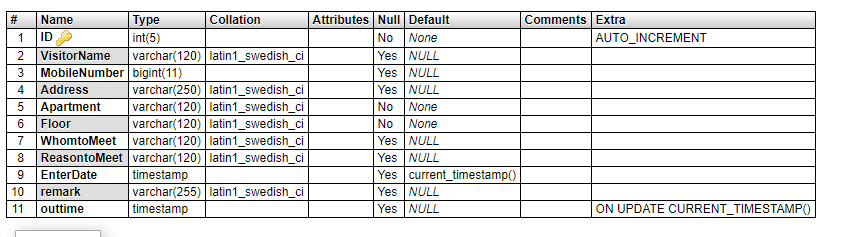
**7.2 Admin Table:(**Table name is admin)

This store admin personal and login details.



**7.3 Visitor Table:** (Table name is tblvisitor)

This store the visitor details and admin remark



**Chapter 8**

**Implementation and System Testing**

After all phase have been perfectly done, the system will be implemented to the server and the system can be used.

## Software Testing

**Software Testing** is a method to check whether the actual software product matches expected requirements and to ensure that software product is[Defect](https://www.guru99.com/defect-management-process.html)free. It involves execution of software/system components using manual or automated tools to evaluate one or more properties of interest. The purpose of software testing is to identify errors, gaps or missing requirements in contrast to actual requirements.

Some prefer saying Software testing definition as a [White Box](https://www.guru99.com/white-box-testing.html) and [Black Box Testing](https://www.guru99.com/black-box-testing.html). In simple terms, Software Testing means the Verification of Application Under Test (AUT). This Software Testing course introduces testing software to the audience and justifies the importance of software testing.

In this Software Testing tutorial, you will learn basics of software testing like:

* [What is Software Testing?](https://www.guru99.com/software-testing-introduction-importance.html#1)
* [Why is Software Testing Important?](https://www.guru99.com/software-testing-introduction-importance.html#2)
* [What are the benefits of Software Testing?](https://www.guru99.com/software-testing-introduction-importance.html#3)
* [Testing in Software Engineering](https://www.guru99.com/software-testing-introduction-importance.html#4)
* [Types of Software Testing](https://www.guru99.com/software-testing-introduction-importance.html#5)
* [Testing Strategies in Software Engineering](https://www.guru99.com/software-testing-introduction-importance.html#6)
* [Program Testing](https://www.guru99.com/software-testing-introduction-importance.html#7)

## Why Software Testing is Important?

**Software Testing is Important** because if there are any bugs or errors in the software, it can be identified early and can be solved before delivery of the software product. Properly tested software product ensures reliability, security and high performance which further results in time saving, cost effectiveness and customer satisfaction.

**What are the benefits of Software Testing?**

Here are the benefits of using software testing:

* **Cost-Effective:**It is one of the important advantages of software testing. Testing any IT project on time helps you to save your money for the long term. In case if the bugs caught in the earlier stage of software testing, it costs less to fix.
* **Security:**It is the most vulnerable and sensitive benefit of software testing. People are looking for trusted products. It helps in removing risks and problems earlier.
* **Product quality:**It is an essential requirement of any software product. Testing ensures a quality product is delivered to customers.
* **Customer Satisfaction:**The main aim of any product is to give satisfaction to their customers. UI/UX Testing ensures the best user experience.

## Types of Software Testing

Here are the software testing types:

Typically Testing is classified into three categories.

* Functional Testing
* Non-Functional Testing or [Performance Testing](https://www.guru99.com/performance-testing.html)
* Maintenance (Regression and Maintenance)

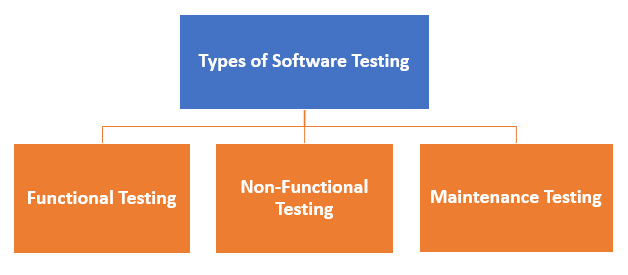


Fig 8.1

**Functional Testing**

Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. Each function is compared to the corresponding requirement to ascertain whether its output is consistent with the end user’s expectations. The testing is done by providing sample inputs, capturing resulting outputs, and verifying that actual outputs are the same as expected outputs.

Some functional testing examples are:

* Can users successfully log in to the application once they provide legitimate credentials?
* Does the payment gateway reject the input and display an error message when a user keys in an invalid credit card number?
* Do inputs to the “Add New Record” screen successfully add and save a new record to the database?

At the end of [functional testing](https://www.microfocus.com/solutions/functional-testing-software-testing), you should have software that has a coherent user.

**NON-FUNCTIONAL TESTING**

is defined as a type of Software testing to check non-functional aspects (performance, usability, reliability, etc) of a software application. It is designed to test the readiness of a system as per nonfunctional parameters which are never addressed by functional testing.

An excellent example of non-functional test would be to check how many people can simultaneously login into a software.

Non-functional testing is equally important as functional testing and affects client satisfaction.

**Maintenance testing**

is a test that is performed to either identify equipment problems, diagnose equipment problems, or confirm that repair measures have been effective. It can be performed at either the system level (e.g., the [HVAC](https://en.wikipedia.org/wiki/HVAC) system), the equipment level (e.g., the blower in an HVAC line), or the component level (e.g., a control chip in the control box for the blower in the HVAC line).

**8.1 System Testing**

The goal of the system testing process was to determine all faults in our project.The program was subjected to a set of test inputs and many explanations were made and based on these explanations it will be decided whether the program behaves as expected or not. Our Project went through two levels of testing

1. Unit testing

2. Integration testing

Unit testing is commenced when a unit has been created and effectively reviewed .In order to test a single module we need to provide a complete environment i.e. besides the section we would require

* The procedures belonging to other units that the unit under test calls
* Non local data structures that module accesses
* A procedure to call the functions of the unit under test with appropriate parameters

**1. Test for the admin module**

* **Testing admin login form-**This form is used for log in of administrator of the system. In this form we enter the username and password if both are correct administration page will open otherwise if any of data is wrong it will get redirected back to the login page and again ask the details.
* **Report Generation:** admin can generate report from the main database.

**INTEGRATION TESTING**

In the Integration testing we test various combination of the project module by providing the input.

The primary objective is to test the module interfaces in order to confirm that no errors are occurring when one module invokes the other module.

**Chapter 9**

**VALUATION**

**Project URL:** [**http://localhost/avms**](http://localhost/avms)

**Login Page**

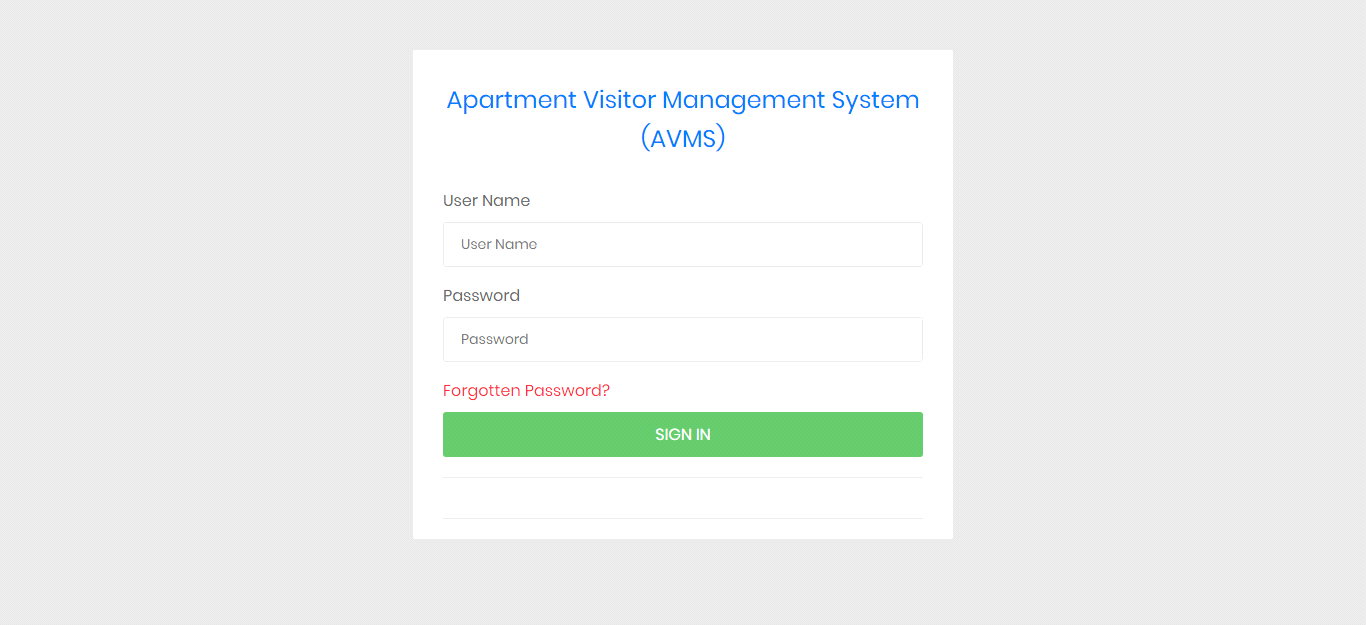


Fig.9.1

**Forgot Password**

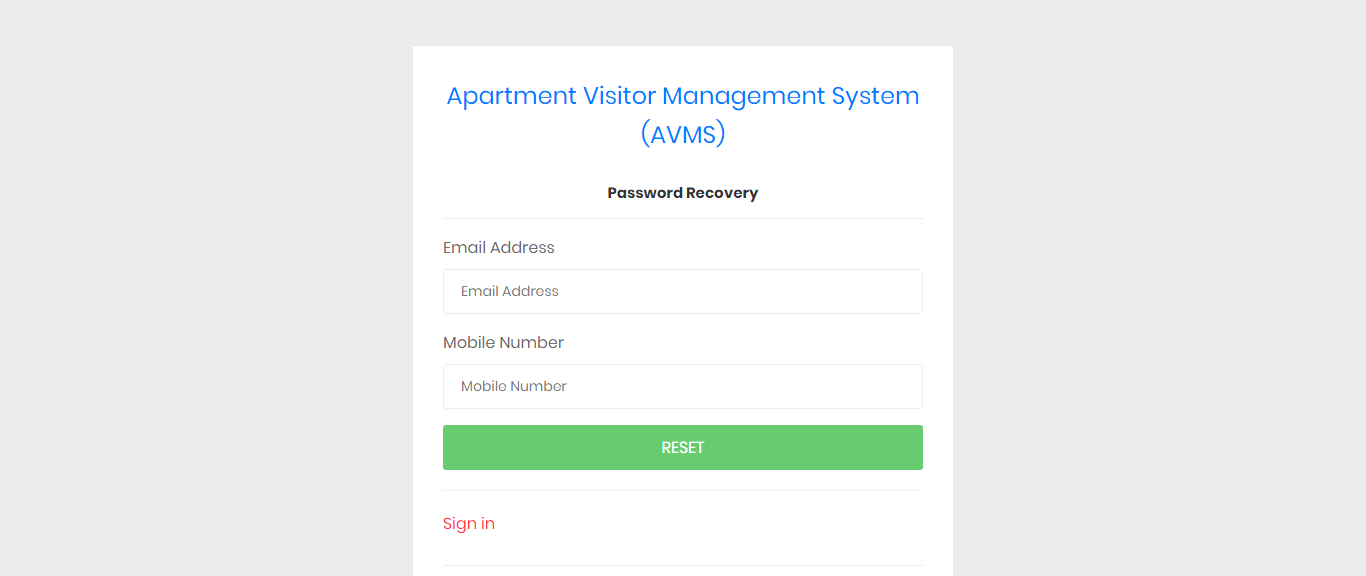


Fig.9.2

**Reset Password**

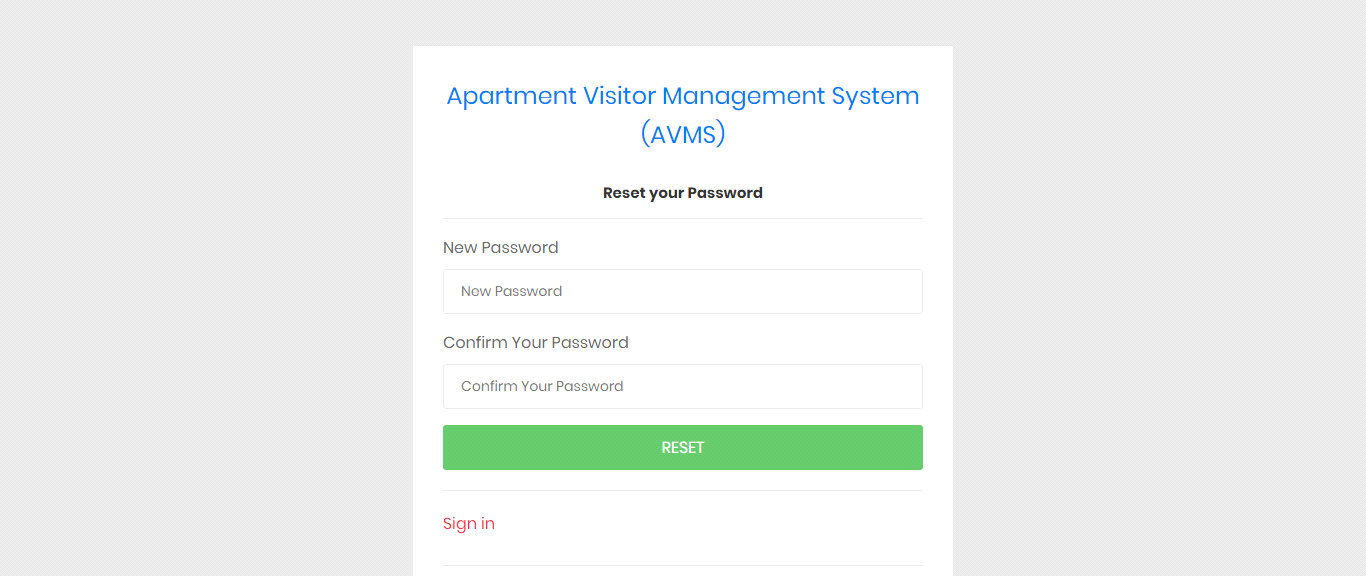


Fig.9.3

**Dashboard**

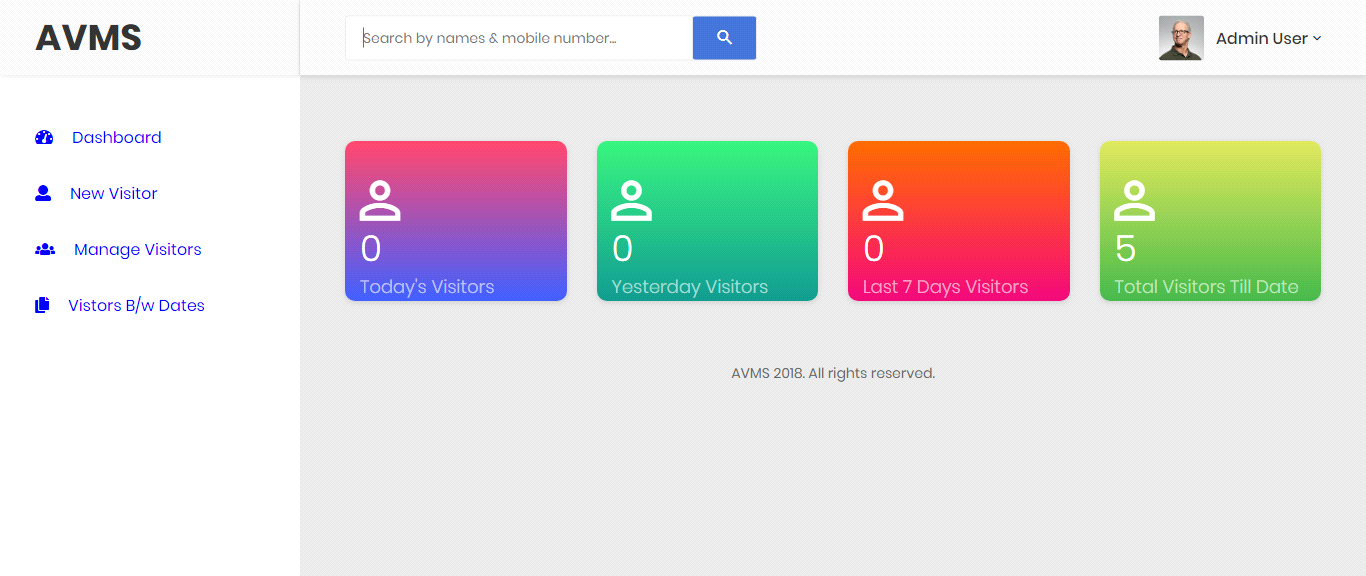


Fig.9.4

**Admin Profile**

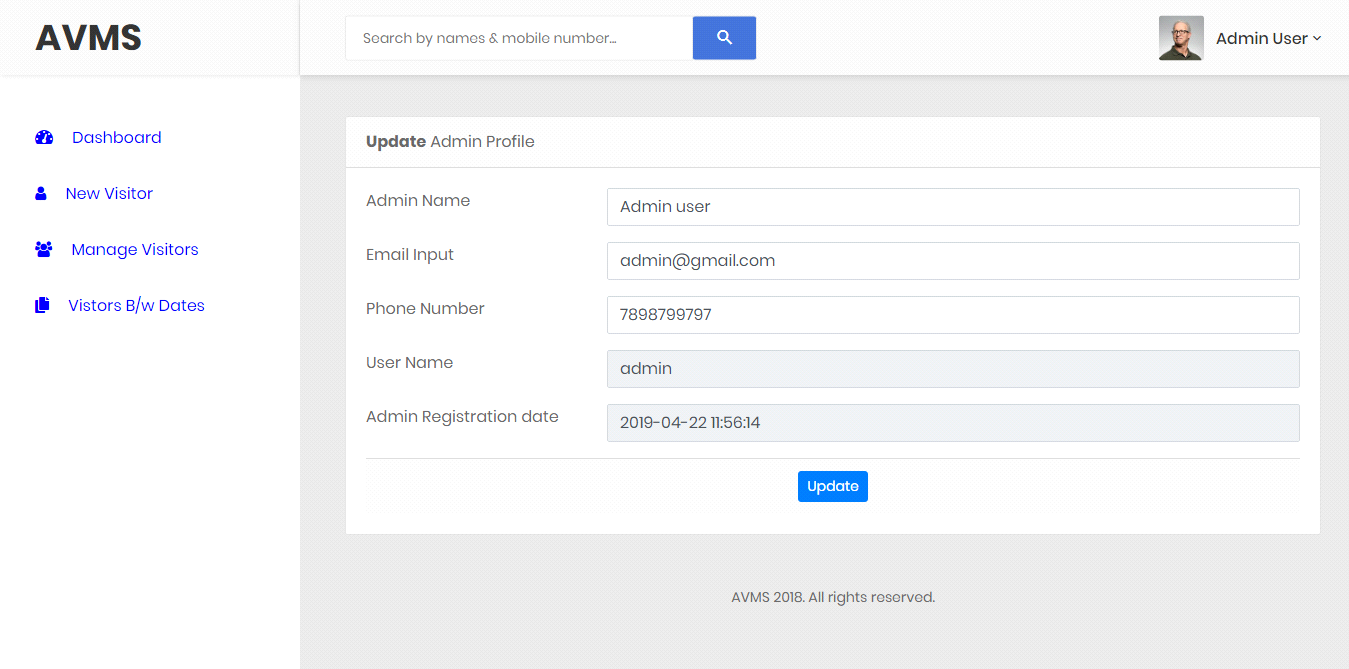


Fig.9.5

**Change Password**

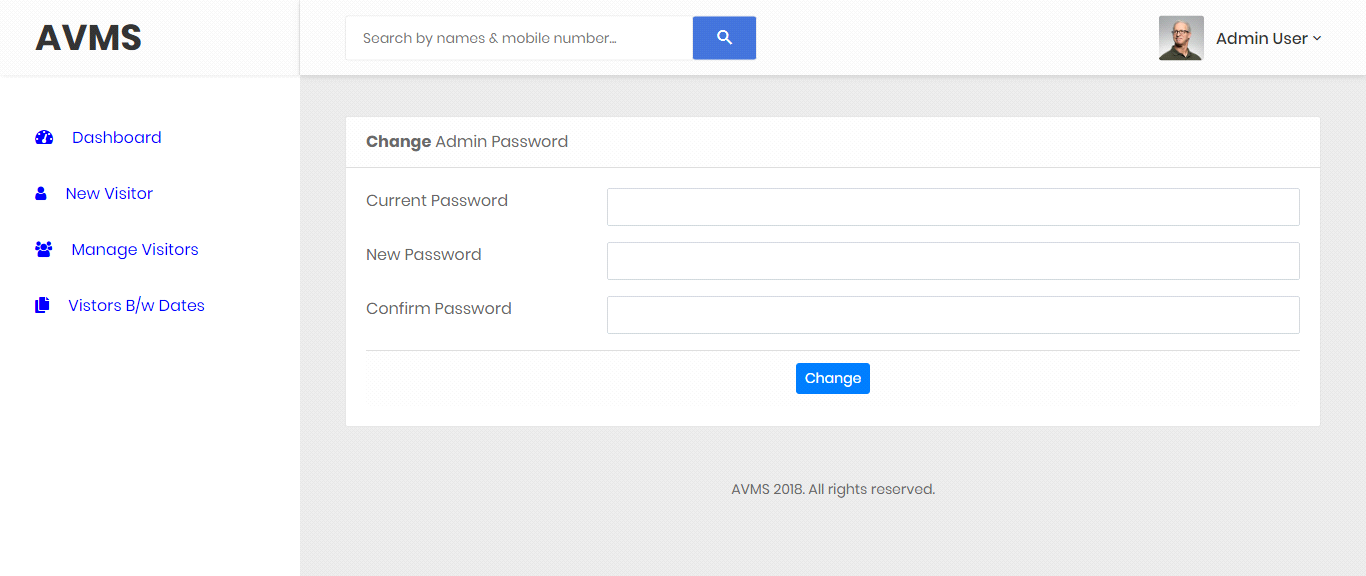


Fig.9.6

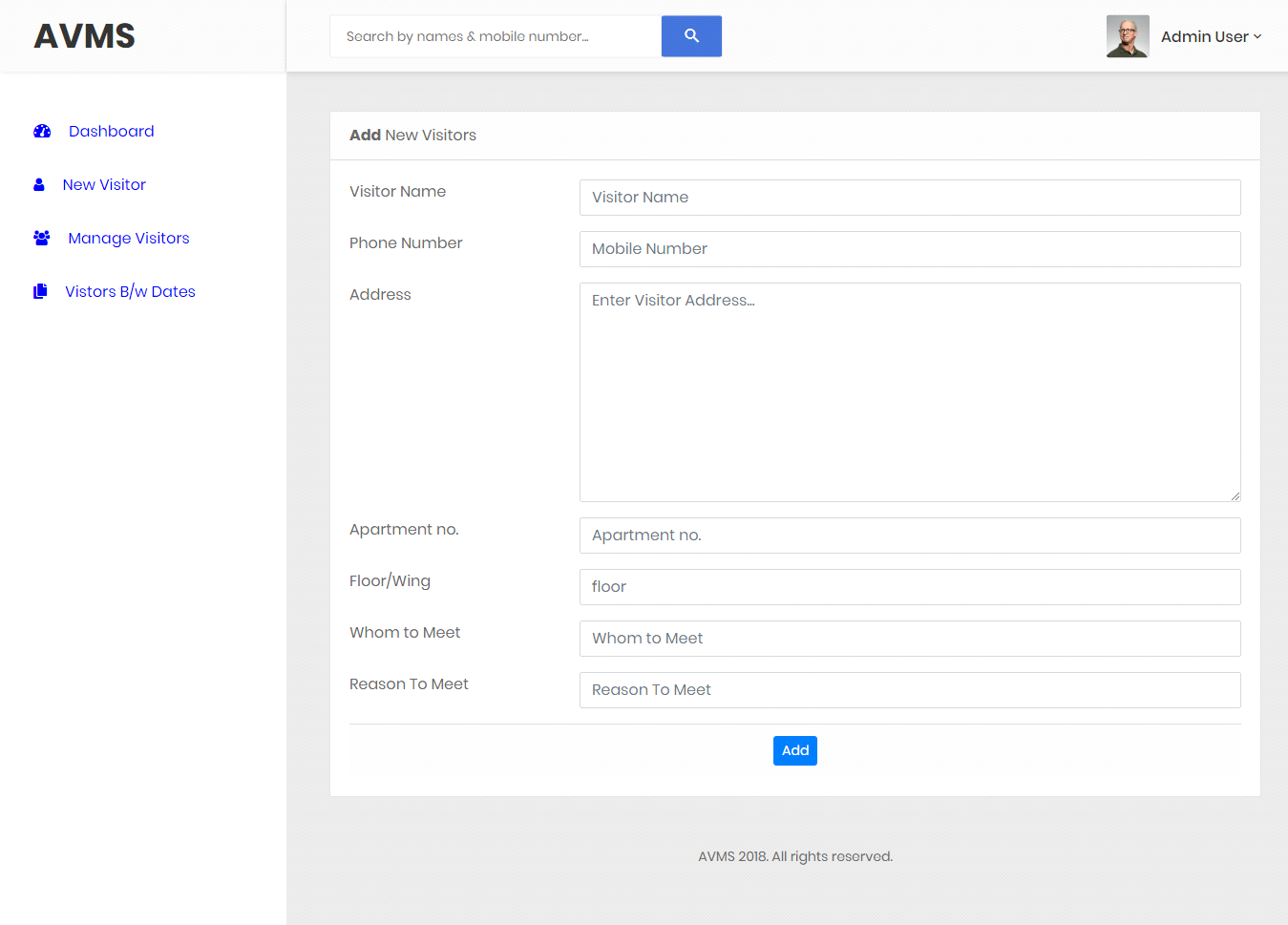
**Add New Visitor**

Fig.9.7

**Manage Visitor**

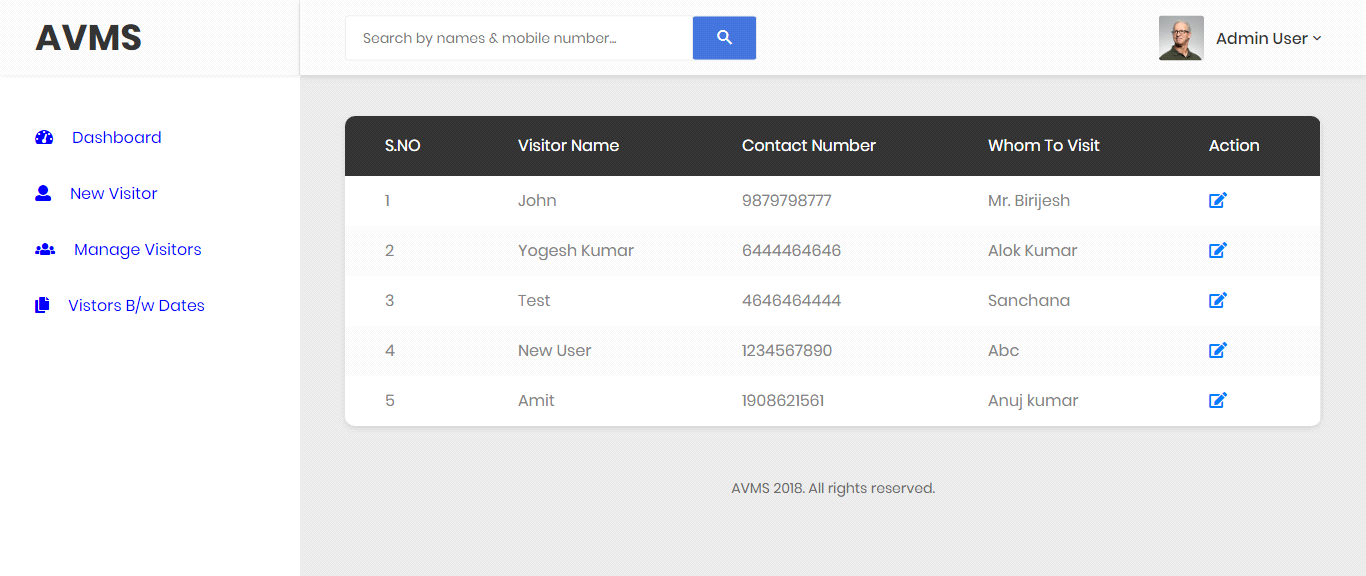


Fig.9.8

**Visitor Detail**

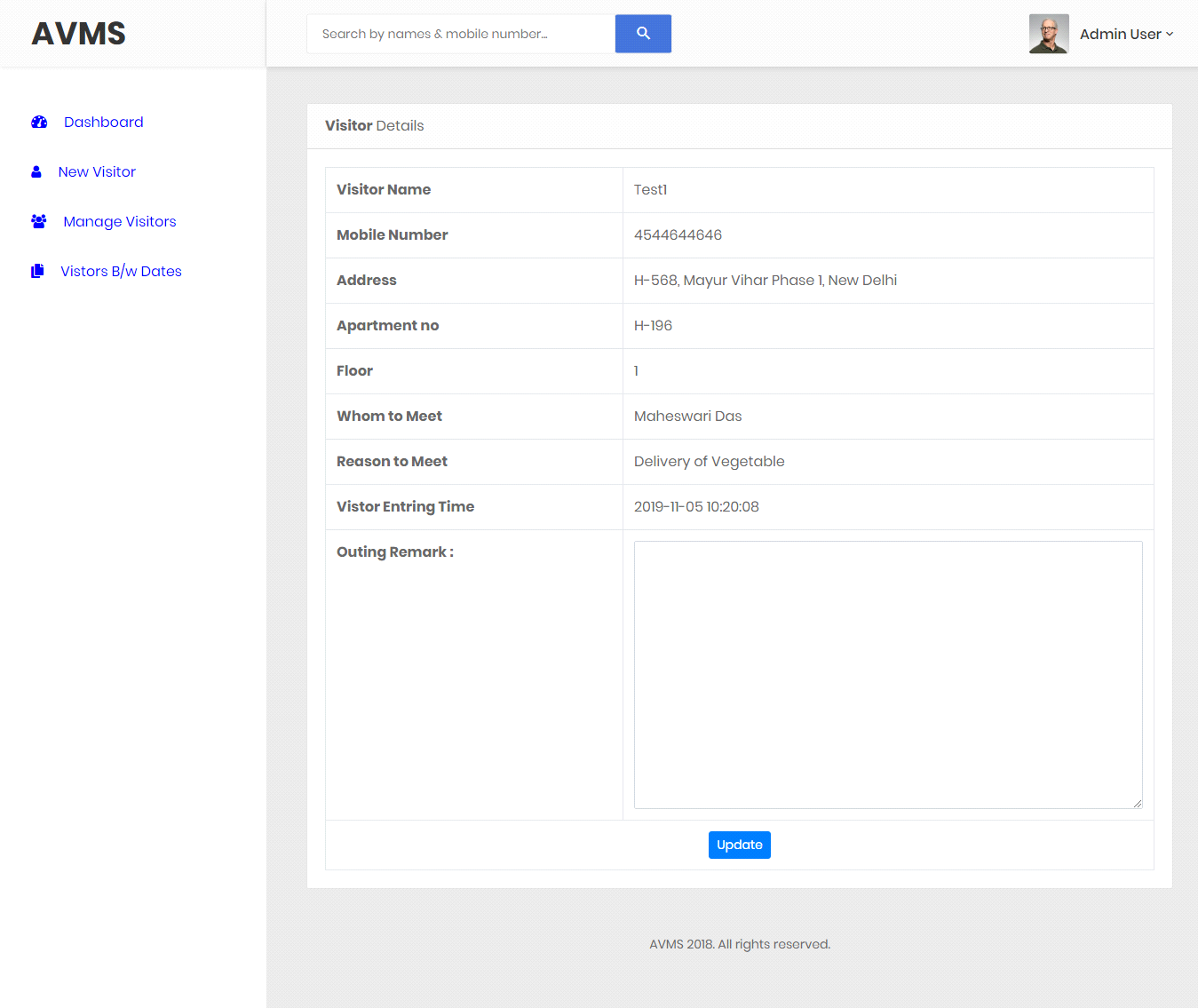


Fig.9.9

**Visitor Detail after Update**

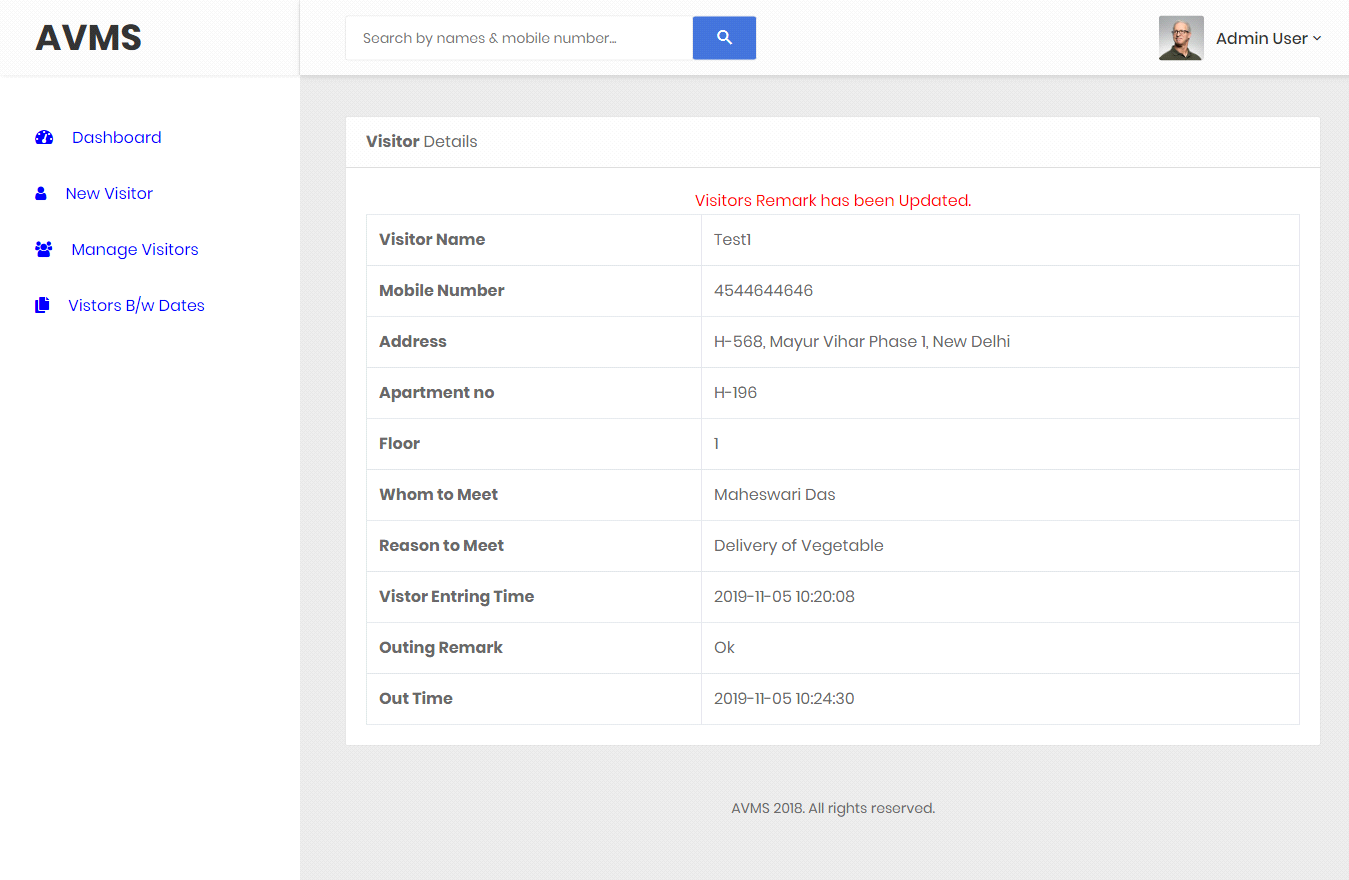


Fig.9.10

**Between Dates Reports**

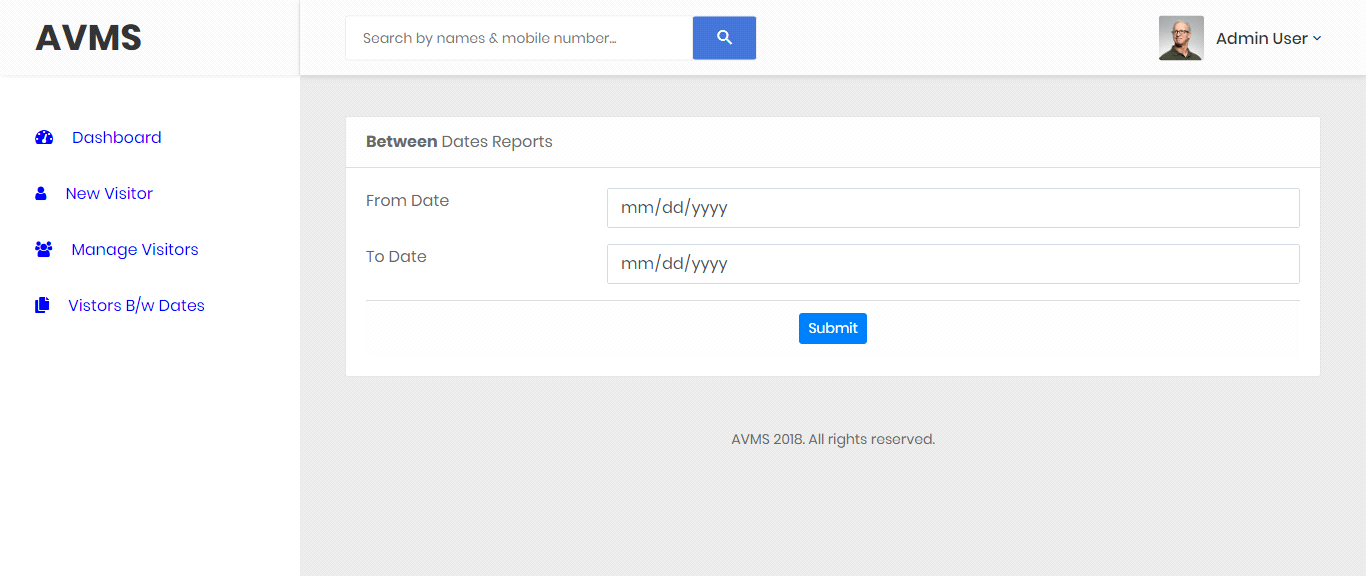


Fig.9.11

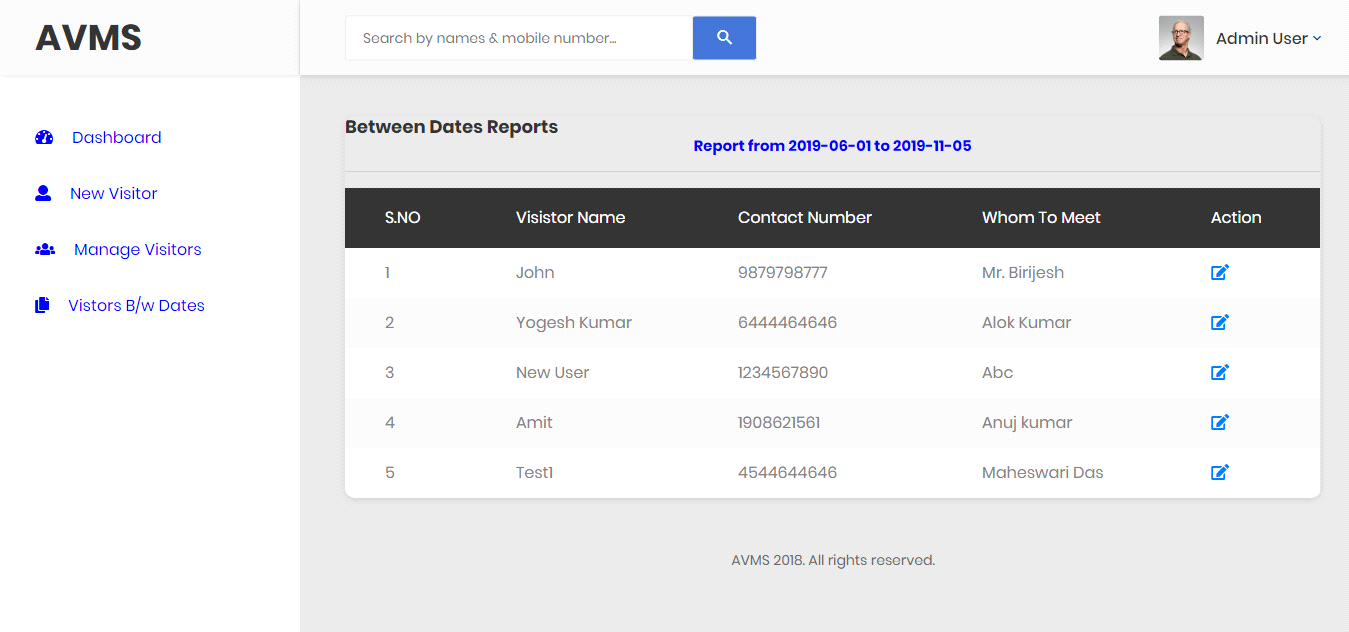


Fig.9.12

**CONCLUSION**

This Application provides a computerized version of Apartment Visitor Management System which will benefit the society.

It makes entire process online and can generate reports. It has a facility of staff’s login where staff can fill the visitor details and generate report.

The Application was designed in such a way that future changes can be done easily. The following conclusions can be deduced from the development of the project.

* Automation of the entire system improves the productivity.
* It provides a friendly graphical user interface which proves to be better when compared to the existing system.
* It gives appropriate access to the authorized users depending on their permissions.
* It effectively overcomes the delay in communications.
* Updating of information becomes so easier.
* System security, data security and reliability are the striking features.
* The System has adequate scope for modification in future if it is necessary.

**References**

1. . Yaish, H., Goyal, M., & Feuerlicht, G. (2014). Evaluating the performance of multi-tenant elastic extension tables. Procedia Computer Science, 29, 614-626.
2. Peng, J., Zhang, X., Lei, Z., Zhang, B., Zhang, W., & Li, Q. (2009, December). Comparison of several cloud computing platforms. In 2009 Second international symposium on information science and engineering (pp. 23-27). IEEE.
3. Li, M., Kankanhalli, A., & Kim, S. H. (2016). Which ideas are more likely to be implemented in online user innovation communities? An empirical analysis. Decision Support Systems, 84, 28-40
4. Kovach, J. J., Atasu, A., & Banerjee, S. (2018). Salesforce incentives and remanufacturing. Production and Operations Management, 27(3), 516-530.
5. Ntayi, J. M., Munene, J. C., & Eyaa, S. (2010). Salesforce behavioural performance of accounts relationship managers (ARMS) in Uganda's commercial banks: A qualitative analysis. Journal of Retail & Leisure Property, 9(1), 5-23
6. Anderson, E., & Oliver, R. L. (1987). Perspectives on behavior-based versus outcome-based salesforce control systems. Journal of Marketing, 51(4), 76-88.
7. Samaraweera, M., & Gelb, B. D. (2015). Formal salesforce controls and revenue production: a meta-analysis. Journal of Personal Selling & Sales Management, 35(1), 23-32.
8. Mason, M. (2018). Overcoming the risks of privileged user abuse in Salesforce. Network Security, 2018(8), 6-8
9. Fisher, S. (2007, May). The architecture of the apex platform, salesforce. com's platform for building on-demand applications. In 29th International Conference on Software Engineering (ICSE'07 Companion) (pp. 3-3). IEEE..
10. Espadas, J., Concha, D., & Molina, A. (2008, October). Application Development over Software-as-a-Service platforms. In 2008 the third international conference on software engineering advances (pp. 97-104). IEEE.
11. Churchill Jr, G. A., Ford, N. M., & Walker Jr, O. C. (1976). Organizational climate and job satisfaction in the salesforce. Journal of Marketing Research, 13(4), 323-332.
12. Dai, Y., & Chao, X. (2016). Price delegation and salesforce contract design with asymmetric risk aversion coefficient of sales agents. International Journal of Production Economics, 172, 31-42.
13. Mietzner, R., & Leymann, F. (2010, December). A self-service portal for service-based applications. In 2010 IEEE International Conference on Service-Oriented Computing and Applications (SOCA) (pp. 1-8). IEEE.
14. Hayes, B. (2008). Cloud computing..
15. Netto, M.; Calheiros, R.; Rodrigues, E.; Cunha, R.; Buyya, R. (2018). "HPC Cloud for Scientific and Business Applications: Taxonomy, Vision, and Research Challenges". *ACM Computing Surveys*. **51** (1): 8:1–8:29.