

# CHAPTER 1

## INTRODUCTION

### 1.1 GENERAL INTRODUCTION

This document is intended to provide the requirement for a web based Online Hostel Management System, which in turn provide the basis of architectural models and development of application. Large increase of strength of students enrolling every year lead to the rise in the accommodation lead to the issue of their management.

#### 1.1.1 PURPOSE

The overall purpose is to develop website that provide an efficient pathway for managing the hostels, reduces inconsistency and redundancy in the data entries, provide ease of access to both student and warden/administration and security at its tip. It will provide the user-friendly interface for both student and administration and thereby provide an ease of access and reduction in the manual input, manual error and a way to reduce the wastage of paper. It serves as a step to the Digital World in this era of Digitalization.

#### 1.1.2 DOCUMENT CONVENTIONS

- Light weighted italics fonts are marked for the various software and dependencies.
- Highlighted texts are available in Index.
- The words marked like “**word**” are the modules used in the project.
- The words like word represents each use-case
- Priorities are based on star symbols where 5 star are more prioritize one

#### 1.1.3 PRODUCT SCOPE

The ‘Online Hostel Management System’ is web development module that provide an efficient way to manage various tasks in the hostels of National Institute of Technology, Puducherry. It consists of eight modules- Student Allocation System, Mess Feedback System, Attendance and Entry/ Exit System, Notice Display System, Grievance/Complaint System, Medicine and sports Accessing System and Sick room/ Visitor room/Common room accessing system and along with hostellers and Wardens Repository.

## **1.2 OBJECTIVE**

To digitize cumbersome, time-consuming and resource draining task of Academic feed-back, Attendance and examination marks management, and registration by reduce the paper work and computerize the labour-intensive tasks through an interactive and user-friendly web-based interface.

## **1.3 MOTIVATION**

- Today, people are more and more dependent on internet for everything.
- More strength and strain of manual worker needed for hostel Management
- Quite Inconsistency in handling security for In/Out.
- Data redundancy and sometime loss of data.
- Lots of paper work is required to handle.
- Difficult to handle large strength of students.
- Consumes lots of time for Management.
- Difficult to update data.
- Delay in communication & Information of Imp Data.
- Increasing Number of Students.

## **1.4 ORGANISATION OF THE CHAPTERS**

Chapter 2 – Literature Review

Chapter 3 – Description of Project

Chapter 4 – Implementation Stage-I: Requirement Analysis

Chapter 5 - Implementation Stage-II: The Project Planning

Chapter 6 – Implementation Stage-III: The Project Modelling

Chapter 7 - Implementation Stage-IV: Construction

Chapter 8 -Experimental Results

Chapter 9- Summary and Conclusion

Chapter 10- References

Chapter 11- Appendix

# CHAPTER 2

## LITERATURE REVIEW

### 2.1 PHP

PHP stands for PHP Hyper-Text Preprocessor. PHP is one of the leading web development languages, however, the various model of PHP including the development model is operated without any structure, which results into blending of the code of data access, the processing of business ideas and trending logic, and web presentation layer together, as a result, it brought about many problems in the web applications, meanwhile, it could not meet the rapid development of web apply any more. He working and algorithms of PHP is based on MVC design patterns which is FDF framework which was provided for PHP developers, which can offer a framework for web applications, separate the data, view and control of web applications, afford to achieve loose coupling and high cohesion, thereby enhanced the efficiency, reliability, maintainability and scalability of application development.

One of the more interesting developments recently gaining popularity in the server-side JavaScript space is MEAN Stack with Node.js. It's a framework for developing high-performance, concurrent programs that don't rely on the mainstream multithreading approach but use asynchronous I/O with an event-driven programming model <sup>[2]</sup>. Node.js is typically a viable option in the present world for web development due to its highly scalability, efficiency, flexible and concurrency.

### 2.2 MongoDB and MEAN

MongoDB is a cross-platform document-oriented database program. Classified as a NoSQL database program, MongoDB uses JSON-like documents with schemata. MongoDB is developed by MongoDB Inc. and licensed under the Server Side Public License (SSPL).

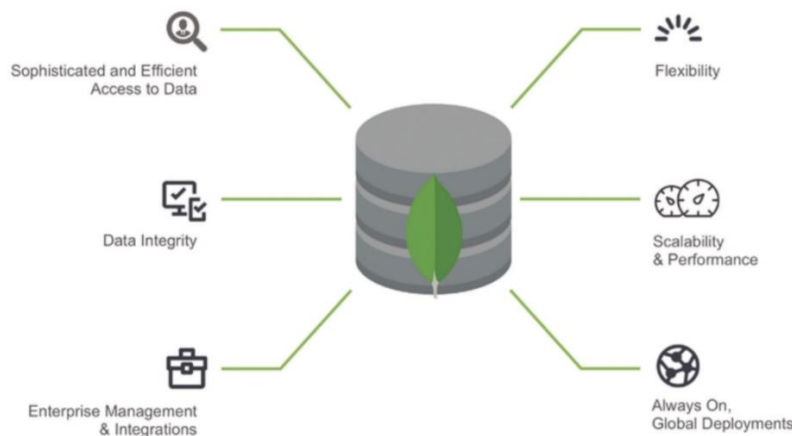


Fig: 2.1: MongoDB Features

MEAN is a free and open-source JavaScript software stack for building dynamic web sites and web applications. The MEAN stack is MongoDB, Express.js, AngularJS (or Angular), and Node.js. Because all components of the MEAN stack support programs that are written in JavaScript, MEAN applications can be written in one language for both server-side and client-side execution environments.<sup>0</sup>

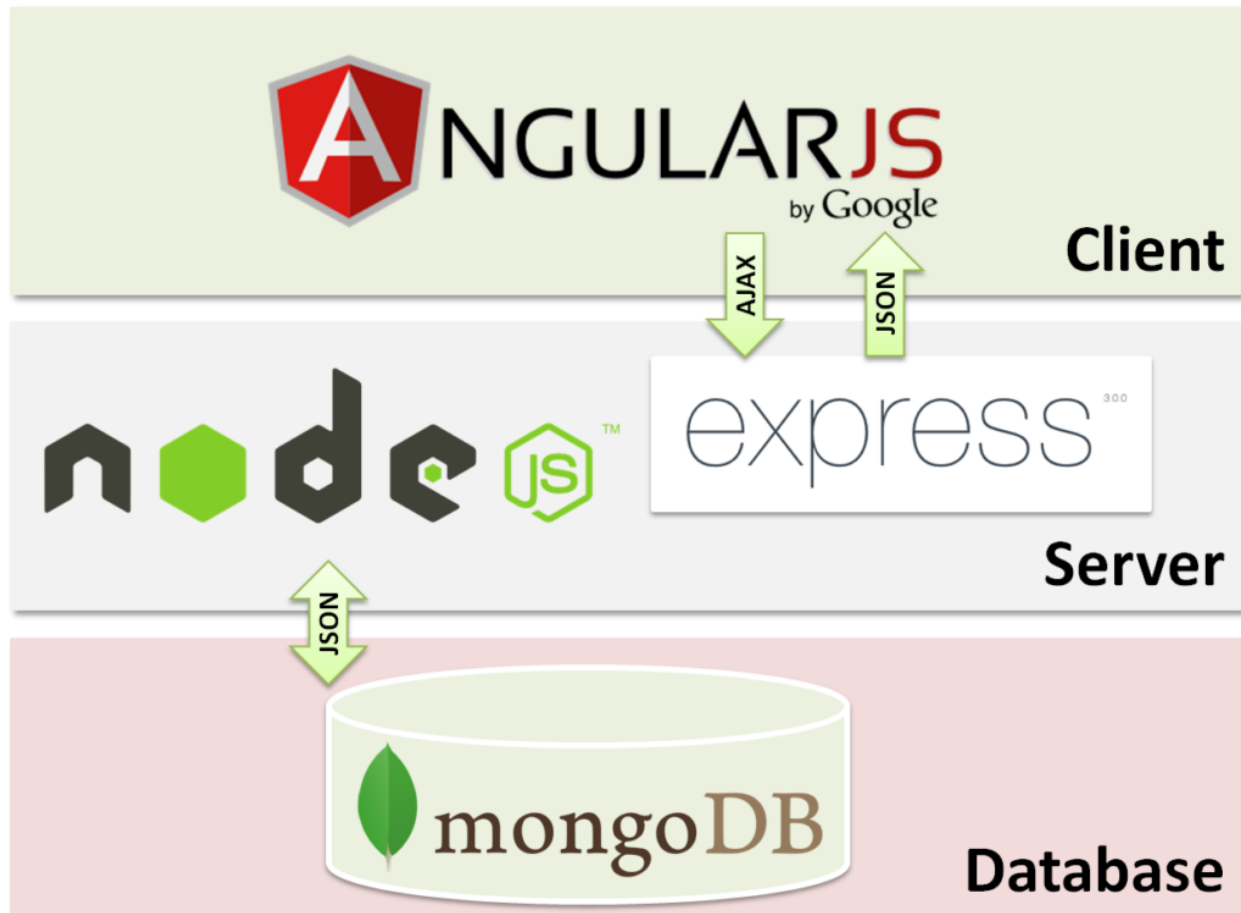


Fig: 2.2: MEAN STACK

## 2.2.1 FEATURES

- **Ad-Hoc Queries**

MongoDB supports field, range query, and regular expression searches. Queries can return specific fields of documents and also include user-defined JavaScript functions. Queries can also be configured to return a random sample of results of a given size.

- **Indexing**

Fields in a MongoDB document can be indexed with primary and secondary indices.

- **Replication**

MongoDB provides high availability with replica sets. A replica set consists of two or more copies of the data. Each replica set member may act in the role of primary or secondary replica at any time. All writes and reads are done on the primary replica by default. Secondary replicas maintain a copy of the data of the primary using built-in replication. When a primary replica fails, the replica set automatically conducts an election process to determine which secondary should become the primary. Secondaries can optionally serve read operations, but that data is only eventually consistent by default.

- **Load- Balancing**

MongoDB scales horizontally using sharding. The user chooses a shard key, which determines how the data in a collection will be distributed. The data is split into ranges (based on the shard key) and distributed across multiple shards. (A shard is a master with one or more replicas.). Alternatively, the shard key can be hashed to map to a shard – enabling an even data distribution.

MongoDB can run over multiple servers, balancing the load or duplicating data to keep the system up and running in case of hardware failure.

- **File Manager Tool**

MongoDB can be used as a file system, called GridFS, with load balancing and data replication features over multiple machines for storing files.

This function, called grid file system, is included with MongoDB drivers. MongoDB exposes functions for file manipulation and content to developers. GridFS can be accessed using mongofiles utility or plugins for Nginx and lighttpd. GridFS divides a file into parts, or chunks, and stores each of those chunks as a separate document.

- **Capped Collection**

MongoDB supports fixed-size collections called capped collections. This type of collection maintains insertion order and, once the specified size has been reached, behaves like a circular queue.

- **Transactions**

Support for multi-document ACID transactions was added to MongoDB with the General Availability of the 4.0 release in June 2018.

- **Aggregation**

MongoDB provides three ways to perform aggregation: the aggregation pipeline, the map-reduce function, and single-purpose aggregation methods. Map-reduce can be used for batch processing of data and aggregation operations. But according to MongoDB's documentation, the Aggregation Pipeline provides better performance for most aggregation operations. The aggregation framework enables users to obtain the kind of results for which the SQL GROUP BY clause is used. Aggregation operators can be strung together to form a pipeline – analogous to Unix pipes. The aggregation framework includes the \$lookup operator which can join documents from multiple documents, as well as statistical operators such as standard deviation.

- **Server-Side Java-Script Execution**

JavaScript can be used in queries, aggregation functions (such as MapReduce), and sent directly to the database to be executed.

## 2.2.2 MongoDB v/s TRADITIONAL SQL (MySQL)

The distinction of MongoDB and MySQL are as follows:

**Table 1: Distinction of MongoDB and my SQL terminologies**

Sl. No.	MySQL	MongoDB
1	Attribute	Field
2	Record	Document
3	Table	Collections
4	Database	Database
5	Index	Index

- **The database:** In simple words it can be called as the physical container for data. Each of the databases has its own set of files on the file system with multiple databases existing on a single MongoDB server.

- **The Collection** : A group of database documents can be called as a collection. The RDBMS equivalent of the collection is a table. The entire collection exists within a single database. There are no schemas when it comes to collections. Inside the collection the various documents can have varied fields but mostly the documents within a collection are meant for the same purpose or serving the same end goal.
- **The Document** : A set of key-value pairs can be designated as a document. Documents are associated with dynamic schemas. The benefit of having dynamic schemas is that document in a single collection does not have to have the same structure or fields. Also the common fields in a collection's document can have varied types of data.

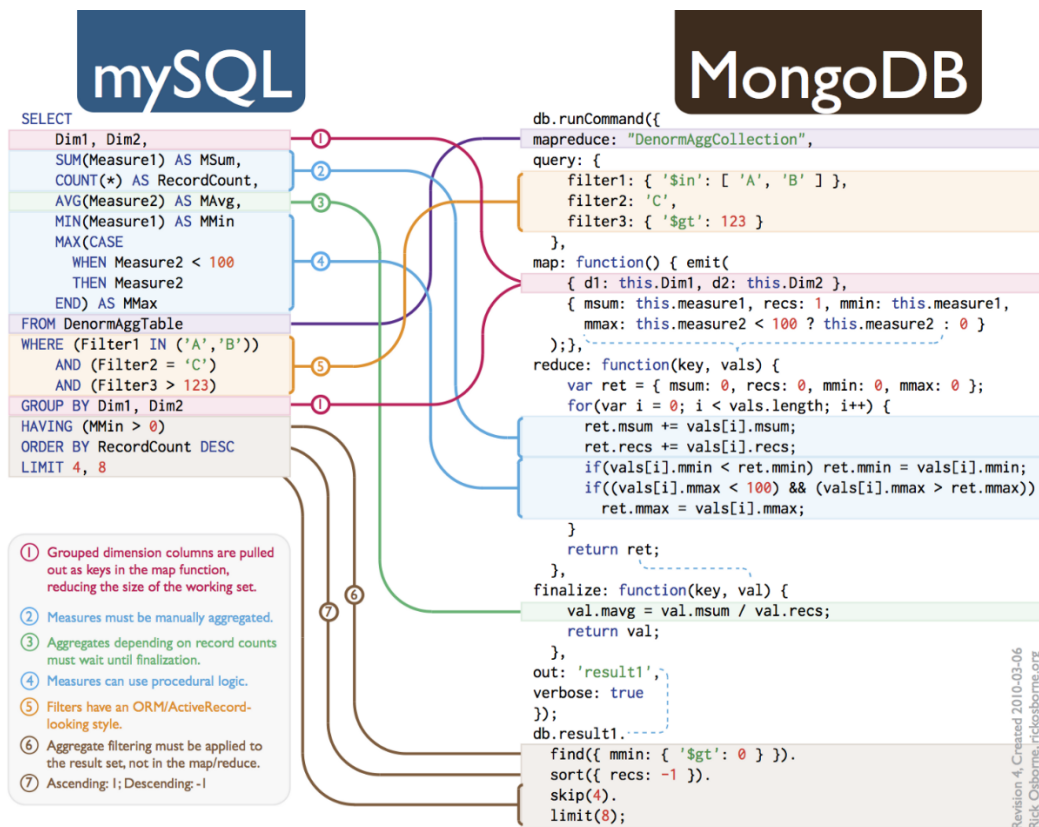


Fig: 2.3: Mongo v/s MySQL command

There are no join operations in MongoDB and no Normalization is required. In MongoDB there can be redundancy of data which in turn may help in scaling of database to larger extent.

## 2.2.3 PROPERTIES



MySQL eg. MariaDB, Oracle	NoSQL eg. MongoDB, Qizx
<b>ACID</b>	<b>BASE</b>
	
<b>Atomicity</b> <b>Consistency</b> <b>Isolation</b> <b>Durability</b>	<b>Basically</b> <b>Available</b> <b>Soft State</b> <b>Eventually Consistency</b>

Fig: 2.4: ACID v/s BASE Properties

### 2.2.3.1 NoSQL

NoSQL in horizontal scaling follows the BASE properties which is derived from CAP Theorem which indicate the properties that a system can cannot have consistency, availability and fault tolerance capabilities at same instance. BASE property indicates the following properties:

- **Basically, Available:** This property state that the database system does guarantee availability of data in the system at all instances.
- **Soft state:** Soft state of the database system indicates that the state of the system may change over time with various transactions. The state may even change without any input to the database system.
- **Eventual consistency:** This property indicates that the system tends to become consistent within a time-interval, given that the system doesn't acquire any input in that particular time-interval.

### 2.2.3.2 RELATIONAL DATABASES

A Relational Database Management System maintain Atomicity, Consistency, Isolation, and Durability which is known as ACID properties which ensure accuracy, reliability, safety and integrity of the transactions which are performed in the database system. These properties are mandatory for the accurately and reliability managing the Transactions in any of any Relational Database.



- **Atomicity**

Atomicity is a property which states that the operations in transaction must either be fully executed or none of them should be executed. There must be no state in a database system where the partial transaction is executed with some of its operations.

- **Consistency**

Consistency is a property which state that database must remain in a consistent state after any transaction. If the database was in a consistent state before the execution of a transaction, it must remain consistent after the execution of the transaction as well.

- **Durability**

This property of state that database should hold all its latest updates and the committed transactions must survived even if the system fails or roll back or the entire system get restarted.

- **Isolation**

This property of a database system applies to a system where more than one transaction is executed simultaneously. The property of isolation states that any transaction executed in the system should not be affected by the existence of any other transaction present in the system.

## **2.3 MEAN STACK**

MEAN is a Web Development Framework overcoming by Traditional LAMP Stack. The components of the MEAN stack are as follows:

- **MongoDB**, a NoSQL database
- **Express.js**, a web application framework that runs on Node.js
- **Angular.js** or **Angular**, JavaScript MVC frameworks that run in browser JavaScript engines
- **Node.js**, an execution environment for event-driven server-side and networking applications

## 2.4 DATABASE ATTACK AND MongoDB

Databases hosted on MongoDB Atlas are secure by default. Atlas encrypts your data, both in-transit and at-rest, and makes it easy to control access with role-based user management.

MongoDB provides various features, such as authentication, access control, encryption, to secure your MongoDB deployments. Some key security features include:

Table 2: MongoDB Security Mechanisms

Authentication	Authorization		TLS/SSL	Type
Authentication  SCRAM  x.509	Role-Based Access Control		TLS/SSL (Transport Encryption)	Kerberos Authentication
	Enable Control	Access	Configure mongod and mongoose for TLS/SSL	LDAP Proxy Authentication
	Manage and Roles	Users	TLS/SSL Configuration for Clients	Encryption at Rest Auditing

# CHAPTER 3

## DESCRIPTION OF PROJECT

### 3.1 PRODUCT PERSPECTIVE

This web-based system on “Online Hostel Management System” provides efficient way to handle hostel tasks in National Institute of Technology, Puducherry. It provides a user-friendly interface, ease of accessibility for hostellers, wardens, chief warden, security and administrators, thereby reduces the inconsistency, redundancy and misplacing of data in the administration and efficiently handles the various issues like attendance, room allocations and mess feedback in an entire academic year.

### 3.2 PRODUCT FUNCTIONS

The web-based development module on Online Hostel Management System focuses on nine perspectives – **hostellers(students), wardens, Assistant warden, Mess Representative, hostel Representative, Student- Council, Chief Wardens, Security and Admin.** There are 12 features will be illustrated by our system:

### 3.3 SYSTEM FEATURES

#### 3.3.1 System Feature 1: Allocation of Students

##### 3.3.1.1 Description and Priority

*Students are allotted based on the availability of Room and allotted student can again can be re-allotted in the hostel after a semester or two. The room allocation data can also contain the information about the room accessories like cots, tables, chairs etc.*

**Priority:** ☆ ☆ ☆ ☆ ☆

##### 3.3.1.2 Functional Requirements

REQ-1: PHP Libraries for accessing the form data

REQ-2: MongoDB database for storing the data and retrieve and update it

#### 3.3.2 System Feature 2: Accessing Sick room/Common rooms

##### 3.3.2.1 Description and Priority

*Student can access the sick room/common room by registering online along with the durations which will0 again be verified by security while providing key of the rooms.*

**Priority:** ☆ ☆ ☆

##### 3.3.2.2 Functional Requirements

REQ-3: Display Notification of the present key-holders

### 3.3.3 System Feature 3: Separate Login for Warden, hostellers, Chief Warden, Security and administration

#### 3.3.3.1 Description and Priority

*Separate the list of User into three categories: student, faculties and administration each having separate area of concern. Each of them must have the separate accessibility based on their Responsibilities.*

**Priority:** ☆ ☆ ☆ ☆ ☆

#### 3.3.3.2 Functional Requirements

REQ-11: A collection in the database to handle the login and other details.

REQ-12: “Phpmailer” library to carry out various email activities with SMTP server

REQ-13: “way2sms” module to carry out various SMS activities

### 3.3.4 System Feature 4: Accessing Visitor Room

#### 3.3.4.1 Description and Priority

*Visitor Room can be accessed by student in case of any visitor visits or can be accessed by warden/chief warden for any chief visitor visiting the hostel.*

**Priority:** ☆ ☆ ☆

#### 3.3.4.2 Functional Requirements

REQ-5: *PhpMailer* for mailing from PHP via SMTP Server provided

### 3.3.5 System Feature 5: Giving Mess feedback

#### 3.3.5.1 Description and Priority

*Mess feedback system is for giving the feedback to the mess based on various Criteria and then taking the cumulative average of the responses and display it to the wardens/ chief warden for further actions*

**Priority:** ☆ ☆ ☆ ☆

#### 3.3.5.2 Functional Requirements

REQ-6: jQuery libraries to accumulate the various feedback on each Criteria

REQ-7: Angular.js to give modular approach to the web module.

### 3.3.6 System Feature 6: Accessing Mess Menu

#### 3.3.6.1 Description and Priority

*Students can see the mess menu and any changes of mess menu can be displayed immediately*

**Priority:** ☆ ☆ ☆

#### 3.3.6.2 Functional Requirements

REQ-8: A database collection for mess menu

### 3.3.7 System Feature 7: Displaying the Notices

#### 3.3.7.1 Description and Priority

*Wardens/Chief wardens draft the notice which can be displayed to all users/hostellers/particular user via this module.*

**Priority:** ☆ ☆ ☆ ☆

#### 3.3.7.2 Functional Requirements

REQ-9: jQuery libraries to accumulate the various feedback on each Criteria

### 3.3.8 System Feature 8: Accessing Medicine and Sport accessories

#### 3.3.8.1 Description and Priority

*Student gets the details of the medicines available and sport accessories that are accessible and available and can register if required.*

**Priority:** ☆ ☆ ☆

#### 3.3.8.2 Functional Requirements

REQ-4: A separate collection for storing medicine a sports information.

### 3.3.9 System Feature 9: Collecting of Grievance and complaint

#### 3.3.9.1 Description and Priority

Student fill the grievance form and submitted form will be displayed to warden/Chief warden for further actions.

**Priority:** ☆ ☆ ☆ ☆

#### 3.3.9.2 Functional Requirements

REQ-10: A separate collection in Database.

### 3.3.10 System Feature 10: Attendance Entry and Exit

#### 3.3.10.1 Description and Priority

*On each entry and exit of the hostel the response would be recorded and attendance will be calculated leading to spontaneous attendance of students. The entry and exit will be verified by wardens and security staffs*

**Priority:** ☆ ☆ ☆ ☆

#### 3.3.10.2 Functional Requirements

REQ-1: Ajax for calculating attendance on each the entry and exit of hostellers in the hostel

REQ-2: TBD (To Be Determine)

### 3.3.11 System Feature 11: General and Emergency Contact Info

#### 3.3.11.1 Description and Priority

*A General and Emergency contact Info are updated and viewed by the students and hostel representative respectively.*

**Priority:** ☆ ☆ ☆ ☆

#### 3.3.11.2 Functional Requirements

PHP and MongoDB

### 3.3.12 System Feature 12: Allocation and De-allocation of Representatives

#### 3.3.12.1 Description and Priority

*A module for allotting the representatives and deallocating them, namely, mess representative, hostel representative and student-council members*

**Priority:** ☆ ☆ ☆ ☆

#### 3.3.12.2 Functional Requirements

PHP and MongoDB

## 3.3 USE CASES AND CHARACTERISTICS

The entire project consists of the nine major perspective-hostellers, wardens, security, mess-representative, hostel representative, Assistant Warden, Student-Council, Care-Taker, Chief warden and Administration each having a separate login and profile view. The entire project consists of five use cases –

- **Hostellers Use Case:** To access the hosteller repository, to view the notice, draft a Grievance or complaint, provide thee rating to mess, access sickroom/common room, medicines and sport accessories.
- **Warden:** to verifies the entry and exit within hostels, access repositories, address grievance, draft a notice, allocates the room and take an action with respect to mess rating.
- **Security Use Case:** To verify the Medicine/sport accessories, to verify the entry and exit, to verify sick room, common room, visitor rooms and view notices.
- **Chief Wardens Use Case:** To check the room allocation, and have access of repository
- **Admin Use Case:** To manage the entire system, to verify the user, ensure the security and proper-functioning of the services provided by the system.
- **Care Taker Use Case:** to verifies the fees and allocation module, to choose representatives
- **Assistant Warden:** to verifies the entry and exit within hostels, access repositories, address grievance, draft a notice, allocates the room
- **Mess Representative Use Case:** To access the hosteller repository, to view the notice, draft a Grievance or complaint, provide thee rating to mess, access sickroom/common room, medicines and sport accessories, update-menu, view feedback.
- **Hostel Representative Use Case:** To access the hosteller repository, to view the notice, draft a Grievance or complaint, provide thee rating to mess, access sickroom/common room, medicines and sport accessories, view-emergency contacts, view room allocation.

### 3.4 OPERATING ENVIRONMENT

MEAN is a software bundle that stands for MongoDB, Express JS, AngularJS, and NodeJS. Together in a stack, these free programs enhance the simplicity of the web development process. MEAN basically describes an application stack:

- MongoDB (data storage)
- Express.js (server-side application framework)
- AngularJS (client-side application framework)
- Node.js (server-side language environment although Express implies Node.js)

Apart from this other application were being used in our project:

- Apache (web server)
- PHP Python (scripting language)

### 3.5 DESIGN AND IMPLEMENTATION CONSTRAINTS

All the five views used in this web-based module should have separate area of concerns. The repository should be secured and the server scripting should be designed to reduce the threads and vulnerability in the system. The web module should be responsive and scalable to handle large number of clients and to run on various platforms and devices.

### 3.6 ASSUMPTIONS AND DEPENDENCIES

As a client the following assumptions are needed to be fulfilled:

- The client must have the updated browsers Version 55.0.3497.100 or above in Chrome, Microsoft Edge 32.17124.1.0 or above, 55.0.2 or above in Firefox or Version 49.0.3051.100 or above in Opera in windows and equivalent versions across the other platform or updated android/iOS/Plasma/blueberry/windows OS phone.
- The Server must have accessibility to the Entire MEAN Stack software- Mongo-DB, node.js, angular.js and Express.js. It is assumed that PHP is installed with a driver for connecting Mongo-DB database. The Apache Server is Expected to be installed. The basic storage space of 1.5 Mb/user is expected to be provided.
- Our project consists of following other dependencies:
  - Way2sms for SMS provider for verification and contact messaging needs
  - Driver to connect PHP and Mongo-DB
  - make2pdf for converting text to pdf for downloading.

# Chapter-4

## IMPLEMENTATION STAGE-I: REQUIREMENT ANALYSIS

Entire Implementation Carried out in 5 stages:

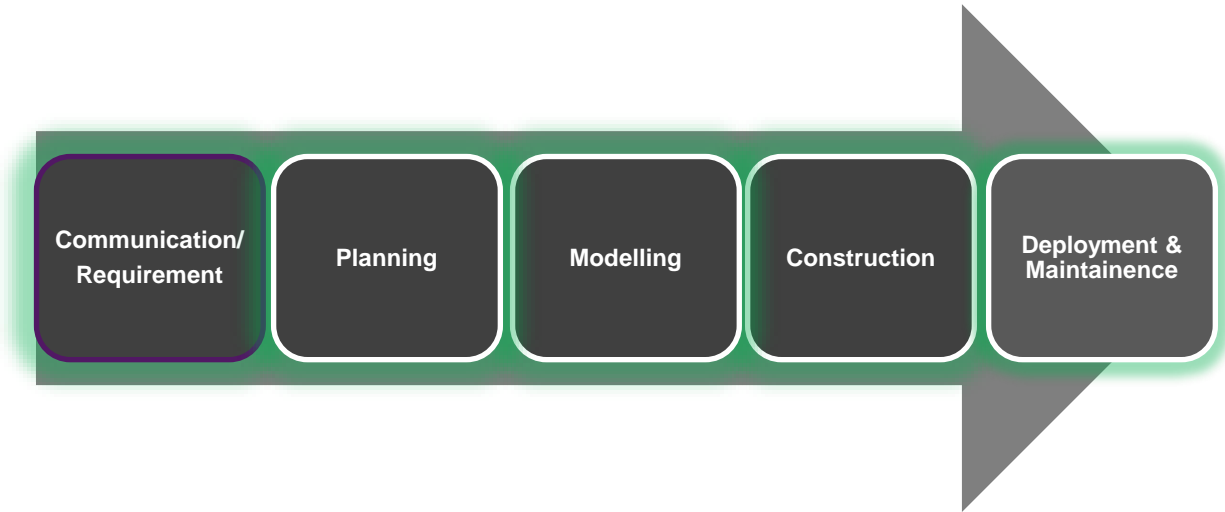


Fig: 4.1: Software Development phases

The first stage consists of development of SRS with various requirements of the future users and the various use-cases. Being a project on the online Hostel Management System for a autonomous Institute like, National Institute of Technology, Puducherry, Karaikal, the various use-cases involve like Wardens, Chief Warden and Care-Taker. Being the end Users as Students, the suggestions from various students were welcomed while making the project. With the consistent effort, after withdrawing the various functionalities, finally the SRS was developed.

### 4.1 SRS

An SRS is generated out of various requirements gathered. It is formed out of all the gathered requirement and consist of 10 pages initially made. It Consist of IEEE standardized format with following Main Topics covered:

- Introduction: Glimpse of the project



- Overall Description: The detail description of the use-cases and the project formed
- External Interface Requirements: The detail description of all the external interfaces, modules, frameworks, and dependencies
- System Features: All the features of the product being developed.
- Other Nonfunctional Requirements: All the non-functional requirement are listed include performance, security and various other functionalities.
- A Case Study of Present System in Appendix.

## 4.2 VARIOUS REQUIREMENTS GATHERED

The most commonly used features of a website for SA includes:

- **Additional Chief Warden:**

Develop a project with features which make allocation Easy. make a user-friendly Website, make a module for Mess-feedback.

- **Care-Taker:** Make the module for fees allocation, mess feedback and In/Out of Student and make them much more convenient.
- **Student:**

**Actor1:** Make a template Much More elegant than Before.

**Actor 2:** make the automation less manual and aesthetic looks

**Actor 3:** provide Emergency contacts for students and make it convenient for room mess and other complaints.

- **Chief Warden:** Make the module for searching the various sort of data associated with students and hostels, payment module for tracking checks(currently out-of-scope)
- **Warden (Girls):** Make a SMS service for faster delivery of Information
- **Warden(boys):** Make a system user friendly and reduce all the paper-work to a greater level, put the grievance module.
- **Hostel Representative:** A faster message accessing system with elegant and user-friendly interface.

# CHAPTER 5

## IMPLEMENTATION STAGE-II: THE PROJECT PLANNING

The project plan is necessary to define what the project is about and to align the expectations from all directly involved stakeholders, including the team-members, other teams, the principal, the coaches, the teaching team and outside resources. The plan is an articulation of the goals and the process to transform the goals into results. The project plan is a temporary agreement. The process is important only until the project is finished. The results will survive the project. The project plan is a living document. Reality cannot always be foreseen and the plan must be updated during the process to accommodate new circumstances.

### 5.1 PROJECT WORK FLOW

The second stage of implementation involved the building of the project plan involving the creation of deadline, a time-based development strategies for the better and convenient coding and better optimization in the future. The plan of action is as follows:

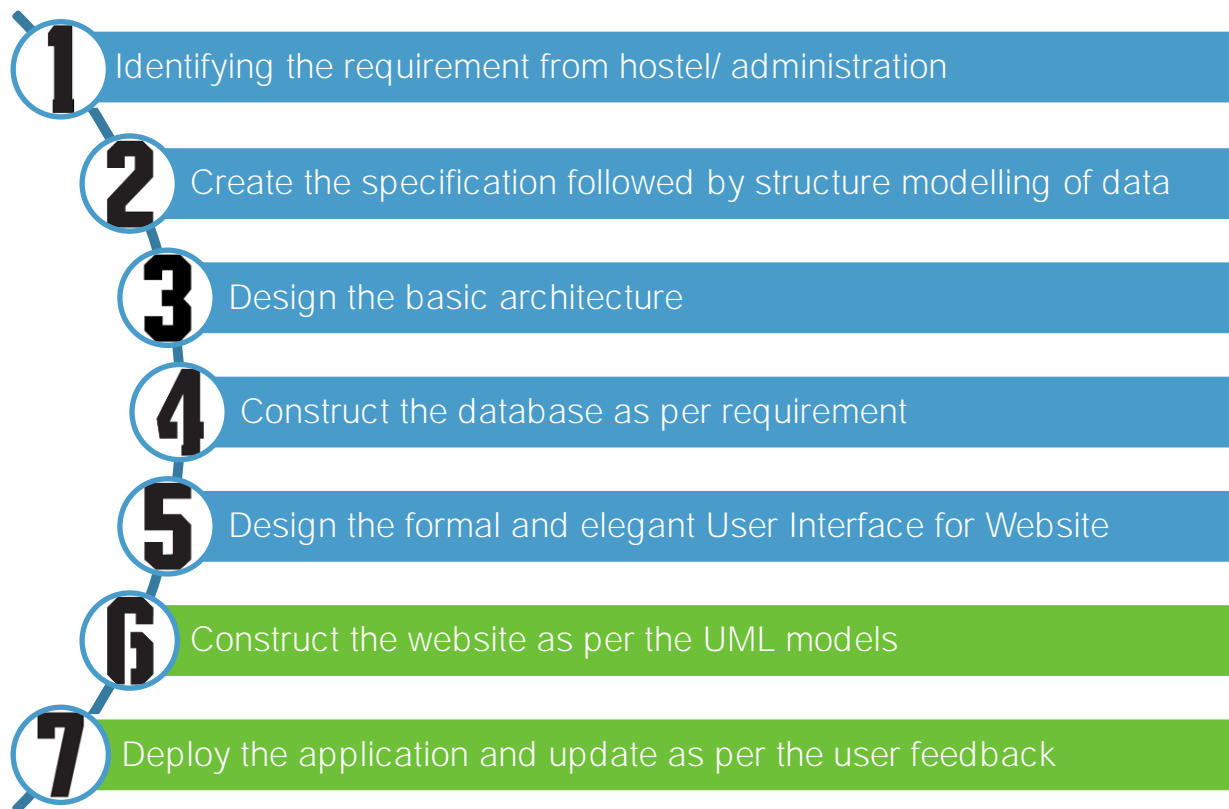
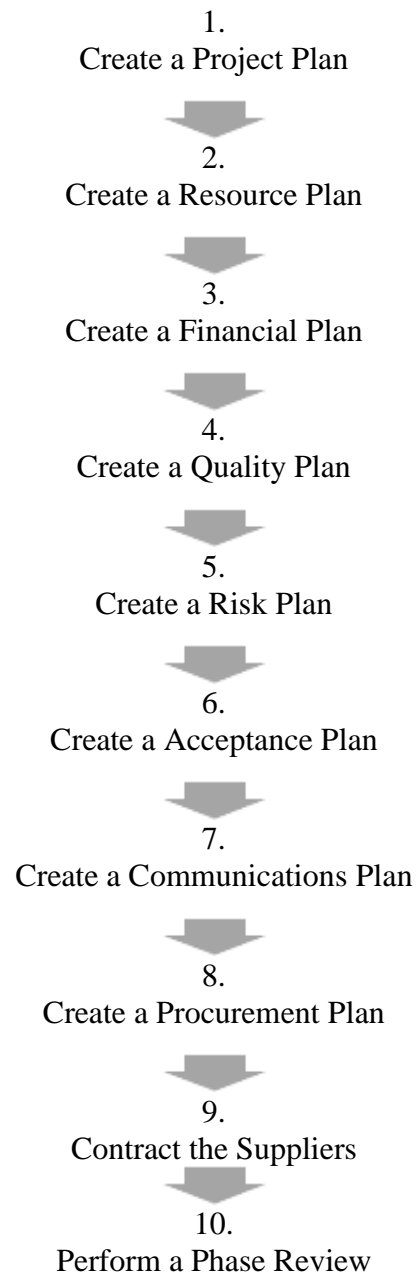


Fig: 5.1: Plan of Action

## 5.2 MAJOR STEPS FOR PLANNING



# CHAPTER 6

## IMPLEMENTATION STAGE-III: THE PROJECT

### MODELLING



Fig: 6.1: Software Development: Iterative Process

The system is designed with iterative modelling. Systems design is the process of defining the architecture, modules, interfaces, and data for a system to satisfy specified requirements. Systems design could be seen as the application of systems theory to product development. There is some overlap with the disciplines of systems analysis, systems architecture and systems engineering.

## 6.1 VARIOUS DESIGN CREATED

### 6.1.1 USE-CASE DIAGRAM

A **use case diagram** is a dynamic or behavior **diagram** in UML. **Use case diagrams** model the functionality of a system **using** actors and **use** cases. **Use** cases are a set of actions, services, and functions that the system needs to perform.

### 6.1.2 SEQUENCE DIAGRAM

A **sequence diagram** simply depicts interaction between objects in a sequential order i.e. the order in which these interactions take place. We can also use the terms event **diagrams** or event scenarios to refer to a **sequence diagram**. **Sequence diagrams** describe how and in what order the objects in a system function.

### 6.1.3.DATA FLOW DIAGRAM

A data-flow diagram is a way of representing a flow of a data of a process or a system The DFD also provides information about the outputs and inputs of each entity and the process itself. A data-flow diagram has no control flow, there are no decision rules and no loops.

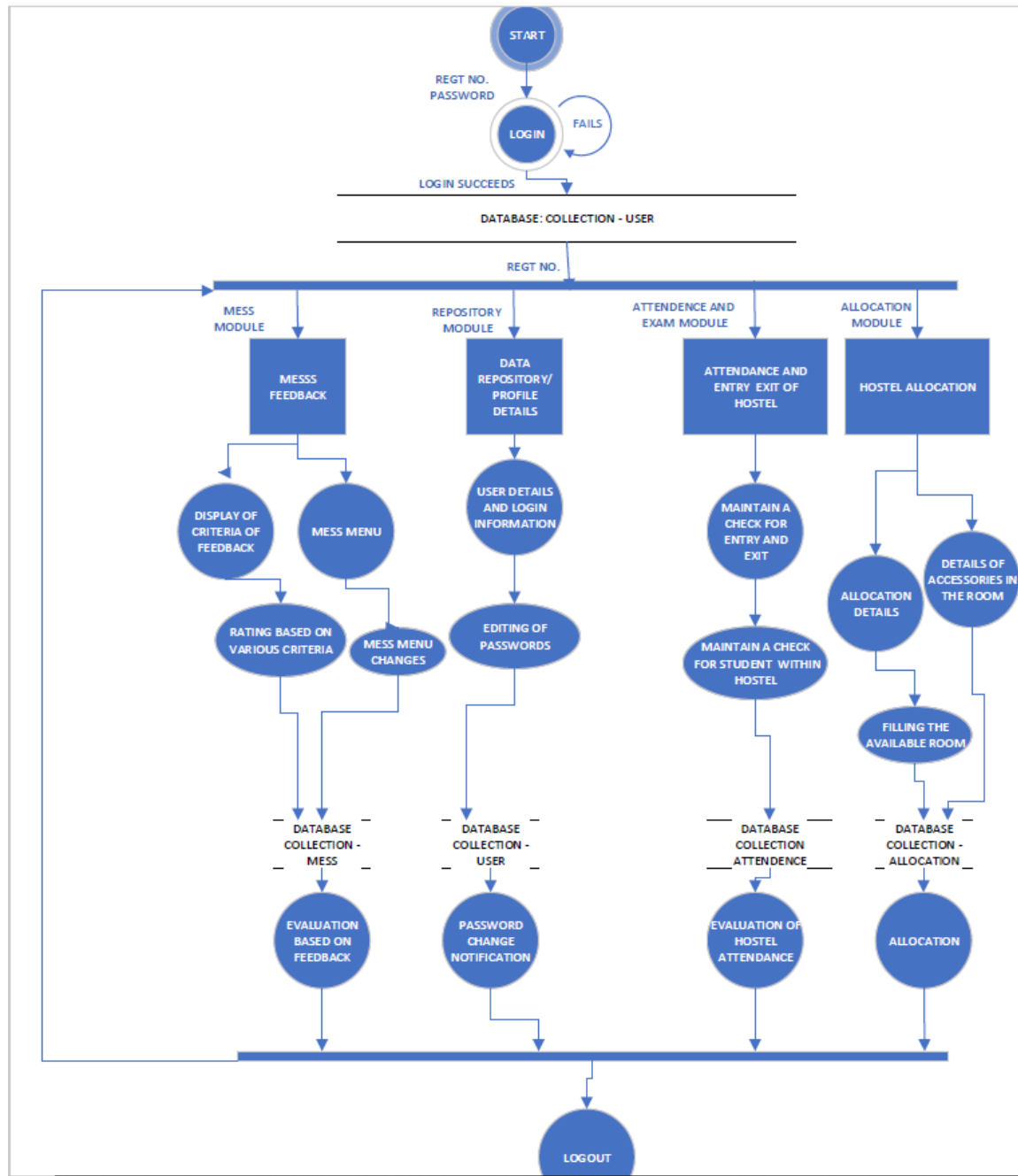


Fig: 6.2: Data Flow Diagram

### 6.1.4 ACTIVITY DIAGRAM

**Activity diagram** is another important **diagram** in UML to describe the dynamic aspects of the system. **Activity diagram** is basically a flowchart to represent the flow from one **activity** to another **activity**. The **activity** can be described as an operation of the system. The control flow is drawn from one operation to another.

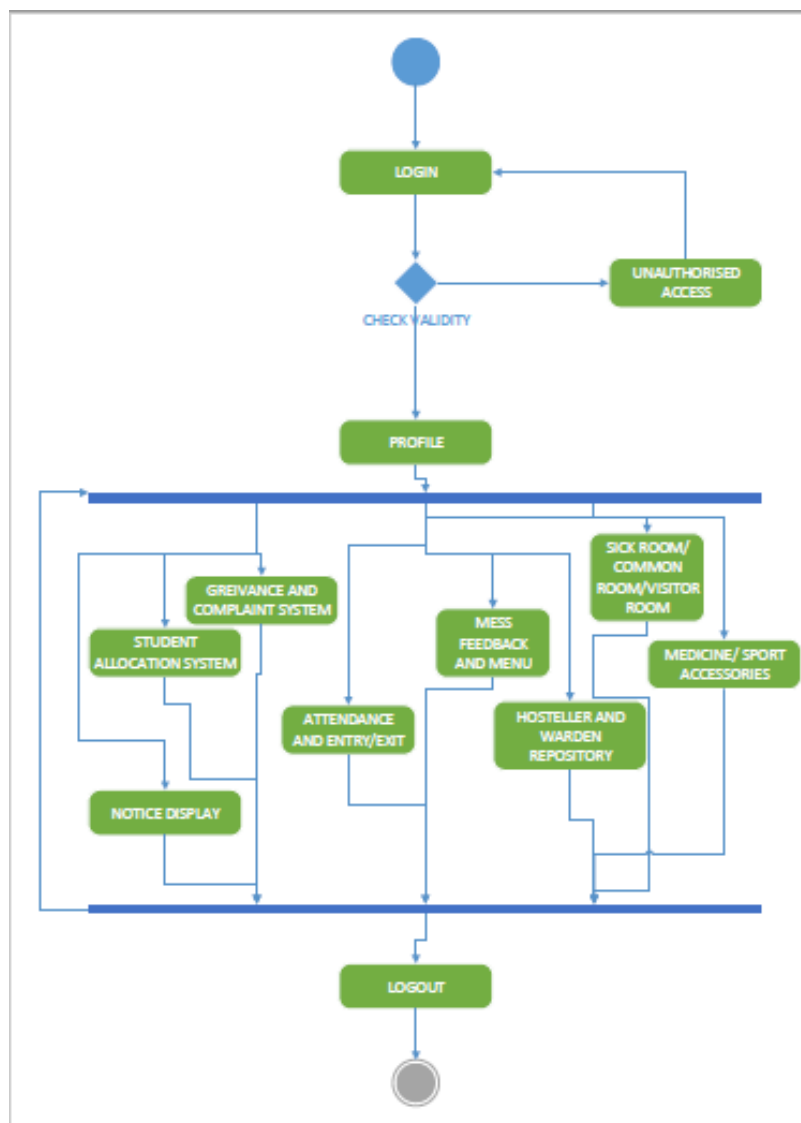


Fig: 6.3: Activity Diagram

# CHAPTER 7

## IMPLEMENTATION STAGE-IV: CONSTRUCTION

The fourth stage of implementation involved the creation of the project as per blue prints.

### 7.1 EXTERNAL REQUIREMENTS

#### 7.1.1 USER INTERFACES

The web-based module of Hostel Management System consists of a login screen which checks for validation of credentials of the user. Beyond this screen various modules are accessible to the user on a role-based privilege basis. A session is created as soon as the user logs in and it ends after the user logs out of his/her respective profile.

#### 7.1.2 SOFTWARE INTERFACES

The web-based module is built on MEAN Stack software- Mongo-DB, Node.js, Angular.js and Express.js. It is assumed that PHP is installed with a php-mongodb.dll driver for connecting Mongo-DB database. The Apache Server is expected to be installed. PHP driver extension has been included in the Apache as well as the PHP php.ini file. Composer has been installed and all PHP related dependencies are handled via Composer. Robo-3T Database Administration Software has been installed and management of the database is regulated using this software. Bootsparap.js and jquery.js files may be required for Responsiveness and advance JavaScript approaches

#### 7.1.3 COMMUNICATIONS INTERFACES

- A web-browser is required having *Version 55.0.3497.100 or above in Chrome, Microsoft Edge 32.17124.1.0 or above, 55.0.2 or above in Firefox or Version 49.0.3051.100 or above in Opera in windows and equivalent versions across the platform or updated android/iOS/Plasma /blueberry/windows OS phone.*
- *An SMTP server is required to be installed that takes care of the whole email delivery process.*
- *Reliable Web-server to handle the request*

## 7.2 NONFUNCTIONAL REQUIREMENTS

### 7.2.1 Performance Requirements

Since this software is going to web-based, it does require a reliable server machine with high band internet access. Server machine should have a powerful Processing Unit and high-speed internet access so that it can handle multiple users at the same time. Another performance requirement is the storage space. Higher storage space means more user information can be stored, leads to the better performance.

Performance requirement by the user side is, web application should be developed as a lightweight web app so that it can work on almost any platform even with slower internet connections. Refactoring of code in PHP and reduction in database entry size lead to overall optimization of the system leading to faster access, responsiveness and a scalable website which reduces the overall server access time and lead to faster server response

### 7.2.2 Security Requirements

The Web based application uses PHP Library's SHA-256 hashing protocol for the passwords of each login along with secure design patterns which tend to make more secure database access for this web module. The various in built php libraries and No-SQL features of MongoDB can reduce the various SQL attacks and various other database vulnerabilities to a very greater extent, hence providing a secured and reliable database which can withstand three objectives of Security of Web Applications: Confidentiality, Integrity and Availability for the Centralized Management System.

### 7.2.3 Software Quality Attributes

**Portability and Responsiveness:** Main purpose of developing web-based module is to improve the portability of software development process. To improve portability, software should run on variety of platforms and variety of connection speeds. As explained in the performance requirements section, software should be lightweight so that it can run on a machine with slow internet connection. To make the web application lightweight, simple libraries and tools should be used at developing phase. Such as using JavaScript and HTML5. This system can also run across various devices including mobile devices, cross platform web browsers due to the Bootstrapping features of the web modules in the projects.



## 7.3 CHALLENGES

### 7.3.1 CHALLENGE OF PERFORMANCE

The demand of highly **Flexible, Responsive** and **Scalable** website is the need of the need today. With emerging number of hosteller and expectation of increasing the number of students in the mere future leads to an alarming need of a scalable website with high performance. For cross-platform client supportability a highly responsive website is required leading to a high need of performance in the Real-time.

### 7.3.2 CHALLENGE OF SECURITY

Security is the protection of computer systems and the data accessed or stored in it from theft or damage to their hardware, software or electronic data, as well as from disruption or misdirection of the services they provide. In today's World, there is an alarming need of Security. Security of data now being a major concern around the world. The information of each of the students, faculties and admin are stored in the database are need to be preserved form the crackers and hackers for the misuse of data and creating the inconsistency of functionalities basically intended to derive from the module.

### 7.3.3 CHALLENGE OF QUALITY

Main purpose of developing web-based module is to improve the **portability** of software development process. To improve portability, software should run on variety of platforms and variety of connection speeds. The written code must be **Reusable**, if incase addition of functionalities or use-cases, if required in the future. The code must be refactored and optimized for better performance in the real-time execution. The code must support the required functionalities, reliability and maintainability for better Quality which is a major challenge.

## 7.4 OPTIMIZATION

### 7.4.1 Avoiding Relative Path In File Inclusion

Generally a `WEB_ROOT` is declared globally providing an absolute path resolution by PHP.

### 7.4.2 Releasing all Resources

Unsetting large arrays, closing a file handler, closing a database connection or a curl request to any other server through an API are amongst the various other things that developers usually overlook. As a result, more memory is occupied in maintaining the resources. It is always better to close a resource after it's opened and our work is done. We can also use destructors in our classes as we know when the program controller exits a class. If a destructor is present, then it would automatically call the garbage collector.

### 7.4.3 Avoid Unnecessary Use of Global Variables

Accessing a global variable is twice as costly as accessing a local variable, so if a global is avoided, it reduce a considerable amount of overhead.

### 7.4.4 No use of `count()` in The Condition Section of a Loop

For example: `for ($x = 0; $x < count($array); $x++)`, in this statement the `count` function is called each time the loop iterates and it does the same thing over and over again. Instead if we declare a variable say `$count = count($array)`; and then `for ($x = 0; $x < $count; $x++)`, we can save processing time.

### 7.4.5 Using `ISSET`

It is safer to use `isset()` Instead of using `count()`, `strlen()`, `sizeof()` as it is better as well as compare the value which is returned with a specific number. Likewise `(strlen($foo) < 50)` can be written as `(!isset($foo{50}))`.

### 7.4.6 Using more of Static Methods/Properties

Static methods/ properties are generally faster than non-static ones. However, if an object is pre-initialized and then a non-static method/property is called repeatedly on the same object, non-static is faster.

Non-static methods skip a step of object initialization making it faster by 20% on an average.

### 7.4.7 `foreach` loops

According to the benchmarks, the `foreach` loop is much faster than the `for` loop and `for` loop is faster than the `while` loop.

#### 7.4.8 Pre and Post Increments

This is again a widely debated topic. It behaves differently in different languages and depends upon the environment. Post-incrementation in PHP has been documented as storing a temporary variable which attributes to this 10% overhead vs. pre-incrementation.

`++$i` is pre-incrementation, `$i` is incremented and the new value is returned

`$i++` is post-incrementation, the value of `$i` copied to an internal temporary variable, `$i` is incremented and the internal copy of the old value of `$i` is returned.

#### 7.4.9 Echo, print, comma, printf, and sprintf

Echo is a language construct whereas print is a function, so echo is faster than print,

`echo $a, $b, $c;` is faster than `echo $a . $b . $c;`, but `echo $a . $b . $c;` is faster than `print $a . $b . $c.`

If the statement is `$query = sprintf( "SELECT * FROM users WHERE username='%s' AND password='%s'", $_user, $_password );`

Using `sprintf` is found to be 10 times faster than using variables within double quotes in string formatting. `printf` outputs the string, whereas `sprintf` stores the string in a variable.

#### 7.4.10 Using Identical Operator (===) Anywhere Possible

The equality operator `==` typecasts or converts the data type temporarily to see if it's equal to the other operand whereas `===` (identity operator) doesn't need to do any converting whatsoever and thus, it does lesser work, making it comparatively fast.

## 7.5 TESTING

### 7.5.1 UNIT TESTING

Unit Testing is a level of software testing where individual units/ components/modules of a software are tested. The purpose is to validate that each unit of the software performs as designed. A unit is the smallest testable part of any software. It usually has one or a few inputs and usually a single output. In procedural programming, a unit may be an individual program, function, procedure, etc. In object-oriented programming, the smallest unit is a method, which may belong to a base/ super class, abstract class or derived/ child class. (Some treat a module of an application as a unit. This is to be discouraged as there will probably be many individual units within that module.) Unit testing frameworks, drivers, stubs, and mock/ fake objects are used to assist in unit testing.

<u>Unit Testing</u>	A level of the software testing process where individual modules of a software are tested. The purpose is to validate that each unit of the software performs as designed. After checking each and every module of the Software various flaws are found.
<u>Integration Testing</u>	A level of the software testing process where individual units are combined and tested as a group. The purpose of this level of testing is to expose faults in the interaction between integrated units. Here after combining it is found software it is highly coupled and the coupling is reduced by changing variables in database
<u>System Testing</u>	A level of the software testing process where a complete, integrated system is tested. Various Test-cases are found by certain actors implementing various inputs and results are recorded.
<u>Acceptance Testing</u>	A level of the software testing process where a system is tested for acceptability. The purpose of this test is to evaluate the system's compliance with the business requirements and checked if it can be delivered. It is the alpha test performed by the user.

Table 7.1: Testing Stages

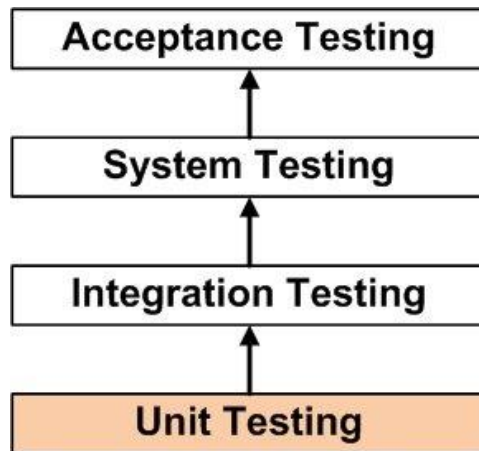


Fig: 7.1: Unit Testing Stack

After Unit Testing various sort of functional errors, errors in module cross-functioning, low cohesiveness and high coupling effect were detected which in turn raises various error and omissions in the project, it leads to various optimization, corrections in the project of hostel Management. The various optimizations were mentioned above.

### 7.5.2 USE-CASE TESTING

Use Case Testing is a software testing technique, which helps identify test cases that could cover the entire system, query after query. Initially there were 5 use cases, scaled up-to 9 after the use case testing. This is basically who all can access which type of functionalities in the project system. A special use-case of student body in the system to view-over-all functioning of hostel.

# CHAPTER 8

## EXPERIMENTAL RESULTS

This chapter included the various screenshots of the Online Hostel Project Management

### 8.1 ONLINE HOSTEL PROJECT MANAGEMENT SYSTEM'S SCREENSHOTS

This section contains the various screen shots of the Online Hostel Project Management

HMS - Hostel Management System  
National Institute of Technology Puducherry

[View Gallery](#)

User Name  
Username

Password  
Password

☐ Save Credentials

[Submit](#)

[Forgot Password? Click Here](#)  
[New Student? Register Here!](#)  
[Admin Signup ? Click Here](#)

Copyrights © 2019 | Shivam Giri and Rohit Baraik | NIT Puducherry

Fig:8.1: Index page



Fig: 8.2: Gallery

Country

India

Pincode

**PASSWORD SECTION**

Enter Password


Confirm Password

Upload a Identity proof driving licence/ Adhaar card/Passport or 10/12th certificate:  N...

Upload a Photograph:  N...


☐ I hereby declare that above Information are best of my knowledge and Wrong Information can cause severe action and even results to non- allocation in the hostel

Fig: 8.3: Registration form



ONLINE HOSTEL MANAGEMENT SYSTEM

NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

 SHIVAM GIRI

- Student Details
- Going for Holiday
- Mess Feedback
- Notice Section
- Room Complaints
- Grievance Cell
- Log Out

NOTICE BOARD

Update of Cricket

NOTICE00000002


Sunday 21-04-2019

Mr. Consistent with his accuracy, great human Being. Stories of Glenn McGrath on Castrol Activ Champions 13.3 Narine to Williamson, no run, slides back and tucks it off the pads to find 13.2 Narine to Williamson, no run, plenty of turn, the batter misses the tuck and the appeal is turned down as it was going down 13.1 Narine to Bairstow, 1 run, clipped away off the pads for a single 13 Runs Scored: 8 1 W 0 2 Wd 3 1 Score after 13 overs SRH 138-1 Bairstow 60(37) Williamson 5(3) Yarra Prithviraj 3-0-29-1 12.6 Yarra Prithviraj to Bairstow, 1 run, slower delivery angling across, dabbed away to deep third man for a single 12.5 Yarra Prithviraj to Williamson, 3 runs, stands up tall and punches it past cover for a triple. That was a length ball and Williamson stood up tall to time it away. Russell did well to cut it off 12.5 Yarra Prithviraj to Williamson, wide, a little harsh? Short ball, going through to the keeper, Williamson let it pass. Was over the head and was wide 12.4 Yarra Prithviraj to Williamson, 2 runs, back-footed punch and it threads the gap between the cover fielders for a brace 12.3 Yarra Prithviraj to Williamson, no run, back of a length delivery and has been punched off the back foot to point Kane Williamson, right handed bat, comes to the crease

Contact: Majhi 9531809590


Uploaded By: SHIVAM GIRI Warden W1900001

Fig: 8.4: Dashboard of Student(View Notice)



ONLINE HOSTEL MANAGEMENT SYSTEM

NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

 SHIVAM GIRI

- Student Details
- Going for Holiday
- Mess Feedback
- Notice Section
- Room Complaints
- Enter Complaints
- Grievance Cell
- Log Out

UPDATE COMPLAINT

Enter Complaint Title

Enter complaint Title


Enter Complaint Content

Enter Complaint Body

Submit

Fig: 8.5: Update Complaint




**ONLINE HOSTEL MANAGEMENT SYSTEM**  
 NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

SHIVAM GIRI

- Student Details
- Going for Holiday
- Mess Feedback
- Give Feedback
- View Menu
- Notice Section
- Room Complaints
- Grievance Cell
- Log Out

RATE YOUR CATERER


FEED-BACK DEACTIVATED

Excellent 5 Points  
 Average 2 Points  
 Poor 1 Points

Very Good 4 Points  
 Good 3 Points  
 Very Poor 0 Points

SNo.	Feedback Category	Points(0-5)
1.	Quality of Food Served	<input type="text"/>
2.	Taste of Food	<input type="text"/>
3.	Cleanliness of Dining and Food Counter Area	<input type="text"/>
4.	Cleanliness of Wash Basins	<input type="text"/>
5.	Cleanliness of Catering Staff	<input type="text"/>

Fig: 8.6: Give- Feedback


**ONLINE HOSTEL MANAGEMENT SYSTEM**  
 NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

SHIVAM GIRI

- Student Details
- Going for Holiday
- New Leave Form
- Travel Within Karaikal
- View Leave status
- Leave register
- Mess Feedback
- Notice Section
- Room Complaints
- Grievance Cell
- Log Out

LEAVE FORM

Name of Applicant	SHIVAM GIRI			
Course				
Roll. No.	CS15B1022			
Year	4th			
Room No.	F38			
Mobile No.	09531809590			
Date of filling the form	22-04-2019			
No. of Days Leave	<input type="text"/>			
Purpose of leave	<input type="text"/>			
Places of Visit with full address	<input type="text"/>			
Travel Plan	Out Date	Out Time	Return Date	Return Time
	22/04/2019	At Exit from Hostel		At time of Return

No  
Student Verified (at Submission)

No  
Warden Verified

No  
Security Verified (at time of Exit)

Submit Info

Fig: 8.7: Student Leave Form

MAKE ENTRY

**Roll Number: CS15B1022**

Enter Place and Reason of Visit

Enter Reason of Visit

Enter Out Date

22/04/2019


Enter Expected Out Time

13:04

Submit

Fig: 8.8: In/Out Karaikal (Local Entry)

## SECURITY


**ONLINE HOSTEL MANAGEMENT SYSTEM**  
 NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

Boys Hostel - II

Allocate Room  
 Gym In/Out  
**Make Entry**  
 Outstanding Entry  
 General Register  
 Health-Care Room  
 Common Room  
 Guest Room  
 Leave Register  
 In/Out Register  
 Log Out

MAKE ENTRY

Enter Student Name

Enter Student Name

Enter Roll Number

Enter Student Roll Number

Enter Out Time

Enter Out Time

Room Selected

Gym Room

Submit

www.nitpy.ac.in

Fig: 8.9: Gym (Make Entry)

ONLINE HOSTEL MANAGEMENT SYSTEM NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY									
Boys Hostel - II									
Allocate Room	Id	Student Name	Roll Number	Phone Number	Room Number	Out Time	Out Date	In Time	In Date
Leave Register	IO1	SHIVAM GIRI	CS15B1022	09531809590	F38	04:14 PM	2019-04-19	04:50 PM	19 April 2019
In/Out Register	IO3	SHIVAM GIRI	CS15B1022	09531809590	F38		2019-04-21		2019-04-21
Out Entry									
Outstanding Entry									
General Register									
Log Out									


Fig. 8.10: General Registry of In/Out

Health-Care Room	New Entry Added!
Common Room	MAKE ENTRY
Make Entry	Enter Student Name
Outstanding Entry	Enter Student Name
General Register	Enter Roll Number
Guest Room	Enter Student Roll Number
Leave Register	Enter Out Time
In/Out Register	Enter Out Time
Log Out	Room Selected
	Common Room
	Submit

Fig. 8.11: New Entry added in Common Room

WARDEN





ONLINE HOSTEL MANAGEMENT SYSTEM

NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

SHIVAM GIRI

Personal Information

Allocation & Room

Academics

Leave and Attendance

Fees Details

Hostel Feedback

Room Complaints

Emergency Contacts

Views Contacts

Insert Contacts

Update Contacts

General Contacts

Notice Section

Grievance Section

Log Out

INSERT EMERGENCY CONTACT

Enter Contact Name

Enter Name

Enter Designation, Occupation

Enter Designation

Enter Contact Number

Enter Contact Number

Enter Contact of the person

Enter Email Id.:(if any)

Remark/ Other Infos

Any other information(optional)

Submit

Fig: 8.14: Insert Emergency Contact

Personal Information

Allocation & Room

Academics

Leave and Attendance

Fees Details

Hostel Feedback

Room Complaints

Emergency Contacts

General Contacts

Notice Section

Update Notice

View Notice

Grievance Section

Log Out

UPDATE NOTICE

Enter Notice Title

Enter Notice Title

Enter Notice Content

Enter Notice Body

Enter Contact person if Any with name

Enter any contact Person Name:(if any)

Enter Contact of the person

Enter any contact Number:(if any)

Fig: 8:15: Update Notice

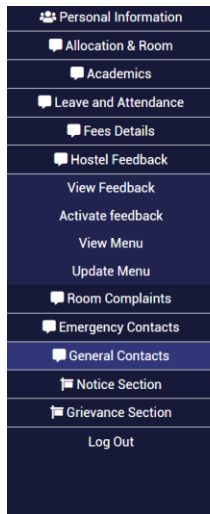

A white dialog box with a black border. At the top, a black button with white text says "ACTIVATE THE FEEDBACK". Below it, the text "Choose date and time for closing the feedback :" is followed by two input fields: "22:00" and "22/04/2019". A blue button with white text says "Activate Feedback". Below the button, the text "Feedback is activated" is displayed in a large font, followed by "Feedback closes at: 0d 8h 11m 18s" in a smaller font.

Fig: 8.16: Activated Feedback

The interface shows the "ONLINE HOSTEL MANAGEMENT SYSTEM" header with the NIT Puducherry logo and the user name "SHIVAM GIRI". The left sidebar menu is the same as in Fig 8.16. The main content area has a white background and a light gray border. It contains a form with the following fields: "Select Registration Number(\* only allocated and verified students are displayed)" with a dropdown menu, "Enter Mode of Payment" with a dropdown menu, "Sem:" with a text input field containing "Even(January-May)", "Year:" with a text input field containing "2019", and "Enter Payment Reference Number/ Transactional Number" with three stacked text input fields: "Enter Transasactional Number", "Enter Transasactional Number if paid in second installment", and "Enter Transasactional Number of fine".

Fig: 8.17: Insert fees Details


**ONLINE HOSTEL MANAGEMENT SYSTEM**  
 NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY


SHIVAM GIRI

- Personal Information
- Allocation & Room
- Academics
- Leave and Attendance
- Fees Details
  - Insert Fees Details
  - Fees Not Paid List
  - View Fees Paid
  - Reset Fees for Next Semester
- Hostel Feedback
- Room Complaints
- Emergency Contacts
- General Contacts
- Notice Section
- Grievance Section
- Log Out

### FEES DETAILS FOR CSE of 4th in B.-Tech

Name of student	Roll Number	Paid	Mode of Payment	Amount	Paid on	Ref_id	Ref_id1	Ref_id2
ROHIT KUMAR BARAIK	CS15B1018	Yes	Demand_Draft	234567	2019-04-10	REF2017	-	-
SHIVAM GIRI	CS19B1099	No					-	-

Fig 8.18: Fee Details


**ONLINE HOSTEL MANAGEMENT SYSTEM**  
 NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY


SHIVAM GIRI

- Personal Information
- Allocation & Room
- Academics
- Leave and Attendance
  - Attendance Graphical View
  - Leave Grant
  - Grant Out for Karaikal
  - View Hosteller-On leave
  - View Hosteller out-in Karaikal
  - View Present Attendance
- Fees Details
- Hostel Feedback
- Room Complaints
- Emergency Contacts
- General Contacts
- Notice Section
- Grievance Section
- Log Out

### ATTENDANCE FOR BOYS HOSTEL -II ON 2019-04-22 AT 01:50:28 PM

Name of student	Roll Number	Course	Room No.	Year	Phone No.	Is Present
SHIVAM GIRI	CS15B1022	B.-Tech	F38	4th	09531809590	Yes
ROHIT KUMAR BARAIK	CS15B1018	B.-Tech	F39	4th	09110013484	Yes
SHIVAM GIRI	CS19B1099	B.-Tech	F38	4th	9434343434	Yes

Fig 8.19: Live Attendance



ONLINE HOSTEL MANAGEMENT SYSTEM

NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

SHIVAM GIRI

Personal Information

Allocation & Room

Academics

Insert Hostel

Insert Courses

Insert Department

Roll No. Mapping

Verify Passout(+4 years Students)

Update Master Password

Update Student's Year Explicitly

Allot Representative

Deallot Representative

View Representative

Leave and Attendance

Fees Details

Hostel Feedback

Room Complaints

Emergency Contacts

CHOOSE THE CREDENTIALS

Representative Specialization

☐ Mess Representative
 ☐ Hostel Representative
 ☐ Student Council member

Submit Course Info

Fig: 8:20: Allot the representative



ONLINE HOSTEL MANAGEMENT SYSTEM

NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

SHIVAM GIRI

Personal Information

Allocation & Room

Academics

Insert Hostel

Insert Courses

Insert Department

Roll No. Mapping

Verify Passout(+4 years Students)

Update Master Password

Update Student's Year Explicitly

Allot Representative

Deallot Representative

View Representative

Leave and Attendance

Fees Details

Hostel Feedback

Room Complaints


Emergency Contacts

VIEW REPRESENTATIVES

Name of student	Roll Number	Hostel	Room No.	Year	Phone No.	Present/Absent
ROHIT KUMAR BARAIK	CS15B1018	F39	Boys Hostel -II	4th	09110013484	Student Council
SHIVAM GIRI	CS19B1099	F38	Boys Hostel -II	4th	9434343434	Hostel Representative

Fig: 8:21: View Representatives




**ONLINE HOSTEL MANAGEMENT SYSTEM**  
 NATIONAL INSTITUTE OF TECHNOLOGY PUDUCHERRY

SHIVAM GIRI

- Personal Information
- Allocation & Room
  - Allocate Room
  - Deactivate a Student
  - Reallocate Rooms
  - Room accessories Details
  - Allocation Graphical View
  - Room Details
  - Search Student
- Academics
- Leave and Attendance
  - Fees Details
  - Hostel Feedback
  - Room Complaints
  - Emergency Contacts
  - General Contacts
  - Notice Section
  - Grievance Section

Room Allocation Information:F38

See Attendance Matrix

Name of student	Roll Number	Attendance	View Profile
SHIVAM GIRI	CS15B1022	Present	View Profile
SHIVAM GIRI	CS19B1099	Present	View Profile

Room Accessories Information

Cots

NITPY/BH/COT/2018/

Chair

NITPY/BH/CHR/2018/

NITPY/BH/CHR/2018/

Table

NITPY/BH/RTBL/2018/

NITPY/BH/RTBL/2018/

NITPY/BH/RTBL/2018/

Fig: 8:22: View Room Information



राष्ट्रीय प्रौद्योगिकी संस्थान पुदुचेरी, कारैक्कल  
 தேசிய தொழில்நுட்பக் கழகம் புதுச்சேரி, காரைக்கால்  
**NATIONAL INSTITUTE OF TECHNOLOGY, PUDUCHERRY**

Dear SHIVAM GIRI,

Greetings from NITPy Online Hostel management!!!  
 We have recieved your request for changing the NITPy Hostel account password.

Click on the below link on forget password your account.

Change Password

Do visit our institute website <http://www.nitpy.ac.in> for more information. Mail : [nitpyhostel@gmail.com](mailto:nitpyhostel@gmail.com)  
 We hope to serve you in the best way and provide you with a great learning experience.

Regards,  
 Hostel Management System.

Fig: 8:22: Mail for Forget Password

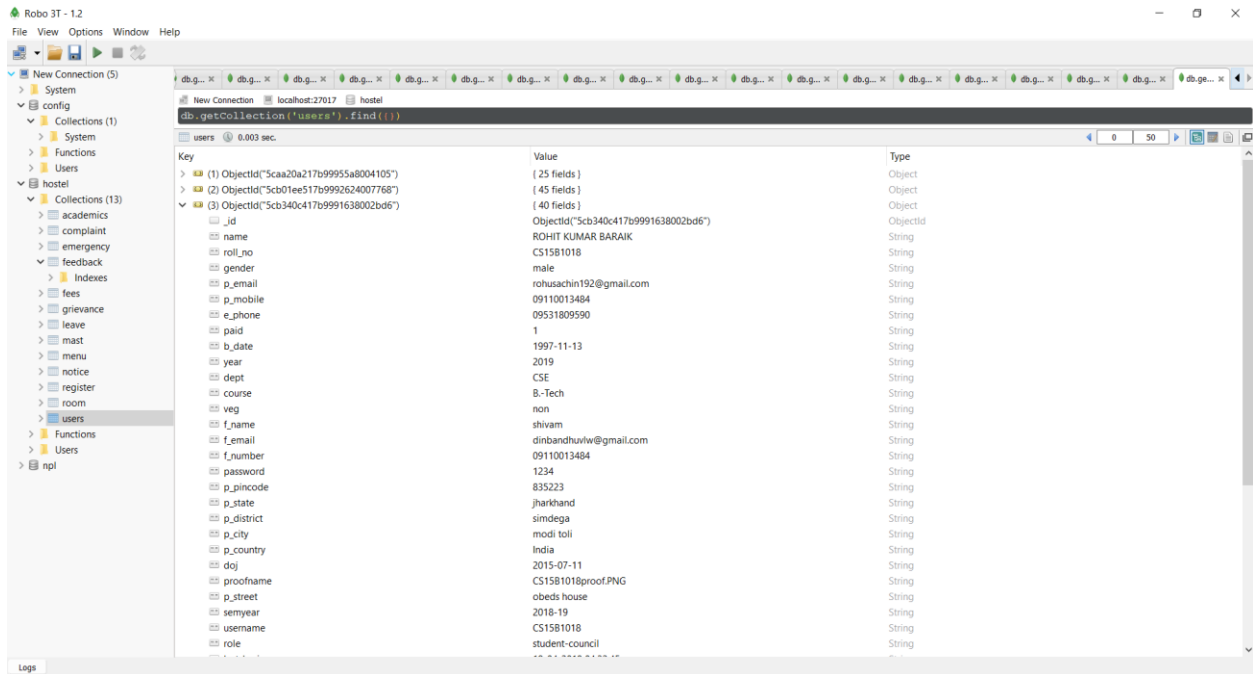


Fig: 8:23: MongoDB Entry

## 8.2 LIMITATIONS OF PROJECT

- 10<sup>th</sup> June of every Year Marked as a date of reference for automatic upgradation of the year of student, if the semester ended after that it may cause inconsistency.
- Although a measure is taken for-year-back students and dismissed students, the process is kept manual for year upgradation.
- Room allocation is manually carried-out for each student
- It can accommodate any course of maximum 8 year duration.
- Non-Pass-out students staying in hostel should be upgraded manually.
- Security system need to upgraded Technically.
- The Future architecture of the Hostels remain same.

### **8.3 ADVANTAGES OF THE PROJECT SYSTEM**

- Can view Allocation and Attendance Matrix of a entire floor in the hostel
- Ease of leave from hostel
- Ease of fees registration, allocation, re-allocation and deallocation.
- Ease of Upgrade of menu.
- A superior feedback system, with csv output representation.
- Attendance at a particular Moment.
- Grievance and complaint registration simplified
- Ease of Management of Student with faster execution from MongoDB.
- Proper partitioning of the functionalities among the Use-cases.

# **CHAPTER 9**

## **SUMMARY AND CONCLUSION**

### **9.1 SUMMARY**

All the above mentions points summarized as the various steps were taken for project Management, project is iteratively constructed. Firstly, the requirement is taken and a SRS is Developed which contained the details of all the functionalities, use-cases and the detailed requirement for the project. It is followed by the planning where the type of development strategy is chosen and the deadline and mile-stones are determined, the works were divided and there was a division of the time required on each mile-stone. Then came modelling where the project is modelled with data-flow diagram, Activity diagram, use-case diagram and sequence diagram. After this step follows construction and testing. During construction as per the various mile-stone, we developed various module sequentially with all functional requirements in SRS it is followed by various optimization show above. The issue of security is taken into account with high priority. We performed unit-testing, followed by the test-stacks and use-case testing of the project, where various uses cases are resolved. Finally the product is bug-free and made as per the various requirement of the users with elegant UI and other functional requirments. Finally, it is fully ready for deployment.

### **9.2 CONCLUSIONS**

Based on the various advantages and disadvantages of the project, we can conclude the project is not fully automated, but can ease the management of the Hostel to a greater degree from the case study of the project system. The paper-based work like feedback, in/out forms and complaint registry is being reduced to a greater extent. The functionalities like allocation, feedback, leave out registry, room registry and fees details is being implemented and tested successfully and the product is scalable, hence the product is fully determined and ought to work foreseeable future.

The goal of this project is to develop a responsive, scalable, efficient and user- friendly website for managing all the hostels in National Institute of Technology, Puducherry. We went through the various recent technologies and recent trend in Web Development and chooses the best which meet the Requirement. Finally, we decided to develop the website with less overhead and a scalable and reliable Website through MEAN Stack.

### **9.3 DIRECTION FOR FUTURE WORK**

The following are the areas of future work in this project:

- Adding a new module for much more convenience of user/hostellers.
- Adding a module for changes in administrative rules and regulations

### **9.4 NEWER IDEAS CAN BE IMPLEMENTED**

The following are the ideas can be implemented in future work in this project:

- Creating formal emailing system with NitPY header banner
- Biometric for In/Out
- Making a better JavaScript for better aesthetic look and UI.
- Making a system for communication if the student fails to arrive on time.
- New forum can be created in the hostel
- Social Login system
- Creating a hostel Instagram and Facebook account
- Making the system more Mobile-friendly, creating a mobile Application of the same

# CHAPTER 10

## REFERENCES

[1] I. K. Chaniotis, K.-I. D. Kyriakou, and N. D. Tselikas, "Is Node.js a viable option for building modern web applications? A performance evaluation study," *Computing*, vol. 97, no. 10, pp. 1023–1044, Mar. 2014.

[2] S. Tilkov and S. Vinoski, "Node.js: Using JavaScript to Build High-Performance Network Programs," *IEEE Internet Computing*, vol. 14, no. 6, pp. 80–83, Nov. 2010.

[3] W. Cui, L. Huang, L. Liang and J. Li, "The Research of PHP Development Framework Based on MVC Pattern," *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*, Seoul, 2009, pp. 947-949. doi: 10.1109/ICCIT.2009.130

[4] MongoDB Manual 4.0: MongoDB CRUD Operations (2019), <https://docs.mongodb.com/manual/>

[5] PHP: Mongo Manual: Php to Mongo Manual <http://php.net/manual/en/mongo.sqltomongo.php>

[6] MongoDB From Wikipedia, the free encyclopedia (2019) <https://en.wikipedia.org/wiki/MongoDB>

[7] Kozielski, Stanisław; Mrozek, Dariusz; Kasprowski, Paweł, Małysiak-Mrozek, Bożena; Kostrzewa, Daniel: *Beyond Databases, Architectures and Structures Volume 521 || Performance Aspects of Migrating a Web Application from a Relational to a NoSQL Databases*, ISBN: 978-3-319-18421-0, 978-3-319-18422-7, year: 2015, volume 10.1007/978-3-319-18422-7, Chapter 9, page: 107—115;

[8] Fonseca, P; Cheng Li; Singhal, V; Rodrigues, R: *2010 IEEE/IFIP International Conference on Dependable Systems & Networks (DSN) - A study of the internal and external effects of concurrency bugs* ISBN: 978-1-4244-7500-1, year = 2010, page: 221-230,

[9] Liu, Yimeng; Wang, Yizhi; Jin, Yi: *Research on the improvement of MongoDB Auto-Sharding in cloud environment*, ISBN: 978-1-4673-0242-5, year: 2012, page: 851—854

[10] SILBERSCHATZ: *Database System Concepts: MCGRAW-HILL EDUCATION: Part 4: Transaction Management*, 2016. Print.

[11] S. M. Zhao, X. L. Xia, and J. J. Le, "A Real-Time Web Application Solution Based on Node.js and WebSocket," *Advanced Materials Research*, vol. 816–817, pp. 1111–1115, Sep. 2013.

**[12]** J. de Leeuw, "Server-Side Statistics Scripting in PHP," *Journal of Statistical Software*, vol. 2, no. 1, 1997.

**[13]** M. Backes, K. Rieck, M. Skoruppa, B. Stock and F. Yamaguchi, "Efficient and Flexible Discovery of PHP Application Vulnerabilities," 2017 IEEE European Symposium on Security and Privacy (EuroS&P), Paris, 2017, pp. 334-349.  
doi: 10.1109/EuroSP.2017.14

**[14]** Dickey, Jeff. *Write Modern Web Apps with the Mean Stack: Mongo, Express, Angularjs, and Node.js*. San Francisco, CA: Peachpit Press, 2015. Print.

# CHAPTER 11

## APPENDIX

### Appendix A: Index

<i>Sl No.</i>	<i>Term</i>	<i>Explanation</i>	<i>Indexed</i>
1)	<b>Responsive web</b>	<i>Responsive web design is an approach to web design that makes web pages render well on a variety of devices and window or screen sizes.</i>	2.5, 5.3, 5.1
2)	<b>SHA-256</b>	<i>The SHA (Secure Hash Algorithm) is one of a number of cryptographic hash functions. A cryptographic hash is like a signature for a text or a data file. SHA-256 algorithm generates an almost-unique, fixed size 256-bit (32-byte) hash.</i>	5.2
3)	<b>Refactoring</b>	<i>Refactoring is the process of changing a software system in such a way that it does not alter the external behavior of the code yet improves and optimizes its internal structure</i>	5.1
4)	<b>No-SQL</b>	<i>A NoSQL (originally referring to "not only SQL" or "non-relational") database provides a mechanism for storage and retrieval of data that is modeled in means other than the tabular relations used in relational databases.</i>	5.2
5)	<b>Portability</b>	<i>Portability in high-level computer programming is the usability of the same software in different environments. The pre-requirement for portability is the generalized abstraction between the application logic and system interfaces.</i>	5.3
6)	<b>Web-server</b>	<i>A <b>Web server</b> is a program that uses <b>HTTP</b> (Hypertext Transfer Protocol) to serve the files that form <b>Web</b> pages to users, in response to their requests, which are forwarded by their computers' <b>HTTP</b> clients.</i>	2.4, 2.6, 3.3, 3.2
7)	<b>Scalable Web</b>	<i>Scalability is the capability of a system, network, or process to handle a growing amount of work, or its potential to be enlarged to accommodate that growth</i>	5.1
8)	<b>SMTP server</b>	<i>An SMTP server is the machine that takes care of the whole email delivery process: that's what to send your messages with an email client or software</i>	4.4.2, 3.3



# **Appendix B: Case Studies**

## **PRESENT SYSTEM ANALYSIS**

### **Hostel Allocation**

In the current scenario, the hostel warden creates an excel sheet containing the details of all the registered students who wish to get rooms allocated to them in the hostel. The warden then manually allocates each and every one of them their respective rooms based on the current status and availability of the rooms.

### **Hosteller and Wardens Repository**

At the moment, such a repository does not exist hence provides no firm ground for comparison.

### **Mess Feedback system**

This is carried out in the form of leaflets that are circulated among the students to fill in and rate the quality of the mess food as well as the caterer.

### **Medicine/Sport Accessories**

There is no regulation and improper maintenance regarding medicine and sports Accessories available in the hostel.

### **Grievance/complaint**

In the event that a hosteller faces any grievance or has any complaint regarding anything, he/she is requested to contact the concerned faculty or representative to address the issue.

### **Other Room Management**

A register is maintained with the security regarding current status of the room – who has the key to the given room along with his details.

### **Entry and Exit Management system**

The students enter their names and their destination in the in/out or local register maintained with the security guard in the hostel.