

## **ABSTRACT**

This project “**HOSTEL MANAGEMENT SYSTEM**” is developed for managing various activities in a hostel. Managing the records of a large no of students has always been a challenging task for a hostel manager. There is a lot of strain on the person who is running the hostel and software are not usually used in this context. Managing everything on paper is now a days not a beneficial way as it is time consuming, error prone and have many more complications like that. Various new features like MESS MANAGEMENT, STAFF SHIFTS, and PAYMENT DETAILS etc. are also included in this project to facilitate an Admin.

Identifying the problems which occur when carried manually, we have tried our best to design the software in such a way that a not so tech-friendly administrator can also handle it an easy way. Interface has been kept simple and user-friendly keeping in mind the complications in managing a hostel. One can enhance his efficiency and productivity being an administrator using this software, thus overcome the drawbacks of the existing traditional system.

**Front End:** - Java has been used to develop the front end.

**Back End:** - My SQL has been used to develop the back end of the Project.

# **CONTENTS**

## **1. Introduction**

- 1.1 Purpose
- 1.2 Scope
- 1.3 Definitions, Acronyms and Abbreviations
- 1.4 Tools Used
- 1.5 References
- 1.6 Technologies to be used
- 1.7 Overview

## **2. Overall Description**

- 2.1 Software Interface
- 2.2 Hardware Interface
- 2.3 Communication Interface
- 2.4 Constraints

## **3. Specific Requirement**

- 3.1 Functional requirements
  - 3.1.0 Login
    - 3.1.0.1 Login Detail
    - 3.1.0.2 Forgot Password
    - 3.1.0.3 Reset Password
  - 3.1.1 Room Matrix
    - 3.1.1.1 Availability
    - 3.1.1.2 Room Allocation

- 3.1.2 Registration
  - 3.1.2.1 New Student Reg
- 3.1.3 Student Details
  - 3.1.3.1 UID
  - 3.1.3.2 Search
    - 3.1.3.2.1 Student's Name
    - 3.1.3.2.2 Father's Name
    - 3.1.3.2.3 DOB
- 3.1.4 Mess
  - 3.1.4.1 Daily expenses
  - 3.1.4.2 Net Bill
  - 3.1.4.3 Electricity Bill
  - 3.1.4.4 Miscellaneous Expenditure

### 3.2 Non-Functional requirements

- 3.2.1 Performance
- 3.2.2 ID Verification
- 3.2.3 Access Permission
- 3.2.4 Maintainability
- 3.2.5 Reliability
- 3.2.6 Other Constraints

## 4. DFD

- 4.1 Symbols to be used
- 4.2 context level DFD
- 4.3 Level 0 DFD
- 4.4 Level 1 DFD
- 4.5 level 2 DFD
- 4.6 Nations Used

## 5. ER diagram

- 6. Design Module Diagram
- 7. Data Table Design

## 8. Conclusion

# 1. Introduction

### 1.1 Purpose

The purpose of this document is to present an overall description and listing of the functionality of the system behind the project “**HOSTEL MANAGEMENT SYSTEM**”. It is developed for the hostel administrator keeping in mind the complications they face during management such as room allocation, mess management, bills etc.

### 1.2 Scope

- The basic users of this software is Hostel administrator.
- The user will be provided a login panel as interface.

### 1.3 Definitions, Acronyms and Abbreviations

- **HMS-** Hostel management system  
It's an application to manage various functionalities of a Hostel.
- **Admin-** Administration  
The one who is intended to use this application.
- **J2EE-** Java 2 Enterprise Edition  
A programming platform which is a part of java platform for developing and distributed java.

- **RAD-** Rational application developer  
It helps to design the diagrams like ER, Database schema Diagrams and to generate DDL.
- **DB -** Database

## 1.4 Tools used

### **JAVA**

Java is an object-oriented programming language developed by Sun Microsystems a company best known for its high end UNIX workstations. Java Language was designed to be small, simple, and portable across platforms, Operating systems, both at the source and at the binary level, which means that Java programs (applet and application) can run on any machine that has the Java virtual machine (JVM) installed.

### **J2EE**

**Java Platform, Enterprise Edition** or **Java EE** is a widely used [platform](#) for [Server](#) programming in the [Java](#) programming language. The [Java](#) platform (Enterprise Edition) differs from the [Java Standard Edition Platform](#) (Java SE) in that it adds libraries which provide functionality to deploy fault-tolerant, [distributed](#), [multi-tier](#) Java [software](#), based largely on [modular components](#) running on an [application server](#).

### **RAD**

IBM Rational Application Developer for Web Sphere Software for Web Sphere Software. RAD is an integrated development environment (IDE), made by IBM's Rational Software division, for visually designing, constructing, testing, and deploying Web services, portals, and Java (J2EE) applications.

### **MySQL**

MySQL is the world's most popular open source database. With its prove Performance, reliability and ease-of-use, MySQL has become the leading

Database choice for web-based applications, used by high profile web Properties including Facebook, Twitter, YouTube, Yahoo! and many more. Oracle drives MySQL innovation, delivering new capabilities to power next generation web, cloud, mobile and embedded applications.

## 1.5 References

- [Java Tutorial,](#)  
[www.tutorialspoint.com/java/index.htm](http://www.tutorialspoint.com/java/index.htm)
- [PHP: MySQL Database,](#)  
[www.w3schools.com/php/php\\_mysql\\_intro.asp](http://www.w3schools.com/php/php_mysql_intro.asp)
- [Software requirements specification.](#)  
[en.wikipedia.org/wiki/Software\\_requirements\\_specification](http://en.wikipedia.org/wiki/Software_requirements_specification)
- [YouTube](#)
- [MySQL Tutorial](#)  
<http://www.tutorialspoint.com/mysql/index.htm>

## 1.6 Technologies to be used

- **DB2:** Relational Database Management System.
- **RAD:** Rational Application Developer.
- **Netbeans:** Official IDE for java 8.
- **MySQL:** Open source Database.
- **JDK:** Java Development Kit

## 1.7 Overview

### Existing System

- Manual Recordkeeping
- Multiple Paper Registers.
- Manager

### **★ Drawbacks**

- Security is not ensured.
- Searching of any record is tough.
- Too much paper work.
- Very hard to keep archives maintained.
- Stressful for an individual person to handle.

### **★ Propose System**

- A software package containing all the functionalities.

### **★ Benefits**

- Automated system
- Requires only a man with very little technical knowledge to operate.
- Data Security is ensured.
- Data retrieval is fast.
- Since it's paperless, we can archive years of data.
- Record keeping is very simple.
- Automatic bill calculations.

## **2. Overall Description**

### **2.1 Software Requirement**

- Windows versions after 98.
- Windows Installer.
- Java
- MySQL Server

### **2.2 Hardware Requirement**

#### **Minimum Requirements**

<b>Processor</b>	<b>RAM</b>	<b>Storage</b>	<b>Display</b>
------------------	------------	----------------	----------------

<b>Intel Pentium III or AMD - 800 MHz</b>	128 MB	2 GB	600*480
---	--------	------	---------

#### **Recommended Requirements**

<b>Processor</b>	<b>RAM</b>	<b>Storage</b>	<b>Display</b>
<b>Intel Pentium III with clock speed 1GHz</b>	512MB	128 GB	1024*768

## **2.3 Communication Interface**

### **➤ Admin**

The administrator will login into the system via an admin panel.

## **2.4 Constraints**

- GUI is only in English.
- Login and password is used for the identification of users.
- Only Admin is authorized to access the software.

## **3. Specific Requirements**

### **3.1 Functional requirements**

#### **3.1.0 Login**

##### **3.1.0.1 Login detail**

**Input:** - Enter user name and password.

**Output:** - A page containing the below mentioned function

**Process:** - Password will be compared. If matched proceed, Requirements. If unmatched, display Invalid.

#### **3.1.0.2 Forgot password**

**Input:** - Enter user name.

**Output:** - A page containing the link “RESET PASSWORD”.

#### **3.1.0.3 Reset password**

**Input:** - Enter the security question and answer.

**Output:** - If matched with DB, Type new password.  
If Unmatched, Wrong Credentials.

**Process:** - Comparison of credentials with DB.

### **3.1.1 Room Matrix**

#### **3.1.1.1 Availability**

**Input:** - Select room availability.

**Output:** - A page containing the availability of room.

**Process:** - Retrieval of details from database.

#### **3.1.1.2 Room allocation**

**Input:** - Enter unique id and available room no.

**Output:** - Confirmation message and print the check in bill.

### **3.1.2 Registration**

### **3.1.2.1 New Student Registration**

**Input:** - Details of Student, Room no & Payment.

**Output:** - Generation of an unique Id no.

**Process:** - Creation of an entry about student in Database

## **3.1.3 Students Details**

### **3.1.3.1 UID**

**Input:** - Enter UID no.

**Output:** - Showing the student profile.

### **3.1.3.2 Search**

#### **3.1.3.2.1 Student's Name**

**Input:** - Enter the first name of student.

**Output:** - A list containing all the names containing that  
Particular first name.

**Process:** - Keyword will be compared with Database.

#### **3.1.3.2.2 Father's Name**

**Input:** - Enter the father's name.

**Output:** - A list containing all the names containing  
that particular name.

**Process:** - Keyword will be compared with Database.

### **3.1.3.2.3 D.O.B**

**Input:** - Enter the D.O.B.

**Output:** - A list containing all the names containing the particular D.O.B's.

**Process:** - Keyword Value will be compared with DB.

## **3.1.4 MESS**

### **3.1.4.1 Daily Expenses**

**Input:** - Update the daily Expenses Column.

**Output:** - Confirmation message “Updated”.

### **3.1.4.2 Net Bill**

**Input:** - Update the Net Bill Column along with Payment date.

**Output:** - Confirmation message “Updated”.

### **3.1.4.3 Electricity Bill**

**Input:** - Update the Electricity Bill Column along with payment date

**Output:** - Confirmation message “Updated”.

### **3.1.4.4 Miscellaneous Expenditure**

**Input:** - Update the Column along with payment date.

**Output:** - Confirmation message “Updated”.

## **3.2 Non Functional requirements**

### **3.2.1 Performance**

The system shall support all type of room.

### **3.2.2 ID Verification**

The system requires the user to identify by using an ID no. at the Checkout point.

### **3.2.3 Access Permission**

This HMS shall have several type of access permissions. For instance, the warden is recognized as the system's administrator. Thus, the warden shall be able to perform any type of activities on the system and both the user's and student profiles. The public in general shall be restricted from accessing any user profile. However, they shall be granted a read access on the student profile.

### **3.2.4 Maintainability**

The system shall provide the capability to back up the database.

### **3.2.5 Reliability**

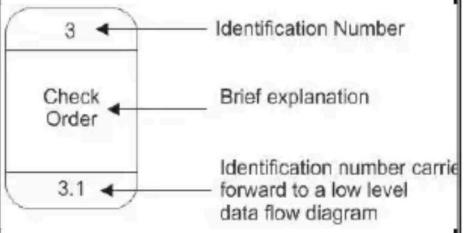
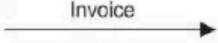
The system shall be available 99.9% of the time.

### **3.2.6 Other constraints**

The system shall support other hostel issues. The HMS shall be flexible and adaptable due to future plans of expanding the system.

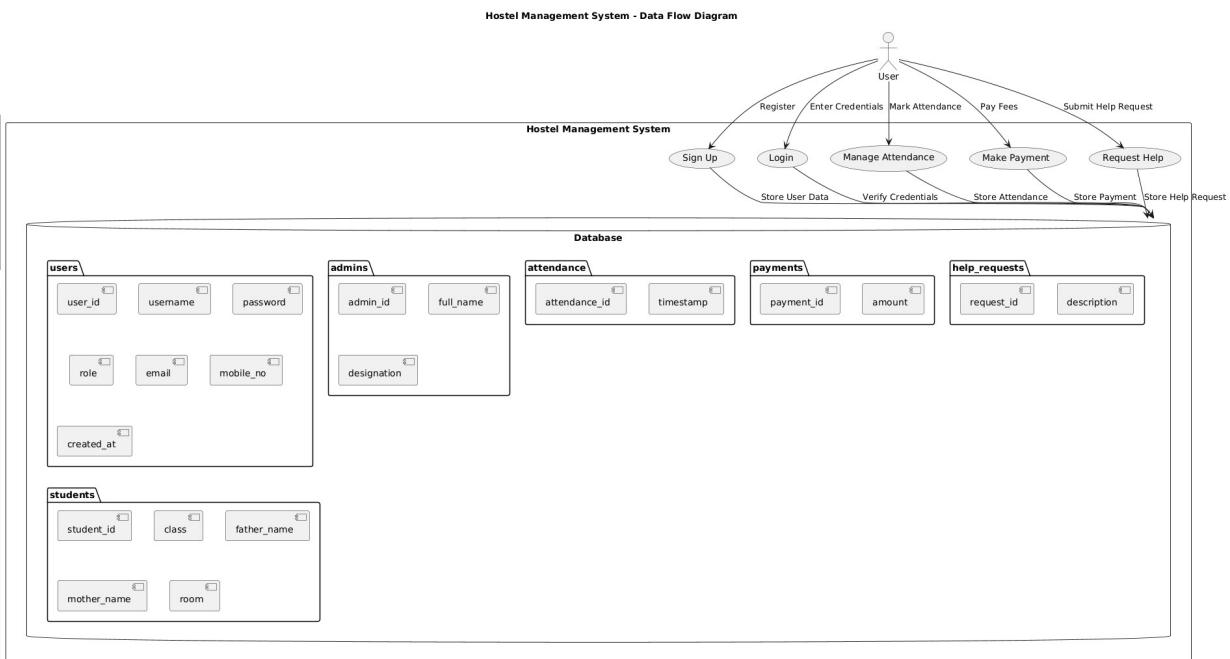
## **DATA FLOW DIAGRAM**

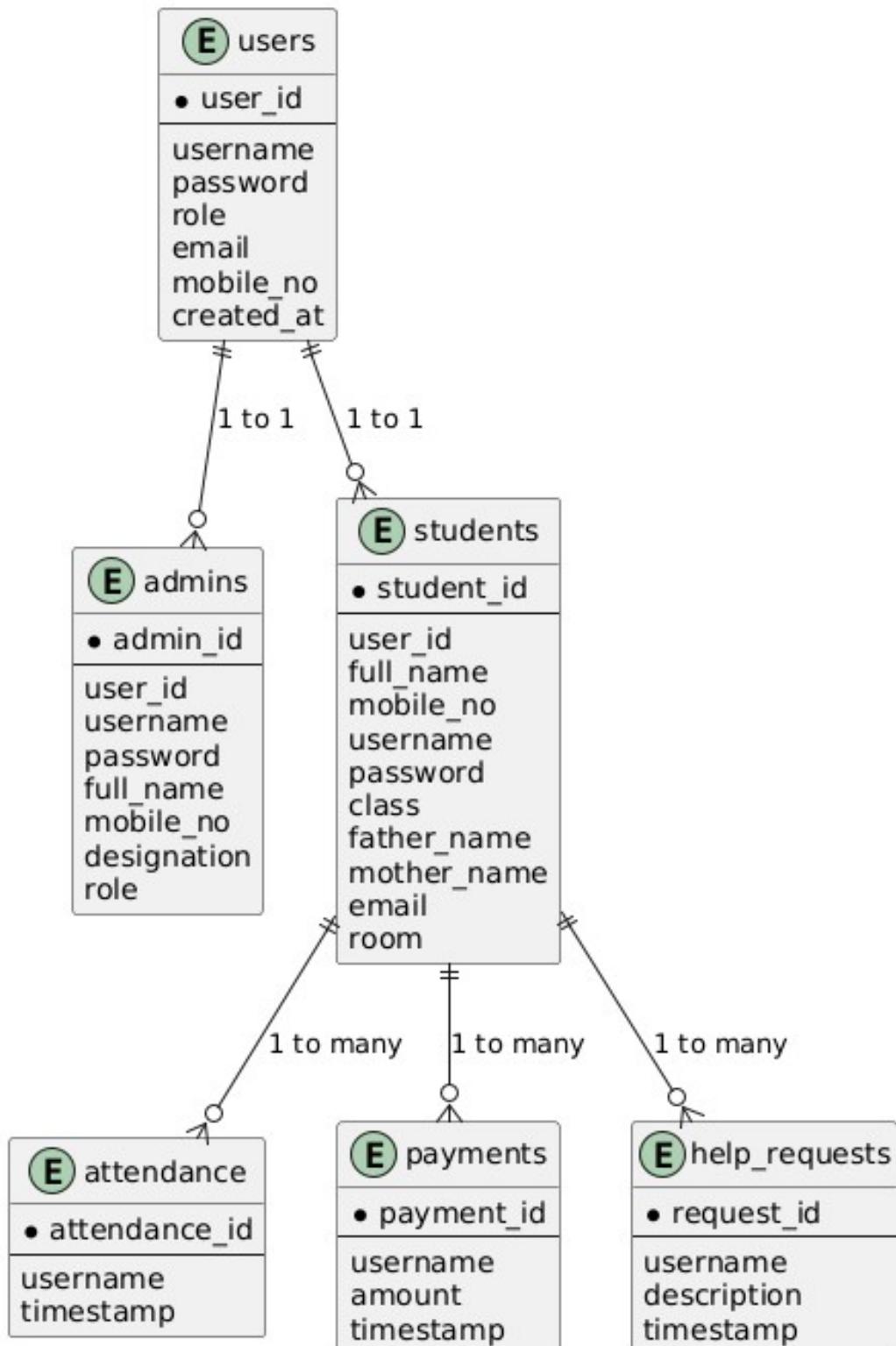
## Symbols to be used:

Symbol	Meaning	Example
	An <b>entity</b> . A source of data or a destination for data.	
	A <b>process</b> or task that is performed by the system.	 <p>3 ← Identification Number      Check Order ← Brief explanation      3.1 ← Identification number carried forward to a low level data flow diagram</p>
	A <b>data store</b> , a place where data is held between processes.	
	A <b>data flow</b> .	



## Construct Context Level DFD



**Hostel Management System - Entity-Relationship Diagram**

## Notations used:-

Notations	Their meaning
<b>HMS</b>	Hostel Management System
<b>UID</b>	Unique Identification Number
<b>D.O.B</b>	Date of Birth
<b>DB 0</b>	Credentials database
<b>DB 1</b>	Room Availability Database
<b>DB 2</b>	Student Details Database
<b>DB 3</b>	Mess Details Database
<b>1.0</b>	Room Matrix Function
<b>2.0</b>	Registration Function
<b>3.0</b>	Student Function
<b>4.0</b>	Mess Details Function
<b>3.1</b>	Function to enter UID
<b>3.2</b>	Function to search UID
<b>3.1.1</b>	Detail Functionality of 3.1
<b>3.1.2</b>	Detail Functionality of 3.2

## **INPUT OUTPUT DESIGN**

### **INPUT DESIGN**

This part deals with the various input from supplied to the Admin so as to get the details which can be further processed in a meaningful way.

#### **Username and Password**

User need to provide a valid username and password. After entering the details first the administrator will check if the entered username and password exist in the data base or not. The process only start if the Admin is found authenticated.

### **OUTPUT DESIGN**

- Room Matrix.
- Student Registration
- Student Details
- Payment Due

## **DATA TABLE DESIGN**

### **CREDENTIALS (DB 0)**

S.NO	Field Name	DATA TYPE	NULLABLE	PRIMARY KEY
1	User Name	VARCHAR(20)	NO	YES
2	Password	VARCHAR(20)	YES	NO
3	Security Question	VARCHAR(30)	YES	NO
4	Security Answer	VARCHAR(20)	YES	NO
5	D.O.B	DATE	NO	NO
6	New Password	VARCHAR(20)	YES	NO

### **ROOM (DB 1)**

S.NO	FIELD NAME	DATA TYPE	NULLABLE	PRIMARY KEY
1	Room Availability	Matrix	NO	NO
2	Room No	Int(3)	NO	YES

**STUDENT (DB 2)**

<b>S.NO</b>	<b>FIELD NAME</b>	<b>DATA TYPE</b>	<b>NULLABLE</b>	<b>PRIMARY KEY</b>
1	UID	INT(10)	NO	YES
2	Std_name	VARCHAR(15)	NO	NO
3	Father's Name	VARCHAR(30)	NO	NO
4	Mother's Name	VARCHAR(30)	YES	NO
5	D.O.B	DATE	NO	NO
6	Address	VARCHAR(100)	NO	NO
7	Payment	DECIMAL(10,4)	YES	NO

**MESS (DB 4)**

<b>S.NO</b>	<b>FIELD NAME</b>	<b>DATATYPE</b>	<b>NULLABL E</b>	<b>PRIMARY KEY</b>
1	Netbill_invoice_no	INT(20)	NO	YES
2	Netbill_payment_date	DATE	NO	NO
3				

	Netbill_amount	INT(5)	NO	NO
4	Electricity_invoice_no	INT(20)	NO	YES
5	Electricity_payment_date	DATE	NO	NO
6	Electricity_amount	INT(5)	NO	NO
7	Daily_Expense_amt	INT(5)	YES	NO
8	Daily_Expense_date	DATE	NO	NO
9	Misc_Expense_amt	INT(5)	NO	NO
10	Misc_Expense_date	DATE	NO	NO

## **CONCLUSION**

The “**HOSTEL MANAGEMENT SYSTEM**” project will serve as a useful project to carry out the complex process in a simple and vivid manner for search of criminals. It serves as a helpful approach for the police for assembling and improving their things. It reduces the time taken and the redundant effort wasted during the criminal details searching as it is a centralised approach. Thus, the project is user friendly.

