**A**

**Project Report On**

**“Hostel Management System”**

**BTech-Sem VI**

**Prepared By**

**Bhikhabhai Dhobi (IT028)**

**Smit Desai (IT024)**



**DEPARTMENT OF INFORMATION TECHNOLOGY**

**FACULTY OF TECHNOLOGY,**

**DHARMSINH DESAI UNIVERSITY**

**COLLEGE ROAD, NADIAD- 387001**

**A**

**Project Report On**

**“Hostel Management System”**

**In partial fulfillment of requirements for System Design Practice**

**B. Tech, Sem VI**

**Submitted By:**

**1 Bhikhabhai Dhobi (IT028)**

**2 Smit Desai (IT024)**

**Under the Guidance of Prof. Yogendra L.Patel**



***DEPARTMENT OF INFORMATION TECHNOLOGY***

**FACULTY OF TECHNOLOGY, DHARMSINH DESAI UNIVERSITY COLLEGE ROAD, NADIAD- 387001**

**CANDIDATE’S DECLARATION**

We declare that semester 6th report entitled

“Hostel Management System” is our own work conducted under the supervision of the guide Yogendra L.Patel.

We further declare that to the best of our knowledge the report for B.Tech. VI semester does not contain part of the work which has been submitted either in this or any other university without proper citation.

Bhikhabhai Dhobi

19ITUBS017

Smit Desai

19ITUOS069

Submitted To:

Yogendra L.Patel

Department of Information Technology,

Faculty of Technology,

Dharmsinh Desai University,

Nadiad

Gujarat.

**DHARMSINH DESAI UNIVERSITY**

## NADIAD-387001, GUJARAT



**CERTIFICATE**

**This is to certify that the project carried out in the subject of System Design Practice, entitled “Hostel Management System” and recorded in this report is a bonafide report of work of**

| **1) Smit Desai** | **Roll No. IT024** | **ID No:19ITUOS069** |
| --- | --- | --- |
| **2) Bhikhabhai Dhobi** | **Roll No. IT028** | **ID No: 19ITUBS017** |

**of the Department of Information Technology, semester VI. They were involved in Project work during the academic year 2021 – 2022.**

Prof. Yogendra Patel

(Project Guide)

Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date:

Prof. (Dr.) Vipul K. Dabhi,

Head, Department of Information Technology, Faculty of Technology,

Dharmsinh Desai University, Nadiad Date:

**ACKNOWLEDGMENT**

On the very outset of this report, we would like to extend our sincere & heartfelt obligations towards all the personages who have directly or indirectly helped in this endeavor. Without their active guidance, help, cooperation & encouragement, we wouldn’t have made the headway in the project.

We are ineffably indebted to our instructor **Prof. Yogendra Patel** for her guidance, encouragement and conscientiousness to accomplish this project. We are extremely thankful for his support.

We would also like to express our gratitude to our Head, **Mr. Vipul Dabhi** along with Dharmsinh Desai University for giving us this opportunity.

Any omission in this brief acknowledgement does not mean lack of gratitude.

Thank You,

Desai Smit

Dhobi Bhikhabhai

**TABLE OF CONTENT**

1. [Introduction… 6](#_30j0zll)

Project Details: Broad specifications of the work entrusted to you… 6

[Purpose 6](#_1fob9te)

[Scope 6](#_3znysh7)

[Objective 6](#_2et92p0)

[Technology and Literature Review](#_tyjcwt) 7

1. [Project Management…](#_3dy6vkm) 8

[Feasibility Study…](#_1t3h5sf) 8

[Technical feasibility…](#_4d34og8) 8

[Time Schedule feasibility…](#_2s8eyo1) 8

[Operational feasibility…](#_17dp8vu) 8

[Implementation feasibility…](#_3rdcrjn) 8

[Project Planning…](#_26in1rg) 9

[Project Development Approach and Justification…](#_lnxbz9) 9

[Project Plan…](#_35nkun2) 10

Roles and Responsibilities… 10

1. System Requirements Study… 11

[Study of Current System… 1](#_1ksv4uv)1

Problems and Weaknesses of Current System… 11

[User Characteristics… 1](#_44sinio)1

[Hardware and Software Requirements… 1](#_2jxsxqh)1

[Constraints… 1](#_z337ya)1

1. [System Analysis… 1](#_3j2qqm3)2

[Requirements of New System… 1](#_1y810tw)2

[Functional Requirements 1](#_41mghml)2

[Non-functional Requirements… 1](#_4i7ojhp)5

1. [System Design… 15](#_2xcytpi)

Use Case Diagram… 15

[Class Diagram… 16](#_2grqrue)

[Sequence Diagram… 17](#_vx1227)

[Flow Chart… 21](#_3fwokq0)

[Component Diagram… 25](#_1v1yuxt)

[Deployment Diagram… 25](#_4f1mdlm)

1. [Implementation Planning… 26](#_3whwml4)

[Implementation Environment… 26](#_2bn6wsx)

[Program/Modules Specification… 26](#_qsh70q)

[Coding Snippets… 28](#_3as4poj)

1. [Testing… 32](#_1pxezwc)

[Testing Plan… 32](#_49x2ik5)

[Testing Strategy… 32](#_2p2csry)

Testing Methods 32

[Test Cases… 32](#_147n2zr)

1. [User Manual… 35](#_3o7alnk)
2. Limitation and Future Enhancement… 55
3. [Conclusion and Discussion… 56](#_23ckvvd)

Conclusions 56

[Discussion… 56](#_ihv636)

[Self-Analysis of Project Viabilities… 56](#_32hioqz)

10.2.3 Problem Encountered and Possible Solutions… 56

10.3.3 Summary of Project work… 56

1. [Bibliography… 57](#_1hmsyys)

# INTRODUCTION

## PROJECT DETAILS

The hostel management system is the system that manages the details of the students

staying in the hostel, admission details, staff’s details and the receipt of the fees paid by

the students. It will be one of the important projects that will simplify the work of the

institutions in getting the seats booked in the hostels with great ease. Hostels are

the part of the institutions which will help the students to stay here who are coming

from far places. It will help the students in getting the seats in the hostels without

any difficulty and will also help in maintaining transparency. It will help in

simplifying life of everyone to acquire seats with great ease. The interface must be

simple and easy to understand.

## PURPOSE

Hostels are the part of the institutions which will help the students to stay here who

are coming from far places. It will help the students in getting the seats in the hostels

without any difficulty and will also help in maintaining transparency. It will help in

simplifying life of everyone to acquire seats with great ease. The interface must be

simple and easy to understand.

## SCOPE

HMS has the main aim of managing room application and allocation in a

Web-based environment. The overall system will consist of HMS Database

System and HMS desktop Interface. The HMS database system will supply the

fundamental database structure of the entire system whereas its desktop

Interface will provide a secure desktop interface between the users and the

database. HMS aims towards establishing a “paperless office” rather

than using a manual system.

## OBJECTIVE

The objectives of the project are: To enable online room application

and allocation in a timely manner, to enable the student’s update

their details online, to eliminate duplicate data, and to enable a

smooth communication channel between the system administrator and its users

## 

## TECHNOLOGY AND LITERATURE REVIEW

We used simple Java JFrame for the frontend as these are all supported by any Desktop. For the backend we used Java and for database we use MYSQL.

# PROJECT MANAGEMENT

## FEASIBILITY STUDY

## Technical Feasibility

After considering our project functionalities and technical needs we decided to go with the following technologies -

Frontend – java JFrame

Backend - java

Database - MySQL

All the above-mentioned technologies are open sources therefore it is feasible to use with the knowledge of basics in each of them it would be easy for us to manage the project too.

## Time Schedule Feasibility

The planning for our project completion is as follows:

Initially we gathered & analyzed all the requirements by the beginning of January 2021. We prepared the SRS document and the GUI design right after that. This was followed by the required diagrams. For coding and unit testing 4 weeks and for system and integration testing another 2 weeks were covered. We are ready for its demonstration in the submission week along with the report. Being a 2 members team, we were able to complete our project in the estimated time.

## Operational Feasibility

In the current situation where the virtual world is new normal, our project will often be very beneficial with the hostel manager. We have kept this app pretty simple and handy so Hostel Manager can take advantage of it and use it without any inconvenience.

## Implementation Feasibility

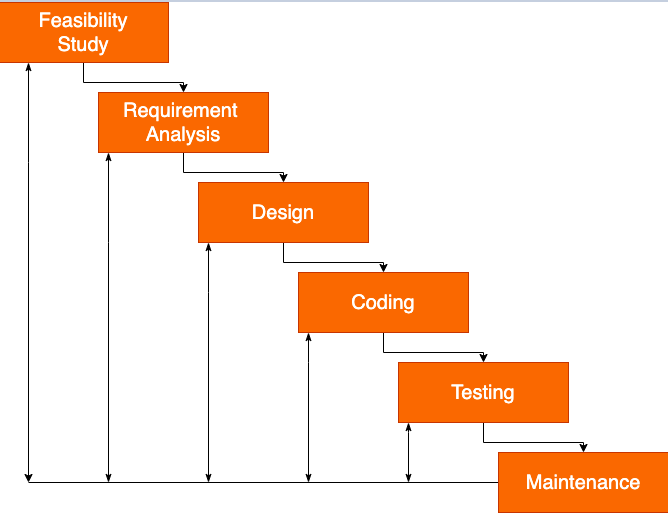
The Functional requirements of our project will be easily fulfilled by the technologies we have mentioned. All the technologies we are using are easily supported by all the Desktop. Hence, the implementation is feasible.

## PROJECT PLANNING

## Project Development Approach and Justification

The software development approach we used for our project : Iterative Waterfall Model This includes the stages like :

1. Feasibility Study
2. Requirement Analysis
3. Design
4. Coding
5. Testing
6. Maintenance



We choose this approach because it gives us the liberty to get back to a stage if it needs some redo. Our entire approach was trial and error and learn as you need. By choosing the iterative waterfall we get the liberty of flexibility and the risk management can be done easily.

## Project Plan

| **Sr. No** | **Documents** | **Timeline** |
| --- | --- | --- |
| **1** | Project Definition, Submission and approval | **December** |
| **2** | System Design | **January** |
| **3** | Implementation | **January** |
| **4** | Testing | **March** |
| 5 | Documentation | **March** |

**Roles and Responsibility**

| **Name** | **Analysis** | **Design** | **Coding** | **Testing** | **Documentation** | **Maintenance** |
| --- | --- | --- | --- | --- | --- | --- |
| **Smit** | ✔ | ✔ | ✔ |  | ✔ |  |
| **Bhikhabhai** | ✔ | ✔ | ✔ | ✔ | ✔ | ✔ |

# 3. SYSTEM STUDY REQUIREMENT

## STUDY OF CURRENT SYSTEM

## Some hostels are managing hostels through pan paper, so it is very stressful and hard work to do.

## PROBLEMS AND WEAKNESS OF CURRENT SYSTEM

## USER CHARACTERISTICS

1. Admin - Maintains the database and the related activities

## 

## HARDWARE AND SOFTWARE REQUIREMENTS

For all the actors the basic software requirements are as follows:

* Windows or Linux

For all the actors the basic hardware requirements are as follows:

* Device to run the desktop application.

## CONSTRAINTS

* As it is a Java based application, it requires Java Environment.
* Response time can vary depending upon pc.

# 4.SYSTEM ANALYSIS

## REQUIREMENTS OF NEW SYSTEM

**Functional Requirements**

R4.1. Administrator Functions

R4.1.1 login

Intro: in login admin logged into system

i/p: In input there is id and password

o/p: if id and password is correct so redirect to home page

process: In process check id and password is correct or not and if not correct so give error

R4.1.2 Manage Room

Intro: in this functionality admin have permission to add room or remove room

i/p: add and remove room details

o/p: display room details

process: add room and if it available so give error or remove room

R4.1.2.1 Add Room

Inro: In this functionality admin add the rooms

I/P: Add room no

O/P: Add room in database if there is not exist and if there exist so give error.

Process: Do appropriate changes in database

R4.1.2.2 Delete Room

Intro: Admin have permission to delete room

I/p: Delete room

O/P: Delete room from the database

Process: Do appropriate changes in database

R4.1.3 Manage Students

R4.1.3.1 Add Students

Intro: in this functionality admin add students in system

i/p: admin add student’s details

o/p: add student details into database

process: add details to database

R4.1.3.2 Update student

Intro: in this functionality admin update student’s details

i/p: admin have permission to update student details

o/p: change details in database

process: change details in database

R4.1.3.3 Delete Students

Intro: in this functionality admin remove student’s details

i/p: admin have permission to remove students

o/p: remove details in database

process: remove details in database

R4.1.3.4 Student Fees

Intro: in this functionality admin add details of student’s fees

i/p: add fees details of students

o/p: show details how many students pay their fees

process: add details of paid fees of students

R4.1.3.5 All Students leaving

Intro: in this functionality admin see all details of all leaving students

i/p:

o/p: show all details of students who leave in hostel

process: read data from database and display

R4.1.3.6 leaved Students

Intro: in this functionality admin see which students leave hostel

i/p:

o/p: show all students details who leave hostel

process: read database and show all details of leaved students

R4.1.4 Manage Employee

R4.1.4.1 New Employee

Intro: in this functionality admin add the details of employees

i/p: add employee details

o/p: add employee details to database

process: add employee details to database

R4.1.4.2 Update Employee Details

Intro: in this functionality admin update employee’s details

i/p: update employee details

process: appropriate change in database

R4.1.4.3 Delete Employee

Intro: in this functionality admin remove employee’s details

i/p: remove employee details

process: appropriate change in database

R4.1.4.4 Employee Payment

Intro: in this functionality admin add employee’s salary details

i/p: add details of payment to employees

o/p: appropriate change into database

process: add details of employee payment into database

R4.1.4.5 All Employee Working

Intro: in this functionality admin see all employee’s details

i/p:

o/p: display all working employee’s details

process: read data from database and display

R4.1.4.6 Leaved Employees

Intro: in this functionality admin see all details of leaved employees

i/p:

o/p: show all leaved employee details

process: read database and show all details of leaved employee

**Non-Functional Requirements**

#### Security:

The system needs to log client’s information of registration such as IP address and time for security purpose.

Password should encrypt and store in the database.

#### 

#### Maintainability:

The application is to be designed so that it is easily maintained. Also, it should allow incorporating new requirements in any module of system.

#### Portability:

The application will be easily portable on any window-based system.

#### Usability:

Relative JDK Environment

The system is user friendly and self-explanatory.

Reliability:

The system has to be very reliable due to the importance of data and the damages incorrect or incomplete data can do.

#### Availability:

This application can work 24x7

**Mean Time to Repair (MTTR)**

Even if the system fails, the system will be recovered back up within an hour or less.

**Access Reliability**

The system shall provide the admin 100% access reliability

**Validation Criteria:**

A Software Testing Plan (STP) will be created to define a set of qualification methods and to specify for each requirement in this document a method of ensuring that the requirement is satisfied.

* How do we recognize a successful implementation?
* What classes of tests must be conducted to validate function, performance and constraints?

Platform: It should be responsive for computers

Database: A database management system that is available free of cost in the public domain should be used

# 

# 5.SYSTEM DESIGN

**5.1 USE CASE DIAGRAM**

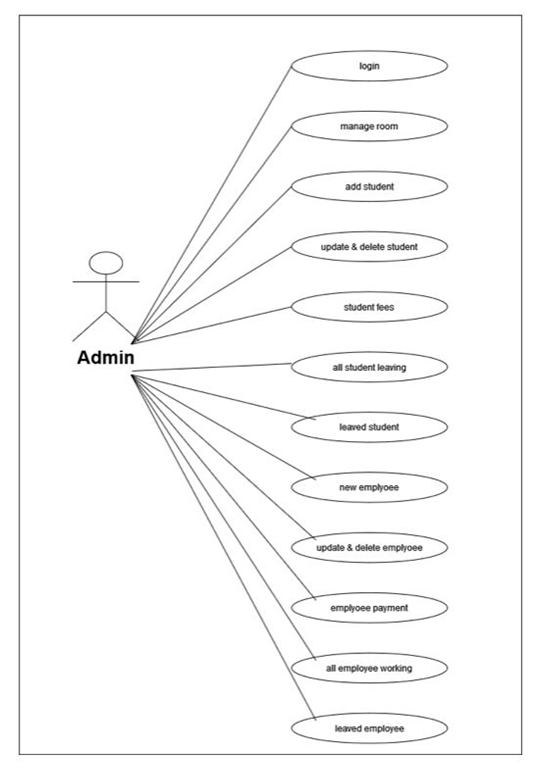
****

Fig. 5.1.1 USE CASE DIAGRAM

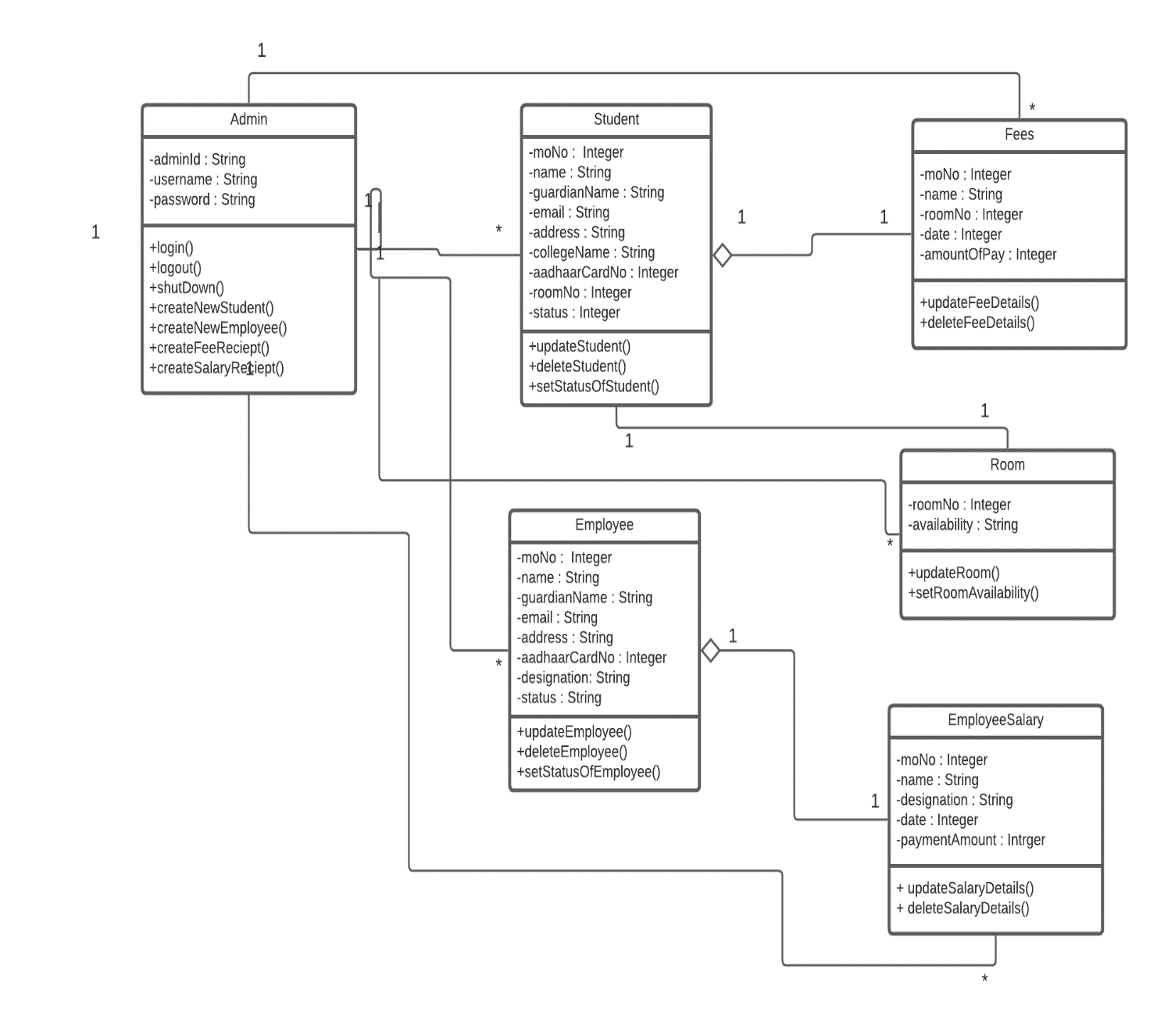
**5.2 Class Diagram**

Fig-5.2.1 Class Diagram

**5.3 Sequence Diagram**

**5.3.1 For Login:**



Fig. 5.3.1.1 For Login:

**5.3.2 For Manage Room:**

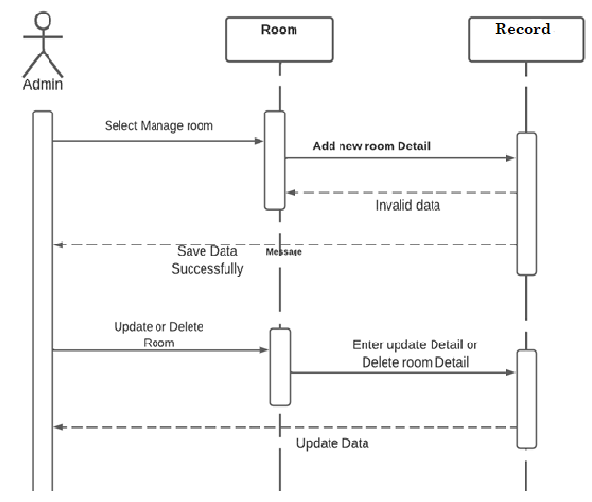


Fig. 5.3.2.1 For Manage Room:

**5.3.3 For Student Fee:**

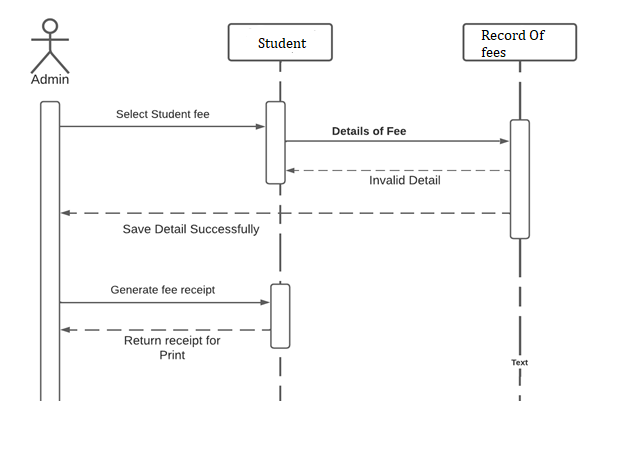


Fig 5.3.3.1 For Student Fee

**5.3.4 For Manage Student Details**:

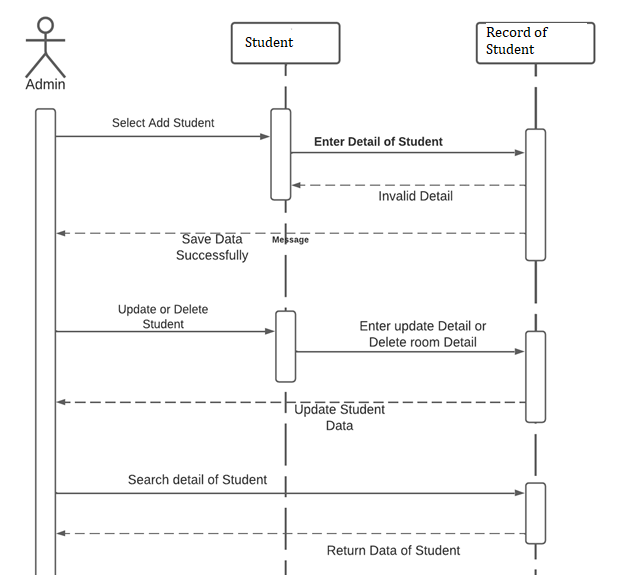


Fig 5.3.4.1 For Manage Student Details

**5.3.5 For Employee Details:**

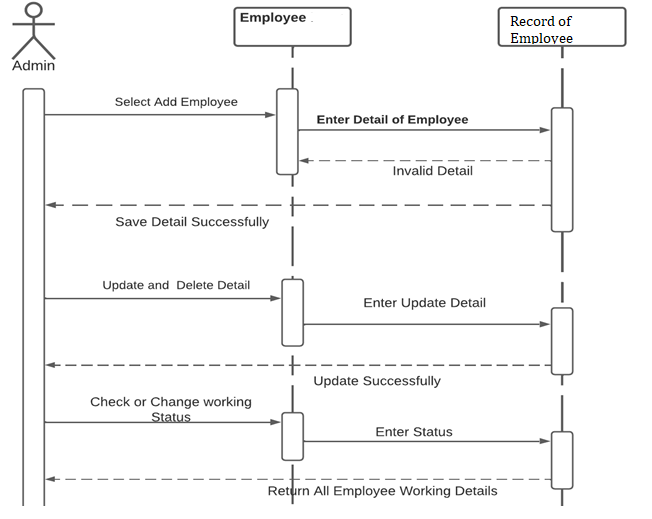


Fig 5.3.5.1 For Employee Details

**5.3.6 For Employee Payment:**

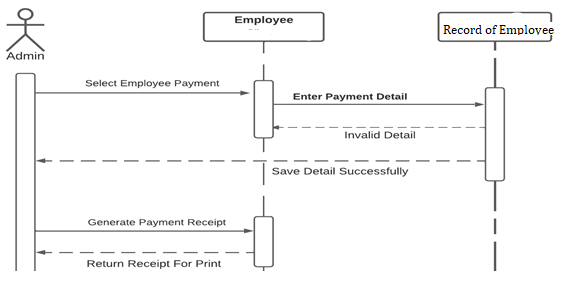


Fig 5.3.6.1 For Employee Payment:

**5.4 Activty Diagram**

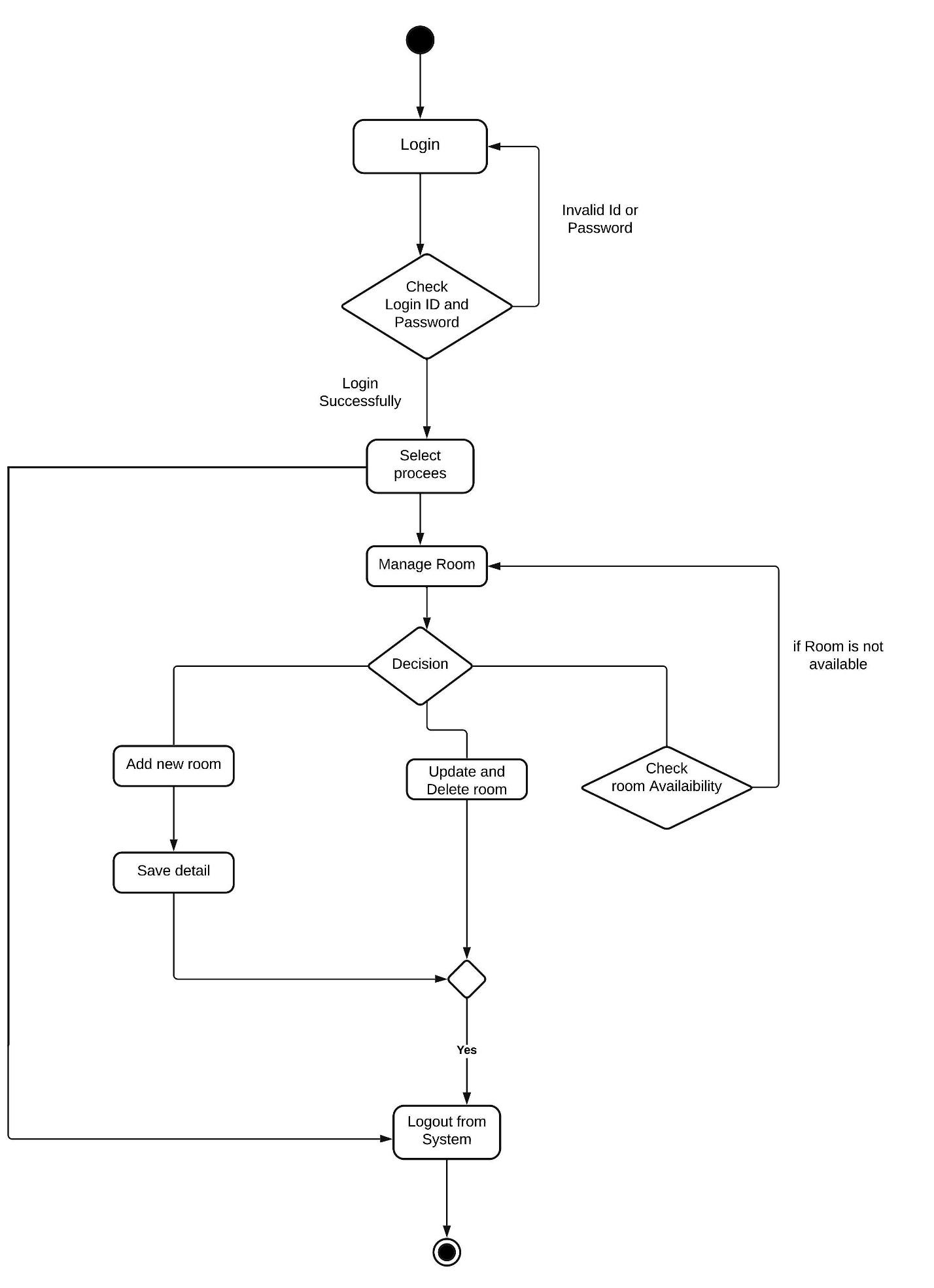
**5.4.1 Activity diagram for For Manage Room**

Fig 5.4.1.1 Activity diagram for For Manage Room

**5.4.2 Activity diagram for Manage Student**

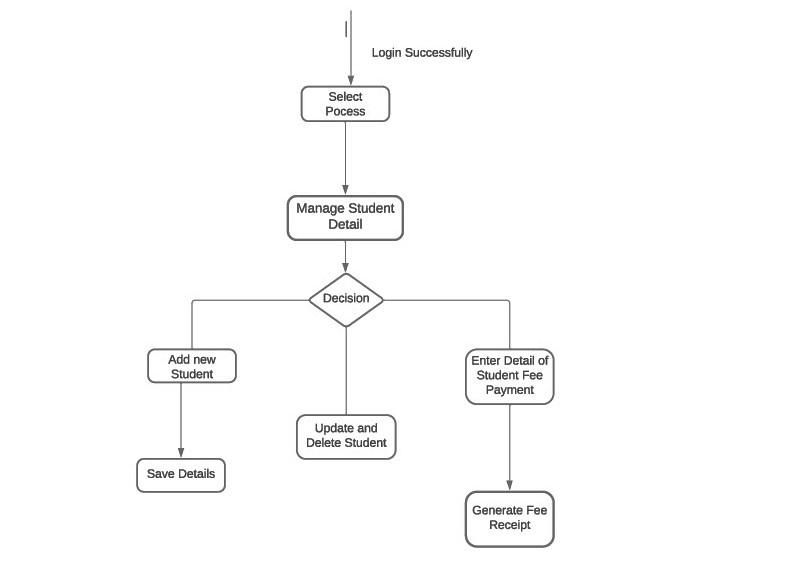


Fig 5.4.2.1 Activity diagram for Manage Student

**5.4.3 Activity diagram for Manage Employee**

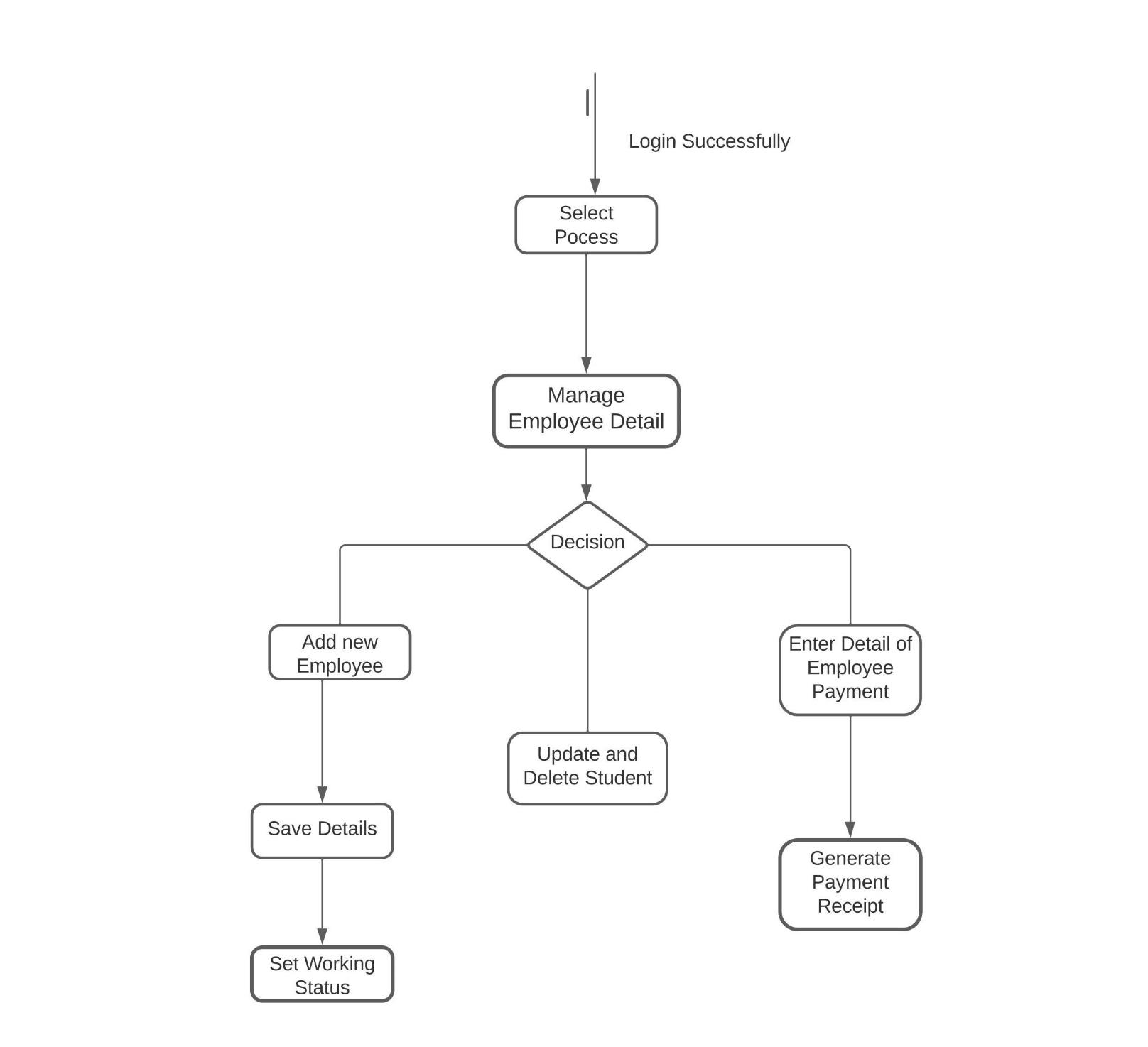


Fig 5.4.3.1 Activity diagram for Manage Employee

**5.5 State-Chart Diagram:**

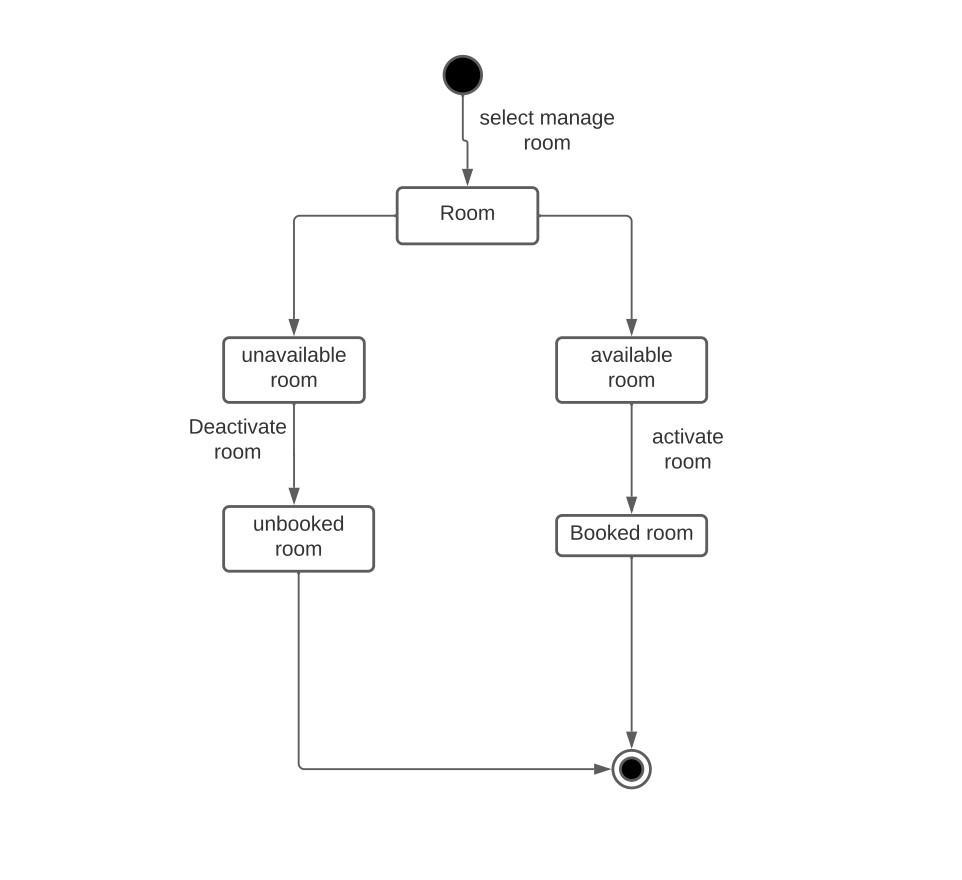


Fig 5.5.1 State-Chart Diagram:

**5.6 Package Diagram**

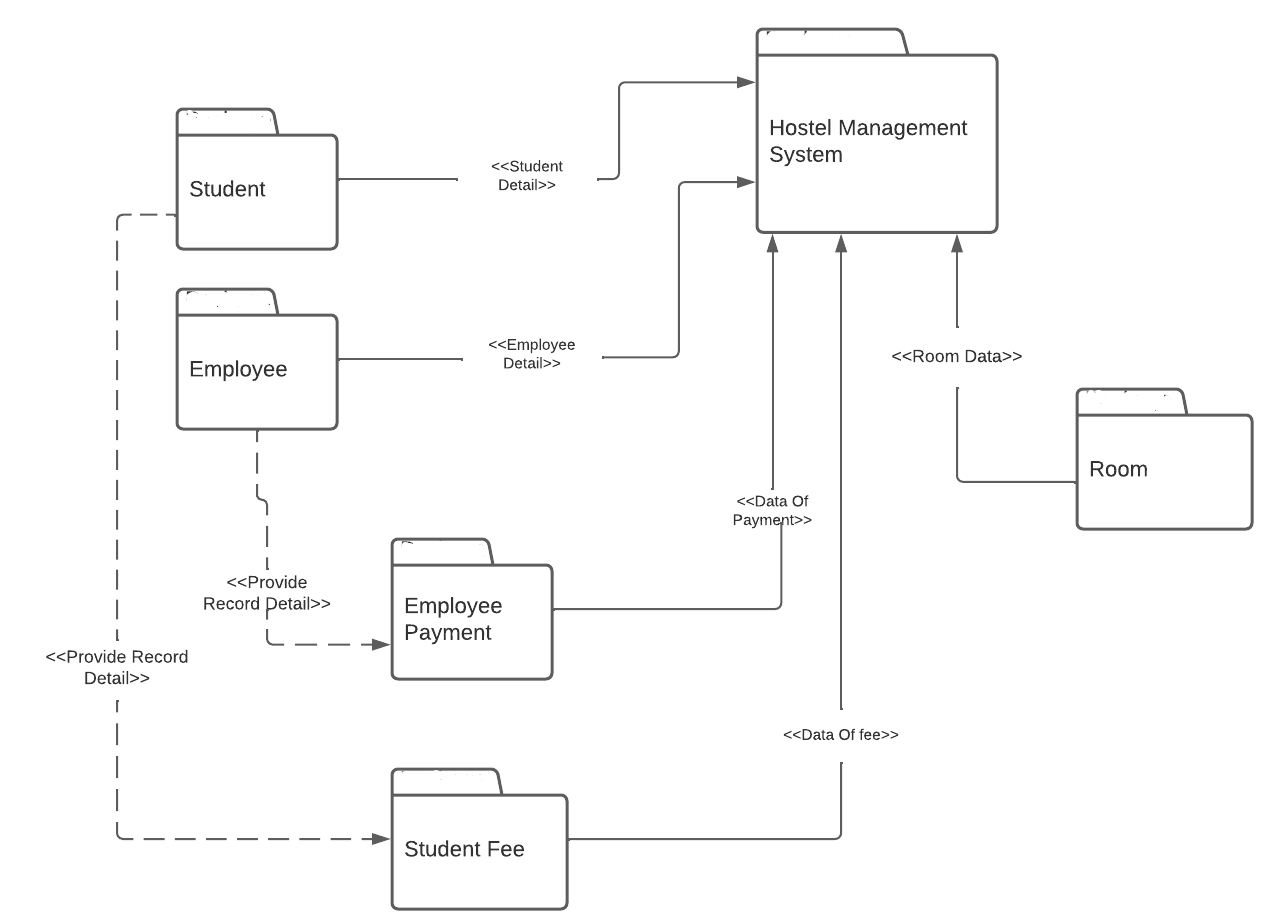


Fig. 5.6.1 Package Diagram

**5.7 Deployment Diagram**

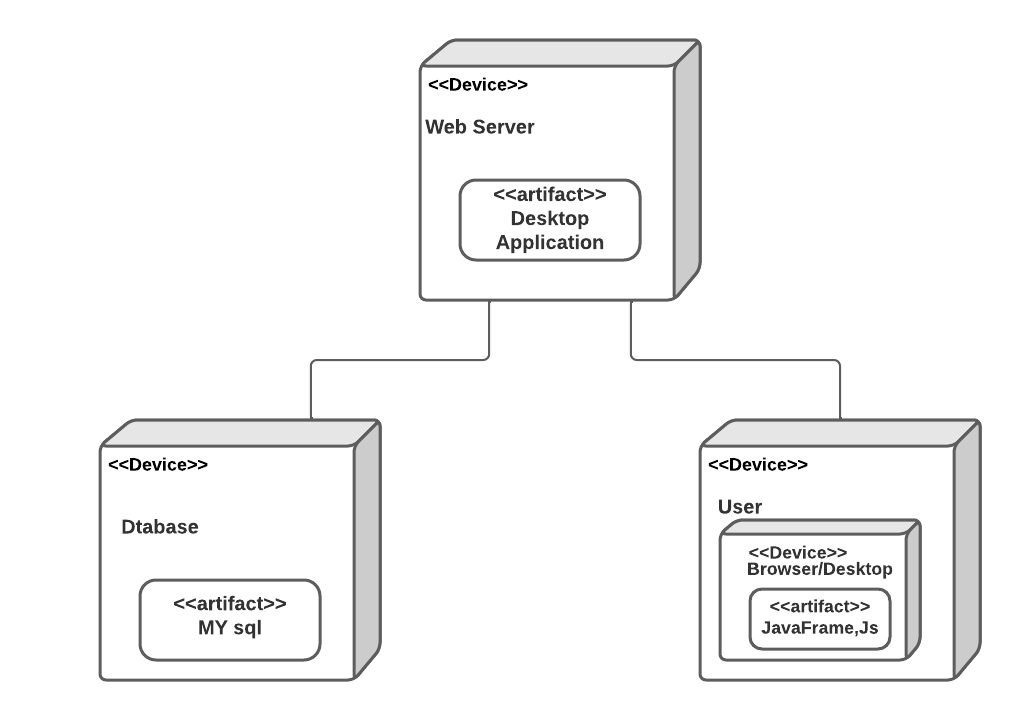


Fig. 5.7.1 Deployment Diagram

**5.8 DFD**

**5.8.1 LEVEL-0 (Context level)**

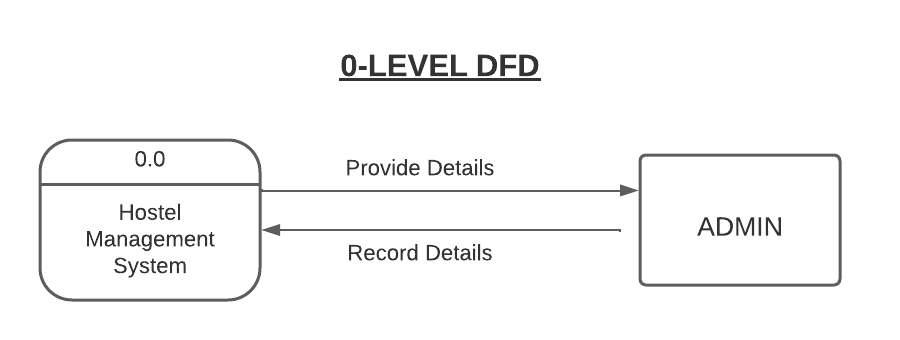


Fig 5.8.1.1 LEVEL-0 (Context level)

**5.8.2 LEVEL-1:**

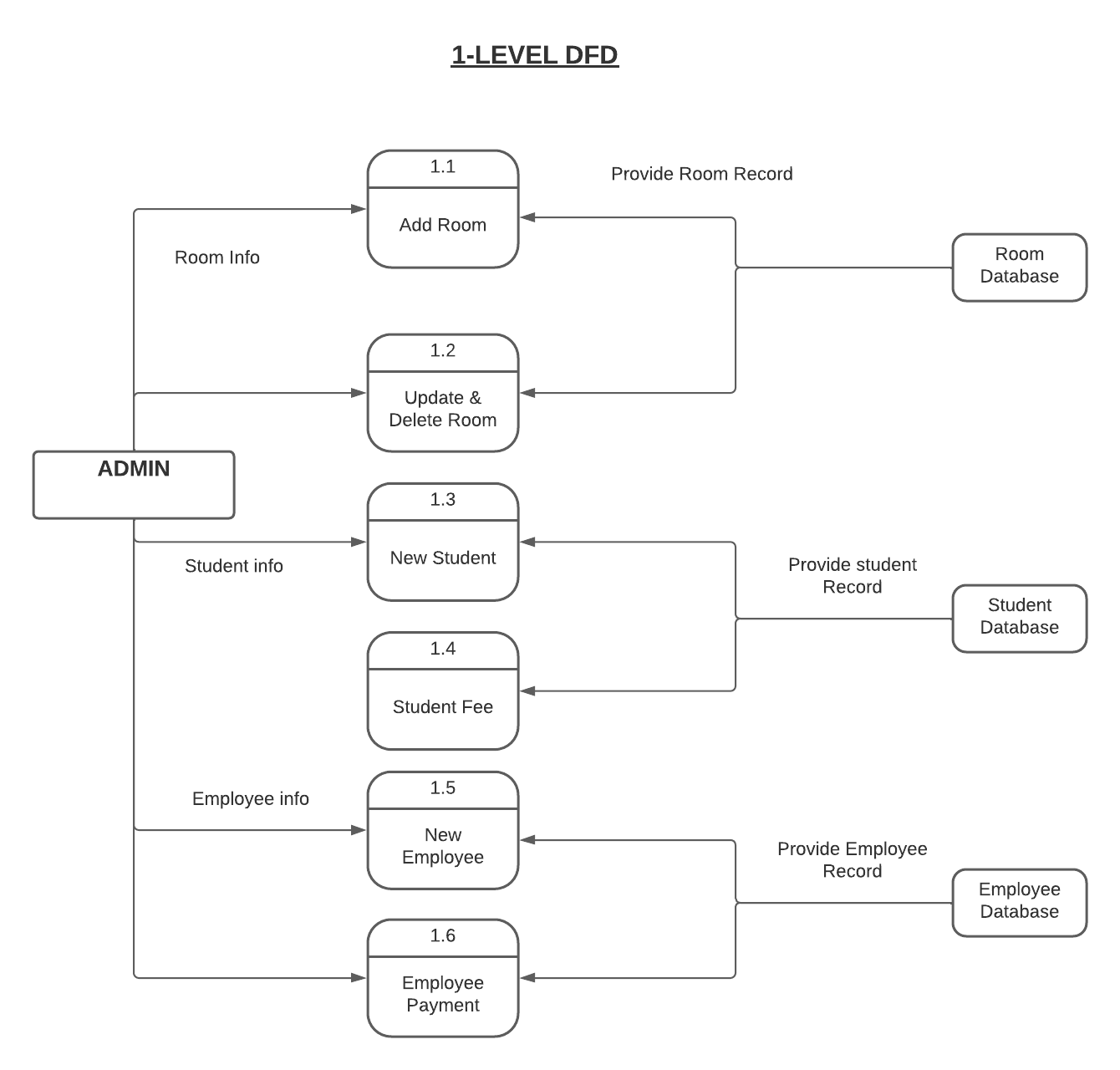


Fig 5.8.2.1 LEVEL-1:

# 

# 6. IMPLEMENTATION PLANNING

## 6.1 IMPLEMENTATION ENVIRONMENT:

During the complete implementation, we’ve worked on NetBeans. NetBeans is free source for GUI Editor

Made by oracle for windows and Linux. Features included support for debugging syntax highlighting intelligent code completion and code refactoring. User can change the theme, keyboard shortcuts, preferences, and additional functionality.

For the data base implementation, we’ve used MySQL. MySQL is powerful, open-source object relational data base system.

## 

## 6.2 PROGRAM / MODULES SPECIFICATION:

Our entire system we have 2 modules.

1. Authentication Module:

This module authenticates the user. It checks the registered credentials with the entered  
credentials. If any discrepancy is found, the user is not allowed to enter the application.

1. Management Module:

This module manages all the use case of this system. It adds or remove room, student and employee.

Update room, students and employee. Additionally, it also keeps and manages the record of student’s fees and employee’s salary payment.

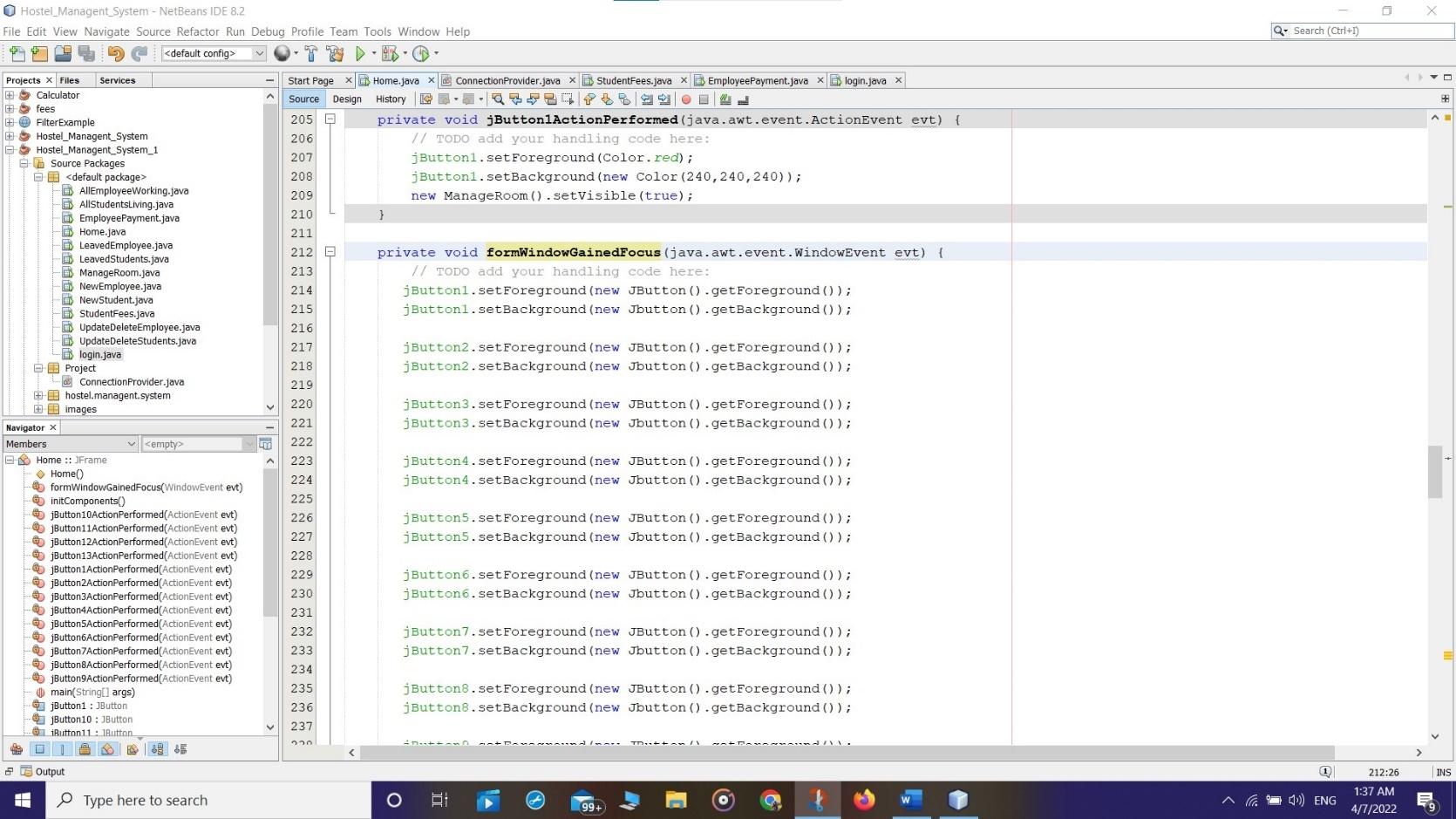
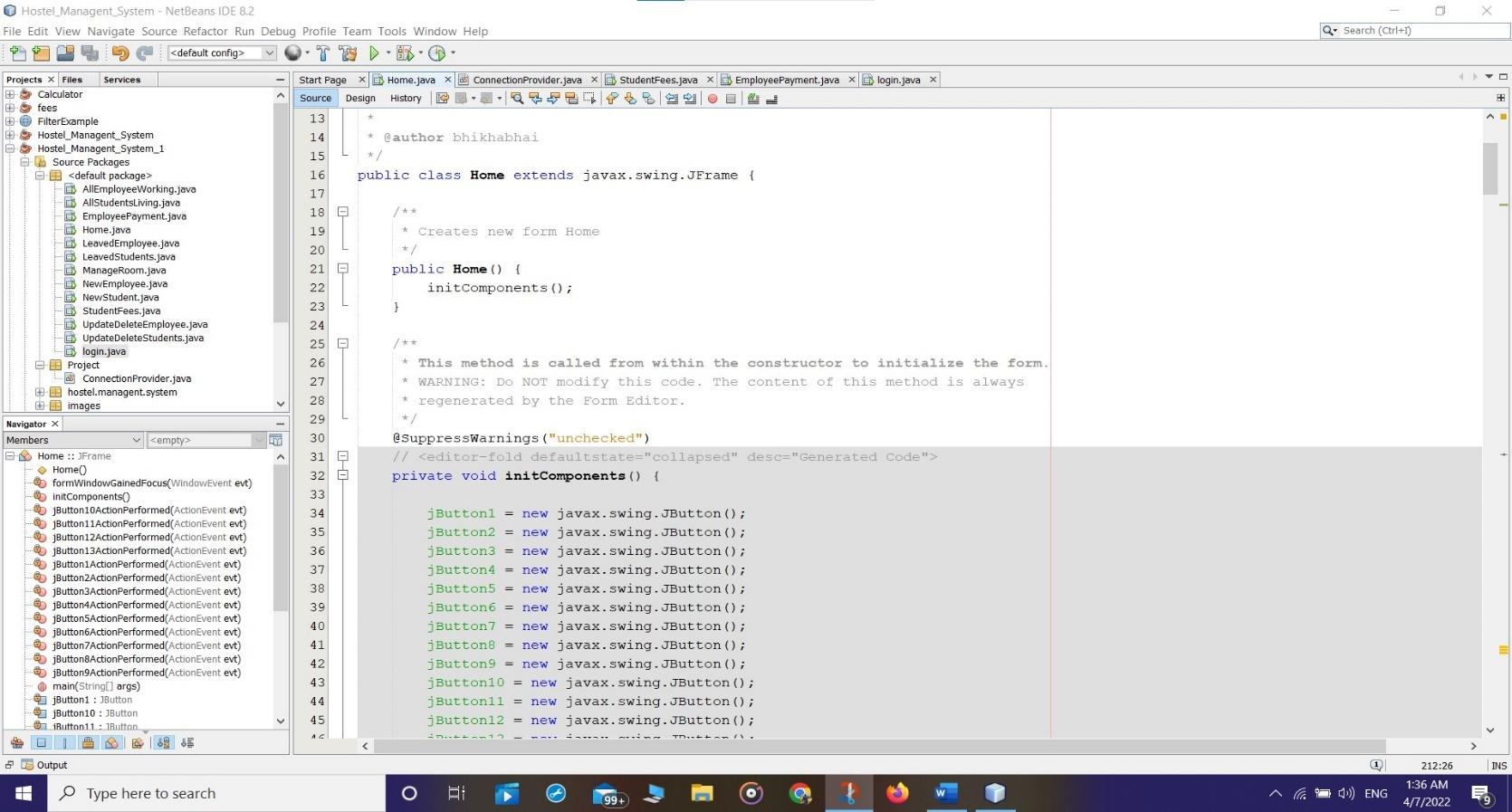
## 6.3 CODING SNIPPETS

While writing our code we took the utmost care to follow the basic coding standards while writing a Java code like,

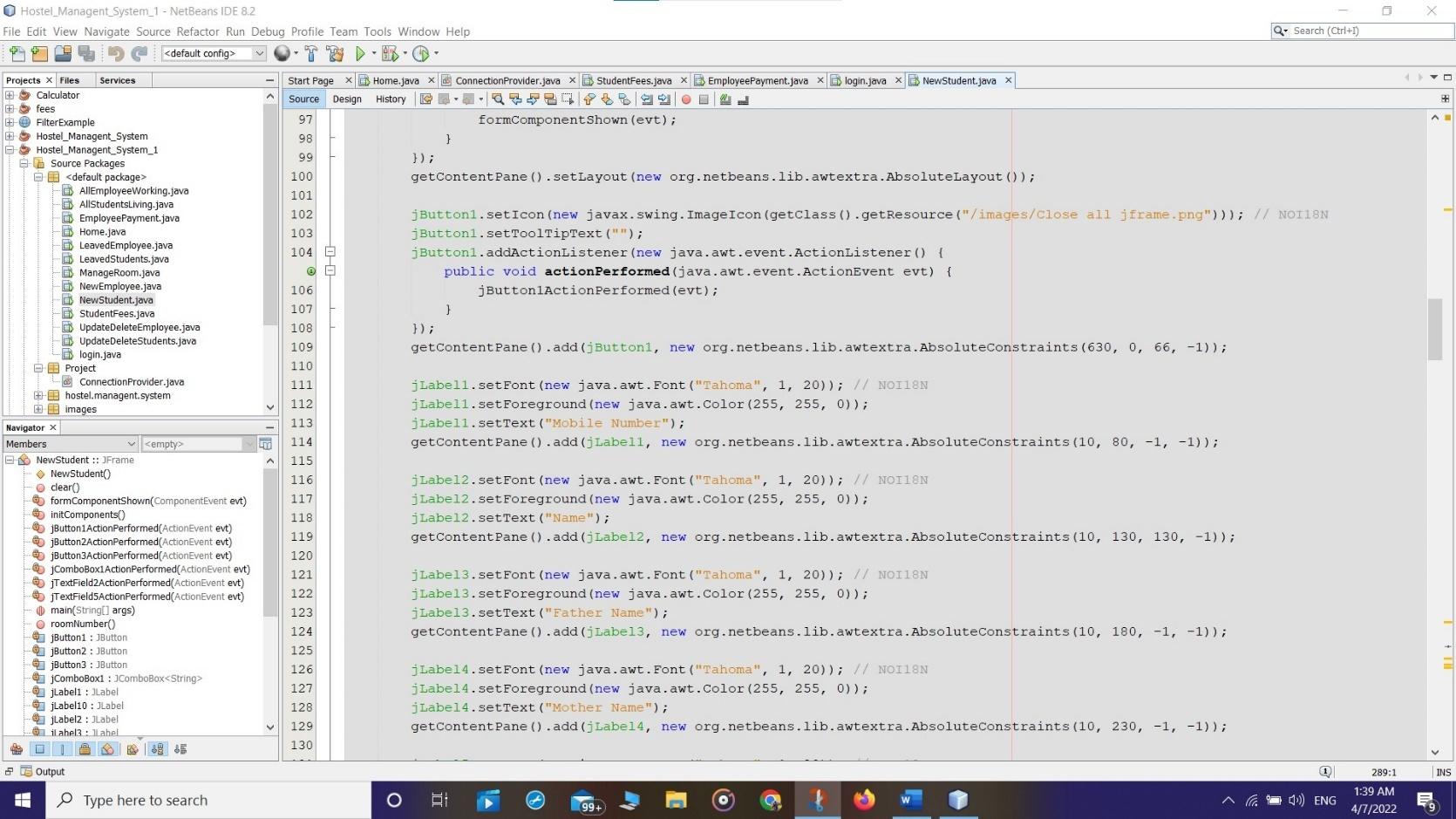
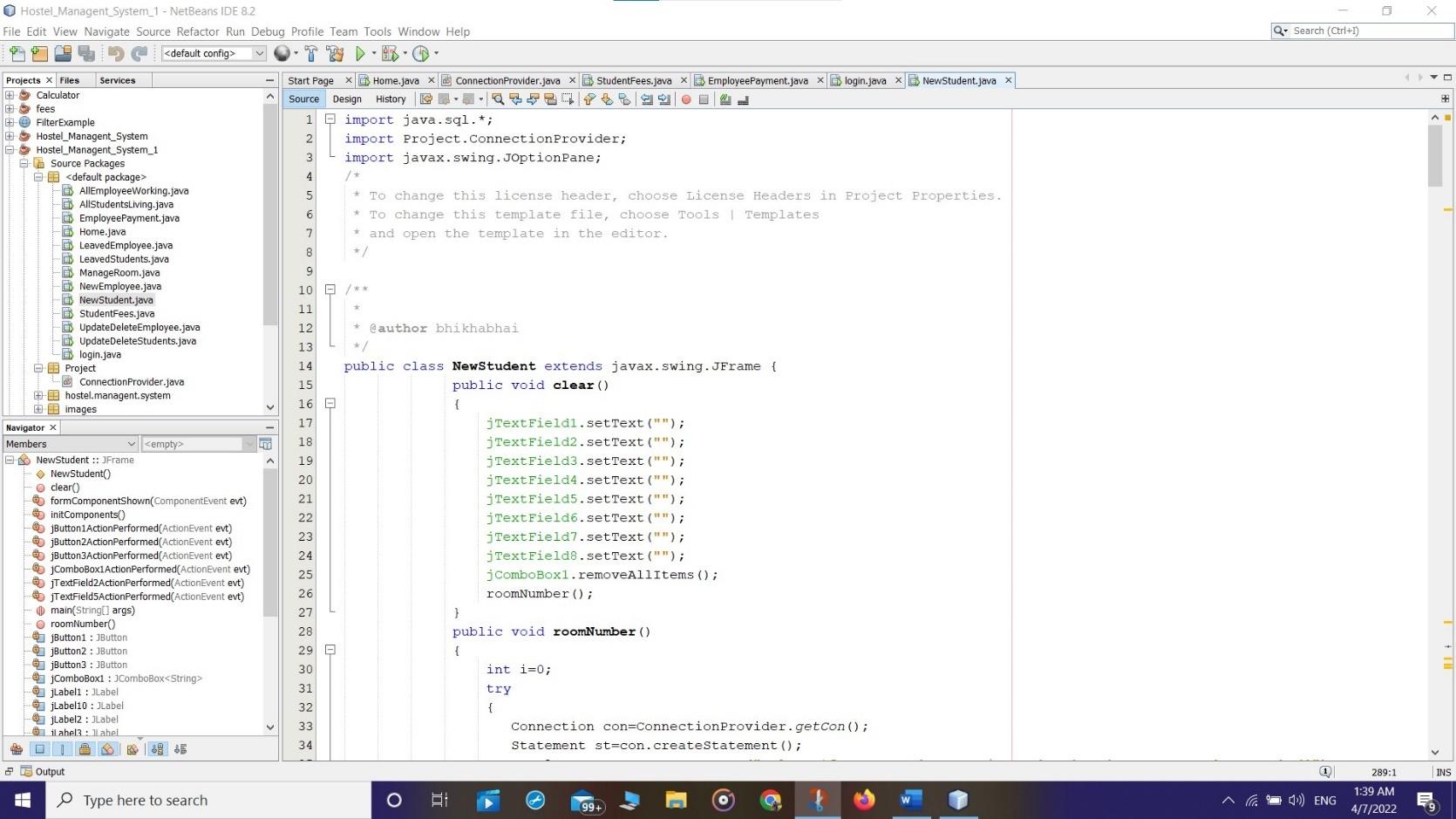
* Following proper naming conventions of local variables, global variables, constants and functions.
* Doing proper indentation.
* Proper error handling.
* Is Powerful
* Adding comments for better understanding.
* Proper Syntax

Here are some of the code snippets written during implementation

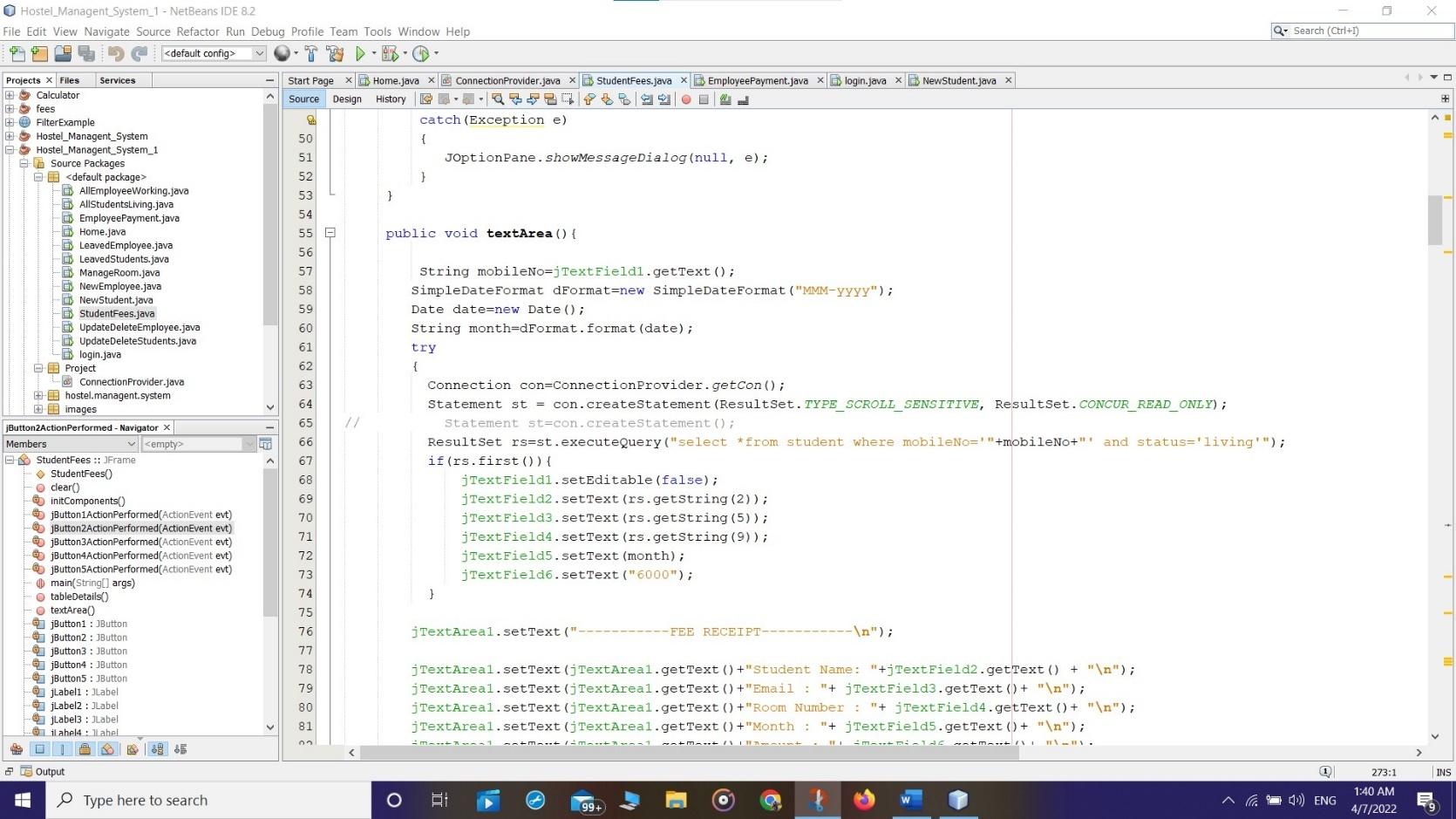
// Home.java

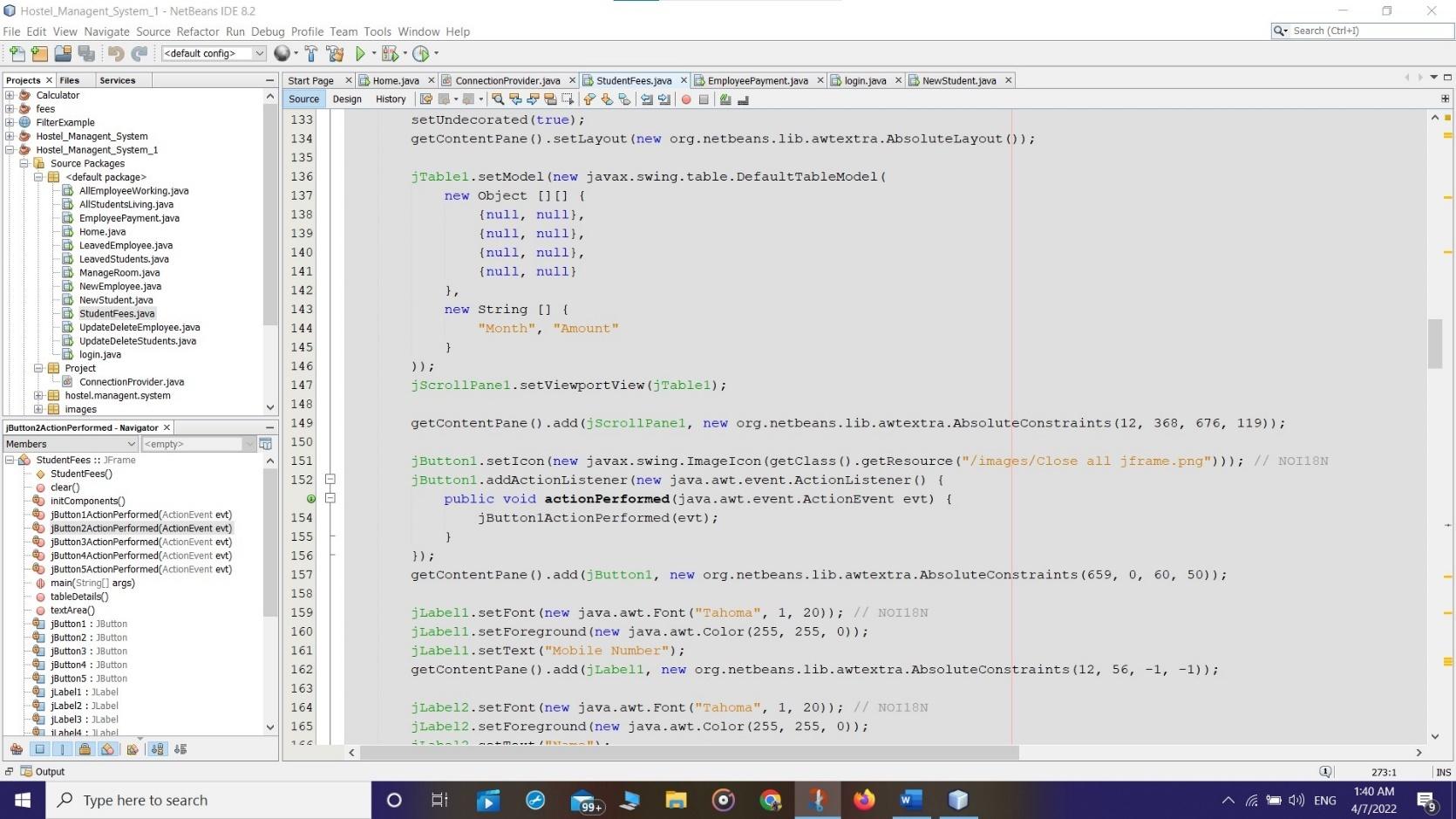


//NewStudent.java



//StudentFees.java





# 7. TESTING

## TESTING PLAN

The method of testing used in this project is White Box testing as in White Box testing, the tester has the complete knowledge and understanding of the code.

## TESTING STRATEGY

The development process repeats this testing sub-process a number of times for the following phases.

* 1. Unit Testing.
  2. Integration Testing

Unit Testing tests a unit of code (module or program) after coding of that unit is completed. Integration Testing tests whether the various programs that make up a system, interface with each other as desired, fit together and whether the interfaces between the programs are correct.

## TESTING METHOD

There are two kinds of testing mainly Blackbox and Whitebox testing:

* + In black-box testing a software item is viewed as a Blackbox, without knowledge of its internal structure or behavior. Possible input conditions based on the specifications (and possible sequences of input conditions), are presented as test cases.
  + In white-box testing, knowledge of internal structure and logic is explored. Test cases are presented such that possible paths of control flow through the software item are traced. Hence more defects than black-box testing is likely to be found.

## TEST CASES

| **ENTITY** | **TEST CASE** | **EXPECTED** | **ACTUAL** | **STATUS** |
| --- | --- | --- | --- | --- |
| **OUTPUT** | **OUTPUT** |
| Login | Validation | Redirect to Home | Redirect to Home | pass |
| Page | page |
| Add room | Adding data | Add room in database | Add room in database | Pass |
|  |  |  |
|  |  |  |
| Delete room | Remove data | Remove room from data base | Remove room from data base | Pass |
|  |  |
|  |  |
|  |  |
| add student | Adding data | Add student in data base | Add student in data base | Pass |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| update student | Update data | Updating student in database | Updating student in database | Pass |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

| add employee |  | Adding data | Adding employee in database | Adding employee in database | Pass |
| --- | --- | --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| update employee | | Update data | Updating employee in database | Updating employee in database | Pass |
|  |  |
|  |  |
|  |  |
|  |  |
|  |  |
| student fees | | Generate report | Generate student’s fees report in database | Generate student’s fees report in database | Pass |
|  |  |
|  |  |
|  |  |
| employee payment | | Generate report | Generate employee’s payment in database | Generate employee’s payment in database | Pass |
|  |  |
|  |  |

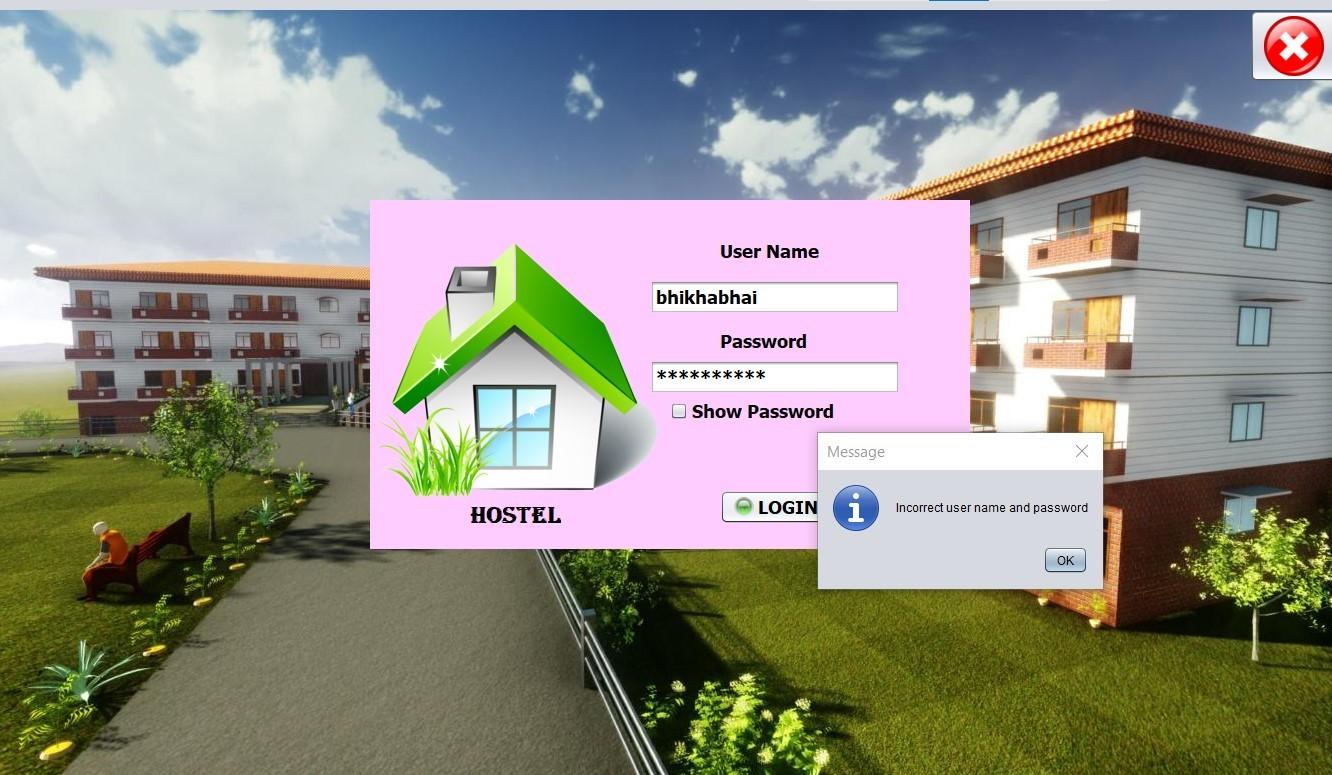
# 

# 8.USER MANUAL

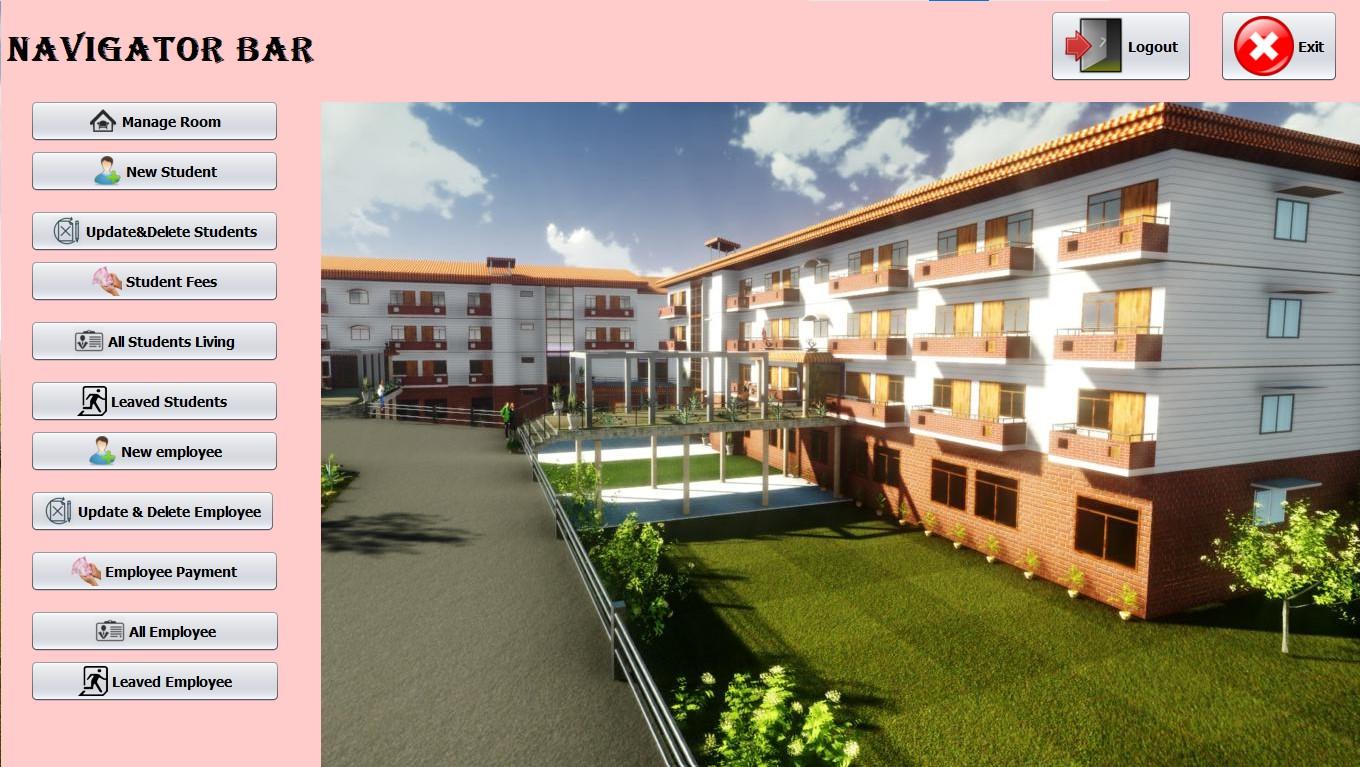
* Home Page



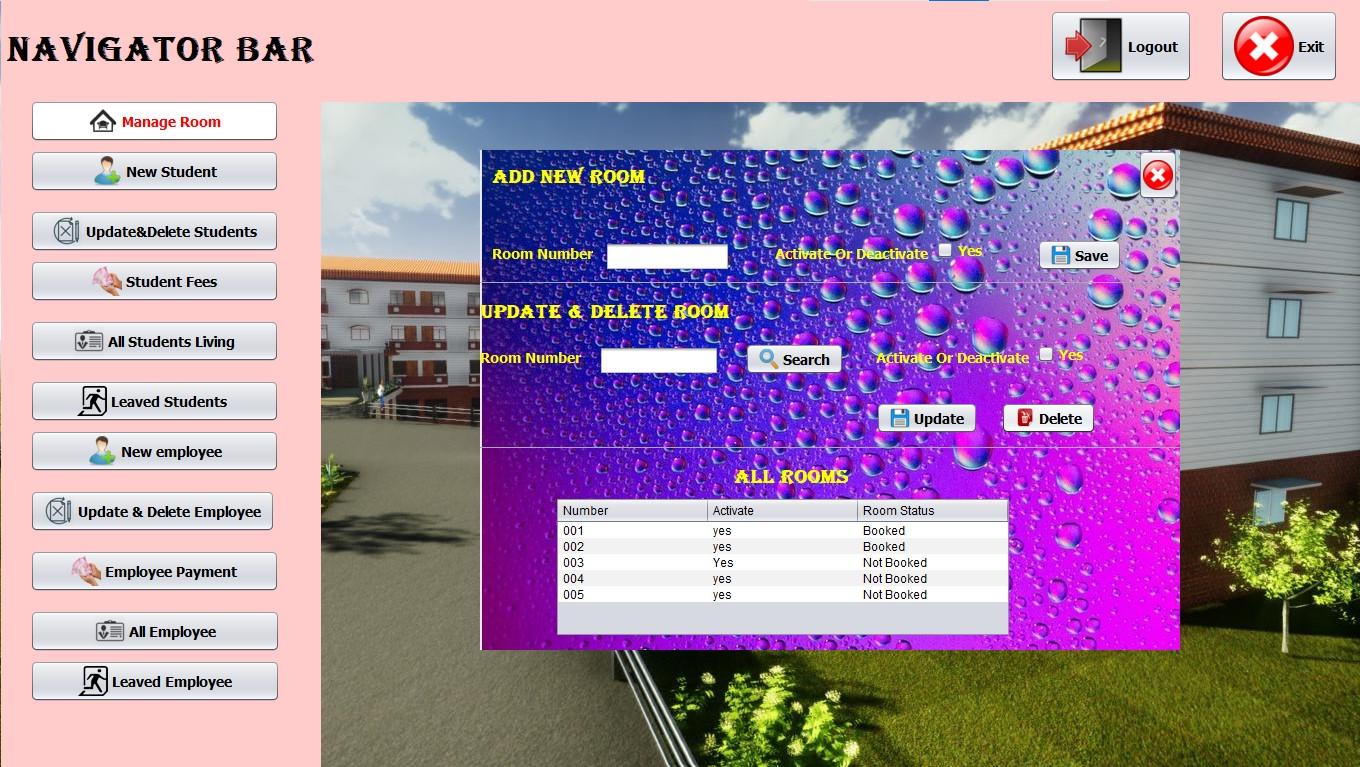
For incorrect username or password



Home Page



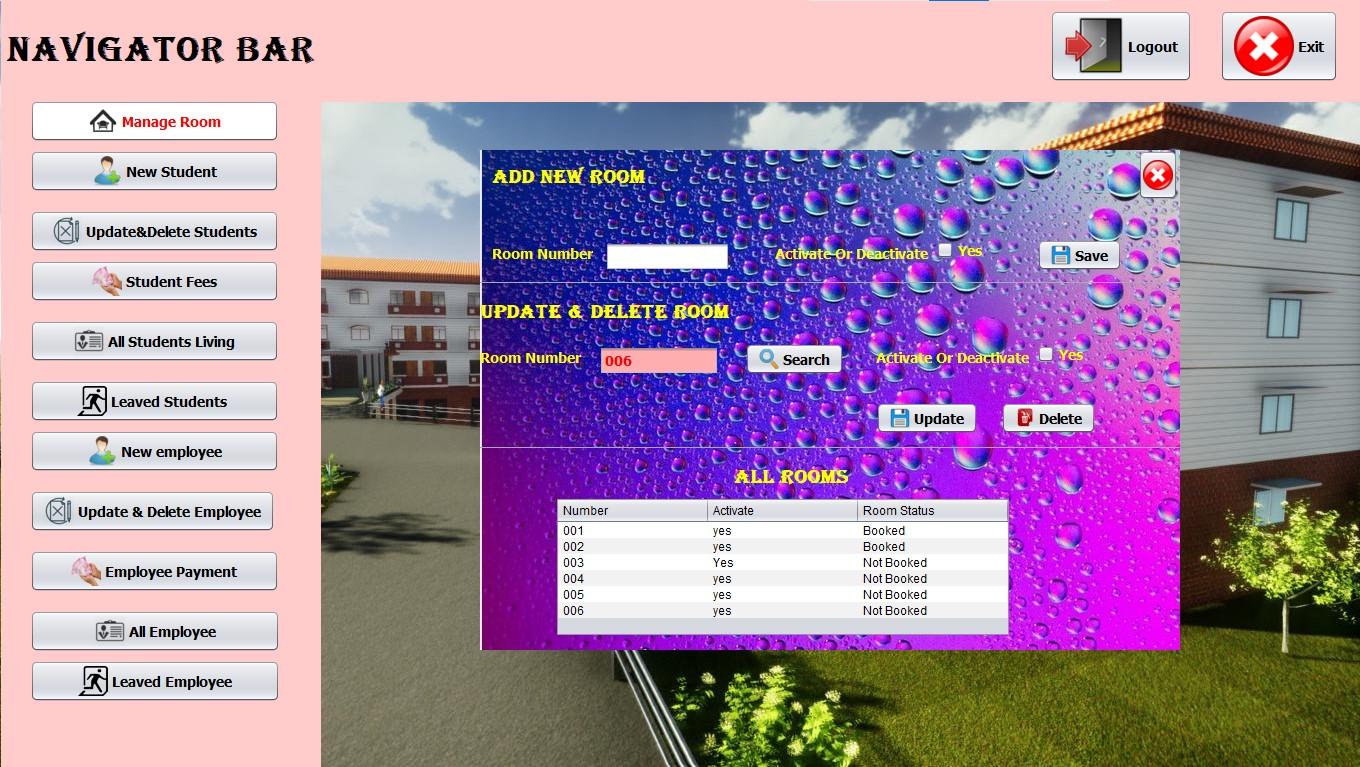
Manage Room



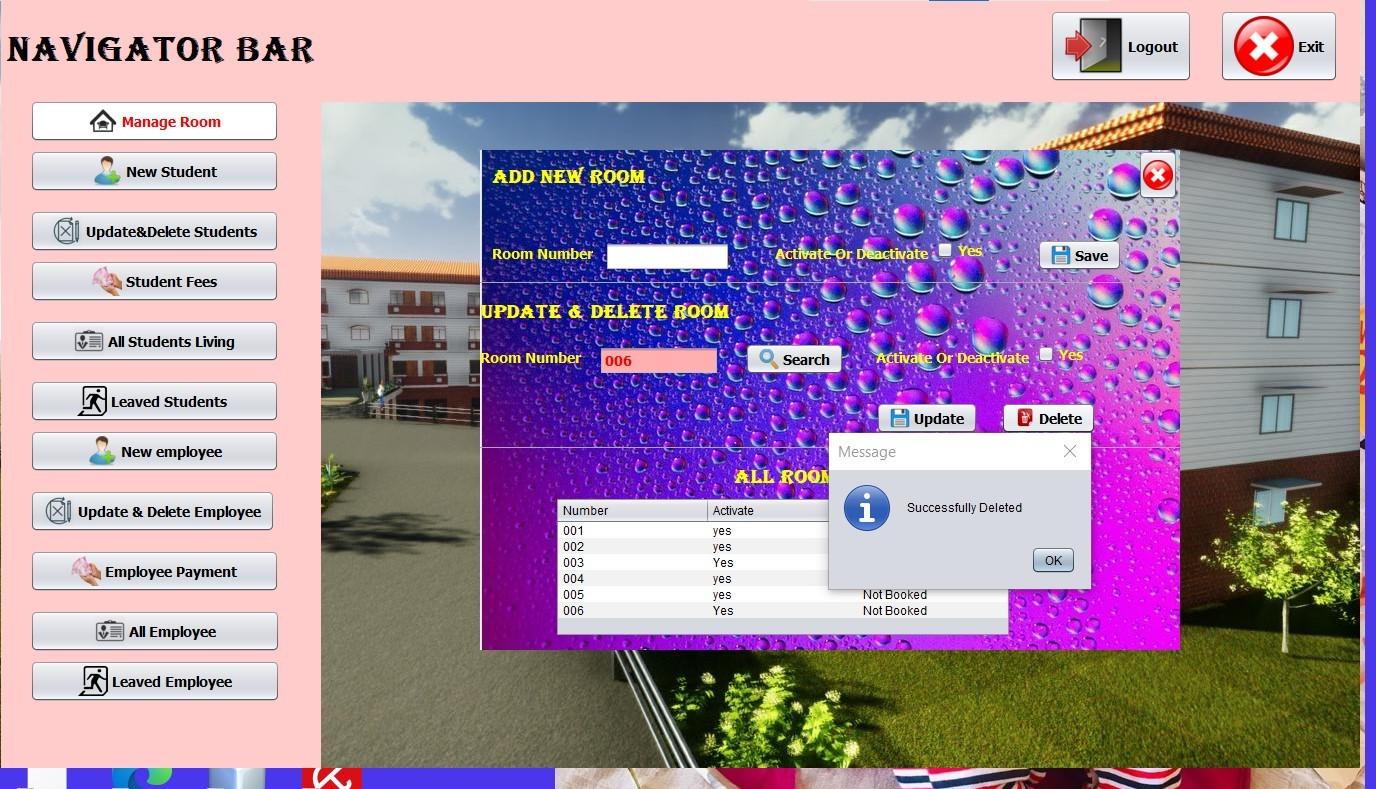
Add Room



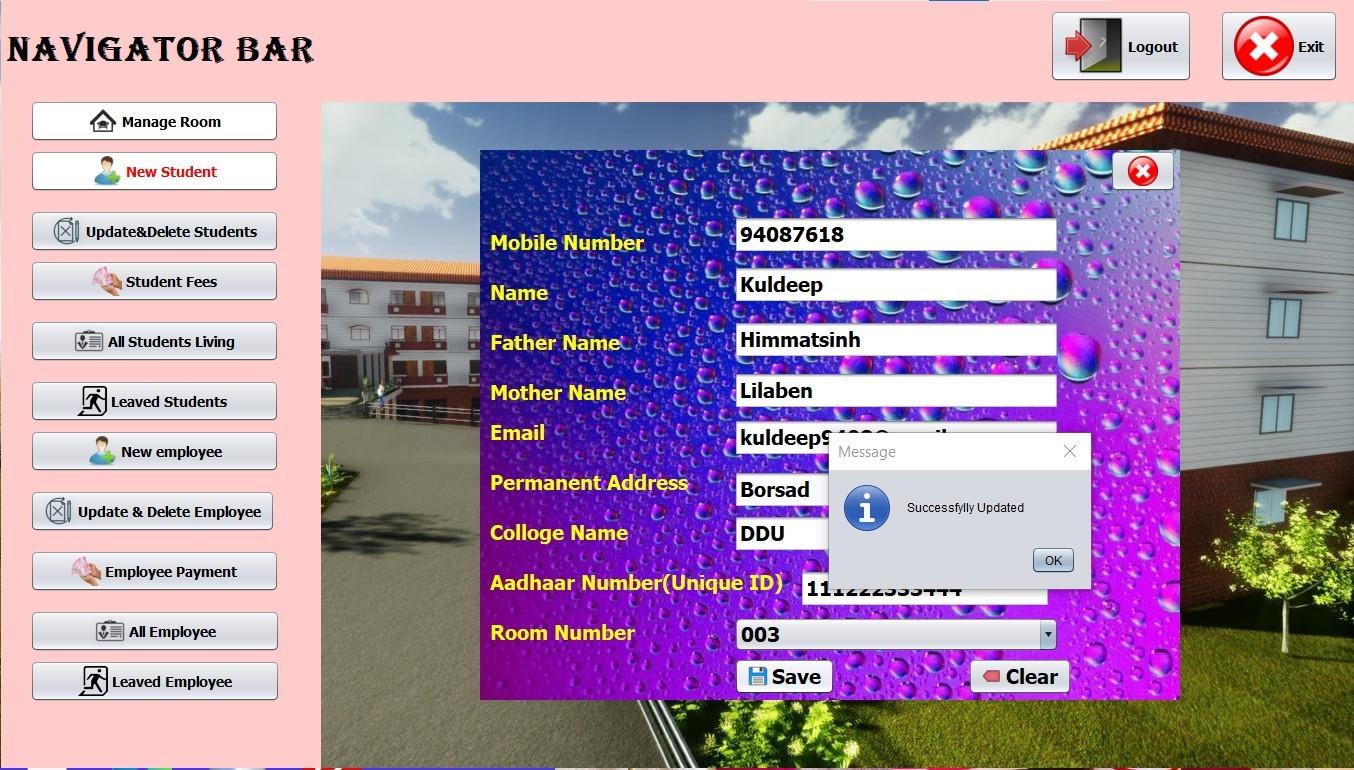
Search Room



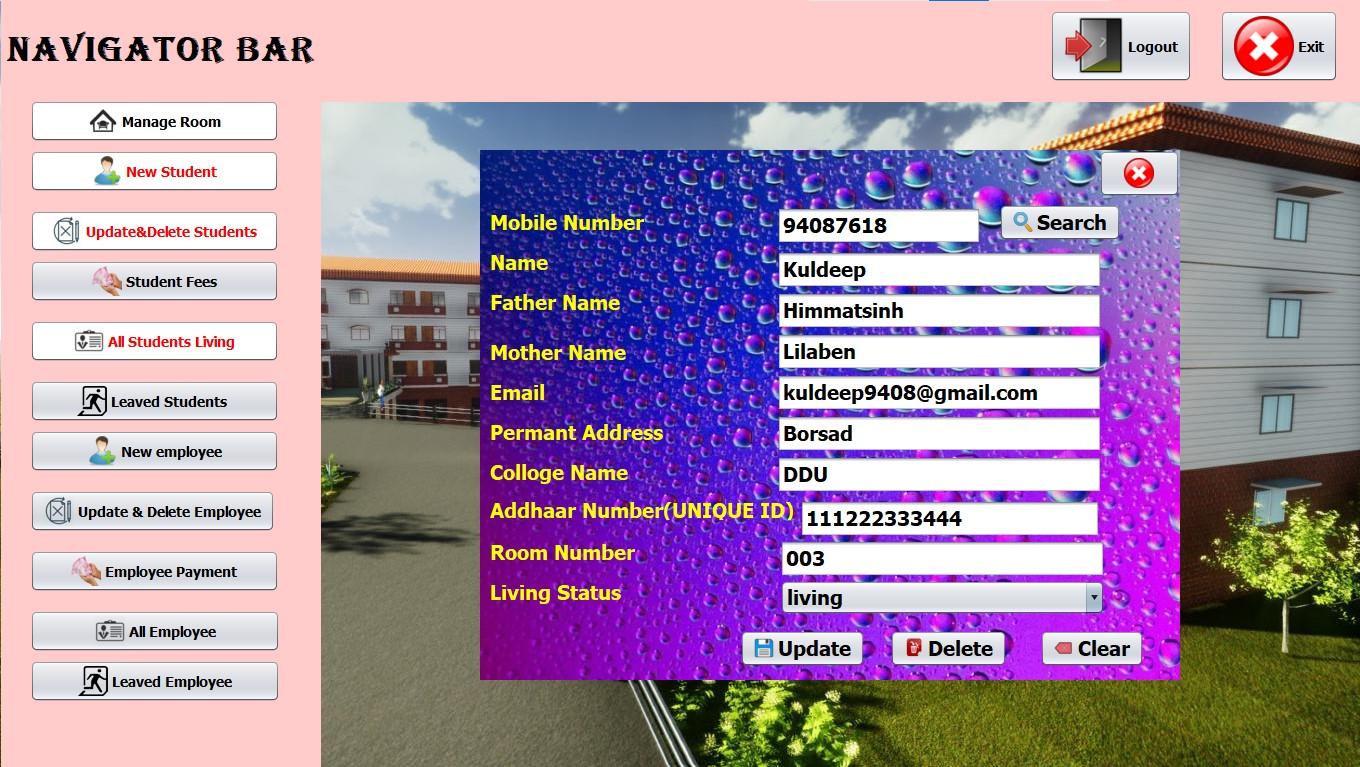
Delete Room



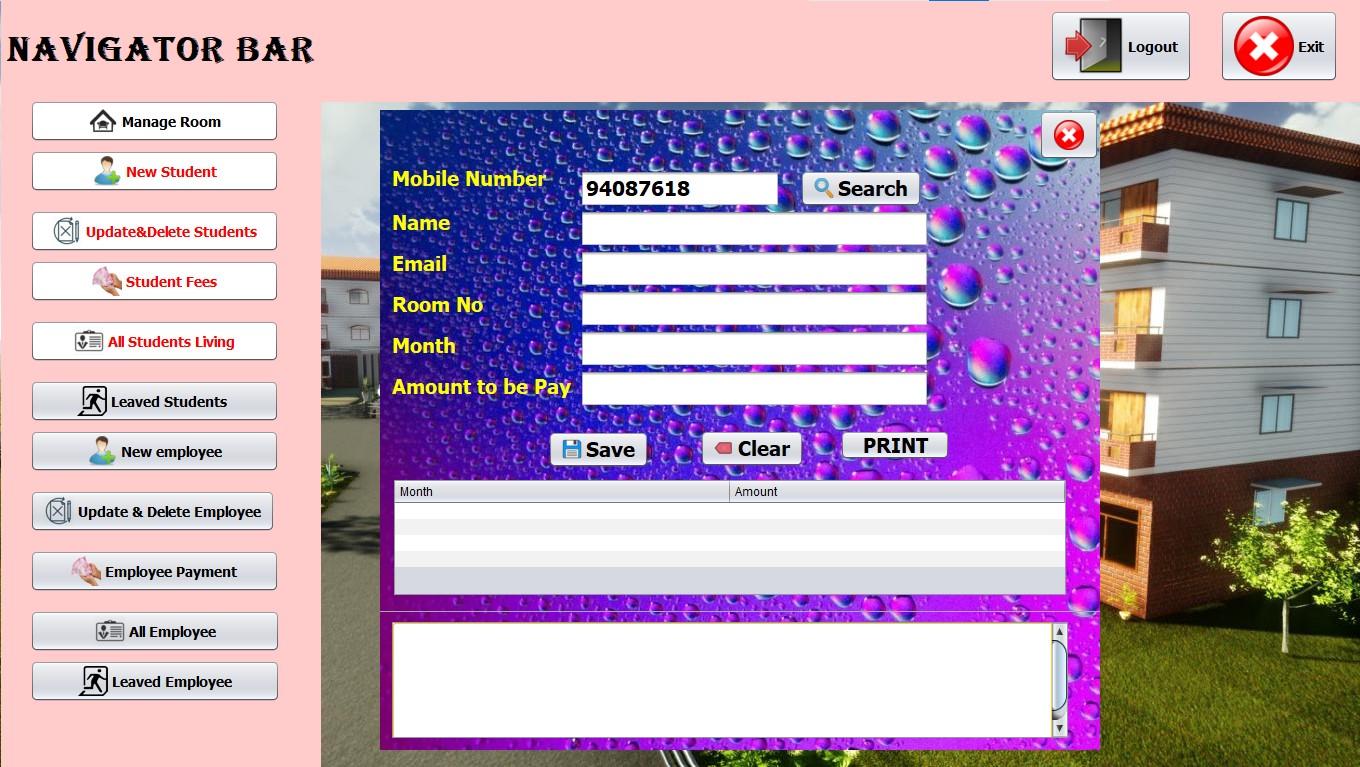
Add New Student



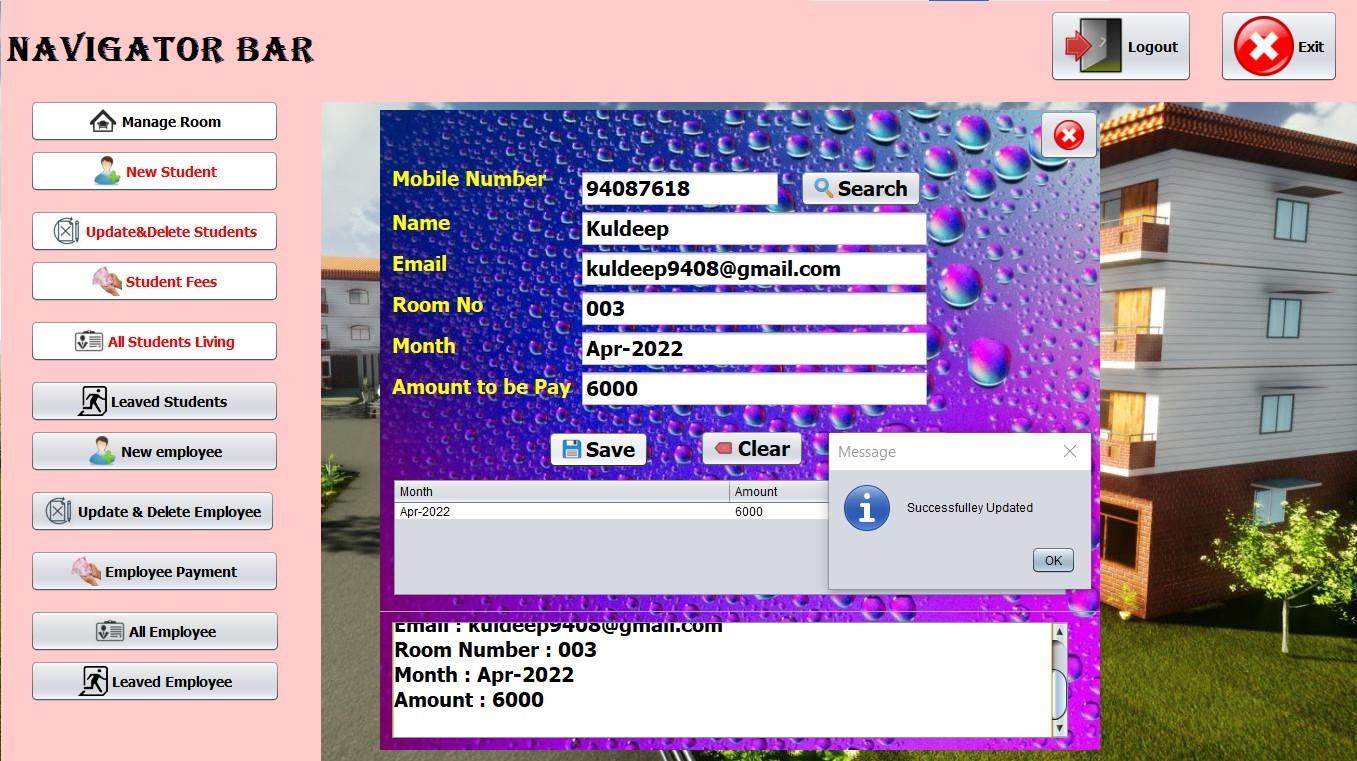
Update And Delete the Student



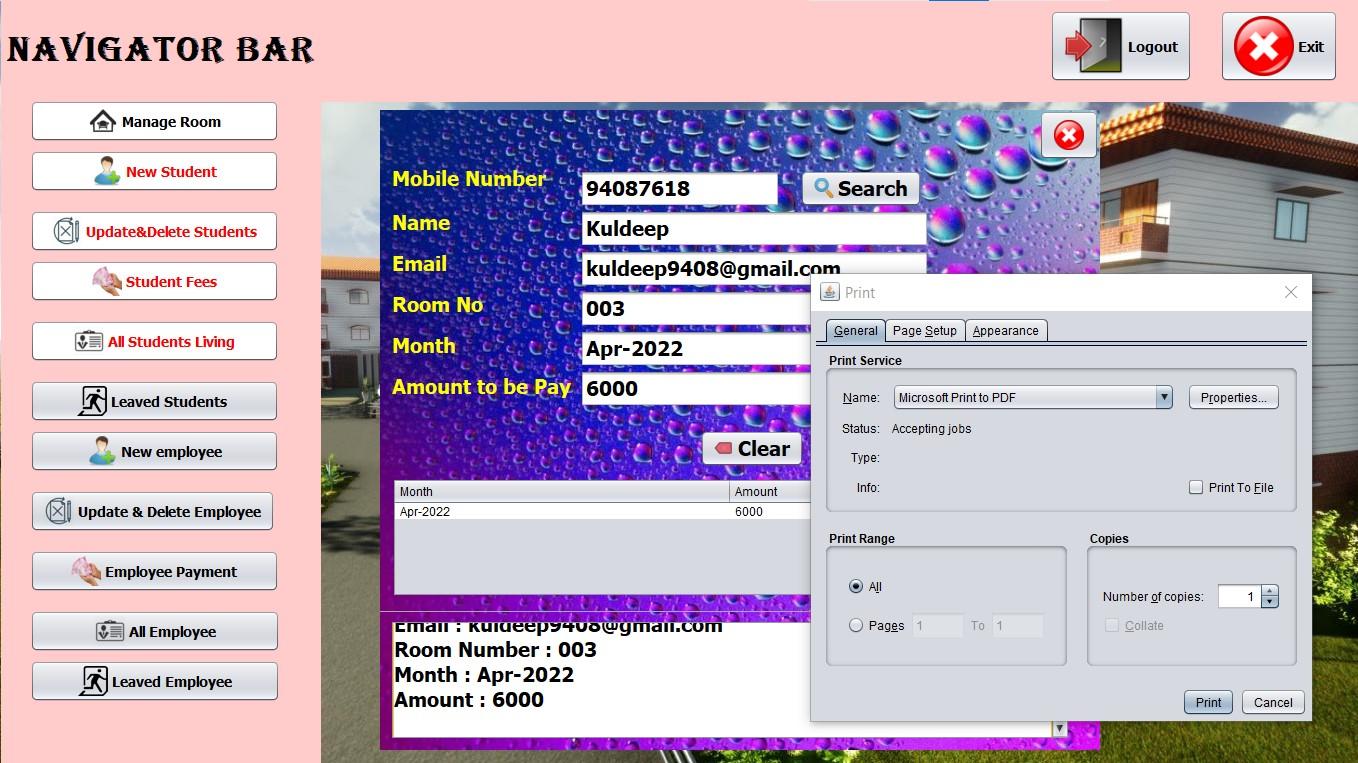
Student Fees



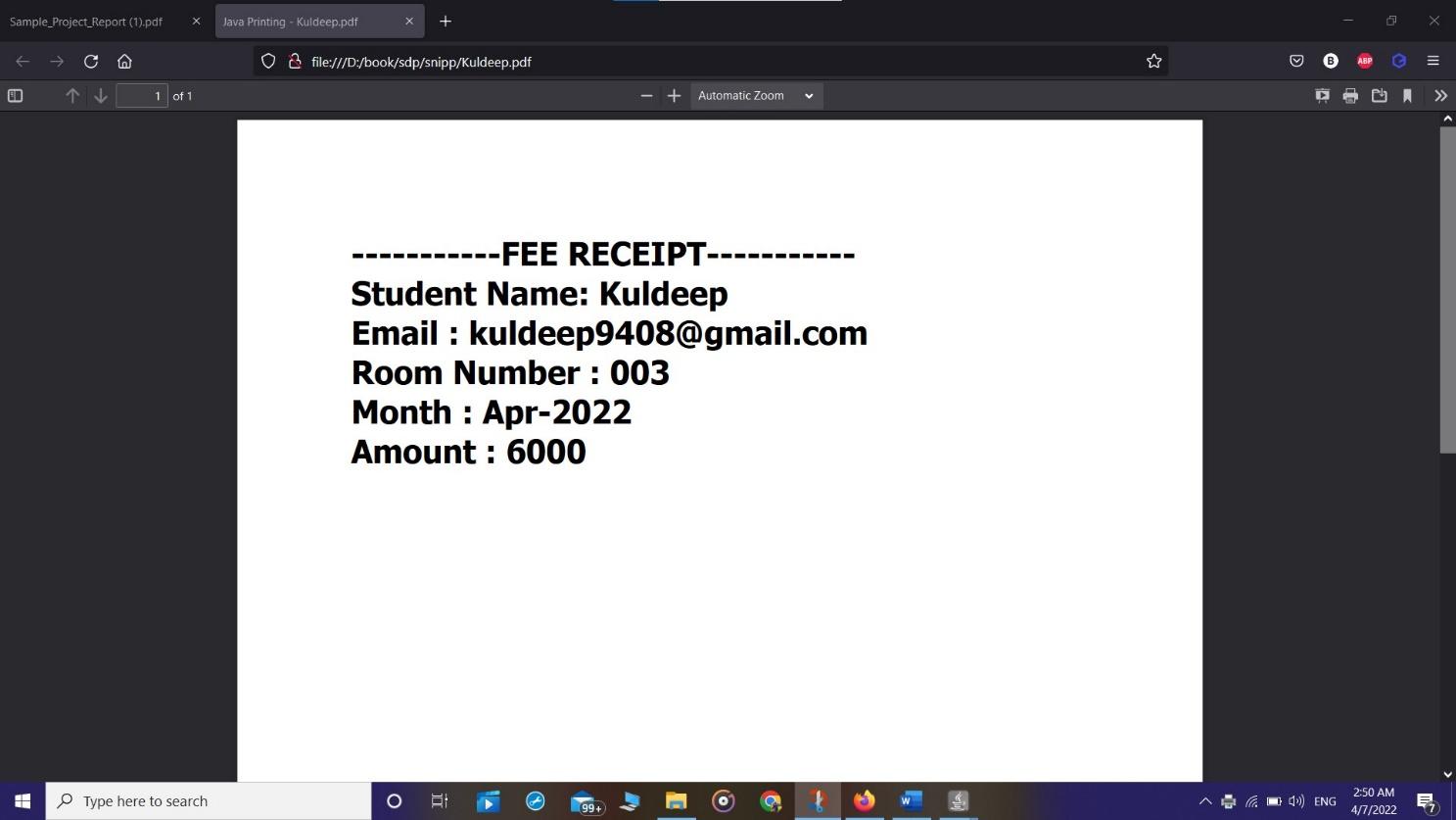
Update Student Fees



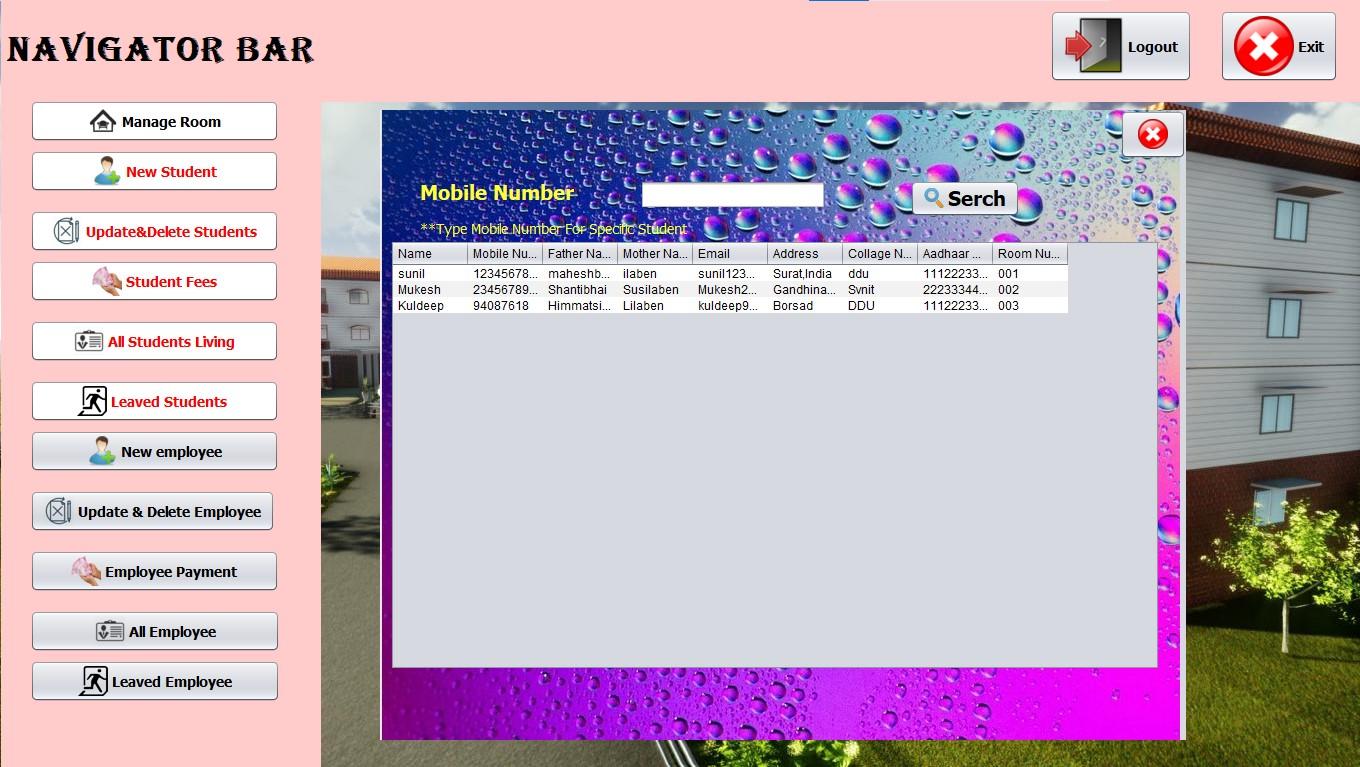
Printing Receipt



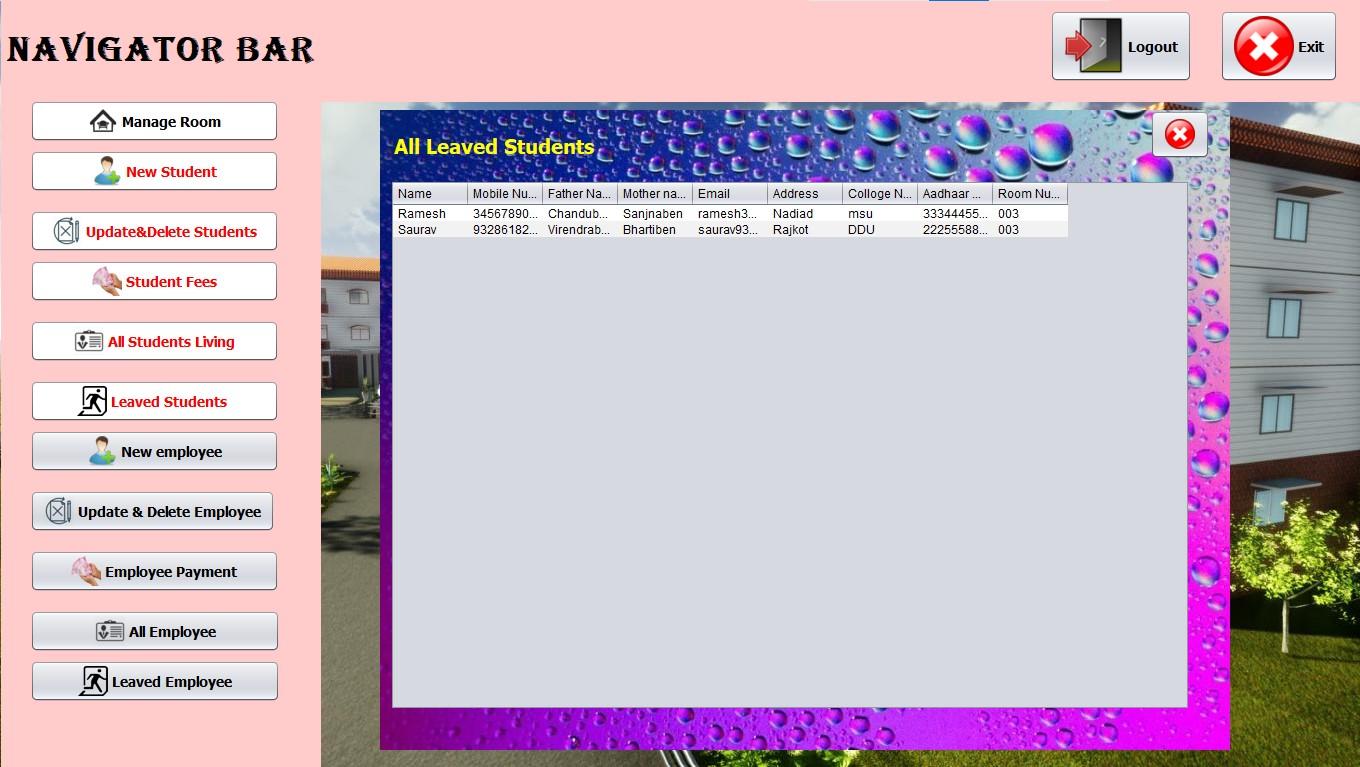
Fee Receipt in pdf form



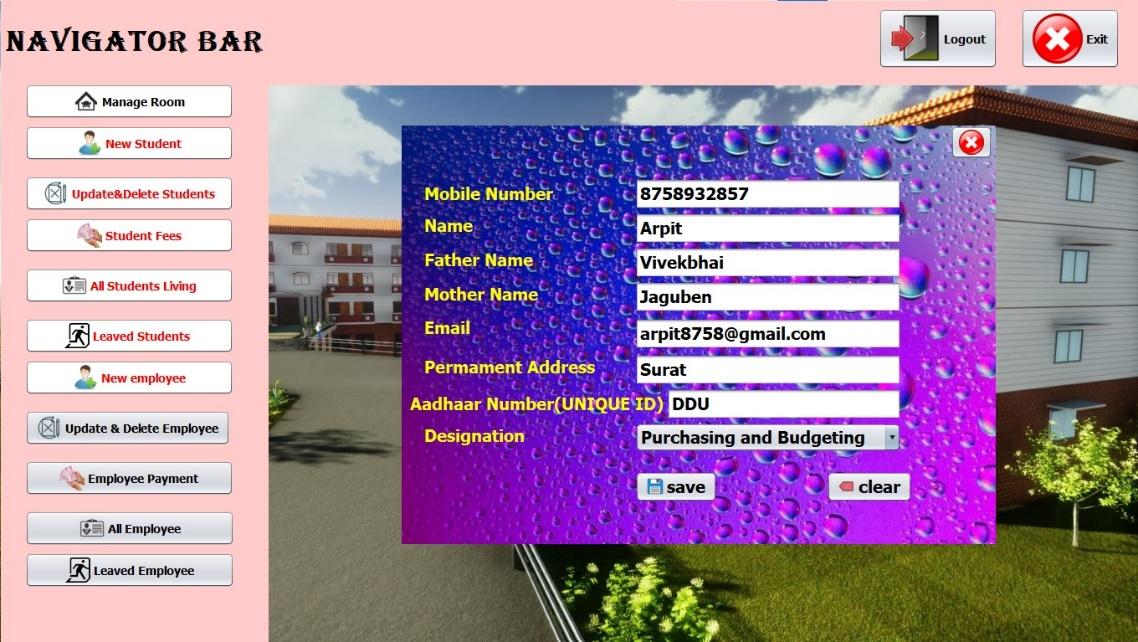
All Student Leaving



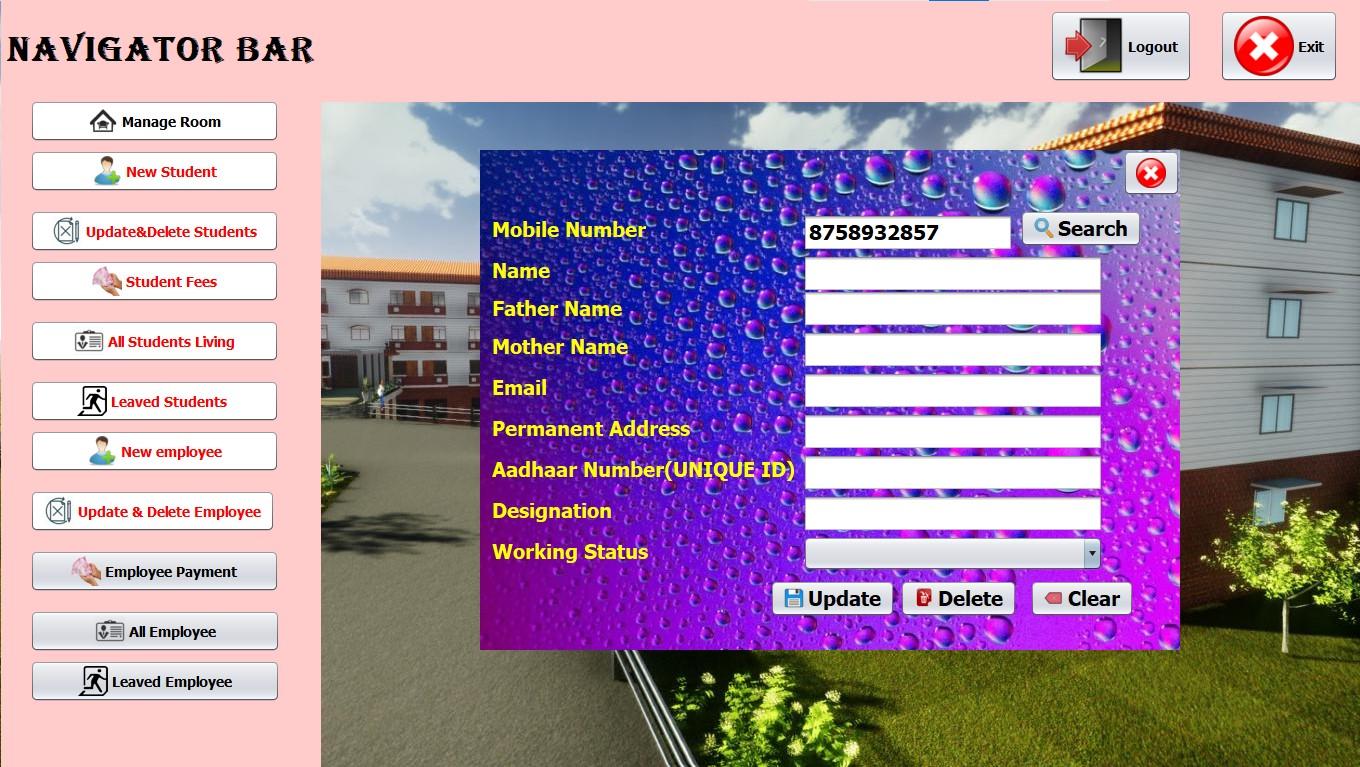
All Leaved Student

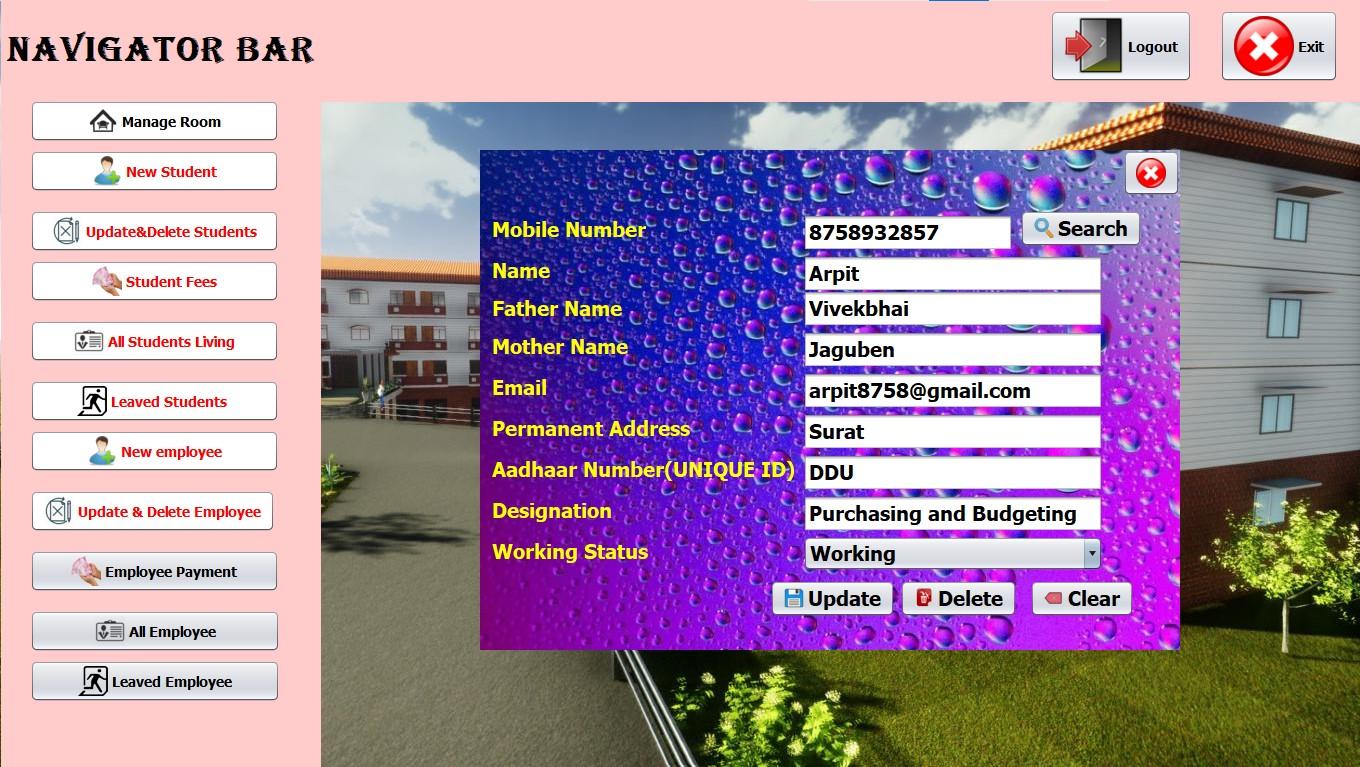


Adding New Employee

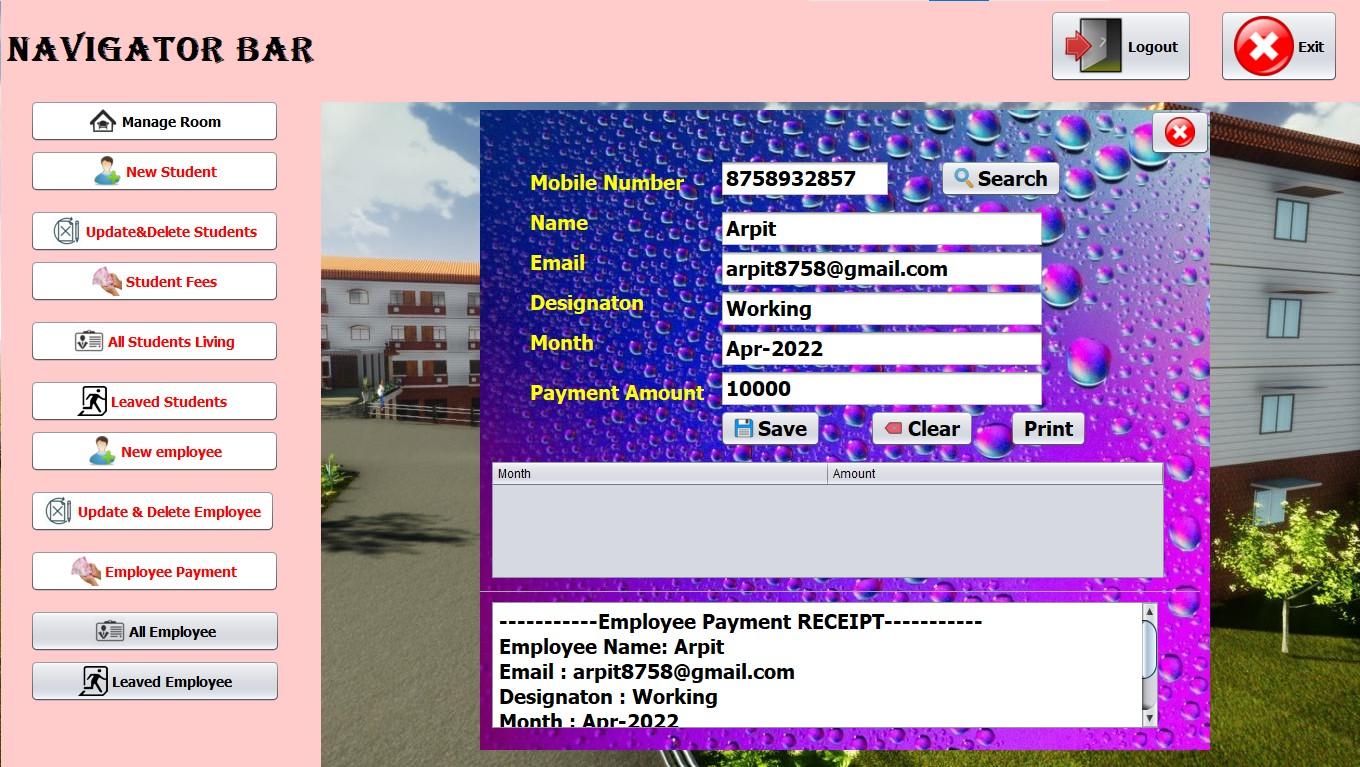


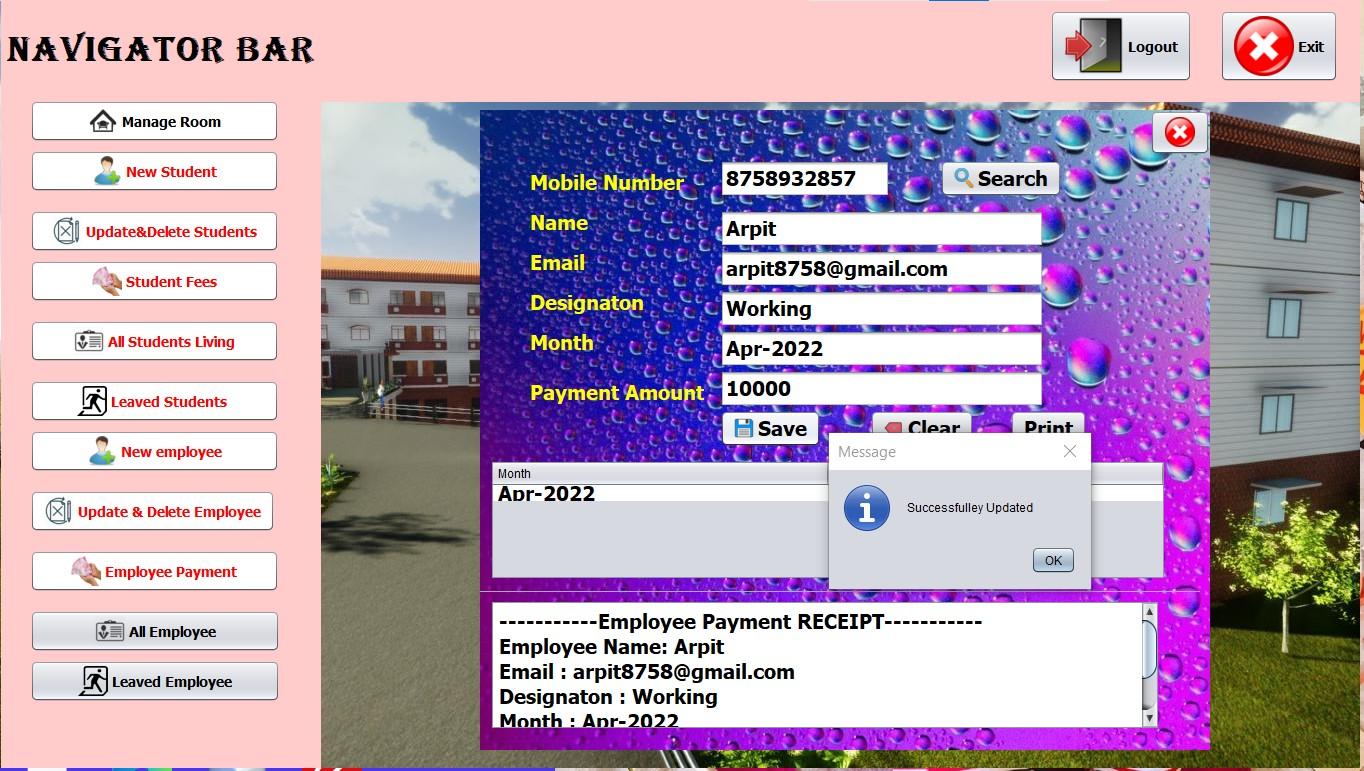
Update And Delete The Employee



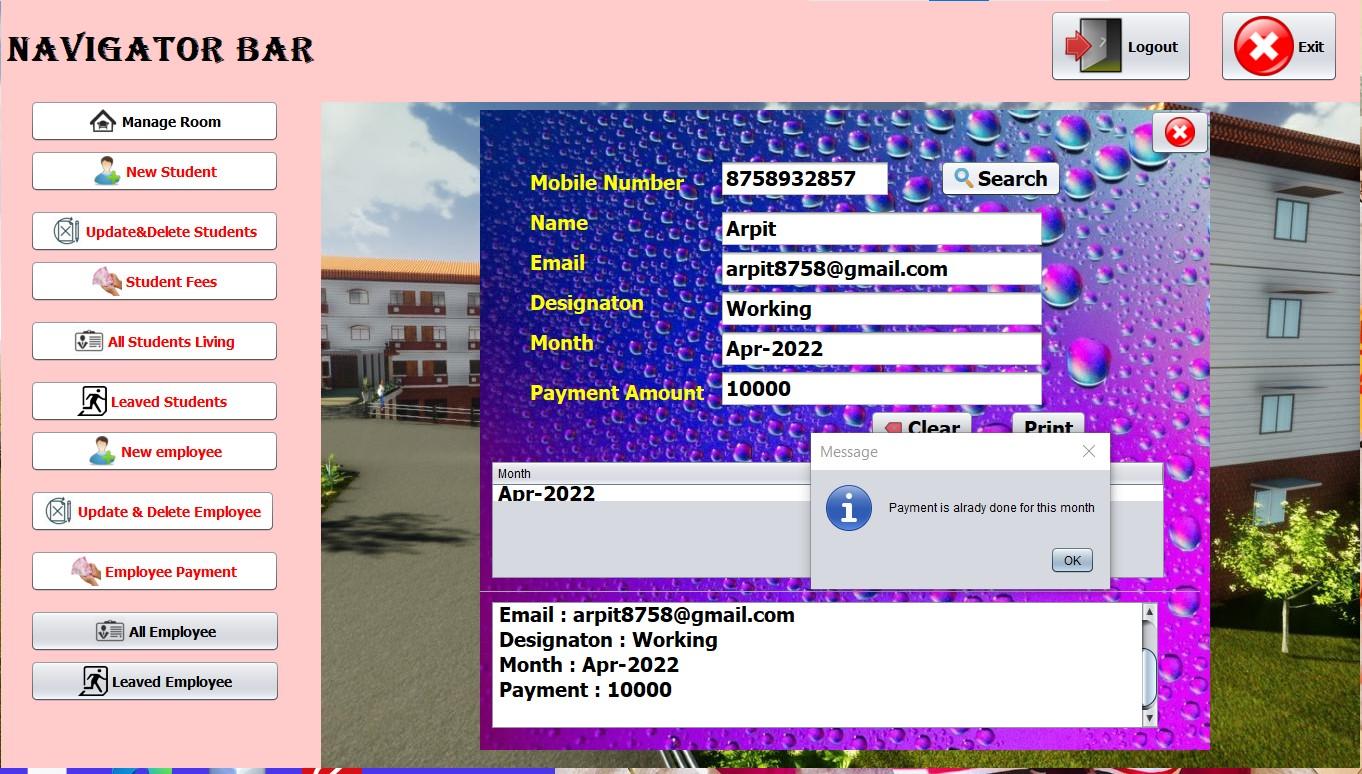


Employee Payment

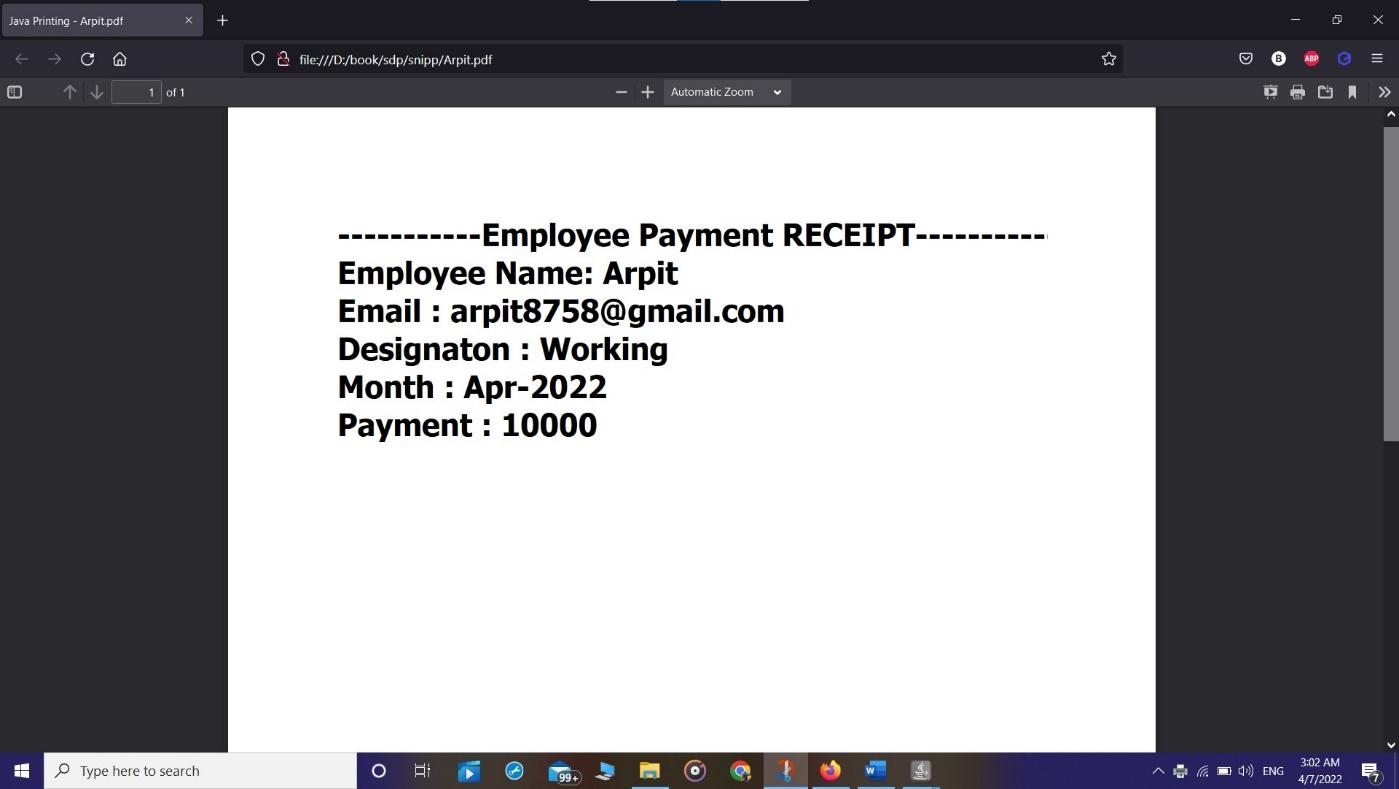




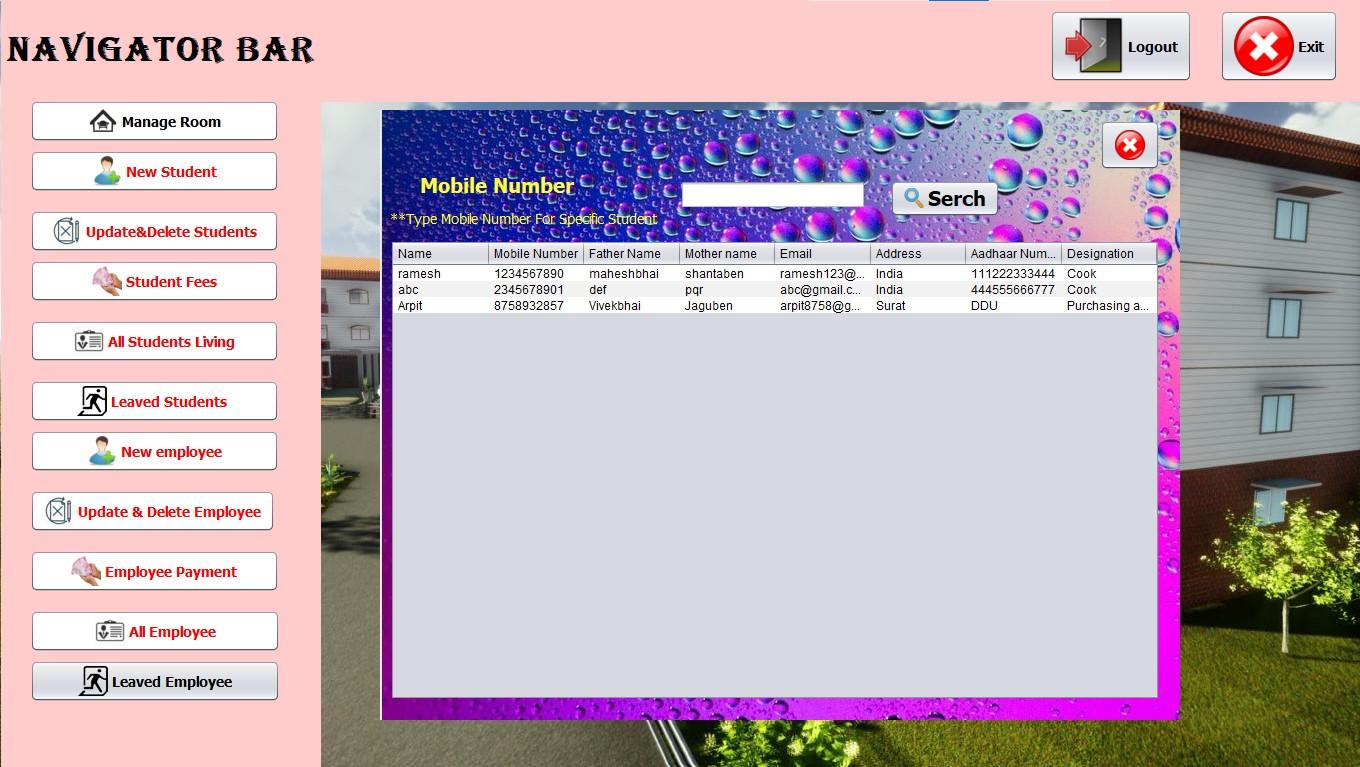
If Payment Is Already Done



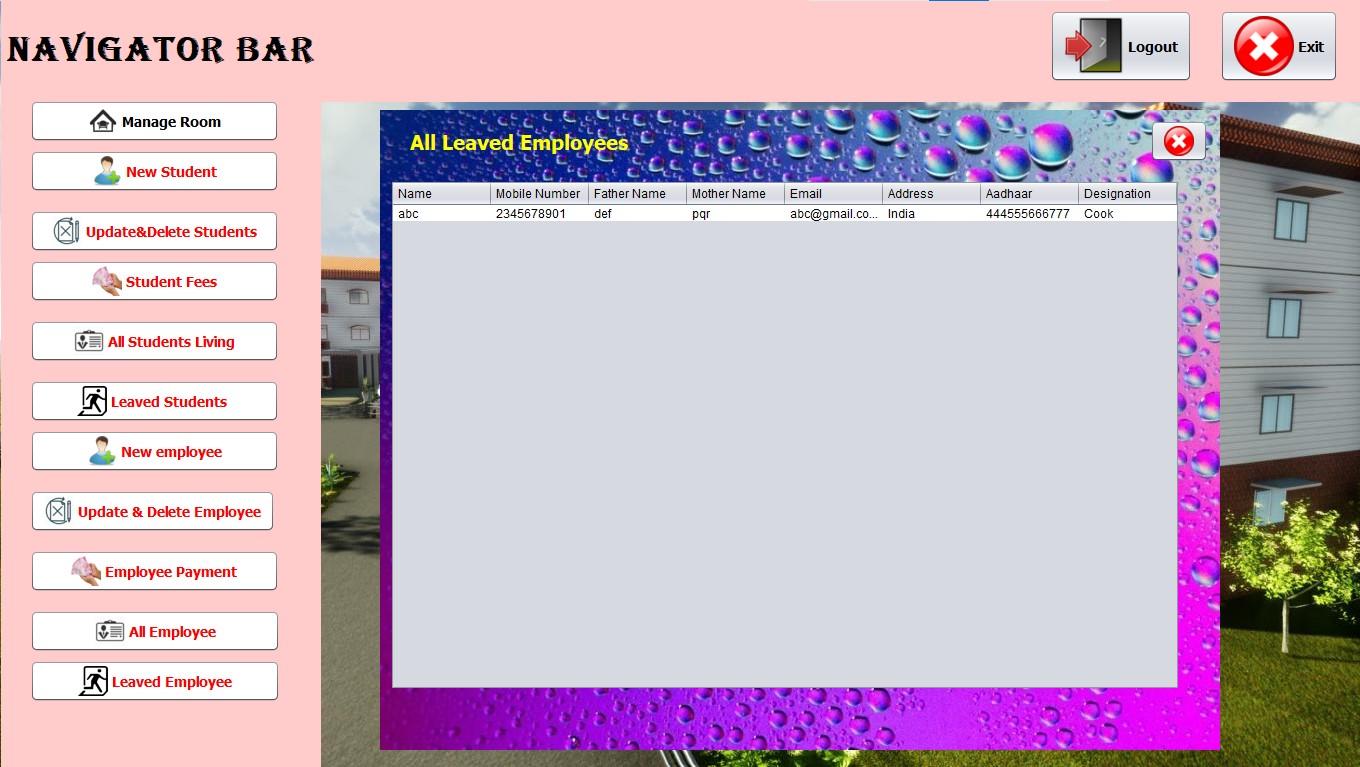
Payment Receipt in pdf



All Employee Working



Leaved Employee



**LIMITATIONS AND FUTURE ENHANCEMENTS**

## LIMITATIONS:

* The system has not able to run online environment.
* The database cannot be converted into centralized.
* Multiuser are not supported in the system.
* Mess management is not provided in this system.
* Attendance management is not included in this system.
* No role for the students in this system.

## FUTURE ENHANCEMENTS:

* Improving the efficiency of the website.
* Adding the attendance management feature.
* Adding the mess management feature for easily handle the mess.
* Keep on the changing on UI-Design to make the application more attractive.
* Add multiuser feature.

# CONCLUSION AND DISCUSSION

## CONCLUSION

* We provide ease to manage hostel
* We have tried to keep the UI alluring and easy enough for our users to feel comfortable

## DISCUSSION

## Self-Analysis of Project Viabilities:

In our opinion, this project has served the goal that we set when we started. As java as platform independent this application can work on all platforms.

## Problems encountered

There were so many problems encountered during this project: -

1. Problem to maintain databases and change them often.
2. Need to change some functionality fully which leads to doing the whole work again.
3. Many instances where the errors weren’t understandable.

## Summary of Project Work:

We have completed our project work using software engineering, system analysis and design approach. We have done our work with planned scheduling pertaining to time constraints and result oriented progress in project development. We have been given an industrial exposure which is very beneficial for our field. This project will be able to manage the hostel and can be further modified to hotel management. As it is platform independent, can be run in all environments.

# 11. BIBLIOGRAPHY

1. YouTube Channels - To learn all the related technologies

**YouTube:** [**https://youtube.com/playlist?list=PLNHw\_0qv1zycVdEX6HqJX2a8ZLHbG3Ox**](https://youtube.com/playlist?list=PLNHw_0qv1zycVdEX6HqJX2a8ZLHbG3Ox)

2. MySQL Documentation - For all the knowledge related to MySQL

YouTube: <https://youtu.be/7S_tz1z_5bA>

3. Stack Overflow - For Debugging

<https://stackoverflow.com/>

4. Convert JRE to .EXE – Convert JRE file to .EXE file format

YouTube: <https://youtu.be/2QNGdKdWSyM>