Final Project Review

Software Engineering

Course code: CSE3001

Project Title:

Hostel Management System

By

ROHAN ALLEN 18BCI0247

RAKSHITH SACHDEV 18BCI0109

HRITHIK AHUJA 18BCE2154

Abstract:

Over the years as more and more students are joining college the strength of the hostels has increased dramatically. Hostel Administration can be a tedious task as the hostels are responsible for the safety and welfare of students. Hostels records were usually kept in large record books making them very difficult and heinous to access. Hostel administration would take forever and ensuring each student is allotted a room is a challenging task. There are many complaints from students about the ineffective and obsolete hostel allotment system. With our Online Hostel Management System, we aim to increase the efficiency of this process so that every student will get allotted room without having to go through a long and difficult process. Computerizing this process will also ensure no errors will occur thereby satisfying students, hostel wardens and all parties involved.

Introduction:

PROJECT OBJECTIVES

• Maintain the students as hostellers and waiting list students separately

• Process allotment list.

• Admin can send the approval notification to every approved student via email.

• Automatically insert student's details to the hosteller's record when the allotment is confirmed by the admin and deleted when vacation is conformed or after the course end date.

• Students can register their complaints.

• Admin can edit notice board and each student can view it.

• Hostel secretary can calculate hostel fee including mess fee and can edit mess menu

• Hostellers can check the status of every month's hostel fee.

Literature Survey/Related Work:

EXISTING SYSTEM

The existing system is manual based and need lot of efforts and
consume enough time. In the existing system we can apply for the hostels online
but the allotment processes are done manually. It may lead to corruptions in the
allocation process as well as hostel fee calculation. The existing system does not
deals with mess calculation and complaint registration.

DISADVANTAGES:

- More human power
- More strength and strain of manual labour needed
- Repetition of same procedure.
- Low security.
- Data redundancy.
- Difficulty to handle.
- Difficulty to update data.

Record keeping is difficult.

- Backup data can be easily generated.

Proposed Model:

PROPOSED SYSYTEM

Alternative Solution to the Problems Identified The alternative solution to these problems is development of an online hostel management system; the hostel management system will eliminate the problems encountered in the manual system. If implemented, it will play a great role such as:

1. Increase efficiency: the computerized system formulates accurate efficiency, faster and effective way of processing hostel activities, with the intervention of computer.

2. Storage: the new system provides a better means of information storage, all records related to hostel are stored on a centralized database and encrypted to avoid unauthorized access.

3. Error free: the new system with the computer intervention in processing, errors will be avoided or eliminated.

4. Speed: the new system offers the students affairs officer and the management an opportunity to retrieve and sort files in the shortest possible time compared to the manual method.

5. Reliability: delay is completely faced out on the retrieval of record about hostel using the computerized system.

Architectural design for the Project

The aim of the software modularization process is to partition a software system into subsystems to provide an abstract view of the architecture of the software system, where a subsystem is made up of a set of software artifacts which collaborate with each other to

implement a high-level attribute or provide a high-level service for the rest of the software system. However, for large and complex software systems, the software modularization cannot be done manually, owing to the large number of interactions between different artifacts, and the large size of the source code. Hence, a fully automated or semi-automated tool is needed to perform software modularization.

List of Modules Identified
1. Administrator module.
2. Student module.
3. Secretary module.

Module Description 1.

1. Administrator module:

In administrator module administrator manages the master data's like server details and student details. Accept the application of students, view the application forms, reject the fake applications, view the complaints of the students in the hostel, accept the vacating form and delete from the database, edit the notice boards and view complaints.

2. Student module:

In student module, they can Submit application form, change password, can check status, view notice board, view monthly hostel fee and submit the vacating form.

3. Secretary module:

In secretary module, the secretary can calculate the mess bill, and edit the mess menu, view the notice board and also change the password. Module Specifications • Software Configuration: A major element in building a system is the section of compatible software since the software in the market is experiencing in geometric progression. Selected software should be acceptable by the firm and one user as well as it should be feasible for the system. This document gives a detailed description of the software requirement specification. The study of requirement specification is focused specially on the functioning of the system. It allows the developer or analyst to understand the

Every software developed is different and requires a suitable SDLC approach to be followed based on the internal and external factors. Some situations where the use of Waterfall model is most appropriate are:

- Requirements are very well documented, clear and fixed.
- Product definition is stable.
- Technology is understood and is not dynamic.
- There are no ambiguous requirements.
- Ample resources with required expertise are available to support the product.
- The project is short.

And our project of Hostel Management System clearly satisfies all these situations and hence it is one of the most appropriate models to use for our project.

Architectural Design (continuation):

Architectural design represents the data structure and program components that are required to build the computer-based system. It considers the structures and properties of the components that constitute the system and the relationship that exists between all architectural components of the system.

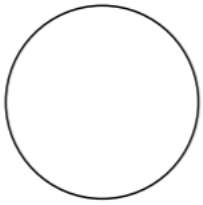
Process Design:

1. Process design plays an important role in project development. In order to understand the working procedure, process design is necessary. Data Flow Diagram (DFD) is the tool used for process design. Data Flow Diagram is the logical representation of the data flow of the project.

2. The DFD is drawn using various symbols. It has a source and a destination. The process is represented using circles and source and destination are represented using squares. The data flow is represented using arrows. One reader can easily get the idea about the project through Data Flow.

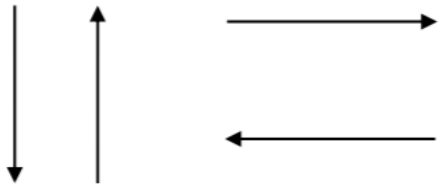
- **Symbols used in DFDs:**

(1) Process: Here flow of data is transformed. E.g. Forms Distribution, Preparing Merit list, etc.



(2) External Entity: A source or destination of data, which is external to the system. E.g. Student, Committee etc.

(3) A data flow: It is packet of data. It may be in the form of document, letter etc.



(4) Data store: Any store data but with no reference to the physical method of storing.

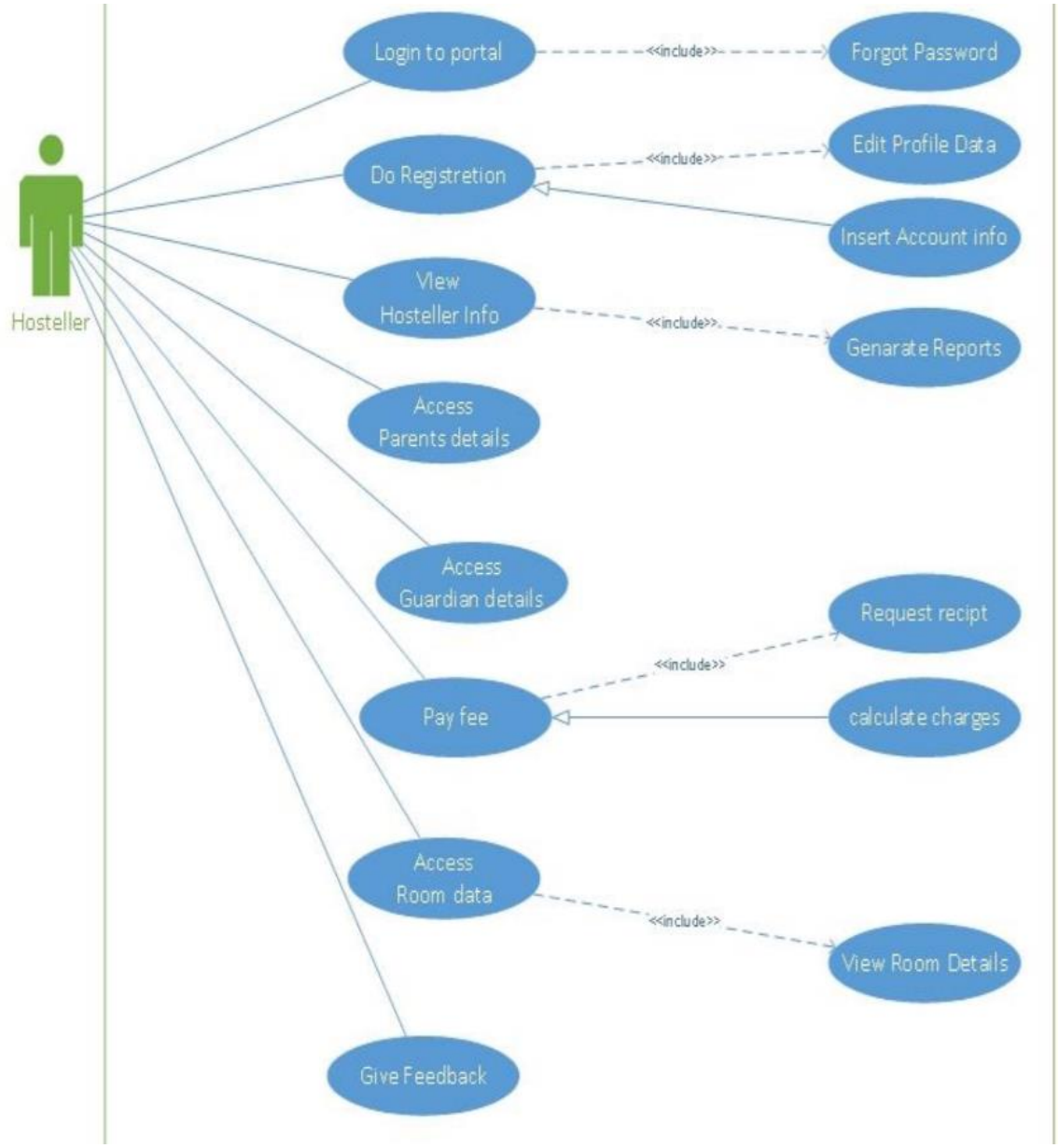
Use Case Diagrams:

1) Use case diagram is a diagram that shows the interaction between user and system to capture the user's goals.

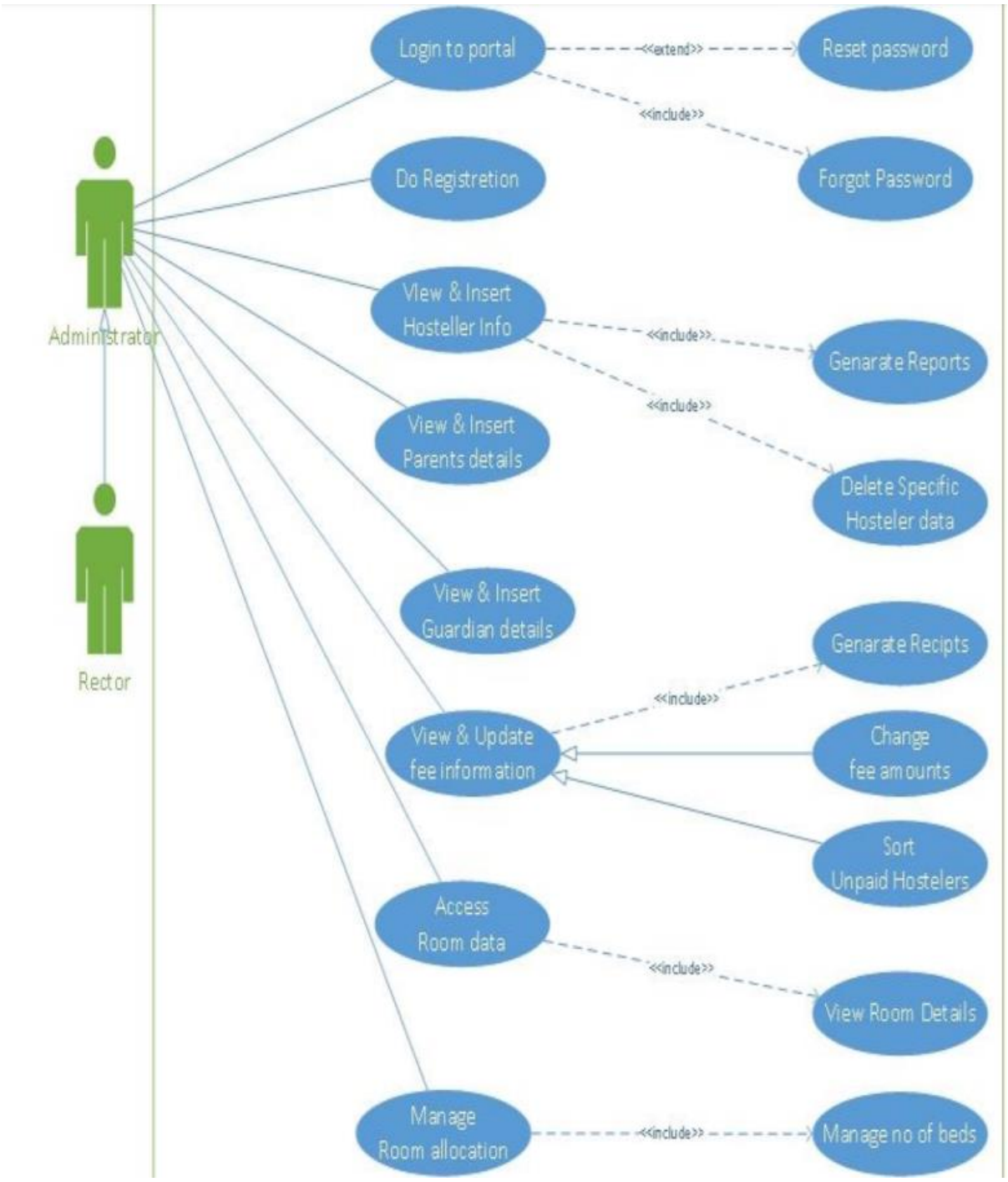
2) It is a graphic depiction of the interactions among the elements of a system. A use case is a methodology used in system analysis to identify, clarify, and organize system requirements.

In Below mentioned use case diagram shows the main features that are accessible by the hosteller or user by using Use cases as for each separated use with system.

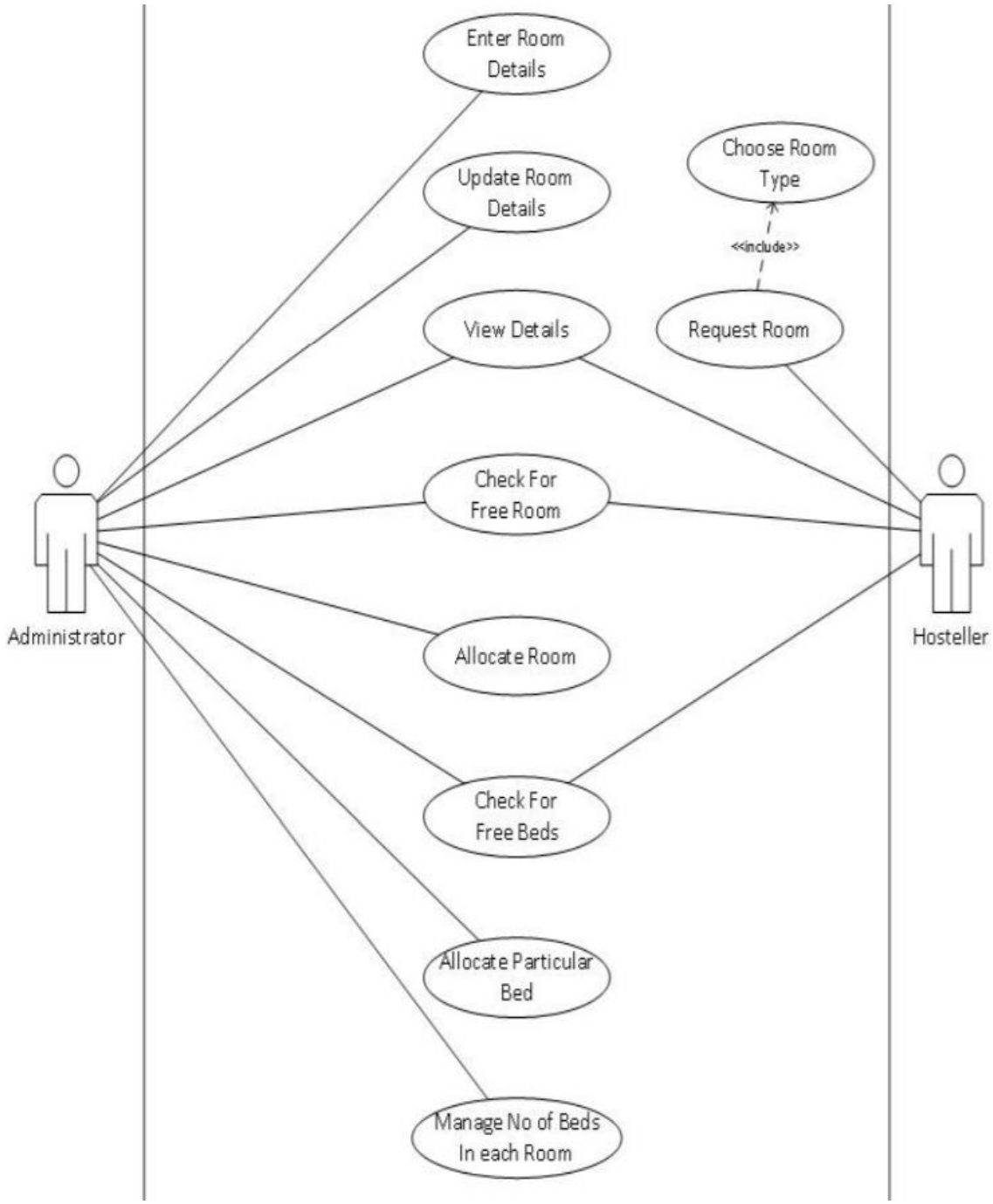
Use case for Hosteller:



Use case for administrator:

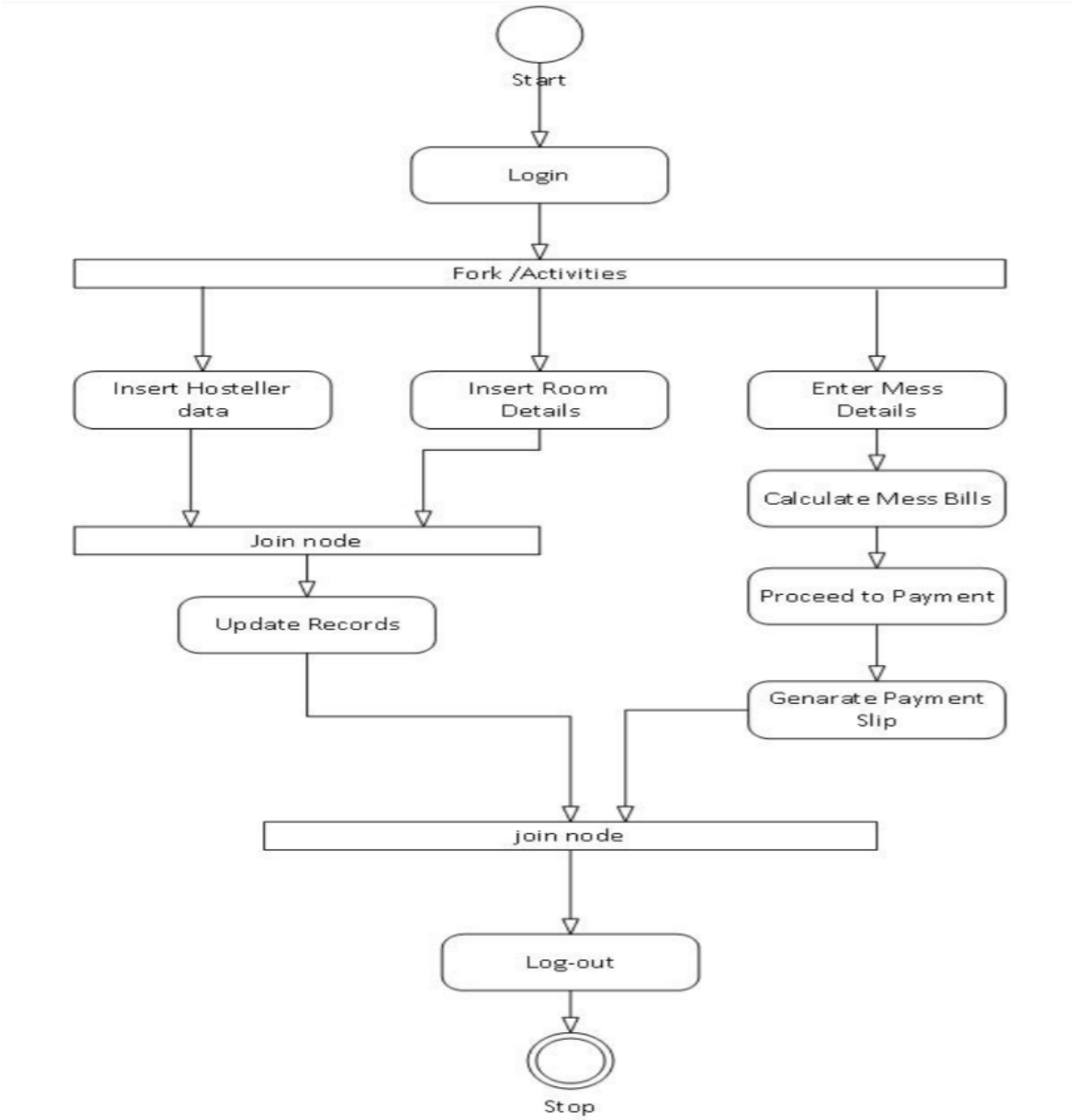


Use case for room allocation:



Activity Diagram:

In an activity diagram the diagram shows the actual process of working activities that are done by system and as well as administrator and hosteller entities.



IMPLEMENTATION CODES:

Administrator account with database connectivity

```
<?php
    session_start();
    $a=$_POST["user1"];
    $b=$_POST["pass1"];
    $c=$_POST["pass1"];
    $_SESSION['user1']=$user1;
    $_SESSION['pass1']=$pass1;
    $cc=mysql_connect("localhost","root","");
    mysql_select_db("hostel");
    $abc="CREATE TABLE IF NOT EXISTS 'adm_account'(
    'id' INT(20) NOT NULL AUTO_INCREMENT,
    'user' VARCHAR(60) NOT NULL,
    'pass' VARCHAR(60) NOT NULL,
    'pass_r' VARCHAR(60) NOT NULL,PRIMARY KEY('id'))";
    mysql_query($abc);
    $sql="insert into adm_account (user,pass,pass_r) values ('$user1','$pass1','$pass1')";
    mysql_query($sql);
    mysql_close($cc);
?>
```

6.1 Administrator Login Code

```
<?
session_start();
if(isset($_REQUEST['sub1']))
{
	$user=$_REQUEST['user1'];
	$pass=$_REQUEST['pass1'];
	$cc=mysql_connect("localhost","root","");
	mysql_select_db("hostel");
	$sql="SELECT * FROM adm_account where user1='$user' AND pass1='$pass'";
	$res=@mysql_query($sql);
	//$a=@mysql_affected_rows();
	//if($a>=1)
	$num=mysql_num_rows($res);
	if($num>0)
	{
		$_SESSION['pass']=$pass;
		$_SESSION['user']=$user;
		header("location:admin_home.php");
	}
	else
	{
		$flag=1;
		$msg="Wrong username or password"
	}
```

Javascript Validation Code

```
function validate(f)
	{
		if((f.user.value=="")||(f.user.value.length<5))
			{
				alert("Please enter a valid username");
				f.user.focus();
				return false;
			}
		if((f.pass.value=="")||(f.pass.value.length<6))
			{
				alert("Please enter a valid password");
				f.pass.focus();
				return false;
			}
		return true;
	}
```

Editing Student details

```
<?php
	session_start();
	if(!(isset($_SESSION['user'])) && !(isset($_SESSION['pass'])))
	header("location:index.php");
	include_once("include_files/db.php");
	$con=new dbconnect();
	$con->open();
	//accepting values from form and inserting them into database
```

```
if(isset($_REQUEST['edit']))
	{

	//receives data from current form
	$year=$_REQUEST['year'];
	$message="For"." "."-".$year.""."Batch";
	$table=$year.'r';
	$f1=0;
	$f2=0;
	$table_a=$year.'a';
	$sql="SELECT * FROM $table_a";
	$result=@mysql_query($sql);
	$total_num=@mysql_num_rows($result);
	if($total_num<=0)
	{
		$f1=1;
	}
	$table_i=$year.'i';
	$sql="SELECT * FROM $table_i";
	$result=@mysql_query($sql);
	$total_num=@mysql_num_rows($result);
	if($total_num<=0)
	{
		$f2=1;
	}

	$s_code=substr($sem,1,1);
	$stud_num=$_REQUEST['stud_num'];
```

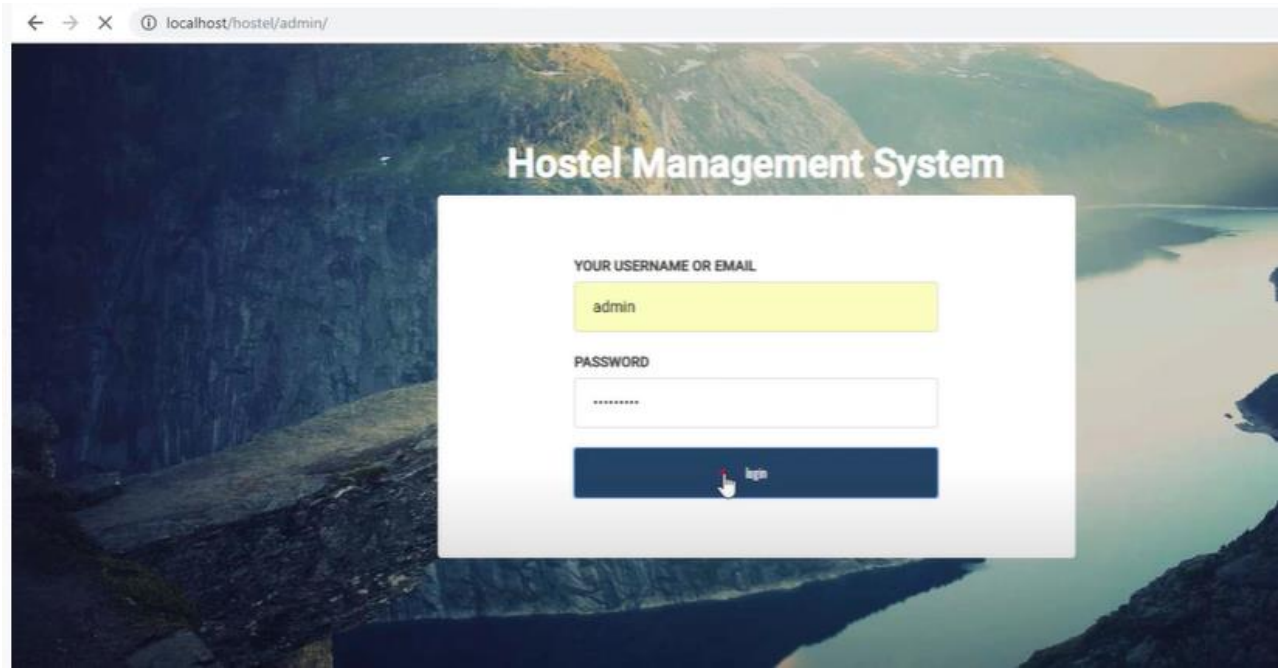
```
for($i=1;$i<=$stud_num;$i++)
		{
			$id[$i]=$_POST["id".$i];
			$roll_no[$i]=$_POST["roll_no".$i];
			$reg_no[$i]=$_POST["reg_no".$i];
			$name[$i]=$_POST['name'.$i];
			$email[$i]=$_POST['email'.$i];

		}
		for($i=1;$i<=$stud_num;$i++)
		{
			$sql="UPDATE $table SET
roll_no='{$roll_no[$i]}',reg_no='{$reg_no[$i]}',name='{$name[$i]}',
				email='{$email[$i]}' WHERE id='{$id[$i]}'";
			$con->update($sql);
			if($f1==0)
			{
				$sql_a="UPDATE $table_a SET roll_no='{$roll_no[$i]}',name='{$name[$i]}'
					WHERE id='{$id[$i]}'";
				$con->update($sql_a);
			}
			if($f2==0)
			{
				$sql_i="UPDATE $table_i SET roll_no='{$roll_no[$i]}',name='{$name[$i]}'
					WHERE id='{$id[$i]}'";
				$con->update($sql_i);
			}

		}
		header("location:stud_edit.php?year={$year}");
	}
```

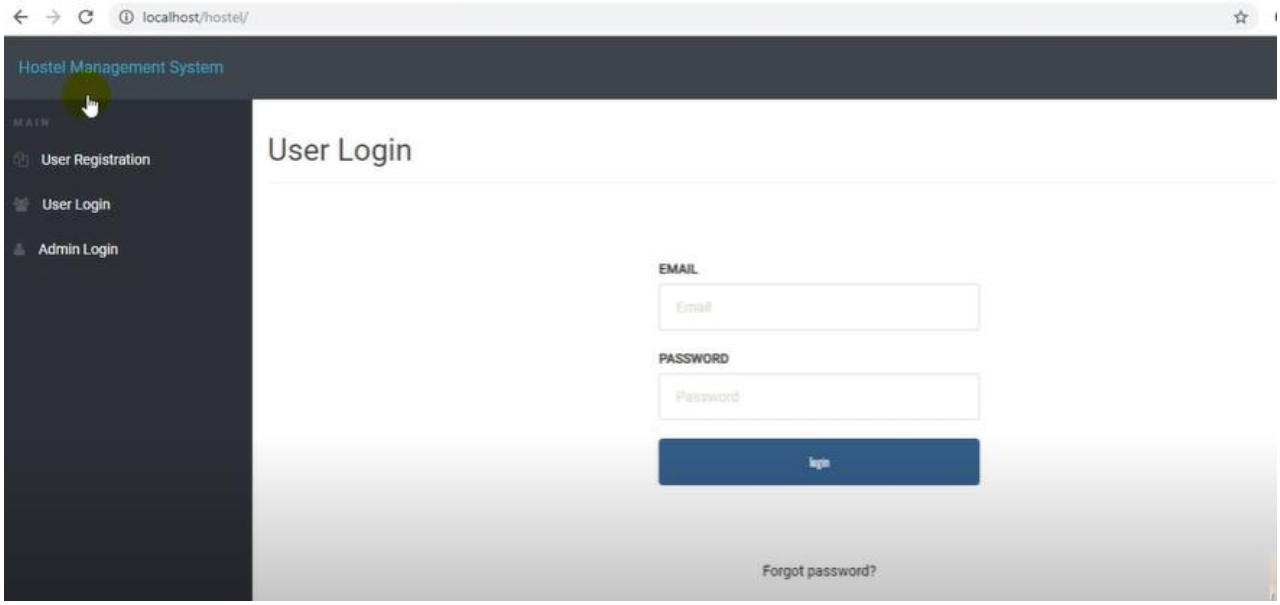
```
if(isset($_REQUEST['delete']))
        {

                $year=$_REQUEST['year'];
                $table=$year.'r';
                $sql="DROP TABLE
                $table";
                @mysql_query($sql);
                header("location:admin_
                home.php");
        }

?>
```

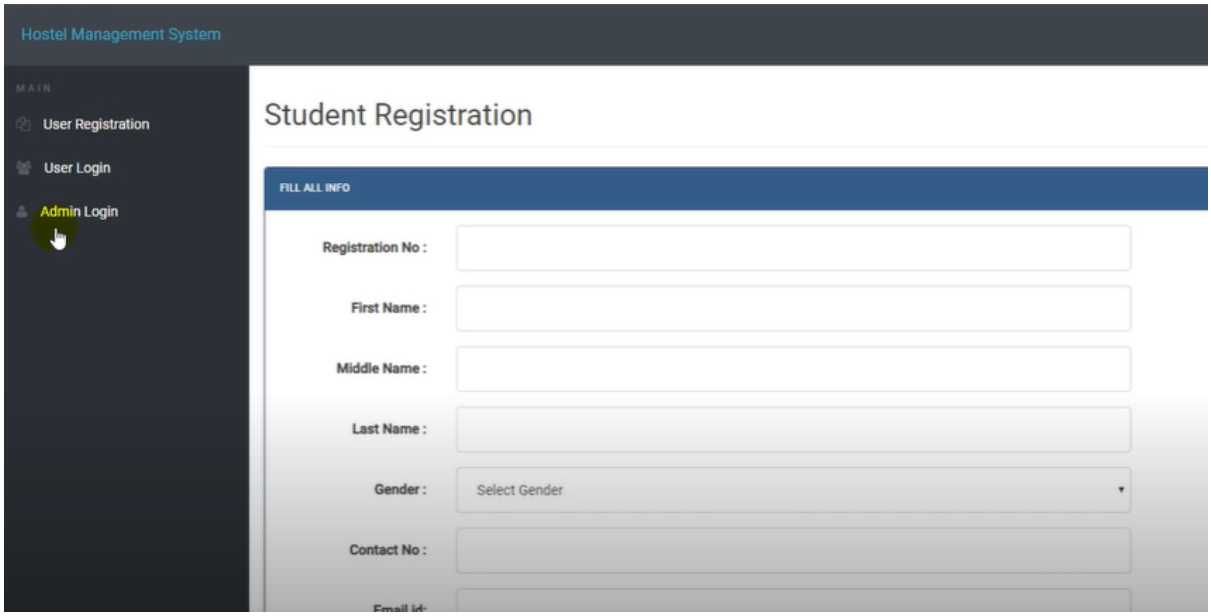
Storing values from database

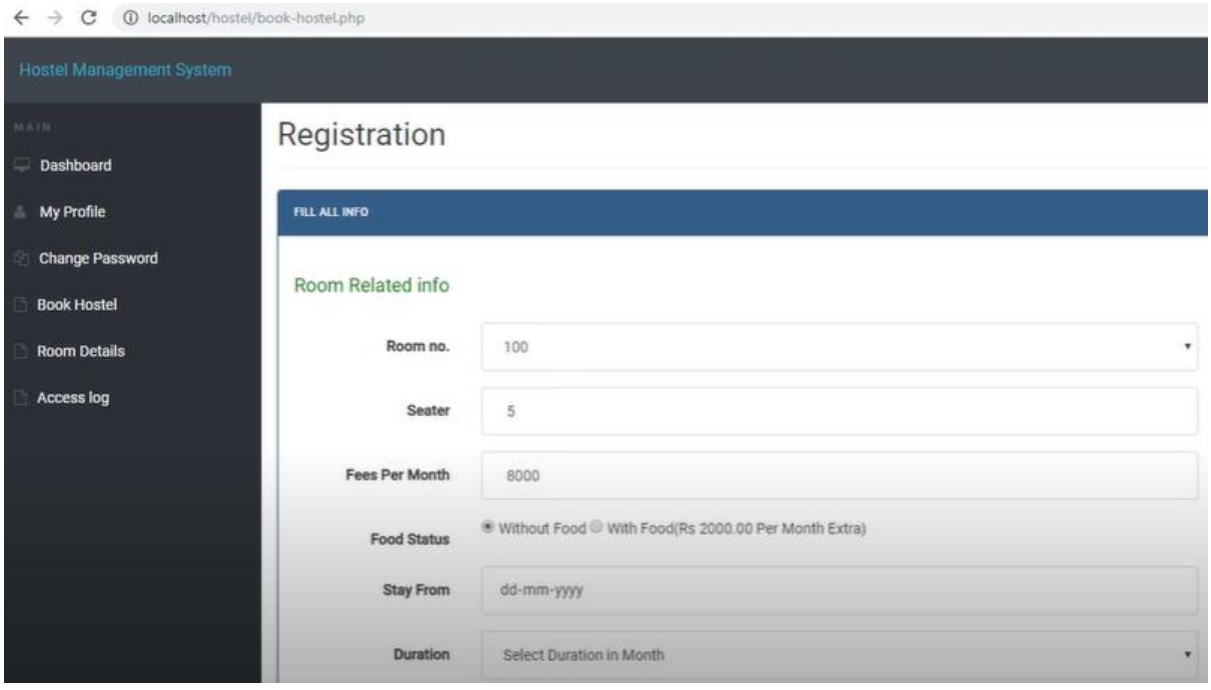
```
                $i=1;
                while($row=@mysql_fetch_arra
                y($result))
                {
                        $id[$i]=$row['id'];
                        $roll_no[$i]=$row['roll_no'];
                        $reg_no[$i]=$row['reg_no'];
                        $name[$i]=$row['name'];
                        $email[$i]=$row['email'];
                        $i++;
                }
```

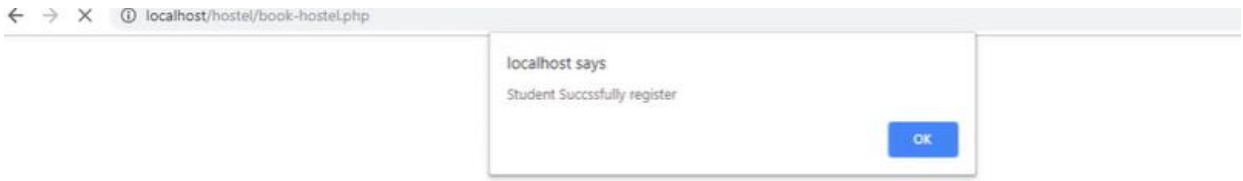
OUTPUT SCREENSHOTS






Hostel Management System
MAIN
User Registration
User Login
Admin Login
Student Registration
FILL ALL INFO
Registration No :
First Name :
Middle Name :
Last Name :
Gender :
Select Gender
Contact No :


localhost/hostel/book-hostel.php
Hostel Management System
MAIN
Dashboard
My Profile
Change Password
Book Hostel
Room Details
Access log
Registration
FILL ALL INFO
Room Related info
Room no.
100
Seater
5
Fees Per Month
8000
Food Status
Without Food
With Food(Rs 2000.00 Per Month Extra)
Stay From
dd-mm-yyyy
Duration
Select Duration in Month


localhost/hostel/book-hostel.php
localhost says
Student Succssfully register
OK

CONCLUSION:

To conclude the description about the project: The project developed using PHP and MySQL based on the requirement specification of the user and the analysis of the existing system, with flexibility for future enhancements.

The expanded functionality of today's software requires an appropriate approach towards software development. This hostel management software is designed for people who want to manage various activities in the hostel. For the past few years, the number of educational institutions is increasing rapidly. Thereby the number of hostels is also increasing for the accommodation of the students studying in this institution, and hence there is a lot of strain on the person who is running the hostel and software's are not usually used in this context. This particular project deals with the problems on managing a hostel and avoids the problems which occur when carried manually.