Dharmsinh Desai University, Nadiad

Faculty of Technology

Department of Computer Engineering

B.Tech CE Semester - IV

Subject :- Software Engineering and Practice

Project Title:- Hostel Management System

By:

Jay Prajapati

(CE109) (19CEUBS052)

Smit Patel

(CE106) (19CEUBS079)

Guided by:

Brijesh S. Bhatt

Prof. Jigar M. Pandya

Prof. Pinkal Chauhan

DHARMSINH DESAI UNIVERSITY
NADIAD-387001, GUJARAT

CERTIFICATE

This is to certify that the project entitled as "Hostel Management System" is a bona-fide report of the work carried out by

- **Prajapati Jay Ashishkumar,** Student Id: **19CEUBS052**
- **Patel Smit Vinodkumar,** Student Id: **19CEUBS079**

of Department of Computer Engineering, semester **IV**, under the guidance and supervision of **Prof. Jigar M. Pandya, Brijesh S. Bhatt, Prof. Pinkan Chauhan** for the subject **Software Engg. & Practices** during the academic year 2020-2021.

Project Guide
Assistance Professor
Prof. Jigar M. Pandya
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad

Head of the Department
Prof. & Head,
Dr. C.K Bhensdadia
Department of Computer Engineering,
Faculty of Technology,
Dharmsinh Desai University,
Nadiad

Contents:

1.Abstract

This project can be used in hostel to facilitate to store new records of students in database, to display the saved records.

We have tried to fulfil most of the desire of the hostel requirements but some of the requirements were solved and some of requirement are not solved.
The problem analysed for the hostel were Check in/ checkout system for students to keep records of room that have been reserved. Room Reservation system to keep records of the Students and find their room number.

The user can also save record of total number of students in the hostel.

This software is totally secure with password protected facility. The administrator can create new user, update them when required and delete them if necessary. The administrator staff can utilize all the facility provided by the software and all the guest staff can get some of the facility. This software also provides the facility to create backup of the database which will help to retrieve data when the accidental loss in the original database happen.

2.Introduction

Hostel Management System in which there are manly three users Admin, Students and Mess staff. This System is for new commer students and students who lived there.

In this system if the student wants to admission in this hostel, then the system takes his details and stores in the database and check for the eligibility. If the student is not eligible, then the system won't allow him for the admission. If the details are correct then his admission is successfully done. If the student is already part of the hostel then he can check his details about fee payment and he can also check notice board if admin put some important notice on that.

If student misbehave or sabotage then admin can give him penalty. Mess staff can only see the details about the student's fee payment for the mess.

3. Software Requirement Specification

1.: User Panel

Description: In this panel student can login and fill the Registration form and conformation that admission.

R.1: User Login

R.1.1: Home

State: This consist of the different pop-up menus showing the details of the different hostels.

Input: "Registration Form" selection…

Output: Gives the message to the screen to enter the registration details.

R.1.2: Registration Form

State: The Registration form has been selected.

Input: Student give that details.

Output: Print the confirmation of the Registration

Form.

Processing: It check the students details and

Gather information.

Next Function: If the student detail is correct then 1.3 else 1.2

R.1.3: Registration conformation:

Input: Give the details and document of the students that is given by the students.

Output: Give the conformation letter

2.: Admin Panel:

R.2.1: Administrator Login:

R.2.1.1: Home

Description: Allot different students to the hostel.

Allot the rooms to the students. List of the vacant

rooms in the hostel.

R.2.1.2: Allotting Rooms:

Input: Student have to enter that details.

Output: display the room number that has been

Allotted.

Processing: from the vacant room's students have

Allots the room.

R.2.1.3: Vacant Room:

State: Check the available rooms to the students.

Input: Click to the vacant room to check.

Output: Display the vacant room.

Processing: in that the system check the how many Rooms data is filled based on that the system will give the list of the vacant rooms.

R.2.1.4: Shifting a room:

Description: If students want to change their rooms, they can do it. We have to delete their old

Record for room and add new record for them.

3.: Fee Payment Process:

Description: In that panel student login through that

ID and generate the fee voucher and after successfully

Paying the fee, they can check the fee payment status

And can print that fee receipt.

R.3.1: Login

State: In that student see the different modes

Of the fee payment.

Input: User selection.

Output: Display the screen that ask to students for

That give the necessary details.

Processing: Check the details of the students that

entered from the students.

Next Function: 3.2 if the details is correct else

3.1

R.3.2: Fee Payment:

State: In that student pay the fee.

Input: Pay the fee.

Output: Print fee receipt option.

Processing: Student pay the fee in the account

that is given.

Next Function: R.3.3 if the transaction is successful else R.3.2

R.3.3: Taking Fee Receipt

Input: click to print the fee receipt.

Output: Get the print of the fee receipt.

R.3.4: Mess fees:

Description: For the mess fee they can also generate fee voucher and pay the fees.

Input: Student generate the fee voucher

Output: Student print the fee voucher.

4.: Manage Student's Details:

R.4.1: Register details:

Description: To register student's details, we take Details like student's photo, address details, phone Number, college name etc.

Input: Student's Details.

Output: Details have been stored.

Processing: when student store the detail of the Detail in the database.

R.4.2: Delete details:

Description: We have to delete the details of those students who would leave the hostel.

Input: Give the information that which students Data have to delete.

Output: Delete that particular student's data**.**

Processing: In that we search the details of the Student's and delete that particular student's data from the database.

5.: Visitor Record:

R.5.1: Store visitor data.

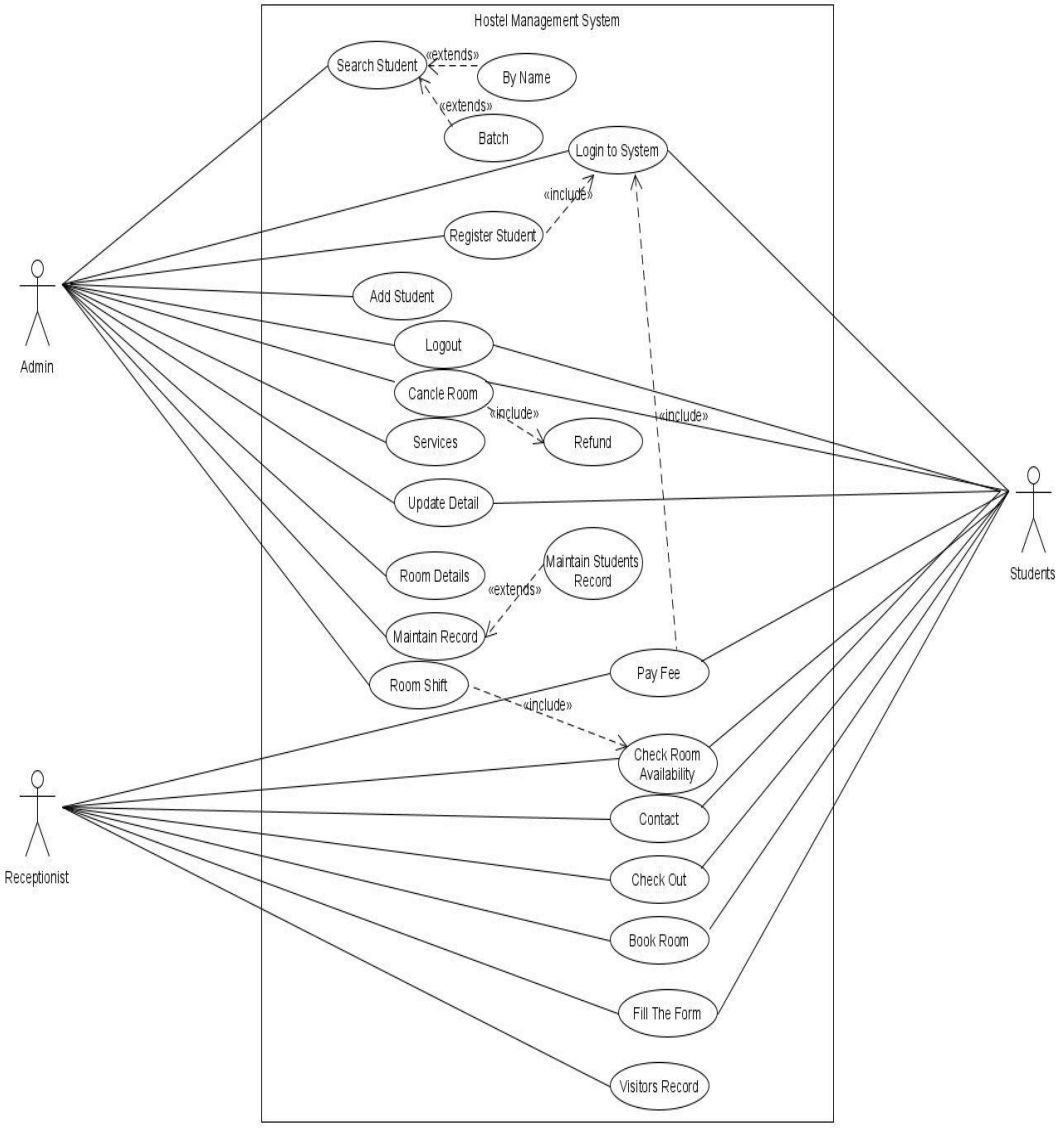
Description: If someone come for visiting their child, we have to keep the information about it.

Input: Parents name and their mobile number and room no. we save the Entry time of that parent. And when parents will go from the hostel, we note that time.

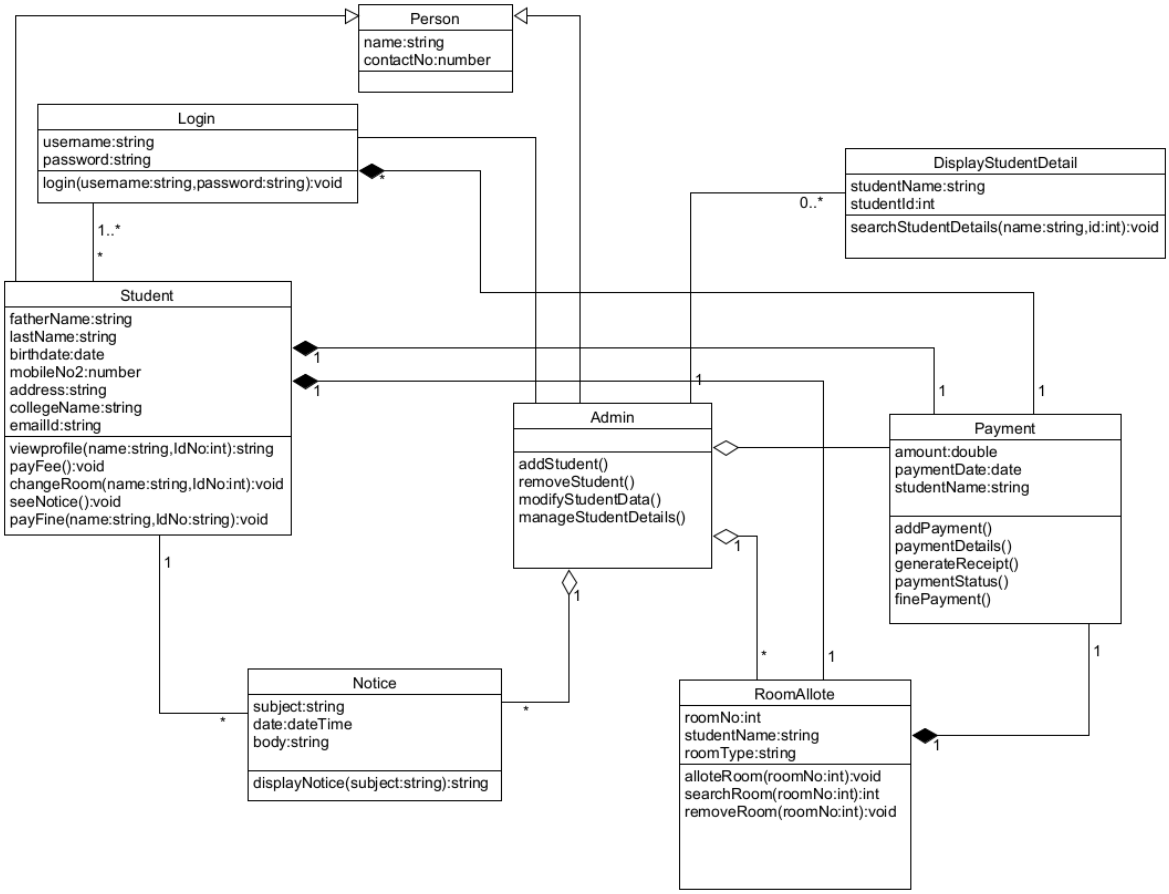
Output: save that record in the visitor record.

4. Design Documents

- **Use case Diagram**

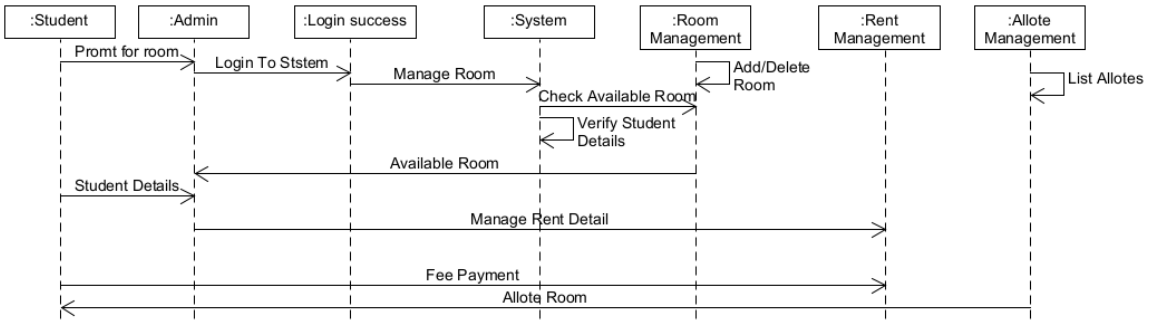


- **Class diagram**

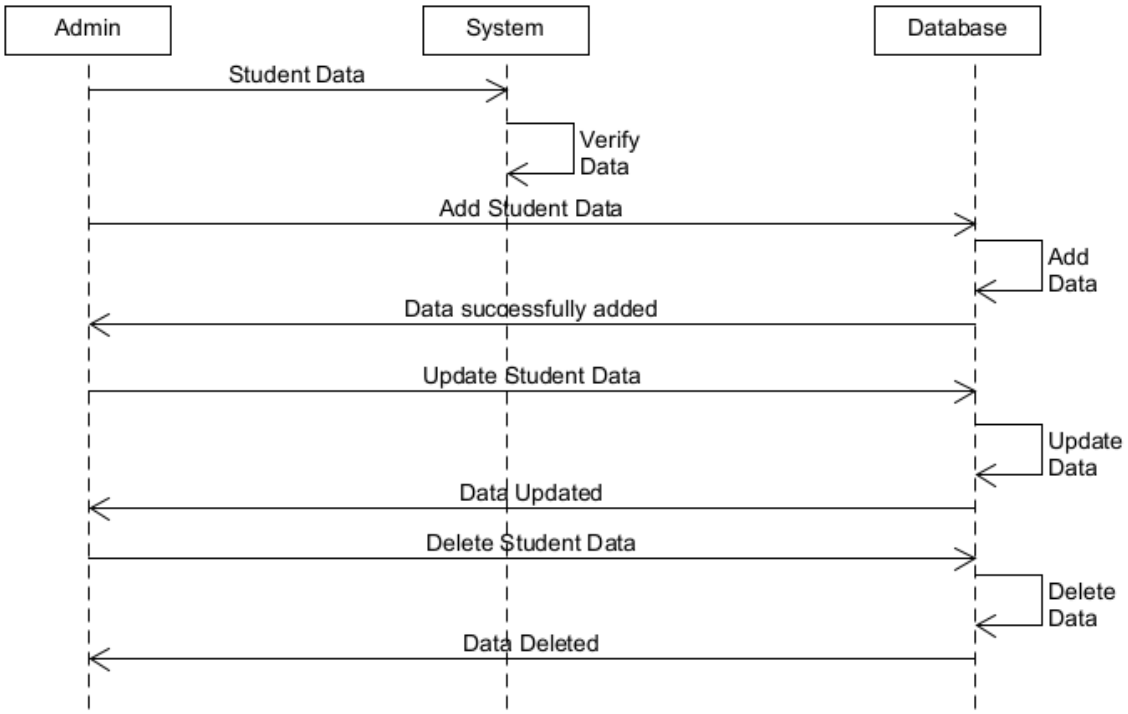


- **Sequence diagram**

1.Room Allotment

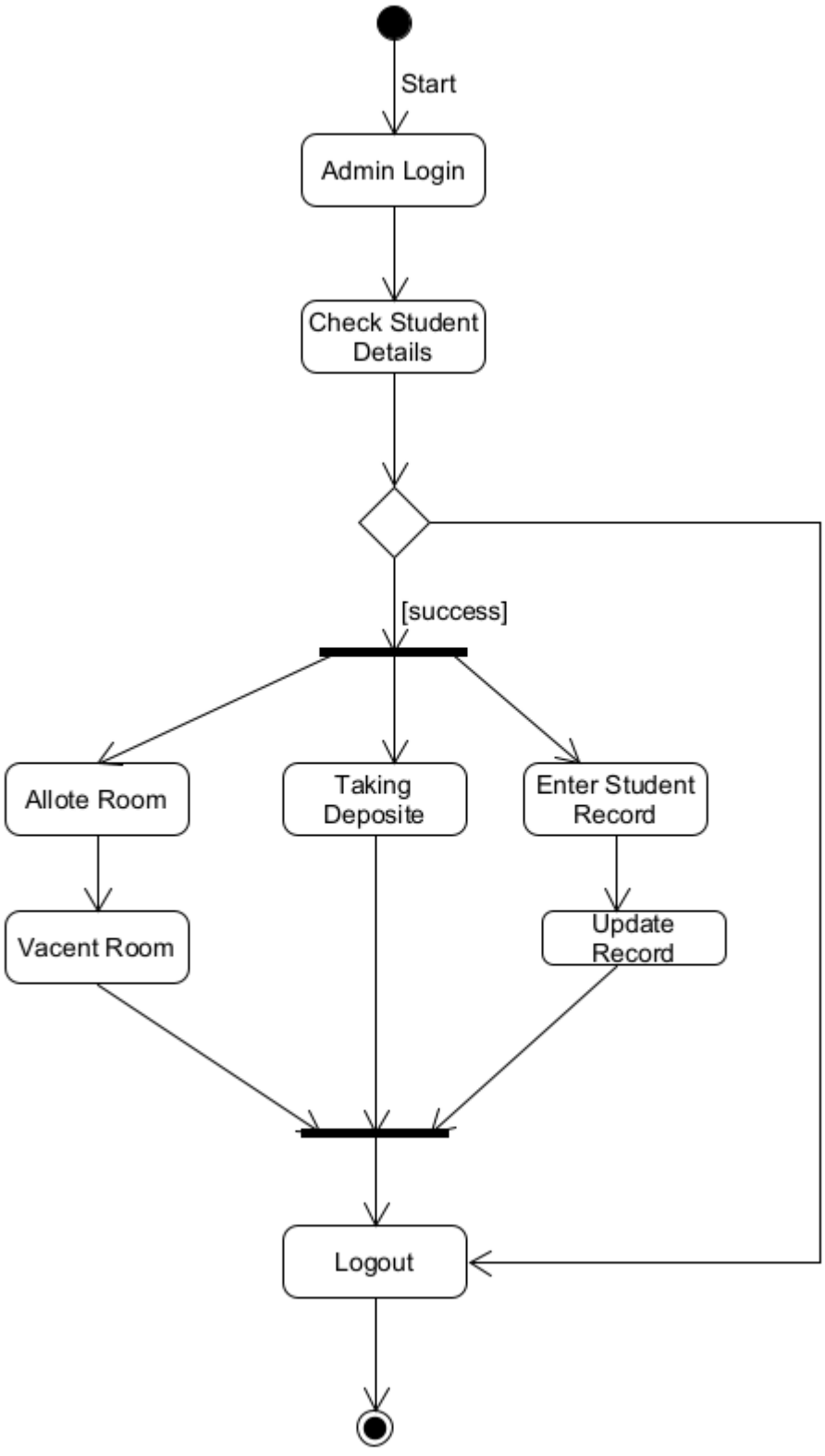


2.Manage Student Details

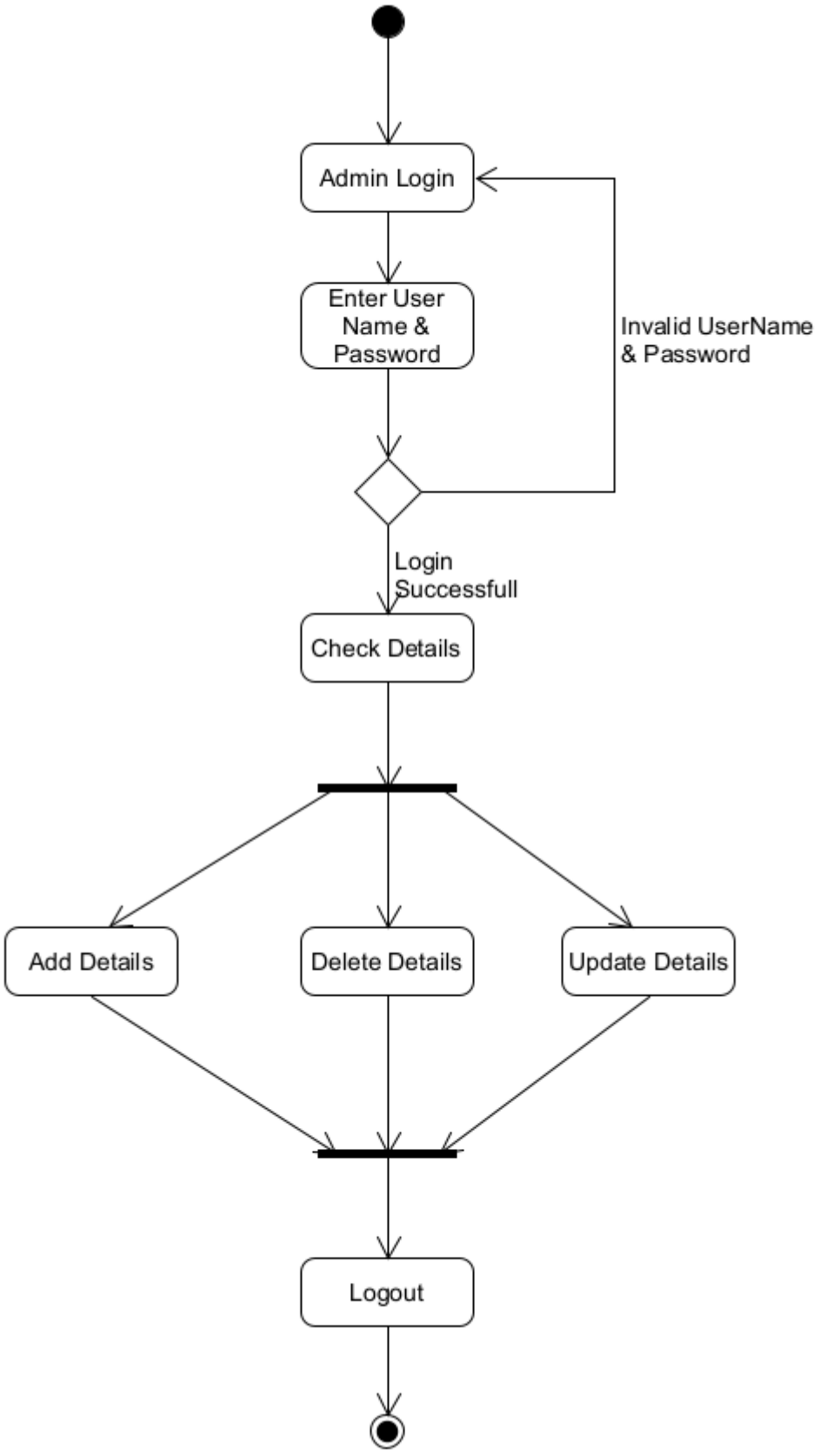


- **Activity diagram**

1.Room Allotment

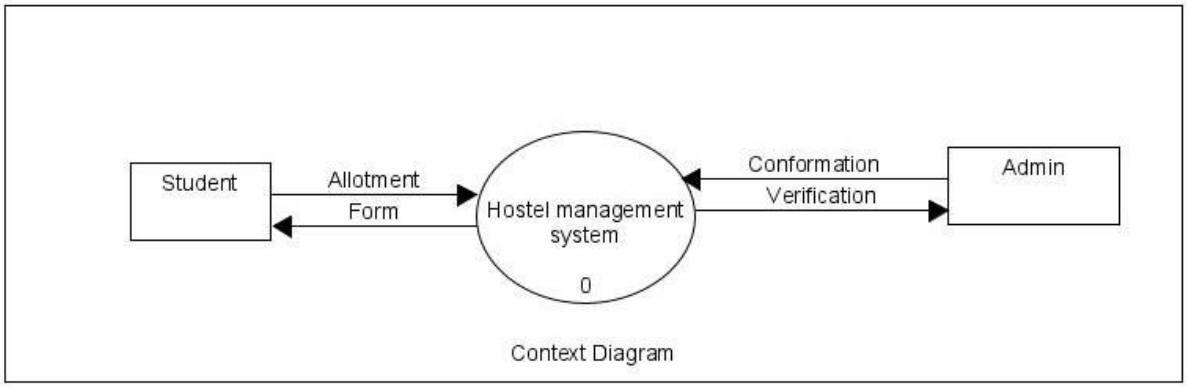


2.Manage Student Details

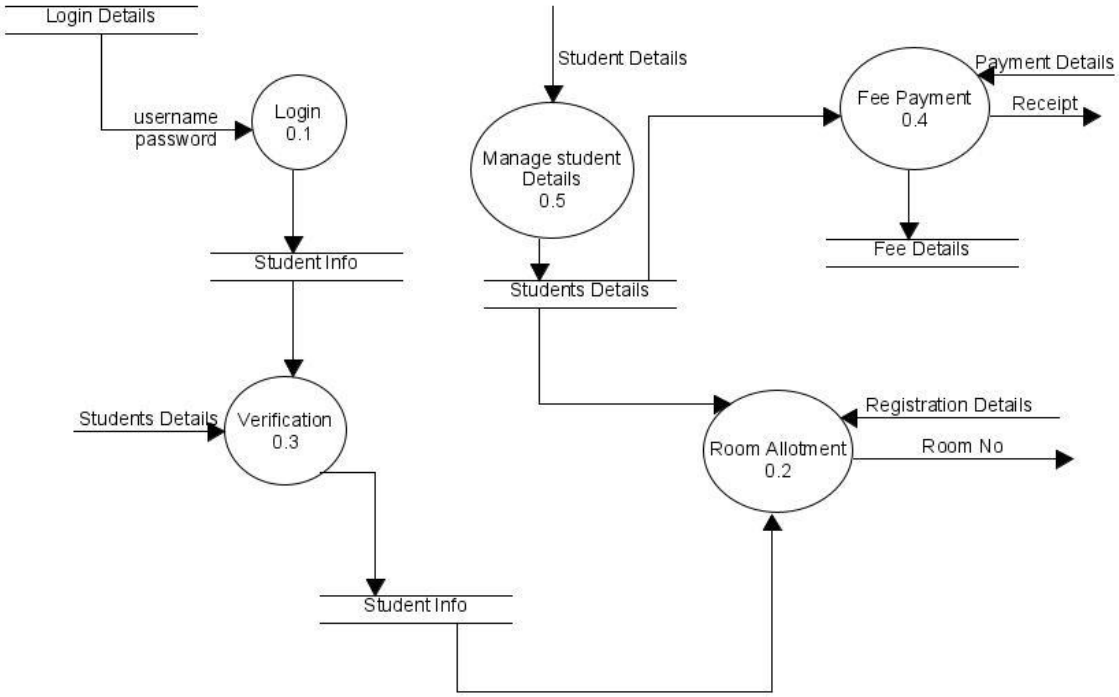


• Data Flow diagram

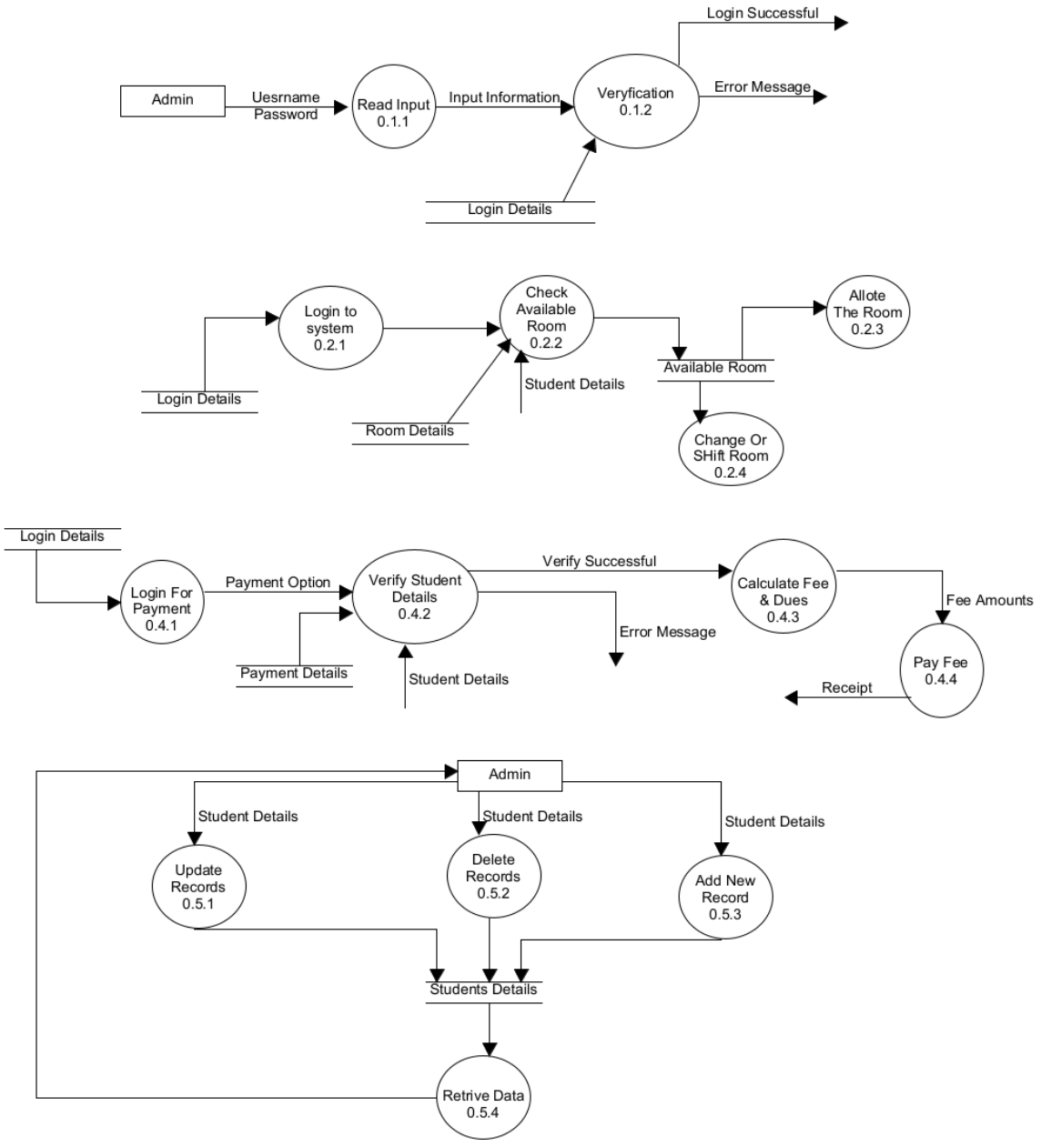
- **Level 0 Diagram**



- **Level 1 Diagram:**



➢ Level 2 Diagram:



5. Implements Details

1.Modules

- **Users:**

  This helps the administrator that he/she only creates new user by providing login details. Admin can modified the information of the students if necessary.

- **Registration:**

  Allows to add the details of students regarding personal information, Academic information,room details etc. students details can also be modified and deleted.

- **Room Details:**

  Allows the user to view the status and details of room. Admin can also modified and delete the data.

- **Fee:**

  Allows to enter the fees details of the students. Students can modified the details also.

2. Major Functions prototypes

```python
def register(request):

    if request.method == 'POST':
        first_name=request.POST['first_name']
        last_name=request.POST['last_name']
        username=request.POST['username']
        password1=request.POST['password1']
        password2=request.POST['password2']
        email=request.POST['email']

        if password1==password2:
            if User.objects.filter(username=username).exists():
                messages.info(request,'Username Taken')
                return redirect('register')
            elif User.objects.filter(email=email).exists():
                messages.info(request,'Email Taken')
                return redirect('register')
            else:
                user=User.objects.create_user(username=username,password=password1,email=email,first_name=first_name,last_name=last_name)
                user.save()
                print('user created')
        else:
            messages.info(request,'password does not match...')
            return redirect('register')
        return render(None,'login.html')
```

Register Students

```python
from django.views.decorators.csrf import csrf_exempt
@csrf_exempt
def addstudent(request):
    s_name=request.POST.get('studentname','')
    fname=request.POST.get('fathername','')
    bdate=request.POST.get('birthdate','')
    sgen=request.POST.get('gender','')
    mno=request.POST.get('mobileno','')
    sbranch=request.POST.get('branch','')
    sage=request.POST.get('age','')
    sarea=request.POST.get('area','')
    scity=request.POST.get('city','')
    sstate=request.POST.get('state','')
    semail=request.POST.get('email','')
    ssemester=request.POST.get('semester','')
    std=Student(student_name=s_name,f_name=fname,b_date=bdate,s_gender=sgen,mobile_no=mno,branch=sbranch,s_age=sage,add_area=sarea,
    add_city=scity,add_state=sstate,s_email=semail,s_sem=ssemester)

    std.save()
    return render(None,'addstudent.html')
```

Add student

```python
from django.views.decorators.csrf import csrf_exempt
@csrf_exempt

def addpayment(request):
    s_year=request.POST.get('year','')
    s_amount=request.POST.get('amount','')
    s_paytype=request.POST.get('paytype','')
    s_paydetails=request.POST.get('paydetails','')
    s_room=request.POST.get('roomno','')
    s_paymentdate=request.POST.get('paymentdate','')

    pay=Payment(p_year=s_year,p_amount=s_amount,p_type=s_paytype,p_details=s_paydetails,p_roomno=s_room,created_date=s_paymentdate)
    pay.save()
    return render(None,'payment.html')
```

Add Payment

```python
from django.views.decorators.csrf import csrf_exempt
@csrf_exempt
def addroom(request):
    st_name=request.POST.get('name','')
    st_lastnane=request.POST.get('lname','')
    st_sem=request.POST.get('sem','')
    st_branch=request.POST.get('branch','')
    st_roomno=request.POST.get('roomno','')

    room=Addstudent(s_name=st_name,s_surname=st_lastnane,s_sem=st_sem,s_branch=st_branch,s_room=st_roomno)
    room.save()
    return render(None,'addroom.html')

def addrooms(request):
    return render(None,'addroom.html')

def getroomdetails(request):
    all_room=Addstudent.objects.all()
    return render(request,"roomdetail.html",{'Addroom':all_room})
```

Add Rooms

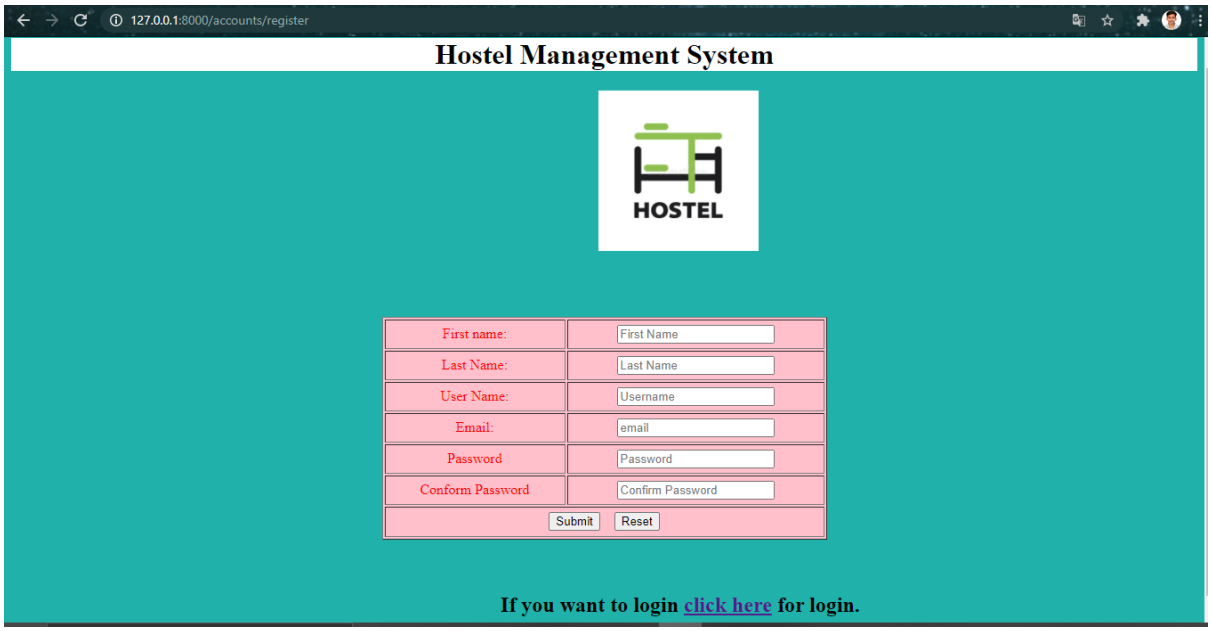
```python
from django.views.decorators.csrf import csrf_exempt
@csrf_exempt
def addnotice(request):
    nts=request.POST.get('notice', '')
    decr=request.POST.get('description', '')
    s=Notice(notice_title=nts,description_data=decr)
    s.save()
    return render(None,'notice.html')
```

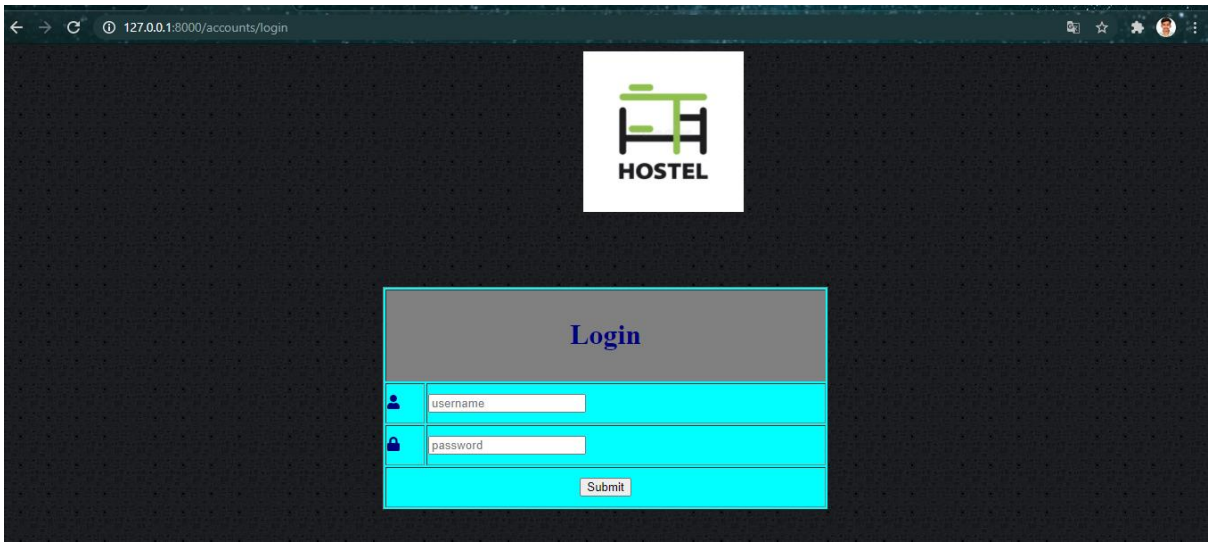
Add Notice

6.Layouts

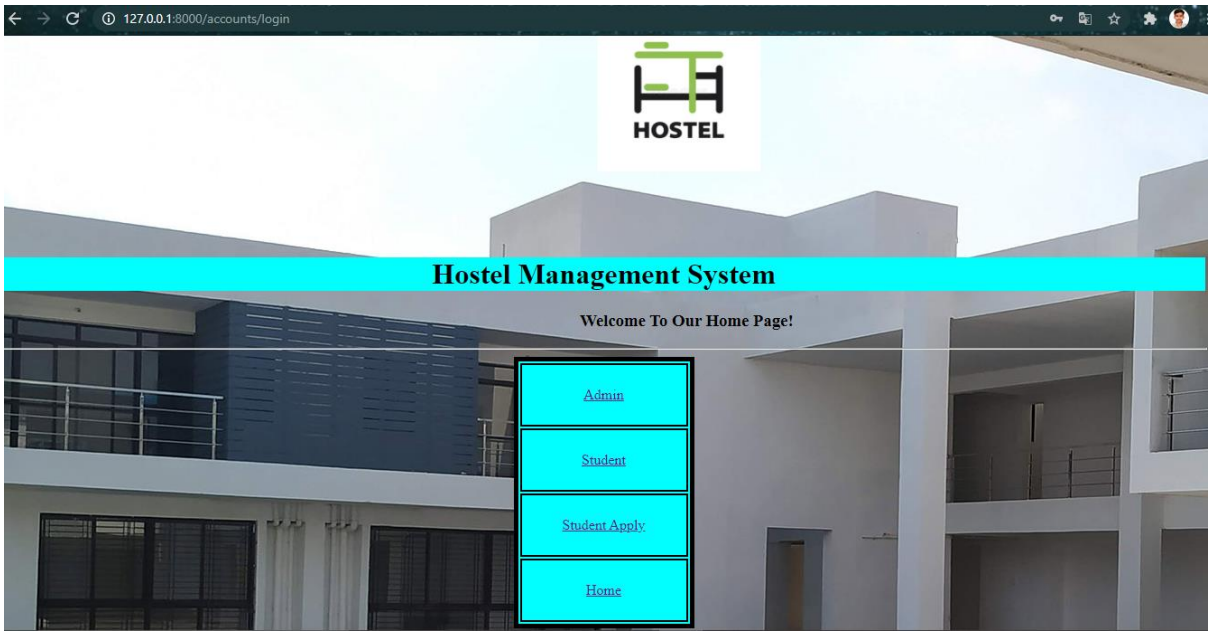
Layouts:

1)Register/Login Module:

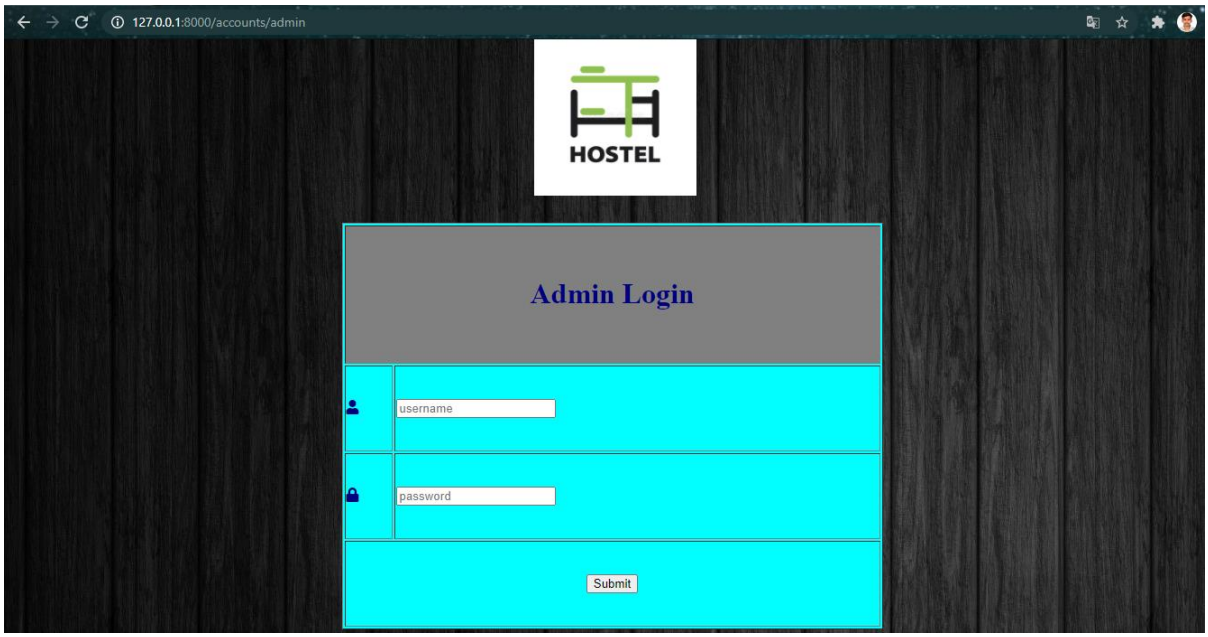


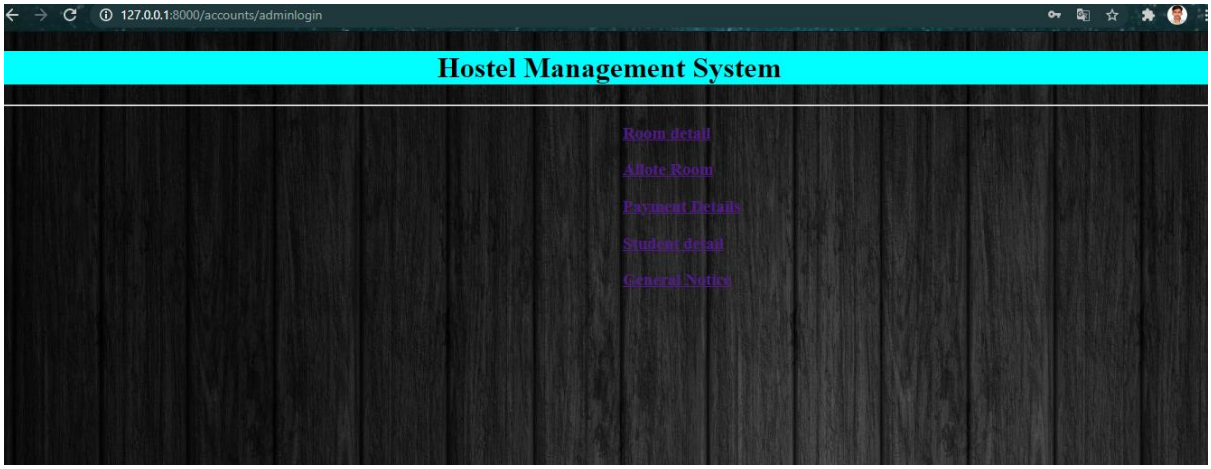


2)Main Page:

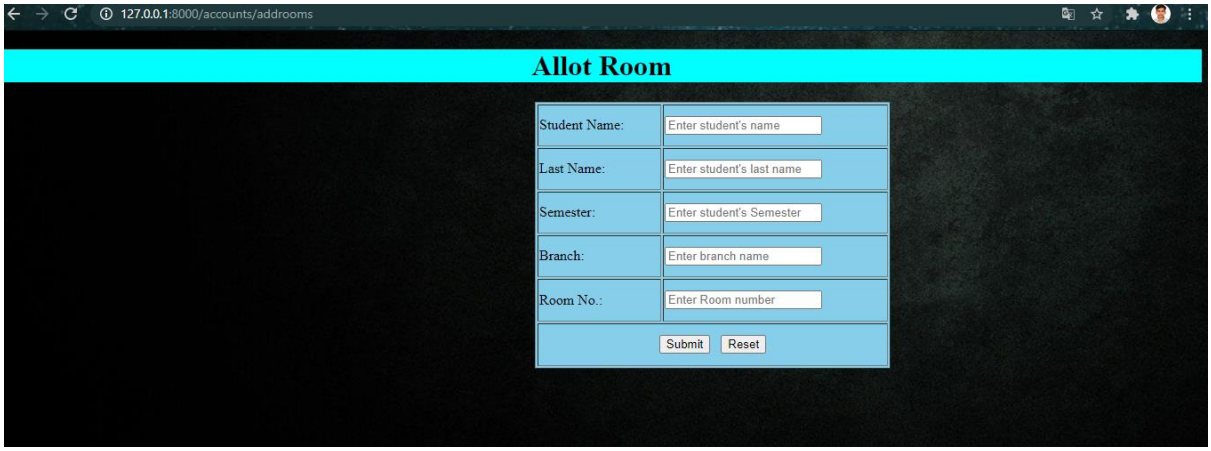


3)Admin Module:

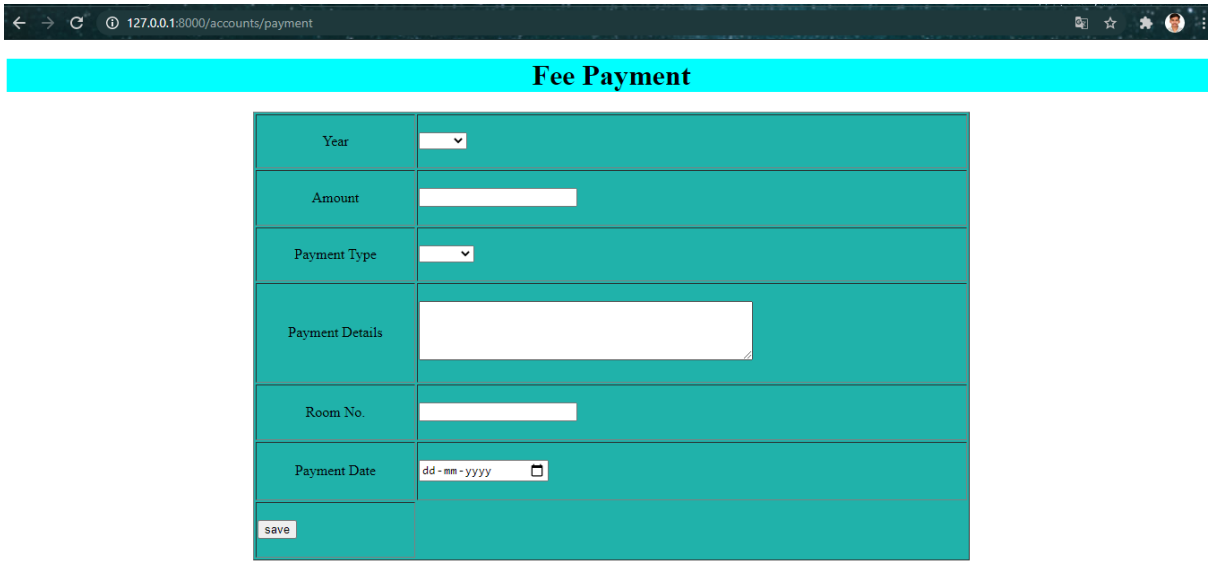




Allot Room:



4)Student Module:



127.0.0.1:8000/accounts/getnotice

Notices List

ID	Notice Title	Description
1	ldchsidu	ldbiuds
2	ldchsidu	ldbiuds
3	ldchsidu	ldbiuds
4	sjdfnskj	kdjcsdsd
5	notice	here is first notice
6	regarding fee payment	please pay yor fee by end of this month
7	jci	ljfhidu

7.Conclusion

Hence, In this project we have successfully implemented Admin Panel, Students Panel, add student Room Allot Functionality. Student can register to the system, Admin will add the students. Admin will allot the rooms and save that students record. Students can see that data and pay the fees of the hostel and mess. That record can be seen by the Admin. Admin can post Notice and students can see that notice.

8. Limitation and future extension

Limitations:

- Hostel Managements System is intended for Hostel (like schools, colleges).
- There will be pre-characterized criteria for the Reserve to the hostels.
- If the students are qualified then they are designated to the hostel Room.
- The system is not able to generate the pdf reporting.

Future Extension:

There are many additional features, which can be planned to incorporated during the future enhancement of the project.

- User interface will be improved to provide better interaction with system.
- Pdf will be integrated for more reliability.

9.Reference / Bibliography

Following links and websites were referred during the development of this project:

- https://getbootstrap.com/
- https://www.djangoproject.com/
- https://github.com/
- https://stackoverflow.com/
- https://codewithharry.com/