SOFTWARE TESTING REPORT

Overview

Our project works as a well-connected system for all students and personnel in the hostel, integrating the benefits of better organizability with effortless accessibility leading to an increased efficiency for the workforce and convenience for all. The project has been implemented as an individualistic yet single well-knit system to provide unique functionalities for every different type of users while keeping the integrity of the hostel held closely

.

TESTING

The tests performed on the system are as follows:

1) **Unit Testing** (done as white box testing)
2) **Manual Testing** (done as black box testing)
3) **Integration Testing** (done as black box testing)
4) **System Testing** (done as black box testing)

Unit Testing

It is a type of software testing where individual units or components of a software are tested. The purpose is to validate that each unit of the software code performs as expected.

This was **white box testing**, where tests were written to ensure that data always returned in a predetermined format. We used npm library mocha to write automated tests. For sample we have added snippets of testing done in the student module and outcome of the tests performed on different module are stated below :

STUDENTS MODULE

```
const chai = require('chai')
const chaiHttp = require('chai-http')
const server = require('../app')
var server2=require('../server');
const expect = chai.expect
const should = chai.should()
chai.use(chaiHttp)

describe('Testing Routes',()=>{

    it('/api/',(done)=>{
        chai.request(server).post('/warden/showStudent').end((err, res)=>{
            expect(res.status).to.be.oneOf([200])
            done()
        })
    })
```

```
    it('/api/',(done)=>{
        chai.request(server).post('/warden/addStudent').send(
            {"name": "Aditya Shara",
            "phNo": "9898098765",
            "rollNo": "1901012",
            "address": "Allahabad, UP",
            "password": "123",
            "branch": "CSE",
            "passingYear" : "2023",
            "email" : "aditya@gmail.com",
            "roomNo" : "451"}
            ).end((err, res)=>{
            expect(res.status).to.be.oneOf([200])
            done()
        })
    })
```

```
it('/api/',(done)=>{
    chai.request(server).post('/warden/deleteStudent').send({"queryId": "625992a151b0abbdea01f1cd"}).end((err, res)=>{
        expect(res.status).to.be.oneOf([200])
        done()
    })
})
```

Testing Result ->

```
listening on port 5000!!


  Testing Routes
database is connected successfully
    ✓ /api/ (1330ms)
    ✓ /api/ (162ms)
    ✓ /api/ (43ms)
    ✓ /api/
    ✓ /api/


  5 passing (2s)
```

GUARD MODULE

```
  Testing Routes
database is connected successfully
    ✓ /api/ (1338ms)
    ✓ /api/ (157ms)
    ✓ /api/
    ✓ /api/ (43ms)
    ✓ /api/ (38ms)
    ✓ /api/
    ✓ /api/
    ✓ /api/
    ✓ /api/
    ✓ /api/
    ✓ /api/


  11 passing (2s)
```

CARETAKER MODULE

```
> backend@1.0.0 test C:\Users\vaibh\Desktop\hms_backend
> mocha --exit

line 10
hi
listening on port 5000!!


  Testing Routes
database is connected successfully
    ✓ /api/ (1308ms)
    ✓ /api/ (177ms)
    ✓ /api/
    ✓ /api/ (39ms)
    ✓ /api/
    ✓ /api/ (38ms)
    ✓ /api/
    ✓ /api/ (38ms)
    ✓ /api/
    ✓ /api/


  10 passing (2s)
```

CLEANING STAFF MODULE

```
  Testing Routes
database is connected successfully
    ✓ /api/ (1221ms)
    ✓ /api/ (187ms)
    ✓ /api/ (47ms)
    ✓ /api/ (42ms)
    ✓ /api/ (41ms)


  5 passing (2s)
```

Manual Testing

Five test accounts were created to test the flow of entire web application.

Student Account

- Verified the credentials of the student form database.
- Only after correct Email-Id and password, student will have access to his/her profile.
- Checked room query by giving a test room query, which was reflected in the database.
- Checked for multiple room query requests.
- Requested for room cleaning request. (A student can only give 1 room cleaning request from his id until his/her room is cleaned)
- Checked working of sports equipment request, which will be approved by guards.
- Few dummy notices were updated in the hostel.
- Checked the dashboard, where all the hostel updates were posted.

Guard Account

- Verified the credentials of the student form database.
- Only after correct Email-Id and password, guard will have access to his/her profile.
- Checked the details of the profile page.
- Checked the test room query given by the test-student and other students.
- Checked the cleaning request by the test-student and other students.
- Checked the delivery updates reflecting on the table from the database.
- Status changed after delivering to "Delivered".
- Checked whether the request for sports eqp by the test-student is visible in the table or not.
- Closed the request after the sports eqp is returned by the student successfully.
- Checked the dashboard, where all the hostel updates were posted.

Warden Account

- Verified the credentials of the student form database.
- Only after correct Email-Id and password, guard will have access to his/her profile.
- Checked the details of the profile page.

- Checked the student details.
- Added a test-student successfully.
- Edited the details of the test-student to check whether the details are edited correctly.
- Checked the dashboard, where all the hostel updates were posted.
- Added a test-notice in the dashboard successfully.
- Deleted the test-notice in the dashboard successfully.

Caretaker Account

- Verified the credentials of the student form database.
- Only after correct Email-Id and password, guard will have access to his/her profile.
- Checked the details of the profile page.
- Checked the guard and cleaning staff details.
- Added a test-guard successfully.
- Edited the details of the test-guard to check whether the details are edited correctly.
- Added a test-cleaning staff successfully.
- Edited the details of the test-cleaning staff to check whether the details are edited correctly.
- Checked the dashboard, where all the hostel updates were posted.
- Added a test-notice in the dashboard successfully.
- Deleted the test-notice in the dashboard successfully.

Cleaning Staff Account

- Verified the credentials of the student form database.
- Only after correct Email-Id and password, guard will have access to his/her profile.
- Checked the details of the profile page.
- Checked the dashboard, where all the hostel updates were posted.
- Accepted request for room cleaning successfully.
- Marked done after cleaning the room successfully.

Taking the account through the complete process from signing up to testing every feature ensured that none of the features were breaking in production and that they were all working properly. Our application is working correctly having all the functions as mentioned in the functional requirement.

This process was repeated with varying the inputs and boundary value inputs to check for integrity of the application.

Integration Testing

Integration testing is the second level of the software testing process after unit testing. In this testing, units or individual components of the software are tested in a group.

The focus of the integration testing level is to expose defects at the time of interaction between integrated components or units and to check the data flow between different components.

Group 1 (Student Interface and Guard Interface)

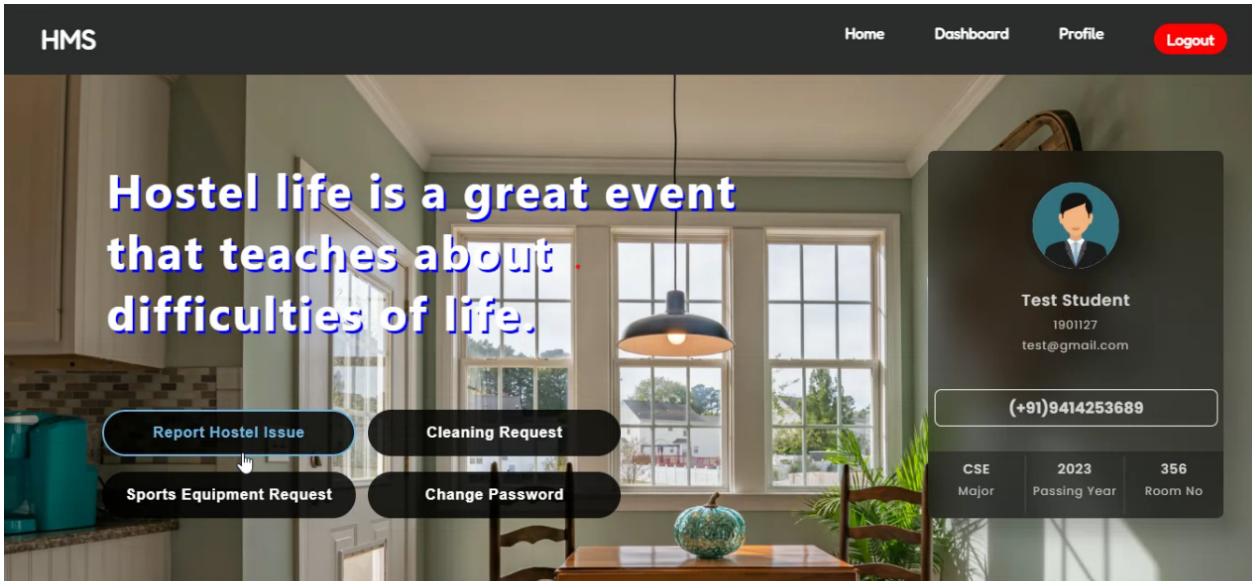
1.1

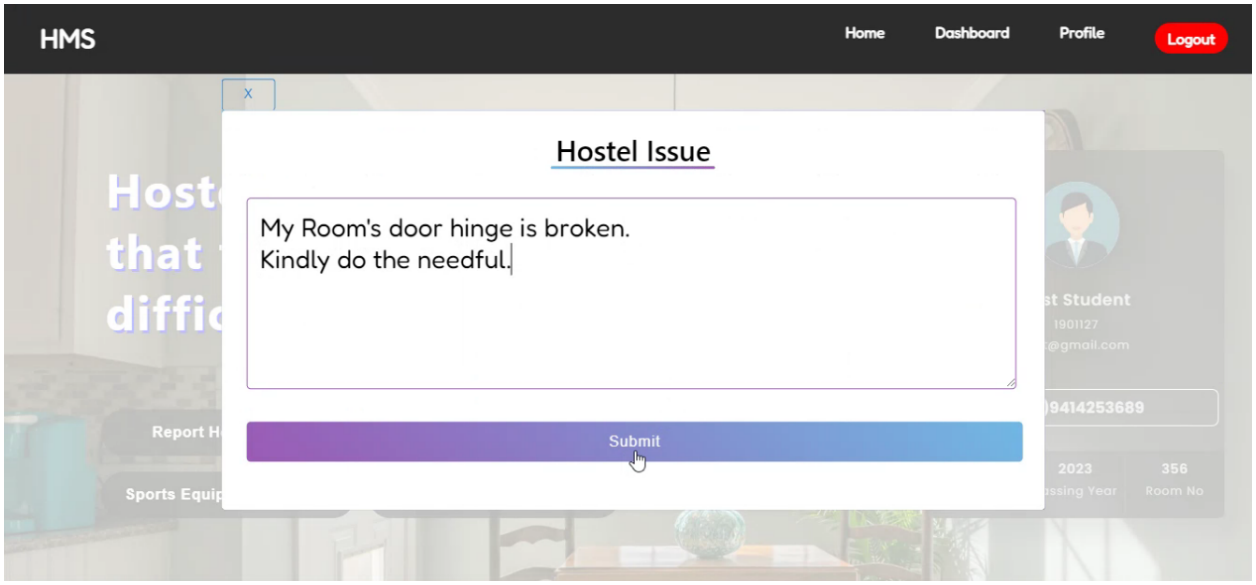
Test Case Description : Students can report an issue regarding hostel or room.

Expected Outcome : Issue is sent successfully and is visible to guard.

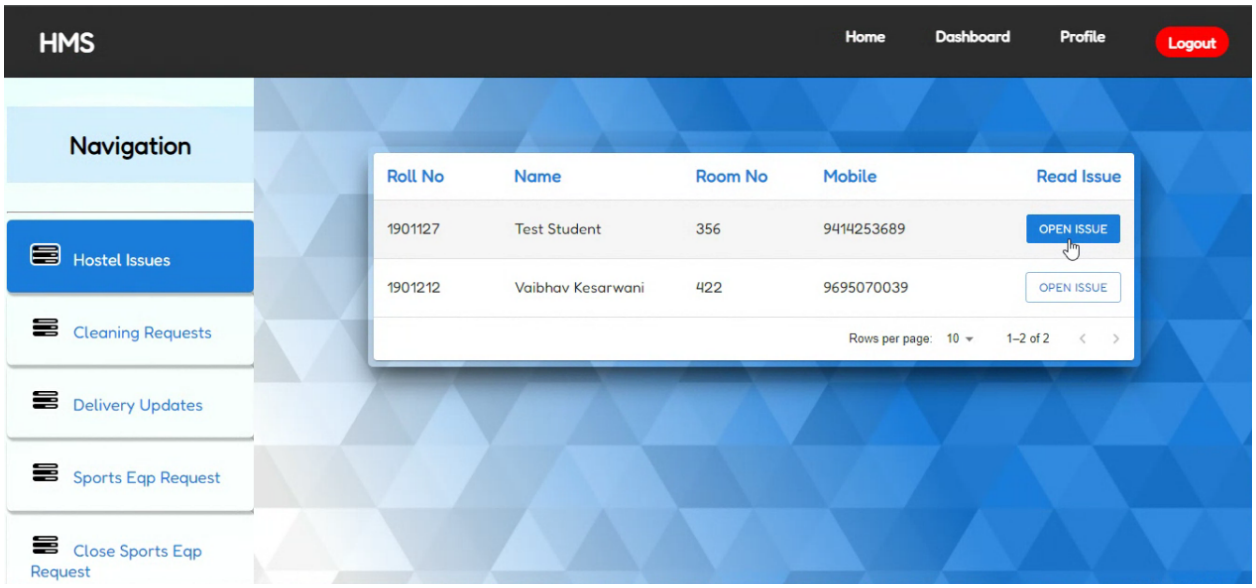
Result : PASSED

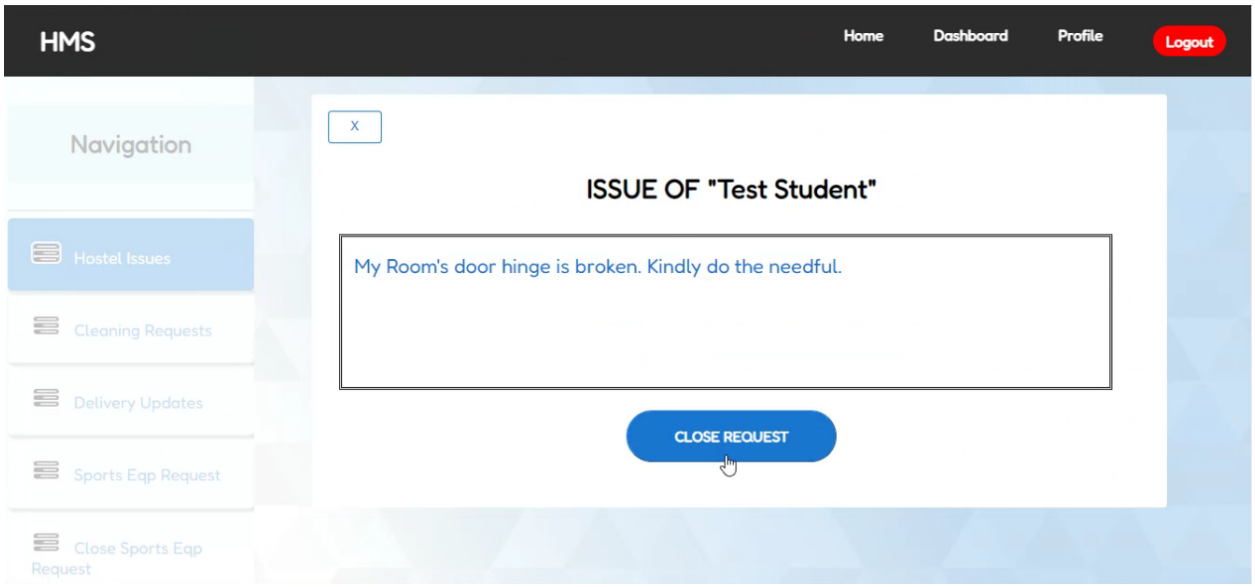
Student Interface ->





Guard Interface ->





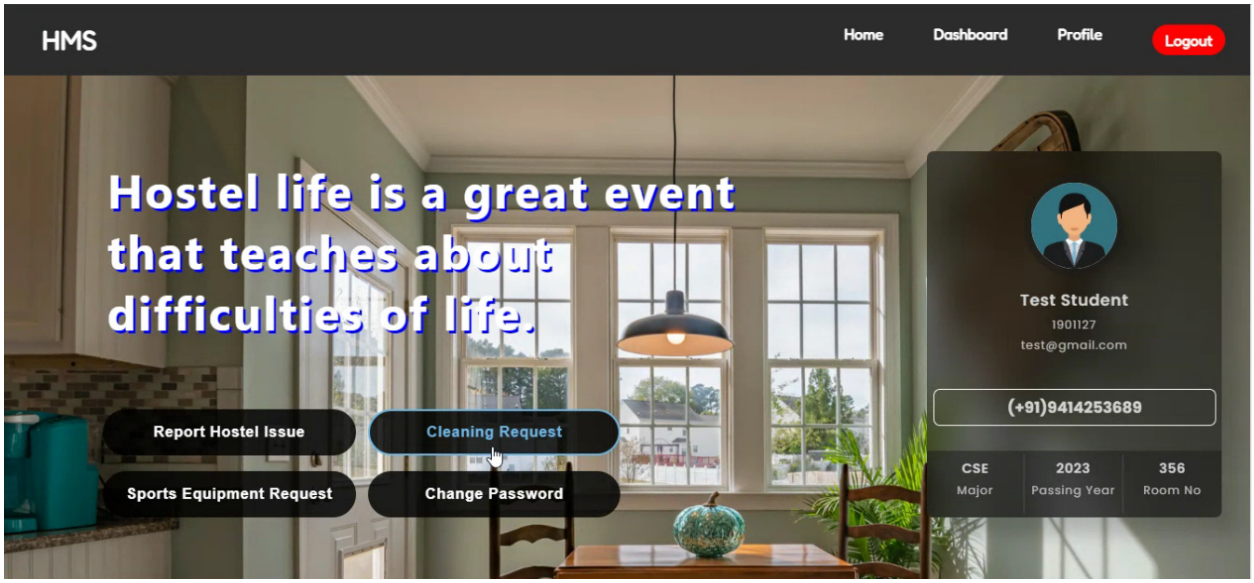
1.2

Test Case Description : Students can request for room cleaning.

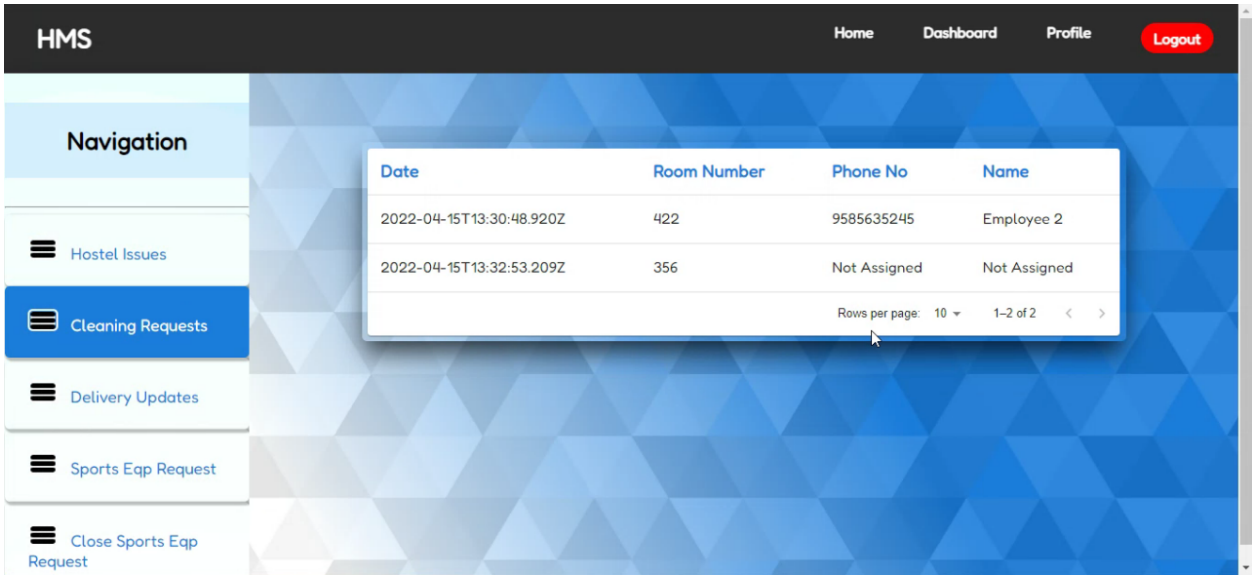
Expected Outcome : Room cleaning request assigned or not assigned in guard interface.

Result : **PASSED**

Student Interface ->



Guard Interface ->



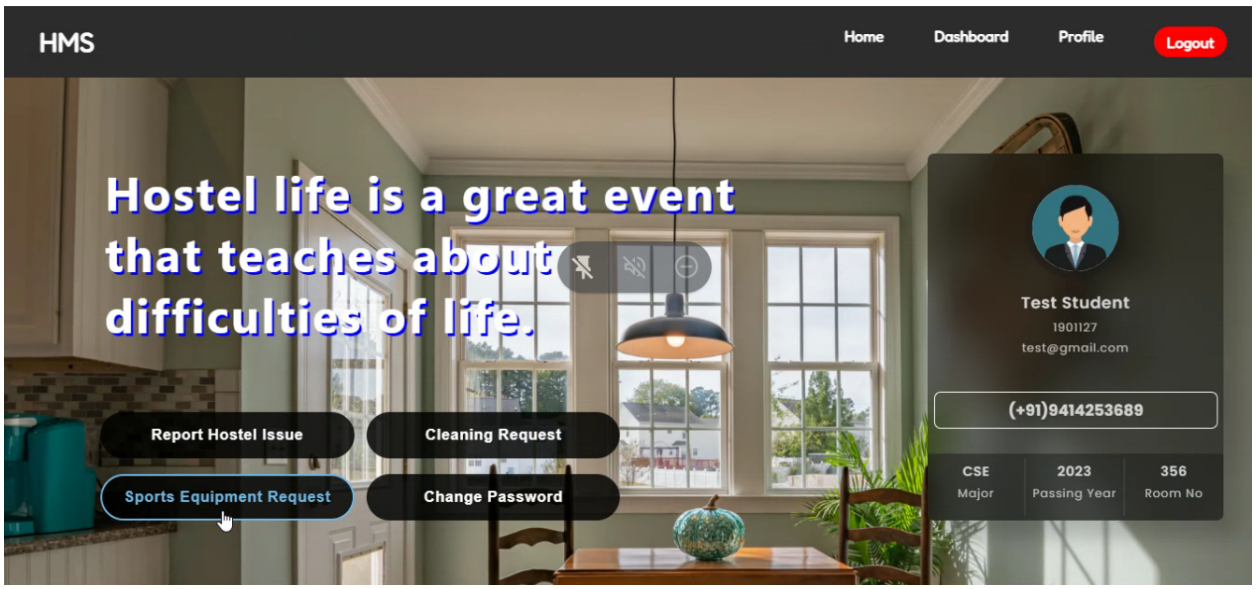
1.3

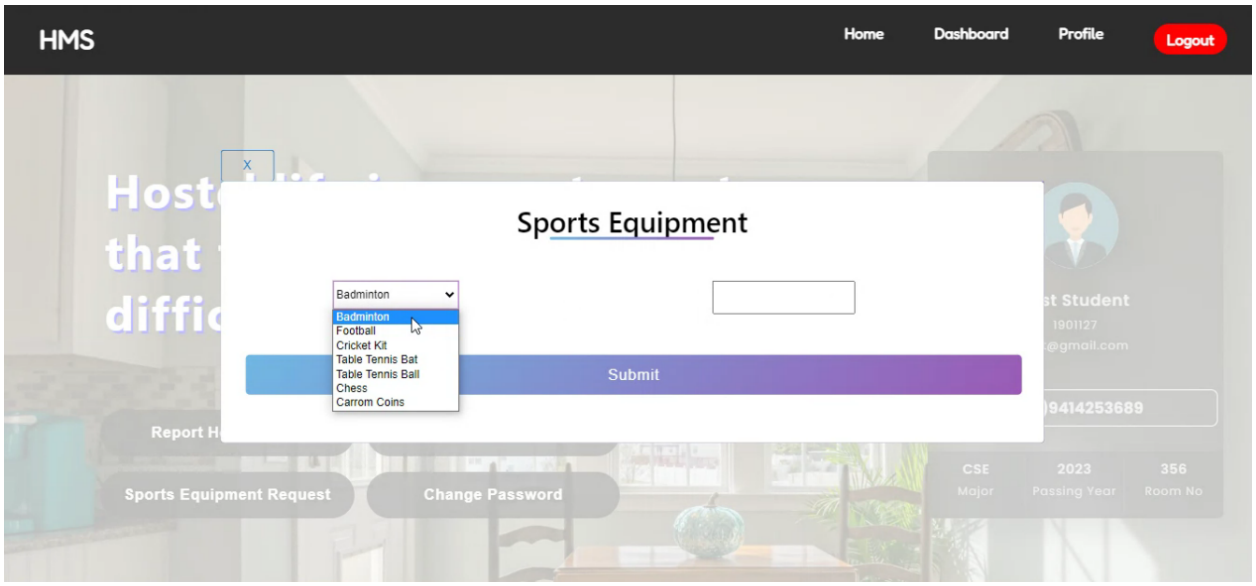
Test Case Description : Students can request sports equipment from the guard.

Expected Outcome : Sports Equipment request is visible to the guard.

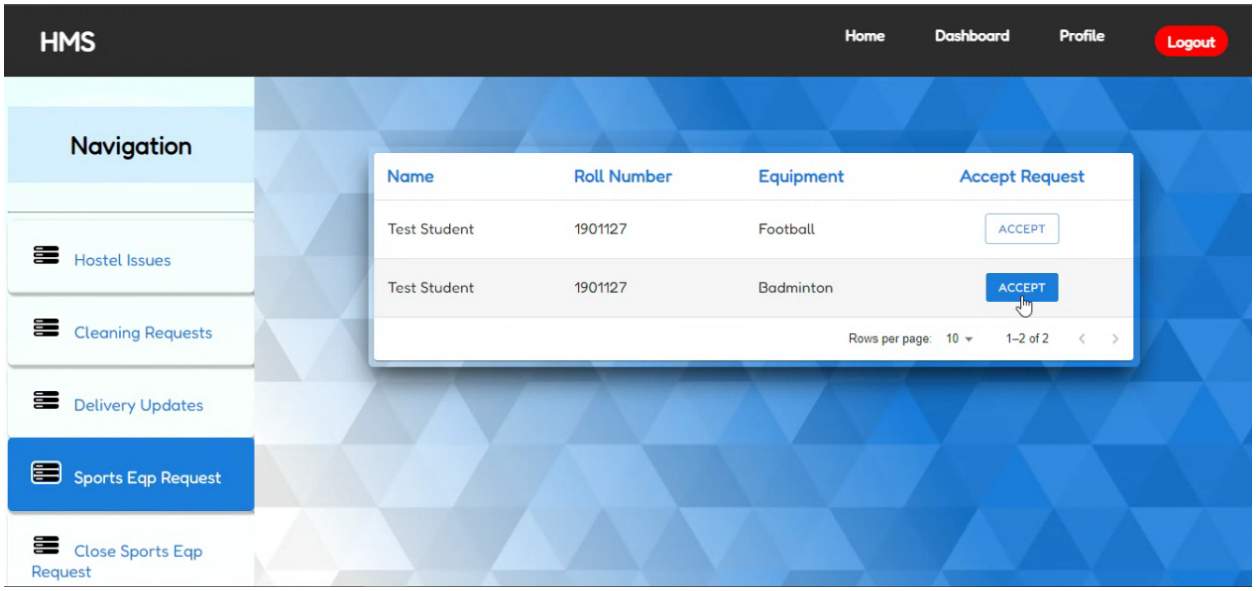
Result : PASSED

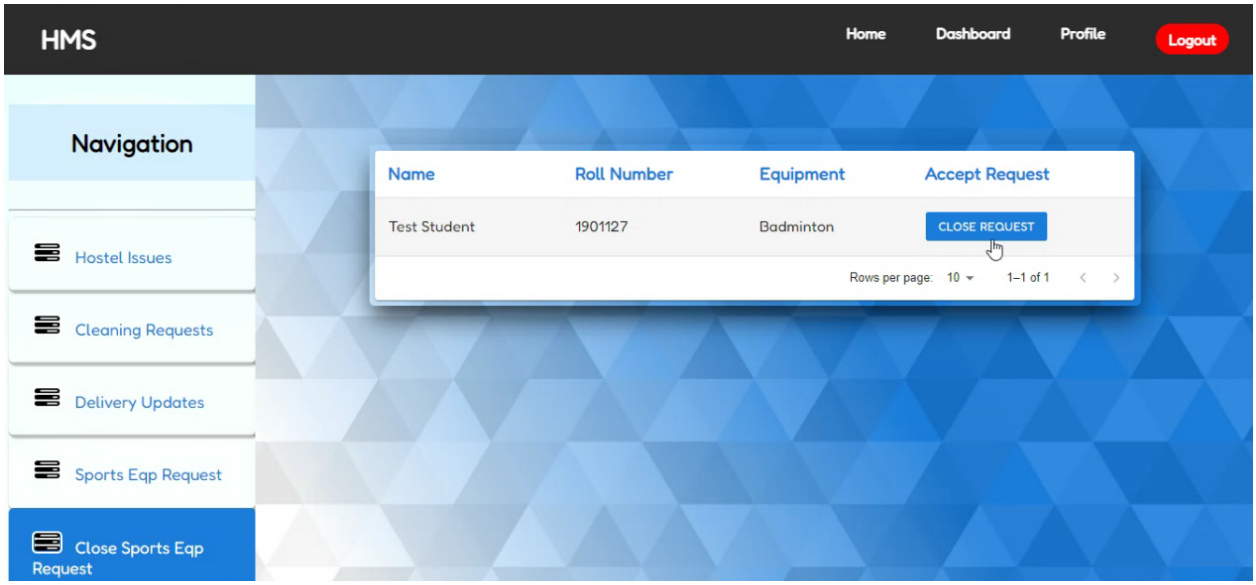
Student Interface ->





Guard Interface ->





Group 2 (Student Interface and Cleaning Staff Interface)

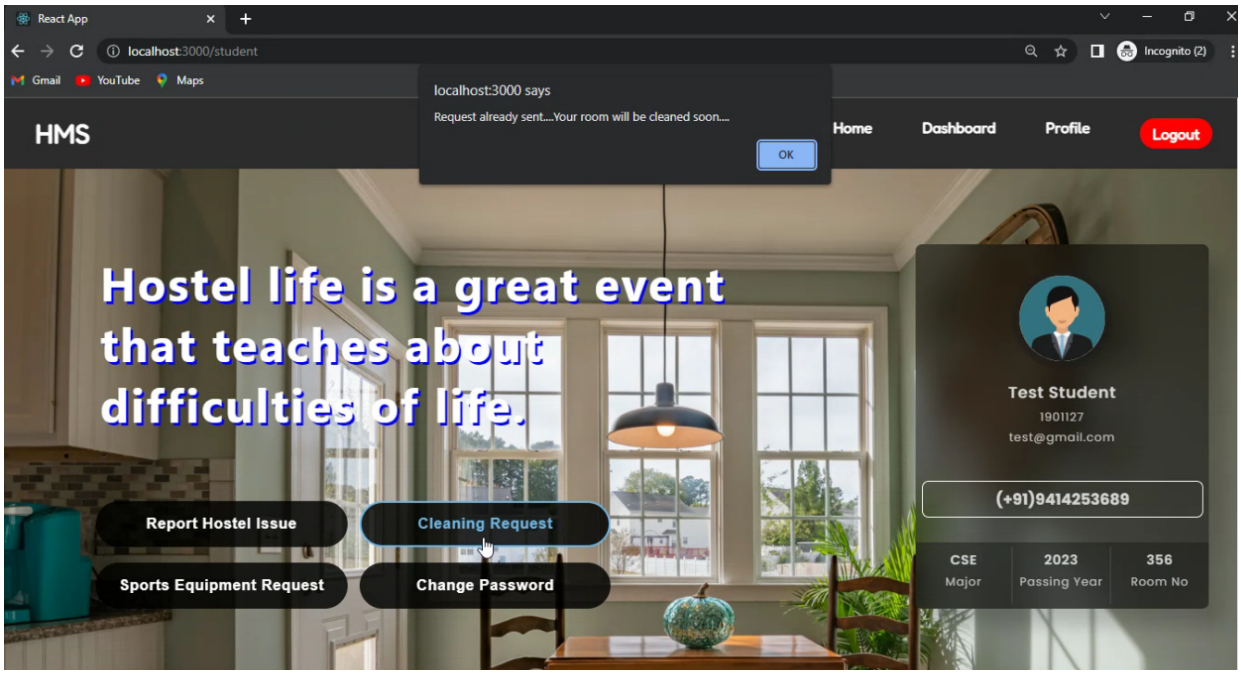
2.1

Test Case Description : Students cannot request multiple cleaning request at a time.

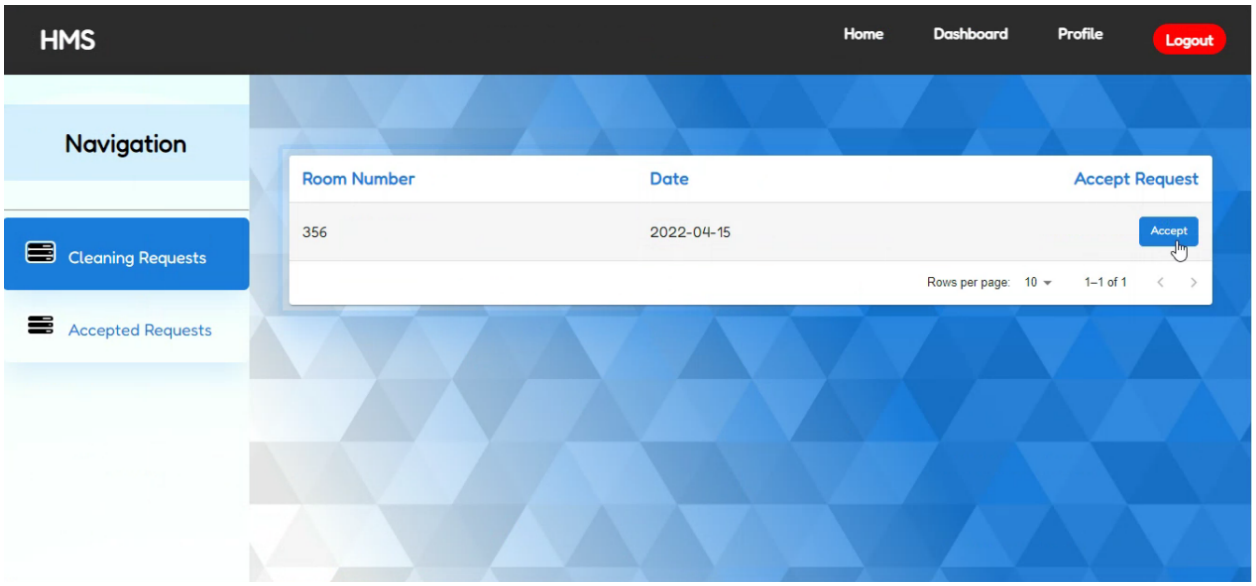
Expected Outcome : Display error message "Request Already Sent" and new request will not be visible in cleaning staff.

Result : PASSED

Student Interface ->



Cleaning Staff Interface ->



Group 3 (Student Interface, Guard Interface and Cleaning Staff Interface)

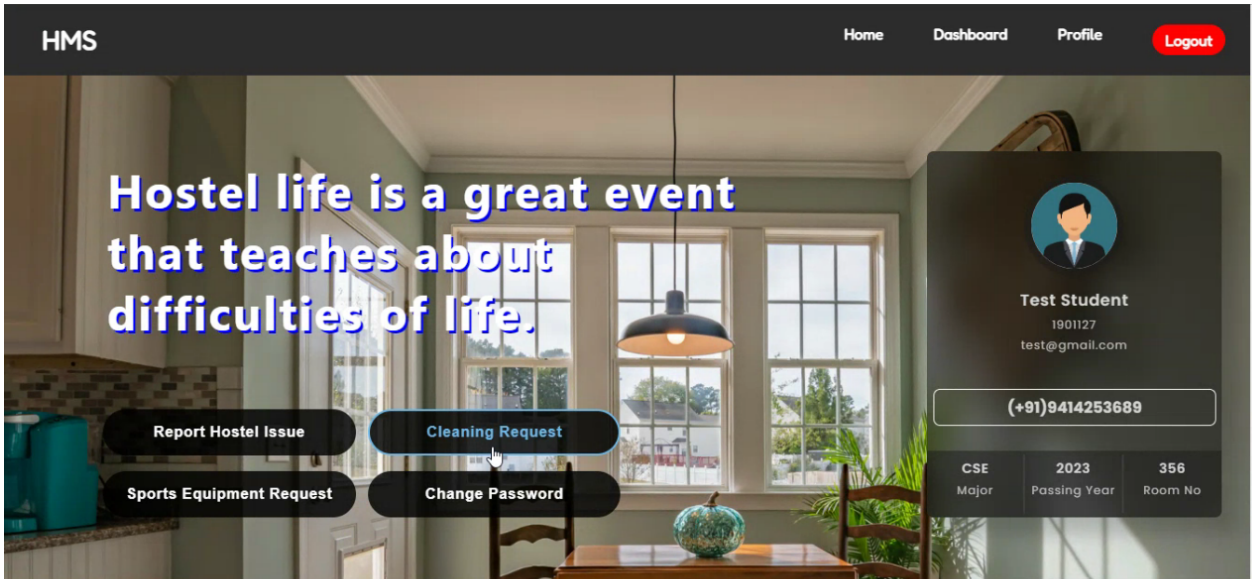
3.1

Test Case Description : Students can request for room cleaning.

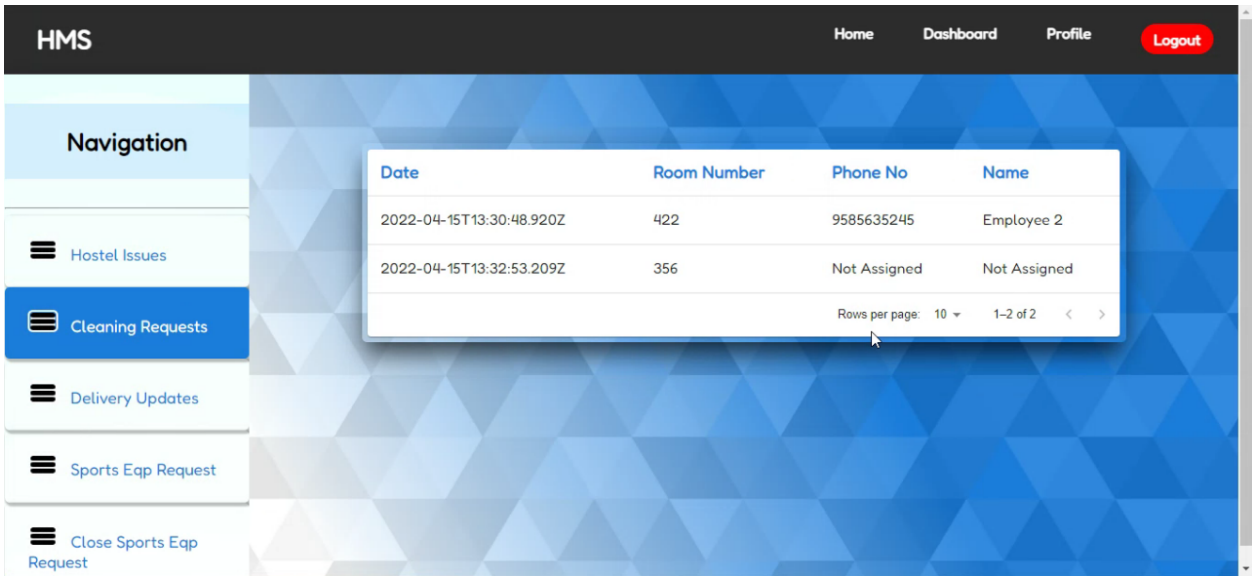
Expected Outcome : Room Cleaning request is sent successfully and is visible to cleaning staff. After a request is accepted by the Cleaning staff, the assigned staff will be reflected in the guard interface.

Result : PASSED

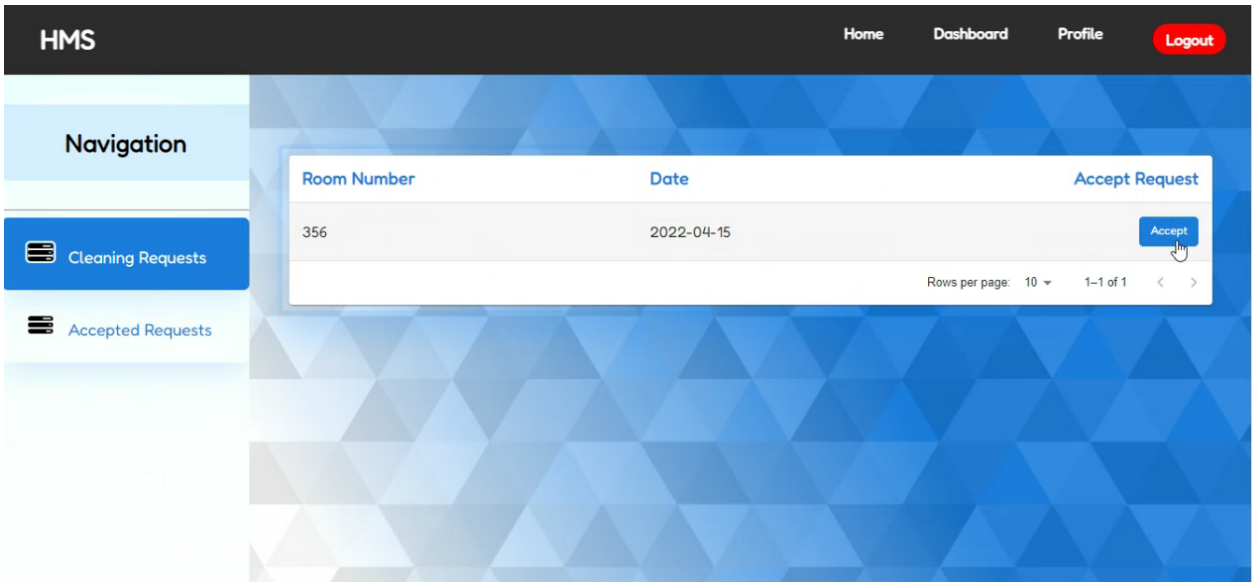
Student Interface ->



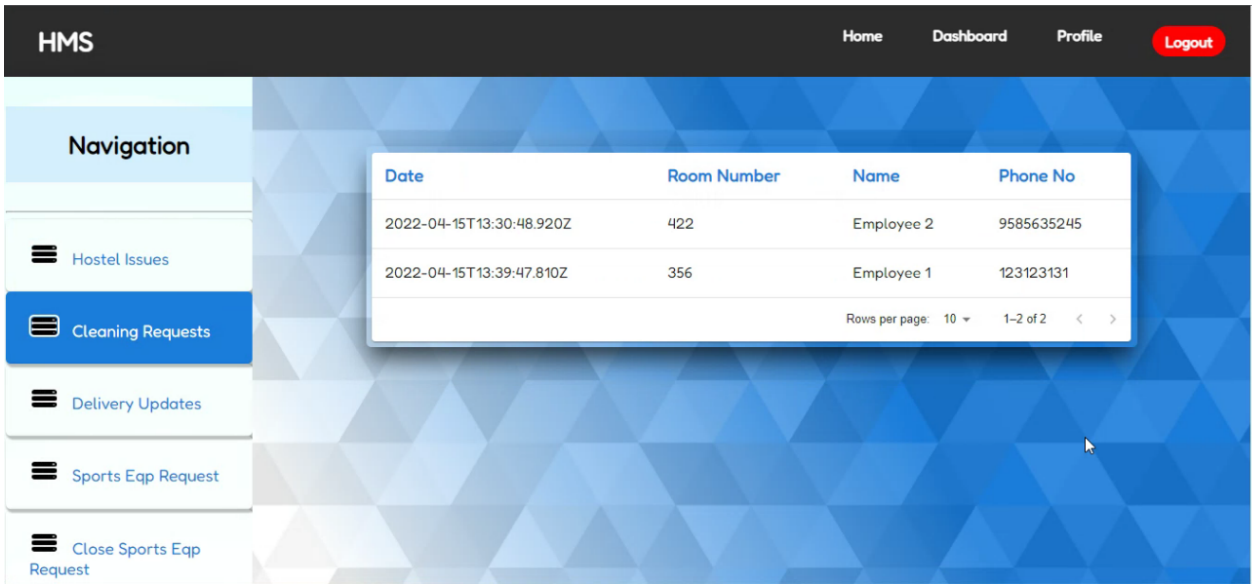
Guard Interface ->



Cleaning Staff Interface ->



Guard Interface ->



After performing the test on the above groups using different components,we fixed all detected irregularities. The Final results are shown above with all the results as PASSED.

System Testing

It is a level of testing that validates the complete and fully integrated software product. The purpose of a system test is to evaluate the end-to-end system specifications.

It is performed on a complete integrated system to evaluate the compliance of the system with the corresponding requirements. We tested the design and behavior of the system and also the expectations of the customer.

Below are the results of the test cases we performed for system testing.

0) LOGIN (For Student, Guard, Warden, Caretaker, Cleaning Staff)

TEST CASES	RESULT
Login with correct email and password	Logged in Successfully
Login with wrong email and password	Invalid credentials error
Login with correct email and wrong password	Invalid credentials error

1) STUDENT :

1.1) SEND HOSTEL ISSUE

TEST CASES	RESULT
On Clicking “Report Hostel Issue” button	Popup appears where we can report hostel related issues.
Report multiple hostel issue	Can successfully report multiple issues.

1.2) ROOM CLEANING REQUEST

TEST CASES	RESULT
Send room cleaning request	Room cleaning request sent, confirmed by message from the backend server
Multiple room cleaning request	Error message displayed : “Request already sent”

1.3) SPORTS EQUIPMENT REQUEST

TEST CASES	RESULT
Request for receiving sports equipment	Request sent successfully

1.4) CHANGE PASSWORD

TEST CASES	RESULT
Entered current password and new password to change the current password	Password changed successfully
Submitted with wrong current password	Error message displayed.
Submitted with wrong confirm new password	Error message displayed.

2) CLEANING STAFF

TEST CASES	RESULT
Accept cleaning request	Accepted successfully
Close room cleaning request	Closed successfully

3) GUARD

TEST CASES	RESULT
Resolved an Issue	Issue resolved successfully
Submitted with wrong confirm new password	Error message displayed.
Change status of delivery after product is handed to student	Changed status to “Delivered”
Add new delivery	Delivery added successfully
Accept sports equipment request	Accepted after checking availability
Received sports equipment back	Closed sports equipment request successfully

4) WARDEN

TEST CASES	RESULT
Add new student	New student added successfully
Edit existing student’s data	Edited data successfully
Delete existing student	Deleted successfully

5) CARETAKER

TEST CASES	RESULT
Add new guard	New guard added successfully
Edit existing guard's data	Edited data successfully
Delete existing guard	Deleted successfully
Add new cleaning staff	New cleaning staff added successfully
Edit existing cleaning staff's data	Edited data successfully
Delete existing cleaning staff	Deleted successfully

On executing the above tests we resolved all detected irregularities among all the unit tests that were integrated together. We made sure that all the features were working as expected and the system as a whole is working smoothly in all circumstances.