
Test Document

for

HallBuddy

Version 1.2

Prepared by

Group : 12

Ritesh Hans
Mridul Gupta
Rohan Batra
Krutuparna Paranjape
Mrdul Agarwaal
Apoorv Tandon
Tanishq Maheshwari
Taneshq Zendey
Ayush
Samarpan Verma

220893
220672
210868
210536
210632
220192
221128
221123
220259
220943

Group Name: CodeMonk

riteshhans22@iitk.ac.in
mridulg22@iitk.ac.in
rohanb21@iitk.ac.in
krutuparna21@iitk.ac.in
mrdula21@iitk.ac.in
apoorvt22@iitk.ac.in
tanishqm22@iitk.ac.in
taneshq22@iitk.ac.in
ayushs@iit.ac.in
samarpanv22@iitk.ac.in

Course: CS253

Mentor TA: *Vaibhav Tanwar*

Date: 29/03/2024



CONTENTS.....	II
REVISIONS.....	II
1 INTRODUCTION.....	1
2 UNIT TESTING.....	2
3 INTEGRATION TESTING.....	3
4 SYSTEM TESTING.....	4
5 CONCLUSION.....	5
APPENDIX A - GROUP LOG.....	6

Revisions

Version	Primary Author(s)	Description of Version	Date Completed
1.1	Mridul Gupta Taneshq Zendey Krutuparna Paranjape Rohan Batra Mrdul Agarwal Tanishq Maheshwari Samarpan Verma Apoorv Tandon Ritesh Hans Ayush	This is the first version of the test document, containing details of unit tests, integration tests and system tests.	01/04/24

1 Introduction

Mention the test strategy

We conducted testing for our software using a combination of automated and manual methods. We prioritized automated testing wherever feasible. Selenium and Django Framework were used to automate tests. Automation of some of the unit test cases was done through the Django Framework. Some aspects of Integration Testing were automated using Selenium. System testing was done manually.

When was the testing conducted?

Testing occurred at multiple stages of application development, including both the implementation and post-implementation phases. During implementation, we primarily utilized manual testing methods, such as incorporating print statements, to assess individual functions and their integration. This proactive approach enabled us to identify and resolve numerous bugs in the code even before completing the implementation phase.

After the implementation was done, we started our testing both manually and using automatic tools. During this phase, we collaboratively wrote unit tests for the various units involved and suggested the required changes to the developers of the respective units/segments.

Who were the testers?

In this project, our team members assumed dual roles as both developers and testers of the application. We conducted testing on applications that we personally developed, as well as those that were implemented by other developers within the team.

What coverage criteria were used?

The testing covered the code exhaustively. All functions were tested with sufficient numbers and a variety of test cases. Also, all possible conditions inside each branching statement were tested using test cases that were set up correctly.

Have you used any tool for testing?

We have used the tools :Django Test and Selenium , which are widely used open-source automated testing frameworks . Django tests were utilised for Unit Testing whereas Selenium was used for Integration Testing.

2 Unit Testing

Authentication

1. User Sign Up

Unit Details: Tested the sign up function to register new users.

Manual Testing: Yes

Automated Testing: Yes

Test Owner: Mridul Gupta

Test Date: [03/22/2024] - [03/24/2024]

Test Results:

Test Case	Expected Result	Test Outcome
Register new hall manager	Raise error, only one hall manager is allowed	Error Raised
Register new student with existing user id	Show message, user already exists	Message displayed
Register new student with non - existing user id	Register user	User is registered

Structural Coverage: [Mention any coverage criteria that you have used to measure test completeness]

Additional Comments: [Enter any summary comments]

2. User Login

Unit Details: Tested user login function

Manual Testing: Yes

Automated Testing: Yes

Test Owner: Mridul Gupta

Test Date: [03/22/2024] - [03/24/2024]

Test Results:

Test Case	Expected Result	Test Outcome
Login user with invalid or unregistered credentials	Show message invalid password or username	Message displayed
Login user with correct credentials	User is logged in	User logged in

Structural Coverage: *[Mention any coverage criteria that you have used to measure test completeness]*

Additional Comments: *No comments*

3. Password function test

Unit Details: Tested set password and reset password functions.

Manual Testing: Yes

Automated Testing: Yes

Test Owner: Mridul Gupta

Test Date: *[03/22/2024] - [03/24/2024]*

Test Results:

Test Case	Expected Result	Test Outcome
Password doesn't meet the constraints specified	Invalid password message	Invalid password message.
New Password and confirm password doesn't match	Error raise that passwords do not match	Error is raised
New password and confirm password match and satisfy the password constraints	Password is changed successfully.	Password is changed

Complaints

4. Lodge Complaint Test

Unit Details: Unit for Lodging a new complaint [Lodge_Request() function]

Manual Testing: Yes

Automated Testing: Yes

Test Owner: *Krutuparna Paranjape*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results:

Test Case	Expected Result	Test Outcome
Lodging new complaint without entering room details	Error raise "Please fill out this field"	Error is raised

Lodging a new complaint without entering category/sub-category	Error is raised "Please enter this field"	Error is raised
--	---	-----------------

A logged-in user is able to see the complete list of all current pending complaints(lodged by all users) including the complaints lodged by the user himself. Every complaint includes details related to location, date,time, and complaint-description.

Structural Coverage: The branch of the 'complaints' view function responsible for lodging a new complaint was covered in this test case.

5. *Resolve complaints Test*

Unit Details: Unit for marking a complaint as 'Resolved'

Manual Testing: Yes

Automated Testing: Yes

Test Owner: *Krutuparna Paranjape*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results: Admin is able to view the current list of all pending complaints with all details correctly validated.

Test Case	Expected Result	Test Outcome
Marking a complaint lodged by other users as resolved	Error raised: "You are unauthorized to mark the status of this request."	Error is raised
Marking an own complaint as resolved	Message raised:"Your request has been removed from pending requests successfully"	Test successful , correct message raised.

Structural Coverage: *The branch of the Pending_Request function responsible for marking a complaint as resolved*

6. Lodging a new complaint

Unit Details: Unit to lodge a new complaint [Lodge_Request()]

Manual Testing: Yes

Automated Testing: Yes

Test Owner: *Krutuparna Paranjape*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results: A logged-in user can successfully lodge a new complaint by clicking on the 'new-complaint' button. The complaint gets appended in the 'Pending Complaints' page.

Structural Coverage: *[Mention any coverage criteria that you have used to measure test completeness]*

7. Updating Status of a Complaint

Unit Details:

Manual Testing: Yes

Automated Testing: Yes

Test Owner: *Krutuparna Paranjape*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results:

A logged-in user is successfully able to update the status of a pending complaint lodged only by him by marking it as 'Resolved'. A user can update the status of only the complaints lodged by him and not others.

Upon marking a complaint as 'resolved' the complaint is removed from the list of current pending complaints (from both user and admin pages).

Structural Coverage: *[Mention any coverage criteria that you have used to measure test completeness]*

Room Cleaning

8. *Record of the Room Cleaning status for the days marked on the calendar(User Login)*

Unit Details: Unit to display/update the room cleaning status for each day.

Manual Testing: Yes

Test Owner: *Taneshq Zendey*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results: A logged-in user can view the Room Cleaning status for each day on the calendar. The cleaning status for a day can be updated by selecting the date and then selecting OK(if the room was cleaned) or Cancel(if the room wasn't cleaned) on the prompt. The user is also prompted to enter the room number.

Structural Coverage: *[Mention any coverage criteria that you have used to measure test completeness]*

9. *Display of the record of the Room Cleaning status of a specified room(Admin Login)*

Unit Details: Unit to display the room cleaning status for a room.

Manual Testing: Yes

Test Owner: *Taneshq Zendey*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results: Admin is able to see the Room Cleaning status(updated by the users) of a room by entering the room number and the date. If no date is entered, the entire record of the cleaning status of the room is displayed.

Structural Coverage: *[Mention any coverage criteria that you have used to measure test completeness]*

Guest Room Booking

10. *Placing a request for guest room booking*

Unit Details: Unit to display the status of booking a guest room

Manual Testing: Yes

Test Owner: *Mrdul Agarwaal*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results

Test Case	Expected Result	Test Outcome
Checkout date is before check in date	Raise error	Error Raised
Check in date is before the present date	Raise error	Error Raised
The room is available on the said date and the above errors are not present	Message displayed of "Booking done"	Success message displayed

Structural Coverage: Branch coverage of booking a new guest room

11. *Approving a request for guest room booking*

Unit Details: Unit to approve/reject a request for booking a guest room

Manual Testing: Yes

Test Owner: *Mrdul Agarwaal*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results:

Test Case	Expected Result	Test Outcome
Approve / reject guest room booking by admin	The request should be approved/rejected	The request is approved/rejected

Structural Coverage: Branch coverage of approving a guest room booking by the admin

12. Adding a guest room

Unit Details: Unit to check the response when a guest room is added

Manual Testing: Yes

Test Owner: *Mrdul Agarwaal*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results:

Test Case	Expected Result	Test Outcome
Guest room is added without room number	Systems prompts admin to add room number	Error Raised
Guest room added without price	Systems prompts admin to add price	Error raised
All Fields are filled correctly	Room should be added	Room is added

Structural Coverage: Branch coverage of adding a guest room by admin

13. Updating a guest room

Unit Details: Unit to check the response when a guest room is updated by admin

Manual Testing: Yes

Test Owner: *Mrdul Agarwaal*

Test Date: *[03/22/2024] - [03/27/2024]*

Test Results:

Test Case	Expected Result	Test Outcome
Guest room is updated without room number	Systems prompts admin to update room number	Error Raised
Guest room updated without price	Systems prompts Hall Manager to update price	Error raised
All Fields are updated correctly	Room should be updated	Room updated

Structural Coverage: Branch Coverage of updating a guest room by admin

Announcements

14. Admin can add new announcements

Unit Details: Unit to create a new announcement

Manual Testing: Yes

Automatic Testing: Yes

Test Owner: Rohan Batra

Test Date: [03/22/2024] - [03/28/2024]

Test Results:

Test Case	Expected Output	Actual Output
New Announcement is added using the Add Announcement Button	The new announcement should be added correctly with the current date and given description	The new announcements is created with the current date and the entered input.

Structural Coverage: Branch Coverage of creating new announcements.

15. Admin cannot create empty description announcements

Unit Details: Unit to create a new announcement

Manual Testing: Yes

Test Owner: Rohan Batra

Test Date: [03/22/2024] - [03/28/2024]

Test Results:

Test Case	Expected Output	Actual Output
Creation of Announcement with empty description	The new announcement should not be created	The description field is a compulsory field, and the announcement is not submitted with an empty field.

Structural Coverage: Branch Coverage of creating new announcements.

16. Date of Announcement

Unit Details: Unit to create a new announcement

Manual Testing: Yes

Test Owner: Rohan Batra

Test Date: [03/22/2024] - [03/28/2024]

Test Results:

Test Case	Expected Output	Actual Output
The date of announcement creation	The admin should not be allowed to provide the date of announcement and should be taken automatically	It was actually possible for the admin to provide the date for the announcement in the form

This bug was identified and corrected, not allowing any manual date input, and the date was taken manually.

Structural Coverage: Branch Coverage of creating new announcements.

17. Admin can delete the created announcements

Unit Details: Unit to delete the announcement

Manual Testing: Yes

Test Owner: Rohan Batra

Test Date: [03/22/2024] - [03/28/2024]

Test Results:

Test Case	Expected Output	Actual Output
The announcement can be deleted by admin using the delete button	The announcement should be deleted	The corresponding announcement is deleted and the message is displayed correctly

Structural Coverage: Branch Coverage of deleting the announcements.

Additional Comments: No comments

18. Admin can edit the announcements

Unit Details: Unit to delete the announcement

Manual Testing: Yes

Test Owner: Rohan Batra

Automatic Testing: Yes

Test Class: Tests

Test Function: test2_edit_annnc

Test Date: [03/22/2024] - [03/28/2024]

Test Results:

Test Case	Expected Output	Actual Output
The announcement can be edited by the admin using the Edit Announcement button.	The admin should be able to update the description of the announcement, and the updated announcement should be displayed henceforth.	The admin is able to edit the announcement description and the updated announcement is displayed with the appropriate message.

Structural Coverage: Branch Coverage of deleting the announcements.

Additional Comments: No comments

3 Integration Testing

1. Checking the interface between the signup and the login module

Module Details: Authentication Section

Test Owner: *Mridul Gupta*

Test Date: 28/03/2024

Test Results: *On successfully signing up for the application using the valid credentials, the user gets redirected to the login page.*

2. Checking the interface between the login module and the home module

Module Details: Authentication Section

Test Owner: *Mridul Gupta*

Test Date: 28/03/2024

Test Results: *On entering the correct credentials on the login page, the user is redirected to the homepage, showing the available products for bidding.*

```
def student_login(self):
    username = self.driver.find_element(by=By.NAME, value="username")
    password = self.driver.find_element(by=By.NAME, value="password")
    submit_button = self.driver.find_element(by=By.CSS_SELECTOR, value="button")

    username.send_keys("mridulg22")
    password.send_keys("P@ssword12")
    submit_button.click()
```



```
def admin_login(self):  
    username = self.driver.find_element(by=By.NAME, value="username")  
    password = self.driver.find_element(by=By.NAME, value="password")  
    submit_button = self.driver.find_element(by=By.CSS_SELECTOR, value="button")  
  
    username.send_keys("admin")  
    password.send_keys("admin")  
    submit_button.click()
```

3. Testing effect of logout on the website

Module Details: Log-Out Section

Test Owner: *Mridul Gupta*

Test Date: 29/03/2024

Test Results: On logging out from the application, the user gets redirected to the login page and the associated cookies get cleared so as not to authorize the user anymore.

```
def logout(self):  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text12").click()
```

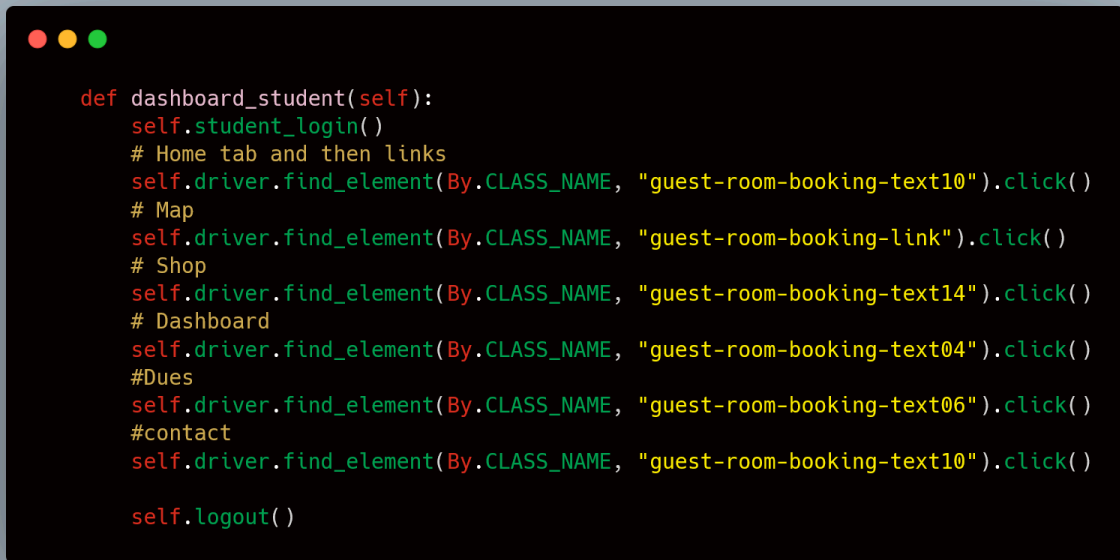
4. Tested the navigation across all segments and subcategories on the student interface.

Module Details: Home - Section

Test Owner: Mridul Gupta

Test Date: *[03/28/2024] - [03/28/2024]*

Test Results: Code was written to open various sections and subsections from the students side and check their navigation. All the navigations were performed successfully.



```
def dashboard_student(self):
    self.student_login()
    # Home tab and then links
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text10").click()
    # Map
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-link").click()
    # Shop
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text14").click()
    # Dashboard
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text04").click()
    #Dues
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text06").click()
    #contact
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text10").click()

    self.logout()
```

5. Tested the navigation across all segments and subcategories on the Hall-Administration interface.

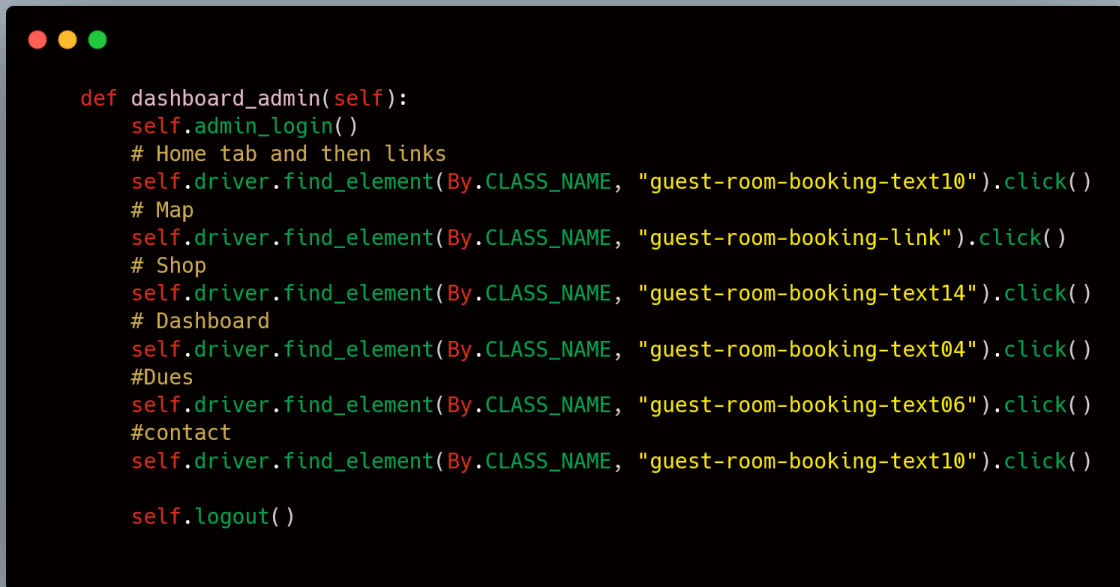
Module Details:

Test Owner: *Mridul Gupta*

Test Date: *[03/28/2024] - [03/28/2024]*

Test Results: Code was written to open various sections and subsections from the Hall Manager's side and check their navigation. All the navigations were performed successfully.

Remarks: The dues section is not viewable by the hall manager.



```
def dashboard_admin(self):  
    self.admin_login()  
    # Home tab and then links  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text10").click()  
    # Map  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-link").click()  
    # Shop  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text14").click()  
    # Dashboard  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text04").click()  
    #Dues  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text06").click()  
    #contact  
    self.driver.find_element(By.CLASS_NAME, "guest-room-booking-text10").click()  
  
    self.logout()
```

6. Tested Visibility of all Pending Complaints (Student login)

Module Details: Complaints-Section

Test Owner: *Krutuparna Paranjape*

Automation : Yes

Test Date: [03/26/2024] - [03/26/2024]

Test Results:

Upon lodging a new complaint, the complaint is displayed on the Complaints page in the list of all pending complaints. The complaint is displayed with correct details of time, place, category and sub-category.

7. Tested Visibility of all Pending Complaints (Admin login)

Module Details: Complaints-Section

Test Owner: *Krutuparna Paranjape*

Automation : Yes

Test Date: [03/26/2024] - [03/26/2024]

Test Results:

Upon login and navigating to the complaints page, the admin is able to view the entire list of all pending complaints with correct details of time, place, etc.

```
def complaints_student(self):
    self.student_login()
    self.driver.implicitly_wait(0.5)
    self.driver.find_elements(By.CLASS_NAME, "guest-room-booking-text16")[0].click()

    #clicking on new-complaint
    self.driver.find_elements(By.CLASS_NAME, "btn.btn-success")[0].click()

    #making requests

    #request 1 :
    self.driver.find_elements(By.CLASS_NAME, "room_text")[1].send_keys("D-213")

    #complaint sub-cateogry and category :
    self.driver.find_element(By.ID, "dropdown1").send_keys("Electrical")
    self.driver.find_element(By.ID, "dropdown2").send_keys("Fan not working")

    self.driver.find_elements(By.CLASS_NAME, "comments_rec")[0].send_keys("Room check")
    self.driver.find_elements(By.CLASS_NAME, "submit_rec")[0].click()
    self.logout()

    #admin_login
    self.driver.implicitly_wait(0.5)
    self.admin_login()
    self.driver.explicitly_wait(5)
    self.driver.find_elements(By.CLASS_NAME, "guest-room-booking-text16")[0].click()
    assert(self.driver.find_elements(By.CLASS_NAME, "a")[0].text == "mridulg22")
```

8. Tested updating of complaint status upon marking as resolved

Module Details: Complaints-Section

Test Owner: *Krutuparna Paranjape*

Automation : Yes

Test Date: [03/26/2024] - [03/26/2024]

Test Results:

Upon login, navigating to the complaints page, and marking a complaint as resolved, the complaint is successfully removed from the list of pending complaints on the complaints-page.

```
def mark_resolved_complaint(self):  
    self.student_login()  
    self.driver.implicitly_wait(0.5)  
    self.driver.find_elements(By.CLASS_NAME, "guest-room-booking-text16")[0].click()  
  
    # mark_resolved  
    self.driver.find_elements(By.CLASS_NAME, "btn.btn-outline-success.btn-sm")[  
        0  
    ].click()  
  
    # try to mark some-one else's complaint as resolved  
    self.driver.find_elements(By.CLASS_NAME, "btn.btn-outline-success.btn-sm")[  
        1  
    ].click()
```

9. Announcements Creation and Visibility by the residents

Module Details: Home-Announcements Section

Test Owner: Rohan Batra

Automatic Testing: Yes

Test Class: Tests

Test Function: test1_new_annc

Test Date: 28/03/2024

Test Results: When the admin adds the new announcement using the Add Announcement Button, it is added successfully with the entered description and date. The admin is able to view the complaint and the correct operation message.

The residents are able to view the newly added complaint marked by “NEW” with the correct description and date.

```
def test1_new_annnc(self,catch_item):
    # Announcement being added correctly in the admin page
    self.admin_login()
    self.driver.find_element(By.NAME, "add_hidden_item").click()
    self.driver.find_element(By.NAME, catch_item).send_keys("Test Announcement")
    self.driver.find_element(By.NAME, "submit").click()
    assert(self.driver.find_elements(By.CLASS_NAME, "alert-success")[0].text == "Changes made
successfully.")
    time.sleep(5)
    self.logout()
    # New Announcement visible to the user
    self.student_login()
    assert(self.driver.find_elements(By.TAG_NAME, "td")[4].text == "Test Announcement" and
self.driver.find_elements(By.TAG_NAME, "td")[5].text ==datetime.date.today().strftime("%B %d, %Y"))
    time.sleep(5)
    self.logout()
```

Additional Comments: None

10.Edited Announcements and updated Visibility by the residents

Module Details: Home-Announcements Section

Test Owner: *Rohan Batra*

Automatic Testing: Yes

Test Class: Tests

Test Function: test2_edit_annnc

Test Date: 28/03/2024

Test Results: After the admin updates the announcement with the new description, the announcement is updated correctly, and the message is displayed.

The residents are able to view the complaint with the updated description.

```
def test2_edit_annnc(self, catch_item):  
    # Announcement being edited correctly in the admin page  
    self.admin_login()  
    self.driver.find_element(By.NAME, "edit").click()  
    self.driver.find_element(By.NAME, catch_item).clear()  
    self.driver.find_element(By.NAME, catch_item).send_keys("Test Announcement Edited")  
    self.driver.find_element(By.NAME, "submit").click()  
    assert(self.driver.find_elements(By.CLASS_NAME, "alert-success")[0].text == "Changes made  
successfully.")  
    time.sleep(5)  
    self.logout()  
    # Edited Announcement visible to the user  
    self.student_login()  
    assert(self.driver.find_elements(By.TAG_NAME, "td")[4].text == "Test Announcement Edited" and  
self.driver.find_elements(By.TAG_NAME, "td")[5].text == datetime.date.today().strftime("%B %d, %Y"))  
    time.sleep(5)  
    self.logout()
```

Additional Comments: None

11. Updation of Room Cleaning status and Visibility by the Admin.

Module Details: Cleaning

Test Owner: *Taneshq Zendey*

Manual Testing: Yes

Test Date: 26/03/2024

Test Results: After a student updates the Room Cleaning status, the changes are reflected on the calendar.

The Admin is able to view the updated status entry in the room's cleaning record

Additional Comments: None

12. Requesting booking of guest room and visibility by the admin

Module Details: Guest Room Booking

Test Owner: *Mrdul Agarwaal*

Manual Testing: Yes

Test Date: 26/03/2024

Test Results: If a student requests to book a guest room, then the hal manager can see his request in his portal and either approve/reject it.

13. Approving the room request via admin and its visibility to the student

Module Details: Cleaning

Test Owner: [Mrdul Agarwaal](#)

Manual Testing: Yes

Test Date: 26/03/2024

Test Results: If the admin approves the booking request, then the user can go to the past bookings page and check the status of the booking request which would have been changed to updated.

4 System Testing

1. Requirement: *Testing the complete functionality of authentication (User, must be able to login or reset password if he is registered, else must be able to create an account)*

Test Owner: Mridul Gupta

Test Date: [03/24/2024] - [03/29/2024]

Test Results: *All the tests related to this requirement were successfully executed.*

- *User is able to login successfully using his login credentials*
- *User is able create new account on Sign Up page*
- *Otp is successfully sent when a user registers or reset his password*
- *Multiple users cannot be created using same user id*
- *Only one Hall Manager is allowed*

Additional Comments:

- *The OTP functionality worked as expected, ensuring security during registration and password reset processes.*

2. Requirement: *Testing the options available on home page.*

Test Owner: Mridul Gupta

Test Date: [03/24/2024] - [03/29/2024]

Test Results:

- *Student and hall manager are able to access all links on the home page.*
- *Hall Manager can't visit the dues page as there are no pending dues for him.*

3. Requirement: *Testing the announcements feature.*

Test Owner: Rohan Batra

Test Date: [03/24/2024] - [03/29/2024]

Test Results:

- *The Hall manager is able to add, delete and edit announcements.*
- *All the announcements are visible to all users.*

4. Requirement: *Testing the Guestroom booking feature.*

Test Owner:

Test Date: [03/24/2024] - [03/29/2024]

Test Results: Mrdul Agarwal

- *A student is able to raise a guest room booking request.*
- *All the requests raised on which action has not been taken are visible to the hall manager.*
- *The mail is sent to the student informing him of the action taken on his request.*
- *If a booking request is accepted then all the other requests for the same room and overlapping booking periods with the accepted request are automatically rejected.*
- *Hall manager is able to view all the approved bookings.*

5. Requirement: *Complaints Management feature testing.*

Test Owner: Krutuparna Paranjape

Test Date: [03/24/2024] - [03/29/2024]

Test Results:

- *Users can submit complaints successfully.*
- *All the complaints are visible to all the users.*
- *Only the user who raised the complaint can mark it as resolved.*
- *Resolved complaints are removed from the database.*

6. Requirement: *Cleaning Status Test.*

Test Owner: Taneshq Zendey

Test Date: [03/24/2024] - [03/29/2024]

Test Results:

- *Users are able to mark the status of their room.*
- *User can view his room status for the previous months.*
- *Admin is able to view the cleaning history of all the rooms.*

5 Conclusion

How Effective and exhaustive was the testing?

Each component of the web-based application was tested both independently using unit testing and in coordination with other units through integration and system testing. The functional and non-functional requirements were tested, which in turn helped us make some of the functions in views.py more precise and accurate.

Which components have not been tested adequately?

We have not been able to test the software behaviour on multiple users performing the same operation concurrently. Extremely large test inputs in the description field of various operations (e.g. complaints, announcements etc) have not been tested.

For now, we have kept aside the issues faced in operating the application on mobile. (or any other platform than PC).

What difficulties have you faced during testing?

- Thinking about the exhaustive set of test cases at different levels - unit level, integration level and the complete system level was challenging.
- Learning automation tools like Selenium and Django Test and creating test cases in them was time-consuming.
- Since we are both the testers and the developers of the software, debugging the code to resolve the found test bugs was challenging, mentally stimulating and time-consuming.
- Since our software has multiple stakeholders, coming up with and thinking of the software behaviour through both the admin and the resident and the impact of the interaction of various residents was challenging and mentally stimulating.

How could the testing process be improved?

- Using better automation tools for system testing and further adding comments to the functions in the application during the system implementation phase would have helped a lot in writing test functions.

Appendix A - Group Log

Date	Members Present	Topic Of Discussion
24/03/2024	All group members	Collaboratively discussed divided tasks for unit testing ,integration testing and system testing
28/03/2024	TA and rest group members	Updated progress with the TA and received feedback from the TA. Discussed possibilities of more tests to exhaustively cover all test-cases.