*Hostel Management*

Test report

Version *1.0*

*04/3/2016*

VERSION HISTORY

|  |  |  |
| --- | --- | --- |
| **Version #** | **Implemented**  **By** | **Revision**  **Date** |
| 1.0 | Team 3 | 04/3/201*6* |

*Tested By:*

1. Aakash Hasija
2. Aman
3. Amit Jain

Shubham Saxena

**Table of Contents**

[**1.0** **Introduction**](#h.1fob9te)

[1.1 Purpose](#h.3znysh7)

[**2.0** **Test PLAN**](#h.2et92p0)

[**3.0** **Test Assessment**](#h.tyjcwt)

[**4.0** **Test Results**](#h.3dy6vkm)

[4.1 Unit/Module/System Testing](#h.1t3h5sf)

[4.2 System Testing](#h.4d34og8)

[4.3 User Acceptance Testing](#h.2s8eyo1)

[4.4 Regression Testing](#h.17dp8vu)

[4.5 Performance Testing](#h.3rdcrjn)

[**5.0** **Variances**](#h.35nkun2)

[**6.0** **Test Instances**](#h.1ksv4uv)

[6.1 Resolved Test Incidents](#h.44sinio)

[6.2 Unresolved Test Incidents](#h.2jxsxqh)

[**7.0** **Recommendations**](#h.z337ya)

[**APPENDIX A: REFERENCES**](#h.3j2qqm3)

[**APPENDIX B: KEY TERMS**](#h.1y810tw)

# 1.0 Introduction

## 1.1 Purpose

This **Hostel Management** Test Report provides a summary of the results of test performed as outlined within this document.

# 2.0 Test PLAN

The following sequence(in order) was followed while conducting the testing for the project

* Basic testing
* System Testing
* User Acceptance Testing
* Performance Testing

Basic Testing: The testing team started with a basic walkthrough of the project searching for any loose ends and broken links.

System testing: In this category, we devised a set of test cases and served them to the software. Some passed and some failed and the results were carefully and duly noted down for further analysis.

User Acceptance Testing: The system was viewed from the perspective of the user and its capability to be served to the end users was tested.

Performance Testing: The system was judged under various parameters such as security, consistency and reliability

# 3.0 Test Assessment

Barring some minor issues, every major fault in the system has been identified by the test cases devised by the testing team.

The software developed by the team is very fragile as it seems to produce erroneous output on putting some tedious test cases. For instance, if a user received multiple swap room requests, the system tends to show server error. The backend of the system has taken care of minor details. Many UI details have also been neglected as is evident by many broken links and unused buttons. Also, there is no consistency among user names being shown on the system display screen

# 

# 4.0 Test Results

1. In the User Interface there were many flaws which were making the user experience hard. For eg. the mouse pointer changes to link when we hover on button, even though it is not clickable. The UI also doesn’t acknowledges any of the action. Eg. If we deallocate any of the batch, it directly redirects us to the home page. Rather it should tell the user that request was successful. Also from the user's point, it should ask for confirmation for vital actions like deleting batch of students. It will prevent accidentalactions**.**
2. Lack of Notifications:- There is no mode of notifying users in the Software. Like whenever a person is requested for swapping room, he will not know about it until he opens the request tab. Ideally he should be sent a request that someone has requested to swap room with him. Same is the case with admin. Also some corner cases need to be handled like Cycle in swap room request[A sending request to C. Then B sending request to C. Now If C opens the Swap room Acknowledgement page, page will not be rendered ].
3. Not showing appropriate details: The software doesn’t shows the user or staff many important details about the rooms like current vacancies, room capacity. (Also there is no way anyone can see more than 15 entries in the list.)
4. Problems in algorithmic implementation:- There is some implementation problem in the software. Like it shows room no 300 vacant even if it has only 2 people living in that. Also Ground Floor and Handicapped are not taken care of.

## Unit/Module/System Testing

Unit, module, and system integration testing activities were performed during the development of the system build or release.

## 4.2 System Testing

The table below summarizes the results of system testing

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? | Comments |
| 1 | 3/4/2016 | aman | Fail | High | Adding of new users is not possible. |  | No such option as “Add new  Student”  [Use case 1] |
| 2 | 3/4/2016 | aman | Fail | High | Adding of a whole batch not possible |  | No such option as “Add new Batch”  [Use case 2] |
| 3 | 3/4/2016 | aakash | Fail | Medium | Doesn’t show a room which is vacant in vacant room list. |  | Also in the vacant room list only 15 rooms are visible at max.  [Use case 3] |
| 4 | 3/4/2016 | Amit | Fail | High | 1) Can send swap request to same roomie  2) if a user receives two swap requests then it gives server error on opening the requests section  3) the user can accept a single swap request multiple times  4) when there is a cycle of swap requests, the site results into server error |  | [Use case 9] |
| 5 | 3/4/2016 | aakash | Pass | none | Staff can accept or decline the swap room request but the room numbers of the students are not visible. |  | No accept or reject button in front of request, creates confusion  [Use case 4] |
| 6 | 3/4/2016 | shubham | Pass | none | Staff can deallocate the room for whole batch |  | No Are You sure dialog forces the staff to be at high caution  [Use case 5] |
| 7 | 3/4/2016 | shubham | Fail | Medium | 1. There is no single option of view or modify details.  2. Only 15 students or rooms are shown even when there are more and there is no option of next page. |  | [Use case 6] |
| 8 | 3/4/2016 | shubham | Pass | None | User can view all the details of the room. |  | Though there is no mention of the floor of the room which was mentioned in SRS.  [Use case 7] |
| 9 | 3/4/2016 | amit | Pass | None | User can send a change room request |  | 1. Though the page name is set as swap room request only which creates confusion.  2. The user should not be allowed to send multiple room change requests.  [Use case 8] |

## User Acceptance Testing

The table below summarizes the test cases employed for user acceptance testing and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? | Comments |
| 1 | 3/4/2016 | Aman | Fail | High | Buttons are not hyperlinked properly |  | When we click the button and not the text, nothing happens which creates confusion that button is inactive. |
| 2 | 3/4/2016 | Aakash | Fail | Low | When we hover on logout button it’s color changes to that of its surrounding. Instead it should be the opposite. |  |  |
| 3 | 3/4/2016 | Amit | Fail | Low | There is a useless “Student” button, which appears when a student clicks on his username |  |  |
| 4 | 3/4/2016 | Shubham | Fail | High | When we click on change room and the room is not vacant (in staff mode), the user name appearing on the top - right corner changes. |  |  |
| 5 | 3/4/2016 | Aman | Fail | Medium | The capacity of the rooms has not been mentioned anywhere. |  |  |

## Regression Testing

The table below summarizes the test cases employed for regression testing and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? | Comments |
|  |  |  |  |  |  |  |  |

## Performance Testing

The table below summarizes the test cases employed for performance testing and the test results obtained for each test case:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Test Case ID | Date Tested | Tester | Pass/Fail | Severity of Defect [Low/ Medium/ High] | Summary of Defect | Closed prior to Production Release? | Comments |
| 1 | 3/4/16 | Amit | Pass | None | There There are no security or speed Issues |  |  |
| 2 | 3/4/16 | Aakash | Fail | Medium | Rooms are vacant still are not shown in the list thus making the software unreliable. |  |  |

# 5.0 Variances

The test cases were manually created as compared to the scripts and software being used in the industries but were sufficient enough to test all the boundary cases as well as major drawbacks in the system. The testing was planned, divided and executed in phases by all the team members involved. The system was also feebly tested with one large test file and the system stuck at many points.

# 6.0 Test Instances

1. The system gave server errors if a cycle of swap requests were given as inputs.
2. We could not identify which room to allot as the vacancy list did not show the complete list.
3. There occurred numerous empty fields in table columns during the testing phase which could have been avoided.

## Resolved Test Incidents

## Unresolved Test Incidents

# 7.0 Recommendations

1. User Interface is not at all user friendly and creates a lot of confusion many times.

Backend functionality needs to be rechecked.

**APPENDIX A: REFERENCES**

The following table summarizes the documents referenced in this document.

|  |  |  |
| --- | --- | --- |
| **Document Name** | **Version** | **Description** |
| *SRS Document* | *2.1* | *It is a description of a* [*software system*](https://en.wikipedia.org/wiki/Software_system) *to be developed. It lays out* [*functional*](https://en.wikipedia.org/wiki/Functional_requirement) *and non-functional requirements, and may include a set of use cases that describe user interactions that the software must provide.* |

**APPENDIX B: KEY TERMS**

The following table provides definitions for terms relevant to this document.

|  |  |
| --- | --- |
| **Term** | **Definition** |
| *Hyperlink* | A **hyperlink** is a reference to data that the reader can directly follow either by clicking or by hovering |
| *User* | College Student |
| *Staff* | College Administration |