

TEST PLAN DOCUMENT

for

Hall Management Software

Prepared by:

Ojas Dubey	22CS30039
Shreya Bose	22CS30050
Nived Roshan Shah	22CS10049

April 9th, 2024

1. Test Plan Identifier

Master Test Plan for Hall Management Software – MTP_HMS_SEL2024

2. References

Software Requirements Specification

3. Introduction

The purpose of this test plan is to provide a framework for verifying the reliability and testing the functionalities of the Hall Management Software, that is to be used by various HMC personnel and the vast student community. Such framework is necessary so that a systematic approach is taken to resolve reported errors (in case they exist), and also to ensure that such errors are not present in the deployed version.

To get a clearer understanding of various use-cases, interactions and functionalities, the Software Requirement Specification document can be referred.

4. Test Items

1. Authentication System and User Profile Management
2. Financial Management
3. Complaints Management
4. Hall Employees Management
5. API Testing (Stripe - Payment Processing Software)
6. Student Room Allocation
7. Hierarchical Responsibility Management and Access Privilege Management
8. Organized display of Entities

5. Software Risk Issues

1. Weaknesses in access controls may lead to unauthorized users gaining entry to restricted features or sensitive data, risking manipulation or breaches. Robust authentication mechanisms and regular security audits are imperative to mitigate such risks and uphold data integrity and privacy.
2. Data integrity risks arise from concurrent access or improper transactions, potentially leading to corruption or erroneous data exchanges within SQLite databases. Stringent transaction management protocols and error detection mechanisms are essential to prevent data loss or corruption, ensuring accurate and reliable data storage and retrieval.
3. Integration with external Payment Portal APIs introduces risks related to transaction success and failure handling. Inadequate error handling mechanisms or communication failures may disrupt payment processes, necessitating robust error detection and recovery strategies to ensure seamless transactions and user satisfaction.
4. Effective error handling and logging mechanisms are vital for facilitating easy debugging and troubleshooting within the software. Inadequate error logging or failure to capture crucial details may impede diagnosis and resolution of issues, compromising system stability and user experience. Implementing comprehensive error handling protocols enhances software reliability and maintainability.

6. Features to be tested

1. Authentication System and User Profile Management
 - a. Register new user
 - b. Login
 - c. Forgot Password
 - d. View Profile
 - e. Change Password
 - f. Update Profile
 - g. Delete Profile
2. Financial Management
 - a. Student Fees and Dues
 - b. Grant allocations for halls
 - c. Budget allocations for hall and mess requirements
 - d. Mess Expense Management
 - e. Record Management through passbooks (downloadable PDFs)
 - f. Employee Salary Management
3. Complaints Management
 - a. Students – Log a complaint
 - b. View complaints with ATR status (Pending or PDF Report) – Can be viewed by Student, Hall Manager and Warden
 - c. Issue ATR (by Hall Manager)
4. Hall Employees Management
 - a. Register Employee (such as Gardener, Sweeper, Cook, etc.)
 - b. Update Employee Profile
 - c. Approve Employee Leaves
 - d. Remove Employee
5. Stripe API Testing
 - a. Payment of Dues of Student through the use of a Dummy Card
6. Student Room Allocation
 - a. Room Preferences asked: If available then allotted, else next free room
 - b. Automatic Room Code generation based on type of room and number of blocks/floors
7. Hierarchical Responsibility Management and Access Privilege Management
As per the Class Functionality Diagram in the Software Requirement Specification Document*
8. Organized Display of (on interface) -
 - a. User Profiles
 - b. Hall Notices
 - c. Student Complaints and ATR Pdfs
 - d. Monthly Mess Menu
 - e. Occupancy Details
 - f. Employee Details
 - g. Student Accommodation Details
 - h. Passbooks (Hall, Mess, Warden) and their Pdfs

* Please refer to the SRS Document for more clarity

7. Features not to be tested

There are no features that are left to be tested. This version of the software serves a platform that touches upon 3 important aspects:

1. Complaints management
2. Digital accommodation records and handling basic book-keeping activities
3. Financial Management

Thus, until the presentation, no other extension to the software is planned and hence no features remain untested.

8. Approach

The testing levels will include the Unit testing, Integration Testing, System Testing and Acceptance Testing

1. Unit Testing

Developers will write unit tests for each component or module of the Hall Management Software using testing framework of Django. Test cases will be designed to cover different scenarios and edge cases, ensuring comprehensive test coverage.

2. Integration Testing

Integration testing will be conducted after unit testing to validate the integration of various components of the software. Test cases will be designed to simulate interactions between modules, including input/output data verification, API testing (stripe), and database integration testing. Both manual and automated testing techniques will be employed to verify integration points and ensure seamless communication between modules.

3. System Testing

System testing will be conducted to validate the overall functionality and behaviour of the hall management software. Test cases will be designed to cover various usage scenarios, including user authentication, complaints management, fees and dues processing, employee management, and student allocation.

4. Acceptance Testing

Stakeholders will participate in user acceptance testing to evaluate the hall management software against predefined acceptance criteria. Test scenarios will be designed based on real-world usage scenarios and user feedback. Stakeholders will interact with the software to perform common tasks, such as registering new users, creating complaints or managing finances. Feedback and issues identified during this testing will be documented and addressed before the software is deployed into production.

9. Items Pass/Fail Criteria

There are certain pass criteria for each of the testing phases and it is important that such criteria are met otherwise the failure of system corresponds to a failure in providing expected requirements to the end-users. Occurrence of a failure would also mean more resource consumption (in terms of tester and developer manpower or developing systems' usage) to further investigate and resolve the failing feature.

The Pass Criteria for various stages are as follows:

Unit Testing: No assertion failures should occur or exceptions should be thrown.

Integration Testing: Components should interact correctly and data exchange between modules occurs as expected.

System Testing: System tests should validate the software's functionality, performance, security, and usability against predefined acceptance criteria.

Acceptance Testing: Software should meet the needs and expectations of end-users.

Some Criteria for other independent testing units are:

Security Testing: Security tests should verify that the software implements appropriate access controls, password hashing, and protection mechanisms to safeguard sensitive information.

10. Suspension Criteria and Resumption Requirements

The tests associated with a subsystem should be suspended in only one case, i.e. when the main web page associated with the specific subsystem does not load in a client. This means other test cases related to that subsystem cannot even proceed. This is the only case where suspension should occur. The tests need not be suspended otherwise, since the isolation of subsystems is a core principle. Test failures in specific subsystems can be used to fix errors pertaining to only those subsystems, thus guaranteeing that this failure is completely contained within a specific unit. This also tells us how various components interact with each other, and suspending the tests means we won't be able to gather sufficient information. Resumption occurs once we fix the errors associated with the test failures that prevent the main pages from loading.

11. Test Deliverables

1. Test Plan
2. Testing Matrix
3. Coverage Report

12. Schedule

Developers perform the unit and integration testing during the development process – Dynamic

User Acceptance Testing (evaluation of project progress by Software Lab Teacher Assistant) – 2nd April, 2024

Multiple System Testing before project presentation (to identify improvement areas) – Dynamic

Submission of Test Plan – 9th April, 2024