Hotel Management System

QA Plan

Submitted by: Sidra Khalid

Table of Contents

1. Introduction

1.1: Test Plan Objectives

Hotel Management developed with the aim of management as well as the reserving facilities for users. This system provides various sorts of facilities to user as well as the hotel managers. The different objectives for testing this automated system are:

- ➢ User can make advance booking using this system.
- ➢ It saves user time in search of rooms.
- ➢ The system is useful as it calculates an exact cost for requested number of days.
- ➢ It saves organization resources and expenses.
- ➢ This system is effective and saves time and cost of users.
- ➢ The system is portable i.e. cab ne used from anywhere.
- ➢ Easy registration.

2: Scope

The scope of testing will be functional testing and black box testing. Black box testing is conducted by creating the necessary test scenarios and cases. After the execution, we identified a list of issues, which are as follows:

From functional to usability flaws, the software contains major issues related to design, function, and accessibility. Some of the design issues include splitting the large name into two parts, cosmetic and table alignment issues. Consequently, the software also contains major usability issues. It doesn't display any message after changing the password or no window is popped up when the user enters an invalid user name or password. So, it has real problems with password verification and changing password test cases.

Functional issues include errors in the booking, billing, and room category issues. Agent permission and user restriction parameters are not defined clearly and there exist some critical errors. Agents can amend and calculate the price according to their own will and there is no check installed to bar agents from doing it. Similarly, the user can also book the blocked and the reserved rooms. Such kinds of issues are also present in the room

categories, not updating the management about the actual status of the rooms. Furthermore, overbooking was another major hitch. For example, if 10 rooms are available in a VIP room category, users can book more than 10 rooms, which are not available in a particular category.   So, most of the test cases and scenarios are negative and my team was startled by such apathy from the side of developers.

3: Testing Strategy

The Testing strategy consists of series of different tests which will be performed according to the agreed methodology to verify the desired results and efficiency of the CR under observation. The primary purpose of these tests is to uncover the application limitations and measure its full capabilities. Scope and strategy is limited to the Change Requests under testing in Release.

The software was tested on the Windows 10 platform using the latest versions of Firefox and Google chrome. Everything went well as we performed cross-platform testing and the response was stable on both browsers.

3.1 Functional Testing

This is the type of black box testing which is included in the quality assurance processes to assess the functional performance of the system against the specifications and the requirements. Functional testing is primarily conducted by feeding the requirements into the testing module, running it on the existing system, and then examining the output against the desired results. If the outcome has a lot of deviations and variations from the expectations, from a developmental perspective, then it is cataloged under the defective components. Functionality testing scope is limited to the specific module under testing including the basic GUI testing. The risk areas will be identified and communicated to the concerned as per the functional understanding communicated to the tester.

All the defects identified in this phase will be added in SIT report will be generated and forwarded to the Release Manager.

4: Environment Requirements

Testing environment is in Java and software is made in Netbeans IDE. NetBeans provides a user friendly environment for project development.

5: Testing Schedule

Activity	Timelines
KT Sessions	13/3/2022 -15/3/2022(9am - 3pm)
Test Cases Development	16/3/2022 (9am - 5:30pm)
Test Cases review by Development Team	17/3/2022 (10am - 12pm)
Functional Testing	17/3/2022 (9am - 3pm)
Deployment	18/3/2022 (9am – 1pm)

6: Control Procedures

6.1: Reviews

The Quality Assurance team will conduct reviews on different phases of the release development i.e. Solution Draft Review, Summary Review). A meeting notice, with related documents, will be emailed to each participant.

6.2: Bug Review meetings

Regular weekly meeting/ on demand meetings will be held to discuss reported defects. The development department will provide status/updates on all defects reported and the testing department will provide addition defect information if needed. All concerned member of the project will participate.

In case proper regular sessions are not conducted, correspondence through E-mails will be maintained by both the teams.

6.3: Change Request

If there will be any change in the Signed-Off requirements, then CHANGE MANAGER / RELEASE MANAGER will be responsible to formally inform QA about the changes so that QA will update the test cases. There would be some changes in assumptions and risk involved after these changes, if needed. Once testing begins, changes to the application are

discouraged and if the changes are immediate then they should be intimated through email by the CHANGE MANAGER OR RELEASE MANAGER. Impact of the change will be analyzed in terms of effect on schedule and updating the documents.

QA will not be responsible for the changes made in functionalities and their impact without testing so this will be put of scope.

6.4: Defect Reporting

When defects are identified, the Testers will compile a defect report on the defect tracking system. The defect tracking systems will be accessible by the testers, developers & all concerned members of the project team.

When a defect will be found, it will be logged in emails or excel sheets with "**NEW**" status. The relevant developer will resolve this issue and will change its status to "**FIXED**". QA team will verify it and will close on complete resolution. If there will be any conflict on any issue, final approval will be made by the consensus of test lead and RM and Dev Lead. For more details, go through the Bug Life Cycle Document.

7: Functions to Be Tested

❖ **Deciding on the requirements and design of the system:**

In the early phases of the development process, the team decided it was essential to have a graphic user interface (GUI) for the software. After the sprint involving creating a GUI for the system, it was tested to check if all the tabs were working using a cout test in C++. Display tabs were set to non-editable tabs just to display the information.

❖ **Checking for the functionality of XML to get room data and check if the form filters it:**

After inputting constant data for the rooms like cost and availability were called in the system using XML, they were tested in the GUI to display rooms by their type displaying all the room numbers for that type and their availability status. By selecting the type of room on the combo box in the GUI and clicking filter, the system displayed all the available rooms of that type.

❖ **Creating a new customer record and storing it using XML:**

After adding parts in the GUI for adding a new customer record by inputting customer details and generating an ID for the customer. The customer record was then stored in the system data using XML and checked if a new current customer was created in the customers sections of the GUI for that customer which is then used to pick the customer and make the booking for the previously selected room.

❖ **Storing rooms and then using insertion sort to sort them by price:**

The rooms can be stored in an array because their room number and details never change because they are already built in the hotel, which makes it easier to sort the rooms by their price from low to high or vice-versa.

❖ **Calculating final bill and customer check-out:**

The final bill is calculated by multiplying the room's price per day of stay by the amount of days the guest stayed. After the member of staff receives the payment from the customer, they click on the check-out button which should set the room status back to available and set the guest Id and number of guests back to zero.

8: Functions Not To Be Tested

- ❖ Scope of Testing and quality assurance starts from the solution draft provided by development team.
- ❖ Solution draft is the base line and requirement substitution from the business in terms of Quality assurance and testing.
- ❖ The manual functional testing will only monitor the inputs and outputs of the functionality and basic level of GUI validations.
- ❖ The impact of some functionality enhancements will only be intimated to the development team in case of QA knowledge base on the same domain.

9: Resources and Responsibilities

QA manager will be responsible for the overall QA & testing activities. These activities will cover the relevant release / change requests at management level coordination as well. Test Lead will be responsible for all the activities relevant to on-

going release. Test Lead will responsible for selection of CR's, collection of SD, preparation of Release Test Plan/Schedules and coordination with Release Manager. Test Lead will track the schedule of release and inform the status.

9.1: Roles

Below mentioned roles will play a major role in execution of the each release.

- ❖ Quality Assurance Manager
- ❖ Release Manager
- ❖ Test Lead
- ❖ Developer
- ❖ Tester
- ❖ Change Manager

9.2: Responsibilities

Quality Assurance Manager	QA Manager will be responsible for overall QA & Testing activities performed within department. He will also involve in coordination and communication with higher management and other stakeholders as well.
Release Manager	Release Manager will be responsible for overall release development activities. He will have authority for go ahead on specific functionality. If there will be any issue then decisions will be made with consensus of release manager.
Test Lead	Ensures the overall success of the test cycles. Test lead will be involved in coordination with development team, business and other relevant stakeholders. He will be responsible for preparation of test plan, testing schedule, SIT Report on a specific

project. He will also be responsible for managing all the QA deliverables and testing tools. He will also coordinate with development team to install testing environment for each release for both types of testing i.e. Functional Testing.

Developer

Developer will be involved in generating the solution draft, reviewing the test cases and use cases developed by testing team and will Sign-off these documents for relevant CR. Developer will also be responsible for defect resolution when any issue will be reported.

Tester

A tester will develop Test cases for the assigned CR and forward these artifacts for review and Sign-off from the relevant developer. Tester will execute these test cases after development of the CR and issues/defects will be logged. Tester will also verify the issues resolved by developer. Tester will also responsible for uploading testing artifacts on share point repository.

Change Manager

A Change manager will be responsible for conducting UAT, making UAT report, he might need help from the QA to conduct UAT. Change manager is responsible for change. Will make relevant documentation against change management process.

10: QA Deliverables:

SD /CR Review Documents	QA Team
Test Case Development	Testing Team
Test Case Review	Test Lead(Internal)/ Development Team
Sign-off on Test Cases	Development Team
Functional Testing (SIT)	Testing team
Test Cycle Reports (Defects)	Tester / Test Lead
Complete Release Defects/Closure Report	Tester / Test Lead
Audit Report	CMMI MR/Process Owner

These deadlines are internally managed by QA and will not harm/affect the release schedule but we need to maintain flexibility during test cases execution in UAT week. Load-Performance testing may be continued after UAT week, if required so that performance analysis reports may be prepared. QA will start functional testing as soon as developed functionalities will be handed over to the relevant tester.

11: Suspension / Exit Criteria

If an issue will impact the whole testing effort and without its resolution the testing team cannot go for further testing then QA Manager may suspend the testing activity. Criteria that will justify test suspension are:

- ❖ Hardware/software is not available at the times indicated in the project schedule.
- ❖ Unavailability of testing environment.
- ❖ The Development team will provide its full support in case of unavailability of hardware and database but does not own the ownership for it. If such an issue arise during testing, coordination will be conducted with the specific teams (Unix Ops or DB Ops).

- ❖ Unavailability of information required for testing regarding the functionality under test. DEV team is completely responsible for Test data preparation, availability and accuracy etc.
- ❖ Source code contains one or more critical defects, which seriously prevents or limits testing progress and system is crashing on each run and bogging down.
- ❖ Assigned testing/development resources are not available when needed by the test team.
- ❖ In case of unavailability of the developer, the person responsible in the absence of the developer will look into the issues and their relevant fixes.

12: Resumption Criteria

If testing is suspended, resumption will only occur when the problem(s) that caused the suspension will be resolved. When a critical defect is the cause of the suspension, the "FIX" must be verified by the test department before testing is resumed.

This criterion only applies to the testing effort made by QA team so if the issue(s) will not be resolved within deadlines then the testing of those functionalities will be out of scope. So QA will not continue the functional testing when the functionalities will go live in production.

However if the End User gives the go-ahead to the functionality despite any resumption or suspension from the QA team, the functionality will be made Live on the set release date. The feedback from the QA team will/can be entertained in the later stages.

13: Dependencies

13.1: Personnel Dependencies

The testing team will need the application developers and these resources are available.
In case of unavailability of the developer, the person responsible in the absence of the developer from the Development team will be look into the issues and their relevant fixes.

13.2: Software Dependencies

For testing this hotel management, there are some software requirements which are necessary for testing. These requirements are given below:

- ❖ Software is designed to run on any platform above Microsoft Windows 7 (32bit).

❖ Microsoft .NET Frameworks 4.0 or above.

❖ Microsoft SQL Server Management Studio Express 2010.

Testing of HMS is dependent on these requirements. The source code must be unit tested and provided within the scheduled time outlined in the Project Schedule. The software to be used in testing should also be installed on the test machines. Testers should have required privileges and prerequisite to perform the testing. Availability of test environment is the responsibility of RM and Tieto Team and their nominated persons.

13.3: Hardware Dependencies

Testing of HMS is dependent on these hardware requirements.

❖ **Operating System** Supports all known operating systems, such as Windows, Linux.

❖ **Computer** 512MB+ RAM, monitor with minimum resolution of 1024x768, keyboard, and mouse.

❖ **Hard Drive** should be in NTFS file-system formatted with minimum 10 GB of free space.

❖ **A Laser printer** will need to be used to print these reports and notes.

The Test Machines/servers (with specified hardware/software) as well as the LAN environment need to be available during normal working hours. Any downtime will affect the test schedule. The availability of testing environment must be intimated by the release manager before testing start.

13.4; Test Data & Database

Test data (different users Login and password) & database will be provided by the RM and dev team is completely responsible for Test data preparation, availability and accuracy etc. Testing data & database environment will remain integrated and will not change until the testing gets complete if Testing data & database environment is changed during load testing in case of any exception then Test lead will be informed about any such scenario at possible earliest. For load testing, the environment availability will be communicated by the RM or Development lead.

QA will only be responsible for the changes which are informed formally and changed functionalities are retested and beyond this QA will not be responsible and it will be out of

scope. The QA team will be informed in advance about the possible changes, updates and compiles.

14: Risks

14.1: Schedule

The schedule should be carefully planned because if the schedule for each phase is very tight then it could affect testing effort. A deviation from schedule may impact the whole testing life cycle. As in QA, a release or change has very strict schedule and it would be difficult to cater any kind of deviation in any phase, so schedule is the great risk. Release calendar/Project plan should be updated accordingly to allocate the time for the overall testing activities and must be tracked.

14.2: Technical

In case of faulty build the application may crash Web server, Application server etc. RM and Tieto team would be responsible to provide the additional support while performing this activity.

There must be a backup plan if any crash happens. The environment should be available in between or after the daily working hours as well, any activity performed with in the testing cycle should be intimated by the Release Manager.

14.3: Resources

All kind of hardware and software resources must be available and if there would be unavailability of human resources then there is a risk that testing effort may be delayed or stopped. In case absence of resources, the person responsible will look after the relevant issues pointed by QA.

14.4: Requirements Change

The test plan and test schedule will cover only the Signed-Off requirements in the solution draft. Any major changes to the requirements may impact the testing schedule and will need to be approved by the management and Project team.

Impact evaluation of the change depending on the effort required will be evaluated by testing and development team.

****