Atlantica Reservation & Management Tool

Test Plan

Table of Contents

[1 Introduction 1](#_Toc505936366)

[1.1 Purpose 1](#_Toc505936367)

[1.2 Assumptions 1](#_Toc505936368)

[2 Version Control 2](#_Toc505936369)

[3 Testing Strategy 3](#_Toc505936370)

[3.1 Requirements Validation 3](#_Toc505936371)

[3.2 Testing Coordination 3](#_Toc505936372)

[3.3 Unit Testing 3](#_Toc505936373)

[3.4 System Integration Testing 4](#_Toc505936374)

[3.5 Security Testing 4](#_Toc505936375)

[3.6 Performance and Stress Testing 5](#_Toc505936376)

[3.7 User Acceptance Testing 5](#_Toc505936377)

[3.8 Recovery Testing 6](#_Toc505936378)

[3.9 Production Verification 6](#_Toc505936379)

[4 Resources/Roles & Responsibilities 7](#_Toc505936380)

[5 Requirement Guide 8](#_Toc505936382)

[6 Test Schedule 9](#_Toc505936383)

[7 Test Scripts and Testing Tools 10](#_Toc505936384)

[8 Control Procedures 12](#_Toc505936385)

[8.1 Defect Reporting 12](#_Toc505936386)

[8.2 Defect Reviews 12](#_Toc505936387)

[8.3 Change Requests 12](#_Toc505936388)

[8.4 Resumption Criteria 12](#_Toc505936389)

[9 Approvals 13](#_Toc505936390)

[9.1 Plan Approval 13](#_Toc505936391)

# Introduction

## Purpose

User testing is made an integral phase of the software lifecycle in order to best align the final product with the target audience and achieve a high degree of user satisfaction. It aids in identifying bugs and glitches in the products that can be rectified before the deployment of the product along with providing valuable user insight into the usability of the product. In the long run, user testing will allow for increase in user retention, reduction in development maintenance and customer support costs and increased profits for the business.

## Assumptions

This document assumes the ARMT system has been created in the manner the hotel owner wanted it to be. Furthermore, the documents assumes that all the criteria have been met that was specified in the Software Requirement Specification Document, and completed according to the points in the High Level Design Document and Detailed Design Document.

# Version Control

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Person | Change |
| 1.0 | 11/1/2018 | Ashwin Sivaraman | Initial Document Creation |
| 1.1 | 11/5/2018 | Ashwin Sivaraman | Code refactor to support older internet browsers |
| 1.2 | 11/7/2018 | Ashwin Sivaraman | Error handling for users who make a booking prior to signing in is added |
| 1.3 | 11/9/2018 | Ashwin Sivaraman | Date format is highlighted in all checkboxes to prevent ambiguity |
| 1.4 | 11/11/2018 | Ashwin Sivaraman | Error handling for invalid username/password was added to website functionality |
| 1.5 | 11/13/2018 | Ashwin Sivaraman | Graphic interactive calendar interface was added for date selection |

# Testing Strategy

The tests performed and their expected results are outlined below:

1. Test: Open hotel website home page

Expected Result: Homepage loads

1. Test: User clicks on ‘Location’ dropdown menu and selects from one of the hotel locations

Expected Result: User is redirected to selected location's webpage

1. Test: User clicks on ‘Make a Booking’

Expected Result: User is redirected to payment page

1. Test: User selects preferred method of payment: credit/debit card or PayPal

Expected Result: User is redirected to the appropriate payment portal’s webpage

1. Test: If selected credit/debit card then enter the credit/debit card number, expiry date and security code respectively

Expected Result: Payment is processed

1. Test: If selected PayPal then login using email ID and password setup with PayPal account

Expected Result: User logs in

1. Test: Click send money on PayPal dashboard

Expected Result: Redirected to next page

1. Test: Enter recipient email address and payment amount

Expected Result: Transaction is completed and review transaction page is opened in the new tab. Confirmation email is sent to the user’s email address.

1. Test: User clicks on ‘Print’ button in order to print payment confirmation

Expected Result: PDF version of payment summary is printed to the user

1. Test: User clicks on transaction history button

Expected Result: User is redirected to the transaction history page

1. Test: User enters date of transaction to be displayed

Expected Result: Transaction history for date entered is opened

1. Test: User scrolls through transactions and clicks on dropdown for order summary of a particular transaction

Expected Result: Order summary appears in dropdown box.

## Requirements Validation

 **Validity**: The functions proposed by the hotel owner during the requirements face of the waterfall software development life cycle are aligned with the features that are being developed for the website.

 **Consistency**: Requirements in the document do not conflict with the component descriptions of the websites that are being developed.

 **Completeness**: The document contains all the requirements and constrains.

 **Realism**: The requirements specified can be implemented using the current knowledge of existing technology, the budget allocated for the development of the website and schedule.

**Verifiability**: Requirements are written in a manner that can be tested and several tests were undertaken to demonstrate that the system meets the specified requirements.

## Unit Testing

The different functionalities of the website were broken down into different units which were then tested individually. This allows for code that is easy to maintain and reliable, encourages code reusability, allows for faster development and easy debugging. The most important user stories in the Waterfall software development life cycle were chosen for performing test cases on. This was done in order to focus on the tests that will most affect the behaviour of the system.

## System Integration Testing

System integration testing is performed in an integrated hardware and software environment in order to verify the behaviour of the complete system. Several tests where done to check the reachability of the Payment REST API, the performance of the website on different web browsers and testing database connections. Changes were made to the code in order to better support older versions of web browsers such as Internet Explorer 6, 7 and 8.

## Security Testing

Security testing is performed in order to detect vulnerabilities in the system and determine whether or not its data and resources are protected from hackers and potential threats. User testing was performed wherein error handling for incorrect login details was performed. Vulnerabilities in network infrastructure, making sure client side browser is not being manipulated and server side application security were contained in the security testing of the application.

# Resources/Roles & Responsibilities

The following are the different roles and responsibilities of the individuals involved in the project:

* Project Manager: Bharat Bhushan Verma
* Scrum Master: Mohd Nawaz Hussain
* Lead Designer-UI: Kothai Kannappan Murugappan
* Lead Designer-Architecture: Sarbottam Thapa Magar
* Quality Assurance Specialist-Security: Ashwin Sivaraman

# Requirement Guide

* Entry Requirement
* Test plan approved
* Test environment stable and ready
* Test cases documented and distributed
* Test tools ready
* Test resources available
* Exit Requirement
* Test case completion with QA Committee approval
* Identification of open defects with quality impact analysis
* Test objectives successfully met
* Suspension Requirement
* Unstable testing environment
* Unavailable resources
* Data corruption
* Resumption Requirement
* Verified stable test environment
* Resource availability meets testing demand
* Data integrity verified with Hotel Atlantica

# Test Cases and Testing Tools

|  |  |
| --- | --- |
| Test Case Field | Description |
| Group (Owner) | Atlantica Hotel |
| Components | Discounts on bookings for members |
| Test ID | TC10 |
| Test Description | Testing the discounts application for the members. |
| Prepared By | Ashwin Sivaraman |
| Date Prepared | 11/12/2018 |
| Test Importance | P1 (Critical) |
| Requirements Traceability | ID FR1 |
| Steps To Perform | |  | | --- | | * Open hotel website home page | | * User clicks on 'Location' dropdown menu and selects from one of the hotel locations | | * User clicks on 'Location' dropdown menu and selects from one of the hotel locations | | * User clicks on 'Make a Booking' | | * User signs in using valid username and password | | * User selects his preferred method of payment : Credit/debit card or PayPal | | * If selected credit/debit then enter credit/debit card number, expiry date and security code respectively | | * If selected credit/debit then enter credit/debit card number, expiry date and security code respectively | | * If selected credit/debit then enter credit/debit card number, expiry date and security code respectively | | * Payment Success webpage is loaded indicating that confirmation email was sent to user's email address | | * If selected PayPal then login using email ID, password setup with PayPal account | | * If selected PayPal then login using email ID, password setup with PayPal account | | * Click 'Send Money' | | * Enter recipient email ID and payment amount | | * User clicks on 'Print' icon to print payment confirmation | | * User opens website homepage | | * User clicks on transaction history button | | * User enters date of transaction to be displayed | | * User enters date of transaction to be displayed | | * User scrolls through transactions and clicks on dropdown for order summary of a particular transaction | |
| Expected Result | User is able to review his transaction history. |
| Date Executed | 11/14/2018 |
| Executed By | Ashwin Sivaraman |
| Actual Results | As Expected |
| Test Result | Pass |
|  |