<ZJU Pay>

SQA Test Plan (STP)

Version: < 1.4 >

< 4/12/2023 >

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Prepared / Revised By** |
| 11/10/2023 | 1.0 | First version of the SQA. | Peilin Sun, Yue Zhang, Pu Chen, Junsheng Huang, Shenan Zhang, Wenjie Huang, Rongchen Wang |
| 11/12/2023 | 1.1 | Review and modification. | Peilin Sun |
| 11/14/2023 | 1.2 | Added a detailed description of the scope for smoke testing, functional testing, and regression testing. | Peilin Sun |
| 12/3/2023 | 1.3 | Remove some unnecessary content. | Peilin Sun |
| 12/4/2023 | 1.4 | Modified the process related to the test | Pu Chen |

Table of Contents

1. Introduction 4

1.1 Document Terminology and Acronyms 4

1.2 References 5

2. Target Test Items 6

2.1 Test Inclusions 6

2.1.1 Account management module 6

2.1.2 Online booking module 6

2.2 Test Exclusions 6

1.1.1 Online shopping module 6

1.1.2 Account reconciliations and audits module 6

3. Test Approach 8

3.1 Test Identification and Justification 8

3.2 Conducting Tests 8

3.2.1 Smoke testing 8

3.2.2 Functionality testing 9

3.2.3 Regression Testing 9

3.3 Defect Management 10

3.4 Test Metrics 10

3.5 Test Criteria 11

4. Environmental Needs 12

4.1 System Hardware 12

4.1.1 Server hardware environment 12

4.1.2 Client hardware environment 12

4.2 System Software 12

4.2.1 Server software environment 12

4.2.2 Client software environment 12

5. Responsibilities, Staffing, and Training Needs 13

5.1 People and Roles 13

5.2 Staffing and Training Needs 13

6. Key Project/ Phase Milestones 14

7. Risks, Dependencies, Assumptions, and Constraints 16

7.1 Risks 16

7.2 Dependencies 16

7.3 Assumptions 16

7.4 Constraints 16

SQA Test Plan

# Introduction

## Document Terminology and Acronyms

|  |  |
| --- | --- |
| **SQA** | Software Quality Assurance, a means and practice of monitoring all software engineering processes, methods, and work products to ensure compliance against defined standards. |
| **Smoke Testing** | Smoke Testing is preliminary testing or sanity testing to reveal simple failures severe enough to, for example, reject a prospective software release. |
| **Functional testing** | Functional testing is a type of testing that seeks to establish whether each application feature works as per the software requirements. |
| **Regression testing** | Regression testing is performed to find out whether the updates or changes had caused new defects in the existing functions. This step would ensure the unification of the software. |
| **Performance testing** | Performance testing is a non-functional software testing technique that determines how the stability, speed, scalability, and responsiveness of an application holds up under a given workload. |
| **Black-box Testing** | Black-box Testing is a method of software testing that examines the functionality of an application without peering into its internal structures or workings. |
| **White-box testing** | White-box testing is a method of software testing that tests internal structures or workings of an application, as opposed to its functionality (i.e. black-box testing). In white-box testing, an internal perspective of the system is used to design test cases. |
| **Code Coverage** | Code Coverage is a percentage measure of the degree to which the source code of a program is executed when a particular test suite is run. |

## References

[1] Pressman R S. Software engineering: a practitioner's approach[M]. Palgrave macmillan, 2005.

[2] GB/T 8567–2006, Specification for computer software documentation[S]. China, 2006.

[3] Software Engineering Course Design[M]. Xiaohu Yang, 2007.

# Target Test Items

## Test Inclusions

### Account management module

* Register
* Login
* View personal information
* Modify password
* Modify personal information
* View balance
* Top up
* View payment statistics

### Online booking module

* Search and filter hotel and flight
* Book Hotel and flight
* View discounted airfares and special room offers
* View booking history

## Test Exclusions

The following modules are under development and are not involved in testing.

### Online shopping module

* Search product
* Purchase product
* Submit complaints
* Refund

### Account reconciliations and audits module

* Audit transaction records
* Query transaction record
* Correct transaction record
* Compile the statuses of transaction records

# Test Approach

## Test Identification and Justification

**Smoke testing** includes the general scenarios of the important core functionalities of both the online booking module and the account management module, primarily including user login, registration, balance inquiry, recharge, hotel reservation, and flight booking.

**Functionality testing** primarily includes the online booking module.

**Regression testing** primarily includes the account management module.

## Conducting Tests

### Smoke testing

|  |  |
| --- | --- |
| **Definition** | Smoke testing, also called build verification testing or confidence testing, is a software testing method that is used to determine if a new software build is ready for the next testing phase. This testing method determines if the most crucial functions of a program work but does not delve into finer details. |
| **Test Scope** | The most crucial functions of a program. |
| **Test Objective** | The primary objective of smoke testing is to swiftly verify the fundamental functionalities and critical paths of the software system, determining its suitability for more comprehensive testing. |
| **Tool** | NA |
| **Test Method** | We'll set up a set of test cases to ensure the application programs stable and can be tested completely. If smoke testing fails, developers will fix bugs and release new versions. |

### Functionality testing

|  |  |
| --- | --- |
| **Definition** | Functional testing is a software testing approach that validates individual functions of a system to ensure they operate in accordance with specifications and expected behaviors. This testing covers various aspects of the system, including basic functionality, user interfaces, database interactions, security, error handling, among others. |
| **Test Scope** | Various functional modules of the system. |
| **Test Objective** | The primary goal of functional testing is to confirm that each function of the system operates as per specifications and expected behavior. It aims to discover functional defects or errors and validate whether the system meets user requirements. Through functional testing, it ensures the system performs various operations reliably and provides expected outcomes. |
| **Tool** | NA |
| **Test Method** | We list all the important functions of the software system, and design the data and operation set of the input and output in the test according to the test case specification. Then we will execute these test cases and compare the actual output with our Expected to be consistent, developers will fix bugs and release new versions. |

### Regression Testing

|  |  |
| --- | --- |
| **Definition** | Regression testing is a software testing process conducted to ensure that recent code changes or modifications in a software application haven't adversely affected the existing functionalities. It aims to identify any new bugs or unintended alterations in the software's behavior caused by modifications. |
| **Test Scope** | The previously developed and tested software functions. |
| **Test Objective** | The primary goal of regression testing is to verify that the previously developed and tested software functions correctly even after making changes. It ensures that new updates, enhancements, or bug fixes haven't introduced unexpected issues or caused regressions in the existing functionalities. |
| **Tool** | Subject7 |
| **Test Method** | After bugs are identified and then fixed, testers will perform regression testing to ensure our changes do not affect the rest of the system. |

## Defect Management

We will use Bugzilla to manage all discovered defects and use it to log newly discovered defects. At the same time, using bugzilla can help us know which bugs should be considered first.

At the same time, defect management helps further regression testing.

## Test Metrics

We'll select some basic test metrics to measure the development quality and test quality of our work. They're as follows:

* count of test cases
* count of passed test cases
* count of un-passed test cases
* count of bugs
* acceptability of bugs

Moreover, we'll make use of the data to calculate some derivate metrics:

* accept rate=count of passed test cases/count of test cases
* fail rate=1-accept rate
* validness=count of bugs found/count of bugs
* cover rate of test=count of test cases run/count of test cases
* cover rate of requirements=count of requirements covered/count of requirements
* cover rate of requirements(weighted)=sum of weight of requirements covered/count of requirements
* time difference=expected date-actual date
* bug rate of 1,000 lines of code=count of bugs/1000 lines of code

## Test Criteria

In order to ensure that the important functions of the product can be used and the realization of the functions is guaranteed, the following test passing standards are formulated. When all test cases with priority of Smoke and High can be passed, and the test cases with priority of Medium and Low have a pass rate of more than 95%, this test round will pass.

Therefore, the passing criteria for the test are:

1. pass rate of test cases with priority of Smoke and High：100%
2. pass rate of test cases with priority of Medium and Low：>=95%

# Environmental Needs

## System Hardware

### Server hardware environment

* CPU main frequency more than 2.0GHZ
* Memory more than 2GB
* Hard disk capacity greater than 128G, hard disk speed more than 5400 rpm
* Network card: Gigabit network card
* Network cable can be used normally and has good data transmission capability
* Power supply is sufficient

### Client hardware environment

* CPU main frequency more than 1.0GHZ
* Memory more than 1GB
* Hard disk capacity greater than 32G, hard disk speed more than 5400 rpm
* Network card: Gigabit network card
* Network cable can be used normally and has good data transmission capability
* Power supply is sufficient

## System Software

### Server software environment

Operating System：Ubuntu 22.04

Database：MySQL 8.0.34.

### Client software environment

Operating System：Windows 10 or later, macOS 10.15 or later

Browsers：All contemporary web browsers, including Edge, Firefox, Chrome, and Safari

# Responsibilities, Staffing, and Training Needs

## People and Roles

Test Manager: Peilin Sun

* Responsibilities include overall test planning, coordination, and ensuring that testing activities align with project goals and objectives. This role will also oversee test execution and defect management.

Test Analyst: Junsheng Huang

* Responsibilities include providing oversight to ensure that testing activities align with the project schedule and goals, as well as evaluation of the testing process.

Test Process Designer: Yue Zhang

* Responsibilities include providing domain-specific knowledge and expertise to assist in defining the testing methodology and associated processes.

Test Automation Architect: Pu Chen

* Responsibilities include designing the overall test automation strategy and framework, as well as identifying the right tools and technologies for test automation.

Test Engineers: All members

* Responsibilities include designing and executing test cases, maintaining test scripts and reporting defects. They will work closely with the Test Manager to ensure comprehensive test coverage

## Staffing and Training Needs

**Staffing**: Peilin Sun, Yue Zhang, Pu Chen, Junsheng Huang, Shenan Zhang, Wenjie Huang, Rongchen Wang

**Training**: Test Engineers will undergo training on testing tools and methodologies to ensure that they are well-equipped to execute their responsibilities effectively. This includes concepts such as unit testing, integration testing, acceptance testing, smoke testing and regression testing to ensure that they are able to conduct thorough and reliable testing.

# Key Project/ Phase Milestones

| **Milestone** | **Planned Start Date** | **Actual Start Date** | **Planned End Date** | **Actual End Date** |
| --- | --- | --- | --- | --- |
| Project/ Phase starts | 11/16 | 11/16 | 11/16 | 11/16 |
| SQA Test Plan agreed | 11/17 | 11/17 | 11/17 | 11/17 |
| Testing resources requisitioned | 11/18 | 11/18 | 11/18 | 11/18 |
| Testing team training complete | 11/19 | 11/19 | 11/19 | 11/19 |
| Requirements baselined | 11/20 | 11/20 | 11/20 | 11/20 |
| Test Case Design baselined | 11/21 | 11/21 | 11/21 | 11/21 |
| QA – Cycle 1 Build Acceptance Test Execution | 11/22 | 11/22 | 11/22 | 11/22 |
| QA – Cycle 1 Functional Test Execution | 11/23 | 11/23 | 11/23 | 11/23 |
| QA – Cycle 2 Build Acceptance Test Execution | 11/24 | 11/24 | 11/24 | 11/24 |
| QA – Cycle 2 Functional Test Execution | 11/25 | 11/25 | 11/25 | 11/25 |
| QA Regression Test Execution | 11/26 | 11/26 | 11/26 | 11/26 |
| QA Performance/Failover Test Execution | 11/27 | 11/27 | 11/27 | 11/27 |
| QA Final Integrated Build Test Execution | 11/28 | 11/28 | 11/28 | 11/28 |
| Project Status Assessment review | 11/29 | 11/29 | 11/29 | 11/29 |
| Project/ Phase ends | 11/30 | 11/30 | 11/30 | 11/30 |

# Risks, Dependencies, Assumptions, and Constraints

## Risks

* Technical Risks: There might be challenges in integrating the personal account management and hotel flight booking modules smoothly.
* Compatibility Issues: The system might face compatibility issues with various browsers and devices.
* Performance Issues: Heavy traffic could lead to server overload and affect the system's performance.

## Dependencies

* Database: The functionality of personal account management and booking modules depends heavily on a stable and well-maintained database.
* Internet Connectivity: The system is dependent on a stable internet connection for users to access and interact with it.

## Assumptions

* User Understanding: Users are assumed to have a basic understanding of online booking systems, and any necessary guidance will be provided through intuitive design.
* Stable Environment: It is assumed that the testing environment will remain stable and consistent throughout the testing phase.
* Accurate Data: The availability and accuracy of hotel and flight information from external sources are assumed to be reliable.

## Constraints

* Time Constraints: There might be a limited timeframe for testing due to project deadlines, potentially impacting thorough testing.
* Budget Constraints: The project might have budget limitations, affecting the tools and resources available for testing.
* Resource Constraints: Availability of testing resources, including hardware and software, might be limited.