## **1 Testing Planning and Control**

### 1.1 Test Planning

#### 1.1.1 Objective:

* Ensure quality and reliability of the registration system.
* Define the scope of testing, including login and payment functionalities.
* Allocate resources efficiently for testing activities.

#### 1.1.2 Components of a Test Plan

**1.Introduction**

* 1. Overview of the registration system project and testing goals.
  2. Identify stakeholders (e.g., developers, testers, project managers) and their roles.

**2.Scope of Testing**

**Features to be tested:**

* + User registration (form validation, email verification).
  + User login (authentication process).
  + Payment integration with Stripe (successful and failed transactions).

**Features not to be tested:**

* + Non-functional requirements (performance, stress testing).

**3.Test Objectives**

* Validate user registration and login functionalities.
* Ensure payment transactions are processed correctly through Stripe.
* Detect any defects in the user experience.

**4.Test Strategy**

* · Levels of testing: unit testing, integration testing, system testing.
* Testing types: manual testing for UI, automated testing for backend logic and payment processes.

**5.Test Deliverables**

* · Test cases/scripts for registration, login, and payment functionalities.
* Test data for different user scenarios (valid and invalid inputs).
* Test reports summarizing findings and defect logs.

6 .**Entry and Exit Criteria**

* **Entry Criteria:** Requirements are finalized; development is complete.
* **Exit Criteria:** All critical test cases pass; defects are resolved.

· 7.**Resources**

* Team members: testers, developers, project manager.
* Tools: Test management software, Stripe API for payment testing.

**8.Schedule**

* Timeline for each phase of testing (test planning, execution, reporting).

9 **Risk Management**

* Identify potential risks related to payment processing and user data security, along with mitigation strategies.

10 **Approval**

· Define who will approve the test plan and deliverables (e.g., project manager, QA lead).

#### 1.1.3 Tools for Test Planning

* Documentation tools (e.g., Confluence, Google Docs).
* Test management tools (e.g., Jira, TestRail).

### 1.2 Test Control

#### 1.2.1 Objectives:

* Ensure testing stays on track and adapts to changes.
* Provide visibility to stakeholders regarding testing progress.

#### 1.**2.2 Activities**

1. ****Monitoring****
   * Track test execution progress against the test plan.
   * Monitor defect metrics (e.g., number of defects found, severity).

2. **Reporting**

* Provide regular status reports (daily/weekly).
* Summarize key metrics (e.g., test coverage, pass/fail rates).

3. **Issue Management**

* Identify and resolve bottlenecks affecting testing progress.
* Reallocate resources as needed.

4.**Change Management**

* Adapt test plans to accommodate changes in requirements or schedules.

**5.Defect Tracking**

* Ensure timely resolution of reported defects related to registration and payment functionalities.

#### **1.2.3 Tools for Test Control**

* Test management software (e.g., Jira, TestRail).
* Defect tracking tools (e.g., Bugzilla, Mantis).

#### 1.2.4 Deliverables

* Test Plan Document: A comprehensive guide for the testing process.
* Test Metrics Reports: Quantitative measures of testing progress.
* Final Test Report: Summary of activities, outcomes, and recommendations.

## 2 Testing Analysis and Design

### 2.1 Test Analysis

#### 2.1.1 Objectives

* Identify what to test in the registration and payment processes.
* Ensure complete test coverage of functional requirements.

#### **2.1.2 Activities**

**1.Requirement Analysis**

* + Review business and technical requirements for registration and payment.
  + Identify any ambiguities or missing details.

1. **Test Basis Identification**

Use artifacts such as:

* + - Functional specifications.
    - Use cases for user registration and payment processing.

1. **Derive Test Conditions**

Identify high-level test scenarios (e.g., successful login, failed payment).

**4.Prioritization**

* + Rank test conditions based on risk and importance (e.g., payment processing is high priority).

**5.Entry Criteria**

Requirements are finalized, and the test environment is set up.

### **2.2 Test Design**

#### 2.2.1 Objectives

* Create detailed test cases for registration, login, and payment functions.
* Ensure test cases are detailed, reproducible, and traceable.

#### 2.2.2 Activities

**1.Test Case Design**

Write detailed test cases for:

* + - User registration (valid and invalid inputs).
    - User login (correct and incorrect credentials).
    - Payment processing (successful and failed transactions).

Use a clear template for each test case.

**2.Test Data Design**

* + Identify data inputs for different scenarios (valid user data, invalid credit card information).

**3.Test Environment Setup**

* + Define hardware, software, and network requirements for testing.

**4.Test Automation Design (if applicable)**

* + Identify test cases suitable for automation (e.g., payment process).
  + Select tools for automation (e.g., Selenium).

**5.Traceability Matrix**

* + Create a Requirements Traceability Matrix to ensure all requirements are covered.

**6.Entry Criteria**

* + Test conditions are approved, and design templates are ready.

#### **2.2.3 Deliverables**

**Test Scenarios**

* 1. High-level conditions derived from requirements.

**Test Cases**

* 1. Detailed steps for registration, login, and payment validation.

**Test Data**

* 1. Valid and invalid inputs for registration and payment.

**Requirements Traceability Matrix (RTM)**

* 1. Ensures all requirements are covered by test cases.

#### 2.2.4 Tools for Analysis and Design

* Test management software (e.g., TestRail).
* Automation tools (e.g., Selenium).
* Traceability tools (e.g., Excel or dedicated RTM tools).