Test Plan for Student Registration System with Payment Module

1. Introduction

The purpose of this test plan is to ensure the functionality, performance, reliability, and security of the Student Registration System with an integrated Payment Module. This system allows students to register for courses and make payments seamlessly. The testing process will identify any bugs, ensure compliance with requirements, and confirm the system's readiness for deployment.

2. Objectives

The key objectives of this test plan are:

1. To verify that the student registration process is working as intended.
2. To ensure that the payment module supports all specified payment methods and processes transactions accurately.
3. To validate the integration between the registration system and the payment module.
4. To test the system's performance under various conditions, including high user loads.
5. To ensure that the system meets security standards, particularly for handling sensitive payment information.
6. To confirm usability and accessibility for a diverse user base.

3. Scope

This test plan will cover the following areas:

1. **Student Registration System**:
    - User account creation and login functionality.
    - Course selection and registration workflows.
    - Validation of prerequisites for course registrations.
    - Notifications (email/SMS) sent upon successful registration.
2. **Payment Module**:
    - Integration with payment gateways.
    - Processing of payments via credit/debit cards, digital wallets, and other methods.
    - Transaction success and failure handling.
    - Refunds and cancellations.
3. **System Integration**:
    - Communication between the registration system and payment module.
    - Handling of errors during data exchange.
4. **Performance Testing**:
    - System behavior under peak usage.
    - Response times for registration and payment processes.
5. **Security Testing**:

    - Protection of sensitive information (e.g., passwords, payment details).
    - Compliance with PCI DSS standards.
    - Prevention of unauthorized access and data breaches.
6. **Usability Testing**:
    - User interface consistency and clarity.
    - Accessibility for users with disabilities.

4. Test Strategy

1. **Testing Levels:**
    - Unit Testing
    - Integration Testing
    - System Testing
    - User Acceptance Testing (UAT)
2. **Testing Types:**
    - Functional Testing
    - Performance Testing
    - Security Testing
    - Usability Testing
    - Compatibility Testing
3. **Testing Tools:**
    - Selenium (for automated UI testing)
    - JMeter (for performance testing)
    - OWASP ZAP (for security testing)
    - Postman (for API testing)
4. **Test Data:**
    - Dummy student accounts with varying roles (new users, returning users, etc.).
    - Test credit/debit card details and payment gateway sandbox accounts.
    - Edge cases, such as expired cards, insufficient funds, and duplicate registrations.

5. Test Environment

1. **Hardware Requirements:**
    - Server: Minimum 8-core CPU, 16GB RAM.
    - Client devices: Desktop, laptop, tablets, smartphones.
2. **Software Requirements:**
    - Operating Systems: Windows, macOS, Linux, Android, iOS.
    - Browsers: Chrome, Firefox, Safari, Edge.
3. **Network Requirements:**
    - Simulated varying network speeds (2G, 3G, 4G, Wi-Fi).

---

6. Test Cases

1. **Student Registration**
    - Verify user can create a new account.
    - Verify login functionality with valid and invalid credentials.
    - Verify the course catalog is displayed correctly.
    - Verify successful course registration and prerequisites validation.
2. **Payment Module**
    - Verify payment initiation for selected courses.
    - Validate processing of different payment methods (credit card, debit card, wallets).
    - Test handling of failed transactions.
    - Verify refund processes.
3. **Integration Tests**
    - Verify data consistency between registration and payment modules.
    - Test error handling during integration failures.
4. **Performance Tests**
    - Conduct stress testing with 10,000 concurrent users.
    - Measure response times for course registration and payment.
5. **Security Tests**
    - Test encryption of payment data.
    - Validate protection against SQL injection, XSS, and CSRF attacks.
    - Test session management (e.g., timeouts).
6. **Usability Tests**
    - Test clarity and consistency of the user interface.
    - Validate accessibility compliance (e.g., WCAG 2.1).

---

7. Risk Management

1. **Potential Risks:**
    - Delayed integration of third-party payment gateways.
    - High error rates due to untested edge cases.
    - Security vulnerabilities in handling payment data.
2. **Mitigation Strategies:**

- Ensure thorough testing of payment gateway integration using sandbox environments.
- Prepare comprehensive test data to include edge cases.
- Conduct regular security audits and code reviews.

---

8. Deliverables

1. Test Plan Document
2. Test Cases
3. Test Results Reports
4. Bug/Issue Reports
5. Final Test Summary Report

---

9. Approval

- Test Plan Approved by: [kiros Gebremedhin
- **Id:** ugr/170100/12**]**
- Date: [09/05/2017]