MEKELLE UNIVERSITY

ETHIOPIAN INSTITUTE OF TECHNOLOGY - MEKELLE

SCHOOL OF COMPUTING

DEPARTMENT OF SOFTWARE ENGINEERING

Software Testing and Quality Assurance:

Assignment on : Test Plan

Name	ID
MIKYAS HAFTAMU	UGR/170241/12

SUBMITTED TO: Inst. Mesele

SUBMITION DATE: Jan 16 2025

Test Plan Document for Payment Form

1. Test Planning

1.1 Objective

- Ensure quality and reliability of the payment form with chapa integration.
- Define the scope of testing to cover all functional and non-functional requirements.
- Allocate resources efficiently to maximize testing effectiveness.

1.2 Components of a Test Plan

1. Introduction

- **Overview of the Project**: The payment form allows users to enter their payment details, including first name, last name, email, and amount to process payments using a payment gateway (Chapa).
- **Testing Goals**: Validate the form's functionality, usability, and security, ensuring it meets business requirements.
- **Stakeholders and Roles**:
  - **Instructor**: Oversees the project.
  - **Collegue Tester**: Conducts all testing activities.
  - **Developers**: Fixes identified defects.
  - **Developer**: Defines requirements and ensures they are met.

2. Scope of Testing

- **Features to be Tested**:
  - Input validation for all fields (first name, last name, email, and amount).
  - Successful submission of valid data.
  - Error handling for invalid inputs.
  - Display of success and error messages.
- **Features Not to be Tested**:
  - Backend payment processing logic (assumed to be tested separately).
  - Integration with the payment gateway (Chapa).

3. Test Objectives

- Detect defects in the payment form.
- Validate that the form meets usability requirements.
- Ensure proper handling of error messages and input validation.

4. Test Strategy

- **Levels of Testing**:
  - **Unit Testing**: Validate individual components (input fields).
  - **Integration Testing**: Test interactions between the form and the backend.
  - **System Testing**: Validate the form as part of the entire application.
  - **Acceptance Testing**: Ensure the form meets user requirements.
- **Testing Types**:
  - **Manual Testing**: For exploratory and usability testing.
  - **Automated Testing**: For regression tests on form validation.
  - **Load Testing**: To assess performance under high traffic.
- **Test Environment Requirements**:
  - Web server running Laravel.
  - Access to the database for saving payment records.

5. Test Deliverables

- Test cases/scripts for all functionalities.
- Test data for various scenarios.
- Test reports summarizing findings.
- Defect logs for tracking identified issues.

6. Entry and Exit Criteria

- **Entry Criteria**:
  - The form must be fully developed and functional.
  - Test environment must be set up and accessible.
- **Exit Criteria**:
  - All critical defects must be resolved.
  - All test cases must be executed with a pass rate of 95% or higher.

7. Resources

- **Team Members**:
  - Instructor
  - Developer
  - Colleague Tester
- **Tools**:
  - Testing tools (e.g., PHPUnit, laravel Dusk for automated testing).
  - Bug tracking tools (e.g., JIRA).
- **Hardware/Software Requirements**:
  - Development machine with access to the application.
  - Browsers for testing (Chrome, Firefox, etc.).

8. Schedule

- **Timeline for Each Phase**:
  - Test Planning: 1 week
  - Test Case Development: 1 week
  - Test Execution: 2 weeks
  - Reporting: 1 week

9. Risk Management

- **Potential Risks**:
  - Delays in development may affect testing timelines.
  - Incomplete requirements could lead to missed functionalities.
- **Mitigation Strategies**:
  - Regular communication with stakeholders.
  - Early involvement of QA in the development process.

10. Approval

- **Approval Process**:
  - The test plan and deliverables will be reviewed and approved by the Project Manager and the QA Lead.

Test Plan Document for Payment Form

1. Test Planning

1.1 Objective

- Ensure the quality and reliability of the payment form.
- Define the scope of testing, including functional and non-functional requirements.
- Allocate resources efficiently to maximize the effectiveness of the testing process.

1.2 Components of a Test Plan

1. Introduction

- **Overview of the Project**: The payment form allows users to input their payment details, including first name, last name, email, and amount in order to process payments using the Chapa payment gateway.
- **Testing Goals**: Validate that the form functions correctly, handles user input appropriately, and integrates with the backend.
- **Stakeholders and Roles**:
  - **Project Manager**: Oversees project execution and timelines.
  - **QA Team**: Responsible for designing and executing test cases.
  - **Developers**: Address identified defects and implement fixes.
  - **Business Analyst**: Ensures that the form meets business requirements.

2. Scope of Testing

- **Features to be Tested**:
  - Input validation for all fields (first name, last name, email, amount).
  - Successful submission of valid data.
  - Error messages for invalid inputs.
  - Integration with the Chapa payment gateway.
  - Handling of success and error messages after form submission.
- **Features Not to be Tested**:
  - Backend payment processing logic (assumed to be tested separately).
  - Performance testing of the payment gateway.

3. Test Objectives

- Detect defects in the payment form and its integration with the backend.
- Validate that the form meets usability and accessibility standards.
- Ensure proper handling of error messages and user feedback.

4. Test Strategy

- **Levels of Testing**:
  - **Unit Testing**: Validate functions in the `PaymentController`.
  - **Integration Testing**: Test interactions between the form and the payment gateway.
  - **System Testing**: Validate the form as part of the overall application.
  - **Acceptance Testing**: Ensure the form meets user requirements and expectations.
- **Testing Types**:
  - **Manual Testing**: For exploratory and usability testing.
  - **Automated Testing**: For regression tests on form validation.
  - **Load Testing**: To assess performance under high user load.
- **Test Environment Requirements**:
  - Web server running Laravel.
  - Database access for recording payment details.

5. Test Deliverables

- Test cases/scripts for all functionalities.
- Test data to cover various scenarios (valid and invalid inputs).
- Test reports summarizing findings and defect logs.

6. Entry and Exit Criteria

- **Entry Criteria**:
  - The payment form must be fully developed and functional.
  - Test environment must be set up and accessible.
- **Exit Criteria**:
  - All critical defects must be resolved.
  - All test cases must be executed with a pass rate of 95% or higher.

7. Resources

- **Team Members**:
  - QA Engineers
  - Developers
  - Project Manager
- **Tools**:
  - Testing tools (e.g., PHPUnit, Selenium for automated testing).
  - Bug tracking tools (e.g., JIRA).

- **Hardware/Software Requirements**:
  - Development machine with access to the application.
  - Browsers for testing (Chrome, Firefox, etc.).

8. Schedule

- **Timeline for Each Phase**:
  - Test Planning: 1 week
  - Test Case Development: 1 week
  - Test Execution: 2 weeks
  - Reporting: 1 week

9. Risk Management

- **Potential Risks**:
  - Delays in development may affect testing timelines.
  - Incomplete requirements could lead to missed functionalities.
- **Mitigation Strategies**:
  - Regular communication with stakeholders.
  - Early involvement of QA in the development process.

10. Approval

- **Approval Process**:
  - The test plan and deliverables will be reviewed and approved by the Project Manager and the QA Lead.