**TDTP**

**Documentation**

Academic year: 2023-2024, Spring Semester

Team Name: ABN

Team members:

* Laurențiu Pricop (936)
* Alin-Gabriel Stanciu (936)

Contents

[1. Application Details. Investigated Features 3](#_Toc164118504)

[2. AC. IOs 3](#_Toc164118505)

[3. Testing Mission 3](#_Toc164118506)

[4. Testing Strategy 3](#_Toc164118507)

[5. Selected Test Design Techniques 3](#_Toc164118508)

[6. Test Design. Test implementation. Test execution. Test Report 4](#_Toc164118509)

[6.1. Test Design 4](#_Toc164118510)

[6.2. Test Implementation. Test Execution 4](#_Toc164118511)

[6.3. Test Report 5](#_Toc164118512)

[7. Issue Reporting 5](#_Toc164118513)

[8. Conclusions. Lessons Learned 5](#_Toc164118514)

# Application Details. Investigated Features

The application under test is a Physical Mail Manager designed to help users manage their physical mail more effectively. Its purpose is to streamline the process of managing and tracking physical mails and recipients, making it easier for users to organize and retrieve mail-related information.

Tested functionalitites:

* **User Registration and Authentication:** Users can create an account by providing a username and password and can log in to access the application's features.
* **Recipient Management:** Users can add new recipients by filling out detailed information such as Name, Street, Number, Block, Stair, Apartment, City, Commune, County, Postal Code, and Observations.
* **Letter Management:** Users can add letters associated with each recipient, including details like Code, Handwritten, Written, Sent, Received, Price, and Observations. Users can also view all letters associated with a specific recipient.
* **Letter Grouping:** Users can group letters by their recipients, then by their status (received or sent).

# AC. IOs

**Application Context:**

Testing is performed after the Physical Mail Manager application has been released. The primary goal is to ensure the main functionalities are working correctly, rather than improving the product itself. Key stakeholders include the project manager and customers, as the project serves a small mailing service.

Several constraints shape the testing process:

* **Stakeholders with Influence:** Focus on the needs and expectations of the project manager and customers.
* **Non-stakeholder Influence:** There are no significant external regulators influencing the testing.
* **Project Goals and Quality Criteria:** Ensure the main functionalities of the application are operational.
* **Available Resources:** Testers have experience with API testing but lack experience in GUI test automation. No pre-existing tests are available.
* **Product Features and Failures:** Identify the core functionalities and potential failure points of the application.
* **Consequences of Failures:** Evaluate how failures impact the service provided to customers and the project's success.
* **Failure Troubleshooting:** Simplify failures to facilitate quick resolution and stakeholder understanding.
* **Exposing Issues:** Highlight critical issues affecting the product's core functionalities and customer satisfaction.

**Suggested Strategy:**

* **Thorough API Testing:** Plan extensive API testing to ensure the backend services are functioning correctly.
* **Exploratory GUI Testing:** Conduct exploratory testing for the GUI to identify and address critical issues without relying on automated tests.
* **Ensure Operational Status:** Focus on verifying that the product's main functionalities are working as expected.

**Information Objectives:**

The information acquired during testing serves various objectives, tailored to the project's specific context:

* **Find Important Bugs:** Identify significant issues affecting the main functionalities of the application.
* **Assess Product Quality:** Evaluate the operational status and quality of the application post-release.
* **Support Project Management:** Provide the project manager with detailed reports on the application's performance and any critical issues.
* **Help Make Maintenance Decisions:** Offer insights to help determine necessary maintenance actions.
* **Prevent Operational Disruptions:** Identify issues that could disrupt the mailing service and ensure they are addressed promptly.
* **Predict Support Costs:** Estimate future support costs based on the identified issues and required fixes.
* **Find Safe Use Scenarios:** Identify scenarios where the application can be used reliably by customers.
* **Minimize Risk of Operational Failures:** Reduce the risk of failures that could impact customer satisfaction and service reliability.

# Testing Mission

**Testing mission for Part I:**

The mission for Part I is to identify critical bugs and assess the overall quality of the Physical Mail Manager application to ensure that user registration, authentication, and the core functionalities of managing recipients and letters operate without issues, thereby supporting a decision for production release.

**Testing mission for Part II:**

The mission for Part II is to validate that the application conforms to all specified requirements and is reliable for third-party use, ensuring that it accurately handles recipient and letter data as per the defined business rules.

# Testing Strategy

**Testing strategy for Part I:**

The testing strategy for Part I involves a combination of analytical and reactive testing approaches. Analytical testing will be used to assess the application’s quality against predefined requirements and specifications, focusing on critical functionalities such as user registration, authentication, and the management of recipients and letters. Exploratory testing, a reactive approach, will be applied to discover unforeseen issues and bugs by simulating various user interactions within the application. This combination ensures thorough coverage of both anticipated and unexpected problem areas.

**Testing strategy for Part II:**

For Part II, a methodical testing strategy will be employed, focusing on ensuring the application meets all specified requirements and standards. This involves detailed testing of business rules related to recipient and letter management, using predefined test conditions and scenarios. Additionally, process-compliant testing will be conducted to evaluate the product’s suitability for third-party use, ensuring adherence to industry standards and regulatory requirements.

# Selected Test Design Techniques

For Part I and Part II, various test design techniques will be selected to cover different test attributes and dimensions. Here's a detailed description of the techniques and a summary table:

**Part I:**

1. **Boundary Testing (BT):** This will focus on testing the boundary values for recipient and letter data fields to ensure they handle edge cases correctly.
2. **Exploratory Testing (ET):** This will involve exploring the application to identify unexpected issues, ensuring critical paths are tested from a user’s perspective.

**Part II:**

1. **Logical Expressions (LE):** This will be used to verify the business rules for managing recipient and letter data, ensuring logical conditions are handled correctly.
2. **Scenario-based Testing (ST):** This will involve testing the application using realistic scenarios to evaluate its behavior in practical use cases.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Part** | **Test Strategy** | **Test Design Technique** | **~~Test Attributes~~** | **Dimension covered** | **Students and Features** |
| Part I | Analytical | Boundary Testing (BT) | ~~Coverage, Reliability~~ | Coverage | Student1 (User Registration) |
| Logical Expressions (LE) | ~~Coverage, Credible, …~~ | Coverage | Student3 (Feature X)  Student4 (Feature Z) |
| Quick Tests (QT) | ~~Power, Representative, …~~ | Risk | Student5 (Feature Y) |
| Scenario-based Testing (ST) | ~~Credible, Motivating, …~~ | Activity | Student4 (Features, X, Z, Y) |
|  | Reactive | Exploratory Testing (ET) | ~~Coverage, Usability~~ | Risk | Student2 (Recipient Mgmt) |
| Part II | Methodical | Logical Expressions (LE) | ~~Coverage, Accuracy~~ | Coverage | Student3 (Letter Mgmt) |
|  | Process-compliant | Scenario-based Testing (ST) | ~~Credibility, Usability~~ | Activity | Student4 (Letter Grouping) |

# Test Design. Test implementation. Test execution. Test Report

# Test Design

*<Include all the information associated to the test design step when a specific test design technique is used. Each team member will fill out the details corresponding to Part I and Part II. The table below indicates a sample for Part Il.>*

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *Information objective (Part II): …* | | | | |
| **Student** | **Feature(s)** | **Test Design Technique** | **Details** | **Input, Expected output** |
| *Student1* | *Feature X* | *BT (Boundary Testing)* | *Variable n in [a, b] is tested* | |  |  | | --- | --- | | ***Input*** | ***Expected Output*** | | *a-1* | *…* | | *a* | *…* | | *a+1* |  | | *b-1* |  | | *b* |  | | *b+1* |  | |
| *Student4* | *Feature Z* | *LE  (Logical Expressions)* | *Business rules for variable a and b are tested* | |  |  |  | | --- | --- | --- | | ***Variable*** | ***Rule1*** | ***Rule2*** | | *a* | *a>0* | *a<=0* | | *b* | *b>0* | *b<=0* | | *result* | *False* | *True* | |  |  |  | |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

# Test Implementation. Test Execution

*<Each student will choose* ***ONE*** *of the test design techniques she/he included in the previous section. For the picked technique the test cases designed will be implemented using an automation framework (see* ***Lab04****) (SeleniumWebDriver + SerenityBDD, Postman, JMeter) should be emphasized. The team can decide if each student will have her/his own automation project, or they will share the same project on git such that it will include a package with the tests implemented by each team member. The table below will consist of the test cases implemented by each student.>*

|  |  |  |  |
| --- | --- | --- | --- |
| **Part** | **Student** | **Feature(s)** | **Input, Expected Output, Actual Output** |
| *Part II* | *Student1* | *Feature X* | |  |  |  |  | | --- | --- | --- | --- | | ***TCs*** | ***Input*** | ***Expected Output*** | ***Actual Output*** | | *TC01* | *a-1* | *…* | *… or* ***Passed*** | | *TC02* | *a* | *…* | *… or* ***Failed*** | | *TC03* | *a+1* |  |  | | *TC04* | *b-1* |  |  | | *TC05* | *b* |  |  | | *~~TC06~~* | *~~b+1~~* |  | *cannot be implemented* | |
| *Part I* | *Student4* | *Feature Z* | |  |  |  |  | | --- | --- | --- | --- | | ***TCs*** | ***Input*** | ***Expected Output*** | ***Actual Output*** | | *TC01* | *a=3, b=5* | *False* | ***False*** | | *TC02* | *a=-1, b=-3* | *True* | ***False*** | | *…* |  |  |  | |  |  |  |  | |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Test Report

*<This section will include the reports of the test execution, e.g., pie charts generated by the used tool, with #TCs run = #TCs passed + #TCs failed. If the entire team has worked on the same project, a single report/pie char should be provided. Otherwise, a pie chart/report should be presented for each team member.>*

# Issue Reporting

*<This section includes the application of the RIMGEA strategy for* ***at least one issue*** *found while performing testing. The type of issue can be coding bug or design issue. Highlight 2-3 relevant RIMGEA elements for the detected issue(s). Thereafter, report the bug or issue using the corresponding template (see* **IssueReport**). The bug/issues discussed in this section (and reported as well) refer to the entire team, not each team member.*>*

# Conclusions. Lessons Learned

*<Please include in this section final conclusions, lessons learned and personal considerations while working on TDTP (3-4 paragraphs). You can focus on the following aspects: type of application to be tested, amount of knowledge to use (related or not to testing), tools required to apply, team collaboration, test project organization, amount of time needed to fulfill the tasks, etc.>*