FINAL PROJECT-SSW567

PART 4-TEST PLANNING

Team Members:
1.Raj Palival
2.Pranav Muralidhar Rao
3.Amith Vishnu
4.Kaijie Ma

Table of Contents

1. INTRODUCTION

1.1 Visual inspection and machine-readable methods (optical character recognition) should be used to transmit information on machine-readable travel documents (MRTDs) required for universal interoperability. There are two of his sections: MRZ (Machine Readable Zone) and VIZ (Visual Early Zone). MRZ has two lines. The passport type, country of issue and name of the holder are all listed on the first line. Passport number, country code, date of birth, gender, expiry date and PIN are all on line 2. In addition to this information, four check digits are added to the middle and end. The check digit from the hardware scanner is compared with the check digit from the database information field. Decoding and encoding routines are used for these purposes.

1.2 The smallest block of code that can be logically separated in a system is called a unit, and unit testing is a way of testing a unit. This is a method, function, subprogram, or property in most programming languages. We are trying to cover the functionality of the software.

1.3 We are using agile method here. Software testing that adheres to Agile development principles is called Agile testing. Agile testing practices align with iterative development practices in which the test team and the customer create requirements incrementally.

1.4 Testing should be done early and often in agile development. As a result, tests are continuously run as features are deployed, rather than waiting until development is complete. Test order is determined by functionality and code coverage. In one iteration, testers try to complete as many tests as possible.

2. Reference Other Relevant Documents

Optional data elements should be entered in the issuing organization's native/working language in addition to English, French, or Spanish for convenience. Zone VIs allow complete entry of arbitrary data in local script and/or local language.
When inserting translations, you must use forward slashes to distinguish between different languages. Punctuation may appear in the VIZ.

3. Testing Scope

3.1 Attempt to test the functionality of the entire software. The software must be able to read the information from his MRZ on the travel document after scanning it with a hardware device scanner. The two strings are the input data for the program. The software must be able to convert her two texts from the MRZ into the appropriate fields and determine the appropriate check digits for the fields. The format of each scan result should be consistent. The application needs to be able to encode two texts in the travel document using data fields retrieved from the database. Compared to the first phase, this process is reversed. Since there is no code from the database component, we can assume that the database function is not ready yet. Create a database interaction method for testing purposes but leave it empty. Each element of the string must be validated, such as passport type, country of issue, name of holder, etc. Applications must be able to indicate discrepancies between specific information fields and check digits and identify where discrepancies occur.

3.2 We do not verify whether the passport type matches your identity or whether the country of issue is related to the holder's name. Only unit tests are run. No system, integration, or acceptance testing is performed.

3.3 Code coverage and test prioritization according to functionality became our main criteria. Test the most important requirements first. B. Get the string from the document. Test newly created or modified code. If any changes were made to the code, test again. The code should be tested. To save time, use equal partitioning.

4. Testing Approach

4.1 key factors:

Response time: Time taken to scan one document.

Availability: Any kind of MRZ strips can be passed in our function.
Functionality: It get all characters from the passport.
Scalability: Any kind of name, DOB, passport number and personal number and expiration date.

4.2 key risks:

- **Development:** debugging during the development.
- **Staffing:** teach the staff how to use the program to avoid wrong operation
- **Technology:** connecting between software and hardware
- **Schedule:** spend enough time to test all the cases.

4.3 Success Criteria:

General function of MRTD.py file with all functions. These processes give the desired result. 99% of the code is covered, which is far more than we expected.

4.4 **Contingency Plans:**
The plan is to debug until the program matches the requirement.

4.5 **Pass/Fail Criteria:**
Any part different form success criteria are failed criteria and the other is pass criteria. If the decode function returns all the information fields from the MRZ string, it's a pass criterion else it's a failure criterion. The process of putting strings of characters, such as letters, numbers, and other special characters, into a structured structure arranged for efficient transmission is called encoding.

4.6 **Entry/Exit Criteria:**
Entry criteria- All the functions of the requirement are completed in the mrtd.py file.

Exit criteria- Code coverage should be greater than 80% and kill the mutants encountered.

4.7 **Testing criteria and Checkpoints:** Testing criteria includes all the criteria's explained above, each point in success criteria is checkpoint.

4.8 **Test Deliverables:**
Software that meets all requirements of the Success Criteria. Additionally, additional test cases should be written to filter out mutants and code coverage should be at least 80%. Databases and hardware scanners should be mocked.

4.9 **Testing Budget:**
Sufficient passport data must be linked. Since everyone's information is private, you might consider creating random cases for testing.

4.10 **Tools are you going to use:**
Static Code Analyzers: Analyze source code to identify anything that doesn't seem right, Syntax errors, Unreachable code, Undeclared variables, Uninitialized variables, Parameter type mismatches, Uncalled functions, Variables used before initialization, Unused function return values, python unit test library, python mock library, python mutpy and python coverage.

4.11 **Automation Strategy:**
Includes configuration management, testing frameworks, and bug tracking systems. A parallel development path may be required to be effective.

4.12 **Types of testing we are performing:**

Requirement reviews: Organize and track requirements; trace tests for each requirement.

Design reviews: Inspections held and passed

Unit testing: code written by the developer to test a small part of the functionality of a method, each unit test should have only one testing purpose

OATS: Frequency of use, Risk or cost of failure, Likelihood of failure.

Integration testing: is the first time that the system starts to be working as a system rather than individual modules, helps to test the design and architecture of a system, might be the first occasion on which reliability, performance, or security issues appear.

Usability testing: Create a test plan – Gather representative users – Ask them to perform relevant tasks – Observe their behavior – Collect measurements– Report results – Iterate to improve your design
Performance testing: A technical Investigation of the System Under Test conducted to provide the Stakeholders with Quality-related information, Performance testing driven by use cases, performance requirements; relation to functional requirements.

Reliability testing: Overview of reliability and availability, Link between reliability and performance testing, Diagnosing reliability issues from performance test results, Reliability metrics and requirements, Operational profiles to plan reliability and performance tests.
System testing: System testing is black box testing Focus on the system's external behaviors, not the system's implementation or architecture.
Test platform: Physical test facilities and dedicated computing platform. A computer which can running our program and a scanner which can scan the passport.
Progress of testing: when near the end of the release, not much time to recover and meet the ship date need to know if you will meet the schedule or not.
Metrics: some measure that characterizes a property of an entity
When ready to ship the system: Accuracy is 90% Precision = 95% Recall = 98% or above. When our unit test report passes the success criteria then its ready to ship.

5.SCHEDULE:

Part 1: Jan 1 to Feb 20

Part 2: Feb 21 to May 10

Part 3: May 11 to Aug 7

Part 4: Aug 7 to Sep 1

6.Approvals:

Name:	
Role:	
Date:	
Signature:	