Test Plan

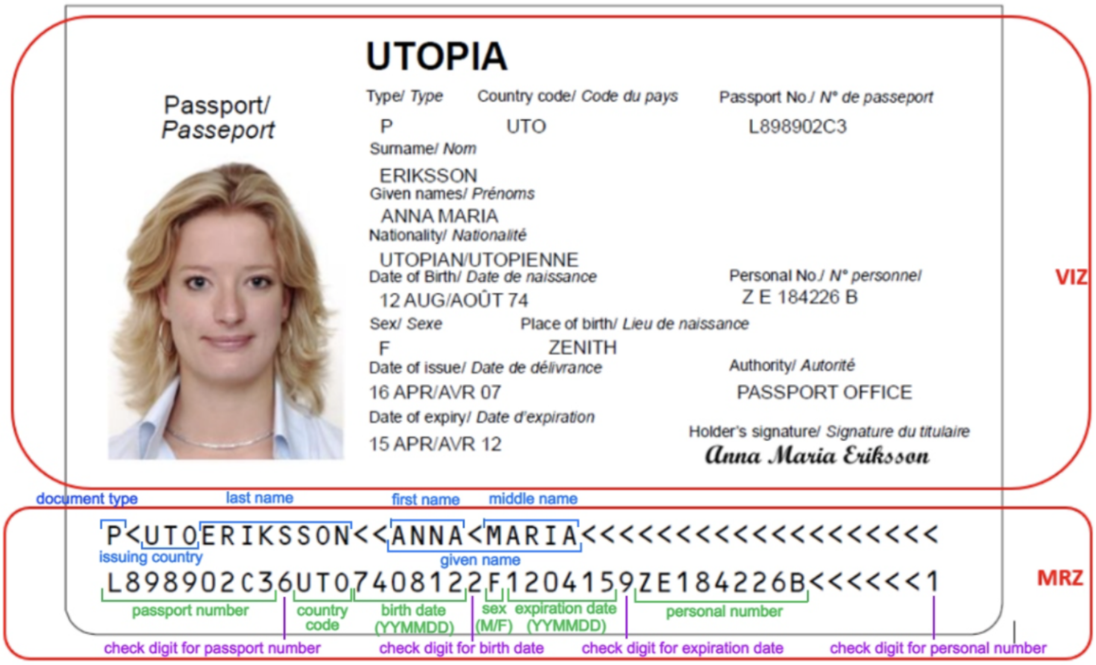
By CYA(Carmen, Yujun, Akash)

Introduction

This is our team's test plan for testing the Machine Readable Travel Documents (MRTD). Here we will be discussing an overview of the system and how we are going to be testing. Along with information regarding the development process. The application is a machine which reads travel documents of a fellow passenger. The goal of the system is to obtain the information from the travel documents of the person. We will be running unit test cases and performance testing procedures to check the system's functionalities. We have chosen the agile methodology to develop the system. The development approach determines when we can begin testing and start rectifying the underlying problems.

Reference from given document

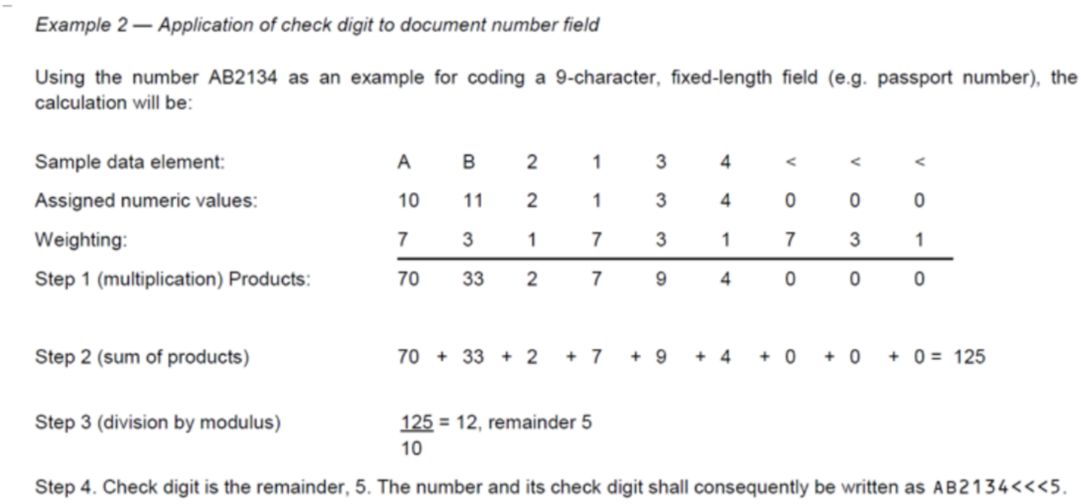
A machine-readable travel document (MRTD) should present information necessary for global interoperability using visual inspection and machine-readable (optical character recognition) means. The following figure is an example. It is composed of two parts: a visual inspection zone (VIZ) and a machine-readable zone (MRZ).


UTOPIA
Passport/ Passeport
Type/ Type
P
Country code/ Code du pays
UTO
Passport No./ N° de passeport
L898902C3
Surname/ Nom
ERIKSSON
Given names/ Prénoms
ANNA MARIA
Nationality/ Nationalité
UTOPIAN/UTOPIENNE
Date of Birth/ Date de naissance
12 AUG/AOÛT 74
Personal No./ N° personnel
Z E 184226 B
Sex/ Sexe
F
Place of birth/ Lieu de naissance
ZENITH
Date of issue/ Date de délivrance
16 APR/AVR 07
Authority/ Autorité
PASSPORT OFFICE
Date of expiry/ Date d'expiration
15 APR/AVR 12
Holder's signature/ Signature du titulaire
Anna Maria Eriksson
VIZ
document type
last name
first name
middle name
P<UTOERIKSSON<<ANNA<MARIA<<<<<<<<<<<<<<<<<<<
issuing country
given name
L898902C36UTO7408122F1204159ZE184226B<<<<<1
MRZ
passport number
country code
birth date (YYMMDD)
sex (M/F)
expiration date (YYMMDD)
personal number
check digit for passport number
check digit for birth date
check digit for expiration date
check digit for personal number

The MRZ contains two lines. The first line specifies the Type of passport, the issuing country, and the name of the holder. The second line specifies the passport number, country code, birth date, gender, expiration date, and personal number. In addition to these information fields, there are four check digits inserted in between and at the end of the information fields.

In the above example, they are “6”, “2”, “9”, and “1”. The check digit serves for checking the correctness of the information fields.

Following is an example illustrating the algorithm for calculating the check code. Assume that the calculation method for composite check digits is the same for all MRTDs.



Example 2 — Application of check digit to document number field

Using the number AB2134 as an example for coding a 9-character, fixed-length field (e.g. passport number), the calculation will be:

Sample data element:	A	B	2	1	3	4	<	<	<
Assigned numeric values:	10	11	2	1	3	4	0	0	0
Weighting:	7	3	1	7	3	1	7	3	1
Step 1 (multiplication) Products:	70	33	2	7	9	4	0	0	0

Step 2 (sum of products) 70 + 33 + 2 + 7 + 9 + 4 + 0 + 0 + 0 = 125

Step 3 (division by modulus) $\frac{125}{10}$ = 12, remainder 5

Step 4. Check digit is the remainder, 5. The number and its check digit shall consequently be written as AB2134<<<5.

Step 1: multiply the numeric values of each digit with a weighting sequence. Note that “A” maps to 10, “B” maps to 11, and thereafter. Special symbols, such as “<”, always map to 0. You should always use the same weighting sequence of 7, 3, and 1 as shown in the above example in the scope of this project.

Step 2: Add up all the products from the previous step.

Step 3: Divide by a modulus of 10.

Step 4: The remainder will be the check digit, which is the final output of this algorithm.

Testing Scope

We are performing 3 types of tests- Requirements, Unit and Performance. We Conduct the requirements testing first as we need to first find out the identify any ambiguity in the current requirement specifications, and rewrite the requirement specification with details and clarifications, by either obtaining additional information from either the project background information or making assumptions that remove the ambiguity.
We then move onto unit test cases which are placed as soon as we are ready with the component to test. Hence all units of the systems are being tested after the completion of its associated component being tested.
At the end of the process we conducted the performance testing where we tested for its efficiency, speed, etc as a system working together.

We did not conduct integration testing, or security testing for the system as it was not required by the constraints and the system has yet a lot of more development to go through before it is perfected.

Testing Approach

The key factors which are taken into consideration while making the system is the reliability of the output given by it. The system should read all the needed data from the travel documents which are being scanned and if it is not able to conduct the function, an override can be performed where the information is manually typed into the system. The keys risks of such a system may be that it reads the characters wrong which leads to the incorrect data being stored, this is a major risk as we cannot afford a misinterpretation as the document data is unique to each person and hence is very sensitive information. The success criteria for the system would be to render the right output, which is reading and storing the data of the travel documents accurately. The contingency plan as stated earlier would be to manually read and enter the document data into the system when it fails to do so by itself.

The pass/fail criteria would be if the system is able to render the document details accurately after it is scanned by the machine and is processed by the system to render the output. The enter criteria would be when the system is able to render output of the information and the exit strategy would be when we can be sure that the data which is read is absolutely accurate.

The testing criteria is that we need to be assured that the data being read and stored to its absolute highest efficiency because there is no room for error. There are several checkpoints with respect to testing which after thorough discussion with the team we have decided are the initial requirement testing where we go through the requirements given and we see for any inadequacies in the given detail and try to dig deep into to reach an effective conclusion. We then use unit testing methods throughout the programming structure, and see the usage and the outcome of the components running in the system and its accuracy. Once we have the system up and ready we go ahead with the performance testing of the system, where we test for the speed, efficiency, etc of the system.

The testing deliverables for each part are:

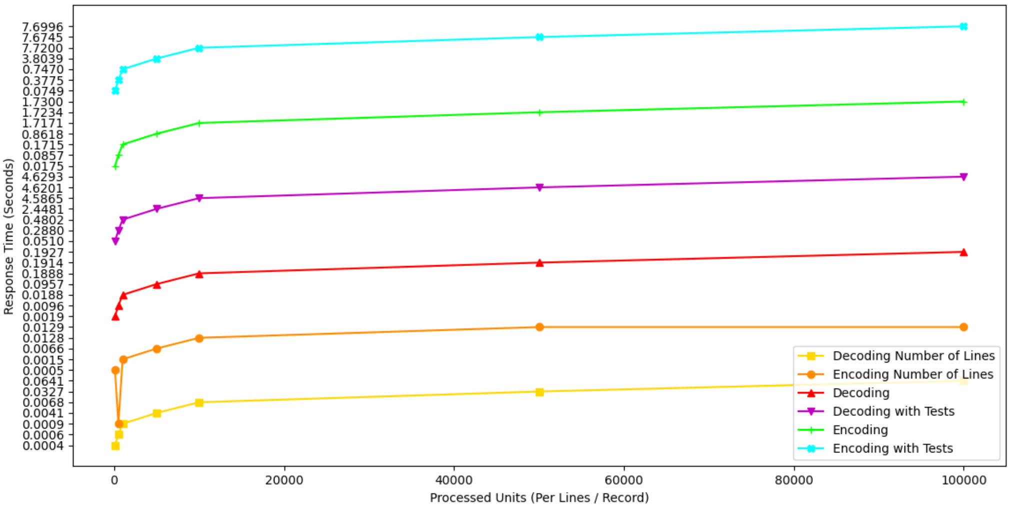
- First we have the requirement testing which we have to deliver the inadequacies of the requirement which is provided, this is in the preproduction stage of the development.
- We then have the unit testing methods whose aim is to deliver the reliability of individual components of the system.
- Lastly we have the performance testing where the goal is to test the running time, efficiency, speed, etc of the entirety of the system.

The budget for testing is minimal as we are students who are developing and testing the systems. We automate it in such a way that when the document is placed below the scanner the correct details are rendered.

The Platforms that were used for unit testing is the unittest package of python, for the performance we used a custom code which ran the system and rendered the output and for the requirement testing it was thoroughly discussed and was drawn out by a fellow team member.

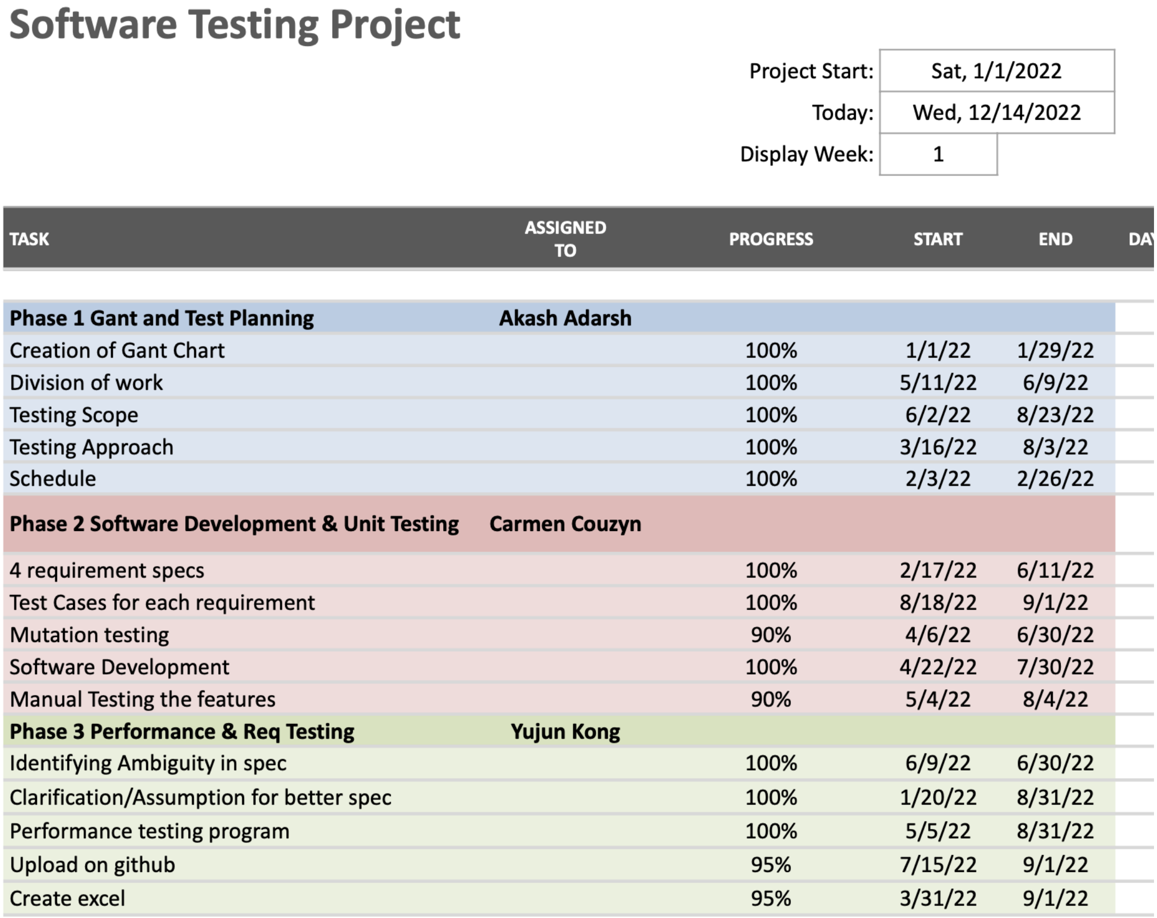
For requirement & design reviews, we conducted within our group which was thoroughly discussed and noted off, all team members had their say in the matter and the requirement plan was drawn out by one of the members. The unit testing eas conducted using the inbuilt function of python called unittest. The integration, usability, reliability and system testing were also conducted using inbuilt functions and custom functions which were built by the team members.

We measure the progress of the testing by the metrics given in the deliverables list and we see that all of them are met. We use reports provided by the graphs of performance testing as shown below:



Once we are sure that the system qualifies all the testing metrics, we get it ready to ship. We then proceed to ship it.

Schedule:



Software Testing Project

Project Start:	Sat, 1/1/2022
Today:	Wed, 12/14/2022
Display Week:	1

TASK	ASSIGNED TO	PROGRESS	START	END	DA
Phase 1 Gant and Test Planning	**Akash Adarsh**				
Creation of Gant Chart		100%	1/1/22	1/29/22	
Division of work		100%	5/11/22	6/9/22	
Testing Scope		100%	6/2/22	8/23/22	
Testing Approach		100%	3/16/22	8/3/22	
Schedule		100%	2/3/22	2/26/22	
Phase 2 Software Development & Unit Testing	**Carmen Couzyn**				
4 requirement specs		100%	2/17/22	6/11/22	
Test Cases for each requirement		100%	8/18/22	9/1/22	
Mutation testing		90%	4/6/22	6/30/22	
Software Development		100%	4/22/22	7/30/22	
Manual Testing the features		90%	5/4/22	8/4/22	
Phase 3 Performance & Req Testing	**Yujun Kong**				
Identifying Ambiguity in spec		100%	6/9/22	6/30/22	
Clarification/Assumption for better spec		100%	1/20/22	8/31/22	
Performance testing program		100%	5/5/22	8/31/22	
Upload on github		95%	7/15/22	9/1/22	
Create excel		95%	3/31/22	9/1/22	

Approval

The stakeholders, that is the people who have asked for the product to be made, the customers. They play a very vital process throughout the testing as we need their inputs several times first during the requirements while finding the inadequacies. Then we need to test the system in real time while doing performance and acceptance testing, they need to be highly cooperative and communicative during the process. Hence they play a vital role throughout.