Functional Specification

Name: Stephen McDonagh
Project Title: Passport Application System
Supervisor: Alistair Sutherland
Module: CA400

0 Table of Contents

1 Introduction

1.1 Overview:

The idea of this project came from my desire to complete and learn about developing a full end to end web application. I thought it would be a useful skill to have some experience of front-end technologies, web services, as well as including some CI/CD into the development cycle. I decided to do a passport application system as the work with facial detection really interested me and I felt it would be a tough part of the project I would enjoy solving. Doing this also gave me the opportunity to include to unique feature for the user which will be detailed in the following document. The application will be developed on a windows PC, fully tested and deployed onto an AWS EC2 t2 micro server. The project will be a fully working passport application system that the user will be able to apply online, ask necessary questions and receive updates about the progress of their passport application.

1.2 Business context:

As passports and the approval of them is a government responsibility, this application process must go through a government department. Realistically ads would not be run on a government run web application, but the software for the application could be distributed to governments around the world. This system will be an improvement on the manual system that occurs around the world.

1.3 Glossary:

AWS - Amazon Web Services
EC2 - Elastic Cloud Computing
SMTP - Simple Mail Transfer Protocol

2 General Description

2.1 Product & System Functions:

The user will be given the option to apply for a new passport if they have never owned one before or if they are looking to renew a current passport. If they want to renew a passport that running out of date they will need to have their details already in the Applications_Completed table and have less than 1 year left on their current passport.
To complete the passport application, there will be several steps the user will need to complete before being given confirmation. They will need to enter in the necessary details some of which include name, address, PPS number as well as a passport photo in JPEG/PNG form. The application will do checks that all the information entered is of a valid form and that the photo is of ‘passport quality’. To ensure the photo is passport quality one face will need to be centred, eyes open, mouth closed etc.
If the user has any queries about the application, they can use the chatbot that will be integrated into the UI which will be trained to be as helpful as possible to answer any questions the user might have. Some questions I’m expecting people to have include “Where can I find my PPS number?”, I will need to do a lot of testing for this part to ensure it answers all types of questions that would be appropriate for a passport application.
The user will need to then pay for their passport. The price will be depending on the length of passport, delivery option etc. This will likely just be a simulation of payment as there would be a lot of ethical problems involved with taking in actual card details.
After successful completion of the application process, the user will be send out a confirmation email telling them that their application has been successful. This will be done using Google's SMTP server.
The user can use the unique number they have been sent to track their application and to get an estimate of how long it may take to get their passport

2.2 User Characteristics & Objectives:

The expected target users will be the Irish public, but more specifically members of the public looking to get a passport. I plan on having the UI clear, consistent, responsive and overall a good experience for the user.
The user will require not previous knowledge knowledge of the system to complete and application.. The user will just need to input information correctly and follow the guidance on screen. I plan on making the margin of error basically zero to eliminate this. They will need to be able to take a picture of themselves and input that into the frontend of the web application.
There will be no remembering/ learning necessary on the users part. This will ensure that they will not get frustrated during the process.
With regards to error handling, if the user has input data incorrectly, in the wrong format, this will be handled by the system and inform the user that they will need to re-enter information.

2.3 Operational Scenarios:

2.3.1 User Enters Passport Photo

Current System State:
The current system state consists of the user entering in their personal information in order to complete the application

Informal State:
The user uploads PNG/JPEG picture.

Next Scenario:
The system will then do checks to see if the passport is a high standard one and meets all the requirements necessary

2.3.2 User Enters Incorrect Passport Photo

Current System State:
The user is entering in their personal information in order to complete the application.

Informal State:
The user uploads picture in incorrect format/ picture doesn't meet standards required.

Next Scenario:
The system will tell the user that the picture is not as expected. Will give the user possible solutions and ask the user to try again.

2.3.3 User Has A question Regarding Application Process

Current System State:
The current system state consists of the user trying to complete the application process. The have a question about a specific part and wants it to be answered.

Informal State:
The user uses the applications in-built chatbot to find out where they can find their PPS number.

Next Scenario:
The system will inform the user of various places that they can find their PPS number.

2.3.4 User Must Pay For Passport

Current System State:
The current system state consists of the user having entered all the relevant informations

Informal State:
The user inputs their visa/credit card information

Next Scenario:
The system will do checks that they have entered visa card information(VISA cards begin with 4319/ have 16 digits)

2.3.5 User Completes Application

Current System State:
The current system state consists of the user having successfully entered card details.

Informal State:
The user completes the application.

Next Scenario:
The user receives email confirming application is now complete. This email will include details about how to check progress of passport.

2.3.6 User Checks Progress

Current System State:
The current system state consists of the user having successfully completed the application process.

Informal State:
The user enters their unique code they received by email.

Next Scenario:
The user can now see how long until their passport will be ready.

2.4 Constraints:

Time:
With a project of this size, running out out time is always going to be a problem. My application will need to be well tested using Junit, selenium & Mockito. I will want to do my testing during the development process to identify bugs early instead of late. This is something I learned from my 3rd year project as well as my internship. I will also need to fully document the process through doing a blog and also complete a user manual as well as functional specification for this project.

Frontend Development:
Using Angular to develop the frontend of my application is something i have never done before. This will require a lot of research at the beginning to understand how to not only build the

frontend, but also how to connect to the backend(database/ java code) using restful web services.

CI-CD/Pipelines:
Another learning constraint is setting up pipelines to run my tests. I believe this would be impressive to do so and it would be worth the time to set up. I would like my unit tests to run after a commit to my gitlab repository.

3 Requirements

The following section will be split into two smaller sections. The functional requirements part will describe what my web application will do, the non-functional requirements look at the criteria that the web app should conform to.

3.1 Functional Requirements:

3.1.1 Input fields

Description:
As part of the process of completing an online passport application, there will be various input fields that the user will need to complete. This needs to be as easy as possible for the user with very little complexity

Criticality:
This is an extremely crucial part of the application. These input fields will need to be well tested so that the user cannot input an incorrect value. These values will likely be stored in the database so an incorrect value could cause problems with the database.

Technical Issues:
Unit tests will need to be completed to ensure no problems will occur. These will be simple enough Junit4 tests to write. These will be run in a pipeline with all the other tests.

Dependencies with other requirements:
As mentioned earlier, there is a dependency on these inputs being correct with the database.

3.1.2 Chatbot

Description:
The user will want to interact with the built-in chatbot in my application. They will have questions about the application process and will want them answered. This will ensure they have a good experience with my web application

Criticality:
This is another crucial part of the app. If the user needs to search elsewhere for answers to their questions then they will be less happy with my application.

Technical Issues:
The integration into the UI is going to be tough with this. It will take some research to get this working correctly.

Dependencies with other requirements:
The only dependency is with the UI as mentioned before.

3.1.3 Web Application Checks Photo Quality

Description:
The user will be required to enter a photo in the correct format. The application will need to do checks that it is of a high enough standard. This will mean checking that there is only one person in the picture, that the users eyes are open and that the users mouth is closed.

Criticality:
This is probably the most important part of the application. It will need to be well tested to make sure it is working as expected

Technical Issues:
I have researched and found that OpenCV can be used for facial detection. I haven't fully decided if this is the best method to go about completing this task. Learning how to integrate this into my application could turn out to be difficult

Dependencies with other requirements:
This will have dependency with the users input.

3.1.4 Email notification

Description:
After the user has successfully completed their passport application, they will receive confirmation of this. The user will have needed to have entered their correct email to get this option.

Criticality:

I believe this is an important part to the project. It is important for the user to be told that their application was successful.

Technical Issues:
From my research, I have found that I could use Google's SMTP server to help with this section.

Dependencies with other requirements:
This will have dependency with the users input.

3.2 Non-Functional Requirements:

3.2.1 Web Application Will Be Easy To Use

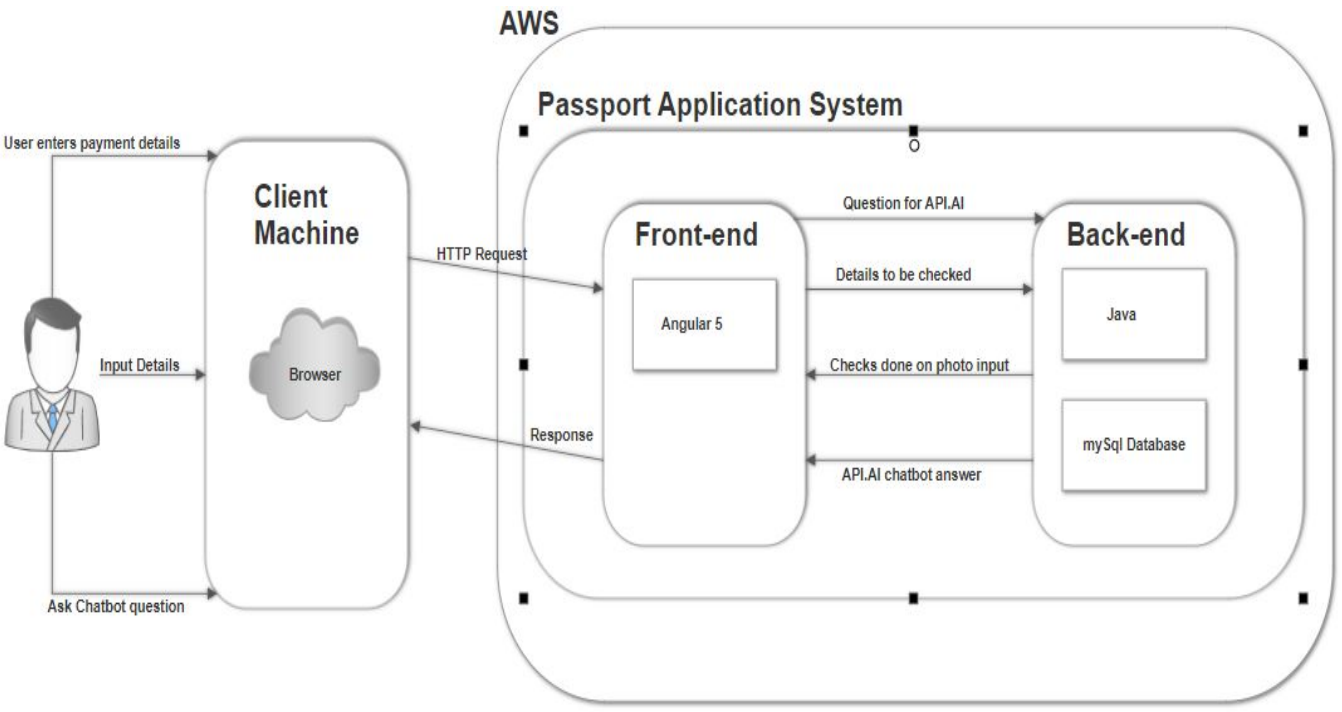
Description:
The process should require no thought from the user. This will mean they have a good UX.

Criticality:
Usually non-functional requirements are less important than functional requirements. But I will need to make sure my IU is as good as possible.

Technical Issues:
I will need to have a good understanding of Angular.

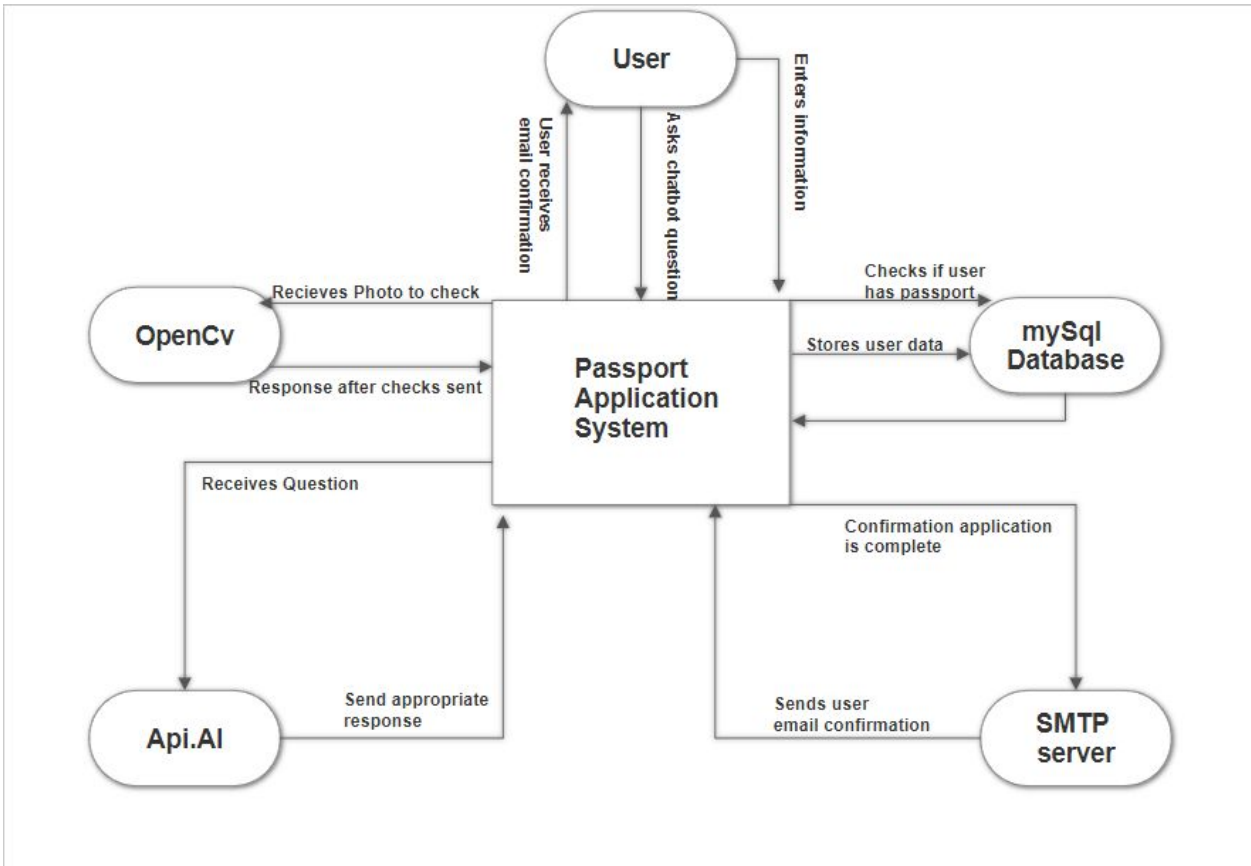
Dependencies with other requirements:
No dependency with other requirements

4 System Architecture

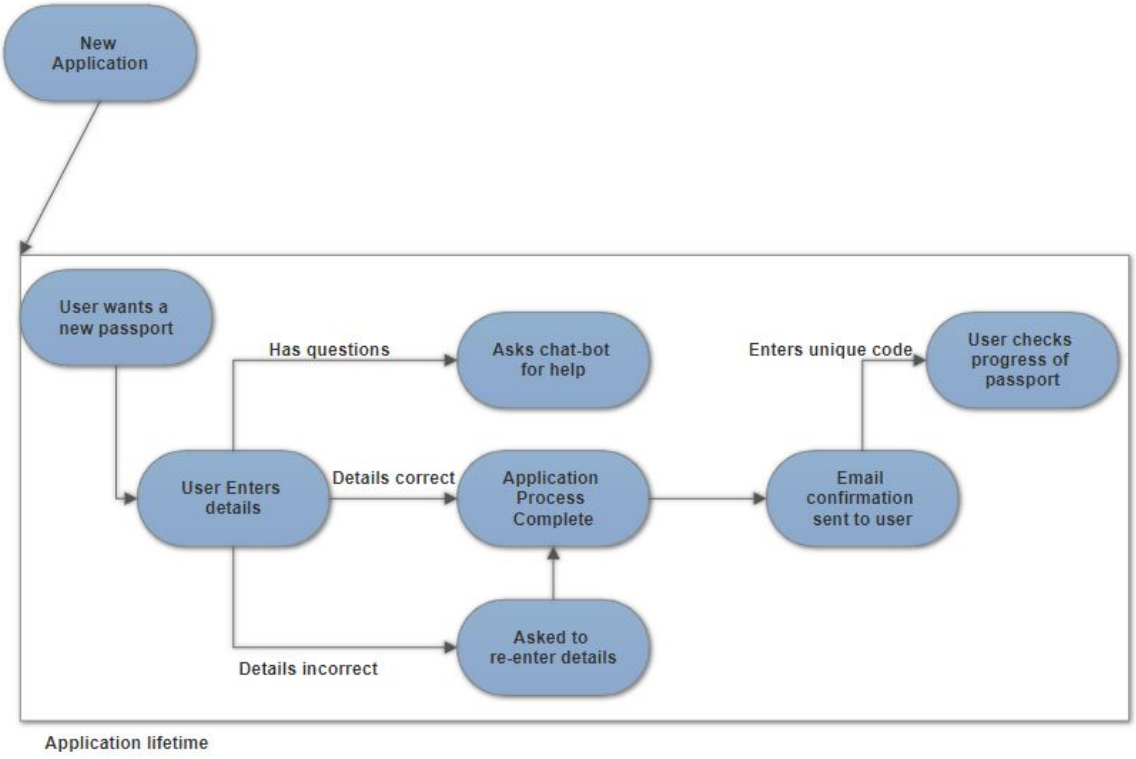


5 High Level Design

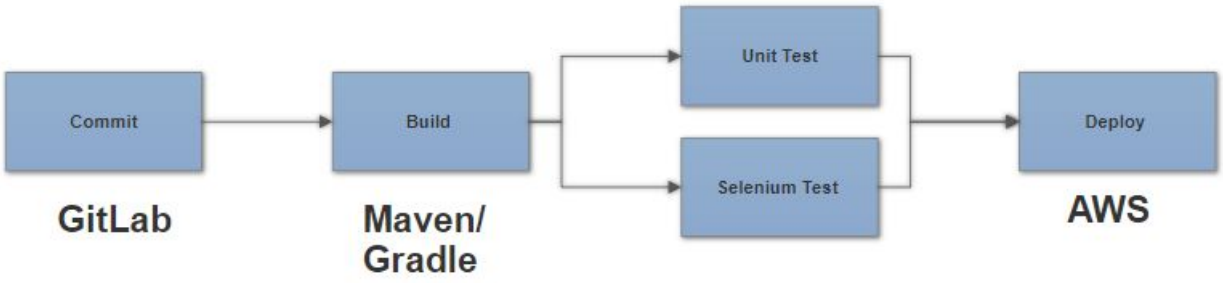
5.1 Context Diagram



5.2 Data Flow Diagram



5.3 CI CD Diagram



6 Preliminary Schedule

6.1 Gannt Chart

	Task	Start	End	Duration (days)
	Passport Application Syste⊖	9/20/18	5/20/19	167
1	Identify Idea	9/20/18	10/3/18	10
2	General Research	9/20/18	11/30/18	50
3	Proposal	10/29/18	11/5/18	6
4	Functional Spec	11/5/18	11/30/18	19
5	Front end/Angular5	12/1/18	12/14/18	10
6	OpevCV	12/14/18	2/1/19	33
7	Exams	1/7/19	1/19/19	10
8	AWS	2/2/19	2/13/19	8
9	Pipeline setup - Jenkins	2/13/19	2/21/19	6
10	mySQL database setup & connection	2/21/19	3/7/19	11
11	Chatbot training ⊖	3/14/19	3/26/19	9
11.1	Chatbot Ui integration	3/14/19	3/26/19	9
12	More Testing	3/25/19	3/28/19	4
13	Bug Fixing	3/28/19	4/4/19	6
14	Ui Improvements	4/4/19	4/16/19	9
15	Google SMTP Server	4/16/19	4/26/19	9
16	Documentation	4/27/19	5/3/19	5
17	Exams	5/7/19	5/20/19	10

7 Appendices

1. https://docs.angularjs.org/tutorial
2. https://docs.docker.com/
3. https://jenkins.io/doc/
4. https://docs.opencv.org/
5. https://www.digitalocean.com/community/tutorials/how-to-use-google-s-smtp-server
6.