

User Manual



Project Name: Passport Application System

Student Name: Stephen McDonagh

Supervisor: Alistair Sutherland

Module Code: CA400

Student Number: 14518233

Submission Date: 17-05-2019

Contents

1 Goal	3
2 Installation Guide	3
2.1 Frontend Set up	3
2.2 Backend Set up.....	4
2.3 Database Set up	4
3 Completing Passport Application.....	5
3.1 Input Required	5
3.2 Date of Birth Incorrect	5
3.3 Successfully completed Application.....	5
3.4 Unsuccessfully Completed Application.....	6
4 Facial Detection.....	7
4.1 Image Not Accepted.....	7
4.2 Image Accepted	7
5 Using Chatbot.....	8
6 Email Confirmation	9
7 About Section.....	9

1 Goal

The goal from my final year project was to implement facial detection in a passport application system. I wanted the application process to be quick and easy to understand for the user. I wanted the user to be told almost instantly if their picture has been accepted or not. I wanted to include features in the application such as email confirmation and a chatbot to help the user have the easiest experience possible.

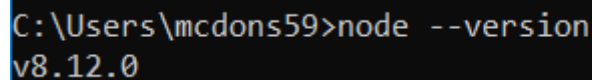
2 Installation Guide

My web application is running locally, therefore to run it on your own machine you will need to follow these installation steps.

2.1 Frontend Set up

For my frontend I am using Angular6, to use on your machine please follow these steps

- Visit the Following link and download Node <https://nodejs.org/en/download/>
- Open your command prompt and check node is installed as follows

A terminal window with a black background and green text. The prompt is 'C:\Users\mcdons59>' and the command entered is 'node --version'. The output is 'v8.12.0'.

```
C:\Users\mcdons59>node --version
v8.12.0
```

- You will then need to install the Angular CLI. To do this run the following command:
npm install -g @angular/cli
- Next you will need to create a basic Angular project. To do this run the following command:
ng new ProjectName
- Then check that this is working correctly by running this command:
cd ProjectName
npm start
- This will run the basic Angular project, you can open this up in your web browser by typing localhost:4200.
- Next you will need to clone my project from my Gitlab repository.
- Then run the same command as before to start up the application
npm start
- You will have some errors but that's okay because you're just missing some of the dependencies that are built into my application.
- Once these dependencies have been installed your front end is set up

For more information on installing an Angular project, I found the following link extremely helpful <https://angular.io/guide/quickstart>

2.2 Backend Set up

For my backend I'm using Springboot framework. But first you will need to install Java8

- Install java here <https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>
- Then run the following command:

```
C:\Users\mcdons59>java -version
java version "1.8.0_191"
Java(TM) SE Runtime Environment (build 1.8.0_191-b12)
Java HotSpot(TM) 64-Bit Server VM (build 25.191-b12, mixed mode)
```

- Generate Springboot project here <https://start.spring.io/>
- Next you will need to install Maven, I am using this as my build tool. Navigate to the following link <https://maven.apache.org/download.cgi>
- To check maven is installed correctly, run the following command:

```
C:\Users\mcdons59>mvn -v
Apache Maven 3.6.0 (97c98ec64a1fdfee7767ce5fffb20918da4f719f3; 2018-10-24T19:41:47+01:00)
Maven home: C:\Users\mcdons59\apache-maven-3.6.0\bin\..
Java version: 1.8.0_191, vendor: Oracle Corporation, runtime: C:\Program Files\Java\jdk1.8.0_191\jre
Default locale: en_IE, platform encoding: Cp1252
OS name: "windows 10", version: "10.0", arch: "amd64", family: "windows"
```

- Now your backend is set up

2.3 Database Set up

For my database I am using SQLite3. You will need to download and install both SQLite & the SQLite browser to view your database easily.

- SQLite: <https://www.sqlite.org/download.html>
- SQLite browser: <https://sqlitebrowser.org/>

3 Completing Passport Application

The following section outlines the process a user must take to successfully complete a passport application.

3.1 Input Required

As part of the application, the user must fill in all input fields. If the user interacts with an input field but does not fill it in, they will get the following error.

Name

Name is required.

3.2 Date of Birth Incorrect

There is customized input validation for some of the user inputs such as date of birth and email address. These inputs are expected to be in a certain format. The format is clearly shown to each user, but if they incorrectly put in their date of birth they will get the following error message

03/10/1996

Please enter a valid date of birth (DD-MM-YYYY)

3.3 Successfully completed Application

Upon successfully completing an application, the user will be told so on the user interface. The completed message looks like this. As you can see below, the user is told that they have completed the application and the form field has also been cleared. The users details have now been stored in the SQLite Database.

Your passport application has been submitted. Please check your email. Thank You.

Name

Date Of Birth (DD-MM-YYYY)

Email Address

Choose File No file chosen

Image Accepted.

PPS Number

Phone Number

SUBMIT

3.4 Unsuccessfully Completed Application

If the user has not added an entry for all input fields and tries to submit the application, they will be shown the following error. As you can see below the user has not filled in the 'Name' input field.

APPLICATION



Please complete application before submitting

Name

Name is required.

4 Facial Detection

4.1 Image Not Accepted

All images inputted must be of type JPG. An image is not accepted if no faces have been detected or if multiple faces have been detected in the image. This is what an image not accepted error looks like

Choose File test2.jpg

Image Not Accepted. Please Try Another

4.2 Image Accepted

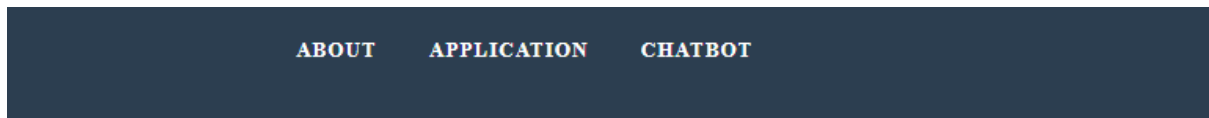
A user's image has been accepted if one face has been detected. This is how an image accepted message looks like

Choose File Me-01.jpg

Image Accepted.

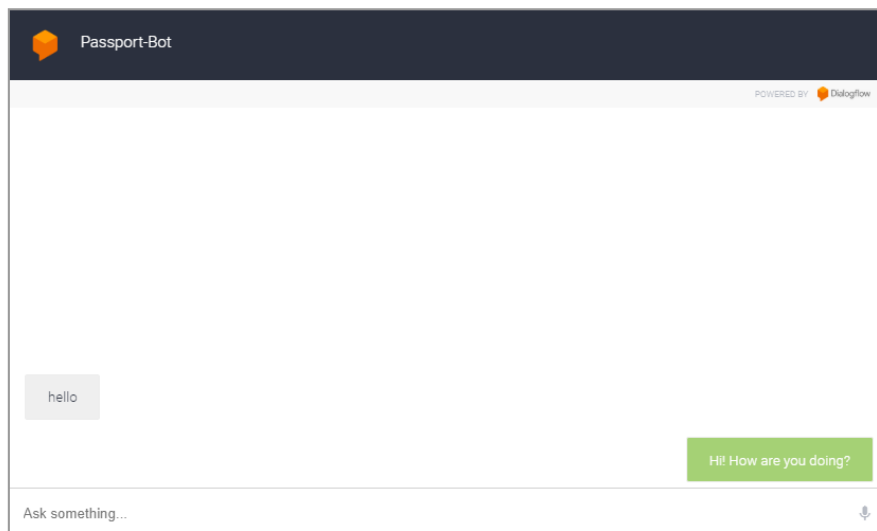
5 Using Chatbot

Included within my web application is a useful chatbot service for users. To use this service the user must navigate to the chatbot section. This can be found in the navigation bar

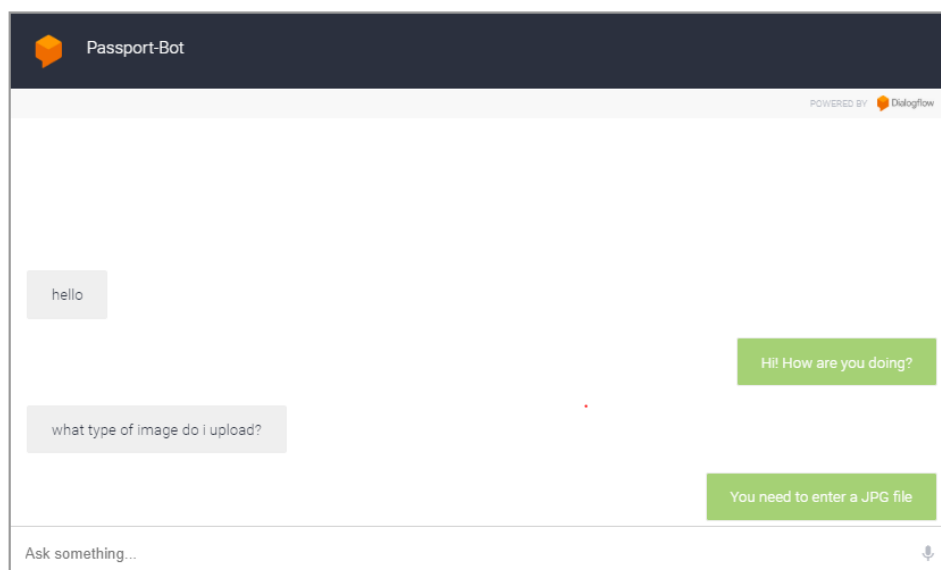


After being redirected to the chatbot page, the user can begin to talk to the chat bot

CHATBOT

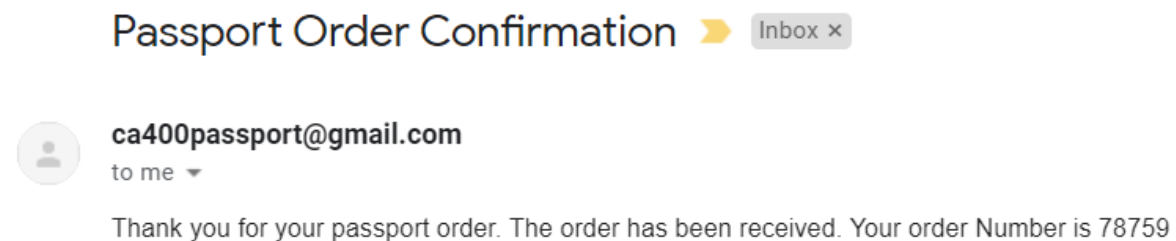


Or if they have questions related to the application process, they can ask those too:



6 Email Confirmation

After the user has completed their application, an email is sent out giving them their order number. I felt this was a nice feature and it also follows a standard set by other businesses. The email confirmation is using Googles SMTP server and a sample email looks like this



7 About Section

The about section was a way to display the different types of technology used to create my final year project. Bootstrap cards are being used to display the data in a more presentable way.

CA400

This Passport Application System was created as part of my final Year Project.

- Frontend: Angular 6
- Backend: Springboot
- Database: Sqlite


Last updated about a week ago

Documents

All documents for this project can be found in my Gitlab repo.

- Function Spec can be found [here](#)
- Technical Manual & User Manual can be found [here](#)
- Blogs for this project can be found [here](#)
- Video Walkthrough for this project can be found [here](#)


Last updated 5 days ago



Springboot

Springboot is a framework used in the development of Web Applications


- This is a Java Framework. I am using Java8
- Maven is being used as my build tool
- REST API's are being used to GET & POST data



Angular6

Angular is a frontend framework. What I used within Angular includes:

- HTML
- CSS
- Bootstrap
- Typescript



Testing

All Of my Tests are available on my Gitlab repo

- Unit/Component Testing using Junit & MockMVC
- Integration Testing using Junit
- UI Testing using Protractor
- User Testing of UI & UX
- REST API testing using Postman during development
- Tests are being run on a gitlab pipeline after a commit