Project Report: TattScan

Joey Tatú  
15015556  
Cloud Platform Programming  
Postgraduate Diploma in Science in Cloud ComputingNational College of Ireland  
joey.tatu@student.ncirl.ie  
16th December 2021

*Abstract*—– a 150-300-word executive summary of the project and the main results

# Introduction

## Background

The inspiration for this project originated from the creation of a vegan health and fitness app. This project's concept was to take a snapshot or image of the eyes and provide a result depending on the sclera's whiteness or yellowness. Against support of the original concept, this application was expanded into an Android app that scans for diabetic retinopathy in the eye. [1] This project has a similar concept where a user can upload a photo of a tattoo and have the application identify it.

## Aims

Google Lens is a comparable image recognition system that can recognise a tattoo but not the pieces that make up a tattoo. Barcodes, QR codes, labels, text, famous buildings, and regions are all recognized by Google Lens. [2] [3]

TattScan works in a similar way to the diabetic retinopathy app. TattScan analyses and identifies a tattoo's primary elements, then develops a hashtag that it uses to search Twitter for comparable tattoos, which it then displays its results to the user. The user can decide if they wish to save the result.

This project has aimed to build a bridge between Google Lens and identifying elements of tattoos via image recognition.

## Technology

TattScan is a web application that has been created using the Amazon Web Services (AWS) Cloud 9 integrated development environment (IDE), primarily using Python code. Possible future iterations will be developed for Android devices.

It was expected that Django would be used as a Python web framework. Comparably, a web site can be hard-coded, but tools are available to simplify this. This is similar to Flask which has been used instead. Flask removes the bulk, hard-coding aspect of a web framework and simplifies creating and deploying a web application. [4]

AWS Simple Storage Service (S3) has been used to store data files (objects) in containers (buckets). [5]

{{DynamoDB}}

# Project Requirements

## Functional Requirements

The functional needs are ranked in this section. These describe the goals of the project.

### Requirement 1: Identify tattoo

This use case describes how a user uploads an image of a tattoo.

#### Description & Priority

This is the focus of this project. This use case explains how a user uploads an image of a tattoo and how the system identifies it. An option then is to allow the user to search the image on Twitter by using a hashtag.

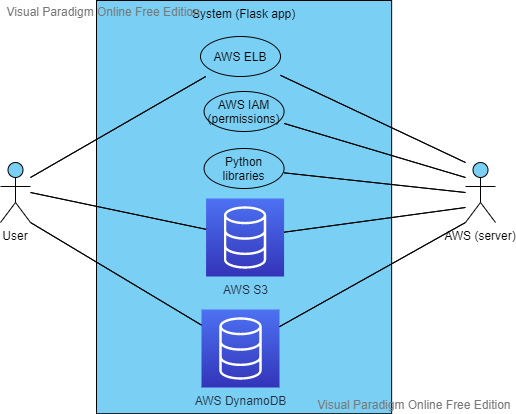
#### Use Case

Unique ID: identify-tattoo

**Scope**

The scope of this use case is for a user to upload an image and for the system to identify it as a tattoo and its main elements.

**Use Case Diagram**



**Flow Description**

**Precondition**

* AWS and its services are active and running correctly.
* The Flask App website is coded correctly and connected to the relevant services (e.g., AWS S3)
* Twitter and its API are active and running.

**Activation**

This use case begins when a user opens the Flask app website.

**Main flow**

1. The user opens the Flask app website.
2. The page loads with options to sign up, login and to upload a file. <See A1, A2>
3. The user selects upload a file

(External: User selects image from their device)

1. Image is stored temporarily in AWS S3
2. The relevant Python library identifies the image
3. The output if what it has identifies is outputted to the user.
4. The User clicks on one of the outputs. <See A3>
5. The System prefixes the output with a hash symbol and adds “tattoo” to the output to create a hashtag. (e.g. #starstattoo)
6. Using the Twitter API via a Python library, the System searches Twitter for images matching the generated hashtag.
7. The Systems then displays the results to the User.
8. The User clicks on an image   
   (External: User is taken to Twitter website)

**Alternative flow**

A1 <User logs in>:

1. The User clicks log in
2. The User enters their username and password
3. The System queries the DynamoDB to check credentials <See E1>
4. The User successfully logged in

<returns to Step 3 in Main flow>

A2 <User signs up>:

1. The User clicks sign up
2. The User enters their username and their password and their password again to confirm
3. The System create a new entry in the DynamoDB and encrypts User’s password.
4. The System logs in the User
5. The User successfully logged in

<returns to Step 3 in Main flow>

A3 <User doesn’t wish to search Twitter>:

1. The User decides not to click on an output.

<Use Case ends>

**Exceptional flow**

E1 < User’s username and/or password are incorrect>

1. The System outputs that the username and/or password are incorrect to the user.

<returns to Step 3 in Alternative flow: A1>

**Termination**

This use case is terminated when the User selects an output to go to the Twitter website.

**Post condition**

The system goes into a wait state.

## Non-functional requirements

### Data Requirements

Encryption is required for all data transmissions. It is not permissible to use raw data for communication from the Flask app website to a service or Twitter API.

Passwords must contain at least one uppercase letter, one lowercase letter, one number, and one special character and must be at least eight characters long. There is a verification in place for this.

### User Requirements

Users should have a basic knowledge of browsing and searching online and uploading files to a website.

### Usability Requirements

The Flask app website will be easy and straightforward to use. Accessibility features have been implemented such as being able to invert colours of the website to make it easier to read, as well as allowing the text to be made larger or smaller to aid those with visual impairments.

# Cloud-based Services

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

# Library Description

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

# Implementation

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

# Integration, Delivery and Deployment

After the text edit has been completed, the paper is ready for the template. Duplicate the template file by using the Save As command, and use the naming convention prescribed by your conference for the name of your paper. In this newly created file, highlight all of the contents and import your prepared text file. You are now ready to style your paper; use the scroll down window on the left of the MS Word Formatting toolbar.

# Conclusions

Possibly talk about identifying issues with healing tattoos.

# References

|  |  |
| --- | --- |
| [1] | J. Tatú, “Automatic Retinopathy Detection Using Digital Image Processing via a Smart Device,” National College of Ireland, Dublin, 2021. |
| [2] | R. Nieva, “Forget rainbow vomit. Google Lens is AR you can actually use,” CNET, 18 May 2017. [Online]. Available: https://www.cnet.com/tech/mobile/google-lens-is-augmented-ar-you-can-actually-use-snapchat-facebook-instagram/. [Accessed 1 December 2021]. |
| [3] | A. Villas-Boas, “Google's new Lens product uses your phone's camera to do clever tricks, like connecting your phone to a WiFi network,” Business Insider, 17 May 2017. [Online]. Available: https://www.businessinsider.com/googles-lens-feature-can-connect-your-phone-to-wifi-using-your-camera-2017-5?op=1&r=US&IR=T. [Accessed 1 December 2021]. |
| [4] | S. Campbell, “Flask vs Django: What’s the Difference Between Flask & Django?,” Guru99, 1 November 2021. [Online]. Available: https://www.guru99.com/flask-vs-django.html. [Accessed 1 December 2021]. |