B.Sc. (Hons) in Software Development

Ollscoil Teicneolaíochta an Atlantaigh

Atlantic Technological University

Profiler

By
Liam Bryant

for
Daniel Craig

May 17, 2023

Minor Dissertation

Department of Computer Science & Applied Physics,
School of Science & Computing,
Atlantic Technological University (ATU), Galway.

Contents

Appendix A

https://github.com/LiamB16/Applied-Project-and-minor-dissertation

A.1 intro

A.1.1 Requirements

For this year's final year project. we are tasked with creating a project throughout 2 semesters. The main idea of this project is to create something that we can show off to potential employers what you are really capable of. In this dissertation, i will be documenting what my project does, the problems i encountered and why i choose this particular project.

This document will also include pictures, diagrams and code snippets to further explain what I'm talking about. The idea is to explain how i created my project and what i have learned from this module.

The important thing to note about the design of this project is that it needs to be 4th year standard. What this means is that we need to above and beyond to what we did in 3rd year so that we can learn something new. For example, if one decides to do a unity game, we can't copy the format from 3rd year because we aren't learning anything new.

Other ideas include making the project in a new language we never done such as Ruby, Go or Flutter. We can also do the projects in languages we are already familiar with like Java, Python or React but if we are going to do this, it needs to be exploring something we didn't covering class.

A.1.2 My Idea

My project idea is called the Profiler which is a facial recognition Security program that is able to identify a person and pull up relative data to that person like name, age and occupation. The Profiler is also able to log the date, time and location of when the individual was last seen as part of an attendance taking system. The data that is pulled is from a MySQL that i created and the only one who can access this information is the admin. The MySQL tables created for this project will contain information that governments will keep track, such as medical records and bank details.

The main use that anyone can get from this project is that it can be used as a security program to identify know individuals. With the attendance taking system, it could save time for universities from taking names down and marking them as "present".

A.1.3 Inspiration for the idea

The inspiration for this idea came from a video game series called Watch Dogs. The series centers around a hacker character who has inside access to the Cen-

tral Operating Systems (Ctos) and now has the ability to exploit it vulnerabilities and using hacking software. These vulnerabilities include changing traffic lights to green and causing an accident, accessing people's personal data, hacking banking account details and causing blackouts. One of these hacking tool called the Profiler, which displays the name, occupation and salary of NPCs (Non Playable Characters) using facial recognition. The image below shows an example of the game's version of the Profiler.

A.1.4 Why its important

The main reason I'm doing this project is to not only show why facial recognition is important, but also show how it could be used for malicious uses if it fell into the wrong hands.

In terms of important uses, the great thing about this project is that it has multiple uses. The whole point of this project isn't just recognising certain people, it idea is to answer the question of "what can i do with this information". The most obvious one is security. The Profiler can be used to identify who shows up at a particular location and if any unknown individuals are in that area. Another use is for attendance taking, colleges could use it for taking student attendance but due to the Data Protection Act, that could be an issue. One other use is for lawsuit cases where employers fired an employee without proper cause. If the argument for the unfair dismissal is that the employee was absent too much, they could use the Profiler's attendance list to prove that employer is right or the employee could use the list to prove that they were in work at the times specified.

While it is interesting to see the uses of this application, I did state that there is the possibility of how it could be used for malicious purposes. All anyone needs to do is change the database and you could pull other data from different places. One example of this could be if the database was changed to ATU's student database. Now you have a Profiler displaying the names, course titles and GPA rating of students. If the Profiler is connected to a government database, the Profiler could

get the PPS numbers of every citizen of Ireland and those numbers could be used for fraud.

A.1.5 Learning Outcomes

In terms of learning outcomes i hope to learn the following.

1. How to implement facial recognition software.
2. Have a deeper understanding of Python
3. Have a deeper understanding of MySQL

A.1.6 success metrics

In order for me to consider this project a success, it must pass the following success metrics.

Reliability

Reliability in this context focuses on how stable is the software and the degree of risk of failure. This means that it once it is running, it must continue to run until the user quits. If at any point the program crashes once it has started or crashes while it is running, this metric will be classed as a "failure".

Security

How likely it is that attackers might breach the system, interrupt it or gain access to sensitive information. While it is known in Cyber Security that nothing is truly security from malicious attacks, there are still steps one can take that can decrease the likely hood of their servers being compromised. If there is a failure to ensure that information stored on the database is secure, this metric will be classed as a "failure".

Maintainability

How easy is the system to debug, troubleshoot, maintain, integrate and extend with new functionality. The one thing to look out for here is putting everything into one Python class. This is considered bad practice and is known as a "God class". If one change can't be made without causing the whole program to return errors, this metric is classed as a "failure".

Performance

How efficient is the code, how optimal is the architecture, is the system able to scale, load times of pages or key functionality. For this metric to be considered a "success", the correct pages must load, the correct data must be retrieved from MySQL and the person being identified must return the correct ID.

A.1.7 Chapters

This section briefly explains the what each chapter contains and how it relates to my project

methodology

This section will cover how i created this project and whether or not my approach was valid.

Technologies Review

This section will go over the technologies used and describe them at a computational level, i.e describing the Python code, Python libraries and MySQL database in detail.

Design

This section will explain exactly how the project works. It will go over how the python code runs and how the database is structured

Evaluation

This section will refer back to the objectives highlighted in the introduction and determine if the objectives are met.

A.1.8 GitHub

https://github.com/LiamB16/Applied-Project-and-minor-dissertation

The link above is the link to the GitHub repository for this project. The repository contains the following.

1. A README that explains how to set up the project on your laptop or desktop.
2. The source code of the project.

3. All related image folders and text files needed for the project.

This is where I will store my project files for the examiners to view and download. Since GitHub keeps track of the date and time of when you submit to the repository, The examiners will look at the commit history as part of the marking scheme. The idea behind this is to show that I'm consistent with my work as this is what's expected in the working world. Employers don't want lazy graduates and this is why examiners are paying attention to the commit history.

A.2 methodology

A.2.1 Waterfall

The waterfall methodology is a project management approach that emphasizes a linear progression from beginning to end of a project. The waterfall methodology is divided up into five sections

1. requirements
2. analysis
3. design
4. implementation
5. Testing
6. Maintenance

A.2.2 Requirements

requirements is what we are expected to produce as 4Th year students. This section of the waterfall methodology depends on the belief that all project requirements can be gathered and understood. Requirements involve cost, risk, dependencies, and success metrics. This phase of the project began on October 29Th and is explained in the introduction of the dissertation.

A.2.3 analysis

analysis, which began on November 8Th, involves . There are two ways that i can conduct an analysis, feasibility study and user stories. The point of this section is to determine whether or not the requirements that i set out for my project are possible to make.

A.2.4 design

designing the project began on November 15Th. This is where a technical solution to the problems set out by the product requirements are created. Given that the project requirements state that we are free to code it in any language and choose what type of project to do, there weren't any limitations to what we could. The only limitation, however, is that it must be of 4Th year standard, meaning that i need to above and beyond to what i did in 3rd year so that i can learn something new.

A.2.5 Implementation

implementation of the project was conducted on November 22ND. This is where i took the design of the project made in the previous week and started the coding part of this project. This is known to be the shortest phase of the project because all the designing and research of this project was conducted. If errors or design changes are constantly popping up, this would mean going back to the design phase of the project to figure out how to solve it

A.2.6 Testing

Testing, also know as the verification phase, was conducted on April 1st. testing is making sure that the project runs correctly with little to no errors. This is done before the project is released to the market because it costs less to fix the bug now than later on. In this case, the market is the college and the cost are marks for it's performance.

A.2.7 Maintainability

The maintainability phase is conducted on November 23rd. The idea of this phase is when a project is released to the market, the developers must listen to the market and fix any bugs reported to them. For this project, i just have to make sure that it is running correctly on the day of the demonstration.

A.2.8 Feasibility Study

Before starting this project, i had to conduct a feasibility study which is an evaluation of whether or not the project idea is practical. This is important because if it is not, i can change project ideas early in the year rather than five or six months down the line and i have n When conducting this study, i had to look into the following.

1. can facial recognition be done on Python
2. Can facial recognition be used for more than just recognising people
3. can facial recognition be used with MySQL

After researching, i have found out the following.

Python can use facial recognition for projects. The facial recognition algorithms are found in a library call "face-recognition". Open-CV is another library that is used to access the web camera of the laptop. This library can also be used to read in images files, which is needed when the project is making comparisons with who is who.

Now that we established that python can be use facial recognition algorithms, we can answer the next question. There have been several projects created that execute tasks if a particular face is recognised. One example of this is a Gmail login where the user can only log into his or her Gmail account if the correct face is shown. Another example includes using facial recognition to unlock a door if it recognises a certain individual.

Now that the two questions above have been answered, the next one can be answered. Python allows you to connect to MySQL and create python files to make queries. Since Python also allows you to create Facial recognition program, you can combine the two by having the facial recognition program return an ID that can be put into a query statement to get the right data.

A.2.9 User Stories

User stories are useful whenever you want to get inside the mind of anyone using this project. We structure the stories as followed.

1. As a (user) I want (feature) so that (reason).

Once we are able to focus on the point of view of the customer, we can develop the features to their specification

A.2.10 Meetings

Every Tuesday, i met with my supervisor Daniel Craig to discuss the project. The idea of these meetings isn't for the supervisor to give me ideas or do my project, but to present my ideas and solutions to them and check if it meets the module's requirements.

A.2.11 feedback

In November, all students did a presentation of their progress so far and what they got done since October. Our lecturer John Healy gave all of us feedback in terms of whats good and bad. The following improvements were suggested to me.

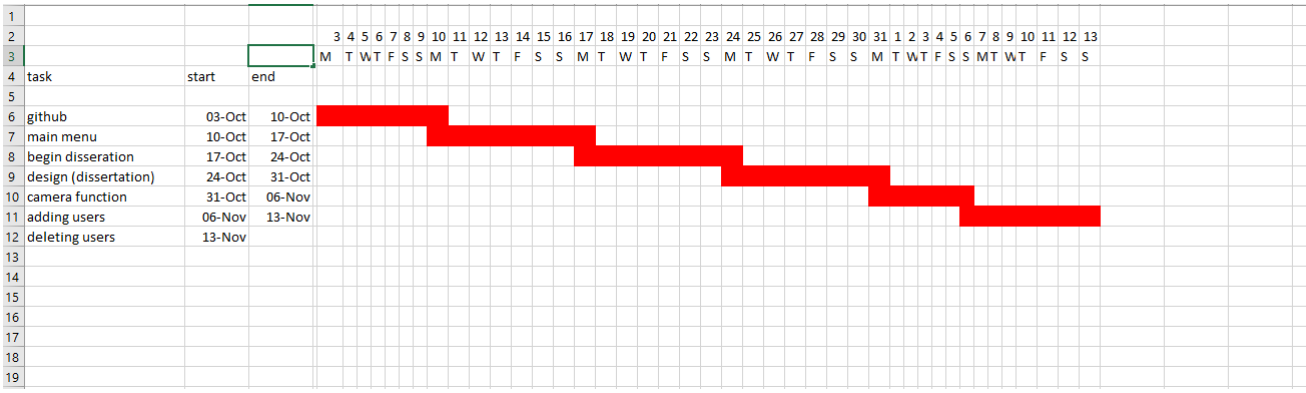
not enough features

It wasn't enough that it was a facial recognition program that recognises people and takes an attendance.

restructure database

The database for the persons table needed to be restructured due to some security issues. The one thing that was pointed out was that the image paths were in plain text on the server. This would be deemed a security issue and this needed to be either changed or gotten rid.

A.2.12 Gantt Chart



This Gantt Chart is used to show my progress from October 12Th to November 15Th. This is able to give a visual representation of my progress.

A.2.13 problems encountered

Throughout this project, there were two major problems that i encountered. While other issues to this project were minor and quickly fixed, this section will focus on these two problems and how i solved them.

database not responding

While obtain data from MySQL wasn't an issue, the problem was writing or deleting information. The queries would execute but the changes made are not seen on the database. The terminal wasn't picking up any errors, so it's likely that something was missing.

After researching, it turns out that when making "insert","delete" or "update" queries to MySQL from Python, you have to follow it up with the command db.commit().This method from MySQL Connector library saves any changes that were made to the database from Python.

Libraries not found

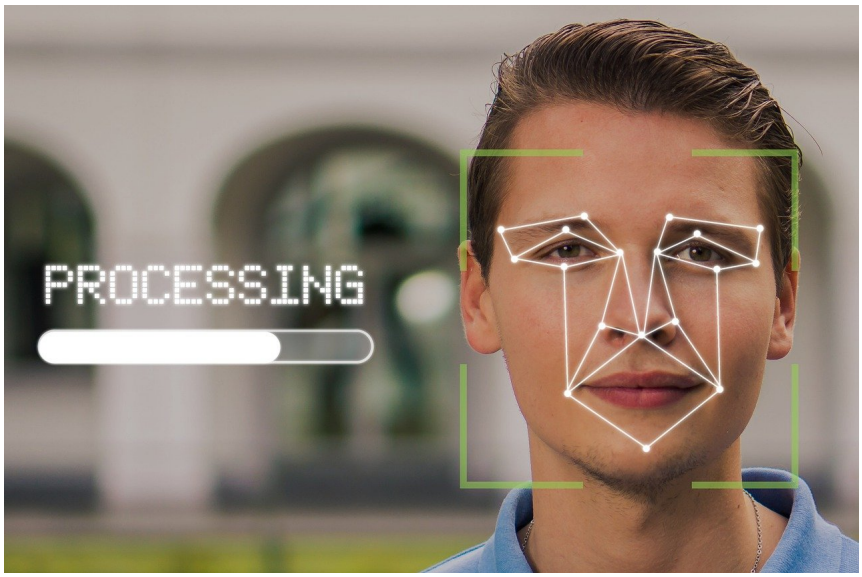
One of the libraries, dlib was difficult to get working. Every other library was imported correctly, including dlib. When i ran the facial recognition feature, the program would crash and say that dlib is not imported, even though it is. This problem was frustrating because it was telling me i had a problem that was fixed, but Python doesn't believe it was fixed.

After addressing this problem with my supervisor, he stated that there were a few students who encountered this same problem when making facial recognition projects. He gave me a specific version of dlib to download and told me to change my Python version to 7. After these change, the facial recognition features were working.

A.3 TechnologyReview

A.3.1 facial recognition

facial recognition works by creating a template" of the target's facial image and compares the template to photographs of preexisting images of a face. These images are usually driving licences, mugshots or student ID cards. The image below (TR-figure 1) shows a man's face being scanned and you notice the white lines measuring the eyes,nose and mouth. This is the template being measured where the software is able to scan the target's face for certain features such as length between eyes, nose size, jawline and mouth. These features are translated into a template with a unique code which will be used to compare with images scanned. The second image below (TR-image 2) is an example of what this binary code is. As mentioned before, the numbers from this binary code must match with the binary code of the image being compared in order for it to be considered a match



(TR-figure 1)

```
  G00377747
● PS C:\Users\Liam\Desktop\Main project> python -u "c:\Users\Liam\Desktop\Main project\Camera2.py"
● [-3.38028371e-02  1.07817218e-01  7.84943551e-02  3.71329673e-02
  -1.10202596e-01  6.75613582e-02 -8.24460238e-02 -4.33206633e-02
   1.98450312e-01 -1.38245538e-01  2.25251347e-01  2.84785796e-02
  -1.96010113e-01 -2.23850943e-02 -4.36720140e-02  1.10514574e-01
  -6.11489937e-02 -7.40451887e-02 -4.92073596e-02 -8.52205902e-02
   1.58115719e-02  1.46437034e-01  2.50257049e-02  3.69005762e-02
  -1.47296205e-01 -3.05231273e-01 -4.57101241e-02 -1.47768423e-01
  -5.36916703e-02 -1.12902954e-01  2.01043170e-02  1.07844740e-01
  -8.95093232e-02 -3.90427932e-02  3.60553302e-02  7.30890632e-02
  -1.56083941e-01  1.72855295e-02  2.02995941e-01  8.38474333e-02
  -7.86805823e-02  2.41252575e-02  4.04151902e-02  3.24348092e-01
   1.20307021e-01  8.38479698e-02  2.20356844e-02 -6.14218339e-02
   1.61905587e-01 -3.13406646e-01  9.29348320e-02  1.25365660e-01
   1.74757183e-01  1.18098095e-01  1.58324748e-01 -1.33179680e-01
   1.83552969e-02  9.40801352e-02 -2.13272735e-01  1.46993548e-01
   3.48865129e-02  2.05254927e-03 -1.54860904e-02 -3.07720639e-02
   2.37968355e-01  1.13483913e-01 -7.56057426e-02 -6.94665536e-02
   1.19837508e-01 -1.63886830e-01 -1.04208522e-01  1.68362513e-01
  -8.59932974e-02 -2.42502093e-01 -1.73876226e-01  6.64816275e-02
   3.62014532e-01  2.23778129e-01 -1.32206798e-01 -5.67838773e-02
  -2.29121372e-03 -6.03206903e-02  9.07181483e-03 -5.17382100e-03
  -7.61302561e-02 -7.97861740e-02 -2.58101486e-02  3.32068466e-03
   2.13896200e-01 -1.27181252e-02 -6.06279112e-02  2.83680350e-01
   6.25528246e-02  1.81497633e-02  7.45234778e-03 -6.69964477e-02
  -7.31954575e-02 -1.93816237e-02 -1.67831793e-01 -6.82389438e-02
  -4.73768935e-02 -5.73449768e-02  1.50528643e-03  3.46395653e-03
  -1.96347401e-01  2.01524824e-01 -4.83316071e-02 -5.46970032e-02
  -2.04270631e-02  5.14518097e-02 -2.36156642e-01  3.04699559e-02
   2.33272567e-01 -2.58881390e-01  1.77522182e-01  8.49289373e-02
   1.07774906e-01  1.60283297e-01  7.69015625e-02  4.05058563e-02
  -1.14151277e-04 -8.44344720e-02 -1.15711391e-01 -1.22682452e-01
```

(TR-figure 2)

Benefits/Uses

Facial recognition is useful in several different ways. The main way is security since identifying the right person is crucial for law enforcement agencies. This can include wanted terrorists who pose a national security threat or find a person who was reported missing. Even some businesses have use facial recognition technology to improve security. Lemon Tree Hotels in India have been able to utilise facial recognition technology to identify guests before they have even stepped into the hotel. This technology can also be used to identify people who were put on a watch list due to their behaviour. Border patrol agencies use facial recognition to scan the passport photo of the person cross the border and allow for a quick, but secure, way of handling border control.

Snap Chat, a social media company, have made use of facial recognition technology. Snap Chat uses filters to identify faces and change their own facial features and appearance with hairstyles, hats, dog ears and other cosmetics. The idea of this was to increase engagement with the app.

controversy

While facial recognition software has it's uses and benefits, that doesn't mean it doesn't have issues. One of the biggest controversies facial recognition has faced is the invasion of privacy. According to security versus privacy [1], the article explains how government overreach with the use of facial recognition technology has caused many to question whether or not people have the right to privacy. In the United States, after 9/11, facial recognition technology was used at the 2001 Super Bowl. At the time, the public didn't see a problem with this because they believed that this kind of technology could've prevented 9/11 or another terror attack from happening. A poll conducted after 9/11 [1] found that 86 per cent of those surveyed believed that "facial recognition should be used to scan for terrorist threats in locations like airports". 63 per cent of those surveyed also believed that "facial recognition software cameras should be expanded to streets and other public places like banks". As the years went on, support for more facial recognition software to be implement went down as more and more Americans questioned whether or it not it's worth trading freedom for privacy for security, especially after Edward Snowden, a former NSA, exposed the NSA for spying on Americans through laptop cameras and mobile phones.

Another country where facial recognition is used is China. Not only does many of China's public institutions use facial recognition software, but private enterprises as well [2]. China uses facial recognition in it's digital payments, hospital waiting rooms, housing complexes, transportation systems, and urban policing. All of this is used as part of China's social credit score. This is an initiate that measures individuals, businesses, social organizations, and government agencies are assessed based on their "trustworthiness." [3]. Any Chinese citizen who was deemed "untrustworthy" or a "traitor to the state" suffered penalties like getting locked out of their bank accounts or unable to use public transport.

A.3.2 GitHub

According to it's own website. GitHub is a code hosting platform for version control and collaboration. It lets you and others work together on projects from anywhere. GitHub has over one hundred million developers signed up to it, four million organisations signed up and over three hundred and thirty repositories created.

repository

Once you have created an account with GitHub, you can create a new repository. A GitHub repository is where your project is stored. repositories can contain

source code, image files, APIs, video demos and excel spreadsheets. You have the option to make the repository public, which means everyone can view your files and download them, or private, where only you can view and download your repository. Repositories also contain README files. These are often use by developers to explain what their project does and how to set it up once it is downloaded.

Branches are another element of repositories that lets you have different versions of the same repository at one time. Your repository has one branch named "main" that is considered to be the default branch. This feature of GitHub is useful when you want to change the project without changing the main source code found on the main branch.

push code

When you have a project you want to publish or "commit" on GitHub, You first go into the command line and change the directory to the project folder. You then enter the following commands.

1. "git init" initialises the repository to be pushed to the remote repository.
2. "git add ." is used to bring all file to the commit stage. you can name specific files like "git add example.py" to be push or you can add a dot to the end of the statement to select all files.
3. git commit -m (commit name) takes everything from the staging area and makes a permanent snapshot of the current state of your repository. The commit name is what will appear beside each file name on the repository.
4. git remote add origin (Link to GitHub repository) validates the existence of the local repository.
5. git branch -m main sets the branch that the changes sent to. If no branch is specified, it will go to the default branch, in this case, the main branch.
6. git push -u origin main uploads all new files created and changes made to existing ones to GitHub.

These commands will upload your code to GitHub for you and everyone to see.

pull requests

Git pull is used for making changes to existing projects. When the developer has finished making changes to the project, they need to use the "git pull (repository name)" command to merge the old repository with the new changes made. If you are collaborating with others in a group project, it is a command that you will run frequently

clone

Git clone can be used to clone your repository, as well as other developers' repositories. All you have to do is open a command terminal and use the command "git clone (repository link)". This will download all the resources on the repository and create a new folder with the same name as the project to put them in. Now, you can edit the cloned project to how you see fit. This can also be used to restart a new project if the developer made a mistake changing the project.

commit history

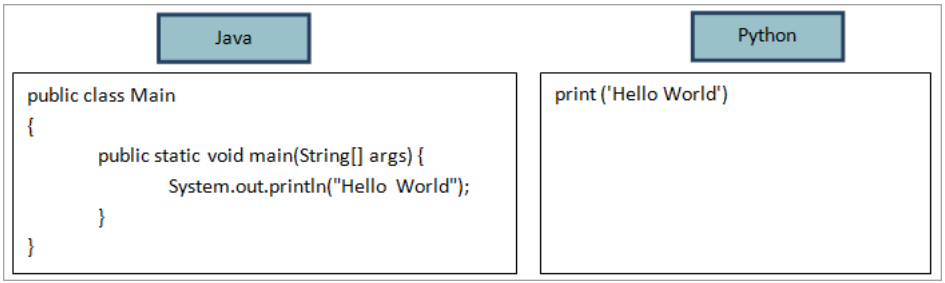
GitHub is able to keep track of the number of commits a developer makes to a repository. Commits can be thought of as snapshots or milestones along the timeline of a Git project. This can help developers show who published what and when. This can also be used in college as a way for students to show their lecturers that they are consistent with their work and that they have the proof to show them.

A.3.3 Python

Python is the main programming language that i will be using to code this project. Python was created in the late 1980s and its implementation was started in December 1989 by Guido van Rossum. We will talk about the advantages and disadvantages of the language and how it will potentially affect the project.

advantages

Python is faster to type out. this is demonstrated below where both java and python are printing out the phrase "hello world" but you notice how python only took one line of code just to print out the phrase while java took six lines of code to print out the phrase. The reason Python is faster is because Python is dynamically typed. This means variables don't have to be known at run time. This means that variables created don't have to be specified as an integer, double or string. In statically typed languages like Java variables must be declared as an integer or string before assigning a value.



(TR-figure 3)

Python is also doesn't have a compile process. This is because Python is an interpreted language where code can execute as soon as it's written. There is no separate compiler and no time-consuming compilation steps like the ones you would find in Java.

Python is open source. This means that it is free to download and use by everyone. All Python releases are available for free under an open source license. Python can even be modified and re-distributed at no cost. This greatly reduces development costs.

Python has a large popularity with developers. This is important for development because that means there are a ton of resources on YouTube and Stack Overflow. Development of the Profiler can be sped up if someone else has already found a solution to a problem i am having. All I would need to do is use the video or web page as a reference and modify the code to fit the requirements of the project.

Rapid development is another advantage of Python. As mentioned before, Python does not have to be compiled, programs take less time to develop

disadvantages

Python, compared to java or c++, is slower at executing programs due to two reasons.

1. Python is a interpreted language which means it only produces a result of the code while java and c++ are compiled languages which produces a program written in assembly language.

2. As mentioned in the advantages section python is faster to typed because it's dynamically typed, but because the variables aren't assigned values, the program runs slower because the type information for each variable has to be retrieved at run time. Since java is statically typed, all variables created are assigned as integer or string so when the code runs, the type checking has already be done.

Python can also impact on RAM because it is memory intensive. This is due to the Python garbage collector cannot gather all discarded resources immediately, which reduces the amount of available memory.

Python doesn't allow for mobile development due to the amount of RAM it uses. There are some Python development tools for mobile apps, but they are more limited than frameworks for other languages.

Python, compared to Java, isn't suited for working with databases. The reason for this is because Python lacks a powerful, high quality, easy-to-use interface like the Java Database Connectivity (JDBC). While this isn't ideal for my project because MySQL is being used along with Python, I can avoid this issue as long as the interactions between the Python program and MySQL aren't too complex.

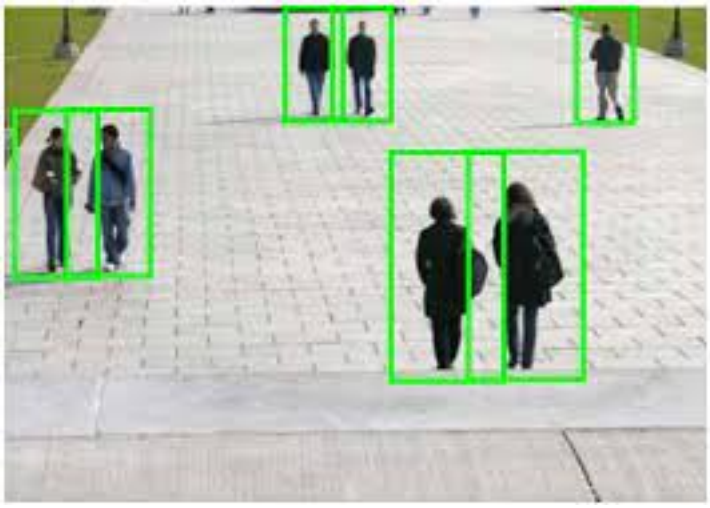
A.3.4 Python Libraries

Open-CV

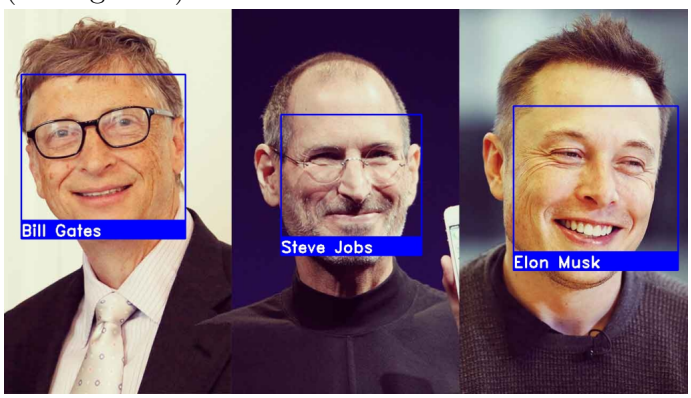
Open-CV is a library that is used for image processing and video capture [4]. The library contains +2500 algorithms that allows us to perform tasks such as finding faces, track moving objects and identify hand gestures. This library is available for Python as well as Java and C++ and is supported by Windows, Linux, Android and Mac OS.

According to Open-CV.org, the library is used by companies such as Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda, and Toyota and it is also used by start-up companies such as Applied Minds, VideoSurf, and Zeitera. Open-CV is also used by many countries such as Israel by detecting intrusions in surveillance video, checking runways for debris in Turkey and inspecting product labels in factories in Japan.

Open-CV contains functions such as rectangle() which will put a rectangle around the object the program is designed to find, such as faces, eyes or cars. The arguments it takes are color of the rectangle and the thickness of the rectangle. The image below (TK-figure 4) shows us an example of this feature in action where the program is designed to find people and draw a green square around them. Another Open-CV function is putText() which allows you to put text around the rectangle. The arguments it takes are image, text, position of text, font, font colour and font thickness. TR-figure 5 shows us the names of the people in the photo shown below the blue rectangle.



(TR-figure 4)



(TR-figure 5)

face-recognition

face-recognition is the library we to identify people. As mention in my research notes in this section, According to the documentation for this library, it is "Built using dlib's state-of-the-art face recognition" and "the model has an accuracy of 99.38 per cent Labeled Faces in the Wild benchmark" .we convert facial features into binary code and compare the two images processed to see if they're the same.

There are three functions within this library to focus on. The first one is the load-image-file(). This takes in the image file that will be examine with argument being the name of the file or the path of the file.

The next Facial recognition function to look at is the encoding() function. The image previously obtained is put into the encoding() function and it returns the 128-dimension face encoding for each face in the image. This is what is being compared with when put up against a photo that could potentially be a match.

The last function to look at is the compare-face() function. This takes two sets of images that have been encoded and compares the two to see if they are a match. The result returned will either be true or false where true means that they are a match and the person has been identified.

MySQL.Connector

MySQL.Connector is a library that allows the user to access the MySQL database using python code. We are able to create functions that send a MySQL query to the database and return a result. Once the library is imported you can start communicating with the MySQL server. MySQL.Connector contains methods such as connect() which allows the user to establish a connection between the python code and the server. The arguments inside connect() are the name of the host of the server, username of the server, password and the database to use.

The execute() function allows users to query the SQL database by putting the query in quotes in the function. Another way you can execute queries with the execute() function is to create a variable called "query" and another variable called "val". This is used for queries that have conditions or getting variables from the user to be inserted into the database. For queries that involve deleting, inserting and updating, the execute() function is followed up with the commit() to save the changes made to the databases.

Tkinter

The Tkinter module ("Tk interface") is the standard Python interface to the Tk GUI toolkit from Scriptics (formerly developed by Sun Labs) [5]. Tkinter is a set of wrappers that implement the Tk widgets, which are instances of classes that represent buttons and frames.

Tkinter uses functions various functions to create customised pages. The label() function lets you create text that can be displayed to the screen. This is used for titles and labelling what each input field is for. The entry() function creates an input field for the user to insert data. This data can be assigned to variables which are used in the functions created by the developer. Grid() functions maps the position by rows and columns. When the row argument increases by one, it's pushed down the page and when the column argument increases by one, it moves to the right of the page.

Pack() is then used at the end of labels(), entry() and grid() to show everything that is created in Tkinter.

A.3.5 MySQL

MySQL database is relational database management system (RDBMS) developed by Oracle that is based on structured query language (SQL)[6]. An SQL database is require fixed table schemas, usually avoids join operations, typically scales horizontally[6]. The opposite of this are NoSQL databases, which are databases system which is distributed, may not require fixed table schemas, usually avoids join operations, typically scales horizontally, does not expose a SQL interface and may be open source [7]. As previously done for python, we will talk about the advantages and disadvantages of this SQL and how it will affect the project. in some of these comparisons, we will also compare it with NoSQL

advantages

MySQL has the advantage of scalability [6]. MySQL supports vertical scalability while NoSQL supports horizontal scalability. Horizontal scalability is when you increase the capacity by connecting the multiple hardware or software entities so that they can work together as a single unit [8]. Vertical scalability refers to scaling up which is the ability to increase the capacity of existing hardware or software by adding more resources to it [8]. This means that for MySQL, one individual server can increase it's power by adding more resources like random access memory (RAM). This will prevent the server from slowing down due to too much traffic from coming in

System Maturity is another advantage of MySQL [6]. SQL has been around for a long time, which means that all issues that were previously discovered after it was launched have now been mostly patched. This issues mostly relate to security such as authentication, data confidentiality integrity. NoSQL doesn't have these issues patched and this could cause the security metric of the project to fail.

MySQL, like Python, is free to download from the MySQL website. This reduces the cost associated with developing this project and I can design the database to my specifications.

disadvantages

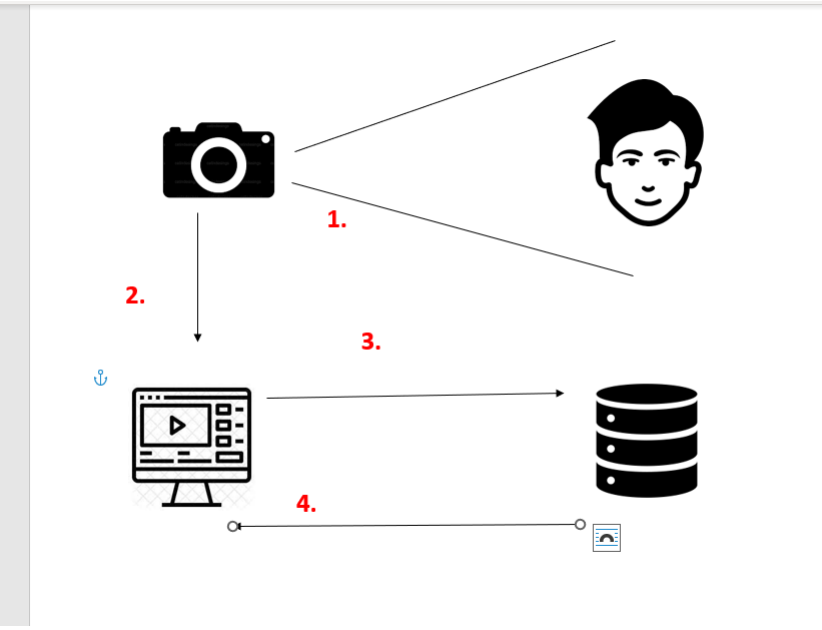
Data Retrieval is slower in MySQL than in NoSQL [6]. In MySQL, tables are linked together and when looking up data from a different table, the user has to use JOIN statements which create Views. This ends up being time consuming.

MySQL is not suitable for handling large databases. The reason for this is because MySQL was originally designed as a single-node system, where Now given the fact that the project i am creating doesn't require a large data set, this should interfere with the performance of the

MySQL is vulnerable to a denial of service attack or DDOS for short. A DDOS attack is when a hacker uses botnets, which are machines controlled by the hacker, to send data request to the server. While a data request from one machine is fine, thousands and thousands of machine all querying the same database in one day can cause the server to shut down due to the volume of requests and being unable to answer them all.

A.4 SystemDesign

A.4.1 UML diagram



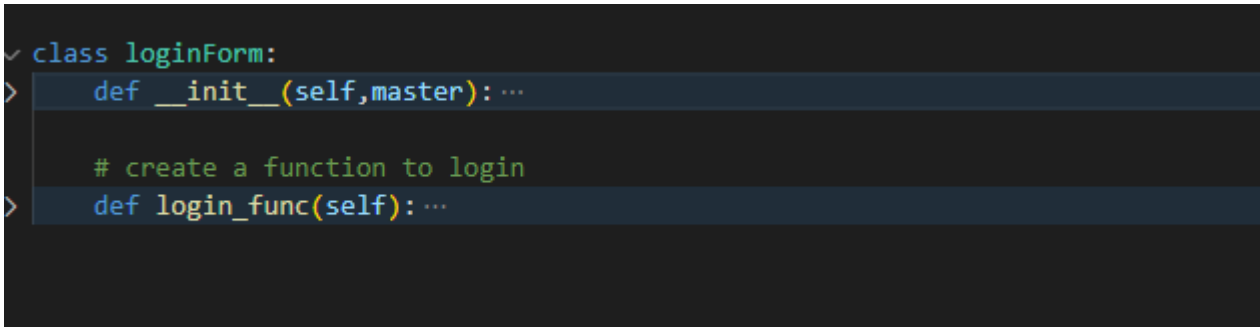
A.4.2 Tkinter

Using the Tkinter library, we are able to create a UI menu for the admin to use and prompt the user for input. The way the Tkinter Menus are set up set up consist of two elements.

1. The function containing the methods.

2. The GUI setup in another function.

Every menu is set up in the same manner and to show exactly what i mean, we will explore this further by looking at one of these menus. In the image below, we examine the login page the user comes across when they first open the profiler. We can see that in the class login Form contains two methods. Login-func() will be explained later on but for now we will focus on the init() function.



The next image below shows us the init() function expanded. We can see that by using the Tkinter library imported and initialising it as TKO for short, we can use the methods to create labels, buttons and message notifications for this project. TK.label() is used to create text fields to show the user what each input box is used for. For example Self.usernamelabel() is used to create the text field for user ID. We can also change the font, colour of the text and text size in the parameters of Tk.label().

Tk.Entry() is used to generate the input field where the user can put in information like ID number or password. This entry method is then used to be inserted in functions as part of the arguments required to perform a task. Inside the Tk.Entry(), you can set the width of the input box, and the border width.

Tk.Button() is also create to activate a function created in the program. As previous to the other two method, you can set the text, font, color, font colour and background colour. The one difference with this function is that you can set the function it activates in the command argument. In this context, we have two buttons, one for login and one to cancel the login. Self.btnlogin() has the command "self.login-func()" which attempts to log in the user based on the details provide. Self.btnCancel() has the command "close window" which will cancel the login process and close the window.

```
# start create widgets
self.frame = tk.Frame(self.master)
self.frame.place(x=200, y=70)

self.btnsFrame = tk.Frame(self.frame, bg='#fff', padx=40, pady=15)

self.windowTitle = tk.Label(self.frame, text='Welcome', bg='#fff',
                            fg='blue', font=('Tahoma',20), pady=30)

self.usernameLabel = tk.Label(self.frame, text='ID:', bg='#fff',
                              font=('Verdana',16))
self.usernameTextbox = tk.Entry(self.frame, font=('Verdana',12), width=25,
                                borderwidth='2', relief='ridge')

self.passwordLabel = tk.Label(self.frame, text='Password:', bg='#fff',
                              font=('Verdana',16))
self.passwordTextbox = tk.Entry(self.frame,show='*', font=('Verdana',12),
                                width=25, borderwidth='2', relief='ridge')

self.btnLogin = tk.Button(self.btnsFrame, text='Login', bg='green',
                          font=('Verdana',12), fg='#fff', padx=25,
                          pady=10, command=self.login_func)
self.btnCancel = tk.Button(self.btnsFrame, text='Cancel', bg='orange',
                           font=('Verdana',12), fg='#fff', padx=25,
                           pady=10, command=close_window)
# end create widgets

# start place widgets
```

The last image to show for the Tkinter setup is to show how each of the Tkinter labels, buttons and input fields are placed on the screen. Grid() functions maps the position by rows and columns. When the row argument increases by one, it's pushed down the page and when the column argument increases by one, it moves to the right of the page. Pack() is then used to show everything that was created in the init() function.

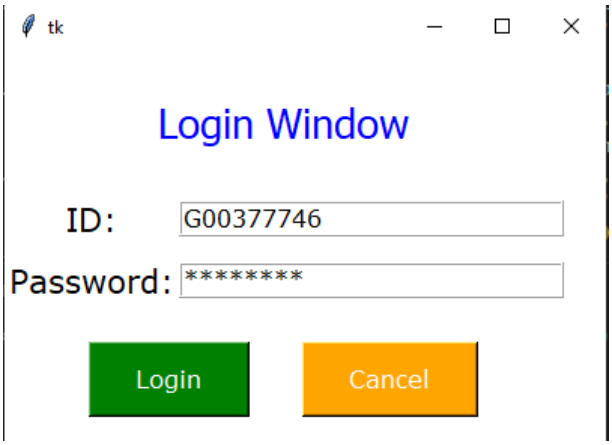
```
# start place widgets

self.frame.pack()
self.windowTitle.grid(row=0, column=0, columnspan=2)
self.usernameLabel.grid(row=1, column=0)
self.usernameTextbox.grid(row=1, column=1)
self.passwordLabel.grid(row=2, column=0, pady=(10,0))
self.passwordTextbox.grid(row=2, column=1, pady=(10,0))
self.btnsFrame.grid(row=3, column=0, columnspan=2, pady=10)
self.btnLogin.grid(row=0, column=0, padx=(0,35))
self.btnCancel.grid(row=0, column=1)
# end place widgets
```

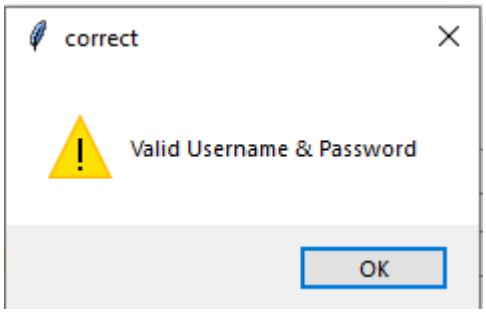
A.4.3 Python

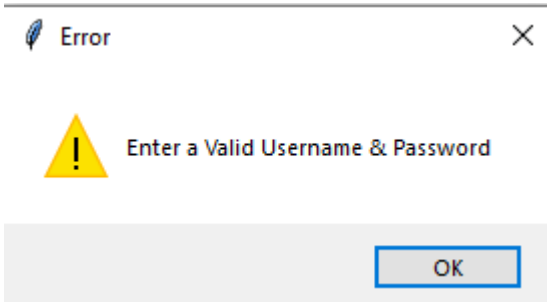
login

The first page the user will come across is the login page. As mentioned before, certain individuals are assigned the role of "admin" which means that only they can log in and use the profiler. Anyone who isn't an admin cannot log in. In

the pictures below show us an early version of this login page. You are prompted for an ID (G00 number) and a password. If the password is correct, you get the response on the left saying that it's valid. You get any of these wrong and you get the response on the right that says the details aren't valid.







Main Menu

Once the admin has logged into the profiler, they are now present with a new menu with new options. These options include

Profile Mode

Profile mode is when a person is identified based on photographs. The admin can use images of people stored on his/her computer and copy and paste the path of the photo to the input box. In a file called "images", photos of the everyone on the database is stored in this folder and used as a reference to identify the person being analysed. Both sets of photos, the one submitted by the admin and ones stored in the images folder, are converted into binary data. If the binary data of the submitted photo matches with any of the photos in the images folder, the submitted photo is flagged as being a match and the admin has the option to view

more about this person or cancel that operation. If the submitted photo's binary data isn't a match, then admin is informed that no matches have been found

Monitor Mode

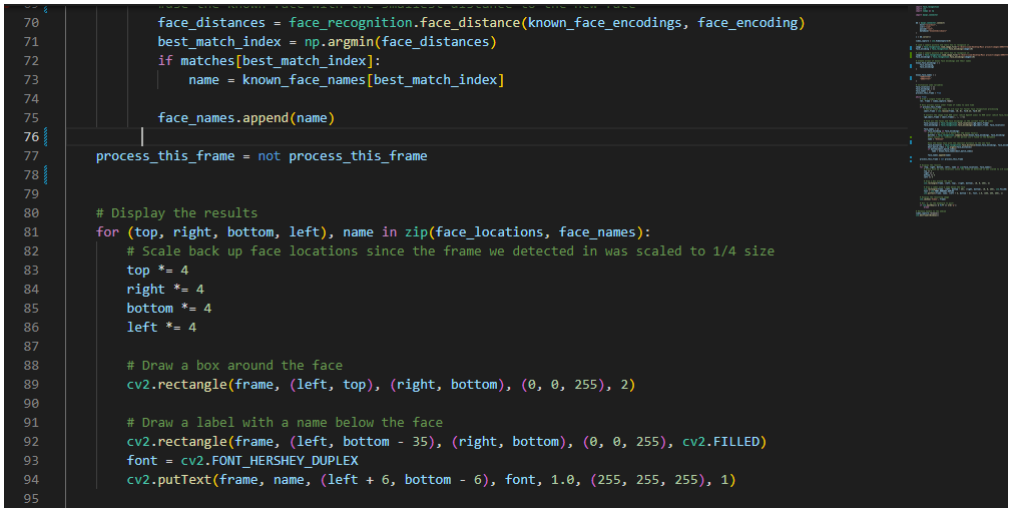
Monitor Mode is used to identify the person using the camera on the laptop or desktop. As explained in the profile mode, monitor mode takes in an image and uses the facial recognition libraries to convert the image into binary data. In the image MM-figure 1, we can see on line 48, the video camera is turned on. From line 51 to 62, Monitor mode sets up the frame of the camera and uses face-location() function to locate a face in view of the webcam.

from line 63 to 75, the compared-face() function is used to compare the two sets of binary data, the one stored in the image files and the one submitted to the profiler. A variable called "name" is assign the value of "unknown". This is the variable that goes into the putText() function where the name of the person is displayed. If a person who is not found on the database is scanned, they are marked as "unknown". On line 72, if the face encodings submitted are a match to the ones found in the image photo, the name variable is changed to the name of the person identified.

```
46  while True:
47      # Grab a single frame of video
48      ret, frame = video_capture.read()
49
50      # Only process every other frame of video to save time
51      if process_this_frame:
52          # Resize frame of video to 1/4 size for faster face recognition processing
53          small_frame = cv2.resize(frame, (0, 0), fx=0.25, fy=0.25)
54
55          # Convert the image from BGR color (which OpenCV uses) to RGB color (which face_recognition uses)
56          rgb_small_frame = small_frame[:, :, ::-1]
57
58          # Find all the faces and face encodings in the current frame of video
59          face_locations = face_recognition.face_locations(rgb_small_frame)
60          face_encodings = face_recognition.face_encodings(rgb_small_frame, face_locations)
61
62          face_names = []
63          for face_encoding in face_encodings:
64              # See if the face is a match for the known face(s)
65              matches = face_recognition.compare_faces(known_face_encodings, face_encoding)
66              #sets values to 'unknown' if the person isn't found in the database
67              name = "Unknown"
68
69              #use the known face with the smallest distance to the new face
70              face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
71              best_match_index = np.argmin(face_distances)
72              if matches[best_match_index]:
```

(MM-figure 1)

In MM-figure 2, from lines 81 to 94, the rectangle is drawn around the person's face. The rectangle() function creates the rectangle to surround the face with the camera feed as the image argument and the putText() function takes in the name variable as the text argument and displays the name of person below the rectangle or display "unknown".



```
            face_distances = face_recognition.face_distance(known_face_encodings, face_encoding)
            best_match_index = np.argmin(face_distances)
            if matches[best_match_index]:
                name = known_face_names[best_match_index]

            face_names.append(name)

    process_this_frame = not process_this_frame


    # Display the results
    for (top, right, bottom, left), name in zip(face_locations, face_names):
        # Scale back up face locations since the frame we detected in was scaled to 1/4 size
        top *= 4
        right *= 4
        bottom *= 4
        left *= 4

        # Draw a box around the face
        cv2.rectangle(frame, (left, top), (right, bottom), (0, 0, 255), 2)

        # Draw a label with a name below the face
        cv2.rectangle(frame, (left, bottom - 35), (right, bottom), (0, 0, 255), cv2.FILLED)
        font = cv2.FONT_HERSHEY_DUPLEX
        cv2.putText(frame, name, (left + 6, bottom - 6), font, 1.0, (255, 255, 255), 1)
```

(MM-figure 2)

Attendance Taking

The attendance taking feature uses the same functions as the ones discussed in monitor mode because it uses the laptop's camera. The Only differences are shown in the images below. In AT-figure 1, the first two lines creates the variables "current date" and "now". current date is formatted as "DD-MM-YYYY" where D is day, M is month and y is year. The format should come out as 06/03/2000. "Now" is a variable that is set to the value of the current date using the datetime.now() function imported from date time library. The last two lines then create a file called "current date" and begins writing to it.

```
# gets cuurent date
now = datetime.now()
current_date = now.strftime("%D-%M-%Y")

# opens and creates file
f = open("C:/Users/Liam/Desktop/Main project/current_date.csv",'w+',newline= '')
lnwriter = csv.writer(f)
```

(AT-figure 1)

```
if name in known_faces_names:
    if name in students:
        students.remove(name)
        print(students)
        #records hour minutes and seconds person was seen
        current_time = now.strftime("%H-%M-%S")
        #writes time to csv fil
        lnwriter.writerow([name, current_time])
```

(AT-figure 2)

In AT-figure 2, this section of the code explains what happens when the person is correctly identified. If the person is known by the profiler, it takes the name of the person and creates a variable called current time and uses the now.shiftime(HH-MM-SS) where H is hours, M is minutes and S is seconds. Both the name, time and date are written to the file.

User data menu

Once the person is identified, the admin is taken to a new menu where they can view the information of the person identified. In UD-figure 1, we can see a code snippet of the code that displays the person's information in the display() function. the first line uses c.execute to get all the data that relates to a specific ID, in this case "G00377746". One the data is found, the data is formatted into an entry box and each entry box is created based on the number of rows in the table. The next lines after the e.insert() function creates a button using the Tk.button(). This button is the delete button that will delete the information from the person's table.

```
c = db.cursor()

def display():
    c.execute("select * from medical_condition where ID = 'G00377746';")

    i=0
    for student in c:
        for j in range(len(student)):
            e = Entry(my_w, width=10, fg='blue')
            e.grid(row=i, column=j)
            e.insert(END, student[j])
        # show the delete button
            e = tk.Button(my_w,width=5,text='Del',fg='red',relief='ridge',
            anchor="w",command=lambda k=student[0]:del_data(k))
            e.grid(row=i, column=8)
        i=i+1
display()
```

(UD-figure 1)

```
def del_data(s_id): # delete record
    try:
        my_var=msg.askyesnocancel("Delete Record",
          "Are you sure ? ",icon='warning')
        print(my_var)
        if my_var:
            query="DELETE FROM medical_condition WHERE id=%s"
            my_data=[s_id]
            c.execute(query,my_data)
            db.commit(); #commits changes to database
            print("Row Deleted  ")

            for row in my_w.grid_slaves():# remove widgets
                row.grid_forget()
            display() # refresh the list
    except SQLAlchemyError as e:
        error = str(e.__dict__['orig'])
        print(error)

my_w.mainloop()
```

(UD-figure 2)

UD-figure 2 shows the delete method to be used as an argument for the command section in Tk.button(). This function will delete the information based on the current user identified and will refresh the page to shown the new results, with the deleted data gone.

Adding new users/data

```
def insertData():
    ID = entry_ID.get()
    firstname = entry_firstname.get()
    Lastname = entry_Lastname.get()
    age = entry_age.get()
    occupation = entry_occupation.get()
    admin = 'N'
    password = ''
    insert_query = "INSERT INTO persons(`ID`, `name`,`sur-name`, `age`, `occupation`, `isadmin`, admin_passwo
    vals = (ID, firstname, Lastname, age, occupation, admin, password)
    c.execute(insert_query, vals)
    db.commit()
    messagebox.showwarning('New User Added')
    close_window()

button_insert = tk.Button(root, text="insert", font=('verdana',14), bg='orange', command=insertData)
btnCancel = tk.Button(root, text='Cancel', bg='orange', font=('Verdana',12), fg='#fff', padx=25, pady=10, com
```

(ANU-figure 1)

Admins can also add new people or add new details to existing profiles. In ANU-figure 1, the function insertData() is created. This function contains get()

methods from the Tkinter variables previously created. The information from the get() function is then put into the SQL query as the values to be inserted into the database. Once the data is inserted into the input boxes, the Insert method is put into the Insert button as the command argument. When the button is pressed, the data is inserted and can be viewed in the Profiler.

A.4.4 MySQL

MySQL tables

For the MySQL part, the database is divided into the following tables, assets, bank, current id, daily activity, education, medical record and persons. The reason for these tables is because as mentioned in the introduction, the SQL table is designed in the data a government would keep on people as opposed to a university or hospital. The primary key is set to ID because it means that pulling the correct data is easier than using first or last name. There could be a scenario that involves two individuals with the same first and last name and the application won't know which data set to pull up so it's easier to identify each user as G00 numbers.

Persons

Persons is the table that contains the list of names for the individuals stored on the database. This table contains the first name, last name, age and occupation for each ID. Is admin and admin password is also important because it lets the programme know who has authorization to log onto the Profiler. Is admin is set to an ENUM type meaning that the only options to choose from is "Y" or "N" where Y means that the person has authorization to log on. Admin password is also used to verify the user and for security purposes, it's encrypted. This is standard practice for Businesses with servers to encrypt passwords in the event that a hacker gains access to the server and gets the password in plain text. With the encrypted password, it's useless to anyone who doesn't have the key or encryption algorithm to decrypt it.

Daily Activity

Daily activity is used to log any changes made to the database. This is done for security reasons to ensure that the client knows what changes are made and who it was done by. The columns contain the ID of the admin, what activities they did, the date and time that activity took place. Activities can include logging on, logging off, adding new user, deleting user, adding information and deleting information. In the example below, we can see G00377746 has logged on and we can also see the date and time they logged on.

Current ID

Current ID only contains one row and that's ID and the reason for this is simple. In order to get the correct information and not waste time, when an individual has been identified by profile mode, the ID is written to the current ID. Now, when the admin is viewing the user's information, rather than go back to profile mode, indemnify the user and pull specific records, the process is sped up. Now the individual has to only be identified once, have his id written to current ID and current ID is used to pull the correct information. When a new individual is identified by profile mode, the previous id is removed from the table and the new one is added. We can see this in the code snippet below. If the result returned is "true" i.e. the person is found on the database, c.execute() will delete everything from current ID before adding the new ID to the current ID table.

Education

education is the table that contains the educational background of everyone on the database. The school row shows the name of the school they attended and ed level row, short for education level, tells us what type of school it was. Examples of types of schools can include primary school, secondary school, university, community college, etc. Award row shows us what they earned from studying at that school and year of graduation row shows us the year the person graduated from that school. If these two rows are blank, but school and ed level rows are not, this means that the person attended the school, but didn't graduate. In the example below, we can see that G00377746 earned a diploma from St Joseph's secondary school in 2019.

Medical Condition

Medical condition table is used to store information relating to the medical conditions of everyone on the database. The condition of what they are suffering from is listed along with the medication they have been prescribed with.

Criminal Record

Criminal record shows the crimes of everyone on the database. The one thing to point out about this table is that the only admin who is allowed to make changes to it are the ones whose occupations are 'garda". The reason for this is because anything relating to criminal data should be handled by the garda in case false data can ruin employment chances for anyone on this database. Another important use for this table is determining who should be assigned as admin. If an admin decides to assign an individual the role of admin and the person in question has a

criminal record, they can't be assigned as admin due to them having a history of being "irresponsible".

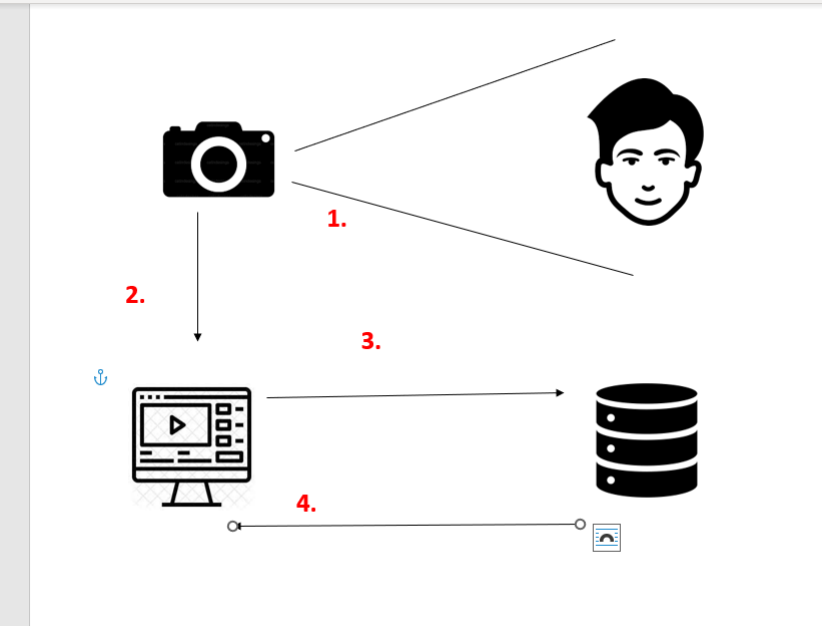
Assets

Assets is a table that records all the financial assets each person on the database possesses. The name of the asset is listed, as well as the value of it and the asset ID. Asset ID can include model number or car registration.

Bank

Bank is used to keep track of details relating to the bank accounts of each user. In this table, the following are logged. The name of the bank such as Bank of Ireland, AIB, etc, the bank ID of each person and the total amount in their account.

A.4.5 UML diagram



The Unified Modeling Language (UML) for the project is displayed above. Each numbered part represents how the program communicates with each component of the Profiler.

1

The first part is the laptop camera capturing the faces of people in view of the frame. These faces can either be people known and record into the database or

it could be people not recorded and known. Once the camera has found the face in it's sights, it will draw a rectangle around the person's face and feeds the data back into the Profiler code.

2

The camera then feeds the information back into the the computer and into the Profiler program. The face is then converted to Binary code and compared with the other images to use as a reference to see if the person is a match.

3

Once the match has been found, if the user decides to view more about this person, the Profiler connects to the MySQL database and queries the database for data relating to the ID of the person. The Profiler can also send queries to add a new person to the SQL database or remove a person from the database.

4

The MySQL database returns the data that the user is seeking and can be view on their computer to see or returns a statement to say if any changes have been made to the databases.

A.5 evaluation

A.5.1 robust

Based on the unit and regression tests conducted, I have come to the following conclusions

Unit Testing

Upon conducting the unit test, it was mostly a success with several issues. The first one being with the profile identifying the wrong person, The profile mode will identify Liam Bryant as "G0037746" correctly. The issue arose when trying to identify Elon Musk, whose ID is "G00377747". It would return Elon Musk as "G0037746" and think it's Liam Bryant.

Another issue that was found in the unit test is that in monitor mode, the name of the person identified is displayed, but not the age and occupation. The error return states that it is an overload resolution failed. this means that the age and occupation data return from MySQL Can't convert object to 'str' for 'text'.

The last error came from the attendance feature. The issue is that the file "current attendance" is unable to be found. The reason this issue wasn't solved quickly was because in the open() function for file reading, the argument "w+" means that you can write to the file specified and if the file doesn't exist, create it.

Regression Testing

After fixing the issues discovered in the unit test, the regression test was a much better success. Profile mode correct identify the correct person. The fix to this was simple because I was encoding the same photo from the image folder and comparing it with itself.

monitor mode is now working better than in the unit test. I had to get rid of the age and occupation variables and just stick with the name. This got rid of the issue previously discovered and monitor mode is able to run and identify people in real time.

The attendance taking feature was also fixed. The solution to this problem was to just create a .csv file called "current data". If the issue was that it couldn't find the file because it didn't exist, even with "+w" as it's argument in the open() function. The attendance system is now running without any problems.

A.5.2 Objectives

This project has helped me achieved my objectives that i set out in the introduction of the dissertation. My understanding of Python has increased because I now have an understanding of the Tkinter library. I know how to customise Tkinter menus and how to incorporate functions to use when i create a Tkinter button.

I know how to implement facial recognition software in Python using the face recognition library. I now know the process of setting up the face recognition algorithm in Python and know what is required in terms of image files, binary data and how to execute code when the correct face is identified.

My knowledge of MySQL has also increased. I made programs in java and React that imports MySQL and execute queries from the code. I learned that although the process of creating a function that executes queries are the same, there are some differences. Python contains the db.commit() to save the changes made by insert, update and delete queries was something I wasn't aware of. I know that in the future, when using a new coding language to execute MySQL, it's important to understand the little differences because I could be stuck for hours wondering why the data won't change.

A.5.3 Metrics

in the introduction of the dissertation, we set out a set of metrics for this project to be deemed successful

Reliability

This metric is deemed as a "success". The menus and the cameras for the profiler will run correctly and won't crash upon start or during run time. Considering that the menus and cameras both passed the unit and regression testing and didn't present any major issues.

Security

The security metric is deemed as a "success". This metric is shown in the login section of the Profiler because the admin password is encrypted. This means that if the server gets compromised and data is stolen, the password will remain safe because hackers only have the encrypted version of it, which is useless to them. Unless the hacker knows what algorithm to use and the key to decrypt, they can't decrypt the password to plain text.

Maintainability

By creating separate Python files for each method, this metric has been deemed as a "success". While developing this project and testing this project, i made several changes to improve the features. This didn't interfere with the features already tested and implemented

Performance

The last metric is also deemed as a "success". The profiler identifies the correct person and calls upon the SQL server to pull the correct information. Login lets the correct admin access the Profiler and doesn't permit any non admins from using it.

A.5.4 limitations

While the profiler is great at identifying the correct person and pulling up the correct data, it does suffer from limitations.

The first limitation is simplistic login screen at the start of the program. In an age where privacy and data are put at risk due to hackers with malicious intent, there needs to be additional security measures such as 2 factor authentication.

Another limitation is the MySQL table structure for the database. The tables designed were suited for governments. If a business or institution like a university were to use this program, they could get in trouble by collecting irrelevant data. For example, A university shouldn't know what assets it's students possess.

A.6 conclusion

A.6.1 objectives completed?

As mentioned in the intro of this dissertation, i set out objectives a

1. do i know how to implement facial recognition software.
2. do i have a deeper understanding of Python?
3. do i have a deeper understanding of MySQL

A.6.2 what i learned

After completing this project, i learned a lot more than i had originally set out in my objectives. These are my findings for the project.

1. I learned how to implement facial recognition software.
2. I now have a deeper understanding of Python
3. I now have a deeper understanding of MySQL

A.6.3 New Knowledge/insight

I learned how facial recognition has truly worked. I was always curious as to how computers can identify people by faces. We humans just look at faces and find notable features like glasses, scars and hair. I always knew that computers' way of processing data was in number and the fact that computers have to convert images to binary to understand what's being presented to them.

I learned that when developing software, the best approach for me is the Waterfall approach. The reason for this is because it is simple to understand and gets you into a routine. When developing software, consistency and planning are the best things that help me to develop good software. I found that during this project, most of the work was just planning and the design the Profiler. Once you have the requirements and all the required research is done, you just follow your own notes and the development process is quick. Even when I got errors in my code, i had to go back to the design phase of the waterfall approach and research solutions online. Once the error was fixed, it was out of the way and i could continue developing the profiler. Now, when im making new software in the future, Waterfall approach is the best project management method to use.

A.6.4 changes i could make

If I could do this project again, I would make the following changes.

I would add an AI component to the profiler. With rise and popularity of ChatGPT and other AI algorithms, it's clear that AI is advancing at a significant rate. To keep up with this trend when making this project again, an AI component could be used alongside the profiler. This AI can help admins with tasks like explain how to use the profiler and what to do if they come across an error. This could potentially increase user engagement and keep people using the program if they feel that they can send feedback to the developer.

I would allow the admin to have the option to remove or create tables based on their profession. The profiler I created is based on data that a government would keep track of and due to Ireland's Data Protection law, that could put businesses or institutions in the middle of a lawsuit. To avoid this in the future, admins can customize their profiler to only collect information relevant to their job. For example, if a university were to use the profiler, they would want to get rid of the asset and bank tables as it has nothing to do with the university. They might want to add tables like modules that contains rows like module name, credits awarded and who teaches this module.

I would also use facial recognition as another verification for log in. At the login screen, the admin will be prompted for a facial ID as well as ID and password. This can increase the security of the Profiler because anyone who isn't the admin trying to use his/her credentials to access the program will be unable to. The one disadvantage to this is that if the person has a photo of the admin, they could use it to bypass the facial ID. This improvement will not be fool proof, but it does help defend against malicious hackers.

A.6.5 Final thoughts

It's good to know that any coding ideas that i see in video games, i have the confidence to bring that to life with the power of code. If that doesn't work out, I now have the knowledge to avoid facial recognition software that will be used by our artificial intelligent overlords who will be fixated on exterminating humanity like in the movie Terminator.

Bibliography

[1] K.W. Bowyer. Face recognition technology: security versus privacy. *IEEE Technology and Society Magazine*, 23(1):9–19, 2004.

[2] Lixiang Li, Xiaohui Mu, Siying Li, and Haipeng Peng. A review of face recognition technology. *IEEE Access*, 8:139110–139120, 2020.

[3] Genia Kostka. China's social credit systems and public opinion: Explaining high levels of approval. *New media & society*, 21(7):1565–1593, 2019.

[4] Alexander Mordvintsev and K Abid. Opencv-python tutorials documentation. *Obtenido de https://media. readthedocs. org/pdf/opencv-python-tutroals/latest/opencv-python-tutroals. pdf*, 2014.

[5] Fredrik Lundh. An introduction to tkinter. *URL: www. pythonware. com/library/tkinter/introduction/index. htm*, 1999.

[6] Sharvari Rautmare and D. M. Bhalerao. Mysql and nosql database comparison for iot application. In *2016 IEEE International Conference on Advances in Computer Applications (ICACA)*, pages 235–238, 2016.

[7] Bogdan George Tudorica and Cristian Bucur. A comparison between several nosql databases with comments and notes. In *2011 RoEduNet international conference 10th edition: Networking in education and research*, pages 1–5. IEEE, 2011.

[8] Anisha Gupta, Rivana Christie, and R Manjula. Scalability in internet of things: features, techniques and research challenges. *Int. J. Comput. Intell. Res*, 13(7):1617–1627, 2017.