The Gamification of Cyber Security Education Using Methods Of Dynamic Challenge Generation

Max Robertson -
40205107

Submitted in partial fulfillment of
the requirements of Edinburgh Napier University
for the Degree of
BEng (Hons) Cyber Security and Forensics

School of Computing

February 25, 2021

Authorship Declaration

I, Max Robertson, confirm that this dissertation and the work presented in it are my own achievement.

Where I have consulted the published work of others this is always clearly attributed;

Where I have quoted from the work of others the source is always given. With the exception of such quotations this dissertation is entirely my own work;

I have acknowledged all main sources of help;

If my research follows on from previous work or is part of a larger collaborative research project I have made clear exactly what was done by others and what I have contributed myself;

I have read and understand the penalties associated with Academic Misconduct.

I also confirm that I have obtained informed consent from all people I have involved in the work in this dissertation following the School's ethical guidelines.

Signed:

Date:

Matriculation no:

General Data Protection Regulation Declaration

Under the General Data Protection Regulation (GDPR) (EU) 2016/679, the University cannot disclose your grade to an unauthorised person. However, other students benefit from studying dissertations that have their grades attached.

Please sign your name below one of the options below to state your preference.

The University may make this dissertation, with indicative grade, available to others.

The University may make this dissertation available to others, but the grade may not be disclosed.

The University may not make this dissertation available to others.

Abstract

It has been widely acknowledged that there is a cyber security skills gap, with companies not having the staff they need to meet their security needs. One proposed solution is to get more people involved and engaged in cyber security at a younger age. Research has shown that gamification can be a useful tool to engage younger audiences with aspects such as freedom of choice, interactive environment, characters, and storylines. This dissertations aims to combine elements of gamification with the already existing CTF (or capture the flag) style learning to design the concept for an application that would be aimed at a high school aged audience with the objective of inciting an interest in cyber security, then to prototype technical elements behind the application such as dynamic challenge generation to test and measure the performance of them and evaluate their use for this purpose. Essentially this dissertation is trying to find out: Can gamification be applied to cyber security learning effectively? and what is the most efficient way to dynamically generate random challenges for this application?

Based on a review of the literature, an application was designed with consideration in the application structure, designing the lessons included in the application and a technical design which discussed how the technology behind the application would work including the dynamic challenge generation, the methods were discussed and prototypes designed. The methods were implemented using python scripts and analysis of the results returned suggested that out of the four methods of dynamic challenge generation (random selection, API request, web scraping and web scraping with API) web scraping with API was the most effective returning performance numbers more than 2x better than any other method. These results indicate that web scraping with the use of APIs is the method best suited for this use with its unique potential to generate random challenges. Further work for this project would include developing the proposed application with further research done on how to scale the application to thousands of users.

Contents

List of Tables

List of Figures

Acknowledgements

I would firstly like to thank my family who have supported me throughout this whole process, accommodating me so that I may dedicate my time to the completion of this project and degree and who have shown encouragement and understanding when I was discouraged.

Next, I would like to thank my close friends at the university who similarly I would not have made in this far into the process if it weren't for their motivation.

I would also like to thank my supervisor Bill Buchanan, who despite being very busy made himself available to me when he could and gave me guidance when I hit some bumps in the road along the course of this project similarly my second marker Richard Macfarlane who gave me advice and referred me to the right people when I was at a crossroads.

Finally, I would like to thank my module leader Robert Ludwiniak who not just throughout this project but throughout my time at the university has excelled in giving me advice and reducing my stress levels.

1 Introduction

1.1 Context

"if organisations don't have employees with the right skills needed to fight back at cyber-criminals, we will be fighting a losing battle. This will ultimately lead to fewer jobs all round as businesses suffer the repercussions of cybercrime". There is a cyber security skills crisis and with the workers in the industry claiming "there simply aren't enough trained people out there right now" [1] a solution needs to be found.

1.2 Background

In recent years a shortage of IT skills has been seen, however, a more severe shortage has been prevalent with Cyber security skills. With statistics like 45 percent of organisations claiming to be lacking in cyber security skills and (found by research done by the Information Systems Security Association) 70 percent of IT workers believe the lack of cyber security professionals is negatively impacting their existing work [2]. This has been a widely discussed issue in recent years and there has been a number of proposals for how to fix this 'cyber security skills gap' from developing the companies HR department so that an individual interested in gaining the relevant skills is given the appropriate training by the company to advancing security technology and crime deterrence so that less personnel are needed to close the gap [3]. Studies have shown [4] alarming numbers both in regards to how many cyber security professionals come from having trained in cyber security vs how many are trained from other roles (see Figure 1-1) and how little corporations feel "very confident" in their employees abilities to perform 8 simple cyber security tasks (see Figure 1-2). Especially so the last one as if the companies employing the staff can't trust them how are the consumers supposed to?

One proposed solution is that an emphasis should be put on cyber security at an earlier age and high school students should be introduced to cyber security elements. If students are introduced to these elements at an early age there is more of a chance of capturing interest in the subject and for students to pursue a career in cyber security. Furthermore if research is done on how to cater this material to the students and how to peak their interests the effect could be greater. This could be done through the use of gamification which has been proposed as a solution for engaging people in individually and socially sustainable behaviours, such as education [5]. CTF (Capture The Flag) competitions are held in cyber security communities which are essentially a gamified environment where teams compete by

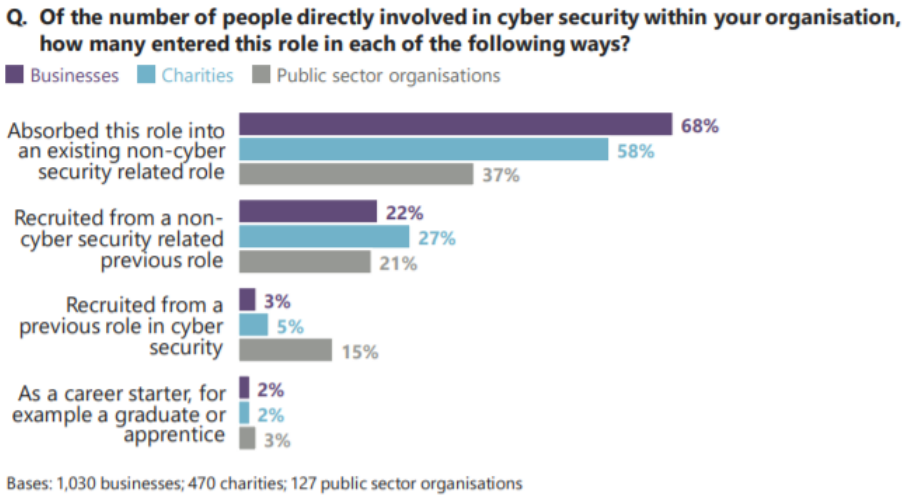


Figure 1-1: Where Cyber security professionals are coming from (Pedley et al 2018)

completing different cyber security related challenges to obtain 'flags' which will score them points. This project proposed to take the structure of a CTF competition and gamify it further for use in the education of cyber security.

1.3 Gamification

Recent years has seen *gamification* rise in popularity [6]. The commercial success of video games, with record-breaking console sales and a huge presence on online multiplayer, has called for research into the effects and relevance in the digital age, this effect made by the digital game medium has encouraged its use in pursuits beyond entertainment [7]. Yu-Kai Chou defines gamification as "the craft of deriving all the fun and addicting elements found in games and applying them to real world or productive activities" [8], educational institutions are interested in the gamification of education, where educators create gamified learning environments to enhance learner engagement and improve learning outcomes [9]. With the concept of gamification becoming more clearly defined in the minds of researchers and practitioners in recent years [10], now is an appropriate time to start looking at the potential of the gamification of cyber security education and the possible methods of doing so.

Capture the flag competitions are one way in which gamification has been adopted in computer science and have been widely used for raising awareness in Cyber security among university students and the like [11] with

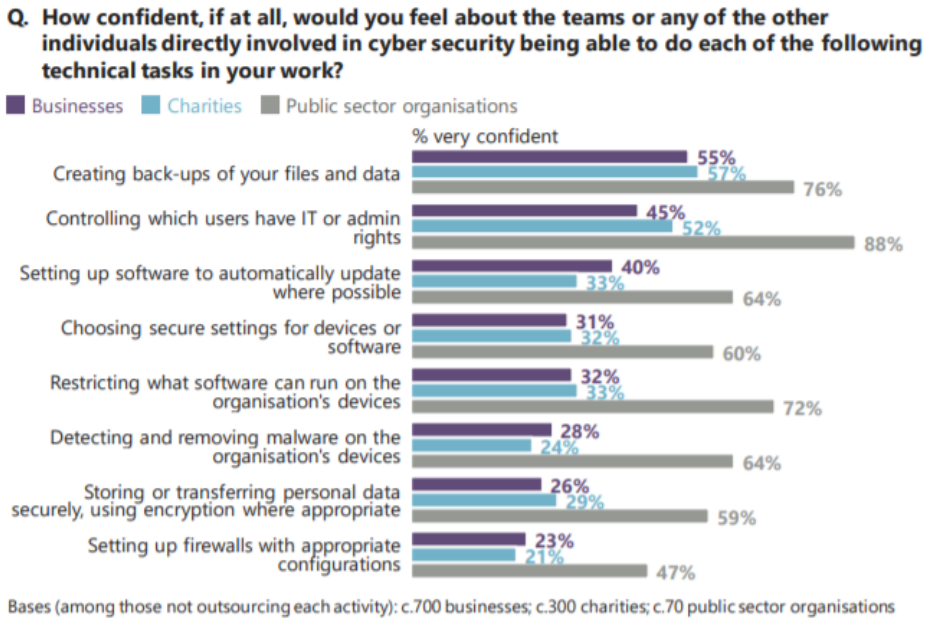


Figure 1-2: What percent feel "very confident" (Pedley et al 2018)

over 70 competitions being organised annually [12]. Some high profile CTF competitions include *Picto CTF* [13] an annual competition which 50,000 students have competed in to date [14], *CSAW (Cyber Security Awareness Worldwide)* which boasts being the largest student-run cyber security event in the world [15] and Defcon [16], who proclaim to be "the biggest underground internet security gathering on the planet" also run a weekend long CTF contest [17]. The activities in these competitions vary. One example of interest is from the previously discussed Defcon competition which contained an activity which tested a security administrators ability to secure a complex system with unknown but required functionality, which has been likened to a real-life scenario faced by a consultant. The difficulty of this activity comes from successfully defending the system while simultaneously providing the required services to the score server, resulting in an action-packed contest [17]. In a competition developed by universities in Texas the teams were told they had gained a contract to run a company network, the team were free to reorganise the equipment in any way they wanted but any service outages would be counted against them. The teams were scored on: their ability to keep required services up and running, reaction to business generated injects (e.g addition or deletion of people) and ability to keep hostiles at bay. This activity took place over days so poor performance early, would be penalised later [18]. As well as being used for competitions CTFs are used

for educational purposes [19], this paper will investigate how to apply this style of learning and combine it with other elements of gamification to design an application that incorporates both.

1.4 Aims and Objectives

As the area of 'the gamification of cyber security education' has been established as the area of research some aims and objectives will be set that it is hoped will be achieved throughout the process of this project:

My first aim is to gain an understanding of different learning techniques that could apply when implementing the project as well as have an in-depth understanding of the concepts that have been proposed to utilise during the implementation. This will be achieved by doing extensive research and produce a literature review, breaking down different learning techniques. Further research will be done into CTFs and the associated available technologies to give a better understanding of how they work and how aspects of them could be utilised for implementation as well as a look into gamification, how to make use of the positives and avoid any pitfalls.

The second aim of this Honours project will be to take the research done in the literature review and understand the potential use of the aspects discussed, prototype aspects of a designed application measuring the effectiveness of different methods of dynamic challenge generation and how they could be utilised to aid in the creation of this proposed application. To achieve this aim the following objectives will be complete:

- Investigate learning models and gamification to develop the understanding needed to design an application for the education of cyber security.
- Design the application with reference to research.
- Design learning materials to be used in conjunction with the application.
- Design technical components of the application to be prototyped and tested.
- Implement the designed prototyped elements.
- Test and evaluate the usefulness of these prototyped elements.

1.5 Dissertation Structure

Here the structure of the dissertation will be outlined and what is included in each chapter will be discussed.

Chapter 2 is the research that has been done to back up the project. A closer look is taken into the cyber security skills gap and the potential solution for this problem is discussed. Then a look into different Learning Models that could be effective is taken which was important when proposing a learning application. After this, the use of CTFs and how their structure can and has been utilised for the teaching of cyber security will be looked into. Finally, gamification is covered discussing aspects such as game design, feedback and user interface as well as its use in computer science.

Chapter 3 is Design; how this application would be structured as well as look, taking what was learned from the literature into account and applying it to designing the elements of the application and what the user experience would be like. The possible "rooms" or lessons that would be offered in the first section of this application are then designed, this is followed by the technical design and a mock-up of how the elements that were decided to be prototyped would be made.

Chapter 4 is the implementation of the code that was required to test and evaluate the different methods of dynamic challenge generation. This chapter goes through each of the scripts that were made with code snippets, discussing what has been done and why they were done and the technology used to make the scripts. After discussing the scripts the implementation of Docker and how it can be used to populate a database is discussed.

Chapter 5 talks about the developed test case scripts to measure the performance of certain aspects of the scripts that were made. This chapter displays the results in graphs which were made from the data returned by the test case scripts. Code snippets of the test case code and example output in some cases are also provided.

In Chapter 6 a look is taken at the results found in the previous chapter and the performance of the aspects implemented is evaluated. Further discussion is done about what was expected and any correlations found and what that may suggest. This is done for each of the tests ran and then summarise and evaluate the implemented aspects as a whole.

Finally, in chapter 7, the project is concluded by discussing my aims and objectives and how successfully this project has met them. After this self evaluation is done. Finally, a discussion of possible future work that could be involved with this project is done, considering aspects of the project that could be realised with further work.

2 Literature Review

2.1 Introduction

This chapter will look at using gamification of cyber security as a way of integrating these skills into high school education and the possibility of it engaging the students in the subject matter more than traditional methods. This review will discuss how to achieve gamification of the learning process through CTF (Capture The Flag) platforms, existing CTF platforms, pedagogies used in computer science, methods of teaching these skills using these pedagogies and setting appropriate learning outcomes.

The demand for Cyber Security has been rising in recent years, as a result, those who possess these skills benefit from a seller's market. Companies that need these skills pay a high price or risk leaving the position unfilled [20]. Driving much of this increased attention is the increased severity of damage that has resulted from failed efforts to secure information systems [21] and with cyber security becoming more and more vital to global security [22] training the next generation of cyber security professionals who understand both the tools and processes involved is the challenge that is currently being tackled in many different ways [23]. With the sheer amount of hours it takes to become a cyber security professional Manson and Pike have suggested that those who start to develop their cyber security skills in university/college are at a disadvantage drawing a parallel between cyber security professionals and athletes, who need to put in hours of dedication before they are ready to pursue a career in university/college and emphasising the importance of engaging students at a younger age [24]. This hope of having more students choose STEM careers, especially in cyber security, is shared by the government/businesses/organisations [25].

2.2 Learning Models

2.2.1 Bloom's Taxonomy

Bloom's Taxonomy is a hierarchy of cognitive learning levels and is presented to help students have a greater level of understanding and abstraction in their course and their entire educational experience [26]. Originally Bloom's taxonomy was defined by the following stages: *Knowledge, Comprehension, Application, Analysis, Synthesis, and Evaluation* [27]. This original model was used to determine the effectiveness of learning objectives and often identified that there was a heavy emphasis on objectives requiring only recall or recognition when actually the objectives falling into the categories from

comprehension to synthesis are regarded as the *most* important [28].

A newer model was developed changing some of the stages order and some of the names to their verb form, the revised stages are *Remembering, Understanding, Applying, Analyzing, Evaluating and Creating* [29]. Furthermore, the new model takes into account multiple domains, a cognitive domain ,and a knowledge domain (see Figure 2-1), This table can be used to effectively determine the use of these domains in a learning objective and identify possible areas of neglection [30].

	The Cognitive Process Dimension					
The Knowledge Dimension	1. ***Remember***	2. ***Understand***	3. ***Apply***	4. ***Analyze***	5. ***Evaluate***	6. ***Create***
A. ***Factual Knowledge***						
B. ***Conceptual Knowledge***				X		X
C. ***Procedural Knowledge***						
D. ***Metacognitive Knowledge***						

Figure 2-1: Taxonomy Table (Krathwohl 2002)

"Is Bloom's taxonomy appropriate to computer science ?", this question has been discussed [31] and it has been observed that it can be difficult to decide upon which cognitive level a certain task lies at, as educators with a more intimate level of the course material are likely to associate a different level with a task than an educator who does not. This has been discussed further by (Thompson et al. 2008)[32] who addressed this issue and noted that there hadn't been an agreed upon interpretation of Bloom's taxonomy for computer science so attempted to do so for a programming module, it was concluded that once staff involved in the teaching of the course an agreement could be made on which cognitive levels tasks belonged to which suggested that the revised Bloom's taxonomy could be effectively used in the programming domain with regards to exam questioning. Trying to cover all the cognitive levels in an examination however, could end up making the exam longer than needed so it has been advised [33] to combine multiple cognitive levels in one question to solve this problem.

Niekerk and Solms discuss applying Bloom's taxonomy to information security education and how it can improve security educational programs. They raise potential issues such as that a program that has not been tailored to the specific needs of an individual, or the needs of a specific target audience

will be ineffective. They also give an example of how to apply Bloom's to information security saying that teaching a student what a password is would lie on the *remember* and *understand* level of Bloom's taxonomy, however, the necessary information to understand why their own password is needed in order to safeguard might be as high as the *evaluate* level of the taxonomy [34]. An example of Bloom's taxonomy applied to information security is shown (see Figure 2-2).

Level	**Verb**	**Sample Activities**
Create	design	Write a new policy item to prevent users from putting sensitive information on mobile devices. **(A6)**
Evaluate	critique	Critique these two passwords and explain why you would recommend one over the other in terms of the security it provides.**(A5)**
Analyze	analyze	Which of the following security incidents involving stolen passwords are more likely in our company?**(A4)**
Apply	execute	Use the appropriate application to change your password for the financial sub-system. **(A3)**
Understand	discuss	Why should non alpha-numeric characters be used in a password? **(A2)**
Remember	define	What is the definition of *access control*? **(A1)**

Figure 2-2: Bloom's applied to information security (Niekerk and Solms 2009)

Buchanan et al. have done research into blending bloom's taxonomy and serious game design suggesting that there may be a way to connect game interaction and the learning objective levels of Bloom's taxonomy concluding serious games are a good medium for this as they allow novice learners to focus on the general concepts and skills of the subject matter, and not become distracted by a tools interface [35].

2.2.2 Psychomotor

While the cognitive domain relates to strategic business intelligence, prudence and applying the psychomotor domain relates to complex overt response, Modus (way in which anything is done) and guided response, essentially cognitive is thinking and psychomotor is doing [36]. Psychomotor is relating movement or muscular activity associated with mental processes, Psych meaning mind and Motor referring to the motor neuronal system in the brain and spinal cord [37].

Psychomotor skills and computer games have been widely researched over the last 20-30 years for example in teaching psychomotor skills in the medical industry using virtual reality [38] [39] [40] [41] [42]. These computer simulation games have been found to improve psychomotor skills [43]. It has

been suggested that psychomotor speed can lead to a fast and accurate performance in cyber operations [44], psychomotor skills and gaming have been linked with one study suggesting that it is likely that frequent and repetitive game playing can improve psychomotor abilities [45].

2.3 CTF platforms

As previously discussed Capture the flag competitions are becoming more and more popular. They consist of a set of challenges each from a different area of cyber security, some of the areas covered include:

- Cryptography
- Exploits
- Forensics
- Networking
- Reversing
- Stenography
- Web

Cyber security skills will have to be used in different challenge scenarios to find the flag, which will usually be text that can be submitted to a scoring server[46]. Typically these platforms will also have features such as a hint system where a user can get help if they are stuck on a challenge and a scoreboard where users can see who is performing best in the competition.

2.3.1 Existing CTF platforms

A variety of platforms have been developed to host CTF tournaments, each have their own advantages and are useful for different situations. **CTFd** The CTFd framework focuses on ease of use and customisability. Written predominantly in python. It handles Static and regex-based flags. Users can access hints if they are stuck on a challenge. File uploads to the server on Amazon S3. Includes features such as Dynamic scoring, Score graphs, team management ,and hiding. CTFd also allows users to customise everything using plugin and theme interface [47].

When running a CTF using this platform the admin can observe the performance of the participants easily using a statistics page, which tracks things such as the ratio of correct submissions to wrong submissions. The

admin can also check the teams, scoreboard and customise the configuration. This flexibility of content creation is CTFd's strength and makes it ideal for changing for intended purposes such as training [19]. The user interface for ctfd is relatively simple (see Figure 2-3).

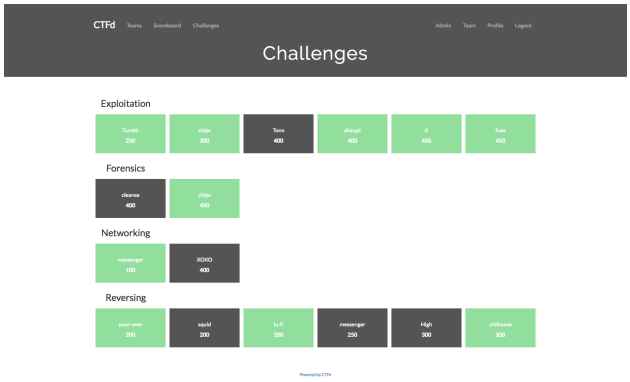


Figure 2-3: CTFd User Interface

As mentioned the CTFd framework is highly customisable allowing the core code to be altered for specific purposes as plugins, some that have already been developed include: different question formats such as multiple questions, different scoreboard formats and ctf event countdown timers [48]. After downloading the platform you can set up a CTF event, logging in to an admin account on the CTF allows you to easily create your own challenge, setting the category of the challenge the type of flag (e.g static or regex) and what the flag is. The customisation allows the creation of new routes, modifying existing routes, Add tables to the database, registering asset types and challenge type creation [49].

Facebook CTF Used for hosting jeopardy and *king of the hill* style CTF competitions. Written in Hack. Can be two players up to several hundred. It has a fancy AI showing a world map, each of the challenges appear in different countries and users will beat the challenges to claim that country. Can be installed in either development mode or production mode. Capturing the countries give you points, you accumulate points which can be compared against other teams/players using a dynamic leader board (see Figure 2-4) [50].

Facebook does not emphasise on customising the platform as much as ctfd. The platform allows the adding of challenges as an admin by giving a title, question, answer, points, hint, hint penalty and what country on the map the challenge will be placed on but not as much as is given with the install to actually change the platform for personal use. Also accessible as

Figure 2-4: Facebook CTF visualization

an admin is the abilities such as configuration, controls, announcement and team management where teams can be monitored and given admin privileges. The platform does allow for new pages to be inserted by editing the IndexController file that is included with the initial install.

Mellivora

CTF engine is written in PHP. Includes features such as arbitrary categories and challenges, scoreboard with optional multiple team types, challenge hints, team progress page, local or amazon s3 challenge file upload and user/email/IP search[51].

Other CTF platforms and technologies With CTF's gaining in popularity there has been an abundance of applications developed to aid CTF competitions. These range from scoring servers such as Hack the Arch [52] which has an optimal dynamic hint system included or Security Scenario Generator (SecGen)[53] which creates vulnerable virtual machines so students can learn security penetration testing techniques, it has been successfully used to enhance security education by providing these randomized targets for lab exercises, large team project security audits and for generating CTF competition VMs [54]. There are also some "always online ctf's" like Hack the Box [55], ctfs.me [56] and Hone your ninja skills [57] all of which offer challenges similar to ones in CTF competitions available all the time on websites.

2.3.2 CTFs in the classroom

CTFs (class capture the flags) have been researched [58], small-scoped competitions engaging students in attack and defence scenarios as teams reporting back that this spiked students interest in learning and allowed them to master skills discussed in lectures and introduced in practicals (for suggested differences between a competition CTF and a class CTF (see Figure 2-5)

Capture the flag platforms are being used in the classroom for networking

Feature		CTFs	CCTFs
F1	Preparation	A few months	A few weeks
F2	Duration	1-2 days	2 hours
F3	Topic	Hacking	Exploits, DDoS, DNS, BGP, crypto, etc.
F4	Team roles	Blue or Red, rarely both	Both Blue and Red
F5	Team pairing	All against one or all against all	Playoff
F6	Occurrence	Annual or semi-annual	2-3 times per semester
F7	Postmortem	Rarely	Always
F8	Advancement paths	None	Built-in
F9	Difficulty	Expert	Beginner to intermediate

Figure 2-5: Comparison of CTFs and CCTFs (Mirkovic, Peterson 2014)

[59], presenting labs in a CTF game with the materials being transferred into challenges, reporting back an increase in the students' comprehension skills as well as being eager and motivated to learn. CTFs have even been used by the united states air force academy for teaching cyber security [60] citing students pushing their own boundary for knowledge as an advantage and state they are looking how to further incorporate CTF into their curriculum. CTFs are a thorough security exercise that integrates well with cyber security educational targets and so can be used effectively to train cyber security professionals [61] as well as an attractive and effective way to raise awareness and interest in security while simultaneously educating for prevention and addressing the private and government sector needs [62]. The competitive element of these events drives and compels all participants to more enthusiastically commit and engage in active experimentation and learning [63], this freedom is a key part which motivates students to experiment and forces those students to hone skills that are not normally covered in a standard information security curriculum [64]. Albert et al performed a study and gathered results by doing a survey, the majority of comments indicated that the students enjoyed themselves and learned more about a field they were interested in and was successful in educating high school students about higher education opportunities [65].

2.4 Gamification

Gamification is the design approach of utilising gameful design in various contexts for inducing experiences familiar from games to support different activities and behaviours and has continued to be a popular topic in both industry and academia since it became popular in the early 2010s [66]. Gamification has been shown across multiple levels of academic instruction to have a positive impact on task completion by augmenting the experiential elements encountered by students who are engaging in the learning process [67].

2.4.1 Gamified elements vs. Gamified experience

Gamification has been defined as "the use of game design elements in non-game contexts" [68]. It is important to know when considering the gamification of learning there is a difference between gamifying a whole experience and adding gamified elements. Gamifying an experience involves an actual game, there is a start state, gameplay and an end state. Users can tell they have reached the end when there is a "win state", along the way the game will have taught some knowledge and skills to the user. Gamifying elements, however, only uses elements of a game. Users will not play through a game start to finish, rather, participate in-game elements such as completing challenges or earning points or badges for completing those tasks [69].

The term serious game is given when a complete game is used for a non-entertainment purpose, whereas gamified applications use elements of games that do not give rise to entire games [68]. Further distinctions between the two have been made. Serious games are usually made to fulfil the role of the tutor providing the instructional content to the learners, gamifying elements ,on the other hand, is designed to augment or support pre-existing instructional content. Furthermore, serious games use all game elements just to varying degrees, whereas gamifying elements involves the extraction and application of particular elements from games for a considered use on a non-game process[70].

2.4.2 Game Design

Looking more into developing a game for educational purposes, game design needs to be extensively looked at. Games have the ability to provide a media for delivering training in an engaging format at levels appropriate for the individual user [71], the idea is also not a new one, games have been used to support health, education, management ,and other sectors have yielded positive results [72]. When designed well, video games can enthrall players. drawing them into a virtual world, motivating them and challenging them [73], with research also indicating that games can support and enhance learning and training [74]. Game mechanics should be considered when designing a game, these are essentially the rules of the game and fall into classes such as *Setup, Victory Conditions* (how the game is won) *Progression of play, player actions* and *game views* [75].

Annetta defines 6 principles to follow when designing games for education [76]: Players should have a unique *Identity* in the game world, which in turn allows the user to become more emotionally involved in the game and care about the consequences. This leads to *immersion* which is a heightened

sense of presence that means the player is more engaged in the content and motivated to succeed. *Interactivity* involves the player being able to interact with NPCs (non playable characters), *increased complexity* keeps the player challenged, *Informed teaching* focuses on providing feedback to instructors and the game should be *instructional*.

Similarly but more specifically, Adams and Makramalla discussed four elements of gamification for cyber security skills training [77]: *Progress mechanics* related to player motivation through the provision of progress tools such as points, leaderboards and badges, *Player control* (which closely relates to Annetta's Identity principal) the use of a character to engage in the gamified training, *Problem Solving* crucial element in gamification when learning and retaining new information and *Story* (which closely relates to Annetta's Immersion principal) a narrative that is the present to create an attachment or bond between the learner and their avatar, stories also motivate the learner to keep on playing to find out the rest of the story. As well as correlating with Annetta's Immersion principal this element is backed up further by literature, an engaging story-line makes the player *want* to play the game [78] and can provide a narrative for the game and begin to immerse the players in an alternate world. These games create goals that a player feels motivated to reach [79] this is beneficial for motivation and the more motivated the student the more likely the educational goals will be achieved [80].

Also when designing a game for cyber security training it would be advantageous to acknowledge the shortcomings of existing cyber security training material such as too many topics being discussed in too little time - users cannot be expected to understand/ retain all of them and a successful cyber security skills training program must be able to do two things: one is get and retain the user's attention for a span of time, and two is to communicate training material to the user effectively in a span of time [81].

2.4.3 Feedback

Hattie and Timperley define feedback as information provided by an agent (e.g teacher) regarding aspects of one's performance or understanding. A teacher or parent can provide corrective information, a peer can provide an alternative strategy, etc. Feedback is a consequence of performance [82]. Simple feedback helps learners verify the performance expectations, judge their level of understanding and become aware of misconceptions, instructional feedback can also provide clues about the best approaches for correcting mistakes and improving performance. Effective performance provides the learner with verification, a judgement of whether an answer is correct, and elaboration, additional information to help the learner [83].

One widely researched area of feedback is multiple choice assessments using immediate feedback. Merrel et al. carried out a study assessing the usefulness of reporting back immediate feedback, allowing the user to make a second attempt and be rewarded partial credit if answered correctly. The results showed that the average grade went up by nearly 6.57 percent [84]. Further research has indicated that methods that permit immediate feedback to students during lectures and tests increase more effective long term understanding [85]. Attali performed a study comparing different types of feedback: NF (no feedback), immediate knowledge of correct response (KCR), multiple try feedback with knowledge of the correct response (MTC) and multiple try feedback with hints after an initial incorrect response. The study found that 38 percent of users were able to correct their incorrect answer on the second attempt without a hint but with a hint 52 percent were able to conclude that a hint after an incorrect response was more effective than multiple try without hints and as well as providing an opportunity to understand the initial error, provides more information and a clearer path [86].

There is also a multitude of different forms of feedback when it comes to games such as losing lives, receiving encouraging messages, or earning rewards (e.g points) [87].

2.4.4 Gamification in Computer Science

The features previously discussed, such as a point system and leaderboards, are being implemented in modules of Computer Science as gamification becomes an emerging pedagogical technique. A study was done in 2014 at Alabama A&M University [88] of students studying Computer Science courses. This study reported a match between the score of virtual points and the final grade for the majority of students. The study also states that this result would make it easier to claim that the gamification mechanism drives student motivation to learn.

More specifically to cyber security, a study was done at a high school camp launched by Purdue University Northwest in the summer of 2016 and 2017[89]. This study was done by using four cyber security game-based learning applications. The results of this study indicated that the camp had not only enhanced the students understanding of cyber security but also increased their motivation to pursue higher education and careers in the field of cyber security. The game based learning applications used in this study may be closer to a serious game than the gamification of learning but perhaps an emphasis on the game elements is what engaged the younger users so much.

2.4.5 Negative effects of gamification

It is important when looking into developing learning material using gamification to look at the possible negative effects it can have. One paper [90] takes a look at multiple studies done on the gamification of learning (the majority of which were in the computing field) and identified four main negative effects that are possible when using gamification: *Loss of performance* where gamification harms or hinders the students learning process by demotivating them. *Undesired Behaviour* where the gamification caused a different effect on the learning context in which it was applied, as a result of poor planning. *Indifference* where gamification simply did not have any impact on the students. *Declining effects* occurring less than the previously discussed effects, this effect is the reported loss of motivation and engagement over time when using the gamified application. The paper concludes that these negative effects are likely due to lack of proper methods/frameworks for planning and deploying gamification in a learning environment. It has also been stated that gamified elements of a course lack alignment to curricular outcomes[91]. Another study, looking more specifically at gamified CTF environment identifies *They aren't calibrated to participants needs, partial credit is typically not supported and the need for expert support personnel* as potential issues [92].

2.4.6 User Interface

HCI (Human Computer Interaction) focuses on how best to design interactive systems so that they are both productive and as pleasurable to use by the intended user as possible [93], getting the user interface right can make a fundamental difference [94]. Traditional user interfaces deliver information with a minimum of subtlety and, due to users limited attention capacity must keep throughput low [95]. One important aspect that should be considered when designing a user interface is consistency, for example, can all the pop-down menus be easily identified by a shared marker such as a downwards arrow. A menu without such notification would be an inconstancy and could initially confuse the user [96]. A dangerous assumption that is made when designing user interfaces is that 'Everyone is like me', this should be avoided and it should be taken into consideration the gap in understanding between the designer and the targeted user [97]. However, when designing a user interface with novice users in mind a balance needs to be struck as to support expert users who are often trapped in "beginner mode" [98]. Many in the computer field cite different ways to design better user interfaces such as "Know the user", "designers are users too, just give them the space and time

to design it" and "make the components good enough and the system will take care of itself" [99].

2.5 Conclusions

To conclude, the Cyber Security industry is crying out for professionals who are skilled and experts in their field. Deploying CTF styled learning in a gamified fashion in the classroom could help combat this shortage, with an almost unanimous agreement across many research papers that the CTF style of learning helps users get to grips with cyber security skills and potentially spark an interest in cyber security. By combining the two it is possible to take advantage of the benefits of both CTF learning and the discussed benefits of a gamified experience. While the impact of this style of learning should not be underestimated it should also be taken into account that there are potential drawbacks which should be taken considered during the development of the platform. Similarly caution should be taken when developing the learning material, making sure the appropriate levels of the cognitive domain are covered while keeping the questions at an achievable level. While it may not have been clear in the past how to apply certain taxonomies to computer science or how to efficiently use gamified learning in a classroom environment, their use is becoming more and more common and with the popularisation of CTFs as a teaching tool and the benefits of gamified learning being widely discussed solutions are being discovered so that this method of learning can succeed and benefit the cyber security industry.

3 Design

3.1 Introduction

This chapter of the dissertation contains the design for an application that will take full advantage of the researched areas maintaining the goal of engaging a younger audience with the possibility of developing an interest in cyber security. This chapter of the dissertation hopes to design this gamified application with elements of a CTF competition used in the design process with Bloom's taxonomy being considered in the design process also. By doing this, elements of serious games which research has shown can engage a younger audience, using elements such as story and player control, will be combined with the structure of CTFs which competitive nature compels students and freedom of choice encourages students to experiment and learn various areas. As it is designed to be an educational tool the research done on Bloom's taxonomy will be applied to ensure the appropriate domains are covered and that the learning element of the application is done at an engaging level and is not overwhelming or boring. Combining these elements into one cohesive application will emphasise gamification, taking advantage of the many positives that research has shown it can have whilst also taking into account the potential pitfalls that can be involved when using gamification to avoid them. Ultimately the goal of this chapter is to combine the research done in the previous chapter with knowledge of available alternatives to design the concept for an application which is effective in handling complex subjects in a way that younger students can understand and apply the skills after use. An additional effect hopefully being, which the research shows is often the case with CTF style learning, an increase in interest for the subject of cyber security itself.

The literature has backed up the need for such an application. How the areas covered can be implemented in the design of the application will be looked into and will go on to discuss how the application will be designed in regard to its structure, taking account what has been discovered in the literature. The design of some of the lessons that application could contain will be done before then going into detail about the technical design of the application. This will take into account methods of research and consider what the best approach to take is to meet the aims of the project.

3.2 Requirements analysis

After doing the appropriate research on the topic, and designing the concept for the application, the best to carry out the research to meet my aims and

objectives was then considered. Williams defines qualitative research as allowing the user to explore and better understand the phenomenon whereas quantitative research provides an objective measure of reality [100]. Greenbaum has further discussed these research methods [101] and cited Michiello's table as useful for deciding a primary research method (see Figure 3-1).

	Qualitative	Quantitative
Conceptual	Concerned with understanding human behaviour from the informant's perspective	Concerned with discovering facts about social phenomena
	Assumes a dynamic and negotiated reality	Assumes a fixed and measurable reality
Methodological	Data are collected through participant observation and interviews	Data are collected through measuring things
	Data are analysed by themes from descriptions by informants	Data are analysed through numerical comparisons and statistical inferences
	Data are reported in the language of the informant	Data are reported through statistical analyses

Source: Adapted from Minchiello *et al.* (1990, p. 5)

Figure 3-1: Quantitative vs Qualitative

As the table shows the qualitative method would involve me studying the human interaction side of things, e.g making the application and through the form of questionnaires, getting data on how the application was useful. A quantitative method involves measuring things, numerical comparisons, discovering facts ,etc This would involve prototyping aspects of the application and measuring their use. Both methods would be valid for meeting the aims and objectives of the project if designed right. However, the qualitative method, in this case, would involve testing the application on the target audience (high school students) and would ,therefore, throw ethics into consideration. Another issue with the qualitative method is if this dissertation was planning to do a questionnaire to gather data on the effectiveness of the application, that questionnaire should really have started to be designed earlier in the project life. The quantitative method ,however, could be useful in after theoretically designing the app, prototyping components of it and gauging how useful it would be in achieving the desired effect by testing and measuring the technical performance of the prototyped components. This would help give insight into the proposed use of the technology and if it should be considered for this use. For this reason, it was decided to choose the quantitative method of research and do a technical evaluation on

prototyped elements of the proposed application.

3.3 Application Design

To design the structure of the application, and how the rooms/levels would be spaced out the literature was referred to, to design an application that would effectively teach cyber security skills and knowledge.

Firstly each "lesson" would be separated into a room in the game. A series of these rooms or "lessons" would make up a floor. These rooms would cover and teach different topics of cyber security. Instead of having floors cover different topics, which is how the application had originally been envisioned, the literature suggests that Bloom's taxonomy Is an effective teaching model to apply to cyber security education so each floor could represent a different level on the Bloom's taxonomy (with the exception of the first floor which could combine the first two "remember" and "understand") level of understanding, as the literature suggested there may be a way of blending serious game elements with Bloom's taxonomy [35] this is a proposed way of doing so. So as the user would progress up the floors the questions would get more complex and require deeper thought. For a visual representation of this structure (see Figure 3-2). This would allow the application to impart skills in an organised manner so that the user would get the most out of it and by the end of the application would be at the final level of Bloom's taxonomy.

Another aspect to consider in regards to the structure is that putting "lessons" / rooms back to back requiring the user to complete the lesson in order to progress to the next would be a bad idea. This has been shown to stunt the learning of younger audiences and discourage them from continuing if they can't get the answer. The literature backed this up implying that freedom of choice is an important factor in gamification and that it encourages the user to explore more and be more engaged with the learning process, with the literature previously discussed saying this freedom motivated students to experiment [64]. To get around this the application would feature something called "reward tokens" which the user would get every time they complete a room/lesson. Say there are 5 rooms on a floor, the user would need to obtain 3 reward tokens to proceed to the next floor. Therefore giving the user the freedom to choose which 3 lessons to complete. The user would also need to be stopped from just completing all 5 lessons on the first floor then only 1 in the second to progress, a solution to this would be to double the reward tokens the user gets for completing a lesson as the floors ascend and adjust the reward tokens needed to go up a floor accordingly.

The user would also need to not be discouraged if they ,for example, could only complete two of the lessons and are stuck on completing a third

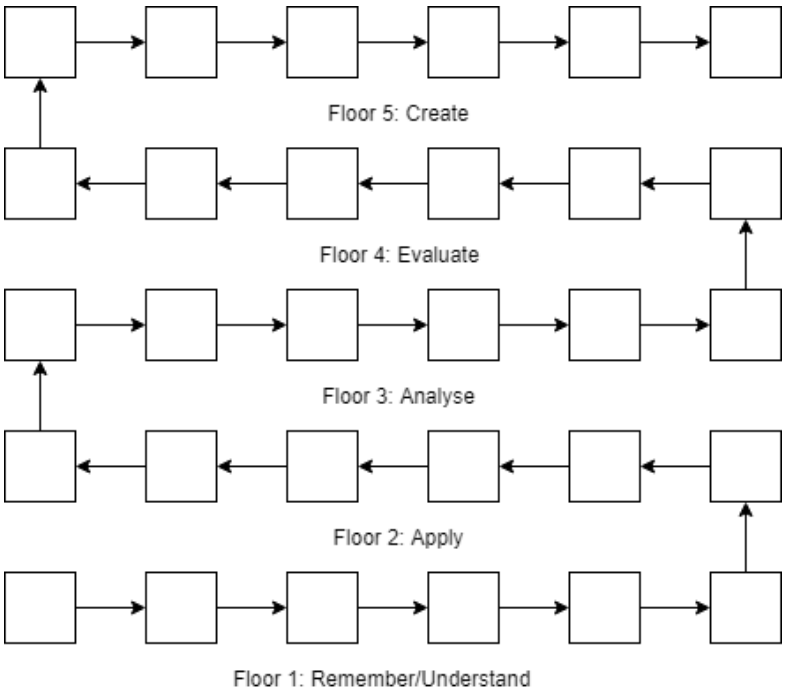


Figure 3-2: Proposed Game Architecture

on a floor. Having looked over the literature on feedback it suggests that just telling the user that they are wrong isn't very effective but that giving the user some form of hint is beneficial for the user [86] allowing them to not give up if they don't know the answer but be pointed in the right direction, with more points being rewarded for an initial correct response and less for a second attempt ,etc. For this reason, a hint system is included in the theoretical design of the application. If the user unsuccessfully attempts a lesson 3 times then a hint room could unlock.

The rooms could be designed to incorporate all of these features. In a typical room, there would be an opening in every direction. One to the left taking the user to the previous room, one to the right taking the user to the next room, one above which will unlock when the user gets the question right and takes them to the reward room where they can collect the reward tokens and finally one below which would unlock on the third incorrect attempt taking the user to the hint room. For a visual representation of this structure (see Figure 3-3).

The user would play as "anonybot" a robot working for moral Corp who when doing his job discovered they were unlawfully harvesting and processing the personal data of customers. When he confronted his superior about it he was locked down on the bottom floor of the headquarters. The user must

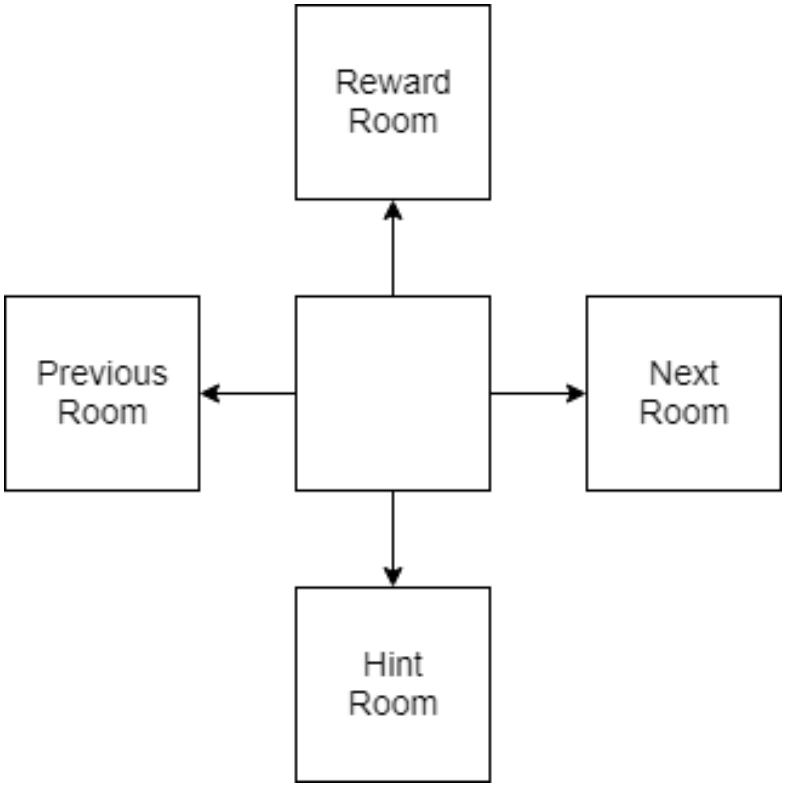


Figure 3-3: Proposed Room Architecture

play as anonybot as he ascends the floors so he can escape the HQ and out the company for what they are doing.

The inclusion of a character, story and an end goal to reach it are all factors which have been cited as useful in, engaging a younger audience with learning material [77] (the narrative creating an attachment/bond with the user and avatar), investing them and making them want to reach the end goal, by the literature. As this application is aimed at the younger high school audience, with the aim of engaging them in cyber security this is also included in the theoretical design.

So ,in summary, the application would essentially be a gamified cyber security application, structured similarly to capture the flag (or CTF) events with the freedom of choice operating as the jeopardy style of CTF shown to be effective in engagement, the lessons serving as the challenges and the reward token serving as a reward for a correct flag. The hint system in place is also represented in typical jeopardy style CTFs. The difference is the character and story orientated take and the elevation of floors representing a respective level of Bloom's Taxonomy.

3.4 Lesson Design

As discussed the level design would be rooms bundled into floors, each floor representing a different level of complexity on Bloom's Taxonomy. For this design chapter, this dissertation will briefly discuss what some of the lessons in the rooms on the first floor would look like, what their content will be, their learning outcomes and then go on to discuss how this dissertation will achieve this.

3.4.1 Caesar cipher

One of the first things that would be covered in the application is basic encryption types, such as the Caesar cipher.

Learning Outcomes
the learning outcomes for this particular lesson would be:

- The user would know and understand what a Caesar cipher is
- The user will be able to use an online decoder to get cipher text into plain text

Lesson Contents
The user would first in a text terminal be told what a Caesar cipher is and would be explained how encryption and decryption work using it. The user would then be given a random Caesar cipher of a famous computer scientist and told to open an online decoder. Using the decoder and referring to the provided steps the user will be able to get the plain text and when entered in the terminal would unlock the door.

How this would be done
Dynamic challenge generation is an important part of this application and would ensure that the students all have a unique experience meaning users can't copy off of each other's answers and will have a different experience if playing again. This is discussed furthermore later in the design chapter. For this lesson test scenarios would be made using random selection and API requested methods, these will also be discussed further later in the design chapter.

How it would look The caesar cipher lesson has been mocked up (see Figure 3-4).

3.4.2 Hash crack

Another lesson that would be covered on the first floor of the application is hashing.

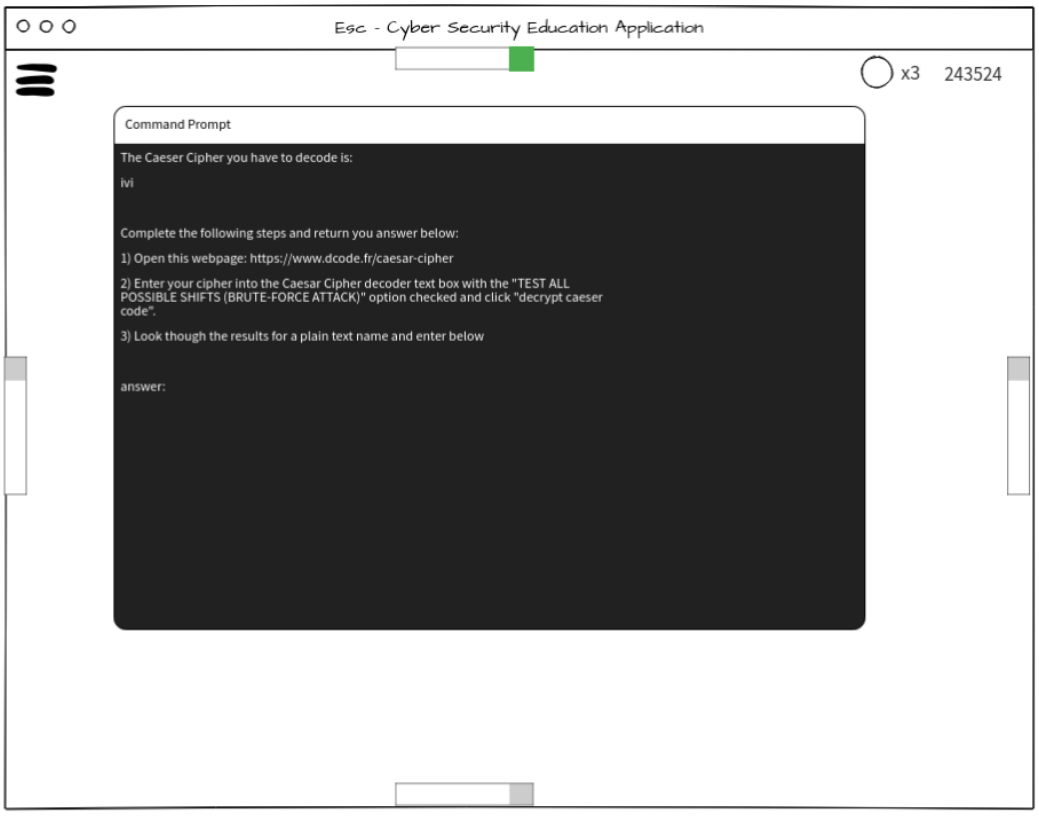


Figure 3-4: Caesar Lesson Mockup

Learning Outcomes

the learning outcomes for this particular lesson would be:

- The user would know and understand what a hash is
- The user will be able to use the command line
- The user will, with instructions, be able to crack a hash using a dictionary and john the ripper.

Lesson Contents

The user would first in a text terminal be told what a hash is, about one-way encryption and the use of it. Then the user will be taught briefly about the command line, basic commands and about john the ripper. the user will then get given a famous computer scientist hashed and be guided with a set of instructions on how to run the hash against a dictionary of computer scientists to get a match using john the ripper. Once the user finds a match they will enter it and the door will unlock.

How this would be done
For this lesson the test scenarios would be made using the web scraping method, this will also be discussed further later in the design chapter. The lesson would then have a dictionary full of the hash value of different computer scientists provided to the user.

How it would look This lesson would look the same as the caesar lesson, with the lesson being done over the command line.

3.4.3 Who is this?

Also, a possible lesson on the first floor would be a simple "which one of these images is...." stage, asking the user to identify a computer scientist out of a selection of pictures.

Learning Outcomes
the learning outcomes for this particular lesson would be:

- The user will be able to recognise famous computer scientists.

Lesson Contents
The user will be presented with a question "which one of these people is *insert person here*", four pictures will then be displayed on the screen, one being the right person and the three others being other computer scientists. When the user gets this right they will proceed to stage 2 and be asked again, this time it will be slightly harder, a less famous computer scientist. Finally, if they get that correct they will proceed to the third stage with a lesser-known computer scientist. By answering the third question correctly the lesson would be complete and the door would unlock. An incorrect answer will result in the user having to go back to the first stage again, this is multiple choice with immediate feedback and has been discussed and backed up by the literature [85]. This technique of try and try again has been shown to be effective for imparting knowledge.

How this would be done
For this lesson, the test scenarios would be made using the web scraping method with the use of two API's, which this dissertation will go into more detail about later in this chapter.

How it would look The "who is this" lesson would show four images on the screen and ask "which one of these pictures is Alan Turing" for example the user would select one of the four pictures and depending on whether they were correct or not the stage counter would change (see Figure 3-5).

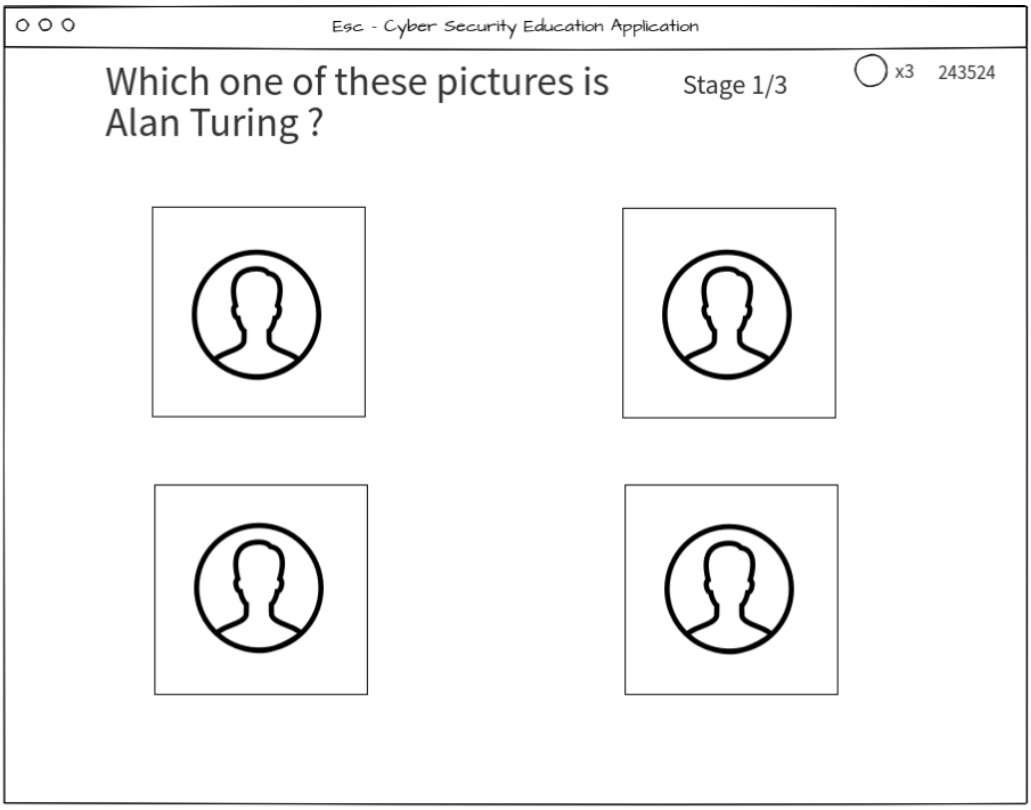


Figure 3-5: Face match Lesson Mockup

3.5 Technical Design

This section of the dissertation covers the design of the technical elements of the project, starting with mocking up the user interface, then designing the methods of dynamic challenge generation and finally a discussion on the back end of the proposed application designing databases.

3.5.1 Front end: User Interface

The user interface will take into account the literature and represent the application in the most effective way possible. The main screen would be relatively simple (see Figure 3-6).

The main screen has quite a lot going on. In a typical room, the player will have a choice to navigate both to the next room and the previous room to their left and right. If they go up they will reach the challenge terminal which when interacted with brings up the challenge. If they enter the correct answer the reward room will open up and they can collect their challenge token. If

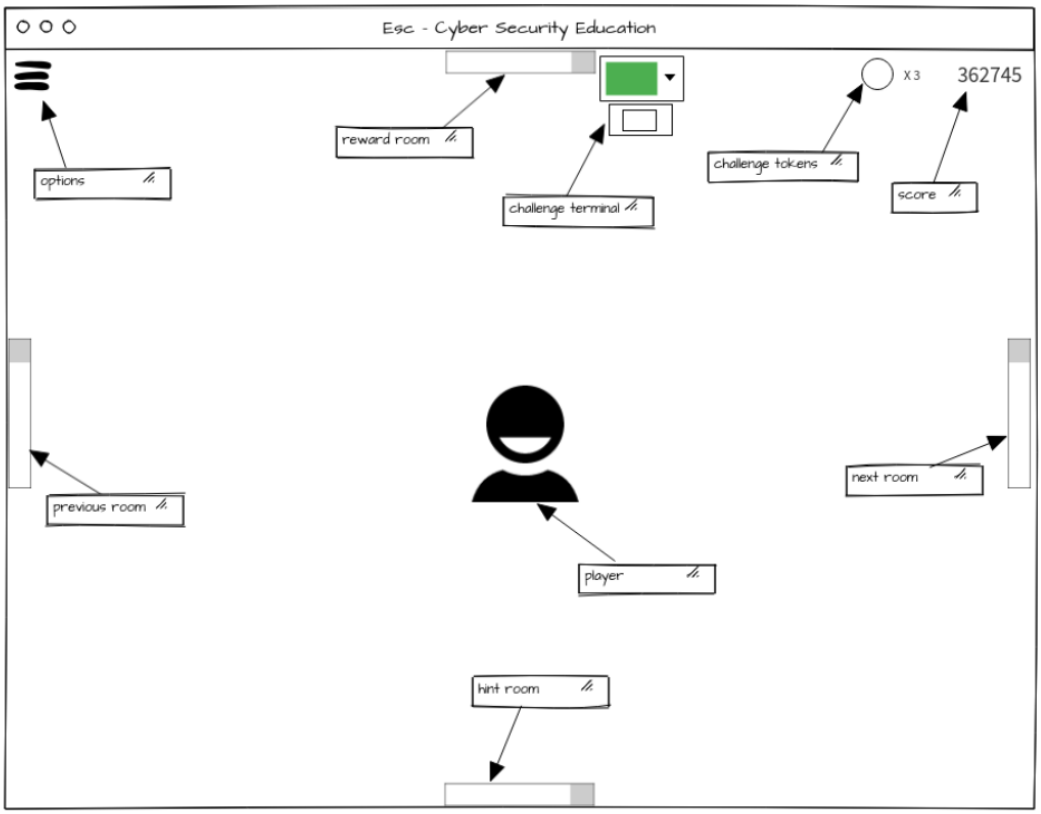


Figure 3-6: Main Screen UI mockup

they get it incorrect the hint room below opens up. How many Challenge tokens they have is displayed and also the score they have accumulated so far. The user can quit or alter their settings with the drop-down menu in the top left-hand corner of the screen (see Figure 3-7). The idea of the design was to try and cover all the necessary elements of the application but keep everything consistent and keeping the delivery of information clear and apparent as per the literature's suggestion [95] [96].

When the user enters the hint room they are presented with a question and a choice of three answers (see Figure 3-8). They will get the hint regardless but is meant to build their knowledge and what's been covered. The reward room is simple and just contains the challenge token (see Figure 3-9).

The feedback received at this level is basic, keeping throughput to a minimum was encouraged and backed up by the literature [95], dependent on what the answer is different doors unlock (see Figures 3-10 3-11).

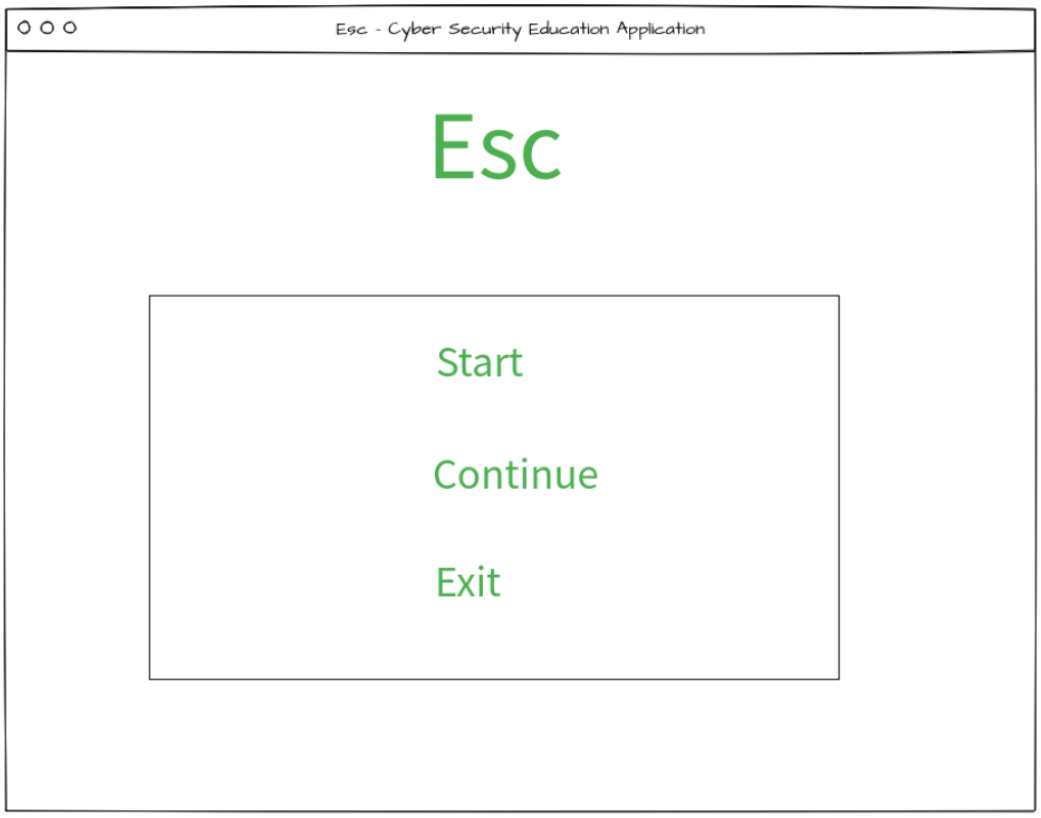


Figure 3-7: Main Menu UI mock up

3.5.2 Middle end: Dynamic Challenge generator

One fundamental part of this application will be the challenge generator. Dynamic challenge generation is an important element of this application, it will ensure that if the user comes back and does it again they will have a different experience and also that users will be having a different experience than the users around them. This dissertation will look at four possible methods of dynamic challenge generation. *Random challenge selection, API request, Web scraping* ,and *Web scraping.*

Random Challenge selection

This will involve setting up a list of possible questions with corresponding answers, then selecting one of those questions at random. A multiple choice question would be asked populated with the correct answer and the answers to the other questions. The script would them check for the user input and check if the answer matches the correct answer giving simple feedback accordingly. This will be the simplest method and the easiest to implement (see Figure 3-12).

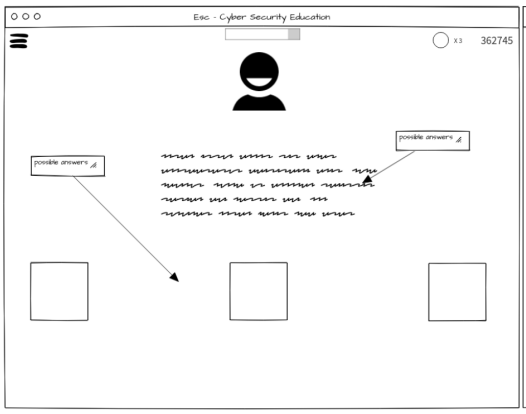


Figure 3-8: Hint room

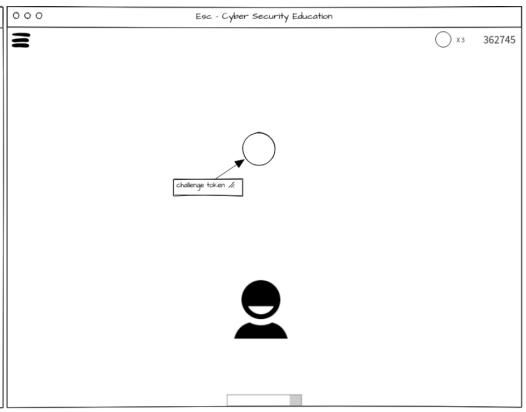


Figure 3-9: Reward room

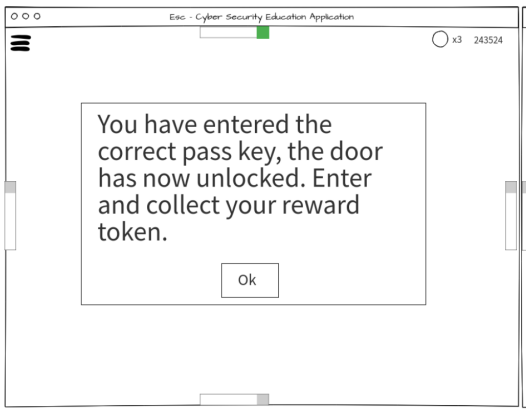


Figure 3-10: correct response screen

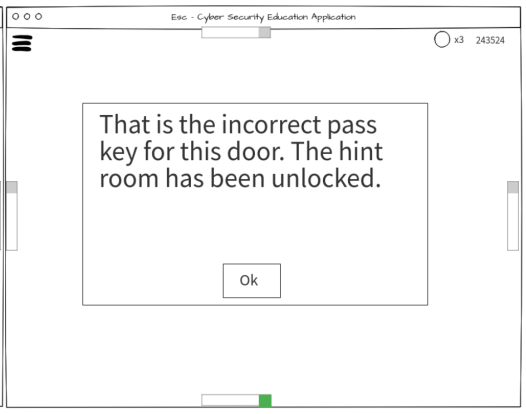
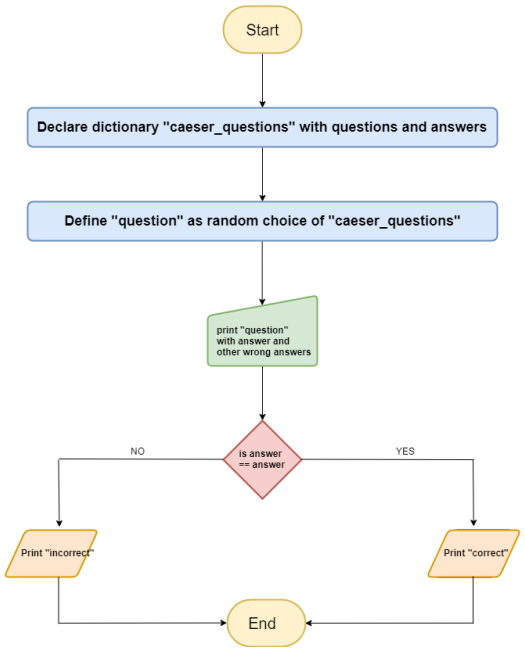


Figure 3-11: incorrect response screen



(1).png

Figure 3-12: random selection flow chart

API request An API will be set up using a python script so a request can be made to it with another script. The script will select a random word from a list, the world will correspond with a question that the API will send back when it receives this word. The user will then send back the answer which will be sent to the API and checked (see Figure 3-13).

Web Scraping Two web scraping applications will be developed to test the effectiveness of it as a method of dynamic challenge generation, one simpler one and one more complex. For the more simple one, the names of influential computer scientists will be scraped from sites like either 'https://www.computerscience degreehub.com/30-most-influential-computer-scientists-alive-today/' this or a Wikipedia entry on something similar. Then will read those names into a list. The challenge would be generated by grabbing a random name from that list and running it through an online hash generator, providing all the information you need for the challenge (see Figure 3-14).

The more complex use of web scraping would involve using the Bing API and the Microsoft cognitive engine. The script would ask the Bing API for 100 cryptographers, it would then store images of them in a list, returns one at random and sends it to the Microsoft cognitive engine to see if it recognised it. This script would be used to ask the users if they recognise the famous cryptographers and check their answer against the answer returned from the Microsoft cognitive engine (see Figure 3-15).

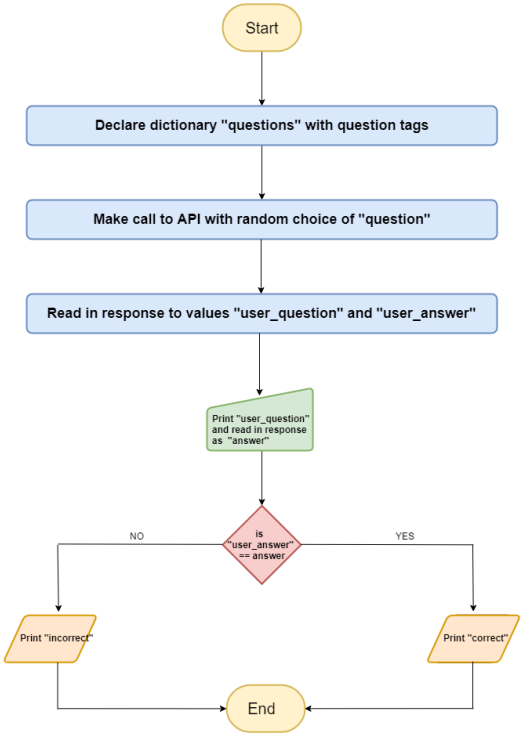


Figure 3-13: API request flow chart

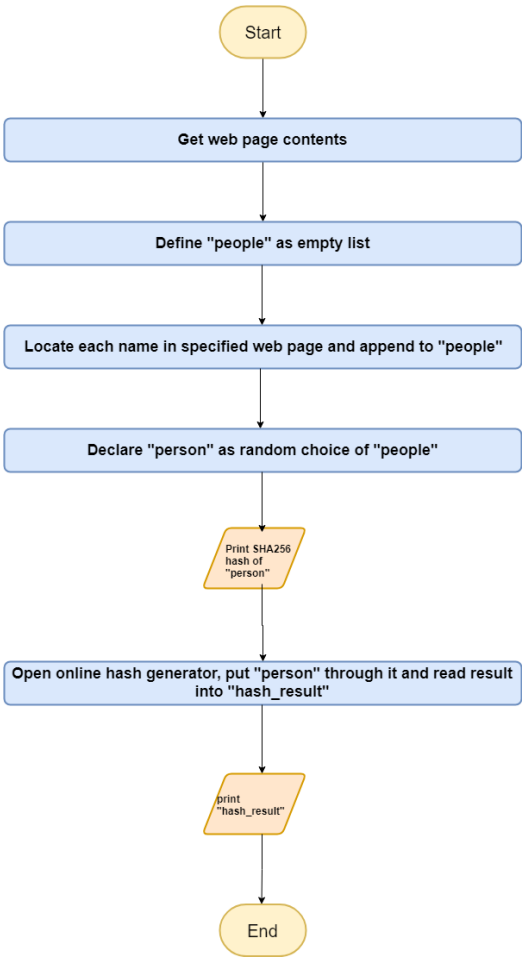
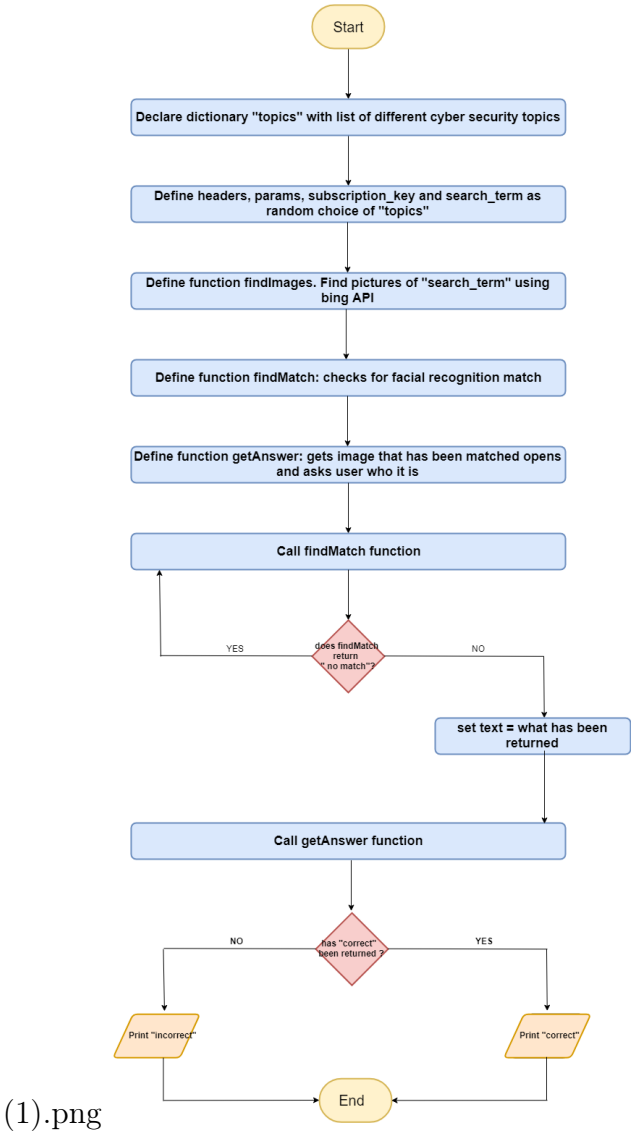


Figure 3-14: Web scraper flow chart



(1).png

Figure 3-15: Face Quiz flow chart

So the script would select a random search term from a list of possible topics. It would then make a call to the Bing API which will return 100 images of the search term. One of these images will be selected at random and be sent to the Face API, the Face API sends the response back in a JSON format. This will loop until a match is found and the Face API matches a name to the face when a match has been found the image is displayed and the user will be asked who it is (see Figure 3-16).

Docker

Docker is a way of systematically automate the faster deployment of ap-

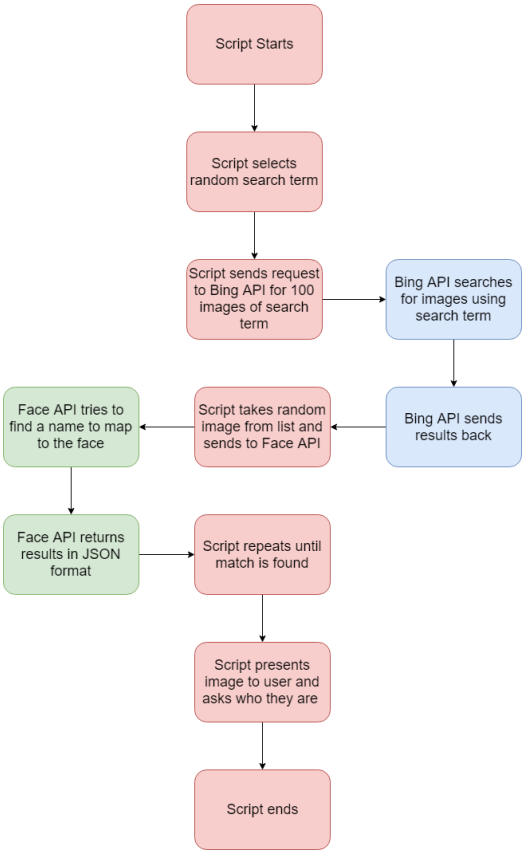


Figure 3-16: API structure

plications using portable containers. Docker containers are created using base images, images are created manually are with Dockerfiles. Dockerfiles contain instructions to be automatically performed. They are used to organise deployment artefacts and simplify the deployment process [102]. This service will be used to populate the challenge database. This will be done by making a script (similar to the Face Quiz script) that instead of opening an image and asking the user who it is being displayed, when a match is found inserts into the challenge database the "question" ("who is this ?"), the image url and the answer (the name of the person) (see Figure 3-17) This would be all that is needed to populate the challenge database (see Figure 3-18). A Dockerfile was set up to run this script and insert data into the database. Then using docker X amount of containers could be spun up and the database would be populated with X entries (see Figure 3-19). By designing it in this manner the database of challenges could be populated in a fast and efficient manner, which would also save processing time when retrieving challenges instead of having to perform the facial matches and image

searches at the time of running the script the script could then just retrieve a set of images from the challenge db.

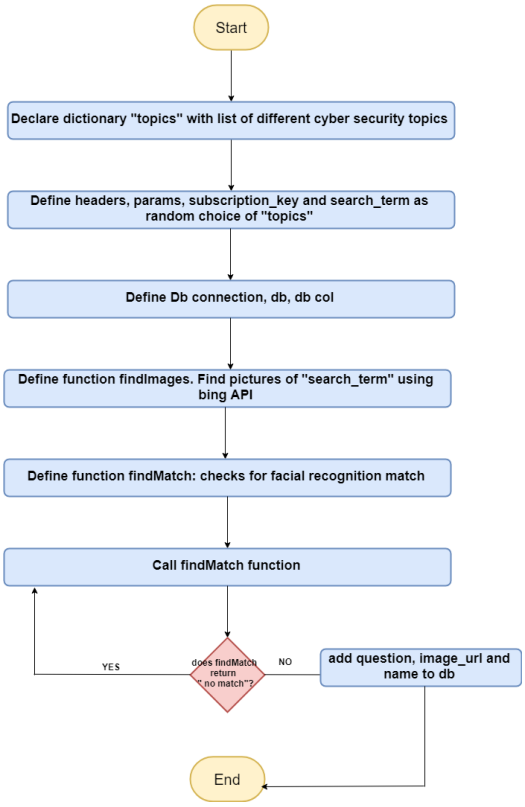


Figure 3-17: Docker face pop flow chart

3.5.3 Back end: Databases

SQL vs NoSQL First off when designing the back end of an application what Database would be most efficient to use for the applications specific needs and potential uses. As this application would ideally be scalable and have data being inserted and taken out at high speeds if it was scaled, it made sense to design choose to use NoSQL over SQL with NoSQL growing in popularity because of its ease of access, speed ,and scalability [103]. Database management is very important when considering scaling because a lot of the load comes from complex queries making results take longer, so techniques such as caching complex query results and effective organising of db tables with keys and index can be effective in reducing the load.

User Database The user Database would be built to service pupils to make an account. So they can register at the beginning and have a score

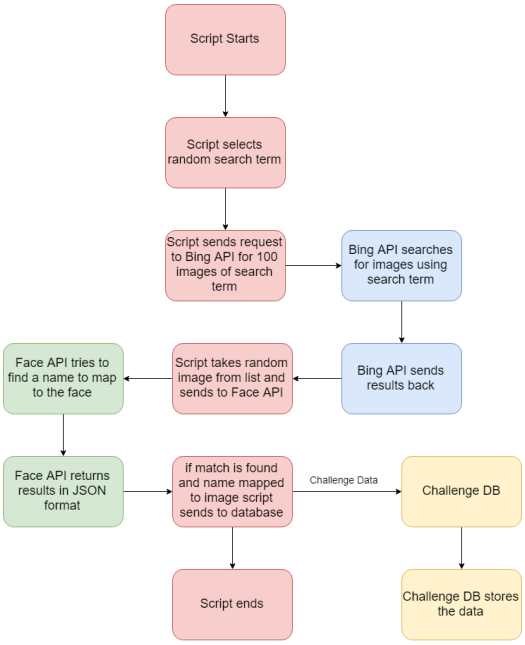


Figure 3-18: Database population structure

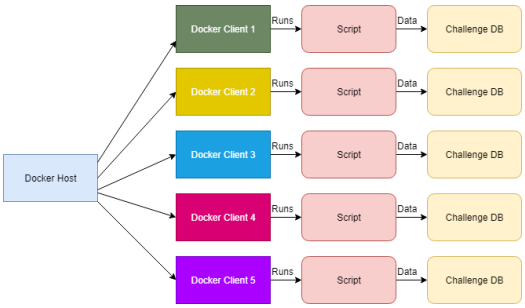


Figure 3-19: Docker Container example

associated with it. Each user would have a username which would be stored in plain text and a password that would be stored as a hash, then a salt that would be used on the hash would be stored as well. When the user inserts a password, it would be hashed with this salt and compared against the stored password hash. The user would have a unique id so they can be identified and found in the database, and would have a table that stores account data such as join date, challenges complete and high scores the last two of which would be used to measure the performance of users (see Figure 3-20).

Challenges Database The challenge database would be used to store each question and answer pair with a challenge Id to identify the challenge and a challenge type to group them by. The challenge database will be

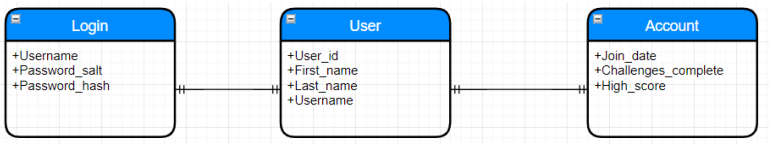


Figure 3-20: User Database

populated using a chosen method and will be designed later in this chapter, the idea is when the database has been populated the application could take a random challenge from the database (see Figure 3-21)

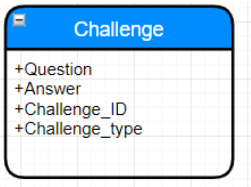


Figure 3-21: Challenge Database

Results Database The Results Database would be used to match a user to more statistics such as total score earned, success rate, hints taken and challenges complete. This would be more used for performance analysis by teachers or supervisors to gauge the effectiveness of the application and how well certain students are performing (see Figure 3-22)

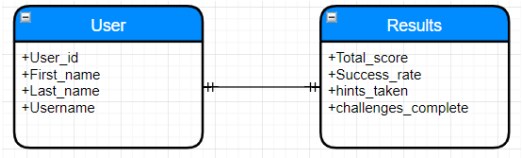


Figure 3-22: Result Database

3.6 Design Methodology

With the application theoretically designed and elements of the application discussed this dissertation will now look at how these prototyped elements will be tested and evaluated after they have been implemented to get useful data out of this project. To do this 3 different aspects of my implemented elements will be tested.

3.6.1 How many times does the same question come up in the same run?

This will be testing the effectiveness of the implemented dynamic challenge generation methods and how effective each method is for randomly generating a random question. This will also answer the question of what method is superior for this intended purpose, or has the most potential for this use.

To do this, for each script that has been made for each of the dynamic challenge generation methods, a separate version for testing that, runs through the script 100 times collecting the questions and answer pair each time then after it has run 100 times it will look at the list of questions and answers and find how many times the same pair occurred in the run, will be made. By doing this it gives a numerical value for comparison across all of the methods and helps gauge how effective that particular method is for this chosen purpose and therefore it's potential.

3.6.2 How many out of 100 images are faces?

This test will be testing the effectiveness of the Bing API that has been chosen to use in the implementation of one of the dynamic challenge methods and for use in populating a database with challenges. By testing this it will tell us how useful this technology is and if it can be utilised for this purpose effectively.

To test this a test case of the script will be made that uses the Bing API to collect faces for the challenge generation. In this proposed script a random topic such as 'famous computer scientists' or 'famous cryptographers' and then that is used as a search term for the Bing API which returns 100 images of that. For the purposes of this dissertation, how many of these 100 images are actually contain faces, will be tested or more accurately how many actually contain people. This will be tested by using search terms with a varying likely hood to return faces, for example, 'Famous movie stars' is very likely to return a set of actors as they by nature are well known but a search term of 'famous cryptographers' on the other hand isn't as well known by nature. Doing this will not only evaluate how useful the Bing API if for general use but analysing different topics it can also be seen how useful it is for the specific proposed purpose

3.6.3 Out of 100 computer scientists, how many does the face API recognise?

This will test how effective the Microsoft cognitive engine Face API, used in my proposed web scraping face quiz script, is at recognising faces. This could

be a very interesting test to carry out and the data gathered from doing this would not only be useful in learning how effective it would be to use this technology for the proposed purpose but for analysing exactly how good this technology is and help determine how it is working.

To achieve this a separate script all together will be made for this test. A list of 100 computer scientists will be gathered and separated into three levels 'world famous' e.g Bill Gates, Steve Jobs, 'industry famous' and 'well known'. Multiple images of each person on the list will then be gathered and 3 runs of each will be done of the images. Outputted will be how many of the 3 images were recognised or a 'recognition score' and how many times out of the 3 runs that API recognised the person or a 'consistency score', this will be outputted in a graphical format such as a csv file. Also featured in the tests will be some 'dud' images of for example people who look like Steve jobs but actually are not to test how accurate the service is being. As mentioned before it is my hope that these tests will determine how useful the Face API is and help pinpoint its behaviour.

3.7 Conclusions

This chapter of the dissertation covered the design of the proposed application with heavy reference to the literature a theoretical design was mocked up oh how the application would look and would be structured as well as what the lesson content would be and how they would be distributed. Following this the chapter discussed the technical elements that would be involved with designing such an application mocking up user interface and an in-depth design of the dynamic challenge generation, discussing different methods that could be used and designing the implementation of those methods before finally talking about how the back end would be set up and designing a way of populating this with challenges. This chapter is then concluded with a discussion about how these features could be tested for their potential and performance after they have been implemented.

4 Implementation

4.1 Introduction

After the research had been done and a theoretical design for what the application would look like and how it would work was completed it was time to start the implementation stage of this dissertation. It was then decided for the implementation to prototype certain parts of my theoretical design. The dynamic challenge generation part of the project was chosen to be prototyped as this was an important part of the learning experience to make sure that a different challenge could be given to a user each time they used the application, or if it was used in a class to ensure that all the users had different experience's using the application and couldn't copy other users answers out of laziness. Dynamic challenge generation was also an interesting area to look at and would be interesting to evaluate the effectiveness of the different approaches in creating a different challenge each time. Another part that was prototyped was the docker element of the project as docker is an incredibly useful service which would benefit the project greatly if made and prototyping an element such as database population would help illustrate that as well as gauge how efficient it would be.

For the implementation stage Python was used for the programming language and Eclipse as the IDE. Python was chosen out of a mix of personal preference as some experience had been picked up leading up to this project and had more than any of the other programming languages as well as the practicality of using it for this project as python has extensive support libraries, useful for completing the variety of tasks that this dissertation proposed to implement, combined with the fact that it is a widely used language and recognised in the industry it seemed like a sensible choice. Eclipse was chosen to use due to exposure with it and for the purposes of this dissertation thought it better to use this rather than learn how to use a new IDE.

The Eclipse IDE will be used to develop python scripts for the different methods of dynamic challenge generation, at least a script for each. A combination of docker and python was then used to create a service which populates a database with challenges in an efficient matter. The aim of this implementation stage is to develop something that in a later stage will give an understanding of which method of dynamic challenge generation is the most effective and how to best distribute this method.

4.2 Starting simple: From selection

random_selection.py

This script was designed to be the simplest method of dynamically generating a challenge, by just selecting from a list of question and answer pairs (see Figure 4-1) takes care of that. First off a dictionary of "caesar questions" is defined, with a key and value of the question and answer. A variable of question is then defined as a random choice of that dictionary and is printed. A for loop then grabs all of the answers, correct and incorrect, and prints them before reading in the user's response to an answer variable.

```
1  import random
2
3  caeser_questions = {
4      'Which answer is crypto put through a caeser cipher  ': 'jyfwav',
5      'Which answer is escape put through a caeser cipher  ': 'lzjhwl',
6      'Which answer is hacker put through a caeser cipher  ': 'ohjrly',
7      'Which answer is coding put through a caeser cipher  ': 'jvkpun'}
8
9  question = (random.choice(list(caeser_questions)))
10 print(question)
11
12 for key, value in caeser_questions.items():
13     print ("---" + value)
14
15 answer = input("Enter answer here:")
16
```

Figure 4-1: Question selection

A function is then defined that loops through all of the items in "caesar questions" and compares the answer given by the user to the answer pair to all of the questions. If a match is found then the value "correctAnswers" is set to the corresponding question to the answer given, otherwise it remains empty. Finally, the "answerChecked" value is set to what has been returned by that function, if this is equal to the question given at the start "correct" is printed if not "incorrect" is printed (see Figure 4-2).

4.3 Using API request

This script was created to implement the API method of dynamic challenge generation. In practice, it would be more useful to do this with an actual website or service designed for this use but for test purposes a script to represent the API being called to for the answers was made.

```
17  def checkAnswer(caeser_questions, answer):
18      correctAnswers = ""
19      listOfItems = caeser_questions.items()
20      for item  in listOfItems:
21          if item[1] == answer:
22              correctAnswers = item[0]
23      return  correctAnswers
24
25  answerChecked = checkAnswer(caeser_questions, answer)
26
27
28  if (answerChecked == question):
29      print ("correct")
30  else:
31      print ("incorrect")
```

Figure 4-2: Answer Check

API.py

First, the API that was to be called for a question was implemented. An API that when ran would run on localhost using python Flask was set up. A route for the request to be sent and read in a string that will represent the question to be sent back was then set. depending on the value of the question string the API selects a question-answer pair and returns it (see Figure 4-3).

```
17  @app.route('/question/<string:question>', methods=['GET'])
18  def get_question(question):
19      if (question == "fred"):
20          return jsonify({"question":" 'mylk'  is what word put through a caesar cipher ?", "answer": "fred"})
21      if (question == "bob"):
22          return jsonify({"question":" 'ivi'   is what word put through a caesar cipher ?", "answer": "bob"})
23      if (question == "alice"):
24          return jsonify({"question":" 'hspjl'   is what word put through a caesar cipher ?", "answer": "alice"})
25      if (question == "eve"):
26          return jsonify({"question":" 'lcl'   is what word put through a caesar cipher ?", "answer": "eve"})
```

Figure 4-3: API

API_request.py Next up a script that would make a call to this API to get a question shown in (see Figure 4-4) was implemented. First off, a

set of tags to be sent are defined in "questions" and "question" is set to a random choice of them. a GET request is then sent to the API and the response is read in, each value contained within the json response is also read into "values", the "user question" and "user answer" variables are set to the question contained in the response and the answer contained in the response respectively, the question is then asked to the user and their answer compared to the answer.

request.PNG

```
1  import requests
2  import json
3  import random
4
5  questions = ['fred', 'bob', 'alice', 'eve']
6  question = (random.choice(questions))
7  values = []
8
9  resp = requests.get('http://127.0.0.1:5000/question/' + question)
10 json_data = json.loads(resp.text)
11 if resp.status_code != 200:
12     print ("something went wrong")
13 for key, value in json_data.items():
14     values.append(value)
15 user_question = values[1]
16 user_answer = values[0]
17 txt = input(user_question)
18 if (txt == user_answer):
19     print ("correct !")
20 else:
21     print ("incorrect")
```

Figure 4-4: API Request

4.4 Web scraping

4.4.1 Basic web scraping

For the more basic attempt at a web scarper a basic website was found listing off "30 influential computer scientists" and used it to pick them off and use them to generate questions. a hardcoded hash method is shown and then a method of opening an online hash generator and putting the text through that as well.

Web_scraper_final.py For this web scraping application the BeautifulSoup library was used for the opening of the website and getting names and the selenium library for opening open a browser and putting the text through an online hash generator. First off (see Figure 4-5) the source was set to the website that was chosen to be used and defined an empty list "people". Then the content returned from the web page is taken and sifted through to find what is wanted. The web page is inspected and the headers were found that contained the information that was wanted to be picked out (the names) so the data returned from the source was split accordingly. After the text was almost down to the strings that were wanted it was filtered further and appended the filtered string to a list of people before "popping" some excess items that came after the desired list of people. A random "person" is selected from the list of "people" found from the website and after this, a SHA256 hash is generated of the person from the list and displayed.

```
8  source1 = requests.get('https://www.computersciencedegreehub.com/30-most-influential-computer-scientists-alive-today/').text
9  soup1 = BeautifulSoup(source1, 'lxml')
10
11 people = []
12
13 for person in soup1.find_all('h3'):
14     computer_person = str(person).split('<h3>')
15     computer_person_name = str(computer_person).split('</h3>')
16     c = " ".join(re.findall("[a-zA-Z]+", str(computer_person_name)))
17     people.append(c)
```

Figure 4-5: Web scraping the names

A web page that calculates md5 hashes of a value that is typed in is opened (see Figure 4-6). By inspecting the web page the elements needed were identified such as the text box to enter the value to be hashed, which was set as "hashField", also identified was the area which the calculated hash is displayed and set that as "hashResult". The "person" that was chosen was sent to the text field "hashField" and once the value was entered read the value of what was in the returned hash field to "hashResult". This value was then printed along with the original person.

```
from selenium.webdriver.chrome.options import Options
options = Options()
options.binary_location = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
browser = webdriver.Chrome(options=options, executable_path=r"C:\Program Files (x86)\Google\Chrome\Application\chromedriver.exe", )
browser.get("https://passwordsgenerator.net/md5-hash-generator/")
hashField = browser.find_element_by_id('txt1')
hashField.send_keys(person)
hashResult = browser.find_element_by_id('txt2').get_attribute("value")
#hashField.send_keys(hashResult)
print (person + ' ' + hashResult)
```

Figure 4-6: Getting hash through browser

4.4.2 Web scraping with facial recognition

For this approach, the services of Microsoft Azure were taken advantage of using two API services available from their cognitive services. The Bing search API and the Computer Vision API which uses the Microsoft cognitive engine. Both were set up on a Microsoft Azure account that was created (see Figure 4-7) and used them to generate the keys and the links that were needed to make my API calls in the script. This script will use the Bing search API to find 100 images of a search term then use the Vision API to recognize a face contained in an image from the search, if it matches a face to the name then it displays to the user and asks who they are.

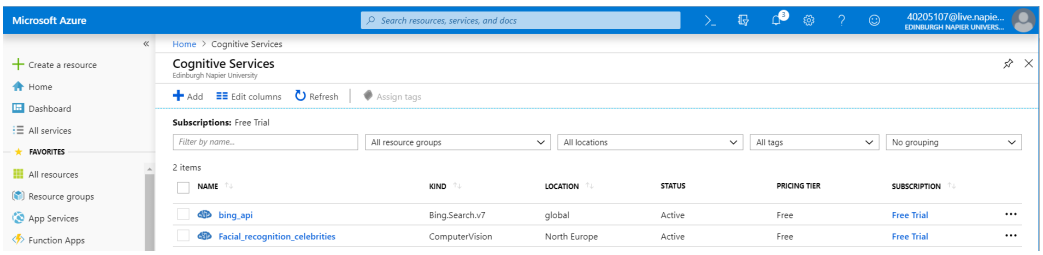


Figure 4-7: Microsoft Azure Cognitive Services

Face_quiz_final.py

At the start of the script variables are declared such as a list of random cyber security topics to be chosen from such as "famous computer scientists" or "famous cryptographers" , a search term to be used for the Bing API which is a random choice of those topics and other variables used for the APIs such as subscription keys and headers and params etc. Then the findImages function is defined (see Figure 4-8). In this function the API client is set up using the credentials set, then call the API with the search term decided and a count of 100 set as well as a filter of face applied. A random image is then chosen and the image url is returned.

Next up the findMatch function was defined (see Figures 4-9 4-10). First

```
38  def findImages (subscription_key, search_term):
39
40      client = ImageSearchAPI(CognitiveServicesCredentials(subscription_key))
41      image_results = client.images.search(query=search_term,count=100,ImageContent=face)
42
43      if image_results.value:
44          first_image_result = random.choice(image_results.value)
45          return(first_image_result.content_url);
46
47      else:
48          print("No image results returned!")
```

Figure 4-8: findImages function

the search term to be used within the function is declared and the findImages function is called with that search term, which as discussed, returns an image url that is concatenated in the "body" variable used to send a request to the vision API. The connection to the vision API is set up and a request is sent using that image url as the body. What gets returned is a json response which is broken down into data and is used in the next part to determine whether a match has been found or not. This is done handling certain errors which would mean a match has not been found. Multiple runs were done and caught the different errors that were happening if a match wasn't found. For example, if the categories or celebrities bracket was empty it meant that there was no match found for the face and when a name value was attempted to be read in an error would happen, with these errors that mean no match had been found the string "no match" is returned at their occurrence. Other exceptions to note was sometimes a match is found but when the image url was trying to be accessed the website denied permission and sometimes it would return an image of multiple people which obviously isn't helpful when trying to ask a user who they are as it could be confusing as to which person the question would be referring to, this was solved by accessing the classification parameter and only accepting pictures that were either of a person or a portrait of a person (dropping images classified as group images). These exceptions were handled and also returned "no match" if they occurred. If none of these exceptions occur then it means a match has been found and a name has been able to be matched to the face so the image url is returned.

After this, the getAnswer function is defined (see Figure 4-11). Firstly the vision API was set up to be used for the question, then sends an image to it, the vision API responds and the name is taken from the response and stored in a variable "name", the certainty score is also taken and rounded up

```
def findMatch ():
    try:
        search_term = random.choice(topics)
        text = findImages(subscription_key, search_term)
        body = "{'url':\'"+text+"\'}"

        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s" % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)

        if (str(data).__contains__('"celebrities":[]') or str(data).__contains__('"categories":[]')):
            #print ("catching empties has worked")
            return("no match");
        else:
            name = (d1['categories'][0]['detail']['celebrities'][0]['name'])
            score = (d1['categories'][0]['detail']['celebrities'][0]['confidence'])
            classification = (d1['categories'][0]['name'])
        try:
            urllib.request.urlretrieve(text, r"C:\Users\MAXBO\Documents\Test_images\file1.jpg")
            os.chdir(r'C:\Users\MAXBO\Documents\Test_images')
        except urllib.error.HTTPError:
            #print("http error handling has worked")
            return("no match")
```

Figure 4-9: findMatch function pt 1

```
        if name == "":
            #print("name is empty so failed")
            return("no match");
        elif score == "":
            #print("score is empty so failed")
            return("no match");
        elif (classification == "people_" or classification == "people_portrait"):
            #print (classification + " has passed the classification test and is therefore equal to people_ or people_portrait")
            return (text);
        else:
            #print (classification + " has failed the classification test and is therefore not equal to people_ or people_portrait")
            return("no match");

    except KeyError:
        #print("Failed because of exeption")
        return("no match");
```

Figure 4-10: findMatch function pt 2

and stored in a variable. Then the image is opened and the user is asked who it is, if their answer matches the name determined from the API "correct" is returned if not "incorrect" is returned.

```
def getAnswer():
    try:
        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s" % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)
        #if (str(data).__contains__('"celebrities":[]' or 'categories":[]')):
        score = (d1['categories'][0]['detail']['celebrities'][0]['confidence'])
        percentage = (score * 100)
        rounded_percentage = round(percentage,2)
        name = (d1['categories'][0]['detail']['celebrities'][0]['name'])
        urllib.request.urlretrieve(text, r"C:\Users\MAXBO\Documents\Test_images\file1.jpg")
        os.chdir(r'C:\Users\MAXBO\Documents\Test_images')
        im = Image.open(r"C:\Users\MAXBO\Documents\Test_images\file1.jpg")
        im.show()
        answer = input("Who is this ?")
        if (answer.lower() == name.lower()):
            return("Correct, I am " + str(rounded_percentage) + " the correct answer is " + name);
        else:
            return("Incorrect, I am " + str(rounded_percentage) + " the correct answer was " + name);

    except KeyError:
        return("Error has occured. Try again.")
```

Figure 4-11: getAnswer function

A variable of correct answers is set to 0. The finally inside a for loop that iterates 5 times, is a while loop that calls the findMatch function setting

the variable "text" to the value it returns. The while loop breaks when the value returned is not equal to "no match", so essentially, if the image has not triggered any of the exceptions and a name has been mapped to the face, when this happens "text" is set to the image url and the while loop is broken. The "body" variable is then defined with that image url and the getAnswer function is called asking the user who is in the image and determining if they are correct. If the function returns correct answer then the correct answer variable is incremented by 1. After this process has been iterated through 5 times a score out of 5 is given and the script terminates (see Figure 4-12).

```
127  correct_answers = 0
128
129  for i in range(0, 5):
130      while True:
131          text = findMatch()
132
133          if text != "no match":
134              break
135
136      body = "{'url':\'"+text+"\'}"
137
138      print("Person " + str(i+1) + ":" )
139      answer = getAnswer()
140      if answer == "correct":
141          correct_answers = correct_answers + 1
142      print(answer)
143
144  print("You scored " + str(correct_answers) + "/5")
145
```

Figure 4-12: Asks user question and iterates

4.5 Challenge distribution with Docker

4.5.1 Python

This python script was adapted from the face quiz script but with the use of the pymongo library used to insert challenge data into a MongoDb database.

"Docker_face_pop_final.py"

The Docker face population script is very similar to the Face Quiz script discussed in the previous section. The only difference is that it doesn't have the getAnswer function as there is no user interaction and is ,therefore, unnecessary to ask the user the questions and iterate through the process 5 times at the end. The other difference is the use of the pymongo library which is used to when the findMatch function determines there has been a match on the face, takes the "question" that would be asked, the image url for the selected image and the "name" of the person in the image (or the answer) and inserts it into the challenge Database (see Figure 4-14). In the code (see Figure 4-13) the "db" referred to as the host on line 43 is established in the docker compose file and is discussed later.

```
43  myclient = pymongo.MongoClient(host='db', port=27017)
44  #myclient = pymongo.MongoClient(os.environ['DB_PORT_27017_TCP_ADDR'], 27017)
45  mydb = myclient["mydatabase"]
46  mycol = mydb["challenges"]
```

Figure 4-13: Establishing Database connection

```
93          elif (classification == "people_" or classification == "people_portrait"):
94              print (classification + " has passed the classification test and is therefore equal to people_ or people_portrait")
95              mydict = { "question": "Who is this ?", "image":text, "name":name}
96              x = mycol.insert_one(mydict)
97              print("Data has been inserted to database")
98              return (text);
```

Figure 4-14: Inserting data into the database

4.5.2 Docker

These are the files used to create Docker images and spin up the containers with my script running inside them so the database can be populated. **requirements.txt** This file is essentially just a list of all the external libraries used in the script that the machine will be running. The Dockerfile uses this file to pip install everything to contained inside (see Figure 4-15).

```
1  Flask
2  azure-cognitiveservices-search-imagesearch
3  pillow
4  cognitive_face
5  pymongo
```

Figure 4-15: Requirements.txt contents

Dockerfile The Dockerfile is the instructions that are needed to build the docker image that will be spun up in containers. This included installing the requirements (as discussed in the previous section), copying the python script (docker_face_pop.py), exposing the port so it can contact the database and finally running (see Figure 4-16).

```
1  FROM python:3-onbuild
2  RUN pip install --trusted-host pypi.python.org -r requirements.txt
3  COPY docker_face_pop.py /docker_face_pop.py
4  #RUN pip install .
5  EXPOSE 27017
6  CMD ["python", "/docker_face_pop.py"]
```

Figure 4-16: Dockerfile contents

docker-compose.yml

Docker compose is a tool for defining and running multi-container applications which can be used to configure the application. After this has been done all services can be created and started with a single command. Docker compose was used to establish the services of "db" (this is the db referred to in the docker face population script) which is a mongoDB image that was downloaded off of Docker Hub (where you can download from a library of images) and expose the correct ports and set the volume. Then the "client" is defined as using the honours-project image (the image created using my requirements.txt and Dockerfile) and link it to the db service so data from the client service can be inserted into the database from the db service. The port is opened and then the environment that will be worked with (which database is going to be used etc) is established (see Figure 4-17).

```yaml
version: '3.3'

services:
  # Name our service will be known by
  db:

    # version of mongo we'll use
    image: mongo

    ports:
      - 27017:27017

    # using a named volume
    volumes:
      - devmongo:/data/db

  client:

    image: honours-project
    links:
        - db
    depends_on:
        - db
    ports:
        - 27017
    environment:
        - MONGO_URI=mongodb:27017/mydatabase

volumes:
  devmongo:
```

Figure 4-17: docker-compose.yml contents

Docker implementation With all of the previously discussed files implemented, Docker can be set up and containers can be spin up each executing the population script and inserting data into the database (see Figure 4-18).

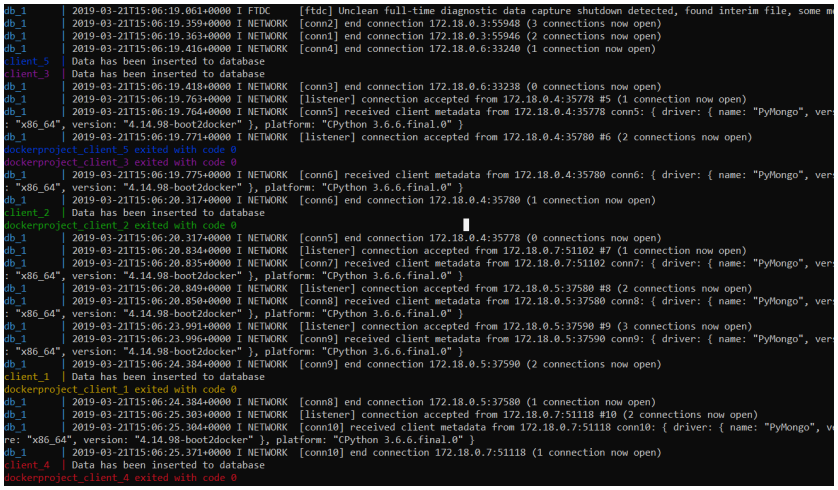


Figure 4-18: Sample Docker output with 5 containers

4.6 Conclusions

This chapter of the dissertation has shown how the four methods of dynamic challenge generation methods were implemented using python scripts as well as the implementation of a database population technique using Docker. The methods were implemented with a varying degree of complexity (with the random selection method being implemented with a limited selection of challenges, whereas the web scraping with API was implemented selecting a random image from 100 returned using a search term which itself was randomly chosen from a list of search terms) with the ultimate goal to measure the potential of the method rather than the performance of the individual script. The web scraping with API method used both the Bing image search API and Vision API which were authenticated using a Microsoft Azure account created for this project, this gives the potential for this project to not only measure the performance of the implemented challenge generation methods but to measure the performance of the utilised APIs and their potential use for this purpose also.

5 Results and Evaluation

5.1 Introduction

With the methods being implemented their potential could now be measured, along with the performance of the APIs taken advantage of in one of the methods. It was decided in the Design chapter that the three tests that would be "Test 1: how many times does the same question come up in the same run", "Test 2: How many out of 100 images are faces ?" and "Test 3: Out of 100 Computer scientists, how many does the face API recognise?" this chapter contains the results of these tests which were carried out by implementing test case scripts and discusses the results evaluating what they indicate.

5.2 Test 1: how many times does the same question come up in the same run?

5.2.1 Making the test cases

To perform this test on the different dynamic challenge methods altercations were made to the code of each of the methods so it would return the data wanted for the purposes of this test and this dissertation (see Figure 5-1) this particular example is taken from the random selection method but the code added to them all was pretty similar, just minor changes depending on how the questions were generated in each of them. These changes consisted of making a list "questions" which each question was added to as the script iterates through the process. "unique_question" is another list declared which will be added to every time a question appears and hasn't yet been appended to "questions" (if this is the first time the question appears and is therefore unique). "occurrence _count" is added to every time a question is retrieved that is in "questions" and therefore has already been asked. it then iterates through the question asking process 100 times. After this "ocount" is declared which is used to add the total number of occurrences for each unique question, this is used at the end of the script to calculate the average occurrence of a question in 100 runs. Also printed out is how many times each question showed up in that run and how many appearances of the same question there was in that run (see Figure 5-2).

```
questions = []
unique_questions = []
occurence_count = []


for i in range(0, 100):
    question = (random.choice(list(caeser_questions)))
    if questions.__contains__(question):
        occurence_count.append(question)
    else:
        unique_questions.append(question)
    questions.append(question)

print (len(questions))
print (len(unique_questions))

ocount = []

for uquestion in unique_questions:
    count = questions.count(uquestion)
    print('This question ' + uquestion + ' appeared  ' + str(count) + ' times')
    ocount.append(count)
print("There was a total of " + str(len(occurence_count)) + " Same occurences of a question")
print (str(average(ocount)))
```

Figure 5-1: Example of changes made to produce test results

```
This question Which answer is crypto put through a caeser cipher   appeared  31 times
This question Which answer is coding put through a caeser cipher   appeared  34 times
This question Which answer is escape put through a caeser cipher   appeared  17 times
This question Which answer is hacker put through a caeser cipher   appeared  18 times
There was a total of 96 Same occurences of a question
25.0
```

Figure 5-2: Example output of test cases

5.2.2 The Results

First data was collected and a graph was made illustrating the average question occurrence in 100 runs. So for each of the unique questions asked how many times on average was it asked across the 100 runs (see Figure 5-3). The Random selection and API request was 25 times, web scraping was 3.44 times and web scraping with web scraping with API was 1.51 times.

Next, the number of times that a question that had already been asked was asked was taken for each of the methods (see Figure 5-4). Random selection and API request methods had 96 occurrences of the same question, the web scraping method had 71 and the web scraping method using APIs had 34.

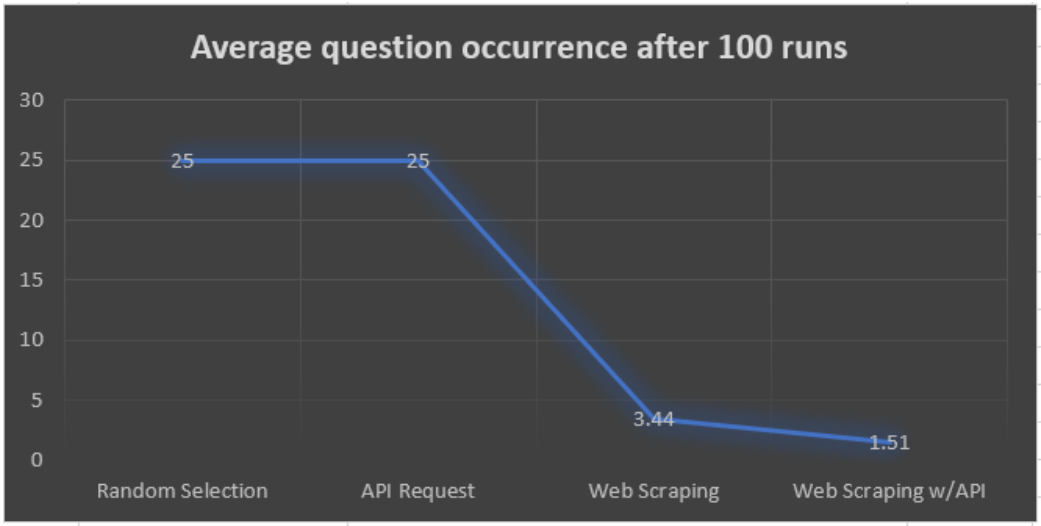


Figure 5-3: Average Question Occurrence

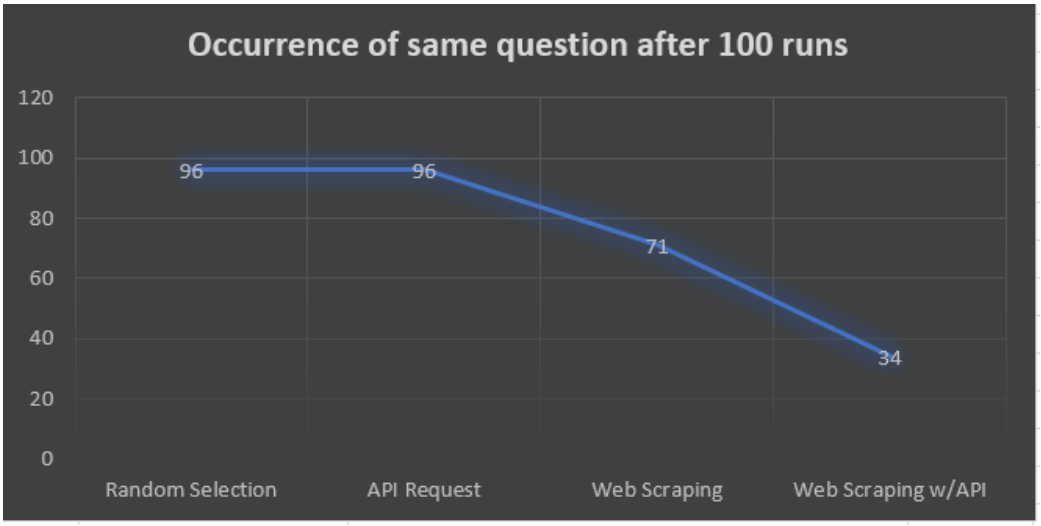


Figure 5-4: Same Question Occurrence

Finally both of the results previous tests were taken plotted on a graph (see Figure 5-5) to show the correlation.

5.2.3 Evaluation of Results

Test 1 was evaluating the performance of the different methods of dynamic challenge generation. When discussing these results and the ones soon to follow it is important to acknowledge that the implementation of both the

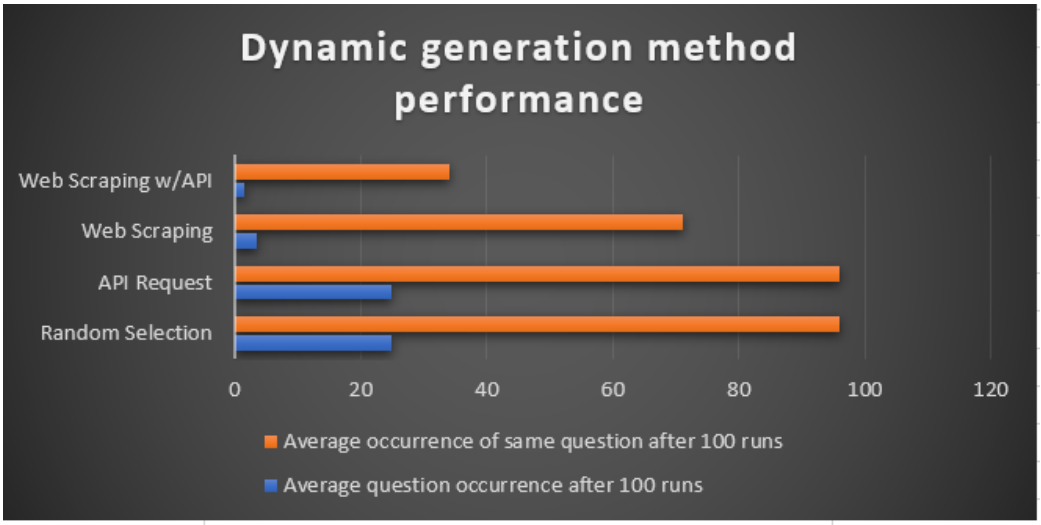


Figure 5-5: Dynamic Generation Method Results Correlation

API request and random selection methods were both very simple. With a selection of 4 questions to choose from the performance was never going to be very high when it comes to dynamic challenge generation. Adding another question to the selection would slightly increase the performance and so on and so on. This could go on forever but for these methods would require a lot of hard work and hard coding in potential challenges. So when looking at these numbers instead of evaluating these results for these specific scripts it should be evaluated as the potential this method carries to be scaled for the intended use based on the results received.

The first set of data collected was how many times on average, per unique question, did it show up in 100 runs. Random selection and API request methods both had 25 whereas web scraping only had 3.44 and web scraping with the use of API had 1.51 times, meaning that in 100 runs the average question only came up 1.51 times. This is significantly better than both the API request and random selection methods and over 2x better than the web scraping method. These results show that the web scraping with API method has a lot more potential and has a proclivity to generate a new unseen question, more so than any of the other methods.

The second set of data collected showed how many times in 100 runs did an occurrence of the same question happen. Random Selection and API request methods both had 96 occurrences of the same question, web scraping had 71 and Web scraping with API had 34. Again web scraping with API method is in front outperforming all of the other methods of dynamic challenge generation. With 34 same occurrences of the same question in 100

runs that means 66 unique challenges were generated in 100 runs (29 for web scraping and 4 for random selection and API request) this is an impressive number and shows the potential this method has.

Overall this test has shown that the use of the web scraping with API method has more potential for the use of dynamic challenge generation with it outperforming the other 3 by quite a bit. As discussed the implementation of this should be considered but even if more challenges were added to the first two methods you would have to add a lot and it would take a lot of time to achieve the same sort of numbers being returned from web scraping with API. Web scraping as a method as a whole has great potential and is superior both to random selection and API request methods for this purpose ,however, the use of APIs to aide this method gives it an edge and increases the performance. The APIs also do a lot of the heavy lifting in that they return images, or facial matches and you don't have to manually mine the elements you need as with web scraping on its own.

5.3 Test 2: How many out of 100 images are faces?

.

5.3.1 Making the test case

For this test, a test script was made which called the Bing API and asked for 100 images with the face tag with four different search terms - "famous people", "famous actors", "famous computer scientists" and "famous cryptographers". The idea is to test the Bing APIs ability to accurately and consistently produce pictures of faces, using different search terms increasing in specificity to gauge how this affects it. A test case script was made for the evaluation of this simply taking a search term and calling the Bing API to return one hundred images with the face tag. The selenium library was then used to iterate through the content url of each of the 100 images returned and open them in a browser. This allowed me to quickly scan through the images looking for ones that did not contain a face. This was done for each search term changing the variable "search_term" accordingly each time (see Figure 5-6).

5.3.2 The Results

The results were collected and each occurrence of "nonface" images in the 100 returned by the Bing API was counted for each search term 3 times to ensure accurate results. Both the "famous people" and "famous movie

```
subscription_key = "71ad32482bbb4d99a0882c34d8b08d9f"
search_term = "famous computer scientists"


client = ImageSearchAPI(CognitiveServicesCredentials(subscription_key))
image_results = client.images.search(query=search_term,count=100,ImageContent=face)


from selenium.webdriver.chrome.options import Options
options = Options()
options.binary_location = "C:\Program Files (x86)\Google\Chrome\Application\chrome.exe"
browser = webdriver.Chrome(options=options, executable_path=r"C:\Program Files (x86)\Google\Chrome\Application\chromedriver.exe", )

if image_results.value:
    for i in range(0,100):
        first_image_result = image_results.value[i]
        browser.get(first_image_result.content_url)
    else:
        print("No image results returned!")
```

Figure 5-6: Code used for Bing API test case

stars" returned 100 out of 100 images containing faces. "famous computer scientists" returned 93 out of 100 images containing faces and "famous cryptographers" returned 37 out of 100 images containing faces (see Figure 5-7).

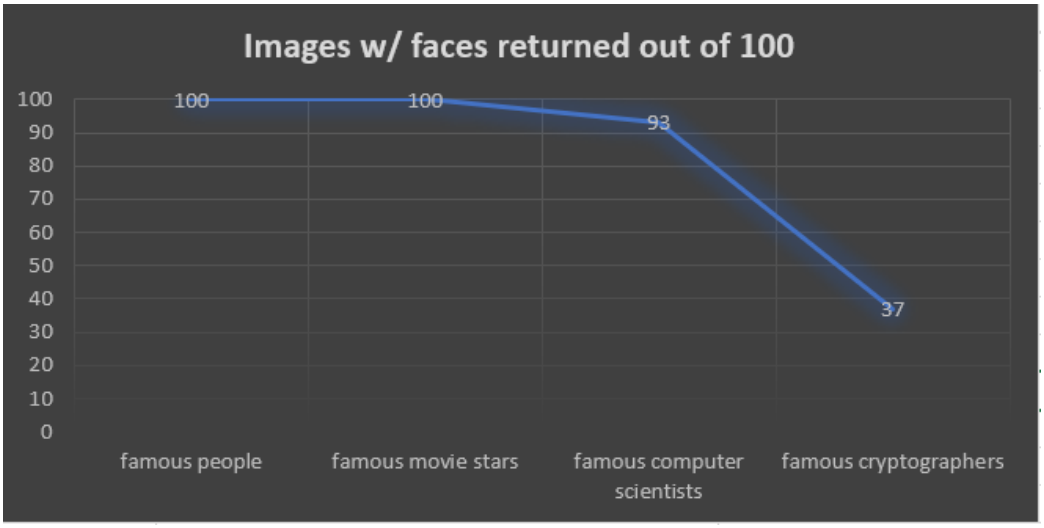


Figure 5-7: Bing API performance

5.3.3 Evaluation of Results

Test 2 was evaluating the performance of the Bing API used in the implementation of my web scraping with API dynamic challenge generation method. The Bing API has a query parameter called ImageContent which you can use to specify you want faces to be returned, this was done for all the runs.

The data collected for this test was measuring how many images out of the 100 that were returned, contained faces (and therefore could be used for the intended purpose more effectively). This was done using four different search terms "famous people", "famous movie stars", "famous computer scientists" and "famous cryptographers" as discussed previously the idea was to gauge how effective the Bing API was at returning faces the more specific the search term got, with "famous people" and "famous movie stars" likely to return very well known people and "famous computer scientists" and "famous cryptographers" with being more specific search terms they have fewer possibilities to collect from, this is important to evaluate as with the intended purpose these specific search terms would be getting used.

The "famous people" search term returned 100 out of 100 images containing faces, as did "famous movie stars", "famous computer scientists" returned 93 out of 100 images containing faces and "famous cryptographers" only returned 37 out of 100 images containing faces. This shows a dramatic decrease in the number of images containing faces returned when the search term gets as specific as "famous cryptographers". The images returned were of graphs explaining cryptography terms, pictures of books, pictures of keys etc essentially things that are associated with cryptography but not actually famous cryptographers a lot of the time. From the context, it seems this is because that there are less big names associated with cryptography, at least in the same manner as the other search terms, and so the results that Bing collects the images from are discussing cryptography rather than a person associated with cryptography so return these images. This isn't a massive blow to using the Bing API for the intended purpose, as has been shown it can still be effective and if you use a variety of search terms with varying specificity as this dissertation has it stops the challenge generation from relying on this specific search term. However, it is important to note that the more specific the search term the less option it has to choose from for faces and therefore will take longer to return a face that has been matched and therefore if this search term is used solely will likely slow down the process.

5.4 Test 3: Out of 100 computer scientists, how many does the face API recognise?

5.4.1 Making the test case

To make the test case for the face API evaluation alterations were made to the face quiz application. The first step however to select a list of 100 computer scientists. 100 names were gathered and grouped them by fame level. 10 very famous 15 well known and 75 industry recognised, this list is available in the

appendix. After this list was decided the alteration was started by declaring three dictionaries, one for each fame bracket. In this dictionary was three image links for each person in the list, which was sourced independently rather than using the Bing API previously used in the project to ensure the data was consistent, in that they were a similar quality of photo, and accurate, in that the images were of whom they were supposed to be. Other changes included adding two functions one similar to the findMatch function used earlier except it returned the certainty score called getScore and a second which just took the digits off of the end of the names of the people (as they were numbered 1-3 so that they were different keys) called mySplit. Then for each of the three dictionaries, the following was done: Three variables were declared; "xcorrect" to count how many correct matches, "xscores" to collect the certainty scores of the matched faces and finally "xmatched" which collects the names of all of the people that have been matched. Then a for loop iterates through each of the values in the dictionary trying to find a match to the face, appending to the lists and increments the correct counter when a match is made (see Figures 5-8 5-9 5-10). After this has been done for all three dictionaries, the results are displayed giving: how many celebrities in that category were matched, the average confidence rating given with matches for that category and the total pictures that were matched (see Figure 5-11).

```
22  Very_famous = {
23   'Bill Gates 1' : 'https://specials-images.forbesimg.com/imageserve/5c76b4b84bbe6f24ad99c370/416x416.jpg?background=000000&cropX1=0&cropX2=4
24   'Bill Gates 2' : 'https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2018/07/11/105322791-1531301768595gettyimages-467620670
25   'Bill Gates 3' : 'https://bitcoinist.com/wp-content/uploads/2018/05/wiki-Bill_Gates_MSC_2017-e1525827667380.jpg',
26   'Steve Jobs  1' : 'https://fm.cnbc.com/applications/cnbc.com/resources/img/editorial/2013/02/26/100496736-steve-jobs-march-2011-getty.1910x
27   'Steve Jobs  2' : 'https://images-na.ssl-images-amazon.com/images/I/61n431BDxGL.jpg',
28   'Steve Jobs  3' : 'https://specials-images.forbesimg.com/imageserve/5b8576db31358e0429c734e3/416x416.jpg?background=000000&cropX1=211&cropX
```

Figure 5-8: Setting up of dictionaries

5.4.2 The Results

For this test fed the Face API 3 different pictures of each of the computer scientists, and did three runs of the script and averaged out the results. The first metric measured was how many people were recognised, the average was 10 out of 10 for the very famous category, 15 out of 15 for the well-known category and 58 out of 75 for the industry recognised (see Figure 5-12). This equates to a total of 83 out of 100 computer scientists being recognised.

For the next part of this test, the average picture recognition was measured, meaning the face API might have recognised the computer scientist but how many of the 3 pictures did it recognise? Essentially how consistent was the Face API across all the 3 fame categories. The averages were 27 out

```
420   vcorrect = 0
421   vscores = []
422   vmatched = []
423
424   for key,value in Very_famous.items():
425
426       link = value
427       text = findMatch(link)
428       name = mysplit(key)
429       if text != "This person could not be matches by the Face API":
430           print(key + " was matched as " + text)
431           vcorrect = vcorrect + 1
432           score = getScore(link)
433           vscores.append(score)
434           if vmatched.__contains__(name):
435               print
436           else:
437               vmatched.append(name)
438       else:
439           print(key + "was not matched my the face API")
```

Figure 5-9: Appending lists and incriminating count

```
493 print("the very famous category matched " + str(vcorrect) + " out of " + str(len(Very_famous)) +
494       " Pictures with an average confidence rating of " + str(average(vscores)))
495
496 print("a total of " + str(len(vmatched)) + " People were matched out of " + str(len(Very_famous)/3))
497
498
499 print("the medium famous category matched " + str(mcorrect) + " out of " + str(len(Medium_famous)) +
500       " Pictures with an average confidence rating of " + str(average(mscores)))
501
502 print("a total of " + str(len(mmatched)) + " People were matched out of " + str(len(Medium_famous)/3))
503
504
505 print("the low famous category matched " + str(lcorrect) + " out of " + str(len(Low_famous)) +
506       " Pictures with an average confidence rating of " + str(average(lscores)))
507
508 print("a total of " + str(len(lmatched)) + " People were matched out of " + str(len(Low_famous)/3))
```

Figure 5-10: Formatting data to be output

of a possible 30 images recognised for the very famous category, 36 out of a possible 45 images were recognised in the well-known category and 120.6 out

```
the very famous category matched 27 out of 30 Pictures with an average confidence rating of 99.43220769917524
a total of 10 People were matched out of 10.0
the medium famous category matched 36 out of 45 Pictures with an average confidence rating of 98.54813764492671
a total of 15 People were matched out of 15.0
the low famous category matched 121 out of 225 Pictures with an average confidence rating of 98.71676529734587
a total of 58 People were matched out of 75.0
```

Figure 5-11: Example output of Face API test case

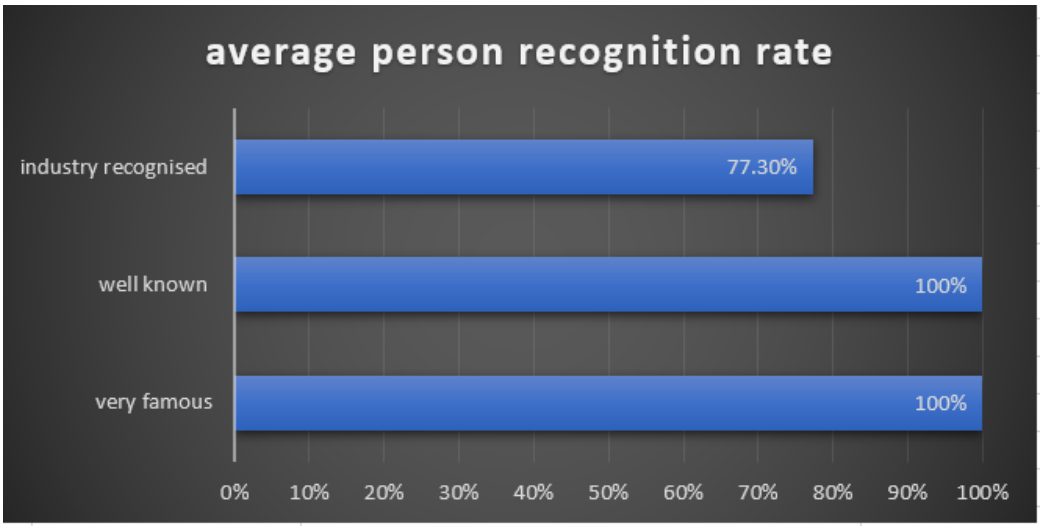


Figure 5-12: Face API people recognised

of a possible 225 images were recognised in the industry recognised category, on average (see Figure 5-13)

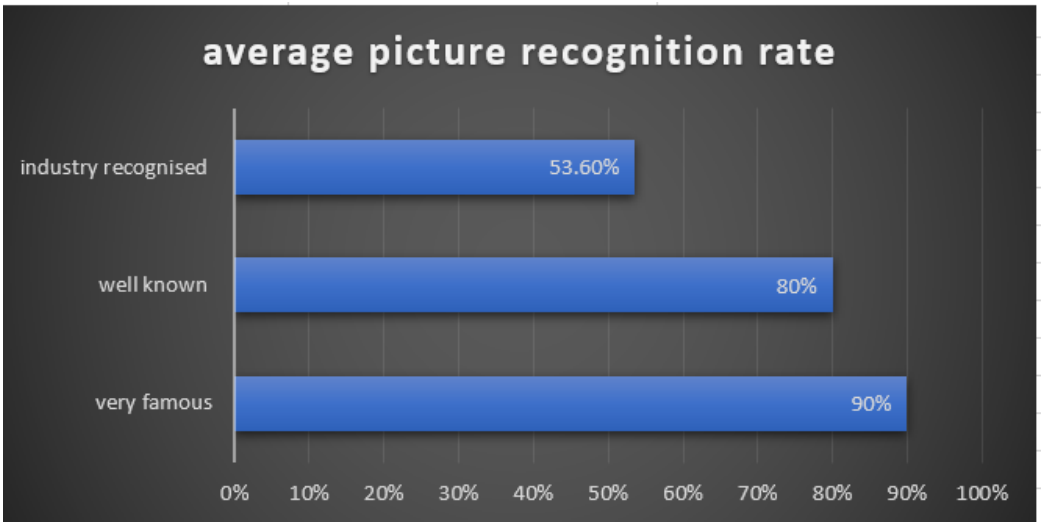


Figure 5-13: Face API images recognised

The third metric gathered was the average confidence rate of the matches made, the Face API gives a confidence rating along with the match stating how sure it is that it is that person. The average confidence rating for the very famous category was 99.43 percent, the average confidence rating for the well-known category was 98.54 percent and the average confidence rating for the industry recognised category was 98.71 percent (see Figure 5-14).

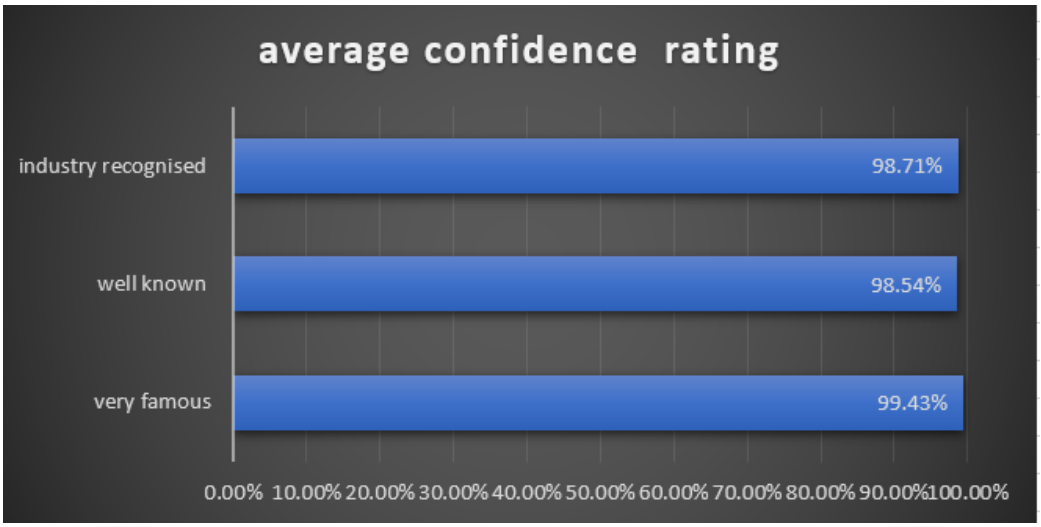


Figure 5-14: Face API average confidence ratings

As an extra final performance test for the Face API, it was decided to try and feed it some "dud" images. By sending the API images of lookalikes such a Bill Gates impersonator or an actor playing the part of Steve Jobs etc, it could be determined how accurate the API is, can it be tricked? After sending the image of Ashton Kutcher in the role of Steve Jobs (see Figure 5-15) the Face API recognised the picture as Ashton Kutcher.

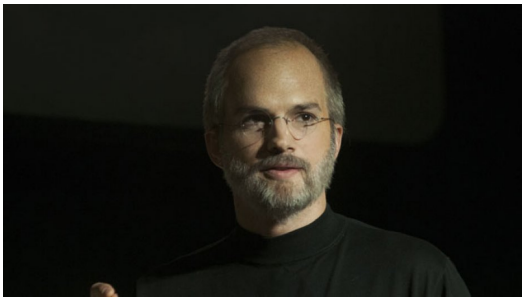


Figure 5-15: "Dud" image sent to API

All of these results were then taken and mapped on a graph to show a

correlation (see Figure 5-16)

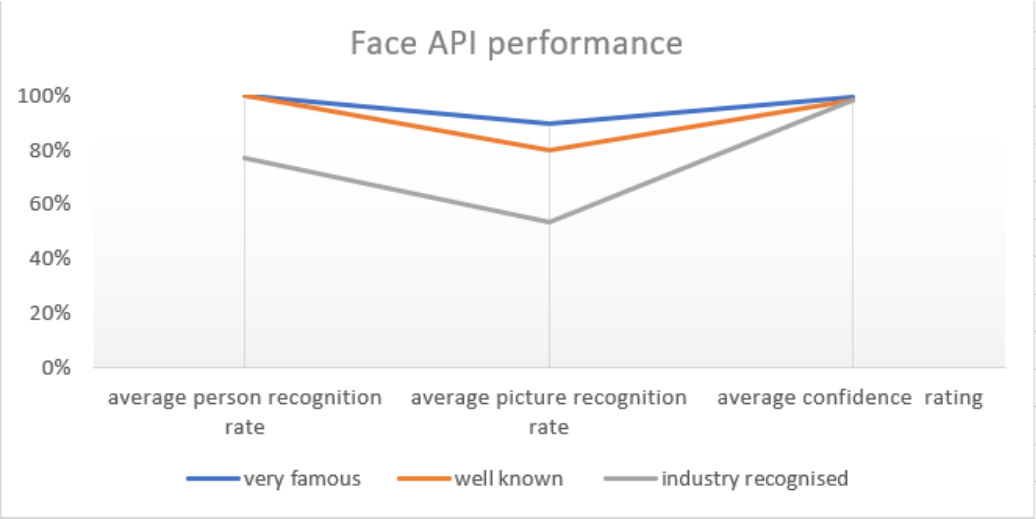


Figure 5-16: Face API performance

5.4.3 Evaluation of Results

Test 3 was evaluating the performance of the Face API used in the implementation of my web scraping with API dynamic challenge generation method. As previously mentioned for this test a list of 100 computer scientists was made and separated them into different fame categories to try and see how effective the Face API was in determining each different category. The first set of data collected for this test was how many people in each category were recognised. In both the very famous and well-known categories all of the computer scientists were recognised whereas with the industry recognised category 58 out of 75 were recognised, resulting in a total of 83 out of 100 computer scientists recognised. This performance was as expected in that the recognition rate of the people was less as they became less known however this performance was still relatively high. Even with 75 people included that were not very famous the Face API still managed to recognise over 80 percent of the 100 people.

The next set of data collected was how many of the actual images were recognised. Here this dissertation was investigating although the Face API may have recognised one of the images of the computer scientist, How consistent was it across the 3 images provided. The scores came back as follows: 27 out of 30 images recognised for the very famous category, 36 out of 45 for well known and 120.6 out 225 for industry recognised. This equates also

to 2.7 out of the 3 images provided on average for very famous, 2.4 out of 3 for well known and 1.6 out of 3 for industry recognised. Here we see a more drastic drop in performance when it comes to the lesser known people, although the Face API was able to recognise over 77 percent of the people included in the industry recognised category it only recognised 53.6 percent of the total images in the same category. This is quite a dramatic drop and should be taken into account for this considered purpose. This would suggest maybe it best to gather multiple images of people ,if intending to use them, if they are lesser known as this could be down to the API learning whom certain pictures are but struggling to match them when a face is at a different angle.

The third set of data collected was the average certainty rating given when a match was made. The results were fairly consistent throughout all of the categories with very famous having 99.43 percent, well known having 98.54 percent and industry recognised having 98.71 percent. While it can be seen that technically the industry recognised category has a marginally higher average confidence rating but this is likely insignificant and the point to take from these results is that the confidence ratings returned from the Face API are very high, and consistently very high which is essential if its to be used for this intended purpose.

Overall the Results show that the Face API is very effective with it recognising 83 out of 100 computer and the average confidence rating being over 98 percent in all three categories, the only area of concern is the rate at which it recognises images in the lesser known category with almost half of the images not being matched. This is not as much of a problem as this can be balanced out by feeding multiple images of the same person to find a match. Finally, the fact that it can determine a face made to look like another correctly increases the reliability of the API as a service. Analysing the performance of this API it can definitely be deemed effective and suitable for the intended use.

5.5 Conclusions

Overall the implementation stage has proved effective in determining the dynamic challenge generation method with the most potential is web scraping with API. The Bing API proved effective but the performance got gradually worse the more specific the search term got, as discussed this isn't fatal but should be considered when using the Bing API for generating cyber security challenges, as proposed. The Face API performed well in the tests provided also with a high recognition rate and the low photo recognition for the lower fame tier being an issue that is easily remedied by supplying a surplus of

images this API can be deemed suitable for this purpose and furthermore can give an edge in challenge generation that wouldn't be possible without.

6 Conclusions and Future Work

6.1 Summary of conclusions

A review of the literature was done, investigating different learning model, their significance and use in the classroom and more specifically their use in computer science. After this the literature on CTF learning was reviewed before finally taking a more in-depth look at the literature on gamification which showed the various uses and the potential it has for cyber security education (as well as various pitfalls to avoid), furthermore suggesting that there maybe be a way to combine gamification and learning models such as Bloom's taxonomy. This knowledge was then taken and applied to a theoretical design for an application that would teach young audiences cyber security before technical elements behind the application were discussed, including dynamic challenge generation methods, which were designed and implemented. After the implementation, test case scripts were made in order to evaluate the performance of the methods and the APIs utilised. The results fed back suggested: Web scraping with the use of APIs is the most effective method of dynamic challenge generation, and that both of the APIs used, while the performance slightly suffers from the specificity that comes with certain areas of cyber security, have impressive performance results that justify their use for this purpose.

6.2 Achievement of aim and Objectives

In this section, this dissertation will discuss how well this project has met the aims and objectives set at the beginning of the project.

The first aim of this project was to do the relevant research looking into different learning techniques and further into CTF learning to provide a literature review. This project succeeded to meet this aim, the literature was well-researched and covered all of the areas of identified interest, as well as learning models and CTF learning, extensive research was also done on gamification and how this could be applied to the classroom. Ultimately the aim was to cover the topic to a point where an application with the intended purpose of engaging a younger audience with cyber security in answer to the cyber security skills gap could be proposed and designed, the literature review in chapter 2 of this dissertation met this aim.

The second aim of this project was to take what was learned from the literature an use it to design a proof of concept for an application that took into account the literature discussed. The design chapter of this project achieved this aim, this dissertation took what was learned about learning

models and gamified learning as well as aspects of CTF styled learning and applied these to design the architecture of how this application would work, mocking up how it would look and technically designing aspects of it that would later be implemented.

The first objective was to investigate both learning models and gamification to better understand how to design an application for the intended purpose, with the literature review this objective was complete. The second objective was designing the application with reference to the literature which as just discussed was achieved. The next step was to design learning materials to be used in conjunction with the application, it was originally intended to mockup about 5 lessons to give a wider span of what the application would look like however due to time it was decided to mock up 3 and concentrate on them. The next two objectives were to design technical aspects of the proposed application and implement them, the first of which being completed in the technical design subchapter and the second of which being completed in the implementation chapter. With more time more detail would have gone into the technical design of the application and maybe how the application could have been built as a web application, how that would have been structured and how it could have been scaled would have been discussed and included in this dissertation but also due to time constraints sound attempt could not have been delivered. The final objective was to test and evaluate the usefulness of the prototype elements, this was a success, this dissertation performed a series of appropriate tests that collected interesting data allowing for it to come to the conclusions that were desired without ambiguity.

Overall, although some sacrifices were made due to time, this dissertation has satisfied for the most part all of the aims and objectives set at the beginning and that this project achieved what it set out to.

6.3 Future Work

This project was researched and took aspects of that research to design the concept for an application that would be used to teach cyber security to a high school aged audience. Further work on it could be developing the application itself taking what has been discussed in the application design subchapter as well as what was discussed in the lesson design chapter and realising it in an actual working application which could be tested against the target audience to see if it had the desired effect and engaged the students as hoped. Further learning material would also need to be developed for it, with the learning material needing to get gradually more complex and embody the different levels of the Bloom's taxonomy this would probably

be a project in itself, ensuring that the material followed this pattern while remaining engaging would be a difficult task to undertake.

Another aspect of this project that could be taken further is the work on dynamic challenge generation, as the potential for web scraping with API as a method of dynamic challenge generation has been shown further work could be done in testing different ways in which to utilise this method, how to produce yet more results from it and increase the performance of it as a dynamic challenge generation method by maybe using different APIs or other aspects of web scraping. It would be an interesting undertaking as there are so many possibilities for this method's potential use.

6.4 Personal Reflections

Over the course of this project I have learned a great deal about researching, project management, independent learning, and have developed many of my practical skills giving me further experience in python, with use of Microsoft Azure and APIs as well as new tools and services such as docker all of which I have enjoyed and has given me thought about which direction to take my career.

First I'd like to discuss my project management as I think this is my weakest part but is one in which I've learned the most. This project did not always look like this from the start and took many different forms in the first few months, I started out with a different idea and was implementing something different, I then decided that this area wasn't going to be worthwhile so pivoted to another idea for practical implementation, going as far as to start developing it before realising that there, not only wasn't enough emphasis on my chosen course so it wasn't accurately showcasing what I had learned at my time at Napier, but it also would have been a nightmare to evaluate. This is when I finally settled on the current idea. This was down to my poor project management and the fact that I settled on an idea without properly investigating it's worth as a project and what it would look like if chosen for this module and what that would mean. I have learned to think further down the line instead of just focusing on trying to get it done, because I ended up in a position where I was behind schedule by quite a bit and this wasn't down to me neglecting this project in favour of my other modules or out of laziness, consistently working on this project I thought was actually one of my key strengths, but was down to me starting a project wasting time on it ,then starting from scratch. In the end once I had my final project decided I dedicated all my time to completing it and getting it to a point where I was proud of it because I was concerned at one point that my project wouldn't flow and while the flow of the project and all of its connected parts did suffer

a little bit, again down to the poor project management and starting over on different ideas, I feel that what the project ended up being flows quite well and is coherent, providing solutions to the problems raised at the beginning. In hindsight, I would dedicate more time at the beginning of the project investigating my ideas further to determine what would work at a far earlier stage.

An aspect of the project I thought I done well was the research element. I was very well organised throughout the research chapter, going through collecting papers that covered the relevant areas, printing out the papers and highlighting different topics with different colours which helped the flow of my literature review. This resulted in a well researched literature review which provided a solid foundation for my project and going forward proved useful in the design chapter as a result. Another aspect I feel I did well was the implementation, After changing the project so many times I was running out of time by the time I got to my actual implementation. Despite this and my relatively limited experience in programming, I feel that I worked very hard on the implementation and making sure it was done well. I made an entire folder full of scripts learning python and how to do web scraping, how to call APIs etc. After learning this I worked on implementing the methods in a way that it could be measured. I learned a lot about programming while doing this project, more than I have in any other module or coursework assignment I truly feel like I've grown a great deal in software development and can see that from what I have produced, this aspect especially has made this project very rewarding. I also feel that I carried out the Results and Evaluation section well selecting interesting aspects to test which help draw meaningful conclusions, answering what dynamic challenge method had the most potential to be used for this task as well as carrying out a performance test on both of the Bing and Face APIs measuring both of their effectiveness in the use of generating random challenges.

Overall there were aspects that I could definitely have done better in and lessons that I will take from that but I feel like I carried out an interesting project in which I have learned a lot and gained valuable experience that will serve me well in the future.

References

[1] T. Caldwell, "Plugging the cyber-security skills gap," *Computer Fraud & Security*, vol. 2013, no. 7, pp. 5–10, 2013.

[2] G. Smith, "The intelligent solution: automation, the skills shortage and cyber-security," *Computer Fraud & Security*, vol. 2018, no. 8, pp. 6–9, 2018.

[3] S. Cobb, "Mind this gap: criminal hacking and the global cybersecurity skills shortage, a critical analysis," 2016.

[4] D. Pedley, D. McHenry, H. Motha, and J. N. Shah, "Understanding the uk cyber security skills labour market," *Ipsos MORI*, 2018.

[5] C.-H. Su and C.-H. Cheng, "A mobile gamification learning system for improving the learning motivation and achievements," *Journal of Computer Assisted Learning*, vol. 31, no. 3, pp. 268–286, 2015.

[6] S. Deterding, D. Dixon, R. Khaled, and L. Nacke, "From game design elements to gamefulness: defining gamification," in *Proceedings of the 15th international academic MindTrek conference: Envisioning future media environments*. ACM, 2011, pp. 9–15.

[7] K. Seaborn and D. I. Fels, "Gamification in theory and action: A survey," *International Journal of human-computer studies*, vol. 74, pp. 14–31, 2015.

[8] W. H.-Y. Huang and D. Soman, "Gamification of education," *Research Report Series: Behavioural Economics in Action, Rotman School of Management, University of Toronto*, 2013.

[9] F. F.-H. Nah, Q. Zeng, V. R. Telaprolu, A. P. Ayyappa, and B. Eschenbrenner, "Gamification of education: a review of literature," in *International conference on hci in business*. Springer, 2014, pp. 401–409.

[10] I. Caponetto, J. Earp, and M. Ott, "Gamification and education: A literature review," in *European Conference on Games Based Learning*, vol. 1. Academic Conferences International Limited, 2014, p. 50.

[11] V. Ford, A. Siraj, A. Haynes, and E. Brown, "Capture the flag unplugged: an offline cyber competition," in *Proceedings of the 2017 ACM SIGCSE Technical Symposium on Computer Science Education*. ACM, 2017, pp. 225–230.

[12] M. Katsantonis, P. Fouliras, and I. Mavridis, "Conceptual analysis of cyber security education based on live competitions," in *Global Engineering Education Conference (EDUCON), 2017 IEEE*. IEEE, 2017, p. 771.

[13] P. ctf, "Picto computer security game," https://picoctf.com/about, 2018, [Online; accessed 27-Oct-2018].

[14] ——, "Picto computer security game," https://picoctf.com/picoCTF_sponsorships.pdf, 2018, [Online; accessed 27-Oct-2018].

[15] csaw, "Cyber Security Awareness Worldwide," https://csaw.engineering.nyu.edu//, 2018, [Online; accessed 27-Oct-2018].

[16] D. con, "DEF CON," https://www.defcon.org/, 2018, [Online; accessed 27-Oct-2018].

[17] C. Cowan, S. Arnold, S. Beattie, C. Wright, and J. Viega, "Defcon capture the flag: Defending vulnerable code from intense attack," in *DARPA Information Survivability Conference and Exposition, 2003. Proceedings*, vol. 1. IEEE, 2003, pp. 120–129.

[18] A. Conklin, "Cyber defense competitions and information security education: An active learning solution for a capstone course," in *System Sciences, 2006. HICSS'06. Proceedings of the 39th Annual Hawaii International Conference on*, vol. 9. IEEE, 2006, pp. 220b–220b.

[19] M. H. b. Noor Azam and R. Beuran, "Usability evaluation of open source and online capture the flag platforms," 2018.

[20] M. C. Libicki, D. Senty, and J. Pollak, *Hackers Wanted: an examination of the cybersecurity labor market*. Rand Corporation, 2014.

[21] R. Albert, G. Markowsky, and J. Wallingford, "High school cyber defense competitions: Lessons from the trenches." in *Security and Management*, 2010, pp. 280–285.

[22] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, p. 256.

[23] L. Buchanan, F. Wolanczyk, and F. Zinghini, "Blending bloom's taxonomy and serious game design," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer ..., 2011, p. 1.

[24] D. Manson and R. Pike, "The case for depth in cybersecurity education," *ACM Inroads*, vol. 5, no. 1, pp. 47–52, 2014.

[25] R. Albert, G. Markowsky, and J. Wallingford, "High school cyber defense competitions: Lessons from the trenches." in *Security and Management*, 2010, p. 281.

[26] T. M. E. BLOOM'S, *Bloom's taxonomy of educational objectives*. Longman, 1965.

[27] L. W. Anderson and L. A. Sosniak, *Bloom's taxonomy*. Univ. Chicago Press, 1994.

[28] D. R. Krathwohl, "A revision of bloom's taxonomy: An overview," *Theory into practice*, vol. 41, no. 4, p. 213, 2002.

[29] M. Forehand, "Bloom's taxonomy," *Emerging perspectives on learning, teaching, and technology*, vol. 41, p. 47, 2010.

[30] D. R. Krathwohl, "A revision of bloom's taxonomy: An overview," *Theory into practice*, vol. 41, no. 4, pp. 214–218, 2002.

[31] C. G. Johnson and U. Fuller, "Is bloom's taxonomy appropriate for computer science?" in *Proceedings of the 6th Baltic Sea conference on Computing education research: Koli Calling 2006*. ACM, 2006, pp. 120–123.

[32] E. Thompson, A. Luxton-Reilly, J. L. Whalley, M. Hu, and P. Robbins, "Bloom's taxonomy for cs assessment," in *Proceedings of the tenth conference on Australasian computing education-Volume 78*. Australian Computer Society, Inc., 2008, pp. 155–161.

[33] T. Scott, "Bloom's taxonomy applied to testing in computer science classes," *Journal of Computing Sciences in Colleges*, vol. 19, no. 1, pp. 267–274, 2003.

[34] J. Van Niekerk and R. Von Solms, "Using bloom's taxonomy for information security education," in *IFIP World Conference on Information Security Education*. Springer, 2009, pp. 280–287.

[35] L. Buchanan, F. Wolanczyk, and F. Zinghini, "Blending bloom's taxonomy and serious game design," in *Proceedings of the International Conference on Security and Management (SAM)*. The Steering Committee of The World Congress in Computer Science, Computer ..., 2011, pp. 3,5.

[36] N. Schutte and N. Barkhuizen, "The development of a strategic leadership competency measure for public sector leaders–a pilot study," *Development*, vol. 2, no. 3, 2016.

[37] U. Tan, "Psychomotor theory: Mind-brain-body triad in health and disease," *NeuroQuantology*, vol. 4, no. 2, 2007.

[38] J. Westwood, H. Hoffman, D. Stredney, and S. Weghorst, "Validation of virtual reality to teach and assess psychomotor skills in laparoscopic surgery: results from randomised controlled studies using the mist vr laparoscopic simulator," *Medicine Meets Virtual Reality: art, science, technology: healthcare and evolution*, p. 124, 1998.

[39] A. G. Gallagher, K. Richie, N. McClure, and J. McGuigan, "Objective psychomotor skills assessment of experienced, junior, and novice laparoscopists with virtual reality," *World journal of surgery*, vol. 25, no. 11, pp. 1478–1483, 2001.

[40] A. G. Gallagher and R. Satava, "Virtual reality as a metric for the assessment of laparoscopic psychomotor skills," *Surgical Endoscopy and Other Interventional Techniques*, vol. 16, no. 12, pp. 1746–1752, 2002.

[41] T. P. Grantcharov, V. Kristiansen, J. Bendix, L. Bardram, J. Rosenberg, and P. Funch-Jensen, "Randomized clinical trial of virtual reality simulation for laparoscopic skills training," *British Journal of Surgery*, vol. 91, no. 2, pp. 146–150, 2004.

[42] K. S. Lehmann, J. P. Ritz, H. Maass, H. K. Çakmak, U. G. Kuehnapfel, C. T. Germer, G. Bretthauer, and H. J. Buhr, "A prospective randomized study to test the transfer of basic psychomotor skills from virtual reality to physical reality in a comparable training setting," *Annals of surgery*, vol. 241, no. 3, p. 442, 2005.

[43] A. Mitchell and C. Savill-Smith, "The use of computer and video games for learning: A review of the literature," 2004.

[44] S. G. Campbell, P. O'Rourke, and M. F. Bunting, "Identifying dimensions of cyber aptitude: the design of the cyber aptitude and talent

assessment," in *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*, vol. 59, no. 1. SAGE Publications Sage CA: Los Angeles, CA, 2015, pp. 721–725.

[45] A. Kennedy, E. Boyle, O. Traynor, T. Walsh, and A. Hill, "Video gaming enhances psychomotor skills but not visuospatial and perceptual abilities in surgical trainees," *Journal of surgical education*, vol. 68, no. 5, pp. 414–420, 2011.

[46] J. Burket, P. Chapman, T. Becker, C. Ganas, and D. Brumley, "Automatic problem generation for capture-the-flag competitions," *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*, 2015.

[47] CTFd, "CTFd CTF platform," https://github.com/CTFd/CTFd, 2018, [Online; accessed 10-Oct-2018].

[48] ——, "CTFd CTF platform," https://github.com/CTFd/plugins, 2018, [Online; accessed 18-Nov-2018].

[49] ——, "CTFd CTF platform," https://github.com/CTFd/CTFd/wiki/Plugins, 2018, [Online; accessed 18-Nov-2018].

[50] fbctf, "Facebook CTF platform," https://github.com/facebook/fbctf, 2018, [Online; accessed 10-Oct-2018].

[51] ——, "Mellivora CTF platform," https://github.com/Nakiami/mellivora, 2018, [Online; accessed 10-Oct-2018].

[52] hack the arch, "Hack the arch scoring platform," https://github.com/mcpa-stlouis/hack-the-arch, 2018, [Online; accessed 10-Oct-2018].

[53] S. Gen, "Security Scenario Generator," https://github.com/cliffe/SecGen, 2018, [Online; accessed 10-Oct-2018].

[54] Z. C. Schreuders, T. Shaw, G. Ravichandran, J. Keighley, M. Ordean *et al.*, "Security scenario generator (secgen): A framework for generating randomly vulnerable rich-scenario vms for learning computer security and hosting ctf events," in *USENIX*. USENIX Association, 2017.

[55] H. the box, "Hack the box pen testing lab," https://github.com/Nakiami/mellivora, 2018, [Online; accessed 10-Oct-2018].

[56] ctfs, "CTF all the time," https://ctfs.me/challenges, 2018, [Online; accessed 10-Oct-2018].

[57] H. your Ninja skills, "Web challenges," https://honeyourskills.ninja/, 2018, [Online; accessed 10-Oct-2018].

[58] J. Mirkovic and P. Peterson, "Class capture-the-flag exercises." in *3GSE*, 2014.

[59] H. W. Prabawa, E. Junaeti, and Y. Permana, "Using capture the flag in classroom: Game-based implementation in network security learning," in *Science in Information Technology (ICSITech), 2017 3rd International Conference on*. IEEE, 2017, pp. 690–695.

[60] M. Carlisle, M. Chiaramonte, and D. Caswell, "Using ctfs for an undergraduate cyber education," *2015 USENIX Summit on Gaming, Games, and Gamification in Security Education (3GSE 15)*, 2015.

[61] C. Eagle and J. L. Clark, "Capture-the-flag: Learning computer security under fire," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 2004.

[62] R. Albert, G. Markowsky, and J. Wallingford, "High school cyber defense competitions: Lessons from the trenches." in *Security and Management*, 2010, p. 280.

[63] ——, "High school cyber defense competitions: Lessons from the trenches." in *Security and Management*, 2010, p. 281.

[64] C. Eagle and J. L. Clark, "Capture-the-flag: Learning computer security under fire," NAVAL POSTGRADUATE SCHOOL MONTEREY CA, Tech. Rep., 2004.

[65] R. Albert, G. Markowsky, and J. Wallingford, "High school cyber defense competitions: Lessons from the trenches." in *Security and Management*, 2010, pp. 283 – 284.

[66] J. Majuri, J. Koivisto, and J. Hamari, "Gamification of education and learning: A review of empirical literature," in *Proceedings of the 2nd International GamiFIN Conference, GamiFIN 2018*. CEUR-WS, 2018.

[67] D. A. Kaufmann, "Reflection: Benefits of gamification in online higher education." *Journal of Instructional Research*, vol. 7, pp. 125–132, 2018.

[68] S. Deterding, R. Khaled, L. E. Nacke, and D. Dixon, "Gamification: Toward a definition," in *CHI 2011 gamification workshop proceedings*, vol. 12. Vancouver BC, Canada, 2011.

[69] K. M. Kapp, *The gamification of learning and instruction: game-based methods and strategies for training and education*. John Wiley & Sons, 2012.

[70] R. N. Landers, "Developing a theory of gamified learning: Linking serious games and gamification of learning," *Simulation & Gaming*, vol. 45, no. 6, pp. 752–768, 2014.

[71] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, p. 256.

[72] M. Prensky, "Digital game-based learning," *Computers in Entertainment (CIE)*, vol. 1, no. 1, pp. 21–21, 2003.

[73] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, p. 259.

[74] B. D. Cone, C. E. Irvine, M. F. Thompson, and T. D. Nguyen, "A video game for cyber security training and awareness," *computers & security*, vol. 26, no. 1, pp. 63–72, 2007.

[75] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, p. 259.

[76] L. A. Annetta, "The "i's" have it: A framework for serious educational game design." *Review of General Psychology*, vol. 14, no. 2, p. 105, 2010.

[77] M. Adams and M. Makramalla, "Cybersecurity skills training: an attacker-centric gamified approach," *Technology Innovation Management Review*, vol. 5, no. 1, 2015.

[78] L. Buchanan, F. Wolanczyk, and F. Zinghini, "Blending bloom's taxonomy and serious game design," in *Proceedings of the International*

Conference on Security and Management (SAM). The Steering Committee of The World Congress in Computer Science, Computer ..., 2011, p. 1.

[79] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, p. 259.

[80] R. Albert, G. Markowsky, and J. Wallingford, "High school cyber defense competitions: Lessons from the trenches." in *Security and Management*, 2010, pp. 280–285.

[81] A. Nagarajan, J. M. Allbeck, A. Sood, and T. L. Janssen, "Exploring game design for cybersecurity training," in *Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), 2012 IEEE International Conference on*. IEEE, 2012, p. 257.

[82] J. Hattie and H. Timperley, "The power of feedback," *Review of educational research*, vol. 77, no. 1, pp. 81–112, 2007.

[83] Y. Attali, "Effects of multiple-try feedback and question type during mathematics problem solving on performance in similar problems," *Computers & Education*, vol. 86, p. 260, 2015.

[84] J. D. Merrel, P. F. Cirillo, P. M. Schwartz, and J. Webb, "Multiple-choice testing using immediate feedback-assessment technique (if at®) forms: Second-chance guessing vs. second-chance learning?" 2015.

[85] H. L. Roediger III and A. C. Butler, "The critical role of retrieval practice in long-term retention," *Trends in cognitive sciences*, vol. 15, no. 1, pp. 20–27, 2011.

[86] Y. Attali, "Effects of multiple-try feedback and question type during mathematics problem solving on performance in similar problems," *Computers & Education*, vol. 86, pp. 260–266, 2015.

[87] M. Adams and M. Makramalla, "Cybersecurity skills training: an attacker-centric gamified approach," *Technology Innovation Management Review*, vol. 5, no. 1, 2015.

[88] Y. Fu and P. Clarke, "Gamification based cyber enabled learning environment of software testing," *submitted to the 123rd American Society for Engineering Education (ASEE)-Software Engineering Constituent*, 2016.

[89] G. Jin, M. Tu, T.-H. Kim, J. Heffron, and J. White, "Evaluation of game-based learning in cybersecurity education for high school students," *Journal of Education and Learning*, vol. 12, no. 1, pp. 150–158, 2018.

[90] A. M. Toda, P. H. Valle, and S. Isotani, "The dark side of gamification: An overview of negative effects of gamification in education," in *Researcher Links Workshop: Higher Education for All.* Springer, 2017, pp. 143–156.

[91] H. Gonzalez, R. Llamas, and F. Ordaz, "Cybersecurity teaching through gamification: Aligning training resources to our syllabus." *Research in Computing Science*, vol. 146, pp. 35–43, 2017.

[92] M. Katsantonis, P. Fouliras, and I. Mavridis, "Conceptual analysis of cyber security education based on live competitions," in *Global Engineering Education Conference (EDUCON), 2017 IEEE.* IEEE, 2017, pp. 771–779.

[93] S. Smith-Atakan, *Human-computer interaction.* Cengage Learning EMEA, 2006.

[94] F. E. Ritter, G. D. Baxter, and E. F. Churchill, "Foundations for designing user-centered systems," *London: Springer. doi*, vol. 10, pp. 978–1, 2014.

[95] A. Bulling, "Pervasive attentive user interfaces," *Computer*, no. 1, pp. 94–98, 2016.

[96] J. Nielsen, *Coordinating user interfaces for consistency.* Elsevier, 2014.

[97] F. E. Ritter, G. D. Baxter, and E. F. Churchill, "Foundations for designing user-centered systems," *London: Springer. doi*, vol. 10, p. 4, 2014.

[98] A. Cockburn, C. Gutwin, J. Scarr, and S. Malacria, "Supporting novice to expert transitions in user interfaces," *ACM Computing Surveys (CSUR)*, vol. 47, no. 2, p. 31, 2015.

[99] S. K. Card, *The psychology of human-computer interaction.* CRC Press, 2017.

[100] C. Williams, "Research methods," *Journal of business & economic research*, vol. 5, no. 3, pp. 65–72, 2007.

[101] T. L. Greenbaum, *Moderating focus groups: A practical guide for group facilitation.* Sage Publications, 1999.

[102] D. Bernstein, "Containers and cloud: From lxc to docker to kubernetes," *IEEE Cloud Computing*, vol. 1, no. 3, pp. 81–84, 2014.

[103] Y. Li and S. Manoharan, "A performance comparison of sql and nosql databases," in *2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM)*. IEEE, 2013, pp. 15–19.

Appendices

A Appendix 1: IPO

Below is my initial IPO. This was actually the second version after the initial one was changed. As the project was changed further down there was no longer a need for an IPO.

Initial Project Overview SOC10101 Honours Project (40 Credits) Title of Project: Developing a Cyber Security game application for educational purposes Overview of Project Content and Milestones

The aim of this project is to develop a game application which main focus would be to effectively teach skills commonly used in Cyber Security (in universities or CTF competitions etc) to users at high school levels of education, utilizing in part psychomotor learning/teaching methods in the process. The goal would be to ease the students into learning skills which might otherwise seem not accessible, by breaking down the activities and using gamification to engage them. Research will be done into how students learn, and what is the best way to teach students psychomotor skills. It will be a functioning game application with levels, the application will be used simultaneously with the command line in another window so the game can instruct the user on how to perform certain activities. A focus will be put on taking the structure of a 'jeopardy' style of CTF and representing it as a game application. The challenges separated by topic (e.g encryption) will be represented as a series of rooms in which the user will have to perform actions on the command line to progress through. A series of rooms will make up a 'floor', all rooms on one floor will focus on the same topic (so essentially will be series of related challenges). Once all of the rooms have been finished on a floor, the user will ascend to the next floor in which all the rooms will be focuses around a different topic (e.g web). The project could look at a way of incorporation a CTF style hint system in which if a user wants a hint they will have to beat a level or puzzle and a scoring system e.g the faster the user completes the challenge the more points they get. The goal is to create a prototype of the application with three floors: One which teaches you the basics of the game on the command line, one which teaches encryption and one which focuses on web challenges. The project could also look at ways of taking advantage of innovative technology so that it is cutting edge.

The Main Deliverable(s): All the code and assets used to make game application Working game application Hierarchy of files which will be used in conjunction with application Code used to develop web page which will be used in conjunction with application.

The Target Audience for the Deliverable(s):

The main target audience for this would be Students/Schools or anyone really looking to get into cyber security. Teachers looking to engage their students in this subject matter would also find it useful. It could also be of interest to those doing research into similar subject matter such as the teaching of cyber security skills effectively to inexperienced users.

The Work to be Undertaken:

This project will require research into gamification, then I will need to develop the game application itself as long as the files needed for learning and a webpage. Designing the game/ learning material so that it is engaging yet accessible to newcomers. Investigation into various learning/teaching techniques. Possibly conduct a survey of its usefulness and analyse the results for evaluation at the end of the project.

Additional Information / Knowledge Required:

Will need to build on my knowledge of developing a game, knowledge of creating a webpage. Will have to gather a lot of information on imparting practical skills, what the best methods for that is and how to apply them in this context.

Information Sources that Provide a Context for the Project:

There are some "always online CTF's" that have attempted to create challenges for "all levels" or a platform to learn such as:

https://www.root-me.org/?page=newslang=en https://www.hackthissite.org/

They in my opinion fail to ease the user in and can be overwhelming despite its intended purpose of training users.

https://www.usenix.org/system/files/conference/3gse15/3gse15-carlisle.pdf

This article looks at integrating CTF's at an undergraduate level for education, so is similar in concept but will be aiming more for a high school level of integration. It cites the usefulness of CTF's in an academic environment particularly in increasing the students motivation and enthusiasm for self directed learning and their desire to push their own boundaries of knowledge.

Boopathi, K., Sreejith, S., Bithin, A. (2015). Learning cyber security through gamification. Indian Journal of Science and Technology, 8(7), 642-649.

This paper is more accurate to what I will be doing, developing an actual functioning game that will impart skills of cyber security.

The Importance of the Project:

Exposing people to cyber security at an accessible level, especially young students is important as it could spark an interest in the subject. The complexity of some aspects of cyber security can put some people off so a simplification of it would prove beneficial. Having easy access to effective teaching

tools would benefit schools also.

The Key Challenge(s) to be Overcome:

The biggest hurdle of the project will be getting to grips with the creation of a game application, and integrating it smoothly with the other parts of the project (webpage, learning files). This will be more complex than anything i've undertaken in this area so will involve a lot of problem solving and finding ways to overcome obstacles. Developing a strategy of how best to teach these skills to newcomers will also be a challenge. Finally finding a way to ethically evaluate the success of the project could prove difficult.

B Appendix 2: Project management evidence

Just as before with the IPO the Gantt Charts I done were for the original proposed ideas. I kept them anyway and have provided evidence of them (see Figures B-1, B-2, B-3)

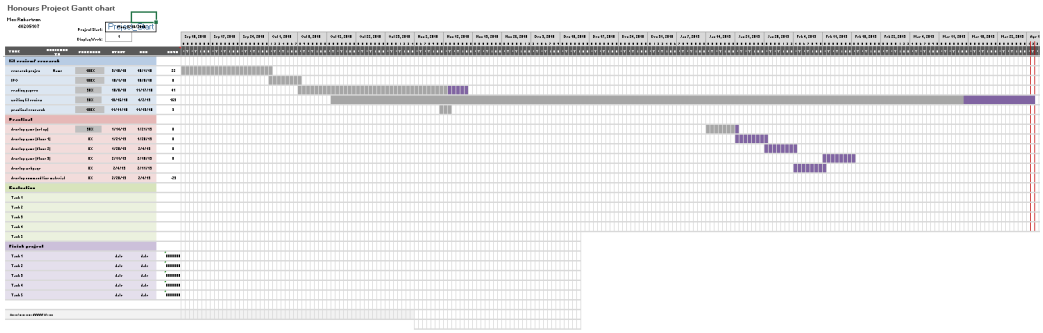


Figure B-1: Gantt Chart Evidence 1

lit review/ research					
research project	Name	100%	9/10/18	10/1/18	22
IPO		100%	10/1/18	10/8/18	8
reading papers		90%	10/8/18	11/17/18	41
writing lit review		90%	10/16/18	4/2/19	169
practical research		100%	11/11/18	11/13/18	3
Practical					
develop game (set up)		90%	1/14/19	1/21/19	8
develop game (floor 1)		0%	1/21/19	1/28/19	8
develop game (floor 2)		0%	1/28/19	2/4/19	8
develop game (floor 3)		0%	2/11/19	2/18/19	8
develop webpage		0%	2/4/19	2/11/19	
develop command line material		0%	2/28/19	2/4/19	-23

Figure B-2: Gantt Chart Evidence 2

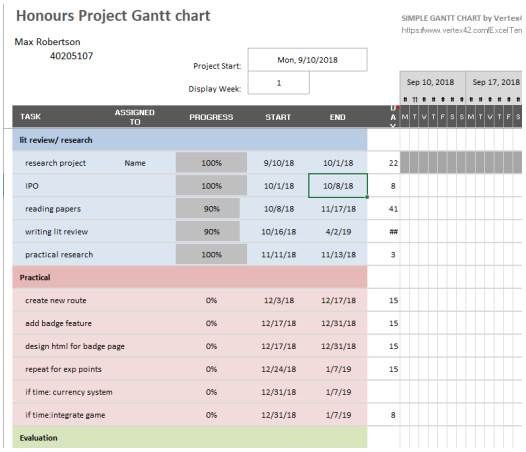


Figure B-3: Gantt Chart Evidence 3

C Appendix 3: random_selection.py code

```
import random

caeser_questions = {
    'Which answer is crypto put through a caeser cipher
        ': 'jyfwav',
    'Which answer is escape put through a caeser cipher
        ': 'lzjhwl',
    'Which answer is hacker put through a caeser cipher
        ': 'ohjrly',
    'Which answer is coding put through a caeser cipher
        ': 'jvkpun'}

question = (random.choice(list(caeser_questions)))
print(question)

for key, value in caeser_questions.items():
    print ("---" + value)

answer = input("Enter answer here:")

def checkAnswer(caeser_questions, answer):
    correctAnswers = ""
    listOfItems = caeser_questions.items()
```

```
    for item  in listOfItems:
        if item[1] == answer:
            correctAnswers = item[0]
    return  correctAnswers

answerChecked = checkAnswer(caeser_questions, answer)


if (answerChecked == question):
    print ("correct")
else:
    print ("incorrect")
```

D Appendix 4: API.py code

```
from flask import Flask, jsonify, request

app = Flask(__name__)

@app.route('/', methods=['GET', 'POST'])
def index():
    if (request.method == 'POST'):
        some_json = request.get_json()
        return jsonify({'you sent':some_json}), 201
    else:
        return jsonify({"about":"Hello World"})

@app.route('/multi/<int:num>', methods=['GET'])
def get_multiply10(num):
    return jsonify({"result":num * 10, "result2":num *
       2, "result3":num * 100})

@app.route('/question/<string:question>', methods=['GET
    '])
def get_question(question):
    if (question == "fred"):
        return jsonify({"question":" 'mylk'  is what
           word put through a caesar cipher ?", "answer
           ": "fred"})
    if (question == "bob"):
```

```
        return jsonify({"question":" 'ivi'   is what
           word put through a caesar cipher ?", "answer
           ": "bob"})
    if (question == "alice"):
        return jsonify({"question":" 'hspjl'   is what
           word put through a caesar cipher ?", "answer
           ": "alice"})
    if (question == "eve"):
        return jsonify({"question":" 'lcl'   is what
           word put through a caesar cipher ?", "answer
           ": "eve"})

if __name__ == '__main__':
    app.run(debug=True)
```

E Appendix 5: API_request.py code

```
import requests
import json
import random

questions = ['fred', 'bob', 'alice', 'eve']
question = (random.choice(questions))
values = []

resp = requests.get('http://127.0.0.1:5000/question/' +
    question)
json_data = json.loads(resp.text)
if resp.status_code != 200:
    print ("something went wrong")
for key, value in json_data.items():
    values.append(value)
user_question = values[1]
user_answer = values[0]
txt = input(user_question)
if (txt == user_answer):
    print ("correct !")
else:
    print ("incorrect")
```

F Appendix 6: web_scraper _final.py code

```
from bs4 import BeautifulSoup
import requests
import random
import hashlib
import re
from selenium import webdriver

source1 = requests.get('https://www.
    computersciencedegreehub.com/30-most-influential-
    computer-scientists-alive-today/').text
soup1 = BeautifulSoup(source1, 'lxml')

people = []

for person in soup1.find_all('h3'):
    computer_person = str(person).split('<h3>')
    computer_person_name = str(computer_person).split
        ('</h3>')
    c = " ".join(re.findall("[a-zA-Z]+", str(
        computer_person_name)))
    people.append(c)

people.pop(34)
people.pop(33)
people.pop(32)
people.pop(31)
people.pop(30)
people.pop(0)

person = (random.choice(people))
print (person + ' ' + hashlib.sha256(str(person).encode
    ('utf-8')).hexdigest())

from selenium.webdriver.chrome.options import Options
options = Options()
options.binary_location = "C:\Program Files (x86)\
    Google\Chrome\Application\chrome.exe"
```

```
browser = webdriver.Chrome(options=options,
    executable_path=r"C:\Program Files (x86)\Google\
    Chrome\Application\chromedriver.exe", )
browser.get("https://passwordsgenerator.net/md5-hash-
    generator/")
hashField = browser.find_element_by_id('txt1')
hashField.send_keys(person)
hashResult = browser.find_element_by_id('txt2').
    get_attribute("value")
print (person + ' ' + hashResult)
```

G Appendix 7: Face_quiz _final.py code

```
import http.client, urllib, base64
from pprint import pprint
import json
import sys
from azure.cognitiveservices.search.imagesearch import
    ImageSearchAPI
from msrest.authentication import
    CognitiveServicesCredentials
from PIL import Image
import urllib.request
import urllib.error
import os
import random
from cognitive_face import face
from azure.cognitiveservices.search.imagesearch.models.
    image_search_api_enums import ImageContent



topics = ['Famous Computer Scientists', 'Famous
    Cryptographers', 'Famous tech People', 'famous
    cryptologists',
            'famous computer programmers' , 'top
               cryptocurrency pioneers']
```

```
subscription_key = "71ad32482bbb4d99a0882c34d8b08d9f"
search_term = random.choice(topics)

headers = {
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': '9
       a899136784940ebadbe150dc9d28b39',
}

params = urllib.parse.urlencode({
    'visualFeatures': 'Categories',
    'details': 'Celebrities',
    'language': 'en',
})

def findImages (subscription_key, search_term):

    client = ImageSearchAPI(
       CognitiveServicesCredentials(subscription_key))
    image_results = client.images.search(query=
       search_term,count=100,ImageContent=face)

    if image_results.value:
        first_image_result = random.choice(
           image_results.value)
        return(first_image_result.content_url);

    else:
        print("No image results returned!")




def findMatch ():
    try:
        search_term = random.choice(topics)
        text = findImages(subscription_key, search_term
           )
        body = "{'url':\'"+text+"\'}"
```

```
h1 = http.client.HTTPConnection('www.cwi.nl')
conn = http.client.HTTPConnection('northeurope.
    api.cognitive.microsoft.com')
conn.request("POST", "/vision/v1.0/analyze?%s"
   % params, body, headers) #
response = conn.getresponse()
data = response.read()
conn.close()
d1=json.loads(data)

if (str(data).__contains__('"celebrities":[]')
   or str(data).__contains__('"categories":[]')
   ):
    #print ("catching empties has worked")
    return("no match");
else:
    name = (d1['categories'][0]['detail']['
       celebrities'][0]['name'])
    score = (d1['categories'][0]['detail']['
       celebrities'][0]['confidence'])
    classification = (d1['categories'][0]['name
       '])
try:
    urllib.request.urlretrieve(text, r"C:\Users
       \MAXBO\Documents\Test_images\file1.jpg")
    os.chdir(r'C:\Users\MAXBO\Documents\
       Test_images')
except urllib.error.HTTPError:
    #print("http error handling has worked")
    return("no match")


if name == "":
    #print("name is empty so failed")
    return("no match");
elif score == "":
    #print("score is empty so failed")
    return("no match");
elif (classification == "people_" or
   classification == "people_portrait"):
    #print (classification + " has passed the
```

```
                classification test and is therefore
                equal to people_ or people_portrait")
            return (text);
        else:
            #print (classification + " has failed the
                classification test and is therefore not
                 equal to people_ or people_portrait")
            return("no match");

    except KeyError:
        #print("Failed because of exeption")
        return("no match");

def getAnswer():
    try:
        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.
            api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s"
            % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)
        #if (str(data).__contains__('"celebrities":[]'
            or 'categories":[]')):
        score = (d1['categories'][0]['detail']['
            celebrities'][0]['confidence'])
        percentage = (score * 100)
        rounded_percentage = round(percentage,2)
        name = (d1['categories'][0]['detail']['
            celebrities'][0]['name'])
        urllib.request.urlretrieve(text, r"C:\Users\
            MAXBO\Documents\Test_images\file1.jpg")
        os.chdir(r'C:\Users\MAXBO\Documents\Test_images
            ')
        im = Image.open(r"C:\Users\MAXBO\Documents\
            Test_images\file1.jpg")
        im.show()
        answer = input("Who is this ?")
        if (answer.lower() == name.lower()):
```

```
            return("Correct, I am " + str(
                rounded_percentage) + " the correct
                answer is " + name);
        else:
            return("Incorrect, I am " + str(
                rounded_percentage) + " the correct
                answer was " + name);

    except KeyError:
        return("Error has occured. Try again.")

correct_answers = 0

for i in range(0, 5):
    while True:
        text = findMatch()

        if text != "no match":
            break

    body = "{'url':\'"+text+"\'}"

    print("Person " + str(i+1) + ":" )
    answer = getAnswer()
    if answer == "correct":
        correct_answers = correct_answers + 1
    print(answer)

print("You scored " + str(correct_answers) + "/5")
```

H Appendix 8: docker_face _pop.py code

```
import http.client, urllib, base64
from pprint import pprint
import json
import sys
from azure.cognitiveservices.search.imagesearch import
   ImageSearchAPI
from msrest.authentication import
   CognitiveServicesCredentials
```

```
from PIL import Image
import urllib.request
import urllib.error
import random
from cognitive_face import face
from azure.cognitiveservices.search.imagesearch.models.
    image_search_api_enums import ImageContent
import pymongo



topics = ['Famous Computer Scientists', 'Famous
    Cryptographers', 'Famous tech People', 'famous
    cryptologists', 'famous computer programmers' , 'top
     cryptocurrency pioneers']


subscription_key = "71ad32482bbb4d99a0882c34d8b08d9f"
search_term = random.choice(topics)

headers = {
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': '9
        a899136784940ebadbe150dc9d28b39',
}

params = urllib.parse.urlencode({
    'visualFeatures': 'Categories',
    'details': 'Celebrities',
    'language': 'en',
})



myclient = pymongo.MongoClient(host='mongoDb', port
    =27017)
mydb = myclient["mydatabase"]
mycol = mydb["challenges"]

def findImages (subscription_key, search_term):
```

```
    client = ImageSearchAPI(
       CognitiveServicesCredentials(subscription_key))
    image_results = client.images.search(query=
       search_term,count=100,ImageContent=face)

    if image_results.value:
        first_image_result = random.choice(
           image_results.value)
        return(first_image_result.content_url);

    else:
        print("No image results returned!")




def findMatch ():
    try:
        search_term = random.choice(topics)
        text = findImages(subscription_key, search_term
           )
        body = "{'url':\'"+text+"\'}"

        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.
           api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s"
           % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)

        if (str(data).__contains__('"celebrities":[]')
           or str(data).__contains__('"categories":[]')
           ):
            print ("catching empties has worked")
            return("no match");
        else:
            name = (d1['categories'][0]['detail']['
```

```
                celebrities'][0]['name'])
            score = (d1['categories'][0]['detail']['
                celebrities'][0]['confidence'])
            classification = (d1['categories'][0]['name
                '])

        try:
            urllib.request.urlretrieve(text)
        except urllib.error.HTTPError:
            print("http error handling has worked")
            return("no match")

        if name == "":
            print("name is empty so failed")
            return("no match");
        elif score == "":
            print("score is empty so failed")
            return("no match");
        elif (classification == "people_" or
           classification == "people_portrait"):
            print (classification + " has passed the
                classification test and is therefore
                equal to people_ or people_portrait")
            mydict = { "question": "Who is this ?", "
                image":text, "name":name}
            x = mycol.insert_one(mydict)
            print("Data has been inserted to database")
            return (text);
        else:
            print (classification + " has failed the
                classification test and is therefore not
                 equal to people_ or people_portrait")
            return("no match");

    except KeyError:
        print("Failed because of exeption")
        return("no match");



while True:
```

```
    text = findMatch()

    if text != "no match":
        break
```

I Appendix 9: Dockerfile contents

```
FROM python:3-onbuild
RUN pip install --trusted-host pypi.python.org -r
   requirements.txt
COPY docker_face_pop.py /docker_face_pop.py
#RUN pip install .
EXPOSE 27017
CMD ["python", "/docker_face_pop.py"]
```

J Appendix 10: requirements.txt contents

```
Flask
azure-cognitiveservices-search-imagesearch
pillow
cognitive_face
pymongo
```

K Appendix 11: docker-compose.yml contents

```
version: '3.3'

services:
  # Name our service will be known by
  db:

    # version of mongo we'll use
    image: mongo

    ports:
      - 27017:27017

    # using a named volume
    volumes:
      - devmongo:/data/db
```

```
  client:

    image: honours-project
    links:
        - db
    depends_on:
        - db
    ports:
        - 27017
    environment:
        - MONGO_URI=mongodb:27017/mydatabase

volumes:
  devmongo:
```

L Appendix 12: random_selection.py (test case) code used for Test 1

```
import random
from numpy.lib.function_base import average

caeser_questions = {
    'Which answer is crypto put through a caeser cipher
        ': 'jyfwav',
    'Which answer is escape put through a caeser cipher
        ': 'lzjhwl',
    'Which answer is hacker put through a caeser cipher
        ': 'ohjrly',
    'Which answer is coding put through a caeser cipher
        ': 'jvkpun'}

questions = []
unique_questions = []
occurence_count = []


for i in range(0, 100):
    question = (random.choice(list(caeser_questions)))
    if questions.__contains__(question):
```

```
            occurence_count.append(question)
        else:
            unique_questions.append(question)
        questions.append(question)

print (len(questions))
print (len(unique_questions))

ocount = []

for uquestion in unique_questions:
    count = questions.count(uquestion)
    print('This question ' + uquestion + ' appeared  '
       + str(count) + ' times')
    ocount.append(count)
print("There was a total of " + str(len(occurence_count
   )) + " Same occurences of a question")
print (str(average(ocount)))
```

M Appendix 13: API_request.py (test case) code used for Test 1

```
import requests
import json
import random
from numpy.lib.function_base import average

questions = ['fred', 'bob', 'alice', 'eve']




questionst = []
unique_questions = []
occurence_count = []


for i in range(0, 100):
    values = []
```

```
    question = (random.choice(questions))

    resp = requests.get('http://127.0.0.1:5000/question
       /' + question)
    json_data = json.loads(resp.text)
    if resp.status_code != 200:
        print ("something went wrong")
    for key, value in json_data.items():
        values.append(value)
    user_question = values[1]
    user_answer = values[0]

    if questionst.__contains__(user_question):
        occurence_count.append(question)
    else:
        unique_questions.append(user_question)
    questionst.append(user_question)

ocount = []

print (len(questionst))
print (len(unique_questions))

for uquestion in unique_questions:
    count = questionst.count(uquestion)
    print('This question ' + uquestion + 'appeared  ' +
        str(count) + ' times')
    ocount.append(count)

print("There was a total of " + str(len(occurence_count
   )) + " Same occurences of a question")
print (str(average(ocount)))
```

N Appendix 14: web_scraper _final.py (test case) code used for Test 1

```
from bs4 import BeautifulSoup
import requests
import random
import hashlib
```

```
import re
from selenium import webdriver
from numpy.lib.function_base import average

source1 = requests.get('https://www.
   computersciencedegreehub.com/30-most-influential-
   computer-scientists-alive-today/').text
soup1 = BeautifulSoup(source1, 'lxml')

people = []

for person in soup1.find_all('h3'):
    computer_person = str(person).split('<h3>')
    computer_person_name = str(computer_person).split
       ('</h3>')
    c = " ".join(re.findall("[a-zA-Z]+", str(
       computer_person_name)))
    people.append(c)

people.pop(34)
people.pop(33)
people.pop(32)
people.pop(31)
people.pop(30)
people.pop(0)

person = (random.choice(people))


questions = []
unique_questions = []
occurence_count = []

for i in range(0, 100):
    question = (random.choice(people))
    if questions.__contains__(question):
        occurence_count.append(question)
    else:
        unique_questions.append(question)
    questions.append(question)
```

```
print (len(questions))
print (len(unique_questions))

ocount = []

for uquestion in unique_questions:
    count = questions.count(uquestion)
    print('This question ' + uquestion + ' appeared  '
       + str(count) + ' times')
    ocount.append(count)
print("There was a total of " + str(len(occurence_count
   )) + " Same occurences of a question")
print (str(average(ocount)))
```

O Appendix 15: Face_quiz _final.py (test case) code used for Test 1

```
import http.client, urllib, base64
from pprint import pprint
import json
import sys
from azure.cognitiveservices.search.imagesearch import
   ImageSearchAPI
from msrest.authentication import
   CognitiveServicesCredentials
from PIL import Image
import urllib.request
import urllib.error
import os
import random
from cognitive_face import face
from azure.cognitiveservices.search.imagesearch.models.
   image_search_api_enums import ImageContent
from numpy.lib.function_base import average



topics = ['Famous Computer Scientists', 'Famous
   Cryptographers', 'Famous tech People', 'famous
   cryptologists', 'famous computer programmers' , 'top
```

```
    cryptocurrency pioneers']
#topics = ['Famous Computer Scientists']


subscription_key = "71ad32482bbb4d99a0882c34d8b08d9f"
search_term = random.choice(topics)

headers = {
    # Request headers. Replace the key below with your
       subscription key.
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': '9
       a899136784940ebadbe150dc9d28b39',
}

params = urllib.parse.urlencode({
    # Request parameters. All of them are optional.
    'visualFeatures': 'Categories',
    'details': 'Celebrities',
    'language': 'en',
})

def findImages (subscription_key, search_term):

    client = ImageSearchAPI(
       CognitiveServicesCredentials(subscription_key))
    image_results = client.images.search(query=
       search_term,count=100,ImageContent=face)

    if image_results.value:
        first_image_result = random.choice(
           image_results.value)
        return(first_image_result.content_url);

    else:
        print("No image results returned!")
```

```
def findMatch ():
    try:
        search_term = random.choice(topics)
        text = findImages(subscription_key, search_term
            )
        body = "{'url':\'"+text+"\'}"

        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.
            api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s"
            % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)

        if (str(data).__contains__('"celebrities":[]')
            or str(data).__contains__('"categories":[]')
            ):
            #print ("catching empties has worked")
            return("no match");
        else:
            name = (d1['categories'][0]['detail']['
                celebrities'][0]['name'])
            score = (d1['categories'][0]['detail']['
                celebrities'][0]['confidence'])
            classification = (d1['categories'][0]['name
                '])


        if name == "":
            #print("name is empty so failed")
            return("no match");
        elif score == "":
            #print("score is empty so failed")
            return("no match");
        elif (classification == "people_" or
            classification == "people_portrait"):
            #print (classification + " has passed the
                classification test and is therefore
```

```
                equal to people_ or people_portrait")
            return (text);
        else:
            #print (classification + " has failed the
                classification test and is therefore not
                 equal to people_ or people_portrait")
            return("no match");


    except KeyError:
        #print("Failed because of exeption")
        return("no match");


correct_answers = 0

questions = []
unique_questions = []
occurence_count = []

for i in range(0, 100):
    while True:
        text = findMatch()

        if text != "no match":
            break



    question = text
    if questions.__contains__(question):
        occurence_count.append(question)
    else:
        unique_questions.append(question)
    questions.append(question)

print (len(questions))
print (len(unique_questions))

ocount = []
```

```
for uquestion in unique_questions:
    count = questions.count(uquestion)
    print('This question ' + uquestion + 'appeared  ' +
        str(count) + ' times')
    ocount.append(count)


print("There was a total of " + str(len(occurence_count
   )) + " Same occurences of a question")
print (str(average(ocount)))

body = "{'url':\'"+text+"\'}"
```

P Appendix 16: bing_eval.py code used for Test 2

```
import http.client, urllib, base64
from pprint import pprint
import json
import sys
from azure.cognitiveservices.search.imagesearch import
   ImageSearchAPI
from msrest.authentication import
   CognitiveServicesCredentials
from PIL import Image
import urllib.request
import urllib.error
import os
import random
from cognitive_face import face
from azure.cognitiveservices.search.imagesearch.models.
   image_search_api_enums import ImageContent
from numpy.lib.function_base import average
from bs4 import BeautifulSoup
import requests
import random
import hashlib
import re
from selenium import webdriver
```

```
subscription_key = "71ad32482bbb4d99a0882c34d8b08d9f"
search_term = "famous computer scientists"


client = ImageSearchAPI(CognitiveServicesCredentials(
    subscription_key))
image_results = client.images.search(query=search_term,
    count=100,ImageContent=face)


from selenium.webdriver.chrome.options import Options
options = Options()
options.binary_location = "C:\Program Files (x86)\
    Google\Chrome\Application\chrome.exe"
browser = webdriver.Chrome(options=options,
    executable_path=r"C:\Program Files (x86)\Google\
    Chrome\Application\chromedriver.exe", )

if image_results.value:
     for i in range(0,100):
         first_image_result = image_results.value[i]
         browser.get(first_image_result.content_url)
     else:
         print("No image results returned!")
```

Q Appendix 17: Face_eval.py code used for Test 3

```
import http.client, urllib, base64
from pprint import pprint
import json
import sys
from azure.cognitiveservices.search.imagesearch import
    ImageSearchAPI
from msrest.authentication import
    CognitiveServicesCredentials
from PIL import Image
import urllib.request
import urllib.error
```

```
import os
import random
from cognitive_face import face
from azure.cognitiveservices.search.imagesearch.models.
    image_search_api_enums import ImageContent
from numpy.lib.function_base import average



#for the purposes of this appendices I have just
    included one computer scientist entry for each
    category dictionary as it was 100s of lines of code
    and would have taken up many pages

Very_famous = {
'Bill Gates 1' : 'https://specials-images.forbesimg.com
    /imageserve/5c76b4b84bbe6f24ad99c370/416x416.jpg?
    background=000000&cropX1=0&cropX2=4000&cropY1=0&
    cropY2=4000',
'Bill Gates 2' : 'https://fm.cnbc.com/applications/cnbc
    .com/resources/img/editorial
    /2018/07/11/105322791-1531301768595gettyimages
    -467620670.1910x1000.jpg',
'Bill Gates 3' : 'https://bitcoinist.com/wp-content/
    uploads/2018/05/wiki-Bill_Gates_MSC_2017-
    e1525827667380.jpg',
}

Medium_famous = {
'James Gosling 1' : 'https://images.computerhistory.
    org/fellows/jgosling.jpg',
'James Gosling 2' : 'http://nighthacks.com/jag/bio/
    JamesInViennaEnjoyingBeer.jpg',
'James Gosling 3' : 'https://upload.wikimedia.org/
    wikipedia/commons/thumb/1/14/James_Gosling_2008.jpg
    /220px-James_Gosling_2008.jpg',
}

Low_famous = {
'Howard Aiken 1' : 'https://upload.wikimedia.org/
```

```
    wikipedia/commons/c/c9/Aiken.jpeg',
'Howard Aiken 2' : 'https://history-computer.com/
    ModernComputer/Relays/images/AikenPortrait3.jpg',
'Howard Aiken 3' : 'https://ethw.org/w/images/8/8b/
    Aiken.jpg',
}

subscription_key = "71ad32482bbb4d99a0882c34d8b08d9f"


headers = {
    # Request headers. Replace the key below with your
        subscription key.
    'Content-Type': 'application/json',
    'Ocp-Apim-Subscription-Key': '9
        a899136784940ebadbe150dc9d28b39',
}

params = urllib.parse.urlencode({
    # Request parameters. All of them are optional.
    'visualFeatures': 'Categories',
    'details': 'Celebrities',
    'language': 'en',
})


def findMatch (link):
    try:
        body = "{'url':\'"+link+"\'}"

        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.
            api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s"
            % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)

        if (str(data).__contains__('"celebrities":[]')
```

```
            or str(data).__contains__('"categories":[]')
            ):
                #print ("catching empties has worked")
                return("This person could not be matches by
                    the Face API");
        else:
            name = (d1['categories'][0]['detail']['
                celebrities'][0]['name'])
            score = (d1['categories'][0]['detail']['
                celebrities'][0]['confidence'])
            percentage = (score * 100)
            rounded_percentage = round(percentage,2)
            classification = (d1['categories'][0]['name
                '])


        if name == "":
            #print("name is empty so failed")
            return("This person could not be matches by
                the Face API");
        elif score == "":
            #print("score is empty so failed")
            return("This person could not be matches by
                the Face API");
        elif (classification == "people_" or
            classification == "people_portrait"):
            #print (classification + " has passed the
                classification test and is therefore
                equal to people_ or people_portrait")
            return (str(rounded_percentage) + ":
                Certain that this is " + name);
        else:
            #print (classification + " has failed the
                classification test and is therefore not
                 equal to people_ or people_portrait")
            return("This person could not be matches by
                the Face API");

    except KeyError:
        #print("Failed because of exeption")
        return("This person could not be matches by the
```

```
            Face API");

def getScore (link):
    try:
        body = "{'url':\'"+link+"\'}"

        h1 = http.client.HTTPConnection('www.cwi.nl')
        conn = http.client.HTTPConnection('northeurope.
           api.cognitive.microsoft.com')
        conn.request("POST", "/vision/v1.0/analyze?%s"
           % params, body, headers) #
        response = conn.getresponse()
        data = response.read()
        conn.close()
        d1=json.loads(data)

        score = (d1['categories'][0]['detail']['
           celebrities'][0]['confidence'])
        percentage = (score * 100)
        return percentage;

    except KeyError:
        #print("Failed because of exeption")
        return("This person could not be matches by the
            Face API");

def mysplit(key):
        head = key.rstrip('0123456789')
        return head




vcorrect = 0
vscores = []
vmatched = []

for key,value in Very_famous.items():

    link = value
    text = findMatch(link)
```

```
    name = mysplit(key)
    if text != "This person could not be matches by the
        Face API":
        print(key + " was matched as " + text)
        vcorrect = vcorrect + 1
        score = getScore(link)
        vscores.append(score)
        if vmatched.__contains__(name):
            print
        else:
            vmatched.append(name)
    else:
        print(key + "was not matched my the face API")



mcorrect = 0
mscores = []
mmatched = []

for key,value in Medium_famous.items():

    link = value
    text = findMatch(link)
    name = mysplit(key)
    if text != "This person could not be matches by the
        Face API":
        print(key + " was matched as " + text)
        mcorrect = mcorrect + 1
        score = getScore(link)
        mscores.append(score)
        if mmatched.__contains__(name):
            print
        else:
            mmatched.append(name)
    else:
        print(key + "was not matched my the face API")


lcorrect = 0
lscores = []
```

```
lmatched = []

for key,value in Low_famous.items():

    link = value
    text = findMatch(link)
    name = mysplit(key)
    if text != "This person could not be matches by the
        Face API":
        print(key + " was matched as " + text)
        lcorrect = lcorrect + 1
        score = getScore(link)
        lscores.append(score)
        if lmatched.__contains__(name):
            print
        else:
            lmatched.append(name)
    else:
        print(key + "was not matched my the face API")
```

```
print("the very famous category matched " + str(
   vcorrect) + " out of " + str(len(Very_famous)) +
      " Pictures with an average confidence rating of "
        + str(average(vscores)))

print("a total of " + str(len(vmatched)) + " People
   were matched out of " + str(len(Very_famous)/3))


print("the medium famous category matched " + str(
   mcorrect) + " out of " + str(len(Medium_famous)) +
      " Pictures with an average confidence rating of "
        + str(average(mscores)))
```

```
print("a total of " + str(len(mmatched)) + " People
   were matched out of " + str(len(Medium_famous)/3))


print("the low famous category matched " + str(lcorrect
   ) + " out of " + str(len(Low_famous)) +
      " Pictures with an average confidence rating of "
        + str(average(lscores)))

print("a total of " + str(len(lmatched)) + " People
   were matched out of " + str(len(Low_famous)/3))
```

R Appendix 18: Face_eval.py output

Bill Gates 1 was matched as 99.97: Certain that this is Bill Gates Bill Gates 2 was matched as 100.0: Certain that this is Bill Gates Bill Gates 3 was matched as 100.0: Certain that this is Bill Gates Steve Jobs 1 was matched as 100.0: Certain that this is Steve Jobs Steve Jobs 2 was matched as 100.0: Certain that this is Steve Jobs Steve Jobs 3 was matched as 100.0: Certain that this is Steve Jobs Elon Musk 1was not matched my the face API Elon Musk 2 was matched as 100.0: Certain that this is Elon Musk Elon Musk 3 was matched as 99.96: Certain that this is Elon Musk Mark Zuckerberg 1 was matched as 97.59: Certain that this is Mark Zuckerberg Mark Zuckerberg 2 was matched as 98.87: Certain that this is Mark Zuckerberg Mark Zuckerberg 3 was matched as 89.13: Certain that this is Mark Zuckerberg Larry Page 1 was matched as 100.0: Certain that this is Larry Page Larry Page 2 was matched as 100.0: Certain that this is Larry Page Larry Page 3 was matched as 100.0: Certain that this is Larry Page Sergey Brin 1 was matched as 99.98: Certain that this is Sergey Brin Sergey Brin 2 was matched as 99.99: Certain that this is Sergey Brin Sergey Brin 3 was matched as 99.67: Certain that this is Sergey Brin Tim Berners Lee 1 was matched as 99.98: Certain that this is Tim Berners-Lee Tim Berners Lee 2 was matched as 99.97: Certain that this is Tim Berners-Lee Tim Berners Lee 3 was matched as 100.0: Certain that this is Tim Berners-Lee Alan Turing 1was not matched my the face API Alan Turing 2was not matched my the face API Alan Turing 3 was matched as 100.0: Certain that this is Alan Turing Steve Wozniak 1 was matched as 99.92: Certain that this is Steve Wozniak Steve Wozniak 2 was matched as 99.98: Certain that this is Steve Wozniak Steve Wozniak 3 was matched as 99.86: Certain that this is Steve Wozniak Linus Torvalds 1 was matched as 99.99: Certain that this is Linus Torvalds Linus Torvalds 2 was matched as 100.0: Certain that this is Linus Torvalds Linus Torvalds 3 was matched as

99.84: Certain that this is Linus Torvalds James Gosling 1 was matched as 99.98: Certain that this is James Gosling James Gosling 2 was matched as 99.98: Certain that this is James Gosling James Gosling 3was not matched my the face API Grace Hopper 1 was matched as 99.99: Certain that this is Grace Hopper Grace Hopper 2 was matched as 99.92: Certain that this is Grace Hopper Grace Hopper 3was not matched my the face API Martin Hellman 1 was matched as 97.34: Certain that this is Martin Hellman Martin Hellman 2 was matched as 99.98: Certain that this is Martin Hellman Martin Hellman 3 was matched as 99.99: Certain that this is Martin Hellman Michael Widenius 1was not matched my the face API Michael Widenius 2was not matched my the face API Michael Widenius 3 was matched as 99.6: Certain that this is Michael Widenius Yukihiro Matsumoto 1 was matched as 100.0: Certain that this is Yukihiro Matsumoto Yukihiro Matsumoto 2 was matched as 99.97: Certain that this is Yukihiro Matsumoto Yukihiro Matsumoto 3 was matched as 100.0: Certain that this is Yukihiro Matsumoto John Resig 1 was matched as 78.35: Certain that this is John Resig John Resig 2 was matched as 99.71: Certain that this is John Resig John Resig 3was not matched my the face API Brian Kernighan 1was not matched my the face API Brian Kernighan 2 was matched as 99.99: Certain that this is Brian Kernighan Brian Kernighan 3 was matched as 99.99: Certain that this is Brian Kernighan Ken Thompson 1was not matched my the face API Ken Thompson 2 was matched as 76.03: Certain that this is Ken Thompson Ken Thompson 3was not matched my the face API David Axmark 1 was matched as 99.97: Certain that this is David Axmark David Axmark 2 was matched as 99.98: Certain that this is David Axmark David Axmark 3 was matched as 98.15: Certain that this is David Axmark Ben Goodger 1 was matched as 99.82: Certain that this is Ben Goodger Ben Goodger 2 was matched as 99.54: Certain that this is Ben Goodger Ben Goodger 3 was matched as 100.0: Certain that this is Ben Goodger Larry Wall 1 was matched as 100.0: Certain that this is Larry Wall Larry Wall 2 was matched as 100.0: Certain that this is Larry Wall Larry Wall 3was not matched my the face API Bjarne Stroustrup 1 was matched as 99.74: Certain that this is Bjarne Stroustrup Bjarne Stroustrup 2 was matched as 99.9: Certain that this is Bjarne Stroustrup Bjarne Stroustrup 3 was matched as 100.0: Certain that this is Bjarne Stroustrup Rasmus Lerdorf 1 was matched as 99.99: Certain that this is Rasmus Lerdorf Rasmus Lerdorf 2 was matched as 100.0: Certain that this is Rasmus Lerdorf Rasmus Lerdorf 3 was matched as 100.0: Certain that this is Rasmus Lerdorf Niklaus Wirth 1 was matched as 99.98: Certain that this is Niklaus Wirth Niklaus Wirth 2 was matched as 99.99: Certain that this is Niklaus Wirth Niklaus Wirth 3 was matched as 100.0: Certain that this is Niklaus Wirth shigeru miyamoto 1 was matched as 99.99: Certain that this

is Shigeru Miyamoto shigeru miyamoto 2 was matched as 99.87: Certain that this is Shigeru Miyamoto shigeru miyamoto 3 was matched as 99.99: Certain that this is Shigeru Miyamoto Howard Aiken 1 was matched as 99.97: Certain that this is Howard H. Aiken Howard Aiken 2 was matched as 91.14: Certain that this is Howard H. Aiken Howard Aiken 3was not matched my the face API Frances E Allen 1 was matched as 99.94: Certain that this is Frances E. Allen Frances E Allen 2 was matched as 99.39: Certain that this is Frances E. Allen Frances E Allen 3 was matched as 99.97: Certain that this is Frances E. Allen John Atanasoff 1was not matched my the face API John Atanasoff 2 was matched as 93.75: Certain that this is John Vincent Atanasoff John Atanasoff 3 was matched as 99.99: Certain that this is John Vincent Atanasoff Charles Babbage1was not matched my the face API Charles Babbage2was not matched my the face API Charles Babbage3was not matched my the face API John Backus1 was matched as 99.97: Certain that this is John Backus John Backus2 was matched as 100.0: Certain that this is John Backus John Backus3 was matched as 99.19: Certain that this is John Backus Corrado bohm1was not matched my the face API Corrado bohm2was not matched my the face API Corrado bohm3was not matched my the face API Fred Brooks1 was matched as 100.0: Certain that this is Fred Brooks Fred Brooks2 was matched as 99.87: Certain that this is Fred Brooks Fred Brooks3 was matched as 100.0: Certain that this is Fred Brooks Vannevar Bush1was not matched my the face API Vannevar Bush2 was matched as 100.0: Certain that this is Vannevar Bush Vannevar Bush3 was matched as 99.85: Certain that this is Vannevar Bush Vint Cerf1was not matched my the face API Vint Cerf2was not matched my the face API Vint Cerf3 was matched as 98.82: Certain that this is Vint Cerf Noam Chomsky1was not matched my the face API Noam Chomsky2 was matched as 99.1: Certain that this is Noam Chomsky Noam Chomsky3 was matched as 99.84: Certain that this is Noam Chomsky Edmund Clarke1 was matched as 98.02: Certain that this is Edmund M. Clarke Edmund Clarke2 was matched as 100.0: Certain that this is Edmund M. Clarke Edmund Clarke3 was matched as 99.99: Certain that this is Edmund M. Clarke Lynn Conway 1 was matched as 95.18: Certain that this is Lynn Conway Lynn Conway 2was not matched my the face API Lynn Conway 3was not matched my the face API Stephen Cook1 was matched as 98.14: Certain that this is Stephen Cook Stephen Cook2was not matched my the face API Stephen Cook3was not matched my the face API Edsger Dijkstra 1was not matched my the face API Edsger Dijkstra 2 was matched as 99.95: Certain that this is Edsger W. Dijkstra Edsger Dijkstra 3 was matched as 99.97: Certain that this is Edsger W. Dijkstra Ernest Allen Emerson 1 was matched as 99.99: Certain that this is E. Allen Emerson Ernest Allen Emerson 2 was matched as 97.98: Certain

that this is E. Allen Emerson Ernest Allen Emerson 3was not matched my the face API Douglas Engelbart1 was matched as 99.97: Certain that this is Douglas Engelbart Douglas Engelbart2 was matched as 100.0: Certain that this is Douglas Engelbart Douglas Engelbart3 was matched as 99.97: Certain that this is Douglas Engelbart Federico Faggin1was not matched my the face API Federico Faggin2 was matched as 99.99: Certain that this is Federico Faggin Federico Faggin3was not matched my the face API Elizabeth Feinler1 was matched as 99.9: Certain that this is Elizabeth J. Feinler Elizabeth Feinler2was not matched my the face API Elizabeth Feinler3was not matched my the face API Tommy Flowers1 was matched as 91.92: Certain that this is Tommy Flowers Tommy Flowers2 was matched as 99.67: Certain that this is Tommy Flowers Tommy Flowers3was not matched my the face API Sally Floyd1was not matched my the face API Sally Floyd2was not matched my the face API Sally Floyd3was not matched my the face API Charles Sanders Pierce1 was matched as 88.08: Certain that this is Charles Sanders Peirce Charles Sanders Pierce2was not matched my the face API Charles Sanders Pierce3was not matched my the face API Stephen Furber1 was matched as 99.92: Certain that this is Steve Furber Stephen Furber2was not matched my the face API Stephen Furber3was not matched my the face API Sophie Wilson1was not matched my the face API Sophie Wilson2was not matched my the face API Sophie Wilson3 was matched as 100.0: Certain that this is Sophie Wilson Seymour Ginsburg1was not matched my the face API Seymour Ginsburg2 was matched as 99.98: Certain that this is George Gurdjieff Seymour Ginsburg3 was matched as 99.97: Certain that this is George Gurdjieff Susan L graham1 was matched as 100.0: Certain that this is Susan L. Graham Susan L graham2 was matched as 100.0: Certain that this is Susan L. Graham Susan L graham3 was matched as 99.97: Certain that this is Susan L. Graham Jim Gray1 was matched as 99.96: Certain that this is Jim Gray Jim Gray2was not matched my the face API Jim Gray3 was matched as 99.18: Certain that this is Jim Gray Barbara Grosz1was not matched my the face API Barbara Grosz2was not matched my the face API Barbara Grosz3was not matched my the face API Margaret Hamilton1 was matched as 99.99: Certain that this is Margaret Hamilton Margaret Hamilton2 was matched as 99.99: Certain that this is Margaret Hamilton Margaret Hamilton3 was matched as 99.08: Certain that this is Margaret Hamilton Geoffrey Hinton1 was matched as 99.99: Certain that this is Geoffrey Hinton Geoffrey Hinton2 was matched as 100.0: Certain that this is Geoffrey Hinton Geoffrey Hinton3 was matched as 100.0: Certain that this is Geoffrey Hinton Charles Anthony Richard Hoare 1 was matched as 99.99: Certain that this is Tony Hoare Charles Anthony Richard Hoare 2 was matched as 99.81: Certain that this is Tony Hoare Charles Anthony

Richard Hoare 3was not matched my the face API Betty Holberton1was not matched my the face API Betty Holberton2was not matched my the face API Betty Holberton3was not matched my the face API Harry Huskey1was not matched my the face API Harry Huskey2 was matched as 86.47: Certain that this is Harry Huskey Harry Huskey3was not matched my the face API Kenneth Iverson1was not matched my the face API Kenneth Iverson2was not matched my the face API Kenneth Iverson3was not matched my the face API Karen Sparck Jones1 was matched as 99.96: Certain that this is Karen Spärck Jones Karen Sparck Jones2was not matched my the face API Karen Sparck Jones3was not matched my the face API Jacek Karpinski 1was not matched my the face API Jacek Karpinski 2was not matched my the face API Jacek Karpinski 3was not matched my the face API Alan Kay1 was matched as 100.0: Certain that this is Alan Kay Alan Kay2 was matched as 99.94: Certain that this is Alan Kay Alan Kay3 was matched as 96.82: Certain that this is Alan Kay Stephen Cole Kleene1was not matched my the face API Stephen Cole Kleene2was not matched my the face API Stephen Cole Kleene3was not matched my the face API Donald Knuth1was not matched my the face API Donald Knuth2was not matched my the face API Donald Knuth3 was matched as 99.99: Certain that this is Donald Knuth Leslie Lamport1 was matched as 99.88: Certain that this is Leslie Lamport Leslie Lamport2 was matched as 99.93: Certain that this is Leslie Lamport Leslie Lamport3 was matched as 99.56: Certain that this is Leslie Lamport Barbara Liskov1 was matched as 99.99: Certain that this is Barbara Liskov Barbara Liskov2 was matched as 100.0: Certain that this is Barbara Liskov Barbara Liskov3 was matched as 99.93: Certain that this is Barbara Liskov Ada Lovelace1was not matched my the face API Ada Lovelace2was not matched my the face API Ada Lovelace3was not matched my the face API John McCarthy1was not matched my the face API John McCarthy2 was matched as 100.0: Certain that this is John McCarthy John McCarthy3 was matched as 99.88: Certain that this is John McCarthy Marvin Minsky1was not matched my the face API Marvin Minsky2 was matched as 100.0: Certain that this is Marvin Minsky Marvin Minsky3was not matched my the face API Yoshiro Nakamatsu1 was matched as 93.03: Certain that this is Yoshiro Nakamatsu Yoshiro Nakamatsu2 was matched as 96.58: Certain that this is Yoshiro Nakamatsu Yoshiro Nakamatsu3was not matched my the face API Gottfried Leibniz 1was not matched my the face API Gottfried Leibniz 2was not matched my the face API Gottfried Leibniz 3was not matched my the face API Peter Naur1 was matched as 84.51: Certain that this is Peter Naur Peter Naur2was not matched my the face API Peter Naur3was not matched my the face API John Von Neumann1was not matched my the face API John Von Neumann2 was matched as 100.0: Certain that this is John von Neu-

mann John Von Neumann3 was matched as 99.98: Certain that this is John von Neumann Kristen Nygaard1 was matched as 99.99: Certain that this is Kristen Nygaard Kristen Nygaard2was not matched my the face API Kristen Nygaard3was not matched my the face API Radia Perlman1 was matched as 100.0: Certain that this is Radia Perlman Radia Perlman2 was matched as 99.98: Certain that this is Radia Perlman Radia Perlman3 was matched as 99.83: Certain that this is Radia Perlman Pier Giorgio Perotto1was not matched my the face API Pier Giorgio Perotto2was not matched my the face API Pier Giorgio Perotto3was not matched my the face API Rosalind Picard1 was matched as 100.0: Certain that this is Rosalind Picard Rosalind Picard2 was matched as 99.88: Certain that this is Rosalind Picard Rosalind Picard3 was matched as 99.74: Certain that this is Rosalind Picard Dennis Ritchie1 was matched as 99.84: Certain that this is Dennis Ritchie Dennis Ritchie2was not matched my the face API Dennis Ritchie3 was matched as 99.28: Certain that this is Dennis Ritchie Bertrand Russel1was not matched my the face API Bertrand Russel2was not matched my the face API Bertrand Russel3was not matched my the face API Claude Shannon 1was not matched my the face API Claude Shannon 2was not matched my the face API Claude Shannon 3 was matched as 99.99: Certain that this is Claude Shannon Masatoshi Shima1was not matched my the face API Masatoshi Shima2was not matched my the face API Masatoshi Shima3was not matched my the face API Herbert A simon1 was matched as 99.94: Certain that this is Herbert A. Simon Herbert A simon2 was matched as 100.0: Certain that this is Herbert A. Simon Herbert A simon3 was matched as 99.98: Certain that this is Herbert A. Simon Richard Stallman1 was matched as 99.99: Certain that this is Richard Stallman Richard Stallman2 was matched as 100.0: Certain that this is Richard Stallman Richard Stallman3 was matched as 99.97: Certain that this is Richard Stallman Michael Stonebreaker1 was matched as 99.99: Certain that this is Michael Stonebraker Michael Stonebreaker2 was matched as 99.99: Certain that this is Michael Stonebraker Michael Stonebreaker3 was matched as 99.36: Certain that this is Michael Stonebraker Ivan Sutherland 1 was matched as 98.97: Certain that this is Ivan Sutherland Ivan Sutherland 2 was matched as 92.28: Certain that this is Ivan Sutherland Ivan Sutherland 3 was matched as 99.87: Certain that this is Ivan Sutherland Chai Keong Toh1was not matched my the face API Chai Keong Toh2was not matched my the face API Chai Keong Toh3was not matched my the face API An Wang1 was matched as 100.0: Certain that this is An Wang An Wang2 was matched as 100.0: Certain that this is An Wang An Wang3 was matched as 100.0: Certain that this is An Wang Willis Howard Ware1was not matched my the face API Willis Howard Ware2was not matched my the face API Willis Howard Ware3was not matched my the face API Maurice

Wilkes 1was not matched my the face API Maurice Wilkes 2was not matched my the face API Maurice Wilkes 3 was matched as 99.66: Certain that this is Maurice Wilkes Konrad Zuse1 was matched as 99.25: Certain that this is Konrad Zuse Konrad Zuse2 was matched as 100.0: Certain that this is Konrad Zuse Konrad Zuse3was not matched my the face API Judea Pearl1 was matched as 100.0: Certain that this is Judea Pearl Judea Pearl2was not matched my the face API Judea Pearl3 was matched as 100.0: Certain that this is Judea Pearl Michael Dell1 was matched as 99.62: Certain that this is Michael Dell Michael Dell2 was matched as 100.0: Certain that this is Michael Dell Michael Dell3 was matched as 100.0: Certain that this is Michael Dell Lawrence Lessig1 was matched as 98.88: Certain that this is Lawrence Lessig Lawrence Lessig2 was matched as 99.97: Certain that this is Lawrence Lessig Lawrence Lessig3 was matched as 99.88: Certain that this is Lawrence Lessig Andrew Ng1was not matched my the face API Andrew Ng2 was matched as 92.47: Certain that this is Andrew Ng Andrew Ng3was not matched my the face API John Carmack1 was matched as 99.69: Certain that this is John Carmack John Carmack2 was matched as 99.78: Certain that this is John Carmack John Carmack3 was matched as 99.98: Certain that this is John Carmack Ken Jennings1 was matched as 100.0: Certain that this is Ken Jennings Ken Jennings2 was matched as 99.91: Certain that this is Ken Jennings Ken Jennings3was not matched my the face API Guido Van Rossum1 was matched as 83.5: Certain that this is Guido van Rossum Guido Van Rossum2 was matched as 79.94: Certain that this is Guido van Rossum Guido Van Rossum3 was matched as 99.69: Certain that this is Guido van Rossum joseph carl robnett licklider 1 was matched as 99.86: Certain that this is J. C. R. Licklider joseph carl robnett licklider 2 was matched as 100.0: Certain that this is J. C. R. Licklider joseph carl robnett licklider 3 was matched as 100.0: Certain that this is J. C. R. Licklider Richard Hamming1 was matched as 99.99: Certain that this is Richard Hamming Richard Hamming2was not matched my the face API Richard Hamming3was not matched my the face API Cuthbert Hurd1was not matched my the face API Cuthbert Hurd2was not matched my the face API Cuthbert Hurd3was not matched my the face API ramon llull 1was not matched my the face API ramon llull 2was not matched my the face API ramon llull 3was not matched my the face API the very famous category matched 27 out of 30 Pictures with an average confidence rating of 99.43220769917524 a total of 10 People were matched out of 10.0 the medium famous category matched 36 out of 45 Pictures with an average confidence rating of 98.54813764492671 a total of 15 People were matched out of 15.0 the low famous category matched 121 out of 225 Pictures with an average confidence rating of 98.71676529734587 a total of 58 People were matched out of 75.0

S Appendix 19: List of 100 computer scientists used for Test 3

–Very Famous–

Bill Gates Steve Jobs Elon Musk Mark Zuckerberg Larry Page Sergey Brin Tim Berners Lee Alan Turing Steve Wozniak Linus Torvalds

–Well Known–

James Gosling Grace Hopper Martin Hellman Michael Widenius Yukihiro Matsumoto John Resig Brian Kernighan Ken Thompson David Axmark Ben Goodger Larry Wall Bjarne Stroustrup Rasmus Lerdorf Niklaus Wirth shigeru miyamoto

—Industry Recognised— Howard Aiken Frances E Allen John Atanasoff Charles Babbage John Backus Corrado bohm Fred Brooks Vannevar Bush Vint Cerf Noam Chomsky Edmund Clarke Lynn Conway Stephen Cook Edsger Dijkstra Ermest Allen Emerson Douglas Engelbart Federico Faggin Elizabeth Feinler Tommy Flowers Sally Floyd Charles Sanders Pierce Stephen Furber Sophie Wilson Seymour Ginsburg Susan L graham Jim Gray Barbara Grosz Margaret Hamilton Geoffrey Hinton Charles Anthony Richard Hoare Betty Holberton Harry Huskey Kenneth Iverson Karen Sparck Jones Jacek Karpinski Alan Kay Stephen Cole Kleene Donald Knuth Leslie Lamport Barbara Liskov Ada Lovelace John McCarthy Marvin Minsky Yoshiro Nakamatsu Akira Nakashima Peter Naur John Von Neumann Kristen Nygaard Radia Perlman Pier Giorgio Perotto Rosalind Picard Dennis Ritchie Bertrand Russel Claude Shannon Masatoshi Shima Herbert A simon Richard Stallman Michael Stonebreaker Ivan Sutherland Chai Keong Toh An Wang Willis Howard Ware Maurice Wilkes Konrad Zuse Judea Pearl Michael Dell Lawrence Lessig Andrew Ng John Carmack Ken Jennings Guido Van Rossum Seymour Ginsburg Richard Hamming Cuthbert Hurd JCR licklider