Software Engineering Group Project

COMP2002 / G52GRP: Final Report

**Project information:**

Project title: UoN Security Training

Project sponsor: Steven Bagely

Academic supervisor: Steven Bagely

Industry Sponsor: (if applicable)

Industry Supervisor(s): (first name, last name), (first name, last name)

**Team information:**

Team number: 43

Rafi Brooks, 20273199, psyrb6

Salman Alshayeb, 20252679, psysa6

Benjamin Jessel, 20255224, psybj2

Dewei Wang, 20126335, scydw1

Leo Meyler, 20273623, psylm10

Yichen Lu, 20215622, scyyl15

**Documentation (links):**

Code repository: https://projects.cs.nott.ac.uk/comp2002/2021-2022/team43\_project/-/tree/main/src

Document repository: https://projects.cs.nott.ac.uk/comp2002/2021-2022/team43\_project/-/tree/main/Docs

Table of Contents goes here

Contents

Part 1: Final Report

**Term 1 Interim**

## Project Background & Understanding

Our project brief outlined two main requirements for us to complete the project. The first is that we research and understand cyber security threats that face students and how students can protect themselves against these threats. This was then to be compiled to explain it to students in a way that anyone can understand. The second requirement was to develop a platform that this content could be delivered in a way that is interesting for student to complete as part of their induction media.

From this brief we decided that the project has two areas for development; an expandable website for content to be added and delivered to students that potentially have very little computer experience; and content deliverables, either through games or explanations of threats that can allow the students to learn about what issues affect them whilst still being interesting and engaging. We completed a textual analysis and drafted a set of requirements for the project and after a meeting with our project sponsor we settled on a main focus for the project being an educational but fun phishing game.

The university has struggled with phishing scams being sent to students of the past year with many emails being sent to all students warning us of current scams that other students had fallen victim to. Because of this we decided that our phishing game would consist of an email browser that would be sent emails of varying danger, some being obvious scams whilst some are less subtle. We felt this would fit the project brief by providing a fun way for students to learn while also allowing them to learn about email security in a situation that they are familiar with by using an email browser for the game. This would allow students to become increasingly confident when faced with a real scam or phishing email as they will have faced similar emails before.

The research for our project began by searching for existing cyber security training. We soon found that most existing resources were simply articles providing ‘tips’ for how to avoid these scam emails or quizzes that gave no hands on experience with dealing with actual emails. We discovered one phishing quiz that was similar to our idea as it had example emails that you then had to select whether they were legitimate or phishing emails however they quiz had a selection of 8 questions that were used in the same order each time you took the quiz. It was also not as engaging and fun as the project we hope to create. We did however like some aspects, such as the explanations for each incorrect and correct answer and the overall email format which we intend to incorporate into our final product.

# Requirements and critical path

Our initial pitch for the project was to create a website which was both informative and interactive. we decided that we would have a home page which would be split into multiple sections about cybersecurity; each would have a information teaching the user and then a small interactive game. After showing our initial design and sharing our ideas for the project during a meeting with the supervisors a clear aspect of the project became a priority. We decide to change the scope from multiple small games to one large email game which would have more functionality, we felt this gamification of cybersecurity issues was more likely to encourage people to learn so that instead of having to force people to do the training they would happily do it voluntarily.

Another aspect of the project that we decide would be too much work for the benefit it provided was the implementation of a log in system. We initially thought this would be beneficial as it would allow the owner to see who had actually completed the cybersecurity training and what progress people had made. This was not a requirement for our supervisors so we decided it would be best to focus our resources on other functionality.

Because we are deciding to implement our project as a web app there is a multitude of software available to use. Originally, we were looking at using Google Web Designer for the project but after a bit of experimenting we decided it wasn’t the right thing. For the mock-up we used Adobe XD and to create the functioning prototype we simply used html, java script and CSS.

When writing the code for the prototype we realised that to create the fake email being sent to the user we were having to hardcode a lot of information, this is not ideal; as such, if we decided to continue with this pure html and JS implementation, we will use JSON files to store our data so that the necessary information for the dummy emails can be stored compactly outside of our html but can still be accessed quickly to ensure the functionality of the game. Furthermore, we may decide to create our game in the development kit UNITY, this may allow us to create a more polished game with functions which would not be practical to implement if coding from scratch ourselves, we could then host this on our website. We need to do more research into this to check if it is practical and if it is accessible enough on the different devices students have accesses to.

Our current requirements moving forward are:

1. Have a platform to inform students on cybersecurity threats
2. Have information on threats relevant to students
3. Create the framework for the website such that it is expandable
4. Work on devices with different screen sizes and resolution
5. Work on different operating systems and in different browsers
6. A simple interface that is intuitive and easy to navigate
7. Not have a high required laptop specification to run so that it is available to all students regardless of device
8. be accessible:
   1. should be compatible with text reading software for the visually impaired

-may not be able to implement on everything due to the nature of the game.

* 1. should follow standards for those with colour-blindness
  2. compatible with translation software to help those with English as a second language

-may not be able to implement on everything due to the nature of the game.

1. able to run on smartphones(this is a stretch goal)

# Initial Software Implementation & Testing

## Wireframes

Graphical user interface

Description automatically generated

1：LOGO of Security Website and its partner

2: Background, miniature

3: Topic button, for four parts: Introduction, Passwords, Phishing, social media

4：diagram for security issues online

5: description of recent security breach

6: Button for entering different parts

7: Remaining content including big event, tips or collaborator

Description: This is the home page for the whole website, the main part for introduction of security training, and providing different entrance for each game web.

Graphical user interface, text, application

Description automatically generated

Description: Entrance for different game options and it will jump to corresponding game page, which is designed for testing visitors’ knowledge base through answering questions.

**Instruction pop-up:**

Graphical user interface, text

Description automatically generated

1. When the player is first introduced to the game, they won’t know what’s going on or what to do. Before entering the game, this pop-up window will appear on screen. It will be filled with the instructions for the game
2. The button at the bottom of the screen will be pressed when the player is satisfied they know how to play the game.

**Phishing Game:**

Graphical user interface, application

Description automatically generated

1. The whole game is meant to directly mimic a real email client. As we can see, the side bar shows the different email folders that would normally be there. There is a counter next to the folders to indicate the unread emails as the score will be determined by whether the player can read the emails before they get to a certain number of unread emails (emails will constantly be coming in). The player will mainly play in the inbox. Depending on how the implementation goes, the player will delete all the emails that they think are dangerous (the other idea we have for the implementation is in 3.). There is also a junk folder which will mainly be filled with fake emails but will have the occasional useful email to mimic what happens in real life.
2. This is the thumbnail of the email, showing the name of the sender and the subject line.
3. This is the other idea we had for implantation. Instead of deleting the emails, they will flag the emails they think is dangerous, and will leave the safe emails
4. This is where the player will see the full email address of the sender. This is crucial in defining if the email is dangerous or not.
5. An attachment bar is something else that is standard in any email client. Not all emails will have attachments, however the dangerous emails that have attachments will penalise the player if clicked on.
6. The body of the email is here. This is the most important part of the email to tell if the email is phishing or regular. It will be the main way the player will determine if the email is dangerous or not.

## Current Git图形用户界面, 文本, 应用程序, 电子邮件 描述已自动生成Graphical user interface, text, application, email Description automatically generated

In a collaborative environment, branches can always help us to share and work on the same course code and share documents easily. While someone will be fixing bugs, others will be implementing new features or adjusting the game style, etc. With so much going on, branches allow each of us to branch out from the original code and isolate different features from the base. Moreover, branches also make it more convenient for us to merge versions later on.

In the design and development phase, we divided branched into three parts, which are named as main, intro-web and the game-design, and each part is responsible for its own aspects as master, homepage and the game。

## Current Phishing Game Code

Currently, we have made a small prototype to what the phishing game will look like with little implantation of functionality.

**Output:**

Graphical user interface, application, Word

Description automatically generated

As you can see, it’s very similar to our wireframe mock-up. It contains the side bar with links to the different folders and also an anonymous picture. The middle section contains a grid where emails will come in at random in intervals. These can be interacted with by clicking on them and this will change the contents section to “this is the main body of the email with id …” depending on the id of the email. When we fully implement all the functions, this will show email, subject, attachments, and the main body.

**HTML:**

Text

Description automatically generated with low confidence

This is the current HTML code that we have. As seen, it is clearly commented what each section does. The HTML makes up the main elements of our website (sidebar, grid of emails and contents section).

**CSS:**

Graphical user interface, text, application

Description automatically generated

(This isn’t all the CSS)

This is a typical CSS document. As you can see in the output screenshot, there isn’t too much styling: this will be improved on as we progress. The main use of the file at the moment is for the positioning of the elements

**JavaScript:**

Graphical user interface, text, application

Description automatically generated

The JavaScript we currently have controls the emails that are coming through. It randomises the emails that are in an array and there is a function that puts the array elements in the grid. There is also a function that allows the element to be clicked on and the body of it to appear.

# Project management and progress

Throughout the project we have used an agile approach, specifically a scrum methodology. This allows us to have frequent meetings to stay updated and divide work. we currently meet every Monday morning during the lab and again on Thursday on a teams call to discuss our progress and bring up any problems we encountered so that they can be resolved before we meet again. By having two meetings a week we also reduced the impact of dependencies between tasks, for example one week we had two goals - create wireframes and a mock-up. wireframes were required to make an initial mock-up of how the website might look; Instead of having to wait a full week to start on the mock-up, we were able to complete both tasks within the week.

Records of our meetings can easily be accessed from the readme file on our git and evidence of meetings seen on our teams group. Furthermore, while originally using git to create issues to organise what we are all doing we have since moved to Trello as we found the software completer and more accessible to us. Informally we stay in touch over a group WhatsApp, which we use frequently to resolve scheduling issues and co-ordinate when working in pairs to complete tasks.

Due to the fact that we are developing a website to host a game, unit testing is very difficult, instead a user testing approach is required to gauge feedback on the design and implementation of the game. Our agile methodology has helped greatly with this as when developing our prototype we were able to give advice to each other weekly and gain feedback from friends after completing sprints to dictate how to move forward with the design.

Throughout our initial design period for the project we split the team effectively based on unique skills, for example:

Shayeb did research into relevant cybersecurity risks that would be appropriate for university students and also which were feasible to implement on our website while following our goals of being, interactive, interesting, and informative. This ended up with doing much more in depth analysis about phishing emails and email security in general after it became a keen aspect of the project for the client; changing our specifications from having multiple smaller games implemented on a website to one larger game with the framework in place to extend and add more games in the future.

Rafi was responsible for creating some initial wireframes for the website displaying how the layout should be displayed and how different webpages should link together, because of his previous experience gained at a-level.

Yichen and Dewei who have experience using a web development tool we hadn’t heard of, Adobe XD, and were able to create a mock-up website that would go on to dictate how the webpage would look and display information to the user as well as the design of the email game specifically.

Ben, who took interest in how the game would work and the functionality behind created diagrams to display different game states and how the user would interact with the game, this provided an initial idea that was then built upon as a group after feedback from the project supervisors.

Leo created the initial functioning prototype for the game in html and java script which allowed us to work out issues with our plan and how we could resolve them as well as what ways to improve implementation; for example we are now looking at JSON files to store data that needs to easily be accessed for the functionality of the game.

# Preliminary reflections

At the start of the year we were only meeting once a week, we discovered this was an inefficient use of our time as we did not need that long to complete deadlines; as a result we started having teams calls on Thursday to help with organisation and to reassign.

So far the only major technical difficulty has been with accessing the git repository. Some members had a hard time connecting to the git during the first few weeks due to the newly required SSH access, as a result we found organising tasks on git to be difficult, so we offloaded this organisation requirement to some software called Trello. We found this much more practical to use and stay up to date with what each of us were doing. Another issue faced with the git was some members uploading docx and zip file types, this was an issue as it meant easily viewing project files was not possible, this has been resolved and we now almost exclusively use markdown files to store information.

The latest significant update to our progress so far has been the creation of a working prototype, which can be used to demonstrate our proposed design for the security training project. Unfortunately, we were unable to get the prototype to a reasonable state before our last meeting for the term with our supervisors; as a result we will have to continue working on it without feedback, making sure to carefully stick to the brief.

While creating the initial prototype we realised that splitting up the work of coding was quite difficult, as the project was quite small it resulted in only one person coding at a time and a disproportionate amount of work being done by one person. Going forward we will be splitting the work into coding the website for the game and coding the website for the home page, this will allow us to split into two smaller groups. From here if we are unable to divide the work again, we will enforce paired programming so that the load of work is shared between people.

# Conclusion

So far we have defined our requirements and specifications, created designs and documents to show our ideas to our supervisor which has allowed the project to naturally evolve and change scope. Since these meetings we have gone on to develop a working prototype which we are eager to share in the new year.

Going forward we will look into more software that could be of use to us or change the way we decide to implement our design and work as a team to tackle the issues highlighted in this report. Our end goal is to create a robust piece of software with the hops of it actually having real world application and practical use by the university.

**Term 2 Changes**

# Reflection on Success and Achievements

As we reach the end of our contract, it’s easy to pinpoint our success and the achievements we’ve made. The best way to reflect upon this is to go over the achieved requirements. The first and main requirement was to have a platform to inform students on cyber security threats and we believe we’ve done that. Our project contains 4 informative pages, each covering a different cyber security threats which goes in depth to explain what these threats are and how to combat them. Furthermore, our phishing game is an excellent way for the user to learn how to spot phishing emails, which is one of the biggest threats facing students.

Another achievement is the intuitive and easy to navigate design of both the website and the game. The website has a consistent header and footer which is easily recognisable to anyone who has used a website. The header shows the logo for the website, along with navigation links to the homepage, the other info pages, and straight to the main game. This makes it very easy for the user to navigate to any page at any point on the website. The footer is just for design purposes as it indicates the bottom of the page but it’s also consistent through the whole website.

The phishing game simulates an email client with emails coming in at regular intervals. The client is loosely based off of Outlook. It’s split into three parts with unequal sizes, each size correlating the needed space needed for the elements. On the left, is the menu for the different email folders (which isn’t functional and purely there for design). The middle section, which is slightly bigger than the menu section, is where the emails appear. The last and biggest section is where the content of the email appears when one is selected. It shows the name of the sender, the email address, an attachment, and the text section. This design is very recognisable to anyone who has ever used an email client, which the target audience almost definitely have.

The functionality of the email game is also a great achievement. It uses a clever mix of html to create the visual elements of the client, like the layout of the page, the way emails are encapsulated in email titles, and the layout of the email and also JavaScript to make it interactive and implement the timings of the emails coming in. The content of the emails are all stored in a json file, allowing it to be separated from the actual code and improving the layout and extendibility of the program. It simulates how an email client runs very well.

# How Requirements Have Changed

Yichen

A description of how requirements have changed over the course of the project and how the team dealt with this.

# Change in Methods and Team Approach

As we have progressed through project, our methods for working and our general approach has gone through a few changed. When we started the project, we had meetings once or twice a week and individually assigned work to each other. As it evolved, more collaborative work was needed. So we used paired programming to increase our efficiency. We had two people working on the code for the phishing game, two people working on the template and layout of the website, and then two people working on the information for the website. Out of the three tasks, the coding takes significantly longer than the other two tasks. This made the other people to all collectively work on finishing off the website, like implementing the information on to the information pages and tweaking the styling with the CSS file.

The git repository took us a while to understand. It was unorganised and we weren’t using it to its full capability. We had a few different branches for the website, and it was making the whole thing very confusing to keep track of who was working on what. We eventually had one branch that the website was being worked on and the main branch where the phishing game was being worked on. We also started to use the issue feature a lot more. We would create a bunch of issues, assign them a priority, assign them to an individual and then when someone wanted to work on an issue, they would create a branch to avoid conflicting commits. Once the issue was completed, a merge request was made and approved back into whatever branch it came from. This greatly helped our efficiency by helping organise our work and allowing us to keep track of what needed to be done in one place.

Overall, we had a bit of a rocky start to the project when it came to organisation. The git wasn’t organised, and people were confused on what they were doing. As we progressed through the project, we became better at organising the project and our team approach was more efficient, allowing us to finish the project.

# Motivation Behind Technical Approaches

Dewei

A motivation behind the technical approaches that the team used in the implementation and a motivation for them.

# Reflection on Project

Rafi

A reflection on what went well, what was hard to achieve, what went badly, what the team has learned from this, and how the team’s functioning could be improved.

# Overview of Work Split

Dewei

An overview of how the work was split between different group members.

# Reflection on Management

Rafi

A reflection on the management of the project (sprints, sprint planning, retros).

# Reflection on Contingency

Rafi

A reflection on how contingency measures helped to deal with unexpected circumstances, or what contingency measures you would take in hindsight to better deal with unexpected circumstances.

# Future Directions

Yichen

• Promising future directions for the project.

Part 2: Software Manual

# Introduction and Scope

Cyber security has become an increasing threat to university students over the last few years, with various scams and malicious files being shared via a variety of sources. One of the most prevalent of these sources is phishing emails, which the University of Nottingham was affected greatly by in the 2020/2021 academic year. Our projects seeks to educate university students about common cyber security threats, focusing on phishing emails, in an engaging way.

We created an email browser based game that students can use to recognise common phishing techniques, allowing them to engage and interact with various emails in a safe environment. The goal of our project was to create an expandable cyber security framework that includes an educational but engaging game to teach students about phishing scams.

# Project Terminology

Definitions of project specific terminology and jargon.

# Overview of Structure

An overview of the high level structure / concepts that underpin the code.

The game consists of an email inbox that fills with emails over time. These emails can be opened and judged to be either phishing attempts or safe regular emails. Alternatively you can also interact with the email, by either downloading files or clicking links which will give a greater point incentive for safe emails but will result in a game over for phishing emails.

To create emails in the inbox we use a function *addEmailButton()* which acts as an email button factory by selecting a random email from the

# Architectural Design

# Overview of Technologies

An overview of data storage mechanisms / technologies / frameworks that were used, external dependences, libraries, proprietary systems and project specific hardware / software (with references to relevant documentation).

# Coding Conventions

Coding conventions, maintainability conventions, and design patterns that you have used.

# Project Testing

A description of project specific testing approaches and frameworks that you have implemented and an introduction on how to use them.

Part 3: User Manual

User manual goes here

User manual goes here

User manual goes here

User manual goes here

User manual goes here

User manual goes here

User manual goes here

User manual goes here

(if using in product manual, please specify how it can be found).