



School of Engineering, Computing, and Mathematics

## **COMP3000 Computing Project**

2020/2021

# **SafeGaze: Sports Facility Observation System**

Reece Davies

10572794

BSc (Hons) Computer Science

## Acknowledgments

I would like to thank my project supervisor, Kimberly Tam, for her incredible support, advice, and guidance that she offered throughout the entirety of this project which ultimately led to a superior end result of SafeGaze.

In addition, I would also like to express my gratitude to the founder and CEO of Exeter Trampoline Academy (ETA), Polly Buzzo-Johns, for acting as a client for SafeGaze, along with providing essential feedback on adjustments to the application's features that would prove to be the much needed alterations to result in the success of this project.

## **Abstract**

This document outlines the software project planning and development process of “SafeGaze”, a web application intended to improve both the physical and cybersecurity of sports clubs and facilities by offering a secure monitoring platform for its members. There are plans to publish this software application in the future and make it publicly available for any facilities that may require such features.

The report begins by discussing the background of SafeGaze, the issues of current livestream systems, and exactly what this project aims to achieve. Subsequently, the legal, social, ethical, and professional issues are then explored alongside with what precautions have been made to ensure those were met.

The main body of the report breaks down the planning and preparation of the software solution alongside with how the entirety of the project has been effectively managed with the use of relevant methods, tools and resources. Furthermore, the different phases of the software’s development life cycle are presented, where each subsection examines the sprint’s objectives and a summary of what challenges were presented and overcome.

Lastly, the report concludes with an end-project report and a post-mortem, where it evaluates the project’s management and development process, highlighting what actions could have been done differently for a more effective outcome.

# Contents

Acknowledgments .....	2
Abstract .....	3
Statement of Word Count.....	7
Code Submission.....	7
1. Introduction .....	8
1.1 Background .....	8
1.2 Deliverable .....	8
1.3 Aims and Objectives.....	8
1.4 The Issues with Current Livestream Systems .....	9
2. Legal, Social, Ethical & Professional Issues.....	9
2.1 Data Protection & other Acts of Parliament.....	9
2.2 Ethical Issues surrounding Livestream Systems .....	10
2.3 Plagiarism of Intellectual Property.....	10
2.4 Software Licensing .....	10
3. Project Management .....	10
3.1 Importance of Project Management.....	10
3.2 Method of Approach.....	11
3.3 Supervisor Meetings & Bi-weekly Reports .....	13
3.4 Time & Risk Management.....	13
3.5 Justification of Technologies Used .....	13
3.5.1 Version Control: Git.....	13
3.5.2 Planned Backlog: Microsoft Office Planner .....	13
3.5.3 Student-Supervisor Contact & Bi-weekly Meetings: Microsoft Teams.....	14
4. Client Side Testing with ETA.....	15
5. Requirements .....	16
5.1 Functional Requirements .....	16
5.1.1 Facility .....	16
5.1.2 Member .....	16
5.2 Non-functional Requirements.....	17
6. System Architecture & Design .....	18
6.1 Use-Case Diagram .....	18
6.2 Flowchart .....	19
6.3 Architecture .....	20
6.4 UI Design .....	21
6.4.1 Facility .....	21

6.4.2 Member .....	24
6.5 Database Schema.....	25
6.6 Risk Plan .....	26
7. Implementation of Technologies & Resources .....	27
7.1 HTML, CSS, & JavaScript.....	27
7.2 Bootstrap .....	27
7.3 PHP.....	27
7.4 MySQL & PhpMyAdmin.....	27
8. Project Phases.....	28
8.1 Development Overview .....	28
8.2 Sprint 0: Planning & Preparation.....	28
8.3 Sprint 1: Facility Focused .....	29
8.4 Sprint 2: Member Focused .....	30
8.5 Sprint 3: Management Focused .....	31
8.6 Sprint 4: Livestream Focused .....	32
8.7 Sprint 5: Message Focused.....	34
8.8 Sprint 6: Testing .....	35
8.9 Sprint 6: Finalisation .....	36
9. ETA's Constructive Criticism.....	37
10. End Project Review .....	38
11. Project Post-Mortem .....	39
11.1 Project Management Evaluation.....	39
11.2 Objectives Evaluation.....	39
11.3 Technologies Evaluation .....	39
11.4 Development Performance Evaluation .....	40
11.5 Future Enhancements .....	40
12. Conclusion.....	41
13. References .....	42
14. Appendix.....	43
14.1 Documentation .....	43
14.2 Project Management .....	46
14.2.1 Backlog.....	46
14.2.2 Risk Plan .....	47
14.2.3 Bi-weekly Reports .....	49
14.2.4 Bi-weekly Meeting Minutes .....	59
14.2.5 Project Plan .....	60

14.2.6 Use Cases .....	62
14.3 Database .....	63
14.3.1 SQL Code .....	63
14.3.2 Table Normalisation .....	70
14.4 ETA's Feedback .....	71
14.4.1 Video Demonstration .....	71
14.4.2 April Meeting .....	72

## **Statement of Word Count**

Word count: 9,999

## **Code Submission**

Source code (invite only): [https://github.com/reecedavies/COMP3000-Assignment-Reece\\_Davies](https://github.com/reecedavies/COMP3000-Assignment-Reece_Davies)

# 1. Introduction

Due to the current status of the COVID-19 pandemic, there are certain restrictions of how many people can be within a given building, especially those intended for sport. This creates a complication for sports facilities where the participants are of a younger age, thus when separated from their parents, both may feel somewhat anxious. Notwithstanding, parents may potentially be given the opportunity to watch their child from outside or a designated viewing area. This therefore creates a security threat of strangers watching the children perform, and such incidences would be particularly troublesome for sports involving tight clothing, such as swimming or gymnastics.

## 1.1 Background

SafeGaze is directed towards the physical safety and cybersecurity of sports clubs and facilities. It specifically relates to the ethical issue of stalking or harassment of strangers and a sports facility's members, especially when the participants primarily consist of children. By implementing such a system, it will benefit the sports facility, the participants involved, and parents of younger participants.

The application is split into two separate parts, each of which is accessed through their respected login page – one for the facility and the other for its members. The member's section is primarily used for allowing parents/guardians to watch their children via one or numerous livestreams, thus eliminating the need for designated viewing areas. This being said, they would only be able to access the livestream within a designated time frame, which is dependent upon when their child is taking part in the activity provided by the facility. The second section would specifically behave as an administrator tool for the sports facility. From within this part of the application, the business would be able to manage their club members and livestreams, as well as issue important notifications or messages to its members that have been given access to an account under the facility's name.

Additionally, the facility would be able to mark the urgency of each message, which could potentially play an important role in announcing crucial information. Furthermore, members would also be able access important information regarding the facility, such as contact details, and street address, which may potentially be improved upon to also include information such as social media links.

With regards to the functionality of a facility managing its members through the application, the facility would be able to view the details of each member's account and alter its account status if deemed necessary. This means that the organisation would be able to remove any users that are no longer members within the facility. In addition, the facility would also be able to alter the times that a member has access to the livestream, which would be important if there was a change in the participant's training hours.

## 1.2 Deliverable

The final deliverable of the project has been implemented as a web application for both the facility and its members, with a possibility of development for a mobile application in the future. Nevertheless, this web app has successfully fulfilled the functional requirements set for it, as well the overall project objectives stated in section 1.3.

## 1.3 Aims and Objectives

**Project management:** Prove to have a great understanding of what is required to effectively plan, design, and manage a large-scale computing project, and see it to its completion in operation.

**Software project:** Introduce a software application to Plymouth University as a final year project, that aims to solve a common problem. This application allows a sports facility's members to safely watch

a secure livestream of their children engaging in such activities. Implementation of the deliverable should be completed by end of February 2021.

**Bi-weekly meetings and reports:** Allow the project supervisor ‘Kimberly Tam’ to be kept informed of current project progress by engaging in bi-weekly meetings, as well as submitting status reports in the form of a short description to the university. Status reports are available in the COMP3000 module page, which is accessible via the Plymouth University DLE.

**Showcase materials:** Create a project poster, software description, and video demonstration of the final deliverable by 22<sup>nd</sup> March 2021. University’s requirements must be met, as stated in the assignment brief. Showcase materials are used for the university’s project showcase, available on the Plymouth University website.

**Documentation:** Create a final year dissertation and have it completely finished by May 2021, consisting of no more than 10,000 words, and abiding to the university’s guidelines. This dissertation should include detailed explanations of the development and planning process of the project, detailed figures, references in Harvard referencing system, and an appendix.

#### 1.4 The Issues with Current Livestream Systems

There are numerous private livestream hosting platforms available to the public, however these often require a large lump sum down payment or monthly subscription to be able to use such software as a service. Furthermore, these software applications do not specifically allow for any modification to the livestream’s security with regards to what times it is accessible by the organisation’s members.

Additionally, these services might be somewhat intimidating to an unexperienced user, or someone with little technical expertise. On the other hand, SafeGaze is designed to be simple and easy to use, meaning – anyone will be able to understand what they can achieve with it, and the exact setup/configuration process. Moreover, SafeGaze allows for real-time configuration to manage its users in an effort to determine who has access to the livestreams and at what specific times.

### 2. Legal, Social, Ethical & Professional Issues

By referring to “legal, social, ethical, and professional issues”, it specifically focuses on the concept of what is deemed ‘right’ and ‘wrong’ in relation to the UKs rules and regulations around software development. Thomson, A. Schmoldt, D (2001) states “*Computer software systems lie at the heart of modern decision making, including data/information storage and manipulation, data availability, and ‘alternatives’ formulation and selection.*” Furthermore, the types of issues that arise are directly linked to the system in question, in terms of how it has been created/implemented, what it specifically focuses on, and what technologies are used in the development process. This is particularly evident when such a system is incorporated into private or public organisations. As a result, considerations must be taken into account when developing and deploying any major computing project.

#### 2.1 Data Protection & other Acts of Parliament

It is crucial that this project abides by the UK’s rules and regulations regarding relevant Acts of Parliament, as failing to do so will incur severe problems relating to legal authority. With the rise of sophistication of IT systems and technology devices, it is apparent that so does the use of its users’ personal data. As technology evolves and advances within the business sector, corresponding laws must adapt to protect its users from criminal infringement of their rights and personal safety. The “Data Protection Act (2018)” specifically relates to how one’s personal information is used by organisations; and as this project heavily relies on its users personal data, it is therefore important that this Act is strictly adhered to. Failing to follow such laws would have a detrimental impact on both

the organisation and users in question because it leads to the possibility of said information being misused by third-parties for acts of fraud. Using the Data Protection Act as a key example, this project strictly follows the UK's laws surrounding the development and deployment of software systems, thus ensuring compliance with relevant Acts of Parliament.

## 2.2 Ethical Issues surrounding Livestream Systems

By transmitting live video footage to the internet, various ethical issues are manifested, particularly when the participants primarily consist of children. It is extremely important that Safeguarding & Child Protection procedures are emphasised by the facility or facilities using SafeGaze. As a result, this ensures that every child, regardless of age, disability, gender, race, religion, belief, or sexual orientation, is not subjected to the possibility of any physical or psychological harm/distress. In instances where a facility implements SafeGaze, it is imperative that persons responsible for children must provide documented consent.

Moreover, improper access to personal information is an issue of privacy. This project must take key considerations into account when designing and developing SafeGaze, as unrestrictive access to sensitive data poses the threat of invasion of privacy. Security of the software application should be finetuned to prevent any unauthorised access to the facility or member's personal information, as a weak system leads to being compromised.

## 2.3 Plagiarism of Intellectual Property

There are many ways to acknowledge the ownership of intellectual property; however, there is the perception that software, ideas, or related concepts are owned by the organisation responsible for its development. Moreover, there are four different types of intellectual property rights related to software: patents, copyright, trade secrets, and trademarks. Regardless of type, new software must not directly take ownership of another organisation's work, as this is plagiarism and therefore considered a criminal offence. Proper market research was conducted prior to the development of SafeGaze to ensure that plagiarism would not pose to be a threat upon its release. There is an abundance of different livestream software services, however none of them relate to the features or aspects that SafeGaze directly tackles.

## 2.4 Software Licensing

With regards to licensing, it is important that all third party software, code and frameworks have been used in accordance with their software licensing to ensure the developer is within their permissions for any assets they might use. This project follows proper jurisdiction with what has been used in the development process of SafeGaze; these technologies are as follows:

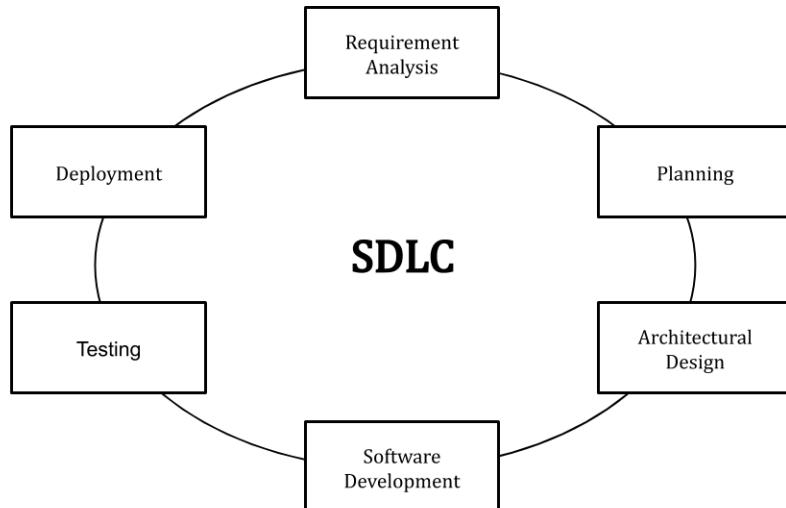
- Visual Studio Code
- Bootstrap
- PhpMyAdmin
- XAMPP

# 3. Project Management

## 3.1 Importance of Project Management

When involved in any major project or event, one would be required to scope out the set tasks, goals, and requirements, and then plan what form of approach is best suited to produce the highest quality of work possible. Failing to do so may have a detrimental effect on the outcome and potentially the reputation of any businesses or organisations involved. Safely planning the development, testing, and maintenance of a software project is often referred to as the 'Software Development Life Cycle'

(SDLC), and defines the practice of producing software at the most optimum efficiency, without negatively affecting the quality or cost of the project.



*Figure 1: Phases of SDLC (Altvater, A (2021))*

Moreover, SDLC is a framework that one would follow in producing software at a professional level, and thus shortens production time whilst simultaneously improving quality of the product. Figure 1 displays how a generic project would be separated. Optionally, a seventh phase could be implemented – ‘Software maintenance’; however this may not be necessary, depending on how the development team handles the “deployment” phase.

### 3.2 Method of Approach

Whilst there are several different models that one may follow when involved in a software development project, they all encompass the importance of proper planning and preparation. With regards to what model has been used for SafeGaze, this project was developed in an ‘Agile Environment’, therefore meaning the entirety of this project was approached in an incremental manner, alongside continuous collaboration, planning, and learning to achieve the end result. By operating on a ‘Scrum’ & ‘Kanban’ like approach, each iteration was referred to as ‘sprints’, with each sprint comprised of several tasks needed to complete the functional requirements and objectives set. As described by Rehkopf, M (2020), “*A sprint is a short, time-boxed period when a scrum team works to complete a set amount of work*”.

The work set for each sprint focused on a particular aspect of the project or software application, and thus prevented potential confusion or temporary setbacks between tasks that may be dependent upon one another. The time allocated per sprint was carefully considered before the ‘development phase’, therefore avoiding the possibility of having a sprint fall short of what might be expected upon reaching its due date. As each sprint focused on a core piece of functionality, it was crucial that the tasks were completed with high standard, and little possibility of returning to said requirement. The first sprint was considered as “Sprint 0” as it did not specifically involve any software development, but focused on the planning, preparation, and design of the project as a whole. The project plan is displayed in figure 2; to see a more detailed version, refer to appendix 14.2.5.

Deliverables		Estimate Hours of Work
Sprint 0: Planning & Preparation	Create project initiation (including market research, project vision, objectives, & risk plan)	5
	Create and populate backlog	3
	Set up repository & web hosting, organise important documents (report, notes, minutes, etc.)	1
	Research frameworks & tools to use	4
	Create requirements and use cases for both users	5
	Create project plan & Gantt chart	10
	Design system flowchart	5
	Create UI design	3
	Design database schema	4
Sprint 1: Facility focused	Set up app's initial pages (index, dashboard, login, etc.) & install Bootstrap to system	3
	Create SQL statements for database	3
	Allow facility to sign up	4
	Allow facility to log in	2
	List all active members	6
	List all member requests	1
Sprint 2: Member focused	List all idle members (declined + inactive)	2
	Allow member to sign up	3
	Allow member to log in	2
	Allow facility to accept/decline member signup requests	3
	Pending/inactive/declined members to have a "Verify member" page to eliminate unauthorised access	4
Sprint 3: Management focused	Display a member's access times to the member	4
	Allow facility to view all a member's details in a separate page ("manage member" page)	5
	Allow facility to change a member's account status	3
	Allow facility to view a member's access times	1
Sprint 4: Livestream focused	Allow facility to alter a member's access times	3
	Conduct further research into method of approach for implementing livestream	10
	Allow facility to create a new livestream	6
	Display all available livestreams to facility	5
	Display all available livestreams to member	3
	Allow member to ONLY access a livestream within their access times	4
	Display popup message for members who are not within their "access times"	1
Sprint 5: Message focused	Allow facility to edit an existing livestream	5
	Allow facility to delete an existing livestream	2
	Allow facility to issue a new message/notification to their members	8
Sprint 6: Testing	Allow facility to view all existing messages	5
	Allow member to view all existing messages	2
	Create video demonstration for FTA	10
	Inform FTA of development state + request feedback from video demo	5
Sprint 7: Finalisation	Log feedback received from FTA for future developments + improvements	2
	Create documentation for facilities on how to use SafeGaze	6
	Adjust system to resolve potential errors/bugs found in testing	10
	Refine UI styling	6
	Create showcase materials	14
	Project report finalisation	N/A

Figure 2: Project plan

By utilising the Agile methodology, it allowed the project to have clear objectives, effective time & risk management, and complete tasks in an orderly process to ensure compliance with the requirements set. Additionally, by separating the tasks into sprints that each focused on a particular theme, it improved time and risk management, as it ensured the project did not lack any time or resources required for a high-quality result. Numerous software applications were also included to benefit project management, as these provided tools that increased the efficiency of workflow (discussed further in section 3.5).

### **3.3 Supervisor Meetings & Bi-weekly Reports**

Alongside the proper planning and preparation that was enforced with SafeGaze, there were additional bi-weekly meetings and status reports that served as sessions to inform the project supervisor of current development status, and an opportunity to amend the current workflow upon any feedback given (see appendix 14.2.3 for bi-weekly reports). By making notes and tracking the dates of each event, it allowed for a detailed record what had been amended, a documented timeline of the development process, and key considerations for future development.

Alternatively, queries could be submitted in these meetings, resulting in vital changes that may improve overall product quality. As these meetings were grouped sessions, it also allowed for different views on a specific topic, as each individual would host their own opinion or perspective on such a matter, therefore enhancing the advice and guidance issued.

### **3.4 Time & Risk Management**

To ensure that SafeGaze was delivered within the given timeframe, it was crucial that proper time management was enforced, as this is vital to any large-scale project that might hold any time restrictions. The tasks within each sprint were made with the project's end date in mind, resulting in a balance of the available resources, especially time, and thus reduced the possibility of setting over-ambitious objectives. It was apparent that by utilising a project backlog, it resulted in better time management as it significantly reduced any time spent trying to identify which tasks had been completed, and which had not.

In addition, it was also important that risks were analysed to ensure they were properly managed whilst in the development process. Minimising the possibility of risks therefore significantly reduces the negative consequences that may occur if not addressed. To target these risks, a risk plan was created in the project's 'planning phase', thus raising awareness of what could go wrong, and what precautions need to be implemented to prevent or mitigate the risks from arising. The risk plan can be seen in section 6.6.

### **3.5 Justification of Technologies Used**

#### **3.5.1 Version Control: Git**

Due to the complexity of this project, it was important that version control was implemented, which heightened the ability to track any changes to the project's files, including code, documents, and graphics. This was achieved through Git because it is the most widely used version control system in today's standards for project management. This in turn allowed for the possibility of reverting to a previous version of the project's source code if any major problems or errors were to occur.

SafeGaze's source code and associated files were hosted to a repository on GitHub, which was also accessible by the Project Supervisor. As a result, they were able to monitor what specific changes had been made to the repository, thus ensuring that the correct quantity of work was completed in order to comply with the project's requirements, goals, and objectives, relative to its time constraints. Atlassian (2020) elaborates further on how an implemented version control system benefits teams when they are involved in the development of any large-scale project or high-performing software.

#### **3.5.2 Planned Backlog: Microsoft Office Planner**

By inputting tasks and other important aspects into a detailed backlog, it therefore allowed for everything to be conveniently organised in one designated area. This project's backlog utilises the contents of the Product backlog, Sprint backlog, and Project backlog, as it was used to plan the project's workflow, sprints' tasks, as well as general ideas/notes. Simply put, "*A project backlog is a*

*prioritized and structured list of deliverables that are a part of the scope of a project. It is often a complete list that breaks down work that needs to be completed.*”, Karlsson, J (2020).

This was achieved with Microsoft’s “Planner” application, and was recommended by Plymouth University due to its availability to the university’s students. This modern app behaves as a teamwork planning system, where it can be used to create Kanban boards in a checklist-like fashion. The tasks can be organised into separate groups, referred to as “Buckets”, thus allowing each item to be systematically arranged to the user’s liking. Moreover, numerous variables can be applied to each task, including: user allocation, labels, progress, priority, start date, due date, notes, sub-tasks, attachments, and comments. These additions proved to be particularly beneficial to SafeGaze due to the complexity of this project.

Project Tasks	Ideas & Notes	Use Cases	Functional Requirements
+ Add task	+ Add task	+ Add task	+ Add task
<input type="radio"/> Research Plymouth University hosting method of use ! <input type="radio"/>	<input type="radio"/> One user admin per sports club. They have control over name for member code for sports club entry (for participants)	<input type="radio"/> Parent/Participant I want to be able to view the sports club's contact information on the information page	<input type="radio"/> Parent/Participant View sports club's contact inform (information page)
<input type="radio"/> Design database		<input type="radio"/> Parent/Participant I want to be able to receive notifications from important messages issued by the sports club	<input type="radio"/> Parent/Participant Receive notifications from import messages issued by the sports cl
<input type="radio"/> Create index page for SafeGaze (admin focused)		<input type="radio"/> Parent/Participant I want to be able to view all messages issued by the sports club	<input type="radio"/> Parent/Participant View all messages issued by the sports club
Show completed 5		<input type="radio"/> Parent/Participant I want to be able to view the livestream	<input type="radio"/> Parent/Participant View the livestream
		<input type="radio"/> Parent/Participant	<input type="radio"/> Parent/Participant View user's livestream accessibility times

Figure 3: Project backlog at project’s initiation (October 2020)

Due to Planner being natively integrated into Office 365, it allowed for easy collaboration, specifically with regards to the Project Supervisor. Ultimately, Planner was an extremely useful tool as it created an intricate visualisation of the project’s scope, all whilst maintaining an organised structure.

### 3.5.3 Student-Supervisor Contact & Bi-weekly Meetings: Microsoft Teams

With the seamless integration of the different Microsoft applications, Microsoft “Teams” was the primary tool for student-supervisor communication, and bi-weekly meetings. It was vital that these meetings would be well-organised and easily accessible due to the ongoing global pandemic, which severely limited face-to-face contact. This in turn prevented any students taking part in meetings with their project supervisor without the use of video chat software.

By making use of Teams, it allowed for direct communication between the Project Supervisor and the students they were supervising; however, Teams also featured “Channels” which behaved as group chats. This proved to be especially beneficial for organising the bi-weekly meetings as a number of students were immediately notified from a single message from the Supervisor.

With regards to video calls, Teams was the preferred system over its competitors because the comments in these meetings were saved to the designated channel where the meeting was hosted. This therefore allowed students to return to the channel for important information they might have missed, even after the meeting had been concluded.

## 4. Client Side Testing with ETA

Due to the nature of SafeGaze, it was important that this application would be tested by an accredited facility which would benefit from its features, in an effort to prevent the development of components that might be a waste of time and resources. It also highlights what aspects must be prioritised, whether it relates to the application's functionality, user interface, or resolving logic or syntax errors. This project was executed in collaboration with Exeter Trampoline Academy (ETA) to gain an insight of an organisation's thoughts surrounding the software's features in question. ETA was thoroughly informed of SafeGaze's UI design, features, errors/bugs, future development, as well as general updates on its overall state upon reaching the end of each subsequent sprint.



Figure 4: ETA's setup

The iterative nature of this particular project plan allowed for constant refinement over the entirety of SafeGaze's development lifespan. By communicating with detailed email messages and video call meetings, it proved useful in the current COVID-19 rules and regulations regarding face-to-face contact. ETA's CEO was able to fully comprehend the project's state and therefore offer invaluable advice and guidance regarding specific matters.

Upon the reopening of gyms and leisure centres, a beta version of SafeGaze was implemented into ETA, which allowed for a greater understanding of what needs to be improved upon, as well as the future developments that will further enhance its capabilities. These aspects are elaborated further in section 9.

## 5. Requirements

A project's requirements are the product's features, functions, conditions, or tasks that must be completed to ensure the success or completion of the product; these are designed in relation to the software/project's stakeholders. By setting clear goals and requirements prior to the developmental phase of the project, it therefore prevents any unnecessary confusion for its developers. This in turn negates the risk of developing a piece of software that is unsuccessful, or does not contain the primary functionality that was initially stated in the project plan.

### 5.1 Functional Requirements

#### 5.1.1 Facility

Requirement	Priority (low, medium, high)
Create account	High
Login to my account	High
Logout of my account	High
Change the password to my account	Low
View sports facility's details (including access code - created in signup)	Medium
Change my sports facility's details	Low
Create a new livestream for my members	High
Edit an existing livestream's details	Medium
Remove an existing livestream	Medium
View all members linked to facility (active, inactive, and declined)	High
View member signup requests	High
Approve member signup requests	High
Deny member signup requests	High
Edit a member's account status	High
View a member's livestream accessibility times	High
Edit a member's livestream accessibility times	High
View all messages issued to members	Medium
Create new messages to members	Medium
Edit existing messages' details	Low
Remove messages issued to members	Low

#### 5.1.2 Member

Requirement	Priority (low, medium, high)
Create new account	High
Login to user account	High
Logout of user account	High
Change password to account	Low
View my livestream accessibility times	High
Watch the available livestreams	High
View all messages issued by the sports facility	Medium
Receive notifications when facility posts "urgent" messages	Medium
Receive notifications when member's account status is changed	Medium

Receive notifications when member's livestream accessibility times are changed	Medium
View the sports club's details and contact information	Medium

## 5.2 Non-functional Requirements

Requirement	Priority (low, medium, high)
Software source code should be regularly backed up	High
All users' passwords must be encrypted prior to storing to the database	High
System must be secure and not vulnerable to potential hacks	High
Conduct proper testing for each feature/functionality of the application, for both use cases member and facility (point of view)	Medium
Use of good programming practice: layout & structure, variable/object names, commenting, etc.	Medium
Code must allow for possibility of future modification/improvement	Medium
Documentation should be produced on how to use the system	Low

## 6. System Architecture & Design

### 6.1 Use-Case Diagram



Figure 5: Use-case diagram

## 6.2 Flowchart

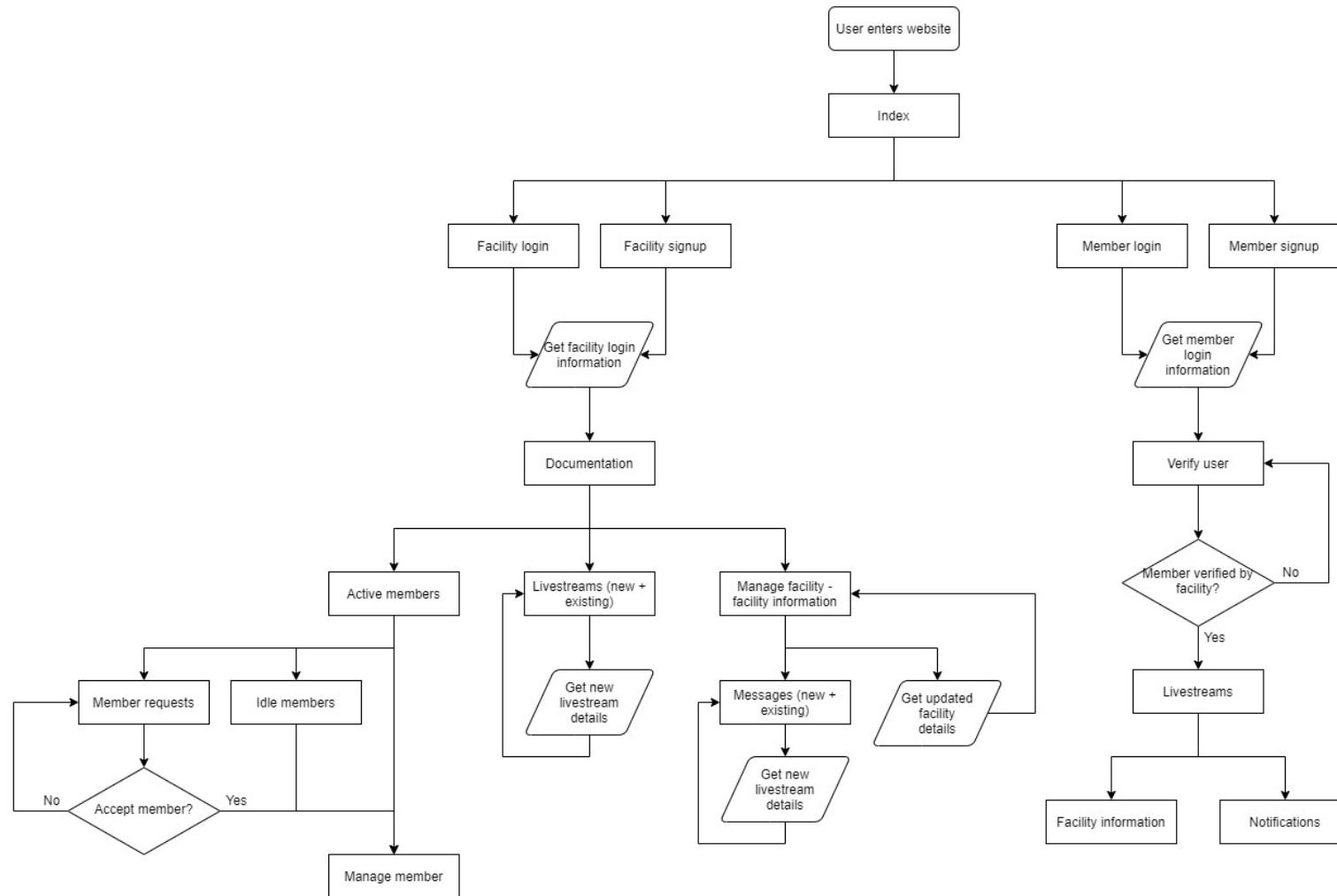


Figure 6: Flowchart diagram

### 6.3 Architecture

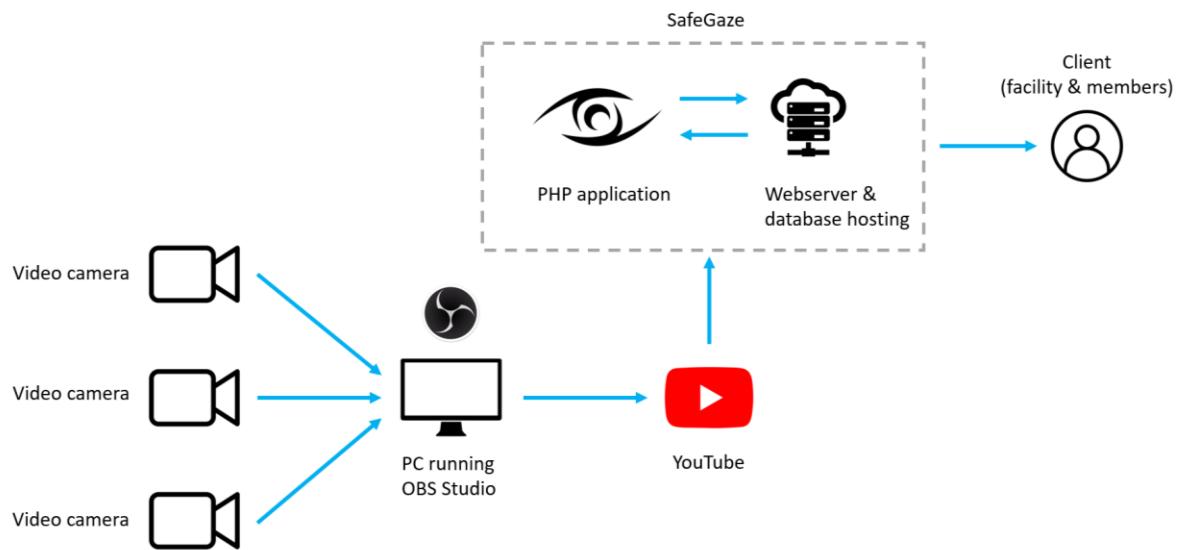


Figure 7: System architecture

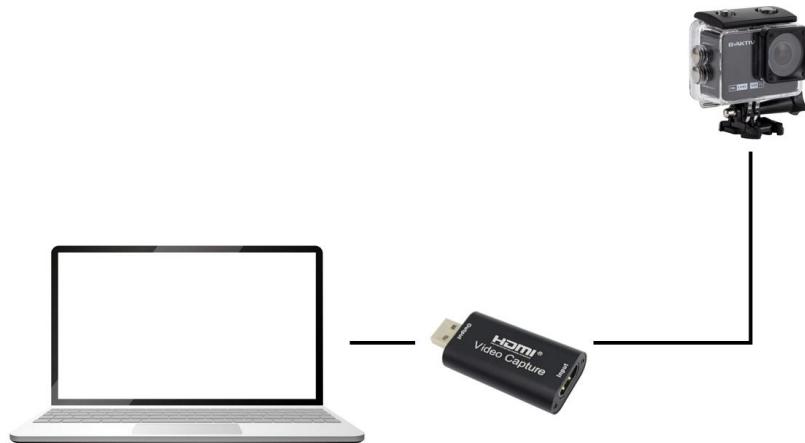


Figure 8: Hardware configuration

## 6.4 UI Design

### 6.4.1 Facility

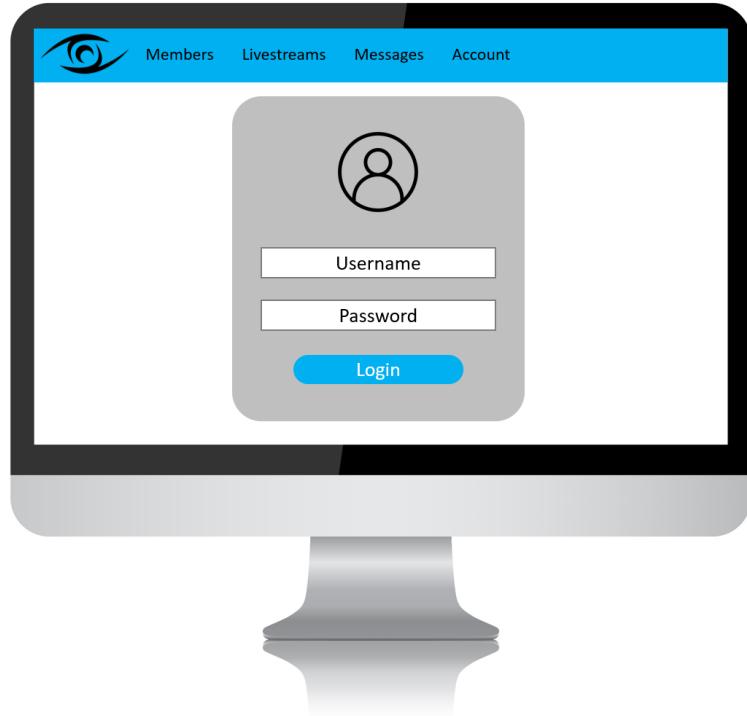


Figure 9: UI design “Login” (facility)

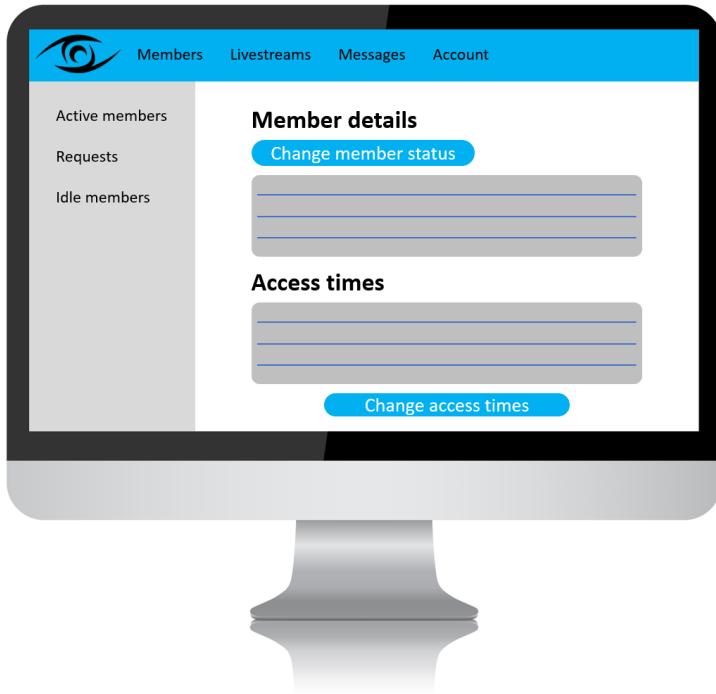


Figure 10: UI design “Manage member” (facility)

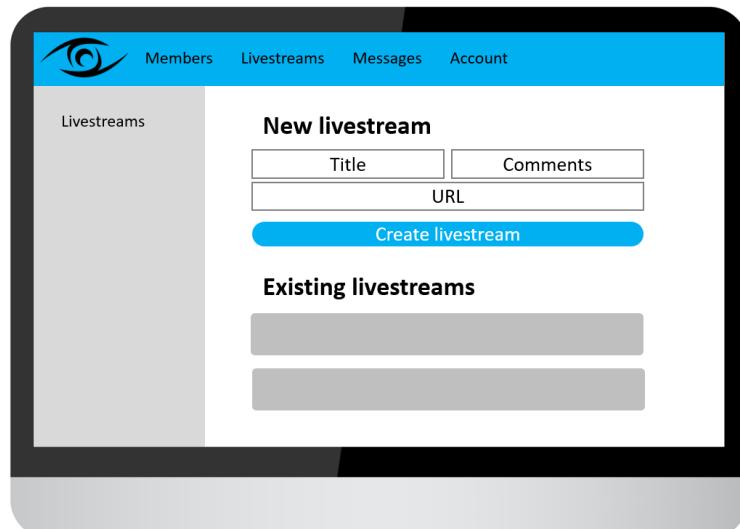


Figure 11: UI design “Manage livestreams” (facility)

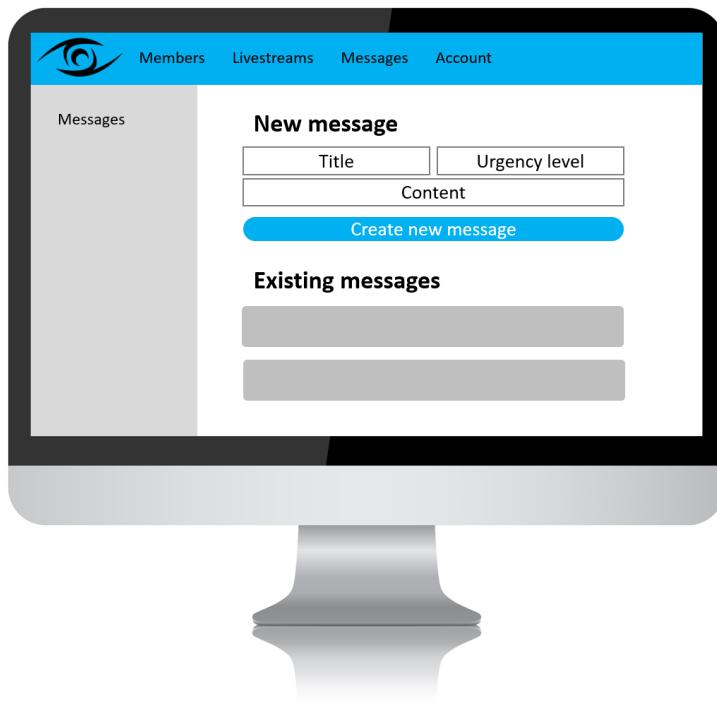


Figure 12: UI design “Manage messages” (facility)

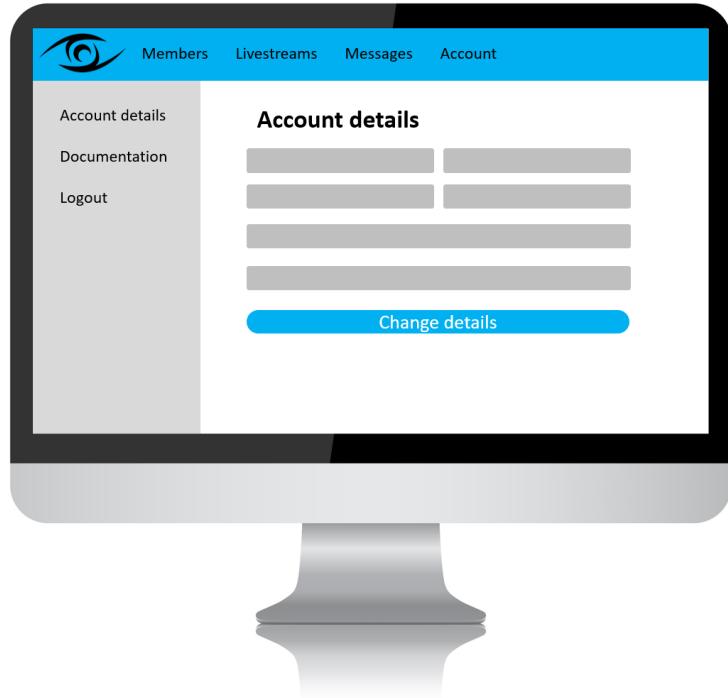


Figure 13: UI design “Manage account” (facility

#### 6.4.2 Member



Figure 14: UI design “Login” (member)

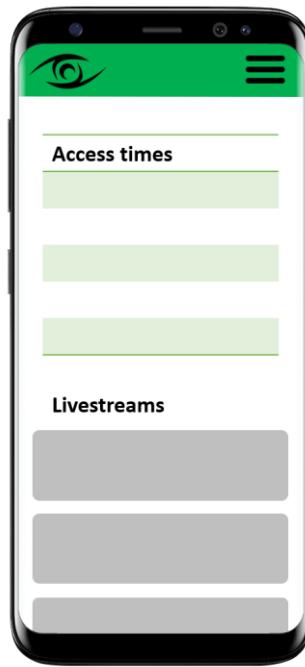


Figure 15: UI design “View livestreams” (member)



Figure 16: UI design “View messages” (member)



Figure 17: UI design “View facility information” (member)

## 6.5 Database Schema

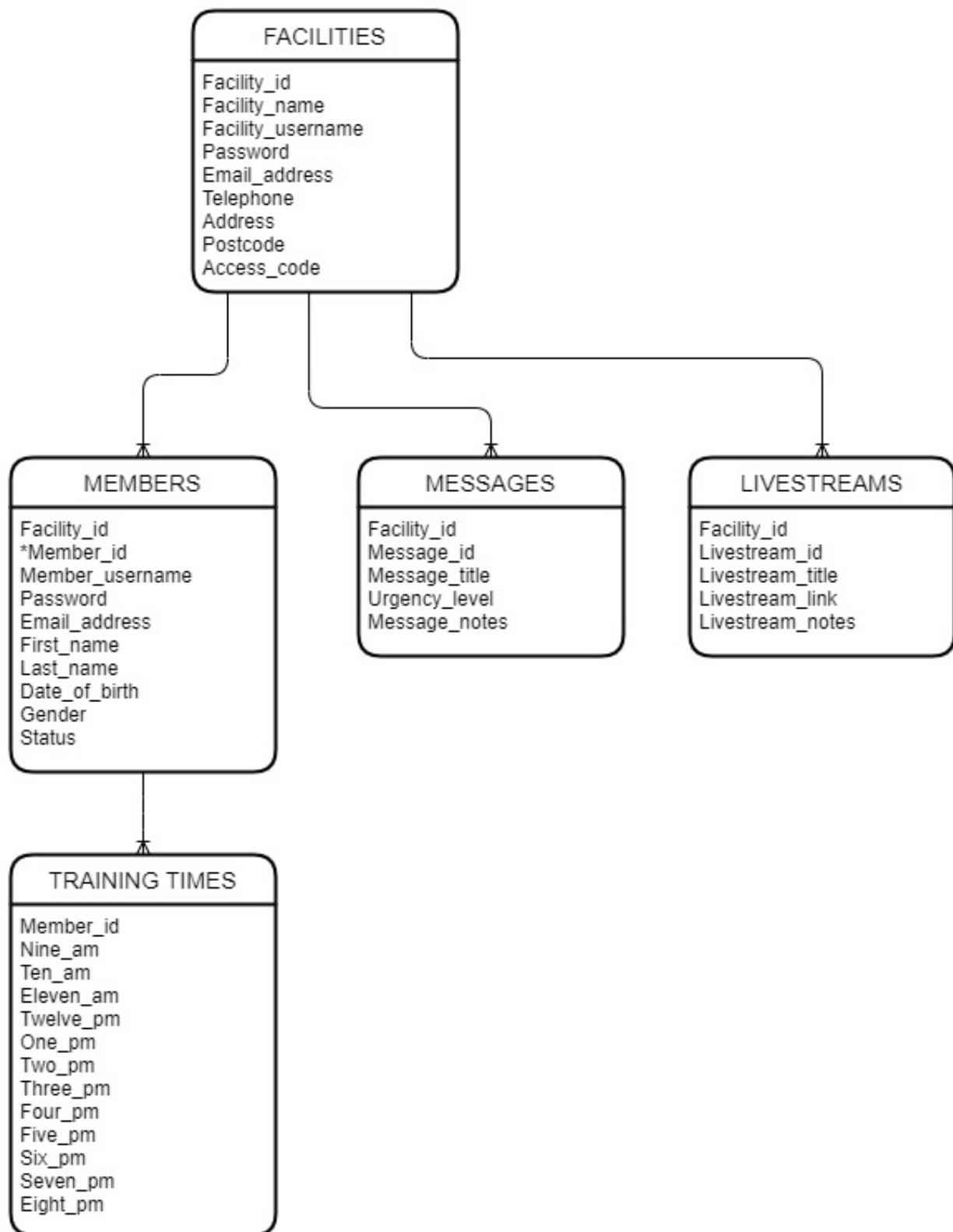


Figure 18: Database schema diagram

## 6.6 Risk Plan

To see the risk plan in its raw format, refer to appendix 14.2.2.

Risk	Likelihood	Impact	Solution
Poor estimation and scheduling per functional requirement	Low	Medium	When creating the Kanban board, provide adequate times per task by accounting for anomalies or time consuming incidences. This therefore means each task will be achievable within the given time frame.
Poor time management due to other commitments outside university	Medium	Medium	Organise tasks into designated working days and weeks, making sure to account for potential changes in the project scope and possibility of outside university events which may affect time management.
Lack of frequent backups or potential loss of data	Low	High	Ensure there are regular git requests to the GitHub repository after making any significant changes to the project. Potentially have a second backup of the project in case of any Git failures.
Loss of valuable time due to obsessive use of “gold plating”	Low	Medium	Prevent use of adding unnecessary features until primary objectives have been reached. Prioritise the functional requirements set, as opposed to adding visually pleasing features that do not contribute to the end goal.
Technical difficulties with related software or hardware resources used in presentations and showcases	Medium	Medium	Thoroughly test available resources in different environments to guarantee a successful delivery in showcases made on personal or public hardware.
Lack of knowledge or experience within specific programming languages, frameworks, or software	Medium	Low	Properly research available resources required for production, as well as potential software resources which may prove to be beneficial.
Poor code quality	Low	Medium	Plan ahead of how the code will be structured to ensure it is properly organised in the different directories or files. When a functional requirement has been met, perform specific test cases to validate that it is working effectively.
Livestream(s) open to all users, therefore creating the potential risk of stalking or harassment from strangers	Low	High	Users require verification to sign up, alongside with proof of being involved in the sports club. Additionally, users will only be able to partake in livestreams at a given time when their child is participating in such activities. Due to how sports clubs are often run, the participant would be given a specific timeslot in a weekly timetable, and thus the parent would only have access to the livestream within this allocated time.
Breach of users' personal data due to software hack or weak security	Medium	High	Ensure all users' personal data is fully encrypted and highly secure within the app and database. Extra steps need to be taken into consideration with the security the mobile app and website. This includes, but not limited to: SQL injection, cross site scripting (XSS), sensitive data exposure, and global variables.
Software efficiency degradation	Low	High	Structure code in the most efficient way possible; prevent use of unnecessary tasks which are CPU intensive.

Figure 19: Risk plan

## **7. Implementation of Technologies & Resources**

This section outlines what specific software, frameworks, and resources were used in the development process of SafeGaze, alongside with detailed justification of why it was favoured over other methods. Prior to development, a comprehensive search was conducted on the different resources that would prove beneficial in developing SafeGaze, therefore improving workflow efficiency.

### **7.1 HTML, CSS, & JavaScript**

HTML, CSS, and JavaScript are by far the most commonly used languages with regards to website development, due to their interconnectivity with one another. Furthermore, as they are extremely popular, there are many tools and resources that take advantage of their capabilities, one being “Bootstrap”.

### **7.2 Bootstrap**

By utilising the Bootstrap framework, it allowed SafeGaze to have a professional and responsive user interface. Moreover, as it is flexible and easy to use, it improved workflow efficiency during the development phase, which resulted to accomplishing the deliverable sooner. Many of Bootstrap’s components are reusable, therefore each page within SafeGaze might be similar to one another, providing the user with some form of ‘familiarity’.

### **7.3 PHP**

PHP (Hypertext Pre-processor) is a server side scripting language that equipped SafeGaze with the ability to interact with database languages, specifically MySQL in this case. This allowed the front-end system to interact with the database nested in the web hosted network. PHP proved to be an efficient and flexible language, alongside providing numerous other advantages to the features of this project.

### **7.4 MySQL & PhpMyAdmin**

MySQL is the most popular database system used along PHP, and is very fast, reliable, and easy to use. Due to previous experience with designing and implementing MySQL databases, it was apparent that this language would be the first choice amongst its competitors. In addition, PhpMyAdmin was chosen as a tool to enhance the efficiency of various database processes or procedures.

## 8. Project Phases

This section fully analyses and explains the different phases of the project, and breaks down each sprint that occurred within the development process. Furthermore, each sub-section provides an outline of the sprint's objectives and tasks in question, as well as key thoughts and considerations that took place within the given time.

### 8.1 Development Overview

Deliverables	Estimate Hours of Work	Start Date (W/C)	End Date (end of W/C)	Milestones (W/C)												
				Oct-20	Nov-20	Dec-20	Jan-21	Feb-21	Mar-21	Apr-21	May-21					
Sprint 0: Planning & Preparation	Create project initiation (including market research, project vision, objectives, & risk plan)	5	05/10/2020	11/10/2020												
	Create and populate backlog	3	05/10/2020	11/10/2020												
	Set up repository & web hosting, organise important documents (report, notes, minutes, etc.)	1	05/10/2020	11/10/2020												
	Research frameworks & tools to use	4	05/10/2020	11/10/2020												
	Create requirements & use cases for both users	9	05/10/2020	18/10/2020												
	Create project plan & Gantt chart	10	12/10/2020	18/10/2020												
	Design system flowchart	9	19/10/2020	26/10/2020												
Sprint 1: Facility focused	Create UI design	3	19/10/2020	26/10/2020												
	Design database schema	4	19/10/2020	26/10/2020												
	Set up app's initial pages (index, dashboard, login, etc.) & install Bootstrap to system	3	26/10/2020	01/11/2020												
	Create SQL statements for database	3	26/10/2020	01/11/2020												
	Allow facility to sign up	4	02/11/2020	08/11/2020												
	Allow facility to log in	2	02/11/2020	08/11/2020												
	List all active members	6	09/11/2020	15/11/2020												
Sprint 2: Member focused	List all member requests	1	09/11/2020	15/11/2020												
	List all idle members (declined + inactive)	2	09/11/2020	15/11/2020												
	Allow member to sign up	3	16/11/2020	22/11/2020												
	Allow member to log in	2	16/11/2020	22/11/2020												
	Allow facility to accept/decline member signup requests	3	16/11/2020	22/11/2020												
	Pending/inactive/declined members to have a "Verify member" page to eliminate unauthorised access	4	23/11/2020	06/12/2020												
	Display a member's access times to the member	4	23/11/2020	06/12/2020												
Sprint 3: Management focused	Allow facility to view all a member's details in a separate page ("manage member" page)	5	07/12/2020	13/12/2020												
	Allow facility to change a member's account status	3	14/12/2020	20/12/2020												
	Allow facility to view a member's access times	3	14/12/2020	20/12/2020												
	Allow facility to alter a member's access times	3	21/12/2020	27/12/2020												
	Conduct further research into method of approach for implementing livestream	10	28/12/2020	03/01/2021												
	Allow facility to create a new livestream	6	04/01/2021	10/01/2021												
	Display all available livestreams to facility	9	04/01/2021	10/01/2021												
Sprint 4: Livestream focused	Display all available livestreams to member	3	04/01/2021	10/01/2021												
	Allow member to ONLY access a livestream within their access times	4	11/01/2021	17/01/2021												
	Display popup message for members who are not within their "access times"	1	11/01/2021	17/01/2021												
	Allow facility to edit an existing livetream	9	18/01/2021	24/01/2021												
	Allow facility to delete an existing livetream	2	18/01/2021	24/01/2021												
	Allow facility to issue a new message/notification to their members	8	25/01/2021	31/01/2021												
	Allow facility to view all existing messages	5	01/02/2021	07/02/2021												
Sprint 5: Message focused	Allow member to view all existing messages	3	01/02/2021	07/02/2021												
	Create video demonstration for ETA	16	08/02/2021	14/02/2021												
	Inform ETA of development state + request feedback from video demo	5	08/02/2021	14/02/2021												
	Log feedback received from ETA for future developments + improvements	2	15/02/2021	21/02/2021												
	Create documentation for facilities on how to use SafeGaze	6	22/02/2021	28/02/2021												
	Adjust system to resolve potential errors/bugs found in testing	10	01/03/2021	07/03/2021												
	Refine UI styling	6	08/03/2021	14/03/2021												
Sprint 6: Testing	Create showcase materials	14	08/03/2021	04/04/2021												
	Project report finalisation	N/A	05/04/2021	16/01/1900												

Figure 20: Project Gantt chart

As previously stated in section 3.2, each sprint focused on a specific aspect of SafeGaze, which can be seen in the project's Gantt chart, displayed in figure 20. The sprints were designed to occur every three weeks, however some sprints needed to be adjusted to be made shorter or longer due to their difficulty, work hours, or nature of the tasks within said sprint. The dates assigned to each task act as a guideline but do not necessarily need to be strictly adhered to, as other university commitments may potentially have an impact on completing certain tasks within the allocated time. That being said, the end dates for each sprint were flexible, therefore allowing for continuation on the sprint's tasks if more time was required, or starting the next sprint if all tasks were already completed prior to the deadline. To see the associated backlog per sprint, refer to appendix 14.2.1.

### 8.2 Sprint 0: Planning & Preparation

The initial sprint heavily revolved around the planning, organisation, and design of the entire project. First, by creating the "Project Initiation" document, it provided a clear scope of what form of approach this project would take. This primarily consisted of the project's vision, where the objectives and goals were outlined in an effort to get an insight of what was to be achieved by the end of the academic

year. Upon the completion of this document, numerous resources and assets were required to be setup prior to the start on the project's planning and design, which included: the project backlog, git repository, Plymouth University web server hosting, and important documents that would aid the project.

Of course, extensive market research was conducted when creating the project initiation, and primarily focused on what approach would be most beneficial in relation with the objectives stated. Nevertheless, preliminary research provided insight into the different software capabilities and scopes that were available. Once this was achieved, numerous designs and plans related to the application were assembled, which therefore outlined the styles, techniques, and procedures that would be taken when constructing the different components of the application. Most of these designs can be seen within section 6.

Moreover, the project's backlog was populated with important planning documents (e.g. use cases) for the purpose of tracking and constant assessment of set requirements. In addition, communication with the project supervisor was particularly frequent within this stage of the project, considering they were most equipped to understand what changes would be beneficial (or mandatory) to the end deliverable.

## Deliverables

- Create project initiation (including market research, project vision, objectives, & risk plan)
- Create and populate backlog
- Set up repository & web hosting, organise important documents (report, notes, minutes, etc.)
- Research frameworks & tools to use
- Create requirements and use cases for both users
- Create project plan & Gantt chart
- Design system flowchart
- Create UI design
- Design database schema

No significant challenges or obstacles were faced over the course of this phase as it revolved around the planning of the project, and subsequently was very straight forward with limited difficulties.

### 8.3 Sprint 1: Facility Focused

This sprint was the commencement of the development of SafeGaze. As the application heavily relied upon PHP and its database for its features, it was crucial that these components were setup and configured once the initial pages had been created. Fortunately, by following the existing user interface design already created in Sprint 0, this meant less time needed to be allocated to the styling of the different webpages. With the use of the Bootstrap framework, basic UI technologies were effortlessly implemented, such as the navigation bar. By following the Database Schema from Sprint 0, the SQL table creation statements were initiated, therefore allowing for a working database to be implemented for the system.

Although, the aim of this sprint was to develop the basic interface for the application, it also focused on providing a facility with basic functionalities. This involved features related to the facility's account, such as being able to sign up to SafeGaze, as well additional features relating to viewing their members according to the member's account status. Upon completion, the PHP backend of each feature was finetuned to ensure the system was highly secure and functioned perfectly.

## Deliverables

- Set up app's initial pages (index, dashboard, login, etc.) & install Bootstrap to system
- Create SQL statements for database
- Allow facility to sign up
- Allow facility to log in
- List all active members
- List all member requests
- List all idle members (declined + inactive)

With the use of Bootstrap, it allowed for easy implementation of a modern and responsive layout. Nevertheless, there were still some annoyances that presented themselves within the initial setup, and resolving such matters became more time intensive than originally anticipated.

### 8.4 Sprint 2: Member Focused

This sprint had very similar deliverables to the previous one, although specifically relating to the facility's participants that would have a SafeGaze account. With regards to the term 'member' – member applies to individuals that are associated with the facility **AND** have/or requested an account for SafeGaze under the facility's name. On the other hand, 'participant' applies to individuals who engage in the facility's activities; moreover, parents or guardians of participants may also be considered as a member to a facility.

Security within this specific area remained pertinent, as specified in the deliverables. For a member to be granted authorised access to their account, their facility would first be required to accept their submitted request. Furthermore, the backend for the "verify member" page was refined to ensure that a member would not be able to access their account illegitimately.

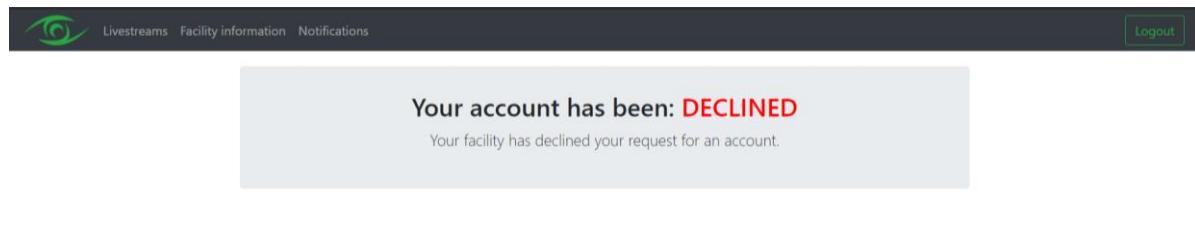


Figure 21: "Verify member" page (showing declined member request)

## Deliverables

- Allow member to sign up
- Allow member to log in
- Allow facility to accept/decline member signup requests
- Pending/inactive/declined members to have a "Verify member" page to eliminate unauthorised access
- Display a member's access times to the member

This sprint presented no major difficulties and surprisingly was very satisfying once the security-related components had been finalised.

## 8.5 Sprint 3: Management Focused

Due to the previous sprints, facilities and members had access to their account, although further management functionalities still needed to be considered and implemented to ensure a facility would be able to govern which of its members have access to the livestream(s) throughout specified times of the day. This resulted in Sprint 3 being centred around the development of such management features that a facility would require.

To implement these functions, the members list would need to be adjusted to make use of an additional column, where a “manage” button would be located. Upon its submission, the facility would be transferred to a dedicated management page, taking the selected member as its argument. This page would therefore allow a facility to perform certain tasks related to viewing a member’s personal details, account information, and adjusting their account’s accessibility within SafeGaze. The buttons within the “Requests” table were adjusted to make use of the new management page; if they facility clicks “Accept”, they are taken to the management page with the accepted member as its argument. This allows for the facility to immediately input the member’s access times if desired.

### Active members

Member ID	First name	Last name	Username	Email	Gender	DOB	
3	Third	User	thirdUser	third@test.com	NULL	NULL	<button>Manage</button>
9	Seventh	User	seventhUser	seventhuser@test.com	male	2020-12-23	<button>Manage</button>

Figure 22: Member list with implemented “manage” button

The screenshot shows a web application interface for managing members. At the top, there is a navigation bar with links: 'Getting started', 'Members', 'Livestreams', 'Manage facility', and 'Logout'. On the left, a sidebar titled 'MEMBERS' contains links for 'Active members', 'Requests', and 'Idle members'. The main content area is titled 'Member : Seventh User'. It displays the following member details in a table format:

Member ID:	9
Username:	seventhUser
Email:	seventhuser@test.com
First name:	Seventh
Last name:	User
Date of birth:	2020-12-23
Gender:	male
Status:	ACTIVE

Below the details, there is a dropdown menu labeled 'Select member status' and a blue button labeled 'Change status'.

At the bottom of the main content area, there is a section titled 'Training times'.

Figure 23: Dedicated “Manage member” page

### Deliverables

- Allow facility to view all a member's details in a separate page ("manage member" page)
- Allow facility to change a member's account status
- Allow facility to view a member's access times
- Allow facility to alter a member's access times

One particular challenge presented itself within the development of this sprint; this being the ability to safely and securely transfer a member's account details to the dedicated management page. A JavaScript function was being executed to take the facility to a new page, however the information was in the form of PHP variables, which was not accessible natively through JavaScript. It was found that to resolve such an issue, the best solution was to retrieve the PHP variable's value from the database and store it in the list as an input text box, with the type "hidden" to ensure the user will not be able to alter it. To demonstrate this, figure 24 displays the HTML code of these adjustments in order to pass the member's ID as an argument when accepting or declining a member request.

```

<tr>
    <td> . $memberId . '</td>
    <td> . $memberFirstName . '</td>
    <td> . $MemberLastName . '</td>
    <td> . $memberUsername . '</td>
    <td> . $memberEmail . '</td>
    <td> . $memberGender . '</td>
    <td> . $memberDOB . '</td>
    <td>
        <form action="php/change-member-status.inc.php" method="post">
            <input type="hidden" name="memberid" value=". $memberId .>
            <button class="btn btn-sm btn-actions btn-success" type="submit" name="accept-member-submit">Accept</button>
            <button class="btn btn-sm btn-actions btn-danger" type="submit" name="decline-member-submit">Decline</button>
        </form>
    </td>
</tr>

```

*Figure 24: PHP code for accepting/declining a specific member request*

## 8.6 Sprint 4: Livestream Focused

Sprint 4 was directly related to the livestreaming capabilities that SafeGaze would offer, and thus the tasks allocated within this sprint were vital to the success of the project. The livestreams would be built upon what had already been created within the previous sprints, taking advantage of the capabilities/features each SafeGaze account offers. It was done in this order to ensure a time-efficient approach in completing all allocated tasks whilst reducing the possibility of set-backs due to task-related dependencies.

Upon the commencement of this sprint, research was conducted on the process of connecting the camera to the facility's computer, however the method which was expected to work did not seem plausible with this projects requirements. To specify: the original plan was to replicate the system that ETA used to playback footage of its athletes performing highly sophisticated skills. This configuration made use of the software "VLC Player", where the DSLR camera was connected directly to the computer through "Real Time Streaming Protocol" (RTSP). As a result, both the coach and athlete(s) are able to watch their performance and therefore identify potential changes that can be done to significantly improve technique. This method required a wireless connection between the computer and camera, resulting in the inability for the computer to connect to the internet wirelessly, which was required for SafeGaze.

Fortunately, an alternative method was found; by making use of additional hardware, the live video footage was successfully encoded to the computer, which could then be used by SafeGaze. A video capture card connects the DSLR camera to the computer's HDMI port through a wired connection; "OBS Studio" is then used as a tool to stream the footage directly to a stream hosting platform. As of the moment, SafeGaze makes use of YouTube livestreams "as a proof of concept", as private stream hosting platforms require a paid subscription to host a livestream. It is important that the YouTube livestream is marked as "Unlisted", which means it is not searchable within YouTube, yet still can be embedded into websites. To apply the livestream to SafeGaze, the video's hyperlink must be copied

and pasted into SafeGaze's "Livestream" section. Upon submitting this form, the hyperlink is altered into embedded HTML code, which allows the livestream to be directly viewed through by the facility and its members.

Figure 25: Facility's livestreams page

YouTube has proved to be beneficial in showing SafeGaze's capabilities, however numerous flaws hinders its potential. One primary issue relates to security, where SafeGaze users are able to access the livestream directly from YouTube by clicking a specific button in the video displayed from within SafeGaze. As a result, the user may disregard SafeGaze entirely, thus eliminating the secure system surrounding the user's specified access times.

Additionally, a facility does not have the ability to keep the YouTube livestream running without continuously providing it some form of video footage. This therefore means the facility would be required to conclude the livestream at the end of each day, as YouTube would automatically stop the livestream if the cameras were disconnected or powered off. Subsequently, the following day would require the facility to create a new livestream and paste the new livestream URL in SafeGaze. Although this is not a significant concern, it may become quite tedious in certain situations. Furthermore, YouTube's default settings grants the viewers the ability to replay a livestream after it has ended, therefore raising an issue with security. The facility can negate this by manually enabling an auto-delete function within YouTube's "creator" settings, which would result in the automatic deletion of a livestream once it has finished.

Upon the future development of SafeGaze, livestreaming will be enhanced by migrating the system over to a secure private streaming platform, therefore significantly improving SafeGaze's security within this sector.

## Deliverables

- Conduct further research into method of approach for implementing livestream
- Allow facility to create a new livestream
- Display all available livestreams to facility
- Display all available livestreams to member
- Allow member to ONLY access a livestream within their access times
- Display popup message for members who are not within their "access times"

- Allow facility to edit an existing livestream
- Allow facility to delete an existing livestream

Due to lack of experience with video streaming through software defined networking (SDN), implementing livestream functionality posed a significant challenge and temporarily halted the development process of SafeGaze. As a result, time-consuming research was required to gain an understanding of what was the best method of approach, and how it would be implemented in this project. This meant more time was spent on Sprint 4 than originally anticipated.

## 8.7 Sprint 5: Message Focused

Upon the completion of Sprint 4, it was clear that the remaining tasks were certainly straightforward and would pose no significant challenges. Sprint 5 focused on allowing the facility to issue notifications/messages to all their members with an active SafeGaze account. Following similar behaviours and UI styling as the previous pages, the facility is provided a form where they can input the Message's "Title", "Contents", and "Priority". The message's priority field was implemented as an asset for issuing important news updates to members, if desired.

The screenshot shows a web application interface for managing messages. At the top, there is a navigation bar with links for 'Getting started', 'Members', 'Livestreams', 'Manage facility', and 'Logout'. On the left, a sidebar titled 'FACILITY' contains 'Facility Information' and 'Messages'. The main content area has a title 'Create New Message (issued to all members)'. It includes fields for 'Title' (set to 'New message') and 'Priority' (set to 'Low'). Below this is a 'Contents' section containing the text 'This is a test!'. A large blue button at the bottom right of this section is labeled 'Notify Members'. Below this section, there is a heading 'Existing Messages' followed by a list of three message categories: 'Urgent message (3)', 'Medium urgency message (2)', and 'Low urgency message (1)'. Each category entry includes the urgency level, a brief description, and a red 'Delete' button on the right.

*Figure 26: Facility's messages page*

Furthermore, additional deliverables were also included within this sprint that were not originally specified in the project plan. These focused on the facility, where they are able to view and update their account information. This was implemented specifically for the sole purpose of allowing a facility to view their access code for any participants who wish to request a SafeGaze member account. It is also insurance for if a facility were to forget their access code.

The screenshot shows a user interface for managing facility information. On the left, there's a sidebar with 'FACILITY' and 'Facility Information' selected. Below that is 'Messages'. The main area is titled 'Facility Information' with a note: 'Please note: input boxes shaded in blue cannot be altered.' It contains several input fields: 'Facility Name' (shaded blue, value 'Final Test Changed'), 'Username' (value 'finaltest'), 'Telephone Number' (value '01752 123456'), 'Email Address' (value 'finaltest@test.com'), 'Street Address' (value 'Testing Street, Plymouth'), 'Postcode' (value 'AB1 2CD'), and 'Facility Access Code' (value 'finaltestaccess'). At the bottom is a large blue 'Update Information' button.

*Figure 27: Facility's information page*

## Deliverables

- Allow facility to issue a new message/notification to their members
- Allow facility to view all existing messages
- Allow member to view all existing messages

Due to the nature of the tasks in this sprint, they were easy to completed; however, the delay which was caused by Sprint 4 created a somewhat troublesome situation, as the deadline was closer than originally planned. It was apparent that the remaining tasks needed to be completed within their allocated time to ensure thorough testing would be done.

## 8.8 Sprint 6: Testing

It was originally planned to have SafeGaze set up in ETA in February 2021 to demonstrate its potential, as well as to highlight important aspects that would improve its functionality in one way or another. However, this application depended on sports facilities being available to the public, and it was not foreseen that they would be closed for three months from the start of 2021 due to COVID-19. This severely halted the client side testing of SafeGaze as it was only plausible to make use of the application after the ease of lockdown restrictions (in April 2021).

An online meeting was set up with ETA's CEO, Polly Buzzo-Johns. This marked an important milestone in the project as it was the first time the client saw a working solution. Once this was done, an in-depth video demonstration was recorded and sent to Polly, allowing her to analyse the application, and find what improvements can be made. Her original feedback can be seen in the form of an email, located in appendix 14.4.1.

Once lockdown restrictions had been lifted, a face-to-face meeting was conducted with Polly, where SafeGaze's setup procedures were analysed. As a result, important feedback was noted on the final enhancements for SafeGaze. This is elaborated in section 9.

## Deliverables

- Create video demonstration for ETA
- Inform ETA of development state + request feedback from video demo
- Log feedback received from ETA for future developments + improvements
- Create documentation for facilities on how to use SafeGaze

Of course, the lockdown created a considerable challenge for this project; nevertheless, certain actions were taken to overcome it. Some testing was conducted, even if it might not have been as thorough as the scope expected. As a result, less time was spent on testing and more on finalising the project, which still proved to be beneficial.

## 8.9 Sprint 6: Finalisation

Sprint 6 marked the final stage of this project, which primarily consisted of making minor adjustments to the application's functionality, as well as refining the CSS styling for better presentation. Once these tasks had been completed, the project's documentation and showcase materials took priority as these were important assets for demonstration SafeGaze's capabilities. Originally, the documentation was set to be done in Sprint 5, however it posed to be far more beneficial to complete it at the end of SafeGaze's development. In addition, creating the documentation was very similar to the tasks set in Sprint 6, as opposed to those in Sprint 5.

With regards to the showcase materials, all contributions are displayed on the Plymouth University "Secam Showcase" website. The entirety of the showcase would consist of four separate segments: the project's vision; a project poster; a thumbnail for said poster; and a hyperlink to a 5 minute video demonstration of the software, posted directly to YouTube. Firstly, the project's poster was to be treated as a highly informative piece of documentation, where it outlines what the software does, its key features, and what technologies were used in its development process. Alongside this, the poster would also include any other important information surrounding the project (E.g. the members responsible for its development). SafeGaze's project poster was created with the use of the software "Gimp".



Figure 28: SafeGaze's project poster

The 5 minute video demonstration used the project's documentation as its script, as it behaved primarily as a guideline for new SafeGaze members. This followed an intricate process, where the software "OBS Studio" was used to capture the use of SafeGaze. To accompany this, "Audacity" was used to record the voiceover for the specific functions being executed. Subsequently, all footage was manipulated and arranged with the video editing software "Shotcut" for a more professional

presentation. In addition, royalty free footage was used to enhance the video's outcome, to which the creators of such assets have been clearly stated in the video's description.

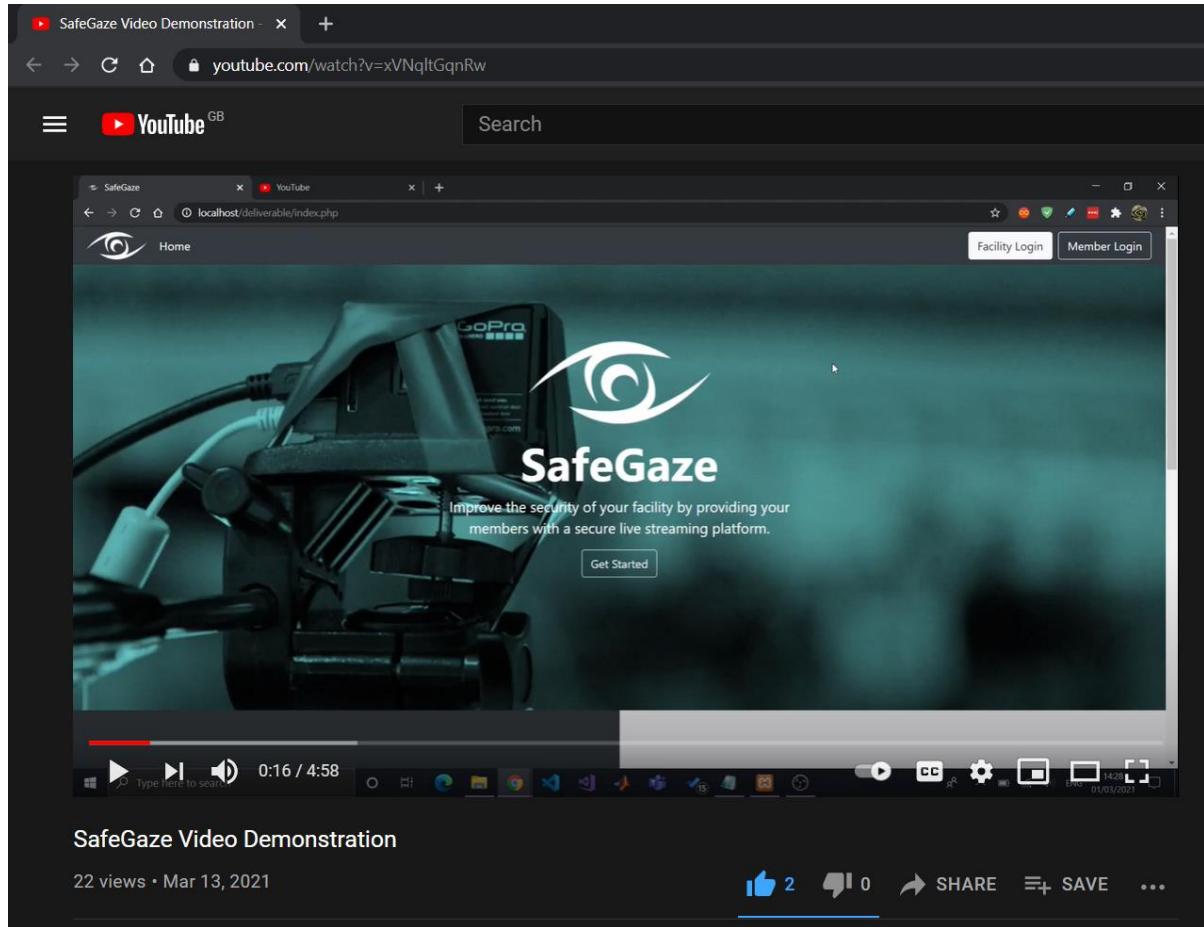


Figure 29: Screenshot of SafeGaze's video demonstration posted to YouTube (<https://youtu.be/xVNqltGqnRw>)

## Deliverables

- Adjust system to resolve potential errors/bugs found in testing
- Refine UI styling
- Create showcase materials
- Project report finalisation

Despite the fact that these tasks were easily accomplishable, they were also very time intensive as each segment of the showcase materials was intricately prepared for the best possible outcome. As a result, more time was spent performing these tasks than originally anticipated.

## 9. ETA's Constructive Criticism

As previously stated in section 8.9, it was originally planned to have SafeGaze setup and configured in ETA's facility for proper testing, however due to certain complications (COVID-19 lockdown in particular), this was not fully achievable. Furthermore, the face-to-face meeting with ETA's CEO that took place on 21<sup>st</sup> April 2021 proved to be extremely beneficial because her constructive criticism outlined specific ramifications that were initially unclear.

The meeting began with positive comments regarding the usability of the application; she found it surprisingly easy for a facility to register a new account as it followed a simple and familiar process most websites use in modern day applications. In addition, the “Getting Started” documentation served as an important guide for ETA for the reason that it was highly detailed, yet still user friendly.

With regards to the recommendations that ETA offered, there were two major aspects that would need to be amended prior to the official release of SafeGaze. Essentially, the current configuration of the members’ access times creates a new ethical issue for participants with specific training times that differ on certain days of the week. Ethically speaking, if the facility were to grant a member a specific access time for a set day, the member would also be able to access the livestream within this time on every other day of the week. It is therefore apparent that the behaviour of SafeGaze’s access times must be changed to allow for each day of the week, as opposed to one designated time for every day.

In addition, it was found that ETA needed their cameras to be wirelessly connected to their computer due to the set angles that were required for recording gymnastics. With SafeGaze’s current configuration that makes use of a wired connection, this poses a challenge for ETA and similar facilities. Future work must be done to allow for a variety of different methods in connecting a camera to SafeGaze, to which the facility may choose based on what approach they feel would be most beneficial to their circumstances.

## 10. End Project Review

This project served as an opportunity to create a piece of software that would improve a leisure centre’s security by providing a safe live streaming platform for the facility’s members, and thus negates the risk of allowing strangers to access the dedicated viewing area. Additionally, in light of the recent COVID-19 pandemic, most designated viewing areas in sports facilities have been closed, or severely limited in an effort to comply with social distancing rules. SafeGaze nullifies the need for these viewing areas, as parents are able to watch their children remotely via one or numerous livestreams.

Overall, this project has been successful as it meets all objectives and most of the requirements stated in the project’s initiation phase. The application does exactly what it had originally aspired to do and allows a facility’s members to view a livestream of their participants engaging in such activities. Alongside this, the facility is able to restrict which users have access the application as well as what specific times the member is able to view the livestream(s). This is proved through the integration of SafeGaze for the client of this project – Exeter Trampoline Academy (ETA). Although, certain complications arose and therefore meant that SafeGaze did not reach its full potential due to the fact that the closure of leisure centres during in COVID-19 pandemic prevented the application being set up for ETA. This therefore interfered with the feedback issued, which could have been used to enhance the application’s functionality.

Some requirements have not been met; however, these do not necessarily make a substantial impact on the application’s functionality. They primarily relate to features involving profile management, such as the ability to change the user’s password once they have registered an account; or the ability to receive notifications from within the app.

SafeGaze was developed with the intention of it being expanded upon further after reaching the end of the academic year. Of course, prior to its official release to the public, important adjustments must be made according to the client’s needs in an effort to fully satisfy any future customers that implement such a system into their facility. It is known that ETA will undoubtedly remain a client, and intends on using SafeGaze when it has been finalised, by summer 2021.

## **11. Project Post-Mortem**

### **11.1 Project Management Evaluation**

Adopting an Agile Methodology to manage this project proved to be highly beneficial as it allowed for an iterative approach in reaching all deliverables, and thus meant that requirements could be altered if any obstacles were to present themselves in the development phase. Additionally, each sprint was laid out with exactly what was expected, which proved beneficial for the client, who did not have a full understanding of a typical software's development lifecycle.

Whilst the development generally followed a smooth process, there moments in which the sprint's dates/times had to be amended to adapt to the circumstances. For instance, Sprint 4 focused on the implementation of the livestreaming functionality for SafeGaze, however certain complications arose and therefore meant that the chosen method of approach had to be reassessed (elaborated in section 8.6).

To deal with such issues, it is clear that the method of approach of implementing these deliverables could have been researched more thoroughly to prevent the possibility of any delays during development. Additionally, the project's plan could have also accounted for potential obstacles that may hinder any development progress and should have been reflected in the times given for each sprint. Overall, this project was managed appropriately, however certain actions could have been taken to more effectively deal with the difficulties faced.

### **11.2 Objectives Evaluation**

This project was a success as it meets all of its aims & objectives (stated in section 1.3); however not all requirements have been fully accomplished, but these do not necessarily affect the functionality that is required to achieve the project's original aims. Upon creating a project plan, all deliverables were given a priority, as well as an estimate number of hours to complete it. This therefore affected the order at which the tasks were attended to, as these variables dictated which tasks were given priority over one another.

As a learning experience, the priority and estimated work hours served an important role in deciding which tasks to complete first within a given sprint, however the requirements could have been managed accordingly to make use a of 'Minimum Viable Product' (MVP). The tasks to achieve the project's aims & objectives would represent the MVP, whilst additional deliverables would be used to enhance the applications features/functionality.

### **11.3 Technologies Evaluation**

The technologies used for the deliverable proved to be optimal as they are by far the most commonly used tools and languages with regards to website development. In addition, due to previous experience, these technologies were easy to use as there was a greater understanding of their potential. However, an argument can be made that it would have been more beneficial to implement a full stack development framework, such as "React" or "Angular", as they potentially could have improved the speed of development, specifically with regards to the user interface. Alongside this, if the deliverable were to be made with a full stack framework, the security could potentially be superior to that which was developed in this project. Nevertheless, the development technologies were still beneficial in reaching the project's objectives.

With regards to project management technologies, certain tools undoubtedly enhanced the workflow of this project. For instance, Microsoft Office Planner was extremely beneficial for laying out the tasks in each sprint, as the user is able to assign a task with beneficial additions, which includes but not limited to: priority, start date, due date, and sub-tasks. Alongside this, other tools were highly effective

in the creation of certain assets, such as the use of Photoshop for the SafeGaze logo, or Gimp for the Project Poster.

#### **11.4 Development Performance Evaluation**

Throughout the entire duration of this project, the effort invested into reaching project's aims and objectives was in accordance with the university's 400 hour guideline and proved to be suitable for developing an system of such a large scale. Within the project's initiation, a lot of time was spent on the planning and organisation of how certain aspects would be directly approached, specifically in relation to the technologies chosen. Subsequently, due to the demand of other university modules and commitments, there was a slight decline of the effort devoted solely to the tasks allocated in this project, specifically within November 2020 and February 2021. Nevertheless, time was managed accordingly, and once other modules were no longer considered a main priority, development on SafeGaze resumed.

The work conducted in this project remained consistent throughout the academic year and there was no significant discontinuity between the different sprints. Ensuring that the project was continuously worked upon, it therefore meant that the development process did not follow an unpredictable pattern. In the end, performance was satisfactory, however the estimated work hours is one aspect that specifically could have been more precise as it was common for some tasks to be underestimated.

#### **11.5 Future Enhancements**

The final deliverable was developed with the aim of it being publicly available once reaching the end of this project. From the relevant market research conducted and feedback from different sports clubs and centres, it has become apparent that such a tool would be highly valued, and something that they would definitely implement. Of course, adjustments must be made prior its official release to fully benefit its clients, primarily the aspects that were addressed by ETA's CEO in the face-to-face meeting on 21<sup>st</sup> April 2021. As already explained in section 9, the behaviours surrounding the members' access times must be addressed as it creates an ethical issue. Furthermore, section 9 also describes why the camera's connection would be another aspect that may potentially stop some facilities from making use of SafeGaze.

Adopting YouTube as the livestreaming platform was used as a proof of concept for SafeGaze, and when it is officially released, a private stream hosting platform will be used in its place. Section 8.6 elaborates on this concept and explains why this would be beneficial. Furthermore, the final improvements to this project specifically relate to the requirements which have not yet been accomplished; for instance, the ability for a member to receive notifications when their account's status has changed.

Additionally, when this application has been released, it would be beneficial for the facility's members to make us of dedicated mobile app as this would present numerous advantages, such as optimised performance, and a more user friendly interface.

## **12. Conclusion**

This project has achieved exactly what was originally set, to improve the security of a sports facility by providing its members with a safe and secure live streaming platform. SafeGaze has demonstrated exactly how implementing such a system would be beneficial to said facility, given how the recent events of the global pandemic has completely overhauled how a business needs to now operate. The lockdown created something that one could never have foreseen, however it also created an opportunity in that software, such as SafeGaze, would now be used and needed by businesses.

All aims and objectives have been accomplished within the deadline; given more time, the system would have been enhanced with additional features, and a dedicated mobile app. Although the project generally followed a smooth process, there were some setbacks which halted development, most notably the 2021 lockdown. Nevertheless, these were quickly resolved, and future work will be conducted on SafeGaze with the aim of releasing it to the public by summer 2021.

## 13. References

- Altvater, A. (2020). What Is SDLC? Understand the Software Development Life Cycle. Viewed 24th Mar 2021. <<https://stackify.com/what-is-sdlc/>>
- Aston, B. (2021). Why Is Project Management So Important To An Organization? The Digital Project Manager. Viewed 24th Mar 2021. <<https://thedigitalprojectmanager.com/why-is-project-management-important/>>
- Atlassian. (2020). What is version control? Atlassian. Viewed 26th Mar 2021. <<https://www.atlassian.com/git/tutorials/what-is-version-control>>
- Curry, L. (2019). What is Office 365 Planner? A Beginner's Guide. Chorus. Viewed 29th Mar 2021. <<https://www.chorus.co/resources/news/what-is-office-365-planner-a-beginners-guide>>
- Digivate. (2013). Responsive Web Design, Dedicated Mobile Site or Mobile App – How to Choose? Digivate. Viewed 13th May 2021. <<https://www.digivate.com/blog/mobile/responsive-web-design-dedicated-mobile-site-or-mobile-app/>>
- Freibrun, E. (1995). Intellectual Property Rights in Software – What They Are and How to Protect Them. Freibrun Law. Viewed 8th Apr 2021. <<https://freibrunlaw.com/intellectual-property-rights-software-protect/>>
- Gehman, C. (2019). What Is Version Control? Perforce. Viewed 26th Mar 2021. <<https://www.perforce.com/blog/vcs/what-is-version-control>>
- GOSS Interactive. (2020). The importance of user testing. GOSS Interactive. Viewed 9th Apr 2021. <<https://www.gossinteractive.com/article/6699/The-importance-of-user-testing>>
- GOV. (2011). Data protection. GOV. Viewed 2nd Apr 2021. <<https://www.gov.uk/data-protection>>
- Jeston, J. (2018). Business Process Management: Practical Guidelines to Successful Implementations. 4th edn. Oxon: Routledge.
- Jevtic, G. (2019). What is SDLC? Phases of Software Development, Models, & Best Practices. Viewed 24th Mar 2021. <<https://phoenixnap.com/blog/software-development-life-cycle>>
- Karlsson, J. (2020). What Is a Project Backlog + Examples? Perforce Software. Viewed 29th Mar 2021. <<https://www.perforce.com/blog/hns/project-backlog-examples>>
- Law Donut. (2021). Data protection and IT. Law Donut. Viewed 2nd Apr 2021. <<https://www.lawdonut.co.uk/business/data-protection-and-it/data-protection-and-it-overview>>
- NSPCC Learning. (2021). Safeguarding children and child protection. NSPCC Learning. Viewed 3rd Apr 2021. <<https://learning.nspcc.org.uk/safeguarding-child-protection>>
- Rehkopf, M. (2020). What are sprints? Atlassian Agile Coach. Viewed 26th Mar 2021. <<https://www.atlassian.com/agile/scrum/sprints>>
- Semusheva, O. (2016). Requirements. Why is it important? Steel Wiki. Viewed 10th Apr 2021. <<https://steelkiwi.com/blog/requirements-why-it-important>>
- Smith, S. (2017). The importance of planning when it comes to systems development. Hypertext. Viewed 24th Mar 2021. <<https://www.hyperext.com/software-development/the-importance-of-planning-when-it-comes-to-systems-development>>

Thomson, A. Schmoldt, D. (2001). Ethics in computer software design and development. Pp. 85-102. Elsevier: Amsterdam.

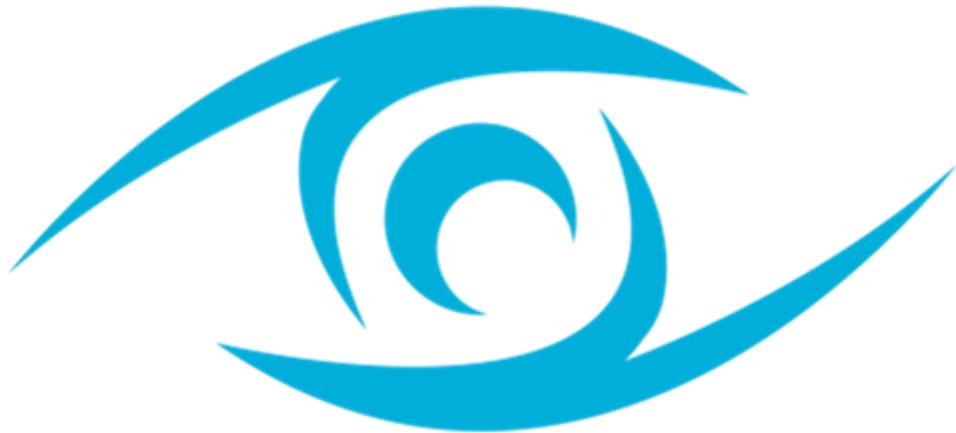
Visual Paradigm. (2020). What is Agile Software Development?. Visual Paradigm. Viewed 25th Mar 2021. <<https://www.visual-paradigm.com/scrum/what-is-agile-software-development/>>

Winqvist, J. (2019). 5 reasons why you should start using Microsoft Teams today. Microsoft Pulse. Viewed 30th Mar 2021. <<https://pulse.microsoft.com/en/work-productivity-en/na/fa2-reasons-why-you-should-start-using-microsoft-teams-today/>>

## 14. Appendix

### 14.1 Documentation

#### Getting Started



## Introduction

### **What is SafeGaze?**

This page acts as a guide for how to properly use SafeGaze at its full potential.

SafeGaze is directed towards the physical safety and cybersecurity of sports clubs and facilities. It focusses on providing a safe streaming platform for your members, and thus negates the risk of stalking or harassment of strangers and your members, especially when the participants primarily consist of children. By implementing our system, you will be able to create a secure livestreaming platform which allows your members to safely watch their children taking part in such activities.

### **How it works?**

You will be able to control which members have a SafeGaze account that is linked to your facility, alongside with what times each specific user has access to the livestreams.

This application is split into two separate systems: one for the facility, and the other for the facility's members, each of which can be access through their respected login pages (facility sections = blue & member sections = green).

## Livestreams

### ***Setting up a livestream***

As of the moment, SafeGaze makes use of YouTube livestreams "as a proof of concept", as private stream hosting platforms require a paid subscription to host a livestream. You will need to create a YouTube livestream that is marked as "Unlisted", which means it is not searchable within YouTube, yet still can be embedded into websites.

To capture video through a DSLR camera and encode the footage to your computer, a "video capture card" is required, which is generally connected with a HDMI cable. Furthermore, OBS software is used to feed the footage directly to the YouTube stream. More information on how to do this can be accessed through <https://www.digitaltrends.com/computing/how-to-live-stream-on-youtube-with-obs/>.

### ***Linking livestream to SafeGaze***

The next step is to link the YouTube livestream to SafeGaze. Click on "Share" within the YouTube livestream and copy the URL. Paste this into the "Create New Livestream" section in SafeGaze alongside with a title and optional livestream notes.

Upon submission, the livestream will now be available to you and your members.

To delete a livestream or edit its details, use the specified buttons available in the livestream List.

## Managing members

A member can sign up for a SafeGaze member's account and link it with your facility with the use of your Facility Access Code (created in sign up). You can view your access code by navigation to the "Facility Information" section in the "Manage Facility" page.

### ***Requests***

When a member signs up, their account will be shown in "Member requests" and will remain pending until you either accept or decline their account.

Upon hitting "Accept" you then need to input what times they will have access to the livestreams.

### ***Active members***

All members that have access to your facility's livestreams will be shown under "Active members". You can change their account's status and livestream access times by clicking on the "Manage" button displayed to the right of each user.

### ***Inactive/declined members***

Any members that have been declined or marked as "Inactive" (meaning their account has been disabled), will be listed in the "IDLE members" page.

## **Facility information**

Your facility's information will be displayed to all active members, which includes the following:

- Facility name
- Facility username
- Facility email address
- Facility telephone number
- Facility street address
- Facility postcode
- Facility access code

Some of these details can be changed by accessing "Facility Information" in the "Manage Facility" page.

## **Issuing messages**

Lastly, you will be able to issue messages to all active members that have a SafeGaze account. By navigating to "Messages" in the "Manage facility" page, you can create new messages, as well as view a list of all existing messages.

Messages are marked with an "urgency level", which can be useful in notifying your members regarding important announcements.

## 14.2 Project Management

### 14.2.1 Backlog

The screenshot shows the Microsoft Planner interface for the 'COMP3000 Assessment - Reece' project. The backlog is divided into four sprints:

- Sprint 0 (Planning & Preparation):** Contains 20 tasks, including 'Create risk plan' (Completed), 'Create project initiation' (Completed), 'Finalise and organise backlog' (Completed), 'Create webflow diagram (flowchart)' (Completed), and 'List member requests' (Completed).
- Sprint 1 (Facility Focused):** Contains 11 tasks, including 'Create dashboard for listing items' (Completed), 'Refine facility signup to check for other users' email address and access code to prevent duplicates' (Completed), 'Organise dashboard' (Completed), and 'Start planning report - different sections and notes' (Completed).
- Sprint 2 (Member focused):** Contains 6 tasks, including 'Display member's training times' (Completed), 'Pending/inactive/declined users to have better "unauthorised access" content on the verify-member page' (Completed), and 'Completed by (s) Reece Davies o...' (Completed).
- Sprint 4 (livestream focused):** Contains 12 tasks, including 'Allow member to receive notifications when their access times have been changed' (Completed), 'Refine your method of approach for a private livestream' (Completed), 'Display message for members when they are active but not in their "access times" (including current time for when access and no access)' (Completed), 'Display all facility's available livestreams for members to view' (Completed), and 'Note client's constructive criticism improvements that can be made to the app' (To be stated in rep...).

The screenshot shows the Microsoft Planner interface for the 'COMP3000 Assessment - Reece' project. The backlog is divided into four sprints:

- Sprint 3 (management focused):** Contains 5 tasks, including 'Allow member to receive notifications when their account status has been changed' (Completed), 'Display proper message for access time(s) for member' (Completed), and 'Allow facility to alter a member's' (Completed).
- Sprint 4 (livestream focused):** Contains 12 tasks, including 'Refine your method of approach for a private livestream' (Completed), 'Display message for members when they are active but not in their "access times" (including current time for when access and no access)' (Completed), and 'Display all facility's available livestreams for members to view' (Completed).
- Sprint 5 (Messages and Facility info focus):** Contains 4 tasks, including 'Allow new notifications to immediately notify the member with an icon/sound/email' (Completed), 'Refine styling of messages/notifications' (Completed), and 'Allow members to view all messages alongside with their details' (Completed).
- Sprint 6 (usability testing focused):** Contains 3 tasks, including 'Allow user to change password account' (Completed), 'Note client's constructive criticism improvements that can be made to the app' (To be stated in rep...), and 'Send video demonstration to client and request feedback on their thoughts and feelings' (Completed).

The image displays two screenshots of the Microsoft Planner application interface. Both screenshots show a board titled 'COMP3000 Assessment - ...' created by 'Reece Davies'. The top screenshot shows a 'Showcase Tasks' section with five items:

- Upload video to YouTube**: Completed by Reece Davies.
- Submit zip file to DLE**: Completed by Reece Davies.
- Video demonstration (using "Getting started" guide as script)**: Status 3/3, completed by Reece Davies.
- Completed**: Completed by Reece Davies.
- Download documents**: Completed by Reece Davies.

The bottom screenshot shows a 'Use Cases' section with four items under the 'Parent/Participant' category:

- I want to be able to view the sports club's contact information on the information page.
- I want to be able to receive notifications from important messages issued by the sports club, including changes to my account status.
- I want to be able to view all messages issued by the sports club.
- I want to be able to view the livestream.

Both screenshots also include sections for 'Functional Requirements', 'Database Tables', and other task categories like 'Styling Improvements' and 'Software Issues / Improvements'.

## 14.2.2 Risk Plan

Risk	Likelihood	Impact	Solution
Poor estimation and scheduling per functional requirement	Low	Medium	When creating the Kanban board, provide adequate times per task by accounting for anomalies or time consuming incidences. This therefore means each task will be achievable within the given time frame.
Poor time management due to other commitments outside university	Medium	Medium	Organise tasks into designated working days and weeks, making sure to account for potential changes in the project scope and possibility of outside university events which may affect time management.

Lack of frequent backups or potential loss of data	Low	High	Ensure there are regular git requests to the GitHub repository after making any significant changes to the project. Potentially have a second backup of the project in case of any Git failures.
Loss of valuable time due to obsessive use of “gold plating”	Low	Medium	Prevent use of adding unnecessary features until primary objectives have been reached. Prioritise the functional requirements set, as opposed to adding visually pleasing features that do not contribute to the end goal.
Technical difficulties with related software or hardware resources used in presentations and showcases	Medium	Medium	Thoroughly test available resources in different environments to guarantee a successful delivery in showcases made on personal or public hardware.
Lack of knowledge or experience within specific programming languages, frameworks, or software	Medium	Low	Properly research available resources required for production, as well as potential software resources which may prove to be beneficial.
Poor code quality	Low	Medium	Plan ahead of how the code will be structured to ensure it is properly organised in the different directories or files. When a functional requirement has been met, perform specific test cases to validate that it is working effectively.
Livestream(s) open to all users, therefore creating the potential risk of stalking or harassment from strangers	Low	High	Users require verification to sign up, alongside with proof of being involved in the sports club. Additionally, users will only be able to partake in livestreams at a given time when their child is participating in such activities. Due to how sports clubs are often run, the participant would be given a specific timeslot in a weekly timetable, and thus the parent would only have access to the livestream within this allocated time.
Breach of users' personal data due to software	Medium	High	Ensure all users' personal data is fully encrypted and highly secure within the app and database. Extra steps need to be taken into consideration with the

hack or weak security			security the mobile app and website. This includes, but not limited to: SQL injection, cross site scripting (XSS), sensitive data exposure, and global variables.
Software efficiency degradation	Low	High	Structure code in the most efficient way possible; prevent use of unnecessary tasks which are CPU intensive.

#### 14.2.3 Bi-weekly Reports

Available at: <https://dle.plymouth.ac.uk/mod/oublog/view.php?id=922745>

##### Weekly report #1 (Fri 23 Oct 2020)

This was the week in which the Project Initiation document was to be submitted on Thursday 22<sup>nd</sup> October 2020. My time primarily focused on this and therefore I have declared what software application I am developing for my final year project. As the Project Initiation has been finalised, I now have a detailed projection vision, a risk plan, project backlog, and a GitHub source code repository. Next steps are to populate the backlog with useful information, such as use cases, functional requirements, and general ideas/comments I might have that was not stated in the project vision.

##### Weekly report #2 (Fri 30 Oct 2020)

I have created a “Documents” folder in the GitHub repository which now contains a word document for all functional requirements and use cases for both users: Administrator and Parent/Participant. This information has also been replicated in the project backlog alongside with planning my current tasks for this sprint. I also requested web hosting from Shirley Atkinson for when I start developing the system. I have been provided with the details to my web host directory as well as guidance on how to access it and upload files. Next steps are to design the system database and the user interface for the application(s) so I can have an understanding of how it will function and how it would look.

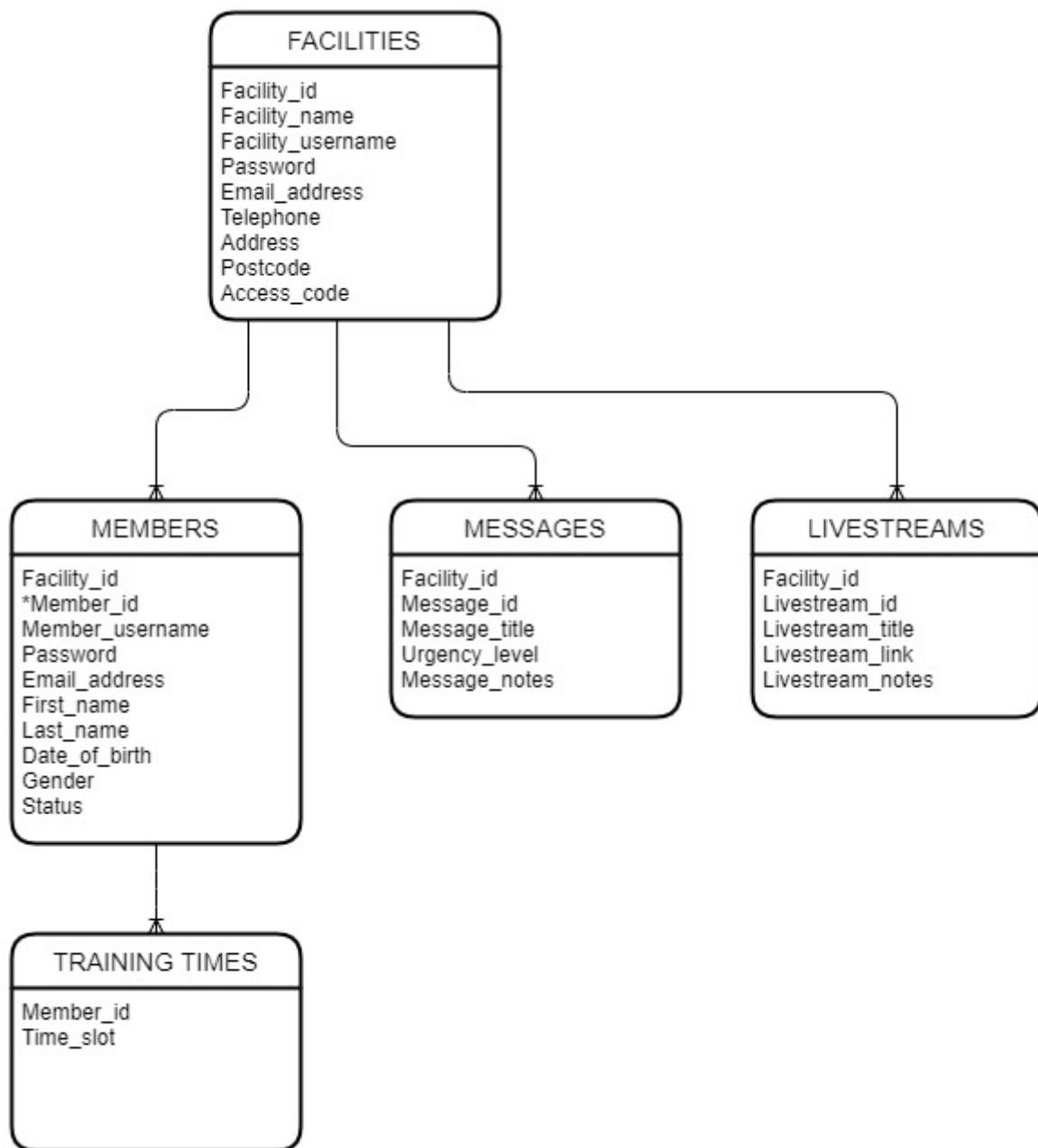
The screenshot shows a project management board with four columns:

- Project Tasks:** Contains three items:
  - Research Plymouth University hosting method of use
  - Design database
  - Create index page for SafeGaze (admin focused)
- Ideas & Notes:** Contains one item:
  - One user admin per sports club. They have control over name for member code for sports club entry (for participants)
- Use Cases:** Contains five items, each labeled "Parent/Participant":
  - I want to be able to view the sports club's contact information on the information page
  - I want to be able to receive notifications from important messages issued by the sports club
  - I want to be able to view all messages issued by the sports club
  - I want to be able to view the livestream
  - View user's livestream accessibility times
- Functional Requirements:** Contains five items, each labeled "Parent/Participant":
  - View sports club's contact info (information page)
  - Receive notifications from import messages issued by the sports club
  - View all messages issued by the sports club
  - View the livestream
  - View user's livestream accessibility times

At the bottom left, there is a "Show completed" dropdown set to 5.

## Bi-weekly report #3 (Fri 13 Nov 2020)

The weekly report has recently changed to Bi-Weekly reports. Within the previous two weeks I have designed the database as well as created the SQL code for creating the tables. To confirm that the database has been normalised properly, I have contacted my project supervisor for any changes that might be beneficial, or mandatory. Additionally, I have also added an SQL script that will add sample data to the database which will be used for testing on the prototype (dev system); this was achieved using the software PhpMyAdmin. Further planning has been done on the project backlog (Microsoft Planner) in which tasks have been allocated into separate sprints, therefore allowing me to track my progress for the entirety of the project. Lastly, the deliverable has been started; the directories have been configured in the correct manner and it currently contains the index page.



## Bi-weekly report #4 (Fri 27 Nov 2020)

Within the previous two weeks I started working on the deliverable where facilities are able to register a new account, as well as then being able to log in and out of their account. The PHP backend was also finetuned for security purposes. Subsequently, the numerous facility pages were set up and a dashboard UI design was created (using Bootstrap). This will be for listing members for the facility and therefore being able to manage them accordingly. Additionally, the navigation bar and dashboard UI were finetuned to be better suited for this specific website.

Section title				
#	Header	Header	Header	Header
1,001	lorem	ipsum	dolor	sit
1,002	amet	consectetur	adipiscing	elit
1,003	integer	nec	odio	Praesent
1,004	libero	Sed	cursus	ante
1,005	dapibus	diam	Sed	nisi
1,006	nulla	quis	sem	at
1,007	nibh	elementum	imperdiet	Duis
	sagittis	ipsum	Praesent	mauris

## Bi-weekly report #5 (Fri 11 Dec 2020)

Unfortunately, due to other university modules having deadlines within these two weeks, I have been unable to do much work on the Computing Project. However, I have organised the tasks in the project backlog into separate sprints for a more organised structure. This means I will understand which tasks I must allocate to myself first and thus the time management for this project will be more efficient.

## Bi-weekly report #6 (Fri 01 Jan 2021)

During the Christmas holiday, the focus was on allowing a Facility to manage its members. The different member pages (active, requests and idle) have been connected to the database and will link the different members according to their account status (accepted, pending, declined, and inactive). The active members page and idle members page have buttons that take the user to a new page which will be used to change the member's status and access times. Furthermore, members are able to request for a new account, which is linked to the facility through the facility's access code. They are then taken to a specific page which will decline the user access if they their status is anything other than "accepted".

Furthermore, a "development process" document has been added which contains screenshots and notes of the deliverable. These notes and screenshots will primarily be used for the report in showing how the final deliverable has changed through time and what it had originally looked like.

## Bi-weekly report #7 (Fri 15 Jan 2021)

Started working on getting the livestream functionality working but having a lot of difficulty with it. Trying to connect IP camera to computer through RTSP. Works on VLC media player but can't seem to connect it to embedded HTML code for the website (<https://videos.cctvcamerapro.com/i/ip-camera->

[streaming-website.html](#)) but might instead use a YouTube (or other streaming service) for the livestream functionality (<https://videos.cctvcamerapros.com/i/ip-camera-streaming-website.html>). Upon further research I found that I required additional hardware – a capture card. This would allow the DSLR camera to output live footage to the computer to be encoded for the stream, and thus I ordered one off amazon.co.uk. In the meantime I continued work on the ‘manage member’ page, specifically the training times and a few smaller things on different pages.

### Bi-weekly report #8 (Fri 29 Jan 2021)

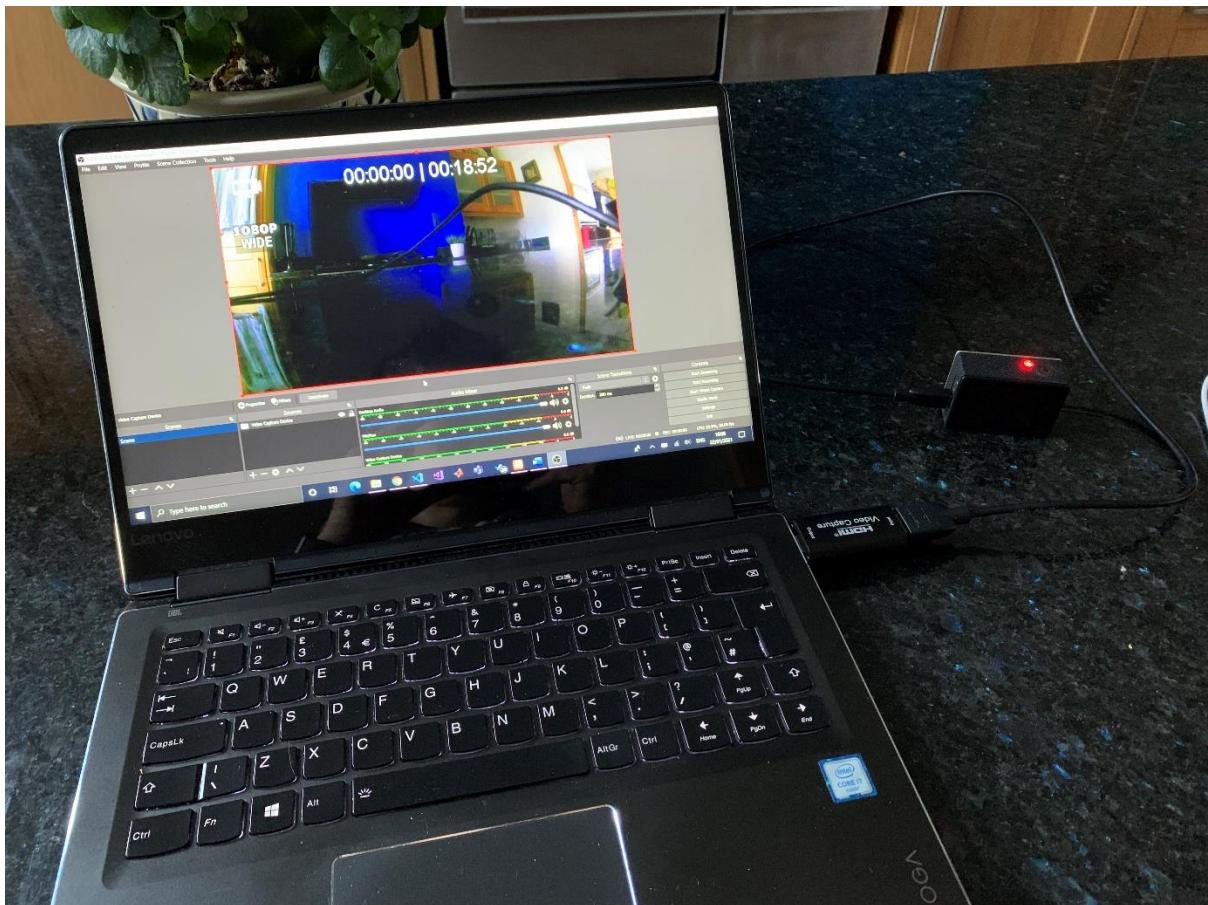
Been very busy due to lockdown the 3rd. Adjusted the training\_times table in DB to allow for check what training times the member has access to. Changed ERD and code in repository to reflect these changes. Created a user verification system that will check if status of the member and allow access if they have status “active” but not if they are anything otherwise (pending, inactive, declined). If they have not been accepted, they will be forced to the verification-page which displays their status (which will give an explanation to user – not yet implemented).

Facility can now manage a member’s training times by clicking the relevant checkboxes; DB row for the specific user is reset and updated with the new values selected.

The screenshot shows a web browser window with the URL 'localhost/deliverable/manage-member.php?memberId=7'. The page is titled 'Member : Fifth User'. On the left, there's a sidebar with 'MEMBERS' section containing 'Active members', 'Requests', and 'Idle members'. The main content area displays the member's details: Member ID: 7, Username: fifthUser, Email: fifthuser@test.com, First name: Fifth, Last name: User, Date of birth: 2020-01-01, Gender: male, and Status: ACTIVE. Below these fields is a dropdown menu labeled 'Select member status' with an arrow pointing down. A large blue button at the bottom right says 'Change status'. At the very bottom of the page, under the heading 'Training times', there is some placeholder text: 'Updated manage-member page for listing the member’s training times in the form of a table. Refined the manage-member styling to exclude unnecessary content. Connected and started testing streaming the capture card to the laptop. Researched process for livestreaming using capture card. Requested streaming access from YouTube for testing (takes at least 24 hours for verification). In the meantime, created manage-livestreams page for facility.'

Updated manage-member page for listing the member’s training times in the form of a table. Refined the manage-member styling to exclude unnecessary content. Connected and started testing streaming the capture card to the laptop. Researched process for livestreaming using capture card. Requested streaming access from YouTube for testing (takes at least 24 hours for verification). In the meantime, created manage-livestreams page for facility.

Testing out livestreams. Complicated. Will be using YouTube as a proof of concept for the time being and then migrate to a secure private streaming service in the future (costs money). YouTube has security flaws with this type of security application and some tedious things (stated in important livestream notes document).



Working on assessing the member's training times and giving restricted access. Started and finished backend script for posting livestream URL to database but need to decide if a facility will have one or numerous livestreams. Multiple livestreams decided.

Started listing all available livestreams facility offers along with their details. Next steps to finish styling livestreams list for facility (styling it as at the moment it does not look professional) and start listing livestreams for members.

LIVESTREAMS  
Manage livestreams

Create New Livestream

Title

URL

Notes

Create Livestream

Existing Livestreams

Facility ID = 7  
Livestream ID = 16  
Title = Test title  
Link = <https://www.youtube.com/embed/BnpjhKG11Kg>  
Notes = Test notes

Reece Dylavies Test Live Stream 00:18:52 Watch Later Share

## Bi-weekly report #9 (Fri 12 Feb 2021)

Styled livestreams listview on facility's manage-livestream page.

Allowed livestream to delete livestream.

Displayed list of member's access times on their livestream page.

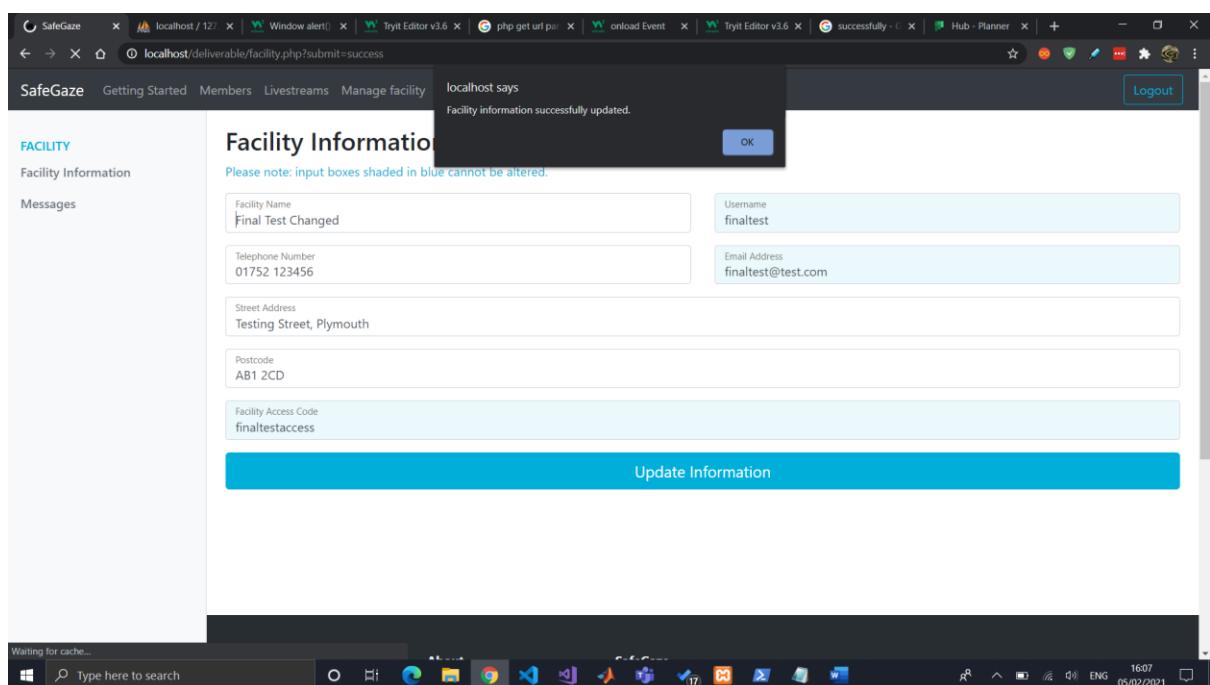
Styled verify-user page for members when their account is pending/inactive/declined.

Styled member's livestream page with notification of access time + short message.

Connected DB livestreams to appear in member's livestream page + styled them.

Created footer and styled it.

Facility can update their information (excluding some like email address and username).



Members can now view their facility's details.

Started page for facility to create and view all messages issued to members.

Then connected it to DB (with form) and improved styling.

The screenshot shows a web application window titled "SafeGaze" with the URL "localhost/deliverable/messages.php". The interface includes a sidebar with "Facility Information" and "Messages" options, and a main content area with a "Create New Message" form and a "Existing Messages" list.

**Create New Message (issued to all members)**

- Title: [Input field]
- Select message urgency level: [Dropdown menu]
- Contents: [Text area]

**Notify Members**

**Existing Messages**

- Medium urgency message**  
Urgency: **Medium**  
Testing medium urgency.  
Edit | Delete
- Low urgency message**  
Urgency: **Low**  
Testing low urgency.  
Edit | Delete
- Urgent message**  
Urgency: **Urgent**  
Testing high urgency message.  
Edit | Delete

Styled facility and member login as well as navbar according to their colour scheme.

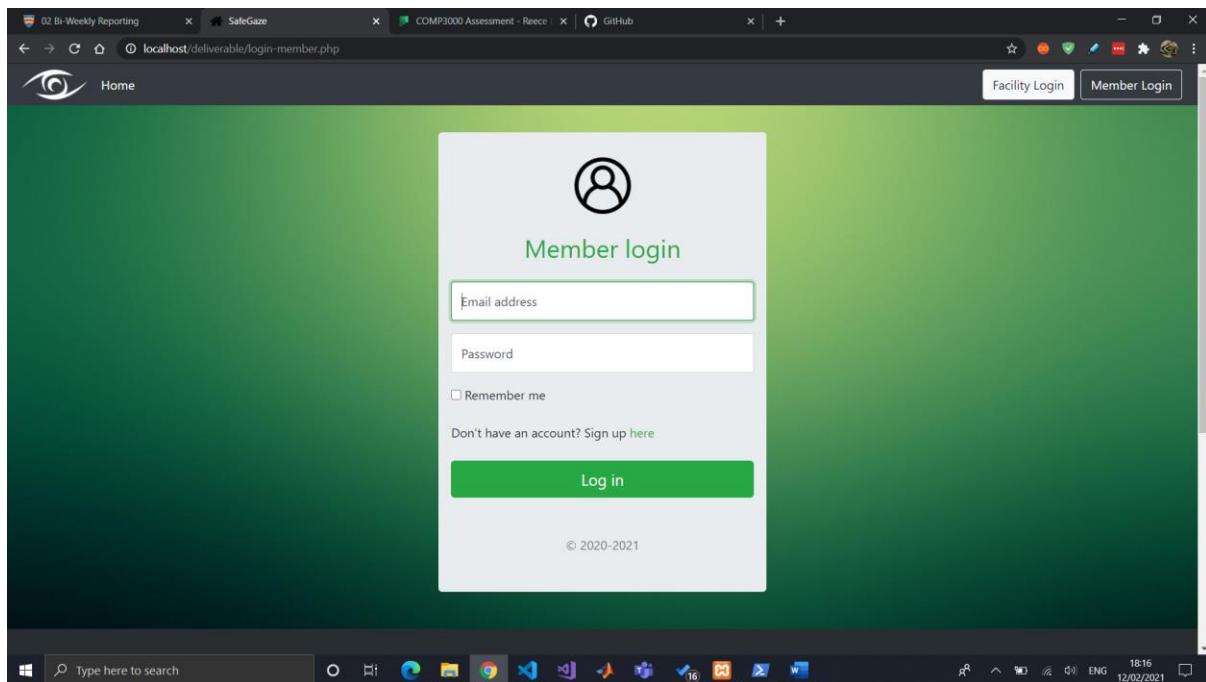
The screenshot shows a web application window titled "SafeGaze" with the URL "localhost/deliverable/login-facility.php". The interface features a dark blue background with light-colored UI elements. It includes a "Facility Login" button in the top right corner and a central login form with a user icon and the text "Facility login".

**Facility login**

Email address  
Password  
 Remember me  
Don't have an account? Sign up [here](#)

**Log in**

© 2020-2021



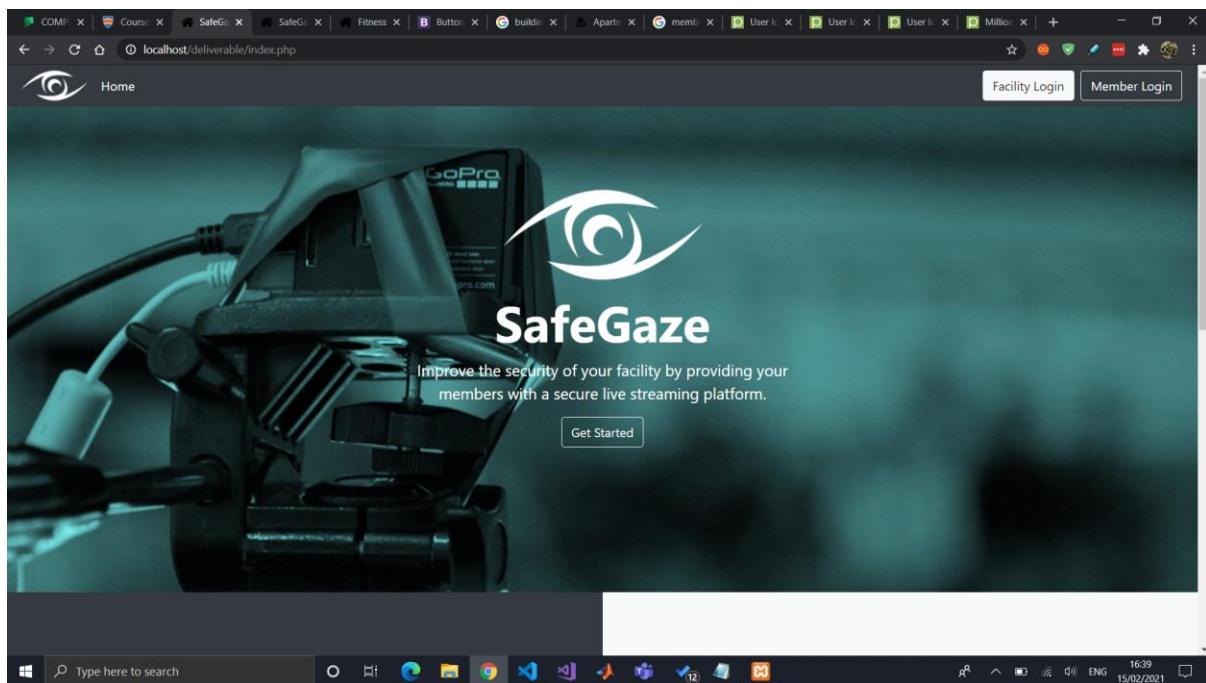
Created SafeGaze logo in Photoshop

Made small styling changes to website, including adding logo to navbar (with respected colour scheme) and footer.

Continued working on report.

### Bi-weekly report #10 (Fri 26 Feb 2021)

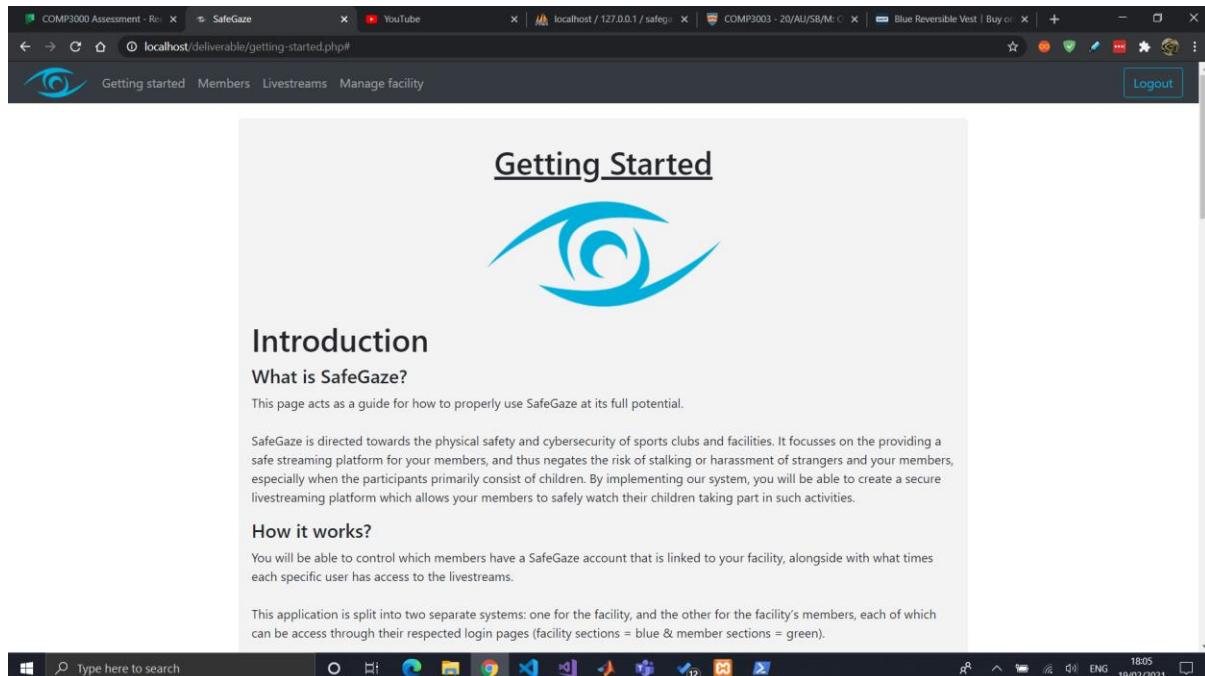
Styled the index page for better presentation.



Started working on the "Getting started guide" document which will act as documentation for the facilities in the "getting-started" page.

Sent video demonstration to client (Exeter Trampoline Academy) for constructive feedback on potential improvements. Response has been received and it has been input into a word document called "ETA feedback.docx".

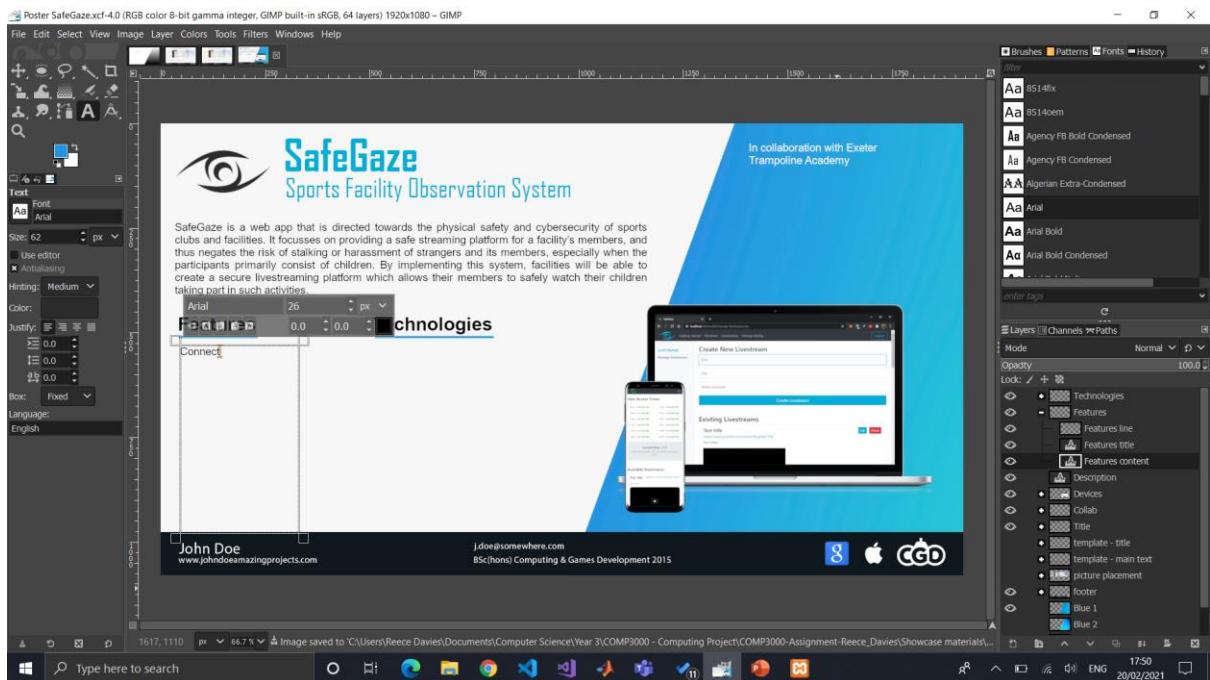
Input content into "Getting started" page, for documentation, and also acting as script for showcase video demonstration.



Took content from "Getting started" guide and transferred it into a script for showcase materials video demonstration.

Did a little more work on Report (introduction section – project aims & objectives).

Started and finished project poster for showcase materials. Was not able to access Photoshop, so it was created with the use of GIMP. Also recorded the audio for the video. Screen video capture will be recorded Monday 1<sup>st</sup> March.



Facility can now edit a livestream's details (all functionality of website is done – except notifications, which might instead be a future improvement to be stated in the report).

### Bi-weekly report #11 (Fri 12 Mar 2021)

Due to other module deadlines, I have not spent too much time working on it within the past week. Although, being very organised with SafeGaze's project, I have finished and submitted the showcase materials. Alongside this, I have finished editing the video demonstration and uploaded it to YouTube.

### Bi-weekly report #12 (Fri 26 Mar 2021)

Been working on report. Finished introduction section and started project management section (the importance of project management + method of approach + explaining bi-weekly supervisor meetings and DLE blog reports).

### Bi-weekly report #13 (Fri 23 Apr 2021)

Report has made decent progress. Currently done just under 6,000 words and on the "Project Phases" section where I explain the development process of each sprint. Once that is done, the remaining sections are the end project report, post-mortem, which will include some things like client feedback, improvements, difficulties, etc.

On Wed 21<sup>st</sup> Apr 2021, I had a face to face sit-down meeting with Polly (ETA CEO) to see if we can get SafeGaze set up. To my surprise, there were a few issues that would prevent this from happening, with the primary reason being that the access times cannot be relative to a specific day of the week. This was a problem because some members at ETA train only on set days, therefore raising the ethical issue of SafeGaze members able to access the livestream at their given access time throughout the entirety of the week, when in fact it would be desirable for them to only access it on a Monday, as an example. I created a document of their feedback and general discussions of the meeting. Just to mention, she really liked the system.

Instead, due to her maternity leave which commences Saturday 24<sup>th</sup> Apr, I will be setting up SafeGaze for her account so she will be able to monitor the facility while away. This is will also allow for her to

provide me with some more useful feedback on testing the somewhat ‘production’ version of the system.

#### Bi-weekly report #14 (Fri 7 May 2021)

Report is nearly completed. Just writing the “end project report” and “project post-mortem”. Once that is done, it is just refining the figures, references, and appendix.

#### 14.2.4 Bi-weekly Meeting Minutes

16 Dec 2020

Point 4 is more for other developers whilst point 5 is more for how the user would use the program.

Demonstration is your computer (so I assume localhost should be fine).

Try checking out YouTube private livestreams for your app.

03 Jan 2021

Ordered new hardware (capture device) - researched it and got it working with livestreaming directly to youtube

Started working on livestream functionality (confusing and tedious)

- Facility livestream page: create new livestream + view their livestreams
  - ^ later connected to DB
- Member's functionality for livestreams (access times, etc)

Also added a lot of screenshots of the development process. Would these be useful in the report and what would be best practise for this?

- Keep hold of critical screenshots
- Put other ones in the appendix

YouTube or other private streaming service

- Can use YouTube as a "viable proof of concept"
- State how this can grow (conclusions, limitations, future work, amendments)

Next steps:

- Added functionality
- Decide method of approach for livestream
- Do more work on report + general things stated in the meeting earlier today (showcase material planning)

## 06 Feb 2021

YouTube would make sense. If that doesn't work you could potentially look into a downloadable client. Web conferencing need the client. So two parts (HTML + Client). (E.g using zoom and the website creates a zoom session for people to watch and then the recording is then available on your website).

Also look into how to encode your video to your computer (<https://colorwhistle.com/live-stream-video-on-website/>).

Guide for youtube livestream (<https://www.youtube.com/watch?v=gch4ReBTirI>) - Elgato Cam Link

## 17 Feb 2021

--- Run through of what I have done ---

Finished all main functionality

- Tasks all over the place at the moment (lots of little ones)

Important but not functional requirements

- Styling aspects
- Logo

Done a decent amount of work on report (would like it to be a little more - but I do have useful notes)

Sent a video demonstration to my client (Exeter Trampoline Academy) - awaiting response (NOT SHOWCASE)

Started planning showcase materials

Started with "Getting started" guide (pretty important)

-- Kimberly notes ---

You basically want the report to be done by 17th May.

### 14.2.5 Project Plan

	<b>Deliverables</b>	<b>Estimate Hours of Work</b>
<b>Sprint 0: Planning &amp;</b>	Create project initiation (including market research, project vision, objectives, & risk plan)	5
	Create and populate backlog	3

<b>Sprint 1: Facility focused</b>	Set up repository & web hosting, organise important documents (report, notes, minutes, etc.)	1
	Research frameworks & tools to use	4
	Create requirements and use cases for both users	5
	Create project plan & Gantt chart	10
	Design system flowchart	5
	Create UI design	3
	Design database schema	4
<b>Sprint 2: Member focused</b>	Set up app's initial pages (index, dashboard, login, etc.) & install Bootstrap to system	3
	Create SQL statements for database	3
	Allow facility to sign up	4
	Allow facility to log in	2
	List all active members	6
	List all member requests	1
	List all idle members (declined + inactive)	2
<b>Sprint 3: Management focused</b>	Allow member to sign up	3
	Allow member to log in	2
	Allow facility to accept/decline member signup requests	3
	Pending/inactive/declined members to have a "Verify member" page to eliminate unauthorised access	4
	Display a member's access times to the member	4
<b>Sprint 4: Livestream focused</b>	Allow facility to view all a member's details in a separate page ("manage member" page)	5
	Allow facility to change a member's account status	3
	Allow facility to view a member's access times	1
	Allow facility to alter a member's access times	3
<b>Sprint 5: Message focused</b>	Conduct further research into method of approach for implementing livestream	10
	Allow facility to create a new livestream	6
	Display all available livestreams to facility	5
	Display all available livestreams to member	3
	Allow member to ONLY access a livestream within their access times	4
	Display popup message for members who are not within their "access times"	1
	Allow facility to edit an existing livestream	5
	Allow facility to delete an existing livestream	2
<b>Sprint 5: Message focused</b>	Allow facility to issue a new message/notification to their members	8
	Allow facility to view all existing messages	5

	Allow member to view all existing messages	2
<b>Sprint 6: Testing</b>	Create video demonstration for ETA	16
	Inform ETA of development state + request feedback from video demo	5
	Log feedback received from ETA for future developments + improvements	2
	Create documentation for facilities on how to use SafeGaze	6
<b>Sprint 7: Finalisation</b>	Adjust system to resolve potential errors/bugs found in testing	10
	Refine UI styling	6
	Create showcase materials	14
	Project report finalisation	N/A

#### 14.2.6 Use Cases

##### Facility

I want to be able to login to my account

I want to be able to logout of my account

**I want to be able to change the password to my account**

I want to be able to view my sports facility's details (including access code - created in signup)

I want to be able to change my sports facility's details

I want to be able to create a new livestream for my members

I want to be able to edit an existing livestream's details

I want to be able to remove an existing livestream

I want to be able to view all members linked to my facility (active, inactive, and declined)

I want to be able to view member signup requests

I want to be able to approve/decline member signup requests

I want to be able to edit a member's account status

I want to be able to view a member's livestream accessibility times

I want to be able to edit a member's livestream accessibility times

I want to be able to view all messages issued to members

I want to be able to create new messages to members

**I want to be able to edit existing messages' details**

I want to be able to remove messages issued to members

## Member

I want to be able to login to my account

I want to be able to logout of my account

**I want to be able to change the password to my account**

I want to be able to view my livestream accessibility times

I want to be able to watch the available livestreams

I want to be able to view all messages issued by the sports facility

**I want to be able to be notified when my facility posted “urgent” messages, changes my account status, or changes my livestream accessibility times**

I want to be able to view the sports club’s details and contact information

## 14.3 Database

### 14.3.1 SQL Code

```
-- phpMyAdmin SQL Dump
-- version 5.0.2
-- https://www.phpmyadmin.net/
--
-- Host: 127.0.0.1
-- Generation Time: Apr 19, 2021 at 04:09 PM
-- Server version: 10.4.11-MariaDB
-- PHP Version: 7.4.4

SET SQL_MODE = "NO_AUTO_VALUE_ON_ZERO";
START TRANSACTION;
SET time_zone = "+00:00";

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET @OLD_CHARACTER_SET_RESULTS=@@CHARACTER_SET_RESULTS */;
/*!40101 SET @OLD_COLLATION_CONNECTION=@@COLLATION_CONNECTION */;
/*!40101 SET NAMES utf8mb4 */;

--
-- Database: `safegaze`
--

-----
-- Table structure for table `facilities`


--
```

```

CREATE TABLE `facilities` (
  `facility_id` int(255) NOT NULL,
  `facility_name` varchar(50) NOT NULL,
  `facility_username` varchar(25) NOT NULL,
  `password` varchar(100) NOT NULL,
  `email_address` varchar(50) NOT NULL,
  `telephone` varchar(20) NOT NULL,
  `address` text NOT NULL,
  `postcode` varchar(15) NOT NULL,
  `access_code` varchar(50) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 
-- Dumping data for table `facilities`
-- 

INSERT INTO `facilities` (`facility_id`, `facility_name`, `facility_username`, `password`, `email_address`, `telephone`, `address`, `postcode`, `access_code`) VALUES
(1, 'Test facility (admin)', 'admin', 'admin', 'admin@test.com', '01752', 'Not a real address', 'XX1234XX', 'newFacility'),
(2, 'new facility', 'new_facility', 'password', 'newfacility@gmail.com', '01752', 'Address stated', 'ZZ1234ZZ', 'testAccess'),
(3, 'sql insert statement test', 'SQLTest', 'password', 'sql@gmail.com', '01752', 'Uh', 'AA12AA', 'sqlTestAccess'),
(4, 'First Test', 'firsttest', 'firsttest', 'firsttest@test.com', '01752', 'First Test Street', 'AB12CD', 'firsttest'),
(5, 'Second Test', 'secondtest', '$2y$10$8LbATAzM/Jy2qLp2p9EsS.zna2MVQus4NoeysJBhAreT/7vGYVDSq', 'secondtest@test.com', '01752', 'Second Test Street', 'AB12CD', 'second-test'),
(6, 'Third Test', 'thirdtest', '$2y$10$/5Qn1AEjGnQ7Vz9AN.AjHucf7YxEyW..x4q6i9J3f3FBM6wArbI8y', 'thirdtest@test.com', '01752', 'Third Test Street', 'AB12CD', 'thirdtest'),
(7, 'Final Test Changed', 'finaltest', '$2y$10$.FgsgffBVC/5rKHxrtWp10swlKoHZXdCFvotE6RzrQF7fc.exYtaG', 'finaltest@test.com', '01752 123456', 'Testing Street , Plymouth', 'AB1 2CD', 'finaltestaccess'),
(8, 'Error Test', 'errortest', '$2y$10$.FgsgffBVC/5rKHxrtWp10swlKoHZXdCFvotE6RzrQF7fc.exYtaG', 'errortest@test.com', '01752', 'Error test street', 'ET11TE', 'errortest');

-- -----
-- 
-- Table structure for table `livestreams`
-- 

CREATE TABLE `livestreams` (

```

```

`facility_id` int(255) NOT NULL,
`livestream_id` int(255) NOT NULL,
`livestream_title` varchar(100) NOT NULL,
`livestream_link` varchar(255) NOT NULL,
`livestream_notes` varchar(255) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 
-- Dumping data for table `livestreams`
-- 

INSERT INTO `livestreams` (`facility_id`, `livestream_id`, `livestream_title`,
`livestream_link`, `livestream_notes`) VALUES
(7, 31, 'Test A', 'ZZPPj83eqKw', ''),
(7, 32, 'Test B', 'UPCCj83zqKwq', ''),
(7, 35, 'Video demonstration', 'LRHmWbyVlOo', '');

-- -----
-- 
-- Table structure for table `members`
-- 

CREATE TABLE `members` (
`facility_id` int(255) NOT NULL,
`member_id` int(255) NOT NULL,
`member_username` varchar(25) NOT NULL,
`password` varchar(100) NOT NULL,
`email_address` varchar(50) NOT NULL,
`first_name` varchar(20) NOT NULL,
`last_name` varchar(20) NOT NULL,
`date_of_birth` date DEFAULT NULL,
`gender` varchar(15) DEFAULT NULL,
`status` varchar(25) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 
-- Dumping data for table `members`
-- 

INSERT INTO `members` (`facility_id`, `member_id`, `member_username`, `password`,
`email_address`, `first_name`, `last_name`, `date_of_birth`, `gender`, `status`)
VALUES
(1, 1, 'firstUser', 'password', 'first@test.com', 'First', 'User', '2020-11-06',
'male', 'pending'),
(1, 2, 'secondUser', 'password123', 'second@test.com', 'Second', 'User', '2020-11-06',
'female', 'active'),

```

```

(7, 3, 'thirdUser', 'password3', 'third@test.com', 'Third', 'User', NULL, NULL
, 'active'),
(7, 4, 'newUser', 'password', 'newuser@test.com', 'New', 'User', NULL, 'male',
'pending'),
(7, 6, 'fourthUser', '$2y$10$.FgsgffBVC/5rKHxrtWp10swlKoHZXdCFvotE6RzrQF7fc.ex
YtaG', 'fourthuser@test.com', 'Fourth', 'User', NULL, 'female', 'declined'),
(7, 7, 'fifthUser', '$2y$10$.FgsgffBVC/5rKHxrtWp10swlKoHZXdCFvotE6RzrQF7fc.exY
taG', 'fifthuser@test.com', 'Fifth', 'User', '2020-01-01', 'male', 'pending'),
(7, 8, 'sixthUser', '$2y$10$.FgsgffBVC/5rKHxrtWp10swlKoHZXdCFvotE6RzrQF7fc.exY
taG', 'sixthuser@test.com', 'Sixth', 'User', '2020-12-
24', 'female', 'inactive'),
(7, 9, 'seventhUser', '$2y$10$.FgsgffBVC/5rKHxrtWp10swlKoHZXdCFvotE6RzrQF7fc.e
xYtaG', 'seventhuser@test.com', 'Seventh', 'User', '2020-12-
23', 'male', 'active'),
(7, 10, 'eighthuser', '$2y$10$tjqe6zf7IVdXHZOr0VmCxu0WmhqzbER.adXZuCs4GZbHhYef
LcrPO', 'eighthuser@test.com', 'Eighth', 'User', '2021-02-
10', 'male', 'pending');

-- -----
-- 

-- 
-- Table structure for table `messages`


CREATE TABLE `messages` (
  `facility_id` int(255) NOT NULL,
  `message_id` int(255) NOT NULL,
  `message_title` varchar(100) NOT NULL,
  `urgency_level` varchar(20) NOT NULL,
  `message_notes` varchar(255) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 
-- Dumping data for table `messages`


INSERT INTO `messages` (`facility_id`, `message_id`, `message_title`, `urgency_level`, `message_notes`) VALUES
(1, 1, 'New message', 'low', 'Message\'s notes'),
(1, 2, 'Another message', 'urgent', 'Some more notes'),
(2, 3, 'Message from new facility', 'low', 'Notes?????'),
(7, 5, 'Low urgency message (1)', 'low', 'Testing low urgency.'),
(7, 6, 'Medium urgency message (2)', 'medium', 'Testing medium urgency.'),
(7, 7, 'Urgent message (3)', 'urgent', 'Testing high urgency message.'),
(7, 13, 'New message', 'low', 'This is a test');

-- -----

```

```

-- 
-- Table structure for table `status`
-- 

CREATE TABLE `status` (
  `member_id` int(255) NOT NULL,
  `member_status` varchar(20) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 
-- Dumping data for table `status`
-- 

INSERT INTO `status` (`member_id`, `member_status`) VALUES
(1, 'pending'),
(2, 'pending'),
(3, 'accepted');

-- -----
-- 

-- 
-- Table structure for table `training_times`
-- 

CREATE TABLE `training_times` (
  `member_id` int(255) NOT NULL,
  `nine_am` tinyint(1) DEFAULT NULL,
  `ten_am` tinyint(1) DEFAULT NULL,
  `eleven_am` tinyint(1) DEFAULT NULL,
  `twelve_pm` tinyint(1) DEFAULT NULL,
  `one_pm` tinyint(1) DEFAULT NULL,
  `two_pm` tinyint(1) DEFAULT NULL,
  `three_pm` tinyint(1) DEFAULT NULL,
  `four_pm` tinyint(1) DEFAULT NULL,
  `five_pm` tinyint(1) DEFAULT NULL,
  `six_pm` tinyint(1) DEFAULT NULL,
  `seven_pm` tinyint(1) DEFAULT NULL,
  `eight_pm` tinyint(1) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4;

-- 
-- Dumping data for table `training_times`
-- 

INSERT INTO `training_times` (`member_id`, `nine_am`, `ten_am`, `eleven_am`, `twelve_pm`, `one_pm`, `two_pm`, `three_pm`, `four_pm`, `five_pm`, `six_pm`, `seven_pm`, `eight_pm`) VALUES
(7, 1, 1, 1, 1, 1, 1, 0, 0, 0, 0, 0, 0),

```

```

(9, 0, 0, 0, 0, 0, 1, 1, 1, 1, 0, 0, 0);

-- 
-- Indexes for dumped tables
-- 

-- 
-- Indexes for table `facilities`
-- 

ALTER TABLE `facilities`
  ADD PRIMARY KEY (`facility_id`),
  ADD UNIQUE KEY `facility_username` (`facility_username`),
  ADD UNIQUE KEY `email_address` (`email_address`);

-- 
-- Indexes for table `livestreams`
-- 

ALTER TABLE `livestreams`
  ADD PRIMARY KEY (`livestream_id`),
  ADD KEY `facility_id` (`facility_id`);

-- 
-- Indexes for table `members`
-- 

ALTER TABLE `members`
  ADD PRIMARY KEY (`member_id`),
  ADD UNIQUE KEY `member_username` (`member_username`),
  ADD UNIQUE KEY `email_address` (`email_address`),
  ADD KEY `facility_id` (`facility_id`);

-- 
-- Indexes for table `messages`
-- 

ALTER TABLE `messages`
  ADD PRIMARY KEY (`message_id`),
  ADD KEY `facility_id` (`facility_id`);

-- 
-- Indexes for table `status`
-- 

ALTER TABLE `status`
  ADD PRIMARY KEY (`member_id`);

-- 
-- Indexes for table `training_times`
-- 

ALTER TABLE `training_times`
  ADD UNIQUE KEY `member_id` (`member_id`) USING BTREE;

```

```

-- 
-- AUTO_INCREMENT for dumped tables
-- 

-- 
-- AUTO_INCREMENT for table `facilities`
-- 

ALTER TABLE `facilities`
    MODIFY `facility_id` int(255) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=9;

-- 
-- AUTO_INCREMENT for table `livestreams`
-- 

ALTER TABLE `livestreams`
    MODIFY `livestream_id` int(255) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=36;

-- 
-- AUTO_INCREMENT for table `members`
-- 

ALTER TABLE `members`
    MODIFY `member_id` int(255) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=11;

-- 
-- AUTO_INCREMENT for table `messages`
-- 

ALTER TABLE `messages`
    MODIFY `message_id` int(255) NOT NULL AUTO_INCREMENT, AUTO_INCREMENT=14;

-- 
-- Constraints for dumped tables
-- 

-- 
-- Constraints for table `livestreams`
-- 

ALTER TABLE `livestreams`
    ADD CONSTRAINT `livestreams_ibfk_1` FOREIGN KEY (`facility_id`) REFERENCES `facilities`(`facility_id`);

-- 
-- Constraints for table `members`
-- 

ALTER TABLE `members`
    ADD CONSTRAINT `members_ibfk_1` FOREIGN KEY (`facility_id`) REFERENCES `facilities`(`facility_id`);

-- 

```

```

-- Constraints for table `messages`
--

ALTER TABLE `messages`
  ADD CONSTRAINT `messages_ibfk_1` FOREIGN KEY (`facility_id`) REFERENCES `facilities`(`facility_id`);

--

-- Constraints for table `status`
--

ALTER TABLE `status`
  ADD CONSTRAINT `status_ibfk_1` FOREIGN KEY (`member_id`) REFERENCES `members`(`member_id`);

--

-- Constraints for table `training_times`
--

ALTER TABLE `training_times`
  ADD CONSTRAINT `training_times_ibfk_1` FOREIGN KEY (`member_id`) REFERENCES `members`(`member_id`);

COMMIT;

/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
/*!40101 SET CHARACTER_SET_RESULTS=@OLD_CHARACTER_SET_RESULTS */;
/*!40101 SET COLLATION_CONNECTION=@OLD_COLLATION_CONNECTION */;
```

#### 14.3.2 Table Normalisation

<b>UNF</b>	<b>1NF</b>	<b>2NF</b>	<b>3NF</b>
<u>Facility_id</u>	<u>Facility_id</u>	<u>Facility_id</u>	<u>Facility_id</u>
Facility_name	Facility_name	Facility_name	Facility_name
Facility_username	Facility_username	Facility_username	Facility_username
Password	Password	Password	Password
Email_address	Email_address	Email_address	Email_address
Telephone	Telephone	Telephone	Telephone
Address	Address	Address	Address
Postcode	Postcode	Postcode	Postcode
Access_code	Access_code	Access_code	Access_code
(Message_id)			
Message_title	<u>Facility_id</u>	<u>Facility_id</u>	<u>Facility_id</u>
Message_urgency_level	<u>Message_id</u>	<u>Message_id</u>	<u>Message_id</u>
Message_notes)	Message_title	Message_title	Message_title
(Livestream_id	Message_urgency_level	Message_urgency_level	Message_urgency_level
Livestream_title	Message_notes	Message_notes	Message_notes
Livestream_link			
Livestream_notes)	<u>Facility_id</u>	<u>Facility_id</u>	<u>Facility_id</u>
Member_id	Livestream_id	Livestream_id	Livestream_id
Member_username	Livestream_title	Livestream_title	Livestream_title
Password	Livestream_link	Livestream_link	Livestream_link

First_name	Livestream_notes	Livestream_notes	Livestream_notes
Last_name			
Email_address	Facility_id	Facility_id	Facility_id
Date_of_birth	*Member_id	*Member_id	*Member_id
Gender	Member_username	Member_username	Member_username
Status	Password	Password	Password
Training_times	First_name	First_name	First_name
	Last_name	Last_name	Last_name
	Email_address	Email_address	Email_address
	Date_of_birth	Date_of_birth	Date_of_birth
	Gender	Gender	Gender
	Status	Status	Member_status
	Training_times	<u>Member_id</u>	<u>Member_id</u>
		Training_times	Nine_am
			Ten_am
			Eleven_am
			Twelve_pm
			One_pm
			Two_pm
			Three_pm
			Four_pm
			Five_pm
			Six_pm
			Seven_pm
			Eight_pm

## 14.4 ETA's Feedback

### 14.4.1 Video Demonstration

Inbox

**Polly Johns**

Wed, Feb 17, 6:53  
PM (2 days ago)

to me

Hi Reece,

So, feedback on your video for your new system SafeGaze.. It looks like exactly what our trampoline club needs to provide a secure login portal so parents and watch their children train during sessions. I love the security features so that we are in control of WHO has access and WHEN! One issue I was concerned with, was that if people left the club they would still be able to gain access, but SafeGaze makes it really easy to change people from Active to Inactive members. I also love that we control when the parents can see the live stream so they will only be watching their child's group rather than login anytime. The only negative of this is that we can't customise the times of SafeGaze to our class times which may cause some confusion for parents. For example, they may miss the last 10 minutes of a session if the class doesn't finish at O'clock. The package means that we can use our own existing cameras and add multiple views so all areas can be visible which is great from a

safeguarding perspective. The ease of the login means all members can be sent the same link to invite them to use Safe Gaze and they can create their own login. The extra function of sending them notifications with an urgency is bonus feature aswell. We can't wait to start using SafeGaze at ETA!

All the best,

Polly Buzz-Johns

CEO at Exeter Trampoline Centre



#### 14.4.2 April Meeting

##### General Comments

Love that it is really easy to sign up for.

I like the "Getting Started" page.

I don't think you need the username, it's "an extra layer you don't need".

##### Improvements

1. The PROBLEM is with the set access times per day. It creates a new ethical issue for participants training specific times on certain days, but different on other days...
2. The YouTube 'complications' should probably also be changed for the annoying parts...
3. Look into wireless camera connection for easier connectivity... Wireless CCTV camera system