## References

Importance of planning systems dev (DONE) (accessed 24th Mar 2021): <https://www.hyperext.com/software-development/the-importance-of-planning-when-it-comes-to-systems-development/>

Project planning books from COMP3005 (NOT DONE)

Importance of project management (DONE) (accessed 24th Mar 2021): <https://thedigitalprojectmanager.com/why-is-project-management-important/>

Explaining SDLC (DONE) (accessed 24th Mar 2021) (section 3.1 figure) (2021): <https://stackify.com/what-is-sdlc/>

{Potential} Explaining SDLC (DONE) (accessed 24th Mar 2021): <https://phoenixnap.com/blog/software-development-life-cycle>

What is agile (DONE) (accessed 25th Mar 2021): <https://www.visual-paradigm.com/scrum/what-is-agile-software-development/>

(Referenced(3.2)-quote) What is a sprint (DONE) (accessed 26th Mar 2021): <https://www.atlassian.com/agile/scrum/sprints>

(Referenced(3.5.1-quote)) What is version control (DONE) (accessed 26th Mar 2021): <https://www.perforce.com/blog/vcs/what-is-version-control>

(Referenced(3.5.1)-elaboration) What is version control (DONE) (accessed 26th Mar 2021): <https://www.atlassian.com/git/tutorials/what-is-version-control>

(Referenced(3.5.2)-quote) What is project backlog (DONE) (accessed 29th Mar 2021): <https://www.perforce.com/blog/hns/project-backlog-examples>

What is Microsoft Planner (DONE) (accessed 29th Mar 2021): <https://www.chorus.co/resources/news/what-is-office-365-planner-a-beginners-guide>

Why use Microsoft Teams (DONE) (accessed 30th Mar 2021): <https://pulse.microsoft.com/en/work-productivity-en/na/fa2-reasons-why-you-should-start-using-microsoft-teams-today/>

(Referenced(2)-quote) Ethical issues in software (DONE) (accessed 2nd Apr 2021): <https://www.srs.fs.usda.gov/pubs/VT_Publications/01t7.pdf>

<https://www.sciencedirect.com/science/article/pii/S0168169900001587>

Data protection and IT (DONE) (accessed 2nd Apr 2021): <https://www.lawdonut.co.uk/business/data-protection-and-it/data-protection-and-it-overview>

{Potential} GOV Data Protection Act (DONE) (accessed 2nd Apr 2021): <https://www.gov.uk/data-protection>

Children safeguarding (DONE) (accessed 3rd Apr 2021): <https://learning.nspcc.org.uk/safeguarding-child-protection>

Intellectual property (DONE) (accessed 8th Apr 2021): <https://freibrunlaw.com/intellectual-property-rights-software-protect/>

{Potential} Benefits of user testing (DONE) (accessed 9th Apr 2021): <https://www.gossinteractive.com/article/6699/The-importance-of-user-testing>

What are project requirements (DONE) (accessed 10th Apr 2021): <https://steelkiwi.com/blog/requirements-why-it-important/>

Benefits of dedicated mobile app (DONE) (accessed 13th May 2021): <https://www.digivate.com/blog/mobile/responsive-web-design-dedicated-mobile-site-or-mobile-app/>

## References before Harvard

Importance of planning systems dev (DONE) (accessed 24th Mar 2021): <https://www.hyperext.com/software-development/the-importance-of-planning-when-it-comes-to-systems-development/>

Project planning books from COMP3005 (NOT DONE)

Importance of project management (DONE) (accessed 24th Mar 2021): <https://thedigitalprojectmanager.com/why-is-project-management-important/>

Explaining SDLC (DONE) (accessed 24th Mar 2021) (section 3.1 figure) (2021): <https://stackify.com/what-is-sdlc/>

{Potential} Explaining SDLC (DONE) (accessed 24th Mar 2021): <https://phoenixnap.com/blog/software-development-life-cycle>

What is agile (DONE) (accessed 25th Mar 2021): <https://www.visual-paradigm.com/scrum/what-is-agile-software-development/>

(Referenced(3.2)-quote) What is a sprint (DONE) (accessed 26th Mar 2021): <https://www.atlassian.com/agile/scrum/sprints>

(Referenced(3.5.1-quote)) What is version control (DONE) (accessed 26th Mar 2021): <https://www.perforce.com/blog/vcs/what-is-version-control>

(Referenced(3.5.1)-elaboration) What is version control (DONE) (accessed 26th Mar 2021): <https://www.atlassian.com/git/tutorials/what-is-version-control>

(Referenced(3.5.2)-quote) What is project backlog (DONE) (accessed 29th Mar 2021): <https://www.perforce.com/blog/hns/project-backlog-examples>

What is Microsoft Planner (DONE) (accessed 29th Mar 2021): <https://www.chorus.co/resources/news/what-is-office-365-planner-a-beginners-guide>

Why use Microsoft Teams (DONE) (accessed 30th Mar 2021): <https://pulse.microsoft.com/en/work-productivity-en/na/fa2-reasons-why-you-should-start-using-microsoft-teams-today/>

(Referenced(2)-quote) Ethical issues in software (DONE) (accessed 2nd Apr 2021): <https://www.srs.fs.usda.gov/pubs/VT_Publications/01t7.pdf>

<https://www.sciencedirect.com/science/article/pii/S0168169900001587>

Data protection and IT (DONE) (accessed 2nd Apr 2021): <https://www.lawdonut.co.uk/business/data-protection-and-it/data-protection-and-it-overview>

{Potential} GOV Data Protection Act (DONE) (accessed 2nd Apr 2021): <https://www.gov.uk/data-protection>

Children safeguarding (DONE) (accessed 3rd Apr 2021): <https://learning.nspcc.org.uk/safeguarding-child-protection>

Intellectual property (DONE) (accessed 8th Apr 2021): <https://freibrunlaw.com/intellectual-property-rights-software-protect/>

{Potential} Benefits of user testing (DONE) (accessed 9th Apr 2021): <https://www.gossinteractive.com/article/6699/The-importance-of-user-testing>

What are project requirements (DONE) (accessed 10th Apr 2021): <https://steelkiwi.com/blog/requirements-why-it-important/>

Benefits of dedicated mobile app (DONE) (accessed 13th May 2021): <https://www.digivate.com/blog/mobile/responsive-web-design-dedicated-mobile-site-or-mobile-app/>

## References before alphabetize tool

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/>

Jeston, J. (2018). Business Process Management: Practical Guidelines to Successful Implementations. 4th edn. Oxon: Routledge.

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/>

Altvater, A. (2020). What Is SDLC? Understand the Software Development Life Cycle. Viewed 24th Mar 2021. <https://stackify.com/what-is-sdlc/>

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>

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/>

Rehkopf, M. (2020). What are sprints? Atlassian Agile Coach. Viewed 26th Mar 2021. <https://www.atlassian.com/agile/scrum/sprints>

Gehman, C. (2019). What Is Version Control? Perforce. Viewed 26th Mar 2021. <https://www.perforce.com/blog/vcs/what-is-version-control>

Atlassian. (2020). What is version control? Atlassian. Viewed 26th Mar 2021. <https://www.atlassian.com/git/tutorials/what-is-version-control>

Karlsson, J. (2020). What Is a Project Backlog + Examples? Perforce Software. Viewed 29th Mar 2021. <https://www.perforce.com/blog/hns/project-backlog-examples>

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>

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/>

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

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>

GOV. (2011). Data protection. GOV. Viewed 2nd Apr 2021. <https://www.gov.uk/data-protection>

NSPCC Learning. (2021). Safeguarding children and child protection. NSPCC Learning. Viewed 3rd Apr 2021. <https://learning.nspcc.org.uk/safeguarding-child-protection>

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/>

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>

Semusheva, O. (2016). Requirements. Why is it important? Steel Wiki. Viewed 10th Apr 2021. <https://steelkiwi.com/blog/requirements-why-it-important/>

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/>

## Things that can be done for the report

* Market research into other apps ( <https://www.socialtables.com/blog/event-technology/live-streaming-apps/> )
* Project initiation – Project Vision + Risk plan
* The different sprints and what they focus on
* Planning and preparation materials (design UI, use case (diagrams), functional requirements, gantt chart, objectives, use cases, aims and objectives, flow diagram)
* Getting a client (ETA) to test the app and get feedback on it (before and or after deliverable)
  + Why and how
  + Their feedback + changes to be made (possibly also in improvements section \/)
* Bi-weekly reports (may contain important notes)
* Development process document screenshots
* **Three different approaches to streaming** (most in important streaming notes document):
  + State security flaws with using YouTube for the streaming service (in important streaming notes document)
  + State why you used ‘this’ method for your product
  + + To Do list for process for facility to create the stream and publish it on SafeGaze
* Importance of private streaming (stated here: <https://www.dacast.com/blog/7-live-streaming-solutions-for-private-events/>)
* Gantt chart statement – livestreams section went on for longer due to complications
* Have a dedicated “Livestream” section for all the above ^ (in a sprint section?)
* Improvements to be made
  + Functional requirements that have not been met
  + Things stated by Polly
  + Other things stated in backlog
* **Feedback / improvements ^^^**
  + Client feedback email (document “ETA feedback”)
  + Backlog
  + TBC – client feedback after gyms reopen (w/c 12th but set up w/c 19th)
  + **Face to face sit-down meeting with Polly on Wed 21st Apr**
    - Useful because it raised awareness of other/specific ramifications that were initially unclear.
    - Information in bi-weekly report 13
    - Meeting notes in document “ETA feedback (Apr)”
* Version control (Project management) by using GitHub
* Creation of logo
* Getting started page – SafeGaze Documentation
* Working on project showcase materials
  + GIMP
  + Icons (<a target="\_blank" href="https://icons8.com/icons/set/blue--screen-of-death">Blue Screen of Death icon</a> icon by <a target="\_blank" href="https://icons8.com">Icons8</a>)
  + Video demonstration logo intro (<https://placeit.net/c/videos/stages/cool-intro-maker-with-a-bold-style-398?stateId=177ee94f8aa347a>)
  + Video demonstration typing sound (<https://freesound.org/people/GeorgeHopkins/sounds/537244/>)
  + Video demonstration background music (<https://freemusicarchive.org/genre/Ambient>) (Lightning Bug, Artist: Bio Unit)
* Elaboration on technologies (technologies + platforms) (coding platforms)
  + Visual studio code
  + PHPMyAdmin
  + XAMPP
* Technologies (coding + frameworks)
  + HTML
  + JavaScript
  + CSS
  + Bootstrap
* Showcase technologies
  + GIMP
  + Photoshop (logo)
  + Shotcut
  + Audacity
  + (Video and poster assets stated above)
* Project management technologies
  + Git + GitHub
  + Microsoft Office Planner
  + Microsoft Teams
  + Sprint planner – Gantt chart (EXCEL or Gantt software)
* Design technologies
  + Database diagrams
  + UI design
  + Poster design
* Beneficial with current circumstances of COVID-19 (written about it – “taken out” section)
* **Assignment brief report guidelines**
* Post-mortem review (Rico)
* Annotated bibliography?
* Functional requirements, non—functional requirements, user requirements (other name)
* **Minimal viable product in requirements**
* Documentation

## Notes

Messages – Message notifications have been stated (needs to be implemented prior to submission)

The term “members” – “Members” applies to everyone that is associated with the facility and requested/has an account for SafeGaze under facility’s name. Participants will apply to individual who engage in the facility’s activities. [ Member’s may apply to parents of participants ]

(Kimberly’s words regarding agile approach):

“ I would classify it as an agile environment. Your splits can be put into 4 stages but with the bi-weekly meetings and everything there was an opportunity to change course after chatting/reflection but if there wasn’t much need for it, that is OK. The structure is still more agile than waterfall. You have 4 main sprints, but you could, if needed, probably break those into smaller sprints. But you probably don’t need to since you have demonstrated some agile aspects.

Hope that makes sense. You don’t need “weekly sprints” to be agile. But even then with alternating blogs on the DLE and supervisor meetings (something weekly but alternating) you can show you have followed the agile structure as much as you need to. If you are worried that your schedule doesn’t follow the online definition 100%, don’t worry, you have adapted the concept to fit your product/style. That both makes sense and is a good thing to do. “

Critique approach with possible alternatives **(Rico post-mortem review)**

Adjusted livestream sprint due to requiring further research and implementation

Potentially add images of logos of software management technologies justification + software licensing

Not all functional requirements have been met – Potentially state how you did not achieve the non-functional requirement “notifications” due to the unexpected scenario with the difficulties when facing the livestream functionality (as it was not working)

Not applying some functional requirements – Some functional requirements have not been implemented and might not be able to be implemented at the moment but can be put into the “Improvements” section.

Potentially add to ethical issues section (article located at “Assessment notes & other docs”):

“Devices such as “nanny cams,” laptop cameras, modern closed-circuit video systems and other small, unobtrusive wireless recorders have given members of the general public an unprecedented ability to put others under surveillance without their explicit knowledge or consent.”

IMPORTANT improvement stated by Polly (Apr feedback) – Improvement to be made show database design diagram for implementing specific times per day

Gantt chart planning & gyms – It depended on the gyms being open, and it was not foreseen that the gyms would be closed for three months from the start of 2021. This severely halted the client side testing of SafeGaze.

## Taken out

[ 1.1 Background ] This mobile app would be specifically dedicated to the facility’s members, which in turn, would improve both the user interface and accessibility for smaller devices, such as mobile phones or tablets.

[ “Getting started” guide – can be useful ] SafeGaze is a web app that is directed towards the physical safety and cybersecurity of sports clubs and facilities. It focusses on providing a safe streaming platform for a facility’s members, and thus negates the risk of stalking or harassment of strangers and its members, especially when the participants primarily consist of children. By implementing this system, facilities will be able to create a secure livestreaming platform which allows their members to safely watch their children taking part in such activities.

[ Catalogue entry project description ] 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 would nullify the need for these viewing areas, as parents will be able to watch their children remotely via one or numerous livestreams.

[ 3.4.1 Version control ] Using version control software is a best practice for high performing software and DevOps teams. Version control also helps developers move faster and allows software teams to preserve efficiency and agility as the team scales to include more developers. (Atlassian)

[ 2.1 ] it is apparent that strict guidelines must be followed to protect any information that is stored relating to a system’s users.

[ 8.1 Sprint overview ] The different sprints are as follows:

1. Planning & Preparation
2. Facility focused
3. Member focused
4. Management focused
5. Livestream focused
6. Message & facility management focused
7. Testing
8. Finalisation

[ 9. ETA’s critique ] After “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”, even when not participating in the facility’s activities.

[ 4. Client side testing ] At end of “It also highlights” sentence: or possibly features that should be implemented in the future

Original statement - It also highlights what aspects must be prioritised, whether it relates to the functionality it offers, design or layout of the user interface, resolving logic or syntax errors.

[4. Client side testing ] After “capabilities”. This particular version of SafeGaze made use of the web hosting server provided by Plymouth University; following informative feedback from ETA, notes were added to the project backlog for beneficial adjustments. It also highlighted that YouTube’s livestreaming features were a much needed change due to the limitations it offers. This is elaborated further in section lorem[youtube-stream-change-improvements].

[ 3.1 SDLC ] After “improving quality of the product” The entirety of the project is split into six smaller phases, of which include:

* Analysis of requirements
* Planning
* Software, system, and architectural design
* Software development
* Testing
* Deployment

[ 2.4 Software licensing ] List explanations:

**Visual Studio Code:** A simple streamlined code editor created by Microsoft, used as the primary tool for the creation and amendment of SafeGaze’s source files.

**Bootstrap:** A popular ‘Cascade Style Sheets’ (CSS) framework used for the development of responsive websites and web-based applications.

**PhpMyAdmin:** A software application intended to handle the administration of MySQL databases. This was used for the creation of SafeGaze’s database prior to its release to a web hosted database server.

**XAMPP:** A PHP development tool used to simulate an Apache server. This was used for testing the styling, responsiveness, and behaviours of SafeGaze prior to its release to a hosted web server.

[ 8.2 Sprint 0 ] Before changes to paragraph: Moreover, the planning documents that were primarily text-based (e.g. functional requirements) were also replicated to the backlog for better organisation for the purpose of tracking and constant assessment of set requirements. Alongside this, the backlog was also populated with other useful information, such as notes, current tasks, and the different sprints/phases of the project.

[ 8.1 ] After “allocated time.” Moreover, the time required per task was deliberately overestimated to allow for some leeway if potential complications arose and would have halted development for a specific reason.

[ 3.5.1 ] After “occur.” Gehman, C (2019) gives this in more detail:

“With the right version control software, you can have a replica of your enterprise repository –– your single source of truth –– operating in another location. If something happens, you can immediately switch over to a replica of the master for uninterrupted availability.”

[ 3.5.2 ] After “comments.” These additions are extremely useful when a task may require additional information, or elaboration of certain aspects, and proved to be particularly beneficial to SafeGaze due to the complexity of this project.

[ 3.5.3 ] After “structure.” This specifically related to the tasks unique to each sprint, as they were identified separately under their designated groups. In the end, through the application of the additional variables, time management was amplified; key information was displayed within the backlog, as opposed to another document/application.

[ 3.2 ] After “requirements set.” At the end of each sprint, the quality of the completed tasks were examined and carefully considered for any adjustments that would prove beneficial. This in turn allowed for better control of the output per sprint.

[ 3.5.3 ] After “with the supervisor.” In addition, these channels were also used for queries, announcements, and general discussion regarding specific topic areas of the COMP3000 module. This included, but not limited to: design layout, deadlines & reporting, technical issues, and usability testing. Furthermore, Teams also featured video calling, resulting in bi-weekly meetings including grouped video chat sessions.

## 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 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. |

## Some Appendix Items (not to forget)

* **REFER TO RICO REPORT APPENDIX**
* Bi-weekly blog reports
* Backlog
* Database excel structure
* (Development process)
* Supervisor meeting minutes
* Gantt chart + project plan unrefined
* Documentation
* (How to complete each task with screenshots)
* (Video demonstration notes)
* Polly’s feedback email
* Database code
* Showcase materials
* Use cases (& functional requirements?)
* Risk plan

## Appendix

* Documentation
* Project management
  + Backlog
  + Risk plan
  + Bi-weekly reports
  + Bi-weekly meeting minutes
  + Gantt chart unrefined
  + Project plan unrefined
  + Use cases
* Database
  + Database SQL code
  + Database excel structure
* Polly’s feedback
  + Email (from video demonstration)
  + April meeting
* Assets (logo, showcase materials, etc)
* **Each page of SafeGaze**

## Areas to elaborate if word count too low

* 3.3: Add benefit “continuous oversight” at end (Importance of project management: <https://thedigitalprojectmanager.com/why-is-project-management-important/>)
* 3.4.1: Version control: replace Atlassian elaborates -> elaborate why version control was beneficial to SafeGaze
* 2.2: “Add comment from report notes (“Nanny cams” bit, taken from article)”
* 1.3: STATED BELOW

## Improvements that can be done to already completed sections

* 8.1: Allow Gantt chart to have time off
* 1.3: Add to aims & objectives. Refer to COMP3005 exercise 2.

## Areas to reduce if word count too high

* 2.0 Ethical issues introduction can merge into section 2.1
* 1.4 Furthermore…”, maybe taken out
* 3.1 “Optionally, a seventh phase…” maybe taken out
* 2.3 Remove software licensing list’s explanations
* 8.2 “Moreover, the planning documents” paragraph can be removed as it touches self-explanatory parts
* 4. Client side testing – reduce
* 6.6 Risk plan moved to appendix and replaced with figure alongside short explanation (a risk plan was created to assess…)
* 3.5.1/2 Remove/alter quotes
* Quite large
  + 3.2 Changing sentence “Numerous software…”
  + 3.3
  + 3.4
  + 3.5.3
* 3.5.2 Potentially remove “moreover, these additions…”
* 8.2
* 8.3
* 8.4
* 8.5
* 8.6
* 8.7 (amber sentence)
* 8.8
* 8.9
* 9

## Main sections for future work

* Development outline (how implementation was approached)
* Development phases
* Client side testing
  + Client user testing feedback
    - Initial communication
    - Video feedback in email
  + About getting it set up in ETA (April)
    - Feedback
* End project report – one section including the following
  + Summary
  + Discussion of results
  + Conclusion
* Post-mortem (basic) **(1)**
  + What went well?
  + What did not go well?
  + What would be performed differently?
  + Future enhancements
    - Polly’s things from Apr meeting (explained in 9)
    - YouTube streaming changes (explained in 8.6)
    - Requirements not met
    - Other things in backlog
* Post-mortem (evaluation approach) **(2)**
  + Management evaluation
  + Technologies evaluation
  + Development performance evaluation
  + Future work
* Post-mortem (evaluation approach) **(3)**
  + Objectives evaluation
  + Technologies evaluation
  + Developer performance evaluation
  + Project management evaluation
  + The future
* Post-mortem **(4)**
  + Technologies
  + Project management
  + Personal performance
  + User testing
  + Disability access
  + Overall
* Post-mortem **(5)**
  + Method of approach review
  + Development review
  + Best aspects
  + Future work
  + Developer performance review

## Tasks for project plan

### Sprint 0: Planning

* [2] Backlog
* [6] Project plan + Gantt chart
* [1] Project initiation
  + Project vision (+ market research)
  + Project objectives
  + Risk plan
* [7] System flowchart
* [5] Functional requirements and use cases (+ diagrams) (both users)
* [8] UI design
* [9] Database table design
* [3] Set up git repository, and login to web hosting (+ misc. docs: report + notes, development process, bi-weekly reports, minutes, etc.)
* [4] Research frameworks to use

### Sprint 1: Facility

* [1] Set up app’s initial pages
  + Index, dashboard, login/signup
  + Implement Bootstrap
  + (Organise dashboard with styling to work properly)
* [2] Create SQL statements for database
* [6] List member requests
* [7] List idle members
* [5] List active members
* (Plan methods for livestream broadcast videos/system)
* [4] Allow facility to log in
* [3] Allow facility to sign up

### Sprint 2: Member

* [5] Display member access times
* [4] Pending/inactive/declined users to have verify-member page to eliminate unauthorised access
* (Start planning report: different sections + notes)
* [3] Allow facility to accept/decline member signup requests
* [2] Allow member to log in
* [1] Allow member to sign up

### Sprint 3: Management

* (Display proper message for access times)
* [3] Allow facility to **view** a member’s access times (in manage member page)
* [4] Allow facility to **alter** a member’s access times (in manage member page)
* [2] Allow facility to change a member’s account status (in manage member page)
* [1] Allow facility to view all a member’s details in a separate page (manage member page)
* (^ Take the user (facility) to a separate page)

### Sprint 4: Livestream

* [1] Refine method of approach for a private livestream / Conduct research into best method of approach
* [6] Display popup message for members when they are active but not in their "access times" (including current time for when access and no access)
* [4] Display all available livestreams for **member** to view
* [3] Display all available livestreams for **facility** to view
* [2] Allow facility to create a new livestream
* [7] Allow facility to edit a livestream’s details
* [8] Allow facility to delete a livestream
* [5] Allow members to ONLY access a livestream within their access times

### Sprint 5: Messages

* (Refine styling of messages/notifications)
* [3] Allow member to view al messages + details
* [2] Allow facility to view all existing messages issued to members
* [1] Allow facility to issue a new message to their members

### Sprint 6: Testing

* [1] Create video demonstration for client
* [2] Inform client of development state + request feedback (from video demonstration + chat)
* [3] Note feedback/constructive criticism for future improvements
* [4] Create documentation for a facility on how to use SafeGaze

### Sprint 7: Refinement & release (setting it up (in Apr)) [potentially in new section]

* Adjust system to resolve potential errors/bugs
* Refine UI styling
* Create showcase materials
* Continue project report

## End Project Report

**Summary**

This project served as the (what it does)

**Requirements/Objectives**

Overall, this project has been a success as it meets the objectives and (%) requirements (stated in section requirements). It allows a facility’s members to watch/engage (elaborate). In the end X% requirements have been met

^ This is proved by implementing the system to the client’s facility. Although certain complications arose which meant SafeGaze could not have reached its potential if lockdown did not persist/delay interactivity/integration and as a result/resulting in feedback /a list of improvements that can be done for SafeGaze for the future.

However, some requirements had not been met, but these do not necessarily made a substantial impact on the application’s functionality (eg, profile management (notifications/change password) are features that were not implemented)

**Future changes**

It has been developed in the manner for future enhancements/improvements to be made

## Post-mortem

### What went well?

It was a success

### What did not go well?

Lockdown prevented things – communication with client could have been more informative to work around lockdown so it would be better/more refined for when gyms/leisure centres reopened

Better project management?

Research before development to understand best approach Comprehensive

### What could have been done/performed differently?

### Future work/enhancements / Improvements

[Use reports]

I will need to adjust it to the client’s needs for it to be fully operational.

**ETA’s is definitely endorsing it. From market research and feedback from different clubs and sports centres, it has become apparent that such a tool would be valued by them / a product that they would definitely use / implement**

When these changes having been made and fully operational ETA will use SafeGaze within the next month

## 10. Port-mortem

### 10.1 Project Management Evaluation

Due to the agile methodology, it meant that the entirety of the project flowed better, and each sprint was laid out with exactly what was expected (useful for the client as they would not understand some of the ‘wording’ used to explain the deliverables/requirements stated). Agile also meant that requirements could be altered due to potential obstacles that may present themselves mid development.

Overall, this project was managed properly, therefore allowing for (elaborate: why it was beneficial)

Whilst the development generally followed a smooth process, there were moments in which the sprints’ dates/times had to be amended to adapt to the situation. For example, the livestream sprint went on for longer due to the fact (didn’t originally work).

[2] While the development generally progressed smoothly, there were moments where sprint scopes and schedules had to be altered. For example, Sprint 4 and Sprint 5 being swapped, and the descoping of procedure returns and logical operators. This kind of flexibility would not have been possible with waterfall, or other such, methods.

The project’s method of approach of implementing/implementation of certain deliverables could have been researched better/in-depth to prevent time delays mid development. (needs to account for possibility of/potential roadblocks/obstacles that may hinder/prevent (the development) progress.

### 10.2 Objectives Evaluation

It was a success as it meets all the aims and objectives stated in section lorem(aims-and-objectives), however not all requirements/deliverables had been fully reached/realised but as already stated in section lorem(), they do not necessarily/significantly affect the functionality that is required to perform the project’s original objectives/original aim.

As a lesson to learn, the deliverables/requirements should be managed accordingly to make use of a minimal viable product. However, the deliverables in this project have each been given a priority, which was considered when with the order at which the tasks are attended to with each given sprint.

### 10.3 Technologies Evaluation

Went with what was safe and did not experiment with new technologies/tools/**frameworks** for programming in case they took too much time to get an understanding of and the possibility that after learning it, realising that it would not be accomplishable with such a/said resource. [2] Initially choosing Magnum as the graphics library was a mistake due to the learning curve associated with a new framework of this scale [refer to 2 and others]

Other technologies/tools significantly enhanced the project, for instance project management ones + ones for the project (photoshop for asset creation, gimp for showcase materials project poster & phpMyAdmin for the creation of sql code)

### 10.4 Development Performance Evaluation

Effort put into this project was in accordance with the university’s 400 hour (development) guideline and proved to be suitable for a project of such a large scale

[2] Averaging the duration of the project, the effort invested was in line with the 400 hour guideline and suitable for the project’s scale, adjusting throughout as needed. The initial six weeks of the project saw a planned decline in the expected work effort due to the author’s personal commitments. This decrease was planned to be made up for in the second half of the project, when more time would be made available to the author.

The second half of the project saw major disruption due to the global health outbreak, and as such minor planned features had to be descoped, as stated in the Development chapter. In all, aside from the resulting solution, the author has gained invaluable knowledge and experience through the undertaking of this project and is proud of the work achieved.

Due to the demand of other university modules, work declined at certain times.

However, the work conducted in this project was consistent and there were no large gaps, and thus prevented (when you get back and forget)

[3] During development of the application the developer attempted to spend between 30 and 40 hours per week on all aspects of the project. In most instances this was achievable however, there were some weeks where other commitments or unforeseen circumstances would get in the way. To compensate, extra time would usually be spent in the following weeks in order to recover the timeline which resulted in the recommended amount of project time being used.

The developer did face difficulties due to their unfamiliarity with dealing with a project of this scale, which did lead to rushing into some tasks and underestimating the challenges that would be faced, this ultimately led to delays in other tasks which should have potentially been done before. However, a lot has been learned from this project and the knowledge can be used in future endeavours.

### 10.5 Future work

This project showed/proved served as a proof of concept and proved/shows that it would be highly beneficial (use 2 for elaboration)

As already stated in project review (further development)

Polly’s things from Apr meeting (explained in 9)

YouTube streaming changes (explained in 8.6)

Requirements not met

Mobile app