Project Proposal

Contents

Introduction

This project is being made to the specifications and requests for our client Rory Hopcraft in line with the COMP2003 module ‘Computing Group Project’. Here, we tasked with developing a multiplayer, educational video game. This software will then be presented to students in an educational environment to aid in their learning.

After some proposals and feedback with our client, we have decided to develop an online multiplayer game – under the current name of ‘Capture the Flag’. This will see a group of students split into two teams, battling each other to earn points to earn a victory over the opposition. Here, students will have to answer series of questions and mini games. These questions will be generated via an AI model with the option for refinement by an educator if necessary.

Justification

Our client requires this project to aid him in his role as an outreach coach. Our software aims to engage his students and provide an education outside of traditional teaching methods.

Whilst other solutions exist to fulfil a role similar to ours – no existing games seem to fulfil the niche of adaptability. Due to our proposed solution of using an AI model to generate a set of questions and answers via a prompt inputted by the educator – we can offer a host of teaching subjects that would not otherwise be available.

Upon completion, the game should be adaptable and reusable over different teaching groups and educational ability.

Requirements

Detailed here are some of the requirements that our solution must be to satisfy the scope:

- Accessible – The webpage must be intuitive, following SOLID design principles. The system must not require any information or training prior to use. If explanation is required, this will be outlined within the website itself.
- Available – The service should be open for use when needed by our client. This means that the service is readily available without the need for a developer to act. This will be mandatory after the software rights are signed to the client.
- Secure – The webpage must have its users be protected at all times. This is especially important in the use case of minors. Any potential threat of an attack could be detrimental to the schools networks.
- Private – Stored data on the students should be minimal to none. There is no need for any personal data on individuals to be stored outside of the use of usernames and passwords – of which – use of personal information in them should be warned. Data stored about users account (such as playtime) should be kept only to the user.
- Managed – Data should be clear and organised. Data structures should be robust and well documented. Data should be validated and not cause any errors.

- Smooth – The service should be able to operate smoothly and with minimal wait times. This includes initial access and runtime.
- Flexible – The service should be made for many use cases. The service must be used for a variety of classroom topics. It should also have option to be up scaled.

Software Specifications and Implementation

In schools, provided hardware for the students can range heavily. This means that students may be using very dated machines or newer, more powerful pieces of equipment. As such, we must be careful as to the technological demands our solution requires.

As an absolute baseline, we should expect their devices to be operating on limited processing power such as Intel Core i3s and have at least 4GB of RAM.

Since we are hosting the servers to the game on a web server, this will take some of the processing out of hands of the students' devices. However, the main issue we will have to face will be with the mini-game sections (as these will require the most amount of processing). Due to the small scale of these games, this shouldn't be much of an issue, providing the games are made optimally.

This implementation will of this project includes:

- HTML - To build the main framework of the website. This will host most screens and functions like question builders, student question time and hosting information
- Unity – To run the mini games. Unity offers many useful features that can aid making mini games like its physics engine. This will be embedded in the HTML structure.
- Web service – To host the website. This will be held locally within the University of Plymouth domain for testing with the end goal of hosting via AWS for implementation
- Database structure – To store information. Users should be able to store statistics about their playtime and accuracy that can be accessed at any time.

Timeline

Following *(Table 1)* is a timeline showing the main goals we will achieve following the 26th of January. These goals are not a full representation of our sprint plans, merely outlining the main goals we will achieve.

Date	Main goals/Expectation
02/02/26	Have basic working build implemented for one user
16/02/26	Implement AI questioning and first fully developed game
02/03/26	Host on university servers. Have multiple demo users available
16/03/26	Implement host functionality (score views and player management)
30/03/26	Implement custom users via account creation
20/04/26	Track users play data including scores
PROCEEDING	Refinement + functional user testing

Table 1 – Timeline

Following is a visual representation of this in the form of a Gantt chart. This has been made via GPT-5.1 with the input from *Table 1*.

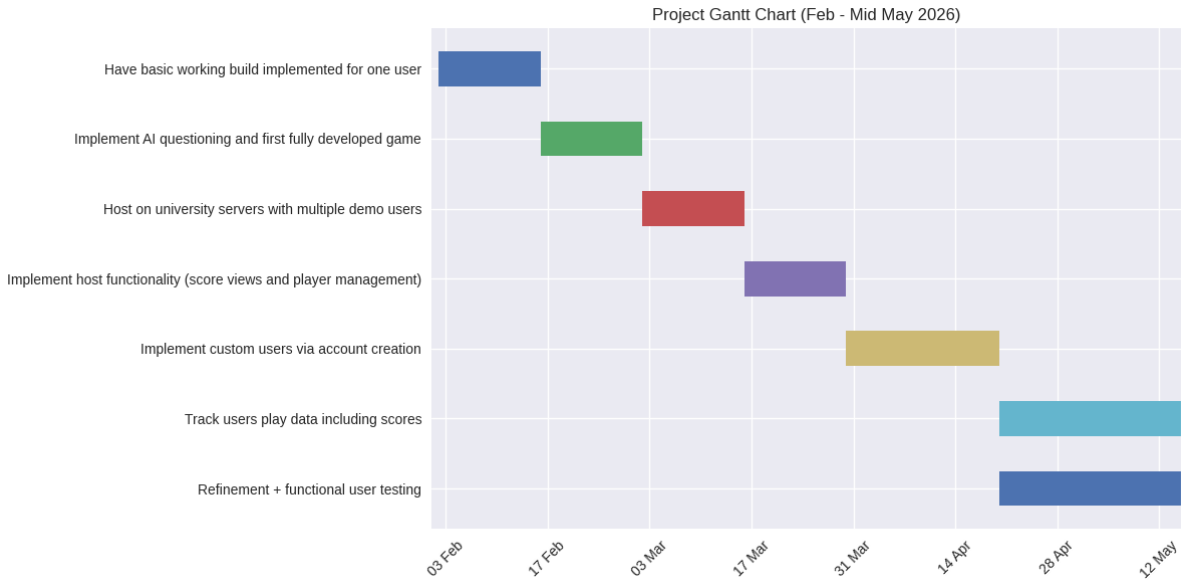


Table 2 – Project Gantt chart

Proposed Solution

Capture the Flag is an educational game designed to help students develop digital-safety and problem-solving skills through interactive challenges. The experience is competitive, fast-paced, and fully sandboxed so that all activities remain secure and appropriate for a classroom environment (of ages 14 to 16).

Two teams compete to earn the highest score by collecting “flags” — short puzzles — across a series of rounds. Players must think quickly, coordinate with teammates, and choose their routes through the challenges strategically.

The updated design introduces mini-games and AI-generated questions, adding variety, unpredictability, and opportunities for teams to gain advantages or sabotage the opposition.

- Supports any number of players, from small groups to full-class sessions
- Players are assigned to Team Red, Team Blue, or additional teams depending on class size
- Teachers can run controlled game rooms or allow open lobbies
- Optional player roles add minor advantages (Analyst, Cryptographer, Engineer, Wildcard) without restricting gameplay

This flexibility allows the game to scale for lessons, workshops, clubs, or large events.

The game will include a variety of challenges, but the exact organisation of these challenges is not finalised. Elements still under development include:

- How challenges will be grouped or categorised
- Whether there will be difficulty tiers
- How scoring or rewards will scale with challenge complexity
- How challenges will appear within each round

These details will be confirmed later once the design team completes testing and refinement.

After the students have completed their set of questions – a short, optional mini-game will appear which allows players to:

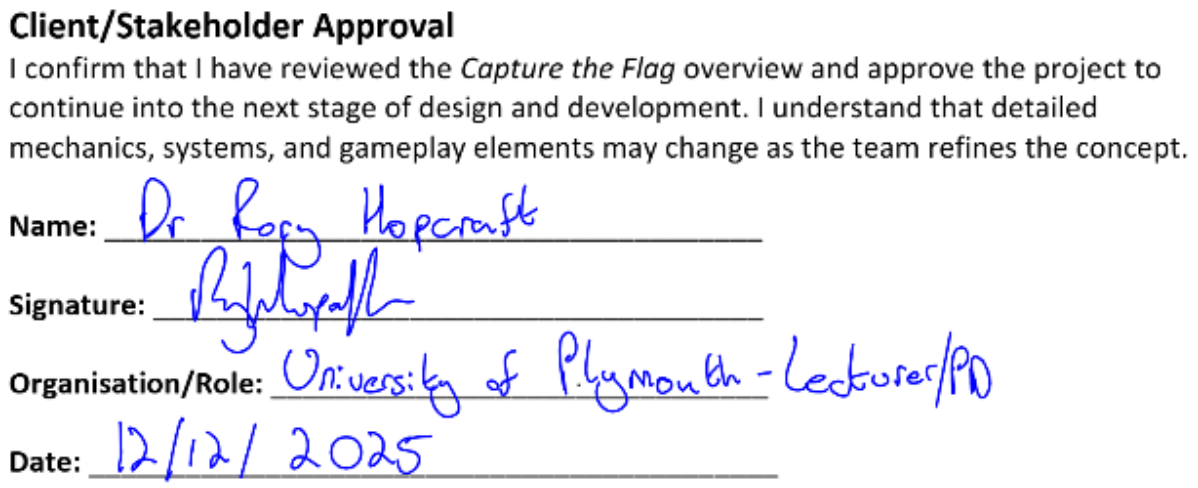
- Earn extra points
- Gain temporary team buffs
- Apply penalties to the opposing team

To increase classroom engagement and spark some excitement - at random or teacher-triggered intervals, the game generates custom challenges using an AI engine. These can:

- Reward the answering team with bonuses
- Send a difficult scenario to opponents, slowing them down

These events add unpredictability and keep matches exciting even with large groups.

This solution has been out decided final concept for how the service will act out. However, this is subject to slight changes should we need to change and features during future development. This has been reviewed and signed off by our client Rory Hopcraft. He has stated that he is content with said implementation and current development. Declaration and signature follow.



Client/Stakeholder Approval

I confirm that I have reviewed the *Capture the Flag* overview and approve the project to continue into the next stage of design and development. I understand that detailed mechanics, systems, and gameplay elements may change as the team refines the concept.

Name: Dr Rory Hopcraft

Signature:

Organisation/Role: University of Plymouth - Lecturer/PD

Date: 12/12/2025

Image 1 – Client proposal sign off

Following *(Image 2)* is a template as to the typical user story. This is from the perspective of a student who would like to play the game alongside their classmates.

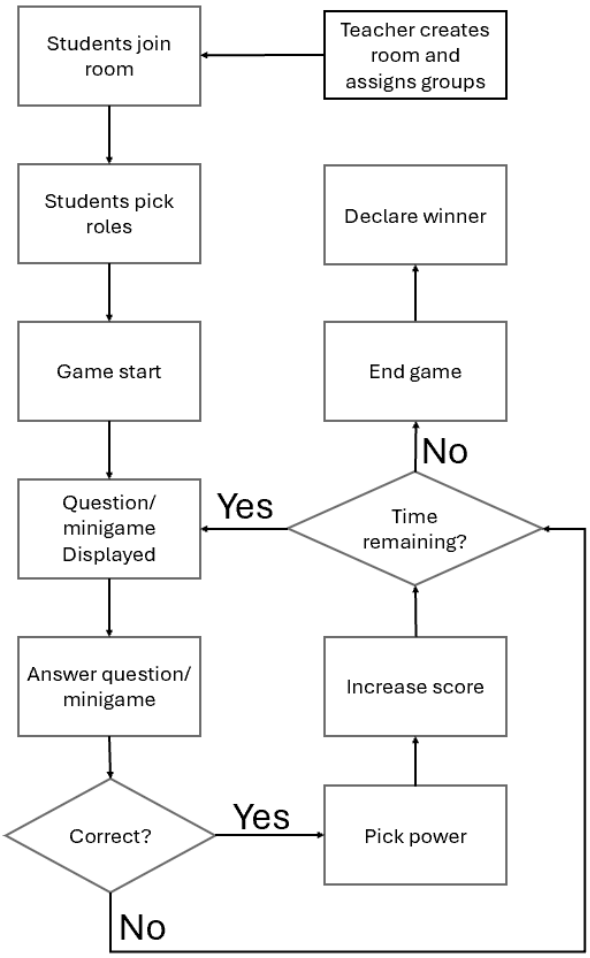


Image 2 – Game flowchart

Work Breakdown Structure

Our work will be split between our four group members to maintain an easy balance. Tasks are delegated based on preference and expertise. Following are each main section of work along with what needs to be completed. These are in no exact order.

UI Design

1. Create wireframes. This is the best start to designing a well presented and accessible webpage. These include:
   - Home
   - Sign in
   - Profile
   - Lobby (teacher and students)
   - Role selection
   - Quiz interface
   - Power up selection
   - Mini game template
   - Scoreboard
   - Quiz question generation.
2. Concept art. Implement these off the design of the wireframes. These designs should remain consistent throughout to make the service look professional.
3. Create assets. By this point, a usable version of the website should have been developed. Create a list of assets that are needed via this demo and implement. Re-using assets were possible is good practice and saves storage. Document each asset with its purpose and where/when it is used.
4. Test application and improve on feel and feedback where necessary. Sound effects and visuals help with this.

Data management and retrieval

1. Identify what attributes are needed for implementation.
2. Create a stable database structure with these attributes, including relationships between tables and data. Form UML diagrams on this structure – including field types and expected data.
3. Form a data structure using the language of choice on the specified domain. Populate with test data.
4. Implement * Complete once website development has commenced *. Establish connection from the website to the database. Create scripts to fetch test data from database and output to user.
5. Create methods of CRUD operations via the website to the database. Stress test these to the point of expected failure to ensure data validation.
6. Access CRUD operations where necessary in future development.

Structure development

1. Identify all needed interfaces from design specification and what they will be used for.
2. Create basic HTML structure with directories to all required blank pages
3. Start implementing content to the design of developed wireframes
4. Import unity projects into project. Embed these into designated page.

5. Implement full design work to standard using provided designs – using assets if needed

Games development

1. Brainstorm a list of small scale, repayable games that last approximately ten seconds. Highlight promising ideas and order via priority.
2. Create blank unity project. Create base objects and scripts (like player gravity, movement etc)
3. Develop game. Implement logic for the game along with a specified win condition (either time based or a goal to reach).
4. Import assets. Use these to make the game look professional and on theme with the rest of the website.
5. Refinement. Tweak values such as speed or points for a clear win condition to make the game feel responsive and to the same difficulty as other games.
6. Repeat steps 2-5 until sufficient variety has been created.

Network Hosting

1. For development, request space to be held for our project under the University of Plymouth domain. Confirm subdomain name
2. Create environment. Implement required hosting environment.
3. Ensure proper security compliance. Includes authentication and security. Define who has developer access
4. Deploy files onto server.
5. * When needed for user testing deployment * Deploy the completed code onto AWS or other web hosting services. This needs to happen – otherwise, devices outside of the University of Plymouth network will need to connect via a VPN to access website.

Communication

For the duration of the project, our group have been gathering for three types of meetings: client, sprints and internal discussions/work sessions. All of our meeting dates and times are recorded on our shared GitHub repository.

Our client meetings (and hence – work on the project) had started later than anticipated due to external reasons. Once these reasons had passed, we have been able to hold valuable and consistent meetings. These are scheduled every two weeks on the same time and day to maintain consistency. However, we have not exclusively been in contact via meetings as we have also had open communication via emails. We ensure we remain professional in our meetings and have a structured plan in place prior to the meeting to present our current work to the client and gather their feedback. We can take this feedback and adapt our schedule and workload. This is also used as a time to ensure that our output is up to the clients standards and is of a suitable quality.

We hold group meetings twice a week providing everyone is available. In these meetings we can openly discuss our visions, difficulties and progress. We try to get shared work done during these times and disband to do our workload of the sprint individually. It is here that we will raise any big issues we have with our project and we attempt to resolve these issues together. If we are unable to, it then escalates to asking for advice during our sprint meetings with our lecturers.

Finally, we have a shared group chat. This allows us to schedule future meetings easily along with asking any questions when we otherwise would not be able to in a meeting. It is also an effective method of communication when we are physically unable to otherwise meet.

Budget

We currently do not have any costs planned for our project. Due to this reason, we do not have any budget in form for the development of this project. Normally, a budget would be set in place with the factors of time, wages and payout upon completion of the project. However, as this is a educational experience, money is not a contributing factor.

With this in mind, we seek out alternative solutions to what would otherwise cost money. As an example of this, we will be tools such as free editing software (and AI when needed) to make designs and assets. Another example being AWS (if implemented) as they provide a free trial for a limited number of use cases before they start charging a premium.

Closure and Evaluation

Upon completion of the project – we will go through two processes; these being: final client signoff/ transfer of copyright and a post-implementation review.

The mentioned client transfer rights is what we are ultimately concerned with. This is the process in which we hand over our completed work to our client for their own free use. At this stage, we will have completed the solution to a high standard and to the client's own image and standards. Only then will we have completed the task sufficiently.

Whilst our post-implementation review is not mandatory for this project specifically, it will help aid us in our future academic and professional product development. Here outlines if the project is sufficient and what we can learn from the developmental process. It also includes how the product is operating after a longer-term testing period. This can help for future improvement.

Appendices

All supporting documents for this report are held within our GitHub repository or Trello board. Includes (but not limited to): client sign-offs, meeting minutes, progress and processes.