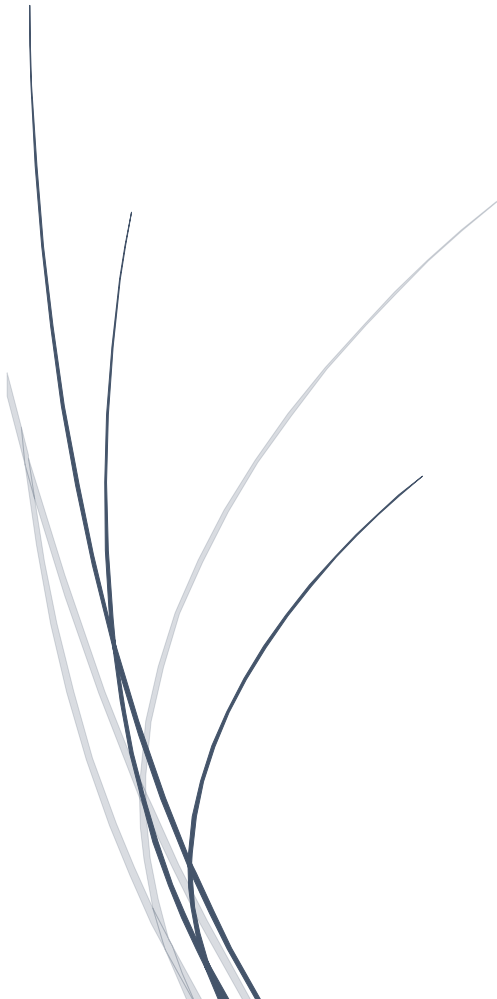




Research plan

TLC Quantum



Maike, Lucas, Esther, Genelle, Gabriella
FONTYS

Opportunity

The Quantum Talent and Learning Centers (TLC) Eindhoven is focused on advancing education in quantum tech, information, and sensing. The goal of Quantum TLC is to promote and ensure education at various levels, including WO, HBO, MBO, secondary, and primary schools, as well as those seeking a career change.

To help with their goal, we have been asked to make a software solution to make an online platform to host Quantum Chess Tournaments.

Quantum Talent and Learning Centers (TLC) Eindhoven has created a game called Quantum Chess, which is a variant of chess with dynamic effects of quantum mechanics. Right now, they can only play a game on one computer, making it so that users need to be physically together with one computer. The task that the software developers need to solve is to offer a software solution where users can use an online platform where they can join tournament-hosted games held by Institutes. This system is used for engaging and collaborative competition that spans all four Quantum Talent and Learning Centers (TLCs) in the Netherlands. A user can log in to the system, sign up for a tournament and play online hosted games. The focus will be on a user-friendly design, participants can sign up and smoothly take part. This end goal is to have a working solution with a great approach to future working software developers for this project.

Research questions

Main question:

“How can an external quantum chess game be integrated into a scalable software platform with a large player base that can participate in tournaments while maintaining security, reliability, and privacy?”

Sub questions:

Methods:

1. What technologies and frameworks will best fit the quantum chess tournament system while looking at performance, scalability, and security?
 - a. What back-end framework best fits our project?
 - b. What building framework for the front end best fits our project?
 - i. What requirements does the project need to fulfill?
 - ii. How scalable does the project need to be?
 - iii. How big is the community and support of the framework?
 - iv. What browsers and devices need to be supported?
 - v. How flexible does the framework need to be?
 - vi. How frequently is the framework updated or is it at risk of becoming outdated?
 - c. What database framework is a satisfactory solution for the tournament system?
 - d. What architecture for the chess tournament system fits the needs of the non-functional requirements?

Methods:

- **Literature study:** Getting knowledge of the different available systems.
- **Community research:** look for communities to get their perspective on different kinds of software.
- **Comparison Chart:** Creating a comparison chart to evaluate various plugins based on factors such as cost and effort.

2. What are the specific needs and preferences of users interested in participating in a quantum chess tournament?

- a. Are there any specific prioritizations to the functionality of these people?
- b. How can we clarify and justify this in an elegant way to the stakeholders?

Methods:

- **User Requirements Exploration:** Gaining comprehensive insights into user needs and preferences for the proposed solution.
- **Focus Group:** Collecting diverse perspectives and insights on the issue at hand.
- **Interviews:** Conducting interviews with stakeholders to delve deeper into their opinions and requirements.
- **Stakeholder Analysis:** Identifying and addressing the needs of relevant stakeholders.
- **Prototyping:** Developing prototypes to materialize conceptual ideas.
- **Brainstorming:** Generating innovative ideas through collaborative brainstorming sessions. Also, in the HackIT (week 1), having sessions with each other to produce ideas.

3. What are the technical requirements for the existing quantum chess game to communicate with the chess tournament system?
- What kind of software and tools are needed for communicating with the system while looking at performance?
 - Are there any specific APIs that can be used to aid with communication to the quantum chess game, while having the consideration on vendor lock-in and performance?
 - How do existing “e-sports” systems handle real-time updates on their platform?

Methods:

- **Literature study:** Getting to know what the embedding options are and how they work.
- **Available product analysis:** Look if any open source projects implemented this kind of technique.
- **SWOT:** analysis to have a better overview of what can be done and what to look out for.
- **Code review:** Once versions are deployed on the embedding part, it is important to check the created code for certain security vulnerabilities and optimize performance.
- **Task analysis:** How is this made possible to integrate it?
 - **IT architecture sketching:** Based on the information and flow, make a diagram together to understand it properly.

4. What are the legal and regulatory requirements for hosting online gaming tournaments, with the consideration of GDPR?

- What are the main data protection rules/requirements?
- How do we apply the measurements of GDPR in our application?

Methods:

- **Literature study:** Looking for the regulations of the GDPR. What is it? How to apply it?
- **Community research:** Get to know the ways other people have implemented these GDPR regulations.
- **Available product analysis:** Looking for existing products that apply regulations and look for solutions that can be applied to this project as well.
- **Problem Analysis:** Thoroughly understanding the problem landscape. So why GDPR?

5. What are the best practices when it comes to the security of the website and the zero-tolerance cheat policy in online gaming tournaments?

- Which security measurements are considered when it comes to online gaming platforms?
- How can cheating be detected and prevented effectively?

Methods:

- **Literature study:** getting to know what the best options are to securing this application, and what the best software solutions can be against cheating in a game.
- **Best good and bad practices:** Looking at secure system solutions.
- **Comparison Chart:** Creating a comparison chart to evaluate various plugins based on factors such as cost and effort.

6. What are the main user interfaces and experiences expected when it comes to online hosted tournaments?
- What design elements in an online gaming platform contribute to a user-friendly application?
 - How can we improve the experience when it comes to using the application on different browsers?
 - What are the existing online tournament platforms, and how do these platforms operate?
 - How do these platforms approach UI/UX?
 - What are the main strengths and weaknesses of these platforms?

Methods:

- **Product review:** Throughout the project, having enough meetings with each other, and discussing the looks with the stakeholders.
- **Usability testing:** Other users can investigate the product and give feedback. With this test, we can assess certain design choices and fix issues before the actual product goes live.
- **Community research:** look at what people find a nice UI/UX regarding a tournament system.
- **Literature study:** Look at the best design principles for creating a clear UI/UX.
- **Problem Analysis:** Thoroughly understanding the problem landscape.

7. How can we make sure that hosting and infrastructure are performing stable enough during a tournament?
- What hosting solutions can oversee a minimum of fifty-plus users at a concurrent time, with the option to automatically scale when needed?
 - How can we maintain the performance and limit potential bottlenecks?

Methods:

- **Code review:** Look at if the structure of the hosting is set up correctly. The system is separated with the correct specifications.
- **Literature study:** Looking at the best solutions to performing and having a stable application.
- **Model validation (ML):** check if the system can oversee the expected traffic.
- **Tinkering:** Engaging in trial-and-error coding to identify optimal solutions.

8. What metrics are available to validate the product that will be used by Quantum TLC Eindhoven to ensure its quality?
- Are there any criteria that need to be met to say the system's satisfaction, keeping in mind any standard requirements?
 - What testing and quality measurements can be considered to ensure the quality of our product?

Methods:

- **Product Review:** Giving a demo and reviews to the stakeholders every sprint, to get continuous feedback from them. This gives a good insight into what requirements need to be met.
- **Unit test:** verify that the logic inside the code is working as expected. Make sure the code is up to quality and prevents unwanted behavior.
- **Code review:** When a person is done with his functionality, another person in the group will review the code to find bugs, coding standards, and improvements.
- **System tests:** having a document with different test flows of functionalities to assess a case with the expected outcome.
- **Peer Review:** Collaboratively reviewing code using GIT to detect and rectify minor errors that may slip through unnoticed.

Deliverables

As mentioned for in the project, (Quantum, 2024, Project Plan, §3.1), there are specific deliverables for the project itself. Other than those documents we have the following:

- Process report, tracking a log on who did what on the research.
- Research Report: research question with the applied methods on how they have been approached and came to a conclusion/result. Also the results, a conclusion, and a reflection on what we have learned and can improve on.

Time estimation

To give an overview of when a sub-question is supposed to be ‘done’, here are the sprints in which they are expected to be finished. Note that we specify multiple sprints for a question since there is always improvement or more feedback to be implemented when showing the solution.

Sprint 0-1: Sub-Questions 1

Sprint 2-3: Sub-Questions 2, 3, and 6

Sprint 3-5: Sub-Questions 4, 7, 5, 8