Research Plan

Name: Tony Jiang

Semester: 6

Class: RB04

Problems and opportunity

The problem is that I currently have no experience developing in Go, and I am unsure how to design my scalable web-based music guessing game. However, this will give me an opportunity to learn about the language and focus on developing a scalable solution.

Research questions and methodology

Main question:

How can I design a scalable web-based music guessing game using Golang within a microservice architecture?

Sub-questions:

1. How should microservices be structured in Golang for optimal scalability and maintainability as the user demand increase?
2. How should the security of the music guessing game be validated?
3. How can CI/CD pipelines be set up for deploying a microservice-based game using Golang?
4. What are the best practices for deploying Golang microservices using serverless cloud functions?
5. What tools can be used to monitor the scalability of microservices?
6. How should data be managed across distributed microservices to ensure consistency, reliability, and scalability?

Methodology

Sub-question number	Methodology	Explanation	Result
1	Best good and bad practice – Design pattern research	This is to explore the best practices and common pitfalls in creating a well-structured	This aims to create a microservice architecture.

		microservice in Golang, with a focus on maintainability and scalability. It also includes research on design patterns to effectively structure microservices in Golang.	
2	Literature study - Security test	This is to study security vulnerabilities and how to validate them. The security testing process involves identifying vulnerabilities, prioritizing them, and assessing the associated risks in my project.	This sub-question aims to create the OWASP Top 10 document, which outlines the vulnerabilities, prioritizes them, and determines which security risks are critical.
3	Literature study - Community research - Prototyping	This study focuses on setting up a CI/CD pipeline for a microservice using Golang, reviewing community research, and creating a prototype for one microservice to understand the process before implementing it for the rest of the services.	This is to create a DevOps document detailing how the CI/CD pipeline is set up for the microservices and to implement the CI/CD process in my repository.
4	Best good and bad practice – Prototyping	This study aims to identify best and worst practices for deploying Golang microservices using serverless cloud functions. Additionally, it includes creating a prototype deployment of one microservice to gain an understanding of how it works.	This is to create a DevOps document on deployment and to understand how to deploy a serverless function on the cloud for scaling purposes.

5	Literature study – Community research – Non-functional test	This study focuses on exploring tools for monitoring the scalability of microservices, including those recommended by the community. Additionally, it involves conducting a non-functional test using one of these tools to assess the scalability of the microservice.	This sub-question aims to validate the scalability of the non-functional requirements.
6	Literature study - Best good and bad practice	Study how to manage distributed data in microservices and explore best and worst practices for managing distributed data in this context.	This sub-question encourages you to consider which services require a database within the microservice architecture.

Estimated time

The estimated completion time for the research would be around week 17.

Deliverables

- Research document
- Technical design document
- Prototype product
- OWASP top 10 document
- DevOps document