Design, implement and evaluate a solution to the chosen problem, including the use of suitable frameworks, methodologies and practices.

Effectively document your work and to communicate the design, implementation, evaluation and conclusions, including motivation, justification and analysis of the approach adopted and the project outcomes.

设计、实施和评估所选问题的解决方案，包括使用合适的框架、方法和实践。

有效地记录您的工作并传达设计， 实施、评价和结论，包括所采用方法和项目成果的动机、理由和分析。

Abstract – maximum one page (max. 1/2 page preferred) that very briefly describes what

your project is about. No technical detail, and as long as you get the key points across,

shorter is almost always better.

2. Introduction – this is perhaps the most important section, which you should spend sufficient

time on re-writing until you are really happy with it. After reading this section alone, a

marker should know what your project is about and what to look out for in the rest of the

thesis. In this section you do not go into too much technical detail except what is really

essential for understanding your project. You do state the problem you are trying to solve

and explain what you have done to solve it. In the introduction, you also state your claims to

quality, and ideally explain how you are going to provide evidence for them.

.介绍 - 这也许是最重要的部分，你应该花足够的时间重写，直到你真正满意为止。单独阅读本节后，标记应该知道你的项目是关于什么的，以及在论文的其余部分要注意什么。在本节中，除了对于理解您的项目真正至关重要的内容之外，您不会过多地介绍技术细节。你确实陈述了你试图解决的问题，并解释了你为解决它做了什么。在引言中，您还陈述了您对

质量的主张，并且理想情况下解释了您将如何为他们提供证据。

22222222222222222

Motivation

Under the COVID-19 background, cities around the world are brought under lockdown frequently, most citizens are told to work from home except circumstances that are extremely necessary, outdoor activities or in-person social activities are restricted, which results in closures of shops \cite{Iacobucci2020Covid19UL}. In the meantime, people still need to show up on video conferences to push forward with their works, including online interviews where video filters are not advised to use. And social patterns out of working time also shifting to online platforms.

Online livings may impact people’s mental health by social media exposure \cite{Gao2020MentalHP}, but it doesn’t affect people’s basic physiological state: the growth of hair is not an exception. In order to keep head clean and a good online appearance, haircuts are always required. Thus, a web app making appointment of in private barber service plays an important role in communicating clients and barbers as barbershops are restricted.

Moreover, even ones are likely to shave hair by themselves, there are still two benefits of in person barber service:

1. Specialized haircut services like perm and hair dye are still needed for whom wants to show up online delicately.

2. In person barber service could help people getting rid of online virtual times and help barbers keeping confident and practiced, opposite to virtual activities like online teaching \cite{Haverback2020MiddleLT}.

1.2 Aims and Objectives

Aims of this project are:

1. Develop a web backend application that communicates with UberForHaircut IOS frontend application (which is developed by my MSc CS classmate Wayne).

2. UberForHaircut App should allow barbers to register their barbershops and upload barber services and allow clients to explore book haircut appointments with barbers through it.

This is done by achieving the following objectives: ➔ Researching and identifying key tasks and responsibilities likely to cause issues between house mates. ➔ Providing flat sharers with a single system to manage their communal tasks and responsibilities. ➔ Having multiple households use the application and release an update of the application integrating user feedback. On estimation, the project will cover 70% type I (implementation) and 30% type II (investigatory). By identifying key tasks and responsibilities likely to cause issues between house mates, a application can be created that 1) automates these tasks or, when automation is not a possibility, 2) distributes them among members of the house in a fair way. In the latter scenario, the project will explore creative and original approaches to motivate members to complete their tasks and keep track of progress. 1.3

Deliverable

A prototype web app is designed and built. In terms of functionality, the application

meets the needs of users

3. Background – think of this as the state of the world before you started your project. In a

research project this is an extremely important section as it places your work in context of

existing literature and shows your awareness of all relevant previous work. In a software

development project, the background section is less crucial, you can use it to introduce the

problem you are trying to solve in more detail, mention technologies that you will use, or

compare existing attempts to solve your problem (and explain why the problem is not fully

solved yet)\

3.背景 - 将此视为您开始项目之前的世界状态。在

研究项目中，这是一个非常重要的部分，因为它将您的工作置于

现有文献的背景下，并表明您对所有相关先前工作的了解。在软件

开发项目中，背景部分不那么重要，您可以使用它来更详细地介绍您要解决的

问题，提及您将使用的技术，或

比较解决您的问题的现有尝试（并解释为什么问题尚未完全解决）\

333333

\section{Traditional architecture}

Traditional applications generally adopt monolithic software architecture during technology selection,

and the monolithic architecture indicates that completion of code writing, compilation, deployment and testing are all in the scope of a single project, where all services are running in a single process.

The characteristic of the monolithic architecture is that all the functionalities (including the representation layer, business logic layer and data resource layer) of the application system is contained inside the same project, and after completing the development of all the functional logic, it is finally compiled and packaged together, and deployed on a server 1351 to provide external services.

With the rising of functionality requirements caused by the expanding business scenario, volume of that project is getting larger and larger, bringing on the growth of development team, and consequently produces uncountable problems and matters:

\section{Drawbacks of traditional architecture}

\\\\

1. Poor development quality: The large scale of code repository and deep coupling of modules make it difficult for newcomers to understand existing code, leading to misuse of existing code during new module development and finally resulting in a continuous decline in code quality.

\\\\

2. Not conducive for continuous delivery: Continuous delivery is a software engineering technique that promises the output process of software products being completed in a short cycle term \cite{Chen2015ContinuousDH}. Monolithic architecture depends on a large code repository, its compiling, deploying and testing take a lot of time, therefore is not friendly to developers to conduct rapid iteration.

\\\\

3. Poor scalability: When the overall access traffic of a system is relatively flat but one particular interface of a single module’s traffic is high, the monolithic architecture system can only be simply deployed to multiple instances to tackle unexpected burst access traffic, which is a waste of server resources and increase of cost.

\\\\

4. Difficulty in technology selection and replacement: Once technology selection of a monolithic architecture system is decided, it is difficult to update technology selection based on existing code library when some modules are intended to switch to an advance technology.

\\\\

In view of these problems, I conducted requirement analysis and technical research on this topic, and know that the server-side system based on the monolithic architecture cannot meet requirements of scalability, agile development or high code quality.

\\\\

Currently the most popular solution among internet industry is a distributed application architecture called microservices architecture\cite{Dragoni2017MicroservicesYT}. Microservices can effectively reduce the coupling intensity between business modules (services) and boost development efficiency of developers. Also, server-side microservices application can increase the resources of specific business modules according to business needs like high concurrency access support, which in other words means high availability.

\\\\

\section{Trend in backend applications}

As the world is entering into mobile Internet era, applications have stepped from offline environment to world wild web, from desktops to laptops and mobiles. Applications are spreading everywhere to every scenario and device, so the web backend applications are facing more frequent changes of demand and a rapid growth of user volume feature of agility, flexibility and scalability of web backend application system development is becoming more and more necessary. As a result, developers urgently need one type of application system architecture which is more efficient and faster in developing and deploying web applications, especially backend applications.

\section{Microservices architecture}

Microservices came into people’s view after Dr. Peter Rodgers proposed Micro-Web-Service at the Web Services Edge conference in 2005, while Juval Lowy was holding a similar preceding idea: Turning categories into granular services as Microsoft's next-stage software architecture. The core idea is to make services being composed by Unix-like pipes, and complex services are covered by simple URIs to expose interfaces, any service, any granularity can be exposed to external environment \cite{granularservices}. The term microservice was first introduced at a software architecture seminar in May 2011 near Venice and officially named microservices in 2012. In 2014, Martin Fowler and James Lewis proposed the concrete concept of microservices and defined it as a micro service consists of a single application and maintaining its own process and lightweight handling. The service is designed according to business functionality requirements and deployed automatically, communicating with other services by HTTP.

\\\\

Microservices is not only an abstract concept, being a new style of software architecture \cite{ThonesJ.Microservices[J].IEEEsoftware,2015,32(1):113-116.},it plays an important role in software industry and receiving more and more attentions with the evolution of cloud computing, Docker container virtualization and DevOps technology\cite{辛园园,钮俊,谢志军等.微服务体系结构实现框架综述[J].计算机工程与应用,2018,54(19):16-23.}.

\\\\

Famous internet companies like Amazon, Netflix, Uber, Alibaba had implemented microservices architecture internally and made success in replacing existing traditional architecture one after another \cite{邓杰文,曹彩凤.微服务若干关键问题研究[J].五邑大学学报(自然科学版),2016,30(2):49-54.}. Meanwhile, open source project Apache Thrift \cite{维基百科编者.Thrift[G/OL].https://zh.wikipedia.org/w/index.php?title=Thrift&oldid=44788943.} and SpringCloud project basing on Netflix microservices open source modules are very prevalent and highly acclaimed among developers.

\\\\

**Body** – here you present everything necessary to understand your project. For example, the

body could contain a “user manual” of your software with screenshots to evidence your

software’s features. You might draw the marker’s attention to design decisions or patterns

used, explain your development process, or show the results of early evaluations and the

conclusions you drew from them for later development. The important point here is to

suitably highlight, that is draw the reader’s attention to, points that you think are

particularly important when marking your work – for example, what you consider to be the

high-quality aspects of your software. As always, back up claims with evidence

身体 - 在这里，你展示了理解你的项目所需的一切。例如，正文可以包含软件的"用户手册"，其中包含屏幕截图，以证明软件的功能。您可以提请标记者注意使用的设计决策或模式

，解释您的开发过程，或显示早期评估的结果以及您从中得出的结论以供以后开发。这里重要的一点是适当地突出，即引起读者的注意，你认为在标记你的工作时特别重要的点 - 例如，你认为你的软件的高质量方面。与往常一样，用证据支持索赔

20

\section{Development methodology}

As the previous chapter mentioned, microservices architecture is always implemented under DevOps methodology. Because the implementation of microservices will inevitably split the original application into dozens, then the compilation, packaging, and deployment of each split microservice will be several times the original workload, if automation tools are not used.

The implementation of microservices will inevitably involve collaboration between multiple microservices, so unit testing, regression testing, and performance testing of microservice functions will become more complex, and if automated tools are not used, the workload is large and the complexity is high, and it is difficult to measure.

The implementation of microservices will inevitably upgrade multiple microservices with different frameworks, so the basic environment on which microservices are deployed will be extremely complex and cumbersome, if you do not use automation tools.

The implementation of microservices will inevitably involve the collaborative development of multiple teams, then the management of microservice requirements, the management of the project will be extremely arduous, if you do not use advanced project management tools.

The implementation of microservices will inevitably update the application frequently, then code compilation, version control, and code quality will not be guaranteed, if mature tools are not used.

In view of the above problems, the implementation of microservices must have a tool chain such as requirements management, code version management, quality management, build management, test management, deployment management, environment management, etc., in addition, it also requires collaboration between the development department and the operation and maintenance department, so DevOps is a sufficient and necessary condition for the implementation of microservices. DevOps is about a set of practices, which combines software development (Dev) and IT operations (Ops). It is aiming at making the application development life cycle shorter and providing continuous delivery with high code quality \cite{http://radar.oreilly.com/2012/06/what-is-devops.html}.

\\\\

DevOps is often been described as a more collaborative and efficient relationship between the development team and the operation team, by which it means that it is primitively intended to be a style of working that crosses multiple departments’ responsibilities. Involved people would use a range of different tools called toolchains when practicing this sort of working methodology \cite{Gartner Market Trends: DevOps – Not a Market, but Tool-Centric Philosophy That supports a Continuous Delivery Value Chain (Report). Gartner. 18 February 2015.}. These toolchains are expected to be used in one or more of the following categories\cite{https://en.wikipedia.org/wiki/DevOps} representing key points during the development and deployment process:

\\\\

1. Coding – code development, code review, source code management, code merging.

\\\\

2. Building – continuous integration, build status.

\\\\

3. Testing – continuous testing.

\\\\

4. Packaging – artifact repository, application pre-deployment staging.

\\\\

5. Releasing – change management, release approvals, release automation.

\\\\

6. Configuring – infrastructure configuration and management, infrastructure as code tools.

\\\\

7. Monitoring – applications performance monitoring, end-user experience.

\\\\

In consideration of the tight relationship between microservices and DevOps, it is indeed useful to practice the workflow during microservices development, and I had considered carrying it out.

\\\\

However, establishing a such sophisticated development workflow system is not something easy. Every tool belonging to the toolchains needed to be carefully selected to make the whole workflow smoother and resultful, inappropriate tools within a toolchain may:

\\\\

Leave potential bugs and vulnerabilities inside an intermediate work that are not easy to detect and finally been brought along with delivered product \cite{Bass2015DevOpsA}.

\\\\

Halt the productivity as the tool may bring about superfluous works and distract developers from major tasks \cite{{Leite2020ASO}.

\\\\

Cause cascading mistakes due to incompatible evaluation criteria and generate unexpected consequences that are scarcely withdrawable \cite{Pal2021SoftwareTU}.

\\\\

\\\\

In spite of the hardness in toolchain selecting, DevOps is always been adopted by big companies and large-scale business projects because the cost of toolchain selection is not comparable to the cost of management regarding large projects and communication between a number of departments and employees \cite{Daz2021WhyAM}.

\\\\

In the case of this project, there are only two developers requiring cooperation and we lack software industry working experience. On the basis of our technology accumulation and the time limitation, we decided not to establish such a long-term framework. I technology learning as our primary target.

\\\\

Nevertheless, a reliable and pragmatic development methodology is still required. In view of the fact that DevOps is complementary with Agile Development which may be of more feasibility on small projects, and microservices architecture's module splitting feature is inherently suitable to iterate more frequently \cite{Tsilionis2021DeterminingTB}, I ultimately choose to process my development in line with agile methodology in contrast to the waterfall development methodology.

Agile development is another methodology which takes the evolution of user requirements as its core connotation, and adopts an iterative, step-by-step approach to software development.

\section{Requirement analysis}

As far as this project goes under agile methodology, every cycle of agile loop starts with one or a set of similar requirements, which could also be translated to what functionalities the users exactly need. Those requirements are like lighthouses navigating the moving direction of each loop of development, and they are just the final harbor where boats of development loop stop. To keep the development processes in a good working flow and prevent from getting lost, the list of requirements should never be empty before a development loop starts. For this reason, it is necessary to conduct requirement analysis, which translates raw requirements from users to structured document which determines what functions the software needs to implement and what work to complete, and so the target list is filled, a new cycle of agile loop is ready to start.

\\\\

For microservices architecture applications, there are generally three different kinds of requirements:

\\

Basic module functional requirements

Functional requirements

Non-functional requirements

\\

As this project is doing with microservices architecture, requirement analysis is also separated into those three parts.

\\\\

\subsection{Basic module functional requirement analysis}

All software systems ultimately serve users, some serve users by communicating directly to people, but there are still some others take responsibility by supporting or acting as platforms or intermedia. They are often regarded as the microservices basic module.

\\\\

The microservices basic module is something provides the underlying support of the microservices architecture for the business logic modules. It could provide specific solutions on building large-scale distributed application systems, including service discovery, service gateways, service communications and other issues. Following is the requirements analysis about functions of the basic system in this project.

\\\\

\subsubsection{service discovery}

\\\\

Microservice architecture advocates the concept of lightweight service architecture, a microservice module may need to be cloned, destroyed, relocated at any time when running, and it is no wonder an inefficient solution to manually maintain and manage service addresses that changes frequently by the developer, so a dynamic service discovery mechanism is needed. The service provider firstly registers the address information of its service instance to the service registry, which take charge of managing service instance addresses and providing the heartbeat check mechanism. As a service caller, the service consumer obtains the list of available service instance addresses from the registry, then selects one of the service instances for service invocation according to a particular algorithm.

\\\\

In this project, service discovery between business logic modules and client-faced modules, or that between modules, requires a mature and reliable service registry center that can automatically obtain the address information of all microservice instances, detect the status of service instances all the time, also actively remove information about microservice instances which are temporarily unable to provide services from the service registration list, as to ensure that service consumers is sending requests to available service providers.

\subsubsection{service gateway}

The microservice system usually runs multiple service providers at the same time, a complete system should be able to unify the authentication, management and forwarding of requests send by external clients. The intermediary that plays the role in a microservices system is the service gateway. The service gateway is mainly located at the edge of the whole system and serves as a unified entry point for external service requests.

\\\\

In this project, when a client sends a service request, the service gateway needs to complete functions including identity verification, traffic control, route management, request forwarding and so on. When forwarding requests, if there are multiple service instances providing the same service in the system, as the middle layer that coordinates the client and the microservice, the service gateway needs to forward multiple requests to different service instances which run the same program according to a specific load balancing algorithm to avoid the situation that a single service instance is overloaded.

\subsubsection{Service communication}

Microservice system is a typical distributed system, and each microservice instance is deployed on different sections and nodes according to requirements. The communication between microservice instances is carried out through the network, so the basic system of this project needs to choose a suitable communication mechanism to make data exchange efficient and stable.

\\\\

According to the message communication mechanism, the communication between microservices is divided into synchronous communication and asynchronous communication. For synchronous communication mode, the client sends a communication request and then been waiting for a response, in the industry there are mainly two protocols adopted, including HTTP and Thrift. This model is simple and easy to use, and convenient for testing, but microservices acting as consumers will be blocked while waiting for response, which will affect the overall performance of the system to a certain extent.

\\

For asynchronous communication mode, consumers do not need to stand alone and wait for the response, it could continue to work after sending data and so the system performance is effectively improved. But using the asynchronous mode increases the difficulty of development and testing and adds the complexity of system deployment, operation and maintenance.

\\\\

In this project, the synchronous communication mode is chosen as there is not a particular service that needs to work intensively and the response would always be quickly received.

\subsection{Functional requirement analysis}

Functional requirements, which in here means the general application functionality requirement, determine the specific business-logic oriented functionalities that the application should accomplish while it is actually running and online, in other words, it means all actions that the user can achieve and get response through the system. Indeed, the functional requirements almost or even all come from user needs.

\\\\

As long as functional requirements are almost coming from users, it is an excellent way to distinguish functionality boundaries and abstract concrete requirements by analyze different roles of users who would directly communicate with the application. Because applications are always been responding users’ actions and the role is a proper level to abstract and differentiate user actions and their motivation.

\\\\

In this project three roles are being discussed. All functional requirements within the project’s scope are derived by a combination of these three types of application users, which I would like to name the combination as “ABC” (Administrator, Barber, Client). The following paragraphs would show why these three types were set to be target user roles and what functional requirements each role has.

\\\\

Firstly, the Client:

\\\\

It is a little wired to start with ‘C’, but as everyone knows there won’t exist service if there is no client, everything starts with a client’s needs, every application starts with either people’s laziness or other objective reasons. So, requirements of the role as client are the foremost to be analyzed.

\\\\

With the context of this project’s background world (lockdown and social distance control), here the client role should be clarified as: people who needs a barber service, but they cannot go to a barbershop for it because the shop is closed, they may need a basic haircut, or some specialized barber services like perm and hair dye.

Their first need is to explore whether there are available barbershops online, and whether

\\\\

Firstly, the Administrator:

\\

Actually, the frontend application development plan does not include providing an administrator application management version or interface, but an administrator user is always essential to manage and analyze data through background application’s functionalities.

\\\\

For non-technical background staffs who are operating business based on the application, they are not expected to all be skilled in sorting and picking data they want from databases by SQL languages, they should always focus on operating business rather than operating massive and raw data in database. Moreover, though all staffs are assumed to be masters of SQL or they have tools to overcome the SQL barrier, they are more like a part of developers and are assigned underlying responsibility to maintain the database, accidents would happen as the database is not only secured by specialized manager. It is possible to assign different authorities to staffs, but when the group grows to a certain scale, staffs would hold various of authorizations to access different regions of data, in this context, authorization control and functionality management would be intertwined, integrating the administrator role into existing system would be more feasible comparing to developing another system.

\\\\

Once the administrator role is decided as a target user role, things are becoming easier. Currently project only consider one kind of ‘super administrator’ like ‘su’ in Linux. As ‘super’ indicates, the administrator could do anything they need on existing user data.

\\\\

The administrator would firstly demand to query appointments made on the application to analyze the business operation status and make decisions on functionalities through the data.

\\\\

Next, in some special situations they need to get user’s information like username and other fields not violating the user’s privacy to help making application updating plans. If the user approved, administrators would like to know some other personal data like age or sex which may be of great support in user experience improvement.

\subsection{Non-Functional requirement analysis}

Non-functional requirement refers to software products to determine the operation of the system according to some quality attributes, while deficiencies refer to a specific achievable demand. As an e-commerce platform, the performance of the self-service micro-mall service terminal greatly affects the user experience, and this paper will analyze the non-functional requirements from the aspects of security, reliability, and response speed. 1) Security: Security is reflected in two aspects, first, the user's identity authentication, micro mall users  enter the self-service micro mall rely on WeChat authorization, so the identity credentials of the micro mall users are granted by WeChat   of the token, the self-service micro mall needs to save the WeChat Openid identification user, and the operator needs  rely on the account password to log in to the background management of the self-service micro mall micro mall. Second, role authorization, with  households can only view their own orders and information, weChat business can only view their own store orders and information,  can not exceed the authority to view the information of others, this paper through WeChat's Openid unique identification of users;  Chapter 3 System Requirements Analysis 2) Reliability: Reliability is the probability or time that the system can operate normally at a certain inspection time. Self-service micro-mall is an enterprise-level software, which needs to provide users with 24-hour uninterrupted service, can quickly restore or forward requests to another same service when a service fails, and needs to avoid dragging down downstream dependent services when a service fails;

\section{System design}

After the methodology choosing and requirement analysis, the applications system designing part should begin.

\section{Technology selection}

To build a microservices architecture application system from the ground up without any framework or external library is something seeming impossible. And the first step to design an application system should be choosing a set of proper framework which would help building up an overall structure quickly.

\subsection{Service design}

Microservices architecture requires the decomposition of all business functionalities belonging to a single application system into multiple subsystems that run independently, these subsystems all are limited inside a clear service boundary and provide simple and specialized services to external in the form of a set of loosely coupled applications.

\\\\

The scheme of service splitting is not fixed, it needs to take specific business functionality needs into consider. The fundamentals of service splitting \cite{[ChrisRich^dson.MicroservicesPatterns:WithexamplesinJava[M].SuffolkCounty,NewYork:ManningPublications,2018.] }are as follows:

\\

1. The service splitting must maintain the stability of the whole architecture.

\\

2. Each microservice should be cohesive, and a set of strongly related functions should be implemented within the microservice.

\\

3. Each microservice must follow the CommonClosurePrinciple, which means that the content being modified at a time is tightly packaged together, ensuring that a single modification affects only one microservice.

\\

4. Each microservice must be loosely coupled, the service encapsulates a specific implementation, and the modification of that implementation should not affect the invocator which is the consumer service side.

\\

5. The granular size of the microservice is generally guaranteed to be developed and maintained by a team of 6 to 10 people.

\\

6. Each microservice team must be autonomous, with minimal need to collaborate with other teams while developing and deploying their own services.

\section{Architecture}

\subsection{The resource layer}

The resource layer is at the bottom of the architecture, which receives data from the upper side, in this project’s case, there are two kinds of implements inside the data layer.

\\\\

One is the open-source database MySQL, it receives data from the data access object layer of the three microservices, and store persistent data into computer storage.

\\

The second is the open source in-memory database Redis, which performs very well in high-speed data reading and writing. For this feature, Redis communicates with Gateway to accomplish the traffic control functionality by storing and providing tokens under the algorithm of Token Bucket Algorithm.

\subsection{The functional microservice layer}

Each microservice is a Spring Boot application, separated into three layers: controller layer, service layer and data access object layer. The microservices are in charge of handling request from client using its business logic. They can

\section{Setting up the environment}

In an application product development life cycle, the construction of development environment should never be skipped, it is always the primary foundation of software development coding stage. Only the development environment is well set up, can code development process be efficient and controllable. A good development environment would must bring great convenience for later stages of development work.

\\\\

In order to make future development work processes better, I had done following works to establish the development environment in this project:

\\\\

1. Installing IDE (Integrated Development Environment):

\\\\

As coding stage of this project is prospected to all be done on my Windows laptop, using an integrated development environment application would be very helpful when coding, compiling, debugging and analyzing rather than using an text code editor and frequently switch to terminals to run and test the application.

\\\\

For JAVA application developing world, Intellji IDEA is a leading and maybe the most powerful java application development application. It is primitively designed for java, with a serious of development supporting features from editing assistance to advanced support for popular frameworks like Spring and Spring Boot. As the technology selection section had decided, Spring and Spring Boot are main frameworks for this project’s microservices, so finally I chose and install Intellji IDEA for window x64 on my computer.

\\\\

2. Installing JDK and Maven:

\\\\

Java applications runs on the java virtual machine, and JDK (java Development Kit) contains the JRE (Java Runtime Environment) as well as other modules, one of which is essential for java application development: the java source code compiler, javac.

\\\\

As stated before, open JDK 8 and Maven 3 is chosen to support this project, so I downloaded it from the official websites and installed them onto this project’s environment directory.

\\\\

Maven is a java-based project management tool to simplify the build processes. In order to achieve its goal, Maven has the following kinds of features:

\\\\

Making the build process easy

\\\\

Providing a uniform build system

\\\\

Providing quality project information

\\\\

Encouraging better development practices

And the result is a tool that can now be used for building and managing any Java-based project. We hope that we have created something that will make the day-to-day work of Java developers easier and generally help with the comprehension of any Java-based project.

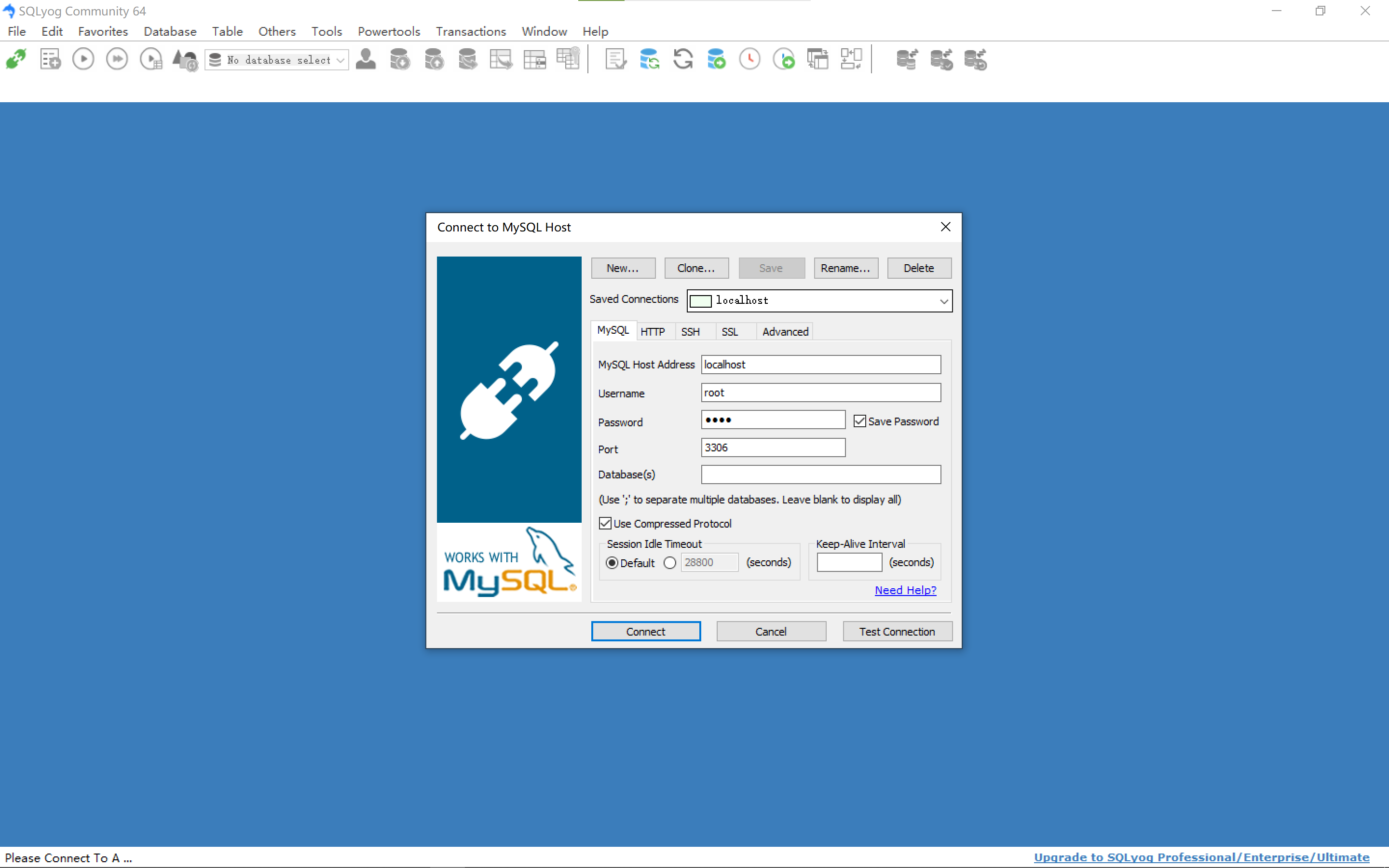
In this project, there are several microservice sub-projects, each depends on various of dependencies to accomplish the functionality, while some dependencies also rely on a few of others, there relationship is almost like a tree. To ensure an overall control of versions of dependencies, prevent unexcepted errors caused by version conflicts between dependency modules or by circular dependence

3. Installing MySql and SQLyog

\\\\

To make the development more convenient and quicker to manage database, after installing MySQL server application on the computer, I installed another database management tool, SQLyog. SQLyog is a free graphical tool which can effectively manage the database by operating the graphic interface, also quickly run SQL script files. It could just be regarded as a MySQL graphic client, transferring user’s button click activities to query language sentences.

\\\\



Likewise,

55555555

**Evaluation** – this is the section where you provide evidence for your claims to quality.

Typically, this will include some form of study with members of the target user group, ideally

including a proper statistical evaluation. You might want to start this section by repeating

your claims, then going through them one by one and providing evidence, or referring back

to evidence in previous sections.

6. **Conclusion** – (you can merge this section with evaluation if you want) here you can be a bit

more informal again. End your project by summarising again what problem you set out to

solve and how well you solved it, talk about what the challenges were during your project

and what you would do differently if you started it again, or what you would do next if you

had another few months to work on it.

– The thesis makes clear who the software is for (target users) and what problem(s)

it solves for them.

– The problem that the software claims to solve is real and does not seem contrived

to justify the project.

– The software does actually solve the problem it claims to solve, or at least makes

some steps in this direction.

A project that is essentially a clone of some existing software but has no ‘unique

selling point’ would normally not get a higher mark than a pass.

Usefulness, Maturity, Technical difficulty, Code quality, User experience

Under the COVID situation, cities around the world are brought under lockdown frequently, and there is a great amount of people work from home. Social activities are more held online, thus, demands of calling taxi is reducing significantly.

However, changing of social patterns and decline of the high street don’t affect people’s physiological state: growing hair is not an exception.

The project makes it possible to call a barber just like calling a uber taxi. Users could get rid of going to barber shops which would close under lockdown on their own or the risk of being infected by passers-by and the even worse situation-being infected by customers inside the barber shop due to not wearing masks while taking a haircut.

The intended user group including: 1. Customers: most people under the pandemic especially who work from home and having lots of video meetings. 2. Barbers: Whose shop is forced to close under lockdown or number of customers daily coming in is much less than it during normal days without pandemic. Key objectives of this project including: 1. Setting up a web backend of booking system and data structures for this app. 2. Communicating with other systems (Front-end, Payment, message...) through suitable APIs. 3. Receive a booking and start processing an order. 4. Assigning an order and waiting for confirmation from barbers then activate the order. 5. Canceling an activate order under excepting situations. 6. Reviewing activate orders and booking history. 7. Dealing conflict under high concurrency. 8. Refresh order states by data from different systems: assigned, confirmed, paid, done, problematic. Main challenges may be designing suitable data structures, designing flexible APIs, refreshing order states by communicating with different systems, persistent storage of histories in database with ability of dealing with high concurrency conflicts in reading and writing data. The project aims to developing a booking system backend which can communicating with other systems and the front-end. Core feature like tracing order states should be implemented considering various situations. Conflicts handling should be able to work well under a real-world scenario, but the performance of high concurrent request may be leaving for some optimisation to some extent. Related projects are mostly about backend systems implementation with ability of dealing high concurrency. Especially projects like ride-hailing software or apps containing multiple subsystems including payment system, assignments system and history reviewing system. The current state of the remoting barbering field is not very prevalent, however similar fields various a lot. Up to high performance SecKill systems which need extreme complicate designs, dealing with hundreds of millions concurrent CRUD. Down to simple apps that would crash under more than a hundred conflict. In this project I plan to build the mentioned system using JAVA language, Spring Boot framework, MySql database and middleware like Mybatis etc. Mange this by Maven and Git. This project would leave some APIs for the front-end and test the system during developing using virtual request tools individually. And some high concurrency simulators to evaluate the loading performance of the system. If everything goes well after the real-world implementation simulating, the objectives have been met!