

UNDERGRADUATE PROJECT PROPOSAL

|  |  |
| --- | --- |
| **Project Title:** | **Web- based Wether forecasting system** |
| **Surname:** | **Li** |
| **First Name:** | **Rox** |
| **Student Number:** | **201918020305** |
| **Supervisor Name:** | **Aymen Chebira** |
| **Module Code:** | **CHC 6096** |
| **Module Name:** | **Project** |
| **Date Submitted:** | **November 3, 2022** |

**Table of Contents**

[1 Introduction 1](#_Toc23133)

[1.1 Background (overview of topic and motivation) 1](#_Toc15945)

[1.2 Aim 1](#_Toc16955)

[1.3 Objectives 1](#_Toc14516)

[1.4 Project Overview 1](#_Toc23052)

[1.4.1 Scope 1](#_Toc1197)

[1.4.2 Audience 1](#_Toc16894)

[2 Background Review 2](#_Toc22232)

[2.1 Summary of existing approaches 2](#_Toc25748)

[2.2 Brief summary of related literature 2](#_Toc32372)

[3 Methodology 3](#_Toc12232)

[3.1 Approach 3](#_Toc1634)

[3.2 Technology 4](#_Toc19586)

[3.3 Version management plan 4](#_Toc16205)

[4 Project Management 4](#_Toc28312)

[4.1 Activities: tasks required to complete each objective 4](#_Toc1059)

[4.2 Schedule 7](#_Toc27117)

[4.3 Data management plan 10](#_Toc21310)

[4.4 Deliverables 11](#_Toc23917)

[5 References 12](#_Toc27118)

# Introduction

## Background (overview of topic and motivation)

Weather forecasting is an important and indispensable procedure in people's daily lives and, according to Selvam and Brorsson (2022), weather and climate science play a crucial role in all forms of life.

## Aim

It can be used to assess the changes that occur in the current atmospheric conditions, so it is very important to predict the weather more accurately.

## Objectives

After some research I found that most weather forecasting systems require downloading a mobile application, so the need to open the website directly for easy real-time viewing is indispensable. Therefore, I decided to use a set of distributed microservices Spring Cloud to complete a convenient and fast Web-based weather forecasting system. This will be a microservices project based on Spring Cloud's microservices architecture and the components or modules covered in it are built on Spring Boot.

## Project Overview

### Scope

To sum of, I will implement the following seven application frameworks in general: registry - service registration and discovery, config - external configuration, monitor - monitoring, zipkin - distributed tracking, gateway - interface gateway to proxy all microservices, auth-service - OAuth2 authentication service, svc-service - business service. Based on these services, I hope to provide a convenient Web-based weather forecasting system for users who do not want to download mobile applications.

### Audience

The weather forecast is relevant to all people's lives and it can provide weather related alerts to all people. Whether you are at home or need to go out (work, travel, etc.), you can be proactive for different situations.

# Background Review

## Summary of existing approaches

Compare my choice of SpringCloud technology, the techniques that can be seen in the current market to develop weather forecasting systems are 1. The use of ARIMA technique to develop weather forecasting tools, Autoregressive Integrated Moving Average (ARIMA) is a data mining technique commonly used for time series analysis and future forecasting (Shivhare et al., 2019). ; 2. now the mainstream microservices framework Dubbo, which uses a custom Dubbo protocol to achieve remote communication, is a typical RPC invocation scheme(Zhao, Jiang and Zhao, 2020). And SpringCloud uses Feign is based on Rest style invocation;

|  |  |  |  |
| --- | --- | --- | --- |
| Compare this to SpringCloud+SpringBoot which I will use to implement microservices development. Specifically, SpringCloud has the core technology for microservices development: RPC remote call technology; SpringBoot's web component integrates SpringMVC by default, which enables lightweight HTTP+JSON transfer and writing microservice interfaces, so SpringCloud relies on SpringBoot framework for micro So SpringCloud relies on SpringBoot framework for microservices development. | | | |
| Using the SpringBoot | Spring Boot Web Starter and Spring Boot Data Redis Starter running on Redis. | By default, spring-boot-starter-web provides a set of HttpMessageconverters for type conversion of request parameters and response results. | SpringBoot avoids a lot of development environment import and saves testing time and effort, Springcloud is a microservice solution - RPC remote calls, relationship Springcloud depends on SpringBoot (SpringMVC for web components). Springcloud will depend on with SpringBoot, because Springcloud write interface is SpringMVC interface, but due to the rapid iteration of the version, some modules will be very large changes, but as a simple implementation of the weather forecast site will not be affected too much. |

## Brief summary of related literature

Regarding Ramachandran et al.'s Healthy Weather app (2021), it is proposed that the app is based on to Android users, so I still prioritize the development of a weather forecasting website when targeting the relevant audience for all masses.

|  |  |  |  |
| --- | --- | --- | --- |
| Technology reference | Summary | Evaluation | Reflection |
| Using the Android Volley Library | The Healthy Weather app uses an HTTP library called Volley Library. Volley's two main classes, Request Queue and Request, are used so that all the necessary information for making network API calls is stored in them. | It makes networking very easy and fast | Volley is designed to be ideal for network operations that do not involve large amounts of data but frequent communication, while for network operations that involve large amounts of data, such as downloading files, Volley performs very poorly. |
| Using the Flask Framework | Flask is a lightweight Python based web framework that supports Python 2 and Python 3 and is easy to use and suitable for rapid development. | It enables fast and easy production of web applications with the ability to improve complex applications. | It is too lightweight and requires more call experience for the developer. |

# Methodology

## Approach

According to the explanatory model of Health Weather App by Ramachandran et al. (2021), the health weather app provides a lot of relevant information in a combination of graphics and text so that it is not challenging for the user to understand. However, after reviewing the article, there are no restrictions for people living at home to use the app, such as download time and mobile device storage space, but for travelers or tourists, an app needs to be downloaded and installed in advance, which proves that it requires the mobile device to provide the appropriate memory storage space in a timely manner. This is a problem that I do not have to consider for the web-based weather forecasting I am going to implement.

## Technology

Flask is a framework written in Python language and easy to understand the code , it is a language compared to the requirements of SpringBoot2.7.5 higher, because it is too lightweight , so developers need to have more experience in the development of third-party library integration calls(Mufid et al., 2019). For comparison, Due to the large base size, data and frequent operations required by the system (pei and peng, 2022), I used the SpringCloud microservices framework when choosing the technology to serve the web-based Weather forecasting system. The Gradle project language is Java. Dependency tool I choose Spring Web.

## Version management plan

Since there are many 24 total tasks to do, I will divide the outline into 6+9 by all needs. In the first half of the semester summarize six versions: 1.0 to 6.0, and in the second half of the semester summarize nine versions 7.0 to 16.0.

# Project Management

## Activities: tasks required to complete each objective

Technology Stack:

Spring boot - entry-level micro framework for microservices, used to simplify the initial build of Spring applications and the development process.

Eureka - Cloud Service Discovery, a REST-based service for locating services for cloud middle-tier service discovery and failover.

Spring Cloud Config - Configuration management toolkit that allows you to put configuration on remote servers and centralize cluster configuration, currently supporting local storage, Git, and Subversion.

Hystrix - Fuse, a fault-tolerant management tool designed to provide greater fault tolerance for latency and failure by controlling the nodes of services and third-party libraries through a fusing mechanism.

Zuul - Zuul is a framework for providing dynamic routing, monitoring, elasticity, security and other edge services on the cloud platform. zuul is the front door to all requests on the back end of the web site for devices and Netflix streaming applications.

Spring Cloud Bus - event, message bus for propagating state changes in clusters (e.g., configuration change events) , which can be used in conjunction with Spring Cloud Config for hot deployment .

Spring Cloud Sleuth - Log collection toolkit that encapsulates Dapper and log-based tracing as well as Zipkin and HTrace operations, enabling a distributed tracing solution for SpringCloud applications.

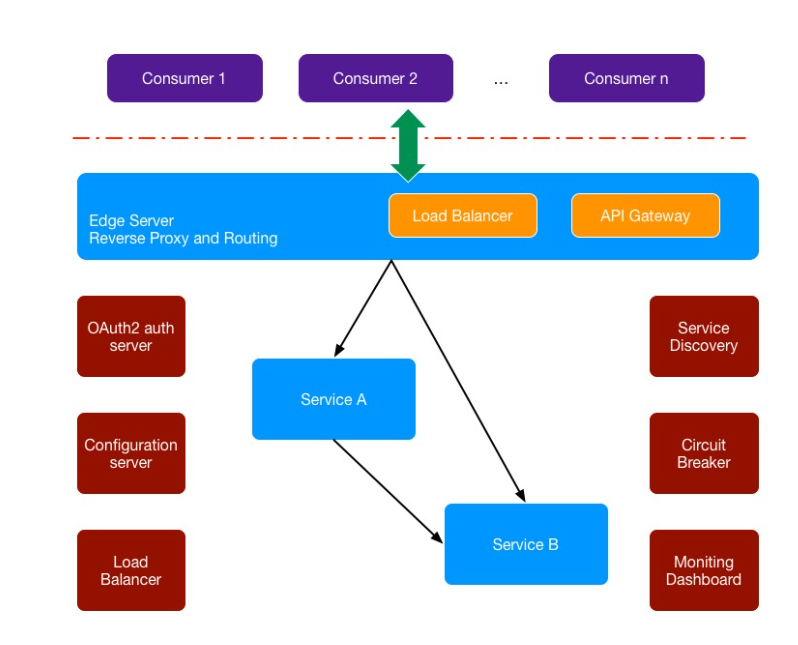
Ribbon - Provides cloud-based load balancing with a choice of load balancing policies that can be used with service discovery and circuit breakers.

Turbine - Turbine is a tool for aggregating event stream data sent by servers to monitor the metrics of hystrix under a cluster.

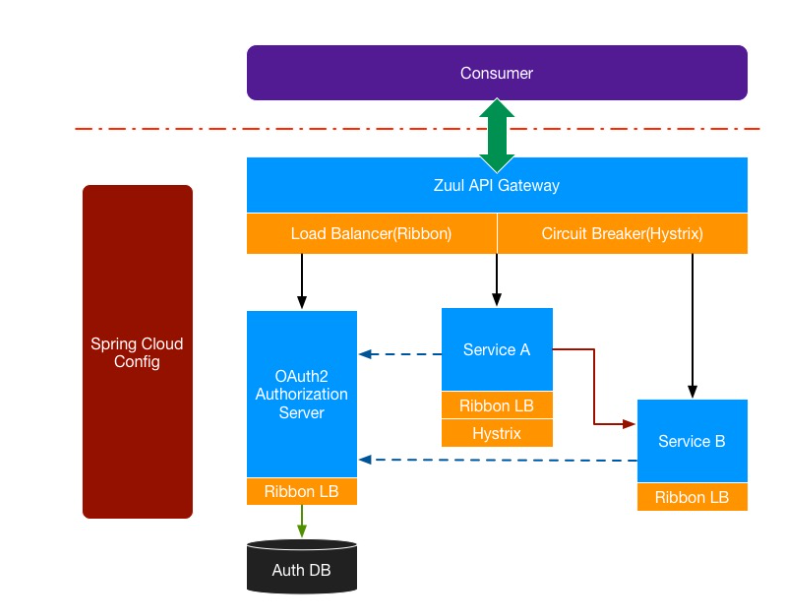
Spring Cloud Stream - Spring data streaming operations development kit that encapsulates sending and receiving messages with Redis, Rabbit, Kafka, etc.

Feign - Feign is a declarative, templated HTTP client.

Spring Cloud OAuth2 - A security toolkit based on Spring Security and OAuth2 to add security controls to your applications.



（Figure1: System Architecture)

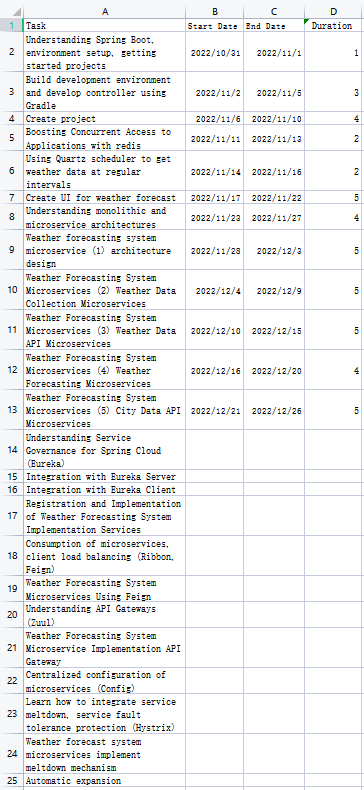


(Figure 2: Application Components)

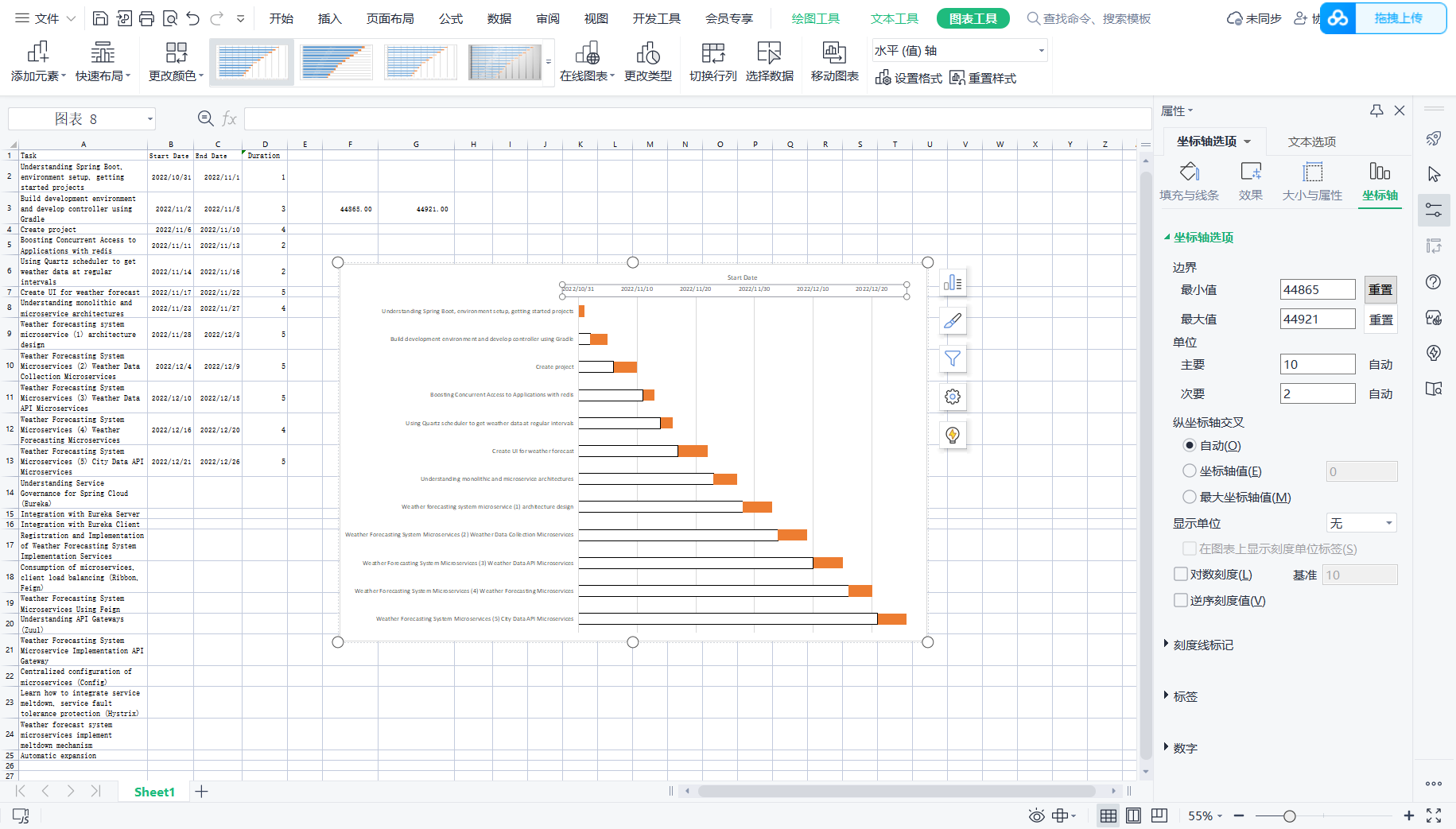
## Schedule

Tasks:

1. Understanding Spring Boot, environment setup, getting started projects
2. Build development environment and develop controller using Gradle
3. Create project
4. Boosting Concurrent Access to Applications with redis
5. Using Quartz scheduler to get weather data at regular intervals
6. Create UI for weather forecast
7. Understanding monolithic and microservice architectures
8. Weather forecasting system microservice (1) architecture design
9. Weather Forecasting System Microservices (2) Weather Data Collection Microservices
10. Weather Forecasting System Microservices (3) Weather Data API Microservices
11. Weather Forecasting System Microservices (4) Weather Forecasting Microservices
12. Weather Forecasting System Microservices (5) City Data API Microservices
13. Understanding Service Governance for Spring Cloud (Eureka)
14. Integration with Eureka Server
15. Integration with Eureka Client
16. Registration and Implementation of Weather Forecasting System Implementation Services
17. Consumption of microservices, client load balancing (Ribbon, Feign)
18. Weather Forecasting System Microservices Using Feign
19. Understanding API Gateways (Zuul)
20. Weather Forecasting System Microservice Implementation API Gateway
21. Centralized configuration of microservices (Config)
22. Learn how to integrate service meltdown, service fault tolerance protection (Hystrix)
23. Weather forecast system microservices implement meltdown mechanism
24. Automatic expansion



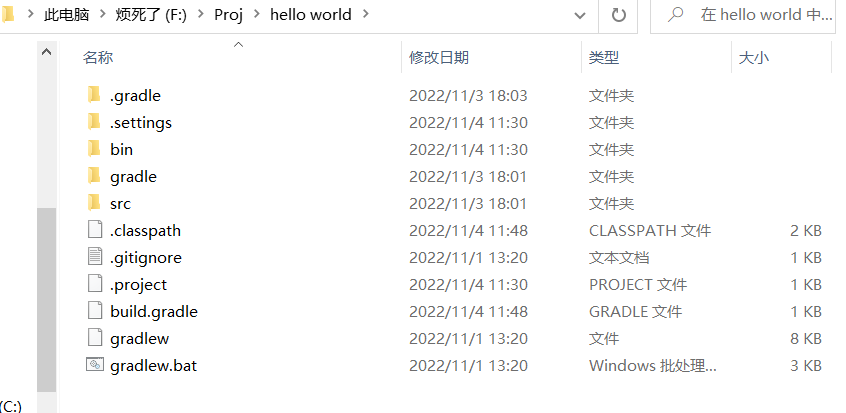
(Figure3: Tasks, Start Date, End Date)



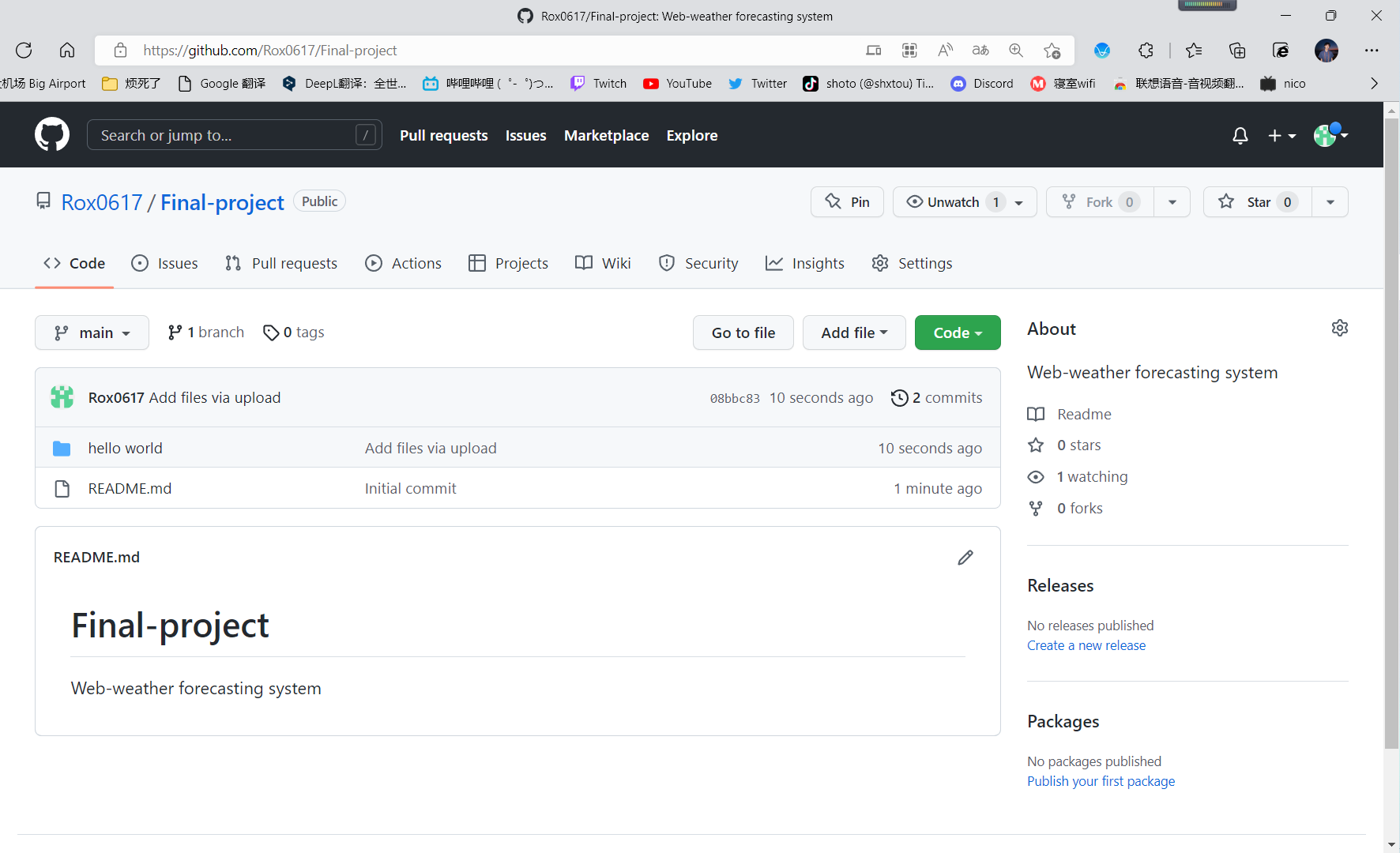
(Figure4: Project-gantt Full flow)

## Data management plan

My understanding of Spring Boot was followed by a Gradle environment build and modification of the Hello World starter project.



(Figure5: The local file)



(Figure6: The github file:https://github.com/Rox0617/Final-project)

## Deliverables

The task 1 to 12 in this semester

(Figure7: Chart)

# References

ABCDEFGHIJKLMNOPQRSTUVWXYZ

1. H. Zhao, Y. Jiang and X. Zhao. (2020). "Design and research of University intelligent education cloud platform based on Dubbo microservice framework," 2020 5th International Conference on Mechanical, Control and Computer Engineering (ICMCCE), 2020, pp. 870-874, doi: 10.1109/ICMCCE51767.2020.00191.
2. K. P. Selvam and M. Brorsson, "Performance Modeling of Weather Forecast Machine Learning for Efficient HPC," 2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS), 2022, pp. 1268-1269, doi: 10.1109/ICDCS54860.2022.00127.
3. M. R. Mufid, A. Basofi, M. U. H. Al Rasyid, I. F. Rochimansyah and A. rokhim, "Design an MVC Model using Python for Flask Framework Development," 2019 International Electronics Symposium (IES), 2019, pp. 214-219, doi: 10.1109/ELECSYM.2019.8901656.
4. M. R. Mufid, A. Basofi, M. U. H. Al Rasyid, I. F. Rochimansyah and A. rokhim, "Design an MVC Model using Python for Flask Framework Development," 2019 International Electronics Symposium (IES), 2019, pp. 214-219, doi: 10.1109/ELECSYM.2019.8901656.
5. R. Ramachandran. et al.(2021) "Health Weather App," 2021 2nd Global Conference for Advancement in Technology (GCAT), 2021, pp. 1-6, doi: 10.1109/GCAT52182.2021.9587785.
6. Shivhare, N., Rahul, A. K., Dwivedi, S. B., & Dikshit, P. K. S. (2019). “ARIMA based daily weather forecasting tool: A case study for Varanasi”, Mausam, 70(1), 133–140. <https://doi.org/10.54302/mausam.v70i1.179.>