Real-world Functional Programming

Project Report

14274056 Junsong Yang (psyjy3)

January 15, 2020

1 Introduction

In this section, the general idea about this project will be introduced as well as how this project would fit for this module.

The project is called recipe house, which is a recipe recommendation service based on the ingredients.The logic behind it is quite simple. It would require users to input whatever ingredients they have, then it will return recipes that used those ingredients. This project includes two parts: a back-end server and a web interface. Since I don't have any recipe database, the back-end server will query a third-party API for recipe info and trim the returned data in JSON and return the trimmed data as a response to the web interface.

The third party APIs that I will use are provided by Edamam. They have over 2 millions of recipes specified by diets, calories, nutrient ranges and simply just ingredients.

This project idea was used to compete in ATOS IT Challenge and was shortlisted(top 20 worldwide). The back-end was written in Go as well as the web interface with a bit vanilla javascript, but it was not quite finished. Hence, this idea will be reimplemented in a different approach for this project.

As for how this project may fit for 10 credits:

learning scala for backend (about 20 hrs)

learning React (with any host language) (20 hrs)

revisit web technologies (HTML CSS) (10 hrs)

implementation of the back-end (15 hrs)

implementation of the web interface (15 hrs)

report writing (20 hrs)

A successful project would be a working web interface that allowed user to use either text-based or image based search for recipes based on what ingredients they have. The front-end would sent the ingredients info to the back-end server via REST API. Then the back-end server will query the third-party API for recipes and parse the returned data and sent back to the front-end as response. The front-end will present the response from server to the end user.

2 Technical Background

In this section, the technological choices made for both the back end and front end will be discussed in detail with justification.

For backend server, the language of choice is Scala. Scala is a multi-paradigm programming language compiles to java bytecode and supports both functional programming and imperative programming. Hence, by using scala, we can not only explore functional programming in depth but also leveraging the java ecosystem.

As for the frontend interface, a javascript library called React will be used. The reason for that choice is that, React supports functional reactive programming paradigm which can be used to create graphical user interface.

3 Implementation

In this section, the general architecture of this project will be introduced as well as some essential implementation details.

4 Reflection