Reflective Report on Programming Clinic Group Projects

Lancaster University

LZPMPC005M: Programming Clinic

Reflective Report

Author: 39224996

Overview

The reflective report records the techniques and skills I have been learning during the 20-week project, including programming techniques and soft skills. After the course I understand that not only the specific coding skills are essential but also the non-technical skills, such as how to build your own workflow to improve efficiency or why do we need to do case study before starting the project.

The first section following would be the techniques and technical skills learnt during the course, which are necessary for smoothly implementing the program, including how to use UML, how to design a system, how to use GIT for version control and other knowledges.

The second section following would be the soft skills, including why do we need the ideas of project management, how to use case study to help designing the blueprint, why do everyone need to build up your own workflow and so on.

The third section consists of the detailed reflections on 5 different projects.

1 Skills and Techniques

1.1 Programming Techniques

1.1.1 UML and System Design

The full name of UML is Unified Modeling Language. It is an intuitive and powerful visualization tool not only for the developers but also for the customers (the developers ultimately serve for).

For the developers, UML can be used for:

1. Organizing the ideas and integrating all the ideas into a well-ordered system (or design/blueprint of the system).
2. Abstracting the core mechanisms of the system and translating them into concentrated diagrams.

3. Expressing how the system works and how the elements/entities interact from a higher dimension.

From my experiences and understandings, the reasons why it is universal are that:

1. UML also have standard syntaxes and icons. Even if for those customers who do not understand how the grammar works, they can still understand the design of the system by reading the diagrams.
2. UML contains a family of different diagrams for different purpose, thus by complementing each other they can comprehensively describe the system. For example, Use Case Diagram describes the activities/actions of the actors (could be human users, organizations or other systems). Class Diagram in another angle describes the entities and the relationships between in the system. While Sequential Diagram step by step explains detailly how those entities, actors or elements work between different sections of the whole system, like describing blood system that blood is flowing by the pumping of heart, carrying oxygen from the lungs to all over the body, and finally recycling again by the heart.

1.1.2 Git and Version Control

Git and GitHub are used for version control. From my perspective the reasons why version control is obligatory in developing are that it not only ensure a smooth cooperation between teammates, but also make sure when accidents happen, it can always save the previous steps for backup.

1.2 Soft Skills

1.2.1 Project Management

One of the most important lessons I learnt from the whole course is that project management is decisive for success. Having a background of management though, I never have a chance to apply management knowledge into practice. At the beginning of the project, I did whatever the lecturer wrote in the introductions week by week and didn’t have an overview from a high level.

It turns out that working like this is time-consuming and frustrating because I was always required to redo to satisfy unexcepted requirements.

But after I realized that I should have the initiative, I changed my methodology:

1. Understand the demands and form my own thinkings. Instead of following all the instructions given by others, I start to climb high and to have an overview at the project. Then I deconstruct the requirements according to my capability and my budget (here is my time).
2. Find examples to imitate and break the requirements down into reasonable tasks. I use UML to organize the ideas, design preliminarily the blueprint of the system and assign the tasks for myself. Still, it is very hard especially for me to accurately estimate the time I need to complete the tasks and I always overestimate my ability to learn and to implement the tasks. I still encountered many times I fail to finish the requirements within a given period.
3. Assess the results and adjust the strategies for next time. The other important thing I learnt is that it is impossible to give a perfect result. Nothing would ideally happen like what is expected in the plan. But what we can do is learn from the mistakes and try to improve little by little in next projects.

1.2.3 Case Study

One of the other important lessons I learnt that we cannot build a metropolis from nowhere. We all stand on the shoulders of the giants.

To build a project efficiently, the best way to begin with is to find existed examples. It is hard to start from zero, but it is not difficult and valuable to learn from others' projects. That doesn't mean copy and paste. What we can do is to form a concrete effect or imagination of what we want to build accordingly. The more specific and more accurate the blueprint is, the smoother it will be in the next steps.

1.2.3 Workflow and SOP (Standard Operating Procedure)

The invariable idea of all the skills above and following is that we should always try to simplify but not to add obstacles.

So, another I learnt from elsewhere and proved effective is that I should form a personal standard workflow. The more standard and clearer the procedures are, the more reproducibility they would have. It is intuitive that if we repeat the same steps serval times (according to same standards), it would be less painful and less time-consuming for us to complete the tasks.

2 Weekly Project

2.1 Bookstore Inventory Management

Techniques learnt:

- Python and Django framework
- IDE

For the first month it is not easy for me to start. Because I have never had the need to use IDE, it is the first time for me to get familiar with it. I have also only known several simple grammars of Python. So, what I decided to do was to follow a tutorial to set up everything.

During the processes, I learnt a lot about web applications and got a little more familiar with Django and Python. I started to get ideas of how the network works, how the client sends request via API to the server and how the server handle and respond.

2.2 Automated Health Clinic Management

Techniques learnt:

- C++ and Crow framework
- UML

For the second project, my teammate and I would like to try new techniques (so that we could explore new techniques). We didn't expect that C++ is not as friendly as Python for the beginner. (And there are not so many web framework in C++ as in Python or Java.) C++ is not easy for beginner to debug and to test, which cost us a lot of time and energy.

At the end of the project, I also started to learn how to draw UML. During my learning, it came into my mind that a good design is crucial for the following development.

2.3 Automated Traffic Management

Techniques learnt:

- MySQL
- Interactions between Multiple Servers

For the third project, I started to learn how to make several servers work at the same time according to my purpose. I also used the ideas mentioned that I should reconstruct the requirements first and implement accordingly.

At the end of the project, although I'm not satisfied about the data analysis section, which should have a better performance, I still managed to complete the majority of the tasks. My teammate and I made a system which can simulate the process of traffic, with a database connected running in the background, and a trigger program for sending emails to the users.

2.4 Digital Zoo Management

Techniques learnt:

- HTML and CSS

The fourth project requires us to implement interactions between the clients, the main server and the database server. It is not hard to connect the main server and the database, but when it added the third subsystem, the complexity boosted. For me I need more time to get familiar with the whole system, and the cooperation with teammate also became more complex. And I found that personally I prefer to study the interactions between the backend servers rather than to study how to display fancy effects on the frontend.

2.5 Online Auctions

Techniques learnt:

- Socket protocol
- Java and Express.js framework

For the last project, I started to think how to implement real-time responses rather than HTTP requests and responses. I also started to think about adding more backend servers for different

purposes. For example, one main server to handle all the main functions, and an authentication server to handle all the login requests. But I failed to manage my progress because I spent too much time on how to implement real-time progress, which also happened in my personal project.

But at least I learnt that we could handle real-time functions by socket (like making phone calls where there would be someone listen to you and give response immediately). And I also started to get familiar with java framework that I haven’t used before thanks to the great contribution of my teammate.