City University of Hong Kong

Department of Computer Science

Summer Storage Management System for CSSAUG

self-assessment report

Collaborators:

DONG Jia Jie

LI Xiao Yang

SHA Xin Chen

ZHENG Shang Kun

ZHOU Yu

ZHANG Tian Tian

Contents

1 DONG Jia Jie **2**

2 LI Xiao Yang **4**

3 SHA Xin Chen **5**

4 ZHENG Shang Kun **7**

5 ZHOU Yu **9**

6 ZHANG Tian Tian **11**

1 DONG Jia Jie

This semester-long project development taught me the importance of team corporation and project management. Compared with CS3342, CS3343 is doing the actual code project. What's learned from books is superficial. Only when doing the real planning and integration do I know that teamwork should not only depend on coding ability but also on communication. Unlike personal assignments or projects, CS3343 is the first time to get in touch with big projects. Thanks to my teammates for helping develop the project together.

During the semester, we have a weekly group meeting, and we exchange the work progress and discuss the further steps. My job is the whole UI component of the project. I write the pages of all the functions. My job needs to be tested on others' work. So communication on the work progress is essential for me to develop the next step. I also checked the consistency with another teammate responsible for reading data every few days. I enhanced my integration ability during the semester.

I also learn how to use GitHub for project uploading and updates. The commitment should be written formally, and the merge branch should be discussed with other team members. The code format is also essential for a large-size coding program. Coding is not your own work, and you should let others know what you are coding. The group work helps a lot with my code style. Naming the parameters meaningfully and also refactoring the code to prevent clutter.

During the testing process, each unit is tested and then integrated. Each team member corresponds to the client using our application and performs all possible functional tests. In the later stages of the project, we try to put ourselves in the user's shoes and constantly

improve the application to enhance the user experience.

At last, thanks to all my teammates. They all spend their effect on the development of the project. We exchange our ideas and design the function. The project can not be well finished without all team's work.

2 LI Xiao Yang

In this project, I worked on implementations of user interfaces, repository management and exception design. During the period of development, I applied my expertise in development practice and what I've learned from the lecture. Also, I gained a lot from this opportunity to collaborate with peers and challenge my self.

One of the most important takeaways for me is that an excellent code of management is incredibly significant in real practice of development. Management practice of good quality largely contributes to reducing development issues and improving efficiency. For instance, resolving conflicts after git pull can be exhausting. An alternative approach is to avoid the conflicts beforehand by good code synchronization manner. Besides, another concern is that bad commit specification leads to problems with finding the corresponding commit when code reset is needed. To facilitate our team work, we divided the duties and assigned different roles to each and every collaborators of our project.

Apart from the ability to propose solution to problems through independent investigation, developing communication skills in the process of development is another lesson to learn as well. Especially when components of the project invokes each other, abilities of defining questions and specifying function prototype count. Our solution is to maintain function specifications and development logs for internal purpose. By doing so, documentation job can be done much easier.

To better develop professional skills in our future career, further improvement can be made in terms of system design and documentations. Last but not least, I would like to thank all collaborators for the constructive suggestions and comments.

3 SHA Xin Chen

I am in charge of database programming, bug reporting, and test case report alongside my collaborators in this project. We have an enjoyable project development experience and have learned much about teamwork, project management, Java programming, and testing procedures.

Although we have carried out a relatively reasonable division of labor in the teamwork process because everyone has different abilities and strengths. We still need to compromise with each other from time to time. For example, I was unfamiliar with git repository management and unaware of its importance at the beginning. Therefore, at the beginning of our project, I sometimes made mistakes that led to the extra work of my team members. Luckily, my teammates are all so patient and helpful that I gradually grasp this skill and successfully finish my part of the programming work. I discovered that collaboration requires everyone's involvement, and communication in this process is crucial. Fortunately, with effective communication, our team performed well in this section. In the first stage, frame building and prototype design are crucial. If the main programming begins without a rigorous and understandable class structure, problems are likely to arise. Once we meet the issue, we will spend time together designing and solving the problem, and the result is a gratifying project that adheres to current and professional standards. Additionally, the time required for major programming and testing has been greatly reduced thanks to effective collaboration.

We have practical hands-on testing experience as we create our projects. We first examine the best testing approach for our product, and then we finally settle on the sandwich testing method. Then, as part of the testing process, we assessed how each

class is related to the others. To obtain the most thorough coverage, we employ testing strategies that are covered in lectures. Nevertheless, we might still discover that our product can include certain problems along the process. As a result, during the final edge, our team members also worked as actual users to mimic the acceptance test and to identify flaws in specific edge instances. We dealt with these bugs swiftly and included them in the bug report. Writing bug reports, doing debugging and testing, and creating new releases are all very beneficial experiences for me.

In terms of project management, our team did quite well. We have a thorough project strategy and cost estimates for staff distribution, and everyone adheres rigidly to the deadline in each phase. This project taught me a lot and I believe will make me more competitive in the workplace.

4 ZHENG Shang Kun

I have learned coding for almost eight years since junior high school. But it was the first time I participated in creating such an app together with five hardworking and enthusiastic teammates. It is my honor to work with them, and I am more than delighted to give my best appreciation to them.

Looking back on the process of this course project, many things teach me and inspire me. First of all, I learned to use Github, which is the star product among programmers. We shared and managed our work in Github, making the cooperation a success. Meanwhile, although it had been taught in course CS3342 last semester, it was the first time that we applied the SOLID principles and the design patterns in the coding. Now I have a higher understanding of these principles and patterns about their content and why they are important. Without them, I think the whole project would become a mess.

This course project also made me deeply realize the importance of unity and cooperation, which is the basic quality of an excellent development team and the key to the success of a project. Although there are only a few people on our team and the time is short, we finally completed the task. What do we rely on? Not only efforts but also unity. During the whole project process, we are very particular about unity and cooperation. We have a clear division of labor and can help each other to solve problems. The timely solution to the problem ensures that the project progresses as planned. We have also had many disputes, which I think is a good phenomenon in our work. We have got a correct understanding of the dispute and learned a lot from each other. Software development is not the work of one person, but the work of a team, so we must consider the whole in the work, and can't do it alone.

Another thing is that I think prototype design plays an important role in software engineering. On the one hand, it can materialize the user's requirements and help us review the requirements specification. On the other hand, providing the prototype to the customer in the requirements phase can get timely feedback on the problems, avoiding the customer telling us that it is not what he wants after the project is completed. Moreover, the prototype provides a good reference for our coding design and improves efficiency.

I think this course project has played a role as a bridge for me to go to society, which is just an important experience and an important step in my life. It will also be of great help to my future career.

5 ZHOU Yu

Through this course and project, I learned a lot about project management and team collaboration. There are indeed differences between teamwork projects and solo projects, since we could not work on the project with all settings as expected. We should always communicate efficiently with other team members, absorb their ideas and express our thoughts. Thankfully, all members of our group worked hard and were enthusiastic about the project. Many thanks to all members of group 13!

The sections I was majorly responsible for were the construction of basic underlying data types and the manipulation of our development branch on GitHub.

Regarding my coding job, since I worked on the most basic underlying data types, it is essentially crucial to determine the partitioning of all the classes and meanwhile correspond to the standard design principles and patterns at the beginning phase. From last semester's CS3342, we got to learn some practical techniques to encapsulate the classes; however, when applying them to the coding work, it was not that easy, as we had to inform other team members of the reasons and ways to comprehend the code either by embedded comments or chats through social media or in person. Besides, due to the uncertainty about the required methods and attributes, we needed to modify our codes according to other members' needs, and thus our coding period lasted much longer than we expected.

In terms of the software branch development, it has allowed me to put the techniques learned from my placement job into use, which I deeply appreciated since the version control tool is really helpful when it comes to multiple people working together on different parts of the job. For the past programming group projects, what we used to do was to

pass the whole package of the codes through direct transmission, which was not only hard to track the history but also made it troublesome to compare the different versions of codes. Though the git version control system requires some preliminaries to fully perform its advantages, I believe it is worthwhile to learn. Meanwhile, it also facilitates us to get familiar with some commit standards and rules, which I believe would benefit us immensely in future real software development work.

In the end, I would like to express my greatest gratitude again to all of our team members who worked eagerly and diligently on this software development project. It is your intelligence and collaboration that brings this well-functioning program eventually!

6 ZHANG Tian Tian

This training also made me deeply realize the importance of unity and cooperation, which is the basic quality of an excellent development team and the key to the success of a project. Although there were only a few people in our team, and the time was very short, we finally successfully completed the task, by what? Not just effort, but unity. During the whole process of the project, we always pay attention to unity and cooperation. We have a clear division of labor and can help each other to solve problems. The timely resolution of the problem ensures that the project progresses as planned. We also had a lot of arguments, and most of the time we argued until we were red in the face. In my work, I think this is a good phenomenon. We got a correct understanding in the dispute and also learned a lot from each other. Software development is not a person's work, but a team's work, so the work must consider the whole, can not be arbitrary. Of course, this month's training also gave me a lot of experience and lessons.

First of all, we should be careful in every part of the project. Any link can not be sloppy, perfunctory. A mistake in one link may lead to a failure in the following links, or even a complete rework. Only successful requirements analysis can lead to successful outline design, and only successful outline design can lead to successful detailed design. Successful detailed design is the premise of ensuring clear coding ideas and fewer code bugs. But I didn't pay enough attention to this point. Start to think that as long as the product to make it OK, the other is not important. Therefore, the requirements document and design document were sloppy at the beginning, and I was a little confused at the beginning of the coding. My mind was in a mess and problems kept appearing. When I was in the middle of the coding, I could not go on. Finally, I have to redesign according to the requirements, and then start coding from the beginning according to

the design. This wasted a lot of time, and I felt that the initial rash start to code took a lot of effort, and nothing came of it. After the design is done, coding is much easier, the idea is clear from beginning to end, and it is very efficient to do. This painful lesson I think is enough to keep in mind in my future work. Coding is not the be-all and end-all of software development, it is just a part of it, and all documentation is equally important. I remember that the life of the software I learned in school is not coding, but maintenance, which cannot be maintained without detailed and correct documentation. Second, the documentation should be as detailed as possible, and the code should be named conventionally. As mentioned above, software development is the work of a team. Maybe you are very clear in your mind, but your colleagues do not know what you think. Only detailed documentation can ensure that the whole project team has a common understanding. Take requirements as an example. If project team members have different understandings of some parts of requirements, it is possible to have a big disagreement in the end. The naming of the canonical code is also a problem that cannot be ignored. On the one hand, there may be problems in the integration, even if it is just the case of file naming (I made this mistake in this project, and the lesson is very profound. In fact, it is the case of the problem. On the other hand, misnamed code is very unreadable, which makes testing very inconvenient. Due to the lack of time, we just did a code search within the group. I found it was very difficult to read due to our irregular naming and individual naming style.