City University of Hong Kong
2024-2025 Semester A
CS3343 Software Engineering Practice

Self Assessment
Group 38
Hong Kong Journey Planner

Conducted by:

Name	Position
Fan Tianrui	Project Manager
Wang Fan	Assistant Project Manager
GAO Nanjie	Developing Analyst
CHEUNG Lok Yi	Testing Engineer
LIU Hengche	Program Developer

Department of
Computer Science
香港城市大學
City University of Hong Kong

Dec 4, 2024

Table of Content

1.FAN Tianrui

As a project manager in the CS3343 course project, I gained deep insights into the importance of teamwork. Unlike the theoretical focus of CS3342, this practical experience taught me that successful projects require not just solid programming skills, but effective communication and coordination.

I was responsible for developing the restaurant module and managing testing tasks. Through weekly team meetings, we synchronized progress and planned future work. To ensure code quality, I regularly reviewed coding standards with teammates working on the recommendation system, maintaining consistency. This process significantly enhanced my system integration capabilities.

In version control, I mastered GitHub practices, learning proper commit messaging and branch management. Team collaboration improved my code readability, including appropriate naming conventions and modular refactoring.

The testing phase followed a progressive approach from unit to integration testing. Team members took turns role-playing as users, comprehensively testing functionality and optimizing experience. The project's success stems from everyone's dedication, with members working together and pooling ideas to deliver an excellent final product.

2.WANG Fan

As the Assistant Project Manager of the Hong Kong Journey Planner project, I primarily focused on the project management, the development of the plaza recommendation system, and the testing of I/O components.

Project management is a brand new experience for me. Despite having led group projects in the past, the difficulty and intensity of this project is far beyond my previous experience. It is only possible for us to deliver the project on time and with high quality by having well-organized project management. While our other team members scheduled the project, allocated resources, and managed the risks, I was responsible for coordinating the team, ensuring the project is on track, and evaluating the deliverables.

Git and GitHub are highly utilized in our project, and I was responsible for managing the repository, merging the branches, and resolving the conflicts. Bugzilla is the repository where we report, track, and manage the bugs. I provided technical support by overseeing the bug reports, assigning the bugs to the responsible team members, and verifying the fixes. To maximize the effectiveness of bug tracking, I also created bug reporting guidelines and a report template for reference. All these are consolidated in the Bug Report document. From this project, I have learned the importance of project management, and the skills to manage a project effectively, including time management, communication, and leadership. We have successfully delivered the project on time, and I am proud of what we have achieved.

As for the technical part, I also contributed to the development of the plaza recommendation and selection management systems. I was responsible for collecting data from TripAdvisor via web scraping, and developing the recommendation algorithm. This is effectively a precise chance to practice the design principles and patterns we have learned last semester in CS3342. I utilized Singleton pattern to create the SelectionManager class, and Strategy pattern to implement different recommendation algorithms across different types of attractions with the same method signature. Command pattern is also used to encapsulate the request as an object, and pass it to the SelectionManager, allowing undo and redo operations. It is satisfying to see all these knowledge and skills are applied in a real-world project, organizing the code in a more structured and maintainable way.

I also conducted thorough testing on the I/O components, from the input validation, to the output formatting. The most challenging part is to extend existing console-based I/O components to accept multiple sources of input and output, including the file and string. By investigating the Java I/O API, I employed InputStream as the universal input source, which can be derived from System.in, FileInputStream, or ByteArrayInputStream. I also acquainted myself with the Java NIO API and reflection API, which allows resetting the Singleton instance for testing, without exposing the private fields and methods to unauthorized access. This is an educational experience for me to gain more insights into Java I/O, reflection, and testing techniques. It is also a good opportunity to practice test-driven development, by writing the test cases before the implementation, and ensuring the code is working as expected.

Finally, I was also responsible for generating the executable JAR file. I utilized Maven to manage dependencies, and to automate the build process. Since the maven package command automatically runs the tests, it can be ensured that the JAR file is built only when all the tests are passed. This is a good practice to ensure the quality of the deliverables, and to prevent the bugs from being introduced in the production environment.

In conclusion, I have learned a lot from this project, both in terms of project management and technical skills. I have gained more insights into the project management, design patterns, Java API, and testing techniques. I am certain that these experiences will be beneficial for my future career, and I am looking forward to applying these skills in my future projects.

3.GAO Nanjie

In this project, my primary responsibility was code refactoring, where I focused on several key aspects aimed at enhancing the readability, maintainability, and scalability of the codebase. My tasks included renaming variables and functions to improve clarity, consolidating redundant code, and applying design patterns such as polymorphism to simplify complex conditionals and eliminate repetitive logic.

I began by thoroughly reviewing the existing code to identify areas that were either difficult to understand or contained redundant patterns. One of my main contributions was replacing complex and repetitive conditionals with polymorphic solutions, which improved the flexibility and extensibility of the system. Additionally, I renamed variables and functions with more descriptive and meaningful names, making the code more self-explanatory and easier for the team to collaborate on. I also focused on breaking down long, monolithic functions into smaller, more manageable pieces, improving modularity, readability, and making it easier to write unit tests for isolated components.

Throughout the refactoring process, I worked closely with my teammates to ensure that the changes I made didn't interfere with the functionality of other modules. Collaboration was essential, as my changes often had a broader impact on the project. I had to ensure that the refactoring improved the overall design without causing new issues or introducing regressions. The iterative process of making changes and then verifying their impact with the team helped refine the codebase and ensured we maintained project momentum.

One of the main challenges I encountered was finding the balance between optimizing the code for efficiency and maintaining its simplicity and readability. Over-engineering solutions can sometimes complicate the code and make it harder for future developers to work with, so I had to be careful to avoid that pitfall. At times, I had to carefully assess the trade-offs and decide whether a more complex refactor would add long-term value or whether a simpler approach would be more effective.

In the end, my efforts in code refactoring contributed to a cleaner, more organized, and more maintainable codebase. It also enhanced the performance and flexibility of the system, making it easier to extend in the future. This experience not only strengthened my technical skills, particularly in applying design patterns and optimizing code, but it also reinforced the importance of clear communication, teamwork, and thoughtful planning in software development. By working closely with my teammates, I learned how to navigate the complexities of refactoring in a collaborative environment and how to make decisions that benefit both the immediate and long-term goals of the project. This experience has been invaluable in helping me grow as both a developer and a team member.

4.Cheung Lok Yi

It is an incredible experience to participate for the first time in creating this system. I am grateful to work with hardworking teammates. I have been happy to work with them, and I want to appreciate their efforts and support.

The whole journey of this course project has taught and inspired me a lot. One was how to use GitHub, which is an important tool for programmers. We shared and managed our work on GitHub, which contributed much to our collaboration's success. I also learned to use Git Fork, a powerful tool that helped us resolve conflicts and track changes efficiently.

Although the SOLID principles and design patterns were introduced to us last semester, this project marked the first time that I applied these in a practical setting. This experience deepened my understanding of these principles, their content, and why they're so important. I can also apply my diagram skills in this project to create sequence diagrams. It is a good opportunity for me to practice my skills.

I also participate in managed project activities, including tasks, timelines, and deliverables. Tools such as sequence diagrams, Gantt charts, and WBS structures ensured that milestones occurred on time and that communications remained appropriate and orderly.

The other thing I learned from this project is collaboration. Each of us had our respective responsibilities, although the tasks were interrelated. We helped each other trace errors and bugs to fix the problem and keep working on schedule. I have also learned a lot from my teammates. While we were assigned to different areas, like testing, code refactoring, and design principles, we shared knowledge and experience, enriching our understanding of each other's systems.

This course project marks one of the major milestones in my life and a significant step toward my future career. It has not only equipped me with technical skills but also taught me interpersonal skills. The lessons learned from this project will continue to guide me in taking on new challenges in life.

5.LIU Hengche

This semester’s project development really opened my eyes to how programming works and the importance of working together with my teammates to hit our deadlines. CS3343 has been a real game changer for me, especially compared to CS3342. It’s one thing to come up with design ideas, but actually putting them into practice using design patterns has been a whole new level of challenge.

One of the biggest perks of our group meetings was how they created a space for open and honest communication. Getting together regularly helped us clarify what we needed to do, brainstorm solutions, and tackle any issues before they got out of hand. It really felt like we were all in it together, and that made a huge difference during crunch times when we needed to make quick decisions.

Learning to use GitHub was another big highlight. It made our workflow so much smoother! We could easily track changes, manage everyone’s contributions, and quickly resolve conflicts. The pull request feature was especially helpful for code reviews, allowing us to give and receive feedback that improved our coding skills. It taught me just how vital version control is for keeping everything organized and for fostering teamwork.

I also learned a lot about the importance of testing and code refactoring in making our project successful. By implementing tests early on, we were able to catch bugs quickly and make sure that new features didn’t mess up what we’d already built. This gave us a lot of confidence in our code. Plus, taking the time to refactor our code helped make it cleaner and easier to read, which is super important for anyone who might work on it in the future.

I can’t thank my teammates enough for their hard work and commitment. Everyone brought something unique to the table, and our different skills really complemented each other. This project taught us how crucial teamwork, communication, and perseverance are. Together, we didn’t just create a solid program; we also developed skills and experiences that will help us down the road.

In the end, this project was about more than just getting a grade; it was a fantastic learning experience that reignited my passion for software development and collaboration. I’m excited to take what I’ve learned here into future projects and keep growing as a developer.