Assessment II

Reflection Report

MSE800 – Professional Software Engineering

Group Members:

Hengpan He

Wen Liang

Arnold Aristotle Tayag

Date:

31 October 2024

Contents

1. Reflection Report..3

2. Demand Gathering 3

3. Requirements Analysis..3

4. Coding..4

5. Testing..4

6. Challenges and Lessons Learned...4

1. Reflection Report

The Tour Booking System is a project that has its origin from the increasingly felt need in the tourism market for an effective and user-friendly online booking system targeting tours around New Zealand, with initial focus in the Auckland region. In this group project, we have worked toward developing a system that would give tourists an enhanced booking experience on the one hand, while giving operators a set of tools to operate tours booking with maximum efficiency. Agile methodology allowed us to adapt flexibly in the face of evolving requirements and user feedback during all stages of the project's lifetime. This report reflects on the stages of requirements gathering and analysis, system architecture design, code implementation, testing, and release, focusing on the lessons learned at each stage.

2. Demand Gathering

This is the demand-gathering phase, during which we have had limited stakeholder interactive engagement in the form of surveys and interviews with potential users: tourists, tour operators, and administrators. Of course, these roles are acted by each of our team members respectively and virtually. Role of tourists had a number of suggestions on how to make their booking simpler, having full information about the tour, and reviewing options. One important challenge was to strike a balance between the seemingly conflicting requirements presented by different needs, and these needs were carefully documented and categorized and had to be balanced.

3. Requirements Analysis

We prioritized requirements into features such as "Must-have," "Should-have," "Could-have," and "Won't-have." The reason for this methodology is that it will enable us to stick to core functionalities while still accommodating some functionalities that can be part of future enhancement plans. Hence, at this stage, the user demands were mapped into functional and non-functional requirements. This helped us outline the project scope for each version and helped plan our Agile sprints in that respect.

4. Coding

The coding was divided into one-week sprints in which we would develop specific features based on the prioritized requirements. We went with Agile methodologies which allowed us to fix our development process as we went and perform feedback intake or any other changes on the go right away. After each sprint, we had a working prototype, reflecting the incremental approach towards our product in its entirety.

5. Testing

Testing forms one of the very key features related to the project, being done through various stages of the project in order to ensure the quality of the system. We started unit testing, verifying that all the components were behaving as anticipated. Then we moved on to integration testing and followed by UAT. But due to the time limitation we skipped performance test in the first release and planned to implement in the second stage for the reason that at the earlier period of online services, the number of registered costumers will be less.

6. Challenges and Lessons Learned

We faced various types of challenges while working on this project; each taught us something important. An important challenge was coordination and communication within the team during different phases of the project. The overall cooperation among each one of the team members taught us the importance of clear communication protocols and collaboration tools, which maintained the alignments and reduced misunderstandings.

Other challenges were technical integrations because rigorous testing and fixing needed to be done to ensure a smooth flow of data across these components, especially when real-time updates occurred in the booking status. This only reinforced the idea of modular and robust documentation since these allowed us to fix problems much faster.

Finally, the time management was an issue, too, especially in the closing stages after the coding phases. In balance with the thoroughness of testing, timewise for the project forced us to prioritize which features were most vital and which we could get away with not focusing on for each sprint. This taught us the importance of setting realistic goals and timelines for each cycle of development, since project timelines have to be respected because one wants to deliver a quality product within expected times.

The challenges we have gone through really imparted lessons on communication, technical integration, time management, and user-centered development. Such insight from our group project will surely help us to develop anything even more effectively and efficiently in our future projects.