**CS 250 Final Project**

One of the lessons I learned during this entire process was the value of adapting on the fly without losing focus on the core deliverables. The overarching goal was to build a user-friendly travel application for SNHU Travel, but shifting stakeholder priorities meant that a robust Scrum-Agile practice became more critical than ever. My experiences with daily stand-ups, user stories, and mid-sprint adjustments consistently reinforced how incremental development combined with constant communication lets a project advance smoothly.

A primary focus for me in this project was understanding the application of the four primary roles, Scrum Master, Product Owner, Developer, and Tester. Although it is possible for one person to handle multiple roles in a smaller project, the importance of distinguishing these responsibilities became apparent as multiple members makes it easier to maintain transparency and prevent overlooked tasks. The Scrum Master role kept the train on its tracks, handling daily standups, making sure discussions didn’t wander and that any blockers were raised right away. This was especially helpful when we implemented test driven development early on since having a dedicated role to encourage TDD allowed for smoother adoption of that practice. The Product Owner oversaw the product backlog and consistently adjusted priorities based on shifting business needs. When we had the major pivot involving emphasizing a detox and wellness theme, the Product Owners responsibility was to introduce new stories that showcased spa amenities and healthy dining information. This approach aligned with the new goal of tapping into the spa-travelers seeking a rejuvenating experience, despite the original backlog being centered primarily on standard vacation searches and bookings. The Developer role I carried out on occasion put each user story into practice. Early on the tasks included the essential booking flow, while later on they centered around adding or updating text and photos for spa amenities. We used test driven development and frequent code reviews to transform user stories into functional features, working closely with the Product Owner to clarify acceptance criteria and confirm that the final implementation accurately reflected stakeholder requirements. The Tester role created specific acceptance tests, verifying that newly added content such as detox descriptions and promotional slideshows met the agreed-upon Definition of Done. At one point the Tester needed some additional details about how text and images needed to appear in the new slideshow. Quick responses from both the Product Owner and the Developer clarified the scope of the tests, ensuring no incomplete or inconsistent functionality would slip into production at the end of a sprint.

From the start, each sprint was fueled by user stories that spelled out exactly what end users needed. At first, I focused on the basics: a booking engine, package listings, and some basic search filters. It was straightforward enough, until we needed to make a pivot and incorporate the entire wellness and detox angle. We needed to revisit the backlog, pulling out user stories that emphasized spa amenities, healthy dining, and a more “mindful” travel experience. This shift forced me to update acceptance criteria like which images belonged in the slideshow, or what descriptions would change. This criteria outlined precisely how a user would see these healthy options, how the search function should behave, and how images would confirm the type of available amenities. By timeboxing each story within a sprint, work could be demonstrated to stakeholders at the end of each iteration and refined based on immediate feedback.

The wellness pivot wasn’t just a minor tweak; it required me to rewrite some of the content, update visual elements, and confirm with the Product Owner that everything still flowed logically for a user. Daily standups were a major help here, since those short, 15-minute meetings allowed me to call out any blockers or confusion as soon as they popped up. I also appreciated the clarity of a simple “What did I do yesterday? What do I plan to do today? What’s blocking me?” format, which kept the entire conversation focused. Outside those standups, I relied heavily on email check-ins to keep everyone aligned. If new acceptance criteria for the spa images emerged, I’d send a quick note confirming how big the images should be or how they fit with the existing page layout. Because I looped in the Tester role, we avoided the risk of pushing an update that tested incorrectly. That synergy between the standups, email follow-ups, and a well-prioritized backlog made all the difference when the project demands shifted so dramatically.

Several Scrum staples kept the project on track. Sprint Planning gave me a chance to break down the highest-priority user stories and ensure they were both sized correctly and fully understood. More than once, clarifying a single story’s acceptance criteria in Sprint Planning saved countless hours of rework later. Daily Scrums may be short but they are critical to surfacing any new questions or clarifications about detox amenities. Sprint Reviews provided a formal space to show what was completed and gather real-time feedback, making it clear whether the newly added features truly aligned with the redefined wellness vision. Lastly, the Sprint Retrospectives allowed me to reflect on what worked well (for instance, refining user stories early in the sprint) and what could improve (such as being clearer on how final text should appear so the Tester didn’t need last-minute clarifications). Regularly pausing to assess and adapt helped me refine not just the product, but also my approach to the project itself.

Overall, this experience solidified my belief that an iterative, collaborative framework is irreplaceable when requirements can shift at any moment’s notice. If this project had been ran on a more rigid fashion, the detox pivot might have tanked the whole timeline. Instead, we simply added or adjusted user stories, updated sprint goals, and kept moving forward. The major challenge was ensuring that any new information like spa marketing or imagery was gathered quickly so the Developer and Tester roles could act on it in the same sprint. Either way the daily standups and consistent backlog grooming made sure that the team never fell too far behind or had to redo massive swaths of code. The TDD approach also dovetailed nicely with Scrum. By writing tests that outlined exactly how the wellness features should work (or display), it became easier to confirm whether the final product matched what the Product Owner actually wanted. If a test failed, it signaled a discrepancy between our assumptions and the real requirements. This clear feedback loop cut down on guesswork, especially when new wellness-related acceptance criteria popped up without warning.

Building the SNHU Travel application under these evolving circumstances truly illustrated why Scrum-Agile excels in dynamic environments. Although the shift toward detox-focused amenities could have disrupted development, the transparent and incremental nature of Scrum helped me weave those new features into the existing product with minimal setbacks. The consistent reliance on daily communication, well-defined user stories, test-driven development, and roles specifically tailored to keep everything running smoothly was key to delivering each increment of value on time.