SAU

SOUTHERN ARKANSAS UNIVERSITY

INTERNSHIP EXPERIENCE REPORT

Zhen Ze Ong

J.B. Hunt Transport Services, Inc.

May 29th, 2024 – July 24th, 2024

Table Of Contents

1 Biographical Sketch

Zhen Ze Ong, who also goes by Justin, is an international student from Selangor, Malaysia. At the time of this internship, he was a Senior at Southern Arkansas University, with a B.S in Computer Science: Cyber Security and Privacy Option along with a minor in Mathematics.

Justin was involved in the Honors College society as well as in the International Students Association. He worked as a student research assistant for his professor under DART (Data Analytics that are Robust and Trusted) and as a peer tutor for the Academic Enrichment Center on campus. He plans to pursue a career in software development after graduating from college.

2 Company Background

J.B. Hunt Transport Services, Inc. is a leading American transportation and logistics company in the trucking industry. It was founded by Johnnie Bryan Hunt and Johnelle Hunt in 1961. Today its headquarters is located in Lowell, Arkansas. In recent years, the company has grown to become a NASDAQ-100 company, leveraging load optimization software to provide efficient and reliable transportation routes for their drivers, ensuring the seamless movement of goods. J.B. Hunt is committed to its five values: integrity, respect, innovation, safety, and excellence, making it a trusted partner in the industry.

3 Internship Program

The internship was a paid, in-person program over the course of two months. There was a total of 198 interns, and I was among the 62 application development interns. The application development interns worked certain days of the week in an intern office in Fayetteville, Arkansas and rest in the main corporate office in Lowell, Arkansas. We were assigned to full-time teams and given user stories to work on.

We were expected to take part in multiple-week scrum sprints with their full-time teams and join daily standup meetings to update their managers about their progress. The user stories and sprints may vary drastically between different full-time teams. The interns also participated in multiple intern events, such as educational, networking, and conference events as part of the program.

On the last week of the internship, the application development interns participated in an Expo event where they presented their internship experience and work to leadership. They were seated at booths within their respective intern groups and gave short 5- to 10-minute presentations to onlookers.

4 Internship Work

My full-time team was mostly working on backend applications. We worked in two-week sprints, where each person was assigned multiple user stories based on the stories' workload. Story workloads in our team would be separated into 3-, 5-, and 8-point stories, where 3-point stories would take a few days to complete while 8-point stories would take up both weeks of the sprint. This logic is not expected to apply to interns, however, as we lack the expertise and experience of full-time employees. The five interns in my team were assigned user stories that were composed of code on the working application. We were split between three mentors, who were responsible for guiding and helping their respective interns in completing their user stories in a responsible manner.

4.1 Work Overview

The repositories of all the company's applications are hosted on Azure Dev Ops (ADO). Our team would use Git to pull the application code from the master branch into a separate branch on our local machines. I used Intellij IDEA as my integrated development environment (IDE) to edit and write code. We used Java and Spring Boot as the backend framework. We are also expected to test and verify that our changes are valid by running the application locally and using testing services such as Postman.

4.2 Work Relevance

Our team was focused on updating and maintaining the current application. I was assigned user stories on removing and replacing deprecated and old API endpoints. APIs are code that allow two software programs to communicate, and the endpoint is the part of the code that connects them together. As an application updates and evolves, certain endpoints are replaced or become unused. Our team was responsible for the ‘cleanup’ of an application that managed bookings and shipments.

4.3 Work Timeline

As our software development life cycle (SDLC), we used two-week scrum sprints. A sprint is a fixed period of time where a team expects to complete a project or multiple tasks. For our team, each sprint would begin with a refinement and planning meeting where team leads will assign user stories or tasks to team members. Team members are expected to complete their stories within the sprint and communicate with their manager if they run into blockers preventing them from doing so.

After completing and testing the changes of a story, we can push our branch to ADO and create a pull request to master, where our code is run in a build pipeline. If the build pipeline runs successfully and the code is approved by two or more required reviewers, the pull request will be completed. The code is then automatically run in a test and development environment, where it will go through another verification process by the system. If tests fail or merge conflicts happen at any point of the process, it is our responsibility to review and fix our code. Installation is the final process of deploying the changes to the working product, updating the application’s infrastructure. I was not involved in any installation process with my full-time team.

Each of our sprints ends with a team retrospective meeting where each full-time team member and intern has a chance to discuss what happened during the sprint. We would say what went well and what did not go well, acknowledge team members, and suggest things that we can improve on.

4.4 In-House Training

The first week of the internship started with an orientation where we were given a tour of both the intern office and main corporate office. In both offices we were assigned hotel desks and given a work laptop to use as our local machines. We were also given a self-paced onboarding training guide to introduce us to all the required software programs and how to install them. The training gave a quick overview of the software system.

In the following weeks, we were given work that consisted of updating and maintaining the working product application. Outside of occasional online training sessions, most of our internship training was done as one-on-one meetings with our mentors. Mentors were expected to instruct and guide their interns in completing their stories, as well as working with them to get what they need.

We also attended leadership conference meetings, where we listened to various leaders within the organization share their past experiences. Other meeting sessions taught us more about J.B. Hunt as a company, such as its values and the workings of other units within the organization. Intern events served more as a networking opportunity rather than for informative purposes. Those events included a pickleball tournament, an intern field day, and a mocktail party.

4.5 Work Tasks and Results

I was assigned three stories over the course of the two-month internship. Two of them were on removing multiple unused endpoints while the last one was on replacing an old endpoint. However, the tasks were not as simple as removing and replacing a couple lines of code. Each endpoint had multiple dependencies and test code, which would break the application if not properly considered. Test code mainly consisted of unit tests, which verifies if a single method works as intended, and integration tests, which verifies if multiple methods work together as intended.

For my first 5-point endpoint cleanup story, I had to remove four endpoints from one application and one endpoint from another application. One endpoint was particularly larger than the rest, which had more than forty integration tests. I had to carefully remove those tests in an integration test class with more than 14,000 lines of code. The endpoint also had client and controller classes that had to be modified to implement the change. For the rest of the endpoints, it was similar but on a much smaller scale, with some of them only having one or two tests. After making sure that the application runs successfully, I did the necessary testing and pushed my branch to ADO to submit a pull request. Including the initial setup of the application and required environment settings, completing this story took about three weeks. Since the required code reviewers were busy, approving the story took even longer.

My second cleanup story was only a 3-pointer. It was similar to my last story, so I managed to complete it in two days with minimal help from my mentor. It was a much smaller process, so the pull request was reviewed and approved within the week as well.

My last story was an 8-point endpoint replacement story. I had to create a new client class to call the replacing endpoint, as the client class that called the old one had other endpoints that were still in use. After writing the client class, I had to go through every reference to the old endpoint and change them to call the new endpoint. These references were in implementation and test classes. I also had to create a mapper class that would map the values from the old endpoint to the new endpoint. The application ran successfully on my local machine, but there was a local issue that prevented me from testing my changes. My full-time team had to take over the story, as there were issues outside of my control that lasted until the end of the internship.

4.6 Work Evaluation

As interns, we are responsible to make sure that the quality of our code is acceptable. Our code must follow certain standards of the company, such as a standard naming convention for our classes and methods. Our work is evaluated first by our mentors, which is further verified and tested automatically by the system as well as by code reviewers during the pull request. Since there are multiple layers of testing, it is almost impossible to cause a fatal error in the final production code.

5 Technical Products

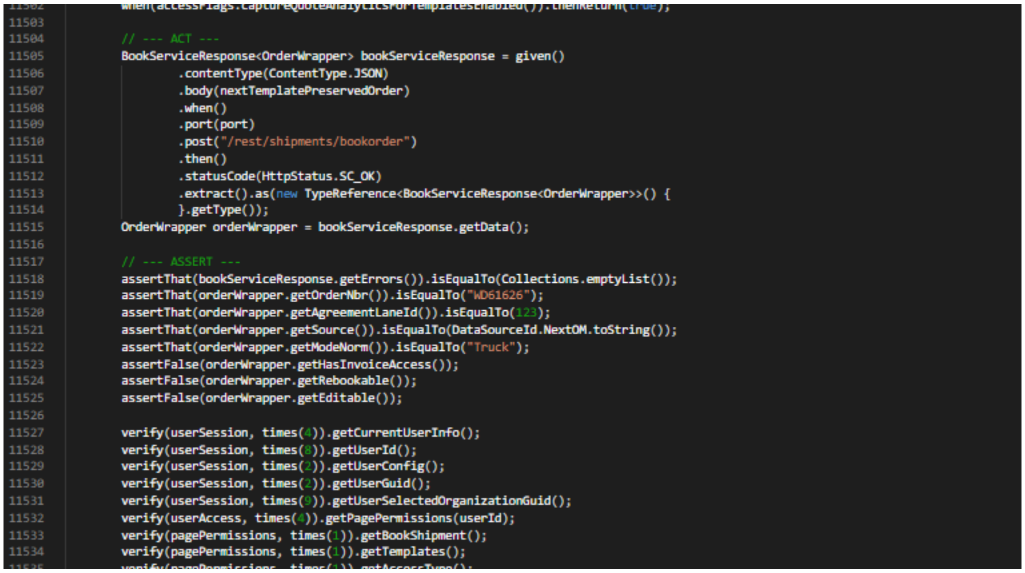
Our tasks were not intern-specific projects, but rather we were assigned stories and worked alongside other software engineers on the applications of J. B. Hunt. My stories were all under the 360 Shipper, a self-service shipment application available to businesses.

5.1 Design Description

My stories were on cleaning up and replacing old endpoints in various classes within 360 Shipper, so there was no specific design I had to follow. I only had to siphon through existing code and carefully change what was required. Some of the classes I had to work with had over 10,000 lines of code. These included client, controller, and test classes, which all depend on each other to work successfully.

5.2 Code or Finished Product

While removing and replacing the endpoints, I had to do the same to their dependencies. One of their dependencies are the integration test classes, which look like the following code:

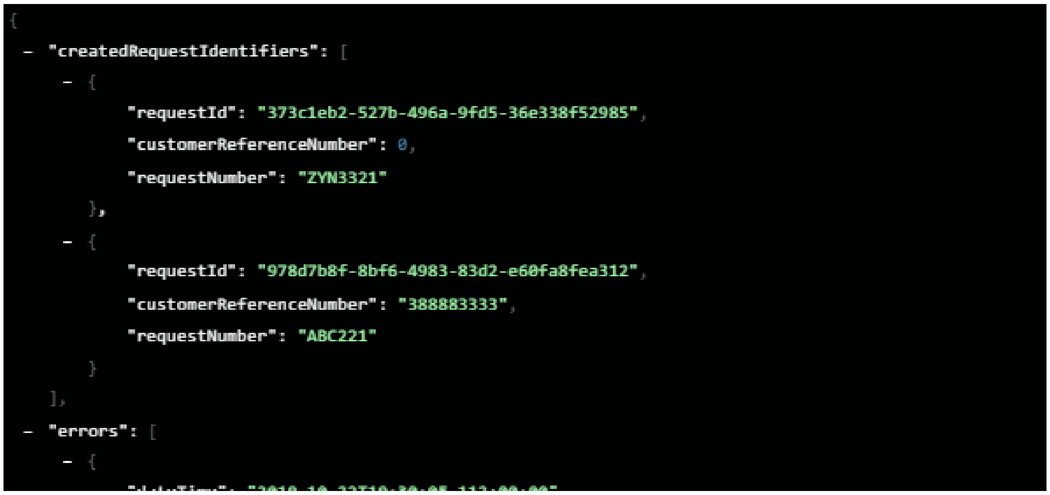


```
        when(accessFlags.captureQuoteAnalyticsForTemplatesEnabled()).thenReturn(true);

        // --- ACT ---
        BookServiceResponse<OrderWrapper> bookServiceResponse = given()
                .contentType(ContentType.JSON)
                .body(nextTemplatePreservedOrder)
                .when()
                .port(port)
                .post("/rest/shipments/bookorder")
                .then()
                .statusCode(HttpStatus.SC_OK)
                .extract().as(new TypeReference<BookServiceResponse<OrderWrapper>>() {
                }.getType());
        OrderWrapper orderWrapper = bookServiceResponse.getData();

        // --- ASSERT ---
        assertThat(bookServiceResponse.getErrors()).isEqualTo(Collections.emptyList());
        assertThat(orderWrapper.getOrderNbr()).isEqualTo("WD61626");
        assertThat(orderWrapper.getAgreementLaneId()).isEqualTo(123);
        assertThat(orderWrapper.getSource()).isEqualTo(DataSourceId.NextOM.toString());
        assertThat(orderWrapper.getModeNorm()).isEqualTo("Truck");
        assertFalse(orderWrapper.getHasInvoiceAccess());
        assertFalse(orderWrapper.getRebookable());
        assertFalse(orderWrapper.getEditable());

        verify(userSession, times(4)).getCurrentUserInfo();
        verify(userSession, times(8)).getUserId();
        verify(userSession, times(2)).getUserConfig();
        verify(userSession, times(2)).getUserGuid();
        verify(userSession, times(9)).getUserSelectedOrganizationGuid();
        verify(userAccess, times(4)).getPagePermissions(userId);
        verify(pagePermissions, times(1)).getBookShipment();
        verify(pagePermissions, times(1)).getTemplates();
        verify(pagePermissions, times(1)).getAccessType();
```

The code above tested and validated the code that was responsible for getting booking requests from the customer. If I modified the code that did the booking, I would have to modify the code that validates it as well, otherwise it would raise an error for something that was removed or replaced.

5.3 Sample Output

After making my changes, I had to test that the application still works as expected. As my work was backend instead of frontend, testing the output can only be done through endpoint testing. I used Postman to do endpoint testing, by making a mock HTTP request to the application. If the code and Postman request are both correct, we expect a response that looks like so:



```
{
 - "createdRequestIdentifiers": [
    - {
          "requestId": "373c1eb2-527b-496a-9fd5-36e338f52985",
          "customerReferenceNumber": 0,
          "requestNumber": "ZYN3321"
      },
    - {
          "requestId": "978d7b8f-8bf6-4983-83d2-e60fa8fea312",
          "customerReferenceNumber": "388883333",
          "requestNumber": "ABC221"
      }
   ],
 - "errors": [
    - {
```

6 Professional Growth

During the two-month program, I was strongly faced with different challenges and had to adapt and grow to push through them. By doing so, I gained many new technical and management skills. Both peers and mentors helped me in this period of time, which I would not have been able to do the same alone.

6.1 Challenges

As this was my first internship, I was not immediately familiar with both industry and J. B. Hunt-specific standards and procedures. Even though I would look for information online, I could not find solutions for many problems as they were company specific. I had to ask questions to bring myself up to speed, even towards the end of the internship. In addition, my full-time team was temporarily working with another team's application and was not entirely familiar with it. Some issues that I encountered had to go through a longer hierarchy process to get fixed.

Our mentors were still working as usual, so they would often be busy with their own duties. When I was unable to figure out an issue, I would sometimes have to wait a couple hours or until the next day before I could get on a call with my mentor. While waiting, I would have work on another story or intern-related work, dividing my focus on the current task.

6.2 Technical Skills Acquired

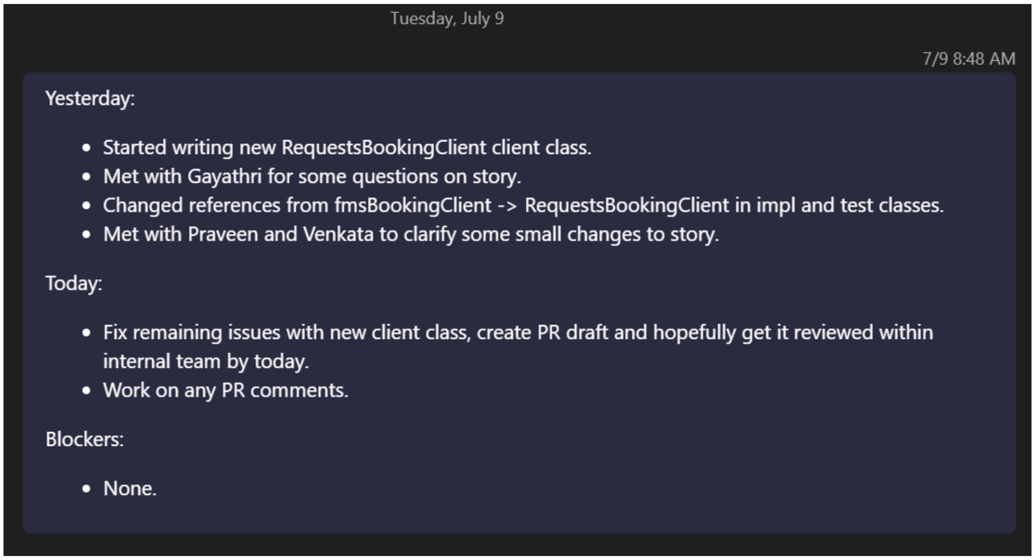
My team worked almost entirely on the backend side of things, so I mainly used Java as the programming language while utilizing Spring Boot and Apache Maven to install dependencies and build the application. The company kept their code repositories for their applications on Azure DevOps, so I had to familiarize myself with the system. I practiced a lot of Git as my version control system, cloning, pulling, and pushing code to and from Azure DevOps so that I can collaborate with other software engineers.

On my local machine, I used Intellij IDEA as the integrated development environment (IDE). I was regularly using Rancher, a substitute of Docker, as the containerization software to get all the necessary dependencies for my local machine. I used Postman to mock some endpoint HTTP requests testing, but due to authorization issues and code freezes my experience with it was minimal.

6.3 Management Skills Acquired

Between intern events, my assigned stories, and online/in-person meetings, I had to manage and report my schedule through written standups every morning. Some events and meetings would take up to a few hours or occasionally the entire day, so progress on my stories could be affected significantly. I needed to adapt to unforeseen issues in my work, such as blockers and external problems that were out of my control. At certain points during the internship, I had to work on two stories at the same time while my mentor was finding a solution for problems out of my control.

I communicated with other individuals and teams through Microsoft Teams. I would reach out to my mentor either in person or through a Teams chat or call if I require help. Outside of Tuesday and Thursday standups with my full-time team at 9 a.m., we had daily written standups in a Teams channel where we would update our manager on our work progress. It includes a clear but short description of what we did the day before, what we plan and expect to do today, and any blockers that are preventing us from progressing. I am expected to deliver whatever I promised to do in my written standup and report any legitimate reasons that I was unable to do so.



Tuesday, July 9

7/9 8:48 AM

Yesterday:

- Started writing new RequestsBookingClient client class.
- Met with Gayathri for some questions on story.
- Changed references from fmsBookingClient -> RequestsBookingClient in impl and test classes.
- Met with Praveen and Venkata to clarify some small changes to story.

Today:

- Fix remaining issues with new client class, create PR draft and hopefully get it reviewed within internal team by today.
- Work on any PR comments.

Blockers:

- None.

We also had a standup session for the interns, where we would send one representative from each intern team each Monday, Wednesday, and Friday at 8 a.m. to update our intern manager on our work. It was also our chance to speak up if we had any issues with our full-time team and needed our intern manager to work with our full-time manager.

6.4 Lessons I Would Apply to Next Internship

Collaboration is one of the best resources I used during this internship, and I would hope to be able to do the same in future internships/workplaces. By working with talented individuals who had more experience than myself, I managed to learn more in two months than I could ever by working alone. Connecting with my mentor and other interns also helped me expand my network.

Networking with both peers and leaders helped me build and strengthen relationships that will last for the rest of my career. Talking to different people helped me see things from different perspectives and develop new ideas. Discussing problems with my mentors also gave me great feedback that can help me improve my abilities. It also makes the workplace a more enjoyable environment to complete my duties.

7 Summary/Conclusions

Internships are valuable work experience to students entering the workforce for the first time. I was fortunate enough to land an internship at J. B. Hunt, a leading company in its industry. My experience there was what I expected and more, meeting amazing people and working with real industry applications.

In J. B. Hunt, I got the opportunity to update and maintain the code with hundreds of other software engineers in the company, by using collaborative tools such as Azure DevOps and Git. I already had years of experience with Java as a programming language, but working with it in real industry code was a great learning experience for me. I also developed personal relationships with employees and other interns, which helped expand the network of my career.