Infosys:

**SDLC requirement gathering and Agile :** The client was APLL based out of flagstaff AZ. We used to design backend codes and front end modules to help

1. First customers look for PO numbers and sizes.
2. Place orders
3. Arrange local shipping until port
4. Cargo weight procedures
5. Container placement and vessel loading
6. Port of departures and port of entry (middle ports )
7. Customs clearance
8. Labels printing
9. Local shipping into warehouse

They use to be multiple child customers like Walmart, Nike, TJMax, TKMax, Adidas, and all major cloth and shoe companies. They wanted the process to happen as they wanted like Nike wanted the warehouse which was building the product to take care of shipping from warehouse to port. Adidas wanted to make a quality check before being taken out of warehouse origin. So we used to gather requirements from clients make the back end and front end modules work accordingly.

**Front components usage :** front end components were built using JavaScript, HTML and CSS. I used SASS to override the default styles like margins, colors, font-weight etc. I have designed booking modules where I get the information from backend of the purchase orders that they have chosen 🡪 create a booking order and make the insertion into the booking table in the DB with their account. To build these pages I have used DHTML concepts.

**Class Diagrams Scenario :** Here I would Like to quote a scenario which I faced when we were on boarding BOBCAT customer. The deal with machinery and equipment so their business was a little different from regular. So, we have to do sit with client and discuss their business operations and their expectations and then we have to figure out what routes they needed, what db’s they needed, how their front modules should look. For backend I wrote several class structures and corresponding UML diagrams as well.

**JSP and Servlets with Development tool Eclipse IDE 3.4 :** There were about 10 customers when I first joined the project and it was a pure Java web based project. All the calls were made using servlets. After I joined we onboarded TKMax and we developed for them in this as well. Mostly everything was developed using JSp’s.

**SpringMVC framework :** Initially customers like Nike, Walmart were pure Java based development. But then when we decided to move the project to Spring. All the customers after that like TJMax, H&M were coded using spring MVC and hibernate to fetch data. Although we did not migrate database. As a test we did it first for TJMax customer since it required less modifications from other customers.

**ORM Mapping :** Initially for one customer New Balance they were producing or manufacturing inside the US so their processes were limited and hence less number of tables and hence we used ORM mapping to map the order booing details in and out of tables. But later it was moved to hibernate as well to unify the process.

**IOC and dependency injection :** We have used both setter and constructor injection in order to inject objects. When you are dealing with cargo loading, we need the booking number, booking order weight and many such information. So, we create a object there use IOC or dependency injection and create cargo label and make an entry into freight cargo table.

**WebTier:** We built servlet links for all clients and maintained the session data of the users logged in using WebTier. It is just a presentation layer. Helper methods like DAO layer classes are written in order to make smooth transition of data from database to objects. Helper methods can also be used to in order to write complex methods inside a service class.

**SpringAOP:** I used Spring AOP to make logs statements. It was a legacy application and quite a big one and hence issues would come every now then. We used study logs and confirm what is causing the issue and thereby we did maintenance activities in the maintenance window. Also while we were developing modules it was really helpful to debug and find mistakes especially while on boarding bobcat customer with almost different backend routes compared to other clients.

Once you develop RESTAPI’s then you need to integrate front-end calls with those calls. Before that check the API calls in postman and see if they are sending in correct data (json).

**REST and SOAP:** All the clients were using REST API service in order to make API calls and get the data from where it is transferred to front-end components.

**Hibernate:** We used Hibernate ORM for relational mapping between tables and model classes. For all the clients we have used the same but in my future project I have used more advanced version ORM version called as JPA. We also used to write Junit test cases for every request mapping to confirm if we are getting the data or not.

**WebSphere Server:** The whole application was deployed in WebSphere. I have not deployed on my own but a part of the team was in charge of it and they would notify us what happened with the deployment. But as team we were responsible for deployment.

**ANT deployment:**

**LDAP authorization:** It is a username, password validation technique. We ask users for either username, email, employee number and based on that we store a DN(distinguished name) and password of every user. Then when user wants to login 🡪 we search using user information to get DN using search in directory 🡪 using DN and password validate the user.