INVENTORY MANAGEMENT SYSTEM (IMS)

QA INDIVIDUAL PROJECT 1: DANIEL AHUCHOGU

INTRODUCTION

Objectives:

- **To create a functional IMS command line application that takes the users inputs to give specific outputs. While connected to a GCP my SQL instance or a JDBC.**
- **The development of the application using supporting tools and technologies learnt throughout the duration of the course.**

Aim:

- **To build an application that a user/client can interact with via a Command Line Interface (CLI)**
- **The application has a JAVA codebase, with a sensible package involved.**
- **Uses the agile framework during the duration of the project.**

CONSULTANT JOURNEY

- **The main technologies learnt and used for this project were:**
  - **Git, SQL, Jira, Java**
- **Git is a version control system, where it can track changes throughout the project with the Involvement of GitHub.**
- **SQL/GCP with SQL: is a database management system, to manage data within tables.**
- **Jira: A project management system, to have the story points for all the user stories and requirements.**
- **Java: A back end build tool software. Main purpose is an object orientated programming language that is class based.**

INITIAL PROJECT MANAGEMENT

- **Before the project started I had an initial plan of what needs to be done related to the priority**
- **I had mock list of the user requirements, user stories and important tasks that I had to do.**
- **The design of my entity relationship diagram related to their relationship links.**
- **A risk assessment of complications and unplanned events that could occurred throughout the project.**
- **Remembering an agile mindset throughout the duration of the project.**

Backlog 15 issues

Create sprint ...

Risk Assessment	DA IP1I-1	
Planning the layout of SQL commands that will be implement with Java.	DA ~~IP1I-4~~	
As a user I want to add information related to the customer, so I can view them in the database/command line.	IP1I-10	5
As a user I want to use the command line, so I can view information related to the customers in the database.	IP1I-11	5
As a user I want to use the command line, so that I can delete any existing information to the customer details that are wrong.	IP1I-13	5
As a user I want to be able to use the command line, so that I can add items or products to the system.	IP1I-14	5
As a user I want to use the command line, so that I can update any existing information related to the customer	IP1I-12	5
As a user I want to use the command line, so that I can view any existing information related to the items or products.	IP1I-15	5
As a user I want to use the command line, so that I can update any existing information related to the item or products.	IP1I-16	6
As a user I want to use the command line, so that I can delete any existing information related to the item/product that are wrong.	IP1I-17	6
As a user I want to recall information from the item and customer within the command line. So, that I can create an order in the system.	IP1I-18	5
As a user I want to recall existing information from the system related to the item and the customer. So, that I can delete orders that are incorrect in the system.	IP1I-23	4
As a user I want to recall existing information from the system related to the item and the customer. So, that I can view orders in the system.	IP1I-19	4
As a user I want to use the system recall an existing order that has been processed. So, that I can add an item to the order.	IP1I-20	5
As a user I want use the command line system. So, that I can calculate the total cost of an order.	IP1I-21	4
As a user I want to recall information within the system related to the order. So that I can delete an item/product from a processed order.	IP1I-22	-

Fig 1: A list of my user stories and user requirements from my Jira backlog

ENTITY RELATIONSHIP DIAGRAMS

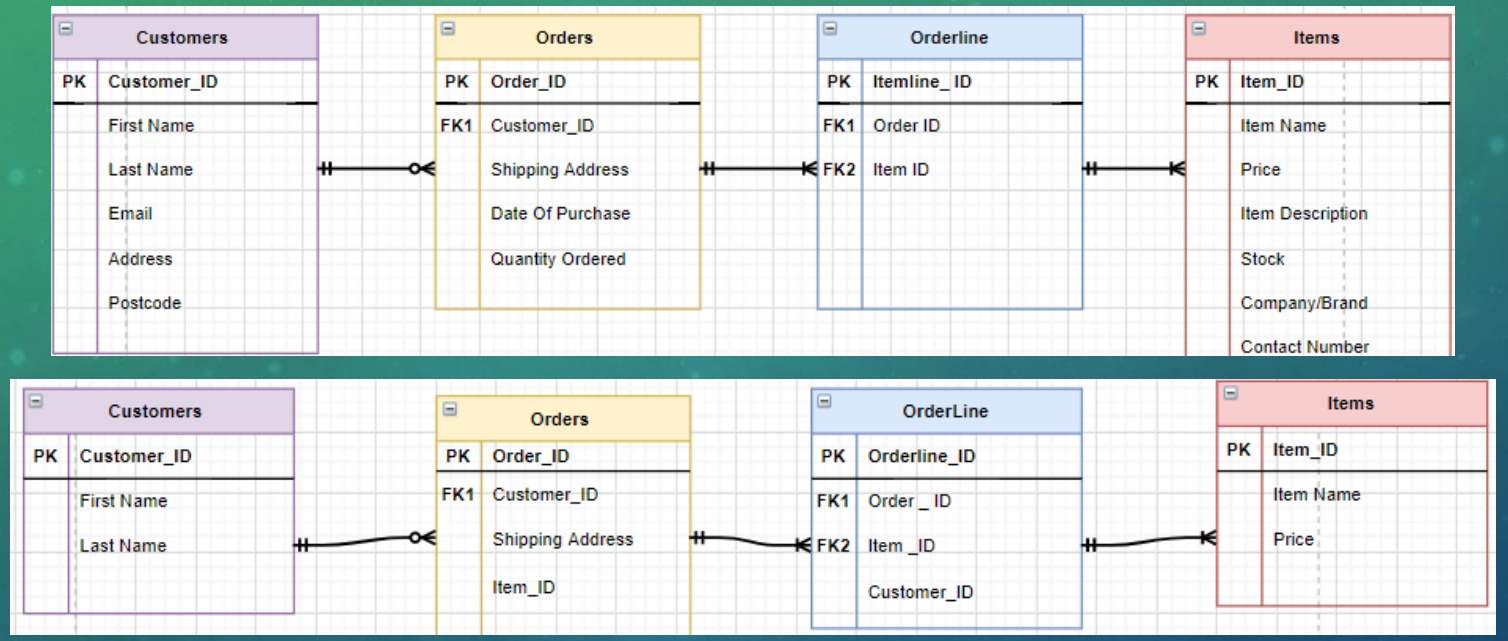

Fig 2: A before and after image of my ERD diagram

- **Entity relationship diagram (ERD) shows the relationships of entities stored in a database.**
- **In this context, the data that needs to be collected is showed above. This ERD models the key information needed for this database.**
- **We use this model to plan the design of the database. To prevent many –to – many relationships we have intermediate table to handle this.**
- **Due to the fact an order can contain 1 –to –many items and an item can be part of 0 – to – many orders.**

PROJECT MANAGEMENT REVIEW

- From the last slide, that was mock project management that I did before looking at the spec.
- After seeing the specification again and the IMS template. Another scrum meeting and project review had to be done.
- The backlog was updated and an active sprint was done. To outline the tasks and important user story tasks.
- An active sprint was started involving MoSCoW. Which help outline key prioritisation.

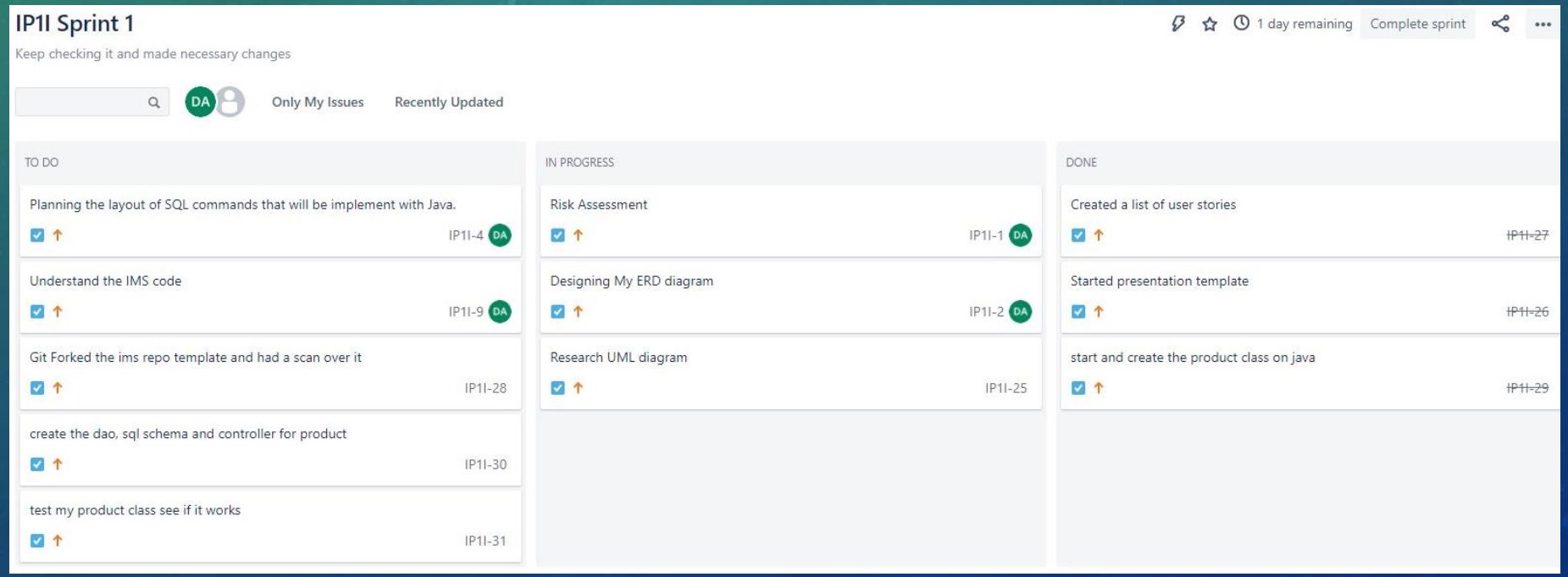

Fig 3: My Jira active sprint Kanban board after my first individual scrum meeting

CONTINUOUS INTEGRATION

- With version control, I approached it as simplest as possible to prevent errors with my code/working.
- Focused working off my DEV branch, to make different feature branches that had different aspects of my work.
- Essentially I didn't want work on my main/master branches.
- Want to minimised has much errors and merged conflicts as possible.
- Due to different features are combined, or integrated, frequently occasionally to expose external coding issues.
- I had multiple branches, because during development you might stop working on a piece of work and restart it on a different branch and works better..

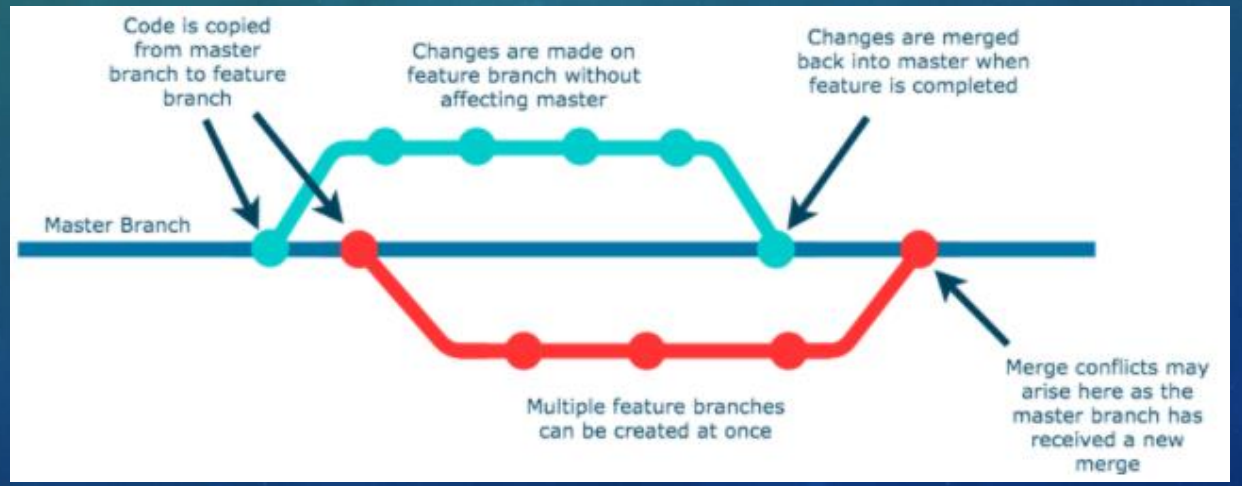

Fig 4:Branch model that I followed

TESTING

- In terms of testing, it's important to test the quality of the code. This means less chance of the code breaking. Helps determine and identify possible errors.
- Need to test the functionality of the code and the coverage the code can take. To see if it meets our expectations or initial prediction .
- JUnit or Unit testing can be an automated process designed to reuse code and provide results after a test.
- From the JUnit test, I had a coverage of 61.6% on the src/main/java folder.

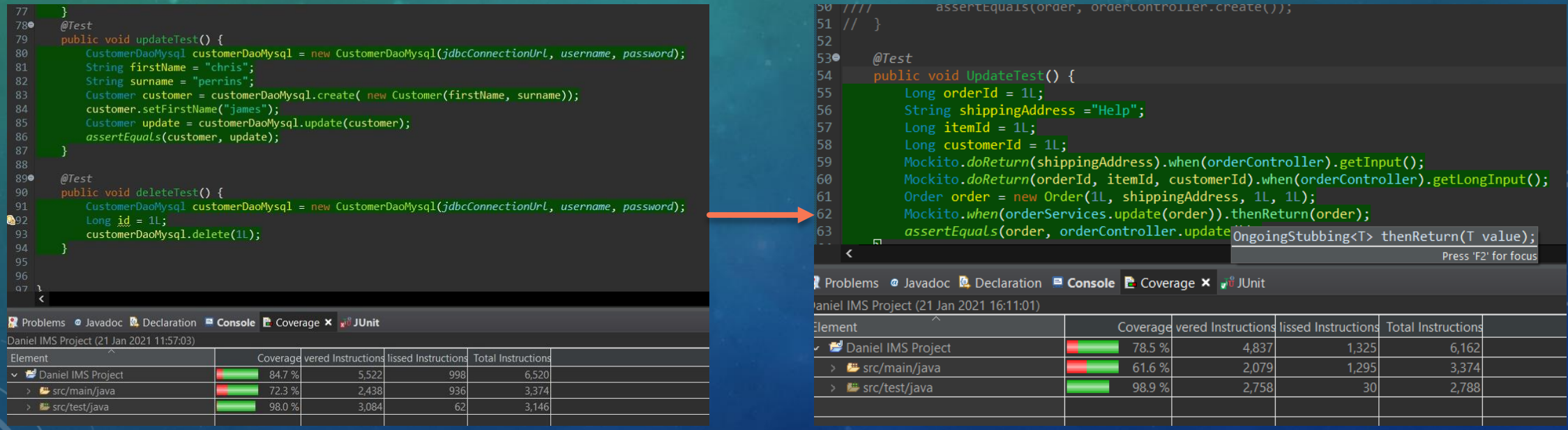

Fig 5: Before and after images for coverage whilst doing test coverage

DEMONSTRATION OF THE PROJECT

- **With some of the user stories I am going to show a piece of my project that is working. After using a Maven Build tool.**
- **Through a Command Line System, because essentially that was the Minimum Value Product (MVP).**

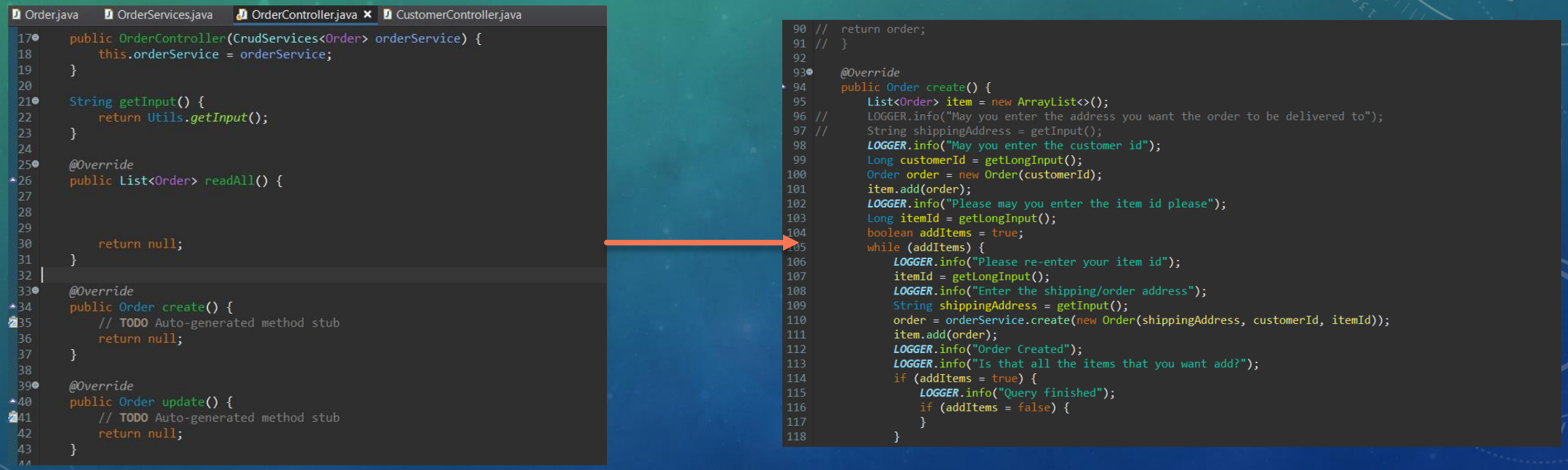

Fig 6: Before and after images for developing an order method for create

UML

- **A UML diagram shows the relationships between the different classes, attributes, method and relationship between different objects in a java program.**
- **UML is showing the basic blueprint of the programme and the design of it.**

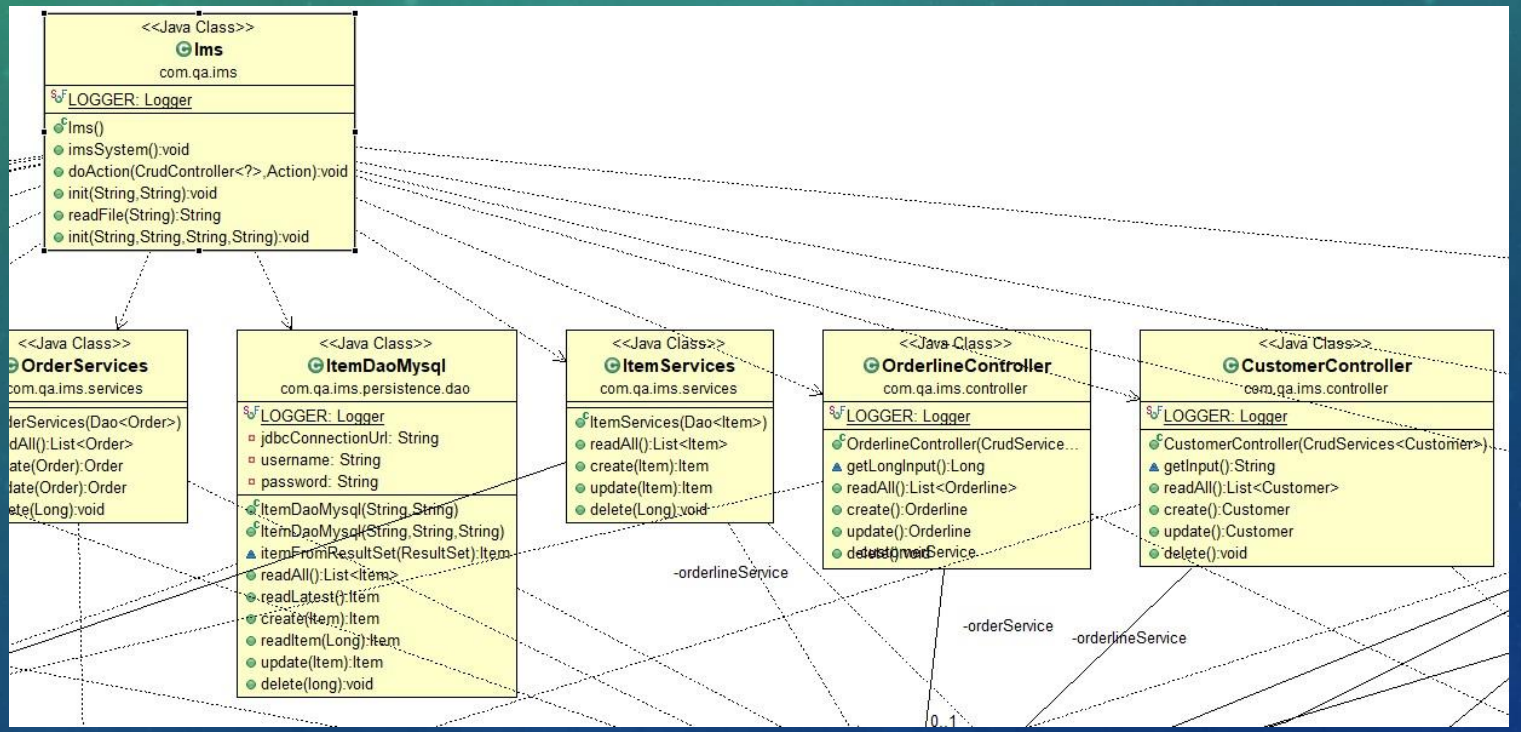

Fig 7 :A section of my UML diagram

SPRINT REVIEW

What did you complete:

- **In general the CRUD functionality for customers, orders and items.**
- **Connects to a local MySQL instance.**
- **Testing and coverage testing.**
- **Git repository with the feature branch model, with a working jar file.**

What got left behind:

- **Calculate cost for orders**
- **Relationship between orders and orderline.**
- **Some aspects of my testing. For example DaoTest is not completed to a high standard.**
- **Connecting to a GCP instance to see if the application works.**

SPRINT RETROSPECTIVE

What went well:

- **What went well was the development of items by implementing CRUD and SQL functions with Java.**
- **The creation of my UML diagram, ERD and some aspects of project management.**
- **Certain aspects of JUnit testing and how I handled my cloud integration/version control.**

What could be improved:

- **Some of my methods/code for orders and orderline could be done better to have the correct relationship between them.**
- **Spending more time on my JUnit test to solve, assertion errors and other type of errors/failures to increase coverage.**
- **Test if my code works on a local GCP earlier.**
- **My README file could be edited and be written more effectively.**

SUMMARY

- **To reflect on the project, as a whole the project went well to a certain extent.**
- **There were certain challenges and risks that occurred.**
- **At start the start I was very overwhelmed of the scale of the project because I had no knowledge of any coding such as Git and Java.**
- **But with an agile mindset I was able to adapt to the situation and I was able to work effectively.**
- **Able to achieve most of what the specification required/the MVP**

Final Thoughts:

- This was a tough project for someone who has a no prior coding experience.
- This is a learn step for me to gain new skill/developed new skills that I have required.

QUESTIONS ?