



Fundamental Project: Inventory Management System

BY MOHAB KHALIFA

Introduction

► Who am I?

I am a QA IT Consultant Trainee. I have completed my first project with QA. In order to achieve this, I have learnt how to use Java, MySQL and Maven to a competent level. I faced many difficult hurdles during the project, but ultimately conditioned me to become a better coder.

► How did I approach specification?

Primarily, I went through the specs of the project to get to grips with all the technologies I need to use in order to complete the project on time. My next step was to build ERDs (entity relationship diagram) and UML diagrams (unified modelling language) in order to create some structure when building my classes. I also created a Jira project in order to follow my plan through a 'sprint'. I turned the domain specs into user stories as my aim was to integrate them into the project in order to link each branch and commit to a user story. This will allow me to keep record of what I completed and what is yet to be finished.

Consultant Journey

- What technologies have I learnt for this project?

Eclipse – this is an IDE (integrated development environment) used to programme in java. I used this to develop my source code and the tests for the fundamentals of the project.

Maven – this is build automation tool used primarily for Java projects. I had to use this in order to package the application in a jar. file so that it can run as a executable without access to the source code, so that I can share it with the trainer.

Git – this is a version control system I used to implement changes to the source code.

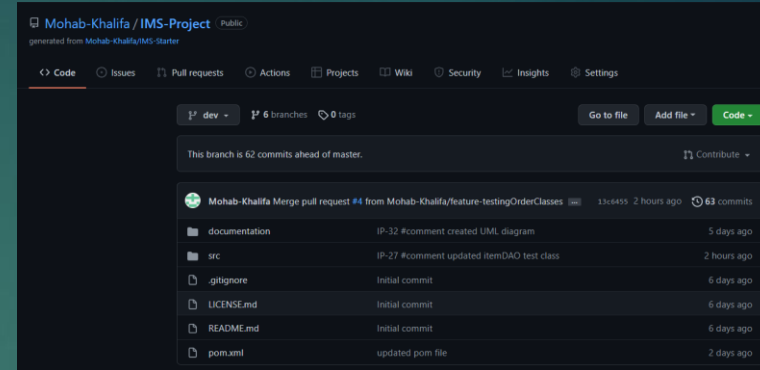
Github – this allowed me to implement changes with the source code in a remote server so that I am able to implement changes from anywhere.

MySQLWorkBench – this is a visual database design tool that integrates SQL development, admin, DB design, creation and maintenance into a single integrated development environment. I wrote my source code in the sql-data.sql and sql-schema.sql files in order to create the ims database and tables.

Continuous Integration

► How did I approach Version Control?

Git/Github – Throughout the development of this project, I tried to maintain continuous commits to update my changes consistently. I made sure that the new updates were on feature branches so that it does not main the source code of the whole programme. Originally, when I forked the repo, I instantly branched off the main branch, to create a dev branch. I then branched off the dev and incorporated the feature branches which I then late merged to the dev. Finally once dev has working source code that is not broken, I merged it into the main branch.



```
Mohab@DESKTOP-FSD9KM3 MINGW64 ~/eclipse-workspace/IMS-Project (dev)
$ git status
On branch dev
Your branch is up to date with 'origin/dev'.

nothing to commit, working tree clean

Mohab@DESKTOP-FSD9KM3 MINGW64 ~/eclipse-workspace/IMS-Project (dev)
$
```

Testing

► What was tested?

I tested all of my source code which resides in the src/main/java folder.

ava (3 Apr 2022 12:27:34)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
IMS-Project	73.8 %	3,008	1,070	4,078
src/main/java	60.7 %	1,550	1,002	2,552
src/test/java	95.5 %	1,458	68	1,526

My aim was to achieve at least 80% coverage, however I managed to reach 60% coverage. I struggled to get over the 80% benchmark as I had issues with my build and runtime as they were not compatible. This caused the EqualsVerifier to fail on all classes.

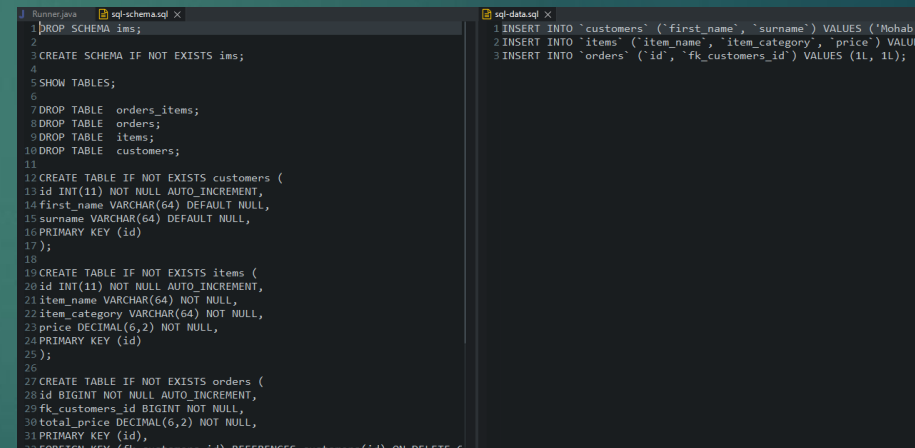
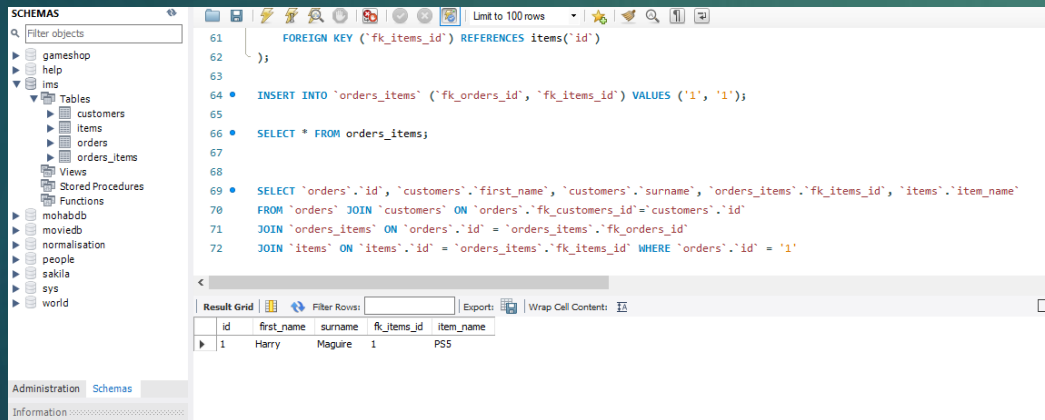
```
@Test
public void testEquals() {
    EqualsVerifier.simple().forClass(Customer.class).verify();
}
```

```
java.lang.AssertionError: EqualsVerifier found a problem in class com.qa.ims.persistence.domain.Customer.
-> Unsupported class file major version 61
```

This hindered me from achieving 80% and took a lot of time trying to fix the issue, however, unfortunately I wasn't able to figure out the issue.

MySQL

I managed to integrate MySQL with my Maven project through the sql-data and sql-schema files. Thus I can store data into MySQL when interacting with the CLI.



Sprint Review

- ▶ What did I complete?

I managed to achieve all of the user stories in the specification. I managed to an application that can create/read/update/delete customers/items/orders. My application can interact with an end user via CLI (command-line interface).

- ▶ What got left behind?

I struggled with testing in general. The EqualsVerifier was not working due to version issues. Additionally, I had a few method tests that returned null objects and I was unsure why. After spending a lot of time fixing bug issues through debugging, I was still unable to fix every issue I had with testing. Ideally, I would've liked to achieve 80% coverage, however was able to reach 60%.

Sprint Retrospective

► What went well?

Creating the item classes (Item, ItemDAO, ItemController) was relatively straight forward. As the forked source code had already pre-built customer classes, I used them as a template to create my item classes. Also implementing the CrudController and Dao interfaces to the classes allowed for easier application to the classes.

► What could be improved?

Possibly create a better plan in the future. I didn't follow everything I aimed to do in the classes. Create more of a flexible plan as I didn't realize all the issues I would come across during this week.

I need to improve on my testing, especially the OrderDAO and OrderController classes. As I ran into failures from testing, I expected to receive a complete object, but unfortunately receiving null objects a few times. Until now I am unsure as MySQL seemed to be working fine.

Conclusion

- Reflections on the project, future steps, any other relevant info

One of my main reflections is definitely time management for the project. To make sure I plan my time accordingly with some leeway to prepare for bug fixes throughout my project.

Also to manage my version control better in the future, creating smaller feature branches along the way.

Also taking breaks accordingly when working. At the end of the week, I felt mentally exhausted as I spent the whole week on my monitor, not taking as many breaks as I probably should have. I noticed when I did take the breaks, it cleared my mind a bit more and helped me identify issues quicker at times.



Any Questions?