Fundamental Project: Inventory Management System

By Peter Vaughan-Williams

Introduction

- Who am I?

My name is Peter Vaughan-Williams; a QA SDET Trainee learning Java and MySQL for the first time and properly utilizing my new found skills in these fields through this fundamental project. It was frustrating but very fun ultimately; leaving me desperate to learn more on what I could have done differently and the different avenues I could have taken.

- How did I approach the specification?

Initially, I ran through the specification to understand what was expected of me and from this project, to understand the deadline and what technologies I would need to incorporate throughout this project. The main planning steps I took were to look over the domain part of the specification to understand what the application needs to be able to do. I turned those into user stories for a Jira board I created and integrated into the project in order to link each branch and commit to a user story; keeping track of what I had completed and what I hadn't yet completed at the time.

Consultant Journey

- What technologies have I learnt for this project?

Eclipse

This was the integrated development environment (IDE) I used to develop both the source code and the tests for the fundamental project.

Git

The Version Control System I used to implement changes to the source code..

Maven

This was the build tool that I had to use in order to clean and package the application in order to get it to run as an executable without access to the source code for sharing purposes. Also used for bringing in dependencies.

JUnit

A dependency brought in using Maven. This was the unit testing technology I utilized for this project in order to write the tests and increase my line coverage throughout the project.

MySQLWorkbench Version 8.0.23

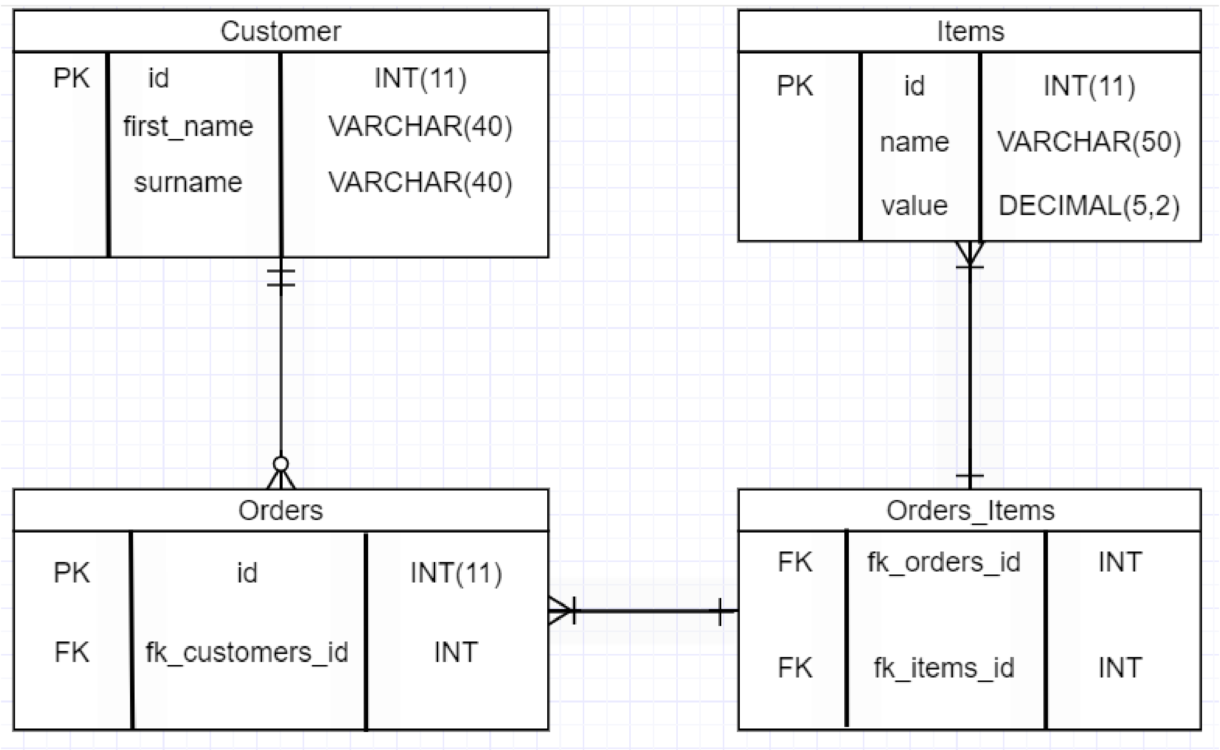
This was the visual database design tool I used to write the source code for my sql-data.sql and sql-schema.sql files; making use of data definition language, data manipulation language and data query language in order to create the database and tables, manipulate data inside the tables, and query the data using the SELECT keyword respectively.

SonarQube

Used to inspect my code and ensure it is written in a high quality, bug free way.

My Entity Relationship Diagram

- This is the Entity Relationship Diagram I created before I wrote the source code for the sql-schema in order to understand the relationships between the different tables and what parameters each one would need.



Customer		
PK	id	INT(11)
	first_name	VARCHAR(40)
	surname	VARCHAR(40)

Items		
PK	id	INT(11)
	name	VARCHAR(50)
	value	DECIMAL(5,2)

Orders		
PK	id	INT(11)
FK	fk_customers_id	INT

Orders_Items		
FK	fk_orders_id	INT
FK	fk_items_id	INT

As you can see, I use the crows feet notation to denote the mandatory and optional relationships as well as in which direction the relationship is a many to one relationship.

Continuous Integration

- How did I approach Version Control?

Git

Throughout the development of this application, I ensured that new updates to the source code would not destroy the whole program by utilizing the Feature Branch model. Hence, starting with the original forked repo for the starting point of this project, I branched off of the main branch into my own, custom dev branch. Thereon, anytime I wanted to create new functionality based on the user stories I had created, I would branch off of those, name the branch based on the key of my Jira board user story (for link and integration purposes), and then merge it back in to the dev branch once I know I have made progress.

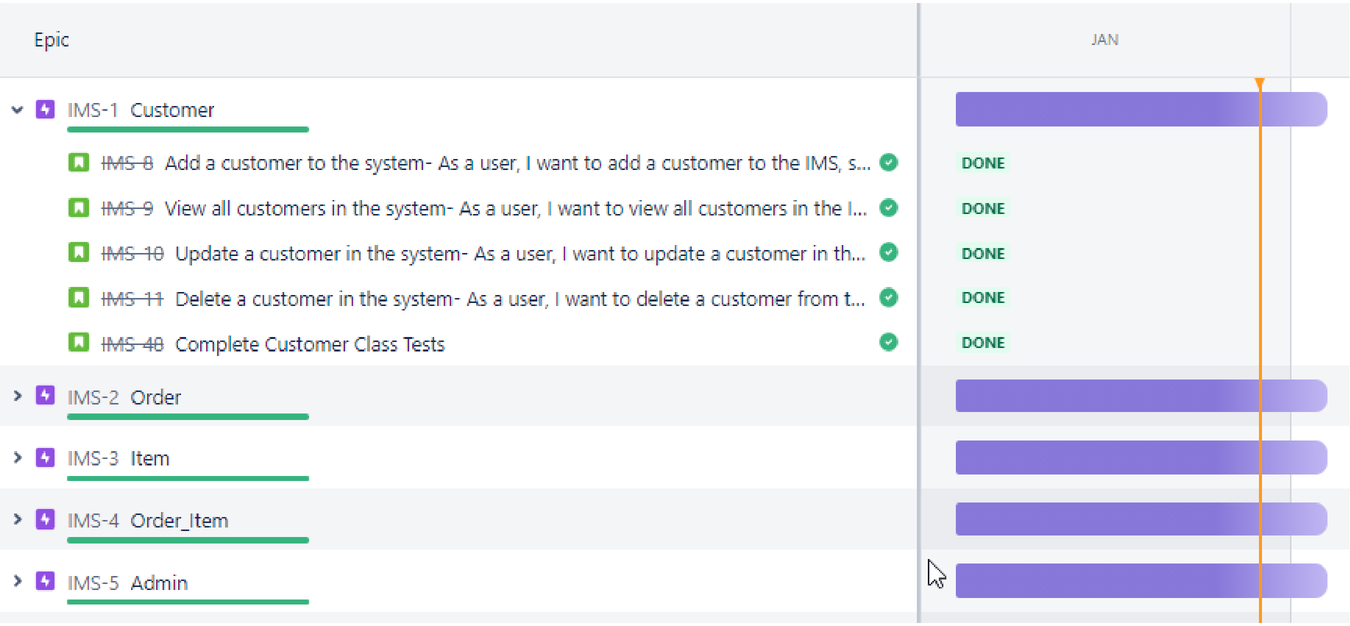
- *Example of the Version Control merging process.*
- Git checkout dev -> git checkout IMS-10_xxx -> (make changes) -> git add .
- (etc)

My Project Roadmap – Achieved in one sprint

Following:
As a []…

I want to []…

So that I can []…



Epic | JAN

IMS-1 Customer

~~IMS-8~~ Add a customer to the system- As a user, I want to add a customer to the IMS, s… DONE

~~IMS-9~~ View all customers in the system- As a user, I want to view all customers in the I… DONE

~~IMS-10~~ Update a customer in the system- As a user, I want to update a customer in th… DONE

~~IMS-11~~ Delete a customer in the system- As a user, I want to delete a customer from t… DONE

~~IMS-40~~ Complete Customer Class Tests DONE

IMS-2 Order

IMS-3 Item

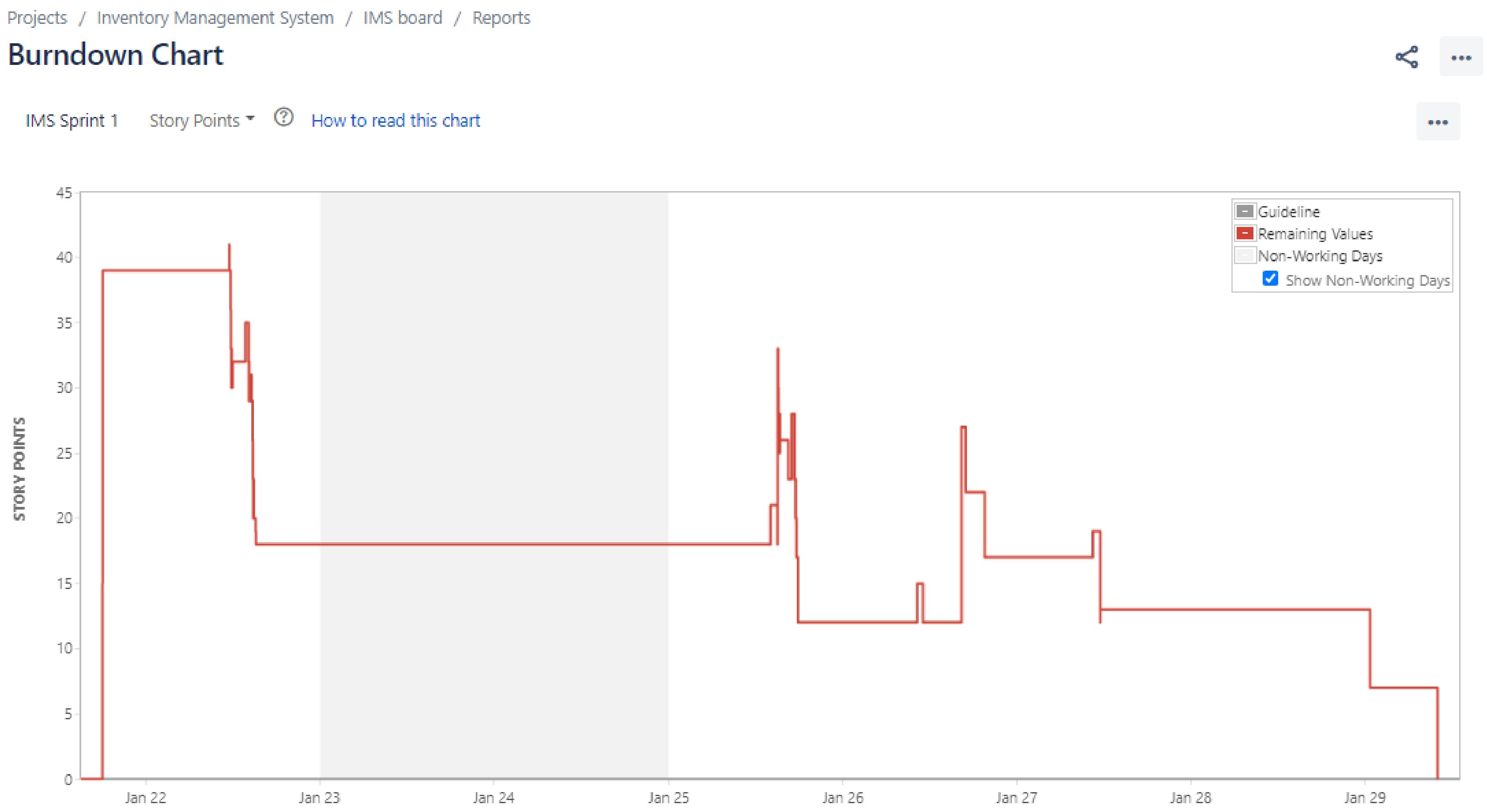
IMS-4 Order_Item

IMS-5 Admin

For each feature the specification asked of the application; I had written a user story to fit what exactly the feature needed to do and why. For example; Add a customer to the system- As a user, I want to add a customer to the IMS, so that I can keep record my customers details.

- Link: https://pvaughan-williams.atlassian.net/jira/software/c/projects/IMS/boards/4/roadmap

Burndown Chart



Projects / Inventory Management System / IMS board / Reports

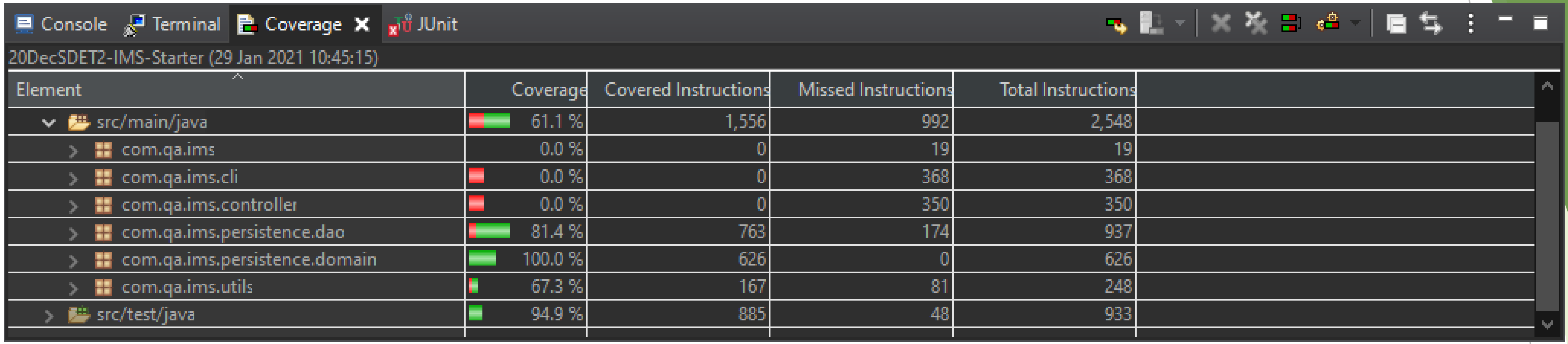
Burndown Chart

IMS Sprint 1 Story Points

Burndown Chart

A visual representation of my progression throughout the project sprint.

Testing

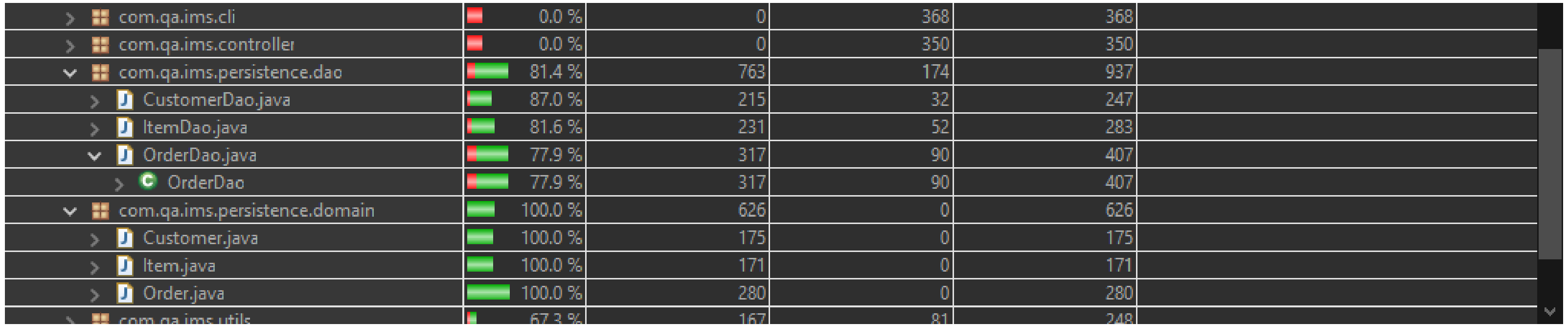


20DecSDET2-IMS-Starter (29 Jan 2021 10:45:15)

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
src/main/java	61.1 %	1,556	992	2,548
com.qa.ims	0.0 %	0	19	19
com.qa.ims.cli	0.0 %	0	368	368
com.qa.ims.controller	0.0 %	0	350	350
com.qa.ims.persistence.dao	81.4 %	763	174	937
com.qa.ims.persistence.domain	100.0 %	626	0	626
com.qa.ims.utils	67.3 %	167	81	248
src/test/java	94.9 %	885	48	933

As you can see:

- I tested and have achieved 100% line coverage of the persistence.domain package in the src/main/java folder
- I tested for 87.4% of the Data Access Objects in the persistence.dao package in the src/main/java comprising of:



com.qa.ims.cli	0.0 %	0	368	368
com.qa.ims.controller	0.0 %	0	350	350
com.qa.ims.persistence.dao	81.4 %	763	174	937
CustomerDao.java	87.0 %	215	32	247
ItemDao.java	81.6 %	231	52	283
OrderDao.java	77.9 %	317	90	407
OrderDao	77.9 %	317	90	407
com.qa.ims.persistence.domain	100.0 %	626	0	626
Customer.java	100.0 %	175	0	175
Item.java	100.0 %	171	0	171
Order.java	100.0 %	280	0	280
com.qa.ims.utils	67.3 %	167	81	248

Extra vocal note regarding Controller testing - Mockito

Live Demonstration

Let’s run through a couple user stories!

Sprint Review

- What did you complete?

In the end, I was able to achieve all of the user stories asked of me in the specification. All the way from add a customer to the system to delete an item in an order. My application can interact with an end user via a CLI (command-line interface) that is capable of many functions.

- What got left behind?

In the testing for my OrderDao, I wasn’t able to fix an issue regarding the returning of null objects for my update function(s). After hours of attempting and bug fixes, I still couldn’t pin down what the issue was, however it hadn’t broken the application itself but perhaps it’s connection to the external database.

I would have also liked to have increased my line coverage in my src/main folder through the use of testing for more null and exception returns as well as the use of Mockito to test all my controllers.

Sprint Retrospective

- What went well?

The initial creation of the domain classes, I was able to follow through with the template of the Customer class that we started with and create an order class and an item class with ease, making use of the ICrudController interface that we were supplied with.

Similarly, creating the controllers was done methodically which meant I was able to get it done in a short space of time compared to the DAO classes.

- What could be improved?

The tests I had written for the OrderDAO class as well as perhaps the OrderDAO class itself? I had gotten it working and running well on the application but had an issue during testing; resulting in null returns for my update functions compared to what I was expecting. I initially written it down to a problem with my SQL, but after review and revision, there still seems to be issues which I'm not entirely sure about…

Conclusion

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

One thing I would say is to not take the pressure on so harshly and allow it to stress me out, at the end of the project I did feel quite burned out getting to this stage but after having completed it, I am proud of what I created and want to improve on it even more.

Other points I would say:

- Skewed burndown chart due to adding backlog issues mid-sprint

Thank you for your time!

Any questions?