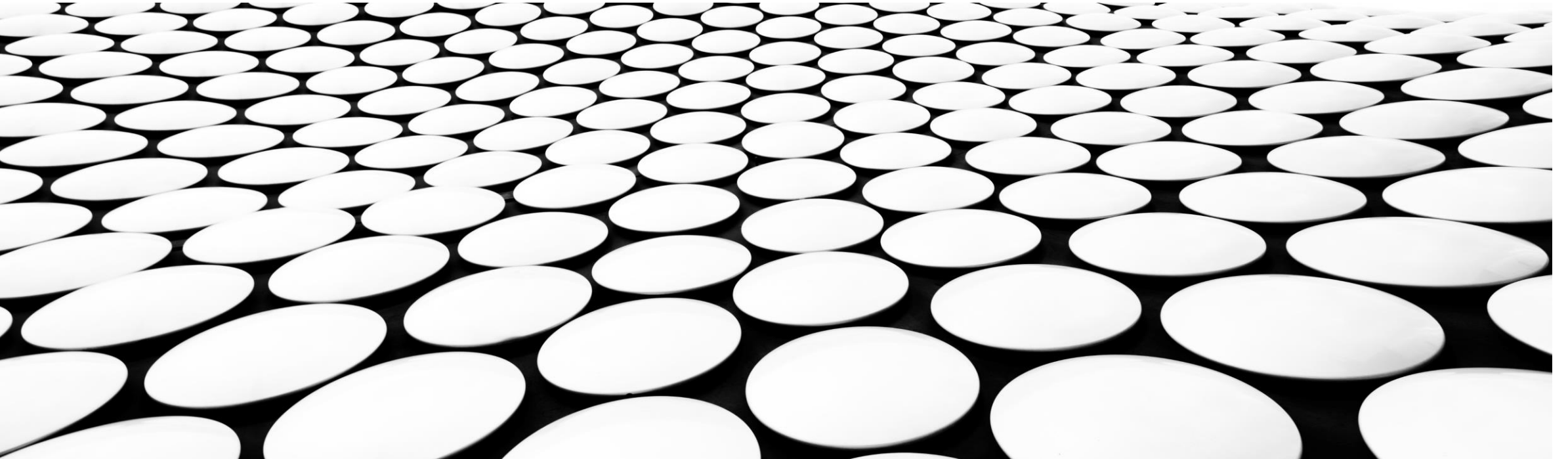

IMS PROJECT

BENJAMIN OLUYOMI



INTRODUCTION

Projects / IMS QA Project / IQP board

Backlog

Backlog 22 issues

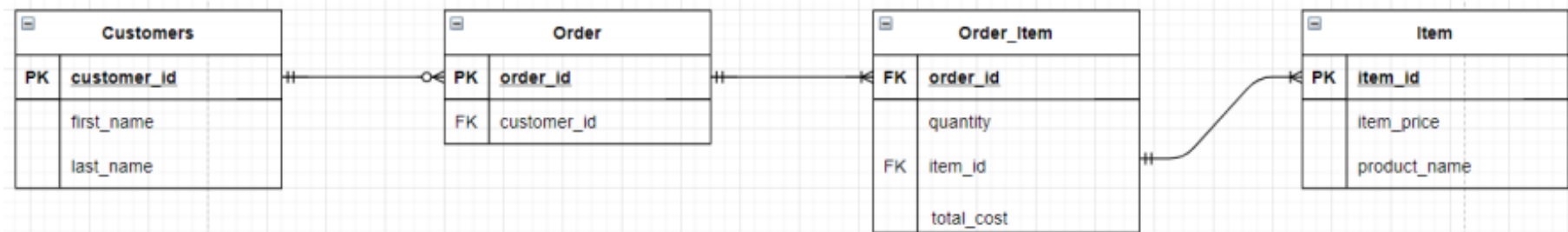
Issue Title	Status	Priority
Entity Relationship Diagram	Create Project Evaluations	IQP-3
Risk Assessment	Create Project Evaluations	IQP-5
MoSCoW	Create Project Evaluations	IQP-6
CRUD functionality following the Enterprise Architecture Model for the customers, items, and orders entities	Codebase	IQP-9
The project connects via JDBC to a local or GCP-based MySQL instance	Codebase	IQP-11
Adherence to best practice (e.g. OOP principles, SOLID, refactoring)	Codebase	IQP-12
Unit test coverage of the src/main/java folder, aiming for 80%	JUnit tests for validation MINIMUM 80%	IQP-13
Git repository utilising the feature-branch model	Continuous Integration	IQP-15
The master branch must compile	Continuous Integration	IQP-16
A build of the application is present in the root folder of your git repo	Continuous Integration	IQP-17
A fat .jar which can be deployed from the command-line	Continuous Integration	IQP-18
Introduction: Who are you? How did you approach the specification?	Presentation	IQP-20
Consultant Journey: What technologies have you learned for this project?	Presentation	IQP-21
CI: How did you approach version control?	Presentation	IQP-22
Testing: What was tested? Show the coverage of the src/main/java folder.	Presentation	IQP-23
Demonstration: Run through a couple of user stories	Presentation	IQP-24
Sprint review: What did you complete? What got left behind?	Presentation	IQP-25
Sprint retrospective: What went well? What could be improved?	Presentation	IQP-26
Conclusion: Reflections on the project, future steps, any other relevant info	Presentation	IQP-27

I started by condensing the brief into digestible points/tasks that I could complete in bursts using sprints.

This was done using Jira, and the tasks were managed into separate epics to further clarify what tasks were related to each other.

This helped to efficiently manage the backlog.

ENTITY RELATIONSHIP DIAGRAM



The customers to order relation ship is a one to zero or many.

The relationship between order to order_item is one to many.

The order_item table connects the order and item table (here items can be added to removed from orders based on order id's). The total cost for each item to order interaction is also calculated here.



CONSULTANT JOURNEY

- Git as a Version Control System
- GitHub Source Code Management System
- Jira Project management
- GCP MySQL
- Java backend programming language
- Maven build tool
- JUnit to test the classes



CI

I used the repository for the IMS starter project to set up my own repo on my GitHub account. I then cloned it down to my personal machine and began working on the code from there.

For each working feature of the code, a commit was made to push up to GitHub

TESTING

Element	Coverage	Covered Instructions	Missed Instructions
IMS prj QA	<div><div></div></div> 49.4 %	1,562	1,601
src/main/java	<div><div></div></div> 41.5 %	1,058	1,489
com.qa.ims.controller	<div><div></div></div> 21.8 %	116	415
com.qa.ims.persistence.dao	<div><div></div></div> 56.3 %	527	409
com.qa.ims.persistence.domain	<div><div></div></div> 43.5 %	251	326
com.qa.ims	<div><div></div></div> 0.0 %	0	183
com.qa.ims.utils	<div><div></div></div> 51.2 %	164	156
src/test/java	<div><div></div></div> 81.8 %	504	112

Element	Coverage	Covered Instructions	Missed Instructions	Total Instru
IMS prj QA	<div><div></div></div> 68.1 %	2,729	1,278	
src/main/java	<div><div></div></div> 62.9 %	1,602	945	
src/test/java	<div><div></div></div> 77.2 %	1,127	333	

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
IMS prj QA	<div><div></div></div> 71.9 %	2,990	1,166	4,156
src/main/java	<div><div></div></div> 65.1 %	1,673	896	2,569
src/test/java	<div><div></div></div> 83.0 %	1,317	270	1,587



DEMONSTRATION



SPRINT REVIEW - WHAT WENT WELL?

- Small daily sprints made it easier to keep track of everything that was required
- Majority of the tasks were completed
- The use of epics further helped to clarify which tasks need to be finished before others



SPRINT REVIEW - WHAT GOT LEFT BEHIND?

- 80% testing
- Extra discount features
- Cost functionality for total



SPRINT RETROSPECTIVE

- I'm fairly happy with the way that I approached the sprints in terms of the short manageable bursts
- Should have used a priority system
- However I would in hindsight make the tasks far more specific to make sure I kept good track of the smaller details



CONCLUSION

- Despite achieving 71.6% coverage for the whole project, I was unfortunately unable to reach 80% test for the src/main folder, in hindsight I would've allocated more time for testing.
- I would also make the order interface a lot simpler and keep it in one table for the end user
- Overall I think the project was successful in delivering the main functionality. However due to some delays in getting certain functions working, I was unable to make time to finish some of the extra features I wanted to include, Like features that apply discounts and so on.



THANK YOU FOR LISTENING