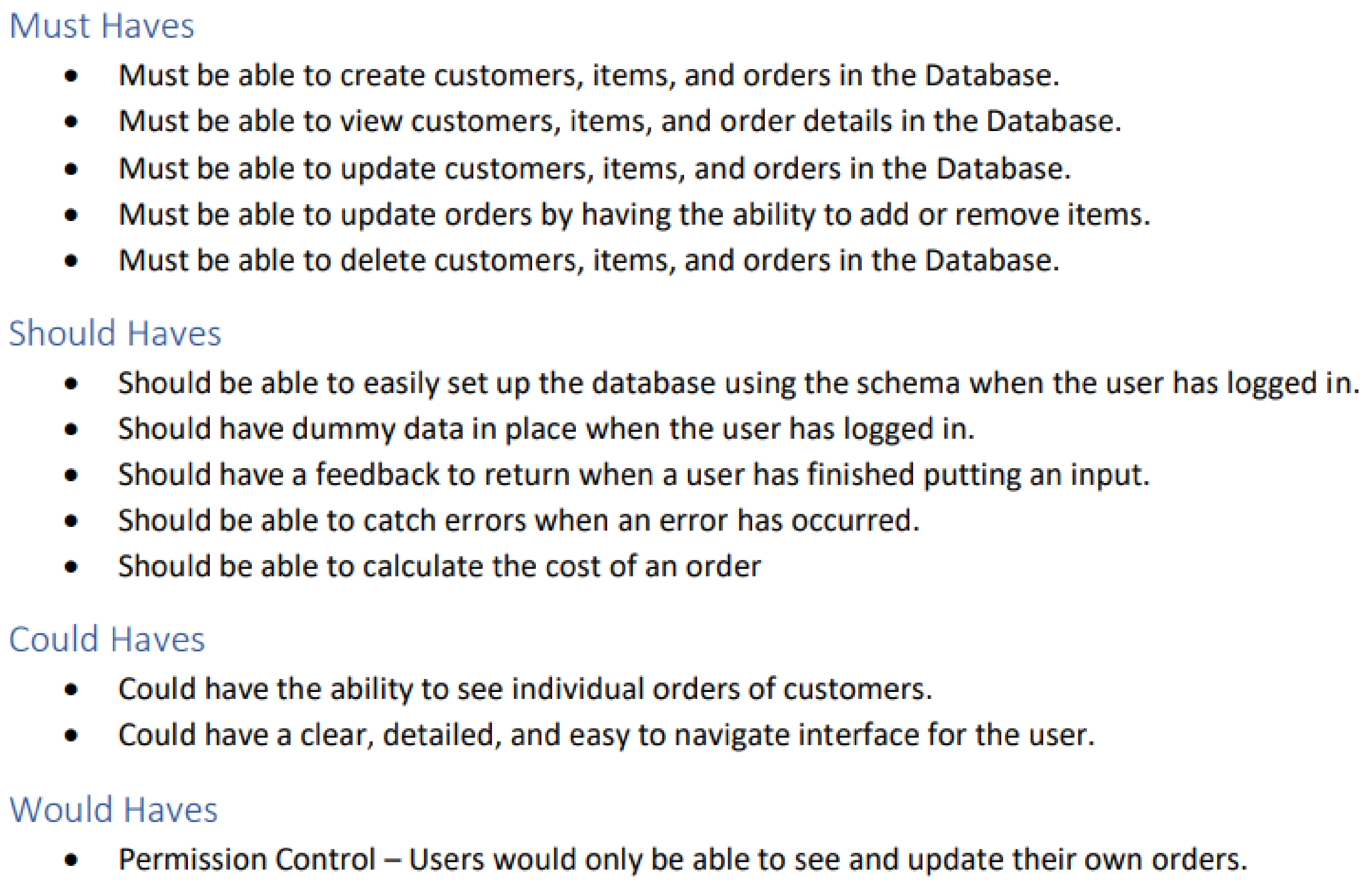
INVENTORY MANAGEMENT SYSTEM

Carl Angeles

DESIGN (RISK ASSESSMENT)

ID	Risk Description	Cause	Effect	Likelihood (1-5)	Impact (1-5)	Risk Rating (1-25)	Action
1	Running out of time	Bad time management	Project not fully completed, requirements are not met or missing features	5	5	23	Daily planning, check outstanding tasks and plan what to do.
2	Insufficient knowledge on technology	Technology not covered during training	Requirements not being met due to features not being implemented	3	4	12	Check back on recordings, notes and ask trainer for help.
3	PC Issues	Internet Issues, Hardware fails	Unable to work on project which results in having an incomplete or lost project	2	3	5	Regular backup on Git, Use Mobile Data (if applicable) if internet issues occur
4	Database Problems	Incorrect data relationship	Cannot test project since it relies on the database	2	5	15	Ensure there is no many to many relationship and if problems occur consult with trainer or peers
5	Version Control not utilized correctly	Incorrect use of Git, no branches for features/testing or lack of pushing	No backups, cannot rollback when an error occurs	2	2	3	Always work on a new branch when implementing a new feature, regular push after each feature/functionality
6	Program not running properly	Not enough testing	Not meeting requirements, program not working as intended	4	4	20	At least 80% coverage in the testing phase.

MOSCOW ANALYSIS



Must Haves

- Must be able to create customers, items, and orders in the Database.
- Must be able to view customers, items, and order details in the Database.
- Must be able to update customers, items, and orders in the Database.
- Must be able to update orders by having the ability to add or remove items.
- Must be able to delete customers, items, and orders in the Database.

Should Haves

- Should be able to easily set up the database using the schema when the user has logged in.
- Should have dummy data in place when the user has logged in.
- Should have a feedback to return when a user has finished putting an input.
- Should be able to catch errors when an error has occurred.
- Should be able to calculate the cost of an order

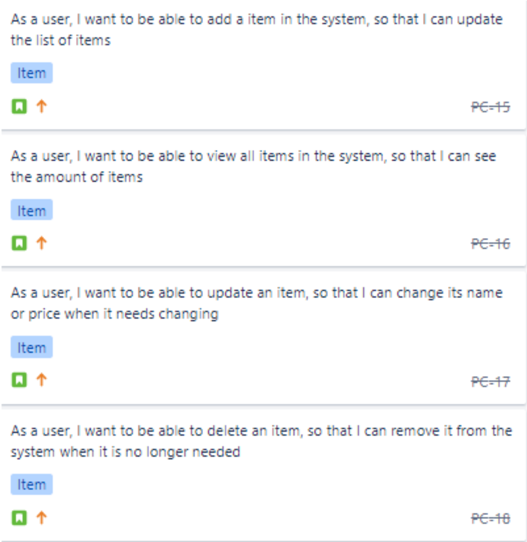
Could Haves

- Could have the ability to see individual orders of customers.
- Could have a clear, detailed, and easy to navigate interface for the user.

Would Haves

- Permission Control – Users would only be able to see and update their own orders.

DESIGN (KANBAN)



As a user, I want to be able to add a item in the system, so that I can update the list of items

Item

~~PC-15~~

As a user, I want to be able to view all items in the system, so that I can see the amount of items

Item

~~PC-16~~

As a user, I want to be able to update an item, so that I can change its name or price when it needs changing

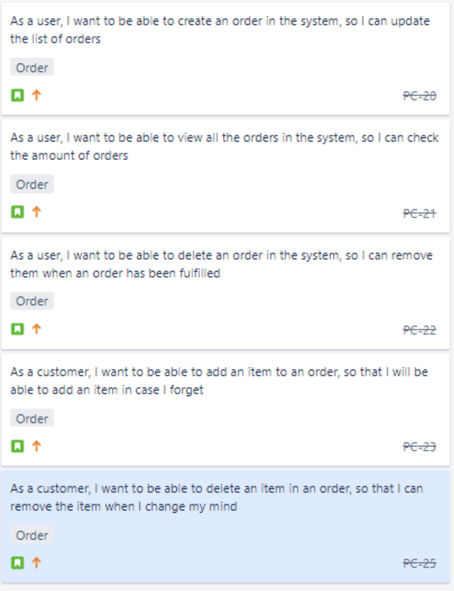
Item

~~PC-17~~

As a user, I want to be able to delete an item, so that I can remove it from the system when it is no longer needed

Item

~~PC-18~~



As a user, I want to be able to create an order in the system, so I can update the list of orders

Order

~~PC-20~~

As a user, I want to be able to view all the orders in the system, so I can check the amount of orders

Order

~~PC-21~~

As a user, I want to be able to delete an order in the system, so I can remove them when an order has been fulfilled

Order

~~PC-22~~

As a customer, I want to be able to add an item to an order, so that I will be able to add an item in case I forget

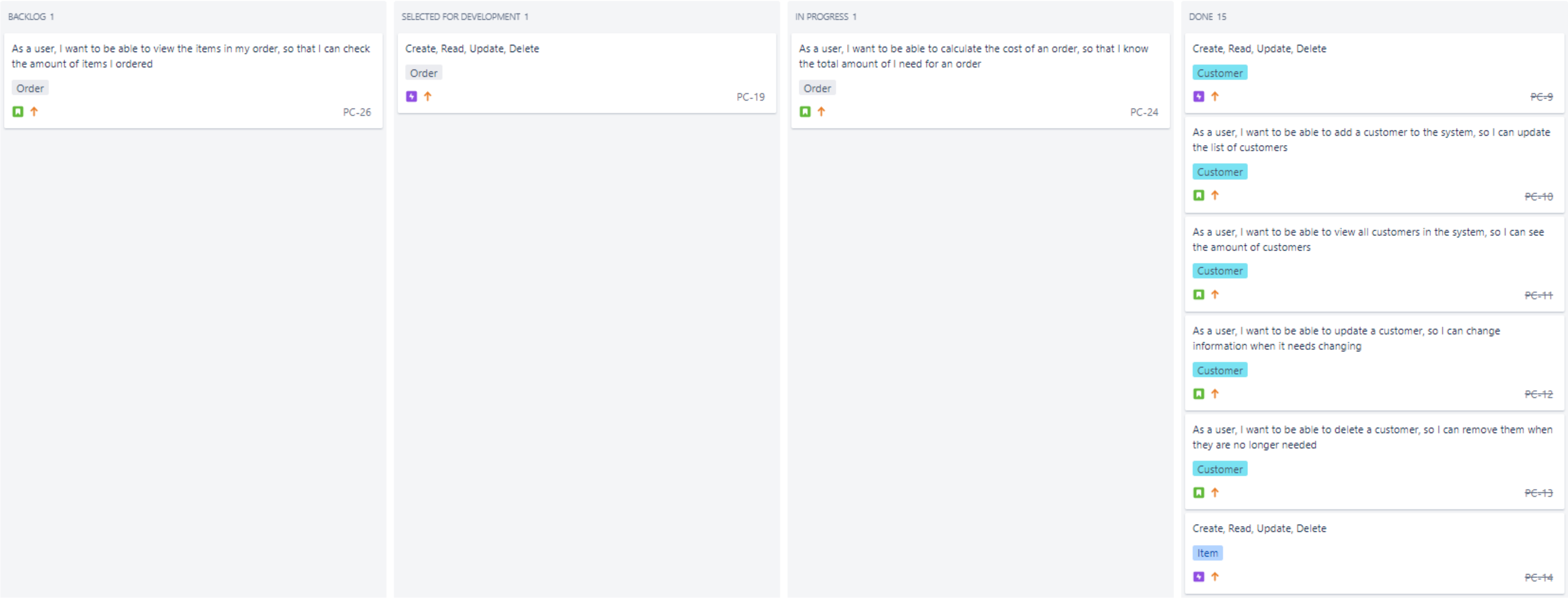
Order

~~PC-23~~

As a customer, I want to be able to delete an item in an order, so that I can remove the item when I change my mind

Order

~~PC-25~~



BACKLOG 1

As a user, I want to be able to view the items in my order, so that I can check the amount of items I ordered

Order

PC-26

SELECTED FOR DEVELOPMENT 1

Create, Read, Update, Delete

Order

PC-19

IN PROGRESS 1

As a user, I want to be able to calculate the cost of an order, so that I know the total amount of I need for an order

Order

PC-24

DONE 15

Create, Read, Update, Delete

Customer

~~PC-9~~

As a user, I want to be able to add a customer to the system, so I can update the list of customers

Customer

~~PC-10~~

As a user, I want to be able to view all customers in the system, so I can see the amount of customers

Customer

~~PC-11~~

As a user, I want to be able to update a customer, so I can change information when it needs changing

Customer

~~PC-12~~

As a user, I want to be able to delete a customer, so I can remove them when they are no longer needed

Customer

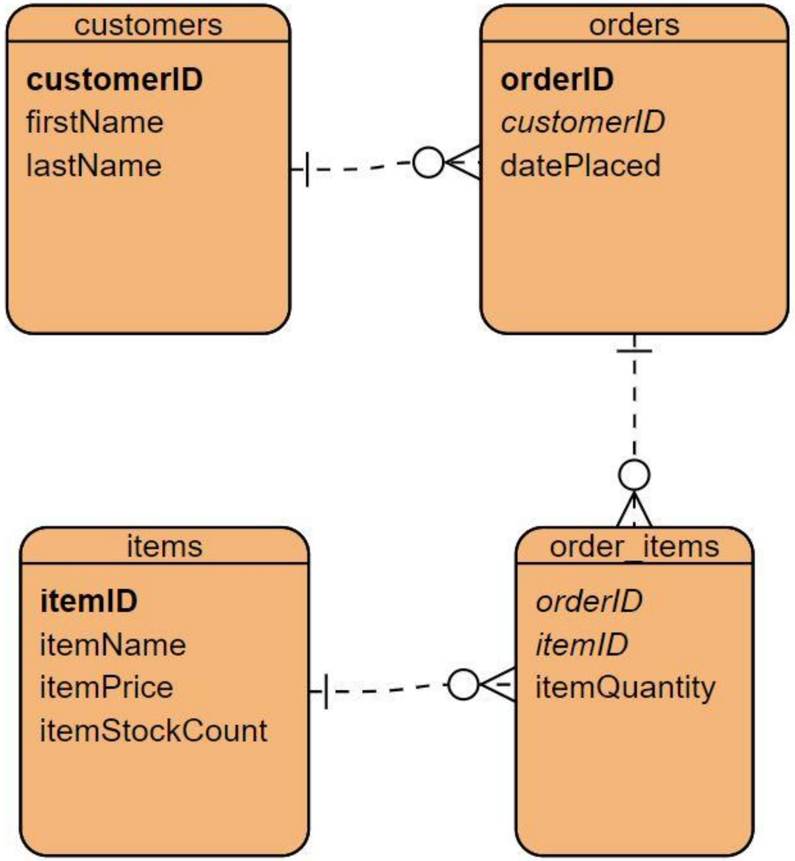
~~PC-13~~

Create, Read, Update, Delete

Item

~~PC-14~~

DESIGN (ERD)



customers

customerID

firstName

lastName

orders

orderID

customerID

datePlaced

items

itemID

itemName

itemPrice

itemStockCount

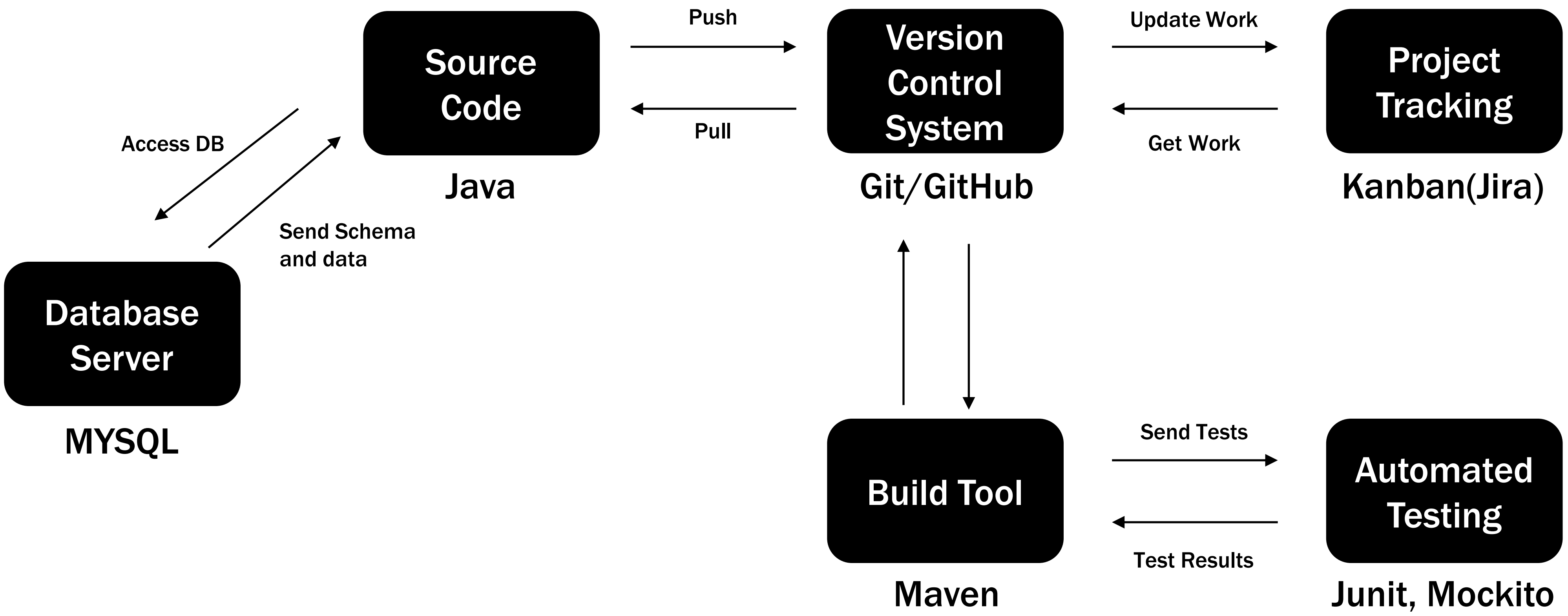
order_items

orderID

itemID

itemQuantity

CI PIPELINE


Source Code
Java
Push
Pull
Version Control System
Git/GitHub
Update Work
Get Work
Project Tracking
Kanban(Jira)
Access DB
Send Schema and data
Database Server
MYSQL
Build Tool
Maven
Send Tests
Test Results
Automated Testing
Junit, Mockito

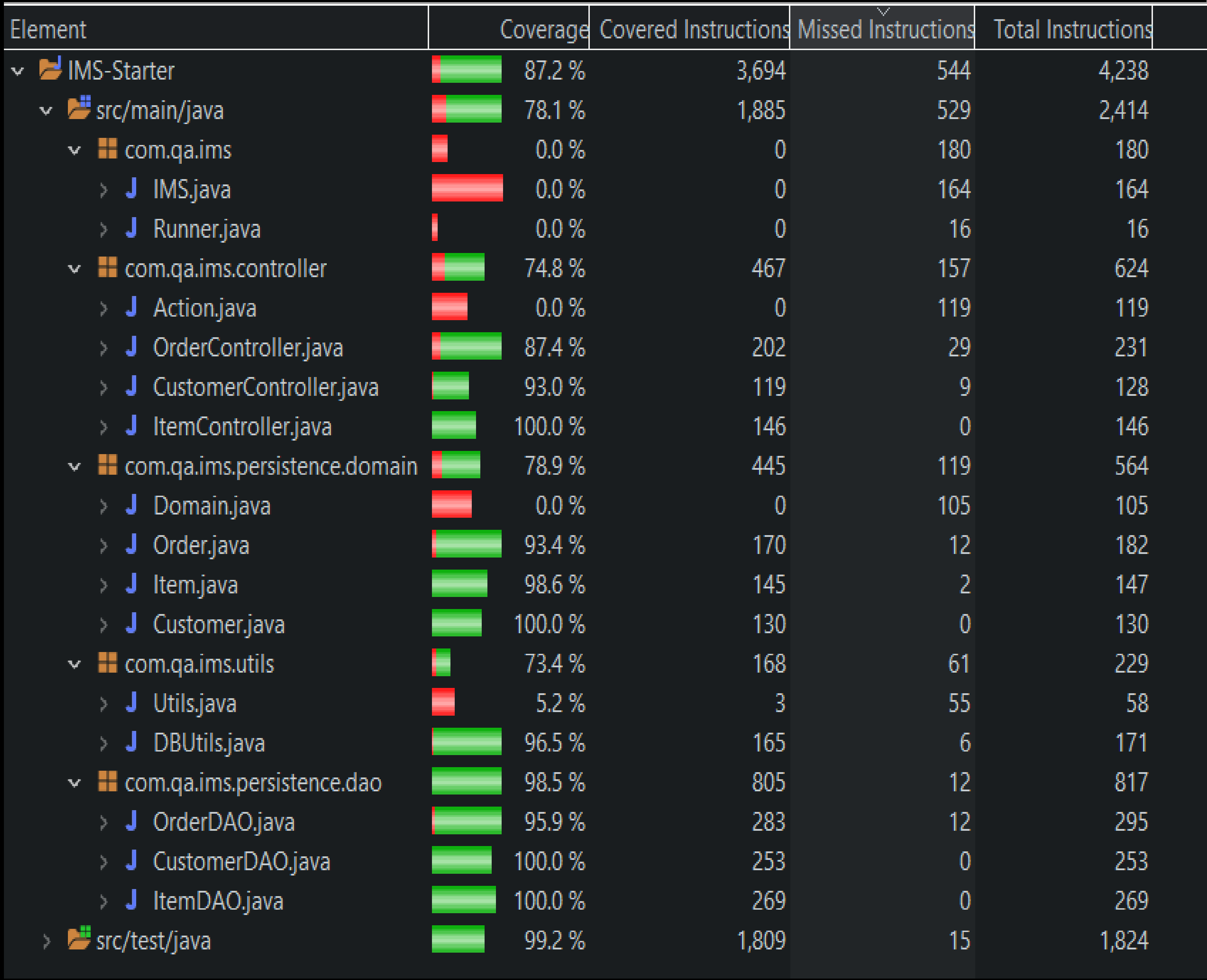
CONSULTANT JOURNEY

- Agile – Scrum, Kanban(Jira)
- Source Control – Git, Github
- Database – MYSQL
- Programming Language - Java
- Build Tool - Maven
- Testing – Junit, Mockito

TESTING

78.1% line coverage in testing

58/58 managed to run



Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
IMS-Starter	87.2 %	3,694	544	4,238
src/main/java	78.1 %	1,885	529	2,414
com.qa.ims	0.0 %	0	180	180
IMS.java	0.0 %	0	164	164
Runner.java	0.0 %	0	16	16
com.qa.ims.controller	74.8 %	467	157	624
Action.java	0.0 %	0	119	119
OrderController.java	87.4 %	202	29	231
CustomerController.java	93.0 %	119	9	128
ItemController.java	100.0 %	146	0	146
com.qa.ims.persistence.domain	78.9 %	445	119	564
Domain.java	0.0 %	0	105	105
Order.java	93.4 %	170	12	182
Item.java	98.6 %	145	2	147
Customer.java	100.0 %	130	0	130
com.qa.ims.utils	73.4 %	168	61	229
Utils.java	5.2 %	3	55	58
DBUtils.java	96.5 %	165	6	171
com.qa.ims.persistence.dao	98.5 %	805	12	817
OrderDAO.java	95.9 %	283	12	295
CustomerDAO.java	100.0 %	253	0	253
ItemDAO.java	100.0 %	269	0	269
src/test/java	99.2 %	1,809	15	1,824

DEMONSTRATION

SPRINT REVIEW

What did I complete?

- Majority of the CRUD functionalities.
- Majority of Testing

What got left behind?

- Viewing of Order Item table
- Calculating cost of order

SPRINT RETROSPECTIVE

What went well?

- Version Control
- Testing

What could be improved?

- Jira
- Testing

CONCLUSION

THANK YOU