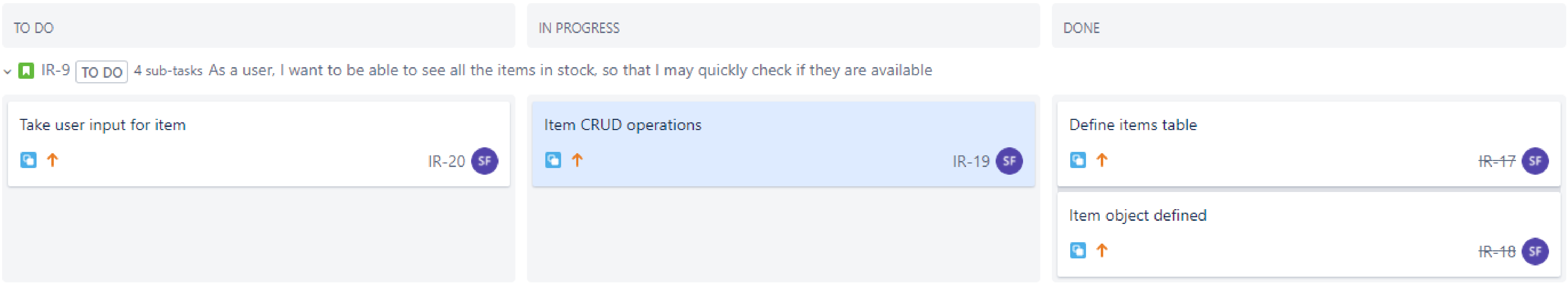
IMS Project

Simon Forster

Project Setup

- Planning processes used:
  - Jira board
  - Entity Relationship Diagram
  - MoSCoW table
  - Risk assessment table



TO DO

IN PROGRESS

DONE

IR-9 TO DO 4 sub-tasks As a user, I want to be able to see all the items in stock, so that I may quickly check if they are available

Take user input for item ↑ IR-20 SF

Item CRUD operations ↑ IR-19 SF

Define items table ↑ ~~IR-17~~ SF

Item object defined ↑ ~~IR-18~~ SF

MoSCoW Table

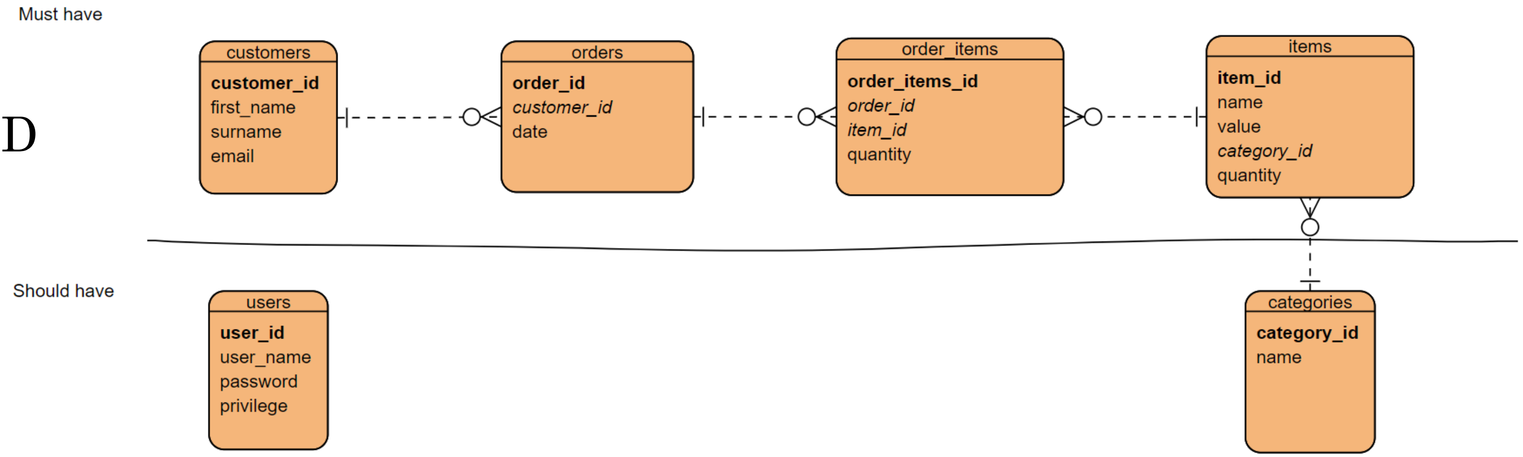
Must have	Should have	Could have	Would like to have
CLI	User login & details	User privilege levels (dependency user login)	Audit log (dependency on user login system)
User interaction with items	Separate file for db details (environment file?)	Item categories	Backup database
User interaction with customers			
User interaction with orders			

Risk Assessment Table

Ref	Risk Description	Cause	Risk Event	Action	Likelihood	Impact
1	Lack of understanding of the technologies.	Not having studied the spec and technologies in detail.	Making mistakes in the assignment.	Make sure to spend time reading and understanding the spec & technologies.	Medium	High
2	Lack of time.	Misusing my time.	Not being able to hand in a complete assignment.	Split up the entirety of the project into major goals, and those into smaller tasks.	Low	High
3	Worldwide disruptions.	COVID-19 outbreak development.	Regulations changes that disrupt services, work and health.	Adhere to hygiene and social distancing standards. Work remotely.	High	Low
4	Data breach.	A non-user accesses the system.	Data stored in the database could be used with malicious intent.	Add a log in system to limit access to necessary personnel.	Medium	High
5	Misuse of system.	A user accidentally modifies or deletes data.	Would lose table data.	Can create an audit log to be able to roll back changes, a backup database for added safety.	Low	Very High
6	Sensitive project data misplaced.	I would accidentally leak my database details online.	Database details accessible by anyone.	Whitelist user if using GCP and avoid putting database details in code.	Medium	High

Entity Relationship Diagram

Planning ERD



Must have

customers
customer_id
first_name
surname
email

orders
order_id
customer_id
date

order_items
order_items_id
order_id
item_id
quantity

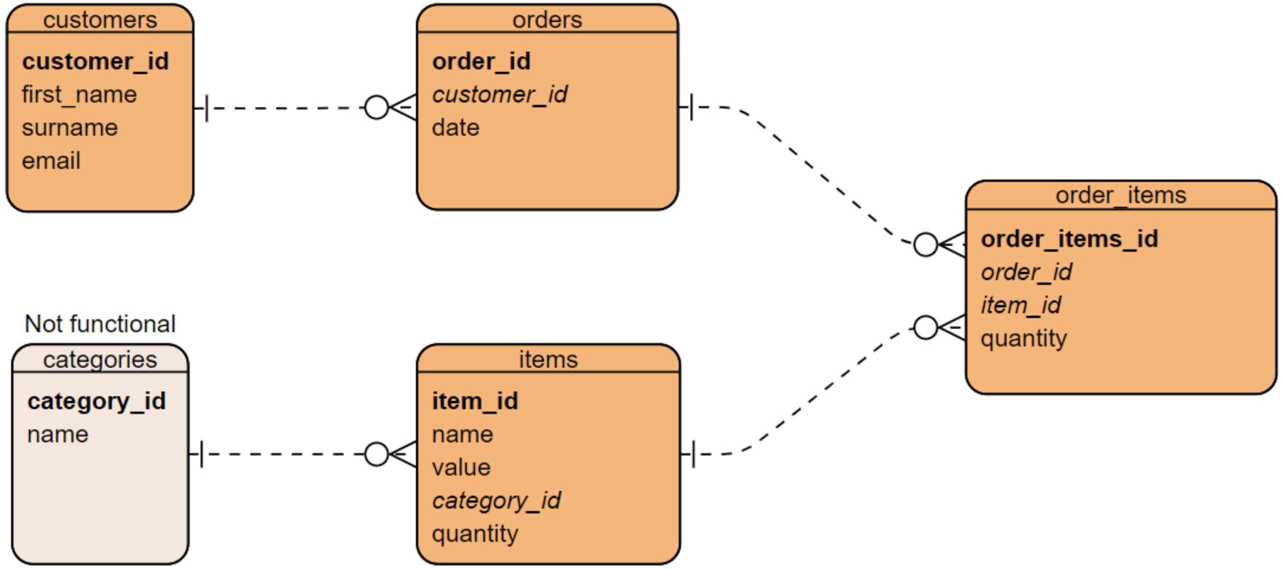
items
item_id
name
value
category_id
quantity

Should have

users
user_id
user_name
password
privilege

categories
category_id
name

Current ERD



customers
customer_id
first_name
surname
email

orders
order_id
customer_id
date

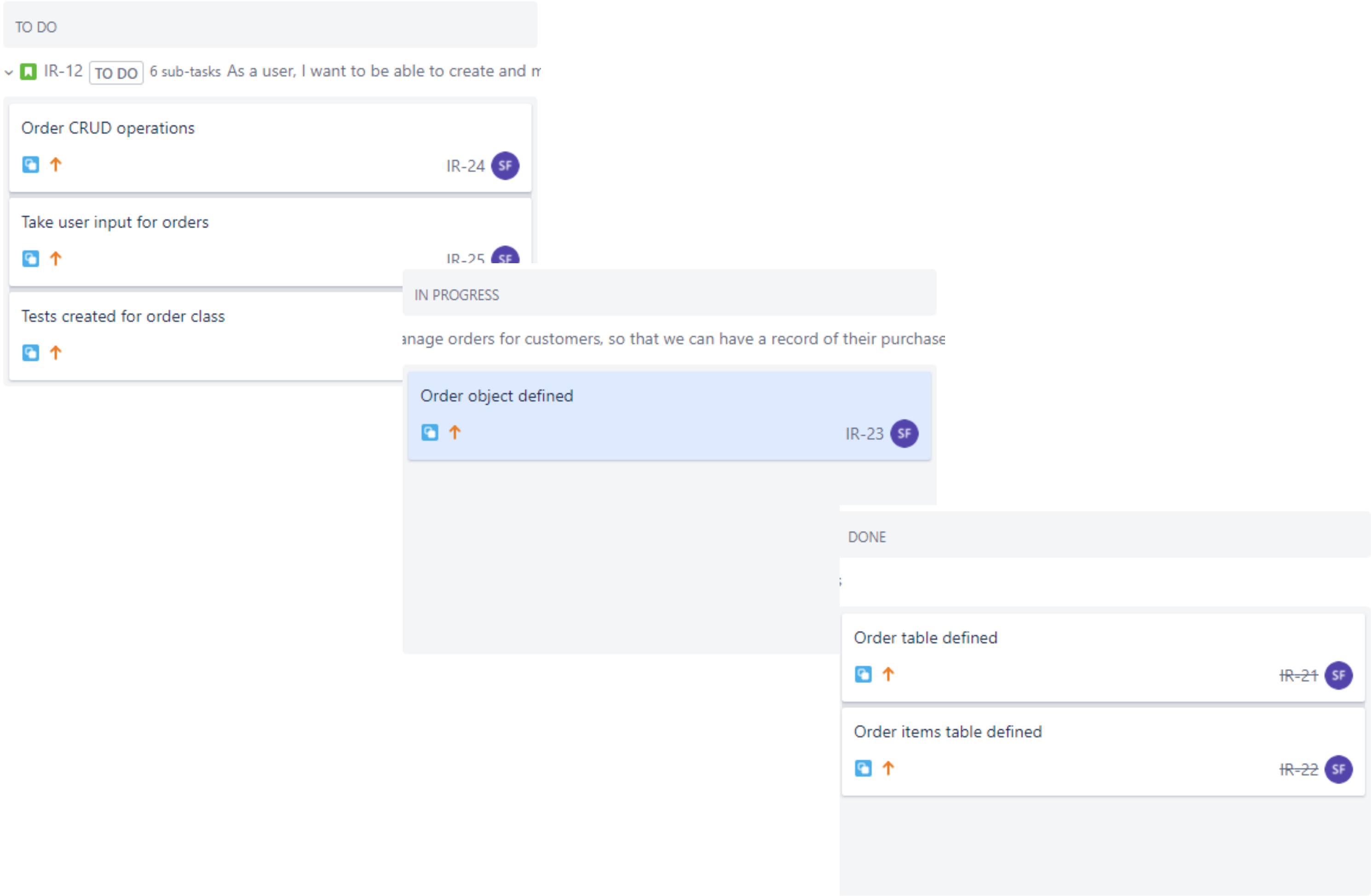
order_items
order_items_id
order_id
item_id
quantity

Not functional

categories
category_id
name

items
item_id
name
value
category_id
quantity

Jira Planning



TO DO

IR-12 TO DO 6 sub-tasks As a user, I want to be able to create and m

Order CRUD operations
IR-24 SF

Take user input for orders
IR-25 SF

Tests created for order class

IN PROGRESS

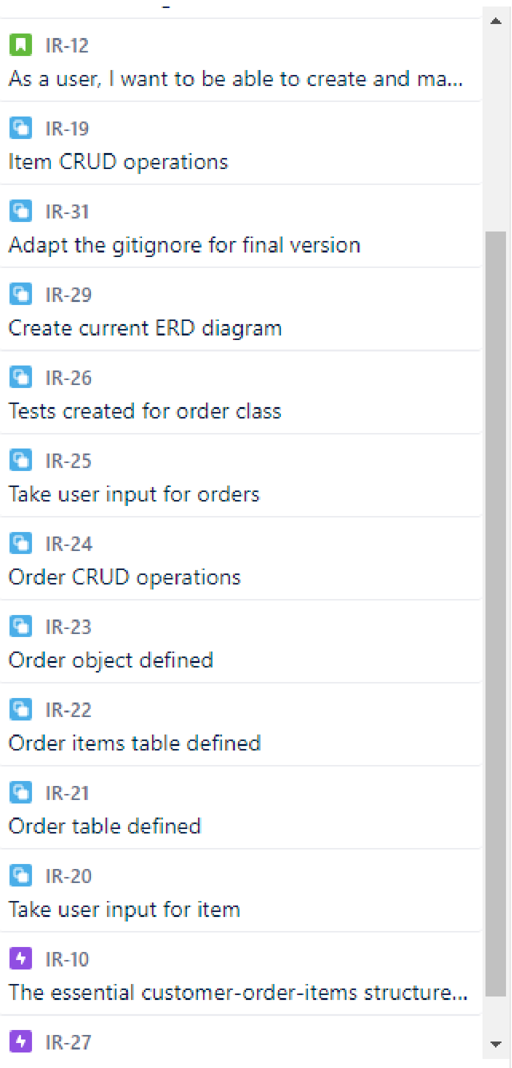
anage orders for customers, so that we can have a record of their purchase

Order object defined
IR-23 SF

DONE

Order table defined
~~IR-21~~ SF

Order items table defined
~~IR-22~~ SF



IR-12
As a user, I want to be able to create and ma...

IR-19
Item CRUD operations

IR-31
Adapt the gitignore for final version

IR-29
Create current ERD diagram

IR-26
Tests created for order class

IR-25
Take user input for orders

IR-24
Order CRUD operations

IR-23
Order object defined

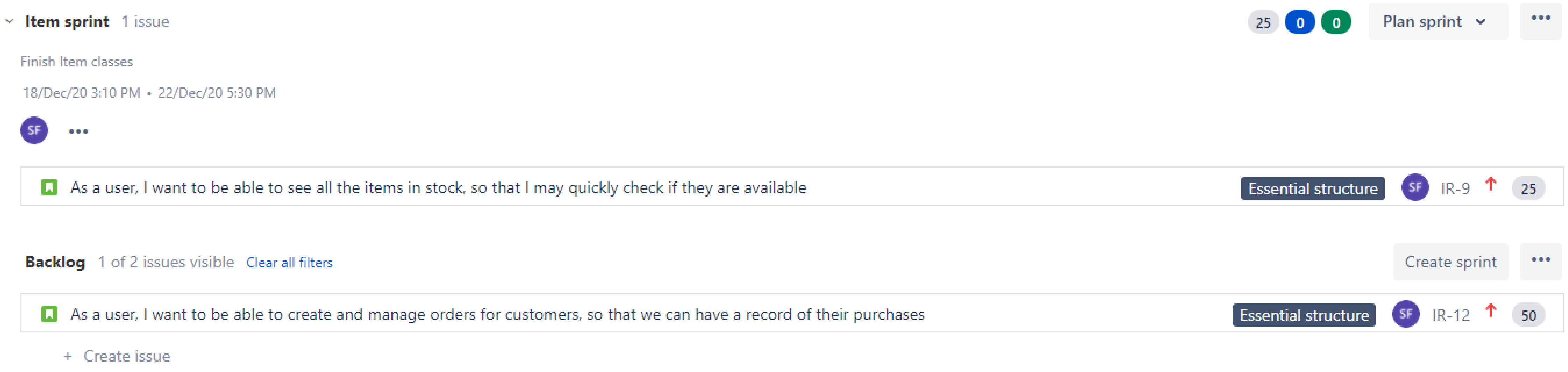
IR-22
Order items table defined

IR-21
Order table defined

IR-20
Take user input for item

IR-10
The essential customer-order-items structure...

IR-27



Item sprint 1 issue

25 0 0

Finish Item classes

18/Dec/20 3:10 PM • 22/Dec/20 5:30 PM

SF

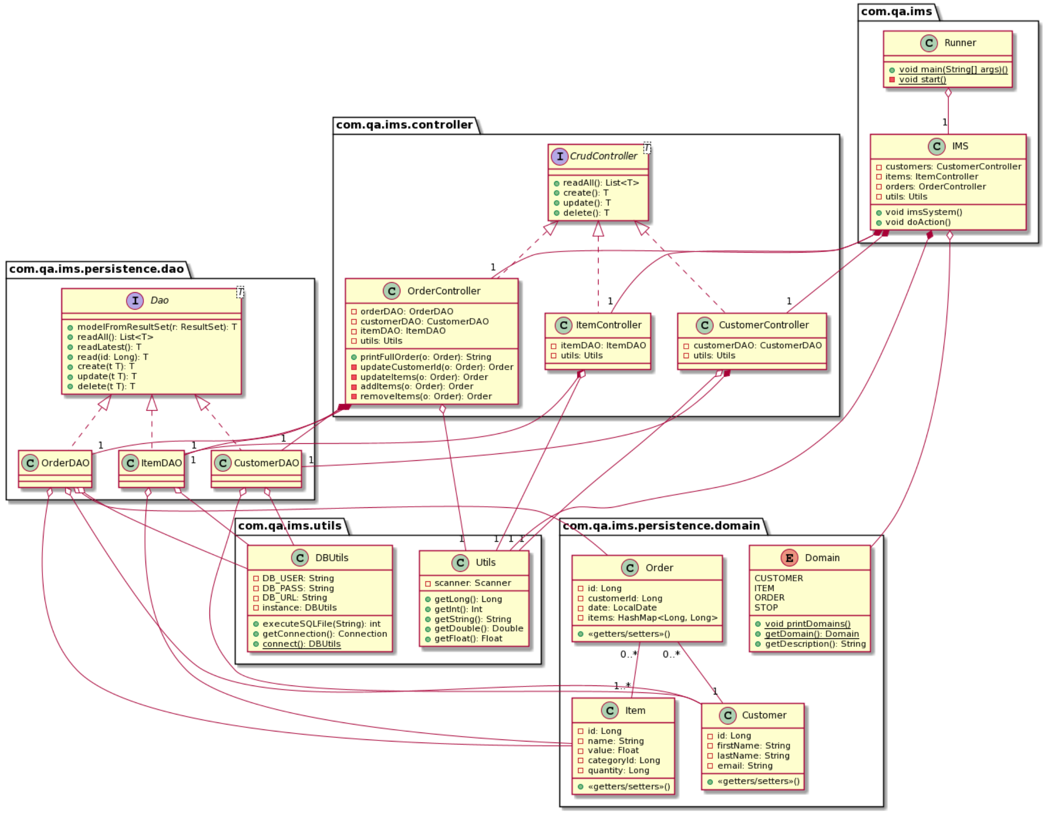
As a user, I want to be able to see all the items in stock, so that I may quickly check if they are available	Essential structure	SF	IR-9 ↑	25

Backlog 1 of 2 issues visible Clear all filters

As a user, I want to be able to create and manage orders for customers, so that we can have a record of their purchases	Essential structure	SF	IR-12 ↑	50

Consultant Journey

- Agile – SCRUM, Kanban
- Source Control – Git, GitHub
- Database – MySQL
- Programming Language – Java
- Build Tool – Maven
- Testing – JUnit, Mockito
- Diagram Usage – ERD, PlantUML



Continuous Integration

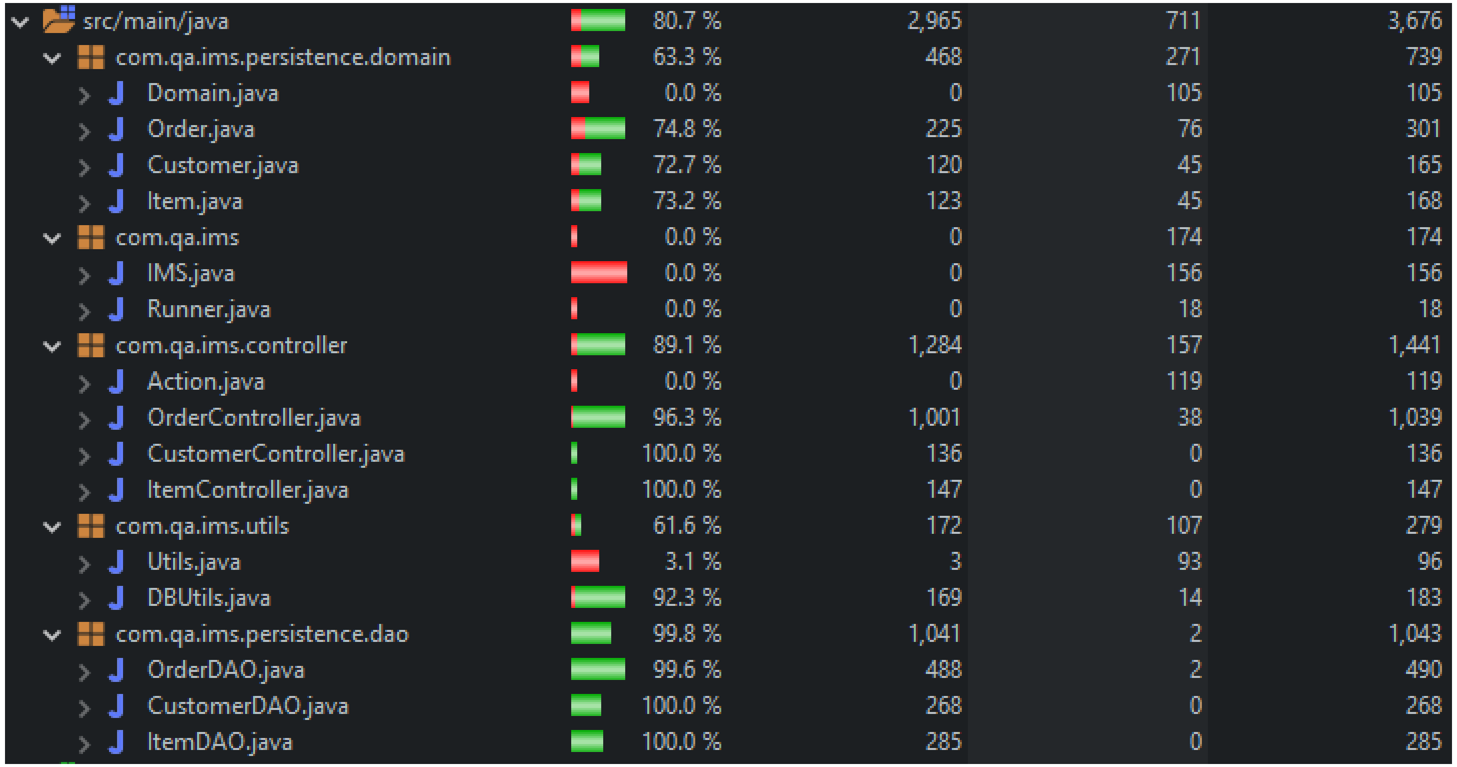
- Used Git & GitHub
- Ticket IDs on commits as frequently as possible
- Commit and push to feature branches often
- Only one feature (or fix/other) branch up at a time
- Only made changes to the developer branch through merges

What could have been done better:

- Clearer usage for branches
- Decent micromanagement of commits, could still have been better

Testing

- 80.7% test coverage



src/main/java	80.7 %	2,965	711	3,676
com.qa.ims.persistence.domain	63.3 %	468	271	739
Domain.java	0.0 %	0	105	105
Order.java	74.8 %	225	76	301
Customer.java	72.7 %	120	45	165
Item.java	73.2 %	123	45	168
com.qa.ims	0.0 %	0	174	174
IMS.java	0.0 %	0	156	156
Runner.java	0.0 %	0	18	18
com.qa.ims.controller	89.1 %	1,284	157	1,441
Action.java	0.0 %	0	119	119
OrderController.java	96.3 %	1,001	38	1,039
CustomerController.java	100.0 %	136	0	136
ItemController.java	100.0 %	147	0	147
com.qa.ims.utils	61.6 %	172	107	279
Utils.java	3.1 %	3	93	96
DBUtils.java	92.3 %	169	14	183
com.qa.ims.persistence.dao	99.8 %	1,041	2	1,043
OrderDAO.java	99.6 %	488	2	490
CustomerDAO.java	100.0 %	268	0	268
ItemDAO.java	100.0 %	285	0	285

- IMS.java has the largest untested instructions

Demo

Sprint Review

What was completed:

- Each sprint would take a few days
- Finish with an essential section of the project (customers, orders, items)
- Ended with a minimum viable product

What got left behind:

- Features from the lower priority columns in the MoSCoW table
- Potential risk handling failures

Sprint Retrospective

What went well:

- Each sprint was defined well and effective
- Breaks between each sprint
- Always knew what I had to do

What could be improved:

- Better use of user stories and epics
- Planned the tasks for documentation earlier
- Better difficulty estimates for testing development

Conclusion

- Expected to add at least couple additional features
- Could have also made better us of Enums for the CLI navigation
- Could have made an abstract class for database object to be able to use generic types for Customer, Item and Order
- Could have done more testing
- Still a success