Inventory Management System (IMS)

Matthew Garbett

January 22nd 2021

Introduction

- I’m Matthew Garbett a trainee software specialist at QA consulting

My Approach

- Used Jira Kanban Board to create sprints with user stories on each task in the specification
- Focused git branching correctly using the feature method assigning the sprints as these features

My Consultant Journey

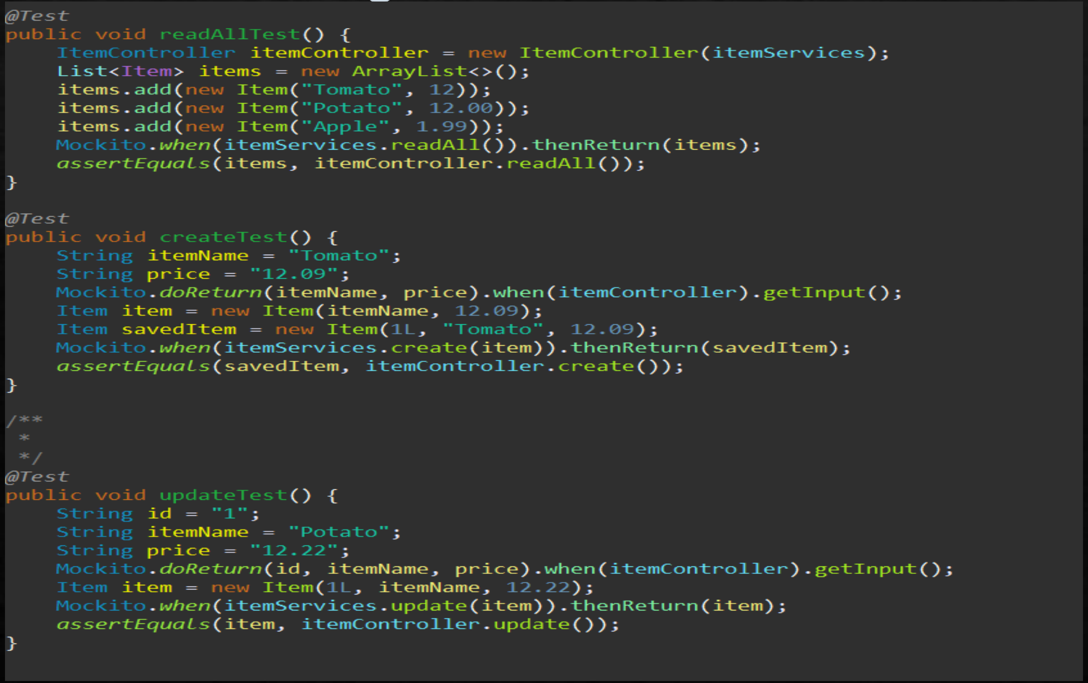
- Technologies adopted for project
  - Java – Used for the code of the project
  - Git – Used as Version Control to keep the work I have done backed up
  - MySQL – Used for the database that goes with the application using MySQL commands
  - JUNIT & Mockito – Used for testing the project
  - Maven – Used for adding dependencies and packaging/compiling the program.

My Approach to Continuous Integration(CI)

- Version Control System – Git
- Using git bash I used the feature branch model and merged with dev once features had been completed
- Committing changes was important throughout the project
  - Created backups to help ease reverting changes

IMS Test Coverage

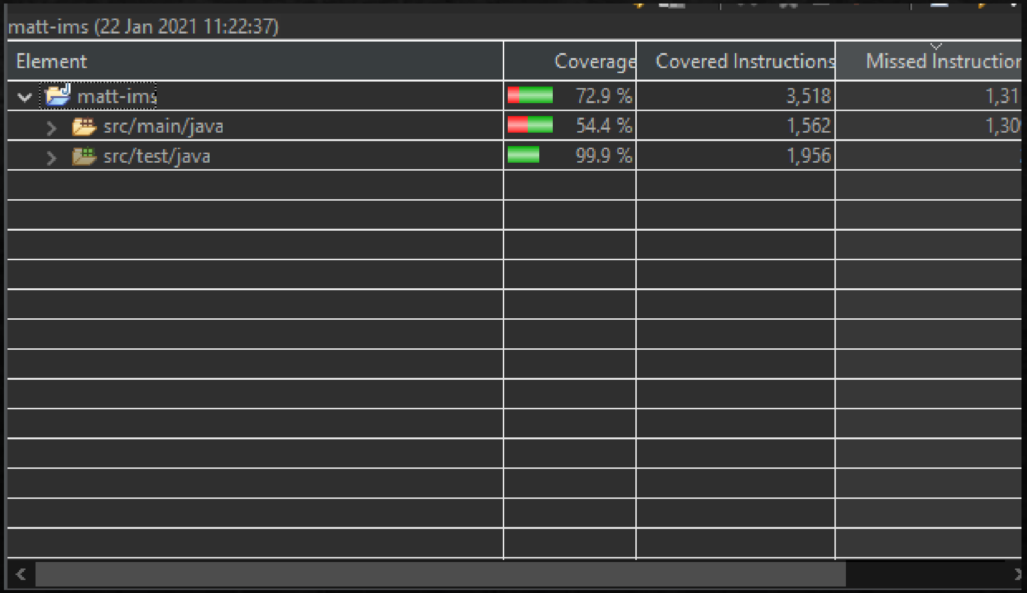
- JUNIT Testing
- Using Mockito to fake functionality on the classes
- The testing allowed for a view of what methods were failing



```
@Test
public void readAllTest() {
    ItemController itemController = new ItemController(itemServices);
    List<Item> items = new ArrayList<>();
    items.add(new Item("Tomato", 12));
    items.add(new Item("Potato", 12.00));
    items.add(new Item("Apple", 1.99));
    Mockito.when(itemServices.readAll()).thenReturn(items);
    assertEquals(items, itemController.readAll());
}

@Test
public void createTest() {
    String itemName = "Tomato";
    String price = "12.09";
    Mockito.doReturn(itemName, price).when(itemController).getInput();
    Item item = new Item(itemName, 12.09);
    Item savedItem = new Item(1L, "Tomato", 12.09);
    Mockito.when(itemServices.create(item)).thenReturn(savedItem);
    assertEquals(savedItem, itemController.create());
}

/**
 *
 */
@Test
public void updateTest() {
    String id = "1";
    String itemName = "Potato";
    String price = "12.22";
    Mockito.doReturn(id, itemName, price).when(itemController).getInput();
    Item item = new Item(1L, itemName, 12.22);
    Mockito.when(itemServices.update(item)).thenReturn(item);
    assertEquals(item, itemController.update());
}
```

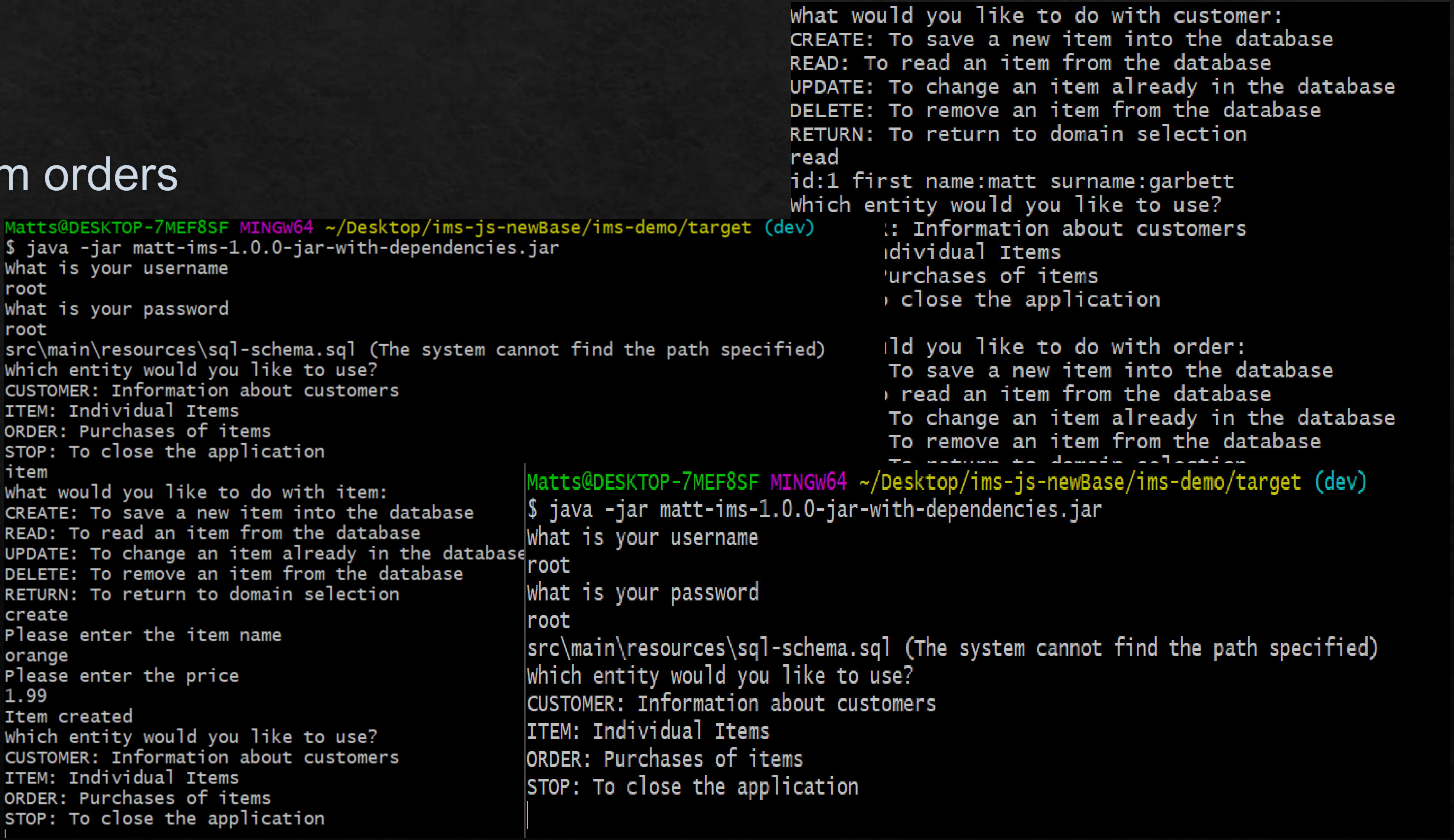


matt-ims (22 Jan 2021 11:22:37)

Element	Coverage	Covered Instructions	Missed Instructio
matt-ims	72.9 %	3,518	1,31
src/main/java	54.4 %	1,562	1,30
src/test/java	99.9 %	1,956	

IMS Demonstration

- Create item
- Read item from orders



```
What would you like to do with customer:
CREATE: To save a new item into the database
READ: To read an item from the database
UPDATE: To change an item already in the database
DELETE: To remove an item from the database
RETURN: To return to domain selection
read
id:1 first name:matt surname:garbett
Which entity would you like to use?
:: Information about customers
ndividual Items
urchases of items
 close the application

ld you like to do with order:
 To save a new item into the database
 read an item from the database
 To change an item already in the database
 To remove an item from the database
```

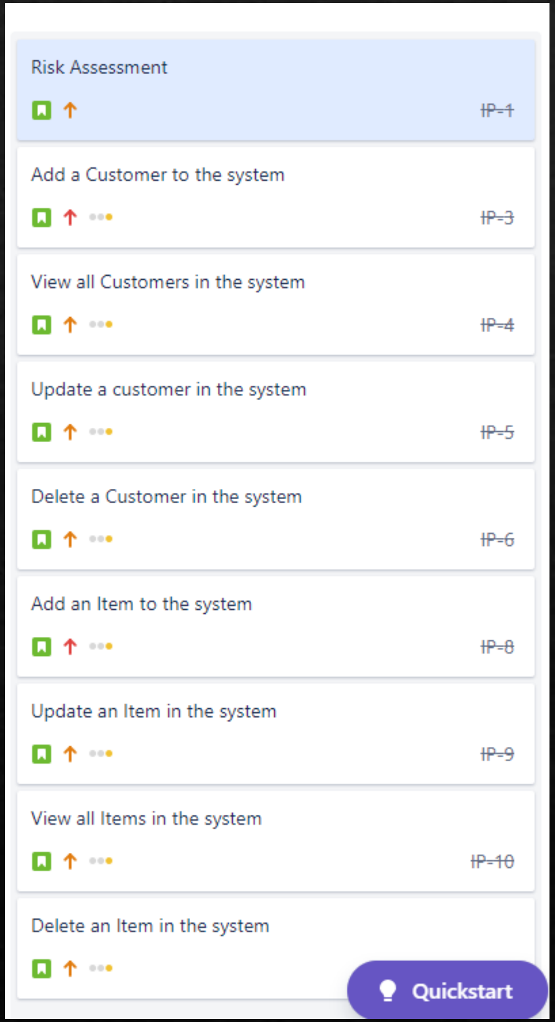
```
Matts@DESKTOP-7MEF8SF MINGW64 ~/Desktop/ims-js-newBase/ims-demo/target (dev)
$ java -jar matt-ims-1.0.0-jar-with-dependencies.jar
What is your username
root
What is your password
root
src\main\resources\sql-schema.sql (The system cannot find the path specified)
Which entity would you like to use?
CUSTOMER: Information about customers
ITEM: Individual Items
ORDER: Purchases of items
STOP: To close the application
item
What would you like to do with item:
CREATE: To save a new item into the database
READ: To read an item from the database
UPDATE: To change an item already in the database
DELETE: To remove an item from the database
RETURN: To return to domain selection
create
Please enter the item name
orange
Please enter the price
1.99
Item created
Which entity would you like to use?
CUSTOMER: Information about customers
ITEM: Individual Items
ORDER: Purchases of items
STOP: To close the application
```

```
Matts@DESKTOP-7MEF8SF MINGW64 ~/Desktop/ims-js-newBase/ims-demo/target (dev)
$ java -jar matt-ims-1.0.0-jar-with-dependencies.jar
What is your username
root
What is your password
root
src\main\resources\sql-schema.sql (The system cannot find the path specified)
Which entity would you like to use?
CUSTOMER: Information about customers
ITEM: Individual Items
ORDER: Purchases of items
STOP: To close the application
```

Sprint review

- Completed Tasks
  - Completed all user stories set on Kanban Board with Risk Assessment and Items being done the quickest

- What got left behind?
  - Full Test Coverage (Dao Testing)
  - Google Cloud Platform



Risk Assessment ~~IP-1~~

Add a Customer to the system ~~IP-3~~

View all Customers in the system ~~IP-4~~

Update a customer in the system ~~IP-5~~

Delete a Customer in the system ~~IP-6~~

Add an Item to the system ~~IP-8~~

Update an Item in the system ~~IP-9~~

View all Items in the system ~~IP-10~~

Delete an Item in the system

Sprint Retrospective

- What went well?
  - Creating methods for each story
  - Git branching used for each feature splitting off from dev branch then merging back into it

- What could be improved?
  - Start Testing earlier
  - Moving from local workbench to GCP
  - Readme file and project documents could have been continually updated

IMS project conclusions

- Putting my learning into practice with new technologies

- Future Improvements
  - Develop logical skills using code to
  - Better project management to leave more time for back-end test coverage
    - For future projects allocate more time to certain tasks

Questions?

Thank you for your time