



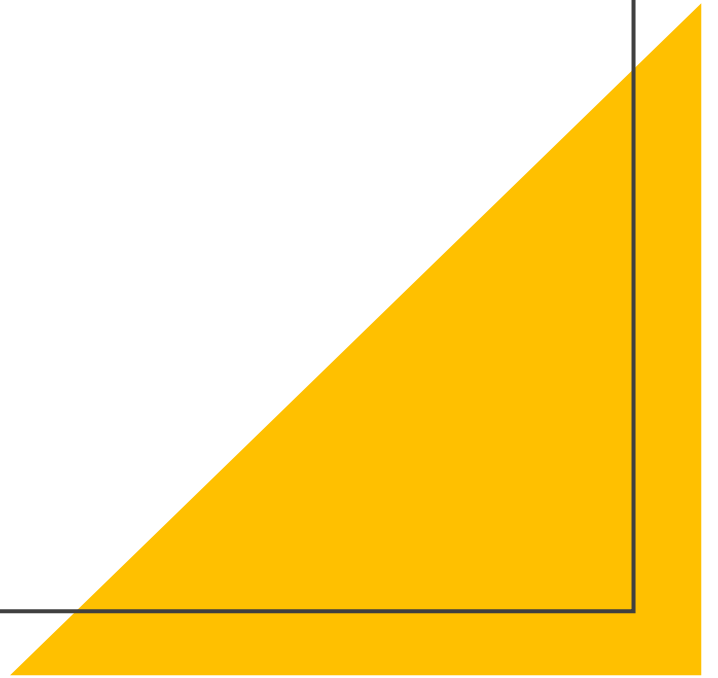
# Inventory Management System

By Adam Hackworth

# Introduction

Started off with documentation:

- Kanban Board using Jira Software
- Risk Assessment document drawn up
- ERD and UML diagrams where also drawn up



# Starting the IMS

- Built a MySQL database using MySQL Workbench which I copied into the SQL Schema on the IMS.
- Started by building an Item class, Data access object and controller.
- Then moved onto making the orders entity in the IMS.

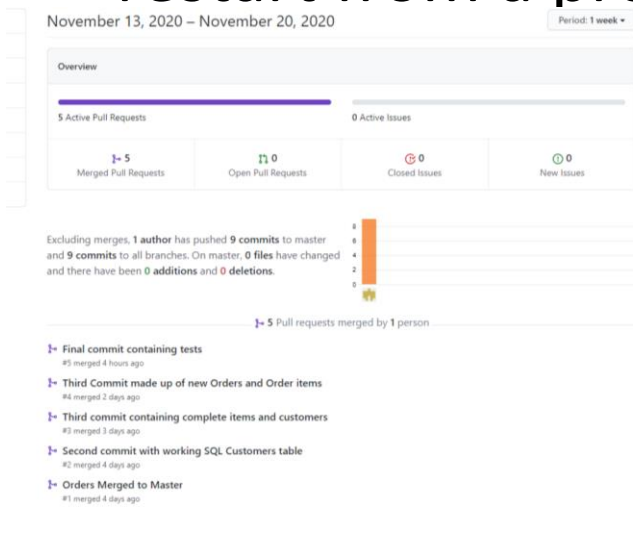
# Consultant Journey

Over the past few weeks I have learnt a variety of new software:

- Java
- Git
- Jira
- Eclipse
- Maven
- Command Line

# Continuous Integration

- Continuous Integration was performed using Git Bash software and Git Hub as a online repository to store completed code from eclipse.
- Using the branch feature on Git Bash I was able to upload various parts of the code I had been working on from a given day while still keep a safe working backup on the GitHub which allowed me to restart from a previous known working IMS.



```
MINGW64/c/Users/adamh/Desktop/IMS/IMS-Starter
modified:  src/test/java/com/qa/ims/persistence/dao/ItemDAOTest.java
modified:  src/test/java/com/qa/ims/persistence/dao/OrderDAOTest.java
modified:  src/test/resources/sql-data.sql
modified:  src/test/resources/sql-schema.sql

Untracked files:
(use "git add <file>..." to include in what will be committed)
src/test/java/com/qa/ims/controllers/ItemControllerTest.java
src/test/java/com/qa/ims/controllers/OrderControllerTest.java

adamh@LAPTOP-GB7VNBFL MINGW64 ~/Desktop/IMS/IMS-Starter (Testing)
$ git add .
$ git status
On branch Testing
Changes to be committed:
  (use "git restore --staged <file>..." to unstage)
  (use "git restore <file>..." to undo changes)
    modified:   src/test/java/com/qa/ims/persistence/dao/OrderDAO.java
    modified:   src/test/java/com/qa/ims/controllers/ItemControllerTest.java
    new file:   src/test/java/com/qa/ims/controllers/OrderControllerTest.java
    new file:   src/test/java/com/qa/ims/persistence/dao/ItemDAOTest.java
    modified:   src/test/java/com/qa/ims/persistence/dao/OrderDAOTest.java
    modified:   src/test/resources/sql-data.sql
    modified:   src/test/resources/sql-schema.sql

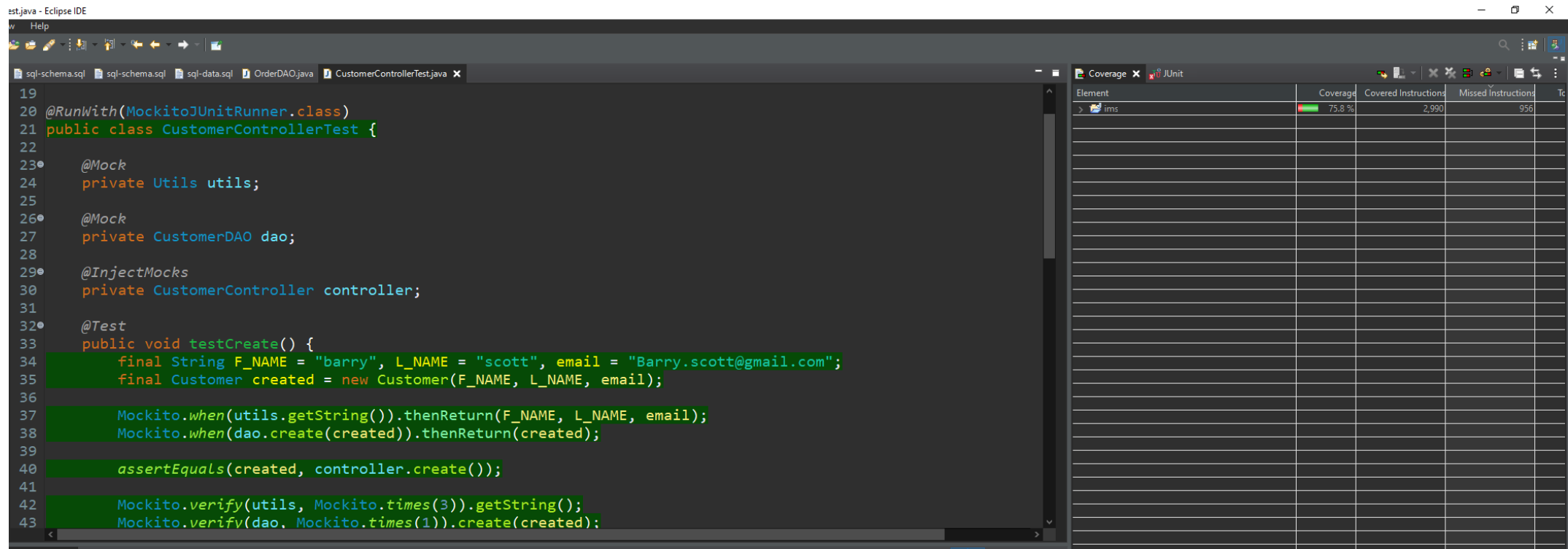
adamh@LAPTOP-GB7VNBFL MINGW64 ~/Desktop/IMS/IMS-Starter (Testing)
$ git commit
hint: Waiting for your editor to close the file... error: There was a problem wi
to the editor 'vi'.
Please supply the message using either -m or -F option.

adamh@LAPTOP-GB7VNBFL MINGW64 ~/Desktop/IMS/IMS-Starter (Testing)
$ git commit -m "Final commit containing tests"
[Testing ffa8f8a] Final commit containing tests
9 files changed, 431 insertions(+), 25 deletions(-)
create mode 100644 src/test/java/com/qa/ims/controllers/ItemControllerTest.java
create mode 100644 src/test/java/com/qa/ims/controllers/OrderControllerTest.java
rewrite src/test/resources/sql-schema.sql (80%)

adamh@LAPTOP-GB7VNBFL MINGW64 ~/Desktop/IMS/IMS-Starter (Testing)
$ git push -u origin Testing
```

# Testing

- Testing was performed with JUnit and Mockito. These two systems allowed me to test the IMS. While also allowing me to see if any bugs where in the system.



The screenshot shows the Eclipse IDE with a Java test class, `CustomerControllerTest.java`, and a JUnit coverage report. The test class is annotated with `@RunWith(MockitoJUnitRunner.class)` and uses `@Mock` for `Utils` and `CustomerDAO`, and `@InjectMocks` for `CustomerController`. The `testCreate()` method tests the `create()` method of `CustomerController` by mocking the `Utils.getString()` and `CustomerDAO.create()` methods.

```
19
20 @RunWith(MockitoJUnitRunner.class)
21 public class CustomerControllerTest {
22
23     @Mock
24     private Utils utils;
25
26     @Mock
27     private CustomerDAO dao;
28
29     @InjectMocks
30     private CustomerController controller;
31
32     @Test
33     public void testCreate() {
34         final String F_NAME = "barry", L_NAME = "scott", email = "Barry.scott@gmail.com";
35         final Customer created = new Customer(F_NAME, L_NAME, email);
36
37         Mockito.when(utils.getString()).thenReturn(F_NAME, L_NAME, email);
38         Mockito.when(dao.create(created)).thenReturn(created);
39
40         assertEquals(created, controller.create());
41
42         Mockito.verify(utils, Mockito.times(3)).getString();
43         Mockito.verify(dao, Mockito.times(1)).create(created);
44     }
45 }
```

The JUnit coverage report shows the following data:

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
> ims	75.8 %	2,990	956	3,946

# Demonstration

- Here is a demo of the system working.

# Sprint Review

The screenshot displays the Jira Software interface for a Kanban board. The browser address bar shows the URL: `qahackworth.atlassian.net/secure/RapidBoard.jspa?projectKey=P1A&rapidView=4`. The Jira navigation bar includes the 'Jira Software' logo, 'Your work' tab, and a 'Projects' dropdown menu. A search bar and a 'Create' button are also present.

The left sidebar contains a list of project items: 'Project 1 - Adam' (Classic software project), 'P1A board' (Board), 'Backlog', 'Kanban board' (selected), 'Reports', 'Issues', 'Components', 'Code', 'Releases', 'Pages', 'Planning Poker', and 'Add item'.

The main content area shows the 'Kanban board' for 'Project 1 - Adam'. The board is divided into three columns: 'SELECTED FOR DEVELOPMENT 4', 'IN PROGRESS 0', and 'DONE 17'. The 'SELECTED FOR DEVELOPMENT' column contains three issues:

- Issue 1: 'As a product owner, I would like to be able to, add an item to an order in the IMS'. Label: 'Orders in IMS'. ID: 'P1A-15'.
- Issue 2: 'Introducing orders entity to the IMS'. Label: 'Orders in IMS'. ID: 'P1A-3'.
- Issue 3: 'As a product owner, I would like to be able to, calculate the total cost of an order in the IMS'. ID: 'P1A-16'.

The 'DONE' column contains three issues:

- Issue 4: 'Introducing customer entity into the IMS'. Label: 'Customers in IMS'. ID: 'P1A-1'.
- Issue 5: 'Introducing item entity to the IMS'. Label: 'Items in IMS'. ID: 'P1A-2'.
- Issue 6: 'As a product owner, I would like to be able to, add an item to the IMS'. Label: 'Items in IMS'.

A 'Quickstart' button is visible in the bottom right corner of the board area. The bottom status bar shows the time '16:59'.



# Sprint Retrospective

The screenshot displays the Jira Software interface for a Kanban board. The top navigation bar includes the Jira logo, 'Your work', and tabs for 'Projects', 'Filters', 'Dashboards', 'People', and 'Apps', along with a 'Create' button. A search bar and user profile icons are on the right. The left sidebar lists project components: 'Project 1 - Adam' (Classic software project), 'P1A board' (Board), 'Backlog', 'Kanban board' (selected), 'Reports', 'Issues', 'Components', 'Code', 'Releases', 'Pages', 'Planning Poker', 'Add item', and 'Project settings'.

The main area shows the 'Kanban board' for 'Project 1 - Adam / P1A board'. It features a search bar, a user profile icon (AH), and filters for 'Only My Issues' and 'Recently Updated'. The board is divided into three columns: 'SELECTED FOR DEVELOPMENT 0', 'IN PROGRESS 0', and 'DONE 21'. The 'DONE' column contains three issues:

- Introducing customer entity into the IMS**: A teal card with the label 'Customers in IMS', a purple icon, an upward arrow, and a status of 'P1A=1'.
- Introducing item entity into the IMS**: A green card with the label 'Items in IMS', a purple icon, an upward arrow, and a status of 'P1A=2'.
- As a product owner, I would like to be able to, add an item to an order in the IMS**: A yellow card with the label 'Orders in IMS', a green icon, an upward arrow, and a status of 'P1A=15'.

A 'Quickstart' button is visible in the bottom right corner of the board area.

# To Conclude

- Enjoyed learning and using the various new software
- Struggled with parts but was able to get guidance
- Managed to successfully build a working IMS which was to the standard of the MVP



Any Questions...?