IMS Project – Inventory Management System

Waleed Tahir

20DecSoftware1

Introduction

Name Waleed Tahir

QA trainee

The overall objective of the project was:

To create a functional application, using supporting tools, methodologies, and technologies, that encapsulates all fundamental modules covered during training.

To Breakdown the project I looked at the Key deliverable features which were the CRUD functionality for the Order, Customer and Item classes and split them into individual user stories for my Kanban Board and then went about completing the required tasks before looking at any further function that could be added to my application.

Consultant Journey

- Version Control – Git,Github/Gitbash
- Database management system – Mysql
- Back-end Programming language – Java,SQL
- Unit testing JUnit and Mockito
- Build Tool Maven
- Project Management platform – Jira-Kanban Board

Continues Integration

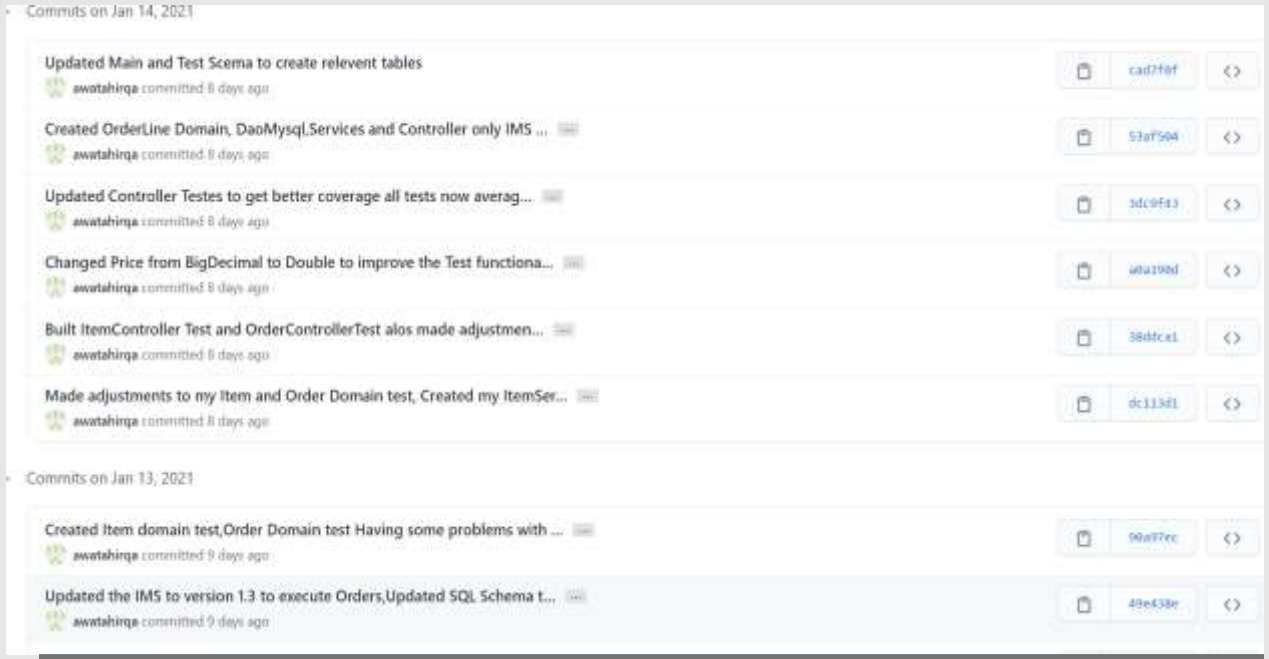
The Version Control System that I used was Git;

- Use a mix of GitBash console and Github to manage iterations of my project working within the Master->Dev->features model to ensure full clarity and safety of my code.
- Regular commits to ensure Code is update operating using swift design principles by ensuring that a copy of my latest working code was always available in my repository.

CI

```
pc@DESKTOP-K4GMEJU MINGW64 ~/Docu
ims-demo (WithoutOline)
$ git branch
  Ordercontroller
  Testing1
* WithoutOline
  dev
  features
  master
```

Branch models



Commit history ensuring regular updates

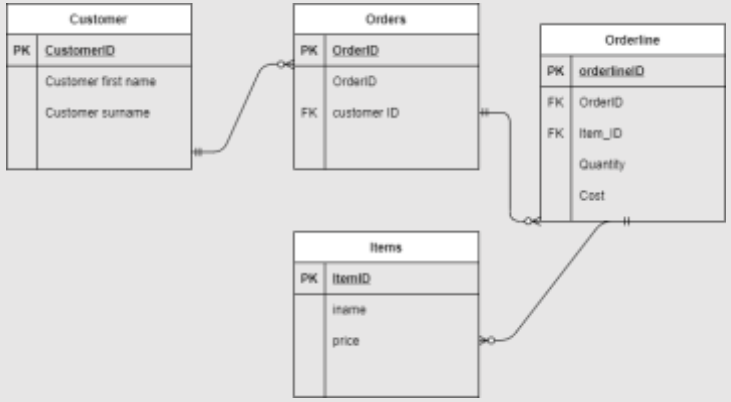
Database management

Code interfacing my Java application with a Database

```
create database if not exists ims;
create table if not exists ims.customers(id int primary key auto_increment, first_name varchar(40), surname varchar(40));
create table if not exists ims.item(id int primary key auto_increment, item_name varchar(40), price DECIMAL(10, 2));
create table if not exists ims.orders(OrderID int primary key auto_increment NOT NULL, CustomerID INT NOT NULL,  FOREIGN KEY (Custo
create table if not exists ims.orderline(orderline_id int PRIMARY KEY  auto_increment NOT NULL, item_id int, order_id int, cost NUM
```

Executing SQL commands
from within my java

```
    try (Connection connection = DriverManager.getConnection(jdbcConnectionUrl, username, password);
            Statement statement = connection.createStatement();) {
        statement.executeUpdate("INSERT into orders(CustomerID) values('" + order.getCustomerID() + "')");
        return readLatest();
```



ERD Diagram showing table relationships

Testing

Testing was completed using a combination of both JUnit and Mockito.

- JUnit was used to test the individual methods within each of my classes by initialising the relevant objects in my test setup and then calling the method to test it.
- Mockito was used to mock objects within my tests when I wanted to use methods that draw their input from dependent classes rather that having a direct data input

JUnit

Mockito

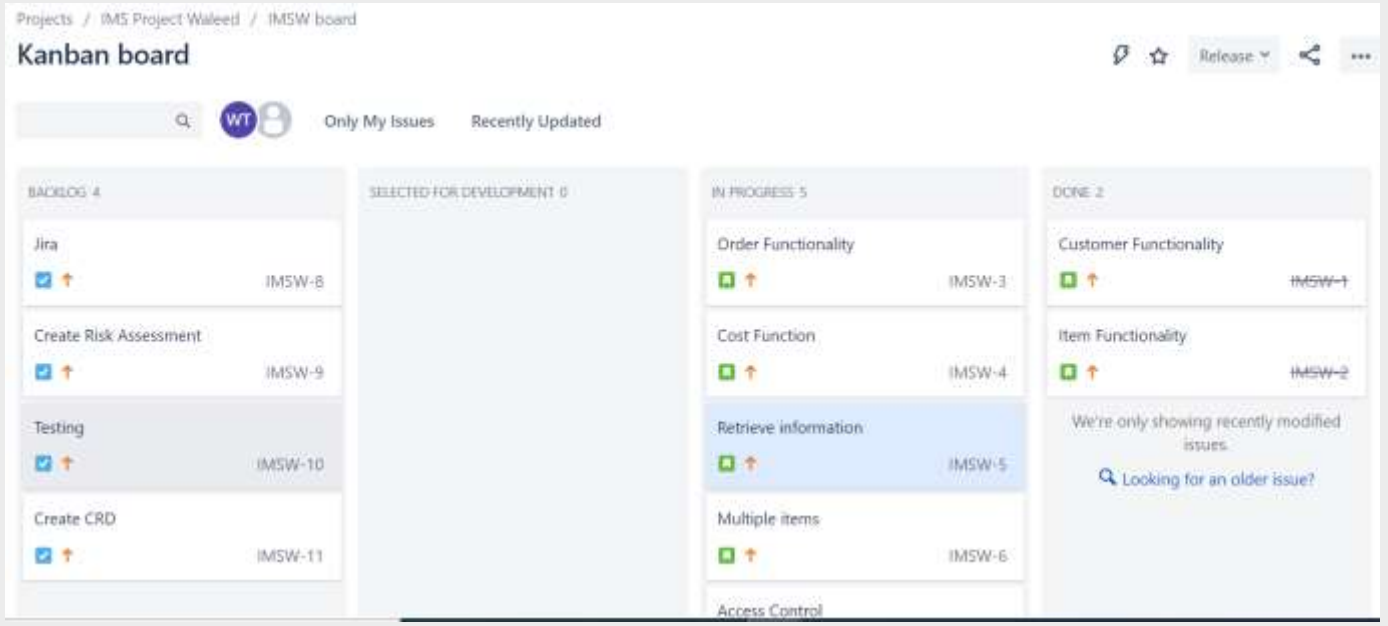
```java
@Test
public void settersTest() {
    assertNotNull(customer.getId());
    assertNotNull(customer.getFirstName());
    assertNotNull(customer.getSurname());

    customer.setId(null);
    assertNull(customer.getId());
    customer.setFirstName(null);
    assertNull(customer.getFirstName());
    customer.setSurname(null);
    assertNull(customer.getSurname());

}
```

```java
@RunWith(MockitoJUnitRunner.class)
public class ItemControllerTest {

    /**
     *  The thing I want to fake functionlity for
     */
    @Mock
    private ItemServices itemServices;

    @Spy
    @InjectMocks
    private ItemController itemController;

@Test
    public void readAllTest() {
        ItemController itemController = new ItemController(itemServices);
        List<Item> items = new ArrayList<>();
        items.add(new Item("Nokia Moon", 599.99));
        items.add(new Item("Samsung Galaxy", 649.99));
        items.add(new Item("Iphone 12", 1200.00));
        Mockito.when(itemServices.readAll()).thenReturn(items);
        assertEquals(items, itemController.readAll());
```

Jira

Jira is one of the leading technologies in agile project management.

With it, it is easy to create a product backlog item, create a sprint etc.



Demonstration

User story : 1 As a user of the application I want to be able to implement Create, Read, Update and Delete functionality on customers so that I can manage my customer information.

User story 2 : As a user I want to be able to implement Create, Read, Update and Delete functionality on items within the Application so that I can flexibly manage my items attributes like price

Sprint Review

- I managed to complete the core functionality of my code;
- CRUD services for Customer
- CRUD services for Item
- CRUD services for Order
- Calculation of cost

- What got left behind;
  - Testing coverage not able to reach 80%
  - Additional functionality such as Users and subsequent access control

Sprint Retrospective

What went well?

- Core code functionality
- Version Control

Improvements

- Adding further functionality to the code
- More efficient/cleaner code
- Higher test coverage

Conclusion

- Core Aim of the Brief met
- I have deepened my understanding of the technologies learnt over the course of the project
- Moving forward continue to improve and as I gain experience complete my project with more functionality to flush out the scope.

- Thank you for Listning are there any Question?