



INVENTORY MANAGEMENT SYSTEM (IMS)

QA INDIVIDUAL PROJECT 1: DANIEL AHUCHOGU

INTRODUCTION

Objectives:

- To create a functional IMS command line application that takes the users inputs to give specific outputs. While connected to a GCP my SQL instance or a JDBC.
- The development of the application using supporting tools and technologies learnt throughout the duration of the course.

Aim:

- To build an application that a user/client can interact with via a Command Line Interface (CLI)
- The application has a JAVA codebase, with a sensible package involved.
- Uses the agile framework during the duration of the project.

CONSULTANT JOURNEY

- The main technologies learnt and used for this project were:
 - Git, SQL, Jira, Java
- Git is a version control system, where it can track changes throughout the project with the Involvement of GitHub.
- SQL/GCP with SQL: is a database management system, to manage data within tables.
- Jira: A project management system, to have the story points for all the user stories and requirements.
- Java: A back end build tool software. Main purpose is an object orientated programming language that is class based.

INITIAL PROJECT MANAGEMENT

- Before the project started I had an initial plan of what needs to be done related to the priority
- I had mock list of the user requirements, user stories and important tasks that I had to do.
- The design of my entity relationship diagram related to their relationship links.
- A risk assessment of complications and unplanned events that could occurred throughout the project.
- Remembering an agile mindset throughout the duration of the project.

Backlog 15 issues		Create sprint	...
<input checked="" type="checkbox"/>	Risk Assessment	DA IP11-1	↑
<input checked="" type="checkbox"/>	Planning the layout of SQL commands that will be implement with Java.	DA IP11-4	↑
<input type="checkbox"/>	As a user I want to add information related to the customer, so I can view them in the database/command line.	IP11-10	↑ 5
<input type="checkbox"/>	As a user I want to use the command line, so I can view information related to the customers in the database.	IP11-11	↑ 5
<input type="checkbox"/>	As a user I want to use the command line, so that I can delete any existing information to the customer details that are wrong.	IP11-13	↑ 5
<input type="checkbox"/>	As a user I want to be able to use the command line, so that I can add items or products to the system.	IP11-14	↑ 5
<input type="checkbox"/>	As a user I want to use the command line, so that I can update any existing information related to the customer	IP11-12	↑ 5
<input type="checkbox"/>	As a user I want to use the command line, so that I can view any existing information related to the items or products.	IP11-15	↑ 5
<input type="checkbox"/>	As a user I want to use the command line, so that I can update any existing information related to the item or products.	IP11-16	↑ 6
<input type="checkbox"/>	As a user I want to use the command line, so that I can delete any existing information related to the item/product that are wrong.	IP11-17	↑ 6
<input type="checkbox"/>	As a user I want to recall information from the item and customer within the command line. So, that I can create an order in the system.	IP11-18	↑ 5
<input type="checkbox"/>	As a user I want to recall existing information from the system related to the item and the customer. So, that I can delete orders that are incorrect in the system.	IP11-23	↑ 4
<input type="checkbox"/>	As a user I want to recall existing information from the system related to the item and the customer. So, that I can view orders in the system.	IP11-19	↑ 4
<input type="checkbox"/>	As a user I want to use the system recall an existing order that has been processed. So, that I can add an item to the order.	IP11-20	↑ 5
<input type="checkbox"/>	As a user I want use the command line system. So, that I can calculate the total cost of an order.	IP11-21	↑ 4
<input type="checkbox"/>	As a user I want to recall information within the system related to the order. So that I can delete an item/product from a processed order.	IP11-22	↑ -

Fig 1: A list of my user stories and user requirements from my Jira backlog

ENTITY RELATIONSHIP DIAGRAMS

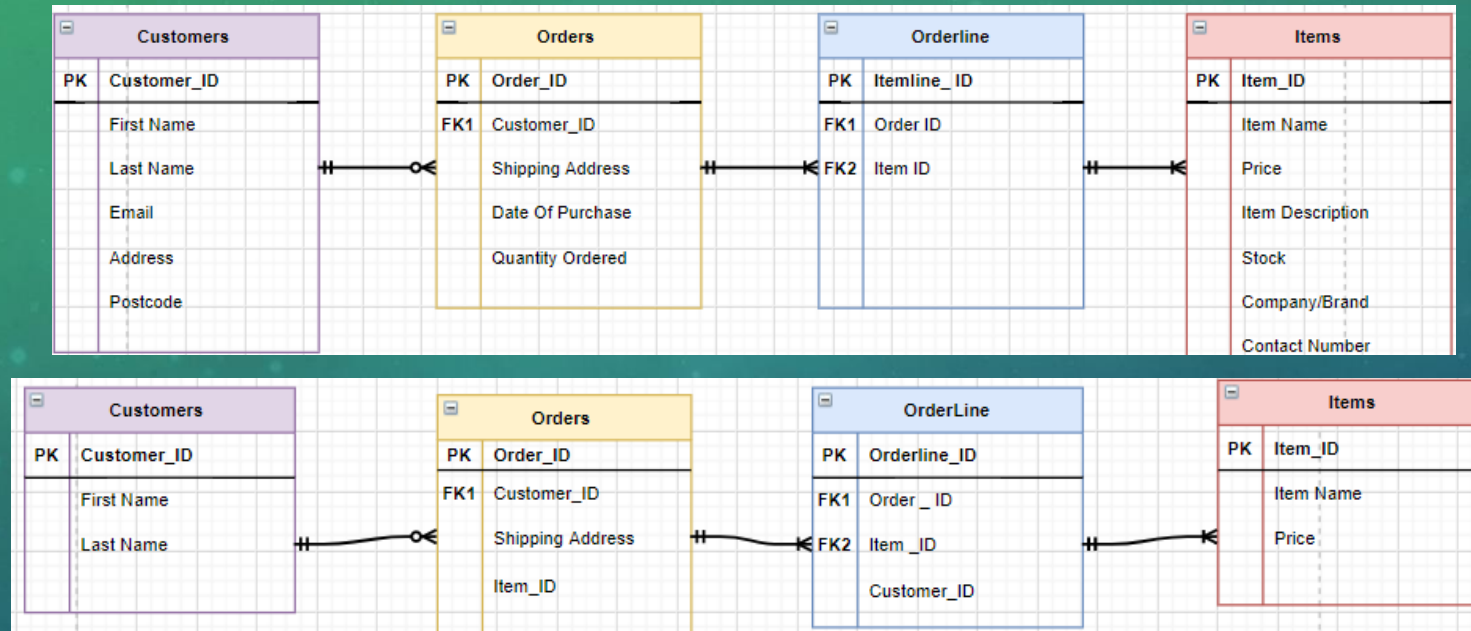


Fig 2: A before and after image of my ERD diagram

- Entity relationship diagram (ERD) shows the relationships of entities stored in a database.
- In this context, the data that needs to be collected is showed above. This ERD models the key information needed for this database.
- We use this model to plan the design of the database. To prevent many –to – many relationships we have intermediate table to handle this.
- Due to the fact an order can contain 1 –to –many items and an item can be part of 0 – to – many orders.

PROJECT MANAGEMENT REVIEW

- From the last slide, that was mock project management that I did before looking at the spec.
- After seeing the specification again and the IMS template. Another scrum meeting and project review had to be done.
- The backlog was updated and an active sprint was done. To outline the tasks and important user story tasks.
- An active sprint was started involving MoSCoW. Which help outline key prioritisation.

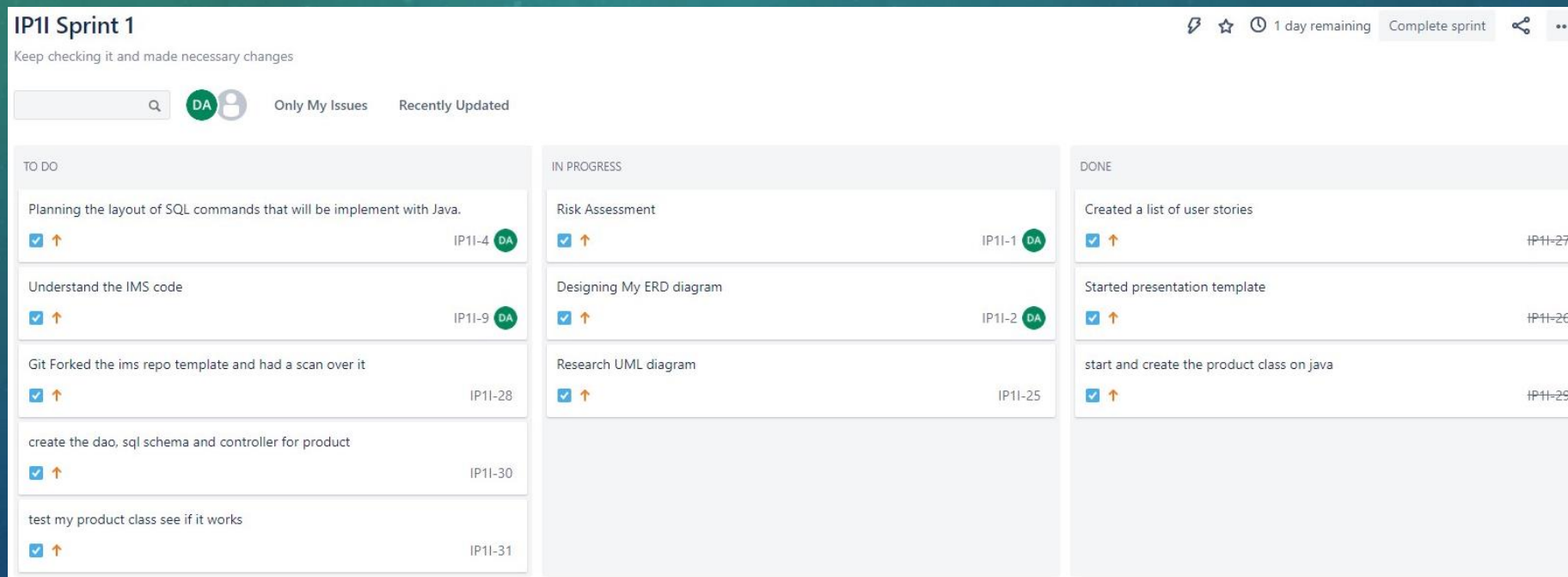


Fig 3: My Jira active sprint Kanban board after my first individual scrum meeting

CONTINUOUS INTEGRATION

- With version control, I approached it as simplest as possible to prevent errors with my code/working.
- Focused working off my DEV branch, to make different feature branches that had different aspects of my work.
- Essentially I didn't want work on my main/master branches.
- Want to minimised has much errors and merged conflicts as possible.
- Due to different features are combined, or integrated, frequently occasionally to expose external coding issues.
- I had multiple branches, because during development you might stop working on a piece of work and restart it on a different branch and works better..

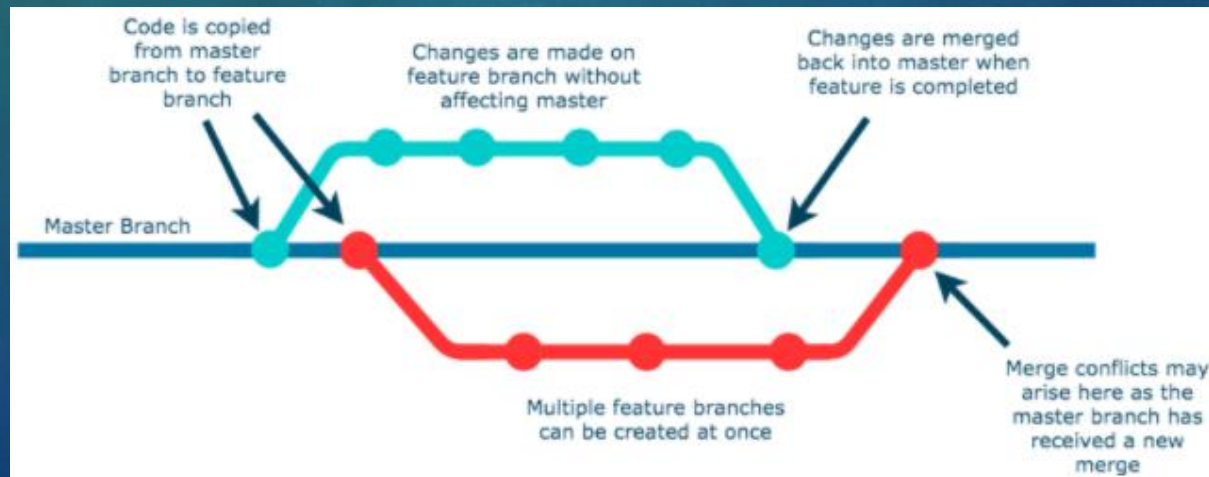


Fig 4: Branch model that I followed

TESTING

- In terms of testing, it's important to test the quality of the code. This means less chance of the code breaking. Helps determine and identify possible errors.
- Need to test the functionality of the code and the coverage the code can take. To see if it meets our expectations or initial prediction .
- JUnit or Unit testing can be an automated process designed to reuse code and provide results after a test.
- From the JUnit test, I had a coverage of 61.6% on the src/main/java folder.

```
77 }
78 @Test
79 public void updateTest() {
80     CustomerDaoMysql customerDaoMysql = new CustomerDaoMysql(jdbcConnectionUrl, username, password);
81     String firstName = "chris";
82     String surname = "perrins";
83     Customer customer = customerDaoMysql.create( new Customer(firstName, surname));
84     customer.setFirstName("james");
85     Customer update = customerDaoMysql.update(customer);
86     assertEquals(customer, update);
87 }
88
89 @Test
90 public void deleteTest() {
91     CustomerDaoMysql customerDaoMysql = new CustomerDaoMysql(jdbcConnectionUrl, username, password);
92     Long id = 1L;
93     customerDaoMysql.delete(1L);
94 }
95
96
97
```

Element	Coverage	Vered Instructions	Missed Instructions	Total Instructions
Daniel IMS Project	84.7 %	5,522	998	6,520
src/main/java	72.3 %	2,438	936	3,374
src/test/java	98.0 %	3,084	62	3,146

```
50 // assertEquals(order, orderController.create());
51 // }
52
53 @Test
54 public void UpdateTest() {
55     Long orderId = 1L;
56     String shippingAddress ="Help";
57     Long itemId = 1L;
58     Long customerId = 1L;
59     Mockito.doReturn(shippingAddress).when(orderController).getInput();
60     Mockito.doReturn(orderId, itemId, customerId).when(orderController).getLongInput();
61     Order order = new Order(1L, shippingAddress, 1L, 1L);
62     Mockito.when(orderServices.update(order)).thenReturn(order);
63     assertEquals(order, orderController.update...
```

Element	Coverage	Vered Instructions	Missed Instructions	Total Instructions
Daniel IMS Project	78.5 %	4,837	1,325	6,162
src/main/java	61.6 %	2,079	1,295	3,374
src/test/java	98.9 %	2,758	30	2,788

Fig 5: Before and after images for coverage whilst doing test coverage

DEMONSTRATION OF THE PROJECT

- With some of the user stories I am going to show a piece of my project that is working. After using a Maven Build tool.
- Through a Command Line System, because essentially that was the Minimum Value Product (MVP).

```
Order.java  OrderServices.java  OrderController.java x  CustomerController.java
17 public OrderController(CrudServices<Order> orderService) {
18     this.orderService = orderService;
19 }
20
21 String getInput() {
22     return Utils.getInput();
23 }
24
25 @Override
26 public List<Order> readAll() {
27
28
29
30     return null;
31 }
32
33 @Override
34 public Order create() {
35     // TODO Auto-generated method stub
36     return null;
37 }
38
39 @Override
40 public Order update() {
41     // TODO Auto-generated method stub
42     return null;
43 }
44
```

```
90 // return order;
91 // }
92
93 @Override
94 public Order create() {
95     List<Order> item = new ArrayList<>();
96     // LOGGER.info("May you enter the address you want the order to be delivered to");
97     // String shippingAddress = getInput();
98     // LOGGER.info("May you enter the customer id");
99     Long customerId = getLongInput();
100     Order order = new Order(customerId);
101     item.add(order);
102     // LOGGER.info("Please may you enter the item id please");
103     Long itemId = getLongInput();
104     boolean addItem = true;
105     while (addItem) {
106         // LOGGER.info("Please re-enter your item id");
107         itemId = getLongInput();
108         // LOGGER.info("Enter the shipping/order address");
109         String shippingAddress = getInput();
110         order = orderService.create(new Order(shippingAddress, customerId, itemId));
111         item.add(order);
112         // LOGGER.info("Order Created");
113         // LOGGER.info("Is that all the items that you want add?");
114         if (addItem == true) {
115             // LOGGER.info("Query finished");
116             if (addItem == false) {
117                 }
118             }
119     }
120 }
```

Fig 6: Before and after images for developing an order method for create

UML

- A UML diagram shows the relationships between the different classes, attributes, method and relationship between different objects in a java program.
- UML is showing the basic blueprint of the programme and the design of it.

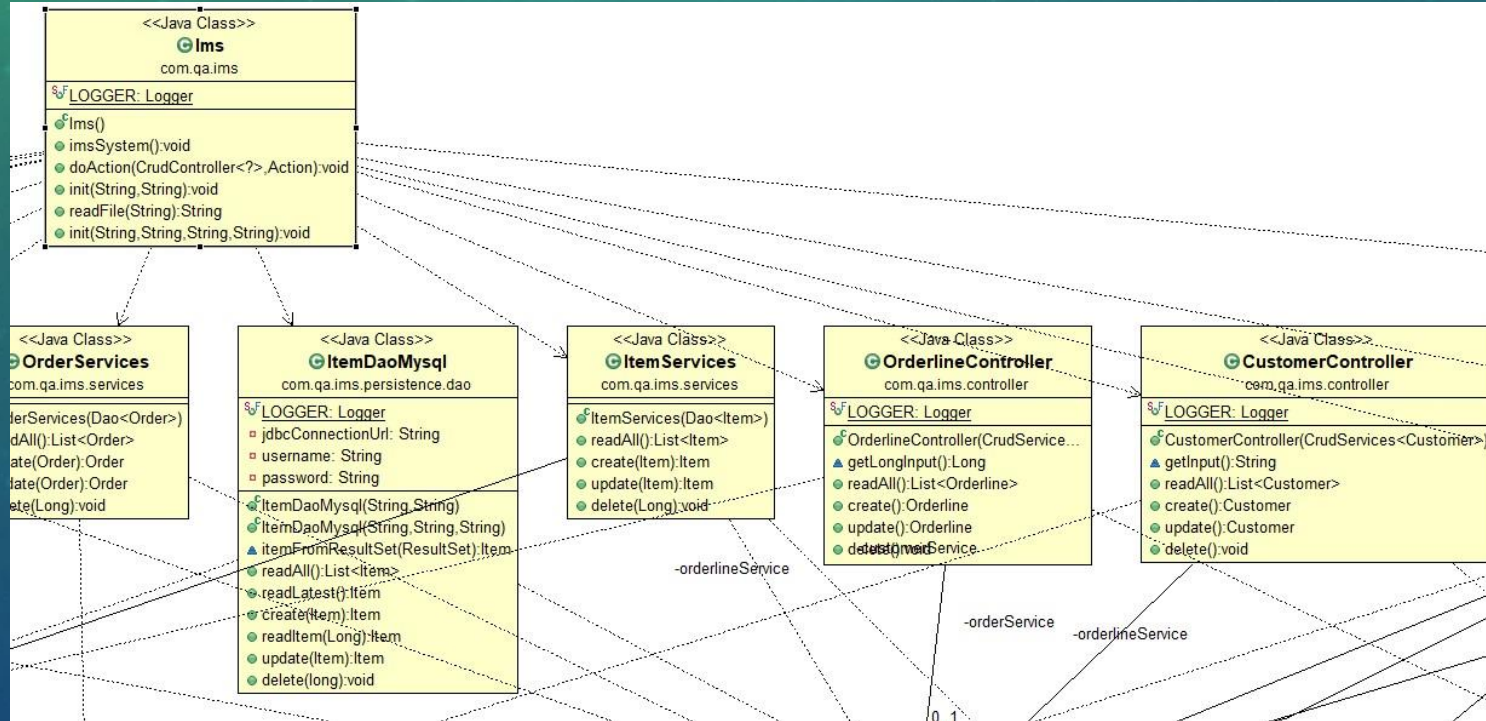


Fig 7 :A section of my UML diagram

SPRINT REVIEW

What did you complete:

- In general the CRUD functionality for customers, orders and items.
- Connects to a local MySQL instance.
- Testing and coverage testing.
- Git repository with the feature branch model, with a working jar file.

What got left behind:

- Calculate cost for orders
- Relationship between orders and orderline.
- Some aspects of my testing. For example DaoTest is not completed to a high standard.
- Connecting to a GCP instance to see if the application works.

SPRINT RETROSPECTIVE

What went well:

- What went well was the development of items by implementing CRUD and SQL functions with Java.
- The creation of my UML diagram, ERD and some aspects of project management.
- Certain aspects of JUnit testing and how I handled my cloud integration/version control.

What could be improved:

- Some of my methods/code for orders and orderline could be done better to have the correct relationship between them.
- Spending more time on my JUnit test to solve, assertion errors and other type of errors/failures to increase coverage.
- Test if my code works on a local GCP earlier.
- My README file could be edited and be written more effectively.

SUMMARY

- To reflect on the project, as a whole the project went well to a certain extent.
- There were certain challenges and risks that occurred.
- At start the start I was very overwhelmed of the scale of the project because I had no knowledge of any coding such as Git and Java.
- But with an agile mindset I was able to adapt to the situation and I was able to work effectively.
- Able to achieve most of what the specification required/the MVP

Final Thoughts:

- This was a tough project for someone who has a no prior coding experience.
- This is a learn step for me to gain new skill/developed new skills that I have required.



QUESTIONS ?