Inventory Management System

Fatima Sheikh-Nur

Intro

- Aspiring software developer/engineer.
- Career switcher: from Biomedical Science to Tech.
- Experience in JavaScript, as well as HTML, CSS used resources such as freecodecamp, as well as CodeFirstGirls.

- Code fully integrated into a Version Control System using the feature-branch model: **main/dev/multiple features**.
- A project management board with full expansion on user stories, acceptance criteria and tasks needed to complete the project.
- A risk assessment which outlines the issues and risks faced during the project timeframe.
- A relational database used to persist data for the project, containing the **customers, items, orders, and orders_items** tables. Relationships should be modelled using an ERD.
- A functional application ‘back-end’, following best practices and design principles, in the language that you have covered during training, meeting the requirements set on your project management board.
- A build of your application, including any dependencies it might need, produced using an integrated build tool.
- Unit tests for validation of the application. You should aim to reach the industry standard of **80%** test coverage.

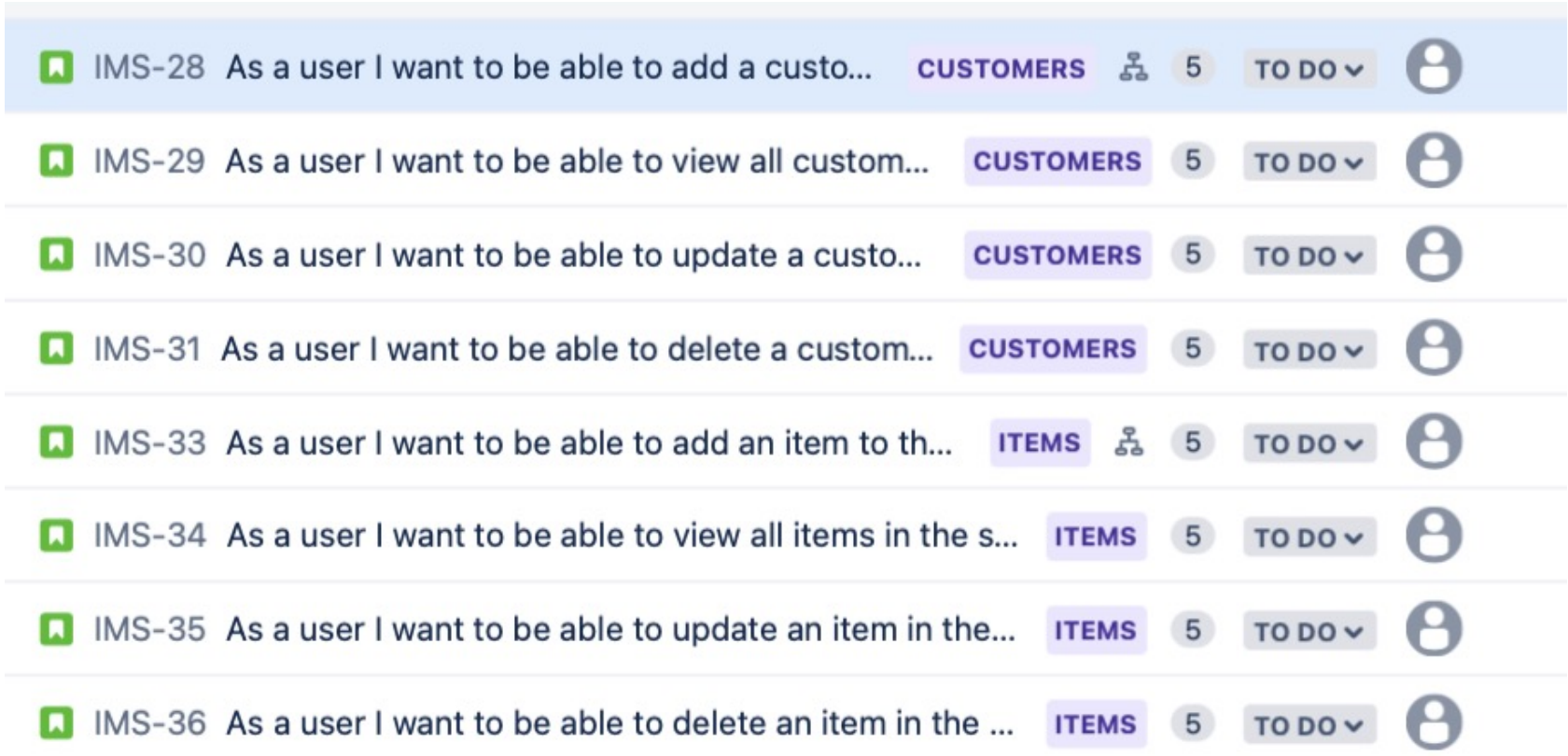
IMS Project:

- A web application app which allows you to create, update, read and delete files.
- Using CRUD
- Entities include customer, items, and orders entities.
- It is usually used in the manufacturing industry but can be used in other sectors also.

Jira

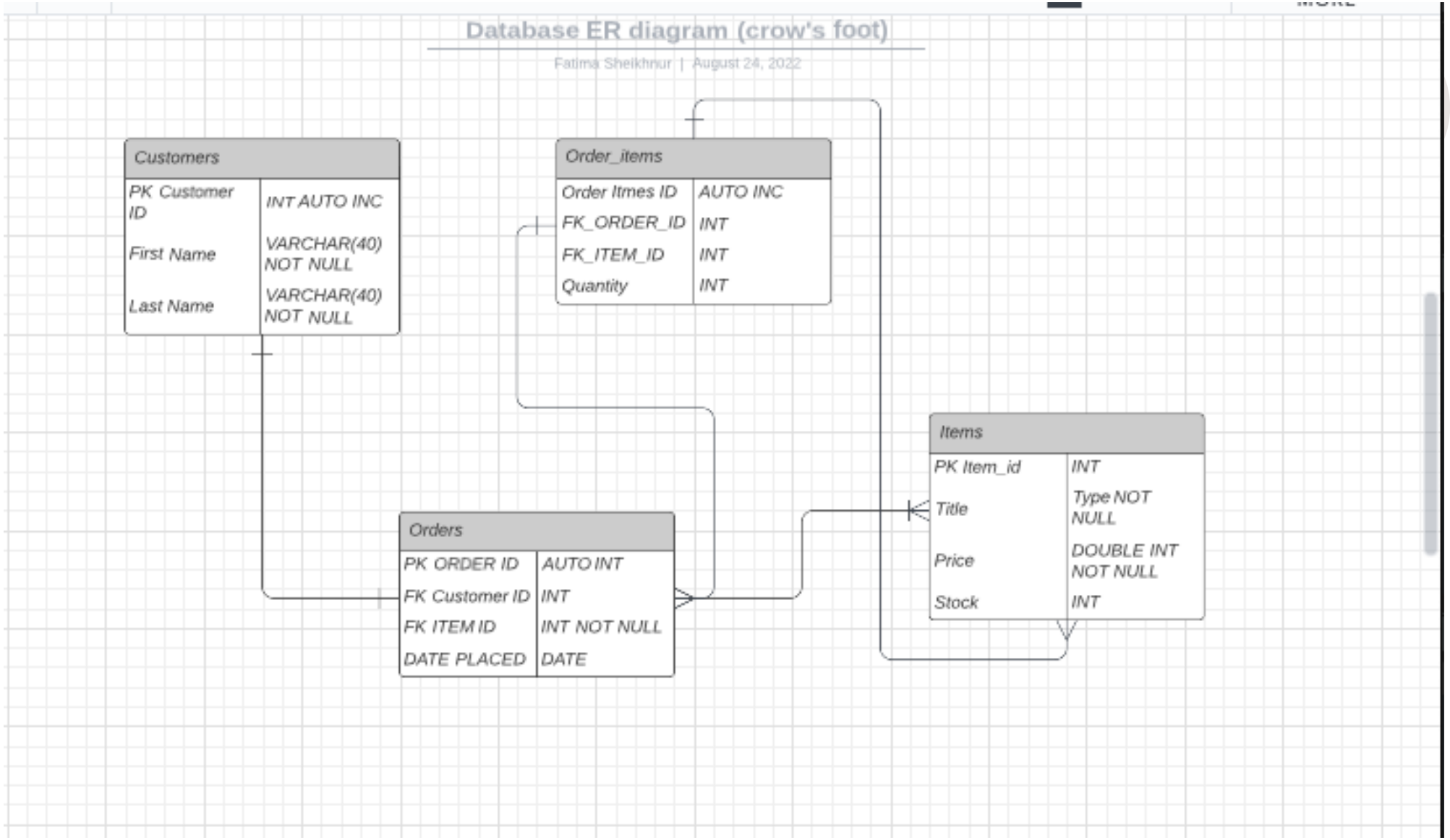
- I used Jira to map out my plan for the project.
- Jira allowed me to create user stories in which I would be able to determine what was crucial to the user experience and what they would want from the app.
- I also made a separate sprint in order to log my specifications for the projects and targets that were needed to be met.

User Stories;



ERD Diagram

- I used an ERD diagram (Entity Relationship Diagram) in order to build my tables seeing how each table interacted with each other and the type of data they were as this would be needed to be cited throughout the project.


Database ER diagram (crow's foot)
Fatima Sheikhnur | August 24, 2022
Customers
PK Customer ID
INT AUTO INC
First Name
VARCHAR(40) NOT NULL
Last Name
VARCHAR(40) NOT NULL
Order_items
Order Itmes ID
AUTO INC
FK_ORDER_ID
INT
FK_ITEM_ID
INT
Quantity
INT
Items
PK Item_id
INT
Title
Type NOT NULL
Price
DOUBLE INT NOT NULL
Stock
INT
Orders
PK ORDER ID
AUTO INT
FK Customer ID
INT
FK ITEM ID
INT NOT NULL
DATE PLACED
DATE

Technologies:

Git: https://git-scm.com- I used git in order to update my work and share it as well as have a backup in case anything was sorrupted. I made sure to commit and push consistently.

Eclipse IDE for JAVA:https://eclipseide.org- I used eclipse for java and made sure to use the correct syntax, I was able to learn how to execute the different commands, as well as use CRUD in order to make my app as functional as possible.

Java: https://www.java.com/en/-- I made sure to use JAVA principles such as SOLID, and CRUD. I made sure to use methods, as well as constructors, as well as hash codes, and getters and setters.

Maven: https://maven.apache.org- I learned how to add maven to eclipse and used it to build the project as it is easy to maintain.

MySQL: https://www.mysql.com- I used SQL to make my main database which was used and referenced across the whole project.

JUNIT- This is a testing tool for each unit of code specifically, which was used to improve and perfect the project.

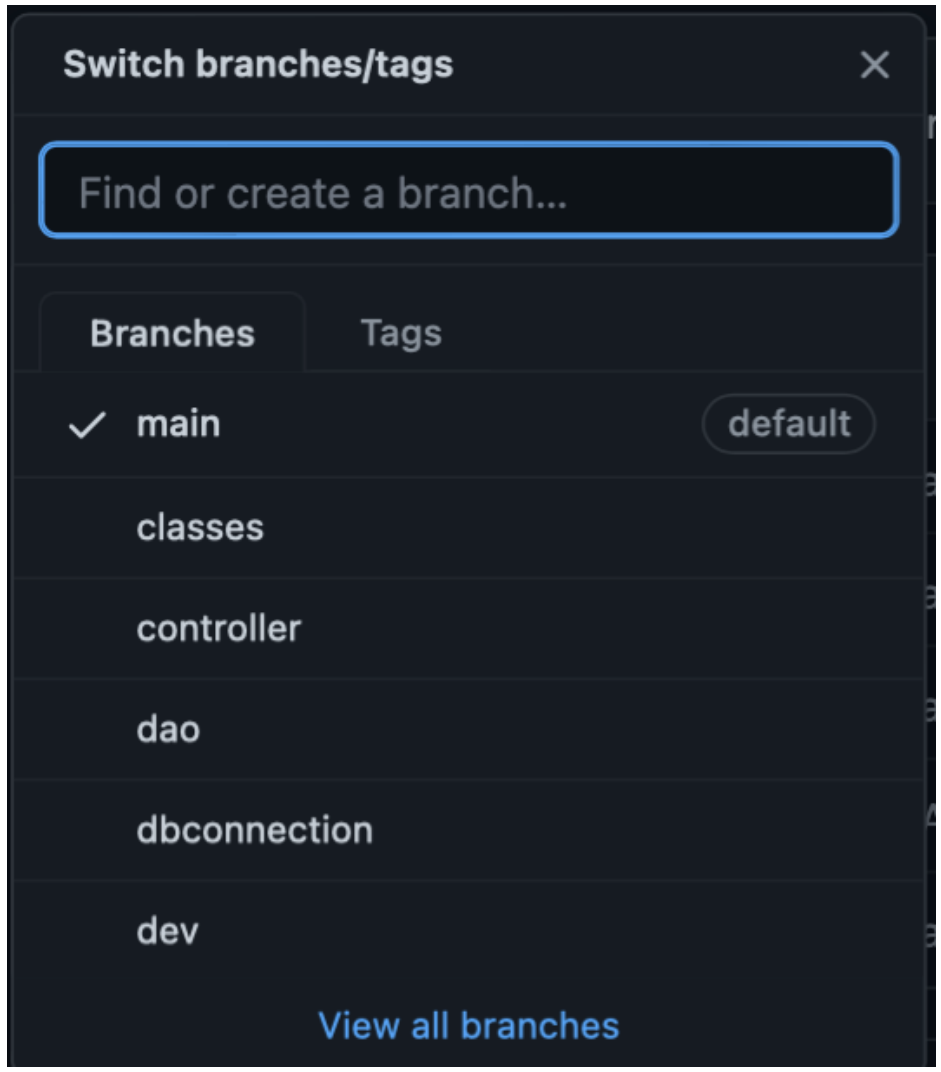
MOCKITO- This is another testing tool which was used to mock the project in order to confirm the running of the project.

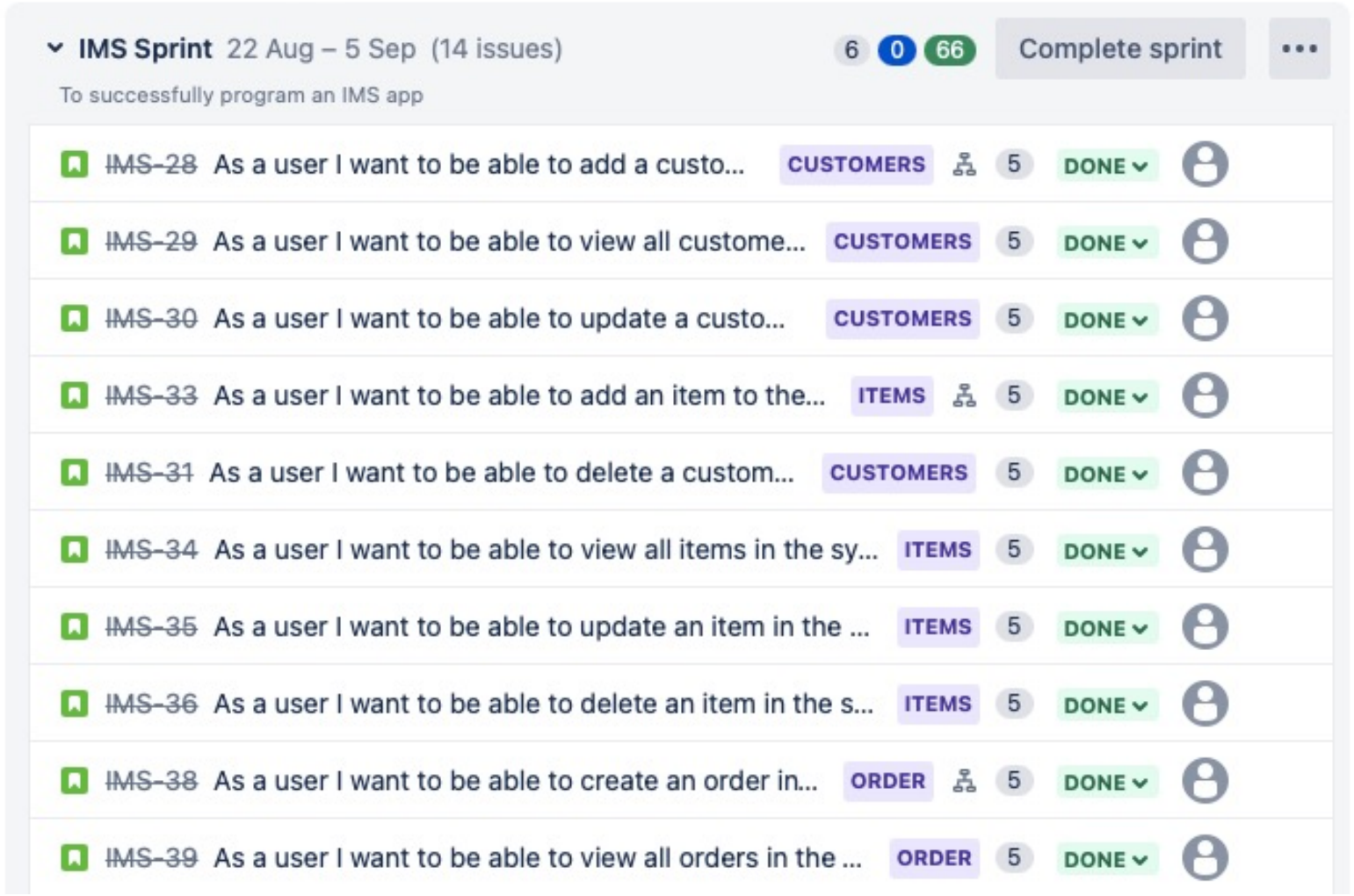
Version Control:

For version control I used GitHub, and tried to commit my changes automatically, using team>push>commit consistently. This is so that I can see the original script if I have made any changes as backup.

I also made sure to write concise commit messages that were clear, making it easier for the reviewer and myself.

I also created multiple branches in order to make it easier to navigate.


Switch branches/tags
Find or create a branch...
Branches
Tags
main
default
classes
controller
dao
dbconnection
dev
View all branches

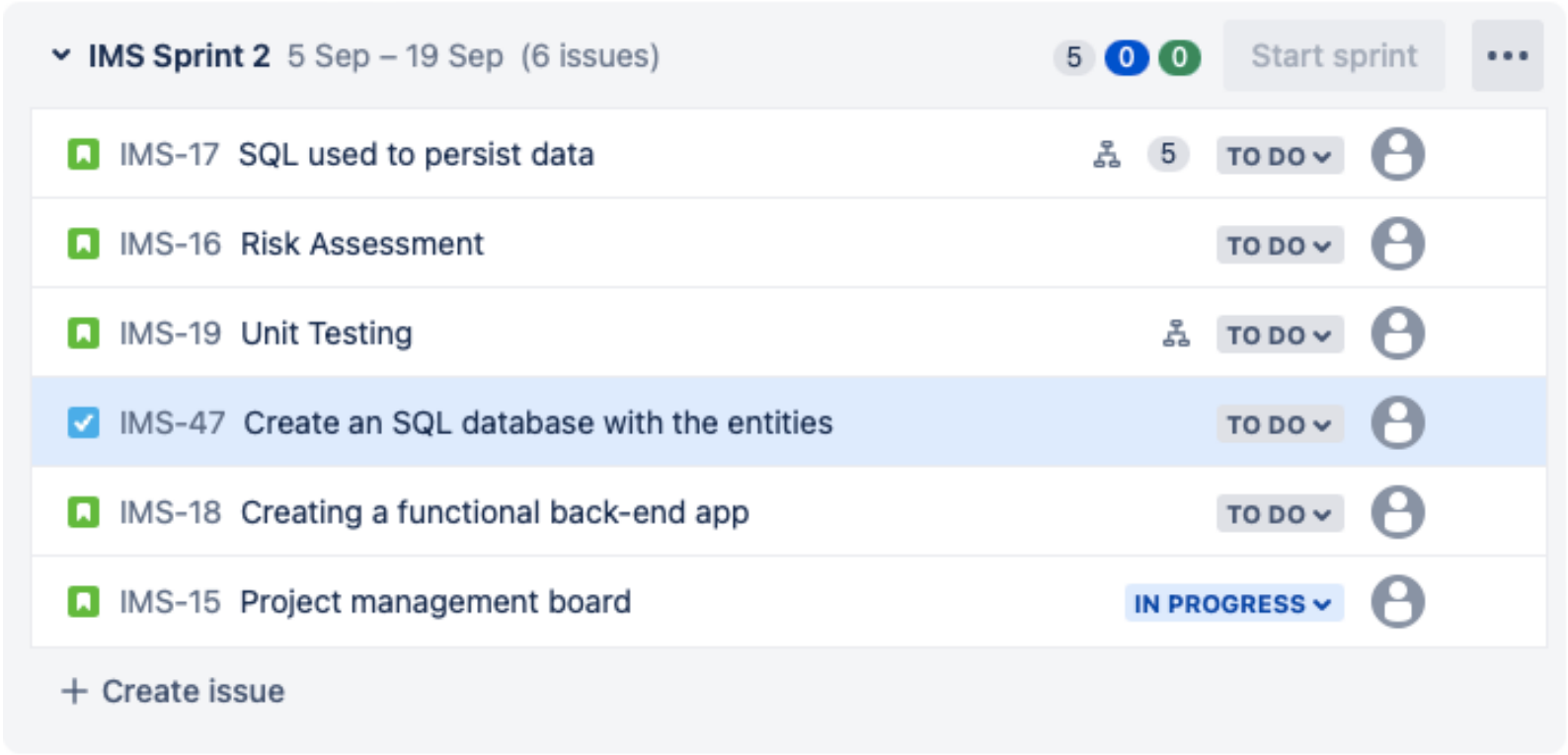

IMS Sprint 22 Aug – 5 Sep (14 issues)
6 0 66
Complete sprint
To successfully program an IMS app
IMS-28 As a user I want to be able to add a custo... CUSTOMERS 5 DONE
IMS-29 As a user I want to be able to view all custome... CUSTOMERS 5 DONE
IMS-30 As a user I want to be able to update a custo... CUSTOMERS 5 DONE
IMS-33 As a user I want to be able to add an item to the... ITEMS 5 DONE
IMS-31 As a user I want to be able to delete a custom... CUSTOMERS 5 DONE
IMS-34 As a user I want to be able to view all items in the sy... ITEMS 5 DONE
IMS-35 As a user I want to be able to update an item in the ... ITEMS 5 DONE
IMS-36 As a user I want to be able to delete an item in the s... ITEMS 5 DONE
IMS-38 As a user I want to be able to create an order in... ORDER 5 DONE
IMS-39 As a user I want to be able to view all orders in the ... ORDER 5 DONE

What went well? What could have been improved.

I have been able to regularly commit changes to my work. I have also been able to make the Jira board, risk assessment and the ERD diagram well.

However, I jumped into the code, and did a little bit of each class and part of specification and did not manage to consistently keep track of the code as I made mistakes such as not consistently having the same thing with the same name. This led to a lot of editing and a lot of cleaning up.

Sprint 2:


IMS Sprint 2 5 Sep – 19 Sep (6 issues)
5 0 0
Start sprint
IMS-17 SQL used to persist data
5
TO DO
IMS-16 Risk Assessment
TO DO
IMS-19 Unit Testing
TO DO
IMS-47 Create an SQL database with the entities
TO DO
IMS-18 Creating a functional back-end app
TO DO
IMS-15 Project management board
IN PROGRESS
Create issue

MYSQL table

```
2
3  CREATE TABLE customers (
4  Customer_ID INT auto_increment,
5  First_Name VARCHAR(40) NOT NULL,
6  SURNAME VARCHAR(40) NOT NULL,
7  PRIMARY KEY (Customer_ID)
8   );
9
10 CREATE TABLE orders (
11 Order_ID INT auto_increment,
12 Date_Placed DATE,
13 fk_customer_id INT NOT NULL,
14 Total Decimal(10, 2),
15 PRIMARY KEY (Order_ID), foreign key (fk_customer_id) REFERENCES customers (customer_id)
16 );
17
18 CREATE TABLE items (
19 Item_ID int PRIMARY KEY auto_increment NOT NULL,
20 Title VARCHAR(80) NOT NULL,
21 Price Decimal(10, 2) NOT NULL,
22 Stock INT
23 );
24
25 CREATE TABLE order_items (
26 Order_Item_ID INT primary key,
27 Quantity INT,
28 fk_item_id INT NOT NULL,
```