22APRENABLE1
THOMAS AYLING

IMS Project

Intro

The task is to create an Inventory Management System with the following requirements:

- The IMS should have at least three tables:
  - Customers
  - Items
  - Orders
- Each with CRUD functionality.
- The IMS should have a CLI for users to interact with the tables in their desired way.
- My version will also have another table to ensure the scope of the project can be fulfilled.

The scope:

- I should provide full documentation including:
  - Jira board with Epics, Tasks and User Stories.
  - ERDs and UMLs to show the architecture of my database and software.
  - A risk assessment that outlines risks and issues faced during the project.
- I should strive to reach a coverage of 80% in my testing phase.
- Use GitHub to manage version control and create a feature branch model.

Risk Assessment

Risk	Statement	Response	Objective	Likelihood	Impact	Risk Level	Recognition Date
RSI	I could develop RSI	Ensure to stretch regularly	Ensure I don't develop RSI which Could impact my future	Very Unlikely	Minor	1	29/04/2022
Coming across new, Unfamiliar technologies	Coming across technologies or techniques that I don't understand could cause delays in my project.	Keep up to date with new coding techniques that could benefit me on my Project.	Try to keep up to date and informed on the techniques and technologies that I will be using throughout my project and if uncertain, research solutions to my problems and learn Effective solutions.	Moderate	Minor	3	29/04/2022
Not completing all requirements as expected	I could misinterpret a requirement which would cause my project to not work as expected.	Ensure to be thorough when reading through the requirements for my project.	Be thorough and specific in my tasks And ensure that I have the correct understanding of each requirement with my trainer before starting my	Unlikely	Minor	2	29/04/2022
Becoming fatigued	Becoming fatigues while working could cause my productivity levels to drop resulting in not completing my project before the deadline.	Ensure to take steps to minimise the risk of becoming fatigued	Use 'eye saver' mode on my monitor to reduce strain on my eyes. Make sure to sleep well and eat healthily to make sure I'm feeling energised before and throughout the day Make sure to take breaks throughout the day and get fresh air to relax my Mind.	Moderate	Minor	3	29/04/2022
Running out of time to Complete my project	Running out of time could cause me to have to submit my project when it is incomplete.	Ensure to manage my time effectively and create achievable goals to complete my work before the deadline	Create time goals for my tasks on my Kanban board to ensure that I stay on track and don't waste too much time on one task.	Moderate	Major	4	29/04/2022
Computer Malfunction	My computer not working as expected could cause me to not be able to continue working on my Project.	Ensure to check my computer health and dust build up regularly.	Monitor computer health regularly and ensure the area around my computer is clean and dust free to reduce the risk of issues such as overheating.	Very Unlikely	Catastrophic	6	29/04/2022
Loss of internet Connection	Loss of internet would prevent me from pushing and pulling from GitHub preventing me from backing my work up, and I would also not be able to contact anyone Through Teams.	Ensure I have a reliable ISP and have data available on my phone as a backup to contact my trainer through teams if necessary.	Regularly push and pull from GitHub To ensure that my work is up to date both locally and in my online Repository.	Unlikely	Hazardous	5	29/04/2022
Power cut	A power cut could cause me to lose work from my project.	Ensure to save my work at regular intervals to reduce the risk of work loss	Regularly save my work and push to GitHub to ensure that if a power cut does occur, I won't loose a large Amount of work	Unlikely	Hazardous	5	29/04/2022
Losing My Work	Loss of work will affect the entirety of my project and cause me to have To restart my project.	Ensure to save regularly and upload my work to my repository at every step of the process.	Regularly push my work to GitHub and keep my work saved in multiple secure locations to decrease the Likelihood of loosing all of my work.	Unlikely	Catastrophic	7	29/04/2022

Risk Matrices						
	Negligible	Minor	Major	Hazardous	Catastrophic	< Impact
Very Unlikely	Low	Low	Low medium	Medium	Medium	
Unlikely	Low	Low Medium	Low medium	Medium	Medium HIgh	
Moderate	Low	Low Medium	Medium	Medium High	Medium HIgh	
Likely	Low	Low medium	Medium	Medium high	High	
Very Likely	Low medium	Medium	Medium high	High	High	
^ Likelihood						

- Each risk has a risk level that shows the impact they would have on the completion of the project if they happened.
- I have also included measures to reduce the chance of each risk occurring.
- My risk matrix allowed me to gain a better understanding of the severity of each risk.

MoSCoW

- I created a document to outline what to prioritize when creating my software.
- This assisted the creation of my Jira board and allowed me to manage my sprints to ensure that my MVP was fulfilled before working on any extra functionality.

I decided that my software:

Must:

- Have a **customers** table.
- Have an **items** table.
- Have an **orders** table.
- Have an **items_ordered** table.
- Have the ability to **add**, **view** (all customers), **update** and **delete** from the **customers** table.
- Have the ability to **add**, **view** (all items), **update** and **delete** from the **items** table.
- Have the ability to **add**, **view** (all orders), **update** (add an item to an order) and **delete** from the **orders** table.
- Have the ability to **add** and **delete** items from the **items_ordered** table.
- Have the ability to **calculate the cost for an order**.
- Have the ability to **delete an individual item** from an **order**.
- Contain a fat .jar in the root folder

Should:

- Have the ability to **view individual customers** from the **customers** table.
- Have the ability to **view individual items** from the **items** table.
- Have the ability to **view individual orders** from the **orders** table.
- Follow best practices.

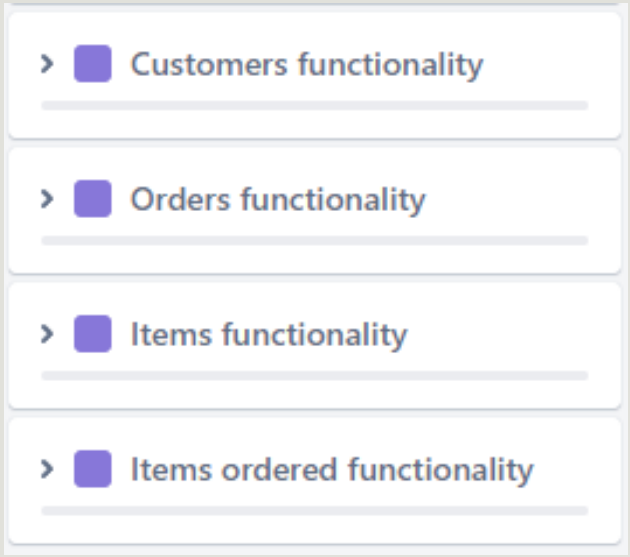
Could:

- Have a **users** table.
- Have the ability to **add**, **view** (all & single users), **update** and **delete** from the **users** table.
- Have a GUI to make the system easier to use.

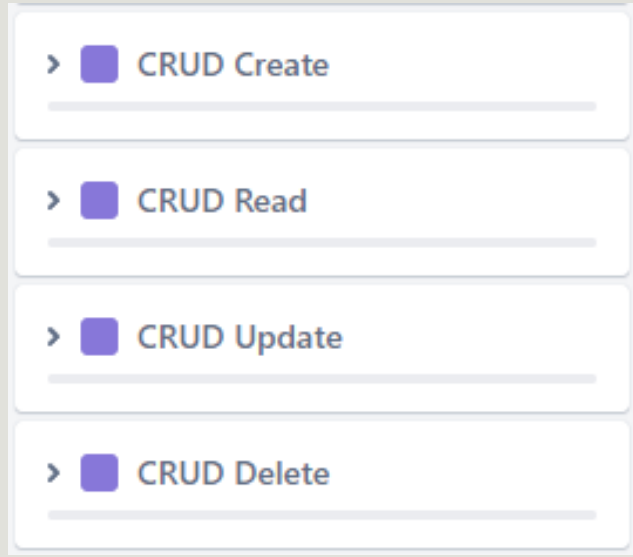
Won't:

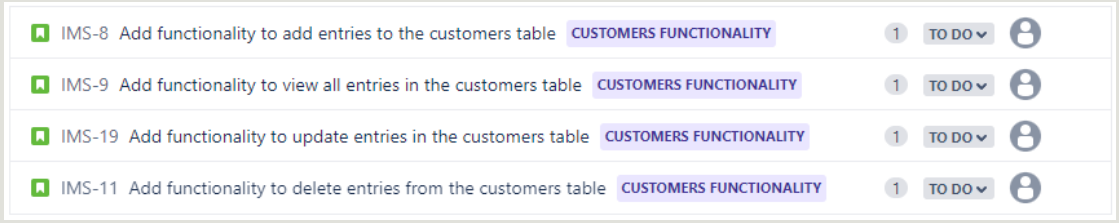
- Be missing documentation
- Contain build-generated files and folders

Jira

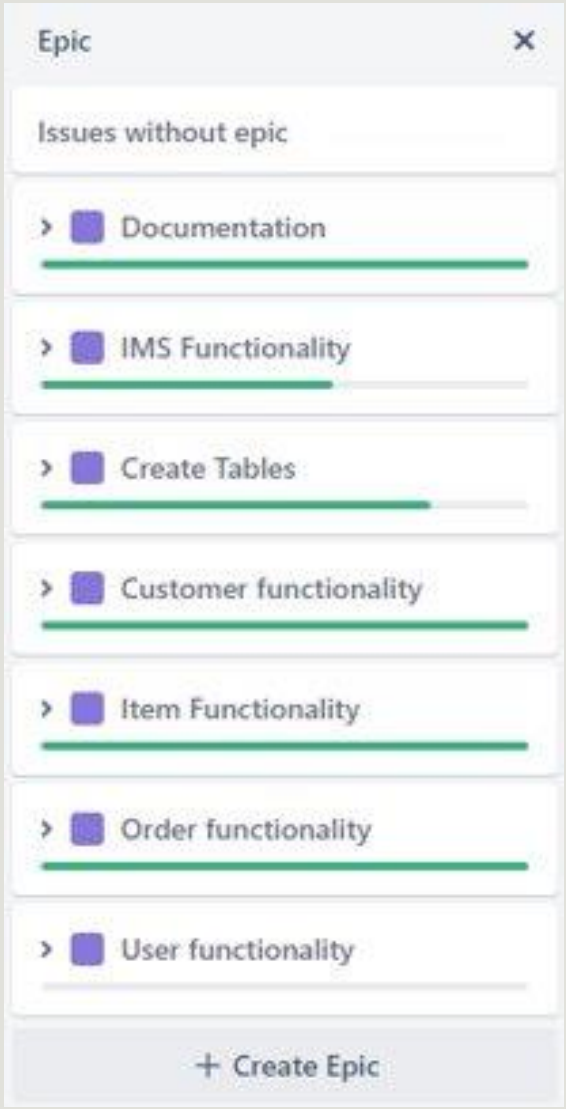


- My Jira board had a few possible structures
- I thought about the development life cycle and decided that working on each the functionality of each table seperately would be a better way of ensuring that my dev branch had functioning code on it at most times.





The final version of my Jira board had the following epics



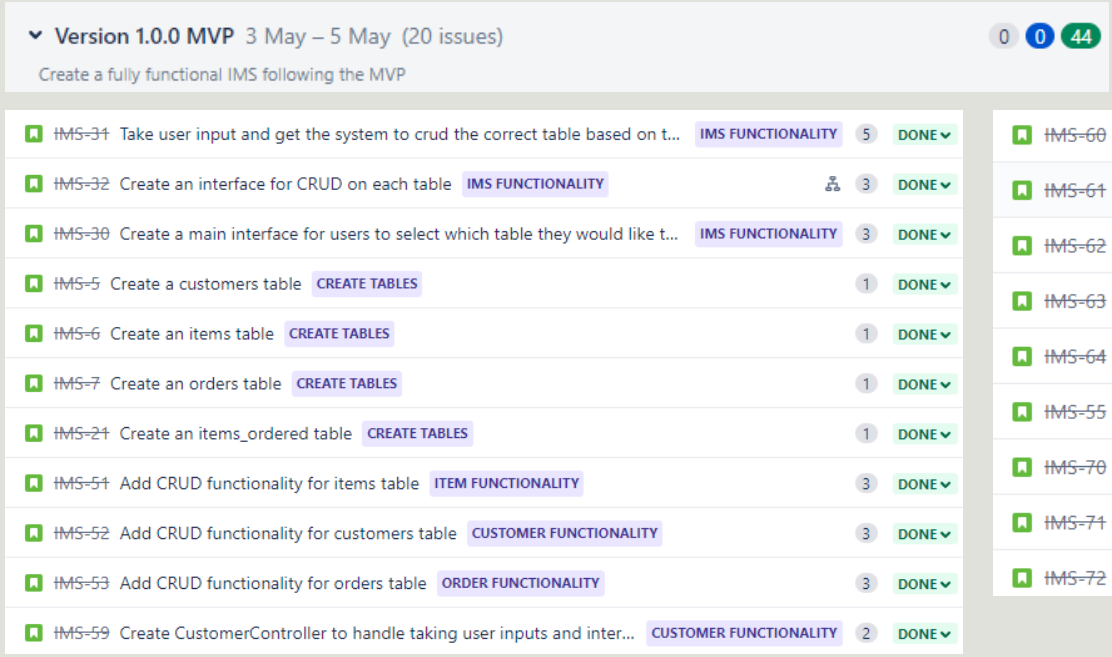
Epic 1: Documentation to be created before coding starts	IMS-70 Create an ERD to describe the architecture of the database DOCUMENTATION 1 DONE IMS-71 Create a UML to describe the structure of the software DOCUMENTATION 3 DONE IMS-72 Create a risk assessment for the project DOCUMENTATION 5 DONE
Epic 2: General IMS functionality to ensure the system works within the scope	IMS-31 Take user input and get the system to crud the correct table based on the inp... IMS FUNCTIONALITY 5 DONE IMS-32 Create an interface for CRUD on each table IMS FUNCTIONALITY 3 DONE IMS-30 Create a main interface for users to select which table they would like to crud IMS FUNCTIONALITY 3 DONE
Epic 3: Create a schema with the necessary tables.	IMS-5 Create a customers table CREATE TABLES 1 DONE IMS-6 Create an items table CREATE TABLES 1 DONE IMS-7 Create an orders table CREATE TABLES 1 DONE IMS-21 Create an items_ordered table CREATE TABLES 1 DONE
Epic 4: Feature 1: version 0.0.2 Add functionality for the items table	IMS-51 Add CRUD functionality for items table ITEM FUNCTIONALITY 3 DONE IMS-60 Create ItemController to handle taking user inputs and interacting with the ... ITEM FUNCTIONALITY 2 DONE IMS-64 Create item object to store item information when adding to or reading fro... ITEM FUNCTIONALITY 1 DONE
Epic 5: Feature 2: version 0.0.3 Add functionality for the orders table	IMS-53 Add CRUD functionality for orders table ORDER FUNCTIONALITY 3 DONE IMS-61 Create OrderController to handle taking user inputs and interacting with the DAO in the intended way ORDER FUNCTIONALITY 2 DONE IMS-63 Create order object to store order information when adding to or reading from table ORDER FUNCTIONALITY 1 DONE IMS-55 Add ability to add and remove single items from an order ORDER FUNCTIONALITY 2 DONE

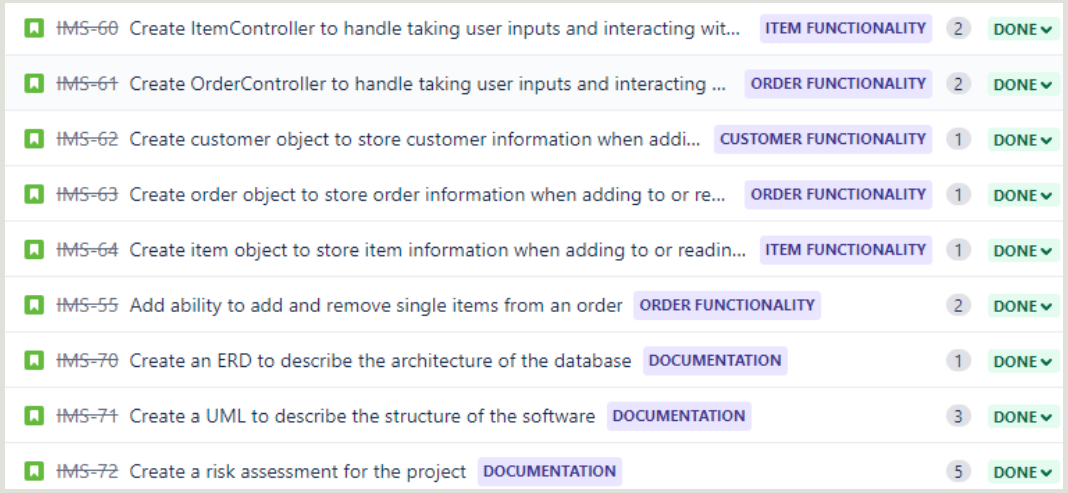
This is all the work that had to be done for the first sprint create my MVP.

I assigned Story Points based on how labour intensive the task would be.

First Sprint

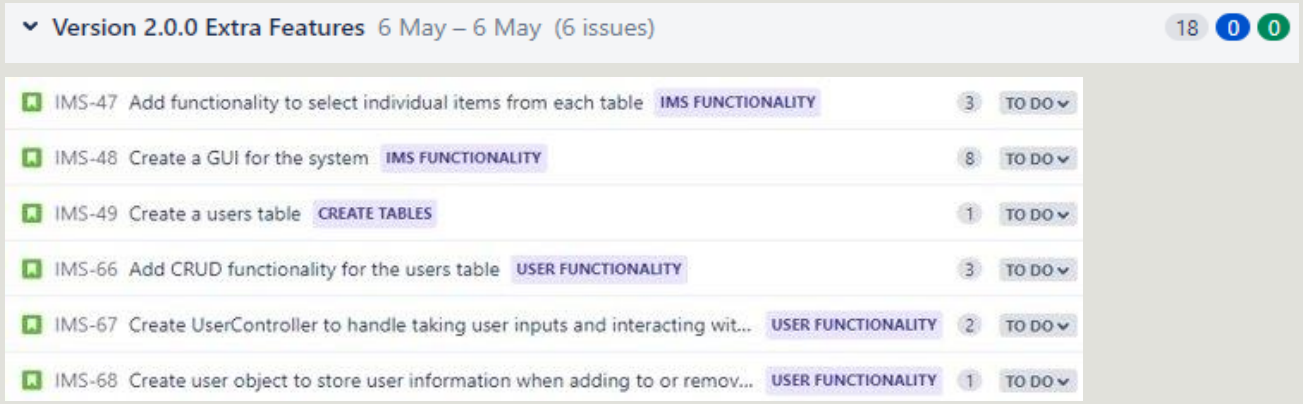
- My first sprint contained all the tasks that would contribute to creating the MVP.

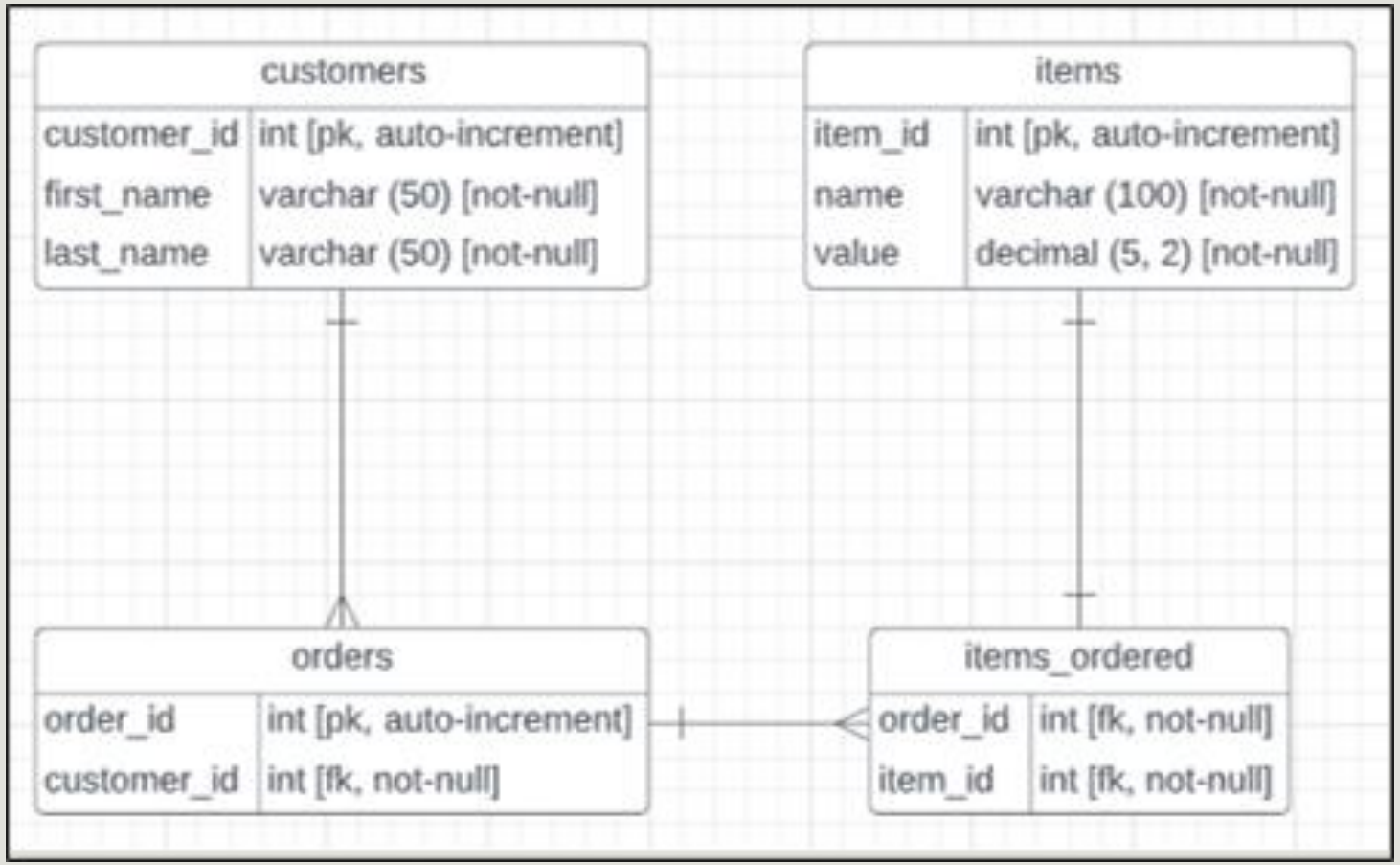




Second Sprint

My second sprint focused on tasks that would contribute more to the should and could sections of my MoSCow document.

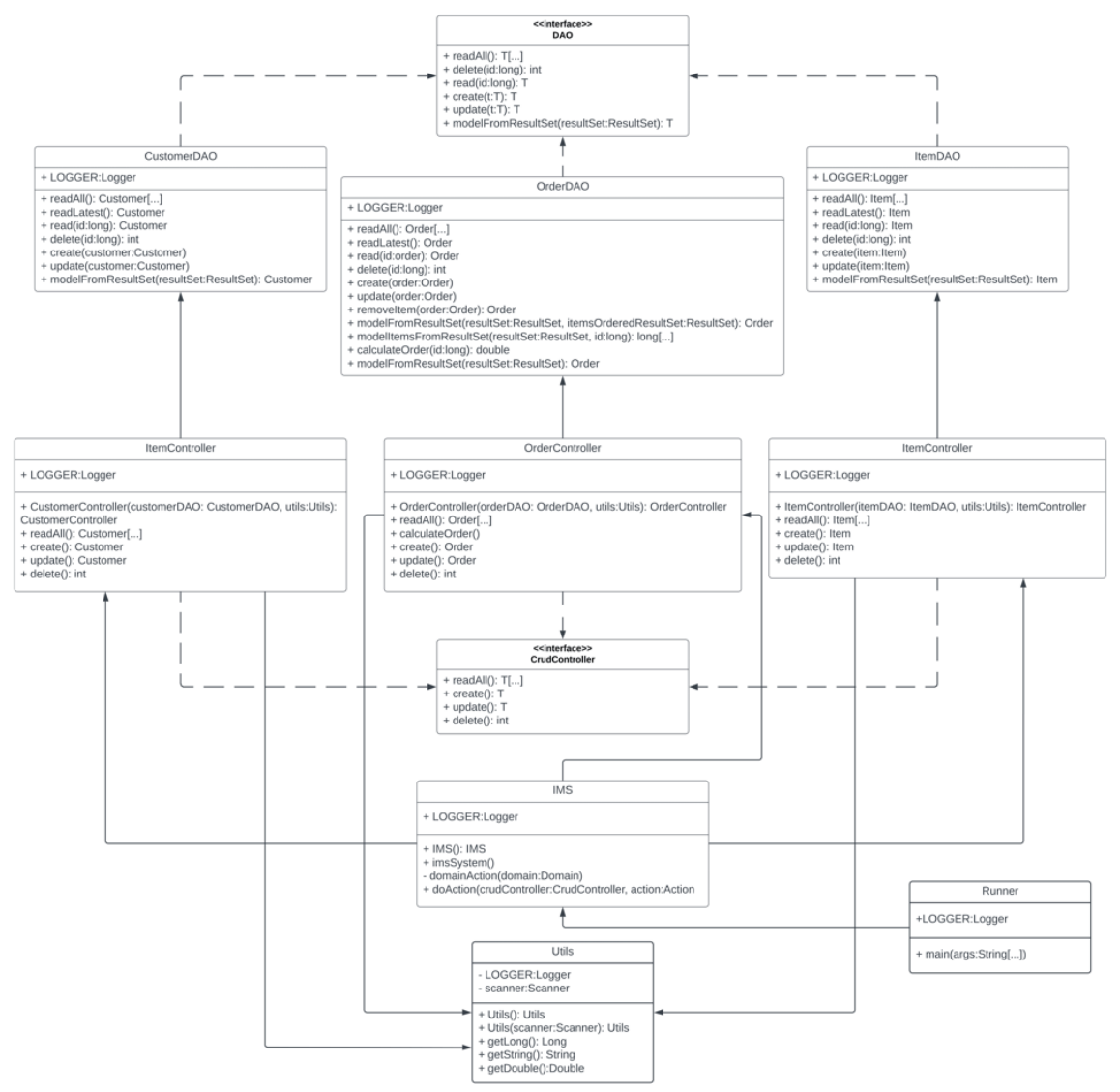




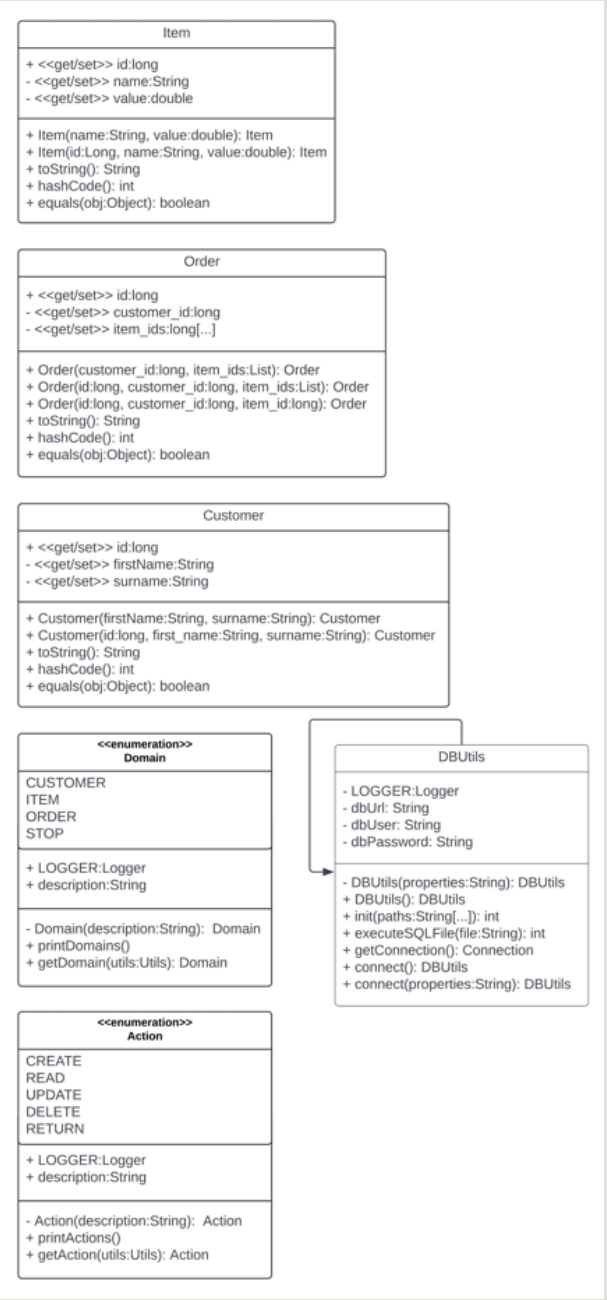
ERD

- My ERD helped my plan the structure of my database before I began editing the schema.
- The items_ordered table linked the items and orders table together to allow for
- To find items that link to an order, we can iterate through the items_ordered table and select all items where the order_id relates to the orders table.

UML Class Diagrams

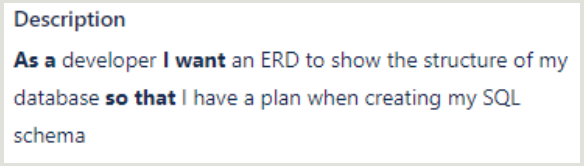


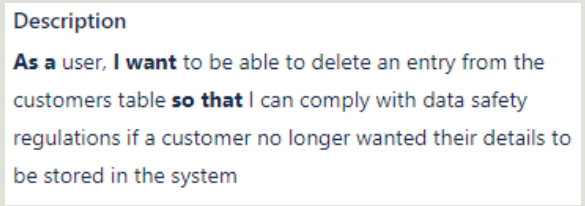
- My UML Diagrams helped me to plan the structure of my project before beginning my coding process.
- The diagram on the left shows the fata flow when the program is running
- The diagram on the right are enumerations, objects that are used for data storage, and the database connection class.



Stories

- Each of my Jira tasks had a story attached to it.





- I then associated a story point to the task that reflected how much effort each task should take to complete. This assisted when it came to managing my time.

GitHub

- My main branch remained untouched throughout the project until my software was running, passing all tests and packaging as expected.
- My dev branch only had a few commits made directly to it – I believe this is because I forgot to switch to my feaure1 branch when implementing version 0.0.1

Feature 2 focused on the order functionality epic.

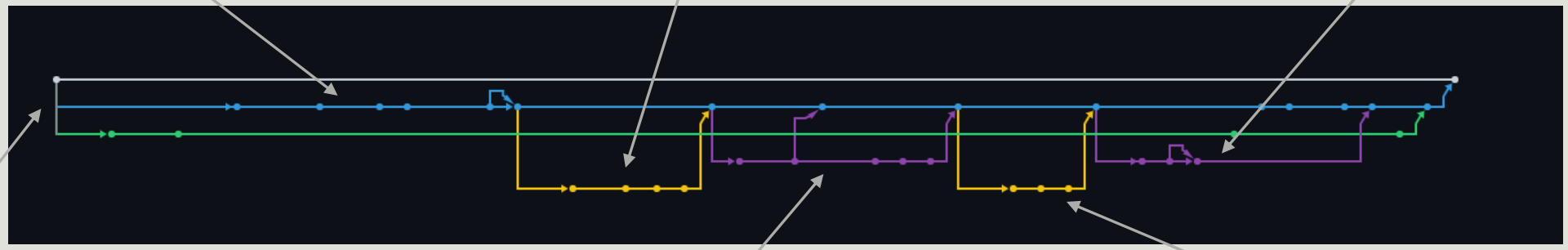
`Added functioning classes OrderController.java, OrderDAO.java and Order.java to feature1`

These commits were supposed to be included in the feature1 branch.

`Added functioning classes ItemController.java, ItemDAO.java and Item.java to feature1`

Finally, I created a BugFixes branch where I focused on fixing bugs that came to light in the testing phase

`Updated OrderDAO.java to fix some bugs that came to light during testing`



Firstly, I created my dev branch from master to isolate the code in master.

I then created a documentation branch where I could add my supporting documents without interfering with the rest of my work.

Feature 3 focused on the general IMS functionality to get the software running from user inputs

`Updated IMS.java to allow the CRUD methods to be called from user input`

I then created a testing branch where I wrote my tests

`Added testing classes for item classes`

CRUD

- My CRUD functionality fulfilled all of my user stories and MVP requirements, and my testing process reflected that.
- Each of the three main tables to operate on had full CRUD functionality and my `items_ordered` table only had the necessary steps of CRUD to allow the software to function as expected.

Test Coverage

- I managed to achieve 75.1% testing coverage on my code.

- A big reason for that was the catch statements in my code proved difficult to test, and time constraints proved a challenge when writing more in-depth tests

- Moving towards version 2.0.0, I would aim to find a way to test these catch statements and reach the goad of 80% coverage

- My DAO coverage was 79%, showing that the CRUD functionality is very capable

- And my CRUD Controller coverage was 89.9%, showing that these classes function as expected.