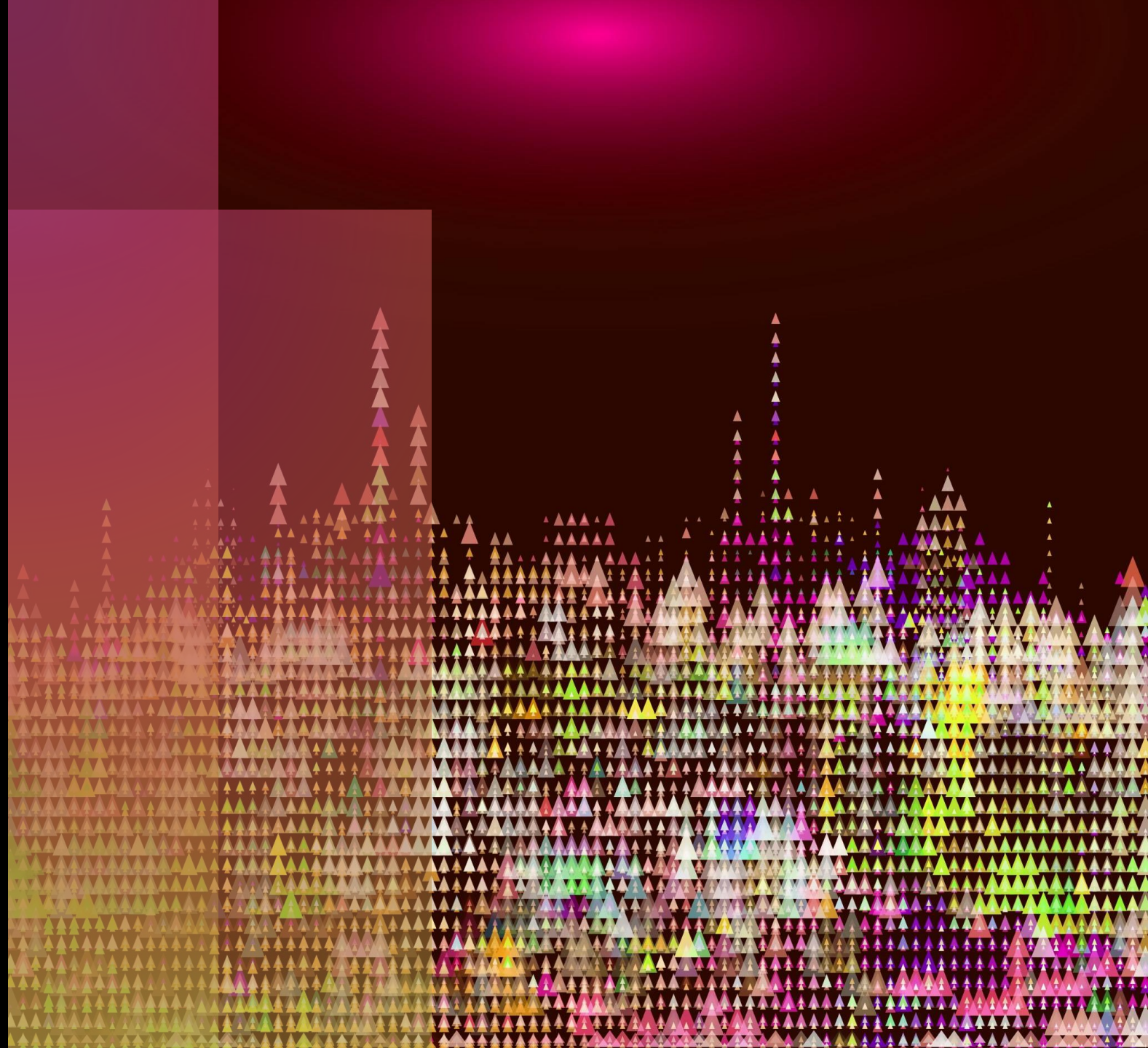
Inventory Management System

Sabiha Alam

22AprEnable1



MVP Objectives

- Build an application that an end user can interact with via a Command-Line Interface (CLI).
- CRUD functionality – needs to be able to Create, Read (view), Update and Delete data from customers, products and orders table.
- Code using a Version Control System **(Git)**.
- 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 containing the **customers, products, orders, and Order_Items** tables. Relationships should be modelled using an ERD.
- A functional application ‘back-end’.
- Unit tests for validation of the application. You should aim to reach the industry standard of **80%** test coverage.

Developer Journey

- **Version Control System**: Git
- **Source Code Management**: GitHub
- **Kanban Board**: Jira
- **Database Management System**: MySQL Server 5.7+ (local or GCP instance)
- **Back-End Programming Language**: Java
- **Build Tool**: Maven
- **Unit Testing**: JUnit

IMS Project RISK ASSESSMENT

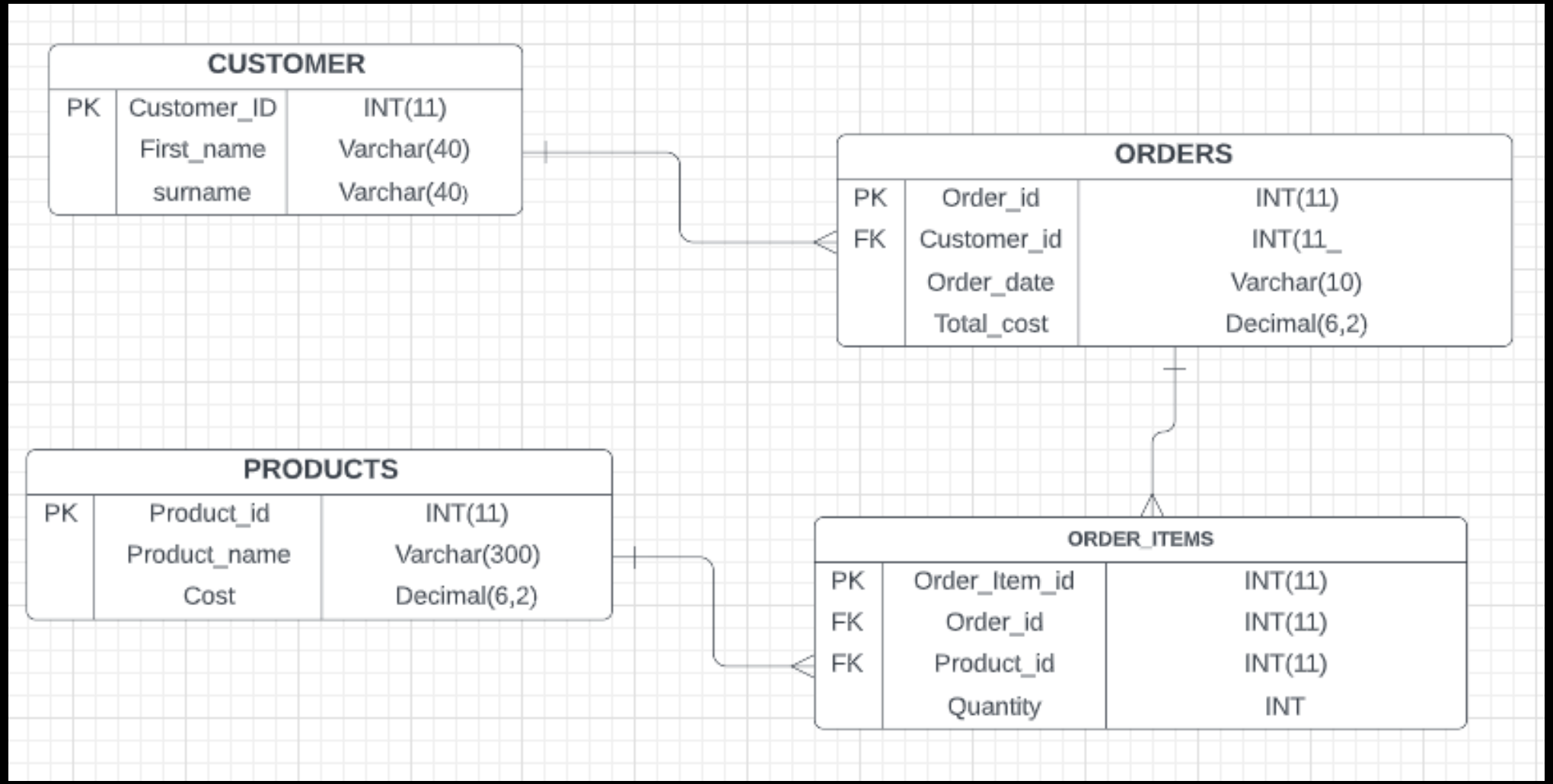
	Negligible	Minor	Moderate	Significant	Severe
Very unlikely	Low	Low	Low Medium	Medium	Medium
Unlikely	Low	Low Medium	Low Medium	Medium	Medium High
Moderate	Low	Low Medium	Medium	Medium High	Medium High
Likely	Low	Low Medium	Medium	Medium High	High
Very Likely	Low Medium	Medium	Medium High	High	High

Risk	Description	Response to avoid or fix	Likelihood	Impact	Risk Level
Device malfunction	If the device I'm working on stops working or faces any issues, this could mean I can't use it to continue my project.	Regularly run device check-ups to ensure it is working fine.	Very unlikely	Severe	Medium
Lose my work	I could lose all my work which would affect the completion of the project in time.	Regularly save and check on my work and create backup copies.	Very unlikely	Severe	Medium
Poor/loss of internet connection	Poor internet connection could slow down and hinder the progress of the project. Loss of internet connection would mean that GitHub can't be accessed, and I won't be able to contact my cohort on teams.	Ensure my internet provider is strong.	Unlikely	Severe	Medium high
GitHub servers being down	This would mean I am unable to access my remote work to complete my project.	Frequently check up on the GitHub server status to check it is running as normal and immediately inform my trainer if there are any issues.	Very unlikely	Severe	Medium
Health issues	Working for long periods of time in front of the screen can pose a range of health risks, such as back problem, wrist aches, eye strain, etc.	Schedule in regular breaks, that includes walking around and not looking at a screen. Also ensure I am well rested before starting work.	Moderate	Minor	Low medium
Project trainer not being available	Due to illness or being busy with helping others, which could limit the support I get on my work.	Inform the mentor that I need support and to get back to me when he is available. In the meantime, I can work on other work so that I am not wasting any time wating.	Moderate	Minor	Low medium
Being severely unwell or injured	I could fall ill or injure myself which could stop me from being able to finish the project in time.	Be careful when carrying out any injury prone activities such as exercise and take extra precautions to avoid falling ill.	Moderate	Significant	Medium high
Merge conflicts (Git)	Merge conflicts could occur when merging branches on git which could impact work being pushed onto the remote.	Frequently push work onto the remote and be mindful of which branches I am working on.	Moderate	Moderate	Medium

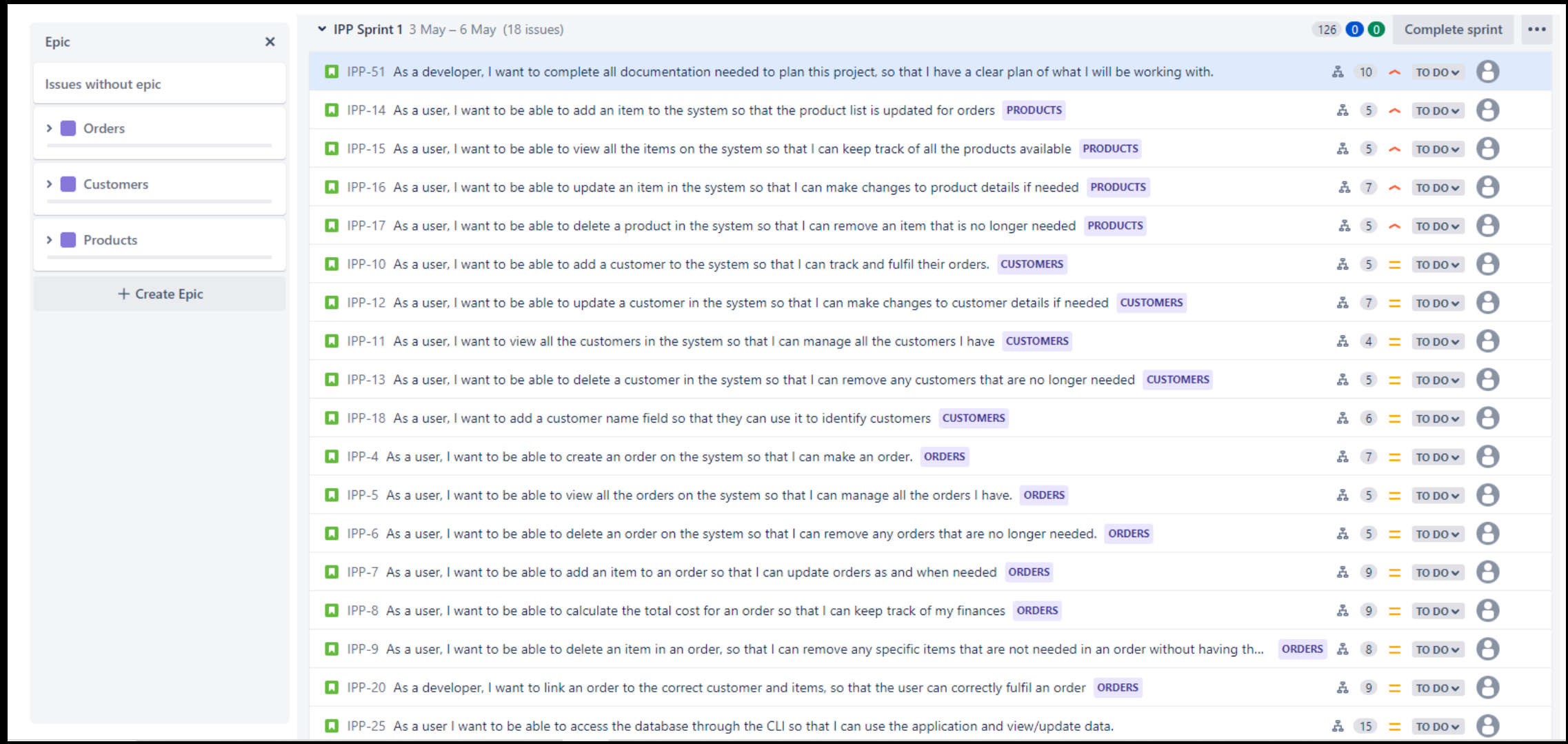
MoSCoW Method Document

Must Have	Should Have	Could Have	Won't Have
CRUD functionality	80% Test Coverage	User stories from the developer's point of view, rather than only the user to make tasks clearer.	Graphical User interface, this will all be executed through the command line.
Command Line Interface that the end user can interact with to make changes/view the data.	Many to Many Order_Items table to link Orders and Products table.		
The use of Git to push work and a Main/Master branch to save all final work.	The use of feature and dev branch on git to organise data and commits.		
A "products", "orders" and "customers" class.	A class for Order_Items		
The use of JDBC to connect the Eclipse to MySQL database.			
Documentation relating to the project. i.e., risk assessment, ERD model, Kanban board, etc.			

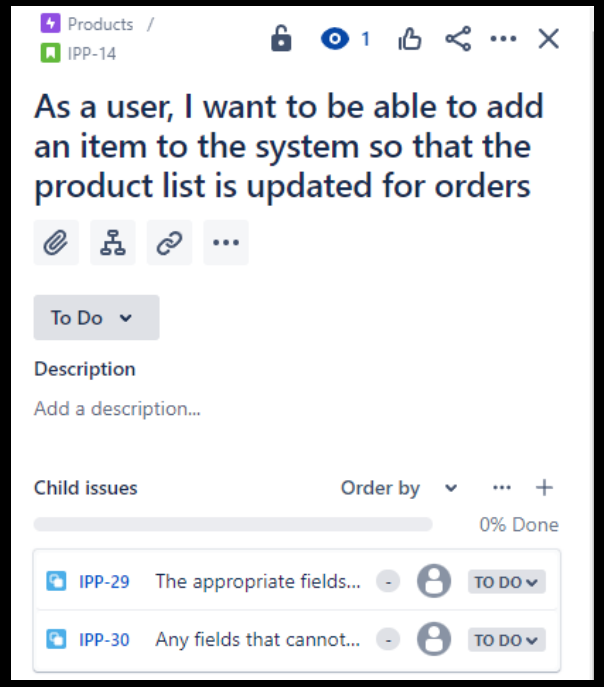
ERD MODEL

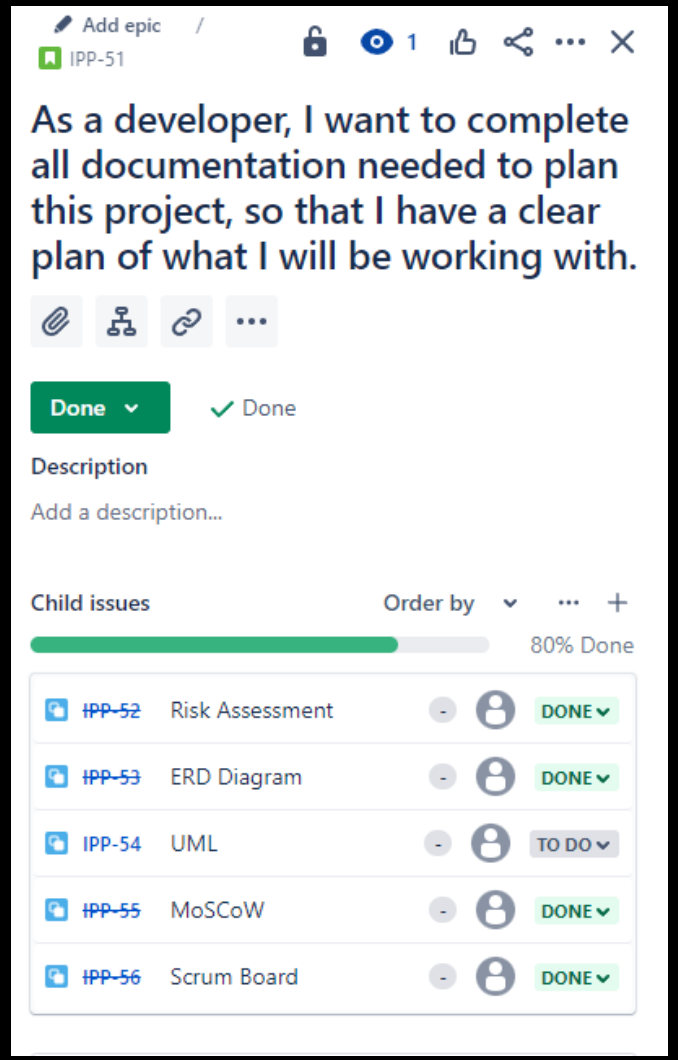

CUSTOMER
PK
Customer_ID
INT(11)
First_name
Varchar(40)
surname
Varchar(40)
ORDERS
PK
Order_id
INT(11)
FK
Customer_id
INT(11_
Order_date
Varchar(10)
Total_cost
Decimal(6,2)
PRODUCTS
PK
Product_id
INT(11)
Product_name
Varchar(300)
Cost
Decimal(6,2)
ORDER_ITEMS
PK
Order_Item_id
INT(11)
FK
Order_id
INT(11)
FK
Product_id
INT(11)
Quantity
INT

Sprint


Epic
Issues without epic
Orders
Customers
Products
Create Epic
IPP Sprint 1 3 May – 6 May (18 issues)
126 0 0
Complete sprint
IPP-51 As a developer, I want to complete all documentation needed to plan this project, so that I have a clear plan of what I will be working with. 10 TO DO
IPP-14 As a user, I want to be able to add an item to the system so that the product list is updated for orders PRODUCTS 5 TO DO
IPP-15 As a user, I want to be able to view all the items on the system so that I can keep track of all the products available PRODUCTS 5 TO DO
IPP-16 As a user, I want to be able to update an item in the system so that I can make changes to product details if needed PRODUCTS 7 TO DO
IPP-17 As a user, I want to be able to delete a product in the system so that I can remove an item that is no longer needed PRODUCTS 5 TO DO
IPP-10 As a user, I want to be able to add a customer to the system so that I can track and fulfil their orders. CUSTOMERS 5 TO DO
IPP-12 As a user, I want to be able to update a customer in the system so that I can make changes to customer details if needed CUSTOMERS 7 TO DO
IPP-11 As a user, I want to view all the customers in the system so that I can manage all the customers I have CUSTOMERS 4 TO DO
IPP-13 As a user, I want to be able to delete a customer in the system so that I can remove any customers that are no longer needed CUSTOMERS 5 TO DO
IPP-18 As a user, I want to add a customer name field so that they can use it to identify customers CUSTOMERS 6 TO DO
IPP-4 As a user, I want to be able to create an order on the system so that I can make an order. ORDERS 7 TO DO
IPP-5 As a user, I want to be able to view all the orders on the system so that I can manage all the orders I have. ORDERS 5 TO DO
IPP-6 As a user, I want to be able to delete an order on the system so that I can remove any orders that are no longer needed. ORDERS 5 TO DO
IPP-7 As a user, I want to be able to add an item to an order so that I can update orders as and when needed ORDERS 9 TO DO
IPP-8 As a user, I want to be able to calculate the total cost for an order so that I can keep track of my finances ORDERS 9 TO DO
IPP-9 As a user, I want to be able to delete an item in an order, so that I can remove any specific items that are not needed in an order without having th... ORDERS 8 TO DO
IPP-20 As a developer, I want to link an order to the correct customer and items, so that the user can correctly fulfil an order ORDERS 9 TO DO
IPP-25 As a user I want to be able to access the database through the CLI so that I can use the application and view/update data. 15 TO DO

Tasks (Acceptance Criteria)





GitHub

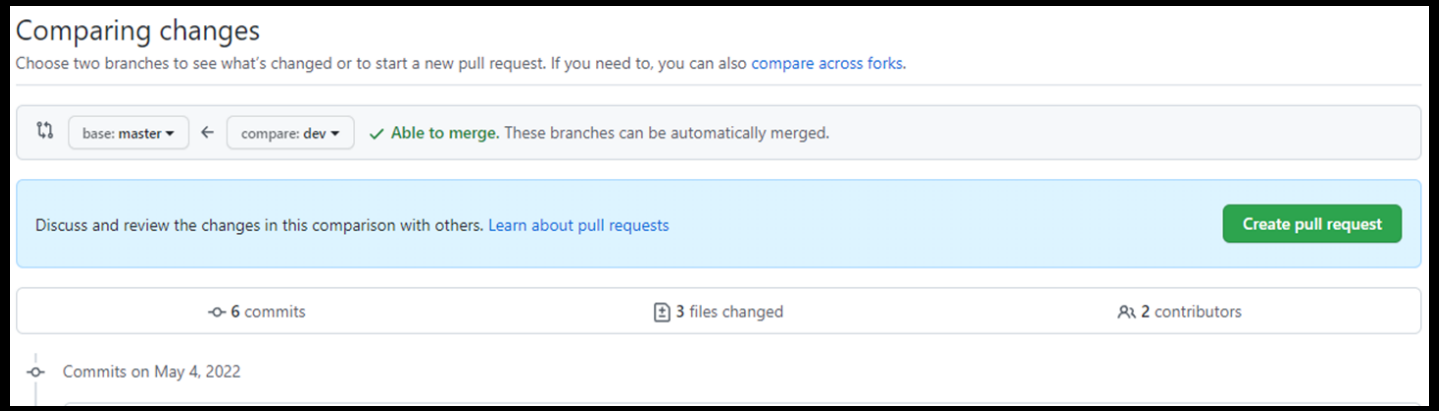
Multiple feature branches

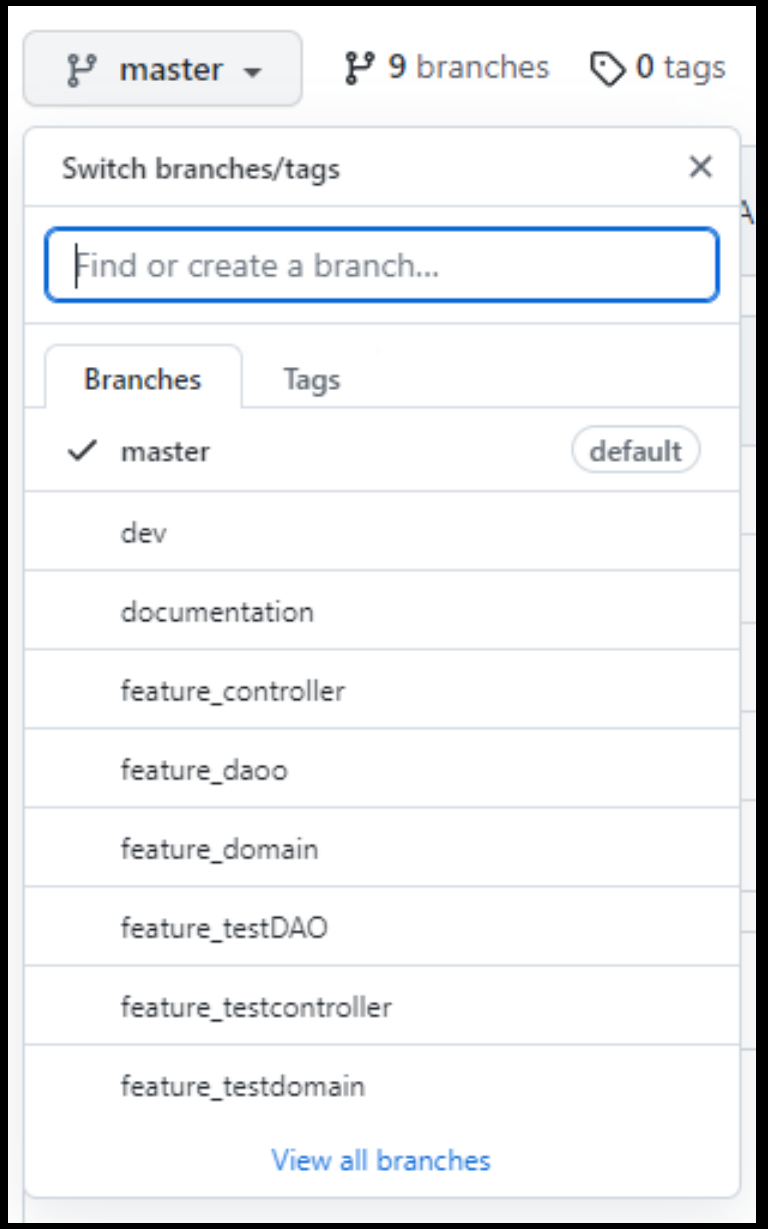
↓

Dev branch

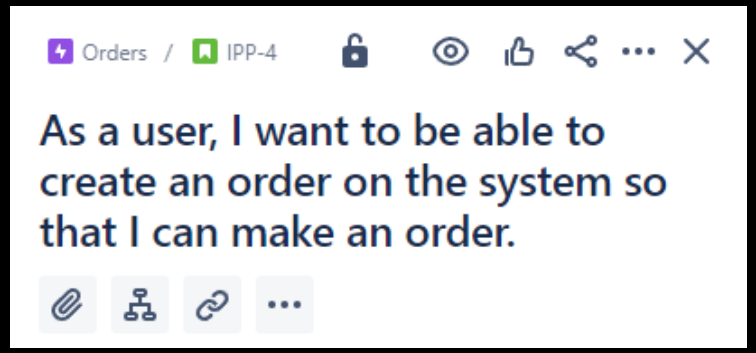
↓

Master





Version Control



Git checkout –b dev

Git checkout –b feature_controller

```
/**
 * Create an order by taking in user input
 *
 */
@Override
public Orders create() {
    LOGGER.info("Please enter order id");
    Long order_id = utils.getLong();
    LOGGER.info("Please enter customer_id ");
    Long customer_id = utils.getLong();
    LOGGER.info("Please enter date of order");
    String order_date = utils.getString();
    LOGGER.info("Please enter the total cost");
    Double totalcost = utils.getDouble();
    Orders orders = orderDAO.create(new Orders(order_id, customer_id, order_date, totalcost));
    LOGGER.info("Order created");
    return orders;
}
```

Git commit –m “commit feature_controller”

Push that onto GitHub

Test Coverage

Element	Coverage	Covered Instructio...	Missed Instructions	Total Instructions
22AprEnable1_IMS	53.1 %	1,797	1,589	3,386

Unit Testing

Unfortunately, my tests did not work. ☹

However, I did attempt to incorporate Mockito for my testing as can be seen in the image.

```
@Test
public void testCreate() {
    final String Product_name = "Foundation";
    final Double Product_cost = 28.00;
    final Products created = new Products(Product_name, Product_cost);

    Mockito.when(utils.getString()).thenReturn(Product_name);
    Mockito.when(utils.getDouble()).thenReturn(Product_cost);
    Mockito.when(dao.create(created)).thenReturn(created);

    assertEquals(created, controller.create());

    Mockito.verify(utils, Mockito.times(1)).getString();
    Mockito.verify(utils, Mockito.times(1)).getDouble();
    Mockito.verify(dao, Mockito.times(1)).create(created);
}
```

LIVE DEMO

Performance Review

- As seen in my demo, whilst I attempted to accomplish all the tasks in my sprint, I was not successful as my tests failed and I was unable to correctly connect my Java to the SQL database.
- Time constraints
- Feeling unwell
- Not being able to ask for support when needed

- Plans for Next Projects...

Thank You ☺

Any Questions?