Software Core Fundamental Project

Planning

First I started to plan out this project by using a Scrum Jira project: [IP1 board - Agile board - Jira (atlassian.net)](https://atlassian.net)

I started with creating 3 epics:

1. As a developer I want to create a MySQL database to store my application data.
2. As a developer I want to create the source code for the inventory management system using JAVA.
3. As a developer I want to TEST my application for any bugs and ensure the functionality is working as expected.

I created a sprint and started with the epic 1 to create the MySQL database as the application will be built around this.

Risk Assessment

Risk	Evaluation	Likelihood	Impact Level	Responsibility	Response	Control Measures
Not being able to understand methods to accomplish tasks	Not being able to complete project before deadline	Medium	High	Developer	Ask mentor or colleagues for advice	Complete more self studying videos out of hours to understand concepts and ask Google
Broken code pushed into main repository	Final code may have bugs and not work as expected	Medium	High	Developers	Rollback to most recent working state	Carry out testing on code before pushing to main branch.
Laptop damage	Unable to carry on working with resources I am used to	Low	High	Developer	Ensure all code has been pushed to git after making changes every time so in the case of the laptop breaking I will still have the code I have worked on and will have to find a back up laptop and download all softwares again which will take a lot of time	Make sure if moving work areas to have extra caution when handling my laptop
Internet outage	I will not be able to push work into github or use resources for guidance	Medium	High	Internet provider	Will have to use hotspot data from my phone	Ensure to have a backup (hotspot) in case of downtime.
Illness	Unable to continue working or working at normal pace	Low	Medium		Will have to recover fully before returning to work or work at a slower pace	Ensure to stay active as much as possible while working and eating healthy.

As it is only myself working on the project I did not have any blockers from other persons or teams.

Revisiting risk assessment: Risk 1: 'Not being able to understand methods to accomplish tasks' has impacted this project. I have spent hours trying to catch up on understanding concepts and figuring out

MySQL information gathering and entity relationship diagram

First, I needed to think about the information required for the database tables and fields then normalize it. This then allowed me to create the entity relationship diagram:

Created here: IMS Database: Lucidchart

I then created the database within MySQL using this ERD.

Later on while working on the code I realised many of the fields were excessive so i then simplified it.
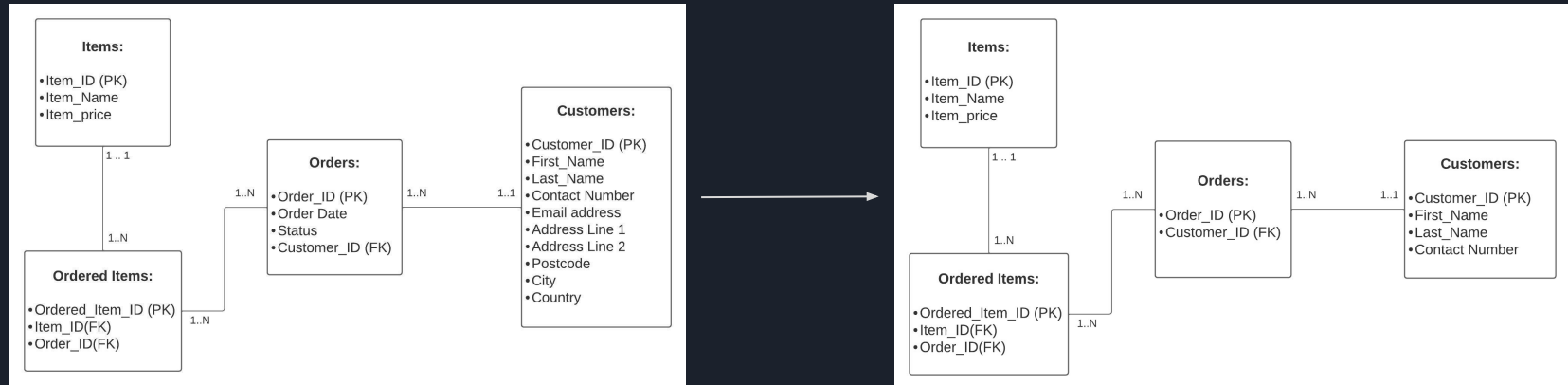
JAVA Customer

1. First i updated the db.properties to have the correct credentials to connect to my DB
2. Updated the customer CREATE method first. I had to ensure all fields from the database was included in the Customer class and that the getters/setters and constructors were also updated with these variables. Had issues with returning the new data as it came back with Null, this was resolved by changing from `this.setFirstName(firstName);` to `this.firstName = firstName;` in Customer.java line 25 and so on.
3. I then created the methods to READ, UPDATE and DELETE which worked successfully.

```
Welcome to the Inventory Management System!
Which entity would you like to use?
CUSTOMER: Information about customers
ITEM: Individual Items
ORDER: Purchases of items
STOP: To close the application
Customer
What would you like to do with customer:
CREATE: To save a new entity into the database
READ: To read an entity from the database
UPDATE: To change an entity already in the database
DELETE: To remove an entity from the database
RETURN: To return to domain selection
read
Customer ID 1, First Name: JT, Last Name: Lana, Contact number: 0738495728]
Customer ID 4, First Name: HArry, Last Name: Pott, Contact number: 123456789]
Customer ID 5, First Name: Bill, Last Name: Paper, Contact number: 3848485938539]
Customer ID 6, First Name: Sally, Last Name: Up, Contact number: 333444555]
Customer ID 8, First Name: JT, Last Name: Lana, Contact number: 987654321]
```

JAVA Item

I created similar classes to the Customer class:
- Item class
- Item DAO
- Item Controller

Creating the methods to CREATE, READ, UPDATE and DELETE was the same process as creating the Customer methods.

I then proceeded to create a number of Items in my database.

Revisiting this I had an issue with the READ method as the Item number was returning the same for all products.

```
item
What would you like to do with item:
CREATE: To save a new entity into the database
READ: To read an entity from the database
UPDATE: To change an entity already in the databas
DELETE: To remove an entity from the database
RETURN: To return to domain selection
read
Item ID: 6, Item: Apple Crumble, Price: £4.99.
Item ID: 6, Item: Choc Muffin, Price: £3.0.
Item ID: 6, Item: Jam Donut, Price: £2.66.
Item ID: 6, Item: Jam Donut, Price: £2.99.
```

This occurred because the itemID within the Item class was using a static variable. This was removed and the Item IDs now return as expected.

JAVA Orders

As modelled in my ERD I had an Order table and an Ordered Items table. In Java I created an Orders class, OrderController and OrderDAO class. Within my Orders Class I added a Customer variable to hold the customer details and also a List variable which was to store a list of items added to the order.

Within the DAO is where I planned to use the JOIN command to combine the table fields to have them listed in the results.

This is where I faced difficulties. After a number of failed attempts I decided to start again as my code had become very hard to read and I also simplified my variables.

Revisited: The Read function is working and showing an order with the customer details along with the item details.

Testing

Even though my systems functionality was not as I had expected, I still performed tests on the methods that are working. These are:

- CustomerTest - Successful
- ItemTest - Successful
- OrderTest - Successful
- CustomerControllerTest - Successful
- ItemControllerTest - Successful
- CustomerDAO - Failed (Returned NULL instead of Customer data in READLATEST method)

The testing coverage is currently at 46.9%. Due to spending the majority of time on the source code:

ava (21 Feb 2022 09:21:51)				
Element	Coverage	vered Instructions	lissed Instructions	Total Instructions
> 📁 ims-project	46.9 %	1,339	1,516	2,855