QA Academy
Week 5 IMS Project
John Fisher

Project Presentation and Overview

Introduction

Technologies Used:

- Jira – Project Planning
- Git/GitHub – Continuous Integration
- Eclipse IDE – To develop IMS programme
- Maven – for Database integration and utilities/plug-ins
- mySQL – Database creation
- Junit and Mockito – Testing (Unit and Integration)
- Draw.io – ERD and UML Creation

Jira – Project Planning

- Jira was essential to plan out the project and keep track of progress.
- 4 Epics were created: 1. Create and IMS Schema in mySQL; 2. Create a Java Project to manipulate the SQL IMS and integrate it Accordingly; 3. Test the functionality and integration of the programme; 4. Utilise a Version Control System to upload the project and include necessary documentation.
- The ability to add and change the status of child issues was invaluable in tracking progress and checking off work as it is completed.
- In future, I will certainly reconsider the priority of some task, e.g. Building the Java Programme – as this took much more time than originally thought, and was late.

Version Control

- Git and GitHub were used for version control.
- In hindsight, this could have been utilised more effectively by creating multiple branches to push smaller segments of work to the dev branch – currently, the bulk of the coding for the programme was submitted in 1 push from featureData(branch), with multiple commits.
- The Tests, Documentation, and SQL database were pushed in their own branches.

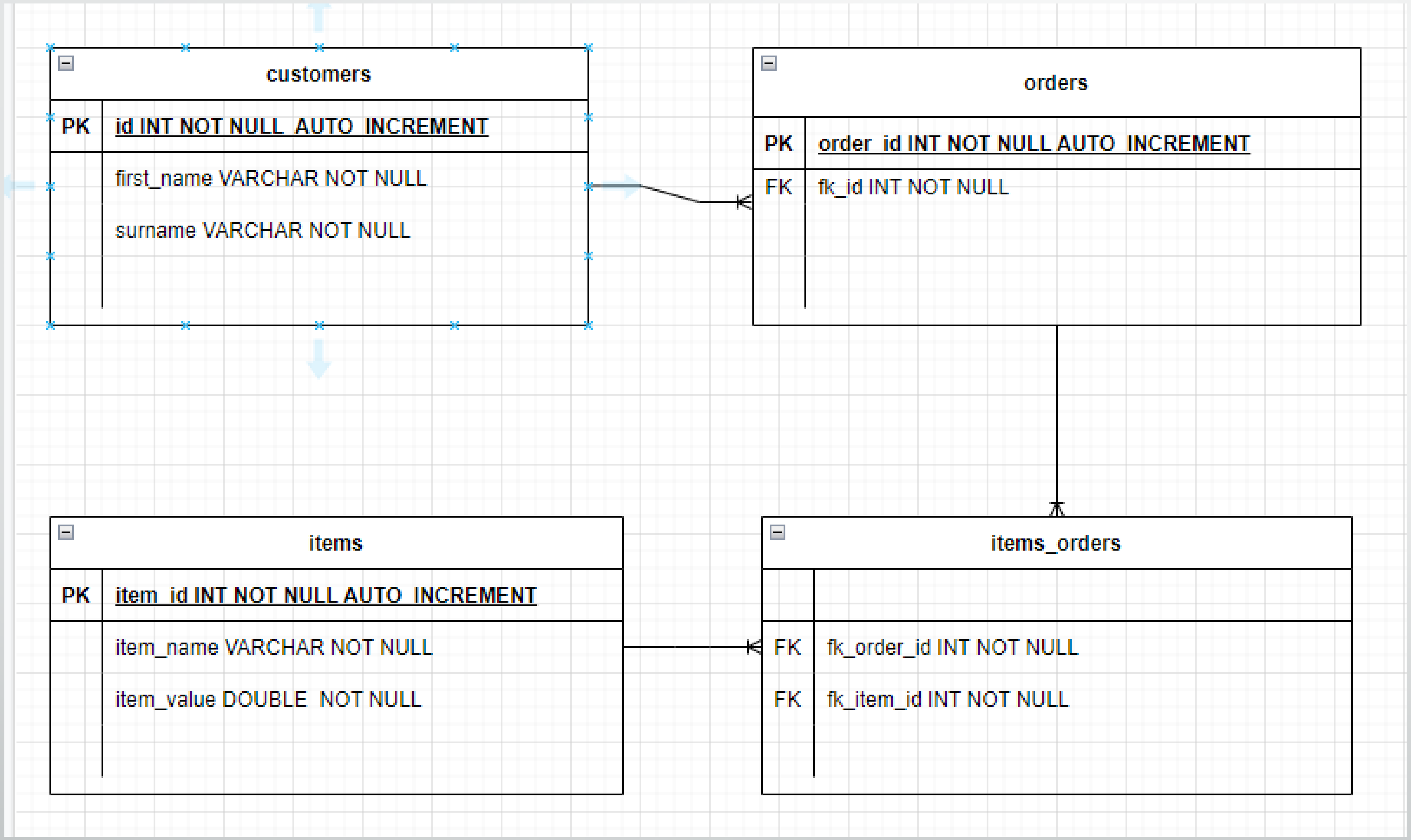
Testing

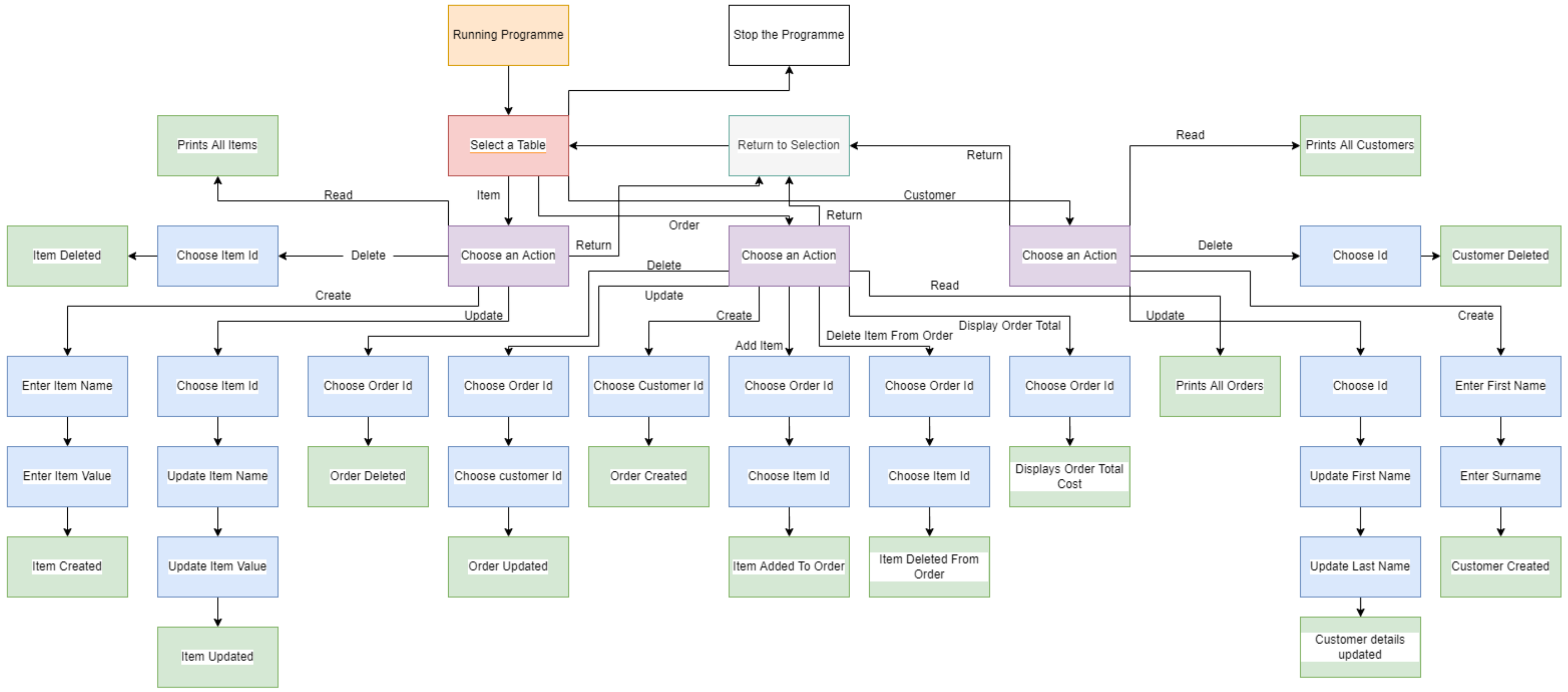
- Both Unit Testing and Integration Testing were utilised when creating this project.
- JUnit allowed for simple unit testing, checking individual methods to ensure they function as intended.
- Using Mockito allowed for a more in-depth test of the programme's functionality, using dummy data to simulate the IMS running without having to call the actual methods in the programme.
- In retrospect, I would ideally have liked to spend more time on testing as I was unable to attain the Coverage I was ideally seeking, and a number of Tests do not function as they should. With more time and further practice with Testing, I believe I could greatly increase the coverage and effectiveness of the tests.

Demonstration

- We will now quickly demonstrate:
- The IMS's functionality;
- Go over some user stories;
- Look at some accompanying documentation and diagrams.

	Risk Description	Evaluation	Likelihood (1 to 10)	Impact Level/ Importance (1 to 10)	Responsibility	Response	Control Measures
1.	IDE / mySQL Crashes	Code/Data is potentially lost	2	8	IDE / mySQL / Hardware	Restore to most recent version	Save work frequently and back-up regularly
2.	Power Outage	Loss of code/potential for hardware failure	1	10	Energy Provider / Act of God	Restore from backup/cloud if possible – replace hardware	Back-up work on cloud/separate hardware; use surge protectors
3.	Failure to meet Project requirements	The project doesn't meet the MVP when deployed	4	9	Developer	Focus on the basic client requirements and reread the Project Spec	Read and fully understand the Project Spec before beginning the project; Prioritise essential tasks during planning
4.	Incorrect Version deployed to main branch	Deployed project on main branch is unfinished/does not work	6	5	Developer	Roll back to last workable version on Git	Restrict access to main branch & ensure version works prior to merging to dev/main branches
5.	Bugs and errors prevent the project functioning	The application does not work correctly due to bugs that have been resolved	9	3	Developer	Test the application using Junit and Mockito to resolve any issues	Plan for testing prior to starting the project and ensure these are carried out and any bugs resolved before deployment
6.	Project not completed within set timeframe	The project is not ready to be deployed by the set deadline	5	7	Developer	Prioritise tasks requiring completion to limit any further delays	Plan ahead using Jira to prioritise essential work and manage your time effectively
7.	The project cannot be easily/effectively used after deployment	The finished project cannot be used/implemented easily by users or developers	7	6	Developer	Create an in-depth README.md and UML to explain implementation and functionality	During planning, ensure you include time to write a comprehensive README file and UML


customers
PK
id INT NOT NULL AUTO INCREMENT
first_name VARCHAR NOT NULL
surname VARCHAR NOT NULL
orders
PK
order_id INT NOT NULL AUTO INCREMENT
FK
fk_id INT NOT NULL
items
PK
item_id INT NOT NULL AUTO INCREMENT
item_name VARCHAR NOT NULL
item_value DOUBLE NOT NULL
items_orders
FK
fk_order_id INT NOT NULL
FK
fk_item_id INT NOT NULL



Running Programme
Stop the Programme
Select a Table
Return to Selection
Prints All Items
Prints All Customers
Read
Return
Item
Customer
Order
Return
Item Deleted
Choose Item Id
Delete
Choose an Action
Return
Choose an Action
Choose an Action
Delete
Choose Id
Customer Deleted
Delete
Read
Create
Update
Update
Create
Update
Create
Add Item
Delete Item From Order
Display Order Total
Update
Enter Item Name
Choose Item Id
Choose Order Id
Choose Order Id
Choose Customer Id
Choose Order Id
Choose Order Id
Choose Order Id
Prints All Orders
Choose Id
Enter First Name
Enter Item Value
Update Item Name
Order Deleted
Choose customer Id
Order Created
Choose Item Id
Choose Item Id
Displays Order Total Cost
Update First Name
Enter Surname
Item Created
Update Item Value
Order Updated
Item Added To Order
Item Deleted From Order
Update Last Name
Customer Created
Item Updated
Customer details updated

Retrospective

- Sprint reviews: The initial planning phase went relatively smoothly – all initial documentation (ERD, Risk Assessment, etc.) was completed on time.
- I underestimated the time/effort required for coding the IMS in Java, and this sprint was completed late. In future, I will adjust the Sprint due date accordingly to take this into account.
- While completed on time, the Sprint for Testing ideally would have been longer as I did not meet the level of coverage I was hoping for. Additional practice and work with Testing would also prove greatly advantageous, and I will consider this moving forward.
- Pushing everything to dev from feature branches was effective and helped to maintain a working version at all times. It was simple to add relevant documentation to the repo as required.

Thanks for Listening!

- This project has been a lot of fun, however extremely challenging! I look forward to future projects to practice and develop the skills I have already learned.