
INTRODUCTION

WHO ARE YOU? HOW DID YOU APPROACH THE SPECIFICATION?

HI MY NAME IS ANGEL AND I'M A TRAINEE IN THE QA ACADEMY.

I APPROACHED THE SPECIFICATION BY FIRST READING AND UNDERSTANDING WHAT THE EXACTLY THE GOAL WAS AND THUS WHAT NEEDED TO BE DONE.

I THEN THOUGHT ABOUT WHAT TOOLS I HAD AT MY DISPOSAL TO TACKLE THE RELEVANT ISSUES.

IN THIS PRESENTATION I WILL EXPLAIN WHAT TOOLS I USED TO AID ME.

CONSULTANT JOURNEY

WHAT TECHNOLOGIES HAVE YOU LEARNED FOR THIS PROJECT?

I LEARNED HOW TO USE **JAVA** CODE AND MANY RELEVANT TECHNIQUES WITHIN TO BUILD ON THE APPLICATION. SOME THINGS THAT WERE IMPLEMENTED IN JAVA WERE THE USE OF:

- CLASSES
- FIELDS
- CONSTRUCTORS
- METHODS/OBJECTS
- ENCAPSULATION

THE JAVA BUILD TOO **MAVEN** WAS ALSO IMPLEMENTED. THIS WAS USED TO EXPAND JAVA FUNCTIONALITY EVEN FURTHER. USING MAVEN, I WAS ABLE TO CONNECT SQL DATABASES.

CONSULTANT JOURNEY

SQL WAS ALSO IMPLEMENTED THROUGH THE USE OF **MYSQL WORKBENCH**.

OTHER TECHNOLOGIES TO AID IN THE PROCESS WERE ALSO USED. SUCH AS

- **GIT** – AS A VERSION CONTROL SYSTEM, AND **GITHUB** – FOR SOURCE CODE MANAGEMENT.
- **JIRA** – AS THE KANBAN BOARD IN ORDER TO TRACK PROJECT PROGRESS.
- **JUNIT** – FOR UNIT TEST.

WE ARE GOING TO TOUCH ON THESE SUBJECTS IN THIS PRESENTATION

CI

IN ORDER TO ENSURE I WAS FOLLOWING CONTINUOUS INTEGRATION I UTILIZED GIT AND GITHUB FOR VERSION CONTROL.

HERE I UTILISED THE FEATURE-BRANCH MODEL. THE MODEL WAS:

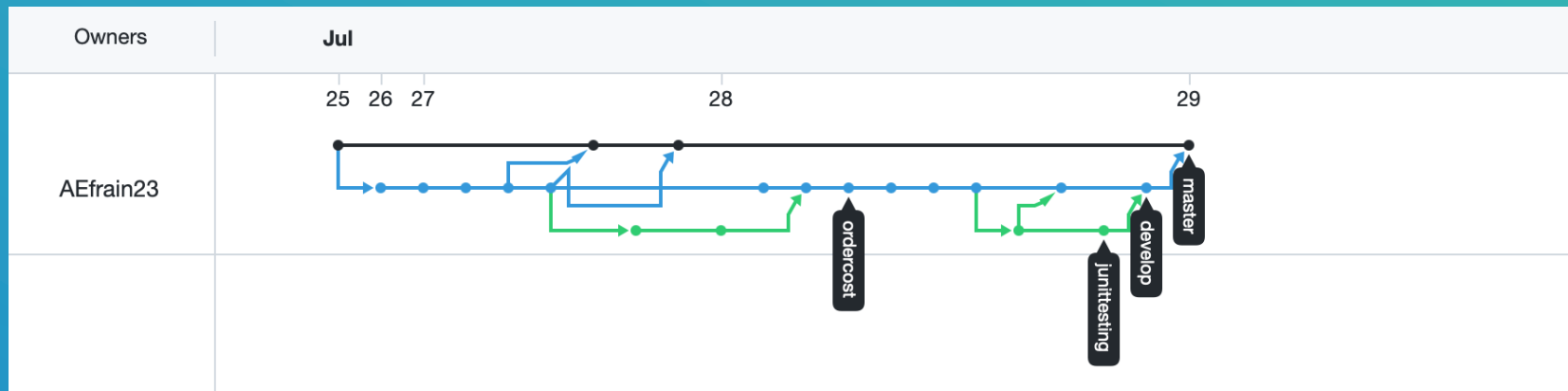
- **MASTER / DEVELOP / FEATURES**

I MADE FEATURE BRANCHES IN ORDER TEST INITIAL ALTERATIONS OF THE CODE. WHEN I CONFIRMED THEY WERE IN WORKING ORDER I MERGED THEM TO MY DEVELOP BRANCH.

ONCE I CONFIRMED EVERYTHING WAS WORKING ON MY DEVELOP BRANCH, I THEN FINALLY MERGED ONTO MY MASTER BRANCH. THIS ALLOWED ME TO ENSURE NO MISTAKES/ERRORS WERE CARRIED ON TO THE MAIN BRANCH OF MY CODE.

CI

HERE I SHOWCASE THE NETWORK GRAPH OF THE DIFFERENT BRANCHES AND COMMITS THAT WERE MADE THROUGHOUT THE PROJECT.



TESTING

WHAT WAS TESTED?

FOR TESTING THE DIFFERENT SECTION OF MY CODE, I IMPLEMENTED JUNIT. I RAN TESTS ON THE FOLLOWING:


- CUSTOMER DETAIL FUNCTIONALITY
- ITEM DETAIL FUNCTIONALITY
- PRODUCT DETAIL FUNCTIONALITY

I.E., FOR EACH OF THESE I ENSURED CRUD FUNCTIONALITY. ***CREATE, READ, UPDATE AND DELETE.***

TESTING

THE COVERAGE OF THE SC/MAIN/JAVA FOLDER.

FOR THE WORK CARRIED OUT AND THE CODE WRITTEN A COVERAGE OF 76.5% WAS REACHED. THIS CAN BE SEEN BELOW:

IMS-Starter (28 Jul 2022 21:22:01)				
Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
> IMS-Starter	 76.5 %	2,396	736	3,132

DEMONSTRATING

SOME OF THE USER STORIES THAT WERE IMPLEMENTED AS PART AGILE, AND THE USE OF JIRA AS A KANBAN BOARD ARE SEEN ON THE NEXT SLIDES.

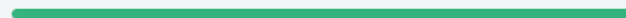
I WILL ALSO SHOWCASE THE EPICS THAT WERE CREATED IN ORDER TO ENCOMPASS ALL THE USER STORIES THAT WERE WRITTEN UP.

Issues without epic

>  Customers



>  Items



>  Orders




>  Testing





>  Databases





As a user I want to view all item details found in the system so I can see all the items that are already part of the system.

 Attach

 Add a child issue

 Link issue





Description

Given: [precondition] User is signed in.

When: [user action] User navigates to section containing all item details.













Then: [outcome] All item details are displayed to user.

Priority  Highest

Child issues

Order by  ... 

100% Done

 IMSS-45	User must be able to view item id in system			DONE 
 IMSS-46	User must be able to view item name in system			DONE 
 IMSS-47	User must be able to view item cost in the system			DONE 

As the developer I want to ensure that item details functionality is tested so that I can ensure it is working

Attach

Add a child issue

Link issue

▼

...

Description

Normal text ▼ | B | I | ... | A ▼ | ☰ ☷ | 🔗 🖼️ @ 😊 🏠 <> ⓘ + ▼

Given: [precondition] Developer is logged into the Java via the use of the selected IDE.

When: [user action] Developer writes up code for unit test in relevant classes etc. Developer navigates to section containing relevant unit tests and Developer runs tests on each section.

Then: [outcome] Unit tests confirm that code for the MIS system is working.

Save









Cancel

Priority ⬆️ High

Child issues

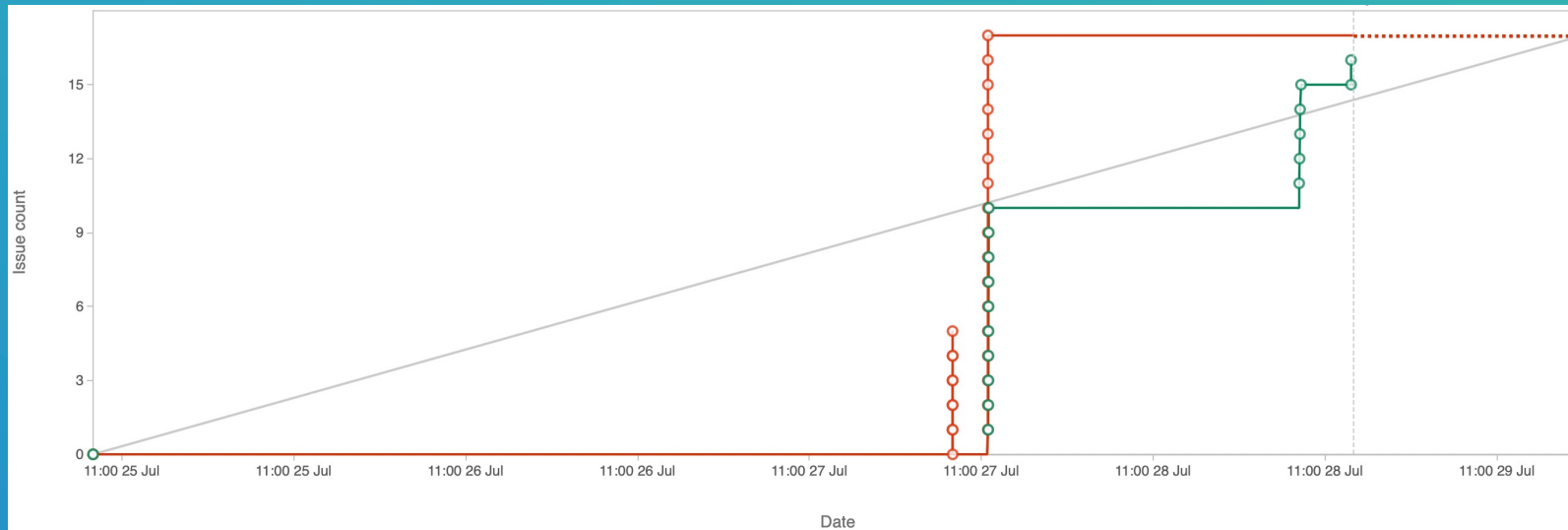
Order by ▼ ... +



 IMSS-64	Developer must test functionality that allows new items to be added	-		DONE ▼
 IMSS-65	Developer must test functionality that allows existing item details to be edited	-		IN PROGRESS ▼
 IMSS-66	Developer must test functionality that allows all existing item details to be viewed	-		DONE ▼
 IMSS-67	Developer must test functionality that allows existing items to be deleted	-		DONE ▼

SPRINT REVIEW

THE FOLLOWING SHOWS THE SPRINT RUN OF THE PROJECT IN RELATION TO USER STORY ISSUES.



SPRINT RETROSPECTIVE

WHAT WENT WELL?

CRUD FUNCTIONING ON ALL SECTIONS OF THE SYSTEM (CUSTOMER, ITEM AND ORDER).

WHAT COULD BE IMPROVED?

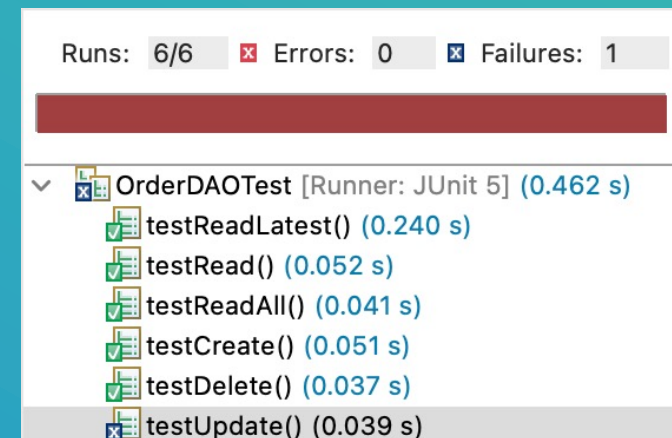
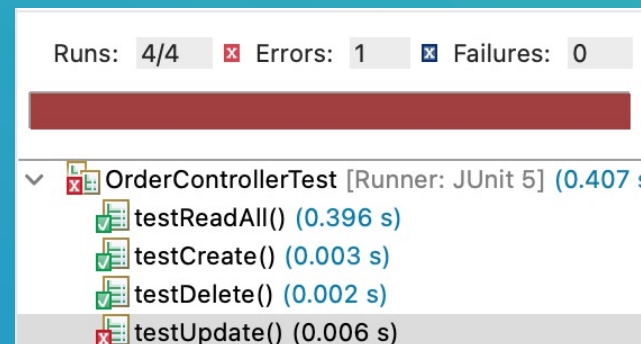
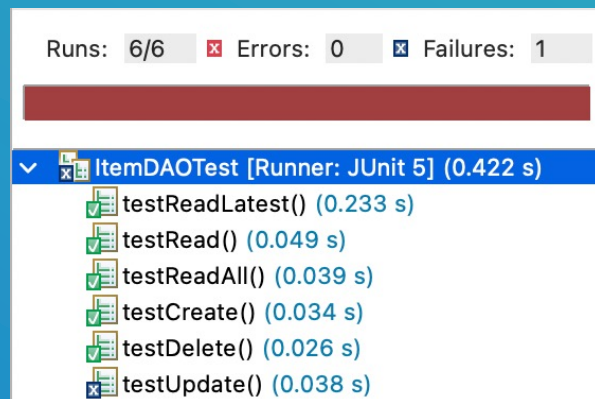
AS YOU CAN SEE FROM THE SPRINT SHOWN IN THE PREVIOUS SLIDE A COUPLE OF THING CAN BE IMPROVED ON.

FIRSTLY, IT SEAMS AS THOUGH COMPLETED WORK IS NOT FOLLOWING THE GUIDELINE ACCURATELY. THIS IS BECAUSE ALTHOUGH ISSUES WERE CREATED EARLY IN THE PROCESS OF THE PROJECT, THEY WERE NOT INITIALIZED INTO A SPRINT UNTIL LATER. THIS WAS A MISTAKE OF THE USER (ME). THIS COULD BE IMPROVED ON NEXT TIME IN ORDER TO GET A MORE ACCURATE IDEA OF PROGRESS OF WORK.

SPRINT RETROSPECTIVE

AS WAS SEEN ON THE SLIDE WITH THE USER STORY FOR ITEM TEST FUNCTIONALITY NOT EVERYTHING WAS COMPLETED IN TIME. SOME THINGS THAT WERE LEFT BEHIND WERE:

- IMPLEMENTATION OF WORKING JUNIT TEST FOR UPDATING AN ITEM IN THE SYSTEM.
- IMPLEMENTATION OF WORKING JUNIT TEST FOR UPDATING AN ORDER IN THE SYSTEM.



CONCLUSION

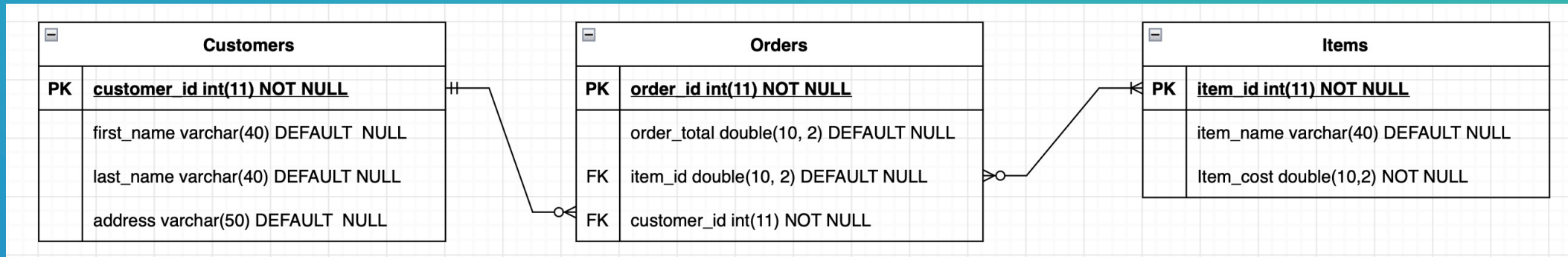
IT IS IMPORTANT TO NOTE THAT SOME MAJOR FEATURES OF THE SYSTEM FUNCTIONALITY WERE NOT INCLUDED. ALTHOUGH ALL SECTIONS (CUSTOMER, ITEM AND ORDER) WORK WITH CRUD, SOME MAJOR AREAS WERE NOT INCLUDED INTO THE SYSTEM.

FOR EXAMPLE, ORDER **TOTAL COST** CANNOT BE VIEWED AS PART OF THE ORDER DETAILS. ADDITIONALLY, **ITEMS** CANNOT BE ADDED TO THE ORDERS. THE ONLY THING THAT IS INPUT INTO THE SYSTEM TO CREATE AN ORDER IS THE CUSTOMER ID.

THE REASON BEHIND THIS IS FIGURING OUT MANY-TO-MANY RELATIONSHIPS CANNOT WORK IN **JAVA** CODE. THIS IS DISPLAYED IN THE FOLLOWING SLIDE BY USE OF ERD DIAGRAMS.

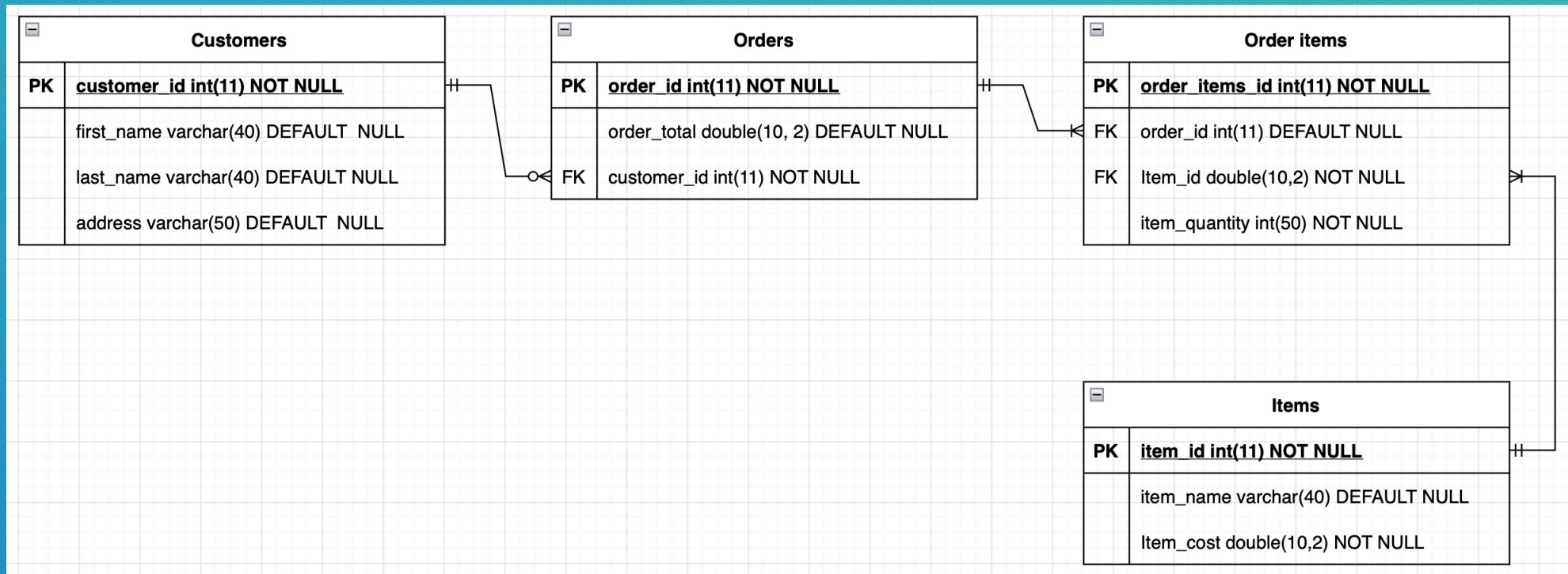
CONCLUSION

INITIAL ERD DIAGRAM



CONCLUSION

UPDATED ERD DIAGRAM



QUESTIONS

ARE THERE ANY QUESTIONS?

+

o

•