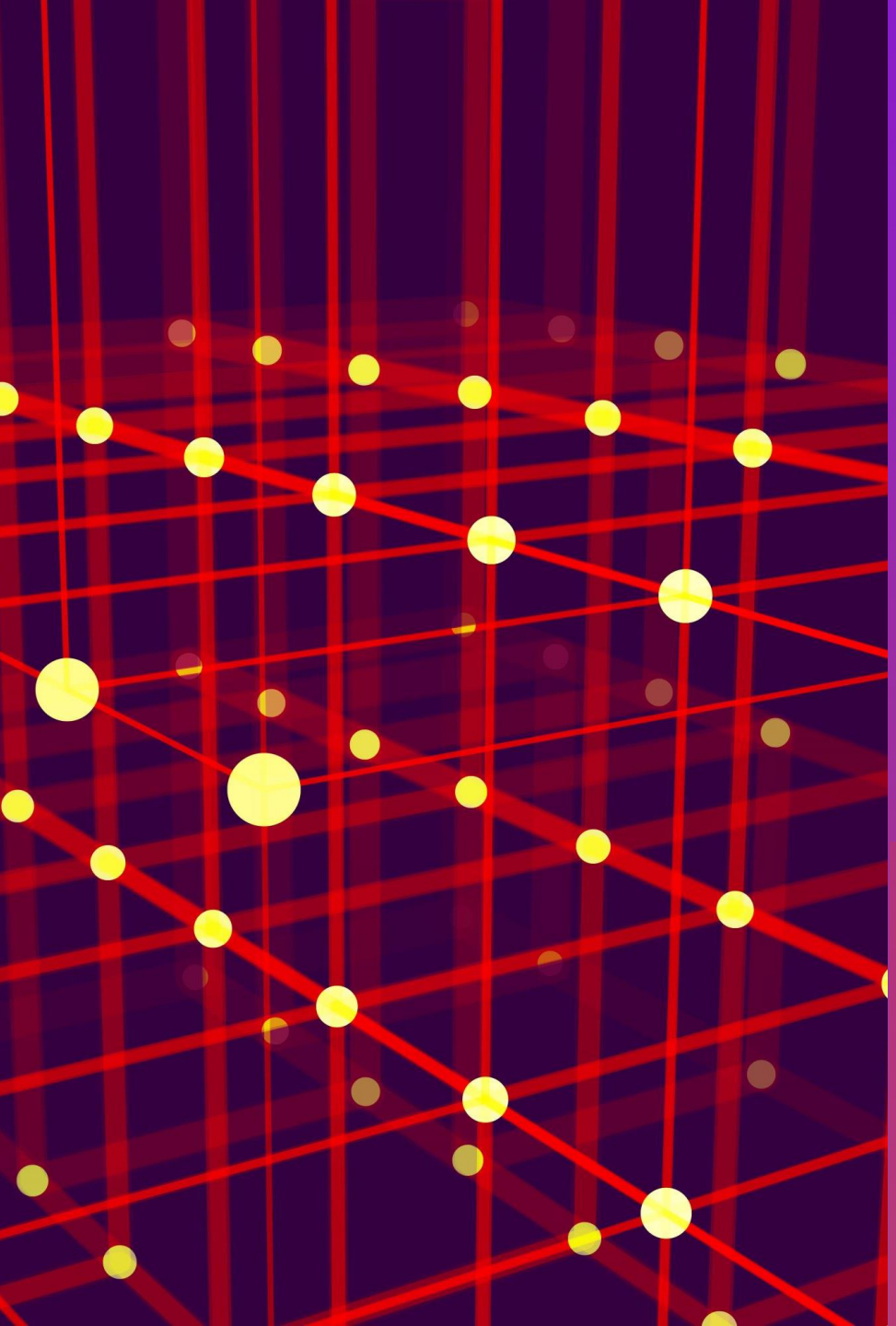


INVENTORY MANAGEMENT SYSTEM(IMS)

QA PROJECT

BY JAMES FERNANDES

INTRODUCTION

APPROACHING THE PROJECT SPECIFICATION

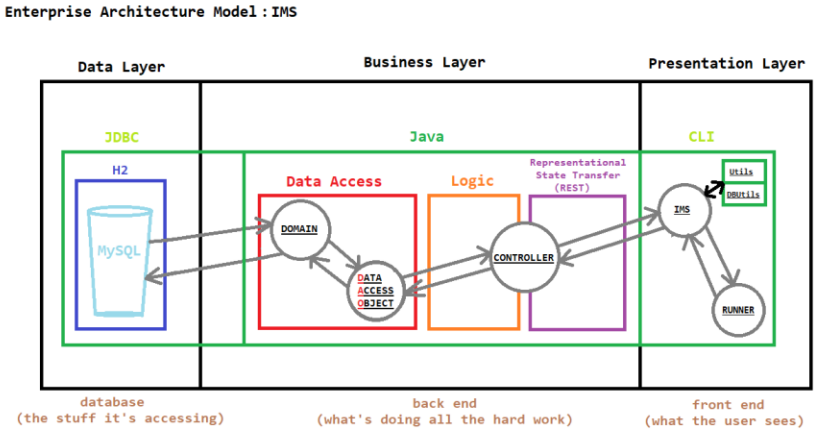
- Reading through the specification and familarsing with concepts learned during QA acadamy to be applied during the project
- Aim of the Project is to create a working IMS system using Java that include Customers, Items and Orders and integrating the code with a MySQL database to store them. Testing of Code with industry standard coverage
- Creating a JIRA board for planning the project with epics,issues and user-stories
- Creating a Risk Assessment and an ERD to model the database structure
- Analysing Base Start Code and start implementing and testing features
- Uploading to GitHub and utilising the Feature-Branch approach

CONSULTANT JOURNEY

TECHNOLOGIES LEARNED FOR THIS PROJECT

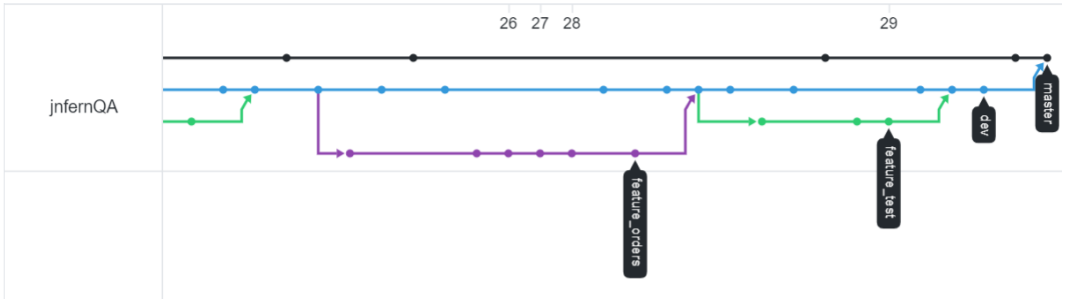
- JIRA : An online project planning software that utilise AGILE/SCRUM approach
- GITHUB : An online software for storing, implementing and development of software in repositories using the version control as Git. Implementing the Feature-Branch approach for better workflow on the software.
- MySQL Database : Either an online platform using gcp or a local database that stores data and utilised query's for implementing changes that manipulate the database
- Java Programming Language : Learned the basics of Java programming with intermediate concepts to solve problems within the code and implement features for the project. Using the Eclipse IDE.
- JUnit Testing : Using Maven build and coverage testing code

CONTINUOUS INTEGRATION



HOW THE VERSION CONTROL WAS APPROACHED

- The Version Control utilised was Git using the Gitbash program to initialise the repository locally that was forked and cloned from a Starter Repo on Github.
- Implementing Changes using the Eclipse IDE creating a MAVEN Build
- Creating multiple branches and represented features to be added to the master branch, each feature to be added was Items, Orders and Testing. Pushing from my local branch to my online repository and merging when a feature was complete to the dev branch.
- After Completion Merging the dev branch to the master branch when the program was working



TESTING

WHAT WAS TESTED?

- The testing was done using JUnit that ran a coverage test on the following classes:
- CUSTOMER, ITEMS, ORDERS, CUSTOMERDAO, ITEMSDAO and ORDERSDAO
- This was done by creating test classes in the source test folder to run the tests from and to compare expected to actual outcomes of the coverage tests.

Results From the Coverage Test:

DEMONSTRATION

SPRINT REVIEW

WHAT WAS COMPLETED?

- Almost all the scope was completed with a working IMS program that managed a database on MySQL for Customers, Items and Orders with all the funtionallity of CRUD.
- JUnit Test of Customers and Items and its DAOs were completed without failures

WHAT GOT LEFT BEHIND?

- As noticed from the testing section the OrderDao was not completing and failing due to a "stackoverflow error" which may have been solved if given more time on the project

SPRINT RETROSPECTIVE

WHAT WENT WELL?

- Getting the main coded program working and its workflow through Github

WHAT COULD BE IMPROVED?

- Testing could be improved as shown by the low-test coverage percent
- Code base could be improved as getting "Too many connections" with MySqL
- Ordering my code more logically
- Planning for the project more efficiently and addressing problems early as possible as this caused delays in the project and incompletion aswell

CONCLUSIONS

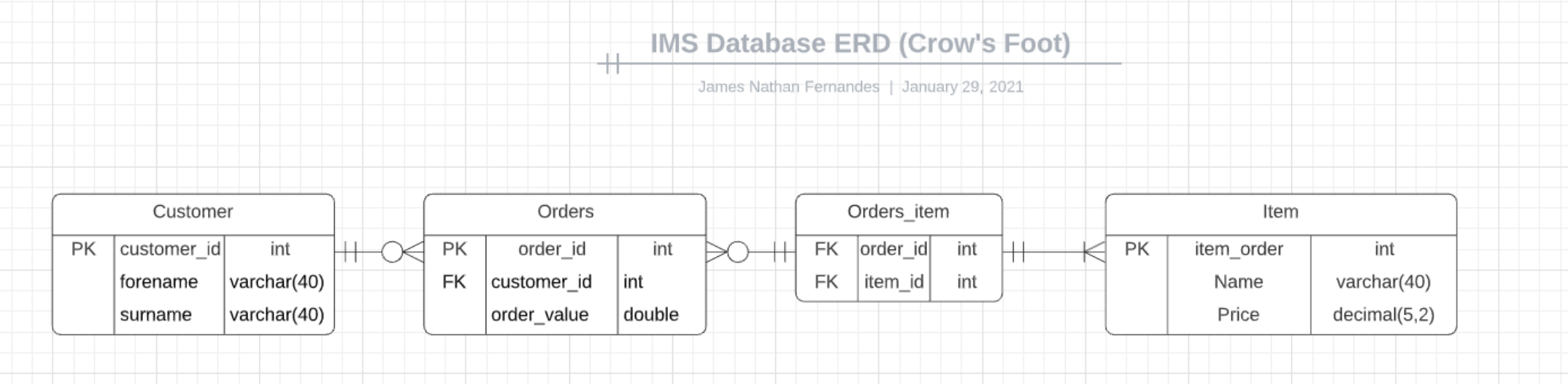
REFLECTIONS ON THE PROJECT

- Learned a lot of concepts during this project and motivated its use in many fields and industry for different projects

FUTURE STEPS

- Reinforcing Concepts learned more often through practice and addressing problems, worries,etc. much earlier to avoid delays in future projects

QUESTIONS


IMS Database ERD (Crow's Foot)
James Nathan Fernandes | January 29, 2021
Customer
PK customer_id int
forename varchar(40)
surname varchar(40)
Orders
PK order_id int
FK customer_id int
order_value double
Orders_item
FK order_id int
FK item_id int
Item
PK item_order int
Name varchar(40)
Price decimal(5,2)