

Software Core Fundamental Project: Inventory Management System (IMS)

By Joseph Price

```
for object to mirror_mod.mirror_object
operation == "MIRROR_X":
    mirror_mod.use_x = True
    mirror_mod.use_y = False
    mirror_mod.use_z = False
operation == "MIRROR_Y":
    mirror_mod.use_x = False
    mirror_mod.use_y = True
    mirror_mod.use_z = False
operation == "MIRROR_Z":
    mirror_mod.use_x = False
    mirror_mod.use_y = False
    mirror_mod.use_z = True

#selection at the end -add
mirror_ob.select= 1
modifier_ob.select=1
context.scene.objects.active
obj("Selected" + str(modifier_ob.name))
mirror_ob.select = 0
obj = bpy.context.selected_objects[0]
obj.data.objects[one.name].select

print("please select exactly one object")

--- OPERATOR CLASSES ---

bpy.types.Operator):
    """Add X mirror to the selected object.mirror_mirror_x"""
    bl_label = "Mirror X"
```

INTRODUCTION

My name is Joseph Price, and I am currently a QA JanSoftware22 cohort Trainee.

- I am currently learning to use a wide range of software applications such as Java, MySQL and Git.
- The next 15 minutes I will be going over and taking you through my first ever full stack project!
- I really enjoyed this project as I had very little experience with Java, mySQL and Git and therefore now was the opportunity to push myself, learn new skills and work within a professional environment, doing something I enjoy.
- Although it was really challenging at times, I can confidently say that my skills have increased drastically, and I have been capable of completing almost the entire project specification.
- Furthermore, I understand all utilised concepts and can also identify my mistakes and already know what I would do next time in order to improve.
- I am entering week 6 as this presentation is taking place and I can happily say I have really been enjoying my time at QA.

APPROACHING THE SPECIFICATION

To start off the project, I initially ran through the specification to identify what was asked of me and whether there were any important restraints or factors that I had to consider.

I also needed to find out the deadline and what tools and technologies we would be using to develop, integrate and test this project.

The project itself was very linear in a sense that in order to complete 1 part, you had to complete something prior, for example, before testing my IMS controllers within Junit, they first had to be created.

My first task was to break down the project into smaller and more concise/manageable objectives and this was done via the creation of User Stories on my Jira Scrum Board.

At this point I was able to identify what I had to do and when, such as creating an ERD, a UML diagram, starting sprints and creating the schema for MySQL database.

CONSULTANT JOURNEY - WHAT TECHNOLOGIES HAVE I LEARNT?

Version Control System - Git

Source Code Management - GitHub

Kanban Board - Jira

Database - MySQL Server 5.7+ (local or cloud hosted)

Back-end Programming Language - Java

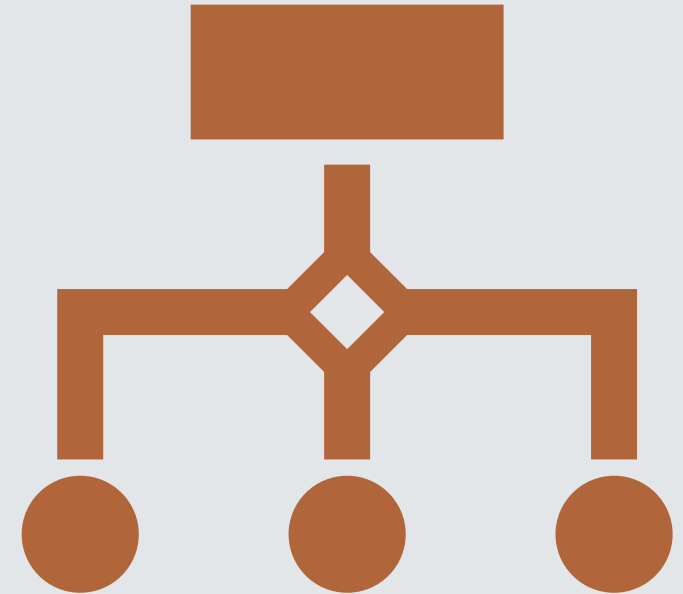
Build Tool - Maven

Unit Testing - JUnit

CONTINUOUS INTEGRATION (CI)

Version Control:

- Utilising the Feature Branch Model.
 - Main -> Dev -> Feature
- Coding scripts and commits would be carried out on the Feature branch.
- When work was to a satisfactory level, it would then be merged into the Dev branch before finally pulling it into the Main branch when all work was complete and project specification was met.



JIRA

Within my Jira application I composed 4 Epics:

- Create an Inventory Management System using MySQL.
- Integrate and further develop my IMS created through MySQL in Java using tools like Maven etc.
- Integrate my IMS into JUnit to test functionalities of CRUD, Domains and DAOs.
- Implement all external, relevant and completed files/code used throughout the project into GitHub.

Each Epic was made up of a handful of concise and useful User Stories, all with Child Issues and broken down into multiple sprints all with a time-frame.

Each User Story was assigned Story Points to dictate what User Story would require the most time, energy and resources.

Each User Story contained a Blocker, in other words, before one User Story could be started, you had to complete the one prior. (Step-by-step format)

On the next slide you will find a screenshot of my project on Jira, including a Roadmap and will notice I stuck to the User Story structure i.e., as a (...), I want (...), so that (...).

ENTITY RELATIONSHIP DIAGRAM (ERD)

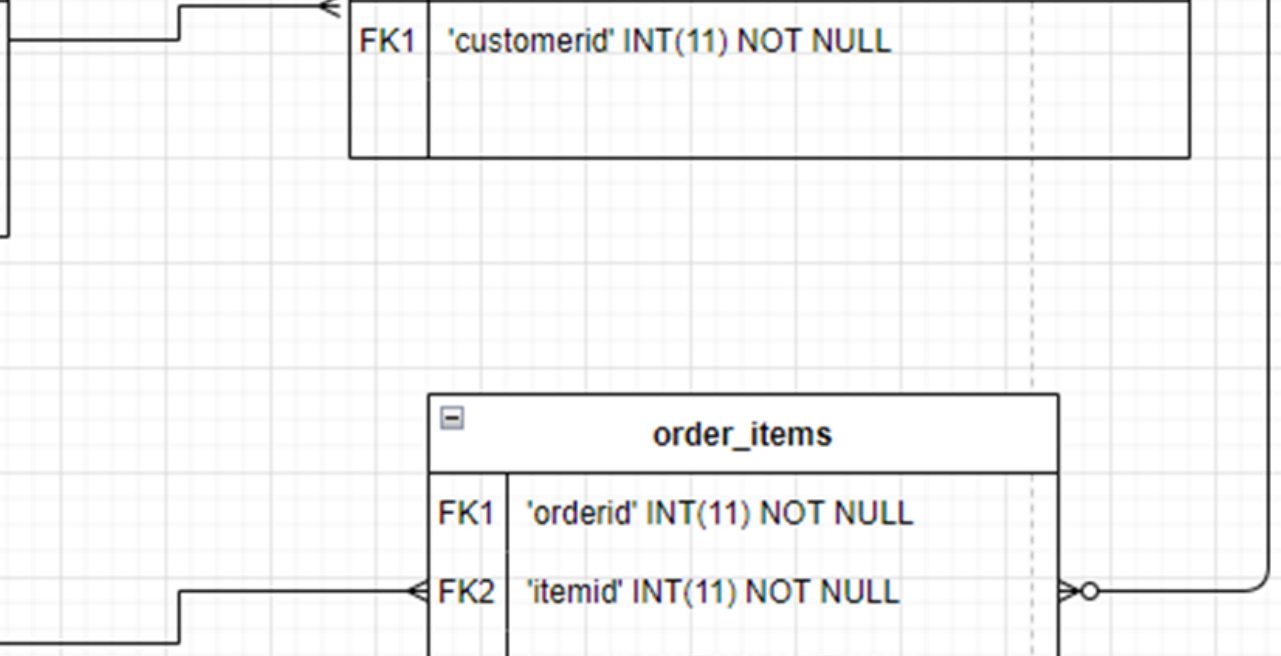
Here you will find my Entity Relationship Diagram that I created via DrawIO. The ERD can be used to identify all entities within my database, their attributes, any Primary and Foreign keys, and can also be used to understand the relationship between each entity i.e., 'One-to-One'.

customers	
PK	<u>'id' INT(11) NOT NULL AUTO-INCREMENT</u>
	'first_name' VARCHAR(40) NULL DEFAULT NULL
	'surname' VARCHAR(40) NULL DEFAULT NULL

orders	
PK	<u>'id' INT(11) NOT NULL AUTO INCREMENT</u>
FK1	'customerid' INT(11) NOT NULL

items	
PK	<u>'id' INT(11) NOT NULL AUTO INCREMENT</u>
	'item_name' VARCHAR(40) NULL DEFAULT NULL
	'item_value' DOUBLE(99,2) NOT NULL DEFAULT 0

order_items	
FK1	'orderid' INT(11) NOT NULL
FK2	'itemid' INT(11) NOT NULL
	'quantity' INT(11) NOT NULL















RISK ASSESSMENT

	A	B	C	D	E	F	G
1	ID	Risk Description	Likelihood(1-5)	Impact (1-5)	Risk Rating	Method of Prevention	Action Plan
2	1	Time Management	3	4	7	Create realistic user stories for accurate prioritisation.	Re-organise or plan new order of prioritisation. Access what can be done in time.
3	2	Network Issues	2	3	6	Regularly save and commit work. Store work on a local device.	Work on a local branch or device. Connect to a hot-spot or use mobile data.
4	3	Data Loss	4	5	6	Backup data regularly. Keep data via passwords.	Revert back to most recent saved file or commit.
5	4	Code Format	5	4	6	Use correct syntax, indentations, use suitable variable names, follow correct layout etc.	Regular static and dynamic testing, spell check work or create separate branches to fix/edit code.
6	5	Cyber-crime	1	3	5	Protect files with passwords, save work regularly, do not share private information.	Revert back to most recent saved file or commit, report to the authorities, change passwords as soon as possible.
7	6	Natural Disasters	1	2	4	Keep work stored on a virtual cloud system like AWS, GoogleDrive etc.	Access a cloud storage system that contains the most recent form of your work/code.
8	7	Poor Data	4	5	6	Check data for errors, remove redundant data, gather data from reliable sources.	Data normalisation.

Matrix: A risk matrix is a matrix that is used during risk assessment to define the level of risk by considering the category of probability or likelihood against the category of consequence severity. This is a simple mechanism to increase visibility of risks and assist management decision making.

TESTING

Element	Coverage	Covered Instructions	Missed Instructions	Total Instructions
ims-project-qa	 70.5 %	3,028	1,266	4,294
src/main/java	 62.2 %	1,830	1,114	2,944
com.qa.ims.persistence.dao	 38.6 %	462	736	1,198
com.qa.ims	 0.0 %	0	186	186
com.qa.ims.persistence.domain	 86.5 %	556	87	643
com.qa.ims.utils	 70.2 %	174	74	248
com.qa.ims.controller	 95.8 %	638	28	666
com.qa.ims.exceptions	 0.0 %	0	3	3
src/test/java	 88.7 %	1,198	152	1,350
com.qa.ims.persistence.dao	 77.7 %	373	107	480
com.qa.ims.controllers	 94.7 %	801	45	846
com.qa.ims.persistence.domain	 100.0 %	24	0	24

As you can see, as of right now, I tested for 70.5% of the total project. Unfortunately, this did not meet the industry standard coverage. However, I believe the reason why was because of test script errors that involved MySQL connectivity.

All functionalities work as intended and reach project requirements.



DEMONSTRATION

SPRINT REVIEW

What Did I Complete?

- I was able to complete the full CRUD functionality for all entities.
- I was able to create a database in MySQL and input relevant data using correct data types and keys.
- I was also able to carry out testing on my code to check for coverage and identify any bugs/errors.
- I was able to commit code within their respective branch, eventually merging them together before pulling code into a repository where it can be marked.
- I was able to compose an ERD and a UML that can be used to visualise the database.
- I was able to compose a Risk Assessment using a Matrix.

What
Got Left
Behind?

Unfortunately, I was unable to achieve an 80% total code coverage on my project.

I was also unable to implement any of the extension features in the project specification

SPRINT REVIEW

SPRINT RETROSPECTIVE

What Went Well?

Overall, I feel that the project went well as I was able to meet the majority of the project requirements.

- CRUD Functionalities
- MySQL Database
- Jira
- Documentation - ERD, UML, Risk Assessment etc.
- Git/GitHub

CONCLUSION

What Could I Do To Improve?

- Time Management
- Improve Test Coverage
- Improve my use of Jira
- Attempt Extension Tasks
- Feature Branch Model





CONCLUSION

Reflection

- Learned a wide range of very important and relevant skills using new software tools.
 - Independent Research.
- Gained a further insight into the professional working world.
 - Fun.



ANY QUESTIONS?