

# IMS PROJECT – 22ENABLEMAY2

Saiful Hussain

# INTRODUCTION

- Create an Inventory management system (IMS) with CRUD functionality.
- User can interact through a command line interface (CLI).
- Use GitHub and GitBash
- SQL
- Java

Run tests on the IMS for functionality with Junit (domain and DAO) and Mockito (controller)

# RISK ASSESSMENT

Risk	Risk Statement	Response strategy	Objectives	Likelihood	Impact	Risk Level
GitHub	Sensitive codes which contain private information pushed to GitHub could potentially be hacked into and manipulated or leaked.	Use strong passwords containing special characters which are regularly updated and known only by you.	Reduce the likelihood of hacking, leaks and corruption of data.	Medium	High	Medium-to-High
Loss of data	Closing tabs or applications by accident or deleting/rewriting codes by mistake could risk losing unsaved work.	Regularly update and push to GitHub especially after completing important sections of work.	Prevent loss of data and time re-doing work.	Low	High	Medium
Hardware failure	Hardware such as PC or input devices such as mouse or keyboard can be damaged or stop working during working could cease and prevent continuation of work.	Replace input devices as soon as possible. If Hardware such as PC becomes damaged or crashes, it may be possible to switch to another device and carry-on work from there.	Ensure that minimal time is wasted due to not having access to a device to do work on.	Low	medium	Medium to low
Inconsistent internet connection	Wi-Fi connection may fluctuate and cut out which may prevent user from connecting to the internet or have difficulty navigating websites for work.	Use a wired internet connection or upgrade broadband package to boost internet speeds. Another alternative may be to use mobile data and hotspot to device to access the internet.	Ensure continuation of work if my Wi-Fi connection fluctuates or cuts out.	Low	Low	Low
Out-dated software or drivers.	If the software or driver is not at the latest or at a newer version, certain functions may not be present or may work differently when trying to integrate with other software application and prevent continuation of work.	Keep programming software and drivers up to date.	Ensure sound and efficient functionality of application with no setbacks or bugs.	Low	Medium	Medium-to-Low

Illness and Injury	Sudden illness or injuries could result in extended periods of time away from work which may result in setbacks to a work project or assignment.	Keep good hygiene habits and perform activities safely. If injuries or illnesses do occur, then consult a doctor at the earliest convenience to reduce time away from work.	Reduce the likelihood of getting an illness, and passing it onto others, and injuries to ensure time away from work is as minimal as possible.	High	High	High
--------------------	--	---	--	------	------	------

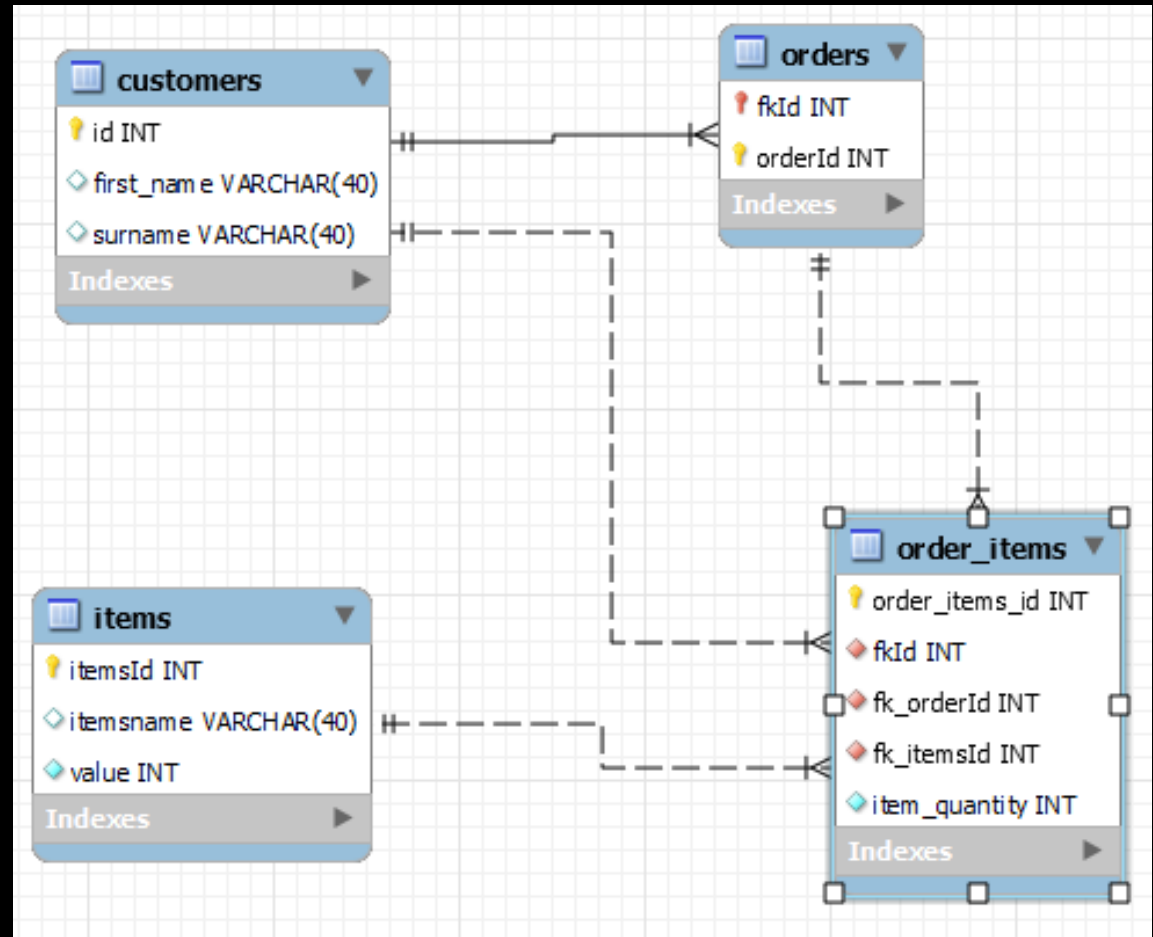
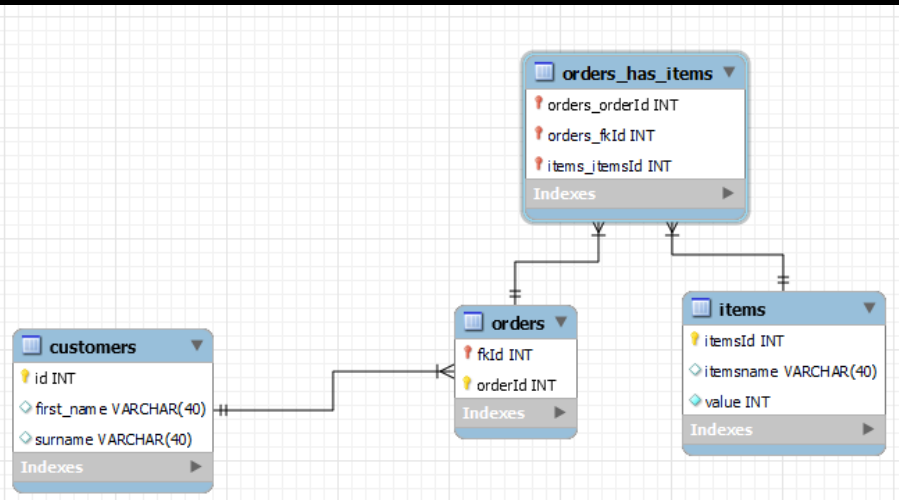
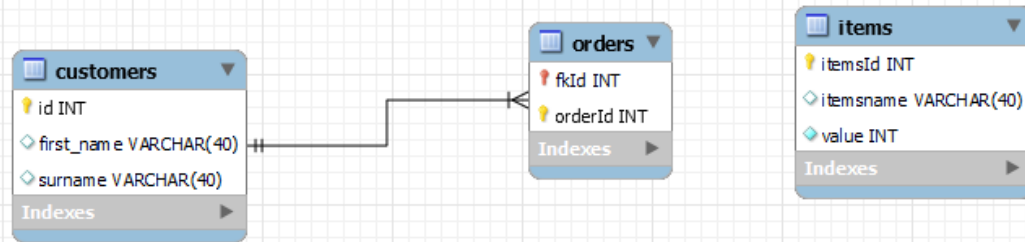
# RISK ASSESSMENT MATRIX

Risk Assessment Matrix		Level of Impact				
		Low	Medium-Low	Medium	Medium-High	High
Likelihood	Low	Low	Low	Medium-Low	Medium	Medium
	Medium-Low	Low	Medium-Low	Medium-Low	Medium	Medium-High
	Medium	Low	Medium-Low	Medium	Medium-High	Medium-High
	Medium-High	Low	Medium-Low	Medium	Medium-High	High
	High	Medium-Low	Medium	Medium-High	High	High

# MOSCOW FRAMEWORK

Must Have	Should Have	Could Have	Won't Have
<ol style="list-style-type: none"><li>1. A risk assessment with 5 risks and a matrix.</li><li>2. A MoSCoW framework to outline the needs, wants and cans of the project.</li><li>3. A completed <b>project management board</b>, including user stories, epics and tasks.</li><li>4. A working application that is interactive with a front-end command-line interface (CLI).</li><li>5. A <b>fat .jar</b> in the root folder.</li><li>6. Using Junit and Mockito to test functionality of the application.</li><li>7. CRUD functionality for the <b>customers</b>, <b>items</b>, and <b>orders</b> entities.</li><li>8. Be able to add and delete an item to and from an order.</li><li>9. The means to calculate the cost of an order.</li><li>10. A relational database containing the <b>customers</b>, <b>items</b>, <b>orders</b>, and <b>orders_items</b> tables.</li><li>11. A primary key in <b>customers</b>, <b>items</b>, <b>orders</b>, and <b>order_items</b> tables.</li><li>12. A <b>README.md</b> file to introduce and explain the information needed to understand project and use the application.</li><li>13. Use of Dev and main branch model to push and update on GITHUB</li><li>14. At least <b>one ERD</b> and <b>one UML</b> diagram, in <b>.png</b> format</li><li>15. Slide-show Presentation of the project.</li></ol>	<ol style="list-style-type: none"><li>1. Proficiently using and updating the feature-branch-model on GITHUB.</li><li>2. AT least 5 risks on risk assessment with a clear, colour coded matrix.</li><li>3. Use ERD to display relationships between tables.</li><li>4. Before and after ERD's and UML.</li><li>5. Unit test coverage of CRUD functionality which should come back approximately 80%.</li><li>6. A fat .jar which can be deployed from the command-line.</li><li>7. Adhering to best practice (OOP principles, SOLID, refactoring).</li><li>8. A working ".ignore" for ignoring build-generated files and folders.</li><li>9.The means to calculate order total.</li></ol>	<ol style="list-style-type: none"><li>1. CRUD Functionality testing to be above 80%.</li><li>2. customer username and password so they can log into unique private accounts.</li><li>3. Orders table to show date and time of order placed.</li><li>4. Description of the item and how many items in stock.</li><li>5. ERD and UML diagram progression.</li><li>6. A Graphical User Interface to make it easier to use the system.</li></ol>	<ol style="list-style-type: none"><li>1. Adverts</li><li>2. Log In functions</li><li>3. Use of customer emails function.</li><li>4. Receipt of customers order emailed to them.</li><li>5. Be missing any documentation.</li></ol>

# ERD'S





## Edit sprint: Week 5 Project

Sprint name \*

Week 5 Project

Start date \*

6/6/2022 12:00 PM

End date \*

6/13/2022 9:00 AM

Sprint goal

Complete Inventory Management System.

Update

Cancel

Projects / QA

## Backlog

SH

Epic

Type

Issues without epic

> Documentation

> Customers

> Items

> Orders

> Order items

> SQL

> Testing

+ Create Epic

Week 5 Project 6 Jun - 13 Jun (37 issues)

Complete Inventory Management System.

10 133 Complete sprint

4 DONE

5 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

8 DONE

## Week 5 Project

Complete Inventory Management System.

SH

Epic

Type

TO DO 2 ISSUES

IN PROGRESS 4 ISSUES

DONE 31 ISSUES

As a user, I want to be able to calculate a cost for an order so that I can view the final/total price of a customer's order.

ORDER ITEMS

QASH-22 10

Create a README.md document to introduce and explain the information needed to understand project.

DOCUMENTATION

QASH-24

Create a before ERD to describe the initial structure of the database.

DOCUMENTATION

QASH-27 2

Create an after ERD to describe the final structure of the database.

DOCUMENTATION

QASH-38 2

Create a UML to describe the building of the software.

DOCUMENTATION

QASH-28

Create a Powerpoint presentation to present the project.

DOCUMENTATION

QASH-29 4

Create a risk assessment to identify possible threats and a plan of action to prevent and fix the problem.

DOCUMENTATION

QASH-25 4

Create a MoSCoW to outline the requirements of the project

DOCUMENTATION

QASH-26 5

As a user, I want to be able to view all customers in the system so that I can find information from them.

CUSTOMERS

QASH-11 8

As a user, I want to be able to update a customer in the system so that I can keep an update database on their information.

CUSTOMERS

QASH-11 8

Epic

Issues without epic

> Documentation

> Customers

> Items

> Orders

> Order items

> SQL

> Testing

+ Create Epic

Customers / QASH-11

As a user, I want to be able to view all customers in the system so that I can find information from them.

Attach Add a child issue Link issue

Description

Add a description...

Child issues

Order by

0% Done

QASH-42 Create a Customer Domain class On Java 4

QASH-43 Create a Customer DAO Class on Java 5

QASH-44 Create an Customer Controller Class on Java 6

Activity

Show: All Comments History

Newest first

SH

Add a comment...

Pro tip: press M to comment

Done Done

Details

Assignee Unassigned

Labels None

Sprint Week 5 Project

Story point estimate 8

Development Create Branch

Create Commit

Reporter SH Saiful Hussain

Created 3 days ago

Updated 1 hour ago

Resolved 1 hour ago

Configure

# JIRA

MySQL Workbench

Local instance MySQL80 x MySQL Model\* x EER Diagram2 x EER Diagram3

File Edit View Query Database Server Tools Scripting Help

Navigator

SCHEMAS

Filter objects

exercise  
gameshopdbwo  
ims  
Tables  
customers  
items  
order\_items  
orders

Administration Schemas

Information

No object selected

SQL Tables

```
1 DROP SCHEMA IF EXISTS ims;
2
3 CREATE SCHEMA IF NOT EXISTS ims;
4
5 USE ims ;
6
7 CREATE TABLE IF NOT EXISTS `ims`.`customers` (
8   `id` INT(11) NOT NULL AUTO_INCREMENT,
9   `first_name` VARCHAR(40) DEFAULT NULL,
10  `surname` VARCHAR(40) DEFAULT NULL,
11  PRIMARY KEY (`id`)
12 );
13
14 CREATE TABLE IF NOT EXISTS `ims`.`items` (
15   `itemsId` INT(11) NOT NULL AUTO_INCREMENT,
16   `itemsname` VARCHAR(40) DEFAULT NULL,
17   `value` INT(40) NOT NULL,
18   PRIMARY KEY (`itemsId`)
19 );
20
21 CREATE TABLE IF NOT EXISTS `ims`.`orders` (
22   `fkId` INT(11) NOT NULL,
23   `orderId` INT(11) NOT NULL AUTO_INCREMENT,
24   PRIMARY KEY (orderId, fkId),
25   FOREIGN KEY (fkId) REFERENCES customers(id)
26 );
27
28 CREATE TABLE IF NOT EXISTS `ims`.`order_items` (
29   `order_items_id` INT(11) NOT NULL AUTO_INCREMENT,
30   `fkId` INT(11) NOT NULL,
31   `fk_orderId` INT NOT NULL,
32   `fk_itemsId` INT NOT NULL,
33   `item_quantity` INT(40) NOT NULL,
34   -- `total_cost_of_order` DEC(10, 2) NOT NULL,
35
36   PRIMARY KEY (order_items_id),
37   FOREIGN KEY (fkId) REFERENCES customers(id),
38   FOREIGN KEY (fk_orderId) REFERENCES orders(orderId),
39   FOREIGN KEY (fk_itemsId) REFERENCES items(itemsId)
40 );
```

Output

Action Output

#	Time	Action
353	05:34:57	CREATE SCHEMA IF NOT EXISTS ims
354	05:34:57	USE ims
355	05:34:57	CREATE TABLE IF NOT EXISTS `ims`.`customers`
356	05:34:57	CREATE TABLE IF NOT EXISTS `ims`.`items`
357	05:34:57	CREATE TABLE IF NOT EXISTS `ims`.`orders`
358	05:34:57	CREATE TABLE IF NOT EXISTS `ims`.`order_

Object Info Session

# SQL SCHEMA





# GITHUB

Overview

Yours

Active

Stale

All branches

Search branches...

Default branch

↗

main

Updated 4 days ago by jordanbenbelaid

Default

✎

Your branches

dev

Updated 28 minutes ago by saifhussain98

0 | 38

New pull request

✎

🗑

Testing

Updated 1 hour ago by saifhussain98

0 | 36

New pull request

✎

🗑

items

Updated 14 hours ago by saifhussain98

0 | 9

New pull request

✎

🗑

order\_items

Updated 14 hours ago by saifhussain98

0 | 12

New pull request

✎

🗑

Items

Updated 15 hours ago by saifhussain98

0 | 7

New pull request

✎

🗑

View more of your branches

>

Active branches

dev

Updated 28 minutes ago by saifhussain98

0 | 38

New pull request

✎

🗑

Testing

Updated 1 hour ago by saifhussain98

0 | 36

New pull request

✎

🗑

items

Updated 14 hours ago by saifhussain98

0 | 9

New pull request

✎

🗑

order\_items

Updated 14 hours ago by saifhussain98

0 | 12

New pull request

✎

🗑

Items

Updated 15 hours ago by saifhussain98

0 | 7

New pull request

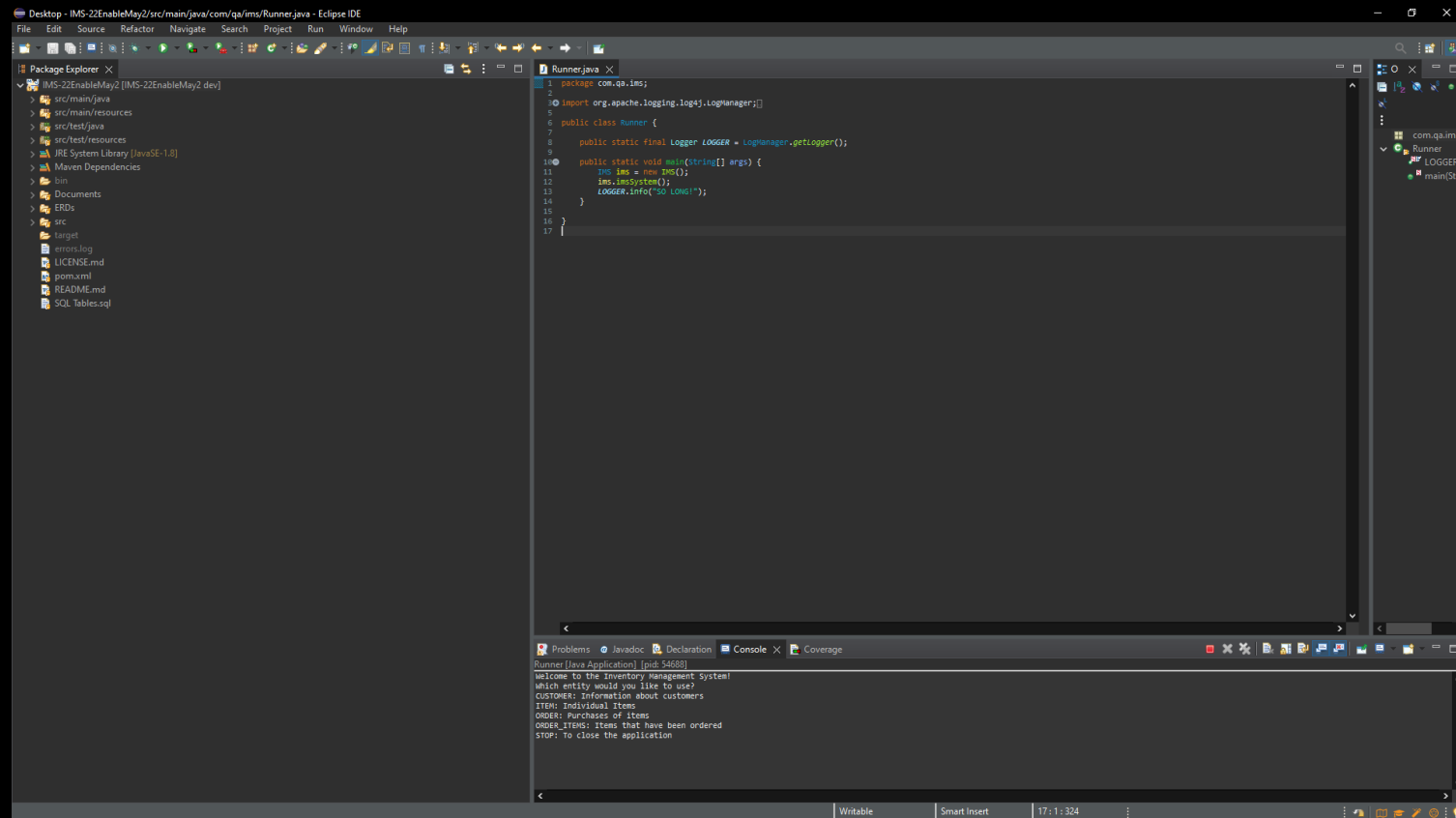
✎

🗑

View more active branches

>




# CODE FULLY FUNCTIONING



```
1 package com.qa.ims;
2
3 import org.apache.logging.log4j.LogManager;
4
5 public class Runner {
6
7     public static final Logger LOGGER = LogManager.getLogger();
8
9     public static void main(String[] args) {
10         IMS ims = new IMS();
11         ims.imSystem();
12         LOGGER.info("SO LONG!");
13     }
14 }
15
16
17
```

Runner [Java Application] [pid: 54688]  
Welcome to the Inventory Management System!  
Which entity would you like to use?  
CUSTOMER: Information about customers  
ITEM: Individual Item  
ORDER: Purchases of Items  
ORDER\_ITEMS: Items that have been ordered  
STOP: To close the application

# TEST COVERAGE

▼	IMS-22EnableMay2		71.8 %	2,933
>	src/main/java		66.3 %	1,772
>	src/test/java		82.3 %	1,161