

Inventory Management System



By Gifty Antwi



Introduction

- Specification – create and test application
- Planning – Jira, ERD (MYSQL), UML
- GitHub
- Java via Eclipse
- Testing – junit & Mockito

Jira

IMSstarterSDET
Software project

Roadmap

Backlog

Board

Code

Project pages

Add item

Project settings

You're in a team-managed project
[Learn more](#)

Projects / IMSstarterSDET

IMSSDET board

GA  Epic 

TO DO 3 ISSUES

Add an item to an order

BUILD AN APPLICATION THAT AN ...

IMSSDET-14

GA

Calculate a cost for an order

BUILD AN APPLICATION THAT AN ...

IMSSDET-15

GA

Delete an item in an order

BUILD AN APPLICATION THAT AN ...

IMSSDET-16

GA

+ Create issue

IN PROGRESS 3 ISSUES

Risk Assessment

BUILD AN APPLICATION THAT AN ...

IMSSDET-24

GA

Update an order in the system

BUILD AN APPLICATION THAT AN ...

IMSSDET-26

GA

Regular testing

BUILD AN APPLICATION THAT AN ...

IMSSDET-23

GA

DONE 18 ISSUES

Delete an order in the system

BUILD AN APPLICATION THAT AN ...

IMSSDET-13

GA

UML Diagram

BUILD AN APPLICATION THAT AN ...

IMSSDET-20

GA

Presentation

BUILD AN APPLICATION THAT AN ...

IMSSDET-25

GA

ERD Diagram

BUILD AN APPLICATION THAT AN ...

IMSSDET-19

GA

Consider the entities in this domain

BUILD AN APPLICATION THAT AN ...

IMSSDET-18

GA

View all orders in the system

BUILD AN APPLICATION THAT AN ...

IMSSDET-12

GA

Maven

BUILD AN APPLICATION THAT AN ...

IMSSDET-22

GA

Create an order in the system

BUILD AN APPLICATION THAT AN ...

IMSSDET-11

GA





Build an application... / IMSSDET-13

  1   ... 

Delete an order in the system

Done

Done

 Attach  Add a child issue  Link issue  ...

Description

As a developer I would like to give the user the ability to delete an order in the system with ease and clarity.

Activity

Show:  All  Comments  History

Newest first 

 Add a comment...

Pro tip: press **M** to comment





Build an application... / IMSSDET-19

  1   ... 

ERD Diagram

Done

Done

 Attach  Add a child issue  Link issue  ...

Description

Relationships should be modelled using an ERD.

Activity

Show:  All  Comments  History

Newest first 

 Add a comment...

Pro tip: press **M** to comment

Done

Done

Assignee  Gifty Antwi

Labels None

Reporter  Gifty Antwi

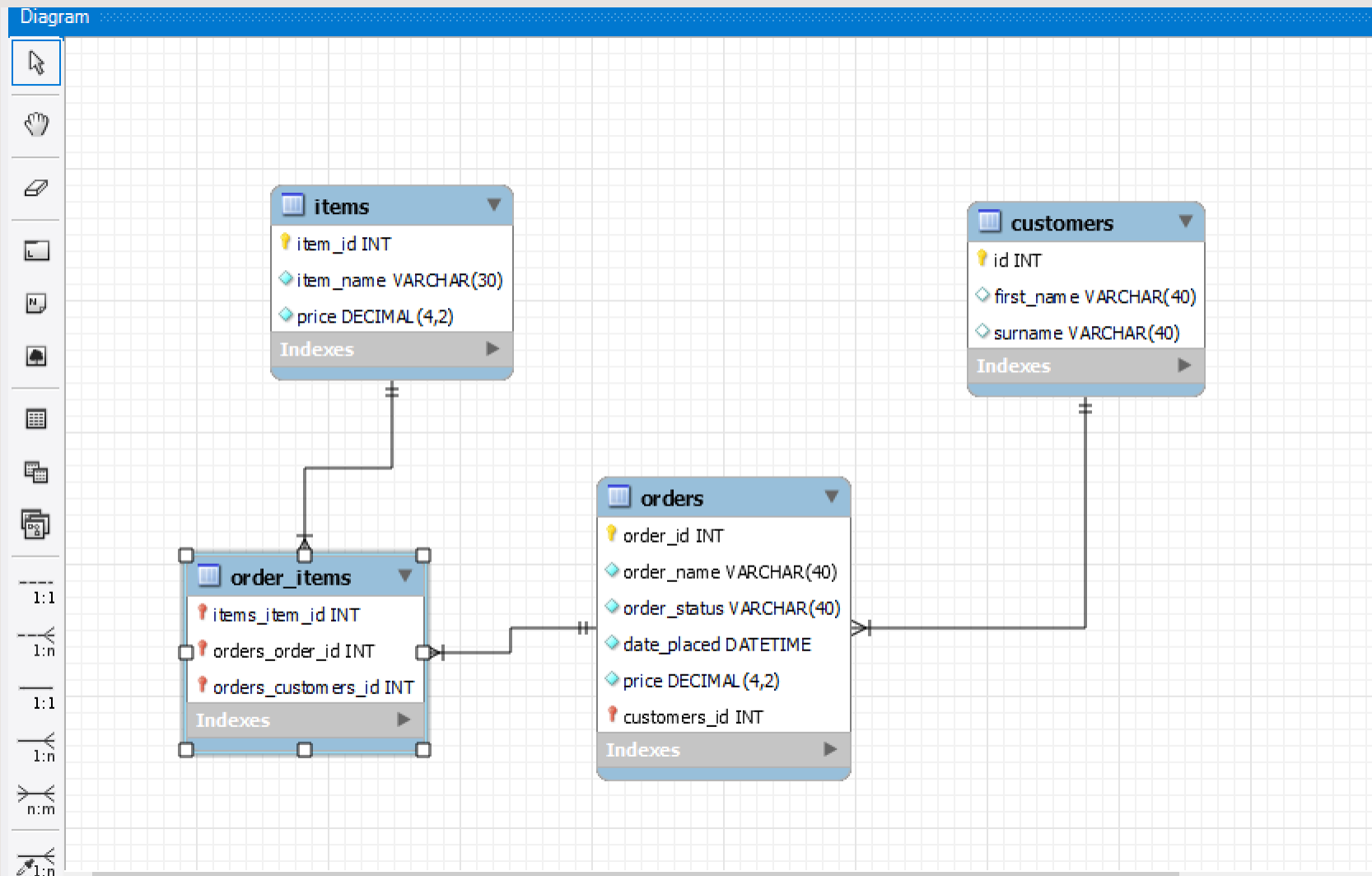
Created 7 days ago

Updated 4 days ago

Resolved 4 days ago

 Configure

ERD Diagram



Database - Schema

```
1 • drop schema ims;
2 • CREATE SCHEMA IF NOT EXISTS `ims`;
3
4 • USE `ims` ;
5
6 • CREATE TABLE IF NOT EXISTS `ims`.`customers` (
7     `id` INT(11) NOT NULL AUTO_INCREMENT,
8     `first_name` VARCHAR(40) DEFAULT NULL,
9     `surname` VARCHAR(40) DEFAULT NULL,
10    PRIMARY KEY (`id`)
11 );
12
13 • CREATE TABLE IF NOT EXISTS `ims`.`items` (
14     `item_id` INT(11) NOT NULL AUTO_INCREMENT,
15     `item_name` VARCHAR(30) NOT NULL,
16     `price` DECIMAL(4,2) NOT NULL,
17     PRIMARY KEY(`item_id`)
18 );
19
20 • CREATE TABLE IF NOT EXISTS `ims`.`orders` (
21     `order_id` INT(11) NOT NULL AUTO_INCREMENT,
```

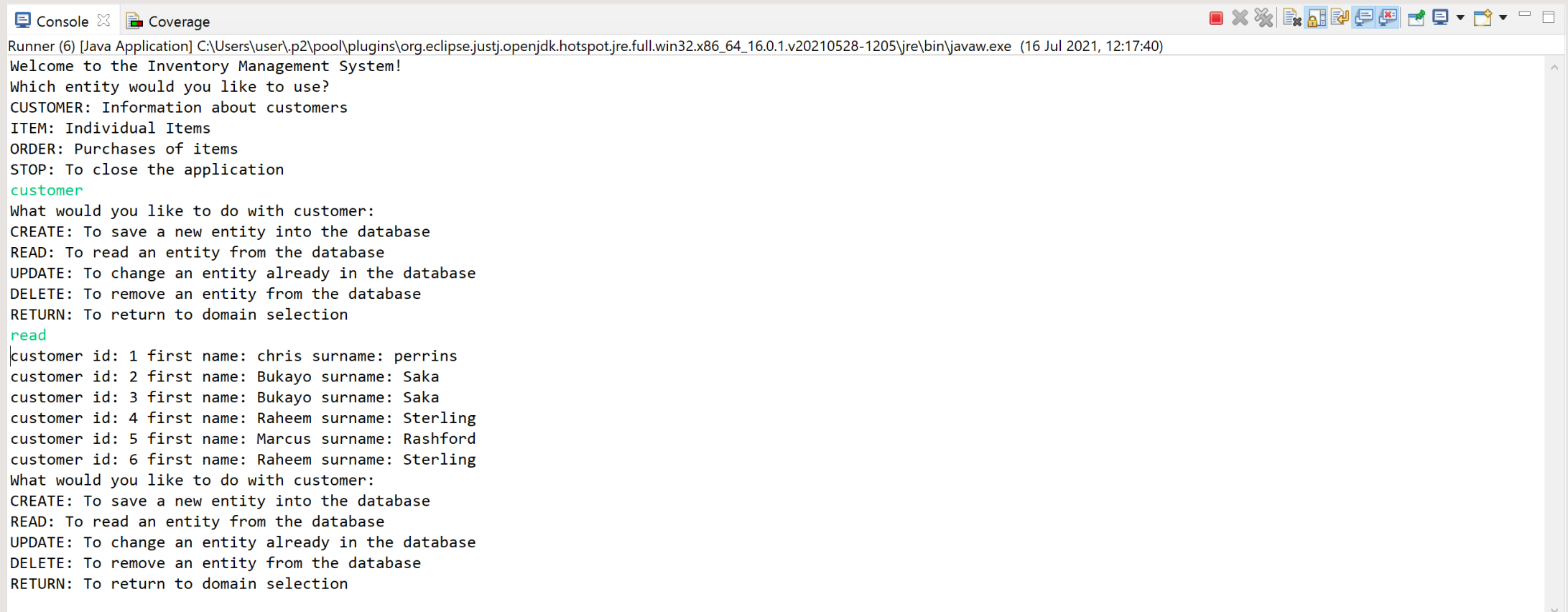
```
19
20 • CREATE TABLE IF NOT EXISTS `ims`.`orders` (
21     `order_id` INT(11) NOT NULL AUTO_INCREMENT,
22     `order_name` VARCHAR(40) NOT NULL,
23     `order_status` VARCHAR(40) NOT NULL,
24     `date_placed` DATETIME(6) NOT NULL,
25     `price` DECIMAL(4,2) NOT NULL,
26     PRIMARY KEY(`order_id`)
27 );
28
29 • CREATE TABLE IF NOT EXISTS `ims`.`orders_items` (
30     `item_id` INT(11) NOT NULL,
31     `order_id` INT(11) NOT NULL,
32     FOREIGN KEY(`order_id`) REFERENCES `orders`(`order_id`),
33     FOREIGN KEY(`item_id`) REFERENCES `items`(`item_id`)
34 );
35
```

Database - Data

```
1 • INSERT INTO `ims`.`customers` (`first_name`, `surname`) VALUES ('Bukayo', 'Saka');
2 • INSERT INTO `ims`.`customers` (`first_name`, `surname`) VALUES ('Marcus', 'Rashford');
3 • INSERT INTO `ims`.`customers` (`first_name`, `surname`) VALUES ('Jadon', 'Sancho');
4 • INSERT INTO `ims`.`customers` (`first_name`, `surname`) VALUES ('Raheem', 'Sterling');
5
6 • INSERT INTO `testingims`.`customers` (`first_name`, `surname`) VALUES ('Raheem', 'Sterling');
7 • INSERT INTO `testingims`.`customers` (`first_name`, `surname`) VALUES ('Bukayo', 'Saka');
8 • INSERT INTO `testingims`.`items`(`item_name`, `price`) VALUES ('Gym Socks', '15.99');
9 • INSERT INTO `testingims`.`items`(`item_name`, `price`) VALUES ('Apple', '0.15');
10 • INSERT INTO `testingims`.`items`(`item_name`, `price`) VALUES ('Water', '1.50');
11 • INSERT INTO `testingims`.`items`(`item_name`, `price`) VALUES ('Protein Shake', '2.29');
12 • INSERT INTO `testingims`.`orders` (`order_name`, `order_status`, `date_placed`, `price`) VALUES ('BS7', 'Processing', '20210712', '3.1');
13 • INSERT INTO `testingims`.`orders_items` (`item_id`, `order_id`) VALUES (1,1);
14
```

Demonstration of Application

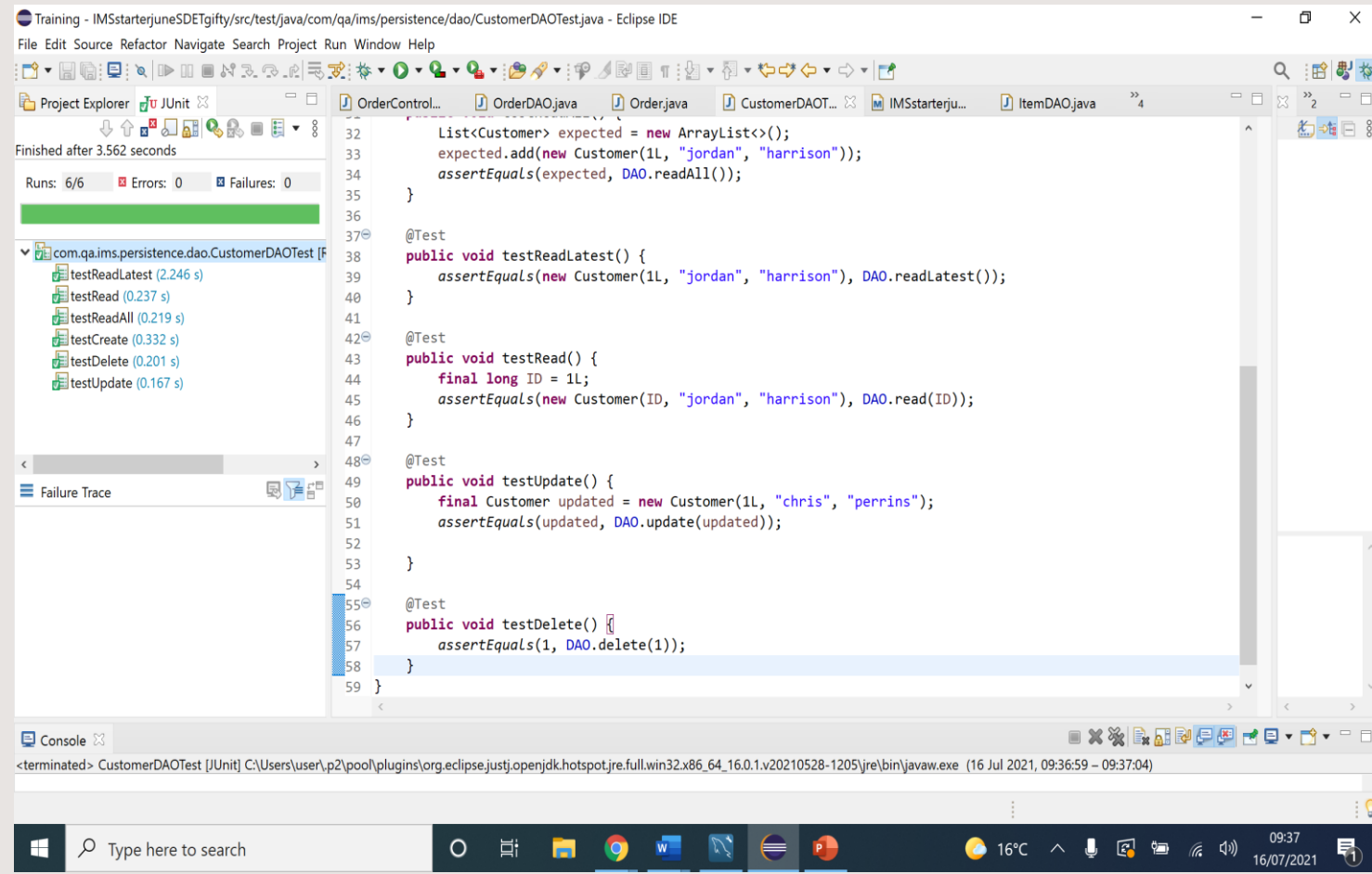
- DAO = Database Access Object – allows controller to access the database information in SQL



```
Runner (6) [Java Application] C:\Users\user\p2\pool\plugins\org.eclipse.justj.openjdk.hotspot.jre.full.win32.x86_64_16.0.1.v20210528-1205\jre\bin\javaw.exe (16 Jul 2021, 12:17:40)
Welcome to the Inventory Management System!
Which entity would you like to use?
CUSTOMER: Information about customers
ITEM: Individual Items
ORDER: Purchases of items
STOP: To close the application
customer
What would you like to do with customer:
CREATE: To save a new entity into the database
READ: To read an entity from the database
UPDATE: To change an entity already in the database
DELETE: To remove an entity from the database
RETURN: To return to domain selection
read
customer id: 1 first name: chris surname: perrins
customer id: 2 first name: Bukayo surname: Saka
customer id: 3 first name: Bukayo surname: Saka
customer id: 4 first name: Raheem surname: Sterling
customer id: 5 first name: Marcus surname: Rashford
customer id: 6 first name: Raheem surname: Sterling
What would you like to do with customer:
CREATE: To save a new entity into the database
READ: To read an entity from the database
UPDATE: To change an entity already in the database
DELETE: To remove an entity from the database
RETURN: To return to domain selection
```

Testing

- Mockito - Creates a mock item so you can test without creating a database
- Junit - running code under various circumstances to highlight where issues occur



Composition line = child cannot exist without the parent

UML Diagram

