



IMS (Inventory Management System)

Siddhartha Gurung



Introduction

Consultant Journey



```

* 5a1cd2e created static ip address variable
* 829fa43 updated ims readFile test to make sonar happy
* b050053 Merge pull request #11 from qasidd/feature/gcp
|/
* ffd5626 (origin/feature/gcp, feature/gcp) version and jar updated
* 6889226 changed path to sql file in tests to work with new init method
* c4f6292 changed url in tests to connect to a gcp instance
* 8281711 changed url so it connects to a gcp instance
* 6f3f1ee Update README.md
|/
* d1833b3 readme img dir
* 04bdeff Update README.md
* 3077b3d docs dir added, updated license.md
* 8853b7d jar updated
* 763bc54 Merge pull request #10 from qasidd/feature/code-smell-fixes
|/
* c920177 (origin/feature/code-smell-fixes, feature/code-smell-fixes) applied fixes sonarqube
suggested
|/
* 4cf28c5 attempt to fix pound symbol
* 89a2513 jar file added to root dir
* ee90340 Merge pull request #9 from qasidd/feature/fixes-for-jar
|/
* 133679c (origin/feature/fixes-for-jar, feature/fixes-for-jar) version number update
* 2f06404 updated orderdaomysql test to work with new methods
* 5711a8a init method updated to include readFile
* a6a54c3 added ioutils dependency, readFile method updated
|/
* e2b2445 artifactId and version number updated
* be27177 Merge pull request #8 from qasidd/feature/utills
|/
* b3ffca1 (origin/feature/utills, feature/utills) final changes to utills
* b405414 controller tests updated
* dc18717 fixed infinite loop error, changed valueOf calls on controller classes
* 3ca653d added getInt, getDouble and getLong to utills
|/
* a909dd2 Merge pull request #7 from qasidd/feature/mvp-tests
|/
...skipping...
* b7d1cec (HEAD -> dev, origin/dev) risk assessment added to docs
* c8eb87e Merge branch 'dev' of https://github.com/qasidd/ims-demo into dev readme
|/
* ac3347a Update README.md
* aa6f199 Update README.md
* 71115e6 Update README.md
* bc039d8 docs updated
|/
* d510d80 Update README.md
* 0b773a9 readme imgs
* 793dc98 nvm can already see ip address in earlier commits
* 325f03d value of ip address to gcp instance redacted
* 5a1cd2e created static ip address variable

```

CI (Continuous Integration)

- Feature/branch workflow model
- Overall, 11 new branches were created



Your branches

<code>dev</code>	Updated 11 minutes ago by qasidd	0 94
<code>feature/gcp</code>	Updated 10 hours ago by qasidd	0 76
<code>feature/code-smell-fixes</code>	Updated yesterday by qasidd	0 67
<code>feature/fixes-for-jar</code>	Updated yesterday by qasidd	0 63
<code>feature/utils</code>	Updated 2 days ago by qasidd	0 57
<code>feature/mvp-tests</code>	Updated 2 days ago by qasidd	0 52
<code>feature/calculate-cost</code>	Updated 3 days ago by qasidd	0 43
<code>feature/orders-orders_items</code>	Updated 3 days ago by qasidd	0 34
<code>feature/orders_items</code>	Updated 6 days ago by qasidd	0 22
<code>feature/orders</code>	Updated 7 days ago by qasidd	0 15
<code>feature/items</code>	Updated 7 days ago by qasidd	0 7



Testing

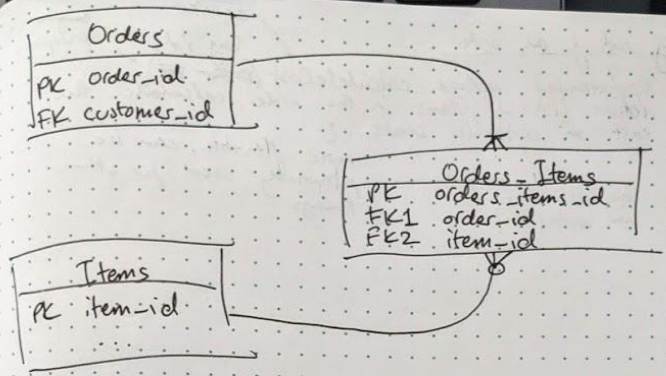
JUnit  



Demonstration

User Stories

- As a customer, I want to be able to add multiple items to my order, so that I can order all these items in one go
- As a customer, I want to know the overall cost of my order, so that I know how much I am spending
- As a customer, I want to be able to delete an item from an order, so that I can change my mind before placing an order
- As a customer, I want to be able to update my details, so that they are up-to-date
- As an administrator, I want to be able to add an item to the system, so that it will be available to customers
- As an administrator, I want to be able to delete an order from the system, so that it is not there any more when the order has been completed, or if the customer changes their mind about placing one
- As an administrator, I want to be able to delete a 'customer' from the system, so that if they wish their details not be stored in our system any longer, we can do that



order-id	customer-id	order-item-id	order-id	item-id
1	9238	1	1	6
2	9239	2	1	9
		3	2	2
		4	2	3
		5	2	9
		6	2	4

multiple records for orders with a single item but more than one quantity? > quantity field in orders-items

orders.update() M
 > add an item to an order
 > delete an item to an order M

orders.create() C
 > option to add items (trigger update()) after creating an order?

OrderItem DAO?
 Order DAO containing everything?

orders.read() M
 > read all orders (return orders)
 > read one order (return items in the order) M
 → include overall cost of order

orders.delete() M
 > delete all references to the M
 delete order as well in orders-items

Appropriate context information S
 (e.g. customers table when deleting a customer) displayed to make actions easier

Calculating cost of an order

Option 1: Dao extended method calculateCost (long id) ✓
 Option 2: return list of items in the order, calculate the cost in controller class. ?

too much logic in controller?

more flexible, can be potentially used for other things

easier long id ✓ single purpose



Sprint Review



Sprint Retrospective

- All user stories and tasks were completed within the time frame
- No major problems at any stage

Improvements for the future:

- Smaller, more frequent commits
- Create new branches when working on functionality unrelated to the current branch



Conclusion



Questions

