

IMS Project

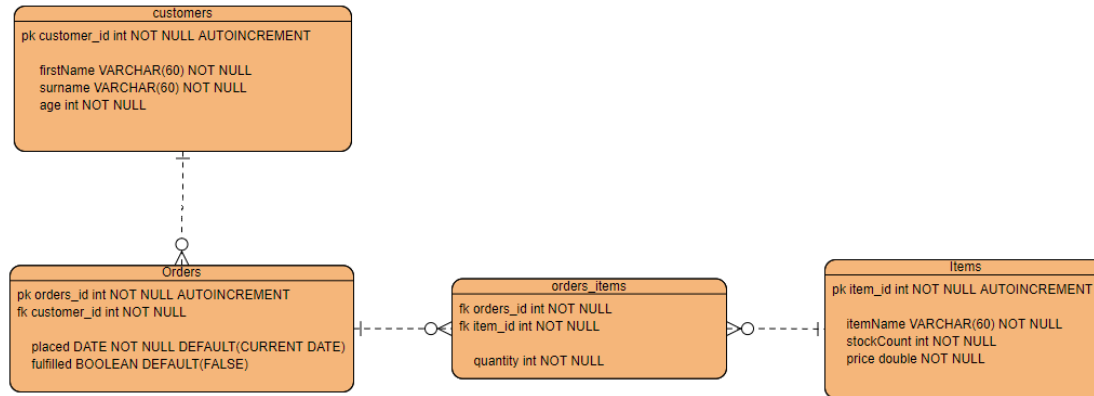
By Nathan Jackson

Technologies Used

- Java
- Maven
- SQL
- Cloud
- Testing
- Git
- Jira



Planning



- 4 Epics created throughout the project
- 20 User Stories
- 46 Tasks

- ERD designed at the start of the project
- Orders Items table used to avoid many to many relationship

IM-55

Covering all tasks which involve handling and validating input data

IM-56

As a user, I want to only access the system when my login is successful, so that I don't perform redundant tasks

IM-57

When trying to login, if successful proceed to main menu, otherwise display message and ask for new login details

IM-58

As a user, I want to be able to either read single entries from a table, or get all entries

IM-59

Edit domain class to branch into generic base function, rather than read all

IM-60

Add read and read by ID methods to CRUD controller and DAO interfaces

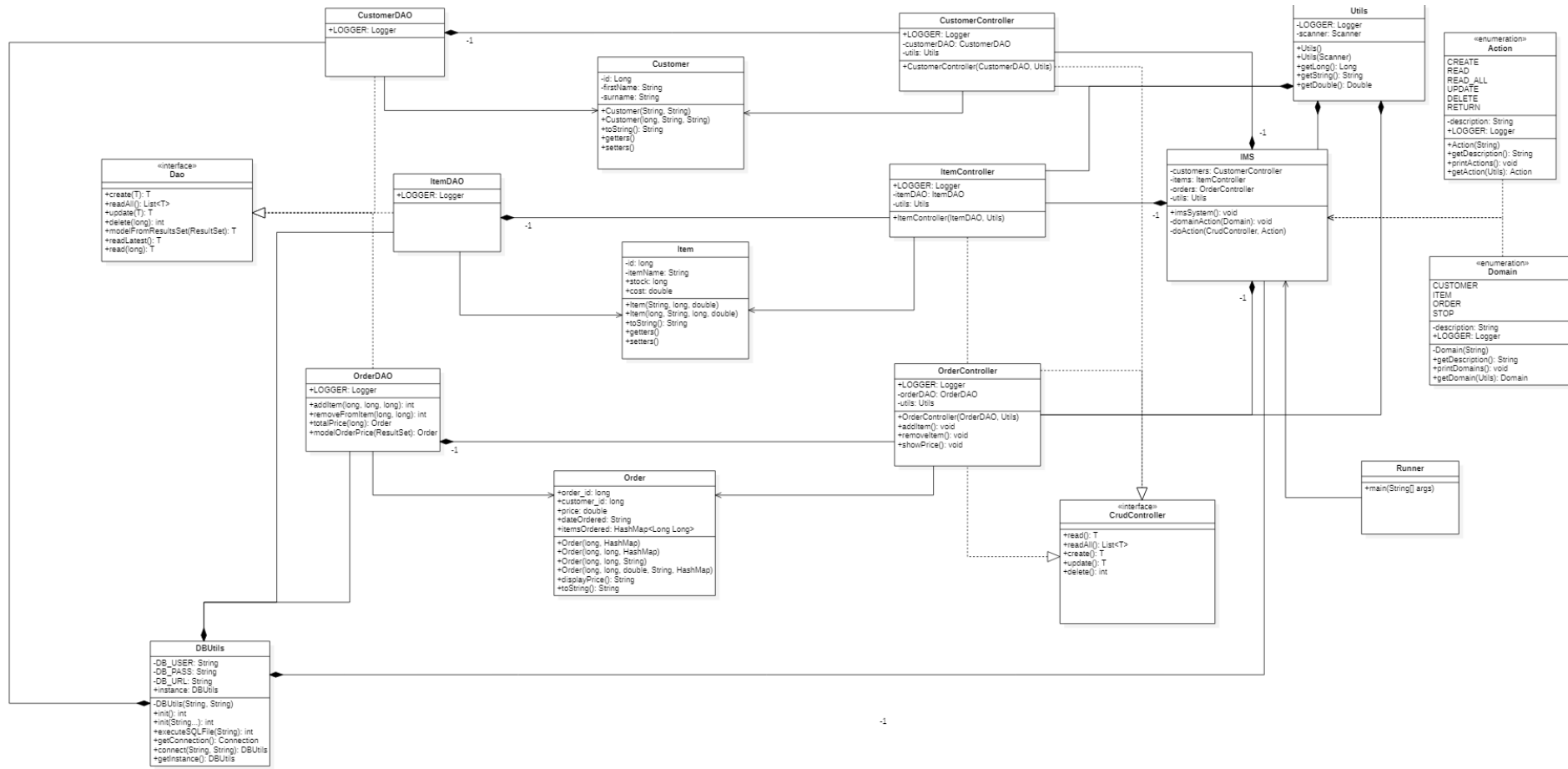
IM-61

Create submenu in each controller's read function to ask for input on what they'd like to read

IM-62

Ensure the DAO can handle and format the queries for read by ID, and read all

Planning – Class Diagram



Version Control

- Branches for each key feature that affects multiple classes, e.g. Customers (Customer, CustomerController, CustomerDAO)
- Testing had it's own branches which was redundant and was result of not testing as I went
- Pushes were done after achieving every milestone in the sprint

dev Updated 5 hours ago by Nj147

testing-Cleanup Updated yesterday by Nj147

testing-DomainClasses Updated yesterday by Nj147

testing-DAOs Updated 2 days ago by Nj147

testing-Controllers Updated 2 days ago by Nj147

feature-inputValidation Updated 3 days ago by Nj147

feature-readChanges Updated 20 days ago by Nj147























feature-orders Updated 21 days ago by Nj147

feature-items Updated 22 days ago by Nj147

feature-sqlCreators Updated 22 days ago by Nj147

Testing

- 82.4% Coverage across the whole project
- All Controllers, DAOs and Domain classes tested to 100%

IMS-Project		91.3 %	5,522	523	6,045
src/main/java		82.4 %	2,453	523	2,976
com.qa.ims		0.0 %	0	206	206
IMS.java		0.0 %	0	190	190
Runner.java		0.0 %	0	16	16
com.qa.ims.controller		84.2 %	693	130	823
Action.java		0.0 %	0	130	130
CustomerController.java		100.0 %	162	0	162
ItemController.java		100.0 %	176	0	176
OrderController.java		100.0 %	355	0	355
com.qa.ims.persistence.domain		86.9 %	698	105	803
Domain.java		0.0 %	0	105	105
Customer.java		100.0 %	175	0	175
Item.java		100.0 %	202	0	202
Order.java		100.0 %	321	0	321
com.qa.ims.utils		69.1 %	161	72	233
Utils.java		4.8 %	3	59	62
DBUtils.java		92.4 %	158	13	171
com.qa.ims.persistence.dao		98.9 %	901	10	911
OrderDAO.java		97.4 %	375	10	385
CustomerDAO.java		100.0 %	253	0	253
ItemDAO.java		100.0 %	273	0	273

Testing Problem

- Due to the Order class feeding 2 different tables in the database, many constructors were used, with different variables being assigned and being left null
- This meant in the testing phase, comparing the returned object wasn't recognised as being the same due to different objects being created in the DAO than were sent there

Testing Solution

- Due to Order structure, Date Ordered variable was assigned in the DB but in the create method
- This meant the test would always fail

```
@Test
public void testCreate() {
    final long custID = 1L;
    HashMap<Long,Long> items = new HashMap<Long,Long>();
    items.put(2L, 100L);
    items.put(3L, 41L);
    final Order created = new Order(6L, custID, items);

    assertEquals(created, dao.create(created));
}
```

```
java.lang.AssertionError: expected:<Order ID: 6, Customer ID: 1, Date Ordered: null> |
but was:<Order ID: 6, Customer ID: 1, Date Ordered: 2021-01-07 17:59:05>
```

```
@Test
public void testCreate() {
    final long custID = 1L;
    HashMap<Long,Long> items = new HashMap<Long,Long>();
    items.put(2L, 100L);
    items.put(3L, 41L);
    final Order created = new Order(6L, custID, items);

    assertEquals(created.getOrder_id(), dao.create(created).getOrder_id());
    assertEquals(created.getCustomer_id(), dao.create(created).getCustomer_id());
}
```

- So instead just have to compare the variables that are received back rather than whole object
- Similar issue occurred in the totalPrice() test

Demonstration

Sprint Review

What Got Done

- All Functional Requirements from specification met
- User input and login validation complete (MoSCoW Should have)
- Testing complete to the required standard

What Missed The Cut

- More comprehensive testing, on IMS specifically
- Some stretch goals I had set
- Restrict orders to only be valid when stock is available
- Handle Multiple logins and/or multiple databases being accessible

Sprint Retrospective

What Went Well

- Always knew what to work on next
- No big unexpected errors occurred
- Version Control
- Time Management

What Could Be Improved

- Test as I go
- Integrate Jira and GitHub
- More detail added in user stories (Story Points, Acceptance Criteria)
- Reduce redundant branches (testing branches)

Conclusions

- Overall development side of things went very well
- Main mistakes were in approach rather than the code
- Going forward would look to refine and improve planning and management techniques