

Assignment-i Submission report

Enterprise Application Development (425)



Gunathilaka D. D. T. M. (IT13011130)

August 14, 2016

SLIIT

The following report describes the implementation of “Order Management” Application which provides the solution for the first assignment of Enterprise Application Development module

# Declaration

I hear by declare that following task is my own creation and it does not violate any constraints of the assignment.

Gunathilaka D. D. T. M. (IT13011130)

Contents

[Declaration 1](#_Toc458957581)

[List of Figures 3](#_Toc458957582)

[Introduction 4](#_Toc458957583)

[Methodology 5](#_Toc458957584)

[Design & Technologies 5](#_Toc458957585)

[Implementation 7](#_Toc458957586)

[Synchronous & asynchronous call usage 7](#_Toc458957587)

[Data access and table structure 7](#_Toc458957588)

[Discussion 8](#_Toc458957589)

[Technical 8](#_Toc458957590)

[Standards 8](#_Toc458957591)

[Sample Screens 8](#_Toc458957592)

# List of Figures

[Figure 1 Use case Diagram of Order Management 4](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957593)

[Figure 2 Tiers and respective classes 5](#_Toc458957594)

[Figure 3 Class diagram of Order Management 6](#_Toc458957595)

[Figure 4 Order Management Table Structure 7](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957596)

[Figure 5 Order Management Home Page 8](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957597)

[Figure 6 View Customer I 9](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957598)

[Figure 7 Create a Customer 9](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957599)

[Figure 8 View Customer II 10](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957600)

[Figure 9 Delete a Customer 10](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957601)

[Figure 10 All Customers I 11](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957602)

[Figure 11 All Customers II 11](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957603)

[Figure 12 Edit Customer 12](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957604)

[Figure 13 All Orders 12](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957605)

[Figure 14 View Order 13](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957606)

[Figure 15 Edit Order 13](file:///C:\Users\Research\Desktop\EAD\Assignment.docx#_Toc458957607)

[Figure 16 Create Order 14](#_Toc458957608)

# Introduction

Java Enterprise Application is created using Java Enterprise Edition (JAVAEE). This application is capable of managing Customers and Orders within the system. The implemented solution is named as “Order Management” and below use case diagram illustrates the functional requirements of the solution.

Figure Use case Diagram of Order Management

System User

Order Management

# Methodology

## Design & Technologies

The application developed following three tier architecture, where it consists with data tire, business & logic tire and presentation tiers. Furthermore solution is deployed as two components namely “EJB” module and “Web” module. In “EJB” module it contains the data-access and business logics while presentation layer resides inside the “Web” module.

In order to facilitate client interaction “Java Servlets” are used in the presentation layer. These servlets are hosted in a Glass Fish server. The business logics are called within these servlets to invoke data access methods. Furthermore data access methods are implemented using Java persistence while Java DB is used as the data storage.

|  |  |
| --- | --- |
| Layer | Classes |
| Data Access Tier | CustomerManagementEntity.Customer  OrderManagementEntity.CustomerOrder |
| Business Logic Tier | SessionFacadeDAO. AbstractFacade  SessionFacadeDAO. CustomerFacade  SessionFacadeDAO. CustomerOrderFacade  Asynchronous. CreateOrderMDB |
| Presentation Tier | Servlet. AllCustomers  Servlet. AllOrders  Servlet. CreateCustomer  Servlet. CreateOrder  Servlet. DeleteCustomer  Servlet. HomePage  Servlet. ManageCustomer  Servlet. ManageOrder  Servlet. ViewCustomer  Servlet. ViewOrder  HTMLCreation. CheckAndValidate  HTMLCreation. HTML  HTMLCreation. POPUP |

Figure Tiers and respective classes

The Following UML Class Diagram describes the organization of classes in the implemented solution.

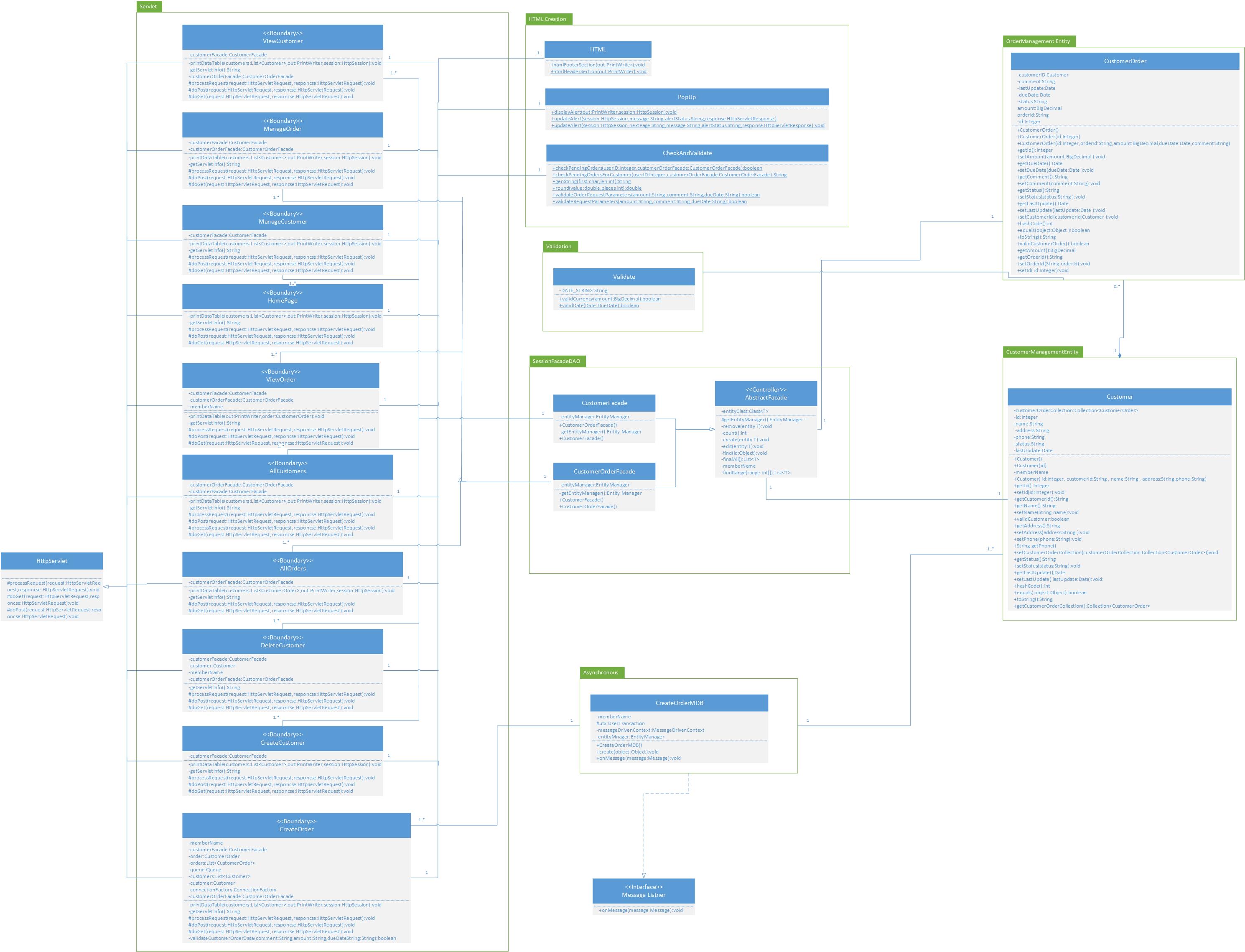


Figure Class diagram of Order Management

# Implementation

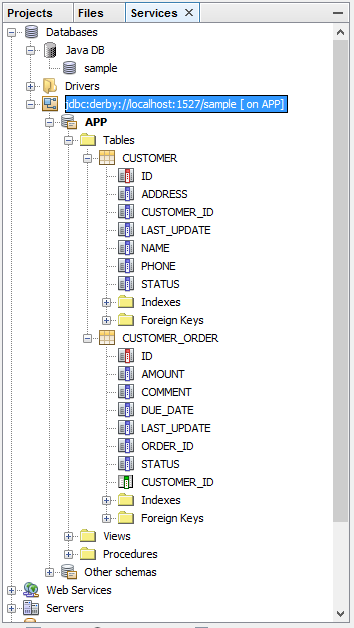
## Synchronous & asynchronous call usage

The new orders are handled asynchronously using Message Driven Beans (MDB).Object Messages are used to pass “CustomerOrder “Entity Bean from presentation tier to business logic tier. The created Object Messages which contain “CustomerOrder” Entity Beans are pushed into a Session Queue which carries respective Object Message to the business logic tier.

In business logic tier the implemented Message Driven Bean instance listens to incoming Object Messages via Session Queue. Once a new Object Message arrives it executes “onMessage (…)“event where it grabs the content of it. This process happens asynchronously due to default behavior of Message Driven Beans.

In this solution the requirement is to save incoming “Order” Entity Beans which arrive via Session Queue as Object Messages. In order to save Entity Bean to database,” Entity Manager” instance is created. This instance consists of” Persist ()” method which allows to save incoming “Order” Entity Bean in database.

As the conclusion Message Driven Bean listens & captures incoming “Order” Entity Beans and save those in database using “Persist” method. This entire process executes asynchronously due to default behavior of Message Driven Beans.

All the other data access logics except creating new order are invoked synchronously by invoking methods in Façade classes. These Façade classes are the Session Beans which manipulate “Customer” and “CustomerOrder “Entity Beans at runtime.

# Data access and table structure

In the database there are 2 tables “CUSTOMER” and “CUSTOMER \_ORDER” to represent Entity Beans. The “CUSTOMER \_ORDER” table refers “id” from “Customer” table.

Tables are mapped with entity classes and also javax.persistance annotations are used to indicate primary and foreign keys. The @id annotation allows creating primary keys in tables. Foreign keys and cardinality ratio are annotated using @JoinColum and @ManyToOne respectively.

Figure Order Management Table Structure

# Discussion

## Technical

* Application is implemented using JavaEE version 6
* The solution compiles and deployed using JDK 1.7.0\_79
* GlassFish server version 3.1.0.2 is used to deploy the application

## Standards

* The “[Google's Java coding standards](http://www.javaworld.com/article/2102551/java-language/googles-java-coding-standards.html)” has followed as the coding standard throughout the solution implementation

## Sample Screens

|  |
| --- |
| Figure Order Management Home Page |
|  |
| Figure View Customer I  Figure Create a Customer |
| Figure View Customer II |
| Figure Delete a Customer |
| Figure All Customers I |
| Figure All Customers II |
| Figure Edit Customer |
| Figure All Orders |
| Figure View Order |
| Figure Edit Order |
| Figure Create Order |