Java based Flight Reservation System Project

Assignment 3

Regis University

MSSE670 – Java Software Development

Spring 2020 Week 3

03/29/2020

Brenda Palmer

Table of Contents

[Introduction 3](#_Toc36299891)

[Purpose 3](#_Toc36299892)

[Scope 3](#_Toc36299893)

[Methodology 3](#_Toc36299894)

[Software Architecture 3](#_Toc36299895)

[Eclipse Images 5](#_Toc36299896)

[Eclipse images of Service Layer Interfaces, Implementation classes and Factory Class 5](#_Toc36299897)

[Eclipse imager of Service Layer JUnit Tests 7](#_Toc36299898)

[What I Learned 8](#_Toc36299899)

# Introduction

The Service Layer is used to take objects from the Domain Layer as inputs of the application where it moves data to and from the database and/or moving data to and from other applications. This week’s assignment begins to focus on the Service Layer implementation which builds upon the Domain Layer.

## Purpose

The purpose of this assignment is to learn how the Service Layer works, interacts with the Domain and Business Layers and then how to implement this knowledge into Java code.

## Scope

This week’s assignment is focused only on creating the factory class, interface classes, implementation classes and corresponding tests by just creating stubs in the implementation classes.

## Methodology

The methodology used for creating the Service Layer is UML.

# Software Architecture

The Service Layer was created in Eclipse and the following classes where created:

**Interface Classes:** ICustomerAccountService,, ILoginSerive, ISearchFlightInformationService, IListAvailableItineraryOptionsServie, IReserveItinearyServie and IBookItineraryService.

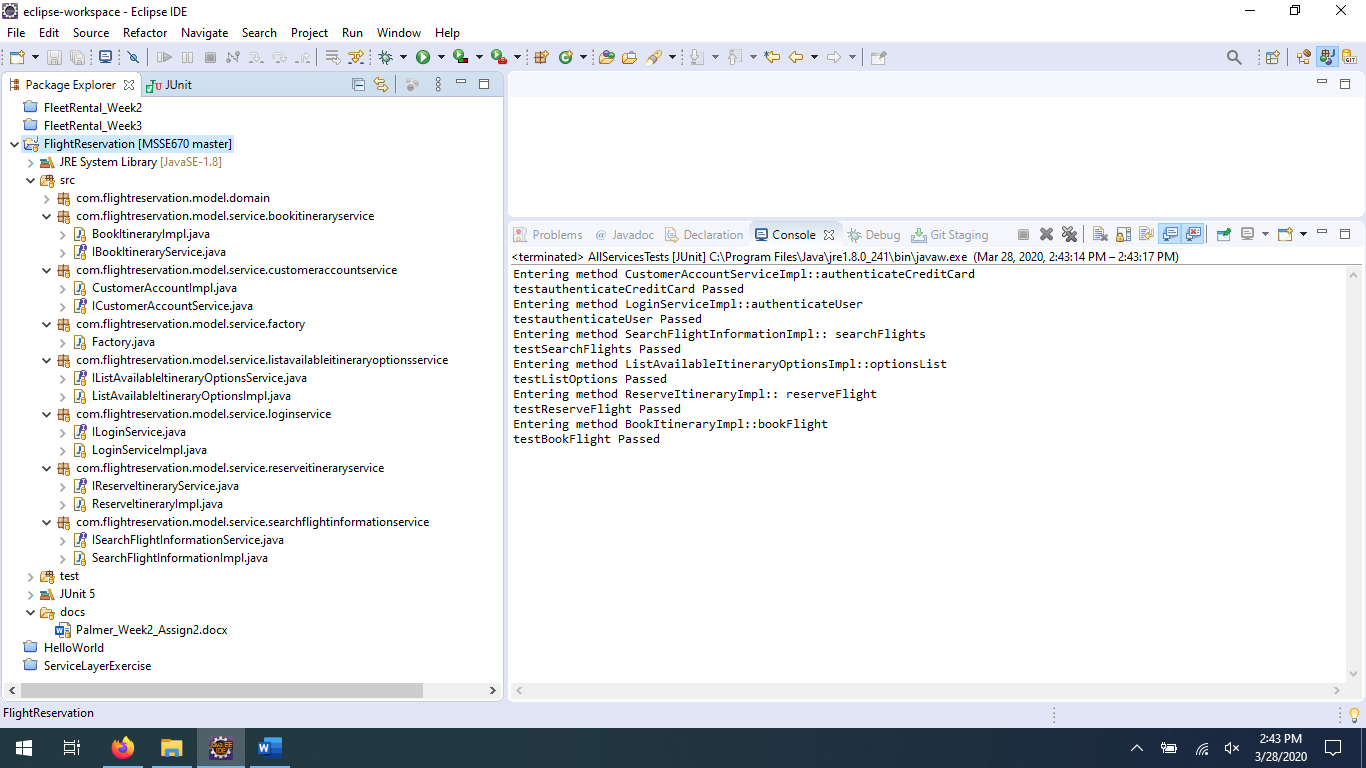
**Implementation Classes:** CustomerAccountImpl, LoginImpl, SearchFlightInformationImpl, ListAvailableItineraryOptionsImpl, ReserveItineraryImpl and BookItineraryImpl.

**Junit Test Classes:** CustomerAccountServiceTest, LoginServiceTest, SearchFlightInformationServiceTest, ListAvailableItineraryOptionsServiceTest, ReserveItineraryServiceTest, BookItineraryServiceTest, FactoryServiceTest and AllServicesTest .

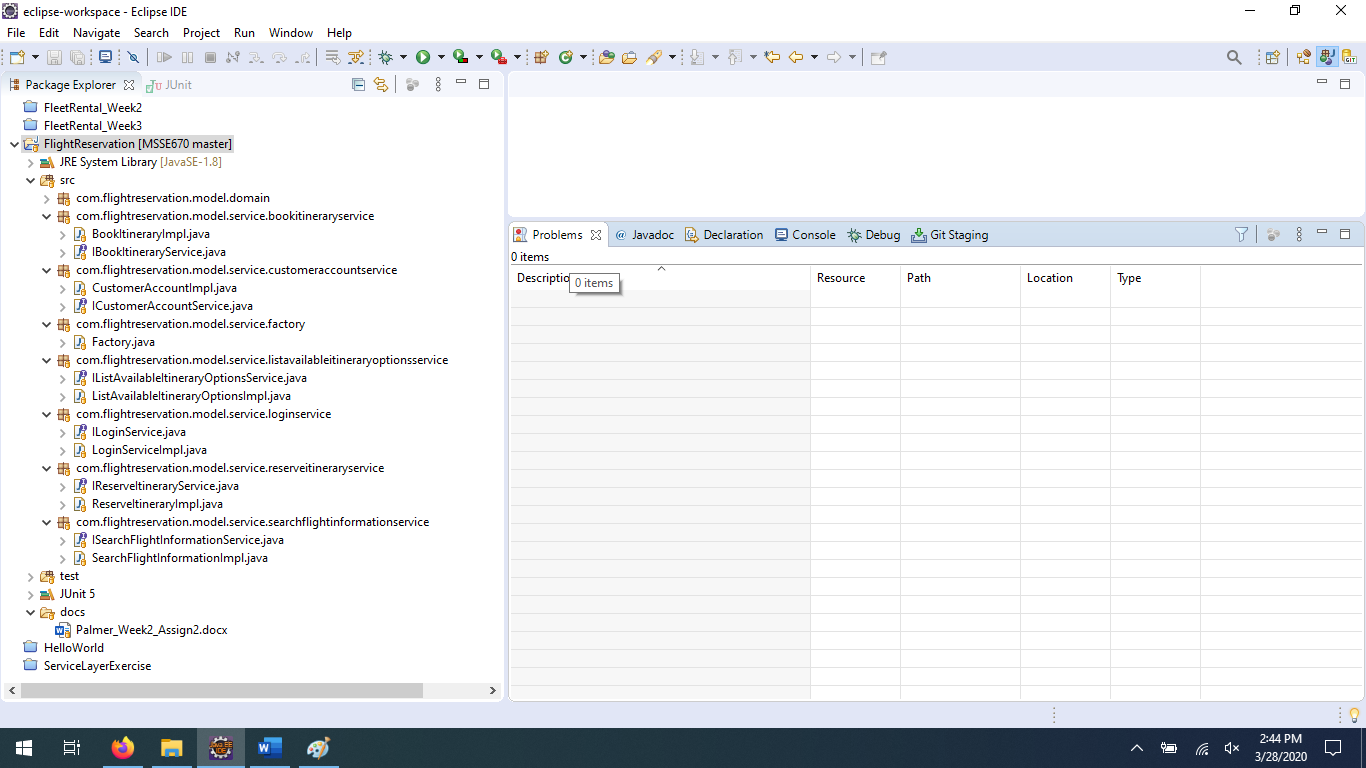
**Factory Class:** Factory

# Eclipse Images

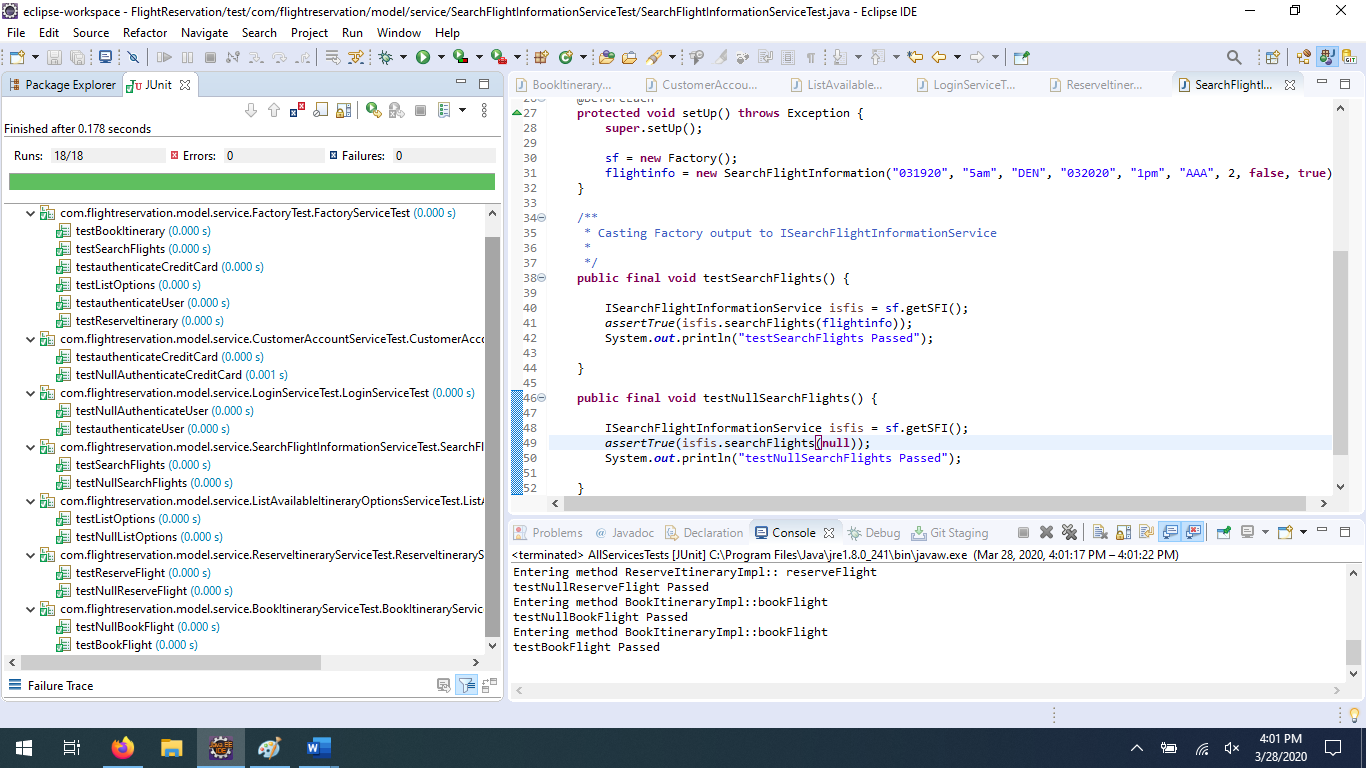
## Eclipse images of Service Layer Interfaces, Implementation classes and Factory Class



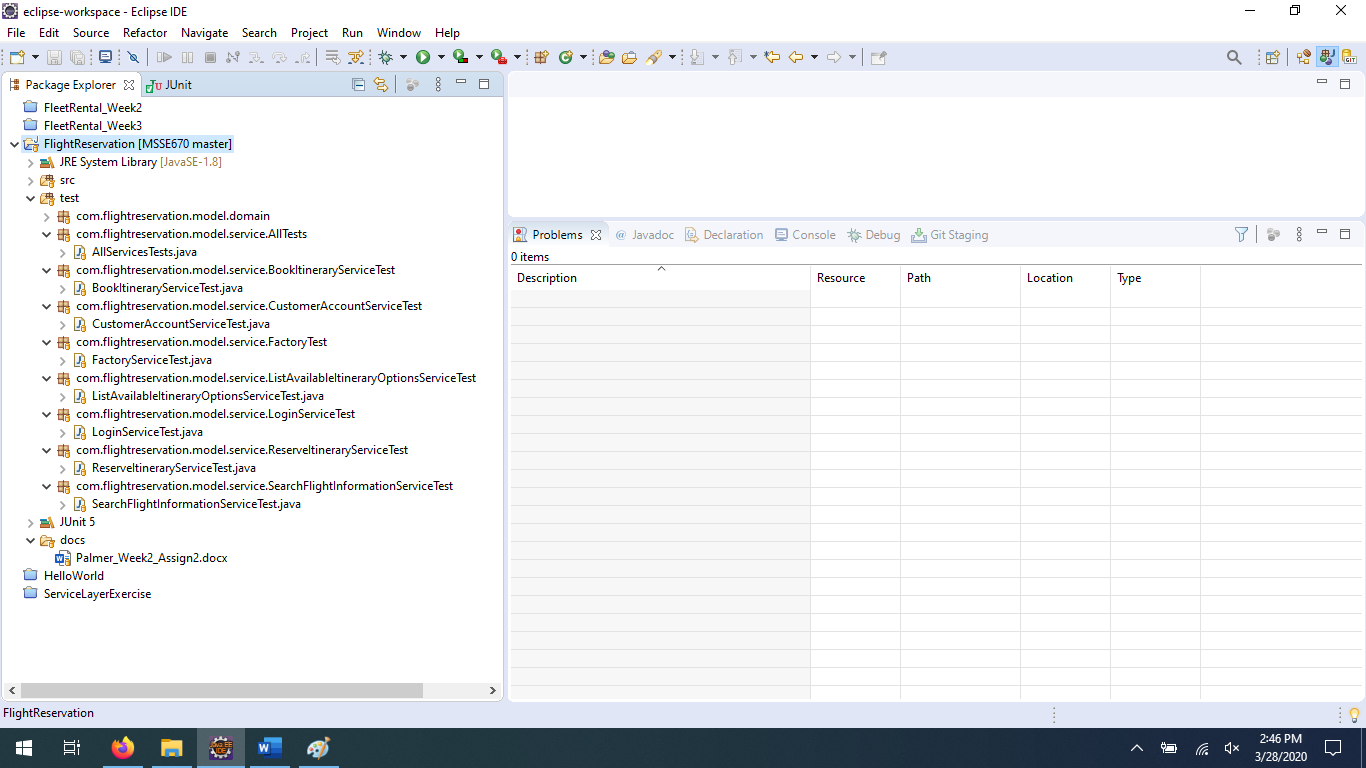
Problems tab:



## Eclipse imager of Service Layer JUnit Tests



Problems Tab:



# What I Learned

Architecture is a new concept to me, so everything moving forward is all new to me. I learned that the Service Layer is what moves data in and out of the database and other applications from the domain layer. Also, how the business layer knows about the Service Layers Interface but is unaware of the implementation classes and their methods. Also learned through coding this week’s exercise that the implementation class can be returned to an interface, which I never knew was possible. For the JUnit tests I wasn’t sure how the tests I created for the implementation classes would run, but once I created the AllServicesTest class I was able to call these tests and run them without using the @Test, like that of the Factory JUnit test class. Overall, I am pleased with what I learned and look forward to what I will learn in week 4.