Java based Flight Reservation System Project

Assignment 4

Regis University

MSSE670 – Java Software Development

Spring 2020 Week 4

04/05/2020

Brenda Palmer

Table of Contents

[Introduction 3](#_Toc37009495)

[Purpose 3](#_Toc37009496)

[Scope 3](#_Toc37009497)

[Methodology 3](#_Toc37009498)

[Software Architecture 3](#_Toc37009499)

[Eclipse Images 5](#_Toc37009500)

[Eclipse images of Service Layer Classes, Exception Classes and Factory Class 5](#_Toc37009501)

[Eclipse imager of Service Layer Part 2 JUnit Passing Tests 6](#_Toc37009502)

[What I Learned 7](#_Toc37009503)

# 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 to use self-created exceptions to then declare them in the interface and implementation and then to create corresponding JUnit tests. As well as serializing and deserializing objects to files.

## Scope

This week’s assignment is focused only on creating the exception classes, marker interface, decoupling the Factory class and creating corresponding tests for the factory and implementation classes.

## Methodology

The methodology used for creating the Service Layer Part 2 is UML.

# Software Architecture

The Service Layer Part 2 was created in Eclipse and the following classes where created:

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

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

**Junit Test Classes:** CustomerAccountServiceTest, LoginServiceTest, SearchFlightInformationServiceTest, ListAvailableItineraryOptionsServiceTest, ReserveItineraryServiceTest, BookItineraryServiceTest, FactoryServiceTest, AllServicesTest, CustomerAccountImplTest, LoginImplTest, SearchFlightInformationImplTest, ListAvailableItineraryOptionsImplTest, ReserveItineraryImplTest and BookItineraryImplTest.

**Factory Class:** Factory

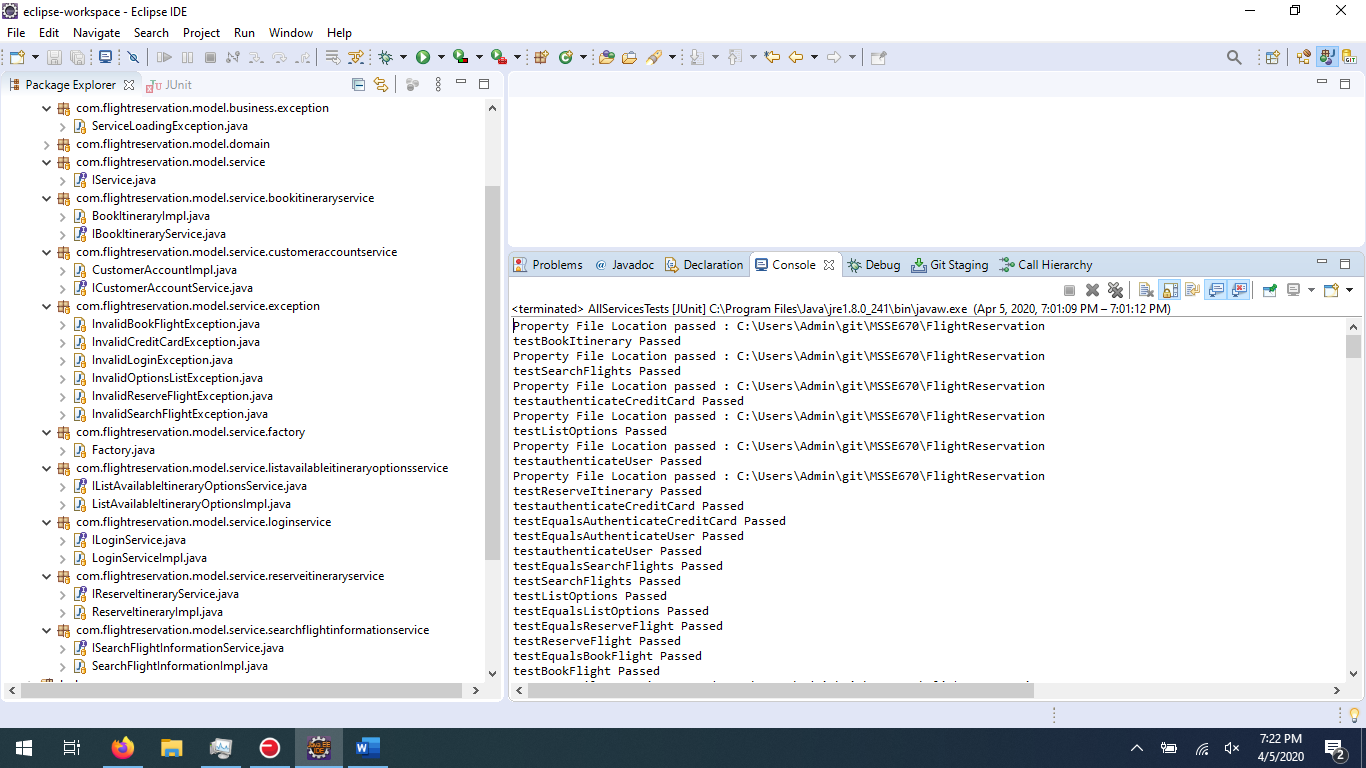
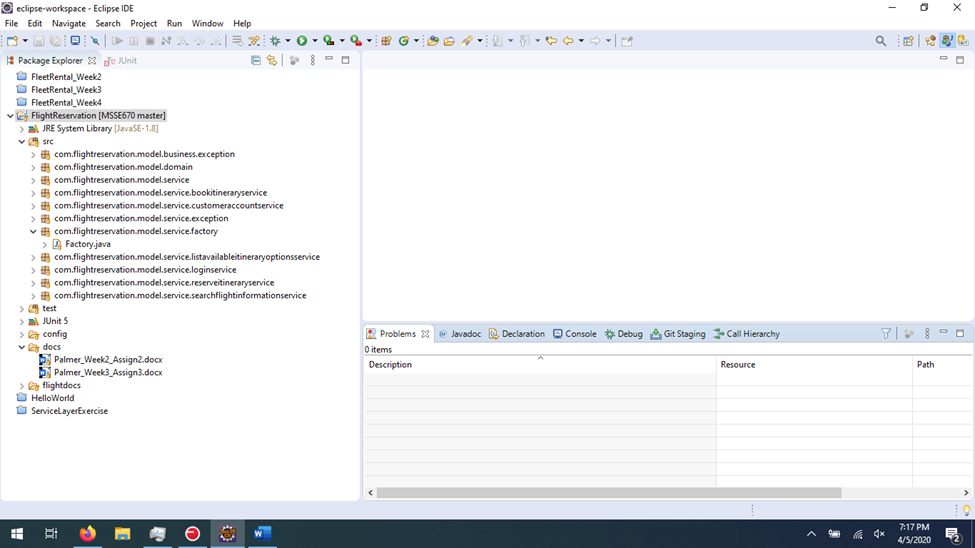
**Exception Classes:** InvalidCreditCardException, InvalidLoginException, InvalidSearchFlightException, InvalidOptionsListException, InvalidReserveFlightException, InvalidBookFlightException and ServiceLoadingException.

**FlightDocs:** CreditCardInformation.out, Login.out, SearchFlightInformation.out, OptionsList.out, ReserveFlight.out and BookFlight.out.

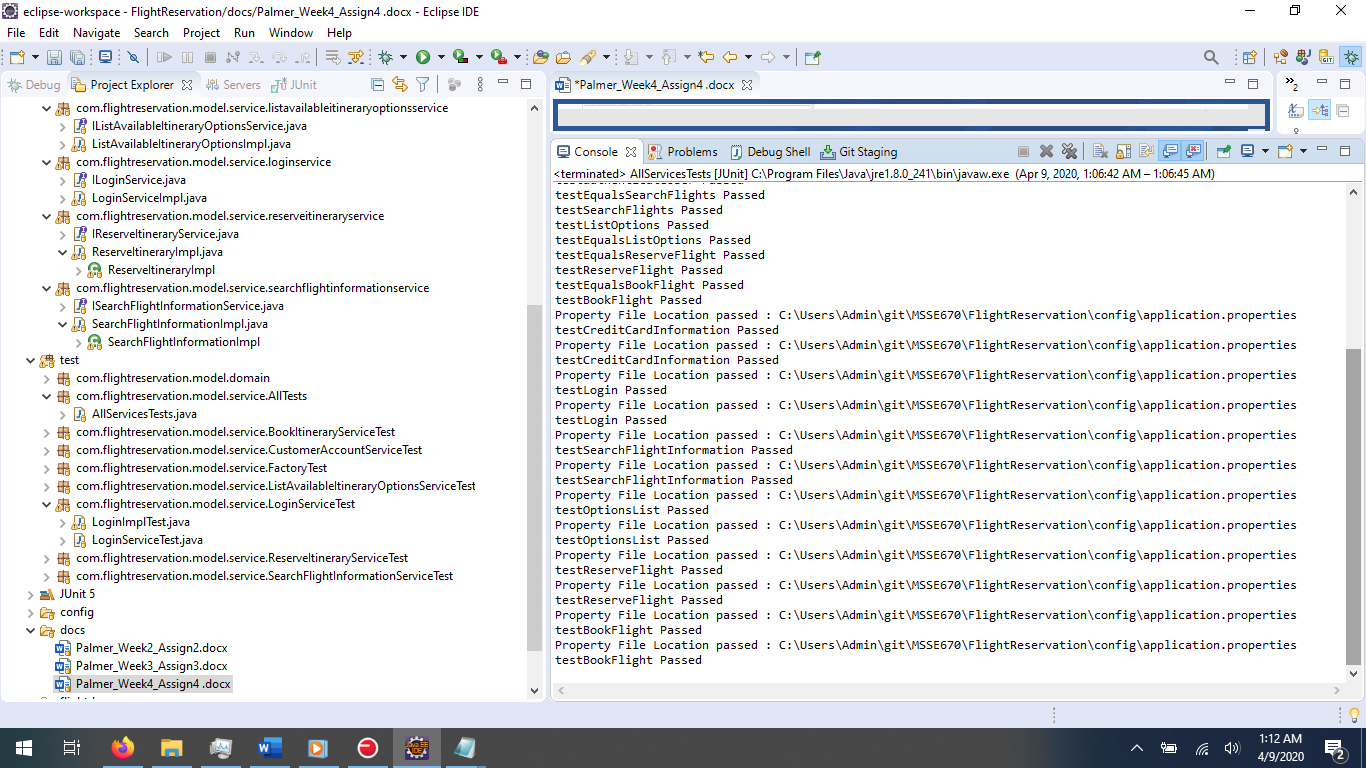
**Config:** application.properties

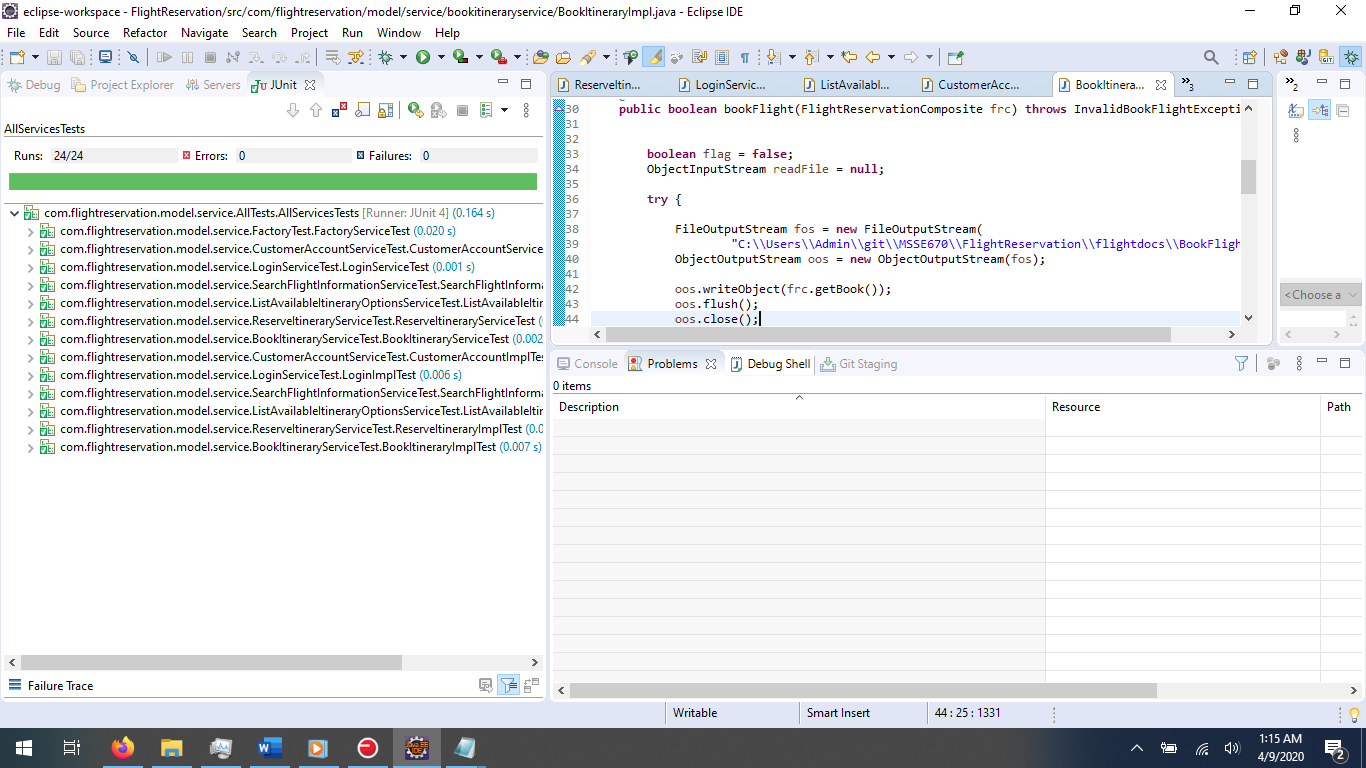
# Eclipse Images

## Eclipse images of Service Layer Classes, Exception Classes and Factory Class

Problems tab:

## Eclipse imager of Service Layer Part 2 JUnit Passing Tests



Problems Tab: 

# What I Learned

I learned how to implement exceptions as well as how to create my own exceptions to then implement them. In all of my experience I never heard of a marker interface, so I was able to use this new concept and see how it works in my code. One issue I am having is with an exception being thrown saying error accessing file. Not sure why this is occurring. I wonder if I need to also create a ObjectOutputStream and not just an ObjectInputStream. Going to try this and if I still have time maybe I will get it resolved. This was a challenging assignment, but I do feel I learned a lot.