California State University Fullerton

CPSC 462



Object Oriented Software Design

Design Model

for the



Tuffy Flights

System

|  |  |  |
| --- | --- | --- |
| **Nathan Marcos** | **Jared Castaneda** | **Jawad Swed** |
| Design Modeling, Implementation | Project Management, Implementation | Business Modeling, Design Modeling |
| [nathanmarcos@csu.fullerton.edu](mailto:nathanmarcos@csu.fullerton.edu) | [jaredcast@csu.fullerton.edu](mailto:jaredcast@csu.fullerton.edu) | [jawadswed@csu.fullerton.edu](mailto:jawadswed@csu.fullerton.edu) |

Revision History:

| Version | Date | Summary of Changes | Author |
| --- | --- | --- | --- |
| 1.0 | 9/28/2020 | * Initial Release | Nathan Marcos  Jared Castaneda  Jawad Swed |
| 1.1 | 11/8/2020 | * Added the story User Login and Edit Account Information * Removed User Login and Edit Account Information story * Added Book Flight, Search Flight, and Show Ticket stories and their software class diagrams (static and dynamic views) * Added descriptions for the new stories * Added all SSD traceability | Nathan Marcos  Jared Castaneda  Jawad Swed |
| 1.2 | 12/5/2020 | * Removed old design model, static and dynamic views reworked * Added design model and dynamic views for all headers * Edited user roles | Nathan Marcos  Jared Castaneda  Jawad Swed |

Table of Contents

[1 Static View 1](#_Toc58181269)

[1.1 Book Flight 1](#_Toc58181277)

[1.1.1 Software Class Diagram 1](#_Toc58181278)

[1.1.2 Description 1](#_Toc58181279)

[1.2 Authenticate Customer 2](#_Toc58181280)

[1.2.1 Software Class Diagram 2](#_Toc58181281)

[1.2.2 Description 2](#_Toc58181282)

[1.3 Show Ticket 3](#_Toc58181284)

[Software Class Diagram 3](#_Toc58181285)

[1.3.1 3](#_Toc58181286)

[1.3.2 Description 3](#_Toc58181287)

[1.4 Get Meal 4](#_Toc58181288)

[1.4.1 Software Class Diagram 4](#_Toc58181289)

[1.4.2 Description 4](#_Toc58181290)

[2 Dynamic View 5](#_Toc58181291)

[2.1 Add Payment Sequence of Execution 5](#_Toc58181301)

[2.1.1 Software Interaction Diagram 5](#_Toc58181302)

[2.1.2 Description 5](#_Toc58181303)

[2.1.3 SSD Traceability 5](#_Toc58181304)

[2.2 Get Meal Sequence of Execution 6](#_Toc58181305)

[2.2.1 Software Interaction Diagram 6](#_Toc58181306)

[2.2.2 Description 6](#_Toc58181307)

[2.2.3 SSD Traceability 6](#_Toc58181308)

[2.3 Get Ticket Sequence of Execution 6](#_Toc58181309)

[2.3.1 Software Interaction Diagram 6](#_Toc58181310)

[2.3.2 Description 7](#_Toc58181311)

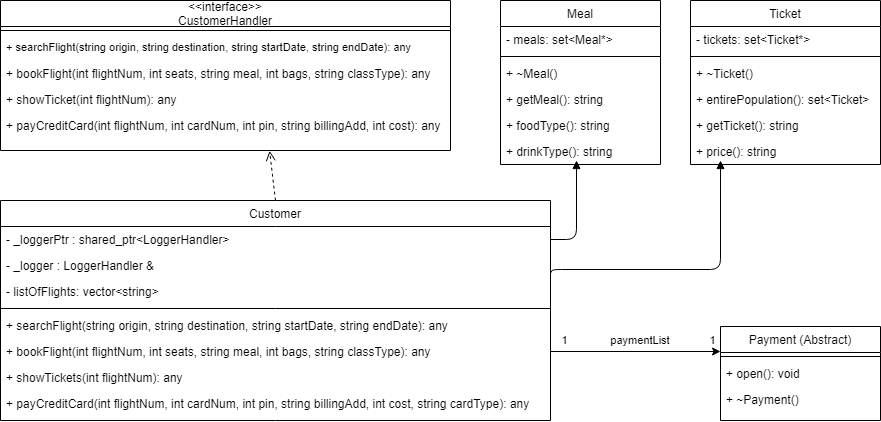
[2.3.3 SSD Traceability 7](#_Toc58181312)

**Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.Error! Hyperlink reference not valid.**

# Static View

## Book Flight

### Software Class Diagram

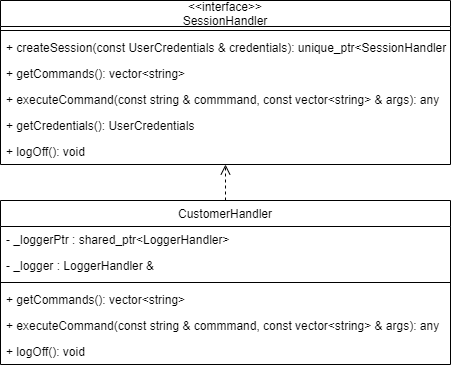


### Description

The Customer is a domain implementation that has access to the other classes in the domain, Meal and Ticket, and the technical service class Payment. A customer calls upon the Payment, Meal, and Ticket classes through various functions. A payment can be created by the customer depending on their meal and ticket chosen. The Meal and Ticket classes are polymorphic classes that can return specific attributes.

## Authenticate Customer

### Software Class Diagram

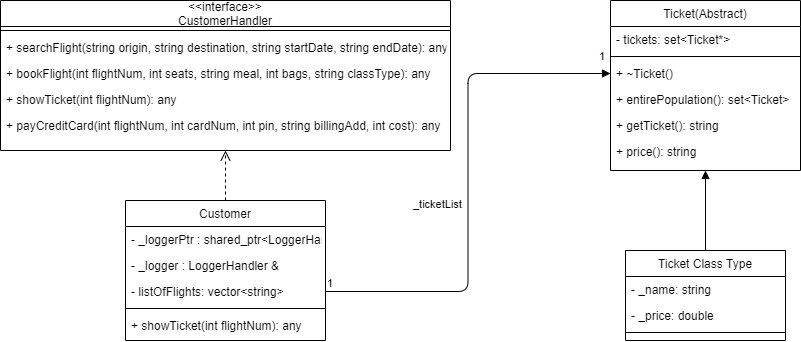


### Description

The SessionHandler is able to create Customer sessions after the user requests to create a customer session. The Customer session has the option to now use its corresponding commands and to log off.

## Show Ticket

### Software Class Diagram

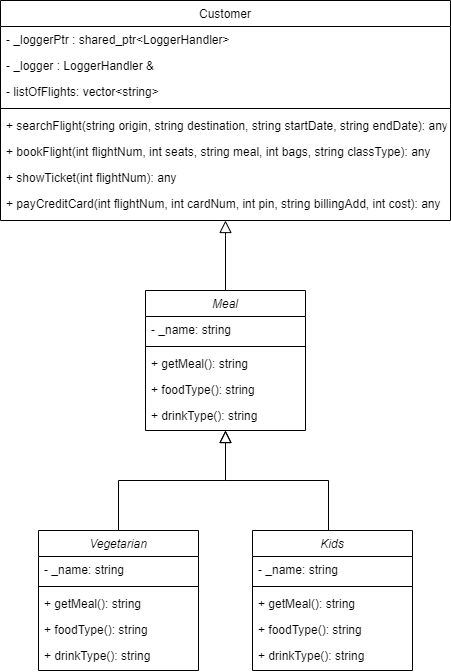


### Description

The Customer class has access to the polymorphic Ticket class. The Ticket holds the different types of tickets, such as First Class and Business Class, and can be called when the customer wants to view the tickets that they purchased. Tickets are saved into a list.

## Get Meal

### Software Class Diagram



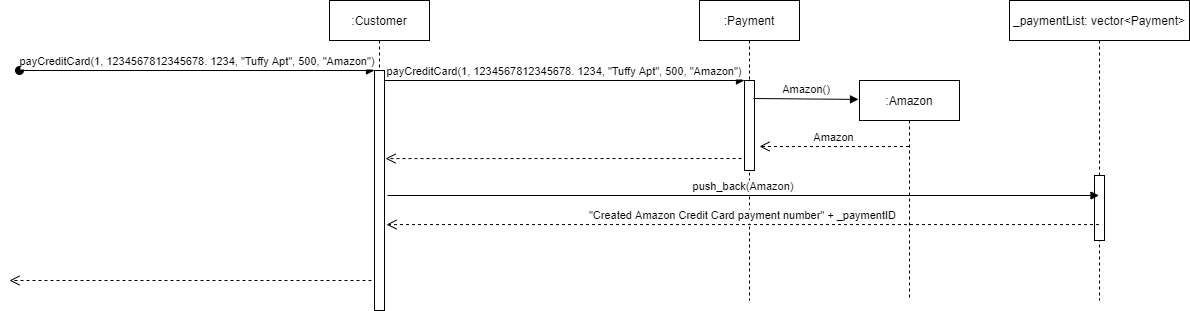
### Description

The Customer class has access to the polymorphic Meal class. The Meal holds the different types of meals, such as Vegetarian and Kids, and can be called when the customer wants to book a flight.

# Dynamic View

## Add Payment Sequence of Execution

### Software Interaction Diagram



### Description

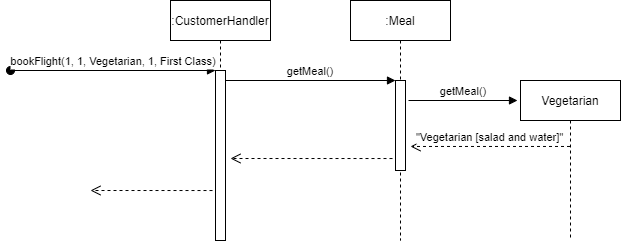
The CustomerHandler received a message of payCreditCard with the parameter Amazon parameter. The SessionHandler calls Amazon() from the Payment abstract class Amazon. PaymentHandler receives the Amazon details and sends it to CustomerHandler. CustomerHandler adds the payment to the \_paymentList vector, which then responds with a confirmation.

### SSD Traceability

This message can be found in the Book Flight use cases.

## Get Meal Sequence of Execution

### Software Interaction Diagram



### Description

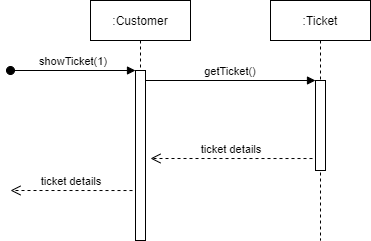
A bookFlight call with the parameters 1, 1, Vegetarian, 1, First Class is sent to the CustomerHandler. The CustomerHandler accesses Meal to retrieve the vegetarian details from the Vegetarian polymorphic class.

### SSD Traceability

This request can be found in the Book Flight use case.

## Get Ticket Sequence of Execution

### Software Interaction Diagram



### Description

The showTicket call with the parameters 1 is sent to the CustomerHandler. The CustomerHandler accesses Ticket to retrieve the ticket details from the Ticket polymorphic class based on a previously booked flight.

### SSD Traceability

This request can be found in the Retrieve Ticket to Board Flight use case.