Flight Management System

Domain Model Description

Version 1.0

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 20/Apr/2015 | 1.0 | Created the domain model description document | Michael Kong |

Table of Contents

1. Introduction 4

1.1 Brief Description 4

2. Domain Model Description 4

2.1 User 4

2.2 Normal Staff 4

2.3 Abstract User 4

2.4 System Administrator 5

2.5 Service System Manager 5

2.6 Reporting System Manager 5

2.7 Profile System Manager 5

2.8 Reservation System Manager 5

2.9 Flight Manager 6

2.10 Customer 6

2.11 Travel Agency 6

2.12 Travel Agency 6

2.13 Report Builder 6

2.14 Ticket Price 7

2.15 Service 7

2.16 Booking 7

2.17 Person 7

2.18 Flight 7

2.19 Airport 7

2.20 Route 8

2.21 Seat 8

Domain Model Description

# Introduction

## Brief Description

This document covers all the classes that are involved in the Flight Management System. For each class that exists in this domain model, the description of it shall be as follows:

1. Class Name
2. Short Description
3. List of Attributes
4. List of Operations
   1. For each operation
      1. Target class
      2. Return value
      3. Brief description

The goal of this document is to assist in understanding the relationships between the various classes within the system and the messages passed between them.

# Domain Model Description

## Class: User

|  |
| --- |
| **Class Name:** User |
| **Superclasses:** None |
| **Short Description:** This class defines the basic attributes and operations that all users of the system must have. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Normal Staff

|  |
| --- |
| **Class Name:** Normal Staff |
| **Superclasses:** User |
| **Short Description:** This class defines common functionalities for all the managers and normal staff. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Abstract Customer

|  |
| --- |
| **Class Name:** Abstract Customer |
| **Superclasses:** User |
| **Short Description:** This abstract class defines the basic attributes and operations that the Customer and Travel Agency classes will have. This class cannot be instantiated. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: System Administrator

|  |
| --- |
| **Class Name:** System Administrator |
| **Superclasses:** User |
| **Short Description:** This class defines what the System Administrator can do. A single instance of this class models the system administrator. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Service System Manager

|  |
| --- |
| **Class Name:** Service System Manager |
| **Superclasses:** Normal Staff |
| **Short Description:** This class defines the attributes that the Service System Manager will have and the operations that the Service System Manager can perform. A single instance of this class models the service system manager. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Reporting System Manager

|  |
| --- |
| **Class Name:** Reporting System Manager |
| **Superclasses:** Normal Staff |
| **Short Description:** This class defines the attributes that the Reporting System Manager will have and the operations that the Reporting System Manager can perform. A single instance of this class models the reporting system manager. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Profile System Manager

|  |
| --- |
| **Class Name:** Profile System Manager |
| **Superclasses:** Normal Staff |
| **Short Description:** This class defines the attributes that the Profile System Manager will have and the operations that the Profile System Manager can perform. A single instance of this class models the profile system manager. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Reservation System Manager

|  |
| --- |
| **Class Name:** Reservation System Manager |
| **Superclasses:** Normal Staff |
| **Short Description:** This class defines the attributes that the Reservation System Manager will have and the operations that the Reservation System Manager can perform. A single instance of this class models reservation system manager. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Flight Manager

|  |
| --- |
| **Class Name:** Flight Manager |
| **Superclasses:** Normal Staff |
| **Short Description:** This class defines the attributes that the Flight Manager will have and the operations that the Flight Manager can perform. A single instance of this class models flight manager. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Customer

|  |
| --- |
| **Class Name:** Customer |
| **Superclasses:** Abstract User |
| **Short Description:** This class defines the attributes that the Customer will have and the operations that the Customer can perform. A single instance of this class models a single customer. Therefore, all customers of the system will interact with an instance of this class to operate the system. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Travel Agency

|  |
| --- |
| **Class Name:** Travel Agency |
| **Superclasses:** Abstract User |
| **Short Description:** This class defines the attributes that the Travel Agency will have and the operations that the Travel Agency can perform. A single instance of this class models a single travel agency. Therefore, all travel agencies will interact with an instance of this class to operate the system. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Report Builder

|  |
| --- |
| **Class Name:** Report Builder |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that the Report Builder will have and the operations that the Report Builder can perform. It builds a different report types based on the message given to it. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Flight Price

|  |
| --- |
| **Class Name:** Flight Price |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that connect a flight to its price and the operations that can be used to manipulate those attributes. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Service

|  |
| --- |
| **Class Name:** Service |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that a Service will have and the operations that can be performed on Service. A single instance of this class models a single service. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Booking

|  |
| --- |
| **Class Name:** Booking |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that a Booking will have and the operations that the Booking can perform. This class links the various parts of a flight booking together by containing instances of Abstract Customer, Service, Person and Flight classes. A single instance of this class models a single booking. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Person

|  |
| --- |
| **Class Name:** Person |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that define a Person and the operations that can be used to manipulate those attributes. This class models the extra persons that are involved in a booking, but do not have an account. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Flight

|  |
| --- |
| **Class Name:** Flight |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that a Flight has and the operations that can be used to manipulate those attributes. This class defines a flight by containing instances of Flight and Plane classes. A single instance of this class models a single flight. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Airport

|  |
| --- |
| **Class Name:** Airport |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that an Airport has and the operations that can be used to manipulate those attributes. A single instance of this class models a single airport. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Route

|  |
| --- |
| **Class Name:** Route |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that a Route has and the operations that can be used to manipulate those attributes. A Route class requires two instances of the Airport class. A single instance of this class models a single route. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Seat

|  |
| --- |
| **Class Name:** Seat |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that a Seat has and the operations that can be used to manipulate those attributes. A single instance of this class models a single seat. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Plane

|  |
| --- |
| **Class Name:** Plane |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that a Plane has and the operations that can be used to manipulate those attributes. A single instance of this class models a single plane. |
| **List of Attributes:** |
| **List of Operations:** |

## Class: Address

|  |
| --- |
| **Class Name:** Address |
| **Superclasses:** None |
| **Short Description:** This class defines the attributes that an Address has and the operations that can be used to manipulate those attributes. A single instance of this class models a single address. |
| **List of Attributes:** |
| **List of Operations:** |