Appendix E

Flight Management System

Domain Model Description

Version 1.1

Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 19/Apr/2015 | 1.0 | Created the domain model description document | Michael Kong |
| 20/Apr/2015 | 1.1 | Added the domain model diagram to the report | Michael Kong |

Table of Contents

1. Introduction 4

1.1 Brief Description 4

1.2 Domain Model Diagram 5

2. Domain Model Description 6

2.1 Class: User 6

2.2 Class: Normal Staff 6

2.3 Class: Abstract Customer 6

2.4 Class: System Administrator 7

2.5 Class: Service System Manager 7

2.6 Class: Reporting System Manager 7

2.7 Class: Profile System Manager 8

2.8 Class: Reservation System Manager 8

2.9 Class: Flight Manager 8

2.10 Class: Customer 9

2.11 Class: Travel Agency 9

2.12 Class: Report Builder 9

2.13 Class: Flight Price 10

2.14 Class: Service 10

2.15 Class: Booking 10

2.16 Class: Person 10

2.17 Class: Flight 11

2.18 Class: Airport 11

2.19 Class: Route 11

2.20 Class: Seat 12

2.21 Class: Plane 12

2.22 Class: Address 12

Domain Model Description

# Introduction

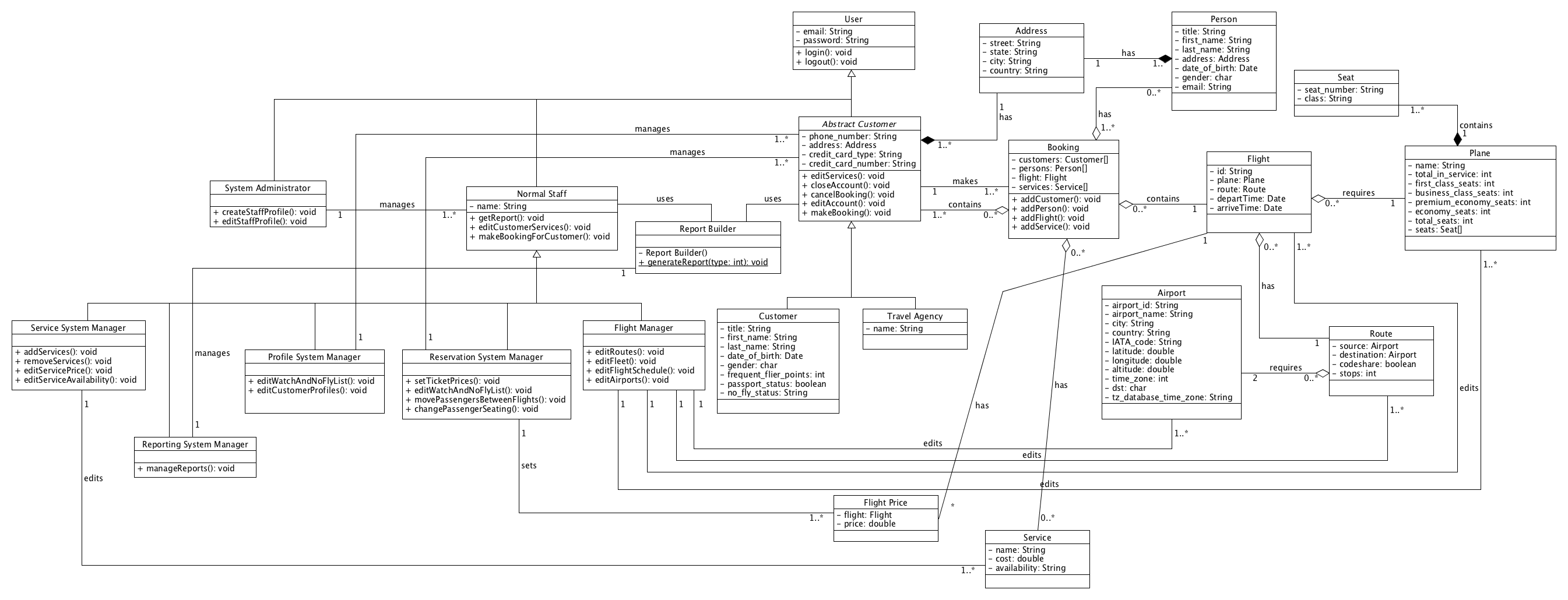
## Brief Description

The first part of this document covers the domain model diagram. It will give a graphical idea of how the different components of the system will interact with each other to produce a working system. The latter part of this document covers the domain model descriptions of 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. Operation name
      2. Return value
      3. Brief description

The goal of this document is to assist in understanding the function of each class and the relationships between the classes in the system.

## Domain Model Diagram



# 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:**  email: String type.  password: String type, but it will be hashed. |
| **List of Operations:**  login(): void  This method allows the user to login to the system.  logout(): void  This method allows the user to logout of the system. |

## 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:**  name: String type. |
| **List of Operations:**  getReport(): void  This method allows the normal staff to get a report from the Report Builder class.  editCustomerServices(): void  This method allows normal staff to edit customer services.  makeBookingForCustomer(): void  This method allows normal staff to make booking for customers. |

## 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:**  phone\_number: String type.  address: Address type. Defined in the Address class.  credit\_card\_type: String type.  credit\_card\_number: String type. |
| **List of Operations:**  editServices(): void  This method allows subclasses to edit services within a booking.  closeAccount(): void  This method allows subclasses to close their account.  cancelBooking(): void  This method allows subclasses to cancel bookings.  editAccount(): void  This method allows subclasses to edit the account details.  makeBooking(): void  This method allows subclasses to make booking. |

## 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:** None |
| **List of Operations:**  createStaffProfile(): void  This method allows the system administrator to create staff profiles.  editStaffProfile(): void  This method allows the system administrator to edit staff profiles. |

## 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:** None |
| **List of Operations:**  + addServices(): void  This method allows the service system manager to add services.  + removeServices(): void  This method allows the service system manager to remove services.  + editServicePrice(): void  This method allows the service system manager to edit the prices of services.  + editServiceAvailability(): void  This method allows the service system manager to edit the availability of the services. |

## 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:** None |
| **List of Operations:**  manageReports(): void  This method allows the profile system manager to manage the types of reports that can be produced by the Report Builder. |

## 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:** None |
| **List of Operations:**  editWatchAndNoFlyList(): void  This method allows the profile system manager to edit the watch and no fly list.  + editCustomerProfiles(): void  This method allows the profile system manager to edit the profiles of customers and travel agencies. |

## 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:** None |
| **List of Operations:**  setTicketPrices(): void  This method allows the reservation system manager to set the prices of flight tickets.  editWatchAndNoFlyList(): void  This method allows the reservation system manager to edit the watch and no fly list.  movePassengersBetweenFlights(): void  This method allows the reservation system manager to move customers between flights.  changePassengerSeating(): void  This method allows the reservation system manager to change the seatings of customers within a flight. |

## 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:** None |
| **List of Operations:**  editRoutes(): void  This method allows the flight manager to edit the routes that the airline can take.  editFleet(): void  This method allows the flight manager to edit the current planes in the airline’s fleet.  editFlightSchedule(): void  This method allows the flight manager to edit the flight schedule of the airline.  editAirports(): void  This method allows the flight manager to edit the source and destination airports. |

## 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:**  title: String type  first\_name: String type  last\_name: String type  date\_of\_birth: Date type defined in the Date class.  gender: char type (M/F)  frequent\_flier\_points: int type (if implementation language allows unsigned, then unsigned would be better because points can never contain a negative value)  passport\_status: boolean type (True/False)  no\_fly\_status: String type (Watch, No Fly, NULL) |
| **List of Operations:** None |

## 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:**  name: String type |
| **List of Operations:** None |

## 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:** None |
| **List of Operations:**  Report Builder()  The constructor here is private because the methods in this class will be class methods.  generateReport(type: int): void  This method is a class method that users of this function will call. It takes an int argument “type” to know what report to generate. |

## 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:**  flight: Flight type. Defined in the Flight class. It is a reference to the Flight object that this price is associated with.  price: double type. |
| **List of Operations:** None |

## 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:**  name: String type.  cost: double type.  availability: String type (Available/Unavailable) |
| **List of Operations:** None |

## 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:**  customers: Customer array type. An array of Customer objects that is associated with the flight.  persons: Person array type. An array of Person objects that is associated with the flight.  flight: Flight type. Reference to the Flight object associated with this booking.  services: Service array type. An array of Service objects that are associated with this booking. |
| **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:**  title: String type  first\_name: String type  last\_name: String type  address: Address type. Defined in the Address class  date\_of\_birth: Date type  gender: char type  email: String type |
| **List of Operations:** None |

## 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:**  id: String type  plane: Plane type. Refers to the Plane object that this flight will use.  route: Route type. Refers to the Route object that this flight will take.  departTime: Date type. The time that this flight departs from the source airport.  arriveTime: Date type. The time that this flight arrives at the destination airport. |
| **List of Operations:** None |

## 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:**  airport\_id: String type. The unique ID of this airport.  airport\_name: String type.  city: String type.  country: String type.  IATA\_code: String type. Stands for International Air Transport Association code that is a unique 3-letter code for an airport.  latitude: double type.  longitude: double type.  altitude: double type.  time\_zone: int type. Plain integer since the time zone can be a negative value.  dst: char type. Denotes the type of daylight savings time that this airport region observes.  tz\_database\_time\_zone: String type. Timezone in "tz" (Olson) format. |
| **List of Operations:** None |

## 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:**  source: Airport type. Defined in the Airport class. Refers to the Airport object that is the source airport of this route.  destination: Airport type. Defined in the Airport class. Refers to the Airport object that is the destination airport of this route.  codeshare: boolean type. This means that this flight is shared with another airline  stops: int type. The number of stops that this route takes. |
| **List of Operations:** None |

## 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:**  seat\_number: String type.  class: String type. (First class, Business, Premium Economy, Economy) |
| **List of Operations:** None |

## 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:**  name: String type. This is the type or model of the plane.  total\_in\_service: int type. The number of plane of this model currently in use.  first\_class\_seats: int type. Number of first class seats available.  business\_class\_seats: int type. Number of business class seats available.  premium\_economy\_seats: int type. Number of premium economy seats available.  economy\_seats: int type. Number of economy seats available  total\_seats: int type. Total number of seats available on this plane type or model.  seats: Seat array type. Refers to all the seats that this plane contains. |
| **List of Operations:** None |

## 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:**  street: String type.  state: String type.  city: String type.  country: String type. |
| **List of Operations:** None |