***Architectural Design***

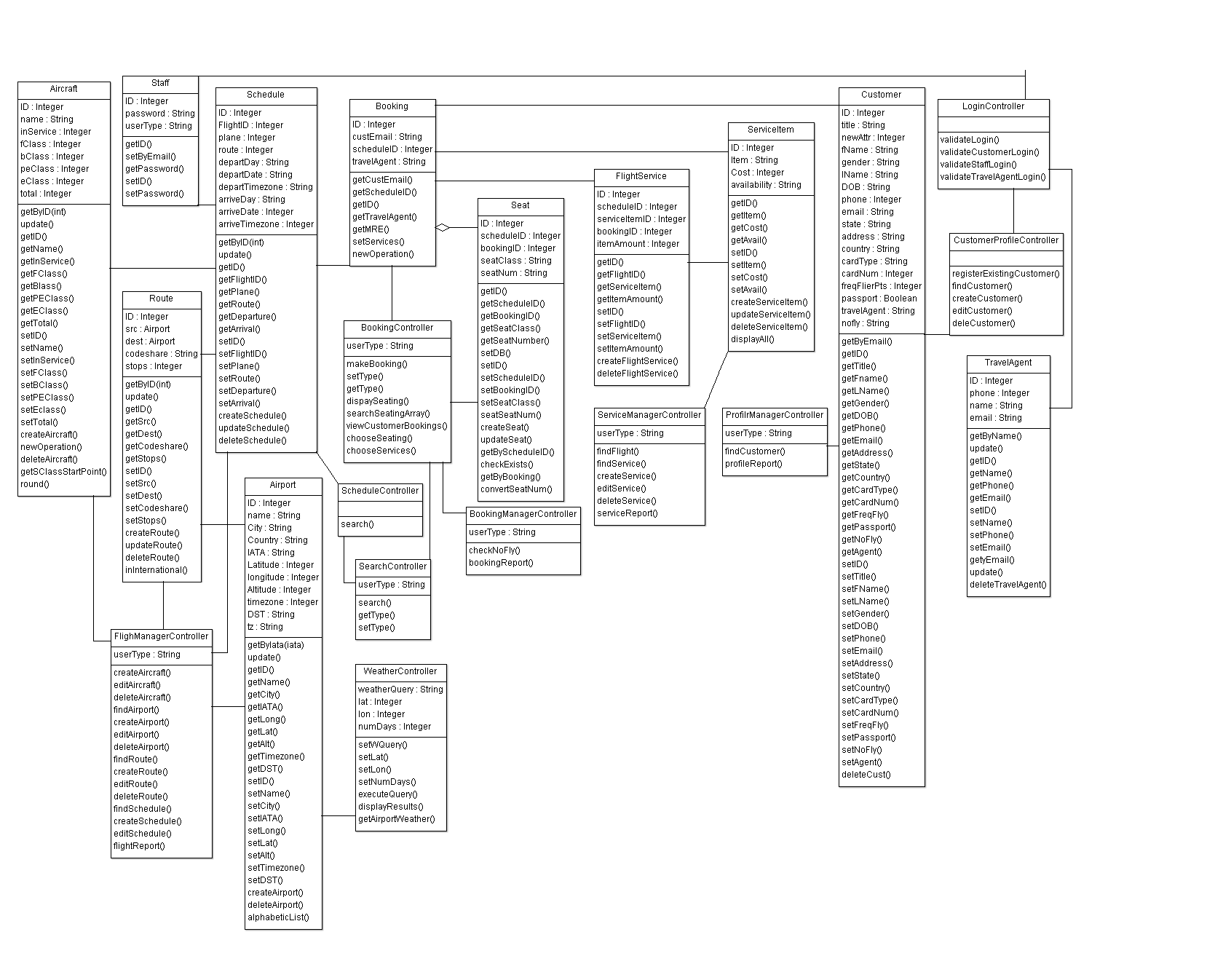
***Logical View:***

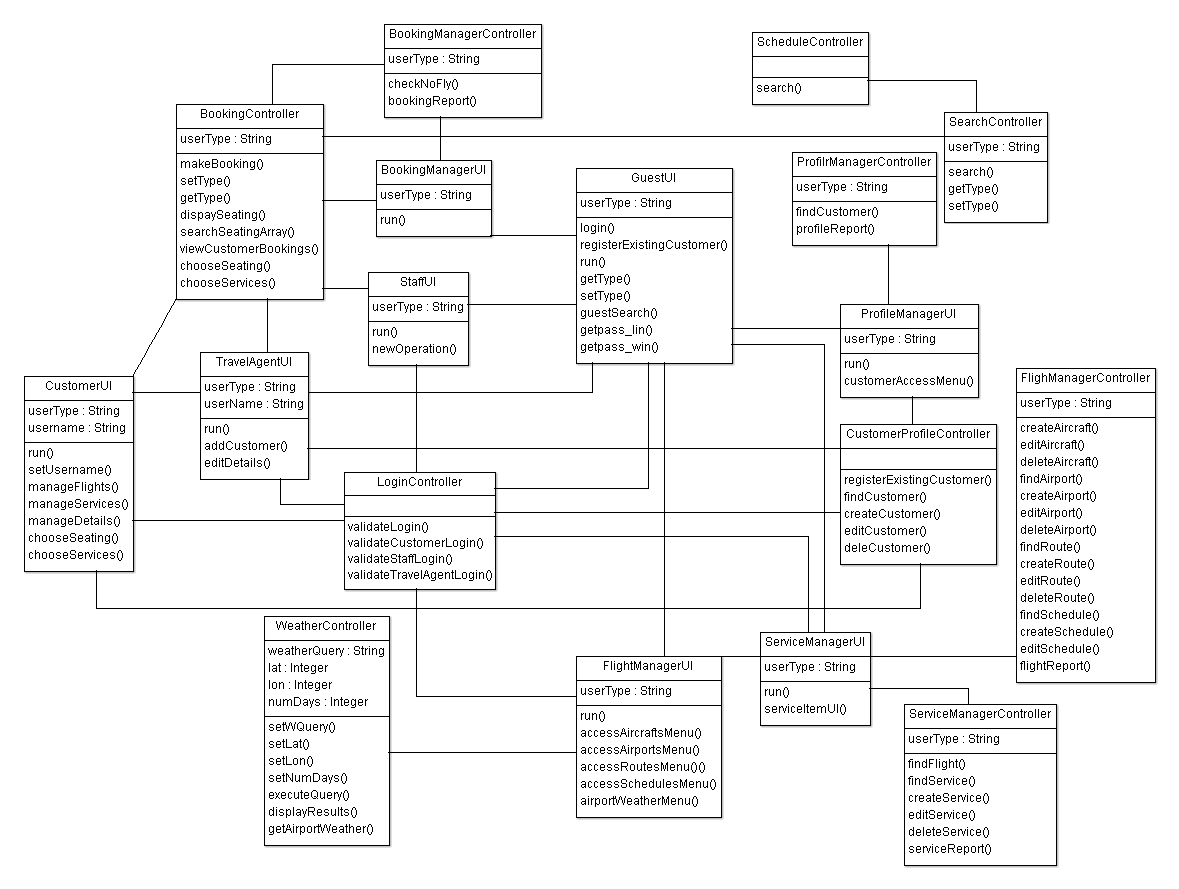
The purpose of the logical view is to describe the static structure of the system and the functionality that the system will provide to the end user. Our model is split into 3 different class diagrams (or “views”). These views were created in order to illustrate parts of the system in a context that is easier to understand (compared to one large diagram).

The following refine domain model has been split into 2 sections for ease of under standing. The first diagram demonsrates the relationships between our entity classes and our controller classes. The second diagram demonstrates the relationships between out controller classes and out user interface classes.

Refined ***Domain Model:***

Refined model part 1: Controller relations to UserInterface



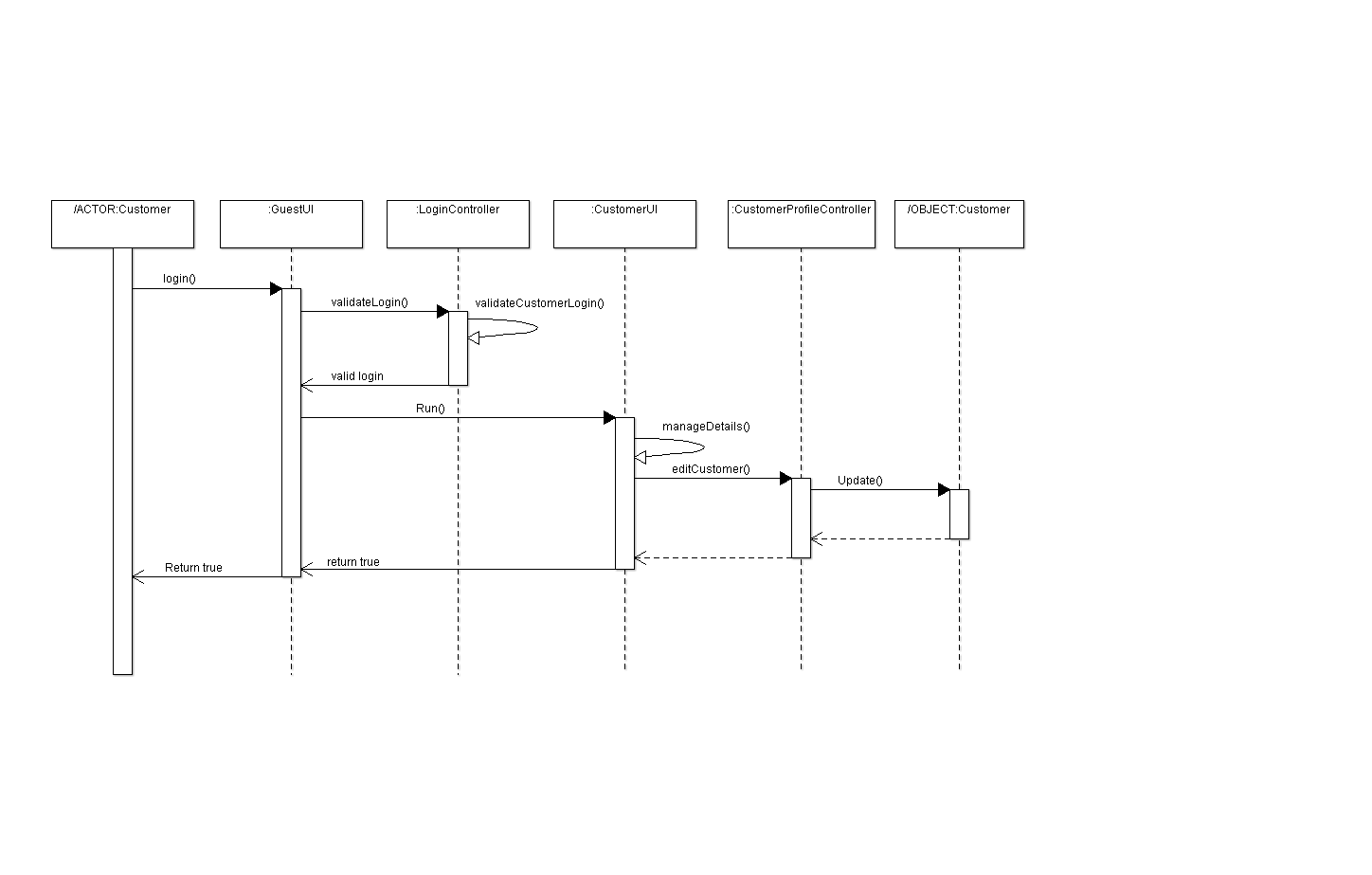


***Process View:***

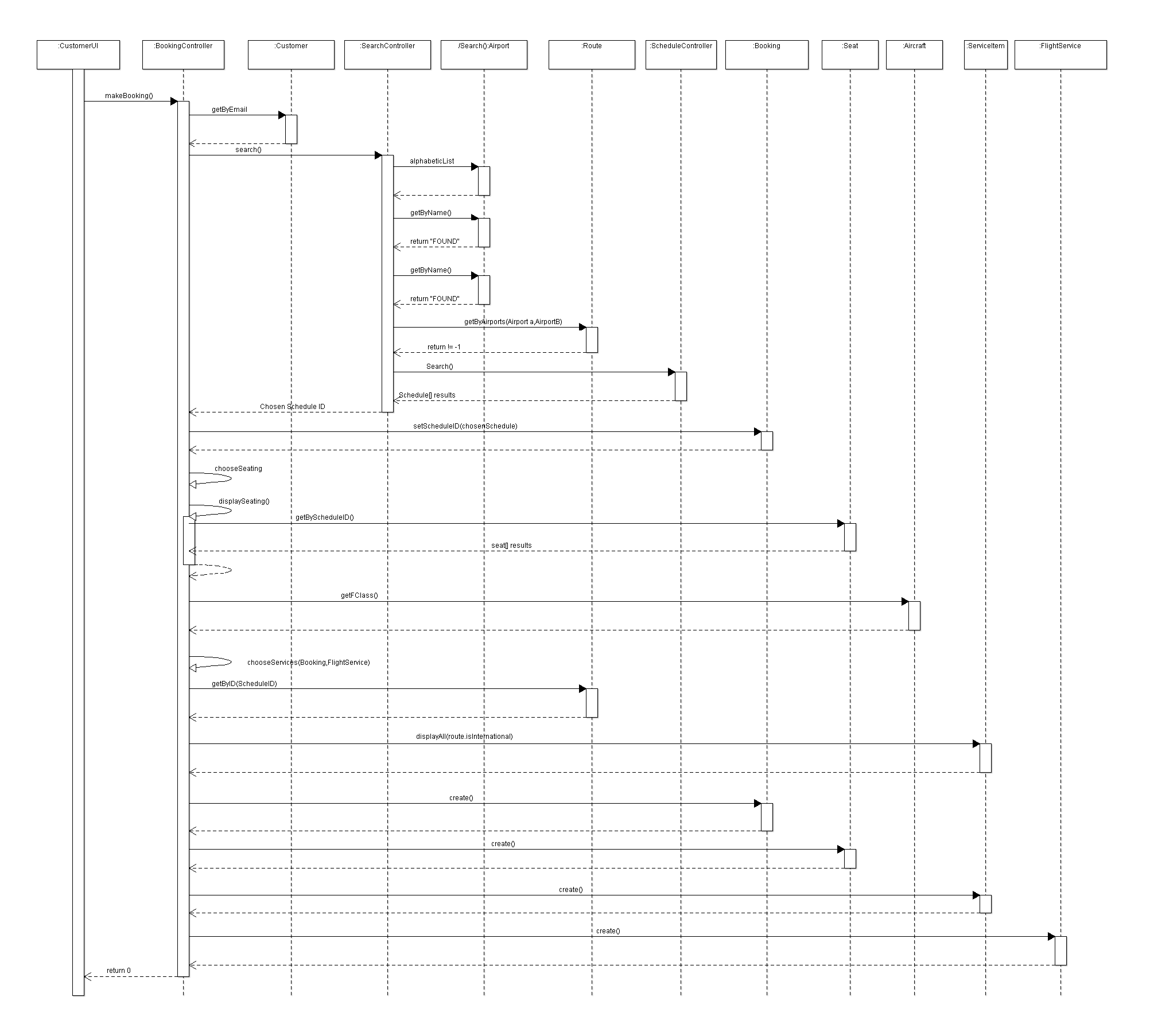
The purpose of the process view is to describe the dynamic nature of the system. We have 4 sequence diagrams that realise selected major use-cases, and depict the run-time behaviour of the system.

*Note 1: ArgoUML does not support the ‘Actor’ stick figure in sequence diagrams. Thus the actor will be defined with ‘ACTOR’ displayed at the top of an object instance.*

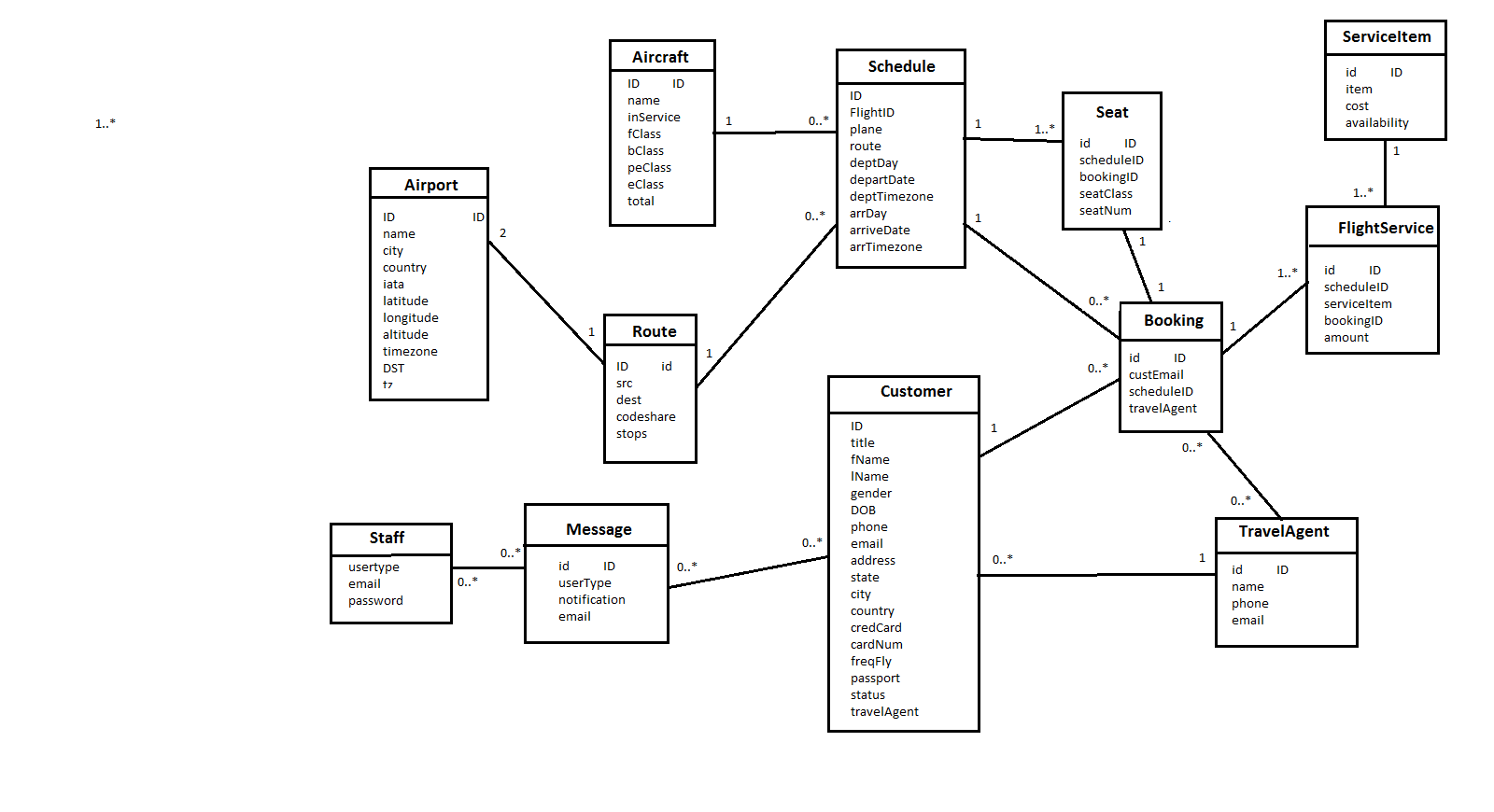
***Customer logs in and edits profile – Sequence Diagram***



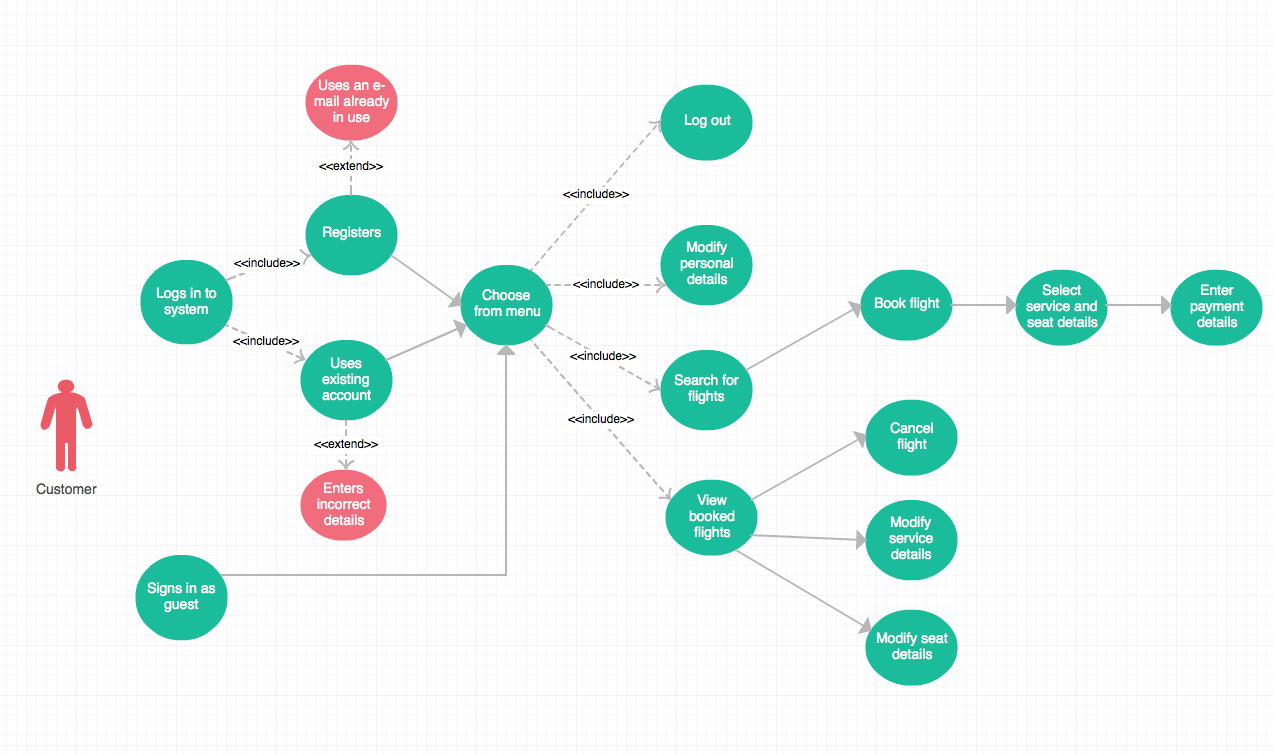
***Customer books flight sequence diagram***



Database model



***Use Case View***

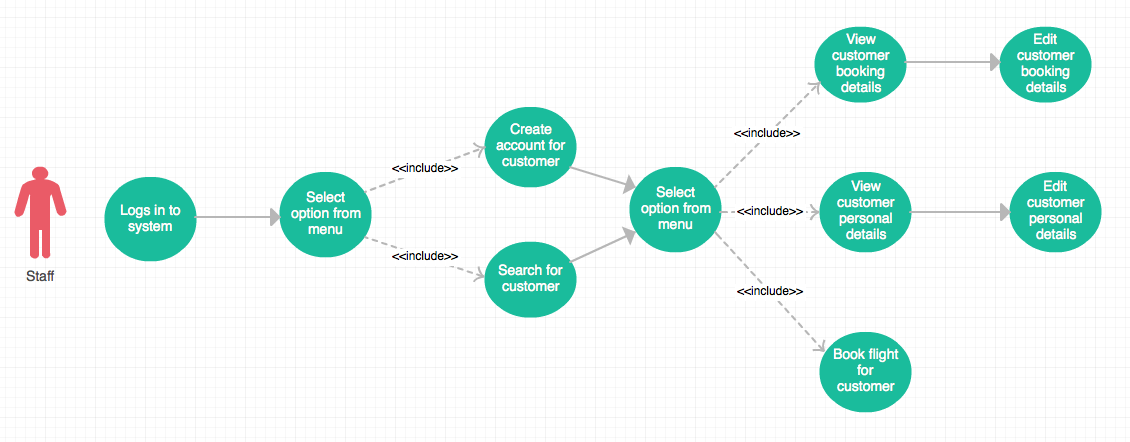


|  |  |
| --- | --- |
| Use Case Number | #1 |
| Use Case Name | Registers |
| Use Case Description | A guest user accesses the system and requests to register an account. After they have chosen a vaild email address the customer logs into the system |
| Primary Actor | User |
| Precondition | * Guest does not have an account |
| Trigger | Guest user wants to register an account. |
| Basic Flow | 1. User selects the option to register to the system 2. Details about the user are entered and stored 3. User chooses a password for account security. 4. User is redirected to log in screen after successful registration. 5. The user logs onto the system as a registered customer. |

|  |  |
| --- | --- |
| Use Case Number | #2 |
| Use Case Name | Guest search for flight |
| Use Case Description | A guest user uses the flight Systems search function to find available flights. |
| Primary Actor | User |
| Precondition | * The guest does not have a registered account. |
| Trigger | A guest wants to search the system for available flights. |
| Basic Flow | 1. Guest selects the option from the menu to begin flight search 2. The user will input an arrival and a departure airport 3. The user then inputs a start date for the search, and an end date, 4. The user then sees a list of available flights corresponding to the search parameters. |

|  |  |
| --- | --- |
| Use Case Number | #3 |
| Use Case Name | Customer books flight |
| Use Case Description | After loggin into the system, a customer proceeds through the route choice, date choice and service choice processes. |
| Primary Actor | Customer |
| Precondition | * User is a registered customer of the system. |
| Trigger | Customer wants to book a flight. |
| Basic Flow | 1) User logs into the system using their email and password.  2) User selects option to manage flights.  3) User selects option to book a flight  4) System give the user a list of available airports to fly from.  5) User chooses both a departure and arrival airport.  6) User enters a beginning and end date from which to search from/to.  7) User is provided with a list of available flights, The user inputs the ID of one of these flights.  8) User is provided with a list of available service Items for the flight.  9) user chooses 0 or more service items for their flights.  10) Booking is finalised. |

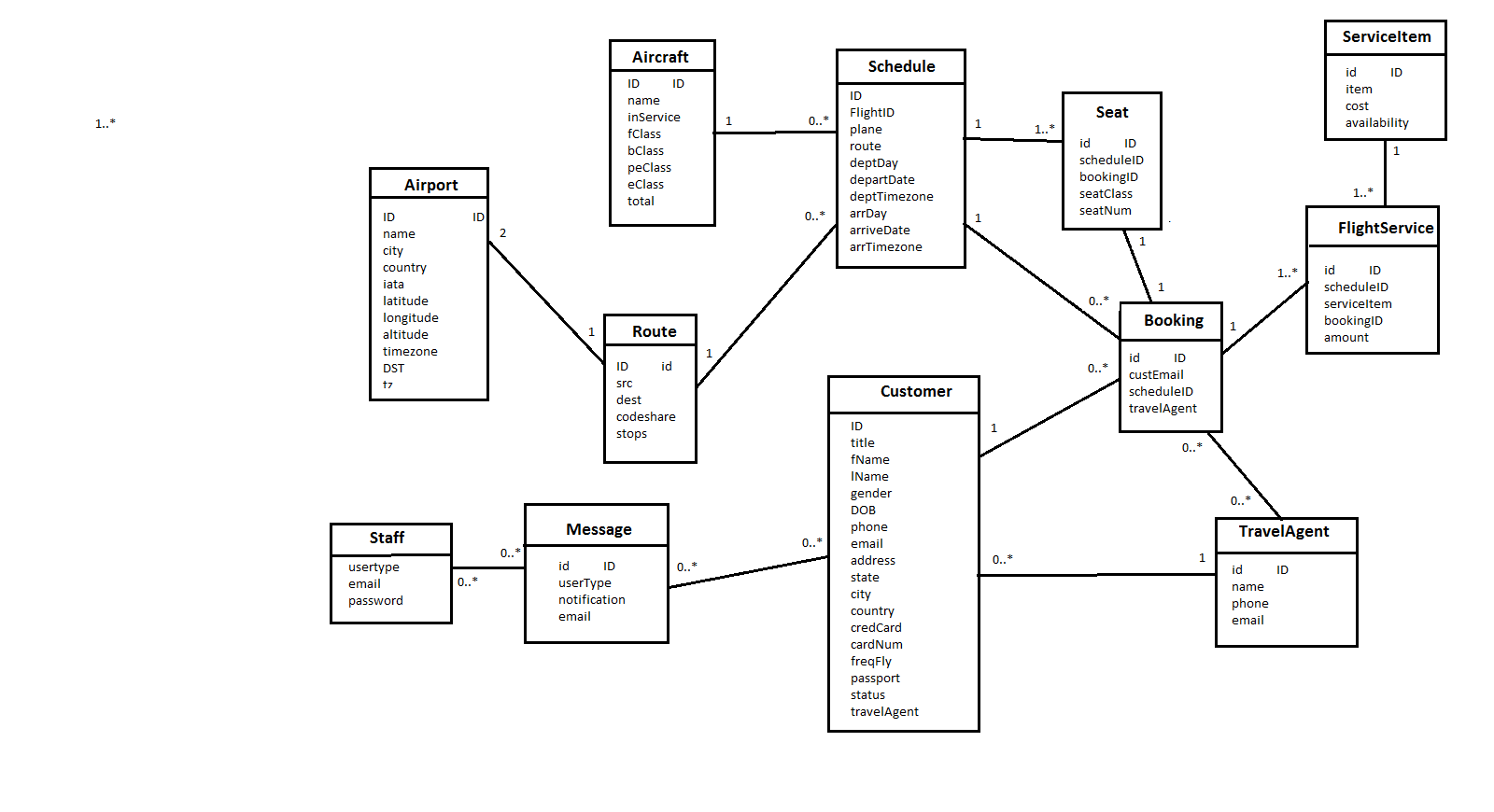
|  |  |
| --- | --- |
| Use Case Number | #4 |
| Use Case Name | Modify Personal Details |
| Use Case Description | A Customer moves through the menu system to change their personal details that are stored on the system |
| Primary Actor | Customer |
| Precondition | * User must have a valid account with the system. |
| Trigger | Customer wants to change their personal details. |
| Basic Flow | 1. User logs into the system with their username and password 2. The user selects the option to edit personal details. 3. The user is provided with fields to enter their new details 4. The user finalises the modification and finished the process. |



|  |  |
| --- | --- |
| Use Case Number | #5 |
| Use Case Name | Staff create account for customer |
| Use Case Description | A staff member fills in all the information required to create a new customer in the system. |
| Primary Actor | Staff |
| Precondition | * User must have staff member permissions |
| Trigger | A staff member wants to create a new customer. |
| Basic Flow | 1) Staff member logs into system with username and password.  2) They navigate to the menu option to create a new customer  3) the system provides the user with fields to enter the new customers details.  4) once complete the staff member confirms the details and customer registration is complete. |

|  |  |
| --- | --- |
| Use Case Number | #6 |
| Use Case Name | Staff Search for customer details. |
| Use Case Description | A staff member uses a customers email to search for the customers details on the system. |
| Primary Actor | Staff |
| Precondition | * User must have staff member permissions |
| Trigger | A staff member wants to create a new customer. |
| Basic Flow | 1) Staff member logs into system with username and password.  2) They navigate to the menu option to create a new customer  3) the system provides the user with fields to enter the new customers details.  4) once complete the staff member confirms the details and customer registration is complete. |

***Persistent Data Model***

******