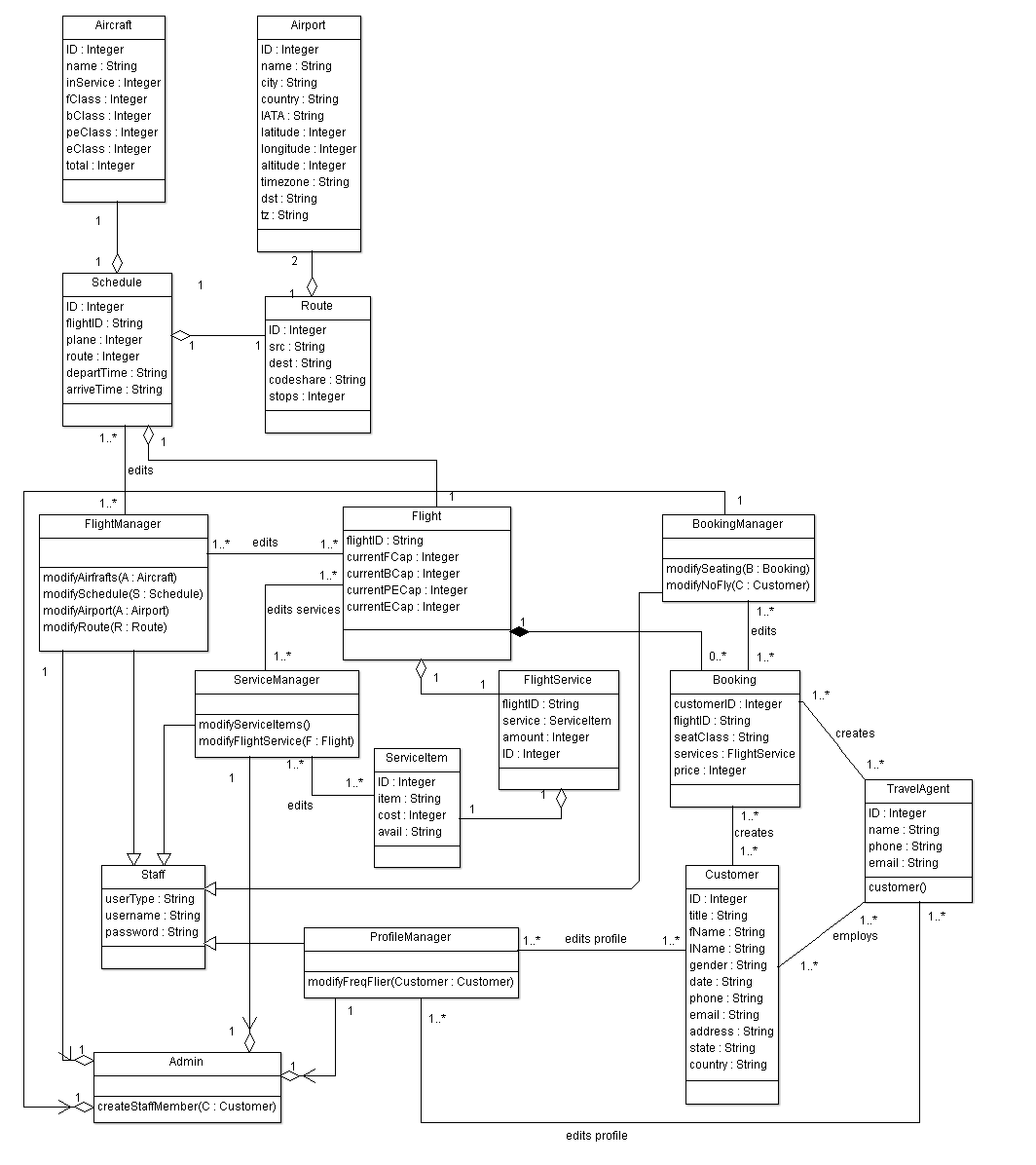
**Architectural Design Document**

***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).

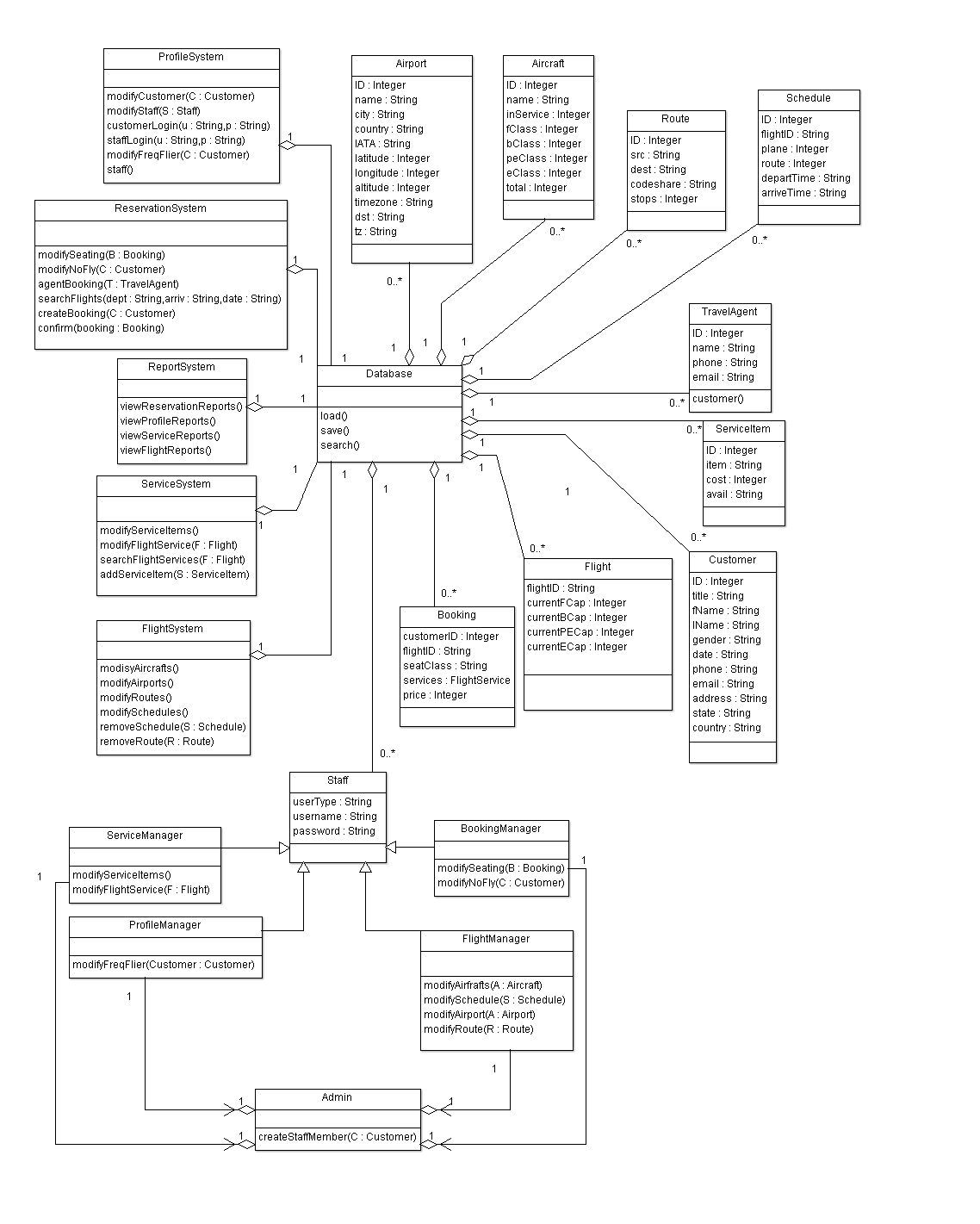
***Domain Model:***

The domain model depicts the key data elements from the system requirements and how they are associated to the users of the system. More complex views of the system seen be seen below.



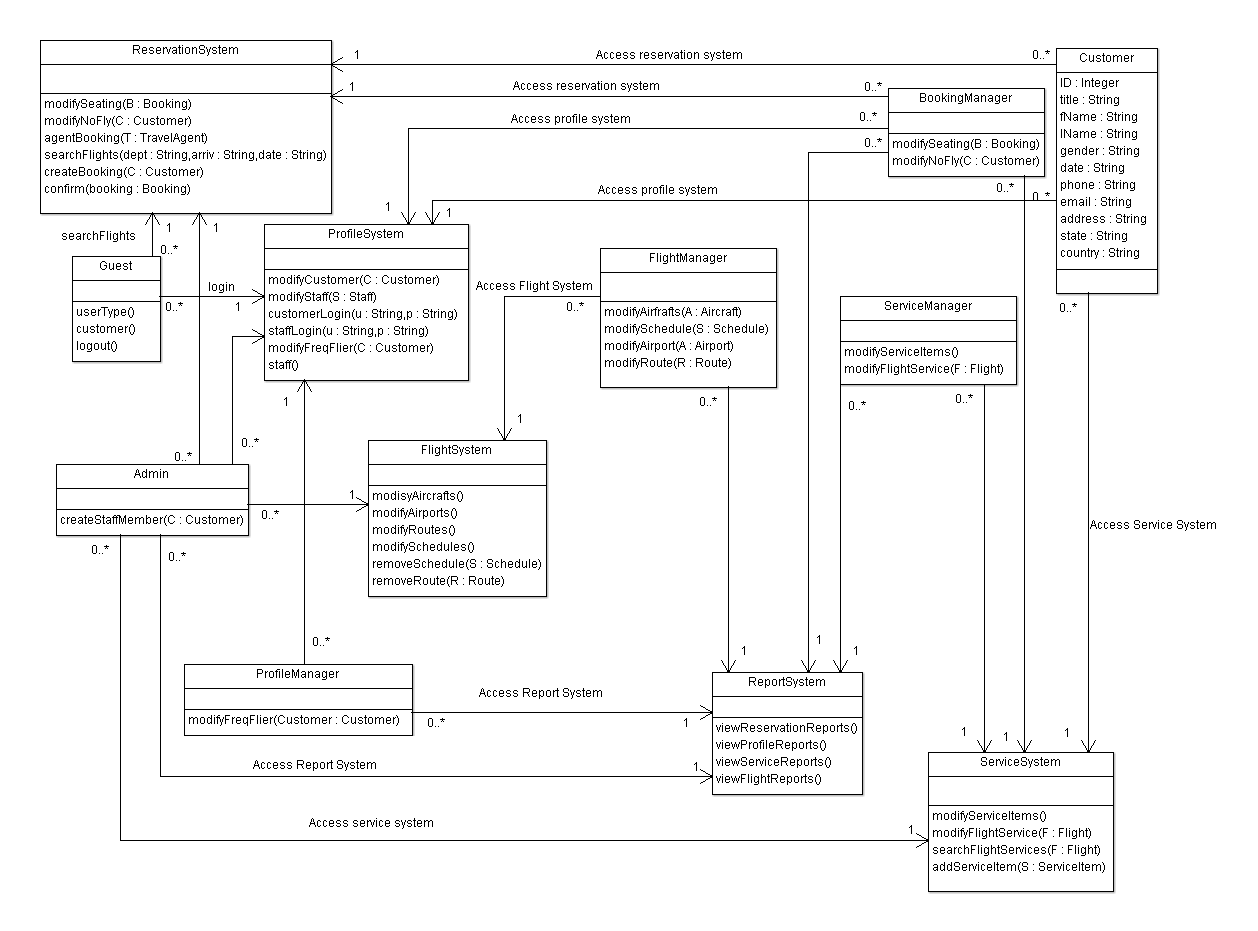
**Sub-System and Database view:**

The goal of this class diagram is to illustrate how data storage classes (e.g. ‘Airport’, ‘Route’) are related to the database system, and how the several sub-systems relate to the database system. The most important concept shown here is that no sub-system can directly access a base class. All access to these classes must be done via database functionality. This will help maintain database integrity.



**User and Sub-System view:**

The goal of this class diagram is to illustrate how user classes relate to the sub-systems. The most important concept shown here is that the user classes can only access the sub-systems that they require to perform their respective jobs. Restricting access to sub-systems based on the user class will help ensure a user cannot access functionality they are not authorized to use.



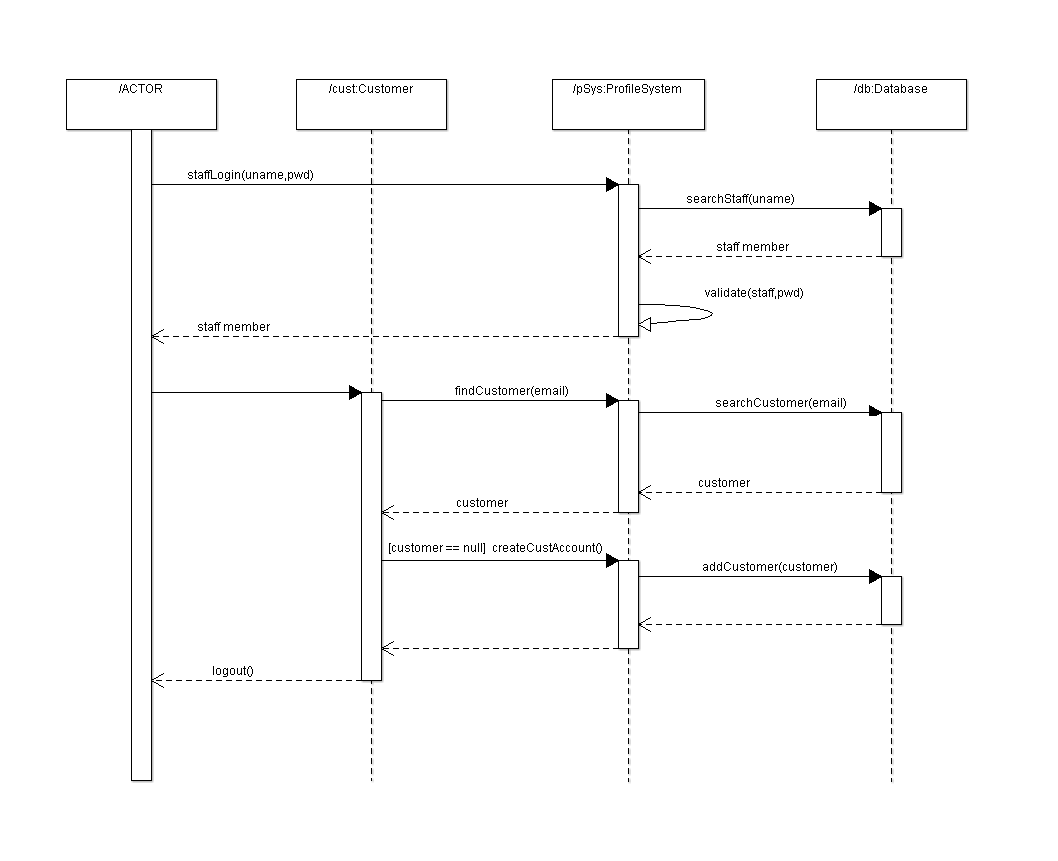
***Process View:***

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.

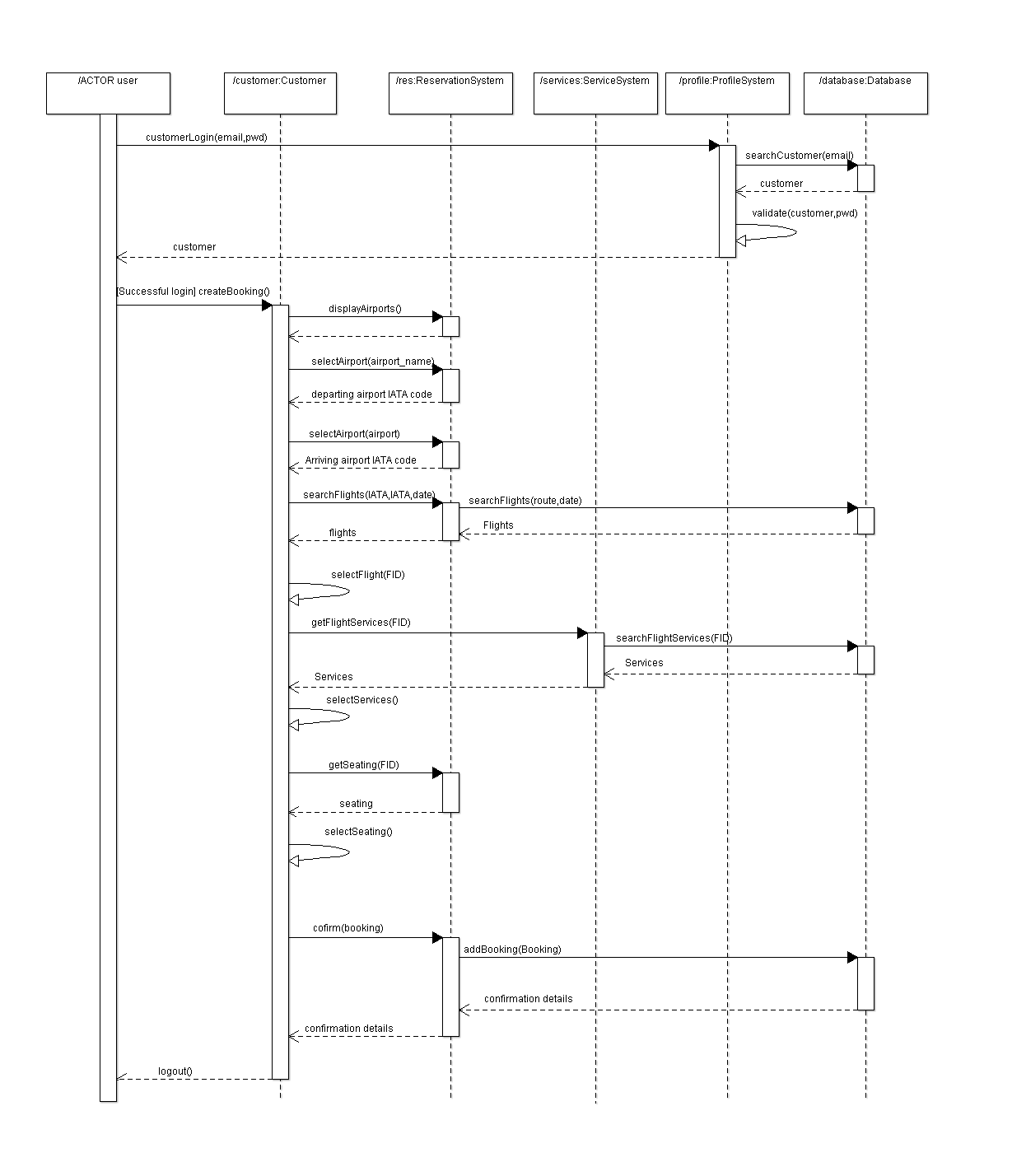
*Note 1: ArgoUML does not support the ‘Actor’ stick figure in sequence diagrams. Thus the actor will be defined with ‘ACTOR’ displayed at the top of an object instance.*

*Note 2: Not all functions in the sequence diagrams are displayed in the class diagram (too many functions created clutter and adversely affect the clarity of the diagram). Full functionality of a class is defined in the Data Dictionary.*

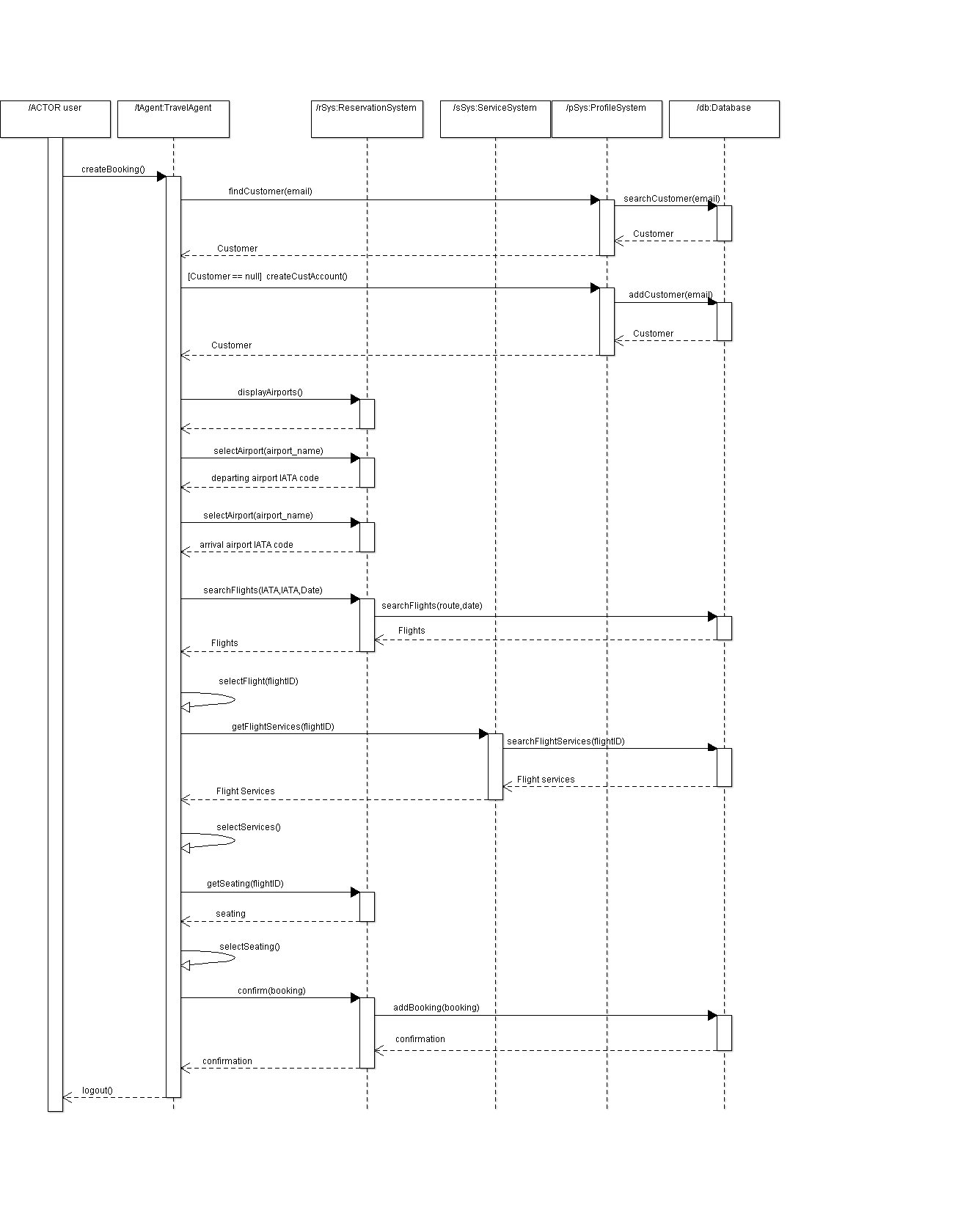
**Staff Member Creates Account for Customer - Sequence Diagram:**



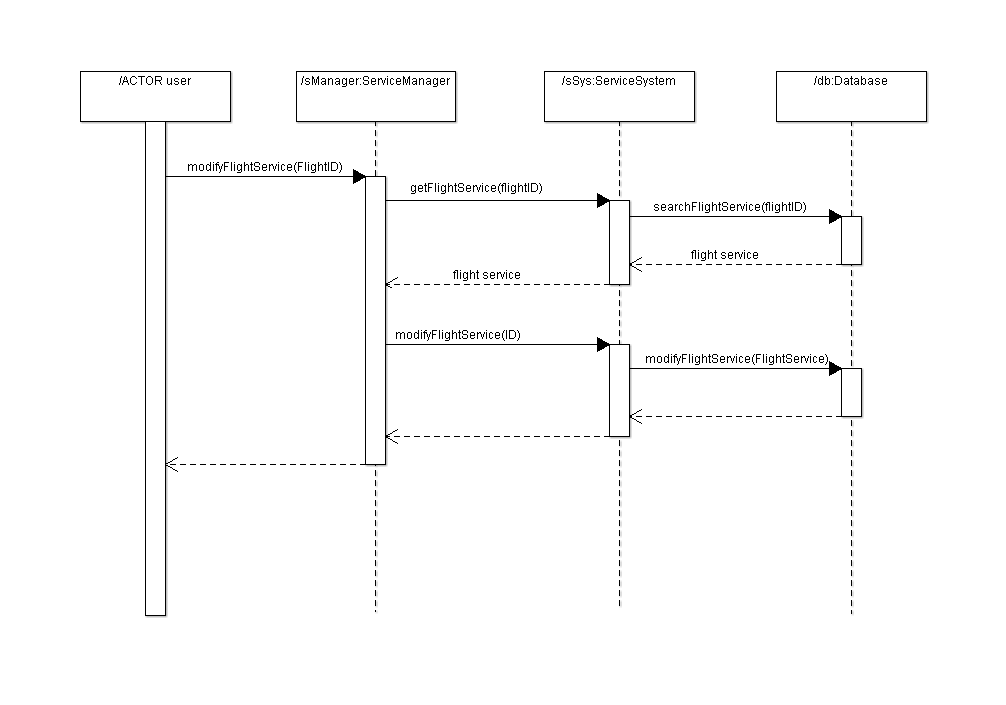
**Customer Books Flight – Sequence Diagram:**



**Travel Agent Books Flight for Customer – Sequence Diagram:**



**Service Manager Modifies Service Items for a Flight – Sequence Diagram:**



***Implementation View:***

The implementation views purpose is to illustrate the systems organisations of static modules. A component diagram is provided below that depicts the dependencies between modules, and some modules dependency on text files.

