# SYSTEM DESIGN:

## 1 U ML D IA GR A MS :

The Unified Modified Language prescribes a standard set of diagrams and notations for modeling object oriented systems, and describe the underlying semantics of what these diagrams and symbols mean. Whereas there has been to this point many notations and methods used for object-oriented design, now there is a single notation for modelers to learn.

UML can be used to model different kinds of systems: software systems, hardware systems, and real-world organizations. UML offers nine diagrams in which to model systems:

* + - **U s e C as e d i agram** for modeling the business processes
    - **S eq u en ce d i agram** for modeling message passing between objects
    - **C o l l ab oration d i agram** for modeling object interactions
    - **S tate d i agram** for modeling the behavior of objects in the system
    - **A c t i v i ty d i agram** for modeling the behavior of Use Cases, Objects, or Operations
    - **C l as s d i agram** for modeling the static structure of classes in the system
    - **Ob j ect d i agram** for modeling the static structure of objects in the system
    - **C om pon en t d i agram** for modeling components
    - **D ep l oy m ent d i agram** for modeling distribution of the system.

UML is a consolidation of many of the most used object-oriented notations and concepts. It began as a consolidation of the work of Grady Booch, James Rumbaugh, and Ivar Jacobson, creators of three of the most popular object-oriented methodologies.

In 1996, the Object Management Group(OMG), a standards body for the object- oriented community, issued a request for proposal for a standard object-oriented analysis notation and semantic meta model. UML, version 1.0, was proposed as an answer to this submission in January of 1997. There were five other rival submissions. During the course of 1997, all six submitters united their work and presented to OMG a revised UML document, called UML version 1.1. This document was approved by the OMG in November 1997. The OMG calls this document OMG UML version 1.1. The OMG is currently in the process of performing a technical.

## 4 . 1 . 1 C LAS S D IA GR A MS :

The class diagram is the main static analysis and design diagram for a system. In it, the class structure of the system is specified, with relationships between classes and inheritance structures. During analysis of the system, the diagram is developed with an eye for an ideal solution. During design, the same diagram is used, and modified to conform to implementation details.

## 4 . 1 . 2 . A C TI VI TY D IA GR A MS :

The Activity Diagram is a multi-purpose process flow diagram that is used to model behavior of the system. Activity Diagram can be used to model a Use Case, or a class, or a complicated method. An Activity Diagram can show parallel processing. This is important when using Activity Diagram to model business processes, some of which can be performed in parallel, and for modeling multiple threads in concurrent programs.

## . 3 S EQU ENC E D IA GR A M:

The Sequence diagram is one of the most effective diagrams to model object interactions in a system. A Sequence diagram is modeled for every Use Case. Whereas the Use Case diagram enables modeling of a business view of the scenario, the Sequence diagram contains implementation details of the scenario, including the objects and classes that are used to implement the scenario, and messages passed between the objects.

## . 4 U S E CA S E DI A GRA M:

Use Case modeling is the simplest and most effective technique for modeling system requirements from a user’s perspective. Use Cases are used to model how a system or business currently works, or how the users wish it to work. It is not really an object-oriented approach; it is really a form of process modeling. It is, however, an excellent way to lead into object-oriented analysis of systems. Use Cases are generally the starting point of object-oriented analysis with UML. The Use Case model consists of actors and Use Cases. Actors represent users and other systems that interact with the system. They are drawn as stick figures. They actually represent a type of user, not an instance of a user. Use Cases represent the behavior of the system, scenario that the system goes through in response to stimuli from an actor. They are drawn as Ellipses.

Each Use Case is documented by a description of the scenario. The description can be written in textual form or in a step-by-step format. Each Use Case can also be defined by other properties, such as the pre- and post conditions of the scenario – conditions that exist before the scenario begins, and conditions that exist after the scenario completes.

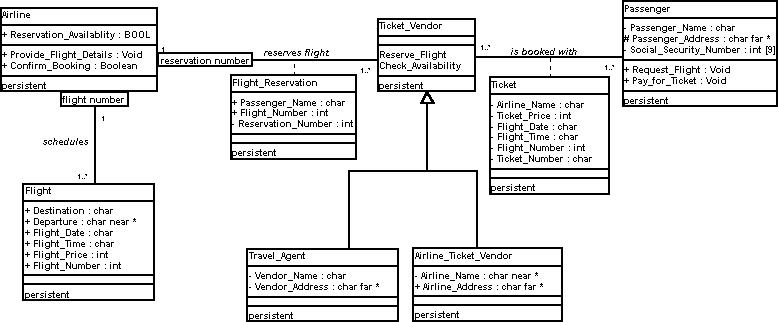
## . 5 C OLLA BOR A TI ON D IA GR A M:

The Collaboration Diagram presents an alternate to the Sequence Diagram for modeling interactions between objects in the system. Whereas in the Sequence Diagram the focus is on the chronological sequence of the scenario being modeled, in the Collaboration Diagram the focus is on understanding all of the effects on a given object

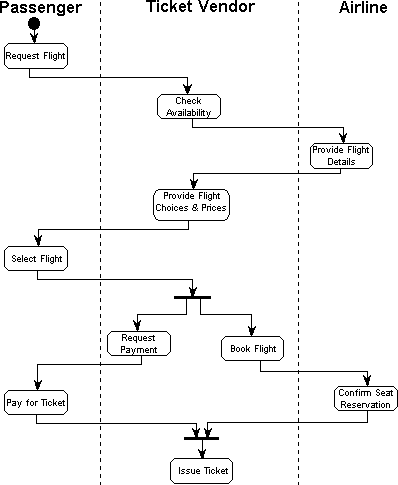
during a scenario.

Objects are connected by links, each link representing an instance of an association between the respective classes involved. The link shows messages sent between the objects, the type of message passed, and the visibility of objects to each other.

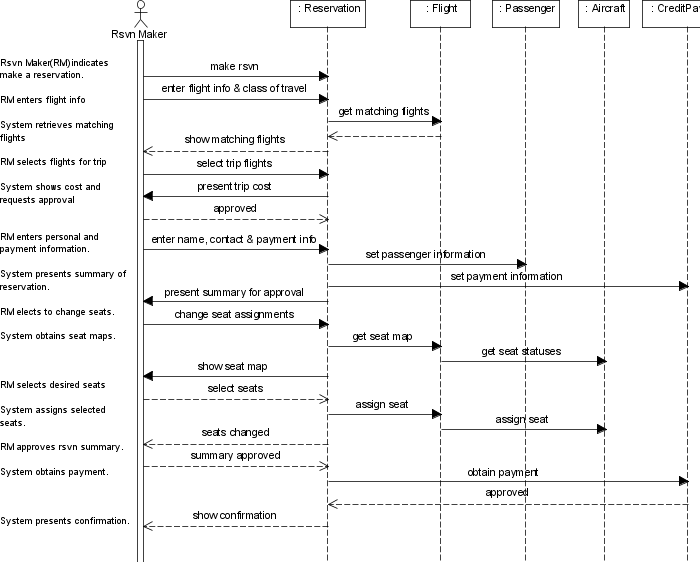
## . 6 D A TA FLOW DI A GR A MS :



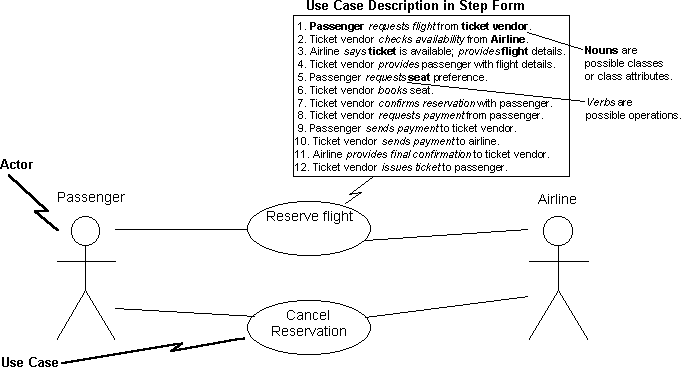
**Figure 1 Class Diagram**



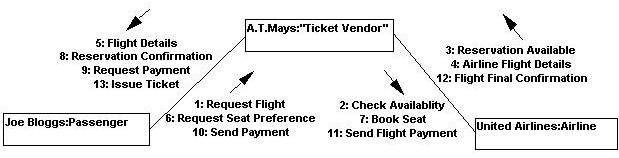
**Figure 2 Activity Diagram**



**Figure 3 Sequence Diagram**



**Figure 4 Use Case Diagrams**



**Figure 5 Collaboration Diagram**

# 2 DATA DICTIONARY:

## . 1 I N TR OD U C TI ON :

**D IC TI ON AR Y** The logical characteristics of current system data stores including Name, Address, Flight code, Source, Destination, Airline code, Flight code, Credit card number, Payment amount etc identifies process where the data are used and where immediate access to information required, Serves as the basis for identifying database requirements during system design.

## U s es of Data D i c t i on ary :

* + - * To manage the details in large systems.
      * To communicate a common meaning for all system elements.
      * To document the features of the system.
      * To facilities analysis of the details in order to evaluate characteristics and determine where system changes should be made.
      * To locate errors and omissions in the system.

## . 2 D A TA D IC TI ON AR Y :

1. **. C an cellation .**

This table is used to store the cancel details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | Constraints |
| cancelid | Cancellation id | int | 10 | PRIMARY KEY |
| reservationid | Reservation id | int | 10 | FOREIGN KEY |
| cancelationdate | Date of Cancellation | date |  | NOT NULL |
| refundmoney | Money to be refundable | decimal | 10,0 | NOT NULL |

## Classes

This table is used to store the class details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| Classid | Id of the class | int | 10 | PRIMARY KEY |
| flightcode | - | int | 10 | FOREIGN KEY |
| classcode | - | varchar | 50 | NOT NULL |
| classname | Name of the class | varchar | 50 | NOT NULL |
| Fare | - | decimal | 10,0 | NOT NULL |
| totalclassseat | Total seats in a class | int | 10 | NOT NULL |

## . Fligh t d ay s

This table is used to store the flight day’s details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| datecode | Code of the date flight  departure | int | 10 | PRIMARY KEY |
| flightcode | Code of the flight | int | 10 | FOREIGN KE |
| Date | date | date |  | NOT NULL |
| departure | Departure time | time |  | NOT NULL |

## . Fligh t d e tails

This table is used to store the flight details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| flightcode | Code of the flight | int | 10 | PRIMARY KEY |
| airlinecode | Code of the airlines | varchar | 100 | NOT NULL |
| flightname | Name of the flight | varchar | 100 | NOT NULL |
| source | Starting place of the | varchar | 100 | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | flight |  |  |  |
| destination | Destination of the flight | varchar | 100 | NOT NULL |
| totalcapacity | Total capacity of the  flight | int | 10 | NOT NULL |

## . Login

This table is used to store the login details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| username | Name of the user | varchar | 50 | PRIMARY KEY |
| password | Password for the user  login | varchar | 50 | NOT NULL |

## . Mail

This table is used to store the mail details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| mailid | Users mail id | int | 10 | PRIMARY KEY |
| Touser | Destination of the mail | varchar | 250 | FOREIGN KEY |
| fromuser | By whom the mail is sent | varchar | 250 | FOREIGN KEY |
| Subject | Subject of the mail | varchar | 250 | NOT NULL |
| message | Message to be sent | text |  | NOT NULL |

## . N ew Us er

This table is used to store the new user details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| Userid | Users id | int | 10 | PRIMARY KEY |
| Firstname | First name of the user | varchar | 250 | NOT NULL |
| lastname | Last name of the user | varchar | 250 | NOT NULL |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| middlename | Middle name of the user | varchar | 250 | NOT NULL |
| mobileno | Mobile number | int | 10 | NOT NULL |
| Dob | Date of birth | date |  | NOT NULL |
| address | - | varchar | 250 | NOT NULL |
| City | - | varchar | 250 | NOT NULL |
| State | - | varchar | 250 | NOT NULL |
| country | - | varchar | 250 | NOT NULL |
| emailid | Mail id | varchar | 250 | NOT NULL |
| password | - | varchar | 250 | NOT NULL |
| usertype | Type of the user | varchar | 250 | NOT NULL |
| username | Name of the user | varchar | 250 | UNIQUE KEY |

## . Pas s en ger

This table is used to store passenger details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| passengerno | Number of the  passengers | Int | 10 | PRIMARY KEY |
| flightcode | Code of the flight | Int | 10 | FOREIGN KEY |
| Dob | Date of birth | Date |  | NOT NULL |
| address | - | varchar | 250 | NOT NULL |
| nationality | - | varchar | 250 | NOT NULL |
| Name | Name of the user | varchar | 250 | NOT NULL |
| Gender | - | varchar | 250 | NOT NULL |
| phoneno | Phone number | varchar | 250 | NOT NULL |
| emailid | Mail id | varchar | 250 | NOT NULL |
| passportno | Passport number | Int | 10 | NOT NULL |
| reservationid | Reservation id | Int | 10 | FOREIGN KEY |

## Payment

This table is used to store payment details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| paymentid | - | Int | 10 | PRIMARY KEY |
| checkno | Checking number | Int | 10 | NOT NULL |
| creditcardno | - | Int | 10 | NOT NULL |
| paidamount | - | decimal | 10,0 | NOT NULL |
| paymentdate | - | date |  | NOT NULL |
| passengerno | - | Int | 10 | NOT NULL |

## . R es ervation

This table is used to store reservation details.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| reservationid | - | Int | 10 | PRIMARY KEY |
| flightcode | - | Int | 10 | FOREIGN KEY |
| journeydate | - | Date |  | NOT NULL |
| Source | Starting place of the  flight | varchar | 50 | NOT NULL |
| destination | Ending position of the  flight | varchar | 50 | NOT NULL |
| Status | - | Int | 10 | NOT NULL |
| journeytime | Time the flight starts | time |  | NOT NULL |

## . R es erve ch eck

This table is used to store reserve check details

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field name | Description | Data type | Size | constraints |
| reservationid | Reservation id number | Int | 10 | PRIMARY KEY |
| username | Name of the user | Varchar | 250 |  |
| Password | Password to login | Varchar | 250 |  |