



COEN 6312

Model Driven Software Engineering

Deliverable-2

Project-Team

<u>Name</u>	<u>ID No.</u>
Ganesh Santhar	40010625
BhanuPrakash Ramineni	27107838
Jithin James	27420854
Rambabu Kunchala	27262957
Rakhi Ubriani	27396333

Index

1. Introduction	2
1.1 Purpose	2
1.2 Scope	2
1.3 General Description.....	2
2. User Requirements	2
2.1 Use Case Descriptions.....	3
2.2 Use Case Scheme	4
2.3 Complete Description of Use Cases.....	5
3. System Requirements	12
3.1 Functional Requirements.....	12
3.2 Non-Functional Requirements	13
Glossary	14
References.....	14
Appendix.....	14

1. Introduction

1.1 Purpose

The purpose of this document is intended to analyze the Flight Reservation System and define the requirements of the system.

1.2 Scope

The scope of this document resides in analyzing the Flight Reservation system in relation to User Interaction. It also list down the User requirements and system requirements based on use case scheme.

1.3 General Description

The Flight Reservation System enables a Traveler to book flights online. It provides a sophisticated interface for the User to do the same. The system offers to create a personal account and can login into it. A Traveler will be able to search Flights based on various criteria namely Domestic or International travel, One way or Round Trip, Source, Destination, Date of travel, Number and Type of passengers. Upon Flight selection it allows the Traveler to book the flight with facility to make online payment. The Traveler will have two modes of online payment through Credit/Debit card, Interac transfer and through Points payment. The system completes the Bank involved payments with a validation from the networked Bank systems. Once the booking is successful, the Traveler can Cancel the booking and can Modify to the booking like changing seats and adding extra baggage. An Administrator who looks after the system can be able to update the Flight database.

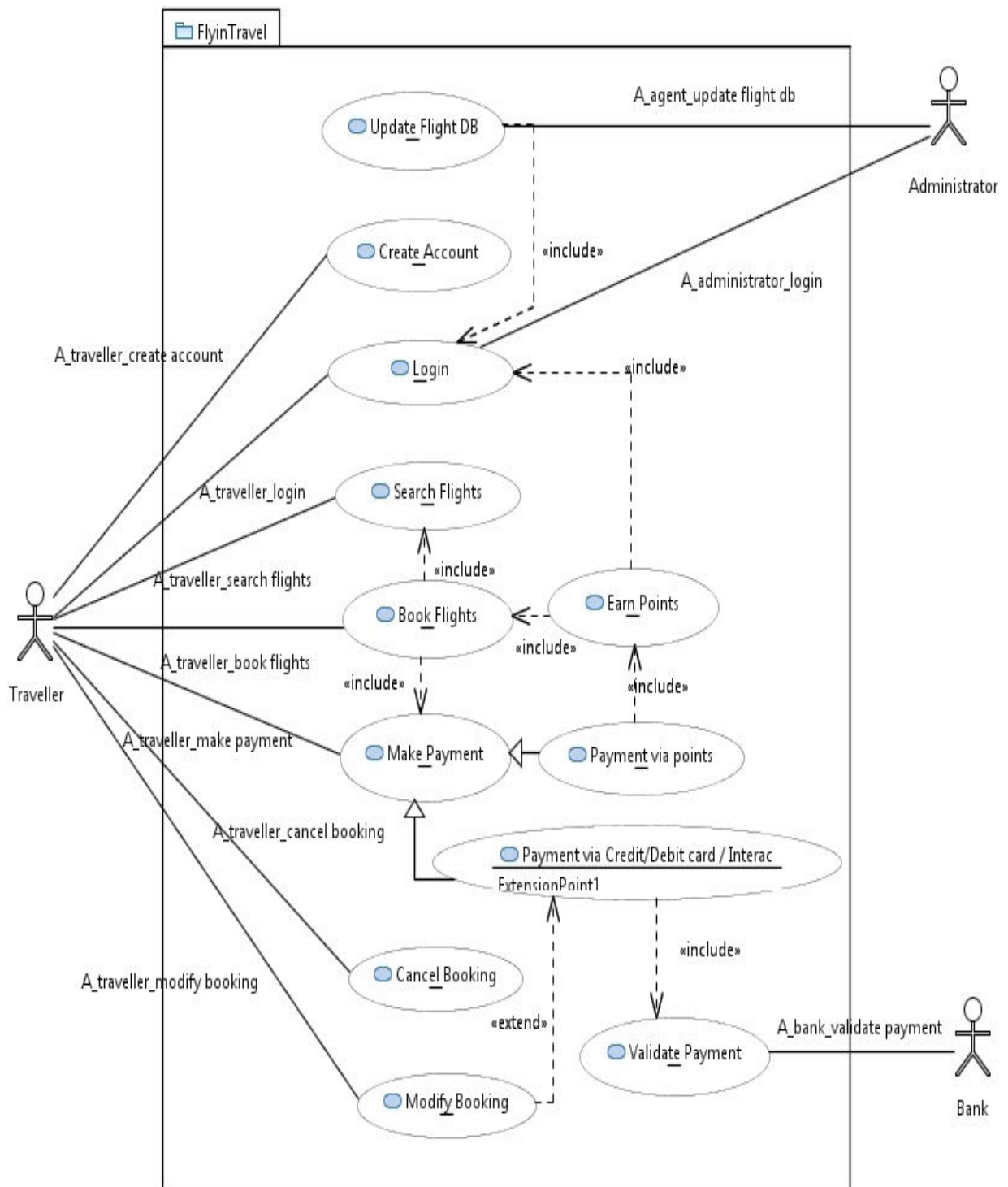
2. User Requirements

This section will analyze the system considering to various User Interactions. The system comprises of 3 actors interacting with the system. The following information will provide the Use Case descriptions and the Scheme demonstrates the Users Interaction with the system.

2.1 Use Case Descriptions

S. No	Use Case name	General Description
1.	Update Flight DB	The Administrator update the flights information in the database.
2.	Create Account	The Traveler can create a personal account in the system.
3.	Login	The User can login into personal account with valid account credentials.
4.	Search Flights	The Traveler can search for the flights based on search criteria.
5.	Flight Booking	The Traveler can book a flight from the retrieved search results.
6.	Make Payment	The Traveler needs to make payment for the booking in process.
7.	Cancel Booking	The Traveler can Cancel the booking with the respective booking reference ID.
8.	Modify Booking	The Traveler can make modifications to the booking.
9.	Earn Points	The Traveler with personal account will earn points for the miles travelled with the previous bookings.
10.	Payment via points	The Traveler can redeem their points as a payment for the current booking.
11.	Payment via Credit/Debit card /Interac transfer	The Traveler can make payment for the booking with a Credit/ Debit card/ Interac transfer.
12.	Validate Payment	The Bank will validate the Credit /Debit card /Interac transfer details of the booking payment.

2.2 Use Case Scheme



2.3 Complete Description of Use Cases

This section defines the Use case details such as Actor(s) involved, Basic Flow, Precondition(s), Post condition(s) and Exception(s) for all of the Use cases from Use Case Scheme.

Use case ID	UC-001
Name	Update Flight DB
Actor(s)	Administrator
Description	Administrator will update the flights information in the database.
Precondition(s)	1. Administrator needs to login into the system. 2. Administrator must have all the required flight information.
Basic Flow	1. The System provides Administrator account with Update Flight DB option. 2. Administrator clicks Update Flight DB. 3. The System provides option to add and drop Flights. 4. Administrator adds and drops Flights. 5. The System requests to confirm the update. 6. Administrator confirms the update.
Post Condition(s)	Flight database updated successfully.
Exception(s)	Error Message is displayed if the updated information is any match with other flights or mandatory fields of flight left blank.

Use case ID	UC-002
Name	Create Account
Actor(s)	Traveler
Description	The Traveler is facilitated to create a personal account in the system.
Precondition(s)	1. The Traveler must not have any existed account in the system. 2. The Traveler should disclose the necessary information in the process of creating account.

Basic Flow	<ol style="list-style-type: none"> 1. The Traveler request the system for creating account. 2. The system redirects the Traveler to signup form page where the personal information should be disclosed. 3. The Traveler submit after entering the valid Username and Password as the credentials for the personal account. 4. The system validates the Username, Password and the personal information with the existed accounts in database, and then store them in the database.
Post Condition(s)	Account Created successfully.
Exception(s)	Error Message is displayed if any of the information provided by the Traveler existed in the system or any mandatory fields left blank.

Use case ID	UC-003
Name	Login
Actor(s)	Administrator, Traveler
Description	The Traveler Login into the personal account with login credentials.
Precondition(s)	The Traveler should have an existed account in the system database.
Basic Flow	<ol style="list-style-type: none"> 1. The Traveler open the login page in the system. 2. The system requests the User to provide the login credentials. 3. The Traveler enters the Username and Password and submit to the system. 4. The system then verify the credentials provided with the stored credentials in the database. 5. The Traveler accesses the personal account.
Post Condition(s)	Login successful
Exception(s)	Error Message is displayed if the credentials are verified as not valid and allow User to enter them again.

Use case ID	UC-004
Name	Earn Points
Actor(s)	Traveler
Description	The Traveler will earn points based on the number of miles travelled with the previous bookings.
Precondition(s)	The Traveler must login into the personal account.
Basic Flow	<ol style="list-style-type: none"> 1. The Traveler made a successful booking with the system through logging into personal account. 2. The system calculates the miles involved in the journey of that booking. 3. The number of miles will be rewarded as points earned for the Traveler.
Post Condition(s)	Earned points will be displayed.
Exception(s)	NA

Use case ID	UC-005
Name	Search Flights
Actor(s)	Traveler
Description	The Traveler search for the flight as per the Traveler's requirements.
Precondition(s)	The Traveler should launch the system and be in the main page.
Basic Flow	<ol style="list-style-type: none"> 1. Traveler clicks search flights. 2. System displays the search flights with below search options <ul style="list-style-type: none"> • One way or Round trip • Domestic or International • Source location • Destination Location • Date of travel • No of Passengers • Type of Passengers(Adult/kids) 3. Traveler enters all the options. 4. System retrieves the search results.
Post Condition(s)	Retrieved flights are displayed.
Exception(s)	Error Message is displayed if any of the mandatory information is not entered.

Use case ID	UC-006
Name	Book Flights
Actor(s)	Traveler
Description	The Traveler books the flight.
Precondition(s)	Search flights results are available.
Basic Flow	<ol style="list-style-type: none"> 1. The Traveler selects a flight from the list of results. 2. The System displays the selected flight and asks for confirmation. 3. The Traveler confirms the booking.
Post Condition(s)	Book flight successful and moves to Payment section.
Exception(s)	Error Message is displayed if the seats are not available in the requested flight.

Use case ID	UC-007
Name	Make payment
Actor(s)	Traveler
Description	The Traveler makes a payment in order to confirm the flight booking.
Precondition(s)	Traveler confirms booking of a flight.
Basic Flow	<ol style="list-style-type: none"> 1. The Traveler selects the make payment option. 2. The System shall display two modes of Payment namely pay by Credit/Debit card /Interac transfer or Pay by points earned. 3. The Traveler selects either of the modes of payment.
Post Condition(s)	The System redirects to the selected mode of payment.
Exception(s)	NA

Use case ID	UC-008
Name	Payment via Credit/Debit card /Interac transfer
Actor(s)	Traveler
Description	The payment is processed with the Credit/Debit card/ Interac Transfer.
Precondition(s)	The Traveler should have selected Credit/Debit card/Interac Transfer mode in Make payment page.
Basic Flow	<ol style="list-style-type: none"> 1. The system asks the credentials for the payment. 2. The Traveler enters the details of Credit/Debit card/Interac transfer. 3. The System request for confirmation. 4. The Traveler confirms the payment.
Post Condition(s)	System redirects the payment details for validation.
Exception(s)	Error Message is displayed if any of the mandatory information not entered.

Use case ID	UC-009
Name	Payment via Points
Actor(s)	Traveler
Description	The payment is processed with the earned points.
Precondition(s)	The Traveler should be logged in and have points in the account.
Basic Flow	<ol style="list-style-type: none"> 1. The system asks for confirming payment using the points. 2. The Traveler confirms the payment using the points.
Post Condition(s)	Payment via points successful and Flight booking done.
Exception(s)	Error Message is displayed when the Traveler does not have enough points in the account.

Use case ID	UC-010
Name	Validate payment
Actor(s)	Bank
Description	The Bank validates the payment initiated by the Traveler and then confirms the booking.
Precondition(s)	The Traveler enters the Credit/Debit card /Interac transfer details and confirms the payment.
Basic Flow	<ol style="list-style-type: none"> 1. The System sends the entered payment details to the Bank. 2. The Bank verifies the details with the database and approves the transaction.
Post Condition(s)	Validation Successful and Flight booking done.
Exception(s)	<p>Error Message is displayed if the entered details do not match with the database.</p> <p>Error Message is displayed if the account does not have enough balance.</p>

Use case ID	UC-011
Name	Cancel Booking
Actor(s)	Traveler
Description	A Traveler wants to cancel a flight booking.
Precondition(s)	Flight must be booked and a booking id available.
Basic Flow	<ol style="list-style-type: none"> 1. Traveler clicks cancel booking option from main Page. 2. System asks to enter booking id. 3. Traveler enters correct booking id and clicks submit. 4. System asks “Do you wish to confirm cancellation”. 5. Traveler confirms Cancellation. 6. System displays “booking id Successfully canceled” and moves to Main Page.
Post Condition(s)	Booking is canceled and booking id added to cancelled database and system moves to main page.
Exception(s)	Error Message is displayed if the entered booking id is wrong or not entered.

Use case ID	UC-012
Name	Modify Booking
Actor(s)	Traveler
Description	A Traveler wants to modify a flight booking.
Precondition(s)	Flight must be booked and a booking id available.
Basic flow	<ol style="list-style-type: none"> 1. Traveler clicks modify booking option from main Page. 2. System asks to enter booking id. 3. Traveler enters correct booking id and clicks submit. 4. System displays main modify page with two options Add baggage and change seats. 5. Traveler selects Add baggage. 6. System displays add 5kgs, 10kgs and 20kgs baggage. 7. Traveler selects one of the baggage option and clicks submit. 8. System comes to main modify page showing added baggage details. 9. Traveler selects Change seats. 10. System shows the available seats. 11. Traveler selects the seat and click submit. 12. System comes back to main page showing added baggage details and changed seat details. 13. Traveler clicks submit. 14. System moves to UC-009 and return outcome is successful. 15. System returns back to Modify booking page displaying Payment successful and Modify booking successful and moves to main page.
Post Condition(s)	Booking is modified for extra baggage and seats been changed and moves to main page.
Exception(s)	Error Message is displayed if the entered booking id is wrong or not entered.

3. System Requirements

This section defines the System requirements in two categories such as Functional and Non-Functional requirements.

3.1 Functional Requirements

Sl. No	Requirement ID	Description
1	FR001	The System should support User login.
2	FR002	The System should support to create a personal account in the system.
3	FR003	The system should support to manage personal account information.
4	FR004	The System should store all the data about User login credentials.
5	FR005	The System should support update flight database.
6	FR006	The System should require special access to update Flight Database.
7	FR007	The System should support add and drop flights in Update Flight DB.
8	FR008	The System should store all the flights information and User details in database.
9	FR009	The System should support search flights with the search criteria.
10	FR010	The System should support flight booking by selecting from the search results.
11	FR011	The System should support to add earn points based on Flight booking.
12	FR012	The System should support payment for flight booking.
13	FR013	The System should support Payment via Credit/Debit card / Interac Transfer.
14	FR014	The System should support Payment via points.
15	FR015	The System should support to modify the booking.
16	FR016	The System should support add baggage and change seats in modify booking.
17	FR017	The System should support to Cancel the booking.

3.2 Non-Functional Requirements

Sl. No	Requirement ID	Description
Usability Requirements		
1	NFR001	The System should be User friendly and strive for effective User Interaction by generating alerts and allowing User to avoid common errors.
2	NFR002	The System shall show the available seats with a pictorial view of the flight in Change seat section of modify booking.
3	NFR003	The System shall have drop down section for Add baggage section of modify booking.
4	NFR004	The System shall take only alphabet and numbers in the booking id field.
5	NFR005	The System shall take maximum of 15 characters in the booking id field.
Performance Requirements		
6	NFR006	The system should support parallel processing for serving various User's requests.
7	NFR007	The System should be accessed from anywhere and at any time.
Security Requirements		
8	NFR008	The System should only provide access to Administrator to update the flight database.
9	NFR009	The System should provide standard security for performing the payment transfers.
Legal Requirements		
10	NFR010	The System should satisfy with the certified Quality Assurance standards.
11	NFR011	The System should protect and never disclose the User personal information by any means.

Glossary

Keyword	Description
Traveler	The Traveler acts on behalf of the person using the system and the member using the system with personal account in it.
Administrator	The Administrator comprises of the System-side handler and the FlyinTravel service executive who lookup after the System and Database.
User	The User resembles both the Traveler and Administrator into consideration with System.
Bank	The Bank is stated as the security based Bank network system that is integrated to the FlyinTravel Database for purpose of approving payments.

References

1. Papyrus Use case-
https://wiki.eclipse.org/Papyrus_for_Education#Use_Case_Diagram
2. IEEE format-
[http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=502838&filter=AND\(p_Publication_Number:3731\)](http://ieeexplore.ieee.org/xpl/articleDetails.jsp?arnumber=502838&filter=AND(p_Publication_Number:3731))

Appendix

1. Use Case Diagram UML file

[usecasediagram_new.rar](#)