Università degli studi di Salerno

***Corso di Laurea in Informatica***

******

***INGEGNERIA DEL SOFTWARE***

***Requirement Analysis Document***

***“UNI-AirLines”***

# Studenti:

##### Nome Matricola

Santoro Mario 0512104850

## Marino Raffaele 0512104508

Pastore Matteo 0512104724

Fortunato Angelo 0512104532

*Anno Accademico: 2018/19*

|  |  |
| --- | --- |
| ***SOMMARIO*** 1. INTRODUCTION........................................................... 3 2. CURRENT SYSTEM……................................................ 3 3. PROPOSED SYSTEM.................................................... 3 3.1 Overwiev……………………….………………………………………………………….3  3.2Functional Requirem…………………………………………………………………3  3.3 Nonfunctional Requirements……………………………………………………..4  3.4 System Models………………………………………………………………………….4  3.4.1 Scenarios…………………………………………………………………………………………4  3.4.2 Use Case Model……………………………………………………………………………….6  3.4.3 Object Model…………………………………………………………………………………15  3.4.4 Dynamic Model  3.4.5 user interface- navigational paths and screen mock-up  **4. GLOSSARY** | ***SSSS*** |
|  |  |

1. **Introduction**

Lo scopo del progetto Uni-Airlines è quello di sviluppare un software per l’acquisto di biglietti aerei online chiamato UnisAir. Il software dovrebbe essere semplice, veloce ed economico.

Un’Europa sempre più a portata di clic, dove per organizzare un viaggio non si va più in agenzia o in biglietteria ma si fa tutto da casa, dietro allo schermo di un computer.

L’obiettivo di questo progetto permette alla compagnia UniAirlines di interfacciarsi con l'utente tramite un sito web dedicato, in modo tale da permettere l'acquisto del biglietto comodamente da casa, richiedere assistenza, news e offerte via e-mail.

1. **Current system**

Attualmente l'agenzia è sprovvista di sito web il che comporta la poca interattività con i clienti, disagi nell'acquisto dei biglietti dovuti a tempi di attesa troppo elevati, mancanza di informazioni dei relativi voli.

Internet ha offerto la possibilità di acquistare biglietti aerei di una determinata compagnia online.

Non avere un software che rende disponibili questi servizi comporta una perdita economica enorme siccome le compagnie aeree concorrenti ne sono tutte provviste.

**3. Proposed system**

**3.1 Overwiev**

UnisAir consiste nel comprare biglietti aerei online solo se l’utente è registrato, possibilità di vedere la time label dei voli, check-in online, tariffe differenti a seconda dell'offerta scelta (classe di viaggio e peso del bagaglio), possibilità di scegliere più tratte con possibili scali, visionare le news, supporto.

**3.2 Functional Requirements**

Il sistema UnisAir dovrebbe fornire le seguenti funzionalità:

1. Un utente deve avere la possibilità di registrarsi: non è obbligatoria per visualizzare le tratte aeree ma necessaria per procedere all’acquisto;
2. Un utente deve avere la possibilità di controllare la time label dei voli;
3. Un utente deve avere la possibilità di effetuare il check-in online: per evitare la coda in aereoporto può farlo comodamente da casa a partire da 3 giorni prima del volo;
4. Un utente deve avere la possibilità di visionare le news sul sito: dove può trovare voli in offerta e notizie di ogni genere;
5. Un utente deve avere la possibilità di ricevere supporto e-mail: tramite un sito esterno l’utente riceve email di offerta e assistenza;
6. Un utente ha la possibilità di imbarcare fino a un massimo di 3 bagagli da stiva, a ogni bagaglio va applicata la tariffa relativa al suo peso;
7. Un utente ha la possibilità di vedere la lista dei voli prenotati;
8. Un utente può prenotare un volo: solo se ha fatto il login, ed entro 6 ore prima del volo.

**3.3 Nonfunctional Requirements**

1. Massimo 30 secondi di attesa nell’interazione con il sistema quando si acquista un biglietto [Performance requirement];
2. Il sito deve essere disponibile 24 ore al giorno [Performance requirement];
3. Un sito web-based indipendente dalla piattaforma [Supportability requirement];
4. Manutenzione da parte dell’admin [Supportability requirement];
5. Uno schema di controllo degli accessi che impedisce l'accesso non autorizzato e consente l'accesso per più utenti [Reliability requirement];
6. In caso di eccezioni o perdita di connessione il sistema fa visualizzare all’utente una pagina di errore [Reliability requirement] ;
7. Interfaccia utente semplice da usare [Interface requirement];
8. In caso di problemi all’utente vengono fornite le informazioni per contattare l’assistenza[Usability requirement];
9. Non sono necessari installazioni, il sistema è completamente sul web[Packing requirement];

10.Non ha costi di utilizzo e non sono previsti diritti d’autore o di componenti [Legal requirment];

11. Tutta l’applicazione deve essere scritta in Java . [Implementation requirement].

## 3.4. System models

### *3.4.1* Scenarios

**3.4.1.1 Prenota Volo**

Raffaele vuole acquistare biglietti di volo per lui e la sua famiglia della compagnia aerea UnisAir, quindi si reca sulla homePage del sito e compila l’apposita form di ricerca composta dai seguenti campi :

1. Aeroporto di partenza: Aereoporto Internazionale di Napoli(Capodichino);

2. Aeroporto di destinazione: Aereoporto Internazionale di Amsterdam;

3. Spunta la checkbox volo andata/ritorno;

4. Data di partenza: 20 dicembre 2018;

5. Data di ritorno: 28 dicembre 2018;

6. Numero adulti: 2;

7. Numero bambini: 1;

8. Numero neonati: 0.

Una volta compilati i campi sopra citati, clicca l’apposito bottone per effettuare la ricerca.

Viene reindirizzato ad una pagina contenente i risultati della ricerca da lui voluta, organizzati in ordine di orario di partenza dall’aeroporto da lui inserito, con relativo costo ed orario di arrivo.

Dopo aver scelto il volo di andata vengono visualizzati anche i risultati dei voli di ritorno per la data da lui inserita, in ordine di orario di partenza con i relativi costi.

Raffaele una volta trovato il volo da lui desiderato, che soddisfa i suoi criteri di orario e costo del volo clicca sul bottone per fare il login, obbligatorio per completare l’acquisto, inserendo email e password.

Subito dopo il sistema fornisce a Raffaele diversi servizi per la classe di volo e per l’imbarco del bagaglio.

Quest’ultimo sceglie i loro posti nella schermata che gli compare, selezionando i posti di classe premium P1, P2, P3, in seguito sceglie bagaglio standard per tutta la sua famiglia, ovvero meno di 15kg di peso il cui trasporto è gratuito.

Ora Raffaele deve inserire:

1. Nome;

2. Cognome;

3. Età;

4. Indirizzo;

5. Città;

6. Codice postale;

7. Paese;

8. Tipo documento;

9. Numero documento.

Per lui e tutta la sua famiglia, compilando quindi 3 form.

Dopodiché sceglie la carta di pagamento:

1. Numero carta di pagamento;

2. Nome proprietario;

3. Data di scadenza;

4. CVC/CVV;

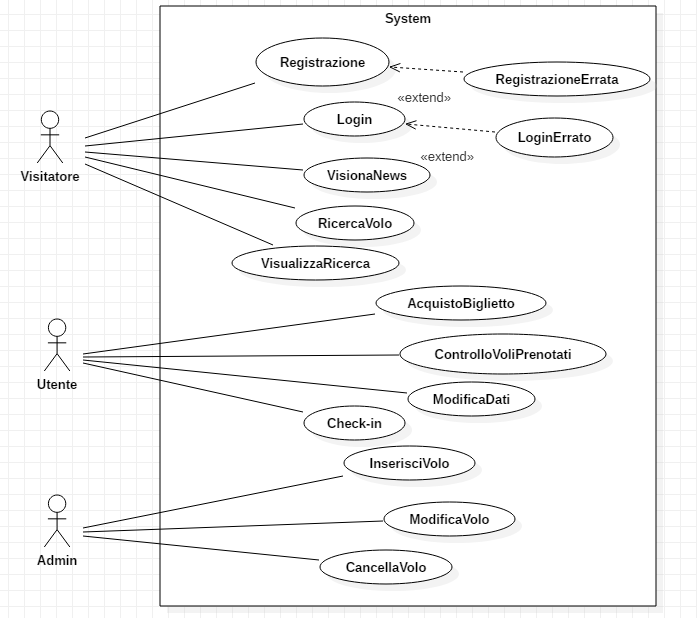
5. Checkbox termini e condizioni.

Raffaele invia le informazioni (ricevendo poi un email di conferma che contiene anche il codice del volo) e viene reindirizzato alla home page.

**3.4.1.2 Check-In**

Raffaele intende effettuare il Check-In della sua prenotazione poiché mancano 3 giorni alla sua partenza. Nella Home-Page del sito preme sul bottone VoliPrenotati. Gli appare la lista dei voli da lui prenotati, in questo caso solo uno, Roma-Parigi e cliccando gli escono tutte le informazioni relative al volo e un bottone di check-in. Gli compare così, una lista di checkbox con i nomi : "Raffele Marino", "Roberta Rossi", "Francesca Rossi". Raffaele spunta tutti i checkbox corrispondenti alle persone per cui ha prenotato il biglietto, dopodiché preme su conferma. A questo punto gli compaiono le carte d'imbarco da stampare, corrispondenti alle sue prenotazioni.

**3.4.2 Use case model**

****

**3.4.2.1 Actors**

Abbiamo identificato i seguenti attori:

* *Visitatore*: esegue le funzioni principali del sito interagendo con il software, però non può acquistare voli;
* *Utente*: esegue le funzioni principali del sito procedendo nell’acquisto del biglietto, controllo voli prenotati e check-in.
* *Gestore:* esegue le funzioni secondarie del sito come manutenzione e amministrazione.

**3.4.2.2 Login**

*Use case name* Login

*Participating actors* Initiated by Visitatore

*Flow of events*

1. il visitatore clicca sul bottone di Login presente sulla Home Page.

2. il visitatore deve compilare una form di Login inserendo Email e Password (se non è un utente ancora registrato può premere il link sottostante che lo rimanda nell’apposita form vedi caso d’uso “Registrazione”), una volta compilata l'utente invia i dati tramite apposito bottone.

3. il sistema controlla i dati, se i dati non sono corretti vedi use case “LoginErrato”.

*Entry condition*

•E’ un utente registrato.

*Exit conditions*

• L'utente è reindirizzato alla Home Page dopo un messaggio di Login effettuato con successo.

*Quality requirements*

• Il completamento della procedura di Login deve essere effettuato in massimo 30 secondi.

**3.4.2.3 Registrazione**

*Use case name* Registrazione

*Participating actors* Initiated by Visitatore

*Flow of events*

1. il visitatore clicca sul bottone di Registrazione.

2. il visitatore compila il form di Registrazione inserendo nome, cognome, tipo documento, numero documento, e-mail e password.

Una volta compilata la form di Registrazione il visitatore invia i dati tramite apposito bottone.

4.il sistema controlla i dati inseriti nella form, in caso di inserimenti non corretto vedere use case “RegistrazioneErrata”.

*Entry condition*

*Exit conditions*

• L'utente è reindirizzato alla Home Page dopo un messaggio di Registrazione effettuata con successo.

*Quality requirements*

• Il completamento della procedura di Registrazione deve essere effettuato in massimo 30 secondi.

**3.4.2.4 Inserimento Volo**

*Use case name* InserimentoVolo

*Participating actors* Initiated by Gestore

*Flow of events*

* 1. Il gestore preme sul bottone per l’inserimento volo
  2. Viene rimandato su una pagina dove inserisce le informazioni del volo:
* Aeroporto di partenza;
* Aeroporto di destinazione;
* Data;
* Ora di partenza;
* Ora di arrivo;
* Prezzo per classe;
* Check-box con gli aerei liberi da assegnare a quel volo.
  1. Il gestore preme il bottone d’inserimento e se ha inserito tutte le credenziali nel modo corretto le informazioni vengono salvate nel sistema.

*Entry condition*

L’utente loggato è un gestore.

*Exit conditions*

• il gestore è reindirizzato al pannello di controllo di gestione dopo un messaggio di inserimento volo effettuato con successo.

*Quality requirements*

• Il completamento della procedura di inserimento deve essere effettuato in massimo 30 secondi.

**3.4.2.5 Modifica Volo**

*Use case name* ModificaVolo

*Participating actors* Initiated by Gestore

*Flow of events*

* + 1. Il gestore visualizza tutta la lista dei voli e preme sul bottone per la modifica volo

2.Viene rimandato su una pagina dove inserisce le nuove informazioni del volo:

* Aeroporto di partenza;
* Aeroporto di destinazione;
* Data;
* Ora di partenza;
* Ora di arrivo;
* Prezzo per classe;
* Check-box con gli aerei liberi da assegnare a quel volo.

1. L’admin preme il bottone di modifica e se ha inserito tutte le credenziali nel modo corretto le informazioni vengono salvate nel sistema.
2. Alla modifica del volo viene inviata un email di avviso a tutti gli utenti che avevano prenotato quel volo, e nel caso in cui il volo non soddisfa più il cliente, quest’ultimo potrà contattare l’assistenza per avere il rimborso del volo.

*Entry condition*

L’utente loggato è un gestore.

*Exit conditions*

• il gestore è reindirizzato al pannello di controllo di gestione dopo un messaggio di modifica volo effettuata con successo.

*Quality requirements*

• Il completamento della procedura di inserimento deve essere effettuato in massimo 30 secondi.

**3.4.2.6 Cancella Volo**

*Use case name* CancellaVolo

*Participating actors* Initiated by Gestore

*Flow of events*

* + - 1. Il gestore visualizza tutta la lista dei voli e preme sul bottone per la cancellazione volo che sarà disponibile solo al massimo fino a 3 giorni prima del volo;
      2. Una volta premuto il volo scomparirà dalla lista dei voli;
      3. Alla cancellazione del volo viene inviata un email di avviso a tutti gli utenti che avevano prenotato quel volo e verranno direttamente rimborsati con un accredito sulla carta di pagamento inserita alla prenotazione.

*Entry condition*

L’utente loggato è un gestore.

*Exit conditions*

• il gestore rimane sul pannello di controllo ma con la lista dei voli aggiornata.

*Quality requirements*

• Il completamento della procedura di cancellazione deve essere effettuato in massimo 15 secondi.

**3.4.2.7 Login non riuscito**

*Use case name* LoginErrato

*Participating actors* Initiated by Visitatore

*Flow of events*

1. il visitatore clicca sul bottone di Login presente sulla Home Page.

2. il visitatore compila la form di Login inserendo Email e Password, una volta compilata l'utente invia i dati tramite apposito bottone.

3. Il sistema controlla i dati.

4. Il sistema non riconosce la combinazione Email-Password poiché una delle due, o entrambe, sono sbagliate o inesistenti.

*Entry condition*

*Exit conditions*

• il visitatore è reindirizzato alla pagina di Login dopo un messaggio di Login non riuscito.

*Quality requirements*

• Il completamento della procedura di Login deve essere effettuato in massimo 30 secondi.

**3.4.2.8 Registrazione non riuscita**

*Use case name* RegistrazioneErrata

*Participating actors* Initiated by Visitatore

*Flow of events*

1. il visitatore clicca sul bottone di Registrazione.

2. il visitatore compila il form di Registrazione inserendo nome, cognome, tipo documento, numero documento, e-mail e password.

3. Una volta compilata la form di Registrazione il visitatore invia i dati tramite apposito bottone.

4. Il sistema controlla i dati inseriti nella form.

5. Il sistema rileva che il Numero documento o l'e-mail sono già presenti.

*Entry condition*

*Exit conditions*

• il visitatore è reindirizzato alla pagina di registrazione dopo un messaggio di Registrazione non riuscita.

*Quality requirements*

• Il completamento della procedura di Registrazione deve essere effettuato in massimo 30 secondi.

**3.4.2.9 utente fa il check-in online**

*Use case name* Check-in

*Participating actors* Initiated by Utente

*Flow of events*

* + - 1. L’utente seleziona il volo di cui vorrebbe effettuare il check-in;
      2. L’utente passa ad una pagina di info dove il bottone di check-in sarà disponibile solo se la data corrente rientra nei parametri della prenotazione;
      3. L’utente clicca sull’apposito bottone per fare il check-in;

4. Dopodichè visualizza tutti i passeggeri della sua prenotazione e spunta la check-box di tutti i passeggeri che vogliono effettuare il check-in online e preme il bottone di conferma;

*Entry condition*

E’ un utente con almeno un acquisto, si trova nella sezione “voli prenotati”, vedere use case “ControlloVoliPrenotati”, la sezione check-in è disponibile a partire da 3 giorni prima del volo .

*Exit conditions*

• l’utente visualizza le carte d’imbarco da stampare.

*Quality requirements*

• la visualizzazione (e consecutiva stampa) del biglietto viene effettuata in massimo 15 secondi.

**3.4.2.10 utente controlla i suoi voli prenotati**

*Use case name* ControlloVoliPrenotati

*Participating actors* Initiated by Utente

*Flow of events*

* + - 1. L’utente entra nel suo profilo cliccando l’apposito bottone;

1. nel menù a tendina troverà la sezione “voli prenotati”.
2. Cliccato il bottone l’utente avrà avanti una pagina dove sono visibili i voli prenotati.

*Entry condition*

E’ un utente con almeno un volo prenotato.

*Exit conditions*

• l’utente riesce a visualizzare i suoi voli prenotati.

*Quality requirements*

• la visualizzazione dei voli viene effettuata in massimo 15 secondi.

**3.4.2.11 Visitatore ricerca volo**

*Use case name* RicercaVolo

*Participating actors* Initiated by Visitatore

*Flow of events*

1.il visitatore accede alla pagina principale dove si ritrova i campi da compilare per effettuare una ricerca:

1.aeroporto di partenza;

2.aeroporto di destinazione;

3.data di partenza;

4.checkbox che conferma di cercare anche il volo di ritorno;

5.data di ritorno;

6.numero adulti;

Una volta compilati correttamente i campi il visitatore cliccherà il bottone di invio il sistema ritorna i voli richiesti dall’utente.

*Entry condition*

*Il visitatore si trova sulla home page*

*Exit conditions*

• l’utente si trova nella pagina di visualizza ricerca.

*Quality requirements*

• la registrazione della prenotazione del volo viene effettuata in massimo 15 secondi.

**3.4.2.12 Visitatore visualizza ricerca**

1. terminata la ricerca da parte del sistema il visitatore avrà avanti una serie di voli che avranno data di partenza, aeroporto di partenza ed aeroporto di arrivo, corrispondenti a quelli inseriti dall’utente nei campi di ricerca.

2.trovato il volo che soddisfa il visitatore, se vuole procedere con l’acquisto egli cliccherà sul bottone per iniziare la procedura di prenotazione. VEDI USE CASE “AcquistaBiglietto”.

*Entry condition*

*L’utente caso d’uso “ricerca volo”*

*Exit conditions*

• l’utente inizia la procedura di prenotazione.

*Quality requirements*

• la registrazione della prenotazione del volo viene effettuata in massimo 15 secondi.

**3.4.2.13 Visitatore prenota un volo tramite offerta dal sito**

*Use case name* VisionaNews

*Participating actors* Initiated by Visitatore

*Flow of events*

1.il visitatore accede alla pagina news dove si ritrova una serie di offerte per dei voli. Trovato un volo che lo soddisfa clicca sul relativo annuncio;

2.ora il visitatore si ritrova avanti la schermata dove viene visualizzata l’offerta da lui selezionata, con tutti i dettagli del volo in questione:

1.aeroporto di partenza;

2.aeroporto di destinazione;

3.data di partenza;

4.checkbox che conferma di cercare anche il volo di ritorno;

5.data di ritorno;

6.numero adulti;

Se soddisfatto della sua scelta l’utente clicca il bottone per iniziare la prenotazione. VEDI USE CASE “AcquistaBiglietto”.

*Entry condition*

*Il visitatore si trova nella home page*

*Exit conditions*

• il visitatore inizia la procedura di prenotazione.

*Quality requirements*

• la registrazione della prenotazione del volo viene effettuata in massimo 15 secondi.

**3.4.2.14 utente acquista biglietto**

*Use case name* AcquistaBiglietto

*Participating actors* Initiated by Utente

*Flow of events*

1. Il sistema verificherà se l’utente è loggato (azione necessaria per proseguire) in caso non sia loggato vedere scenario “Login”
2. L’utente si troverà davanti una schermata che gli permetterà di selezionare il posto(o posti) sull’aereo; tali posti sono suddivisi per classi(Premium, Business e Economy) e in base a dove si trova il posto scelto tra le classi può essere applicato un sovraprezzo sul costo del biglietto.
3. A questo punto dovrà selezionare tramite alcuni checkbox il peso del bagaglio che vorrà portare con se. Anche in questo caso verranno applicati dei sovraprezzi all’aumentare del peso del bagaglio.
4. L’utente inserirà i dati anagrafici di tutti i passeggeri:

1. Nome;

2. Cognome;

3. Età;

4. Indirizzo;

5. Città;

6. Codice postale;

7. Paese;

8. Tipo documento;

9. Numero documento.

1. La schermata che apparirà adesso all’utente gli permetterà di confermare il pagamento ,tramite i campi seguenti:

1. Numero carta di pagamento;

2. Nome proprietario;

3. Data di scadenza;

4. CVC/CVV;

1. Una volta fatto ciò l’utente termina la procedura di prenotazione e gli sarà recapitata un e-mail di conferma. Se la partenza del volo è programmata entro 3 giorni allora l’utente sarà abilitato a fare il check-in(vedi caso d’uso “Check-in”).

*Entry condition*

E’ un utente già loggato e deve aver già selezionato il volo tramite la procedura descritta nei precedenti casi d’uso(vedi casi d’uso “RicercaVolo” e “visionaNews”)

*Exit conditions*

•L’utente termina con successo la procedura di prenotazione.

*Quality requirements*

• La transizione e la conferma della prenotazione del volo viene effettuata in massimo 30 secondi.

**3.4.2.15 utente modifica propri dati**

*Use Case Name* Modifica Dati Utente

*Partecipatinng actors*  Initiated by Utente

*Flow of events*

1. L’utente tramite un menù potrà selezionare l’opzione “Modifica i miei dati.
2. Verrà reindirizzato a una pagina che gli permetterà di modificare i suoi dati anagrafici(es. Residenza ecc.) e modificare i dati di pagamento nel caso in cui abbia già effettuato almeno un acquisto ed abbia salvato il metodo di pagamento durante l’ultimo acquisto effettuato.
3. L’utente terminerà la procedura cliccando l’apposito bottone di conferma.

*Entry Conditions*

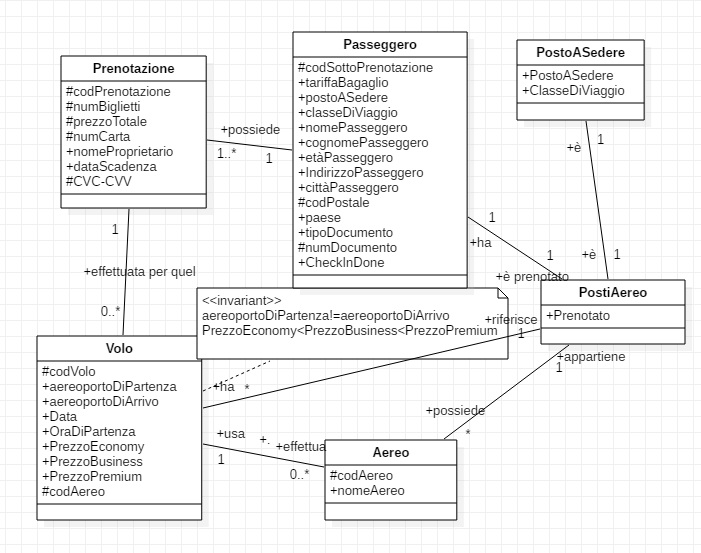
* L’utente deve aver già effettuato il Login

*Exit conditions*

* L’utente avrà aggiornato i propri dati.

*Quality requirements*

* L’aggiornamento dei dati viene effettuato in un massimo di 30 secondi.
  + 1. **Object Model**

****

**3.4.3.1 Prenotazione**

Prenotazione è una classe che dipende da Utente (che effettua la prenotazione), essa rappresenta la prenotazione del posto per un volo in un determinato aereo. È caratterizzato da il codice della prenotazione che differenzia ogni prenotazione, il numero di biglietti che identifica quanti biglietti sono stati prenotati per quel volo, il prezzo totale della prenotazione che è dato dal costo dei biglietti che dipende dalla classe di volo e dal bagaglio portato. Poi è presente il numero della carta di pagamento con il nome del proprietario data di scadenza e il CVV. Le operazioni di questa classe sono i getter e setter delle istanze.

**3.4.3.2 Passeggero**

La classe Passeggero è una classe che dipende dalla classe Prenotazione, essa ha come attributi il codice della prenotazione, la tariffa del biglietto data dalla classe di viaggio (definito in PostoASedere) i dati del passeggero, formati da: nome, cognome, età, indirizzo, città, CAP, paese e tipo di documento (che servono per il biglietto da stampare , che verrà associato ad una ed una sola persona), relativo al biglietto, ed una variabile booleana che ci dice se è stato effettuato o no il Check-In. Le operazioni di questa classe sono i getter e setter delle istanze.

**3.4.3.3 PostoASedere**

PostoASedere è una classe che (serve a definire PostiAereo) identifica il numero esatto di posti a sedere di tutti gli aerei e anche la loro classe di viaggio. Le operazioni di questa classe sono i getter e setter delle istanze.

**3.4.3.4 PostiAereo**

PostiAereo serve a identificare quali posti di quel determinato aereo,

riferito a quel determinato volo, sono prenotati. Ha una sola istanza

booleana che identifica se quel posto è prenotato. Le operazioni sono i

getter e setter dell’attributo.

**3.4.3.5 Volo**

Volo è una classe che dipende dal gestore che memorizza le nformazioni. Queste informazioni definiscono il volo:

CodVolo, aereoporto di partenza, aereoporto di destinazione,

Data, Ora di partenza, prezzo economy, prezzo business,

prezzo premium e il codice aereo.

I prezzi economy, business e premium sono diversi siccome

la qualità e i servizi sono differenti (i prezzi sono:

economy<business<premium), il codice aereo definisce

l’aereo che verrà usato in quel volo, selezionato tra i possibili

aerei della compagnia se disponibili per fare quella tratta in quella determinata data e ora, gli aeroporti di partenza e destinazione devono essere ovviamente diversi.

Le operazioni di Volo sono i getter e i setter degli attributi.

**3.4.3.6 Aereo**

La classe aereo viene associato al volo, in quanto per quel determinato volo verrà usato quell’aereo, i suoi attributi sono codice aereo e nome aereo. Le operazioni sono i getter e setter degli attributi.