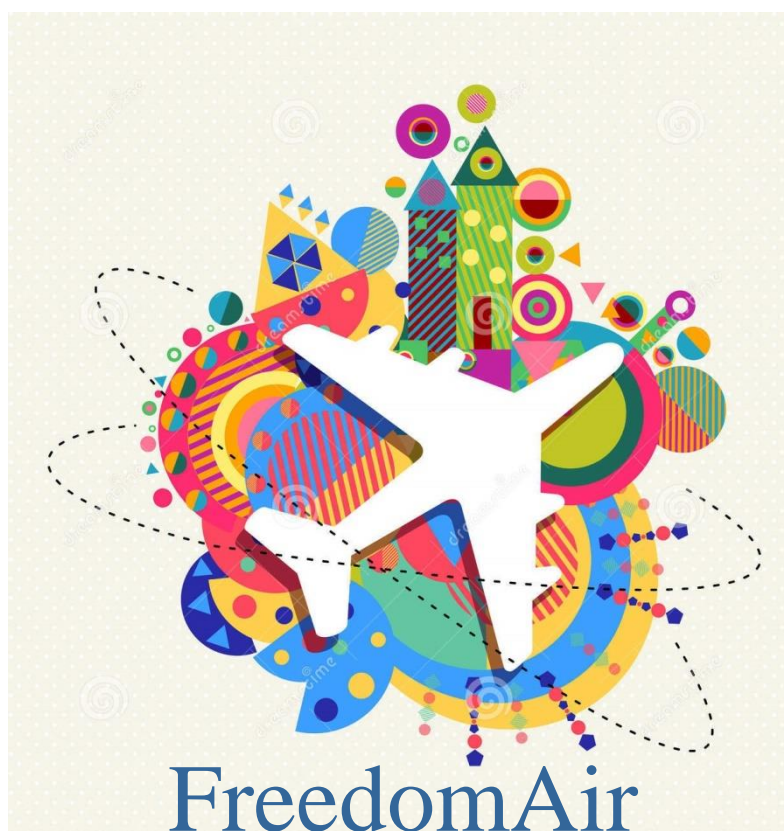


# Università degli Studi di Salerno

Corso di Ingegneria del Software

**FreedomAir**  
**SDD**  
**Versione 0.5**



Data: 29/11/2017

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

**Partecipanti:**

Nome	Matricola
Ferraioli Rosa	0512103450
Alfonso Del Gaizo	0512103462

<b>Scritto da:</b>	Ferraioli Rosa, Alfonso Del Gaizo
--------------------	-----------------------------------

## Revision History

Data	Versione	Descrizione	Autore
21/11/2017	0.1	Paragrafo 1	Ferraioli Rosa
22/11/2017	0.2	Paragrafo 2	Del Gaizo Alfonso
24/11/2017	0.3	Paragrafo 3 e 4	Ferraioli Rosa, Del Gaizo Alfonso
27/11/2017	0.4	Paragrafo 5 e schema EER con tabelle	Ferraioli Rosa, Del Gaizo Alfonso
29/11/2017	0.5	Fine SDD	Ferraioli Rosa, Del Gaizo Alfonso

## INDICE DEI CONTENUTI

1. Introduzione	
1.1 Scopo del sistema.....	3
1.2 Design Goals .....	3
1.3 Definizioni, Acronimi ed Abbreviazioni.....	6
1.4 Riferimenti.....	6
1.5 Visione generale.....	6
2. Sistema Software Corrente.....	7
3. Sistema Software Proposto.....	7
3.1 Visione generale.....	7
3.2 Decomposizione in sottosistemi.....	7
3.3 Hardware/Software mapping.....	10
3.4 Controllo degli Accessi e Sicurezza.....	11
3.5 Controllo Globale del Software.....	12
3.6 Condizione limite .....	12
3.7. Start-up.....	12
4 Servizi dei Sottosistemi.....	13
5. Glossario.....	16

## 1. Introduzione

### 1.1 Scopo del sistema

Con il seguente progetto ci apprestiamo ad analizzare e gestire l'insieme delle problematiche relative all'amministrazione di un aeroporto. Tale attività è suddivisa in tre aree fondamentali, una prima riguardante l'amministrazione delle risorse, la seconda che riguarda il promoter ossia colui che si occupa della sponsorizzazione del sistema e la terza area riguarda l'interazione tra il cliente e il sistema.

L'obiettivo generale del progetto FreedomAir è, dunque, quello di implementare un sistema software semplice da utilizzare e che permetta l'acquisto di un biglietto senza recarsi in un negozio abilitati.

### 1.2 Design Goals

Abbiamo diverse categorie di design goals:

- Criteri di performance
- Criteri di affidabilità
- Criteri di costo
- Criteri di manutenzione
- Criteri utenti finali

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

## Criteri di performance (Performance Criteria)

**Tempo di Risposta:** A qualsiasi sollecitazione fornita dagli attori, il sistema deve rispondere entro 30 secondi oppure in caso di errore presentare un messaggio di errore.

**Memoria:** La dimensione complessiva dipende dalla memoria utilizzata per il mantenimento del database. Verrà utilizzato il programma MySQL; in presenza di grandi quantità di dati, non vi sarà un rallentamento del sistema.

## Criteri di affidabilità (Dependability Criteria)

**Robustezza:** Il sistema, nelle interazioni con gli attori, effettuerà dei controlli sui dati in input allo scopo di validarli. Se la validazione darà esito positivo i dati saranno successivamente elaborati; altrimenti verrà mostrato un messaggio di errore.

**Tolleranza ai guasti:** In caso di errore, il sistema non continuerà ad operare ma, mostrerà una notifica all'utente, ed eventualmente istruzioni su come proseguire. Inoltre i dati verranno salvati per evitare delle perdite.

**Sicurezza:** Nessun utente può vedere dati relativi ad altri utenti. Inoltre l'accesso avviene tramite un username, unico per ogni utente, e password. Inoltre dati sono resi sicuri, in quanto l'accesso al database è limitato solo ad alcuni operatori.

**Affidabilità:** Il sistema deve attenersi alle specifiche funzionali che sono state descritte nell'apposita documentazione (RAD).

## Criteri di mantenimento (Maintenance criteria)

**Tracciabilità dei requisiti:** La tracciabilità dei requisiti è possibile grazie ad una matrice di tracciabilità, attraverso la quale è possibile retrocedere al requisito associato ad ogni parte del progetto. La tracciabilità è garantita dalla fase di progettazione fino al testing.

**Portabilità:** Il sistema sarà portabile in quanto l'interazione avviene mediante un browser quindi non c'è una dipendenza dovuta al sistema operativo.

## Utente finale (End User Criteria)

**Usabilità:** Il sito sarà di facile utilizzo e User Friendly. Inoltre vi è un manuale utente che, fornirà un supporto cartaceo agli amministratori del sistema.

**Utilità:** Il sistema risponderà in modo esauritivo alla richieste degli utenti, che verranno supportati dal sistema.

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

## 1.3 Definizioni, Acronimi ed Abbreviazioni

**RAD** = Requirement Analysis Document.

**RNF** = Requisito Non Funzionale.

**SDD** = System Design Document.

**Mysql** = è il più diffuso database Open Source basato sul linguaggio SQL.

**FreedomAir** = il sistema di gestione dell' Aeroporto sviluppato..

**User-friendly** = software di facile utilizzo anche per chi non è esperto.

**UML** = è un metodo per descrivere l'architettura di un sistema in dettaglio.

**File** = contenitore di informazioni/dati in formato digitale.

## 1.4 Riferimenti

- Prentice Object Oriented Software Engineering Using UML Patterns and Java
- RAD\_Completo

## 1.5 Visione Generale

Nei successivi paragrafi del documento di System Design verranno descritti:

- **Sistema Software Corrente:** rappresenta l'architettura del software che è stato sostituito o, se esso non esiste, descrive l'architettura dei software per sistemi simili esistenti.
- **Sistema Software proposto:** documenta il modello di System Design del nuovo sistema in cui vengono descritti i seguenti elementi:
  - Decomposizione in sottosistemi, nella quale viene descritta la suddivisione del sistema in vari sottosistemi.
  - Hardware/Software mapping.
  - Gestione Dati Persistenti.
  - Controllo degli Accessi e Sicurezza.
  - Controllo Globale del Software.
  - Condizioni Boundary in cui sono descritte le condizioni limite del sistema.
- **Servizi dei Sottosistemi:** descrive i servizi forniti da ciascun sottosistema in termini di operazioni.

## 2. SISTEMA SOFTWARE CORRENTE

Attualmente esiste un sistema software che si occupa di gestire le richieste di un aeroporto tramite un sistema digitalizzato. Ma il nostro scopo è quello di creare un software che sia facile da utilizzare, dove chiunque può accedervi in completa sicurezza e affidabilità.

Si tratta, perciò, di un sistema che rientra nel campo della Re-Engineering.

Attraverso una Re-Engineering un progetto software viene sviluppato avendo come riferimento un

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

sistema esistente utilizzando nuove tecnologie.

### 3. SISTEMA SOFTWARE PROPOSTO

#### 3.1 Visione Generale

Il sistema che si vuole proporre è una applicazione web il cui obiettivo è fornire un supporto a tutte le figure che vengono coinvolte nella fase di acquisto di un biglietto online.

Il sistema fornirà al cliente una lista di tutti i voli possibili di una determinata tratta.

Un sistema di gestione del progetto Freedomair è la funzionalità principale del sistema, ciò permetterà al cliente di evitare di recarsi in un'agenzia per l'acquisto di un biglietto aereo, questo garantisce un minor spreco di tempo in quanto il cliente può effettuare tutto da casa. Per le restanti figure coinvolte (Amministratore/ Promoter) garantiscono la gestione e la visibilità necessaria al sistema. L'amministratore può gestire il sistema in quanto può gestire i voli presenti nel database mentre il promoter gestisce la parte di sponsorizzazione del sistema.

Il sistema, in generale, sarà organizzato secondo il criterio client/server: il client sarà adibito alla gestione della parte di presentazione e della parte direttamente connessa all'interfaccia grafica; il server sarà invece adibito alla gestione della parte logica relativa ai dati e del database all'interno del quale verranno resi persistenti i dati.

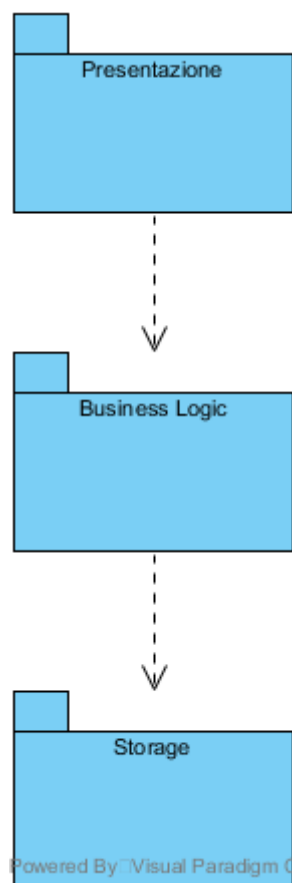
Per fornire una massima coesione e minimo accoppiamento tra i sottosistemi che comporranno il sistema proposto, le gestioni verranno individuate in base alle funzionalità. Ciò permetterà al sistema di essere flessibile nei cambiamenti: un cambiamento in un sottosistema non influenzerà gli altri. Le funzionalità saranno, infine, divise in layer logici distinti tra di loro in base al loro tipo: sarà presente il layer di presentazione, di business logic e di memorizzazione (storage).

#### 3.2 Decomposizione in sottosistemi

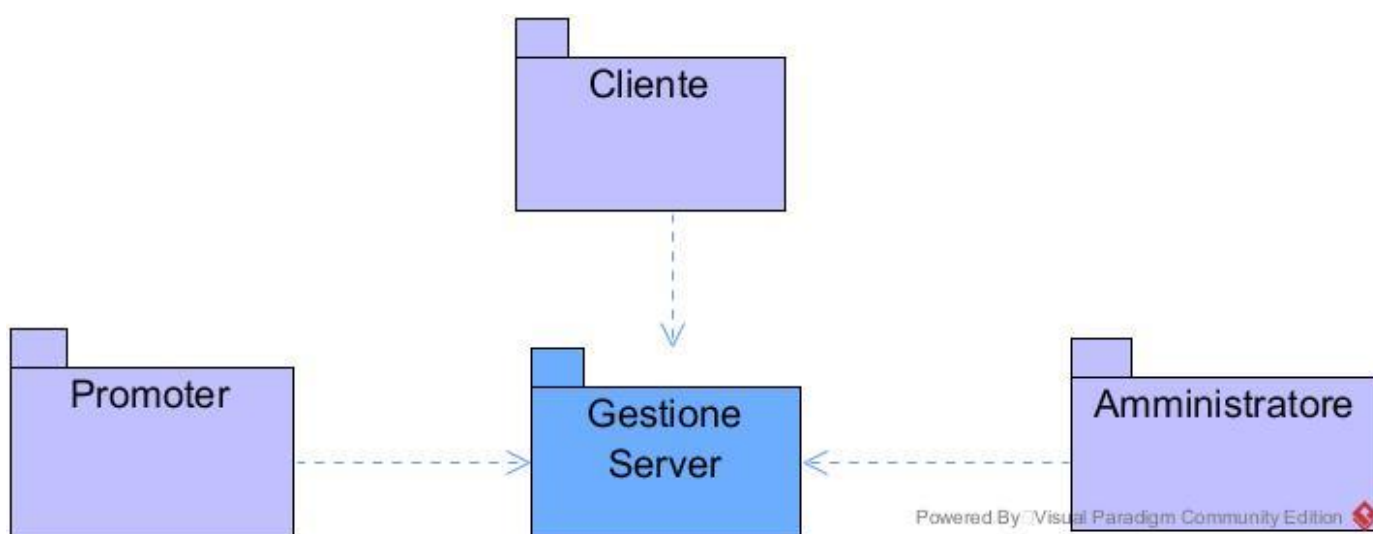
##### 3.2.1 Decomposizione in Layer

La decomposizione prevista per il sistema è composta da tre layer che si occupano di gestirne aspetti e funzionalità differenti:

- Presentazione: raccoglie e gestisce l'interfaccia grafica e gli eventi generati dall'utente;
- Business Logic: si occupa della gestione della logica del sistema;
- Storage: si occupa della gestione e dello scambio dei dati tra i sottosistemi;



### 3.2.2 Decomposizione in Sottosistemi

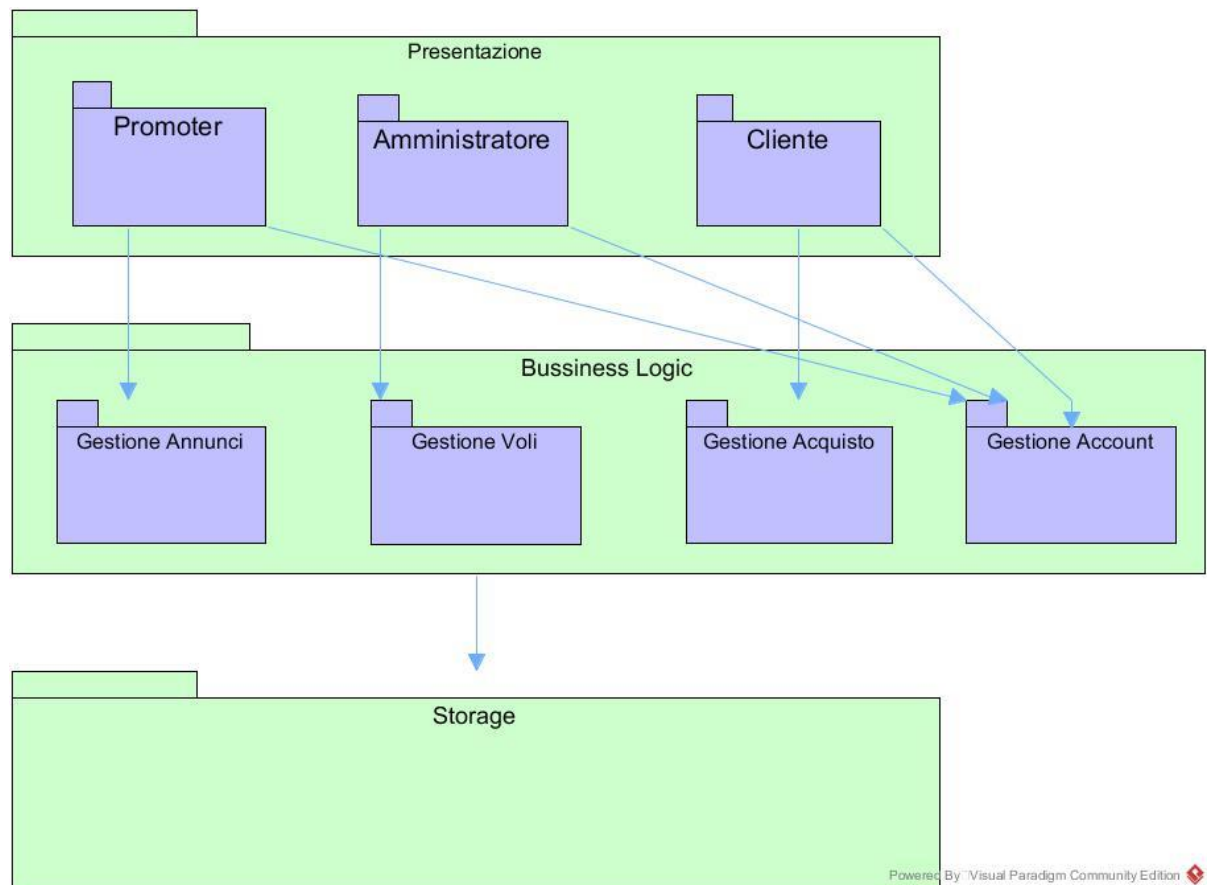


Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

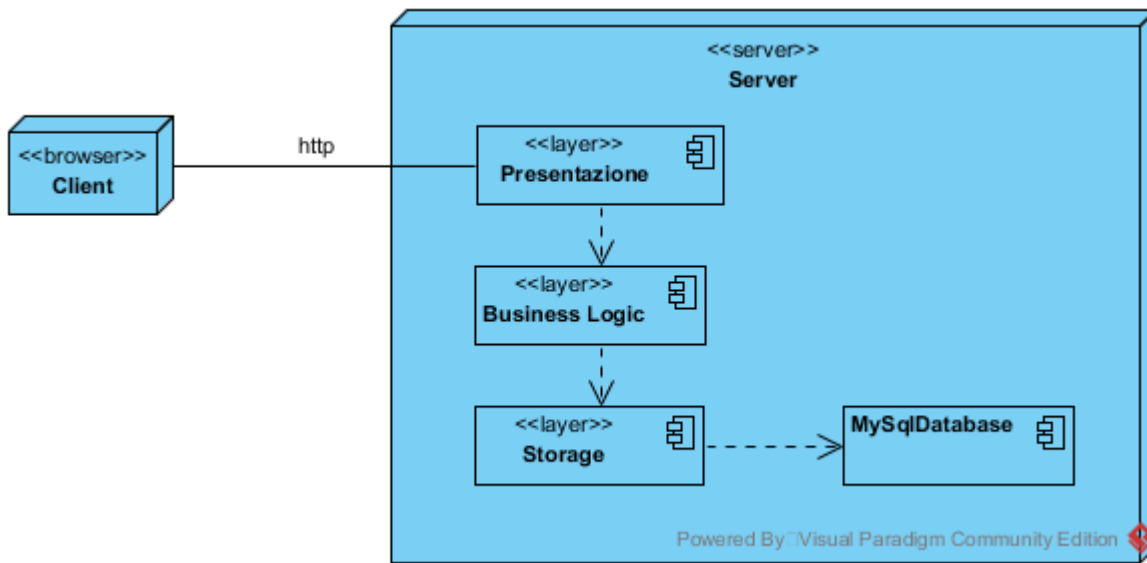
Il sistema è stato diviso in tre strati:

- Il livello di **Presentazione** è stato diviso in tre sottosistemi principali:
  - ❖ Le partizioni fanno riferimento ai tre principali attori che accedono al sistema:
    - Sottosistema **Cliente**: Include tutte le interfacce grafiche a cui il cliente può accedere, ovvero: registrazione, login, logout, visualizza voli, ricerca volo, acquisto biglietto, visualizza biglietto, modifica dati personali.
    - Sottosistema **Amministratore**: Include tutte le interfacce grafiche a cui l'Amministratore può accedere, ovvero: login, logout, modifica dei voli, visualizzazione del volo, Inserimento volo
    - Sottosistema **Promoter**: Include tutte le interfacce grafiche a cui il promoter può accedere, ovvero: login, logout, cancellazione annunci, visualizzazione degli annunci pubblicati, Inserimento annunci.
- Il livello di **Business Logic** è stato diviso in 4 sottosistemi principali.
  - ❖ Questi 4 sottosistemi rappresentano le quattro funzionalità base che svolgono gli attori:
    - Sottosistema **Gestione Voli**: In esso sono incluse tutte le operazioni che permettono di modificare, inserire o visualizzare i voli.
    - Sottosistema **Gestione Annunci**: In esso sono incluse tutte le operazioni che permettono di visualizzare, inserire o rimuovere gli annunci
    - Sottosistema **Gestione Acquisto**: In esso sono incluse tutte le operazioni che permettono di acquistare un biglietto
    - Sottosistema **Gestione Account**: in esso sono incluse tutte le operazioni di accesso al sistema.
- Il livello di **Storage**: sistema che gestisce ed immagazzina i dati persistenti. Questo è l'unico sottosistema che deve cambiare se cambia l'interfaccia del sottosistema DATABASE.





### 3.3 Mapping hardware/software



Per il sistema che si vuole sviluppare è stata scelta un'architettura Client/Server, dove ognuno dei due nodi contiene specifiche componenti.

Tramite una qualunque device dotata di browser l'utente può accedere all'interfaccia grafica del sistema.

Le componenti di Business Logic, dello Storage e il database sono istanziate nel Server.

Questo consente di centralizzare le risorse, dato che il server è al centro della rete. In generale la comunicazione tra client e server avverrà sfruttando il protocollo HTTP.

L'HyperText Transfer Protocol (HTTP) (protocollo di trasferimento di un ipertesto) è un protocollo a livello applicativo usato come principale sistema per la trasmissione d'informazioni sul web ovvero in un'architettura tipica Client/Server.

### 3.4 Controllo Accessi e Sicurezza

Il controllo degli accessi è garantito dall'utilizzo di username e password per tutti gli utenti; tali dati verranno richiesti per ogni singolo accesso. La sicurezza è garantita, in quanto tutti i dati personali dei singoli utenti sono protetti dall'accesso controllato. Nel sistema ci sono informazioni che possono essere visualizzate da ogni tipo di utente: i voli e le relative informazioni. L'utente responsabile della sponsorizzazione cioè il Promoter, invece, potrà accedere alle informazioni relative agli Annunci pubblicati, mentre l'utente Amministratore responsabile alla gestione dei voli potrà accedere alla modifica dei dati.

Nel sistema vi sono diversi tipi di utenti che hanno in comune le seguenti funzionalità:

-Login

-Logout

Sottosistema Attore	Gestione Account	Gestione Acquisto	Gestione Voli	Gestione Annunci
CLIENTE	-Registrazione	<ul style="list-style-type: none"> <li>- Ricerca Volo</li> <li>- Aggiunta Volo al carrello</li> <li>- Acquisto Biglietto</li> <li>- Visualizza Biglietto</li> </ul>		
AMMINISTRATORE			<ul style="list-style-type: none"> <li>-Inserimento Volo</li> <li>-Modica Voli</li> <li>-Visualizzazione Voli</li> </ul>	
PROMOTER				<ul style="list-style-type: none"> <li>-Inserimento Annuncio</li> <li>- Cancellazione Annuncio</li> <li>-Visualizzazione Annuncio</li> </ul>

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

## 3.5 Controllo flusso globale del sistema

Il flusso di controllo globale è la sequenza di azioni nel sistema. Il sistema FreedomAir ha un flusso guidato di eventi; le funzionalità richiedono un'interazione continua da parte dell'utente; per questo motivo, il controllo del flusso globale che utilizziamo è di tipo event-driven.

Quindi non abbiamo una sequenza di operazioni prestabilite ma, è l'utente che sceglie l'operazione da eseguire.

## 3.6 Condizione limite

Nel sistema vi sono degli oggetti persistenti; per ognuno di essi bisogna definire le fasi di creazione, eliminazione e di archiviazione nel DB.

### Oggetti persistenti

#### **Utente /Cliente**

Viene creato nel caso d'uso Registrazione Cliente. Tale utente viene archiviato dopo la creazione e non può essere cancellato.

#### **Utente /Amministratore**

Viene creato quando viene creato l'archivio e vengono inseriti i dati nel DB. L'amministratore non può essere cancellato.

#### **Utente /Promoter**

Viene creato quando viene creato l'archivio e vengono inseriti i dati nel DB. Il Promoter non può essere cancellato.

#### **Voli**

Viene creato quando è creato l'archivio e vengono inseriti i dati nel DB, oppure quando l'amministratore decide di aggiungere un volo al sistema. Questo oggetto non può essere cancellato e, viene archiviato dopo la creazione.

#### **Biglietto**

Viene creato quando il cliente acquista il biglietto e i dati vengono inseriti nel DB. Il biglietto viene immediatamente archiviato e non può essere cancellato.

#### **Annunci**

Viene creato quando è creato l'archivio e vengono inseriti i dati nel DB, oppure quando il promoter decide di aggiungere un annuncio al sistema. L'annuncio può essere cancellato dal promoter.

## 3.7. Start-up

Per lo start-up iniziale del sistema FreedomAir, è necessario l'avvio di un web server che fornisca il servizio di un Database MySQL per gestire i dati persistenti e l'interpretazione e l'esecuzione del codice dal lato server. Tramite l'interfaccia di Login, sarà possibile autenticarsi tramite credenziali(username e password) come utente e, sarà permesso l'accesso a determinate funzionalità del sistema. Dopo aver effettuato l'accesso, sarà caricata la home e saranno rese disponibili tutte le operazioni che il sistema fornisce.

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

### 3.7.1 Terminazione

Alla corretta chiusura del servizio, il sistema termina con un Logout regolare e, vengono inviati gli aggiornamenti ai sottosistemi al fine di gestire correttamente i dati in fase di chiusura. La consistenza dei dati viene assicurata annullando eventuali operazioni in esecuzione.

### 3.7.2 Fallimento

I casi di fallimento del sistema possono essere diversi:

1. Guasti dovuti al sovraccarico e successivo fallimento del database. In questo caso, come provvedimento di ha quello del salvataggio periodico dei dati sotto forma di codice SQL per la successiva rigenerazione del DB.
2. Interruzione inaspettata dell'alimentazione, non sono previsti metodi di ripristino dello stato del sistema prima dello spegnimento.
3. Software che causa la chiusura inaspettata dovuta ad errori commessi durante la fase di implementazione. In questo caso non sono previste politiche di correzione; l'unico processo eseguibile sarà la chiusura del sistema e il suo riavvio.
4. Errore critico nell'hardware. In questo caso non sono previste risoluzioni.

## 4. SERVIZI DEI SOTTOSISTEMI

### 4.1 Application Layer

#### 4.1.1 Gestione Account

Sottosistema: Gestione Account	
<b>Descrizione</b>	Sottosistema appartenente alla stratificazione <i>Application Layer</i> , contiene un'interfaccia che gestisce l'accesso al sistema.
Servizi del sottosistema	
Operazioni	Comportamento
<b>Login()</b>	Operazione che offre la possibilità a un Utente Registrato di accedere alla sezione ad egli riservata.
<b>LogOut ()</b>	Operazione che offre la possibilità a un Utente Registrato di uscire dal sistema
<b>Registrazione()</b>	Operazione che offre la possibilità a un cliente di registrarsi al sistema.

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

#### 4.1.2 Gestione Voli

Sottosistema: Gestione Voli	
<b>Descrizione</b>	Sottosistema appartenente alla stratificazione Application Layer, contiene un interfaccia che gestisce i voli, con funzionalità dedicate all'Amministratore.
Servizi del sottosistema	
Operazioni	Comportamento
VisualizzaVolo()	Operazione che offre la possibilità all'Amministratore di visualizzare i voli presenti nel sistema.
ModificaVolo()	Operazione che offre la possibilità all'Amministratore di effettuare l'update del volo.
InserimentoVolo ()	Operazione che offre la possibilità all'Amministratore di aggiungere un nuovo volo sistema.

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

### 4.1.3 Gestione Annunci

Sottosistema: Gestione Annunci	
<b>Descrizione</b>	Sottosistema appartenente alla stratificazione Application Layer, contiene un interfaccia che gestisce gli Annunci, con funzionalità dedicate solamente al Promoter , al quale è affidato totalmente l'incarico di Gestione degli Annunci.
Servizi del sottosistema	
Operazioni	Comportamento
<b>VisualizzaAnnuncio()</b>	Operazione che offre la possibilità al Promoter di visualizzare gli Annunci presenti nel sistema.
<b>InserimentoAnnuncio()</b>	Operazione che offre la possibilità al Promoter di effettuare l'aggiunta dell'Annuncio all'interno del Sistema.
<b>CancellazioneAnnuncio()</b>	Operazione che offre la possibilità al Promoter di eliminare l'annuncio dal sistema.

### 4.1.3 Gestione Acquisto

Sottosistema: Gestione Acquisto	
<b>Descrizione</b>	Sottosistema appartenente alla stratificazione <i>Application Layer</i> , contiene un'interfaccia che gestisce l'acquisto del biglietto.
Servizi del sottosistema	
Operazioni	Comportamento
<b>RicercaVolo()</b>	Operazione che offre la possibilità a un Cliente di ricercare i voli in base alle proprie esigenze
<b>AggiuntaCarrello()</b>	Operazione che offre la possibilità a un Cliente di aggiungere al carrello il volo desiderato
<b>AcquistoBiglietto()</b>	Operazione che offre la possibilità ad un cliente di acquistare il biglietto desiderato

Progetto: FreedomAir	Versione: 0.5
Documento: SDD	Data 29/11/2017

<b>ModificaDati()</b>	Operazione che offre la possibilità ad un cliente di modificare i propri dati
<b>VisualizzaBiglietto</b>	Operazione che offre la possibilità ad un cliente di visualizzare in modo elettronico il biglietto che ha acquistato

## 5. Glossario

**FreedomAir**= Nome del sistema

**RAD** = Requirement Analysis Document.

**RNF** = Requisito Non Funzionale.

**SDD** = System Design Document.

**Mysql** = è il più diffuso database Open Source basato sul linguaggio SQL.