

**Mille Miglia In Viaggio**  
**System Design Document**  
**Versione 1.0**



Data: 01/02/2019

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### Coordinatore del progetto:

Nome	Matricola
De Lucia Andrea	

### Partecipanti:

Nome	Matricola
Antonio Scognamiglio	0512104868
Giuliana Muto	0512104598
Matteo Fasolino	0512103038

<b>Scritto da:</b>	Giuliana Muto
--------------------	---------------

## Revision History

Data	Versione	Descrizione	Autore
04/02/2019	1.0	Introduzione	Matteo Fasolino
04/02/2019	1.0	Architettura del sistema corrente	Matteo Fasolino
04/02/2019	1.0	Architettura del sistema proposto	Matteo Fasolino
04/02/2019	1.0	Panoramica	Matteo Fasolino
04/02/2019	1.0	Decomposizione in sottosistemi	Matteo Fasolino
04/02/2019	1.0	Mapping hw/sw	Matteo Fasolino
04/02/2019	1.0	Gestione dati persistenti	Matteo Fasolino
04/02/2019	1.0	Diagramma ER & Ristrutturato	Antonio Scognamiglio
04/02/2019	1.0	Schema logico	Antonio Scognamiglio
01/02/2019	1.0	Struttura delle tabelle	Giuliana Muto
01/02/2019	1.0	Tavola dei volumi	Giuliana Muto
01/02/2019	1.0	Tavola delle operazioni	Giuliana Muto
01/02/2019	1.0	Tavola degli accessi	Giuliana Muto
01/02/2019	1.0	Controllo della sicurezza	Giuliana Muto
01/02/2019	1.0	Controllo del flusso globale del sistema	Matteo Fasolino
01/02/2019	1.0	Condizioni Boundary	Matteo Fasolino
01/02/2019	1.0	Gestione Acquisto pacchetto	Giuliana Muto
04/02/2019	1.0	Gestione Vendita pacchetto	Antonio Scognamiglio
04/02/2019	1.0	Glossario	Antonio Scognamiglio

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

## Sommario

1.	INTRODUZIONE .....	4
1.1	Obiettivi del sistema.....	4
1.2	Obiettivi di design .....	5
1.3	Definizioni, acronimi e abbreviazioni.....	8
1.4	Riferimenti .....	8
1.5	Panoramica .....	8
2.	ARCHITETTURA DEL SISTEMA CORRENTE.....	9
3.	ARCHITETTURA DEL SISTEMA PROPOSTO .....	9
3.1	Panoramica .....	9
3.1.1	Vantaggi del Modello MVC .....	10
3.1.2	Svantaggi del Modello MVC .....	10
3.2	Decomposizione in sottosistemi.....	11
	.....	14
3.3	Mapping hardware/software.....	15
3.4	Gestione dati persistenti .....	16
3.4.1	Diagramma ER .....	16
3.4.2	Diagramma ER ristrutturato .....	17
	.....	17
3.4.3	Schema logico .....	19
	.....	19
3.4.4	Struttura delle tabelle.....	20
3.4.5	Controllo degli accessi e Controllo Sicurezza .....	24
3.4.6	Tavola dei volumi .....	25
3.4.7	Tavola delle operazioni .....	26
3.4.5.3	Tavola degli accessi.....	26
3.5	Controllo del flusso globale del sistema .....	28
3.6	Condizioni boundary .....	29
3.6.1	Avvio del sistema.....	29
3.6.2	Terminazione del sistema .....	31
3.6.3	Fallimento del sistema.....	32
4.	SERVIZI DEI SOTTOSISTEMI.....	33
4.1	Gestione profilo utente .....	33
4.2	Gestione Acquisto.....	33
4.3	Gestione Vendita.....	34
5.	GLOSSARIO.....	35

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

# 1. INTRODUZIONE

## 1.1 Obiettivi del sistema

- Oggi giorno il cartaceo abbandona sempre più il campo alla nuova tecnologia e con l'avvento di nuovi dispositivi mobile la connettività e l'informazione ha raggiunto apici elevati rispetto al precedente ventennio. Chi intende fare determinati acquisti si proietta con maggiore sicurezza dal web, sicuri che alle spalle ci siano tramite user-friendly che trasmettono fiducia e garanzia. Il sistema proposto vuole prima di tutto dare un forte impatto, attraendo gli utenti con una interfaccia grafica usabile ed intuitiva, per poi progredire ad un successivo stato di interesse portando l'utente nella sua sicurezza a voler acquistare e/o vendere un determinato pacchetto di viaggi.
- Si vuole agevolare l'utente fornendo al sistema azioni semplici e che richiedono poco tempo per portarle al termine, fornendo un alto fattore di piacere sia nell'acquisto che nella vendita.
- Le informazioni dovranno essere concise, di rapido apprendimento e istantaneamente disponibili, in modo da indurre l'utente ad avere chiarezza e spontaneità nell'affrontare determinati task.
- Si vuole portare l'utente acquirente ad essere sempre interessato ad acquisire informazioni circa i propri interessi, seppur non propenso all'acquisto imminente; pensiamo che le continue informazioni fornite in modo gratuito e poco pervadente possano donare piacere e al contempo attrarre al momento opportuno l'acquirente.
- Vogliamo che l'acquirente si sente in un angolo di scelte sicuro, poco invadente e sempre pronto a fornire le dovute informazioni.
- Si vuole agevolare il partner in modo che possa avere tutte le informazioni utili per la vendita dei pacchetti; la sicurezza che quello che faccia corrisponda alla realtà e che sia sinonimo di sicurezza.
- **Utilizzo di Java:** per mettere in pratica tutte le promesse proposte è indispensabile l'uso del linguaggio java, in quanto ci fornisce i mezzi utili per far rispettare i canoni di usabilità, robustezza, tempo di risposta e personalizzazione.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

## 1.2 Obiettivi di design

n.	NOME	Tempo di risposta.
	<b>CATEGORIA</b>	Performance.
	<b>ORIGINE</b>	RNFP
	<b>DESCRIZIONE</b>	Il sistema deve garantire tempi di risposta inferiori a 1 secondi, ad eccezione nei casi in cui vengono caricate immagini per la messa in vendita di un pacchetto.
	<b>TRADE OFF</b>	Se il sito non rispetta il requisito allora si è costretti ad aumentare la memoria del sistema hardware, oppure a migliorare la qualità della banda del segnale internet.

n.	NOME	Rispecchio delle realtà
	<b>CATEGORIA</b>	Usability
	<b>ORIGINE</b>	RNFU
	<b>DESCRIZIONE</b>	Il sistema deve assicurare che i pulsanti che vengono premuti rispecchino quello che l'utente intende fare.
	<b>TRADE OFF</b>	Nel caso in cui questo non accada il sistema deve provvedere a lanciare delle eccezioni gestite, in modo che l'utente venga immediatamente a conoscenza dell'evento interrotto e agendo di conseguenza.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

n.	NOME	Sicurezza.
	<b>CATEGORIA</b>	Releability.
	<b>ORIGINE</b>	RNFR
	<b>DESCRIZIONE</b>	Il sistema deve assicurare all'utente registrato l'accesso, l'acquisizione e la gestione dei dati in maniera sicura, dimostrando l'affidabilità sempre con delle notifiche di una qualsiasi operazione avvenuta.
	<b>TRADE OFF</b>	Se il requisito non viene rispettato il sistema lancia un'eccezione dove comunica in primis l'impedimento all'accesso dei dati e successivamente a segnalare lo staff tecnico, il quale provvederà alla dovuta manutenzione.

n.	NOME	Compatibilità.
	<b>CATEGORIA</b>	Supportability.
	<b>ORIGINE</b>	RNFS
	<b>DESCRIZIONE</b>	Il sistema deve garantire la corretta visualizzazione del sito sul browser, questo grazie ai protocolli di navigabilità.
	<b>TRADE OFF</b>	Nel caso in cui il sito non è visualizzabile il sistema segnala all'utente tramite un'eccezione che il proprio browser non è compatibile con la versione predisposta dai progettisti; successivamente il sistema invia una notifica al team di progettazione per una futura manutenzione.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

n.	NOME	Interfacciamento
	<b>CATEGORIA</b>	Interface
	<b>ORIGINE</b>	RNFI
	<b>DESCRIZIONE</b>	Il sistema deve garantire il corretto collegamento con i server per la comunicazione client-sever durante le fasi di registrazione, accesso, acquisto e vendita.
	<b>TRADE OFF</b>	La mancata o errata comunicazione può essere causata anche da problemi esterni ed in tal caso il sistema si limita a lanciare un'eccezione in modo da indurre l'utente a controllare la propria connessione, oppure a contattare le terze parti.

n.	NOME	Indipendenza da software
	<b>CATEGORIA</b>	Packaging
	<b>ORIGINE</b>	RNFPK
	<b>DESCRIZIONE</b>	Il sistema non permette installazioni di alcun tipo nei dispositivi essendo che lavora su un Browser Web.
	<b>TRADE OFF</b>	

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 1.3 Definizioni, acronimi e abbreviazioni

**MilleMigliaInViaggio:** il nome del sito da sviluppare.

**User-friendly:** aggettivo utilizzato per definire un software di facile utilizzo anche per persone non esperte nell'utilizzo del computer.

### 1.4 Riferimenti

RAD: Requirements Analysis Document.

### 1.5 Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design. Verrà esplicata successivamente l'architettura del sistema corrente e l'architettura del sistema proposto. Illustriamo rapidamente le parti del documento:

Nel capitolo 2 viene spiegato che non esiste un sistema attuale;

Nel capitolo 3 viene mostrata l'architettura del sistema proposto mostrando una decomposizione del sistema in sottoinsiemi sulla gestione del profilo utente, dell'acquisto e della vendita;

Mapping hardware/software circa la scelta del sistema lato hardware del sistema, la comunicazione tra i vari nodi ed infine l'incapsulamento dei servizi in un sottosistema.

La gestione dei dati persistenti descrive la gestione di questi da parte del sistema, e l'infrastruttura richiesta.

Controllo degli accessi e della sicurezza descrive, tramite una matrice degli accessi, le operazioni effettuabili e le informazioni consultabili da ogni tipologia di utente e come questi si autenticano al sistema;

Controllo del flusso globale descrive quali operazioni eseguire ed in che ordine, per garantire il corretto flusso di controllo del sistema;

Le condizioni boundary descrivono le fasi salienti quali l'avviamento, la terminazione e gestione dei fallimenti;

Il capitolo 4 descrive i servizi offerti dai sottosistemi ed il loro relativo range di azione;

Il capitolo 5 descrive un glossario, il quale elenca la terminologia e le definizioni utilizzati nel documento.



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

## 2. ARCHITETTURA DEL SISTEMA CORRENTE

Non esiste un Sistema attuale da sostituire, e le attività di MilleMigliaInViaggio sono complete.

## 3. ARCHITETTURA DEL SISTEMA PROPOSTO

### 3.1 Panoramica

L'architettura scelta per questo sistema è il Model-View-Controller (MVC). Questo tipo di architettura permette di dividere nettamente la logica di business e quella di presentazione.

#### **Servlet (Controller):**

- processes the request;
- handles the application logic;
- instantiates Java beans (Model).

Accetta l'input dell'utente dalla richiesta e capisce cosa significa per il modello.

Indica al modello di aggiornarsi e rende disponibile lo stato del nuovo modello per la vista (JSP).

#### **(Model):**

Contiene la vera logica di business e le regole per ottenere e aggiornare lo stato. I contenuti di un Carrello e le regole su cosa fa con esso farebbero parte del Model in MVC.

#### **JSP (View):**

- ottiene i dati del form dal bean;
- formatta la risposta senza essere a conoscenza di nulla circa quello che sta accadendo.

Responsabile della presentazione. Ottiene lo stato del modello dal controller (anche se non direttamente, il controller inserisce i dati del modello in un punto in cui la vista può trovarlo). È anche la parte che riceve l'input dell'utente che torna al controller.

A seguire vengono descritti i passi di questo modello.

- Chiamata al form:

Il client effettua una richiesta per la pagina form.html

Il container recupera la pagina form.html

Il container restituisce la pagina al browser, in cui l'utente risponde alle domande sul model;

Esecuzione dell'applicazione:

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

Il browser invia i dati della richiesta al Container;  
 Il Container invia la servlet corretta in base all'URL e passa la richiesta alla servlet;  
 La servlet chiama il Bean per la gestione dei dati dettagliati;  
 La classe restituisce una risposta e la servlet aggiunge all'oggetto di request;  
 La servlet inoltra la richiesta alla JSP;  
 La JSP ottiene la risposta dall'oggetto request;  
 La JSP genera una pagina per il Container;  
 Il Container restituisce la pagina all'utente.

### 3.1.1. Vantaggi del Modello MVC

I vantaggi per questo tipo di architettura sono visibili fin dall'inizio da un punto di vista di velocità nella manutenibilità; gli sviluppatori possono lavorare separatamente e parallelamente sui tre fronti del modello, semplificando e modulando anche il lavoro stesso.

Questo modello inoltre elimina ridondanze di classi, permettendo ad una classe model di poter gestire più jsp; con il cambiamento dei protocolli diventa anche più semplice per uno sviluppatore apportare modifiche o miglioramenti da questo punto di vista, perché si è costretti a modificare poche righe di codice.

### 3.1.2. Svantaggi del Modello MVC

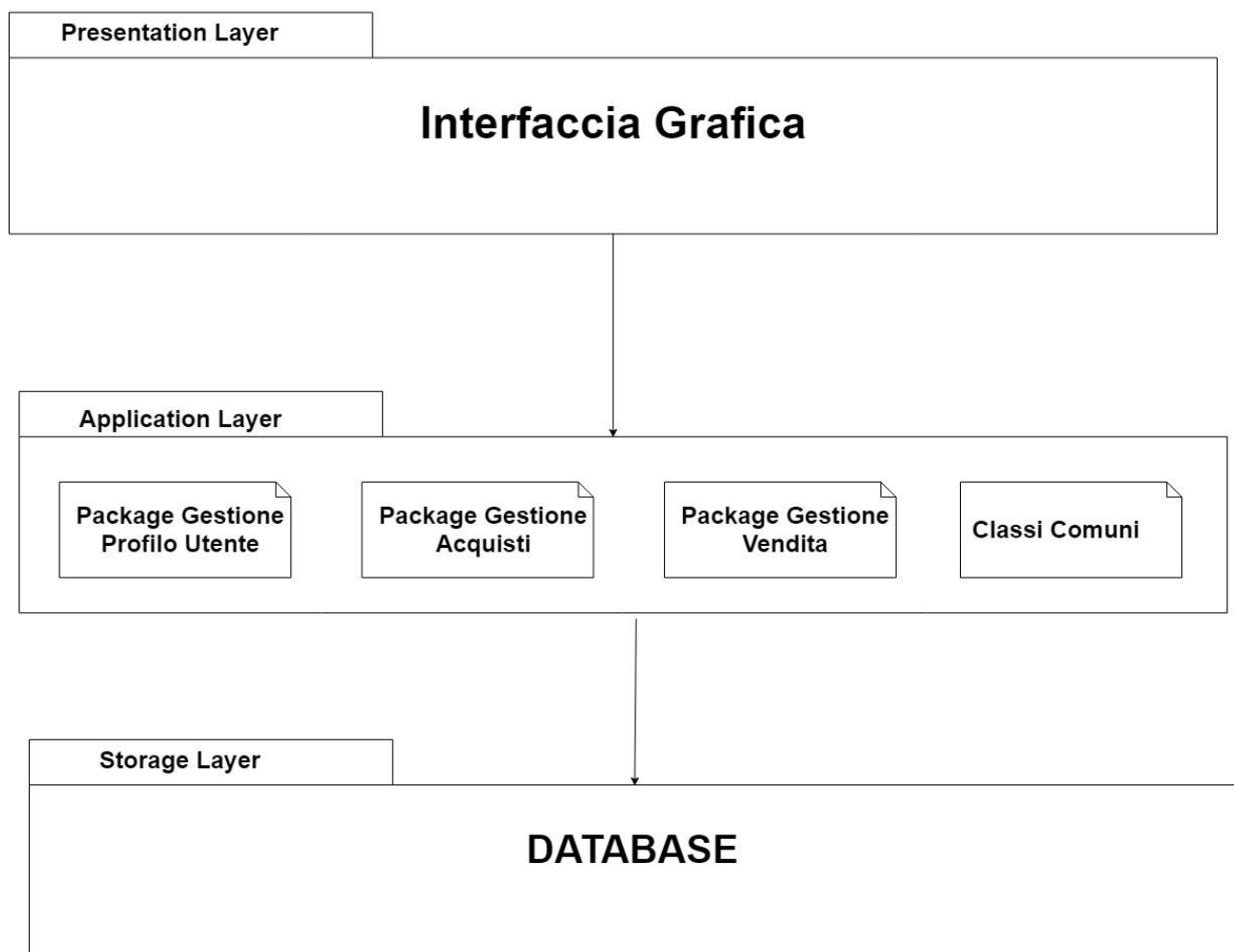
Con questo modello architetturale la qualità del servizio è dipesa molto se non completamente dalla qualità di lavoro del team di progettazione. È un sistema molto articolato e richiede un'ottima dose di coordinazione nei tempi e nei modi.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

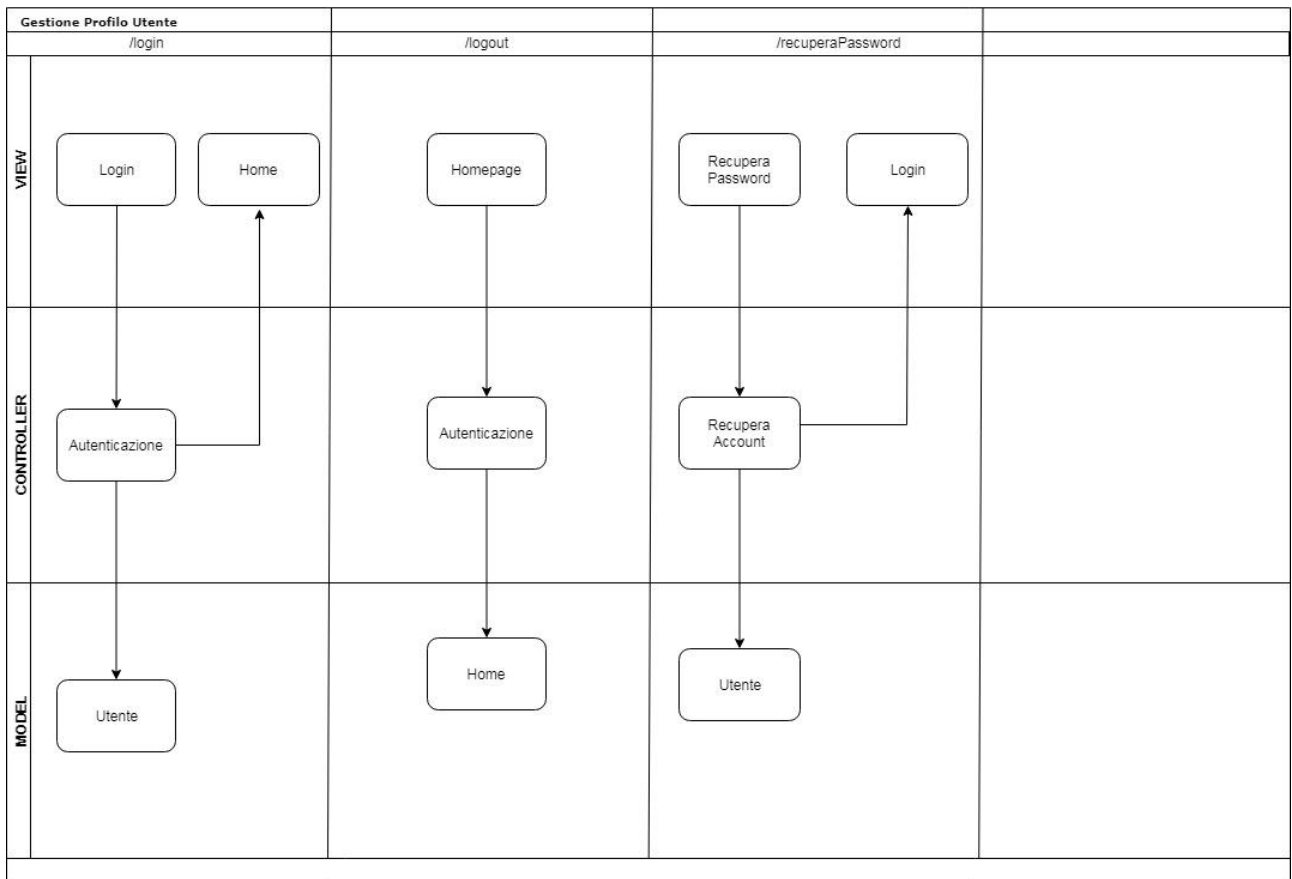
### 3.2. Decomposizione in sottosistemi

Il sistema è suddiviso in 3 layer che si occupano a sua volta di tre gestioni principali: Gestione Profilo Utente, Gestione Acquisto Pacchetto e Gestione Vendita Pacchetto.

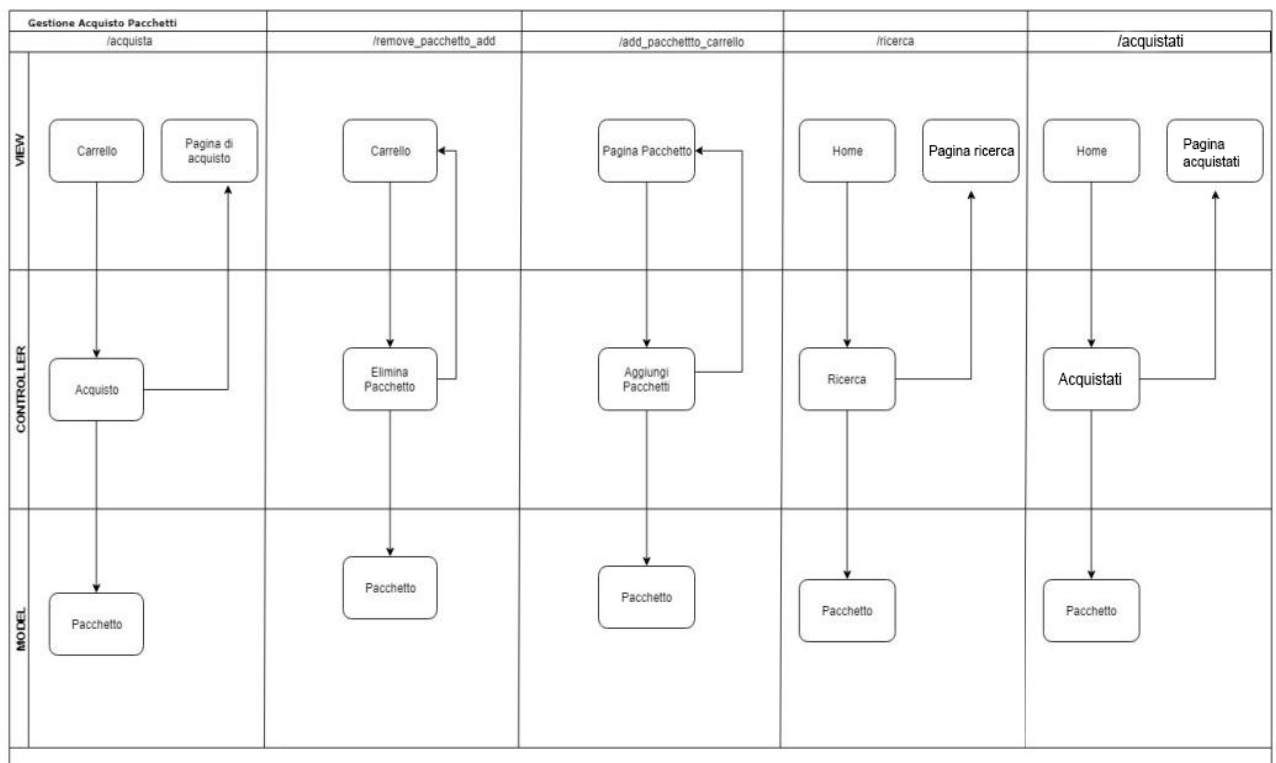
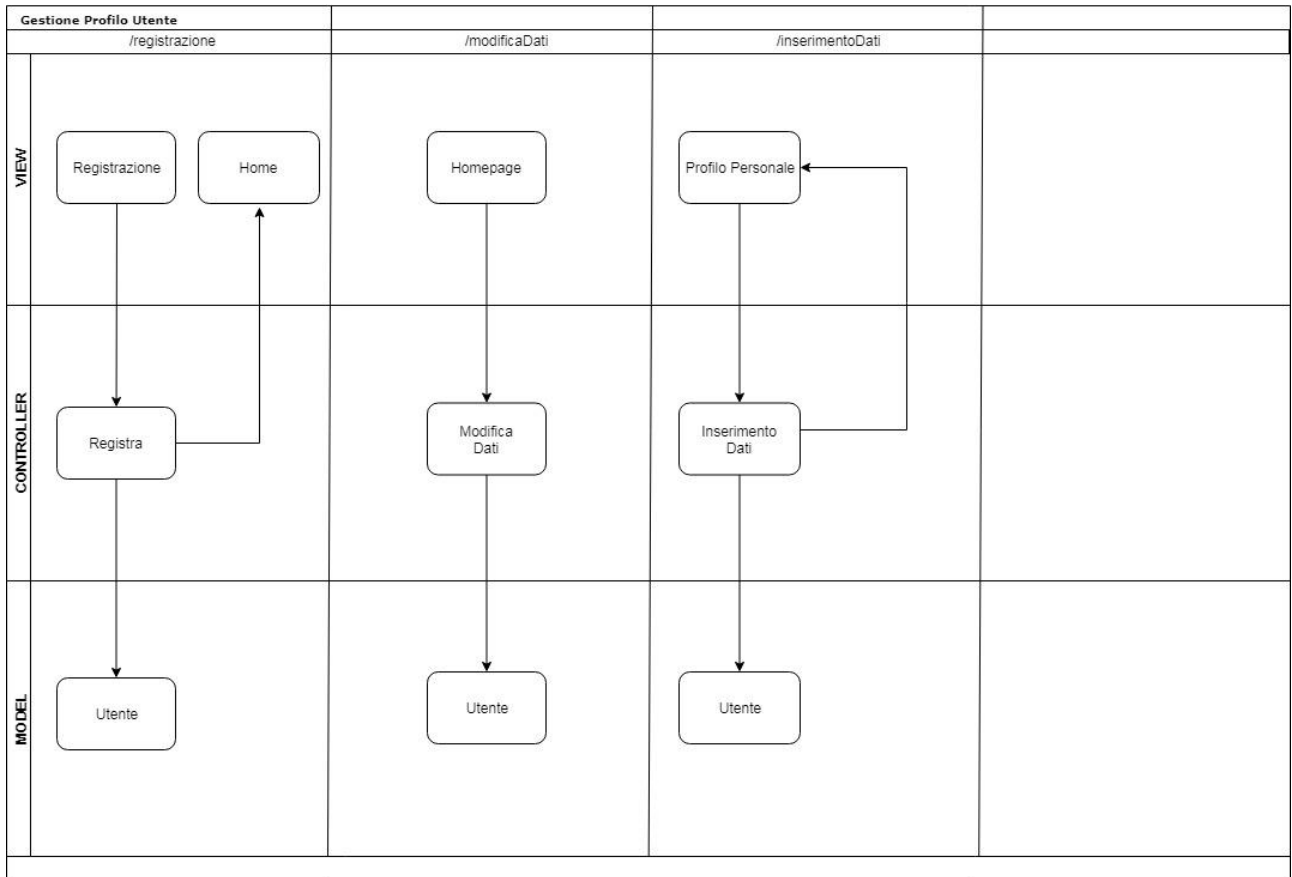
- **Presentation Layer:** è composta dall'interfaccia utente, la quale interagisce con tutti i sottosistemi dell'Application Layer; questa permette la gestione dell'interfacciamento grafico e gli eventi inerenti all'utente;
- **Application Layer:** si scompone nella gestione della gestione del profilo utente, dell'acquisto dei pacchetti e della vendita; gestisce la logica Control del sistema interagendo sempre con i database e fornendo le funzionalità elencate nelle varie scomposizioni dettagliate;
- **Storage:** è lo strato ultimo contenente i dati persistenti del sistema e là dove avviene la gestione dei dati provenienti dai vari sottosistemi.



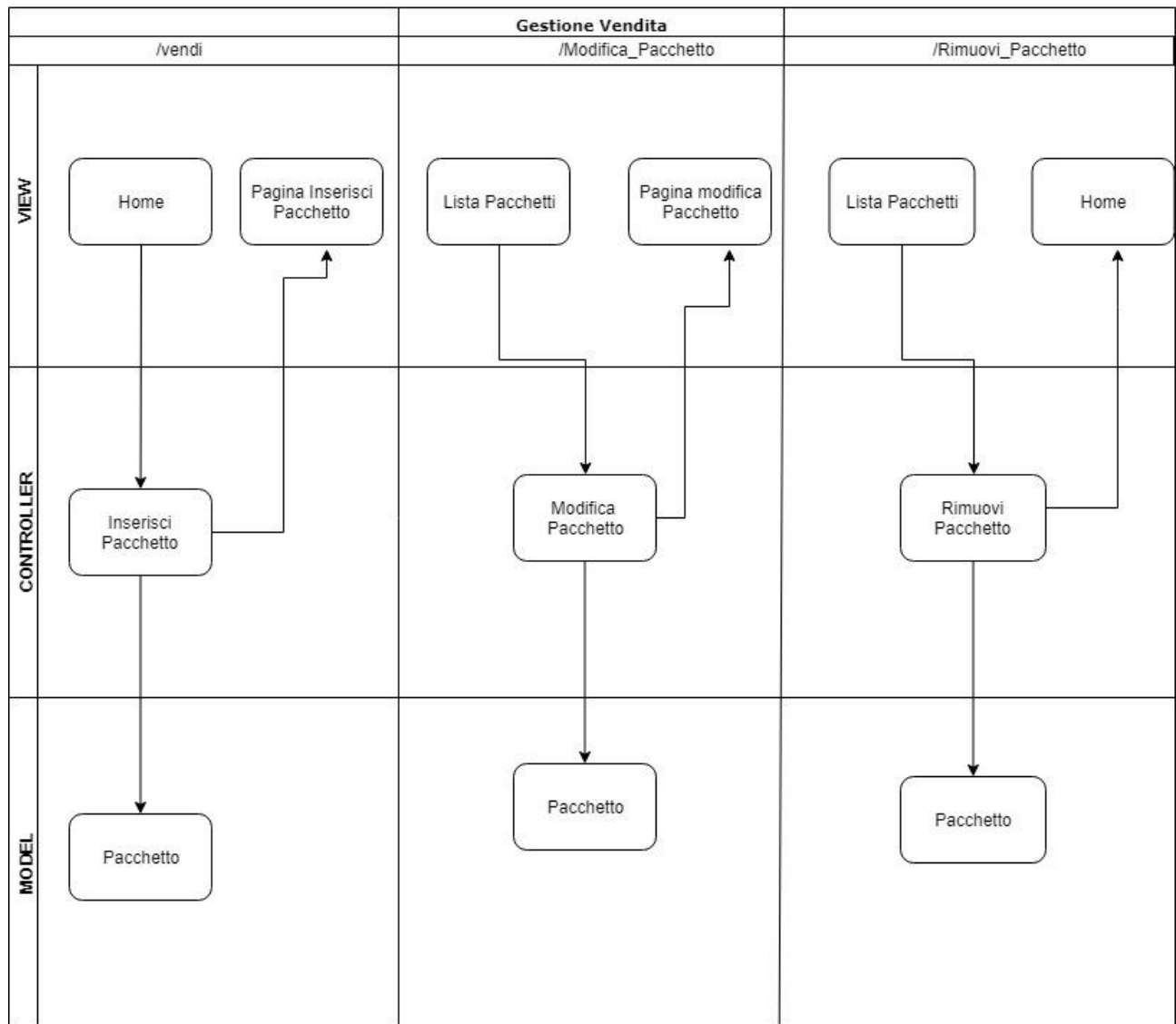
Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

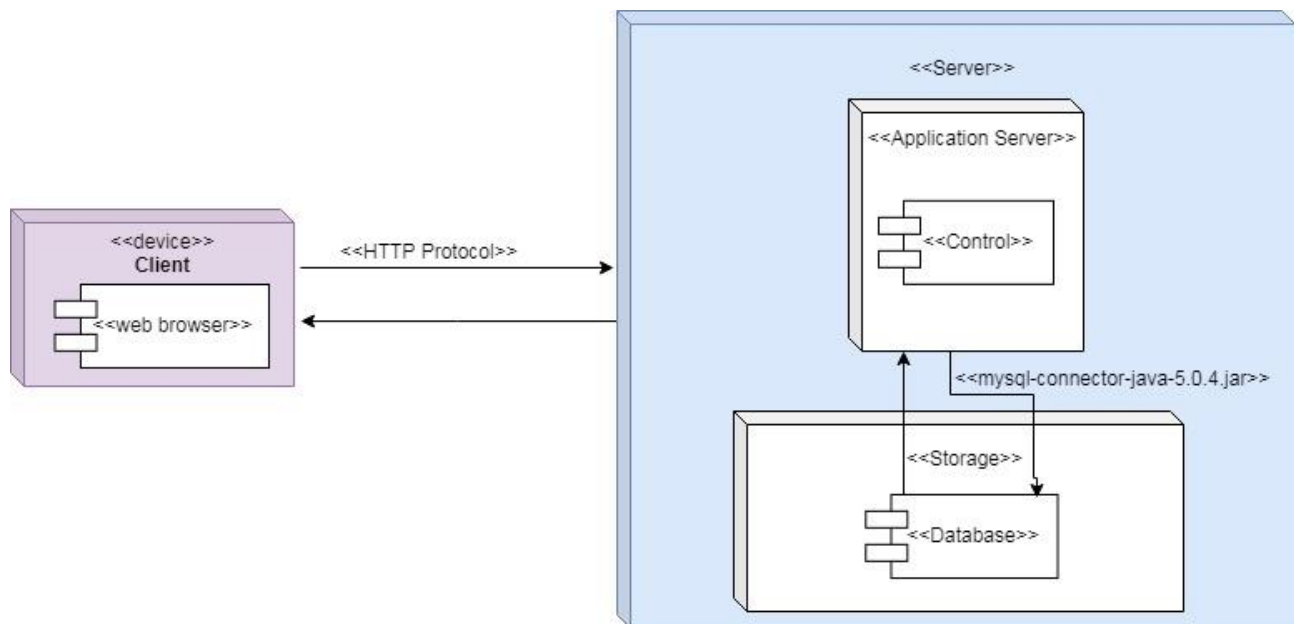
### 3.3. Mapping hardware/software

Il sistema si presenta con un'architettura di tipo client/server divisa in due "Layer", in cui un server espone i servizi ad uno o più client. Nel primo layer, il client è eseguito da un qualsiasi browser web che supporti HTML5.

Il secondo layer è formato dal web Server che gestisce le azioni del "control".

Il Client e il Server comunicheranno tramite "protocollo http" ( un protocollo di trasferimento di ipertesti, che consente a due macchine di tipo client/server di interagire tramite request/response)

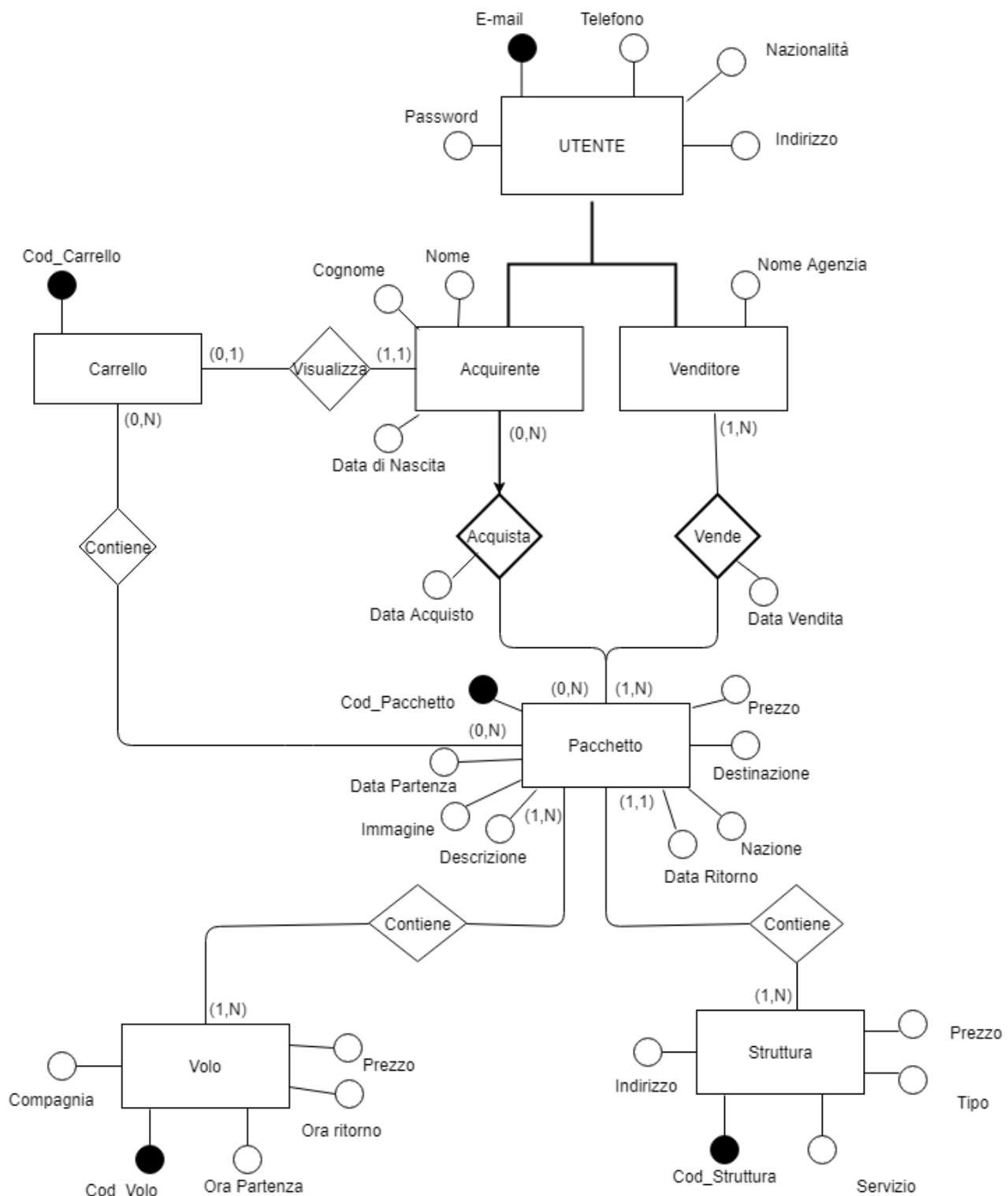
Nella gestione del server ritroviamo l'utilizzo del Database che comunica con il precedente attraverso un Application Server; qui avviene la gestione dei dati persistenti tramite il DBMS MySQL.



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4. Gestione dati persistenti

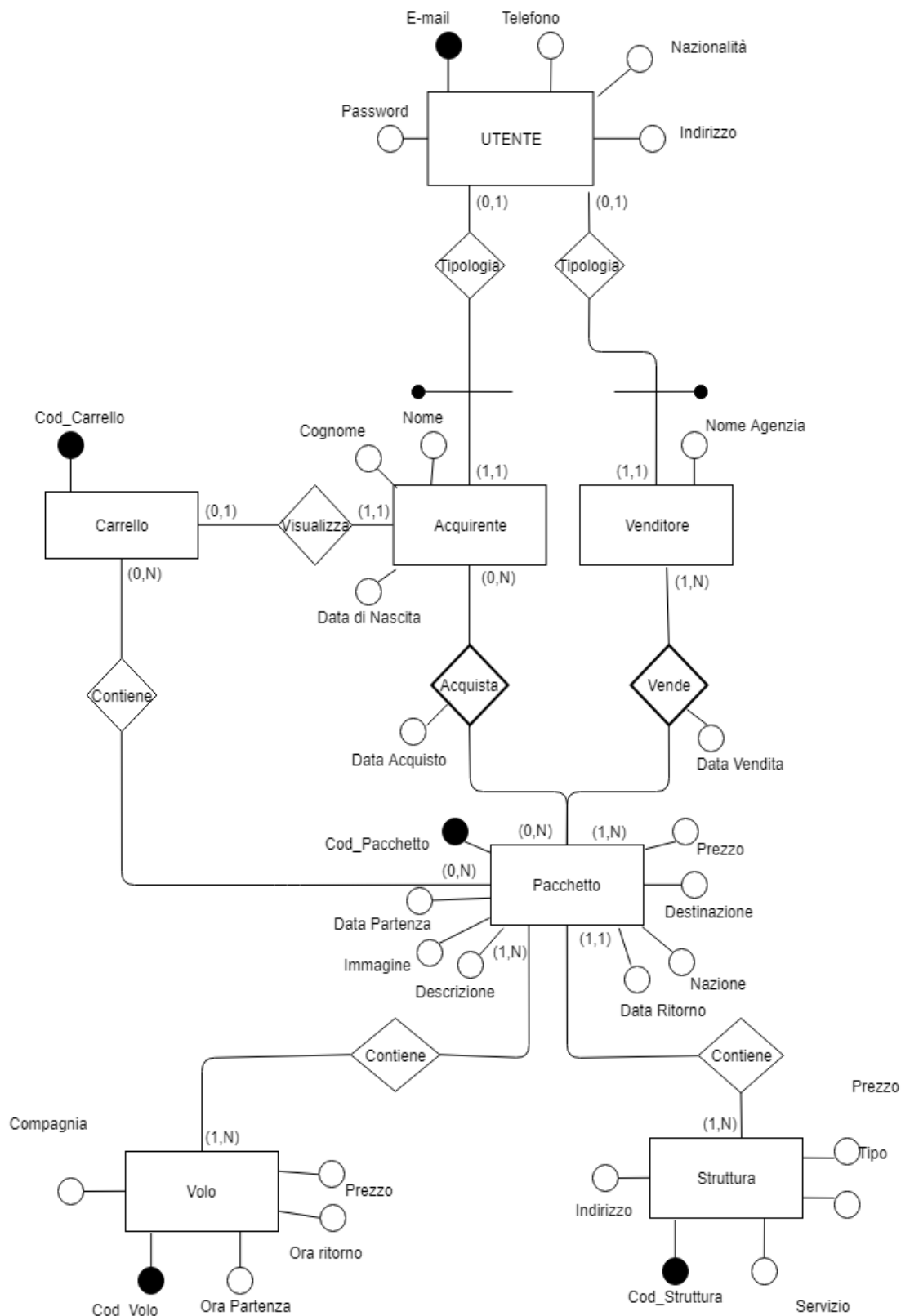
#### 3.4.1. Diagramma ER



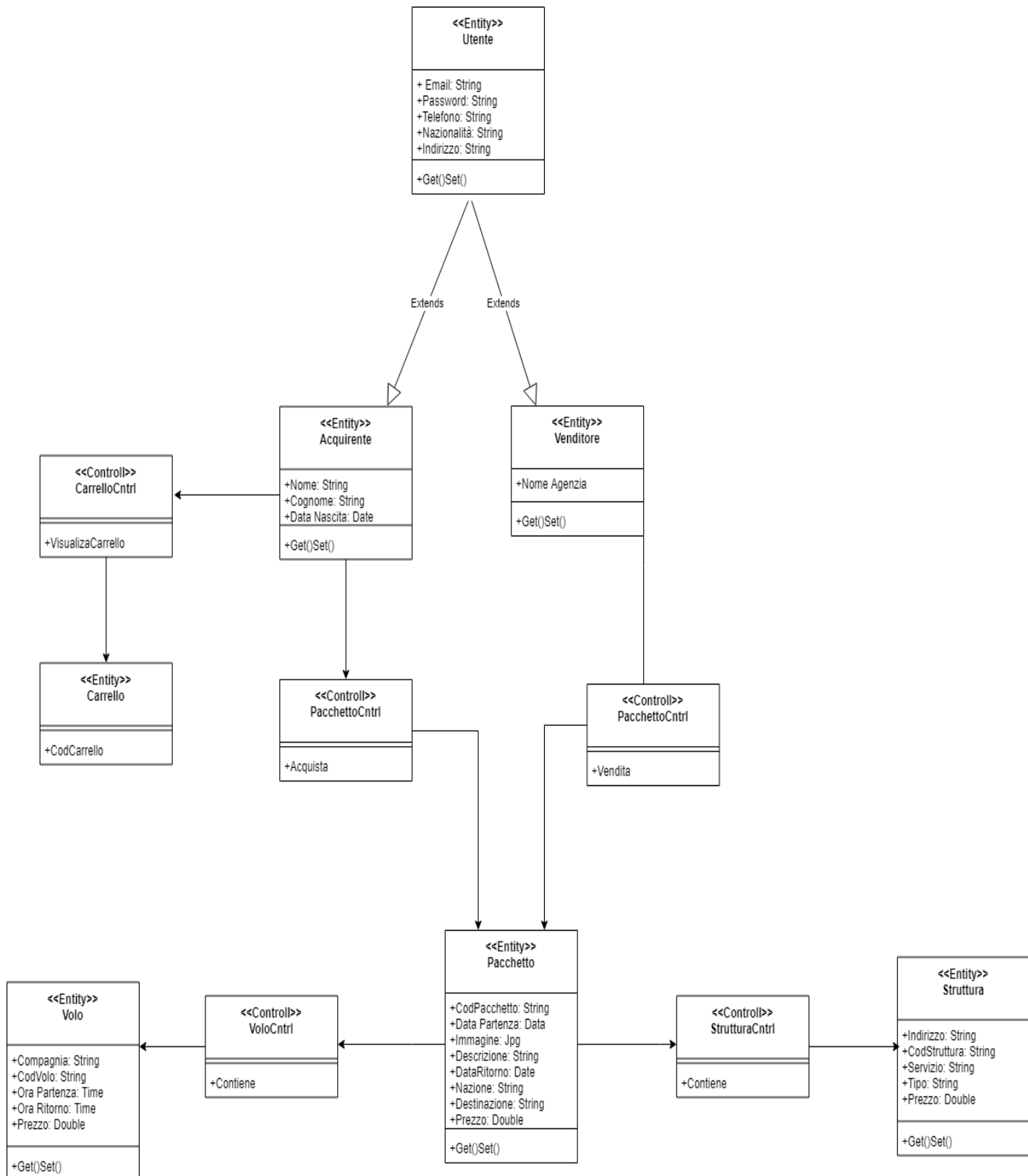


Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4.2. Diagramma ER ristrutturato

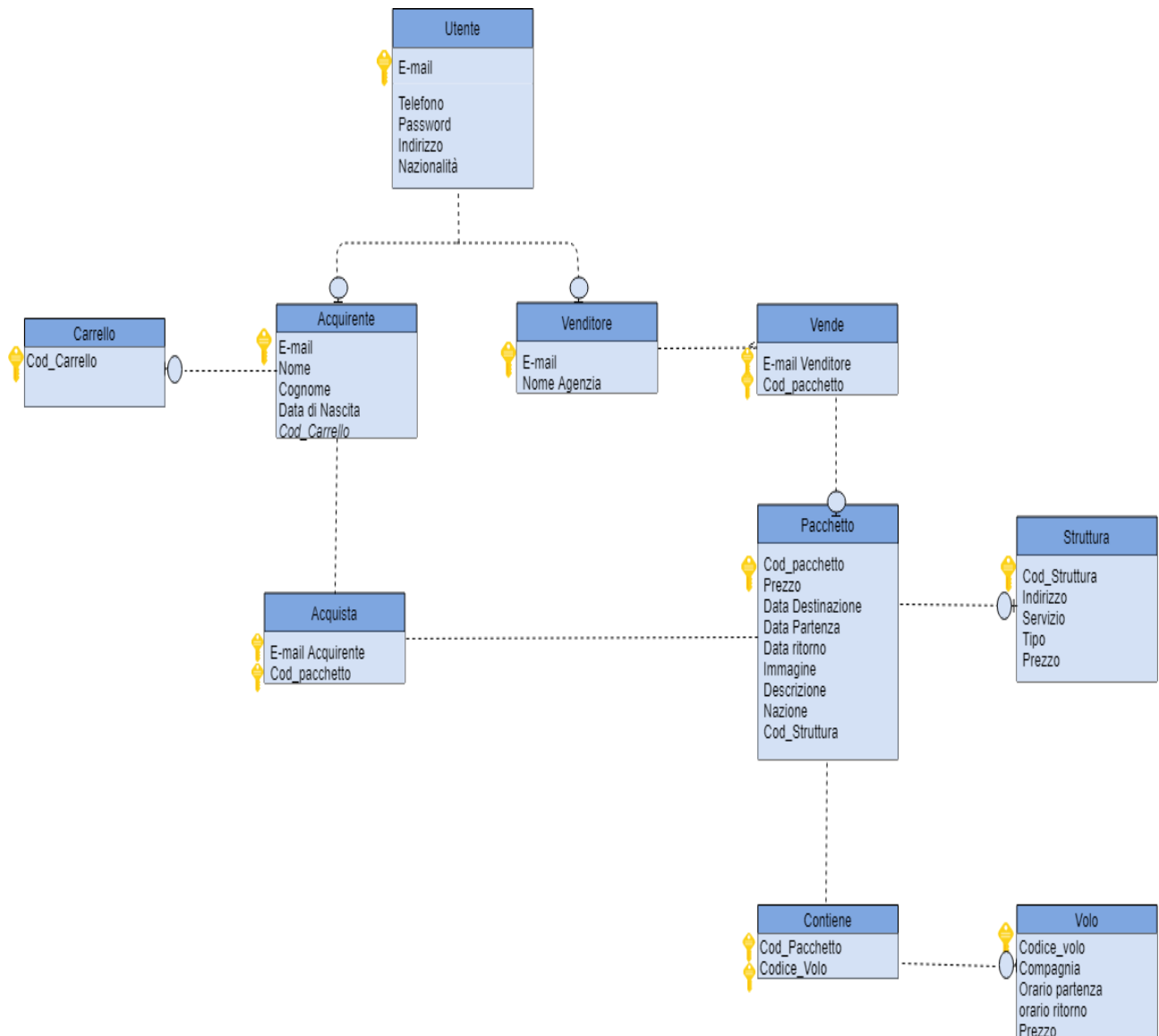


Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4.3. Schema logico



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4.4. Struttura delle tabelle

Di seguito sono riportate tutte le tabelle che vanno a formare il nostro database per la gestione di tutte le informazioni del sistema. In ogni tabella è indicato: il nome, il compito e tutti gli attributi ad essa associati, con relativi vincoli e tipo.

Utente		
Campo	Vincoli	Tipo
<b>E-mail</b>	Lunghezza massima: 50 Primary Key	Varchar
<b>Password</b>	Lunghezza massima: 25 Not null	Varchar
<b>Telefono</b>	Lunghezza massima:10 Not null	Int
<b>Nazionalità</b>	Lunghezza massima: 20 Not null	Varchar
<b>Indirizzo</b>	Lunghezza massima:50 Not null	Varchar

Acquirente		
Campo	Vincoli	Tipo
<b>Email</b>	Lunghezza massima: 50 Primary Key, Foreign key	Varchar
<b>Nome</b>	Lunghezza massima: 25 Not null	Varchar
<b>Cognome</b>	Lunghezza massima: 25 Not null	Varchar
<b>Data di nascita</b>	Data	Date

Venditore		
Campo	Vincoli	Tipo
<b>Email</b>	Lunghezza massima: 50 Primary Key, Foreign key	Varchar
<b>Nome Agenzia</b>	Lunghezza massima: 25 Not null	Varchar

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

Carrello		
Campo	Vincoli	Tipo
<b>Cod_carrello</b>	Lunghezza massima: 50 Primary Key	Varchar

Pacchetto		
Campo	Vincoli	Tipo
<b>Cod_pacchetto</b>	Lunghezza massima: 50 Primary Key	Varchar
<b>Prezzo</b>	Lunghezza massima: 25 Not null	Doube
<b>Descrizione</b>	Lunghezza massima: 100 Not null	Varchar
<b>Data partenza</b>	Data Not null	Date
<b>Data ritorno</b>	Data Not null	Date
<b>Immagine</b>	Not null	Blob
<b>Destinazione</b>	Lunghezza massima 25 Not null	Varchar
<b>Nazione</b>	Lunghezza massima 25 Not null	Varchar

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

Pacchetto		
Campo	Vincoli	Tipo
<b>Cod_pacchetto</b>	Lunghezza massima: 50 Primary Key	Varchar
<b>Prezzo</b>	Lunghezza massima: 25 Not null	Double
<b>Descrizione</b>	Lunghezza massima: 100 Not null	Varchar
<b>Data partenza</b>	Data Not null	Date
<b>Data ritorno</b>	Data Not null	Date
<b>Immagine</b>	Not null	Blob
<b>Destinazione</b>	Lunghezza massima 25 Not null	Varchar
<b>Nazione</b>	Lunghezza massima 25 Not null	Varchar
<b>Data acquisto</b>	Data	Date
<b>Data vendita</b>	Data	Date

Volo		
Campo	Vincoli	Tipo
<b>Cod_volo</b>	Lunghezza massima: 50 Primary Key	Varchar
<b>Prezzo</b>	Lunghezza massima: 25 Not null	Double
<b>Ora partenza</b>	Data Not null	Date
<b>Ora ritorno</b>	Data Not null	Date
<b>Compagnia</b>	Lunghezza massima 25 Not null	Varchar

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

Struttura		
Campo	Vincoli	Tipo
<b>Cod_struttura</b>	Lunghezza massima: 50 Primary Key	Varchar
<b>Prezzo</b>	Lunghezza massima: 25 Not null	Doube
<b>Tipo</b>	Lunghezza massima: 25 Not null	Varchar
<b>Servizio</b>	Lunghezza massima: 50 Not null	Varchar
<b>Indirizzo</b>	Lunghezza massima:50 Not null	Varchar

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4.5. Controllo degli accessi e Controllo Sicurezza

Il software MilleMigliaInViaggio permette l'accesso tipi di utenti diversi: venditore, acquirente e utenti non registrati.

Per poter effettuare l'accesso alla piattaforma come acquirente o venditore bisognerà inserire le credenziali utilizzate nella registrazione (email e password) altrimenti si può utilizzare come utente non registrato ma con diverse restrizioni sulle proprie azioni.

Attori	Gestione utente	Gestione vendita	Gestione Acquisto
<b>Venditore</b>	<ul style="list-style-type: none"> <li>• Visualizzare il proprio profilo utente</li> <li>• Modificare il proprio profilo</li> <li>• Recuperare Password</li> <li>• Log-out dal sistema</li> </ul>	<ul style="list-style-type: none"> <li>• Aggiungere pacchetti</li> <li>• Eliminare pacchetti</li> <li>• Modificare pacchetti</li> </ul>	
<b>Acquirente</b>	<ul style="list-style-type: none"> <li>• Visualizzare il proprio profilo utente</li> <li>• Modificare il proprio profilo</li> <li>• Recuperare Password</li> <li>• Log-out dal sistema</li> <li>• Ricerca per filtri</li> </ul>		<ul style="list-style-type: none"> <li>• Aggiungere pacchetti</li> <li>• Eliminare pacchetti</li> <li>• Comprare pacchetti</li> </ul>
<b>Utente non registrato</b>	<ul style="list-style-type: none"> <li>• Ricerca per filtri</li> </ul>		



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4.6. Tavola dei volumi

Ogni anno circa un miliardo di persone ogni anno in viaggio per conoscere il mondo. Meno della metà si rivolgono ad agenzie di viaggio. Prima dell'avvento dello shop online, le agenzie di viaggio erano più di 12.000 solo in Italia, ad oggi ne contano 8.500. MilleMigliaInViaggio nasce con l'esigenza di aiutare le agenzie di viaggio a far vendere i propri pacchetti viaggio, ma anche di aiutare l'utente finale all'acquisto di viaggi in modo sicuro e vantaggioso.

Si auspica alla presenza di più di 1500 agenzie di viaggio. Ogni agenzia di viaggio può avere a disposizione una piattaforma dove poter caricare tutti i dettagli del pacchetto. Si possono caricare circa 20 pacchetti. Ci si aspetta quindi circa 5.500 pacchetti viaggio di ogni tipo. Inoltre l'utente ha possibilità illimitata di acquistare i pacchetti viaggio.

Concetto	Costrutto	volume
<b>Acquirente</b>	E	4000
<b>Visualizza (Acquirente-carrello)</b>	R	40000
<b>Acquista (Acquirente-Pacchetto)</b>	R	1500
<b>Venditore</b>	E	1500
<b>Vende (Venditore-Pacchetto)</b>	R	2500
<b>Pacchetto</b>	E	30.000
<b>Contiene (Pacchetto-Volo)</b>	R	30.000
<b>Contiene (Pacchetto-Struttura)</b>	R	30.000
<b>Volo</b>	E	30.000
<b>Struttura</b>	E	3000

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.4.6 Tavola delle operazioni

Operazione	Tipo	Frequenza
<b>Registrazione</b>	Interattiva	5000/anno
<b>Log-in</b>	Interattiva	80.000/anno
<b>Log-out</b>	Interattiva	70.000/anno
<b>Recupera password</b>	Interattiva	1000/anno
<b>Acquisto</b>	Interattiva	15.000/anno
<b>Ricerca pacchetto</b>	Interattiva	60.000/anno
<b>Selezione pacchetto</b>	Interattiva	100.00/anno
<b>Eliminazione Pacchetto(Carrello)</b>	Interattiva	45.000/anno
<b>Aggiunta pacchetto(carrello)</b>	Interattiva	67.000/anno
<b>Vendita</b>	Interattiva	15.000/anno
<b>Modifica pacchetto(venditore)</b>	Interattiva	10.000/anno
<b>Eliminazione pacchetto(venditore)</b>	Interattiva	23.000/anno

### 3.4.7 Tavola degli accessi

#### Registrazione

Concetto	Costrutto	Accessi	Tipo Accessi
<b>Acquirente/venditore</b>	E	1	S
<b>Totali accessi = 2 X5.000 = 10.000 accessi / anno</b>			

#### Log in

Concetto	Costrutto	Accessi	Tipo Accessi
<b>Acquirente/venditore</b>	E	1	L
<b>Totali accessi = 1 X80.000 = 80.000 accessi / anno</b>			

#### Log out

Concetto	Costrutto	Accessi	Tipo Accessi
<b>Acquirente/venditore</b>	E	1	L
<b>Totali accessi = 1 X70.000 = 70.000 accessi / anno</b>			

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

#### Recupera password

Concetto	Costrutto	Accessi	Tipo Accessi
Acquirente/venditore	E	1	L
<b>Totali accessi = 1 X1000 = 1000 accessi / anno</b>			

#### Acquisto

Concetto	Costrutto	Accessi	Tipo Accessi
Acquirente	E	1	L
Pacchetto	E	1	L
Acquisto	R	1	S
<b>Totali accessi = 1 +1+2*15.000 = 60.000 / anno</b>			

#### Ricerca

Concetto	Costrutto	Accessi	Tipo Accessi
Pacchetto	E	1	L
<b>Totali accessi = 1 X60.000 = 60.000 accessi / anno</b>			

#### Selezionare pacchetto

Concetto	Costrutto	Accessi	Tipo Accessi
Pacchetto	E	1	L
<b>Totali accessi = 1 X100.000 = 100.000 accessi / anno</b>			

#### Eliminazione pacchetto

Concetto	Costrutto	Accessi	Tipo Accessi
Carrello	E	1	L
Pacchetto	E	1	L
Carrello	E	1	S
<b>Totali accessi = 1X1X2X70.000 = 140.000 accessi / anno</b>			

#### Aggiungi pacchetto

Concetto	Costrutto	Accessi	Tipo Accessi
Pacchetto	E	1	L
Carrello	E	1	S
<b>Totali accessi = 1 X2X70.000 = 140.000 accessi / anno</b>			

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### Vendita

Concetto	Costrutto	Accessi	Tipo Accessi
Venditore	E	1	L
Pacchetto	E	1	L
Vende	R	1	S
<b>Totali accessi = 1 X1X2X150.000 = 300.000 accessi / anno</b>			

### Modifica pacchetto

Concetto	Costrutto	Accessi	Tipo Accessi
Pacchetto	E	1	L
Pacchetto	E	1	S
<b>Totali accessi = 1 X2X10.000 = 20.000 accessi / anno</b>			

### Eliminazione pacchetto

Concetto	Costrutto	Accessi	Tipo Accessi
Pacchetto	E	1	L
Pacchetto	E	1	S
<b>Totali accessi = 1 X2X23.000 = 46.000 accessi / anno</b>			

## 3.5 Controllo del flusso globale del sistema

Il controllo del flusso del software è regolato da classi Java che fungono da ricevitori di eventi e che rispondono alle attivazioni di client. Le richieste vengono generate da un client, e la classe preposta a gestire quel determinato evento associato alla richiesta, prendendo talvolta degli input, si preoccupa di inizializzare le richieste e di inoltrarle alle classi per lo svolgimento dell'operazione. Una volta ottenuto il risultato dell'operazione, la classe gestore si preoccupa di inoltrarlo al client che aveva generato la richiesta. Il sistema software è gestito con l'uso di Servlet e JSP. Il server centrale attende le richieste di un client (web browser) e una volta ricevuta una richiesta, la processa e la smista alla Servlet deputata.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.6 Condizioni boundary

Sono le condizioni di inizio e di fine di ogni sequenza del sistema.

#### 3.6.1 Avvio del sistema

Servizi forniti agli utenti collegati al sistema avviato

##### 3.6.1.1 GPU\_LI

L'utente accede al sito, clicca sul pulsante "accedi" ed il sistema mostra un piccolo menu bootstrap, dove gli permette di compilare i campi "Email" e "Password".

L'utente clicca su "Accedi" per confermare l'accesso, e il sistema lo indirizza alla homepage con la comparsa del nome dell'utente nel bootstrap dell'area di accesso.

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

<b>Nome: UC_LI</b>
<b>Attori: Utente</b>
<b>Flusso di eventi</b>
1. L'utente clicca sul tasto "Accedi"
2. Il sistema mostra un form da compilare con "e-mail" e "password"
3. L'utente compila il form
4. L'utente clicca sul pulsante "accedi"
5. Il sistema controlla i dati inseriti
6. Il sistema mostra la pagina Home.
<b>Condizioni in entrata</b>
<ul style="list-style-type: none"> <li>Il caso d'uso inizia dopo l'esecuzione del caso d'uso UC_Registrazione_Acquirente oppure UC_Registrazione_Venditore</li> </ul>
<b>Condizioni in uscita</b>
<ul style="list-style-type: none"> <li>il caso d'uso termina quando il sistema mostra la pagina Home.</li> </ul>
<b>Eccezioni</b>
<ul style="list-style-type: none"> <li>Al punto 5 nel caso in cui i dati inseriti sono errati viene eseguito UC_DatiErrati</li> </ul>
<b>Requisiti Speciali</b>
<ul style="list-style-type: none"> <li>La combinazione dei dati, che permette di identificare i singoli utenti durante la navigazione verrà gestita dai cookie, impegnati per trattare i dati personali.</li> <li>Viene utilizzato HTTPS che si occupa della crittografia e dell'autenticazione dei dati trasmessi.</li> </ul>

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.6.2 Terminazione del sistema

La terminazione avviene con l'arresto dei sottosistemi, preceduto da quello delle connessioni alla sessione e circa la gestione dei dati persistenti; i dati vengono prima salvati nel database e a seguire viene arrestato il suo funzionamento per poi passare alla chiusura della sessione.

#### 3.7.2.1 GPU\_LO

L'utente accede al sito, clicca sul pulsante "Logout" posto in alto a destra del navbar, e a seguire il sistema provvede a chiudere la sessione di accesso ed i servizi attivi ripristinando l'homepage di sessione chiusa.

<b>Nome:</b> UC_LO
<b>Attori:</b> Utente
<b>Flusso di eventi</b>
1. L'utente clicca sul pulsante "Logout"
2. Il sistema disconnette l'utente dal sistema
3. Il sistema mostra la schermata della pagina principale
<b>Condizioni in entrata</b>
<ul style="list-style-type: none"> <li>Il caso d'uso inizia dopo l'esecuzione del caso d'uso UC_Accesso_Utente</li> </ul>
<b>Condizioni in uscita</b>
il caso d'uso termina quando l'utente viene reindirizzato alla pagina principale

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

### 3.6.3 Fallimento del sistema

Il fallimento a livello hardware porterà la conseguente sostituzione della parte danneggiata oppure ad un upgrade dello stesso per mantenere la funzionalità del sistema.

Il fallimento a livello software viene gestito dalle eccezioni che nel migliore dei casi permette all'utente di continuare o ripetere le azioni, mentre nel caso peggiore il sistema invia un messaggio all'utente per segnalare l'errore, con un suggerimento finale.

Un ulteriore fallimento può nascere a livello di database nel senso più generico; in caso di problemi di connessione il problema è demandato a livello hardware, mentre in caso di errori lato server il problema viene gestito come di tipo software, gestendo il tutto con le eccezioni. Se i dati non vengono salvati parte un'eccezione che richiede all'utente di reinserire i dati persistenti, in modo da rendere concluso il task.

Il database ha un sistema di backup periodico che permette di salvare i dati persistenti migliorando il servizio di sicurezza.



Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

## 4. SERVIZI DEI SOTTOSISTEMI

### 4.1. Gestione profilo utente

Sottosistema	Gestione Autenticazione
Descrizione	Questo sottosistema permette di effettuare tutte le operazioni relative agli account
Servizi offerti	
Servizi	Descrizione
Login	Questo servizio consente l'accesso ad alcuni servizi disponibili solo agli utenti registrati
Logout	Consente di uscire dal sistema
Recupero Password	Questo servizio consente il recupero della propria password per l'accesso al sistema
Registrazione nuovo utente	Questo servizio consente ad un utente non registrato di creare un nuovo account per accedere al sistema
Modifica profilo	Questo servizio consente agli utenti che hanno effettuato il login di modificare i propri dati.

### 4.2. Gestione Acquisto

Sottosistema	Gestione Carrello
Descrizione	Questo sottosistema permette di immagazzinare i pacchetti che l'utente intende acquistare
Servizi	
Servizi	Descrizione
Aggiungere pacchetto	Questo servizio consente di poter aggiungere i pacchetti nel carrello
Eliminare pacchetto	Questo servizio consente di poter eliminare un pacchetto dal carrello
Acquisto pacchetto	Questo servizio consente di acquistare un pacchetto.

Sottosistema	Gestione Ricerca
Descrizione	Questo sottosistema permette di cercare pacchetti
Servizi	
Servizi	Descrizione
Ricerca Per Filtri	Questo servizio permette di poter cercare, secondo determinati criteri, un pacchetto

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

Sottosistema	Gestione Visualizza Acquisti
Descrizione	Questo sottosistema permette di visualizzare i pacchetti acquistati
Servizi	Descrizione
Acquistati	Questo servizio permette di poter visualizzare lo storico degli acquisti

#### 4.3. Gestione Vendita

Sottosistema	Gestione Gestione Pacchetti
Descrizione	Questo sottosistema permette di gestire tutte le operazioni inerenti alla vendita dei pacchetti.
Servizi	Descrizione
Inserisci pacchetto	Questo servizio consente di aggiungere un pacchetto al sistema
Rimuovi pacchetto	Questo servizio consente di Rimuovere un pacchetto dal sistema
Modifica pacchetto	Questo servizio consente di Modificare un pacchetto già esistente nel sistema

Sottosistema	Gestione Visualizza Pacchetti
Descrizione	Questo sottosistema permette di visualizzare i pacchetti messi in vendita
Servizi	Descrizione
Venduti	Questo servizio permette di poter visualizzare lo storico delle vendite

Progetto: Mille Miglia In Viaggio	Versione: 1.0
Documento: System Design Document	Data: 01/02/2019

## 5. GLOSSARIO

**Utente:** rappresenta l'utilizzatore del sistema;

**Venditore:** rappresenta l'utilizzatore del sistema che intende vendere dei pacchetti;

**Acquirente:** rappresenta l'utilizzatore del sistema che intende acquistare dei pacchetti;

**Pacchetto:** servizio offerto dal sistema;

**Client:** componente che accede a servizi e risorse di un altro componente detto server;

**Server:** Componente che gestisce traffico di informazioni e fornisce servizi e risorse attraverso la rete;

**http:** protocollo di trasferimento di ipertesti che consente a due macchine, client e server di interagire attraverso un meccanismo di richiesta/risposta. Il client inoltra una richiesta al server, che verrà soddisfatta con la risposta di quest'ultimo;

**Web Browser:** applicazione software installata sul client che permette di visualizzare e navigare le risorse del Web;

**JSP:** tecnologia di programmazione web utilizzata per fornire contenuti dinamici;

**Servlet:** oggetti Java all'interno del server web che permettono di creare web applications la combinazione con JSP;

**DBMS:** sistema software per creazione, manipolazione e interrogazione efficiente di database;

**StartUP:** avvio di un sistema;

**Shutdown:** spegnimento di un sistema.