

POLITECNICO DI MILANO  
SCUOLA DI INGEGNERIA INDUSTRIALE E DELL'INFORMAZIONE  
CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA



Progetto di Ingegneria del Software 2  
**Travel Dream**

# DESIGN DOCUMENT

Responsabile:

Prof. Raffaella Mirandola

**Davide Piva**

Matricola: 820040

**Luca Putelli**

Matricola: 760123

**Matteo Varalta**

Matricola: 818913

# Sommario

1.	Descrizione generale .....	3
1.1	Panoramica .....	3
1.2	Descrizione dell'architettura .....	3
1.3	Scelte tecnologiche: .....	5
2.	Struttura dei dati.....	6
2.1	Modello relazionale generale.....	6
2.2	Modello logico dettagliato .....	7
2.3	Proprietà non trascurabili dei dati .....	12
3.	Struttura dell'applicazione .....	14
3.1	Modello EJB.....	14
3.2	Elementi della logica applicativa.....	15
4.	Struttura del client.....	17
4.1	Introduzione e obiettivi .....	17
4.2	Struttura dell'area pubblica e modello IDM .....	17
4.3	Page sketches dell'area pubblica .....	19
4.4	Struttura dell'area privata.....	23
4.5	Page sketches dell'area privata .....	26
5.	Modifiche al RASD .....	33
5.1	Errata corrige per il modello Alloy .....	33
5.2	Rimozione di permesso .....	33
5.3	Modifica dell'indirizzo e-mail.....	34
5.4	Numero di persone e viaggi regalabili.....	34
	Appendice: computo ore.....	35

# 1. Descrizione generale

Questo documento specifica, in maniera schematica e generale, la struttura e le scelte di design dell'applicazione web commissionata. Si tratta di una descrizione che completa e amplia il RASD, concentrandosi sul come i requisiti verranno implementati.

La struttura di questo Design Document ricalca quella dell'applicazione, per cui verranno presentate:

- le strutture dati, solamente da un punto di vista logico, senza occuparsi invece di come queste debbano essere fisicamente realizzate;
- il modello applicativo, per cui le basi su cui si regge l'applicazione web;
- il funzionamento e l'aspetto dell'interfaccia utente.

Il Design Document si rivolge soprattutto agli sviluppatori del progetto, sebbene possa essere utile (specialmente nelle parti più semplici) anche per gli stakeholders, come base di confronto su alcune scelte fatte.

## 1.1 Panoramica

TravelDream è un'applicazione web con due funzioni principali:

- la presentazione dei propri contenuti (offerte per viaggi);
- la vendita di offerte viaggio.

Queste due funzioni sono rigorosamente distinte, in quanto differiscono molto nella loro rappresentazione e implementazione.

Nel sito web, l'utente di fatto segue (con una certa libertà di navigazione) dei percorsi predefiniti: può vedere offerte e come si compongono, può vedere alberghi e come fare per soggiornarci, per farsi un'idea su quanto acquistare. Una volta effettuata la selezione del viaggio, può apportare modifiche, invitare amici oppure creare una gift list, in una propria area personale.

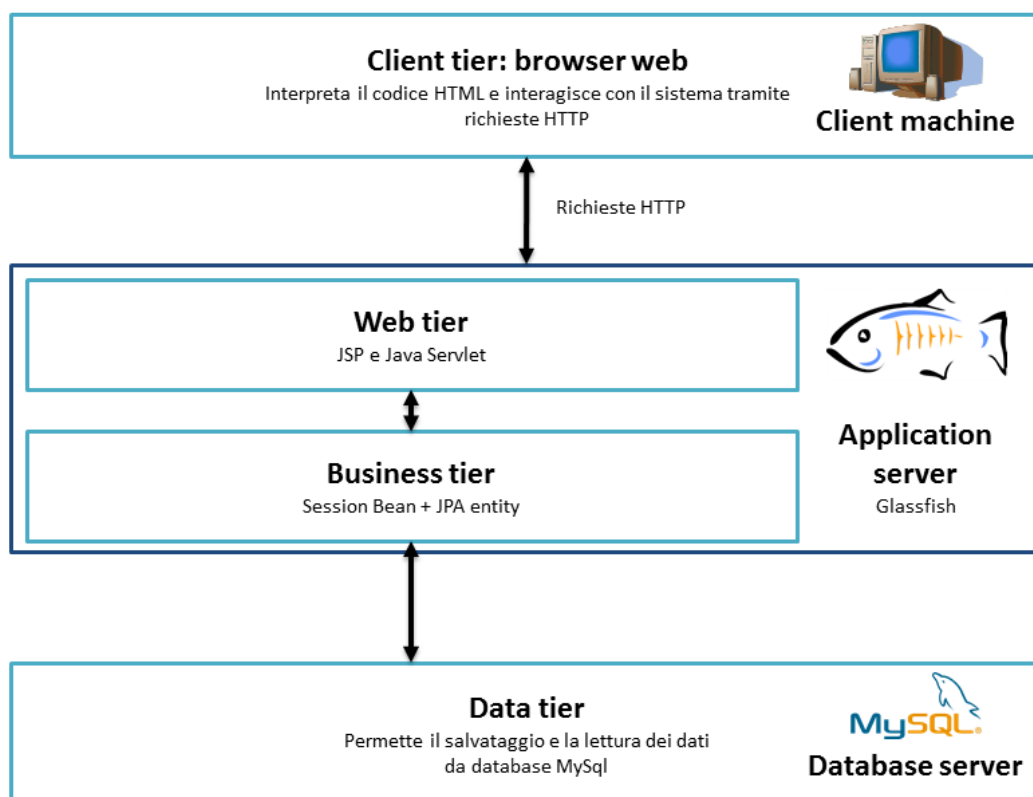
Lo stile che si è deciso di adottare è di tipo classico: il layout del sito è intuitivo, molto semplice e serve di fatto solo da cornice, da interfaccia per la navigazione, lasciando invece maggiore spazio a foto e descrizioni delle offerte, in modo che l'utente possa farsi un'idea immediata del prodotto e interessarsi all'acquisto.

## 1.2 Descrizione dell'architettura

La piattaforma da realizzare, schematizzata qui di seguito, si basa sulla tipica architettura multilivello: client tier, web tier, business tier e data tier.

- Il client tier, a contatto con l'utente, permette di visualizzare le informazioni in maniera semplice ed efficace attraverso diverse pagine web. Questa tecnologia è stata preferita a quella di una tipica applicazione client Java poichè è più semplice per l'utente usufruirne necessitando di un solo browser web. Inoltre non permette al sistema centralizzato di interfacciarsi con una versione obsoleta del client, la quale potrebbe essere incompatibile (anche magari solo parzialmente) con la tecnologia server.
- Il web client, sfruttando web beans, permette di generare la pagina relativa alla richiesta dell'utente interagendo con esso e con il Business tier. Le pagine generate sono statiche, a queste inoltre verrà applicato un insieme di fogli di stile in modo da avere un'interfaccia coerente fra pagine differenti, soddisfacendo anche requisiti estetici del sito web, importanti per promuovere il brand.
- Il business logic tier contiene la logica principale dell'applicazione, è formato da EJB Beans che permettono di interfacciare tra loro il web tier e il persistence tier
- Il persistent tier permette l'interazione con il database sfruttando, per ogni tabella, un entity bean che la rappresenta sotto forma di oggetti. Ciò permette di interagire con il database tramite le classiche operazioni CRUD, sfruttando una logica ad oggetti

Se la macchina client è unica per ogni utente, lo stesso non si può dire dei server DB e JEE: nel nostro caso questi sono in esecuzione sulla stessa macchina, ma nulla vieterebbe di installare il server su una macchina diversa del server per JEE.



### 1.3 Scelte tecnologiche:

Quasi tutte le scelte tecnologiche sono state imposte dai vincoli presentati nel RASD, esaminiamole comunque con ordine:

- la logica applicativa è realizzata con tecnologia J2EE, con Enterprise Java Bean (EJB) per ogni classe, utilizzando come application server Glassfish;
- l'interazione tra il client e la logica applicativa è realizzata con Servlet, che generano pagine HTML;
- il client vero e proprio è costituito da una serie di pagine HTML, con inserti di Javascript, corredate da fogli di stile CSS;
- il database è realizzato con il DBMS MySQL (Oracle), l'accesso a questo è gestito invece negli EJB;
- l'interazione tra i componenti del team, durante l'implementazione, avviene con l'ausilio della piattaforma Google Code e di Subversion, per garantire sincronizzazione e possibilità di lavoro anche a distanza;
- l'ambiente di sviluppo è l'IDE Eclipse Kepler.

## 2. Struttura dei dati

La progettazione della base di dati è avvenuta per gradi e, nello stesso ordine, viene presentata in questo Design Document:

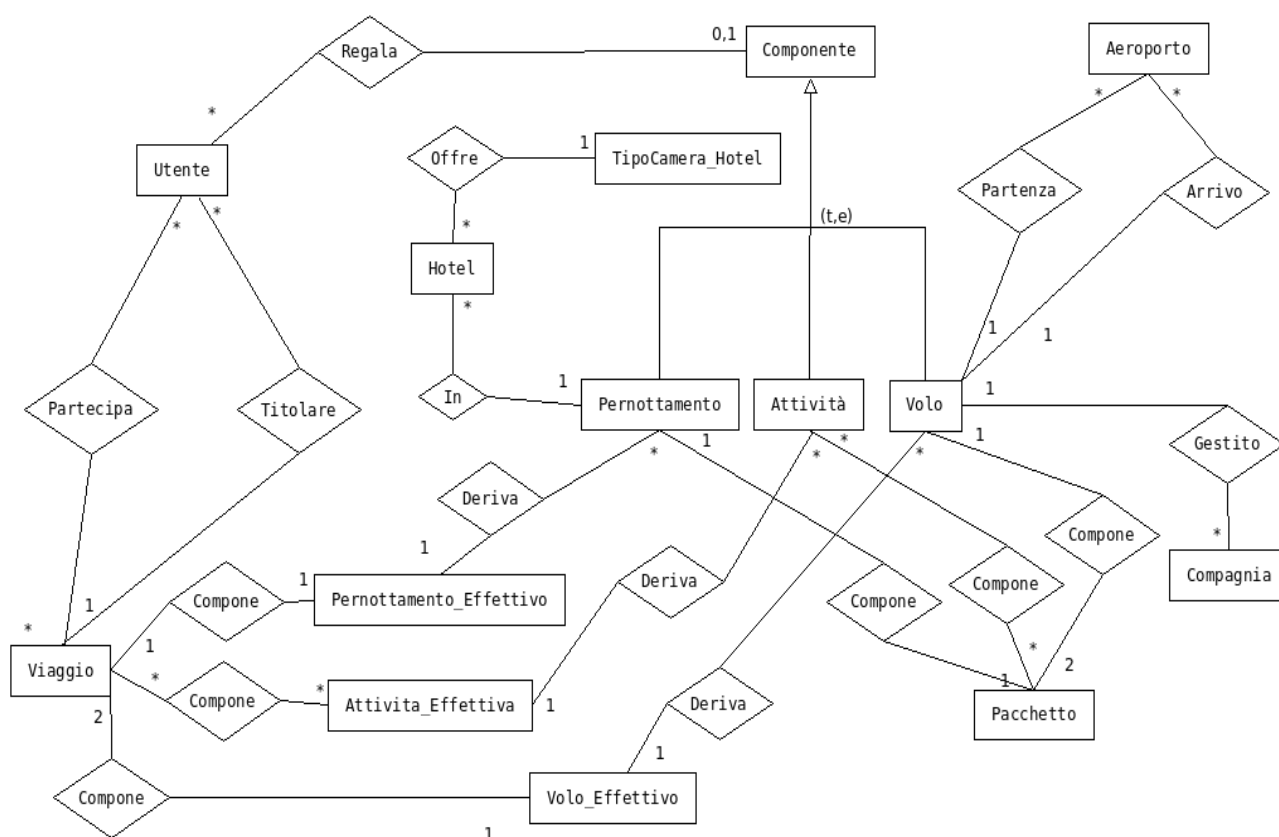
- innanzitutto si è dovuto delinare le entità principali, andando anche a capire come si relazionano tra loro; per questo, è stato costruito un diagramma E-R, senza attributi, al solo scopo di dare un'idea generale;
- successivamente, si è cercato di analizzare le singole entità, definendo gli attributi fondamentali e facendo alcune scelte di progettazione per rappresentare le diverse configurazioni delle istanze;
- poi si è analizzata l'integrità referenziale, andando a specificare maggiormente le chiavi esterne e risolvendo alcune problematiche di cardinalità, ponendo altre scelte e convenzioni;
- infine si è potuto creare un modello logico completo, che viene presentato come risultato finale della progettazione della base di dati.

In questa parte di documento, si rende conto solamente della parte logica e non dell'implementazione fisica. Il modello logico infatti è indipendente dal DBMS e necessario in caso di future modifiche nelle scelte tecnologiche.

### 2.1 Modello relazionale generale

Prima di mostrare il diagramma E-R, alcune precisazioni:

- l'entità Pacchetto rappresenta l'offerta proposta da TravelDream così com'è;
- l'entità Viaggio invece rappresenta l'effettivo viaggio acquistato dall'utente, derivato quindi da un Pacchetto ma con possibili modifiche;
- la differenza tra l'offerta teorica e l'acquisto vero si ripercuote anche sui vari componenti;
- il modello logico comporta la realizzazione di numerose tabelle ponte, per la rappresentazione delle relazioni molti a molti e di alcuni attributi che si palesano solo al momento della selezione del pacchetto;



## 2.2 Modello logico dettagliato

Qui di seguito, presentiamo prima in forma scritta le diverse entità con i rispettivi attributi. Come da standard, la chiave primaria è sottolineata; le chiavi esterne, invece, non sono rappresentate direttamente per non creare ulteriore confusione, tuttavia il nome utilizzato per indicare l'attributo dovrebbe rendere immediatamente comprensibile, per chi ha un po' di dimestichezza con la progettazione concettuale, la presenza o meno di un riferimento ad un'altra tabella.

Non sono presenti gli attributi necessari per la visualizzazione delle foto, che verranno aggiunti in fase di implementazione.

VIAGGIO(id,pernottamento,volòAndata,volòRitorno,città,prezzo,titolare,pagato,numeroPersone)

PERNOTTAMENTO(id,idHotel,tipo)

VIAGGIO\_PERNOTTAMENTO(idViaggio,idPernottamento,regalabile)

HOTEL(id,nome,città,indirizzo,telefono,descrizione)

TIPOCAMERE\_HOTEL(hotel,tipo,prezzo)

ATTIVITA(id,data,ora,titolo,descrizione,città,prezzo,)

VIAGGIO\_ATTIVITA(idAttivita,idViaggio,regalabile)

PACCHETTO\_ATTIVITA(idAttivita,idPacchetto)

VOLO(id,aeroportoPartenza,aeroportoArrivo,prezzo,idCompagnia,data,oraPartenza,oraArrivo)

VOLO\_VIAGGIO(idVolo,idViaggio,regalabile)

AEROPORTO(id,nome,città)

COMPAGNIA(id,nome,descrizione)

PACCHETTO(id,pernottamento,voloAndata,voloRitorno,città,titolo,descrizione,prezzo,tipologia,target)

UTENTE(email,nome,cognome,password,tipologia,attivo)

PARTECIPAZIONE(idViaggio,partecipante,pagato)

DONAZIONE\_VOLO(donatore,id)

DONAZIONE\_ATTIVITA(donatore,id)

DONAZIONE\_PERNOTTAMENTO(donatore,id)

Procediamo ora con una descrizione dettagliata delle entità e del tipo di attributi delineati nel modello logico:

UTENTE: contiene i classici attributi che identificano qualsiasi tipo di utente su un sito (email, nome, cognome e password). Per distinguere tra loro le varie categorie di utente che interessano al sistema abbiamo scelto di introdurre l'attributo di tipo enum tipologia (generico, impiegato, amministratore). Infine è presente l'attributo booleano attivo che sarà FALSE nel caso in cui una persona, invitata a partecipare ad un viaggio, non si sia ancora registrata su TravelDream.com, e verrà posta a TRUE non appena esso si sarà registrato. Nel frattempo il sistema conserverà solamente l'indirizzo e-mail dell'invitato ed aggiornerà gli altri campi una volta completata la registrazione. Le relazioni partecipa e regala saranno realizzate a livello logico da tabelle ponte: nel primo caso sarà presente l'indirizzo e-mail dei partecipanti e un



booleano pagato che indicherà l'avvenuto pagamento o meno, mentre nel secondo vi sarà l'indirizzo e-mail del donatore e gli id dei vari componenti regalati.

**PACCHETTO:** descrive i pacchetti inseriti nel sistema dagli impiegati. Essi vengono identificati con un id. Oltre ad un titolo incisivo, l'entità contiene informazioni sul pernottamento, sulle attività proposte (queste ultime saranno elencate in una tabella ponte, come da modello logico), sui voli, sulla città di destinazione e sul prezzo offerto, il tutto corredato da una descrizione. L'attributo tipologia sarà di tipo enum ed indicherà che tipo di viaggio l'utente andrà ad acquistare: romantico, avventura e così via. Infine l'attributo target, anch'esso di tipo enum, suddividerà i pacchetti in tre diverse categorie a seconda del numero di persone per cui sono pensati: singolo, coppia e gruppo.

**VIAGGIO:** in gran parte equivalente a **PACCHETTO**, ma con la differenza che le attività, il pernottamento e i voli possono essere stati modificati da un utente titolare prima della conferma dell'acquisto. In più raccoglie gli identificativi del titolare e di eventuali partecipanti (questi ultimi saranno elencati in una tabella ponte, come da modello logico). L'attributo numeroPersone conterrà il numero di persone che parteciperanno al viaggio. Questo attributo avrà un valore temporaneo solo fino a quando il booleano pagato diventerà TRUE, ovvero dopo che TravelDream avrà correttamente ricevuto il pagamento del viaggio.

**VOLO:** oltre all'id che identificherà univocamente un volo, inteso come singolo tragitto da un aeroporto ad un altro compiuto da tutti i partecipanti al viaggio, all'interno del nostro database, questa entità conterrà informazioni sugli aeroporti di partenza e di arrivo, sulla data e sull'ora di partenza ed arrivo, sulla compagnia che offre il servizio ed infine sul prezzo offerto da TravelDream.

**VOLO\_VIAGGIO:** questa entità metterà in relazione tramite i corrispettivi id un volo con un viaggio in particolare. In più permetterà di tenere traccia del fatto che il dato volo per il dato viaggio sia regalabile o meno.

**ATTIVITA:** ogni singola attività viene identificata da TravelDream con un id. Esse sono caratterizzate da una data ed un'ora, un titolo e una breve descrizione. Ci sarà inoltre la città in cui la determinata attività si svolgerà in modo da poter verificare che l'utente non scelga di partecipare ad attività che si svolgono in città diverse da quella del viaggio. Infine sarà anche qui presente l'attributo con il prezzo offerto da TravelDream.

**ATTIVITA\_VIAGGIO:** questa entità metterà in relazione tramite i corrispettivi id una o più attività con un viaggio in particolare. In più permetterà di tenere traccia del fatto che la data attività per il dato viaggio sia regalabile o meno.

**PERNOTTAMENTO:** anch'esso identificato tramite un id univoco, un pernottamento per TravelDream è inteso come una permanenza (anche per più persone) in un determinato hotel e in un determinato tipo di stanza (lowcost, smart e dream). Il periodo del pernottamento coincide con quello del viaggio quindi, per evitare ridondanze, si è scelto di tralasciare gli attributi con la data di checkin e di checkout.

**PERNOTTAMENTO\_VIAGGIO:** questa entità metterà in relazione tramite i corrispettivi id un pernottamento con un viaggio in particolare. In più permetterà di tenere traccia del fatto che il dato pernottamento per il dato viaggio sia regalabile o meno.

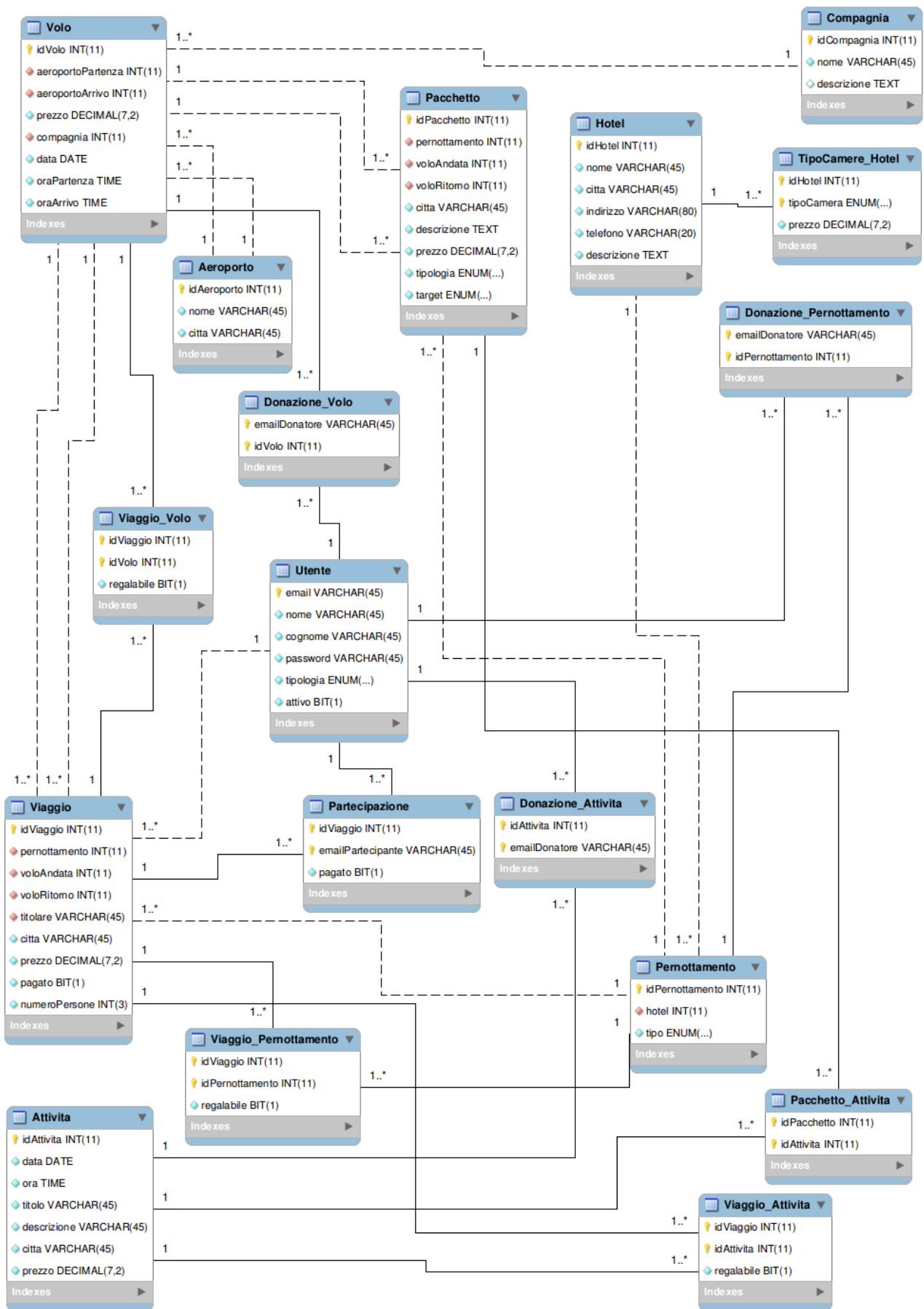
**HOTEL:** per semplicità abbiamo scelto di identificare anche ogni hotel partner di TravelDream con un id. Ogni hotel sarà descritto da un nome e dalla città (per la quale valgono le stesse considerazioni fatte per le attività). Ci sarà poi una descrizione testuale dell'hotel accompagnata dall'indirizzo e dal numero di telefono. Un hotel può offrire una o più tipologie di camera, esse saranno elencate in una apposita tabella ponte che indicherà anche il prezzo per le determinate camere per notte.

**CAMERA\_HOTEL:** questa entità conterrà l'identificativo dell'hotel se il dato hotel possiede un determinato tipo di camera (lowcost, smart e dream).

**AEROPORTO:** ogni aeroporto, descritto dal proprio nome e dalla città in cui risiede, verrà identificato con un'id dato che esistono città con più di un aeroporto.

**COMPAGNIA:** anche le compagnie aeree saranno semplicemente individuate da un id univoco e dal loro nome. Sarà inoltre presente un campo descrizione che conterrà eventuali informazioni sulla compagnia (regolamenti, suggerimenti, ecc.).

Nella pagina seguente, proponiamo una rappresentazione grafica di quanto qui descritto:



## 2.3 Proprietà non trascurabili dei dati

La definizione iniziale, il modello logico e altri strumenti affini non pongono dei problemi di coerenza della base di dati; tuttavia, quando si modella una situazione reale, si deve tenere conto anche di tutta una serie di vincoli “di buon senso” che vengono dati per scontati da qualsiasi persona.

Su espressa richiesta del committente, in questa sezione vengono presentati alcuni vincoli fondamentali della base di dati (che saranno poi implementati), affinché l'applicazione modelli esclusivamente una situazione reale come l'organizzazione di viaggi.

### Coerenza temporale

La partenza precede l'arrivo e il viaggio di andata si svolge prima di quello di ritorno. Durante il soggiorno inoltre si possono svolgere anche diverse attività, ma solamente in momenti diversi.

Questo si traduce nelle seguenti regole per il database:

- nella tabella Viaggio, la data di voloAndata deve precedere quella di voloRitorno. Questo è un vincolo che si ripercuote su due tuple di tipo Volo, che appunto contengono la data;
- per la tabella ponte Viaggio\_Actività, in ogni tupla deve valere che la data dell'attività (che è chiave esterna, per cui si deve controllare l'attributo data della tabella Attività) deve essere sempre compresa, esclusi i giorni di arrivo e partenza, tra la data di voloAndata e voloRitorno;
- analogamente, nella tabella Viaggio\_Actività, fissando un idViaggio, ogni attività correlata a quello specifico viaggio non potrà avere sia la data che l'ora uguali;
- tutti questi vincoli si applicano alla stessa maniera per le tabelle Pacchetto e Pacchetto\_Actività.

### Coerenza spaziale

E' assolutamente naturale che, in un viaggio aereo, ci si sposti da una città all'altra e quindi da un aeroporto all'altro. Una volta arrivati, il soggiorno è limitato all'interno della città e lì avviene il pernottamento in hotel e tutte le attività.

Nel database, questo significa che:

- nella tabella Viaggio, aeroportoPartenza deve essere diverso da aeroportoArrivo. Nei casi in cui inoltre ci siano più aeroporti in una città (caso comunque piuttosto frequente), questo vincolo non basta e deve essere necessariamente implementato che la città della tupla collegata da aeroportoPartenza (di tipo Aeroporto), abbia città diversa da quella di aeroportoArrivo;
- nella tabella Viaggio, l'attributo città deve coincidere con la città di aeroportoArrivo;
- per la tabella ponte Viaggio\_Actività, in ogni tupla deve valere che la città dell'attività (che è chiave esterna, per cui si deve controllare l'attributo città della tabella Attività) deve coincidere con quella del Viaggio a cui è collegata;

- l'hotel connesso dalla tabella Pernottamento (che è collegata a Viaggio mediante la chiave esterna pernottamento) deve avere lo stesso attributo città contenuto nella tupla di Viaggio;
- si procede allo stesso modo per questi componenti in relazione con Pacchetto.

### Impostazioni per le offerte

Se il viaggio viene modificato, il prezzo finale è dato dalla somma dei prezzi dei componenti.

Questo è un vincolo che non si ripercuote direttamente nella progettazione del database, bensì va implementato nell'applicazione, mediante un opportuno metodo ricalcolo che possa andare a recuperare i prezzi dei diversi componenti e a calcolarne il nuovo.

Tutti i componenti di un viaggio regalabile sono regalabili

Questo invece è un vincolo che va realizzato sia nel database sia via software, garantendo che le diverse tabelle ponte (Viaggio\_Pernottamento, Viaggio\_Volo, Viaggio\_Attività) abbiano l'attributo booleano regalabile impostato a true solamente se il viaggio è stato impostato regalabile dal cliente titolare.

## 3. Struttura dell'applicazione

In questa sezione viene mostrata la composizione, a grandi linee, della logica applicativa.

Innanzitutto, vengono presentati gli EJB che delineano le entità coinvolte e le relazioni che intercorrono tra queste, analogamente a quanto già fatto per il database.

Successivamente, si dà conto anche del lato vero e proprio di funzionamento dell'applicazione web, con tutte le operazioni base che il sistema deve supportare ad ogni richiesta dell'utenza.

Per entrambi gli obiettivi, si farà largo utilizzo di diagrammi UML delle classi, questa volta però con un maggiore livello di dettaglio, presentando anche i principali attributi e metodi.

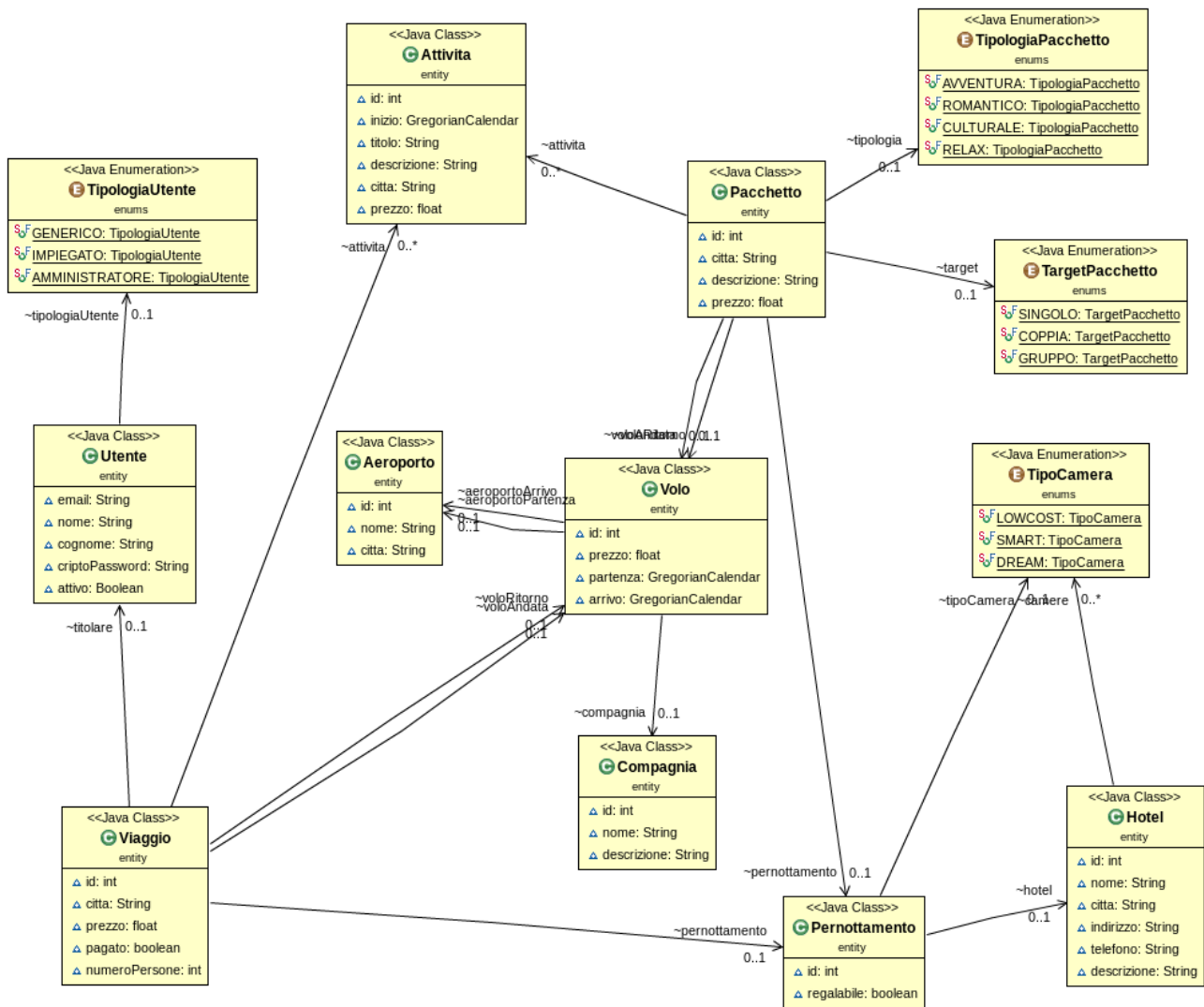
Questa parte di design è, come si può facilmente intuire, la più complessa e articolata e potrà subire modifiche di varia natura durante l'implementazione, all'incorrere di problemi o nuove esigenze.

### 3.1 Modello EJB

L'applicazione, come espressamente richiesto dal committente, è realizzata in J2EE e deve quindi rispettarne gli standard. Per fare ciò, è necessario che tutte le classi che delineano il modello dell'applicazione siano JavaBean, cioè delle particolari classi Java con le seguenti proprietà:

- devono necessariamente avere un costruttore di default;
- devono contenere tutti i getter e setter;
- devono essere serializzabili;
- non devono contenere metodi per la gestione degli eventi.

Attenendosi a queste proprietà, il risultato è quello di una classe che è fortemente legata all'oggetto rappresentato e soprattutto rispecchia quasi perfettamente una tabella del database (chiaramente con un cambio di logica, da quella relazione a quella ad oggetti tipica di Java).



## 3.2 Elementi della logica applicativa

Oltre chiaramente al modello dell'applicazione, sono necessarie diverse classi Java per organizzare e compiere le funzioni vere e proprie, sulle entità modellate.

La logica applicativa è una delle parti più complesse e passibili di modifica, quindi la soluzione qui ipotizzata è da intendersi come guida, come traccia su cui si potrà costruire l'applicazione reale. In ogni caso, quanto mostrato nel seguente diagramma UML delle classi (rappresentate solamente nei metodi principali e di fatto non hanno praticamente interazioni tra di loro) sta a significare le entità fondamentali e le operazioni fondamentali che dovranno essere svolte e, se non in questa forma, in una del tutto analoga a questa.

Autenticazione
+login(email:String,password:String): boolean +registra(nome:String,cognome:String,email:String, password:String): boolean +logout(): boolean

GestioneProfili
+aggiungiProfilo(utente:Utente): void +modificaProfilo(utente:Utente): void +rimuoviProfilo(utente:Utente): void

GestioneImpiegati
+aggiungiPermesso(impiegato:Utente): void +rimuoviPermesso(impiegato:Utente): void +getListaPermessi(): ArrayList<Utente>

DBInterface
+connetti(): boolean +disconnetti(): boolean

GestioneViaggio
+nuovoViaggio(pacchetto:Pacchetto): void +aggiungiAttivita(attivita:ArrayList<Attivita>): void +rimuoviAttivita(attivita:ArrayList<Attivita>): void +modificaPernottamento(pernottamento:Pernottamento): boolean +modificaDate(data1:Date,data2:Date): boolean +modificaVolo(volo:Volo,isAndata:boolean): boolean +modificaViaggiatori(numero:int): void +rendiRegalabile(): void +aggiungiPartecipazioni(partecipanti:ArrayList<Utente>): void +rimuoviPartecipante(partecipante:Utente): boolean +paga(): boolean

GestioneOfferte
+creaPacchetto(p:Pernottamento,v1:Volo,v2:Volo, attivita:ArrayList<Attivita>, prezzo:float): boolean +modificaVoloAndata(volo:Volo): boolean +modificaVoloRitorno(volo:Volo): boolean +modificaPernottamento(pernottamento:Pernottamento): boolean +aggiungiAttivita(attivita:ArrayList<Attivita>): boolean +rimuoviAttivita(attivita:Attivita): boolean +modificaPrezzo(prezzo:float): void +modificaTitolo(titolo:String): void +modificaDescrizione(descrizione:String): void

GestioneComponenti
+creaPernottamento(hotel:Hotel,camera:TipoCamera, ): void +creaHotel(nome:String,citta:String,indirizzo:String, telefono:String,descrizione:String): void +creaAttivita(data:Date,ora:Time,titolo:String, descrizione:String,citta:String, prezzo:float): boolean +cancellaHotel(hotel:Hotel): void +cancellaAttivita(attivita:Attivita): void



## 4. Struttura del client

### 4.1 Introduzione e obiettivi

Per la modellazione del client, abbiamo preferito suddividere fortemente le fasi di presentazione dei contenuti e quella di gestione. Dal punto di vista dell'interazione con l'utente, la suddivisione è lampante: la prima è infatti aperta a qualsiasi visitatore, che si limita a navigare facendo qualche click, passando da una pagina all'altra secondo la propria fantasia, limitandosi a vedere ciò che il sito offre; nella seconda, l'utente deve avere effettuato l'accesso, può fare solo alcune procedure predefinite (acquistare, modificare un viaggio, modificare il proprio profilo) rigidamente controllate dall'applicazione, in quanto fondamentali nell'erogazione del servizio.

Abbiamo mantenuto questa divisione anche in questo Design Document, pertanto la prima parte viene presentata mediante il diagramma P-IDM, che dà un resoconto intuitivo su come si relazionano le pagine tra di loro; la seconda parte invece viene presentata con l'UX Diagram, con intento maggiormente descrittivo.

Obiettivo principale dell'applicazione web è l'intuitività: l'utente deve poter vedere e progettare il proprio viaggio in poche semplici mosse, pena la perdita di interesse e la rinuncia; non è trascurabile però nemmeno la chiarezza dell'esposizione: l'utente deve avere ben chiari quali siano le conseguenze di quanto sta per fare, specialmente nei passaggi chiave (la modifica, il pagamento), per questo si ritiene che, in alcune fasi, sia opportuno chiedere più volte conferma, fornire riepiloghi, affinché l'utente possa capire e riflettere bene prima di impegnarsi in una transazione vera e propria con la compagnia.

### 4.2 Struttura dell'area pubblica e modello IDM

Il C-IDM presentato nel RASD dà solo un'idea generale di come si relazionano gli argomenti all'interno dell'applicazione, ma non dà nessuna informazione vera e propria di come si sistemano le pagine. Un'ulteriore specificazione degli argomenti è l'L-IDM (che qui non riteniamo utile presentare, in quanto poco rappresentativo), ma il diagramma che si può veramente tradurre come un'insieme strutturato di pagine web è il P-IDM.

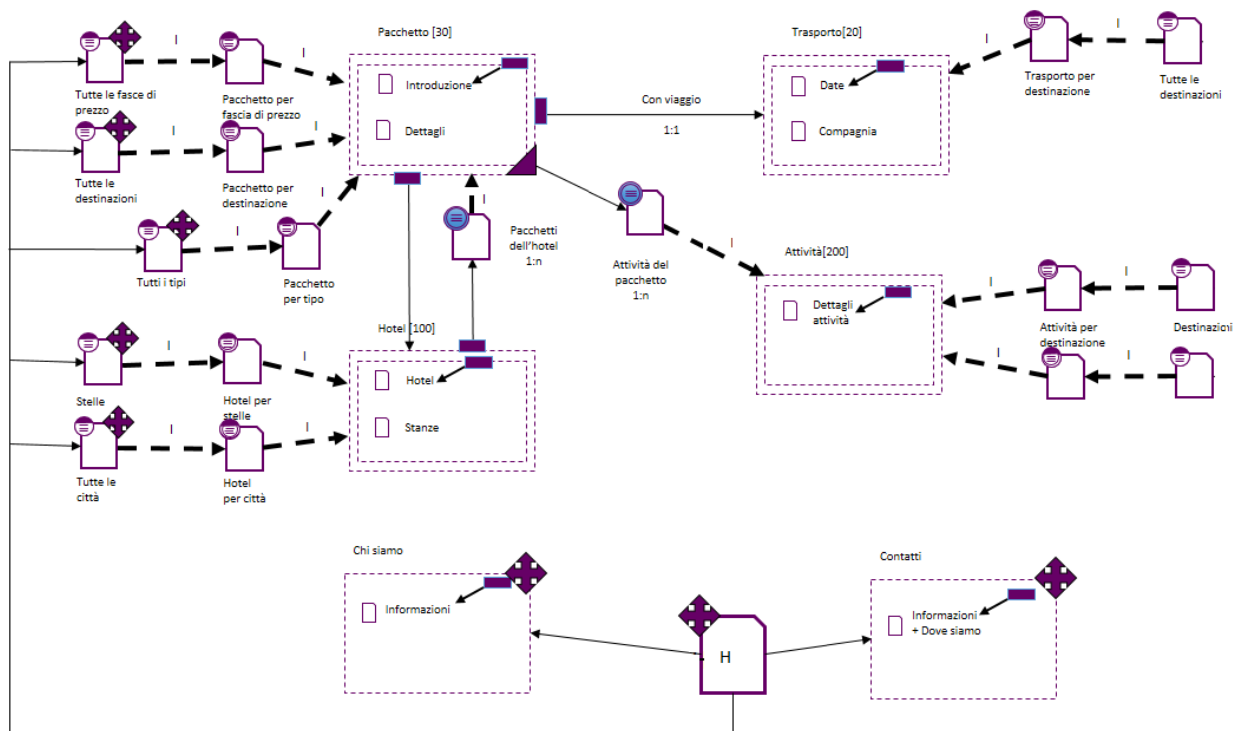
Anche in questo caso, non essendo l'IDM uno standard, abbiamo ritenuto necessario fornire una spiegazione-legenda di quanto rappresentato in seguito.

Nel P-IDM si mantiene la suddivisione per topic dei precedenti, si possono però vedere le pagine fisiche che lo compongono. I riquadri multipli (del Multiple Topic Pacchetto, per esempio) sono quindi scomposti in due pagine, rappresentati con dei fogli all'interno (in questo caso Introduzione e Dettagli). Ogni istanza dell'entità Pacchetto, per cui ogni viaggio presentato, avrà quindi una sua pagina di introduzione, che è la prima che si vede quando si accede a questo genere di contenuto, e una di dettaglio, in cui si potranno avere informazioni supplementari sui componenti e altro.

I Group Of Topic di fatto vengono sostituiti da delle pagine che forniscono liste di istanze, raggruppate secondo lo stesso filtro. Alcune di queste possono anche essere essere presenti in tutte le pagine del sito, come landmark (rappresentato con la freccia quadrupla), per un accesso più comodo da parte dell'utente, dovunque esso si trovi. La selezione può essere chiaramente multipla (per esempio, da una lista di fasce di prezzo si può vedere i viaggi di una particolare fascia di prezzo), e si mantiene ugualmente la doppia linea per identificare la molteplicità.

Le relazioni tra entità, che nel C-IDM sono semplicemente suggerite, possono essere di due tipi:

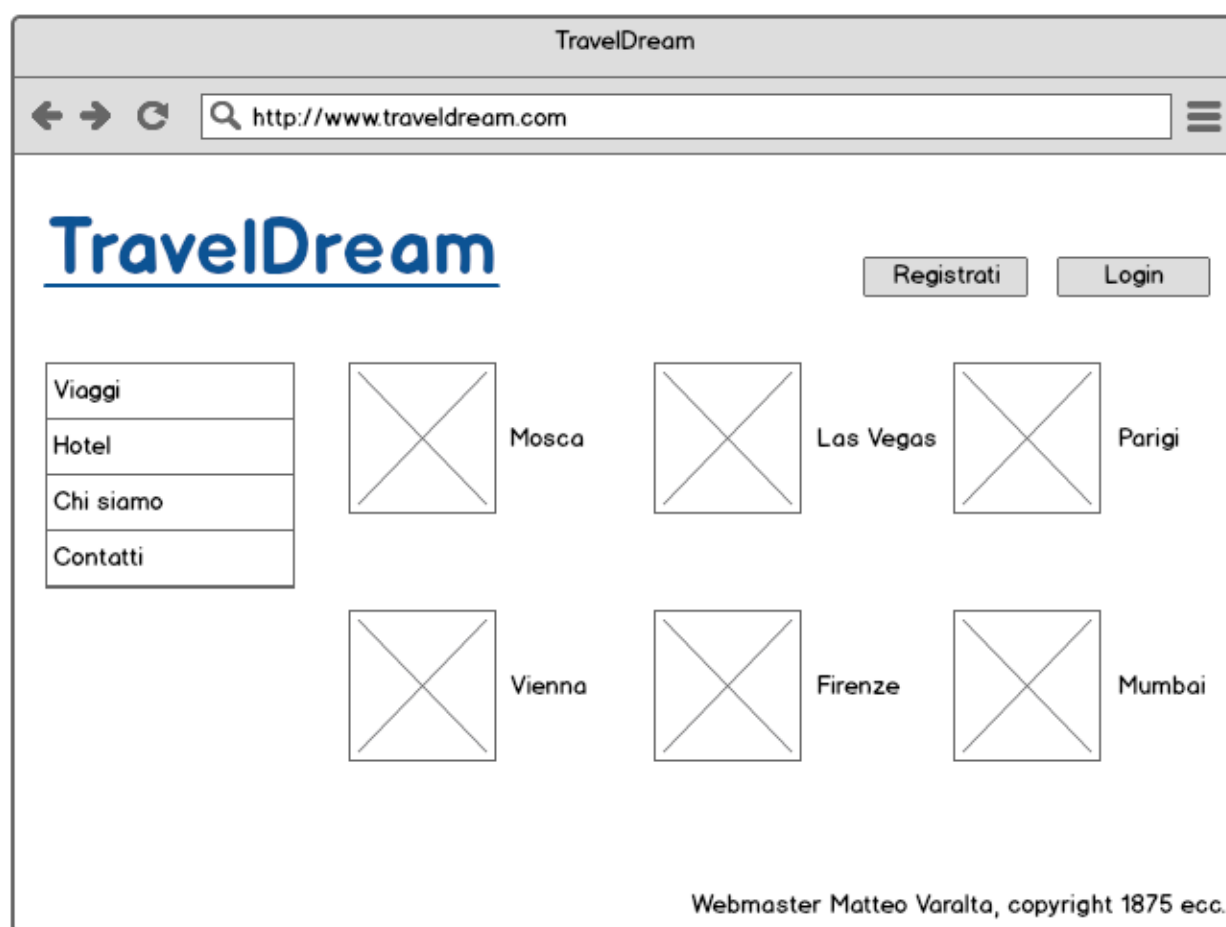
- se si passa a una singola pagina (per esempio l'hotel di un pacchetto), tramite un link diretto, e quindi una semplice freccia nel diagramma;
- se invece si vogliono vedere una serie di pagine che sono correlate all'argomento in cui ci si trova (per esempio le molteplici attività di un pacchetto), tramite una pagina di transizione, che fornirà quindi una lista di link alle pagine specifiche, con possibilità di scelta per l'utente.



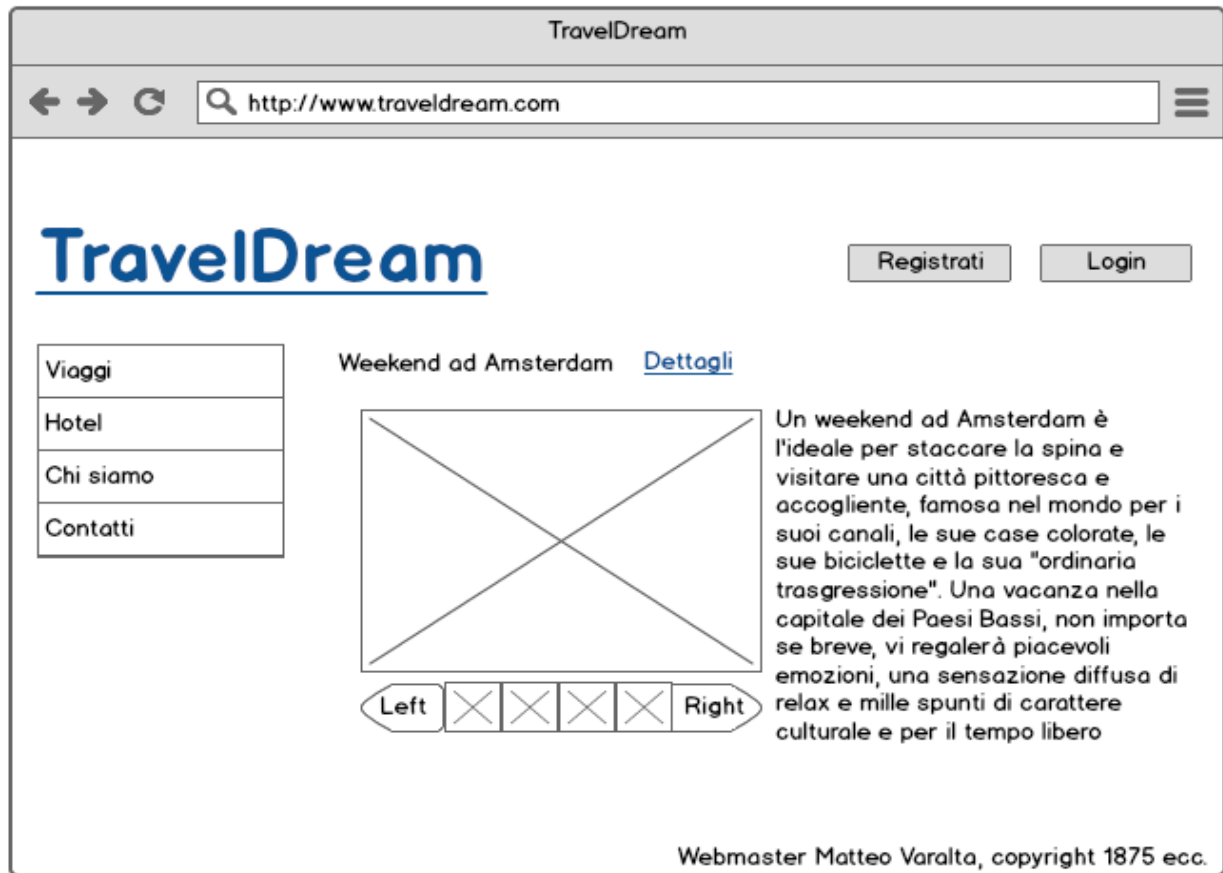
## 4.3 Page sketches dell'area pubblica

Di seguito proponiamo alcuni modelli di pagina significativi della nostra applicazione. Questi cosiddetti page sketches sono stati realizzati con il software Balsamiq Mockups e sono, per scelta, molto schematici e privi di dettagli. Sono utili al committente per avere un'idea generale del colpo d'occhio che avrà sull'utente la nostra applicazione e per gli sviluppatori come modello da seguire nell'implementazione.

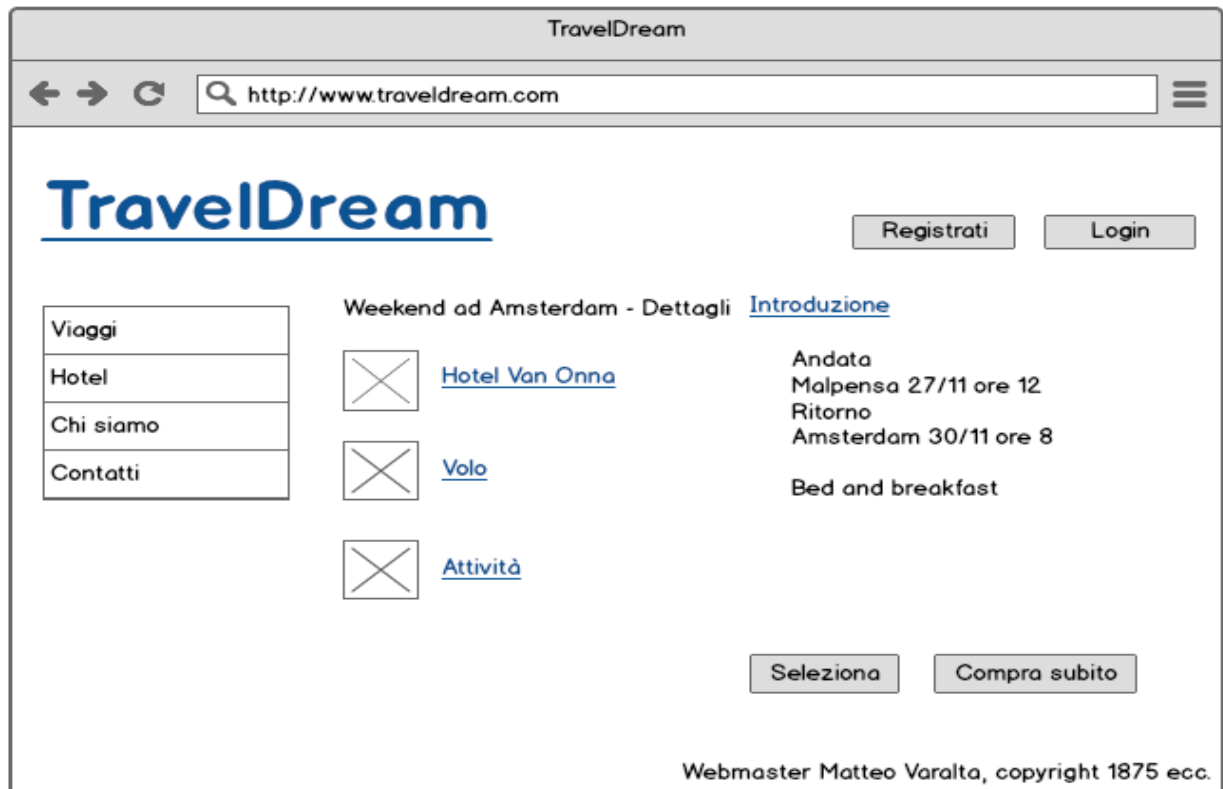
Home page



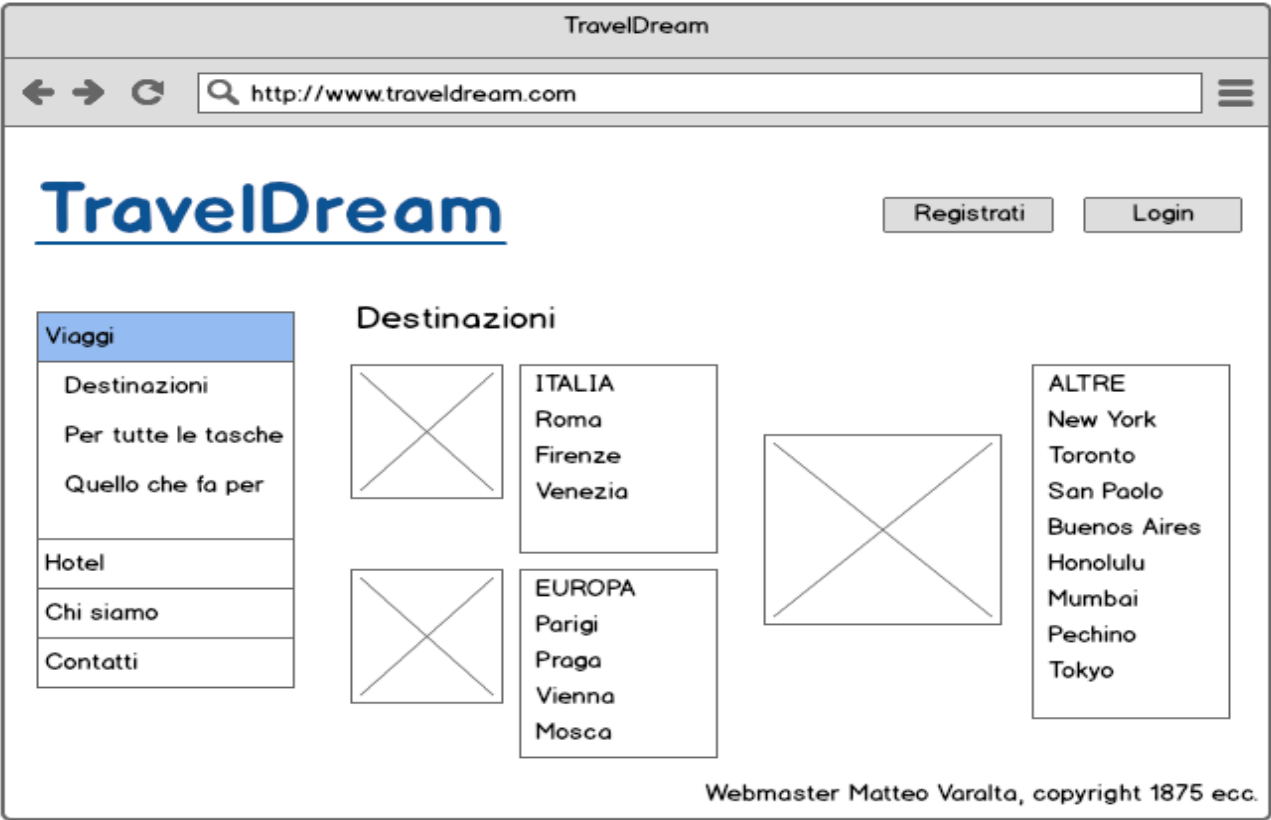
## Pacchetto – Introduzione



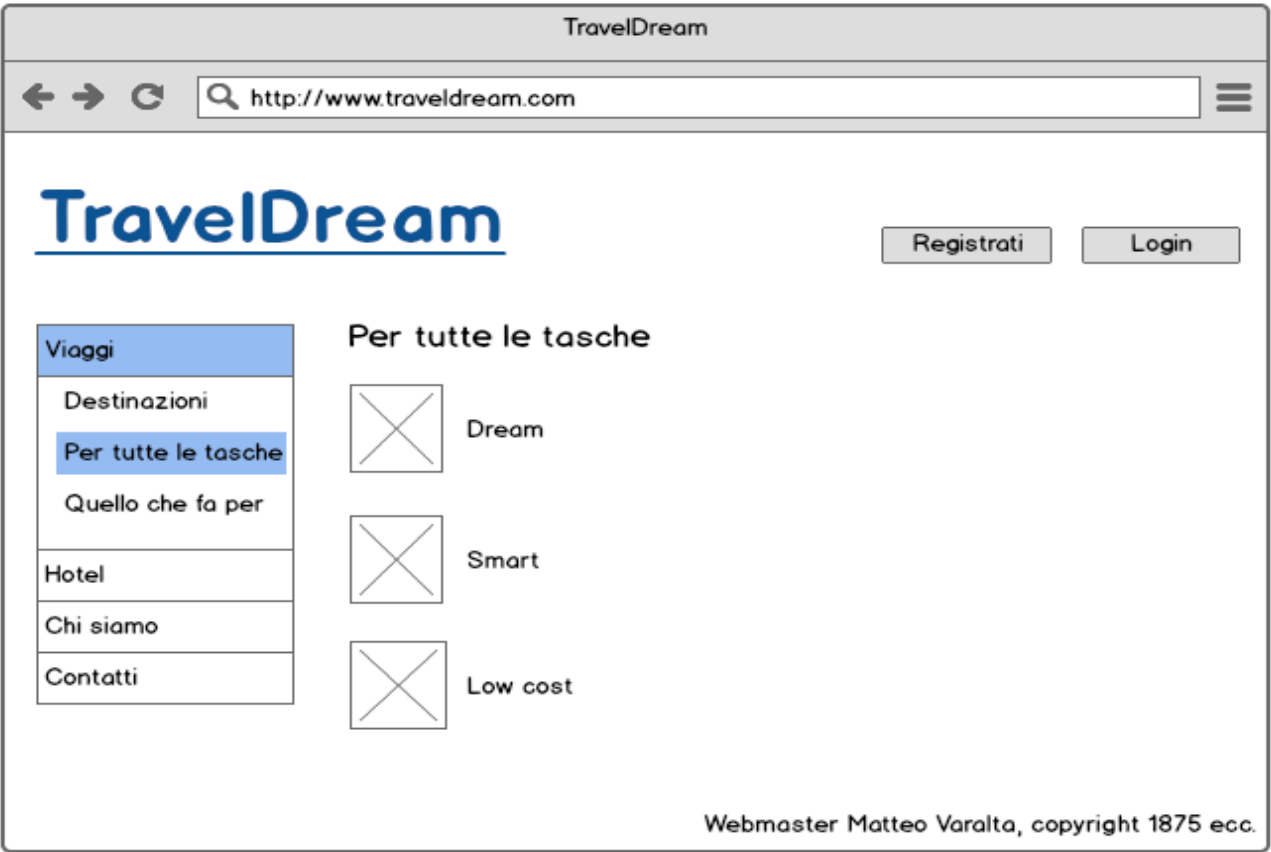
## Pacchetto – Dettagli



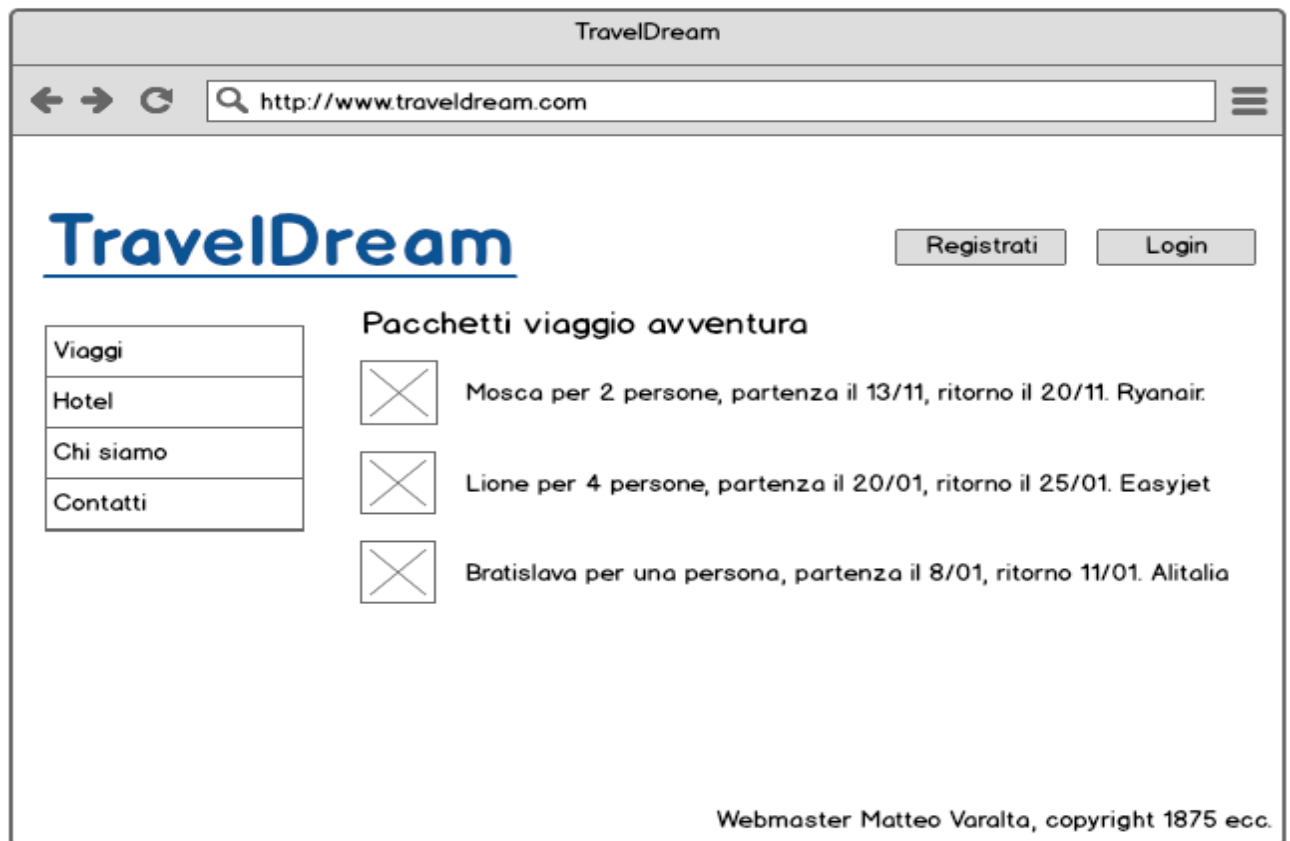
Destinazioni possibili



Fasce di prezzo

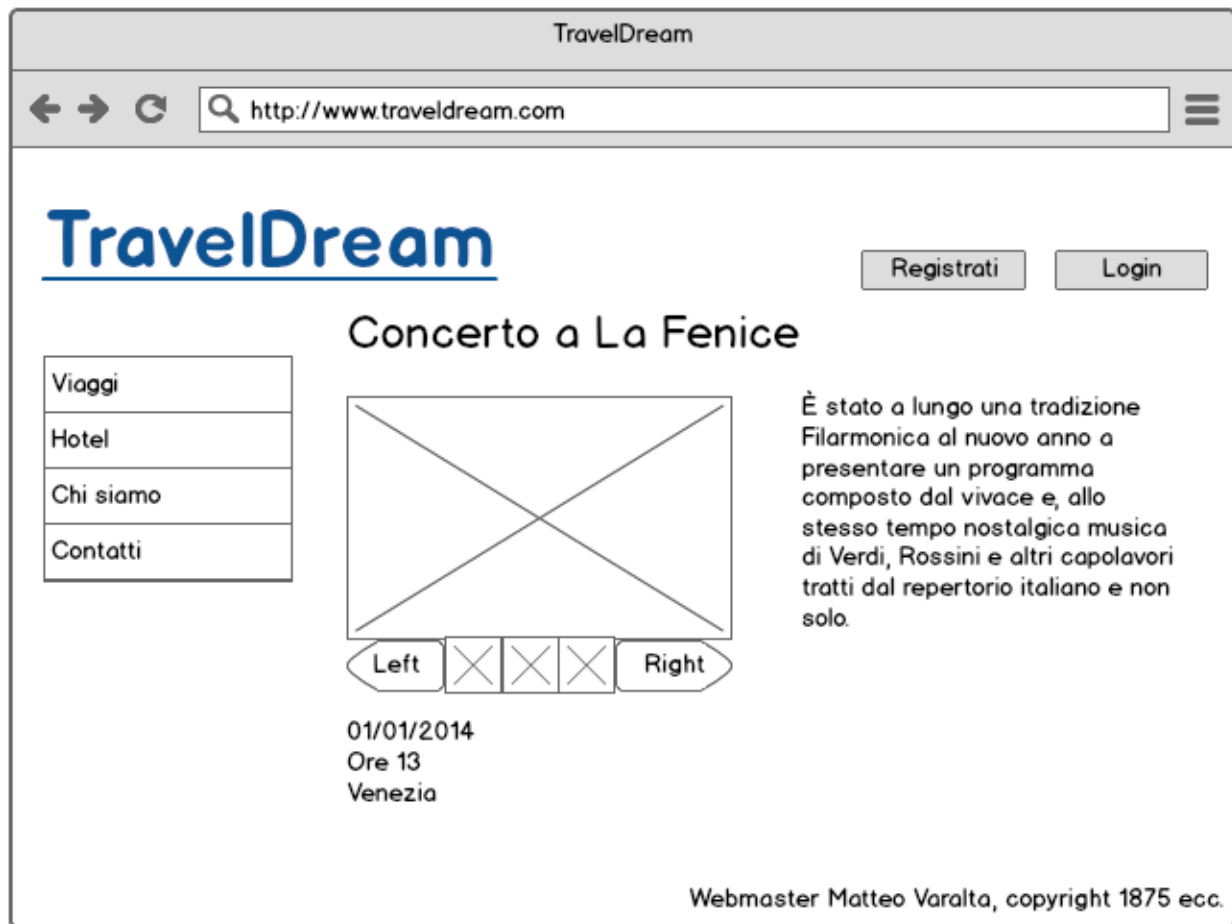


Pagina di suddivisione dei pacchetti (secondo varie possibilità)



Hotel





#### 4.4 Struttura dell'area privata

Il sito ha un'ampia area privata nella quale l'utente può effettuare le operazioni più importanti del suo rapporto con TravelDream.

Da qui l'utente può accedere a diverse procedure:

- la modifica del proprio profilo;
- l'acquisto e la modifica di pacchetti selezionati;
- vedere lo storico dei propri viaggi, compresi quelli che deve ancora fisicamente effettuare;
- vedere i viaggi a cui è stato invitato oppure che può regalare.

In aggiunta, gli impiegati hanno una consolle di gestione che permette loro di:

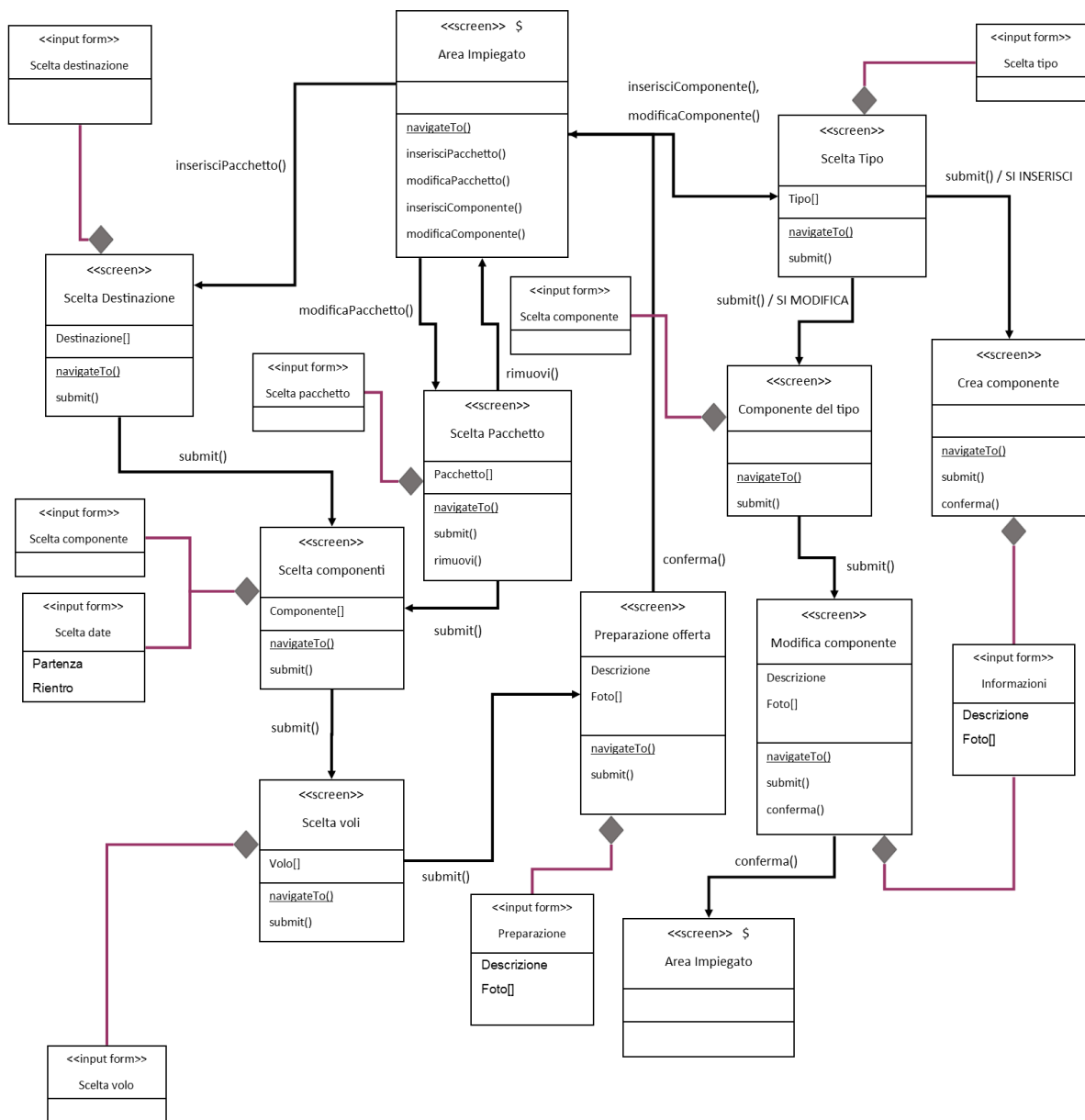
- inserire nuovi componenti;
- modificare o eliminare componenti già esistenti;
- inserire nuovi pacchetti;
- modificare o eliminare componenti già esistenti.

L'amministratore di sistema avrà chiaramente una consolle di gestione modificata, nella quale potrà modificare componenti e pacchetti oppure occuparsi dei permessi degli impiegati, aggiungendone o revocandone.

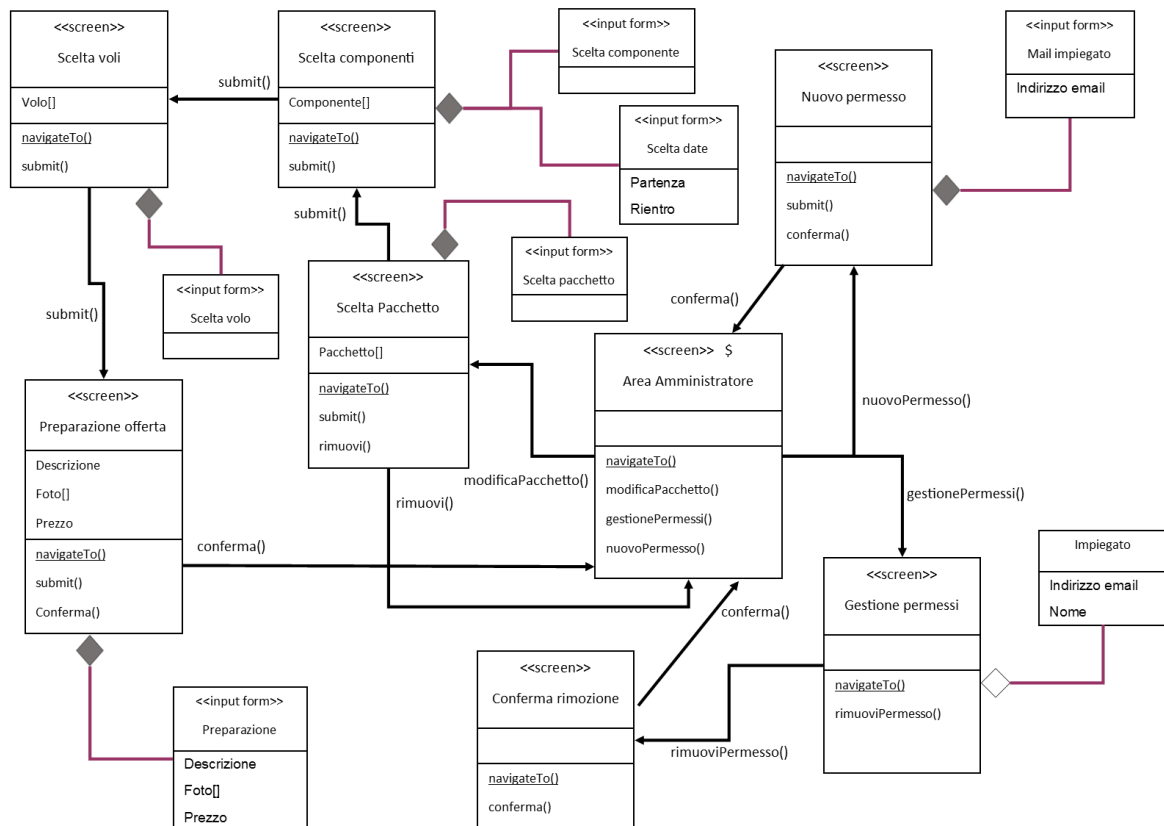
[illegible]



## UX dell'Impiegato



## UX dell'Amministratore di sistema



## 4.5 Page sketches dell'area privata

Analogamente all'area pubblica, proponiamo i page sketches della parte privata del sito.

La sostanziale differenza tra le due tipologie è che da un lato abbiamo una presentazione (per cui conta molto anche l'aspetto estetico, il posizionamento degli elementi), dall'altro abbiamo una serie di moduli, in cui l'obiettivo principale è la raccolta di dati: è necessaria quindi chiarezza e precisione, senza dar luogo alla possibilità per l'utente di confondersi durante una procedura.

## Area personale dell'utente

TravelDream

← → ↺

Q http://www.traveldream.com

☰

# TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

Davide Piva

davidepiva@libero.it

Modifica del profilo

- Francesco Ruggeri ti ha invitato a partecipare a [questo viaggio](#)
- Fabiana Volpi ti suggerisce [un regalo](#)

Stai organizzando:

- Natale sul Canal Grande

23/12/2013 - 3/01/2014

- Carnevale a Rio

14/02/2014 - 19/02/2014

I miei prossimi viaggi:

- Weekend berlinese

10/11/2013 - 12/11/2013

Webmaster Matteo Varalta, copyright 1875 ecc.

## Modifica di un viaggio

TravelDream

← → ↺

Q http://www.traveldream.com

☰

# TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

## Natale sul Canal Grande

Viaggio per 2 - 950 euro a persona

Hotel Le Frasche

Viale Europa 25 - Venezia

Attività

- Gita in gondola 21/12/2013
- Concerto di Capodanno a La Fenice 01/01/2014

Date e voli:

- Malpensa 19/12/2013 ore 8 (Ryanair)
- Venezia-Tessera 03/01/2014 ore 17 (Ryanair)

Acquista

Cancella

Modifica

Modifica

Modifica

Modifica

Vorrei...

Invita

Se vuoi che qualcuno te lo regali, clicca su 

Vorrei...

Se invece vuoi partire in compagnia 

Invita

Webmaster Matteo Varalta, copyright 1875 ecc.

## Partecipazione a un viaggio

TravelDream

← → ↻

Q http://www.traveldream.com

☰

# TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

## Natale sul Canal Grande

Hotel Le Frasche

Viale Europa 25 - Venezia

Attività

- Gita in gondola 21/12/2013
- Concerto di Capodanno 01/01/2014

Date e voli:

- Malpensa 19/12/2013 ore 8 (Ryanair)
- Venezia-Tessera 03/01/2014 ore 17 (Ryanair)

La tua quota è: 950 euro

Partecipo

Non partecipo

Webmaster Matteo Varalta, copyright 1875 ecc.

## Viaggi effettuati in passato

TravelDream

← → ↻

Q http://www.traveldream.com

☰

# TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

## Esperienze passate

- Surfing USA

15/07/2013 - 30/07/2013

- Oktoberfest

03/10/2013 - 10/10/2013

Webmaster Matteo Varalta, copyright 1875 ecc.

Pagina degli inviti

TravelDream

← → ↻

Q http://www.traveldream.com

≡

TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

Natale sul Canal Grande

Partiranno con te: 950 euro a persona  
Il tuo totale: 1900 euro

2 persone 

Modifica

In più, hai invitato

Andrea Panella

Matteo Vergani

Paolo Siligardi

Rimuovi

Rimuovi

Rimuovi

Hanno già confermato

Sara Ferrari

Hai dimenticato qualcuno?

e-mail utente

Invita

Webmaster Matteo Varalta, copyright 1875 ecc.

Pagina in cui si effettua il pagamento

TravelDream

← → ↻

Q http://www.traveldream.com

≡

TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

Effettua il pagamento

L'importo è di: 950 euro

Inserisci il numero di carta di credito e la scadenza della carta

/

Inserisci il codice di 3 cifre sul retro:

Informazioni su modalità di pagamento, tipologie supportate,  
trattamento dei dati personali ecc. ecc.

Confermo

Annulla

Webmaster Matteo Varalta, copyright 1875 ecc.

Pannello dell'impiegato (quello dell'amministratore è una lista analoga)

TravelDream

← → ↻

Q http://www.traveldream.com

≡

TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

Gestione dei contenuti

Inserisci una nuova offerta

Modifica un'offerta

Inserisci un nuovo componente

Modifica un componente

Webmaster Matteo Varalta, copyright 1875 ecc.

Scelta dei componenti durante la modifica

TravelDream

← → ↻

Q http://www.traveldream.com

≡

TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

Scelta dei componenti

Hotel a Venezia

Attività dell'offerta

Partenza

Hotel Le Frasche

☐ Visita Guidata

☒ Concerto a La Fenice

☒ Gita in gondola

/ /

Rientro

/ /

Procedi

Webmaster Matteo Varalta, copyright 1875 ecc.

## Preparazione delle descrizioni

TravelDream

← → ↻

Q http://www.traveldream.com

☰

TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

## Preparazione dell'offerta

Inserisci la descrizione

Prezzo a persona

Foto dell'offerta

[Foto 1](#)  
[Foto 2](#)

Carica una foto

url della foto

Carica

Procedi

Webmaster Matteo Varalta, copyright 1875 ecc.

## Modifica di un componente

TravelDream

← → ↻

Q http://www.traveldream.com

☰

TravelDream

Ciao, Davide 

Logout

Viaggi

Hotel

Chi siamo

Contatti

## Modifica di "Concerto a La Fenice"

Modifica la descrizione

Un'occasione unica per ascoltare, in un prestigioso teatro veneziano, i grandi capolavori della musica classica italiana

Prezzo a persona

130

Foto del componente

[Foto 1](#)  
[Foto 2](#)  
[Foto 3](#)

Rimuovi

Rimuovi

Rimuovi

Carica una foto

url della foto

Carica

Procedi

Webmaster Matteo Varalta, copyright 1875 ecc.

TravelDream

← → ↺

🔍 http://www.traveldream.com

☰

# TravelDream

Ciao, Matteo Logout

Viaggi

Hotel

Chi siamo

Contatti

## Elenco degli impiegati

Davide Piva davidepiva@tiscali.it	<button>Rimuovi permesso</button>
Luca Putelli lucaputelli@libero.it	<button>Rimuovi permesso</button>
Franco Bonetti bonettifranco@hotmail.com	<button>Rimuovi permesso</button>

Webmaster Matteo Varalta, copyright 1875 ecc.



## 5. Modifiche al RASD

### 5.1 Errata corrige per il modello Alloy

Nel modello Alloy va apportata la seguente modifica:

```
sig Hotel{  
  camera : set Camera,  
  sedeHotel : one Citta  
}
```

E' di fatto una modifica marginale, in quanto non varia il modello generato, che rimane valido. E' evidente che in un Hotel ci possano essere più tipi di camere invece che uno solo.

### 5.2 Rimozione di permesso

Viene aggiunto il seguente caso d'uso:

Titolo	Rimozione di permesso
Attori	Amministratore di sistema
Precondizioni	L'utente è impiegato di TravelDream e ha ottenuto in passato i permessi di modifica dei pacchetti
Eventi	<ol style="list-style-type: none"><li>1. L'amministratore, nel proprio pannello personale, clicca sul bottone di gestione dei permessi</li><li>2. Il sistema riceve la richiesta e rimanda l'utente a una pagina contenente tutti i permessi conferiti</li><li>3. L'amministratore sceglie l'impiegato (in base a nome e cognome) a cui revocare il permesso e clicca sul bottone di rimozione</li><li>4. Il sistema chiede un'ulteriore conferma all'utente</li><li>5. L'amministratore conferma la volontà di revocare il permesso all'utente selezionato</li><li>6. Il sistema revoca il permesso all'impiegato</li></ol>

Postcondizioni	L'impiegato non ha più i permessi di modificare i pacchetti
Eccezioni	

### 5.3 Modifica dell'indirizzo e-mail

Contrariamente a quanto specificato nel RASD è resa possibile, da parte dell'utente, la modifica del proprio indirizzo e-mail di riferimento. La modifica è avvenuta perchè, nonostante questa rimanga chiave primaria nella tabella Utente del database, non sono stati riscontrati problemi derivati dal permettere all'utente questo tipo di azione.

### 5.4 Numero di persone e viaggi regalabili

Viene effettuata anche una supposizione di buon senso riguardo al rendere un viaggio regalabile. Prima dell'acquisto, un titolare può modificare varie volte il numero di persone che verranno con lui (sono esclusi i partecipanti). Questa decisione tuttavia si ripercuote sul prezzo e quindi, in caso di viaggio regalabile, anche sui donatori. Al fine di evitare conflitti quindi, una volta che un viaggio è stato reso regalabile da vari donatori, il cliente titolare non potrà più effettuare modifiche al numero di persone coinvolte.

## Appendice: computo ore

Data	Piva	Varalta	Putelli
28/11/2013	4	2	2
04/12/2013	2	2	
06/12/2013			1
09/12/2013		2	
10/12/2013			2
11/12/2013	3	2	4
12/12/2013	3	2	4
13/12/2013			1
14/12/2013	1		
16/12/2013	1	1	1
17/12/2013			2
18/12/2013	1	1	1
Totale	15	12	18