

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



RASD

TRAVEL DREAM

Progetto di ingegneria del software II

Sara Marchesini, Sebastiano Mariani, Riccardo Mastellone

Contents

| | | |
|----------|--|-----------|
| 1 | Introduzione | 4 |
| 1.1 | Scopo del documento | 4 |
| 1.2 | Scopo del software | 4 |
| 1.3 | Definizioni e abbreviazioni | 4 |
| 1.3.1 | Definizioni | 4 |
| 1.3.2 | Abbreviazioni | 5 |
| 1.4 | Referenze | 5 |
| 1.5 | Panoramica | 5 |
| 2 | Descrizione generale | 6 |
| 2.1 | Descrizioni componenti del Prodotto | 6 |
| 2.1.1 | Interfacce di sistema | 6 |
| 2.1.2 | Interfaccia utente | 6 |
| 2.1.3 | Interfacce hardware | 11 |
| 2.1.4 | Interfacce software | 11 |
| 2.1.5 | Interfacce di comunicazione | 11 |
| 2.1.6 | Memoria | 11 |
| 2.1.7 | Operazioni | 11 |
| 2.2 | Assunzioni | 11 |
| 2.2.1 | Pacchetto | 11 |
| 2.2.2 | Ricerca di un pacchetto | 12 |
| 2.2.3 | Prezzo dei prodotti base | 12 |
| 2.2.4 | Personalizzazione di un pacchetto | 12 |
| 2.2.5 | Cancellazione di un pacchetto | 12 |
| 2.2.6 | Modifica di un pacchetto | 12 |
| 2.2.7 | Inserimento di un componente singolo | 12 |
| 2.3 | Implementazioni future | 12 |
| 2.4 | Descrizione funzionalità prodotto | 13 |
| 2.4.1 | Attori e funzionalità | 13 |
| 2.4.1.1 | Cliente | 13 |
| 2.4.1.2 | Dipendente | 13 |
| 2.4.1.3 | Amministratore | 14 |
| 2.5 | Vincoli | 14 |
| 3 | Requisiti specifici | 15 |
| 3.1 | Scenari | 15 |
| 3.1.1 | Registrazione al sito | 15 |
| 3.1.2 | Ricerca di un pacchetto | 15 |
| 3.1.3 | Personalizzazione di un pacchetto | 15 |
| 3.1.4 | Invito a un amico | 15 |
| 3.1.5 | Acquisto pacchetto | 16 |
| 3.1.6 | Aggiungi a lista desideri | 16 |
| 3.1.7 | Invia lista desideri | 16 |
| 3.1.8 | Crea pacchetto | 16 |
| 3.1.9 | Aggiungi pacchetto | 17 |
| 3.1.10 | Modifica pacchetto | 17 |
| 3.1.11 | Cancella pacchetto | 17 |
| 3.1.12 | Storico ordini | 18 |
| 3.1.13 | Elimina dipendente | 18 |
| 3.1.14 | Aggiungi dipendente | 18 |

| | | |
|----------|--|-----------|
| 3.2 | Casi d'uso | 19 |
| 3.2.1 | Registrazione | 20 |
| 3.2.2 | Login | 21 |
| 3.2.3 | Logout | 22 |
| 3.2.4 | Ricerca di un pacchetto per destinazione | 22 |
| 3.2.5 | Ricerca di un pacchetto per tipo | 23 |
| 3.2.6 | Visualizzazione scheda di un pacchetto | 24 |
| 3.2.7 | Personalizzazione di un pacchetto | 26 |
| 3.2.8 | Invita amico | 28 |
| 3.2.9 | Acquista pacchetto | 29 |
| 3.2.10 | Aggiungi a lista desideri | 30 |
| 3.2.11 | Visualizza lista desideri | 31 |
| 3.2.12 | Invia lista desideri | 31 |
| 3.2.13 | Cancella dalla lista desideri | 32 |
| 3.2.14 | Crea pacchetto | 33 |
| 3.2.15 | Modifica catalogo | 35 |
| 3.2.16 | Aggiungi pacchetto | 35 |
| 3.2.17 | Ricerca di un pacchetto per nome | 37 |
| 3.2.18 | Edita pacchetto | 38 |
| 3.2.19 | Cancella pacchetto | 39 |
| 3.2.20 | Storico ordini | 40 |
| 3.2.21 | Aggiungi dipendente | 40 |
| 3.2.22 | Cancella dipendente | 40 |
| 3.3 | Diagramma delle classi | 41 |
| 4 | Appendice | 42 |
| 4.1 | Alloy | 42 |
| 4.1.1 | Assunzioni | 42 |
| 4.1.2 | Analisi formale | 42 |
| 4.1.3 | Mondi generati | 48 |
| 4.1.3.1 | Mondo pacchetto consistente | 48 |
| 4.1.3.2 | Mondo prenotazione consistente | 49 |
| 4.1.3.3 | Mondo generale | 50 |

1 Introduzione

1.1 Scopo del documento

Lo scopo del documento è dare una descrizione dettagliata del nuovo software proposto all'azienda TravelDream, illustrando come una soluzione software-based permetta di semplificare il processo di vendita e gestione dei pacchetti vacanza.

Il documento intende mostrare tutti i vari aspetti dello sviluppo del sistema spiegandone funzionalità e vincoli.

Questo documento è principalmente rivolto ai dirigenti della TravelDream e servirà come riferimento al team di programmatori per lo sviluppo della prima versione.

1.2 Scopo del software

Lo scopo dell'applicazione TravelDream è quello di fornire un sistema software ottimale che permetta ai dipendenti dell'azienda di gestire l'offerta e la vendita di pacchetti vacanze, e ai clienti di consultare e acquistare il pacchetto desiderato.

L' applicazione sarà disponibile su qualsiasi browser visitando il sito www.traveldream.com.

A seconda delle credenziali immesse il sistema renderà disponibili le funzioni adeguate al grado di privilegio dell'utente loggato:

- il cliente potrà liberamente consultare il catalogo delle offerte, scegliere, personalizzare e creare un pacchetto e invitare gli amici a prendere visione del sito;
- il dipendente potrà aggiungere, modificare o eliminare i pacchetti vacanza;
- l'amministratore potrà assegnare ed eliminare le credenziali d'accesso dei dipendenti.

Gli obiettivi prefissati da TravelDream sono i seguenti:

- [G1] Estendere la visibilità dell'azienda grazie a internet
- [G2] Agevolare i clienti nella consultazione del catalogo
- [G3] Aiutare i clienti nella ricerca e creazione della vacanza adatta a loro
- [G4] Facilitare i clienti nella prenotazione della vacanza
- [G5] Agevolare i clienti nella condivisione della vacanza con altre persone
- [G6] Offrire servizi ad hoc ai clienti, quali la lista desideri
- [G7] Semplificare l'aggiornamento delle offerte dell'agenzia
- [G8] Semplificare la gestione dell'intero sistema

1.3 Definizioni e abbreviazioni

1.3.1 Definizioni

| Parola | Definizione |
|----------------|---|
| Cliente | Utente registrato al sito |
| Dipendente | Impiegato della TravelDream con credenziali per l'accesso da dipendente al sito |
| Amministratore | Capo della TravelDream con credenziali uniche per la gestione dei dipendenti |
| Prodotto base | Volo/hotel/attività secondarie |
| Pacchetto | Vacanza organizzata comprendente volo, hotel e attività secondarie |

1.3.2 Abbreviazioni

| Abbreviazione | Definizione |
|---------------|-----------------------------------|
| G | Goal (obiettivo) |
| FR | Requisito funzionale |
| NFR | Requisito non funzionale |
| DBMS | Database Management System |
| JEE | Java Enterprise Edition |
| API | Application Programming Interface |

1.4 Referenze

- Linee guida progetto: Progetto AA 2013-2014 (TravelDream).pdf
- IEEE Recommended Practice for Software Requirements <http://ieeexplore.ieee.org/xpl/mostRecentIssue.jsp?punumber=>
- Modello Alloy: travelDream.als

1.5 Panoramica

Il documento è organizzato nel seguente modo:

- Sezione 1: Introduzione, fornisce una presentazione del sistema software da sviluppare;
- Sezione 2: Descrizione generale, descrive i fattori generali che influenzano il sistema, le funzionalità che dovrà mettere a disposizione degli utenti e i vincoli che dovrà rispettare;
- Sezione 3: Requisiti specifici, contiene un'analisi dettagliata dei requisiti attuata tramite la stesura di scenari e casi d'uso e utilizzando diagrammi UML per identificare le funzionalità che il sistema deve offrire per il soddisfacimento dei requisiti;
- Sezione 4: Appendice, fornisce la rappresentazione schematica della realtà attraverso Alloy.

2 Descrizione generale

Questa sezione non descrive i requisiti specifici, ma si limita a fornire una panoramica generale di tutti gli aspetti e di tutte le funzionalità del sistema

2.1 Descrizioni componenti del Prodotto

Il software è un sistema stand-alone che non estende nè fa parte di un software già esistente in commercio.

L'accesso al software avviene esclusivamente via browser e per il momento non prevede applicazioni esterne (applicazioni mobile, ecc.) che si interfacciano con esso.

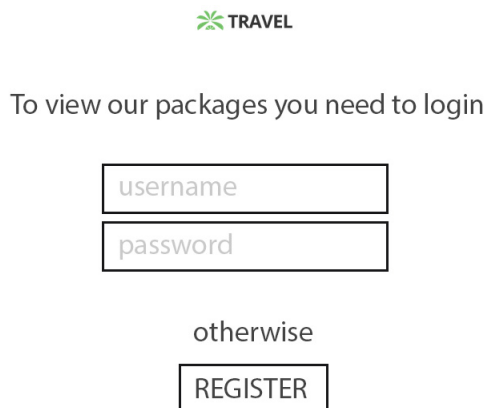
2.1.1 Interfacce di sistema

Il sistema non prevede la creazione di API per permettere ad altri sistemi esterni di interfacciarsi con esso, ma si limiterà ad interfacciarsi con un database MySQL.

2.1.2 Interfaccia utente

Il sistema fornisce un'intefaccia estremamente semplice ed intuitiva sia ai clienti sia ai dipendenti sia all'amministratore.

Questi mokup illustrano in modo schematico e minimale le varie pagine di navigazione:



✱ TRAVEL

To view our packages you need to login

username

password

otherwise

REGISTER

Figure 1: pagina di benvenuto

Il primo rappresenta la pagina di benvenuto che compare digitando l'indirizzo base www.traveldream.com. In questa clienti, dipendenti e amministratore possono loggarsi e un utente non registrato può registrarsi.



Please fill up the form to register

| |
|----------|
| name |
| surname |
| email |
| password |
| REGISTER |

Figure 2: pagina di registrazione

Si accede a questa pagina cliccando l'opzione "registrati" nella pagina di benvenuto.



Home Build your package Contacts Your Profile

WHERE WOULD YOU LIKE TO GO?

| | | |
|--|--|--|
| Type a city, the kind of place or an activity (eg beach, romantic, skiing, mountain) | | |
| <div>IMAGE</div> <div>PackageNAME Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer fringilla adipiscing.</div> <div>SELECT</div> | <div>IMAGE</div> <div>PackageNAME Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer fringilla adipiscing.</div> <div>SELECT</div> | <div>IMAGE</div> <div>PackageNAME Lorem ipsum dolor sit amet, consectetur adipiscing elit. Integer fringilla adipiscing.</div> <div>SELECT</div> |
| <div>IMAGE</div> | <div>IMAGE</div> | <div>IMAGE</div> |

Figure 3: homepage cliente

Questa è l'homepage del sito per gli utenti loggati come clienti: questi possono effettuare una ricerca oppure scegliere tra i pacchetti sponsorizzati.

YOUR WHISHLIST

You sent this package to **2** people. 1 confirmed, 1 pending.

PackageName 04/12/2013 - 20/12/2013
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris sagittis lorem dictum,
venenatis enim ac, vestibulum mi. Nam dignissim nisl a dignissim commodo.
Hotel HotelName, Flight AirLines

DELETE


PackageName 04/12/2013 - 20/12/2013
Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris sagittis lorem dictum,
venenatis enim ac, vestibulum mi. Nam dignissim nisl a dignissim commodo.
Hotel HotelName, Flight AirLines

SEND

DELETE

Figure 4: pagina lista desideri

Questa pagina racchiude tutti i pacchetti aggiunti da un cliente alla sua lista desideri e le operazioni possibili su questa.



[Home](#)
[Build your package](#)
[Contacts](#)
[Your Profile](#)

PackageNAME

Hotel

HotelNAME 4* 126\$

Flight

Brussels Airlines

From Milan Malpensa to Brussels National

Departure 02/12/2013 (14.23-17.11)

Return (08/12/2013 19.25-21.13)

Total Cost: XXX \$

<

IMAGES

>

IMAGE

HotelName 4*

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.

Integer fringilla adipiscing.

DETAILS

CHANGE

LOGO

Brussels Airlines

Lorem ipsum dolor sit amet,

consectetur adipiscing elit.

Integer fringilla adipiscing.

CHANGE

IMAGE

Planned Activities

. City Tour

- New Year's Eve Party

- Jogging in the park

CHANGE


INVITE FRIEND

ADD TO WISHLIST

BUY

Figure 5: pagina info pacchetto

Questa è la pagina cui si accede premendo il tasto “select” di un pacchetto, e riporta tutte le informazioni e le operazioni disponibili.




[Add Package](#) [Orders History](#) [Manage Hotels](#)

Type any package detail (name, description, tags) to filter results Q

| | |
|--|--|
| PackageName 04/12/2013 - 20/12/2013 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris sagittis lorem dictum, venenatis enim ac, vestibulum mi. Nam dignissim nisl a dignissim commodo. Hotel HotelName, Flight AirLines | <input type="button" value="EDIT"/> <input type="button" value="DELETE"/> |
| PackageName2 04/12/2013 - 20/12/2013 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris sagittis lorem dictum, venenatis enim ac, vestibulum mi. Nam dignissim nisl a dignissim commodo. Hotel HotelName, Flight AirLines | <input type="button" value="EDIT"/> <input type="button" value="DELETE"/> |
| PackageName3 04/12/2013 - 20/12/2013 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris sagittis lorem dictum, venenatis enim ac, vestibulum mi. Nam dignissim nisl a dignissim commodo. Hotel HotelName, Flight AirLines | <input type="button" value="EDIT"/> <input type="button" value="DELETE"/> |
| PackageName4 04/12/2013 - 20/12/2013 Lorem ipsum dolor sit amet, consectetur adipiscing elit. Mauris sagittis lorem dictum, venenatis enim ac, vestibulum mi. Nam dignissim nisl a dignissim commodo. Hotel HotelName, Flight AirLines | <input type="button" value="EDIT"/> <input type="button" value="DELETE"/> |

Figure 6: homepage dipendente


Questa è l'homepage del sito per gli utenti loggati come dipendenti: comprende la lista di tutti i pacchetti e le operazioni che si possono svolgere su di essi.




[Home](#) [Orders History](#) [Manage Hotels](#)

Package name

Package description

Departure datetime 

Return Datetime 

Hotel name

Calculated Hotel Cost: XX \$

How many to sell? ▼

How many people? ▼

Airline

Flight Cost

+ Add Activity

SAVE

Total Calculated Cost: XXX \$

Figure 7: pagina aggiungi pacchetto

Questa è la pagina che permette a un dipendente di aggiungere un pacchetto al catalogo: fornisce i campi da riempire con i dettagli del pacchetto.

2.1.3 Interfacce hardware

l'unica interfaccia hardware necessaria per l'utilizzo del sistema è un normale personal computer o eventualmente un dispositivo mobile dotato di connessione ad internet

2.1.4 Interfacce software

- **DBMS:**

- Nome: MySQL;
- Versione: 5.6.14;
- Source: <http://dev.mysql.com/downloads/>;

- **Application Server:**

- Nome: Glassfish;
- Versione: 4.0;
- Source: <https://glassfish.java.net/download.html>;

- **Sistema operativo:**

- il software deve girare su un qualunque sistema operativo che supporti la Java virtual machine e il DBMS descritti sopra

2.1.5 Interfacce di comunicazione

Il sistema userà le porte di default per le comunicazioni Http e l'interfacciamento con il DBMS

| Protocollo | Porta | Servizio |
|------------|-------|----------|
| TCP | 80 | Http |
| TCP | 3306 | MySQL |

2.1.6 Memoria

Il sistema deve essere installato su una macchina che rispetti i seguenti requisiti minimi di memoria:

- RAM: 2 GB
- HDD: 20 GB

2.1.7 Operazioni

Un utente può interagire con il sistema secondo 3 livelli di privilegio distinti (cliente, dipendente e amministratore).

Per maggiori informazioni si veda la sezione 2.3.

2.2 Assunzioni

2.2.1 Pacchetto

Il pacchetto è identificato da un periodo di validità, un nome, un tipo, un hotel e delle attività secondarie. I voli sono prestabiliti e a cadenza specifica per il pacchetto, e il check-in e il check-out dell'hotel coincidono con le date di arrivo e partenza del volo. Quindi un'eventuale prenotazione dovrà essere di una durata prestabilita (esempio: se la vacanza è tipo settimanale, da sabato al sabato seguente, non potrà essere effettuata una prenotazione con giorni di arrivo e partenza diversi da sabato).

2.2.2 Ricerca di un pacchetto

I risultati della ricerca del pacchetto vengono calcolati in base alla destinazione o al tipo, e alla data di partenza indicata dall'utente. In particolare vengono proposti tutti i pacchetti validi (tutte le componenti disponibili in quel periodo) che soddisfano il criterio di ricerca e la cui data di partenza è compresa tra quella indicata dall'utente e 7 giorni successivi.

2.2.3 Prezzo dei prodotti base

Esistono due tipi di prezzo: il prezzo del pacchetto e il prezzo del prodotto base.

Il primo si ha acquistando un pacchetto prestabilito; questo prezzo sarà inferiore alla somma dei prezzi dei prodotti base che compongono il pacchetto, grazie alle convenzioni stabilite con le compagnie alberghiere e aeree.

Il secondo, invece, si ha quando si decide di modificare o creare il pacchetto e sarà il prezzo intero non scontato.

2.2.4 Personalizzazione di un pacchetto

Se il cliente decide di modificare un prodotto base del pacchetto, verranno proposte tutte le alternative disponibili in quelle date e in quella località, indipendentemente dal tipo di pacchetto. Il prezzo verrà ricalcolato in base al costo dei prodotti base cambiati.

2.2.5 Cancellazione di un pacchetto

Quando un dipendente cancella un pacchetto questo non viene più mostrato quando si effettua una ricerca, ma le eventuali prenotazioni pendenti rimangono. Viene invece cancellato dalle liste desideri dei clienti che lo hanno aggiunto.

2.2.6 Modifica di un pacchetto

Quando un dipendente modifica un campo di un pacchetto, la modifica si ripercuote sulle future ricerche e sulle liste desideri dei clienti che lo avevano aggiunto. Queste modifiche invece non si ripercuotono sulle prenotazioni già effettuate.

2.2.7 Inserimento di un componente singolo

Il sistema non prevede l'inserimento di prodotti base se questi non fanno parte di uno specifico pacchetto.

2.3 Implementazioni future

Futuri aggiornamenti del software potranno prevedere:

- prenotazione di macchine a noleggio sul posto di villeggiatura,
- prenotazione di vacanze in crociera,
- trasporti diversi dall' aereo (treno, pullman),
- integrazione con i social network,
- servizi di contabilità automatica basati sulle fatture emesse,
- servizi di statistica basati sulle richieste al sito.

2.4 Descrizione funzionalità prodotto

2.4.1 Attori e funzionalità

Sono state individuate 3 diverse tipologie di attori che interagiscono con il sistema.

Le tipologie sono state specificate di seguito insieme ad una descrizione delle funzionalità che il sistema deve permettere

2.4.1.1 Cliente

Requisiti funzionali:

- [FR1] Registrazione
- [FR2] Modifica dati
- [FR3] Login
- [FR4] Logout
- [FR5] Visualizzazione catalogo pacchetti
- [FR6] Ricerca pacchetto
- [FR7] Personalizzazione pacchetto esistente
- [FR8] Creazione di un pacchetto
- [FR9] Acquisto prodotto base
- [FR10] Acquisto pacchetto
- [FR11] Visualizzazione delle descrizioni dettagliate di pacchetti e prodotti base
- [FR12] Aggiunta pacchetto a lista desideri
- [FR13] Invito amico a prendere visione del pacchetto scelto

Requisiti non funzionali:

- [NFR1] Le password degli utenti devono essere archiviate in modo sicuro
- [NFR2] Il sistema deve essere affidabile e disponibile anche con un alto traffico
- [NFR3] Il sistema deve essere disponibile 24/7
- [NFR4] Il tempo di risposta deve essere inferiore a 3 secondi
- [NFR5] Il pagamento deve avvenire in modo sicuro e crittografato

2.4.1.2 Dipendente

Requisiti funzionali:

- [FR1] Login
- [FR2] Logout
- [FR3] Aggiunta pacchetto
- [FR4] Modifica pacchetto

- [FR5] Cancellazione pacchetto
- [FR6] Ricerca pacchetto
- [FR7] Visualizzazione catalogo pacchetti
- [FR8] Visualizzazione storico ordini

Requisiti non funzionali:

- [NFR1] Le modifiche apportate al sistema devono essere permanenti
- [NFR2] Il sistema deve essere disponibile 24/7
- [NFR3] Il tempo di risposta deve essere inferiore a 5 secondi

2.4.1.3 Amministratore

Requisiti funzionali:

- [FR1] Login
- [FR2] Logout
- [FR3] Aggiunta dipendente
- [FR4] Rimozione dipendente

2.5 Vincoli

Il sistema deve rispettare i seguenti vincoli

- **Limiti hardware:**
 - Il sistema non ha alcuna limitazione hardware se non le minime richieste sulla memoria descritte sopra ed una capacità di calcolo del server in grado di gestire tutta l'utenza;
- **Operazioni in parallelo:**
 - Il sistema deve supportare operazioni contemporanee di più utenti ed essere in grado di gestirle in modo trasparente all'utente;
- **requisiti sul linguaggio di alto livello:**
 - Il sistema dovrà essere sviluppato in linguaggio Java sfruttando le tecnologie JavaEE;
- **Requisiti di affidabilità:**
 - Per garantire l'affidabilità del sistema è sufficiente che la memoria secondaria su cui sarà salvato il database sia replicata, in modo da impedire la perdita dei dati in caso di guasto;
- **Criticità dell'applicazione:**
 - Aspetto fondamentale è quello di tenere traccia delle informazioni che vengono salvate su database (credenziali utenti, informazioni pacchetti, prenotazioni, ecc..) poichè da questi dipende l'integrità del sistema e la sua fruibilità;

3 Requisiti specifici

3.1 Scenari

3.1.1 Registrazione al sito

Laura, studentessa universitaria, dopo un estenuante e impegnativo anno di studi, decide di organizzare le sue vacanze estive.

Laura non è soddisfatta dei pacchetti che le sono stati proposti nelle agenzie viaggio della sua città, così decide di cercare su internet.

Si collega a internet, apre il browser installato sul suo computer e inizia la ricerca, durante la quale trova il sito TravelDream. Laura decide di aprire il sito, e scopre di dover essere registrata per accedere al catalogo delle offerte.

Incuriosita e accattivata dalla pagina di benvenuto del sito che si promette di trovare la vacanza ideale ad ogni tipo di esigenza, decide di premere sull'opzione "registrati".

Il sistema le restituisce la form di registrazione, da riempire con le informazioni necessarie.

Laura inserisce nome, cognome, data di nascita, indirizzo, email e password, tutti campi obbligatori. Alla fine deve premere il pulsante "annulla" o il pulsante "salva".

Laura salva tutto, il sistema le invia un messaggio di avvenuta registrazione e la reindirizza all'homepage del sito.

3.1.2 Ricerca di un pacchetto

Laura, utente appena registratosi, può ora loggarsi e accedere alle funzionalità del sito.

Indirizzata sulla homepage di TravelDream, Laura immette email e password nella form di login.

Il sistema le restituisce l'homepage provvista di tutte le aree e funzioni riservate ai clienti registrati.

Laura può ora consultare le centinaia di pacchetti offerti dal sito, e decide di restringere la scelta filtrando quelli che effettivamente rispecchiano le sue idee con l'immediato motore di ricerca del sito.

Volendo andare in Svezia, Laura digita nella barra di ricerca per località "Stoccolma".

Il sistema restituisce una pagina con la lista di tutti i possibili pacchetti vacanza a Stoccolma.

Laura, essendo una ragazza particolarmente sportiva e dinamica, viene attirata dal pacchetto "SportiveStoccolma" e clicca sul pulsante "info" per saperne di più su tale proposta.

Il sistema restituisce la pagina con le informazioni dettagliate sulla vacanza: periodo, località, prezzo, hotel, volo, attività secondarie e descrizione del pacchetto.

3.1.3 Personalizzazione di un pacchetto

Laura analizzando il pacchetto "SportiveStoccolma" offertole dal sito rimane soddisfatta del volo, compagnia e costo, e delle attività secondarie organizzate, escursioni e arrampicate. Resta un po' dubbiosa invece sull'alloggio, un ostello spoglio e essenziale nella periferia di Stoccolma.

Laura decide quindi di premere sull'opzione "cambia hotel" presente all'interno della pagina del pacchetto, sperando di trovare qualcosa di più confortevole.

Il sistema restituisce la pagina con la lista degli hotel proposti in quella località. Laura sceglie l'hotel "Mercure", accogliente e in una posizione più centrale.

Il sistema restituisce la pagina del pacchetto aggiornata con l'hotel selezionato e il nuovo prezzo.

3.1.4 Invito a un amico

Laura, utente già registrato in cerca della sua adrenalinica vacanza estiva, è soddisfatta del pacchetto "SportiveSvezia" e delle attività che include, e, una volta modificato l'hotel è decisamente contenta del risultato ottenuto. Decide quindi di proporre questa vacanza alla sua amica e compagna di viaggio Maria. Laura preme sul pulsante "invita amico" e il sistema restituisce la pagina da riempire con l'email del destinatario.

Laura inserisce l'email dell'amica e preme su "invia", inoltrando così il codice identificativo e la scheda riepilogativa del viaggio.

Il sistema la informa dell'avvenuto invio. Prima di procedere con l'acquisto aspetta l'approvazione dell'amica.

3.1.5 Acquisto pacchetto

Laura, avuto parere positivo sulla vacanza dall'amica, decide di procedere con l'acquisto del pacchetto.

Dalla pagina del pacchetto scelto ("SportiveStoccolma") preme l'opzione "acquista".

Il sistema restituisce la pagina contenente le informazioni di pagamento,

Laura inserisce il suo numero di carta e prosegue con il pagamento.

Il sistema restituisce il messaggio di avvenuta prenotazione.

3.1.6 Aggiungi a lista desideri

Laura, navigando nel sito, scopre la possibilità di creare una lista desideri, che le permette di selezionare alcuni pacchetti e dare l'opportunità a amici o parenti di pagare e farle così un regalo sicuramente apprezzato.

Laura, volendosi far regalare una spassosa vacanza per le prossime vacanze invernali, una volta loggatasi, ricerca una vacanza di tipo "divertimento" dal motore di ricerca della homepage.

Il sistema restituisce una pagina con la lista dei pacchetti inerenti la ricerca, e Laura vuole saperne di più su "Capodanno a Cortina", per cui clicca sul pulsante "info" di tale proposta.

Il sistema restituisce la pagina con le informazioni dettagliate sulla vacanza: 5 giorni di soggiorno a cavallo dell'inizio dell'anno, hotel a due stelle, cenone dell'ultimo, skipass incluso.

Soddisfatta, Laura preme sul pulsante "aggiungi a lista desideri" e il sistema restituisce un messaggio di conferma.

Laura per assicurarsi che tale pacchetto sia stato effettivamente aggiunto alla sua lista desideri, clicca sull'opzione "vai alla lista desideri", fornita ad ogni utente registrato, e il sistema ritorna la pagina contenente l'elenco aggiornato con il pacchetto "Capodanno a Cortina".

3.1.7 Invia lista desideri

Avvicinandosi il periodo natalizio, Laura decide di inviare la sua lista desideri a sua sorella affinché possa regalarle la vacanza che aveva scelto.

Laura accede al sito, si logga e preme "vai alla lista desideri".

Il sistema le restituisce la pagina contenente i pacchetti che aveva aggiunto, solo "Capodanno a Cortina" nel suo caso.

Laura preme "invia lista desideri", il sistema le chiede di inserire la email della persona cui vuole mandarla, e lei inserisce la mail della sorella e preme "conferma".

Il sistema la informa con un messaggio del successo dell'operazione.

3.1.8 Crea pacchetto

Maria vuole organizzare, con un amico, un weekend particolare in Toscana per distrarsi e svagare dallo stress del lavoro. L'amica Laura le ha parlato bene di TravelDream, sito che offre l'opportunità di creare pacchetti vacanza personalizzati, costruendoli da zero, così Maria decide di provare questo servizio.

Si connette a internet e si registra al sito.

Una volta loggatasi e acceduta alla homepage del sito immediata è l'opzione "crea pacchetto", Maria ci clicca.

Il sistema le chiede per prima cosa di specificare la destinazione e il periodo.

Maria riempie i campi inserendo Livorno e scegliendo data di partenza e data di arrivo.

Il sistema le restituisce la pagina per la creazione del pacchetto, composta dalle tre opzioni scegli volo, scegli hotel, scegli attività.

Maria, bolognese, non necessita del trasporto aereo, pertanto traslascia la prima sezione. Preme invece il pulsante “scegli hotel”. Il sistema restituisce la pagina con l’elenco degli hotel in quella zona. Maria vuole avere più informazioni su “La tenuta”, quindi clicca sul tasto “info” accanto al nome dell’hotel. Il sistema restituisce la pagina con le informazioni dettagliate dell’hotel. Maria non ne rimane particolarmente entusiasta perché non offrono garage o parcheggio, per cui decide di cliccare su “torna alla lista” per guardarne altri. Il sistema restituisce la pagine originale. Maria decide di guardare le informazione su un altro hotel “Castello”, preme su info, legge i dettagli e, rimanendone soddisfatta, preme sul tasto “conferma” in fondo alla pagina. Il sistema le restituisce la pagina originale con l’ hotel “Castello” contrassegnato, e il prezzo del pacchetto settato a quello dell’ hotel. Infine cliccando su “scegli attività” le proposte sono “degustazione di vini” e “giro panoramico per le colline toscane” . Maria, appassionata di vini, decide di scegliere la prima opzione. Il sistema ritorna la pagina del pacchetto aumentando automaticamente il prezzo del pacchetto. Maria contenta del risultato decide di confermare e proseguire con la prenotazione, per cui preme “conferma”.

3.1.9 Aggiungi pacchetto

Matteo, impiegato dell’azienda, in seguito alla presa visione delle ultime statistiche di vendita e mercato, e constatate le scarse vendite di pacchetti costosi, decide di aggiungere un pacchetto vacanza economico in Italia. Matteo quindi effettua il login, e, immettendo le credenziali da dipendente, accede al sito con specifiche funzionalità riservate ai dipendenti. Matteo preme l’ opzione “modifica catalogo” e successivamente “aggiungi pacchetto” e il sistema restituisce la pagina con la form da compilare per la creazione di uno di questi. Matteo riempie i campi del pacchetto “SalentoRisparmiando”: tipo di vacanza, periodo, località, volo d’andata, volo di ritorno, hotel, attività secondarie e relativi prezzi, e descrizione del pacchetto, e preme il pulsante “conferma”. Il sistema restituisce la lista dei pacchetti aggiornata.

3.1.10 Modifica pacchetto

Matteo, in seguito alla chiamata dell’hotel termale “La Terrazza” che comunica di dover chiudere un paio di mesi per ristrutturazione, deve modificare il pacchetto viaggio “Rigenerarsi in un weekend” che prevede il soggiorno in quell’ hotel, sostituendolo con un’alternativa. Matteo quindi effettua il login in sito, inserendo le credenziali da dipendente, e preme il pulsante “modifica catalogo”. Il sistema ritorna la lista dei pacchetti. Non trovando immediatamente il pacchetto cercato tra i numerosi altri, Matteo decide di usare l’opzione di ricerca e digita quindi il nome del pacchetto nella barra di ricerca. Il sistema restituisce il pacchetto “Rigenerarsi in un weekend” e Matteo clicca sul pulsante “edit” accanto a questo. Il sistema restituisce la pagina contenente tutti i campi che compongono il pacchetto, riempiti con i dati attuali. Lorenzo cambia il nome, l’ indirizzo e il prezzo dell’ hotel, e preme “conferma”. Il sistema restituisce la lista dei pacchetti aggiornata.

3.1.11 Cancella pacchetto

Matteo controllando lo storico ordini, nota che il pacchetto “a spasso nella Val Nure” non è mai stato acquistato, e parlandone con un superiore, decide di eliminarlo. Matteo effettua quindi il login immettendo le credenziali da dipendente, e preme il pulsante “modifica catalogo”.

Il sistema ritorna la lista dei pacchetti e Matteo ricerca quello “a spasso nella Val Nure”.

Il sistema ritorna il pacchetto cercato e Matteo clicca sul pulsante “cancella” accanto a questo.

Una finestra di conferma chiede se si vuole proseguire con la cancellazione del pacchetto, Matteo preme “sì” e il sistema restituisce la lista dei pacchetti aggiornata.

3.1.12 Storico ordini

Matteo, incaricato di tenere controllate le prenotazioni e farne delle analisi, deve accedere alla funzionalità “storico ordini” offerta da TravelDream per i dipendenti.

Matteo effettua quindi il login immettendo le credenziali da dipendente, e preme il pulsante “storico ordini”.

Il sistema restituisce la pagina contenente la lista con tutte le prenotazioni e i relativi dettagli (data, cliente, prodotti prenotati, prezzo).

Matteo controlla le ultime fatture registrate e, terminata la sua analisi, preme “torna all’homepage”.

Il sistema restituisce la pagina di benvenuto per i dipendenti.

Non dovendo fare altre operazioni sul catalogo pacchetti, Matteo effettua il logout.

3.1.13 Elimina dipendente

Marco, amministratore del sistema, in seguito al licenziamento di Luca deve aggiornare la lista dei dipendenti, eliminando le sue credenziali.

Marco accede alla pagina di benvenuto del sito e effettua quindi il login immettendo le credenziali, uniche, da amministratore.

Il sistema gli ritorna la lista dei dipendenti con i loro dati e le loro credenziali.

Marco ricerca la riga contenente Luca e preme il pulsante “elimina” affianco a questo.

Una finestra di conferma chiede se si vuole proseguire con la cancellazione del dipendente, Marco preme “sì” e il sistema restituisce la lista dei dipendenti aggiornata.

3.1.14 Aggiungi dipendente

Marco, in seguito all’assunzione di Daniele deve aggiornare la lista dei dipendenti, aggiungendo le nuove credenziali di Daniele.

Marco accede alla pagina di benvenuto del sito e effettua quindi il login immettendo le credenziali, uniche, da amministratore.

Il sistema ritorna la lista dei dipendenti con i loro dati e le loro credenziali.

Marco seleziona immediatamente la sezione “aggiungi dipendente” sopra l’elenco e il sistema restituisce la pagina con la form da compilare per l’aggiunta di uno di questi.

Marco riempie i campi inserendo email e password valide, nome, cognome e indirizzo di Daniele, e preme il pulsante “conferma”.

Il sistema restituisce la lista dei dipendenti aggiornata

3.2 Casi d'uso

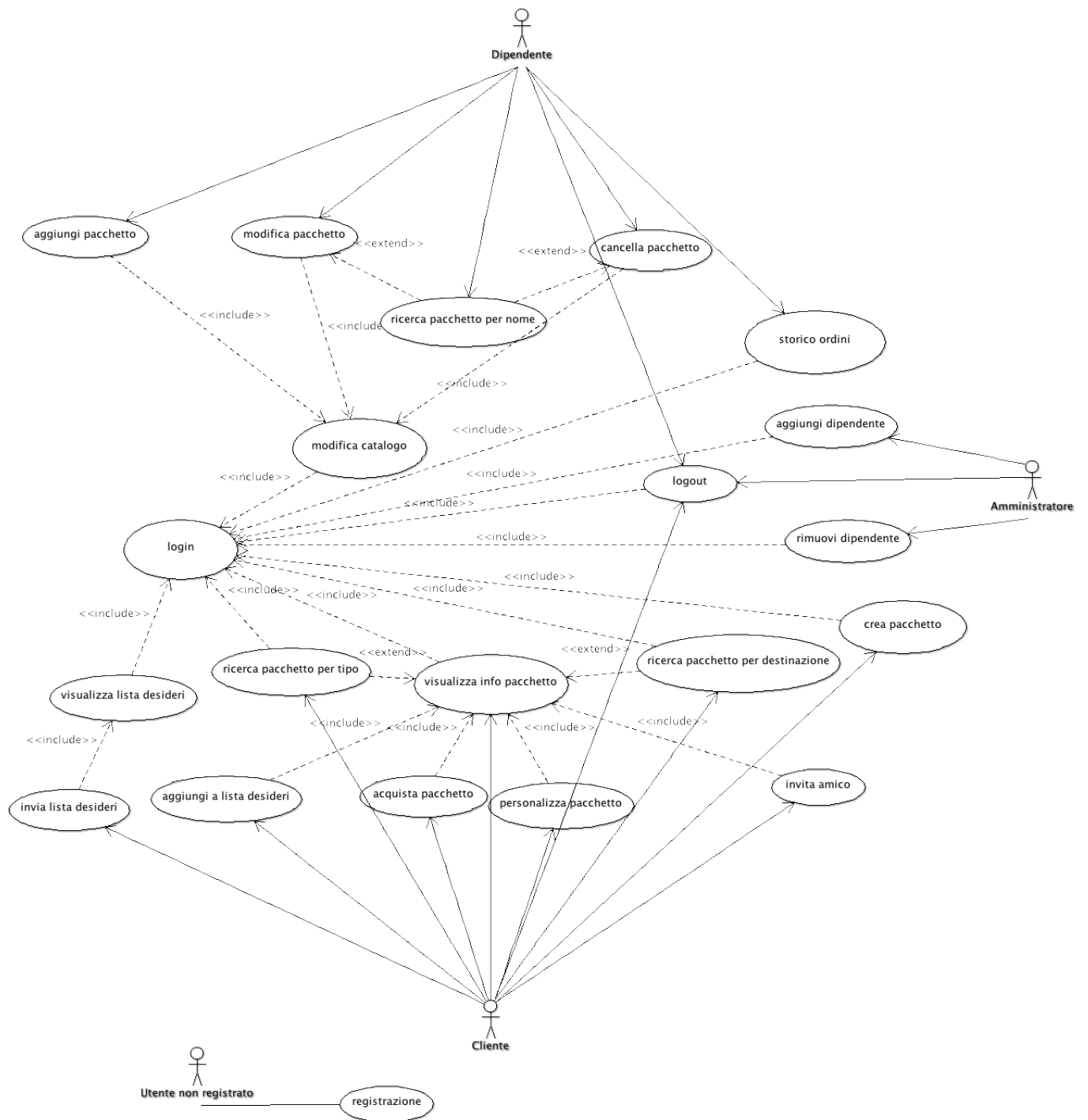


Figure 8: Modello dei casi d'uso

3.2.1 Registrazione

| | |
|-----------------------|---|
| Attori | Utente non registrato |
| Condizione d'ingresso | L'utente non ha mai effettuato la registrazione |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. L'utente non registrato accede alla pagina di benvenuto del sito 2. L'utente non registrato seleziona l'opzione "registrati" 3. Il sistema restituisce la form da compilare con i dati personali 4. L'utente non registrato completa i campi e preme il pulsante "salva" 5. Il sistema restituisce un messaggio di avvenuta registrazione e reindirizza l'utente alla homepage |
| Condizione di uscita | L'utente è registrato al sito e può effettuare il login |
| Eccezioni | L'email inserita non è univoca ; l'utente non completa la registrazione |

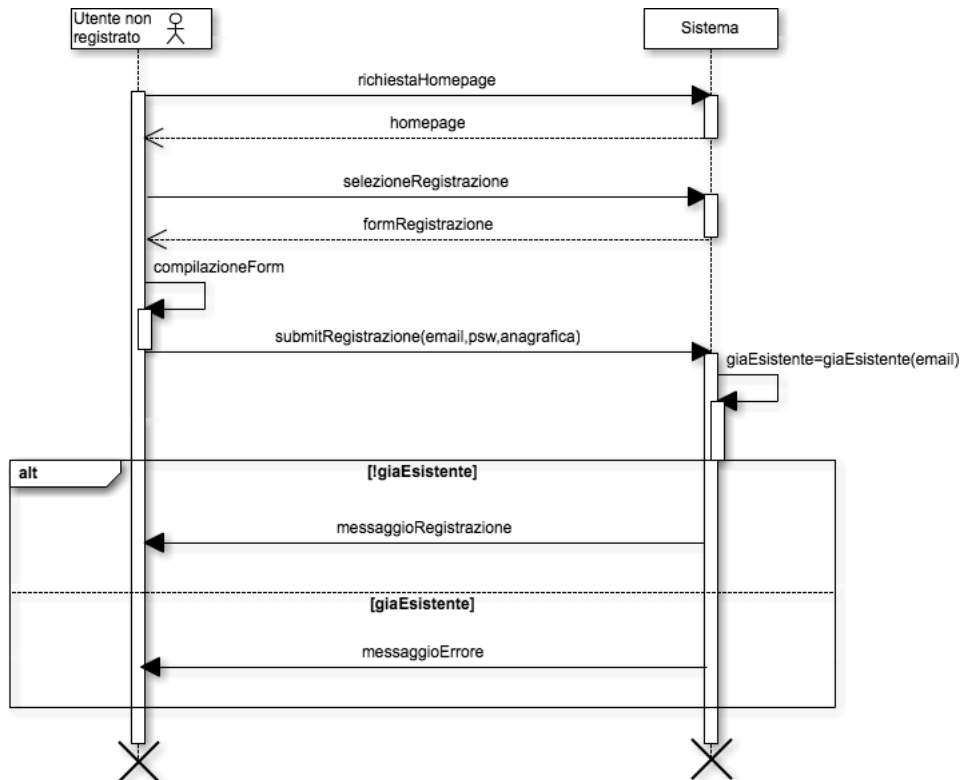


Figure 9: Registrazione: sequence diagram

3.2.2 Login

| | |
|-----------------------|---|
| Attori | Cliente/Dipendente/Amministratore |
| Condizione d'ingresso | Il cliente o il dipendente o l'amministratore è registrato ma non ha ancora effettuato il login |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. L'utente accede alla homepage del sito 2. L'utente inserisce indirizzo email e password 3. L'utente viene reindirizzato alla pagina principale |
| Condizione di uscita | L'utente è autenticato |
| Eccezioni | Il sistema non riconosce email e password; l'autenticazione fallisce |

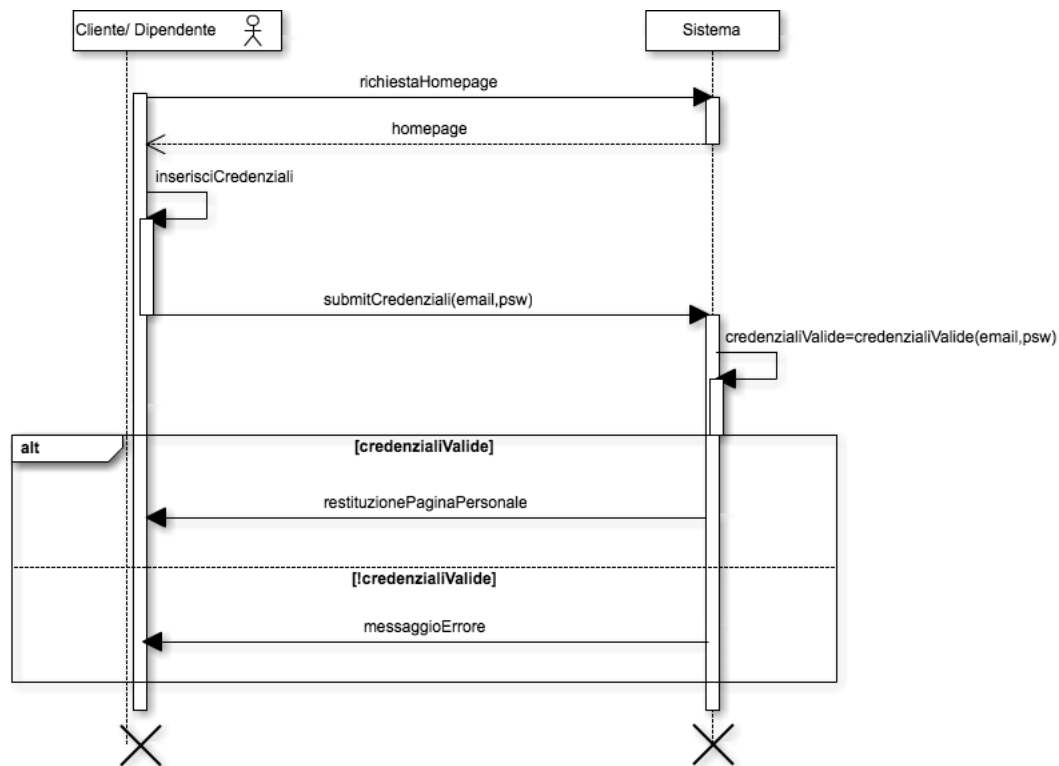


Figure 10: Login: sequence diagram

3.2.3 Logout

| | |
|-----------------------|---|
| Attori | Cliente/Dipendente/Amministratore |
| Condizione d'ingresso | Il cliente o il dipendente o l'amministratore ha effettuato il login e si trova in qualsiasi sezione del sito |
| Flusso d'eventi | <ol style="list-style-type: none">1. L'utente preme il tasto "logout" presente in ogni pagina del sito2. Il sistema lo riporta alla pagina iniziale di login e registrazione |
| Condizione di uscita | L'utente è disconnesso dal sito e non può più effettuare operazioni |
| Eccezioni | |

3.2.4 Ricerca di un pacchetto per destinazione

| | |
|-----------------------|--|
| Attori | Cliente |
| Goal | [G2] Agevolare i clienti nella consultazione del catalogo [G3] Aiutare i clienti nella ricerca e creazione della vacanza adatta a loro |
| Condizione d'ingresso | Il cliente si è loggato con successo |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il cliente digita nella barra di ricerca per destinazione la località che desidera, e seleziona le data di partenza2. Il sistema restituisce la lista di pacchetti coerenti con la ricerca |
| Condizione di uscita | Al cliente è fornita una lista di pacchetti che soddisfano la sua ricerca |
| Eccezioni | Non previsto alcun pacchetto vacanza nella destinazione digitata; il sistema notifica l'assenza di risultati |

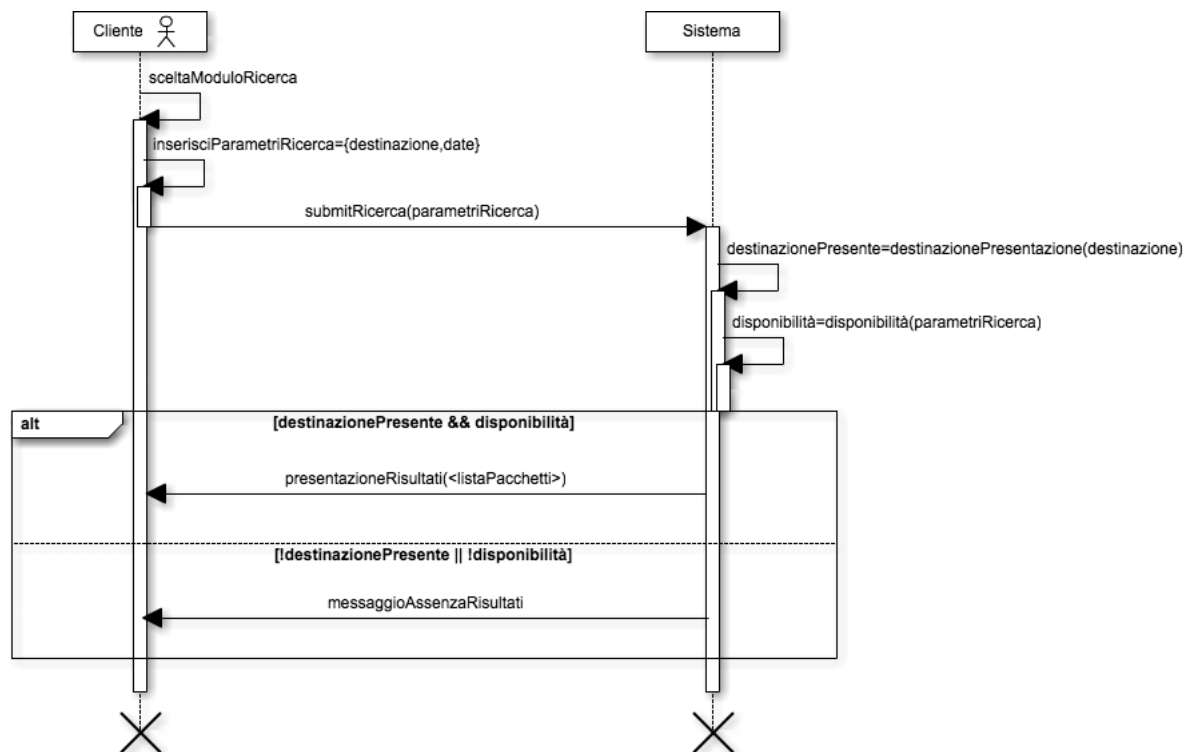


Figure 11: Ricerca per destinazione: sequence diagram

3.2.5 Ricerca di un pacchetto per tipo

| | |
|-----------------------|--|
| Attori | Cliente |
| Goal | [G2] Agevolare i clienti nella consultazione del catalogo [G3] Aiutare i clienti nella ricerca e creazione della vacanza adatta a loro |
| Condizione d'ingresso | Il cliente si   loggato con successo |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il cliente sceglie dal men  a tendina il tipo di vacanza che sta cercando, e seleziona le date 2. Il sistema restituisce la lista di pacchetti coerenti con la ricerca |
| Condizione di uscita | Al cliente   fornita una lista di pacchetti che soddisfano la sua ricerca |
| Eccezioni | Esaurita la disponibilit  di pacchetti vacanza di un determinato tipo in determinate date; il sistema notifica l'assenza di risultati |

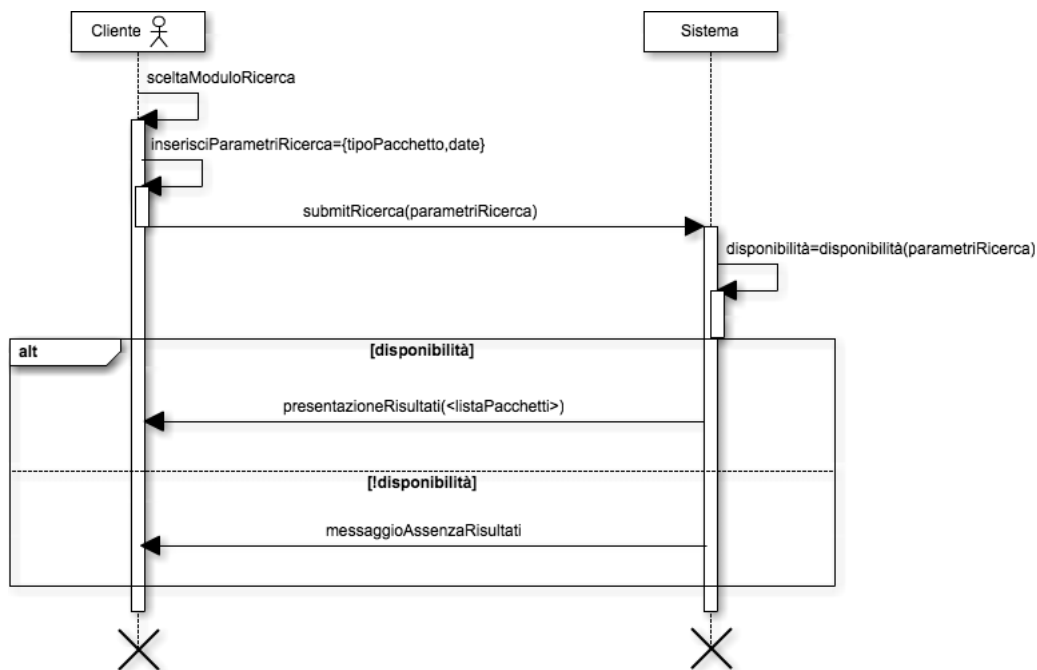


Figure 12: Ricerca per tipo: sequence diagram

3.2.6 Visualizzazione scheda di un pacchetto

| | |
|-----------------------|--|
| Attori | Cliente |
| Goal | [G2] Agevolare i clienti nella consultazione del catalogo [G3] Aiutare i clienti nella ricerca e creazione della vacanza adatta a loro |
| Condizione d'ingresso | Il cliente si è loggato con successo e ha effettuato una ricerca |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il cliente clicca sul pulsante "info" accanto al nome di un pacchetto per avere maggiori dettagli 2. Il sistema ritorna la pagina con le informazioni dettagliate del pacchetto 3. Il cliente preme "torna alla lista" 4. Il sistema ritorna la pagina con la lista 5. Il cliente può consultare e visualizzare gli altri pacchetti |
| Condizione di uscita | Al cliente è fornita la pagina di descrizione di un pacchetto |
| Eccezioni | |

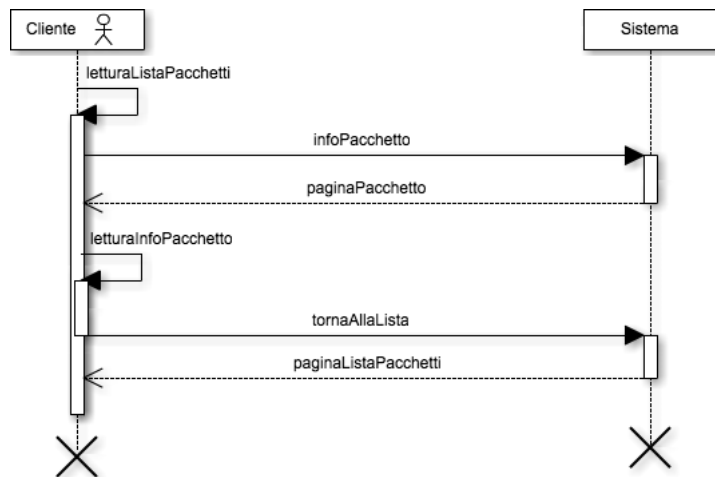


Figure 13: Visualizzazione scheda pacchetto: sequence diagram

3.2.7 Personalizzazione di un pacchetto

| | |
|-----------------------|--|
| Attori | Cliente |
| | [G3] Aiutare i clienti nella ricerca e creazione della vacanza adatta a loro |
| Condizione d'ingresso | Il cliente si è loggato con successo, ha effettuato una ricerca e accede alla pagina info di un pacchetto |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il cliente preme “cambia hotel”2. il sistema restituisce la pagina con la lista degli hotel proposti in quella località3. Il cliente sceglie un hotel e preme “conferma”4. Il sistema restituisce la pagina delle informazioni aggiornate del pacchetto con l'hotel selezionato e il nuovo prezzo5. Il cliente preme “cambia volo”6. Il sistema restituisce la pagina con la lista dei voli disponibili per quella destinazione7. Il cliente sceglie un volo e preme “conferma”8. Il sistema restituisce la pagina delle informazioni aggiornate del pacchetto con il volo selezionato e il nuovo prezzo9. Il cliente preme “cambia attività”10. Il sistema restituisce la pagina con la lista delle attività proposte e organizzate in quella località11. Il cliente sceglie un' attività e preme “conferma”12. Il sistema restituisce la pagina delle informazioni aggiornate del pacchetto con l'attività selezionata e il nuovo prezzo |
| Condizione di uscita | Al cliente sono fornite le informazioni del pacchetto modificato |
| Eccezioni | Il sistema non restituisce alcuna alternativa perchè terminate le disponibilità |

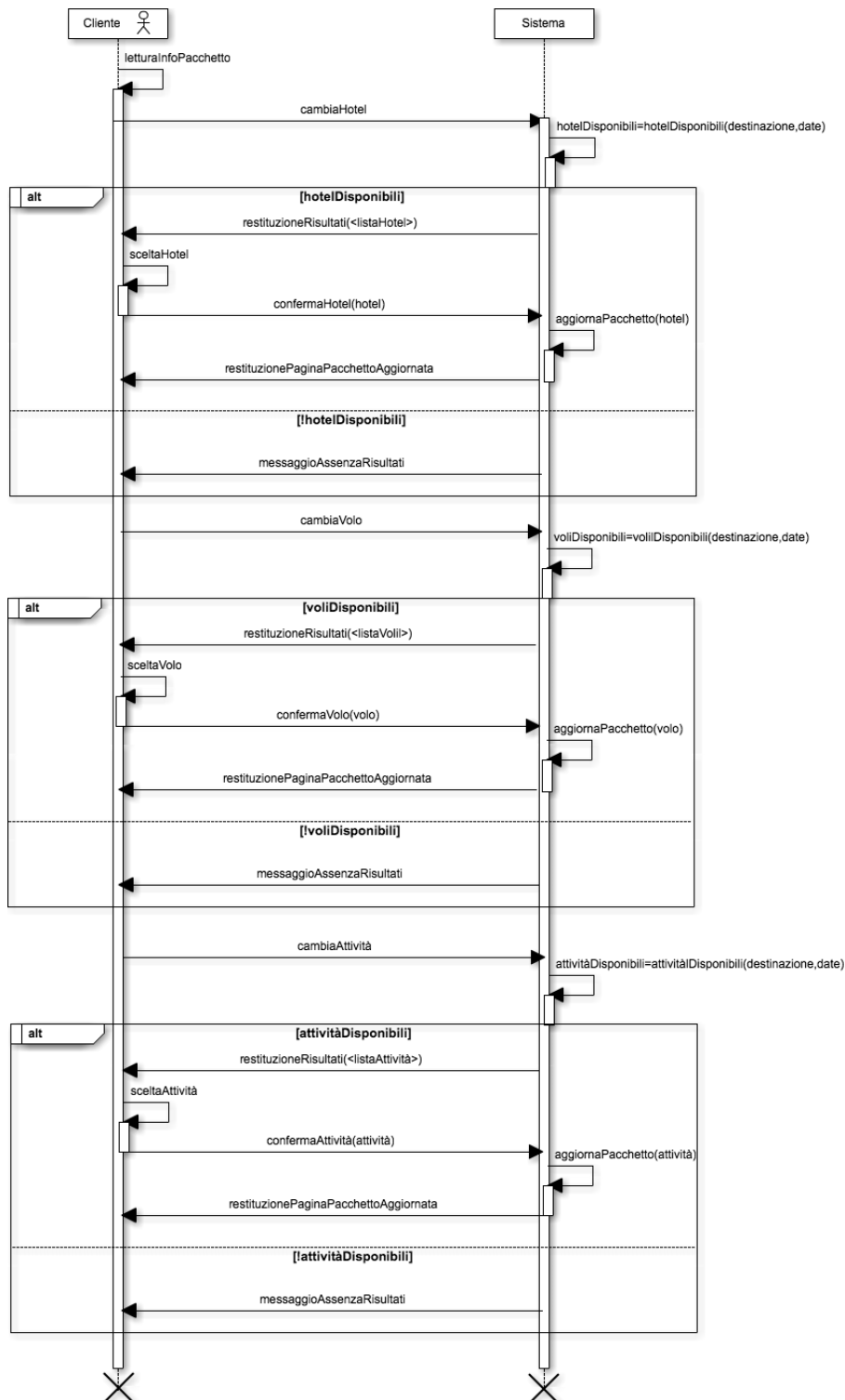


Figure 14: Personalizza pacchetto: sequence diagram

3.2.8 Invita amico

| | |
|-----------------------|--|
| Attori | Cliente |
| Goal | [G5] Agevolare i clienti nella condivisione della vacanza con altre persone |
| Condizione d'ingresso | Il cliente si è loggato con successo, ha effettuato una ricerca e accede alla pagina info di un pacchetto |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il cliente clicca sul pulsante “invita amico” 2. Il sistema restituisce la pagina contenente il campo email del destinatario 3. Il cliente inserisce l’email della persona che vuole coinvolgere, e preme “invia” 4. Il sistema restituisce un messaggio di avvenuto invio |
| Condizione di uscita | Il sistema notifica il successo dell’operazione |
| Eccezioni | Email inserita non valida; il sistema segnala che il campo non è stato riempito correttamente |

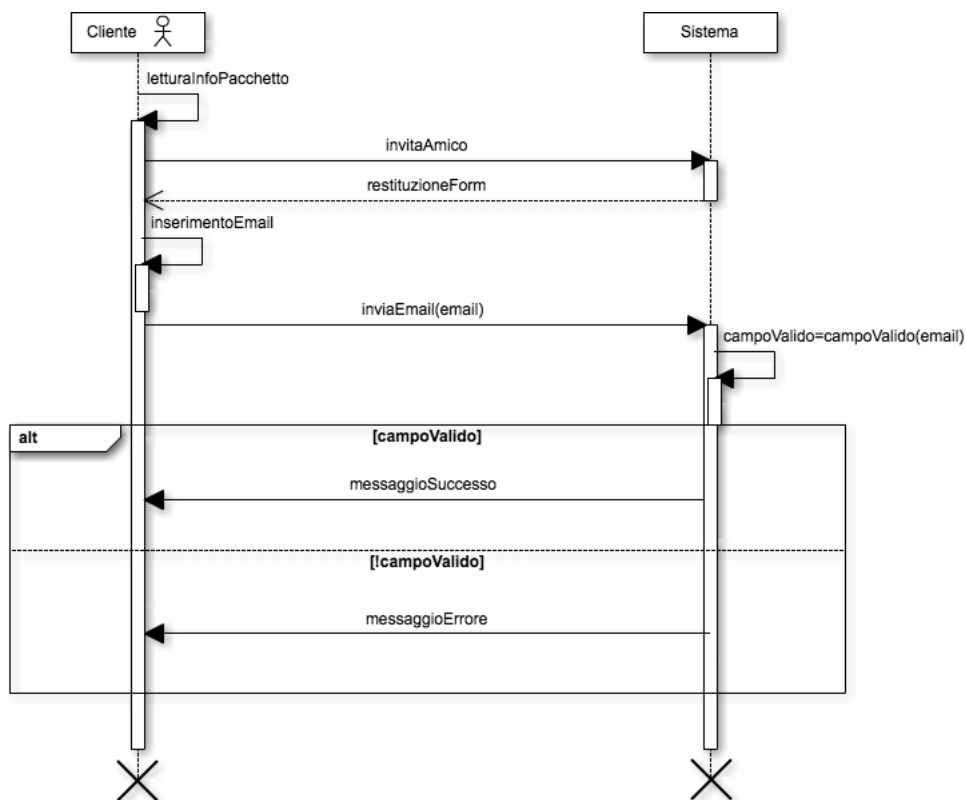


Figure 15: Invita amico: sequence diagram

3.2.9 Acquista pacchetto

| | |
|-----------------------|--|
| Attori | Cliente |
| Goal | [G4] Facilitare i clienti nella prenotazione della vacanza |
| Condizione d'ingresso | Il cliente si è loggato con successo, ha selezionato e visualizzato le info di un pacchetto (eventualmente dopo averlo personalizzato) |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il cliente clicca su “acquista” 2. Il sistema restituisce la pagina contenente le informazioni di pagamento 3. Il cliente inserisce i dati necessari al pagamento e preme “conferma” 4. Il sistema restituisce il messaggio di avvenuta prenotazione |
| Condizione di uscita | Il sistema notifica il successo della prenotazione |
| Eccezioni | Dati inseriti per il pagamento non validi; il sistema segnala che i campi non sono stati riempiti correttamente |

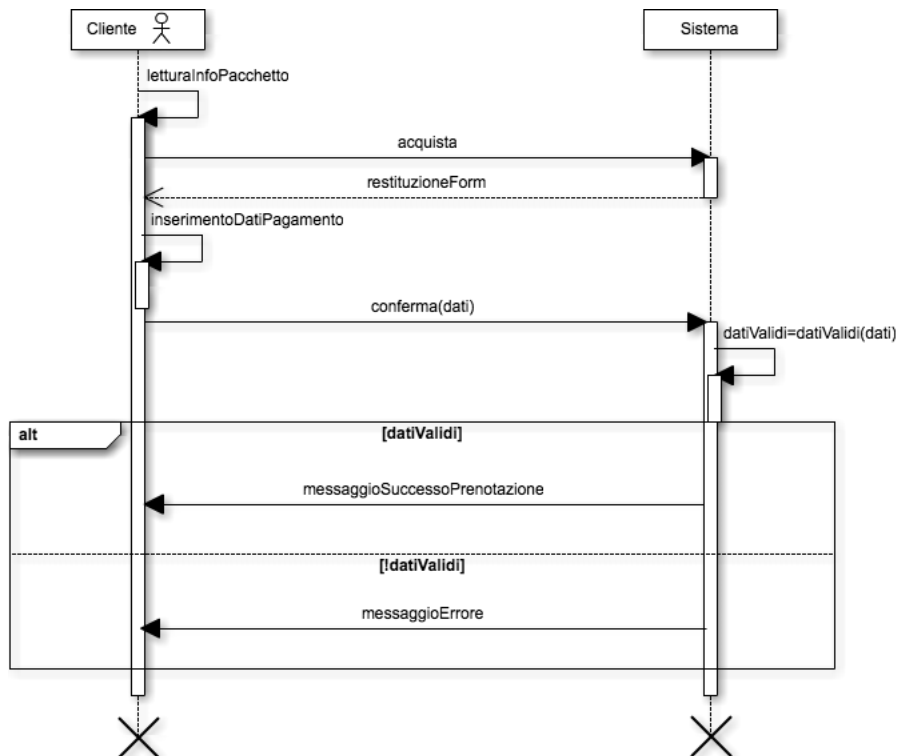


Figure 16: Acquista pacchetto: sequence diagram

3.2.10 Aggiungi a lista desideri

| | |
|-----------------------|---|
| Attori | Cliente |
| Goal | [G6] Offrire servizi ad hoc ai clienti, quali la lista desideri |
| Condizione d'ingresso | Il cliente si è loggato con successo, ha selezionato e visualizzato le info di un pacchetto |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il cliente clicca su “aggiungi a lista desideri” 2. Il sistema mostra una finestra di conferma 3. Il cliente preme “sì” 4. Il sistema restituisce un messaggio di avvenuta aggiunta |
| Condizione di uscita | Il sistema notifica il successo dell'operazione |
| Eccezioni | |

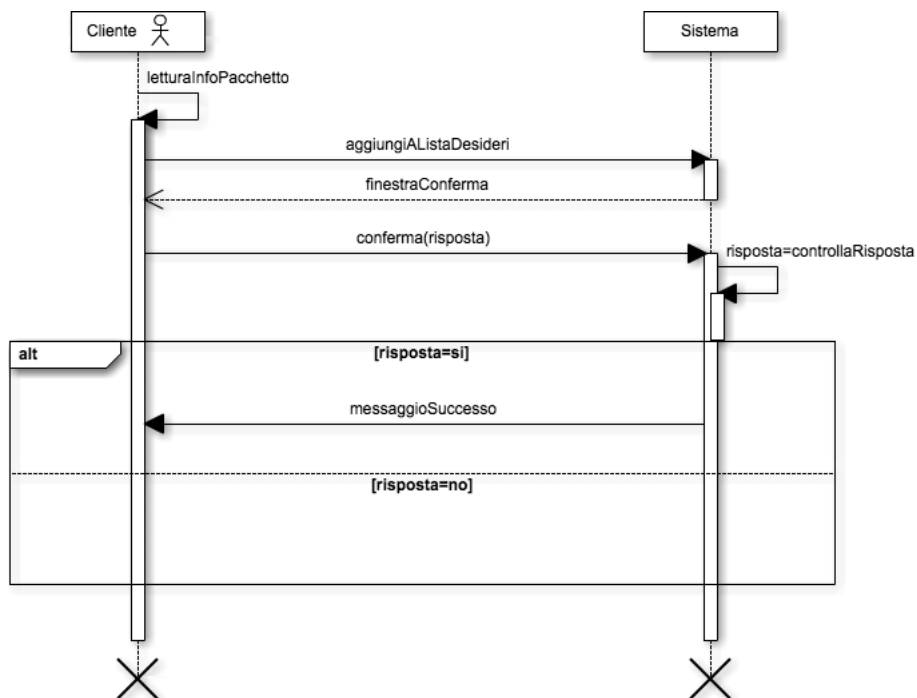


Figure 17: Aggiungi a lista desideri: sequence diagram

3.2.11 Visualizza lista desideri

| | |
|-----------------------|---|
| Attori | Cliente |
| Goal | [G6] Offrire servizi ad hoc ai clienti, quali la lista desideri |
| Condizione d'ingresso | Il cliente si è loggato con successo |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il cliente clicca su “vai alla lista desideri”2. Il sistema restituisce l'elenco dei pacchetti aggiunti dal cliente |
| Condizione di uscita | Al cliente è fornita la sua lista desideri |
| Eccezioni | |

3.2.12 Invia lista desideri

| | |
|-----------------------|---|
| Attori | Cliente |
| Goal | [G6] Offrire servizi ad hoc ai clienti, quali la lista desideri |
| Condizione d'ingresso | Il cliente si è loggato con successo e ha visualizzato la sua lista desideri |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il cliente clicca su “invia lista desideri”2. Il sistema restituisce la pagina contenente il campo email del destinatario3. Il cliente inserisce l'email della persona che vuole coinvolgere, e preme “invia”4. Il sistema restituisce un messaggio di avvenuto invio |
| Condizione di uscita | Il sistema notifica il successo dell'operazione |
| Eccezioni | Email inserita non valida; il sistema segnala che il campo non è stato riempito correttamente |

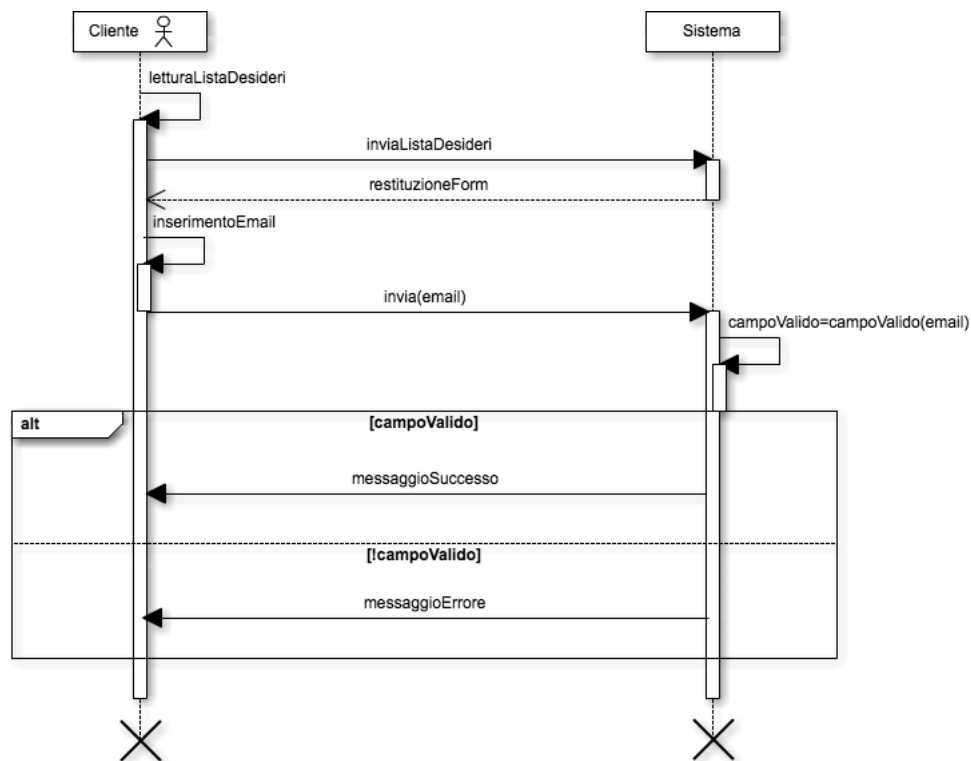


Figure 18: invia lista desideri: sequence diagram

3.2.13 Cancella dalla lista desideri

| | |
|-----------------------|---|
| Attori | Cliente |
| Goal | [G6] Offrire servizi ad hoc ai clienti, quali la lista desideri |
| Condizione d'ingresso | Il cliente si è loggato con credenziali valide, è acceduto alla sezione lista desideri |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il cliente clicca sul pulsante “cancella” accanto al pacchetto 2. Il sistema restituisce la lista desideri aggiornata |
| Condizione di uscita | Il pacchetto è ora rimosso dalla lista desideri |
| Eccezioni | |

3.2.14 Crea pacchetto

| | |
|-----------------------|---|
| Attori | Cliente |
| Goal | [G3] Aiutare i clienti nella ricerca e creazione della vacanza adatta a loro |
| Condizione d'ingresso | Il cliente si è loggato con successo |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il cliente preme “crea pacchetto”2. Il sistema chiede di specificare periodo e destinazione3. Il cliente inserisce le date e la località e preme “conferma”4. Il sistema restituisce la pagina per la creazione del pacchetto5. Il cliente preme il pulsante “scegli volo”6. Il sistema restituisce la pagina con l’elenco dei voli disponibili per quella destinazione7. Il cliente ne seleziona uno e preme “conferma”8. Il sistema ritorna la pagina del pacchetto aggiornata9. Il cliente preme il pulsante “scegli hotel”10. Il sistema restituisce la pagina con l’elenco degli hotel disponibili per quella destinazione11. Il cliente ne seleziona uno e preme “conferma”12. Il sistema ritorna la pagina del pacchetto aggiornata13. Il cliente preme il pulsante “scegli attività”14. Il sistema restituisce la pagina con l’elenco delle attività disponibili per quella destinazione15. Il cliente ne seleziona una e preme “conferma”16. Il sistema ritorna la pagina del pacchetto aggiornata |
| Condizione di uscita | Al cliente è fornita la pagina del pacchetto aggiornata con le opzioni scelte e il prezzo aumentato. |
| Eccezioni | |

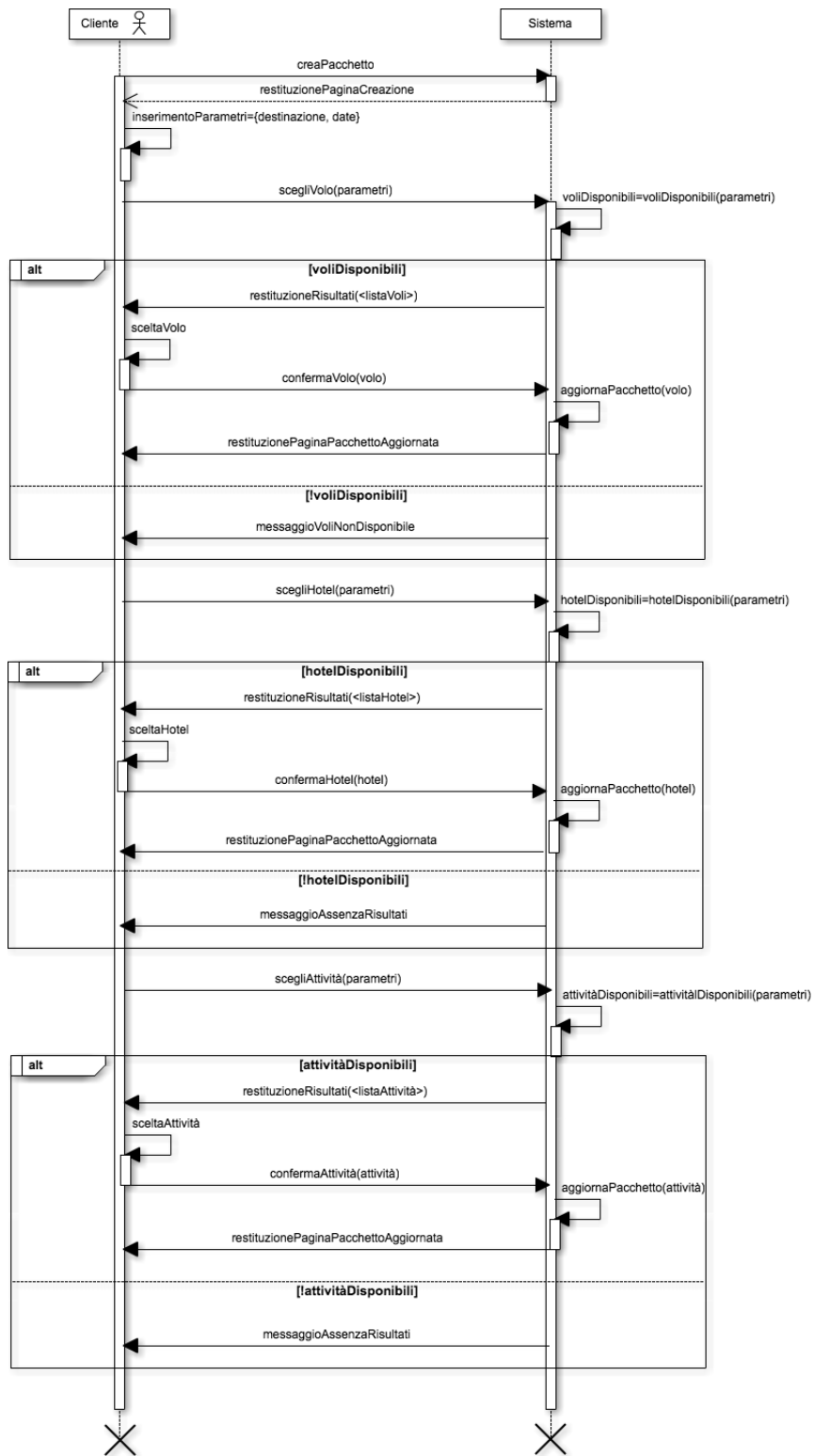


Figure 19: Crea pacchetto: sequence diagram

3.2.15 Modifica catalogo

| | |
|-----------------------|---|
| Attori | Dipendente |
| Goal | [G7] Semplificare l'aggiornamento delle offerte dell'agenzia |
| Condizione d'ingresso | Il dipendente si è loggato con credenziali valide |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il dipendente clicca su “modifica catalogo”2. Il sistema ritorna la pagina contenente tutti i pacchetti |
| Condizione di uscita | Al dipendente è fornita la lista di di tutti i pacchetti e può ora eseguire le operazioni di modifica |
| Eccezioni | |

3.2.16 Aggiungi pacchetto

| | |
|-----------------------|---|
| Attori | Dipendente |
| Goal | [G7] Semplificare l'aggiornamento delle offerte dell'agenzia |
| Condizione d'ingresso | Il dipendente si è loggato con credenziali valide ed è acceduto alla sezione modifica catalogo |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il dipendente preme il pulsante “aggiungi pacchetto”2. Il sistema restituisce la pagina con tutti i campi che compongono il pacchetto da completare3. Il dipendente riempie i campi tipo, periodo, località, hotel e prezzo, volo e prezzo, attività secondarie e prezzo e descrizione del pacchetto, e preme il pulsante “conferma” -4. Il sistema restituisce la lista dei pacchetti aggiornata |
| Condizione di uscita | Il pacchetto è ora incluso nel sistema |
| Eccezioni | Dati mancanti o dati inseriti non validi nella form di aggiunta; il sistema segnala i campi riempiti scorrettamente |

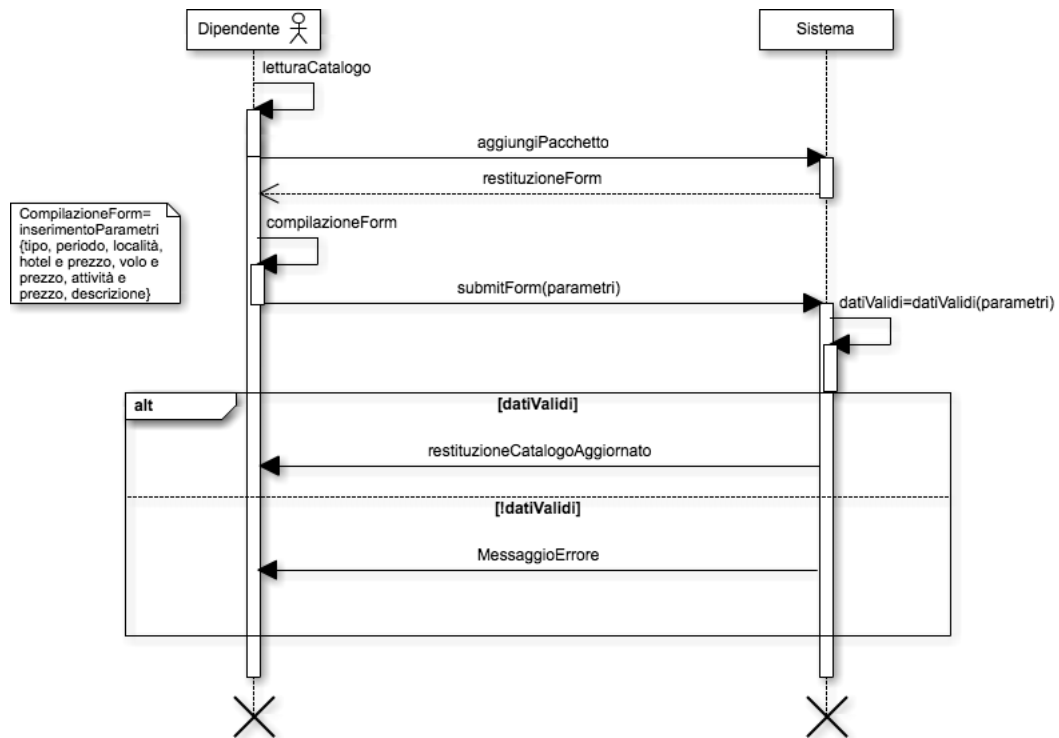


Figure 20: Aggiungi pacchetto: sequence diagram

3.2.17 Ricerca di un pacchetto per nome

| | |
|-----------------------|--|
| Attori | Dipendente |
| Goal | [G7] Semplificare l'aggiornamento delle offerte dell'agenzia |
| Condizione d'ingresso | Il dipendente si è loggato con credenziali valide ed è acceduto alla sezione modifica catalogo |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il dipendente cerca il pacchetto che deve modificare digitandone il nome nella barra di ricerca 2. Il sistema restituisce il pacchetto cercato |
| Condizione di uscita | Al dipendente è fornito esattamente il pacchetto cercato |
| Eccezioni | Nome del pacchetto inesistente; il sistema notifica l'assenza di risultati |

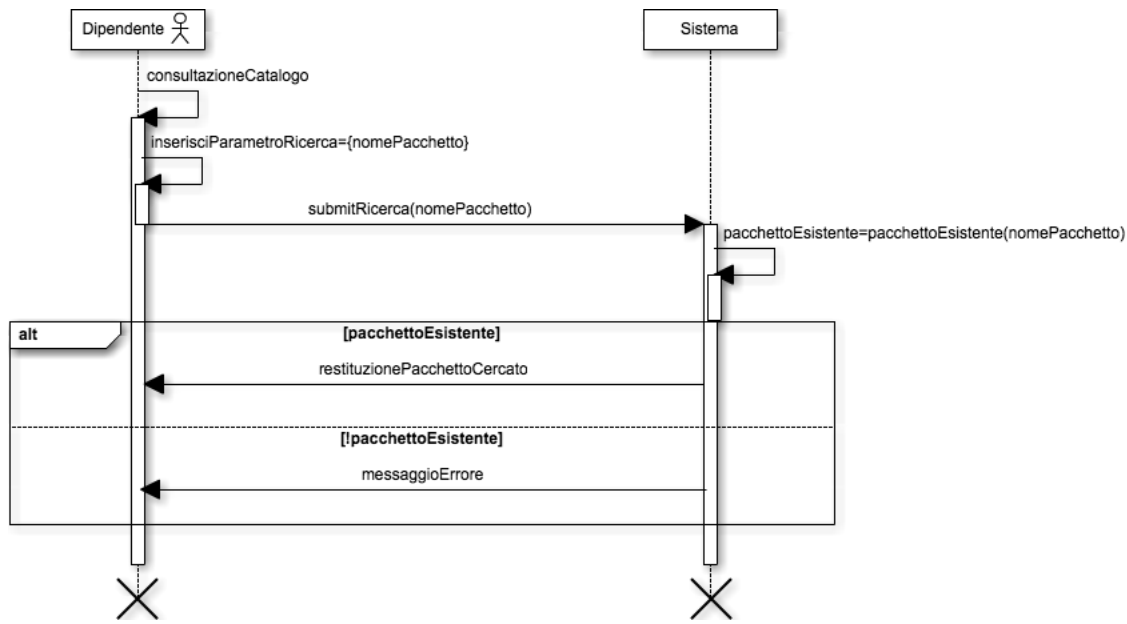


Figure 21: Ricerca per nome: sequence diagram

3.2.18 Edita pacchetto

| | |
|-----------------------|--|
| Attori | Dipendente |
| Goal | [G7] Semplificare l'aggiornamento delle offerte dell'agenzia |
| Condizione d'ingresso | Il dipendente si è loggato con credenziali valide, è acceduto alla sezione modifica catalogo e ha selezione il pacchetto da modificare (eventualmente cercandolo) |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il dipendente clicca sul pulsante "edit" accanto al pacchetto 2. Il sistema restituisce la pagina con tutti i campi che compongono il pacchetto, riempiti con i dati attuali 3. Il dipendente cambia quelli che deve, e clicca su "conferma" 4. Il sistema restituisce la lista dei pacchetti aggiornata |
| Condizione di uscita | Il pacchetto è ora modificato |
| Eccezioni | Dati mancanti o dati inseriti non validi nella form di modifica; il sistema segnala i campi riempiti scorrettamente |

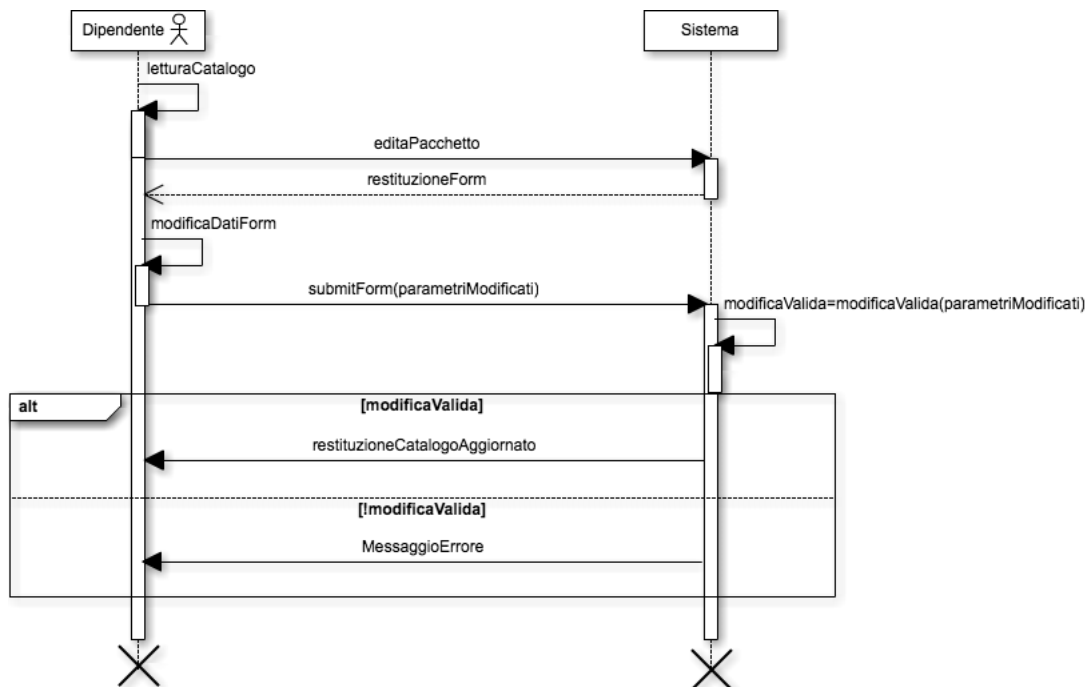


Figure 22: Edita pacchetto: sequence diagram

3.2.19 Cancella pacchetto

| | |
|-----------------------|--|
| Attori | Dipendente |
| Goal | [G7] Semplificare l'aggiornamento delle offerte dell'agenzia |
| Condizione d'ingresso | Il dipendente si è loggato con credenziali valide, è acceduto alla sezione modifica catalogo e ha selezione il pacchetto da modificare (eventualmente cercandolo) |
| Flusso d'eventi | <ol style="list-style-type: none"> 1. Il dipendente clicca sul pulsante “cancella” accanto al pacchetto 2. Il sistema restituisce un popup di conferma 3. Il dipendente clicca sul pulsante “sì” 4. Il sistema restituisce la lista dei pacchetti aggiornata |
| Condizione di uscita | Il pacchetto è eliminato dal sistema |
| Eccezioni | |

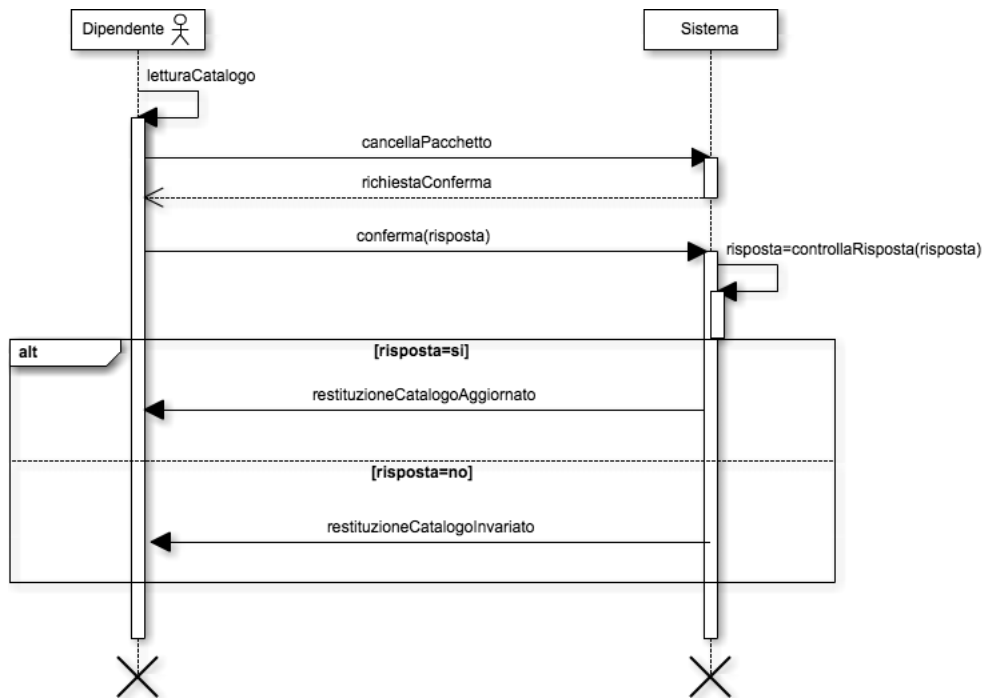


Figure 23: Cancella pacchetto: sequence diagram

3.2.20 Storico ordini

| | |
|-----------------------|--|
| Attori | Dipendente |
| Goal | [G8] Semplificare la gestione dell'intero sistema |
| Condizione d'ingresso | Il dipendente si è loggato con credenziali valide |
| Flusso d'eventi | <ol style="list-style-type: none">1. Il dipendente clicca sulla sezione "storico ordini"2. Il sistema restituisce la pagina elencante tutte le prenotazioni con relative informazioni |
| Condizione di uscita | Al dipendente è fornita la lista delle prenotazioni |
| Eccezioni | |

3.2.21 Aggiungi dipendente

| | |
|-----------------------|--|
| Attori | Amministratore |
| Goal | [G8] Semplificare la gestione dell'intero sistema |
| Condizione d'ingresso | L' amministratore si è loggato con credenziali uniche e valide |
| Flusso d'eventi | <ol style="list-style-type: none">1. L' amministratore preme il pulsante "aggiungi dipendente"2. Il sistema restituisce la pagina con tutti i campi che identificano il dipendente da completare3. L'amministratore riempie i campi email e password, nome, cognome e indirizzo, e preme il pulsante "conferma"4. Il sistema restituisce la lista dei dipendenti aggiornata |
| Condizione di uscita | Il dipendente e le sue credenziali valide sono ora inserite nel sistema |
| Eccezioni | Credenziali inserite già esistenti; il sistema notifica questa irregolarità |

3.2.22 Cancella dipendente

| | |
|-----------------------|---|
| Attori | Amministratore |
| Goal | [G8] Semplificare la gestione dell'intero sistema |
| Condizione d'ingresso | L' amministratore si è loggato con credenziali uniche e valide, e ha cercato il dipendente da eliminare |
| Flusso d'eventi | <ol style="list-style-type: none">1. L'amministratore clicca sul pulsante "cancella" accanto al dipendente2. Il sistema restituisce un popup di conferma3. L'amministratore clicca sul pulsante "sì"4. Il sistema restituisce la lista dei dipendenti aggiornata |
| Condizione di uscita | Il dipendente è eliminato dal sistema e le sue credenziali non sono più valide |
| Eccezioni | |

3.3 Diagramma delle classi

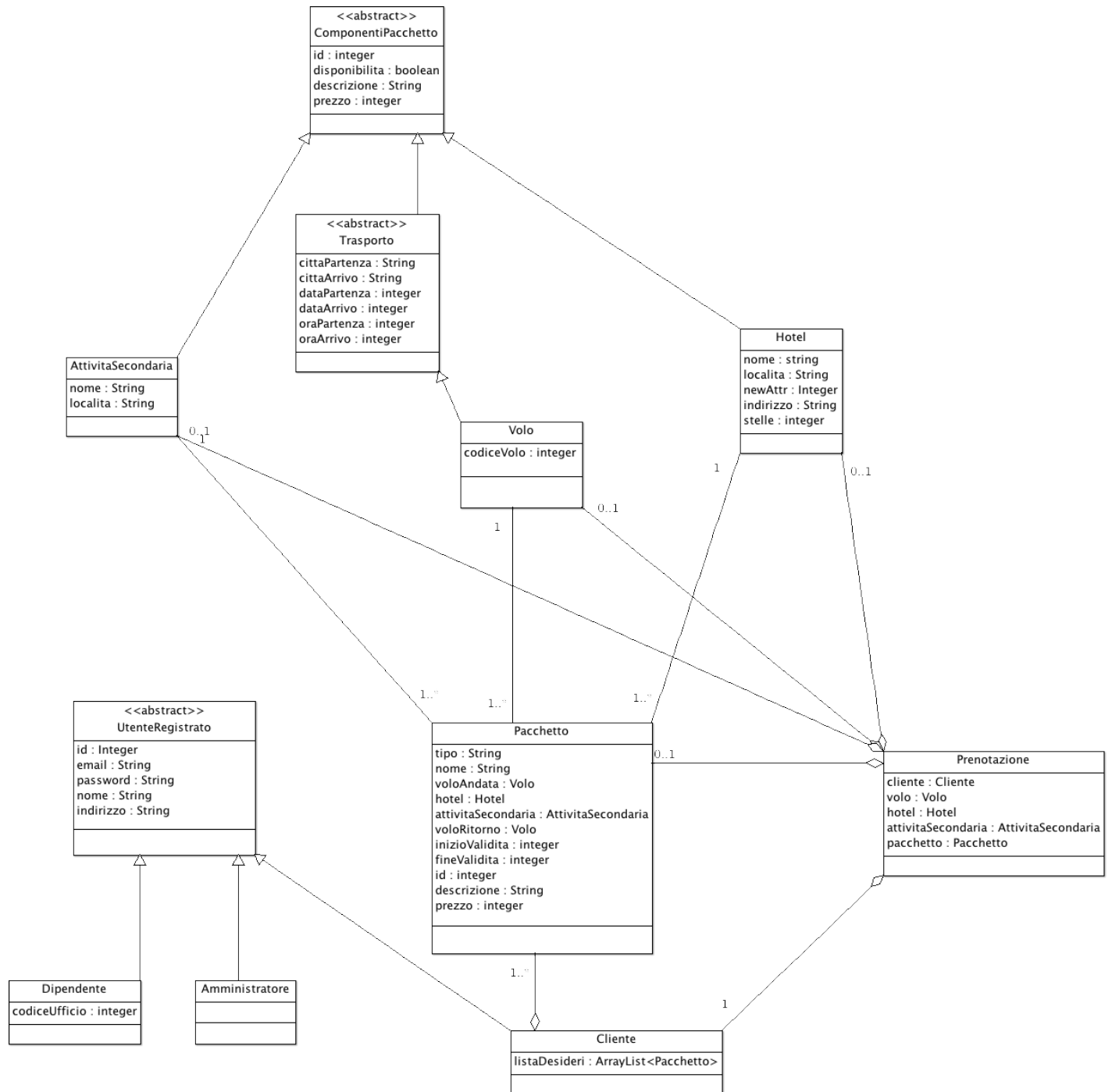


Figure 24: Diagramma delle classi

4 Appendice

4.1 Alloy

Di seguito verranno descritte la modellizzazione e le assunzioni fatte utilizzando Alloy Analyzer

4.1.1 Assunzioni

Per semplicità verranno trattate le date come interi positivi, in cui date successive avranno interi successivi, e la disponibilità come parametro booleano.

Nell'implementazione finale la data dovrà essere trattata come data del calendario, e la disponibilità delle varie componenti dovrà essere ricavata ripettando i vari vincoli come il numero di posti disponibili, la data scelta ecc.

Verrà riportato solo il codice più significativo; per la visione dell'intero si veda il file allegato.

4.1.2 Analisi formale

L'analisi di seguito proposta deriva dallo studio del diagramma delle classi e dal rispetto dei vincoli descritti finora.

```

/----- TIPI DI DATO -----
sig Testo{}
/----- ENTITA' RELATIVE AL PACCHETTO -----
//attributi comuni a tutti gli elementi del pacchetto
abstract sig ComponentePacchetto{
    disponibilita: one Bool
}

//attributi comuni a tutte le classi di trasporto (utile per le future implementazioni)
abstract sig Trasporto extends ComponentePacchetto {
    cittaPartenza: one Testo,
    cittaArrivo: one Testo,
    dataPartenza: one Int,
    dataArrivo: one Int,
    oraPartenza: one Int,
    oraArrivo: one Int
}

sig Pacchetto{
    nome: one Testo,
    tipo: one Testo,
    voloAndata: one Trasporto,
    voloRitorno: one Trasporto,
    hotel: one Hotel,
    attivitaSecondaria: one AttivitaSecondaria,
    inizioValidita: one Int,
    fineValidita: one Int
}

sig Volo extends Trasporto{
    codiceVolo: one Testo
}

sig Hotel extends ComponentePacchetto{
    localita: one Testo,
    nome: one Testo,
    indirizzo: one Testo
}

sig AttivitaSecondaria extends ComponentePacchetto{
    nome: one Testo,
    localitaValida: one Testo,
}

sig Prenotazione{
    cliente: one Cliente,
    volo: one Volo,
    hotel: one Hotel,
    attivitaSecondaria: one AttivitaSecondaria,
    pacchetto: one Pacchetto,
}
{ (#pacchetto > 0) implies (#volo=0 and #hotel=0 and #attivitaSecondaria = 0)) and
  ((#volo>0 or #hotel>0 or #attivitaSecondaria > 0) implies (#pacchetto =0))}

//il catalogo deve essere unico
one sig Catalogo{
    insiemePacchetti: set Pacchetto
}

//lo storico ordini deve essere unico
one sig StoricoOrdini{
    insiemePrenotazioni: set Prenotazione,
}

/----- ENTITA' RELATIVE AGLI UTENTI REGISTRATI -----

//attributi comuni a clienti e dipendenti
abstract sig UtenteRegistrato{
}

//l' amministratore è unico
one sig Amministratore extends UtenteRegistrato{

}

sig Cliente extends UtenteRegistrato{
    listaDesideri: set Pacchetto,
}

sig Dipendente extends UtenteRegistrato{
}

```

```

//----- DEFINIZIONE DEI FATTI -----

//non ci possono essere 2 utenti registrati con la stessa mail
fact NoEmailDuplicate{
    no disj u1, u2: UtenteRegistrato | u1.email = u2.email
}

//non ci possono essere 2 pacchetti con lo stesso nome
fact NoPacchettiUguali{
    no disj p1, p2: Pacchetto | (p1.nome = p2.nome) and ( (p1.voloAndata = p2.voloAndata) and (p1.voloRitorno = p2.voloRitorno) ) or
    (p1.attivitaSecondaria = p2.attivitaSecondaria) or
    (p1.hotel = p2.hotel) )
}

//non ci possono essere 2 voli con lo stesso codice
fact NoVoliDuplicati{
    no disj v1, v2: Volo | v1.codiceVolo = v2.codiceVolo
}

//non ci possono essere 2 hotel uguali nella stessa località (stesso nome, indirizzo, località)
fact NoHotelDuplicati{
    no disj h1, h2: Hotel | (h1.indirizzo = h2.indirizzo) and
    (h1.nome = h2.nome) and
    (h1.localita = h2.localita)
}

//non ci possono essere 2 attività con lo stesso nome con la stessa località di svolgimento
fact NoAttivitaSecondarieDuplicate{
    no disj a1, a2: AttivitaSecondaria | (a1.nome = a2.nome) and
    (a1.localitaValida = a2.localitaValida)
}

//tutti pacchetti devono appartenere al catalogo
fact PacchettiInCatalogo{
    all p: Pacchetto | p in Catalogo.insiemePacchetti
}

//tutte le prenotazioni devono appartenere allo StoricoOrdini
fact PrenotazioniInStoricoOrdini{
    all p: Prenotazione | p in StoricoOrdini.insiemePrenotazioni
}

//tutte le prenotazioni devono avere almeno un componente prenotato
fact NoPrenotazioniVuote{
    all p : Prenotazione | ( (#p.pacchetto = 1) or
    (#p.volo = 1) or
    (#p.hotel = 1) or
    (#p.attivitaSecondaria = 1) )
}

//tutte le prenotazioni devono avere i componenti disponibili
//il vincolo sul pacchetto non c'è perchè non possono esistere pacchetti non disponibili
fact NoPrenotazioniNonDisponibili{
    all p : Prenotazione | (#p.volo = 1 implies isTrue[p.volo.disponibilita]) and
    (#p.hotel = 1 implies isTrue[p.hotel.disponibilita]) and
    (#p.attivitaSecondaria = 1 implies isTrue[p.attivitaSecondaria.disponibilita])
}

//non posso aggiungere 2 volte lo stesso pacchetto alla lista desideri
fact NoDuplicatiInListaDesideri{
    no disj p1, p2 : Pacchetto, c : Cliente | (p1 in c.listaDesideri) and
    (p2 in c.listaDesideri) and
    (p1 = p2)
}

//il volo non puo partire e arrivare nella stessa citta
//NB: questo fatto è solo su volo e non su Trasporto, un ipotetico pullman potrebbe partire ed arrivare nella stessa citta
fact CittaDistinteInVolo{
    all v: Volo | v.cittaPartenza != v.cittaArrivo
}

```

```

//non può esistere un componente pacchetto che non sia almeno in un pacchetto
fact NoComponentePacchettoStandAlone{
  (all a : AttivitaSecondaria |
    some p: Pacchetto | a in p.attivitaSecondaria)
  and
  (all v : Volo |
    some p: Pacchetto | (v in p.voloAndata) or (v in p.voloRitorno))
  and
  (all h : Hotel |
    some p: Pacchetto | h in p.hotel)
}

//tutti i pacchetti devono avere tutte le componenti disponibili
fact NoPacchettiComponentiNonDisponibili{
  all p: Pacchetto | (isTrue[p.voloAndata.disponibilita] ) and
                    (isTrue[p.voloAndata.disponibilita] ) and
                    (isTrue[p.hotel.disponibilita]) and
                    (isTrue[p.attivitaSecondaria.disponibilita])
}

//fatto che modella tutte le date in gioco nel nostro modello
fact DateValidePacchetto{
  //fine validita del pacchetto deve essere > inizio validita (entrambe positive)
  (all p: Pacchetto | (p.fineValidita > p.inizioValidita) and (p.fineValidita > 0) and (p.inizioValidita > 0) and
    (p.voloAndata.dataArrivo < p.voloRitorno.dataPartenza))
  and
  //dataArrivo di un volo deve essere >= di quella di partenza, nel caso siano uguali l'ora di arrivo deve essere > di quella di partenza
  (all v: Volo | (v.dataArrivo >= v.dataPartenza) and (v.dataArrivo > 0) and (v.dataPartenza > 0) and
    (v.oraArrivo > 0) and (v.oraPartenza > 0)
    and (v.dataArrivo = v.dataPartenza implies v.oraArrivo > v.oraPartenza))
}

//il pacchetto deve avere i voli di andata e ritorno che arrivano e partono nella stessa città rispettivamente
//le date del volo di andata e del volo di partenza devono essere comprese tra l'inizio e la fine della validità del pacchetto
//il volo di andata e ritorno devono arrivare e partire nella stessa città dell'hotel del pacchetto
//il volo di ritorno e partenza devono arrivare e partire nella stessa città di partenza
fact PacchettoCoerente{
  all p: Pacchetto | (p.voloRitorno.cittaPartenza in p.voloAndata.cittaArrivo) and
    (p.voloAndata.dataPartenza >= p.inizioValidita) and (p.voloRitorno.dataPartenza <= p.fineValidita) and
    (p.hotel.localita = p.voloAndata.cittaArrivo) and (p.hotel.localita = p.voloRitorno.cittaPartenza) and
    (p.voloAndata.cittaPartenza = p.voloRitorno.cittaArrivo) and
    (p.attivitaSecondaria.localitaValida = p.hotel.localita)
}

```

Eseguito il check con Alloy Analyzer è stato restituito il seguente risultato:

6 commands were executed. The results are:

- #1: No counterexample found. NoAggiuntaPacchettoEsistente may be valid.
- #2: No counterexample found. NoPrenotazioneVuota may be valid.
- #3: No counterexample found. TuttePrenotazioniInStoricoOrdini may be valid.
- #4: No counterexample found. NoPacchettiNonDisponibili may be valid.
- #5: No counterexample found. NoPrenotazioneDiComponentiNonDisponibili may be valid.
- #6: No counterexample found. TuttiPacchettiConsistenti may be valid.

```

//----- DEFINIZIONE ASSERT -----

//se aggiungo un pacchetto gia esistente il catalogo deve restarmi uguale
//( inserimento duplicato in un insieme da insieme stesso)
assert NoAggiuntaPacchettoEsistente{
  all p: Pacchetto, c, c1: Catalogo |
    (p in c.insiemePacchetti and aggiungiPacchetto(c,c1,p))
  implies (c.insiemePacchetti = c1.insiemePacchetti)
}

check NoAggiuntaPacchettoEsistente for 10

//non può esistere una prenotazione contenete solo il cliente
assert NoPrenotazioneVuota{
  all p: Prenotazione |
    (one p.cliente) implies ( (one p.volo) or (one p.hotel) or (one p.attivitaSecondaria) or (one p.pacchetto) )
}

check NoPrenotazioneVuota for 10

assert TuttePrenotazioniInStoricoOrdini{
  no p : Prenotazione |
    p not in StoricoOrdini.insiemePrenotazioni
}

//check TuttePrenotazioniInStoricoOrdini for 10

//verifico che non ci siano pacchetti con componenti non disponibili
assert NoPacchettiNonDisponibili{
  no p: Pacchetto | ((not isTrue[p.voloAndata.disponibilita] ) and
    (not isTrue[p.voloRitorno.disponibilita] )) or
    (not isTrue[p.hotel.disponibilita]) or
    (not isTrue[p.attivitaSecondaria.disponibilita])
}

check NoPacchettiNonDisponibili for 10

//verifico che non ci siano prenotazioni su componenti non disponibili
assert NoPrenotazioneDiComponentiNonDisponibili{
  no p : Prenotazione | (#p.volo = 1 and not isTrue[p.volo.disponibilita]) or
    (#p.hotel = 1 and not isTrue[p.hotel.disponibilita]) or
    (#p.attivitaSecondaria = 1 and not isTrue[p.attivitaSecondaria.disponibilita])
}

check NoPrenotazioneDiComponentiNonDisponibili for 10

//verifico che tutti i pacchetti siano consistenti
assert TuttiPacchettiConsistenti{
  no p: Pacchetto | (p.voloRitorno.cittaPartenza != p.voloAndata.cittaArrivo) or
    (p.voloAndata.dataPartenza < p.inizioValidita) or (p.voloRitorno.dataPartenza > p.fineValidita) or
    (p.hotel.localita != p.voloAndata.cittaArrivo) or ( (p.hotel.localita != p.voloRitorno.cittaPartenza) or
    (p.voloAndata.cittaPartenza != p.voloRitorno.cittaArrivo) or
    (p.attivitaSecondaria.localitaValida != p.hotel.localita)
}

check TuttiPacchettiConsistenti for 10

```

```
//----- DEFINIZIONE DEI PREDICATI -----

pred show() {}

run show for 10

pred aggiungiPacchetto( c, c1 : Catalogo, p: Pacchetto ){
  c1.insiemePacchetti = c.insiemePacchetti + p
}
|
run aggiungiPacchetto for 10

pred aggiungiAListaDesideri(c, c1: Cliente, p : Pacchetto){
  c1.listaDesideri = c.listaDesideri + p
}

run aggiungiAListaDesideri for 10

pred eliminaDaListaDesideri(c,c1: Cliente, p: Pacchetto){
  c1.listaDesideri = c.listaDesideri - p
}

run eliminaDaListaDesideri for 10
```

Anche i run restituiscono predicati consistenti, come mostrato da questo risultato:

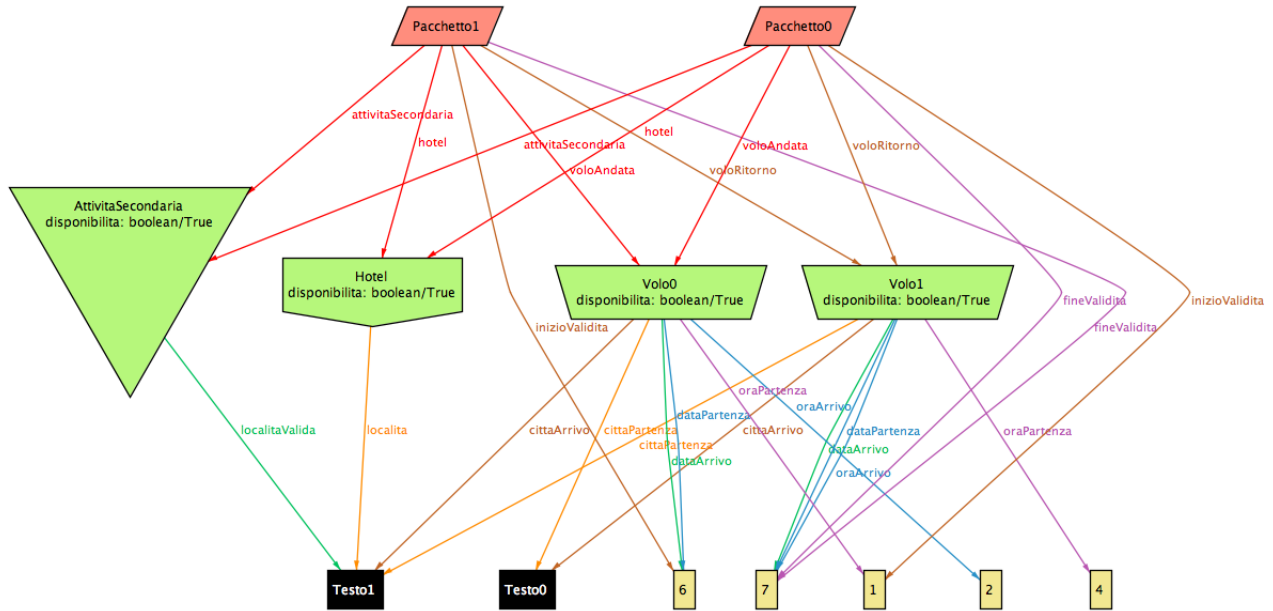
```
4 commands were executed. The results are:
#1: Instance found. show is consistent.
#2: Instance found. aggiungiPacchetto is consistent.
#3: Instance found. aggiungiAListaDesideri is consistent.
#4: Instance found. eliminaDaListaDesideri is consistent.
```

Questo risultato conferma la validità del modello, per un mondo di dimensione 10, ossia che contiene al massimo 10 elementi per ogni signature che è stata definita.

Inoltre, analizzando i mondi che sono stati generati, non sono state rilevate inconsistenze.

4.1.3 Mondi generati

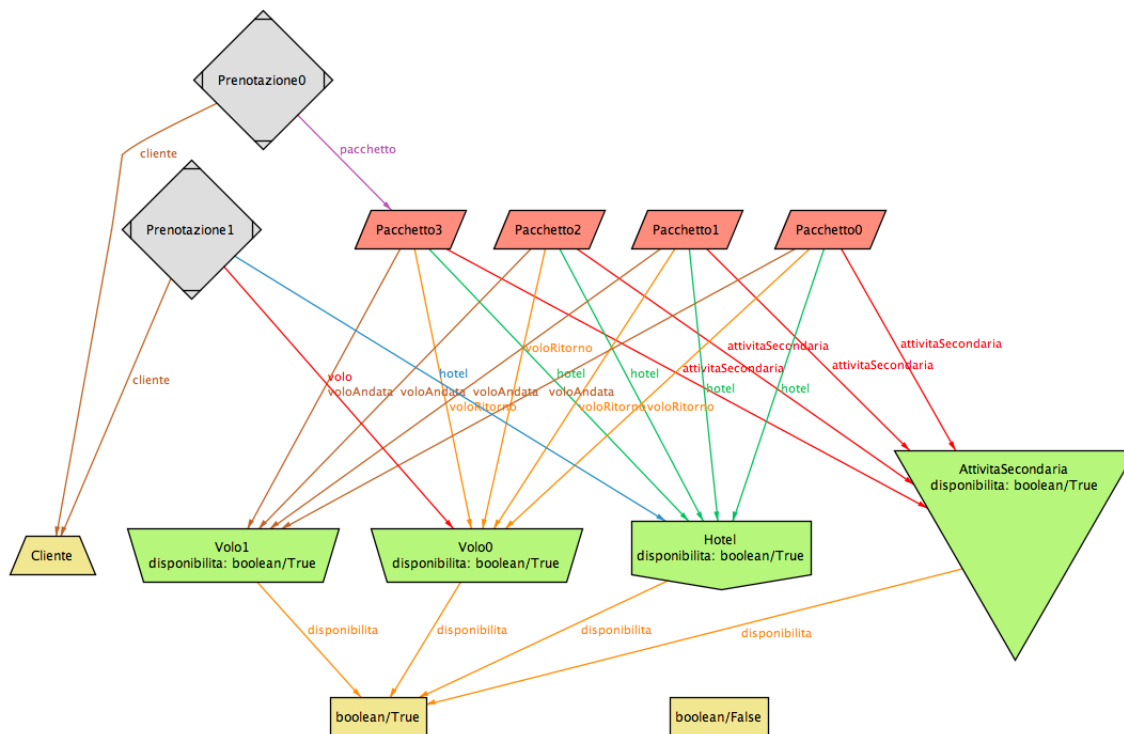
4.1.3.1 Mondo pacchetto consistente



Questo mondo verifica che tutti i pacchetti creati rispettino i vincoli imposti dal modello. Infatti (tutti gli esempi seguenti si riferiscono al pacchetto 0):

- tutti i pacchetti hanno tutte le componenti necessarie (volo d'andata, volo di ritorno, hotel, attività secondarie)
- le date dei voli di un pacchetto sono comprese tra inizio e fine della validità dello stesso (volo 0 d'andata in data 6 e volo 1 di ritorno in data 7, entrambe comprese tra data 1 di inizio validità e data 7 di fine validità)
- la data di fine validità di un pacchetto è sempre successiva alla data di inizio validità dello stesso (fine validità in data 7 > inizio validità in data 1)
- per ogni volo, nel caso in cui la data di arrivo e partenza sia la stessa, l'ora di partenza è minore dell'ora di arrivo (volo 0 ha data di arrivo e di partenza 6, ma ora arrivo 2 > ora partenza 1)
- la data di arrivo del volo di andata è sempre precedente la data di partenza del volo di ritorno (volo 0 di andata ha data di arrivo 6 < data di partenza 7 del volo 1 di ritorno)
- la città di partenza del volo di andata coincide con la città di arrivo del volo di ritorno e, viceversa, la città di arrivo del volo di andata coincide con la città di partenza del volo di ritorno (volo di andata 0 parte da Testo0 e arriva in Testo1, mentre il volo 1 di ritorno parte da Testo1 e arriva in Testo0)
- hotel e attività secondaria hanno la stessa località, e questa è la città di arrivo del volo di partenza, quindi quella di partenza del volo di ritorno (hotel, attività secondaria, città arrivo volo 0 di andata e città partenza del volo 1 di ritorno, hanno tutte località Testo1)

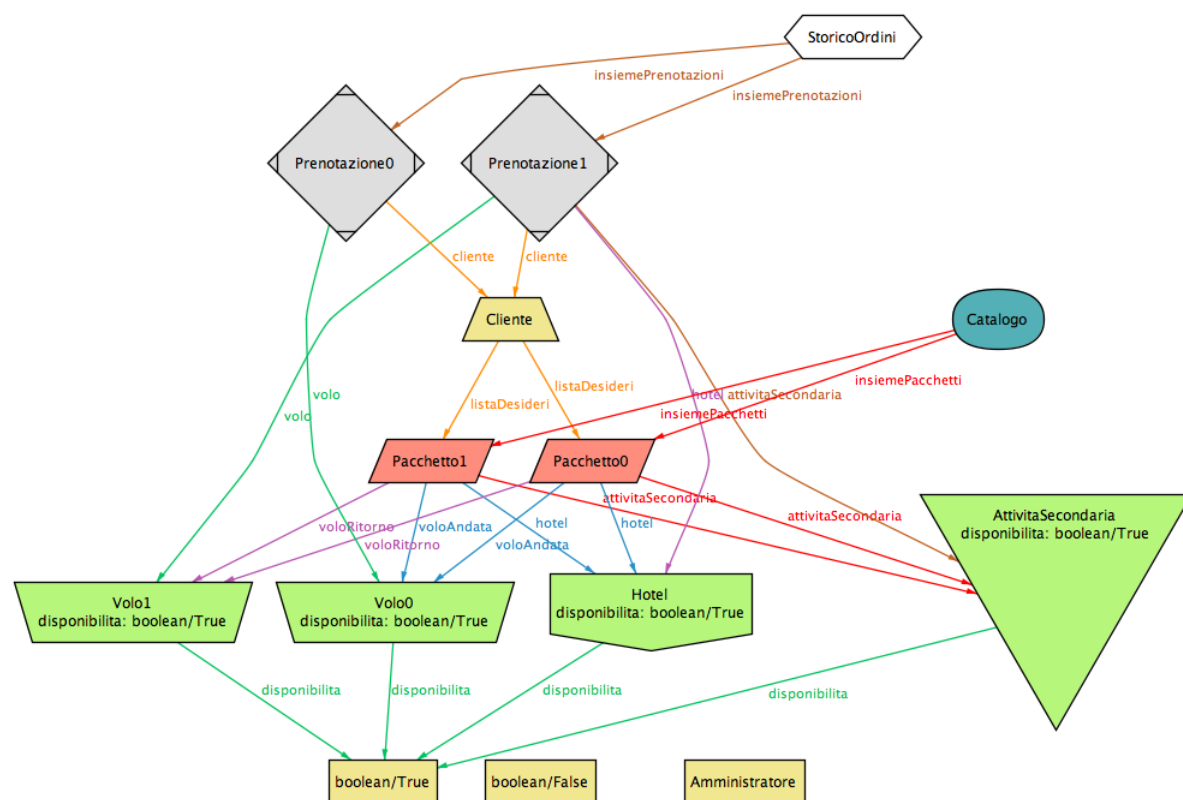
4.1.3.2 Mondo prenotazione consistente



Questo mondo verifica che tutte le prenotazioni create rispettino i vincoli imposti dal modello. Infatti:

- ogni prenotazione deve avere uno e un solo cliente
- ogni prenotazione deve avere uno o pi  componenti base, oppure uno e un solo pacchetto (prenotazione 1 ha volo e hotel, prenotazione 0 ha pacchetto 3)
- se la prenotazione comprende un pacchetto tutte le componenti del pacchetto devono essere disponibili, se la prenotazione comprende prodotti base questi ultimi devono essere disponibili

4.1.3.3 Mondo generale



Questo mondo fornisce una visione generale di tutti le entità e le relazioni significative del mondo non ancora mostrate, quali:

- il catalogo, che comprende tutti i pacchetti esistenti
- lo storico ordini, che comprende tutte le prenotazioni fatte da tutti i clienti
- la lista desideri, che comprende tutti i pacchetti aggiunti da un cliente nella stessa