

**POLITECNICO DI MILANO**  
Scuola di Ingegneria Industriale e dell'Informazione  
Corso di Laurea Magistrale in Ingegneria Informatica



Corso di Ingegneria del Software II  
**Referente Prof.ssa Mirandola**

**Applicazione web “TravelDream”**  
**Documento di Analisi e Specifica dei Requisiti 1.0.1**

Selenia Vincenza Rusalen - Marcello Felappi - Giovanni Battista Conserva

*Anno accademico 2013/2014*

# Indice

<b>1. Introduzione.....</b>	<b>3</b>
1.1. Scopo del progetto.....	3
1.2. Terminologia.....	3
<b>2. Generalità.....</b>	<b>4</b>
2.1. Caratteristiche dell'utente.....	4
2.2. Descrizione dell'applicazione.....	4
2.3. Funzionalità dell'applicazione.....	4
2.4. Assunzioni.....	5
2.5. Limitazioni.....	5
2.6. Possibili estensioni.....	5
2.7. Documentazione dell'applicazione.....	5
<b>3. Requisiti specifici.....</b>	<b>6</b>
3.1. Requisiti di interfaccia.....	6
3.1.1. Ipotesi di interfaccia grafica.....	6
3.2. Requisiti funzionali.....	8
3.2.1. Scenari d'utilizzo.....	8
3.2.1.1. Nuovo impiegato crea pacchetto.....	8
3.2.1.2. Nuovo cliente acquista pacchetto personalizzato.....	9
3.2.1.3. Amico conferma partecipazione ad un viaggio.....	9
3.2.1.4. Invitato acquista prodotto.....	9
3.2.2. Diagramma di Caso d'Uso.....	10
3.2.3. Casi d'uso.....	11
3.2.3.1. Registrazione.....	11
3.2.3.2. Login.....	11
3.2.3.3. Logout.....	11
3.2.3.4. Navigazione.....	12
3.2.3.5. Ricerca.....	12
3.2.3.6. Inserimento prodotto.....	12
3.2.3.7. Modifica prodotto.....	13
3.2.3.8. Eliminazione prodotto.....	13
3.2.3.9. Creazione pacchetto.....	14
3.2.3.10. Modifica pacchetto.....	14
3.2.3.11. Eliminazione pacchetto.....	14
3.2.3.12. Personalizzazione pacchetto.....	15
3.2.3.13. Acquisto.....	15
3.2.3.14. Invito amici.....	16
3.2.3.15. Accettazione partecipazione.....	16
3.2.3.16. Creazione gift list.....	16
3.2.3.17. Scelta gift.....	17
3.3. Requisiti di performance.....	17
3.4. Limitazioni.....	17
3.4.1. Limitazioni hardware.....	17
3.4.2. Limitazioni software.....	17
3.5. Proprietà del software.....	18
3.5.1. Affidabilità.....	18
3.5.2. Disponibilità.....	18

3.5.3. Sicurezza.....	18
3.5.4. Manutenibilità.....	18
3.5.5. Portabilità.....	18
3.5.6. Accuratezza.....	18
3.5.7. Scalabilità.....	19
3.5.8. Robustezza.....	19
3.6. Altri requisiti.....	19
<b>4. Appendice.....</b>	<b>20</b>
4.1. Modelli di analisi.....	20
4.1.1. Diagrammi di Attività.....	20
4.1.1.1. Impiegato crea pacchetto.....	20
4.1.1.2. Impiegato modifica pacchetto.....	21
4.1.1.3. Impiegato elimina pacchetto.....	22
4.1.1.4. Cliente invita amici.....	23
4.1.1.5. Cliente crea giftlist.....	24
4.1.2. Diagramma di Stato.....	25
4.1.3. Diagramma delle Classi.....	26
4.1.4. Alloy.....	27
4.1.4.1. Modello sistema in generale.....	27
4.1.4.2. Modello controllo consistenza.....	31
4.2. Modifiche in itinere.....	33
4.3. Tempi di elaborazione.....	33

# Introduzione

## 1.1. Scopo del progetto

Il progetto si pone come obiettivo lo sviluppo di un'applicazione web che consenta all'impresa TravelDream di vendere i suoi prodotti online. Tale applicazione dovrà dunque essere in grado di supportare il personale aziendale nella pubblicazione di contenuti da offrire alla clientela e la clientela stessa nell'atto di selezione dei prodotti disponibili che ritenga consoni alle sue esigenze. Nella fattispecie, l'impresa si occupa di vendita di pacchetti viaggio da lei confezionati.

## 1.2. Terminologia

Nel seguito si utilizzeranno frequentemente i seguenti termini con la connotazione indicata:

Termine	Accezione
Impiegato o <i>Employee</i>	Impiegato dell'impresa TravelDream con mansioni di gestione dei prodotti da pubblicare sul sito.
Cliente o <i>Customer</i>	Utente che si registri al sito con l'intenzione di acquistare un pacchetto viaggio, o ne abbia acquistato uno in precedenza.
Amico o <i>Friend</i>	Utente invitato sul sito da un cliente allo scopo di farlo diventare a sua volta un cliente.
Invitato	Utente che abbia ricevuto da un cliente un invito alla propria gift list, allo scopo di effettuare un pagamento per un prodotto di cui usufruirà il cliente invitante.
Utente o <i>User</i>	Generico utilizzatore del sito.
Prodotto o <i>Product</i>	Singolo componente di un viaggio, quale ad esempio un volo, una prenotazione alberghiera, o un'escursione.
Tappa o <i>Stage</i>	Insieme di prodotti che facciano riferimento ad una medesima località.
Pacchetto o <i>Package</i>	Insieme di una o più tappe acquistabile da un cliente.

# Generalità

## 2.1. Caratteristiche dell'utente

Si prevede che l'applicazione web da sviluppare sia fruita principalmente ad opera di due categorie di utenti, quali gli impiegati e gli utenti generici potenzialmente clienti o invitati di clienti.

Per i primi si suppone si tratti di individui adulti, di media cultura e discrete conoscenze informatiche, mentre per i secondi si suppone si tratti di individui giovani o adulti, di media cultura e conoscenze informatiche elementari.

## 2.2. Descrizione dell'applicazione

L'applicazione sviluppata si connoterà come una nuova applicazione web *self-contained*, realizzata *ex novo* e non in sostituzione o ad integrazione di strutture preesistenti.

## 2.3. Funzionalità dell'applicazione

L'applicazione dovrà mettere a disposizione alcune funzionalità per espletare il suo scopo, quali:

- Registrazione – dovrà essere consentito ad impiegati e clienti di registrarsi al sito tramite l'inserimento di alcuni dati personali, al fine di autenticare gli accessi;
- Autenticazione – gli utenti registrati dovranno aver modo di autenticarsi sul sito tramite una procedura di login;
- Inserimento prodotti – gli impiegati dovranno poter pubblicare sul sito informazioni relative ai nuovi prodotti disponibili;
- Creazione pacchetti – gli impiegati dovranno poter creare dei pacchetti viaggio assemblando vari prodotti disponibili;
- Navigazione – gli utenti dovranno poter navigare all'interno dell'elenco di pacchetti per visualizzarne le informazioni;
- Ricerca – gli utenti dovranno avere la facoltà di cercare un determinato prodotto in base ad esigenze specifiche;
- Personalizzazione – i clienti dovranno poter formulare una propria combinazione di prodotti personalizzata;
- Acquisto – una volta scelta la propria soluzione, il cliente dovrà essere in condizioni di procedere all'acquisto;
- Invito amici – al cliente dovrà essere offerta l'opportunità di invitare degli amici a viaggiare con lui, permettendo loro di visualizzare il suo pacchetto personalizzato;
- Conferma partecipazione – gli amici del cliente dovranno poter confermare la partecipazione al viaggio previa registrazione al sito;
- Creazione gift list – al cliente dovrà essere permessa la creazione di una gift list di prodotti che i suoi invitati possano acquistare per lui;
- Selezione gift – agli invitati del cliente dovrà essere consentito di scegliere quale prodotto regalarli.

## **2.4. Assunzioni**

Nell'elaborazione di un modello ci si baserà sulle seguenti assunzioni:

- La navigazione e la ricerca di pacchetti all'interno del sito siano disponibili per qualsiasi utente;
- Per il riconoscimento degli impiegati si utilizzi una matricola impiegato gestita dall'azienda e resa disponibile all'applicazione;
- La modifica e l'eliminazione ad opera di un impiegato di prodotti e pacchetti ancora disponibili al pubblico sia da ritenersi un evento saltuario dovuto a cause di forza maggiore. Si ipotizza pertanto che eventuali disagi per i clienti siano gestiti da un apposito servizio clienti;
- L'acquisto di un pacchetto debba seguire immediatamente la sua selezione e/o personalizzazione, e non ne possa essere garantita la disponibilità altrimenti;
- La creazione di una gift list avvenga solo in seguito alla selezione e/o personalizzazione di un pacchetto;
- Una volta confermata la gift list, i prodotti presenti vadano ritenuti acquistati. In caso di mancato acquisto da parte degli invitati, il cliente sarà invitato al saldo di quanto dovuto.

## **2.5. Limitazioni**

Nel processo di sviluppo dell'applicazione dovranno essere prese in considerazione le seguenti limitazioni:

- specifiche indicazioni riguardo al linguaggio di programmazione e alla tecnologia da utilizzare;
- ridotto tempo a disposizione per il completamento del processo;

## **2.6. Possibili estensioni**

Si prevede che versioni successive dell'applicazione richiedano di estendere la gamma di prodotti inclusi nell'offerta anche ad altri generi di servizio, come ad esempio trasporto via treno e noleggio auto.

## **2.7. Documentazione dell'applicazione**

I documenti che accompagneranno il rilascio dell'applicazione saranno:

- documento di analisi e specifica dei requisiti, in cui saranno illustrati gli obiettivi che si intende perseguire nella realizzazione del progetto;
- documento di design del progetto, in cui saranno dettagliate le scelte progettuali;
- documento di piano di testing, nel quale saranno illustrate le modalità con cui si sarà verificato il corretto funzionamento di quanto prodotto;
- documento con risultati del testing;
- manuale d'uso per l'utente.

## Requisiti specifici

### 3.1. Requisiti di interfaccia

Benché non siano stati formulati particolari requisiti concernenti l'interfaccia, si considera comunque opportuno realizzare un'interfaccia grafica che minimizzi il rischio di sovraccarico cognitivo e sia quanto più possibile scevra da ambiguità, in modo da risultare intuitiva e immediatamente utilizzabile dal target d'utenza ipotizzato. Si intende quindi ridurre al minimo il tempo di addestramento necessario agli impiegati per l'espletamento delle loro mansioni e massimizzare l'efficacia del sito in generale.

#### 3.1.1. Ipotesi di interfaccia grafica

Seguono alcune ipotesi di interfaccia dell'applicazione.

The wireframe shows the layout of the TravelDream homepage. At the top center is the title "TravelDream". On the left is a vertical sidebar with four buttons: "I NOSTRI PACCHETTI", "PROMOZIONI", "ALTRO", and an empty button. The main content area has a header with two tabs: "Cliente" (selected) and "Impiegato". Below the "Cliente" tab, there are two sections: "Accedi" (Login) with fields for "Username" and "Password", and "Registrati" (Register) with fields for "Nome", "Cognome", "Email", and "Altro". To the right of the login/register section is a "Cerca nel sito" (Search) box with fields for "Data", "Località", and "Nome pacchetto". Below the search box is a "Devi fare un regalo?" (Do you want to give a gift?) section with a field labeled "Inserisci il tuo codice qui:".

- Homepage -

# TravelDream

[Home](#) > [Pannello Impiegato](#)

PACCHETTI

PROMOZIONI

ALTRO

Nuovo prodotto

Modifica prodotto

Elimina prodotto

Nuovo pacchetto

Modifica pacchetto

Elimina pacchetto

- Pannello Impiegato -

# TravelDream

[Home](#) > [Pannello Impiegato](#) > [Nuovo pacchetto](#)

PACCHETTI

PROMOZIONI

ALTRO

Inserisci i dati del nuovo pacchetto:

Nome pacchetto:

Descrizione:

Data partenza:  

Data rientro:  

Prodotti correnti:

Aggiungi volo

Aggiungi albergo

Aggiungi escursione

Rimuovi prodotto

Conferma creazione

Annulla

- Pannello Impiegato: Creazione pacchetto -



# TravelDream

[Home > Pacchetti](#)

PACCHETTI	<a href="#">Nella terra dei canguri</a>	Feste in musica - Viaggio per 1 persona. Località: Vienna Partenza: Aeroporto Milano Malpensa Ore 14.00 Giorno 23/12/2013 Ritorno: Aeroporto Wien-Schwechat Ore 17.00 Giorno 02/12/2014 Soggiorno: Hotel Plaza, Schottenring 11, 1010 Vienna, Austria Escursioni: Concerto di Natale, Solisti Orchestra da camera di Vienna Ore 18.00 Giorno 24/12/2013 Luogo: Kursalon Wien Concerto di Capodanno, Orchestra Filarmonica di Vienna, Ore 11.15 Giorno 01/01/2014 Luogo: Musikverein Visita guidata Vienna Ore 11.00 Giorno 28/12/2013 Visita guidata Vienna Ore 14.00 Giorno 29/12/2013  Goditi la magia della musica nella terra di grandi compositori, nella suggestiva cornice di una Vienna innevata...
PROMOZIONI	<a href="#">Dalla Cina con furore</a>	
ALTRO	<a href="#">Feste in musica</a>	
	<a href="#">Pirati dei Caraibi</a>	
...		

- Elenco pacchetti -

## 3.2. Requisiti funzionali

### 3.2.1. Scenari d'utilizzo

Si ipotizzano i seguenti scenari come probabili sequenze di operazioni svolte dagli utilizzatori dell'applicazione.

#### 3.2.1.1. Nuovo impiegato crea pacchetto

- **Attore:** Roberto, un impiegato
- **Descrizione:** Roberto, impiegato TravelDream da poco assegnato alla gestione del sito, viene incaricato di confezionare un nuovo pacchetto viaggio. Accede dunque al sito e seleziona l'opzione di registrazione per impiegati. Inserisce quindi alcuni dati personali tra cui la matricola impiegato, il suo nuovo nickname "Bob70" e la sua nuova password "iosonoBoB70".

Completata con successo l'operazione, può loggarsi sul sito con i suoi nuovi dati di login (nickname e password). Una volta loggato seleziona l'opzione di creazione di un nuovo pacchetto. Inserisce perciò il nome del nuovo pacchetto "Feste in musica" e le date di inizio e fine vacanza, rispettivamente 23/12/2013 e 02/01/2014. Seleziona dunque dall'elenco di voli disponibili un volo da Milano a Vienna alle 14.00 del giorno 23 dicembre. Procede quindi alla selezione di una camera singola presso l'albergo Plaza di Vienna per il periodo indicato. Seleziona anche un posto per il volo di rientro da Vienna a Milano previsto per le 17.00 del 2 gennaio.

Per completare l'opera, aggiunge al pacchetto le partecipazioni a due concerti d'orchestra previsti per la Vigilia e l'ultimo dell'anno e due visite guidate per la città previste rispettivamente per il 28 e il 29 dicembre.

Ritenendosi soddisfatto del risultato, Roberto conferma la creazione del nuovo pacchetto.

### 3.2.1.2. Nuovo cliente acquista pacchetto personalizzato

- **Attore:** Filippo, un utente
- **Descrizione:** Filippo, intenzionato a prenotarsi una vacanza al mare, approda sul sito di TravelDream. Digita dunque il termine “Caraibi” nell'apposito modulo di ricerca. Il primo risultato a comparire è il pacchetto “Pirati dei Caraibi”. Dopo averlo esaminato per qualche minuto conclude di potersi ritenere soddisfatto con l'apporto di qualche piccola modifica. Procede dunque alla registrazione inserendo i dati personali richiesti, il nuovo nickname “Pippo76” e la password “iosonoPippo76”.  
Conclusa con successo l'operazione procede all'autenticazione inserendo i nuovi dati di login nell'apposito form.  
Una volta autenticato si dedica alla modifica del pacchetto prescelto. Rimuove quindi le serate a tema piratesco previste dall'offerta e le sostituisce con una gita in barca.  
Ritenendo a questo punto il pacchetto conforme ai suoi desideri lo conferma e procede al pagamento.

### 3.2.1.3. Amico accetta partecipazione ad un viaggio

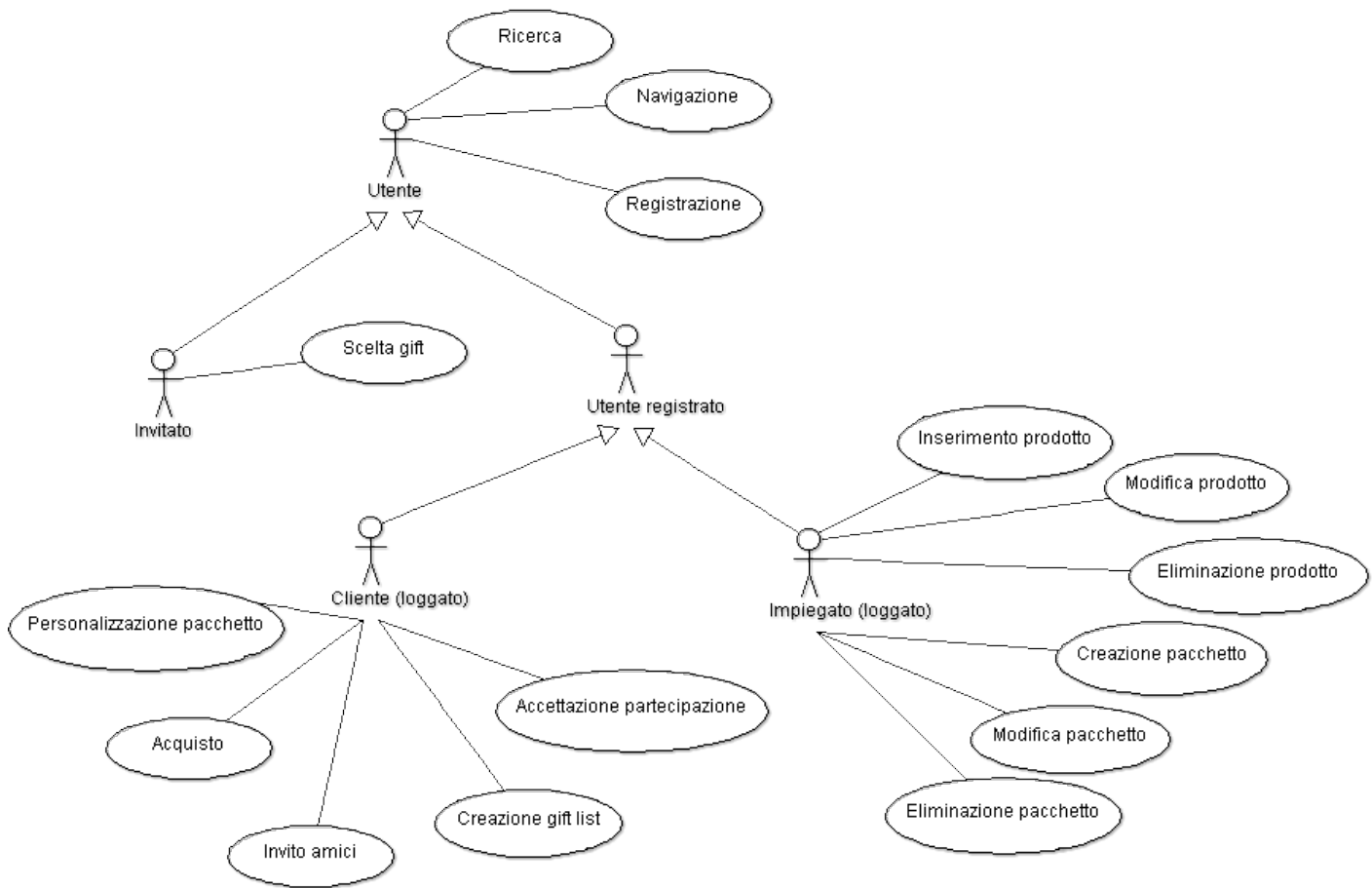
- **Attori:** Filippo, Marco
- **Descrizione:** Filippo, soddisfatto della precedente vacanza, decide di avvalersi nuovamente dei servizi TravelDream e di invitare anche il suo amico Marco a viaggiare con lui. Perciò, una volta individuato il pacchetto di suo gusto “Nella terra dei canguri”, decide di acquistarlo e di procedere all'invito di amici. Compila dunque l'apposito form e conferma l'invio dell'invito.  
Marco, controllando la sua posta elettronica, nota la mail speditagli dal sito con l'invito a partecipare al viaggio. Si reca dunque al link indicato nella mail e inserisce il codice in essa riportato nel form apposito, avendo in questo modo la possibilità di visualizzare il pacchetto.  
Convinto della proposta, accetta la partecipazione al viaggio e procede alla registrazione e all'acquisto.

### 3.2.1.4. Invitato acquista prodotto

- **Attori:** Marco, Giulia
- **Descrizione:** Marco, prossimo alle nozze, decide di usufruire della possibilità offerta dal sito TravelDream di farsi offrire un viaggio dagli invitati. Una volta individuato il pacchetto per un interrail in varie località asiatiche “Dalla Cina con furore”, seleziona quindi l'opzione di creazione di una gift list. Preleva dunque il codice da inviare agli invitati e lo spedisce in primo luogo alla sorella Giulia.  
Giulia, ricevuto il codice, si reca sul sito e lo inserisce nell'apposito form, accedendo così al pacchetto personalizzato dal fratello. Decide di pagare per il volo da Pechino a Canton e quindi seleziona tale prodotto e procede al pagamento.

### 3.2.2. Diagramma di Caso d'Uso

Le funzionalità dell'applicazione in base alla tipologia di utilizzatore possono essere schematizzate come nel seguente diagramma:



- Diagramma di Caso d'Uso generale -

### 3.2.3. Casi d'uso

Le principali modalità di utilizzo sono trattate in modo maggiormente dettagliato nel seguito.

#### 3.2.3.1. Registrazione

Il sistema dovrà consentire ai dipendenti dell'azienda e ai potenziali clienti di registrarsi inserendo alcuni dati personali e, nel caso dei dipendenti, l'apposita matricola.

<b>Titolo</b>	Registrazione
<b>Attori</b>	Utente del sito
<b>Precondizione</b>	L'utente accede alla schermata di registrazione.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• L'utente compila i campi di un apposito form con i suoi dati personali;</li><li>• Conferma l'intenzione di registrarsi.</li></ul>
<b>Postcondizione</b>	Il sistema accetta i dati inseriti confermando l'avvenuta registrazione.
<b>Eccezioni</b>	<ul style="list-style-type: none"><li>• L'utente risulta già registrato. Il sistema segnala l'impossibilità a procedere.</li><li>• I dati inseriti non risultano validi. Il sistema richiede all'utente di correggerli e/o integrarli.</li></ul>

#### 3.2.3.2. Login

Il sistema dovrà consentire ad impiegati e clienti che abbiano effettuato la registrazione di autenticare i propri accessi al sito.

<b>Titolo</b>	Login
<b>Attori</b>	Utente del sito
<b>Precondizione</b>	Gli utenti si sono precedentemente registrati come impiegati o clienti.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• L'utente inserisce i propri dati di login;</li><li>• Conferma l'intenzione di loggarsi.</li></ul>
<b>Postcondizione</b>	Il sistema riconosce i dati di login dell'utente.
<b>Eccezioni</b>	<ul style="list-style-type: none"><li>• L'utente non risulta registrato. Il sistema invita a procedere in tal senso.</li><li>• I dati inseriti non sono corretti. Il sistema richiede di ritentare l'inserimento.</li></ul>

#### 3.2.3.3. Logout

Il sistema dovrà consentire agli utenti loggati di effettuare un logout e tornare a visualizzare il sito come utenti generici.

<b>Titolo</b>	Logout
<b>Attori</b>	Cliente o impiegato
<b>Precondizione</b>	Il cliente o impiegato ha precedentemente effettuato un login.

<b>Flusso di eventi</b>	Il cliente o impiegato conferma l'intenzione di effettuare il logout.
<b>Postcondizione</b>	Il sistema accetta il comando concludendo la sessione autenticata dell'utente.
<b>Eccezioni</b>	/

#### 3.2.3.4. Navigazione

Il sistema dovrà consentire a tutti gli utenti di visualizzare alcune pagine pubbliche, e in particolare di prendere visione dell'offerta di pacchetti.

<b>Titolo</b>	Navigazione
<b>Attori</b>	Utente del sito
<b>Precondizione</b>	/
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• L'utente accede ad una schermata con l'elenco dei pacchetti disponibili;</li> <li>• Prende visione dei pacchetti;</li> <li>• Procedo con l'esplorazione.</li> </ul>
<b>Postcondizione</b>	L'utente decide alternativamente di: <ul style="list-style-type: none"> <li>• Procedere all'acquisto o alla personalizzazione di un pacchetto.</li> <li>• Lasciare il sito.</li> </ul>
<b>Eccezioni</b>	/

#### 3.2.3.5. Ricerca

Il sistema dovrà permettere a tutti gli utenti di cercare un pacchetto in base ad esigenze specifiche.

<b>Titolo</b>	Ricerca
<b>Attori</b>	Utente del sito
<b>Precondizione</b>	/
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• L'utente inserisce in un apposito form alcune informazioni circa il pacchetto desiderato;</li> <li>• Conferma l'intenzione di effettuare una ricerca.</li> </ul>
<b>Postcondizione</b>	Il sistema restituisce, se esistono, i pacchetti corrispondenti alle richieste dell'utente.
<b>Eccezioni</b>	/

#### 3.2.3.6. Inserimento prodotto

Il sistema dovrà permettere ad un impiegato di inserire nuovi prodotti.

<b>Titolo</b>	Inserimento prodotto
<b>Attori</b>	Impiegato

<b>Precondizione</b>	L'impiegato ha effettuato il login.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• L'impiegato seleziona l'opzione di inserimento prodotto;</li> <li>• Compila un apposito form con i dati del prodotto richiesti;</li> <li>• Conferma l'inserimento.</li> </ul>
<b>Postcondizione</b>	Il sistema accetta l'inserimento del prodotto.
<b>Eccezioni</b>	<ul style="list-style-type: none"> <li>• Il prodotto è già presente. Il sistema segnala l'impossibilità a procedere.</li> <li>• I dati del prodotto sono incompleti. Il sistema richiede di correggerli e/o integrarli.</li> </ul>

### 3.2.3.7. Modifica prodotto

Il sistema dovrà consentire ad un impiegato di modificare i dati relativi ad un dato prodotto.

<b>Titolo</b>	Modifica prodotto
<b>Attori</b>	Impiegato
<b>Precondizione</b>	L'impiegato loggato ha visualizzato il prodotto che intende modificare.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• L'impiegato seleziona l'opzione di modifica;</li> <li>• Compila un apposito form con i nuovi dati del prodotto;</li> <li>• Conferma la modifica.</li> </ul>
<b>Postcondizione</b>	Il sistema accetta le modifiche effettuate.
<b>Eccezioni</b>	<ul style="list-style-type: none"> <li>• I dati modificati del prodotto sono incompleti. Il sistema richiede di correggerli e/o integrarli.</li> <li>• Il prodotto da modificare appartiene ad un pacchetto. Nel caso la modifica non rispetti la coerenza interna del pacchetto, si rende necessaria un'operazione di modifica di quest'ultimo, con relativa notifica ad eventuali acquirenti.</li> </ul>

### 3.2.3.8. Eliminazione prodotto

Il sistema dovrà consentire ad un impiegato di eliminare un dato prodotto non più disponibile.

<b>Titolo</b>	Eliminazione prodotto
<b>Attori</b>	Impiegato
<b>Precondizione</b>	L'impiegato loggato ha visualizzato il prodotto che intende eliminare.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• L'impiegato seleziona l'opzione di eliminazione;</li> <li>• Conferma l'eliminazione.</li> </ul>
<b>Postcondizione</b>	Il sistema accetta la cancellazione del prodotto.
<b>Eccezioni</b>	Il prodotto appartiene ad un pacchetto. Si rende necessaria una modifica di quest'ultimo, con relativa notifica ad eventuali acquirenti.

### 3.2.3.9. Creazione pacchetto

Il sistema dovrà consentire ad un impiegato di confezionare un pacchetto da vendere sul sito.

<b>Titolo</b>	Creazione pacchetto
<b>Attori</b>	Impiegato
<b>Precondizione</b>	L'impiegato ha effettuato il login.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• L'impiegato seleziona l'opzione di creazione di un nuovo pacchetto;</li><li>• Inserisce alcuni dati caratterizzanti del pacchetto (come nome e date);</li><li>• Seleziona una o più località in cui sarà effettuato il viaggio;</li><li>• Aggiunge alcuni prodotti al pacchetto prelevandoli dall'elenco di quelli disponibili;</li><li>• Conferma la creazione del pacchetto.</li></ul>
<b>Postcondizione</b>	Il sistema accetta il pacchetto creato.
<b>Eccezioni</b>	<ul style="list-style-type: none"><li>• I prodotti selezionati manifestano incoerenze temporali e/o spaziali. Il sistema segnala l'impossibilità a procedere.</li><li>• Il pacchetto è già presente. Il sistema segnala l'impossibilità a procedere.</li></ul>

### 3.2.3.10. Modifica pacchetto

Il sistema dovrà permettere ad un impiegato di modificare un pacchetto presente sul sito.

<b>Titolo</b>	Modifica pacchetto
<b>Attori</b>	Impiegato
<b>Precondizione</b>	L'impiegato loggato ha visualizzato il pacchetto che intende modificare.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• L'impiegato seleziona l'opzione di modifica di un pacchetto;</li><li>• Compila un form con gli eventuali nuovi dati del pacchetto;</li><li>• Eventualmente aggiunge o rimuove uno o più prodotti dal pacchetto;</li><li>• Conferma le modifiche effettuate.</li></ul>
<b>Postcondizione</b>	Il sistema accetta le modifiche effettuate.
<b>Eccezioni</b>	Il pacchetto in questione è già stato acquistato. Il sistema notifica l'accaduto al cliente in modo che possa decidere se adeguarsi alla modifica o richiedere un rimborso.

### 3.2.3.11. Eliminazione pacchetto

Il sistema dovrà consentire ad un impiegato di eliminare un pacchetto presente sul sito.

<b>Titolo</b>	Eliminazione pacchetto
<b>Attori</b>	Impiegato
<b>Precondizione</b>	L'impiegato loggato ha visualizzato il pacchetto che intende modificare.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• L'impiegato seleziona l'opzione di modifica di un pacchetto;</li></ul>

	<ul style="list-style-type: none"> <li>• Compila un form con gli eventuali nuovi dati del pacchetto;</li> <li>• Eventualmente aggiunge o rimuove uno o più prodotti dal pacchetto;</li> <li>• Conferma le modifiche effettuate.</li> </ul>
<b>Postcondizione</b>	Il sistema accetta le modifiche effettuate.
<b>Eccezioni</b>	Il pacchetto in questione è già stato acquistato. Il sistema notifica l'accaduto al cliente.

### 3.2.3.12. Personalizzazione pacchetto

Il sistema dovrà mettere il cliente in condizione di personalizzare un pacchetto viaggio secondo le sue esigenze.

<b>Titolo</b>	Personalizzazione pacchetto
<b>Attori</b>	Cliente
<b>Precondizione</b>	Il cliente loggato ha individuato il pacchetto che intende personalizzare.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• Il cliente seleziona l'opzione di personalizzazione di un pacchetto;</li> <li>• Inserisce opzionalmente alcuni dati concernenti le sue esigenze specifiche in termini di tempo e luogo;</li> <li>• Eventualmente aggiunge o rimuove prodotti dal pacchetto;</li> <li>• Conferma la personalizzazione.</li> </ul>
<b>Postcondizione</b>	Il sistema accetta il pacchetto personalizzato e permette di procedere all'acquisto.
<b>Eccezioni</b>	Il pacchetto presenta delle incoerenze interne in termini di tempo e spazio. Il sistema notifica la loro presenza, ma permette ugualmente di procedere.

### 3.2.3.13. Acquisto

Il sistema dovrà consentire al cliente di acquistare un pacchetto confezionato o personalizzato. Le transazioni monetarie sono delegate ad un sistema di pagamento esterno.

<b>Titolo</b>	Acquisto
<b>Attori</b>	Cliente
<b>Precondizione</b>	Il cliente loggato ha confermato un pacchetto.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• Il cliente seleziona l'opzione di acquisto;</li> <li>• Procede all'acquisto tramite un apposito sistema di pagamento.</li> </ul>
<b>Postcondizione</b>	Il sistema conferma l'avvenuta transazione.
<b>Eccezioni</b>	Eventuali errori durante la transazione. Il sistema consente di riprovare.



### 3.2.3.14. Invito amici

Il sistema dovrà permettere ad un cliente di invitare degli amici a partecipare al suo stesso viaggio.

<b>Titolo</b>	Invito amici
<b>Attori</b>	Cliente
<b>Precondizione</b>	Il cliente loggato ha confermato un pacchetto personalizzato e lo visualizza nella sua pagina personale.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• Il cliente seleziona l'opzione di invito amici;</li><li>• Compila un form con gli indirizzi degli amici da invitare;</li><li>• Conferma l'invito;</li></ul>
<b>Postcondizione</b>	Il sistema conferma il successo dell'operazione.
<b>Eccezioni</b>	Non vi sono abbastanza prodotti disponibili per creare un pacchetto viaggio per l'amico. Il sistema segnala l'impossibilità a procedere.

### 3.2.3.15. Accettazione partecipazione

Il sistema dovrà permettere agli amici del cliente di visualizzare il pacchetto che sono stati invitati ad acquistare ed eventualmente di accettare l'offerta.

<b>Titolo</b>	Accettazione partecipazione
<b>Attori</b>	Amico
<b>Precondizione</b>	Un cliente ha concluso con successo una procedura di Invito amici in relazione al proprio pacchetto.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"><li>• L'amico riceve il codice del pacchetto inviatogli dal cliente;</li><li>• Accede al sito e inserisce il codice in un apposito form;</li><li>• Prende visione del pacchetto;</li><li>• Procede alla registrazione (vedi caso d'uso);</li><li>• Procede al login (vedi caso d'uso);</li><li>• Inserisce nuovamente il codice nell'apposito form;</li><li>• Procede all'acquisto (vedi caso d'uso).</li></ul>
<b>Postcondizione</b>	Il sistema conferma l'avvenuto pagamento.
<b>Eccezioni</b>	Il pacchetto selezionato non è più disponibile in quanto i prodotti siano già stati acquistati da altri nel frattempo. Il sistema segnala l'impossibilità a procedere.

### 3.2.3.16. Creazione gift list

Il sistema dovrà permettere al cliente di far sì che siano i suoi invitati a pagare i vari componenti del suo pacchetto per lui.

<b>Titolo</b>	Creazione gift list
<b>Attori</b>	Cliente

<b>Precondizione</b>	Il cliente loggato ha confermato un pacchetto.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• Il cliente seleziona l'opzione di creazione di una gift list;</li> <li>• Visualizza il codice da spedire ai suoi invitati;</li> </ul>
<b>Postcondizione</b>	Il sistema accetta la creazione della gift list come modalità di pagamento.
<b>Eccezioni</b>	/

### 3.2.3.17. Scelta gift

Il sistema dovrà consentire agli invitati di un cliente di procedere alla scelta e al pagamento di una componente del viaggio da questo selezionato.

<b>Titolo</b>	Scelta gift
<b>Attori</b>	Invitato
<b>Precondizione</b>	L'invitato ha ricevuto un codice da un cliente.
<b>Flusso di eventi</b>	<ul style="list-style-type: none"> <li>• L'invitato inserisce il codice nell'apposito form;</li> <li>• Seleziona il prodotto che intende regalare al cliente;</li> <li>• Conferma l'acquisto.</li> </ul>
<b>Postcondizione</b>	Il sistema conferma la corretta conclusione dell'operazione.
<b>Eccezioni</b>	Eventuali errori durante la transazione. Il sistema consente di riprovare.

## 3.3. Requisiti di performance

Benché non siano stati formulati specifici requisiti di performance, si ritiene che il tempo di risposta alla richiesta di un utente debba essere mantenuto inferiore ai 10 secondi al fine di evitare eventuali perdite di interesse da parte dello stesso.

## 3.4. Limitazioni

### 3.4.1. Limitazioni hardware

Non sono stati formulati particolari requisiti in termini di hardware impiegato.

### 3.4.2. Limitazioni software

Per specifiche esigenze di consegna sarà necessario impiegare:

- Java EE come piattaforma di sviluppo;
- Enterprise JavaBeans come architettura client-server;
- MySQL come gestore della base di dati.

## **3.5. Proprietà del software**

Il software realizzato dovrà presentare le seguenti caratteristiche.

### **3.5.1. Affidabilità**

Al fine di massimizzare l'affidabilità dell'applicazione saranno prese alcune precauzioni in fase di sviluppo, quali il riutilizzo nei limiti del possibile di librerie standard di comprovata affidabilità e la sottoposizione dell'applicazione ad una fase di testing allo scopo di intercettare le possibili cause di *failure* prima del rilascio. Non sono tuttavia prevedibili i valori di metriche di affidabilità quali *Failure Rate* o *Mean Time to Failure* in stadi così primitivi di progettazione.

### **3.5.2. Disponibilità**

Si prevede che l'applicazione debba risultare virtualmente disponibile 24 ore su 24 per 365 giorni l'anno. In considerazione tuttavia della scarsa criticità delle funzioni rivestite si ritiene possa essere ammissibile garantire una disponibilità a due 9 (99%).

### **3.5.3. Sicurezza**

È previsto un sistema di autenticazione che regolamenti l'accesso ai dati confidenziali durante la sessione. Si considera inoltre auspicabile l'impiego di adeguati meccanismi di cifratura delle password. Per la sicurezza dei dati stoccati occorre tuttavia affidarsi alla tecnologia sottostante. Analogamente, non essendo prevista l'implementazione del sistema di pagamento, non sono previste misure di sicurezza in tal senso.

### **3.5.4. Manutenibilità**

Per migliorare la manutenibilità del codice si prevede di seguire pratiche di modularizzazione delle funzionalità e di eliminazione delle ambiguità, e di fornire una documentazione esaustiva di quanto prodotto. Al fine di facilitare la comprensione dell'elaborato anche ad eventuali futuri programmatori di lingua diversa si ritiene auspicabile l'accostamento di commenti in inglese e una nomenclatura delle classi e delle tabelle della base di dati nel medesimo idioma.

Per la manutenibilità della base di dati si rimanda invece alle *best practice* del settore (quali periodici controlli di integrità, back-up dei dati ecc...).

### **3.5.5. Portabilità**

La portabilità dell'applicazione sarà limitata alla scelta del linguaggio di sviluppo. Nella fattispecie dunque l'applicazione risulterà utilizzabile da qualsiasi sistema supporti Java.

### **3.5.6. Accuratezza**

Non saranno implementati meccanismi di verifica della correttezza dei dati in termini di corrispondenza col mondo reale, in quanto questa esuli dal controllo dell'applicazione e risulti ad appannaggio degli utenti. Sarà invece cura dell'applicazione verificare che i dati inseriti nella base di dati siano completi e consistenti in base a criteri di prossimità spaziale e temporale.

### **3.5.7. Scalabilità**

La capacità del software di incrementare le proprie prestazioni al crescere delle risorse hardware dedicate sarà garantita dalla scelta della struttura architetturale, quale nella fattispecie Enterprise JavaBeans.

### **3.5.8. Robustezza**

L'applicazione sviluppata dovrà essere in grado di fronteggiare eventuali input errati degli utenti senza che ciò comporti una sospensione del servizio.

## **3.6. Altri requisiti**

L'applicazione necessiterà dell'impiego di un database relazionale per lo stoccaggio dei dati utilizzati. In base alle linee guida per lo svolgimento del progetto fornite si prevede di optare per la scelta di un database MySQL.

# Appendice

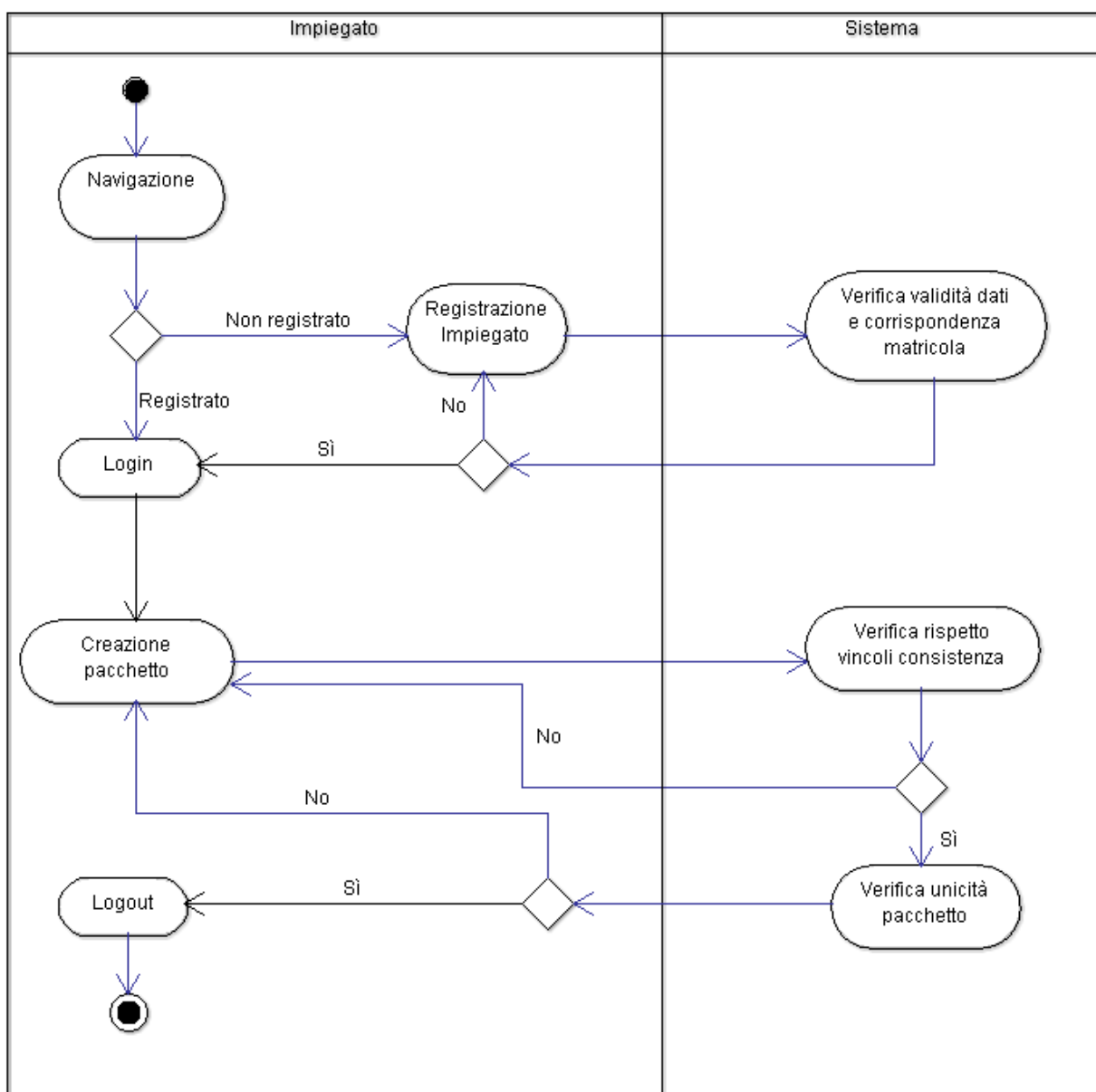
## 4.1. Modelli di analisi

### 4.1.1. Diagrammi di Attività

Nel seguito sono rappresentate quelle che si ritengono essere le più significative attività che gli utenti necessitano di compiere nell'utilizzo dell'applicazione.

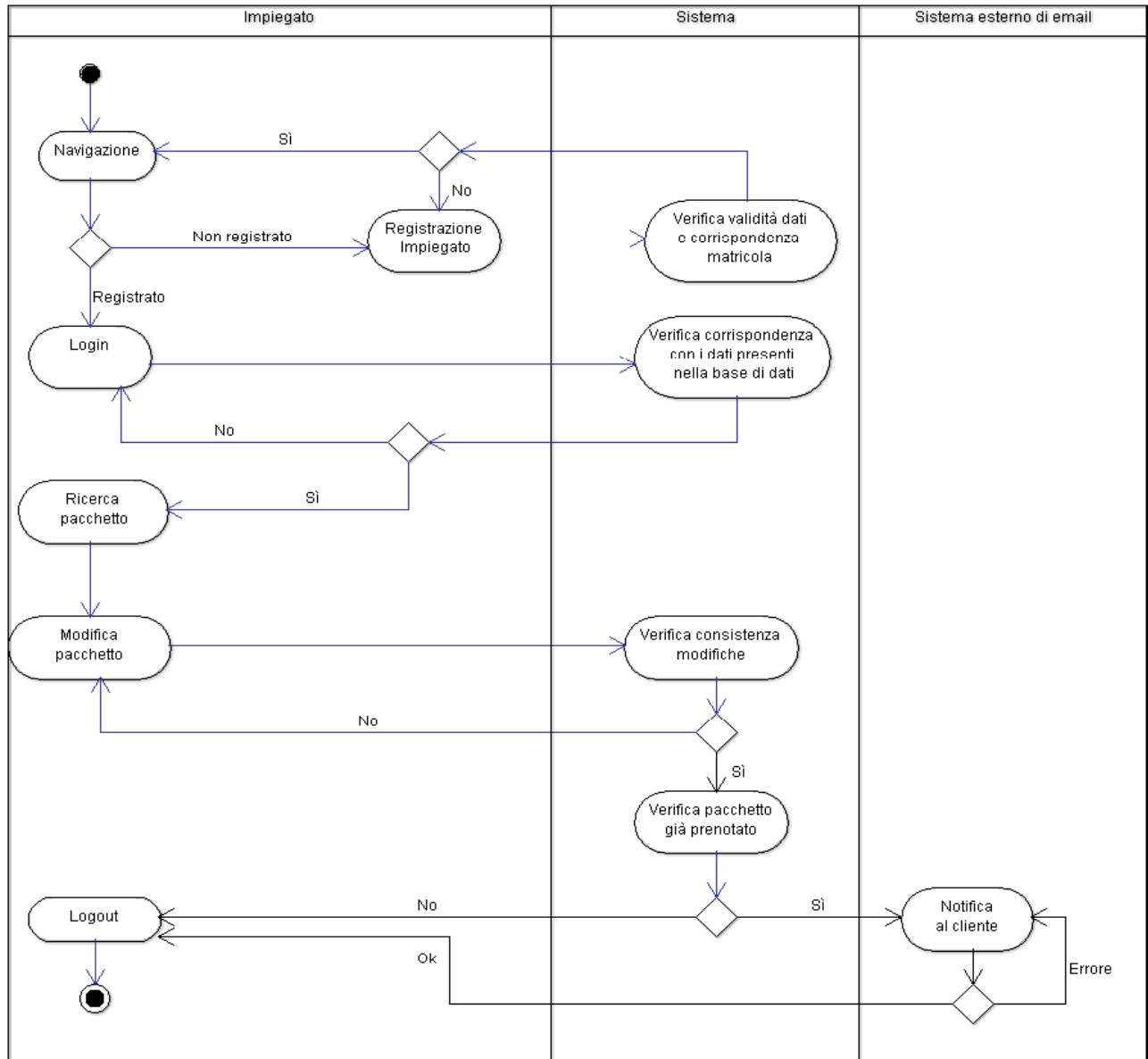
#### 4.1.1.1. Impiegato crea pacchetto

Segue un possibile svolgersi di eventi relativamente alla creazione di un nuovo pacchetto.



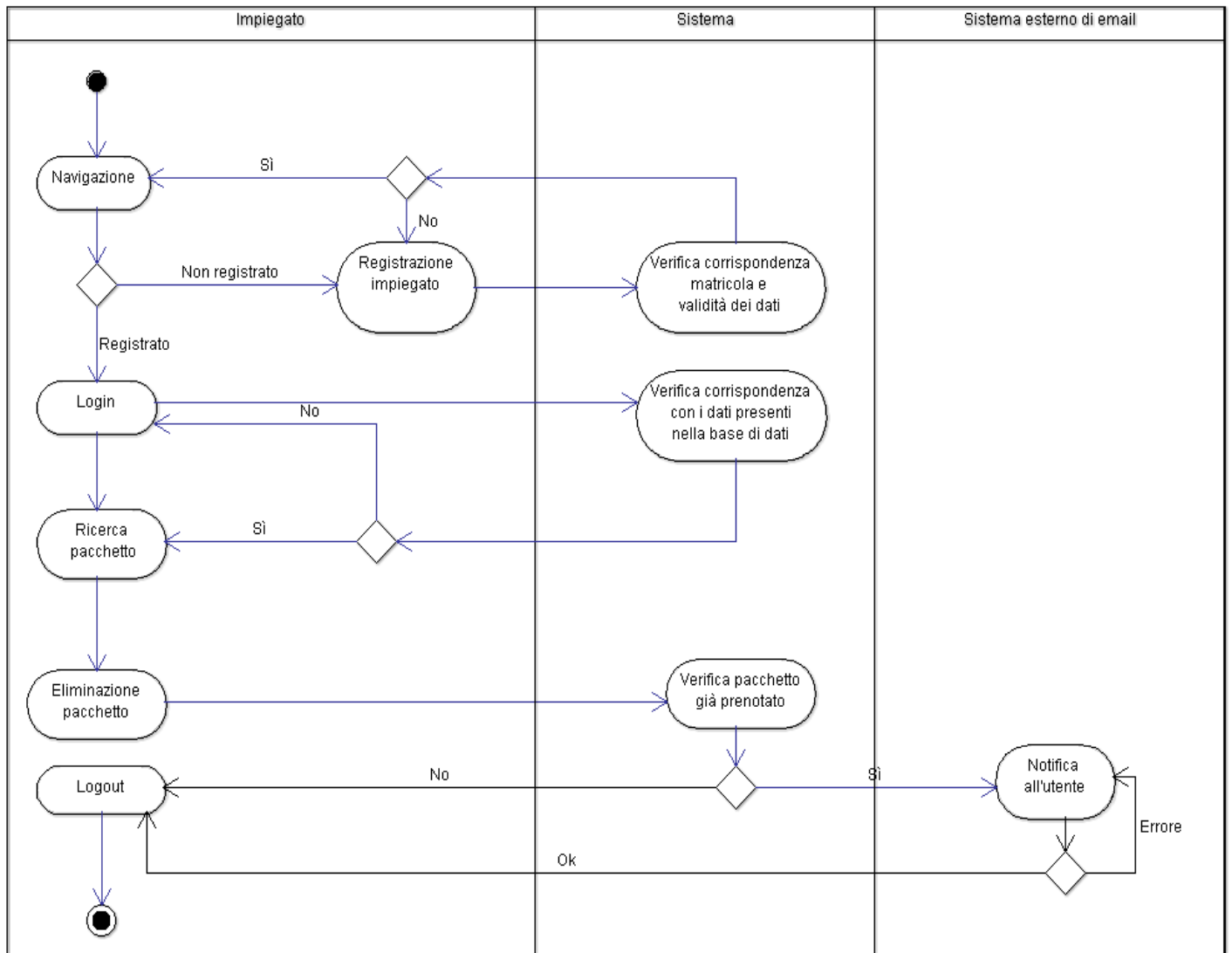
#### 4.1.1.2. Impiegato modifica pacchetto

Segue un possibile svolgersi di eventi relativamente alla modifica di un pacchetto esistente.



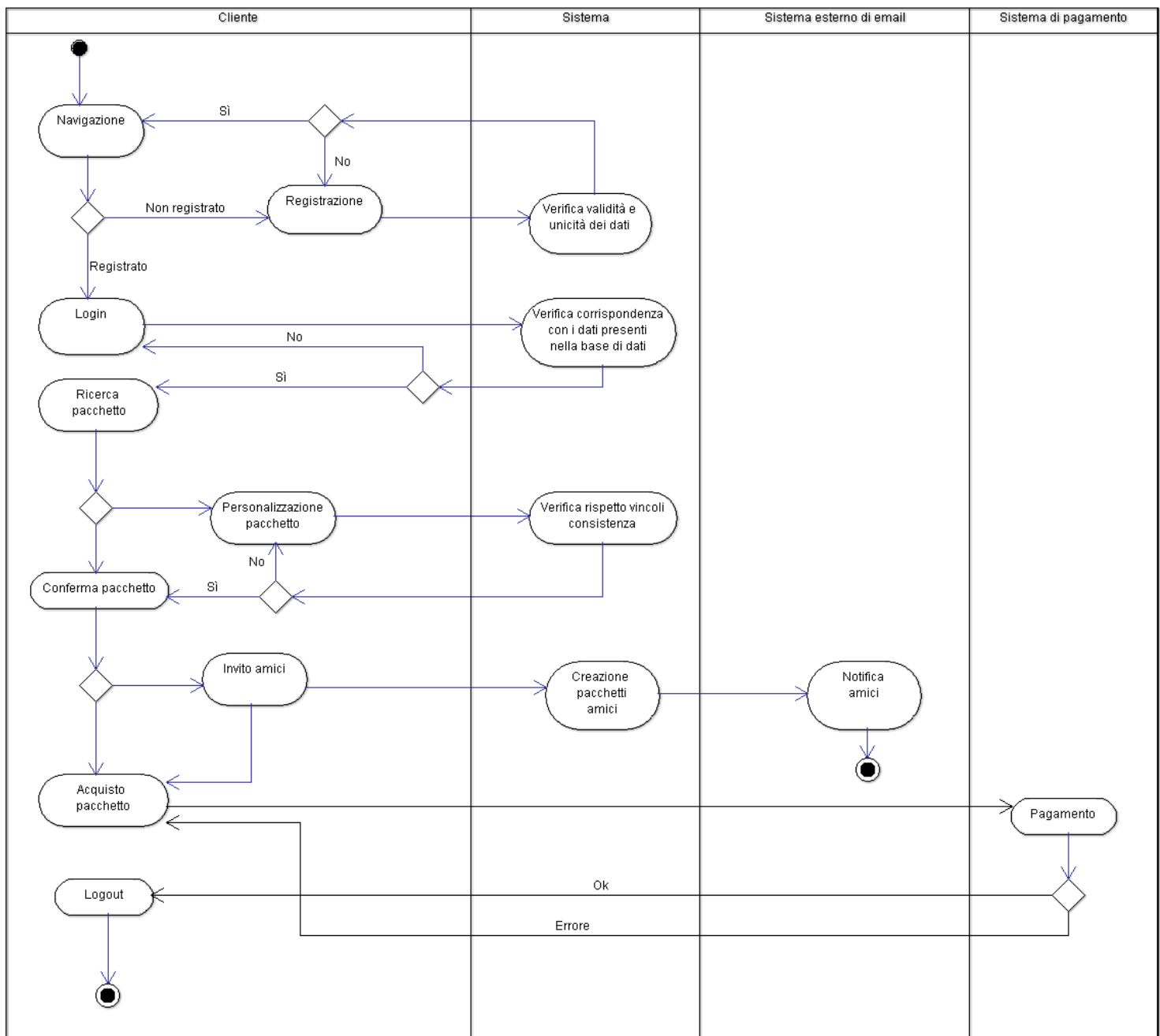
#### 4.1.1.3. Impiegato elimina pacchetto

Segue un possibile svolgersi di eventi relativamente all'eliminazione di un pacchetto esistente.



#### 4.1.1.4. Cliente invita amici

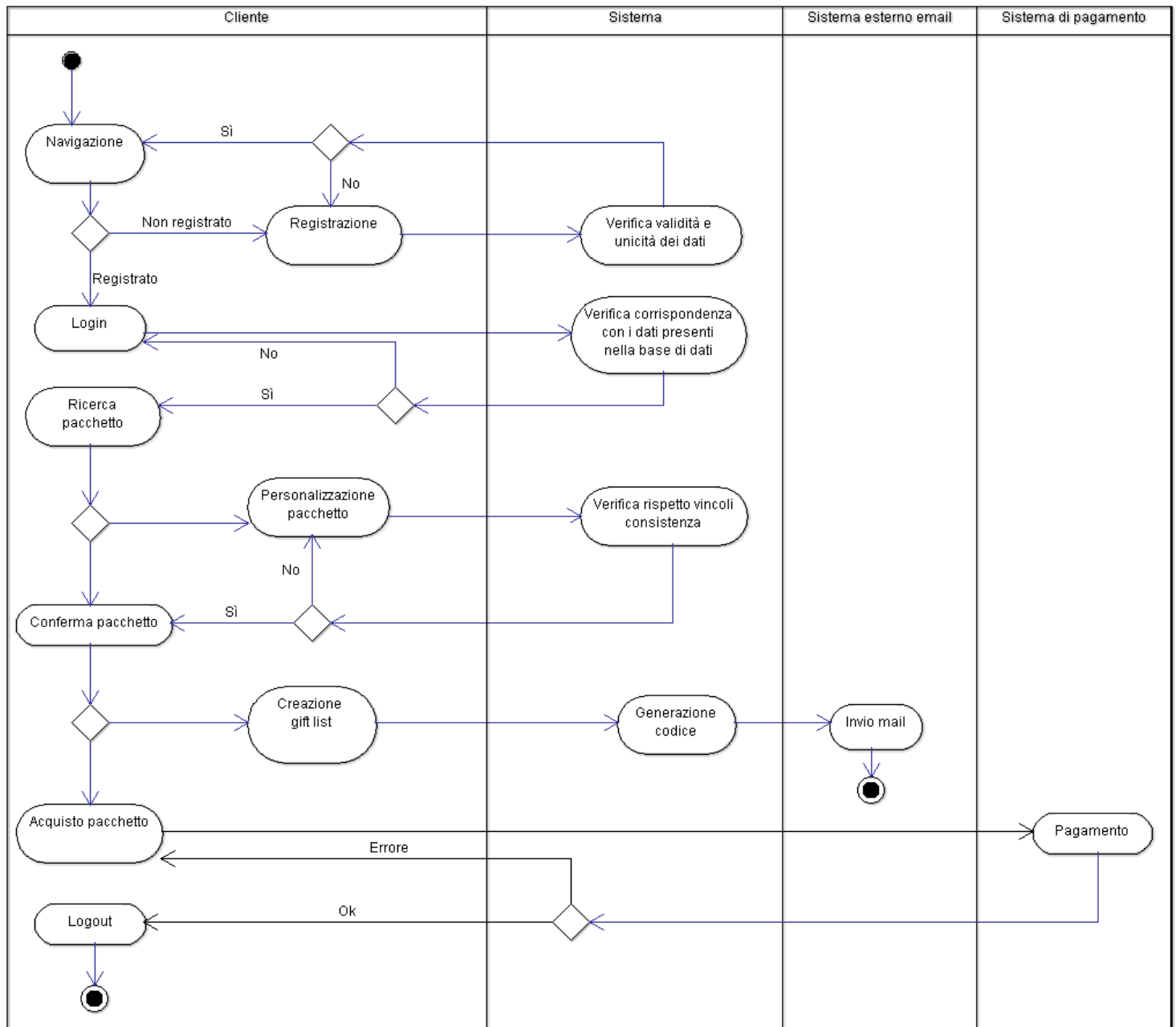
Segue un possibile svolgersi di eventi relativamente all'invito a viaggiare insieme a lui da parte di un cliente nei confronti di un amico.





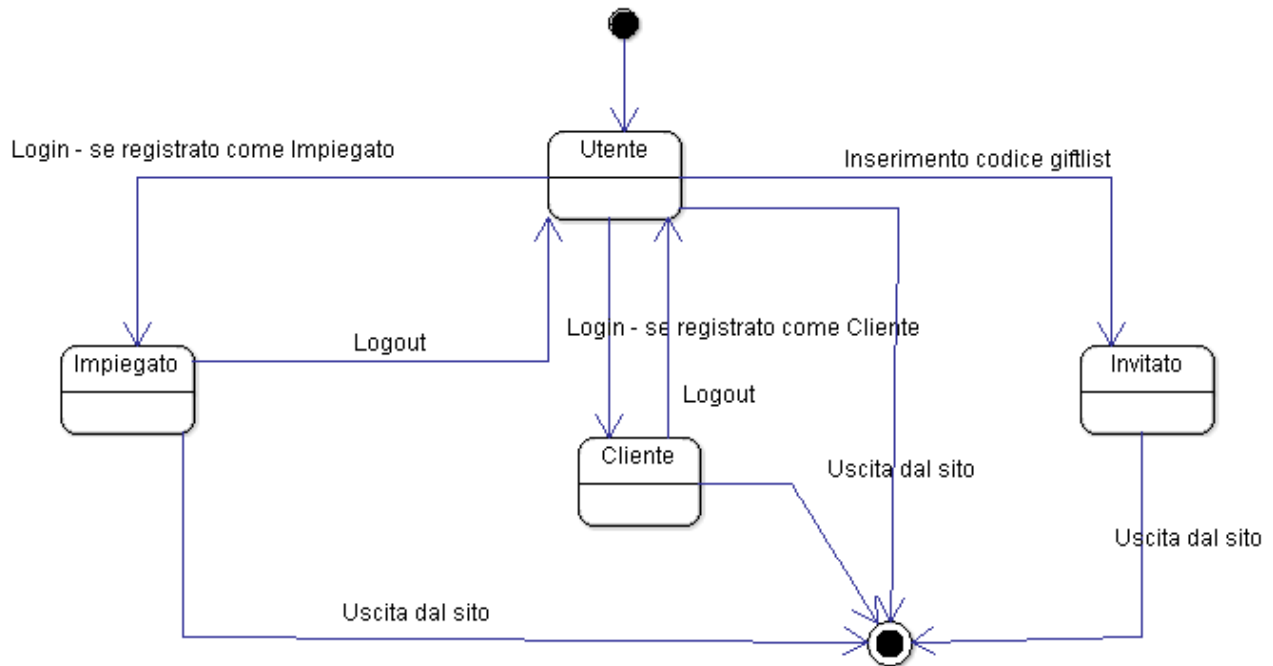
#### 4.1.1.5. Cliente crea giftlist

Segue un possibile svolgersi di eventi relativamente alla creazione di una gift list da parte di un cliente.



### 4.1.2. Diagramma di Stato

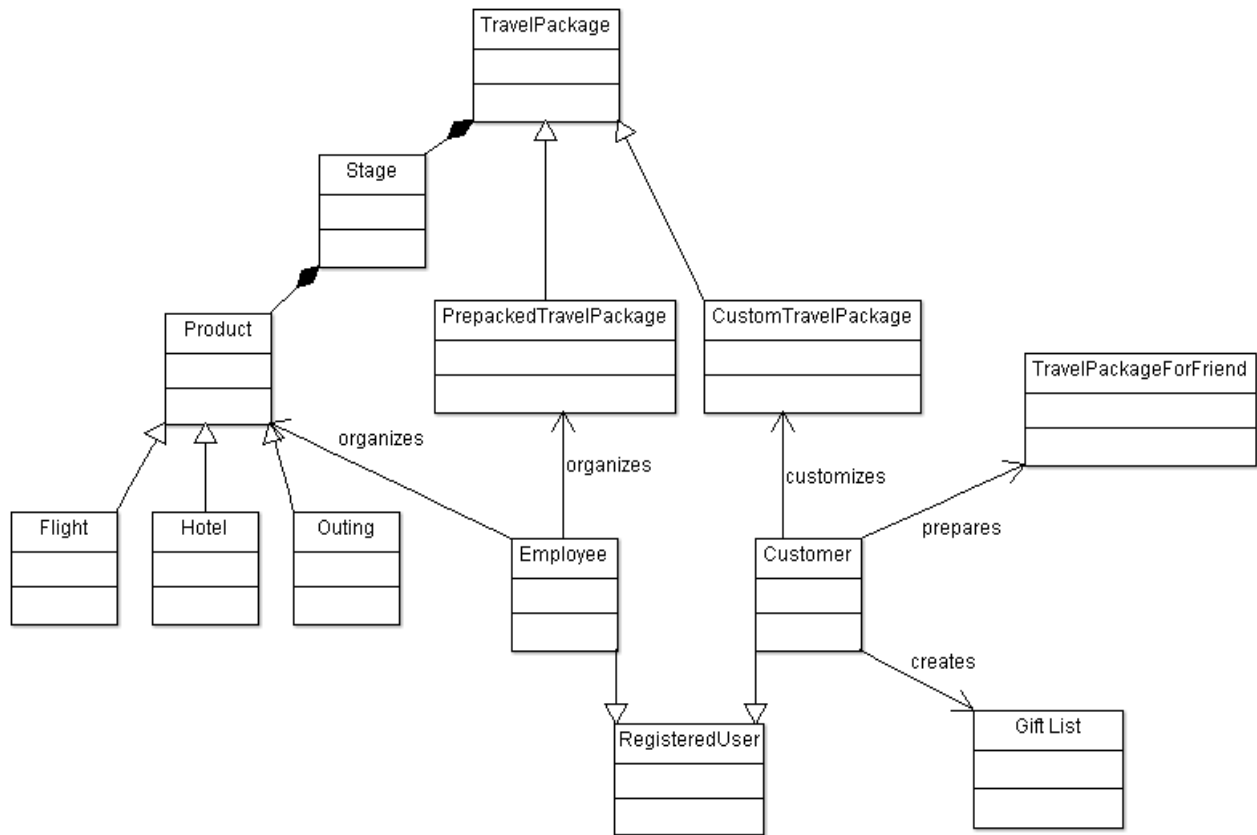
L'evoluzione dell'utente in base alle sue caratteristiche durante la sua permanenza sul sito può essere modellizzata come nel seguente diagramma.



- Diagramma di Stato -

### 4.1.3. Diagramma delle Classi

Una prima stima delle classi che si intende sviluppare potrebbe essere la seguente:



- Diagramma delle Classi -

## 4.1.4. Alloy

La modellazione in Alloy dell'applicazione è allegata al presente documento. Segue una panoramica dei risultati ottenuti.

### 4.1.4.1. Modello sistema in generale

Sono stati effettuati dei test relativi a:

- Corrispondenza login-logout:

Frammento di codice relativo:

```
assert loginlogoutCorrect
{
  all u1,u2,u3 :UtenteRegistrato , ut:Utente | u1.state=offline && login[u1,u2,ut]and logout [u2,u3,ut] implies u1.utente=u3.utente
}
check loginlogoutCorrect for 4 Utente, 2 Invitato ,4 Cliente , 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt,6 Attributi
```

Risultato:

```
Executing "Check loginlogoutCorrect for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi"
Sig this/Utente scope <= 4
Sig this/Invitato scope <= 2
Sig this/Cliente scope <= 4
Sig this/Password scope <= 4
Sig this/Connessione scope <= 6
Sig this/Username scope <= 4
Sig this/CodiceInvito scope <= 2
Sig this/CodiceAmico scope <= 2
Sig this/Impiegato scope <= 2
Sig this/posInt scope <= 6
Sig this/Attributi scope <= 6
Sig this/online scope <= 1
Sig this/offline scope <= 1
Sig ordering/Ord scope <= 1
Sig this/UtenteRegistrato scope <= 6
Sig this/Stato scope <= 2
Sig this/Stato forced to have exactly 2 atoms.
Field ordering/Ord.First == [[ordering/Ord$0, online$0]]
Field ordering/Ord.Next == [[ordering/Ord$0, online$0, offline$0]]
Sig this/Connessione in [[Connessione$0], [Connessione$1], [Connessione$2], [Connessione$3], [Connessione$4], [Connessione$5]]
Sig this/Username in [[Username$0], [Username$1], [Username$2], [Username$3]]
Sig this/Password in [[Password$0], [Password$1], [Password$2], [Password$3]]
Sig this/Attributi in [[Attributi$0], [Attributi$1], [Attributi$2], [Attributi$3], [Attributi$4], [Attributi$5]]
Sig this/CodiceInvito in [[CodiceInvito$0], [CodiceInvito$1]]
Sig this/CodiceAmico in [[CodiceAmico$0], [CodiceAmico$1]]
Sig this/posInt in [[posInt$0], [posInt$1], [posInt$2], [posInt$3], [posInt$4], [posInt$5]]
Sig this/Utente in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]]
Sig this/Invitato in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]] with size<=2
Sig this/UtenteRegistrato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]]
Sig this/Stato == [[online$0], [offline$0]]
Sig this/online == [[online$0]]
Sig this/offline == [[offline$0]]
Sig this/Impiegato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=2
Sig this/Cliente in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=4
Sig ordering/Ord == [[ordering/Ord$0]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
6043 vars. 536 primary vars. 12041 clauses. 62ms.
No counterexample found. Assertion may be valid. 16ms.
```

- Unicità della connessione:

Frammento di codice relativo:

```
assert noSharedConnections{
all disj u1,u2:Utente | u1.connessione!=u2.connessione
}
check noSharedConnections for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi
```

Risultato:

```
Executing "Check noSharedConnections for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi"
Sig this/Utente scope <= 4
Sig this/Invitato scope <= 2
Sig this/Cliente scope <= 4
Sig this/Password scope <= 4
Sig this/Connessione scope <= 6
Sig this/Username scope <= 4
Sig this/CodiceInvito scope <= 2
Sig this/CodiceAmico scope <= 2
Sig this/Impiegato scope <= 2
Sig this/posInt scope <= 6
Sig this/Attributi scope <= 6
Sig this/online scope <= 1
Sig this/offline scope <= 1
Sig ordering/Ord scope <= 1
Sig this/UtenteRegistrato scope <= 6
Sig this/Stato scope <= 2
Sig this/Stato forced to have exactly 2 atoms.
Field ordering/Ord.First == [[ordering/Ord$0, online$0]]
Field ordering/Ord.Next == [[ordering/Ord$0, online$0, offline$0]]
Sig this/Connessione in [[Connessione$0], [Connessione$1], [Connessione$2], [Connessione$3], [Connessione$4], [Connessione$5]]
Sig this/Username in [[Username$0], [Username$1], [Username$2], [Username$3]]
Sig this/Password in [[Password$0], [Password$1], [Password$2], [Password$3]]
Sig this/Attributi in [[Attributi$0], [Attributi$1], [Attributi$2], [Attributi$3], [Attributi$4], [Attributi$5]]
Sig this/CodiceInvito in [[CodiceInvito$0], [CodiceInvito$1]]
Sig this/CodiceAmico in [[CodiceAmico$0], [CodiceAmico$1]]
Sig this/posInt in [[posInt$0], [posInt$1], [posInt$2], [posInt$3], [posInt$4], [posInt$5]]
Sig this/Utente in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]]
Sig this/Invitato in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]] with size<=2
Sig this/UtenteRegistrato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]]
Sig this/Stato == [[online$0], [offline$0]]
Sig this/online == [[online$0]]
Sig this/offline == [[offline$0]]
Sig this/Impiegato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=2
Sig this/Cliente in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=4
Sig ordering/Ord == [[ordering/Ord$0]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
5859 vars. 522 primary vars. 11599 clauses. 47ms.
No counterexample found. Assertion may be valid. 0ms.
```

- Unicità dello username:

Frammento di codice relativo:

```
assert noSharedUsername
{
  no disj u1,u2 :UtenteRegistrato | u1.username=u2.username
}
check noSharedUsername for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi
```

Risultato:

```
Executing "Check noSharedUsername for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi"
Sig this/Utente scope <= 4
Sig this/Invitato scope <= 2
Sig this/Cliente scope <= 4
Sig this/Password scope <= 4
Sig this/Connessione scope <= 6
Sig this/Username scope <= 4
Sig this/CodiceInvito scope <= 2
Sig this/CodiceAmico scope <= 2
Sig this/Impiegato scope <= 2
Sig this/posInt scope <= 6
Sig this/Attributi scope <= 6
Sig this/online scope <= 1
Sig this/offline scope <= 1
Sig ordering/Ord scope <= 1
Sig this/UtenteRegistrato scope <= 6
Sig this/Stato scope <= 2
Sig this/Stato forced to have exactly 2 atoms.
Field ordering/Ord.First == [[ordering/Ord$0, online$0]]
Field ordering/Ord.Next == [[ordering/Ord$0, online$0, offline$0]]
Sig this/Connessione in [[Connessione$0], [Connessione$1], [Connessione$2], [Connessione$3], [Connessione$4], [Connessione$5]]
Sig this/Username in [[Username$0], [Username$1], [Username$2], [Username$3]]
Sig this/Password in [[Password$0], [Password$1], [Password$2], [Password$3]]
Sig this/Attributi in [[Attributi$0], [Attributi$1], [Attributi$2], [Attributi$3], [Attributi$4], [Attributi$5]]
Sig this/CodiceInvito in [[CodiceInvito$0], [CodiceInvito$1]]
Sig this/CodiceAmico in [[CodiceAmico$0], [CodiceAmico$1]]
Sig this/posInt in [[posInt$0], [posInt$1], [posInt$2], [posInt$3], [posInt$4], [posInt$5]]
Sig this/Utente in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]]
Sig this/Invitato in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]] with size<=2
Sig this/UtenteRegistrato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]]
Sig this/Stato == [[online$0], [offline$0]]
Sig this/online == [[online$0]]
Sig this/offline == [[offline$0]]
Sig this/Impiegato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=2
Sig this/Cliente in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=4
Sig ordering/Ord == [[ordering/Ord$0]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
5885 vars. 526 primary vars. 11647 clauses. 47ms.
No counterexample found. Assertion may be valid. 15ms.
```

- Unicità della matricola impiegato:

Frammento di codice relativo:

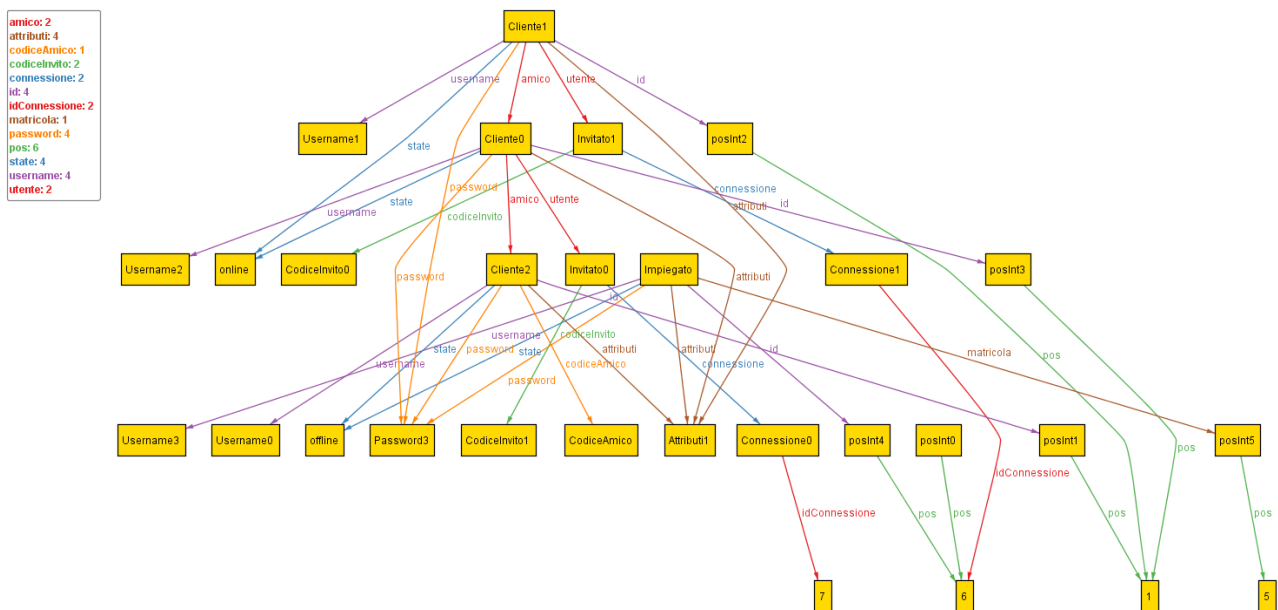
```
assert noSharedMatricola
{
  no disj u1,u2 :Impiegato | u1.matricola=u2.matricola
}
```

check noSharedMatricola for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi

Risultato:

```
Executing "Check noSharedMatricola for 4 Utente, 2 Invitato, 4 Cliente, 4 Password, 6 Connessione, 4 Username, 2 CodiceInvito, 2 CodiceAmico, 2 Impiegato, 6 posInt, 6 Attributi"
Sig this/Utente scope <= 4
Sig this/Invitato scope <= 2
Sig this/Cliente scope <= 4
Sig this/Password scope <= 4
Sig this/Connessione scope <= 6
Sig this/Username scope <= 4
Sig this/CodiceInvito scope <= 2
Sig this/CodiceAmico scope <= 2
Sig this/Impiegato scope <= 2
Sig this/posInt scope <= 6
Sig this/Attributi scope <= 6
Sig this/online scope <= 1
Sig this/offline scope <= 1
Sig ordering/Ord scope <= 1
Sig this/UtenteRegistrato scope <= 6
Sig this/Stato scope <= 2
Sig this/Stato forced to have exactly 2 atoms.
Field ordering/Ord.First == [[ordering/Ord$0, online$0]]
Field ordering/Ord.Next == [[ordering/Ord$0, online$0, offline$0]]
Sig this/Connessione in [[Connessione$0], [Connessione$1], [Connessione$2], [Connessione$3], [Connessione$4], [Connessione$5]]
Sig this/Username in [[Username$0], [Username$1], [Username$2], [Username$3]]
Sig this/Password in [[Password$0], [Password$1], [Password$2], [Password$3]]
Sig this/Attributi in [[Attributi$0], [Attributi$1], [Attributi$2], [Attributi$3], [Attributi$4], [Attributi$5]]
Sig this/CodiceInvito in [[CodiceInvito$0], [CodiceInvito$1]]
Sig this/CodiceAmico in [[CodiceAmico$0], [CodiceAmico$1]]
Sig this/posInt in [[posInt$0], [posInt$1], [posInt$2], [posInt$3], [posInt$4], [posInt$5]]
Sig this/Utente in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]]
Sig this/Invitato in [[Utente$0], [Utente$1], [Utente$2], [Utente$3]] with size<=2
Sig this/UtenteRegistrato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]]
Sig this/Stato == [[online$0], [offline$0]]
Sig this/online == [[online$0]]
Sig this/offline == [[offline$0]]
Sig this/Impiegato in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=2
Sig this/Cliente in [[UtenteRegistrato$0], [UtenteRegistrato$1], [UtenteRegistrato$2], [UtenteRegistrato$3], [UtenteRegistrato$4], [UtenteRegistrato$5]] with size<=4
Sig ordering/Ord == [[ordering/Ord$0]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
5917 vars. 526 primary vars. 11755 clauses. 47ms.
No counterexample found. Assertion may be valid. 15ms.
```

Mentre una rappresentazione di una possibile istanza del sistema è la seguente:



- Rappresentazione del sistema -

#### 4.1.4.2. Modello controllo consistenza

Con maggiore specificità sono stati effettuati dei test relativi al controllo di consistenza dei pacchetti, in particolare riguardo a:

- Non sovrapposizione delle escursioni:

Frammento di codice relativo:

```
assert noEscursionOverlapping
{
  no disj e1,e2 :Escursione | (e1.dataA.data<e2.dataB.data &&e2.dataA.data<e1.dataB.data) || (e2.dataA.data<e1.dataB.data &&e1.dataA.data<e2.dataB.data)
}
check noEscursionOverlapping for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo|
```

Risultato:

```
Executing "Check noEscursionOverlapping for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo"
Sig this/Viaggio scope <= 1
Sig this/Tappa scope <= 2
Sig this/Volo scope <= 4
Sig this/Data scope <= 16
Sig this/Escursione scope <= 4
Sig this/Hotel scope <= 2
Sig this/Luogo scope <= 2
Sig this/Viaggio == [[Viaggio$0]]
Sig this/Data == [[Data$0], [Data$1], [Data$2], [Data$3], [Data$4], [Data$5], [Data$6], [Data$7], [Data$8], [Data$9], [Data$10], [Data$11], [Data$12], [Data$13], [Data$14], [Data$15]]
Sig this/Volo == [[Volo$0], [Volo$1], [Volo$2], [Volo$3]]
Sig this/Hotel == [[Hotel$0], [Hotel$1]]
Sig this/Escursione == [[Escursione$0], [Escursione$1], [Escursione$2], [Escursione$3]]
Sig this/Luogo in [[Luogo$0], [Luogo$1]]
Sig this/Tappa == [[Tappa$0], [Tappa$1]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
38062 vars. 648 primary vars. 71990 clauses. 2824ms.
No counterexample found. Assertion may be valid. 100979ms.
```

- Non sovrapposizione degli alberghi:

Frammento di codice relativo:

```
assert noHotelOverlapping
{
  no disj h1,h2:Hotel | (h1.dataA.data<h2.dataB.data &&h1.dataB.data>h2.dataA.data) || (h2.dataA.data<h1.dataB.data && h2.dataB.data>h1.dataA.data)
}
check noHotelOverlapping for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo
```

Risultato:

```
Executing "Check noHotelOverlapping for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo"
Sig this/Viaggio scope <= 1
Sig this/Tappa scope <= 2
Sig this/Volo scope <= 4
Sig this/Data scope <= 16
Sig this/Escursione scope <= 4
Sig this/Hotel scope <= 2
Sig this/Luogo scope <= 2
Sig this/Viaggio == [[Viaggio$0]]
Sig this/Data == [[Data$0], [Data$1], [Data$2], [Data$3], [Data$4], [Data$5], [Data$6], [Data$7], [Data$8], [Data$9], [Data$10], [Data$11], [Data$12], [Data$13], [Data$14], [Data$15]]
Sig this/Volo == [[Volo$0], [Volo$1], [Volo$2], [Volo$3]]
Sig this/Hotel == [[Hotel$0], [Hotel$1]]
Sig this/Escursione == [[Escursione$0], [Escursione$1], [Escursione$2], [Escursione$3]]
Sig this/Luogo in [[Luogo$0], [Luogo$1]]
Sig this/Tappa == [[Tappa$0], [Tappa$1]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
37904 vars. 644 primary vars. 71426 clauses. 2792ms.
No counterexample found. Assertion may be valid. 121103ms.
```



- Non sovrapposizione dei voli:

Frammento di codice relativo:

```
assert noFlightOverlapping
{
no disj v1,v2:Volo | (v1.dataA.data<v2.dataB.data &&v1.dataB.data>v2.dataA.data) || (v2.dataA.data<v1.dataB.data && v2.dataB.data>v1.dataA.data)
}
check noFlightOverlapping for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo
```

Risultato:

```
Executing "Check noFlightOverlapping for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo"
Sig this/Viaggio scope <= 1
Sig this/Tappa scope <= 2
Sig this/Volo scope <= 4
Sig this/Data scope <= 16
Sig this/Escursione scope <= 4
Sig this/Hotel scope <= 2
Sig this/Luogo scope <= 2
Sig this/Viaggio == [[Viaggio$0]]
Sig this/Data == [[Data$0], [Data$1], [Data$2], [Data$3], [Data$4], [Data$5], [Data$6], [Data$7], [Data$8], [Data$9], [Data$10], [Data$11], [Data$12], [Data$13], [Data$14], [Data$15]]
Sig this/Volo == [[Volo$0], [Volo$1], [Volo$2], [Volo$3]]
Sig this/Hotel == [[Hotel$0], [Hotel$1]]
Sig this/Escursione == [[Escursione$0], [Escursione$1], [Escursione$2], [Escursione$3]]
Sig this/Luogo in [[Luogo$0], [Luogo$1]]
Sig this/Tappa == [[Tappa$0], [Tappa$1]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
38062 vars. 648 primary vars. 71990 clauses. 2808ms.
No counterexample found. Assertion may be valid. 89934ms.
```

- Presenza di almeno un prodotto per tappa:

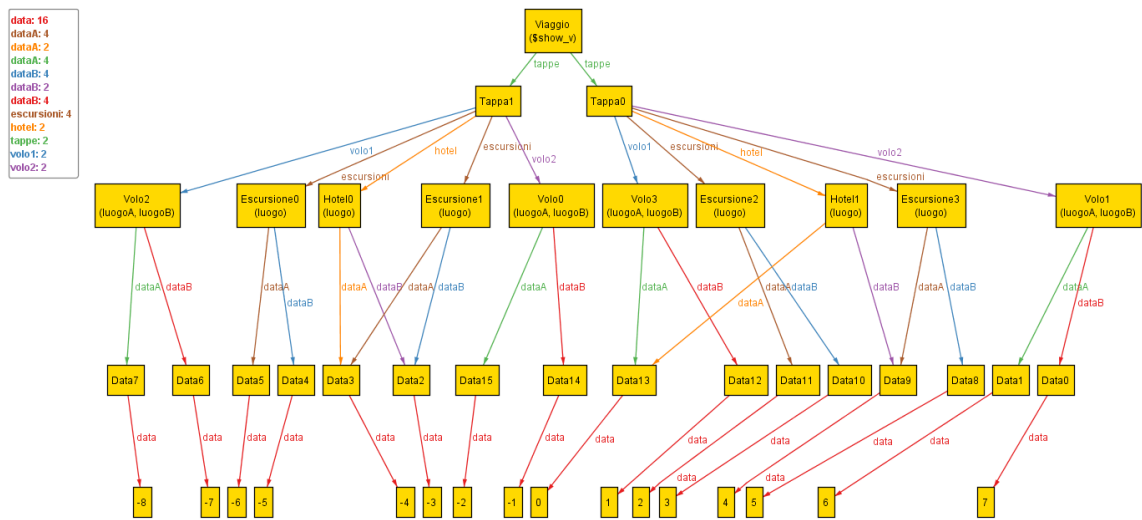
Frammento di codice relativo:

```
assert atLeastOne
{
no t:Tappa | #t.volo1=0 && #t.volo2=0 && #t.escursioni=0 && #t.hotel=0
}
check atLeastOne for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo
```

Risultato:

```
Executing "Check atLeastOne for exactly 1 Viaggio, exactly 2 Tappa, exactly 4 Volo, exactly 16 Data, exactly 4 Escursione, exactly 2 Hotel, 2 Luogo"
Sig this/Viaggio scope <= 1
Sig this/Tappa scope <= 2
Sig this/Volo scope <= 4
Sig this/Data scope <= 16
Sig this/Escursione scope <= 4
Sig this/Hotel scope <= 2
Sig this/Luogo scope <= 2
Sig this/Viaggio == [[Viaggio$0]]
Sig this/Data == [[Data$0], [Data$1], [Data$2], [Data$3], [Data$4], [Data$5], [Data$6], [Data$7], [Data$8], [Data$9], [Data$10], [Data$11], [Data$12], [Data$13], [Data$14], [Data$15]]
Sig this/Volo == [[Volo$0], [Volo$1], [Volo$2], [Volo$3]]
Sig this/Hotel == [[Hotel$0], [Hotel$1]]
Sig this/Escursione == [[Escursione$0], [Escursione$1], [Escursione$2], [Escursione$3]]
Sig this/Luogo in [[Luogo$0], [Luogo$1]]
Sig this/Tappa == [[Tappa$0], [Tappa$1]]
Solver=sat4j Bitwidth=4 MaxSeq=4 SkolemDepth=1 Symmetry=20
36123 vars. 642 primary vars. 65373 clauses. 2839ms.
No counterexample found. Assertion may be valid. 4586ms.
```

Mentre una rappresentazione di una possibile istanza di pacchetto è la seguente:



- Rappresentazione pacchetto -

### 4.2. Modifiche in itinere

L'elaborazione del documento di specifica dei requisiti ha subito la seguente evoluzione:

Data	Evento
28/11/13	Prima versione del documento
25/02/14	Adattamenti dei casi d'uso

### 4.3. Tempi di elaborazione

Per l'elaborazione del presente documento è stata impiegata la seguente quantità di tempo:

Membro del gruppo	Ore di attività
Selenia Vincenza Rusalen	44
Marcello Felappi	28
Giovanni Battista Conserva	10