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

Progetto di Ingegneria del Software 2

TravelDream

Parte I: RASD

(Requirement Analysis and Specification Document)

Responsabile:
Prof. Raffaella Mirandola

Progetto di:
Piazza Enrico Matricola n.760240
Polvara Riccardo Matricola n.817572
Zamperetti Niccolò Matricola n.814656

ANNO ACCADEMICO 2013-2014

Indice

Capitolo 1

Introduzione

L'obiettivo di questo documento è fornire una specifica dei requisiti che dovranno essere soddisfatti dal sistema realizzato. Il documento presenta una descrizione dei requisiti funzionali e non-funzionali, impiegando vari diagrammi UML, oltre a una specifica del sistema ad alto livello. Nell'ultima parte è invece presentato un modello formale delle specifiche e i risultati della sua analisi, realizzata mediante Alloy. A chi è rivolto: Questo documento è rivolto sia agli sviluppatori del progetto TravelDream che a tutti gli altri stakeholder del progetto. Per quanto riguarda i primi, il documento assume importanza nel momento in cui è necessario stabilire una coerenza tra lo sviluppo e le richieste elaborate. Per gli stakeholder, questo documento rappresenta una traccia del lavoro di analisi svolto dagli sviluppatori, utile per interpretarne il lavoro di sviluppo.

1.1 Obiettivo

Il sistema che verrà realizzato è un applicativo software denominato TravelDream. L'obiettivo del prodotto è proporre agli utenti offerte riguardo pacchetti compositi di voli, hotel ed escursioni. Il sistema permetterà inoltre agli utenti di creare dei pacchetti personalizzati partendo da quelli già proposti e interagire con degli amici, invitandoli a partecipare a un proprio viaggio o inviando loro una gift list di componenti di proprio interesse.

1.2 Descrizione del problema

Gli impiegati della TravelDream possono modificare l'insieme di offerte proposte ai clienti. In particolare, possono aggiungere, rimuovere o modificare singoli voli, hotel o escursioni, oppure creare dei pacchetti predefiniti da proporre agli utenti. Gli utenti registrati alla piattaforma possono sfogliare tutte le offerte presenti e aggiungere uno o più pacchetti al loro carrello. Da quest'ultimo sono possibili diverse azioni: confermare l'acquisto dell'oggetto aggiunto, modificarlo, rimuoverlo, invitare un amico (registrato o meno) a prendervi parte o inviargli alcuni degli oggetti presenti come gift list. Gli utenti non registrati possono solo sfogliare le offerte, oppure (se invitati da un utente registrato) visualizzare viaggi a cui può prendere parte (previa registrazione), o acquistare alcuni componenti di una gift list inviatagli da un amico.

1.3 Glossario

Di seguito sono fornite le definizioni di alcuni termini che verranno utilizzati nel documento, al fine di prevenire eventuali ambiguità nel loro utilizzo o nella loro comprensione.

- Prodotto: Componente atomico di un viaggio (voli, hotel, escursioni).
  - Sinonimi: componente, ComponenteViaggio (Alloy).
- Pacchetto predefinito: Insieme di prodotti; il termine "predefiniti" indica che sono proposti dalla TravelDream (creati quindi da un impiegato).
- Pacchetto personalizzato: Insieme di prodotti; Il termine "personalizzato" indica che il pacchetto è associato ad un utente, p.e. perché ha subito delle variazioni o delle aggiunte rispetto ad un pacchetto predefinito.
- Carrello: Insieme di pacchetti personalizzati , anche in fase di attesa (associato ad un utente).
- Storico: Insieme di pacchetti personalizzati che sono stati acquistati (associato ad un utente).
- Gift List: Lista associata ad un utente (necessariamente registrato) di prodotti, che un amico può acquistare al suo posto come, appunto, un regalo.
- Amico: Un utente (non necessariamente registrato) a cui è stata inviata una Gift List o una richiesta di partecipazione a un viaggio.

Capitolo 2

Descrizione generale

2.1 Prospettiva del prodotto

Il sistema proposto è un'applicazione totalmente indipendente da altri sistemi e autocontenuta, pensata per essere rilasciata al pubblico generale. Il client sarà realizzato come applicazione web, a cui gli utenti possono accedere attraverso la rete Internet da un comune browser (e.g. Mozilla Firefox) inserendo l'indirizzo web associato.

2.2 Considerazioni preliminari

Nella descrizione fornita sono presenti alcuni punti poco chiari oppure ambigui. Abbiamo quindi aggiunto ipotesi aggiuntive, allo scopo di risolvere le ambiguità presenti ed esplicitare i requisiti lasciati sottointesi.

- Pagamento di un oggetto in Gift List: giacchè non è specificato se l'azione richieda che il destinatario sia registrato, abbiamo ipotizzato che si possa procedere all'acquisto anche non essendo registrati alla piattaforma (immaginando sia possibile selezionare ciò che si desidera pagare, per poi passare direttamente a una pagina in cui vadano inseriti i dati della carta di credito)

- Carrello: tutti i pacchetti (personalizzati), prima di confermarne l'acquisto, passano attraverso un carrello; tutte le azioni possibili su un oggetto (acquisto/rimozione/modifica/invito partecipazione/invio Gift List) sono effettuabili dal carrello stesso.

- Visibilità dei prodotti: dalla specifica si evince che l'utente possa sfogliare/ricercare solamente pacchetti, quindi si è ipotizzato che non potesse cercare e/o acquistare singoli componenti. Inoltre, sempre per lo stesso motivo, si è ipotizzato che la personalizzazione di un pacchetto consista nella sola sostituzione di prodotti con altri equivalenti (p.e. sostituire un Hotel con un altro più/meno caro, sostituire un volo con un altro in una data/ora diversa etc.). Se infatti all'utente fosse data la possibilità di aggiungere o rimuovere alcuni elementi di un pacchetto, si potrebbe arrivare alla riduzione di un pacchetto a un singolo componente, rendendo quindi il vincolo di non potere sfogliare/ricercare prodotti del tutto inutile e incoerente. Ne deriva anche, ovviamente, che l'utente non possa creare pacchetti personalizzati da zero.

- Struttura dei pacchetti: dall'assunzione precedente deriva necessariamente che dal punto di vista di un utente il pacchetto abbia una struttura "rigida", ovvero può cambiare i prodotti che ne fanno parte, ma non può aggiungerne di nuovi o rimuovere quelli presenti. Qualora fosse quindi interessato a un pacchetto dalla diversa struttura, potrebbe scegliere un altro pacchetto che ne presenti una differente, e personalizzare quest'ultimo. Diverso discorso per gli impiegati che, potendo creare nuovi pacchetti predefiniti da zero, non sono soggetti a questi vincoli.

2.3 Caratteristiche degli utenti

Il target dell'applicazione sono utenti adulti del web che desiderano pianificare un viaggio, loro "amici" (in senso stretto) che possono essere invitati e gli impiegati dell'azienda che gestiscono i dati dell'applicazione.

2.4 Vincoli e assunzioni

Il principale vincolo risiede nell'utilizzo della piattaforma J2EE e della tecnologia degli Enterprise Java Beans (EJB) per la realizzazione dell'applicazione. Assumiamo che gli utenti del sistema abbiano a disposizione un browser web recente e una connessione a Internet. Per quanto riguarda le interfacce con altri sistemi, TravelDream dovrà avere accesso a un server SMTP per l'invio di notifiche agli utenti via posta elettronica (in particolare per poter inviare il messaggio di conferma durante la registrazione e gli inviti agli amici), e a un DBMS (database management system) per poter memorizzare i dati necessari al funzionamento. Ulteriori dettagli sulle assunzioni relative ai sistemi con cui TravelDream deve interagire sono rimandati a una fase successiva, e saranno contenuti nel Design Document.

2.5 Attori e funzionalità

Abbiamo individuato quattro diverse tipologie di attori che interagiscono con il sistema. Le tipologie sono specificate di seguito insieme a una descrizione sintetica delle funzionalità di base che il sistema deve permettere.

2.5.1 A1 Utente non registrato

Descrizione: Un utente dell'applicativo che non ha effettuato la registrazione oppure che non è stato ancora autenticato. Funzionalità:

- Iscriversi alla piattaforma (diventando utente Customer A3)
- Effettuare il login (diventando utente Customer A3 o un utente Impiegato A4)
- Visualizzare i pacchetti predefiniti

2.5.2 A2 Amico

Descrizione: Un utente che accede alla piattaforma su invito da parte di un utente registrato. Funzionalità:

- Visualizzare il pacchetto suggeritogli da un utente customer ed eventualmente parteciparvi
- Visualizzare la Gift List inviatagli da un utente customer e scegliere quali prodotti "regalargli"

2.5.3 A3 Customer

Descrizione: Un utente registrato alla piattaforma e che si è autenticato al sistema. Funzionalità:

- Visualizzare pacchetti predefiniti
- Personalizzare pacchetti
- Invitare amici a prendere parte e a visualizzare i loro pacchetti personalizzati

2.5.4 A4 Impiegato

Descrizione: L'impiegato dell'agenzia che può modificare i prodotti offerti. Funzionalità:

- Creare pacchetti predefiniti
- Creare/Modificare singoli prodotti (Voli, Hotel, Escursioni etc)

2.6 Requisiti non funzionali

Abbiamo individuato alcuni requisiti non funzionali necessari per un corretto funzionamento del sistema ed alcune qualità che il software che realizzeremo dovrà avere.

- Semplicità di utilizzo: è necessario rendere l'interfaccia utente il più possibile intuitiva: il sistema è rivolto a un pubblico eterogeneo anche per utilizzi occasionali. Per questo motivo non è possibile assumere che l'utente abbia letto il manuale d'uso.
- Portabilità: il client dev'essere compatibile con tutte le piattaforme hardware e software più diffuse, al fine di raggiungere il maggior numero possibile di utenti. Questo requisito è soddisfatto realizzando il client come applicazione web: in questo modo, le uniche assunzioni che devono essere soddisfatte dagli utenti sono la disponibilità di un browser web e di una connessione alla rete Internet.
- Prestazioni: per poter fornire un servizio adeguato, il sistema dev'essere sufficientemente reattivo e capace di rispondere a un numero, anche elevato, di richieste contemporanee: pertanto, la potenza di calcolo dei server che erogano il servizio e la larghezza di banda a disposizione devono essere dimensionate in maniera adeguata al carico. Poiché non è possibile fare assunzioni sulla velocità di connessione degli utenti (che in generale può variare molto), è necessario che il volume di traffico che deve essere scambiato tra client e server sia contenuto, e che nel caso sia necessario scambiare dati di grosse dimensioni, l'interfaccia dell'applicazione renda l'utente consapevole di quello che sta avvenendo.
- Disponibilità: il sistema dovrà soddisfare dei requisiti di disponibilità, in particolare dovrà essere sempre disponibile, 24 ore su 24, e i sistemi su cui funzioneranno i server dovranno permettere gli interventi di manutenzione più ordinari (ad esempio la sosti- tuzione dei dischi rigidi) senza compromettere la disponibilità del servizio. Il sistema comunque non è critico, pertanto alcuni rari brevi periodi di indisponibilità in caso di problemi al server sono accettabili.
- Affidabilità dei dati: è necessario, poiché il sistema deve gestire dati scambiati tra gli utenti, che tali dati in condizioni normali non vadano persi: il sistema su cui tali dati sono salvati deve essere affidabile, e quindi dovrà essere opportunamente replicato (ad esempio sfruttando un dispositivo RAID con mirroring) e soggetto a backup in modo da minimizzare le perdite di dati in caso di guasto del sistema.
- Sicurezza: una qualità fondamentale di ogni servizio erogato tramite Internet e che gestisce i dati degli utenti è la sicurezza. In primo luogo, è necessario gestire in modo sicuro e corretto l'autenticazione degli utenti, ossia distinguere i ruoli di utente non registrato, utente registrato e amministratore. Questo può essere realizzato chiedendo all'utente di autenticarsi inserendo il proprio indirizzo e-mail e la password. La pas- sword può essere memorizzata nel database salvandone, al posto del testo in chiaro, il valore di una opportuna funzione di hash, in modo che l'accesso diretto al database da parte dell'amministratore di sistema o di altri utenti non permetta di recuperare le pas- sword (che sono il dato più sensibile tra quelli memorizzati nel sistema). Gli account degli amministratori sono predefiniti, mentre quelli

degli utenti registrati sono soggetti a una registrazione autonoma. In secondo luogo, poiché il sistema permette l'inseri- mento di dati da parte degli utenti, destinati all'inserimento in query sul database oppure in pagine visualizzate da altri utenti, sarà necessario prevederne un opportuno filtraggio al fine di impedire vulnerabilità di sicurezza, ad esempio, SQL injection o cross site scripting (XSS). Inoltre, potrà essere migliorata la sicurezza del software attuando una protezione dei dati in trasmissione mediante il supporto al protocollo HTTPS, almeno per la fase di autenticazione. Tuttavia, il supporto a questo protocollo viene rimandato a una versione futura del software poiché richiede l'acquisto di un certificato firmato da un'autorità di certificazione riconosciuta.

Capitolo 3

Requisiti funzionali

3.1 Scenari

3.1.1 Giovanni e il viaggio in Slovenia

Giovanni, grande appassionato di pallacanestro, desidera ardentemente seguire la sua nazionale nel torneo continentale ospitato quest'anno in Slovenia. Dal momento che l'Italia ha superato il girone di qualificazione, Giovanni decide di recarsi a Lubiana per seguire da vicino la seconda fase della competizione. Informato da un amico sull'ottimo rapporto qualità/prezzo dei pacchetti viaggio della compagnia TravelDream, si reca sul loro sito e sfoglia i pacchetti messi in evidenza nella homepage. Trovatone uno di suo interesse, con partenza da Bari la mattina stessa della prima partita, procede alla registrazione del suo profilo per poter completare la fase d'acquisto. Vista la possibilità di aggiungere al pacchetto viaggio un'escursione della capitale slovena senza costi aggiuntivi, decide di inserirla al pacchetto selezionato. Procede quindi all'acquisto del pacchetto cliccando sul pulsante apposito. Forza Italia!

3.1.2 Stefano organizza un viaggio in Russia

Stefano, un giovane studente di ingegneria, è prossimo alla laurea. Non convinto delle proposte offerte dalla sua università, per il prosieguo dei suoi studi si sta guardando attorno. Viene a conoscenza che a Mosca è presente uno degli atenei più quotati a livelli mondiale per quanto riguarda il suo settore d'interesse. Dopo aver preso contatto via email con l'università, decide di recarsi sul luogo per poter dare un'occhio da vicino alle strutture disponibili e conoscere da vicino i professori che lo accompagneranno durante la sua esperienza di studio all'estero. Si reca pertanto sul sito tramite smartphone, e dopo essersi loggato in quanto utente già registrato, sfoglia i pacchetti predefiniti dopo aver filtrato la ricerca per destinazione. Individua una soluzione che fa al caso suo, comprendente un ostello che si affaccia sulla Piazza Rossa, a soli 15 minuti a piedi dall'università. Non avendo però soldi sulla carta di credito, si limita solo all'inserimento del pacchetto all'interno del suo carrello, da dove sarà poi accessibile per il futuro acquisto.

3.1.3 Aldo invita Enrico ad un viaggio

Aldo,un giovane ragazzo di belle speranze fissato con il fisico, ha programmato da tempo una vacanza in Colombia. Enrico, anch'esso studente, da tempo desideroso di un viaggio in SudAmerica, viene a sapere da Aldo che il sito presso cui egli ha prenotato il viaggio offre la possibilità di invitare un amico a prendere parte ai pacchetti acquistati. Aldo, pertanto invia ad Enrico il codice identificativo ed univoco del suo viaggio. Questo, accedendo dalla pagine principale di TravelDream, inserisce il

codice ricevuto nell'apposita barra di ricerca e gli viene mostrato a video il pacchetto acquistato da Aldo,comprendente data di partenza e arrivo, albergo di soggiorno ed escursioni prepagate. Per poter confermare l'ordine è richiesto però essere registrati al sistema: Enrico quindi accede alla pagina apposita ed inserisce username, password ed indirizzo email per poter creare un nuovo profilo. Una volta loggato con le sue credenziali, procede alla conferma di partecipazione.

3.1.4 Andrea e le vacanze anticipate

Andrea, tecnico del suono di una nota società che opera nel mondo dello spettacolo, ha progettato una vacanza romantica con la sua ragazza Roberta in Salento. Purtroppo per lui, che aveva inizialmente prenotato per l'ultima settimana di Agosto una camera in un agriturismo vicino a Santa Maria di Leuca, il suo datore di lavoro gli ha organizzato una conferenza a Francoforte il 27 Agosto e pertanto si trova costretto ad anticipare di alcuni giorni la data di ritorno. Dopo aver acceduto al suo carrello sul sito di TravelDream, il sistema che ha gestito la sua prenotazione, seleziona il pacchetto che aveva messo in attesa della conferma del suo capo, e clicca sul pulsante modifica. Dopo essersi consultato con Roberta, decide di anticipare le vacanze dal 18 al 25 di Agosto; apportate le giuste correzioni, Andrea può confermare l'acquisto in via definitiva.

3.1.5 Steve invia come Gift List un pacchetto viaggio ad una lista di amici

Steve, che ha appena finito di personalizzare un pacchetto secondo le sue esigenze, decide di inviare il codice associato a questa gift-list ad una serie di amici, con l'obiettivo che questi gli regalino parte di ciò che lui ha scelto. Malcom, ricevuto il codice, lo inserisce nell'apposito campo e visualizza i dettagli delle componenti del pacchetto, quali hotel ed escursioni. Senza che sia richiesta la sua registrazione, Malcom decide di pagare l'escursione nel deserto a Steve, seleziona quindi la tick box in corrispondenza dell'escursione e conferma l'ordine.

3.1.6 Asia crea un nuovo pacchetto e aggiorna i prodotti già esistenti

Asia, un'impiegata di TravelDream, è stata incaricata dalla direzione di aggiornare i pacchetti predefiniti accessibili all'utente. Attraverso le funzionalità riservate ai dipendenti TravelDream, Asia inserisce un nuovo pacchetto in aggiunta a quelli già esistenti. Di ogni suo componente inserisce i dati richiesti, quali il nome nel caso dell'albergo, le date di partenza e arrivo in merito al volo, ed eventuali escursioni, e ne conferma l'inserimento nel database. Asia si accorge poi di essersi dimenticata di modificare il nome dell'hotel presente nel pacchetto "Profondo Blu" in quanto è stato comprato da un altro gestore; posizionatasi all'interno del suddetto pacchetto procede al suo aggiornamento.

3.2 Use Case

Partendo dagli scenari illustrati nella sezione precedente e dall'analisi generale effettuata nella prima parte di questo documento, abbiamo individuato i casi d'uso dell'applicazione che verrà realizzata. In figura 3.1 è rappresentato un modello che rappresenta complessivamente gli attori, i casi d'uso, e le loro interazioni. Il modello è stato realizzato utilizzando la notazione UML opportuna. I singoli casi d'uso saranno descritti in dettaglio nel seguito del paragrafo.

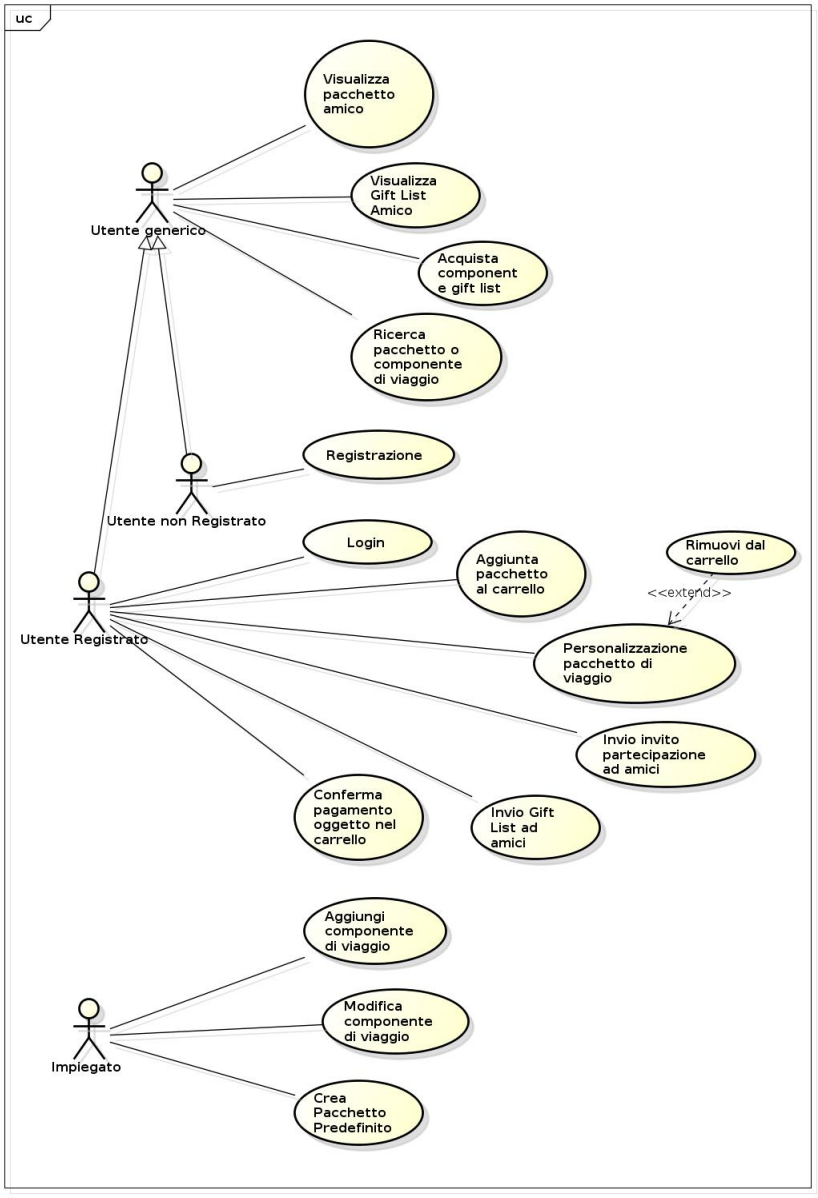

Figura 3.1: Diagramma dei casi d'uso

Registrazione

Attori Utente non registrato (A1)

Precondizioni L'utente non ha mai effettuato la registrazione.

Flusso di Eventi

- L'utente accede alla pagina di registrazione
- Il sistema richiede l'inserimento di nome e cognome, email e password, n° di carta di credito e indirizzo di fatturazione
- Il sistema verifica l'univocità dell'indirizzo email
- Il sistema notifica l'avvenuta registrazione e rimanda alla pagina principale

Postcondizioni L'utente è registrato come Customer(A3) e ha effettuato il login .

Eccezioni La mail inserita risulta già registrata al sistema, viene notificato all'utente che non può procedere.

Login

Attori Utente Customer (A3), A

Precondizioni L'utente è nella pagina principale dell'applicazione.

Flusso di Eventi

- A si posiziona sul form di login
- A inserisce le credenziali quali username e password
- A inoltra la richiesta di login

Postcondizioni Viene notificato all'utente A l'avvenuto login.

Eccezioni Le credenziali inserite sono sbagliate.

Invito di partecipazione a un viaggio

Attori Utente Customer (A3), A Utente Amico (A2), B

Precondizioni A ha un pacchetto personalizzato che decide di condividere con B.

Flusso di Eventi

- A accede al carrello
- A accede alla pagina del pacchetto che vuole condividere con B

- A inserisce l'indirizzo di posta elettronica di B nell'apposito campo
- A clicca sul pulsante di invio

Postcondizioni All'utente A viene notificato che una mail è stata inviata a quell'indirizzo email.

Eccezioni L'indirizzo email inserito viene riconosciuto come non valido.

Utente effettua una ricerca

Attori Utente Customer (A3), o un Utente non registrato (A1) B

Precondizioni

Flusso di Eventi

- B accede alla pagina di ricerca
- B seleziona il un luogo di destinazione come obiettivo di ricerca
- B inoltra i parametri della ricerca
- Visualizza i risultati

Postcondizioni L'utente si trova sulla pagina che mostra i risultati della sua ricerca .

Eccezioni L'utente non inserisce una data di andata (viene automaticamente considerata quella odierna). La destinazione non esiste. L'utente inserisce una data di ritorno precedente a quella di andata.

Utente customer aggiunge un elemento al carrello.

Attori Utente Customer (A3)

Precondizioni L'utente A è loggato e ha effettuato una ricerca, oppure è sulla pagina principale dove è visualizzato un elenco di pacchetti predefiniti.

Flusso di Eventi

- A visualizza l'elenco di componenti
- A seleziona il pacchetto che più gli aggrada
- Dalla pagina dettaglio del pacchetto conferma il suo interesse premendo l'apposito pulsante

Postcondizioni Il pacchetto selezionato viene inserito nel carrello dell'utente.

Eccezioni E' già presente nel carrello un pacchetto con le stesso identificativo.

Utente customer personalizza un pacchetto predefinito

Attori Utente Customer (A3)

Precondizioni L'utente si trova nel carrello, nel quale è stato aggiunto un pacchetto predefinito.

Flusso di Eventi

- L'utente sceglie un pacchetto predefinito da personalizzare
- L'utente selezione l'elemento da modificare (ad. es. l'Hotel)
- L'utente visualizza l'elenco di elementi sostitutivi compatibili col suo pacchetto (ad. es. altri Hotel nello stesso luogo)
- L'utente conferma il pacchetto

Postcondizioni All'utente vengono notificate le avvenute modifiche al pacchetto tramite un piccolo riepilogo. Il pacchetto viene quindi inserito nel suo carrello in attesa di ulteriori modifiche o conferma d'acquisto.

Eccezioni Il controllo sulla consistenza dei prodotti nel pacchetto ha esito negativo, l'utente viene avvisato e non viene permesso di procedere all'acquisto del pacchetto

Un utente riceve l'invito per partecipare ad un viaggio

Attori Utente Customer (A3), A Utente Amico (A2), B

Precondizioni A ha inviato a B il codice del suo pacchetto e quest'ultimo decide di parteciparvi.

Flusso di Eventi

- B riceve una mail contenente il codice corrispondente al pacchetto dell'utente A.
- B visualizza il pacchetto di A inserendone il codice nell'apposito form presente nella pagina principale
- B, avendo deciso di parteciparvi, si registra divenendo un utente customer a sua volta, qualora non sia già un utente riconosciuto.
- Da ultimo B conferma la sua partecipazione al viaggio.

Postcondizioni All'utente B viene notificato a video che la sua partecipazione è stata confermata, mentre all'utente A viene inviata una mail informativa.

Eccezioni

Un utente riceve una gift-list e decide di regalare alcuni componenti

---

Attori Utente generico, C

Precondizioni L'utente ha ricevuto il codice di una gift-list.

Flusso di Eventi

- C inserisce nell'apposito campo di ricerca il codice ricevuto.
- Viene mostrata a video la giftlist, con i relativi componenti.
- C seleziona i componenti che è suo interesse acquistare.
- C conferma l'acquisto dei componenti selezionati.

Postcondizioni Viene notificato a C e al proprietario della gift-listche è stato effettuato correttamente l'acquisto.

Eccezioni

Un impiegato TravelDream aggiunge un nuovo pacchetto alla lista di quelli predefiniti

---

Attori Impiegato (A4), C

Precondizioni L'impiegato ha acceduto alla funzionalità apposita per l'inserimento di un nuovo pacchetto.

Flusso di Eventi

- C seleziona una località tra quelle disponibili.
- Il sistema elabora la scelta di C e fornisce una lista degli alberghi disponibili.
- C seleziona un albergo e procede con l'aggiunta di attività integrative quali escursioni.
- C aggiunge quindi al pacchetto una data di partenza e una di ritorno.
- Se è tutto corretto, C prosegue nella pubblicazione del pacchetto.

Postcondizioni Il pacchetto viene aggiunto alla lista dei pacchetti predefiniti e a all'impiegato viene visualizzato un piccolo riepilogo dell'avvenuta operazione.

Eccezioni Il controllo sulla consistenza delle date ha esito negativo, l'impiegato viene avvisato e può procedere alla modifica. Il pacchetto proposto è già inserito all'interno del database.

Un impiegato TravelDream modifica un pacchetto predefinito

Attori Impiegato (A4), C

Precondizioni L'impiegato ha acceduto alla pagina contenente l'elenco dei pacchetti predefiniti.

Flusso di Eventi

- Selezionato il pacchetto d'interesse, C procede alla sua modifica tramite l'apposito pulsante.
- C, volendo modificare le escursioni, rimuove tramite l'apposita funzione quelle esistenti.
- Il sistema chiede all'impiegato se è sicuro dell'operazione.
- C conferma e prosegue aggiungendo una nuova attività al posto di quelle presenti in precedenza, scegliendola da una lista proposta.
- C prosegue confermando il pacchetto su richiesta da parte del sistema.

Postcondizioni Il pacchetto viene aggiunto alla lista dei pacchetti predefiniti e a all'impiegato viene visualizzato un piccolo riepilogo dell'avvenuta operazione.

Eccezioni Non del database.

3.3 Diagrammi di sequenza

In questa sezione sono presentati i diagrammi di sequenza di alcune interazioni fondamentali.

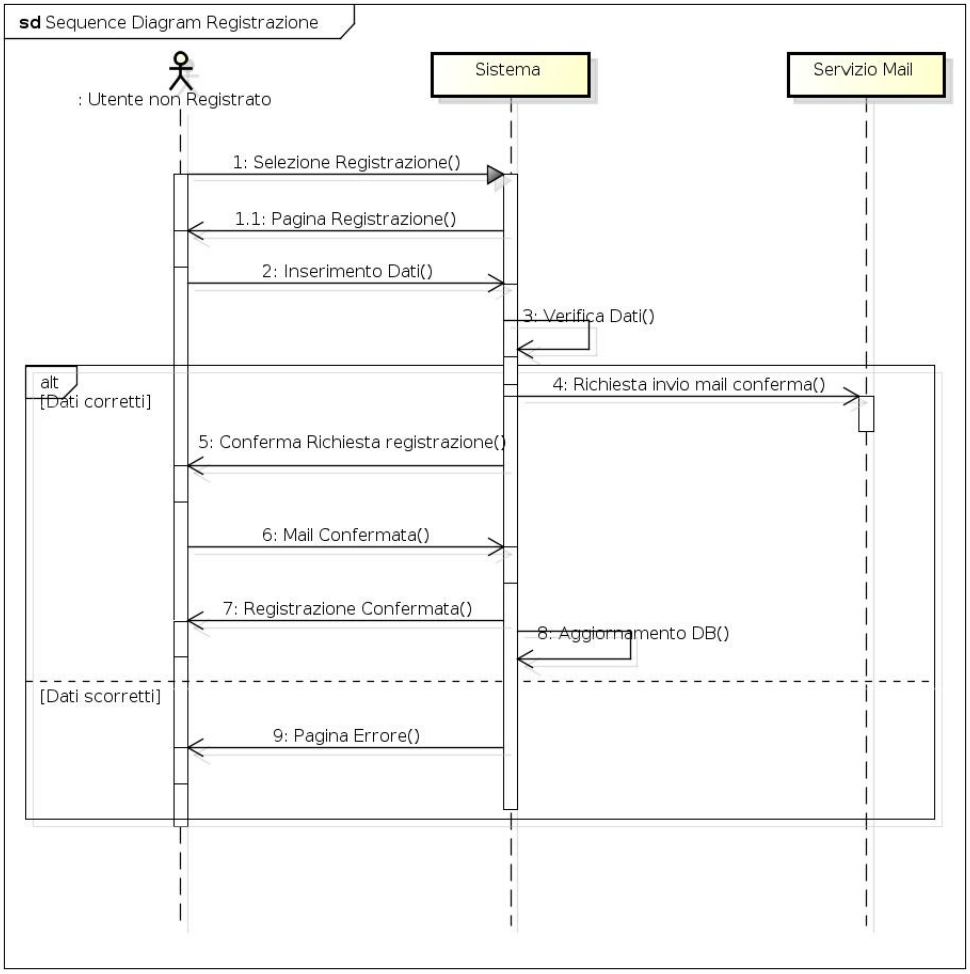

Figura 3.2: Diagramma di sequenza relativo alla fase di registrazione.

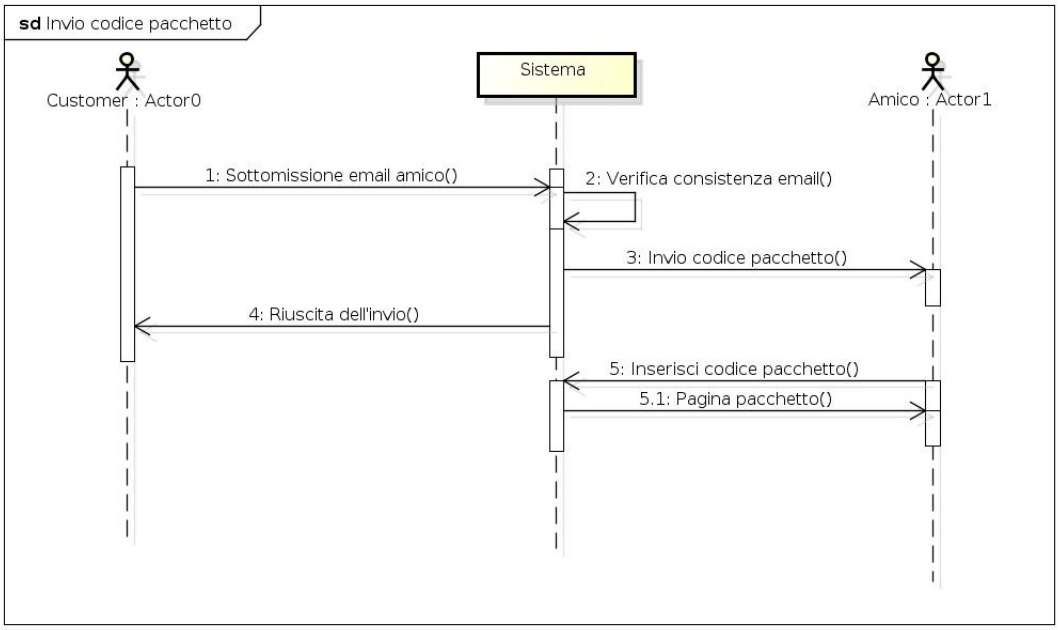

Figura 3.3: Diagramma di sequenza relativo all'invio del codice pacchetto.

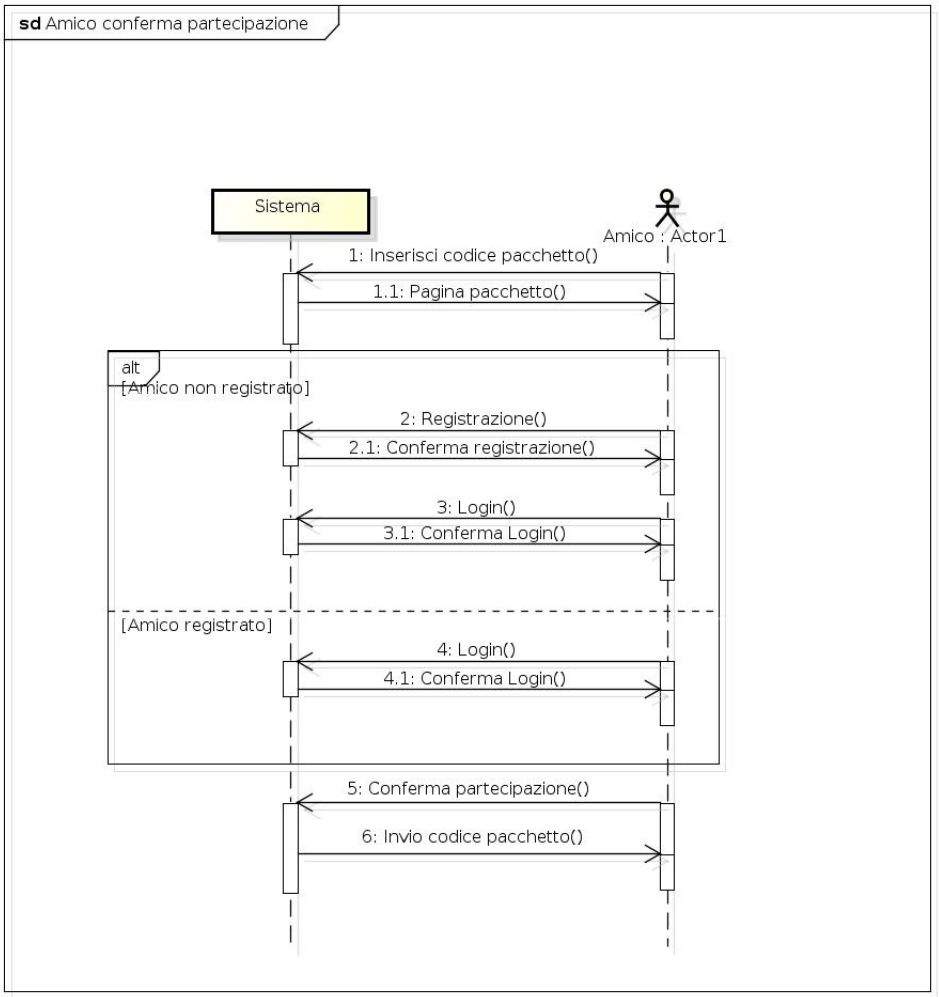

Figura 3.4: Diagramma di sequenza relativo alla conferma di partecipazione al pacchetto da parte di un amico.

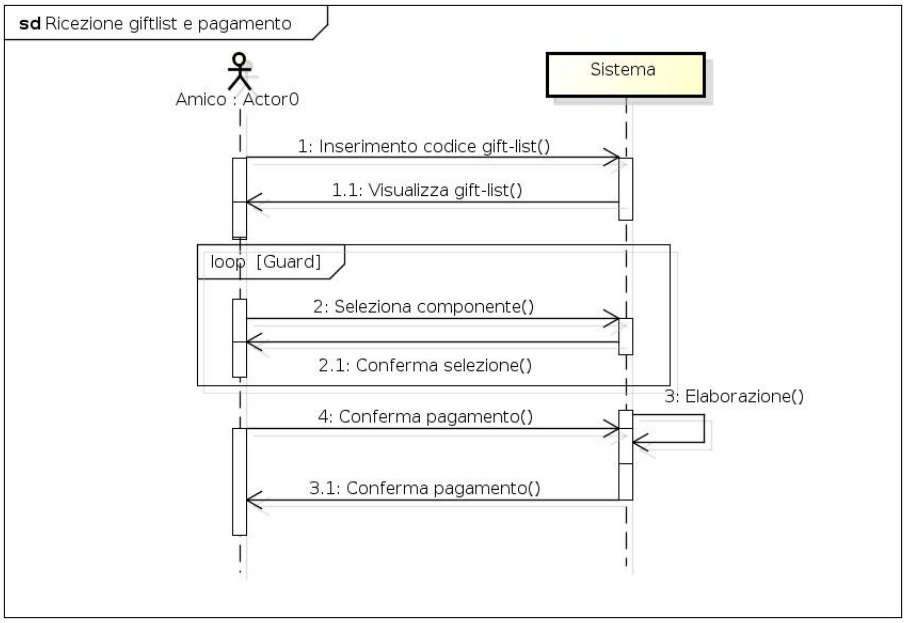

Figura 3.5: Diagramma di sequenza relativo alla ricezione della giftlist e al pagamento di almeno uno dei suoi componenti.

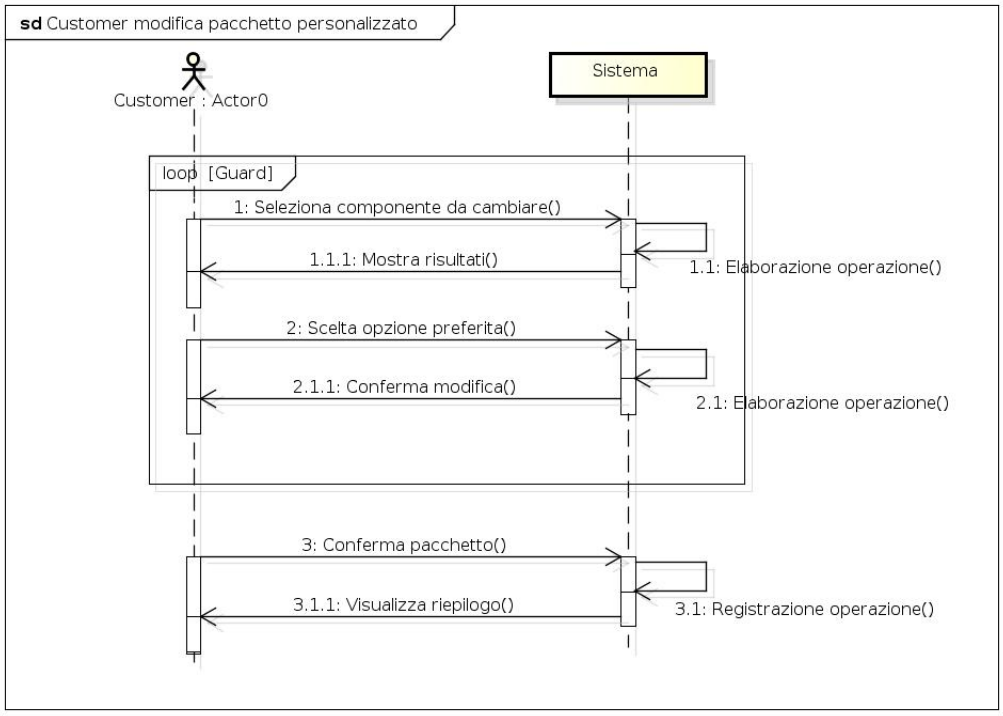

Figura 3.6: Diagramma di sequenza relativo alla modifica di un pacchetto presente nel proprio carrello.

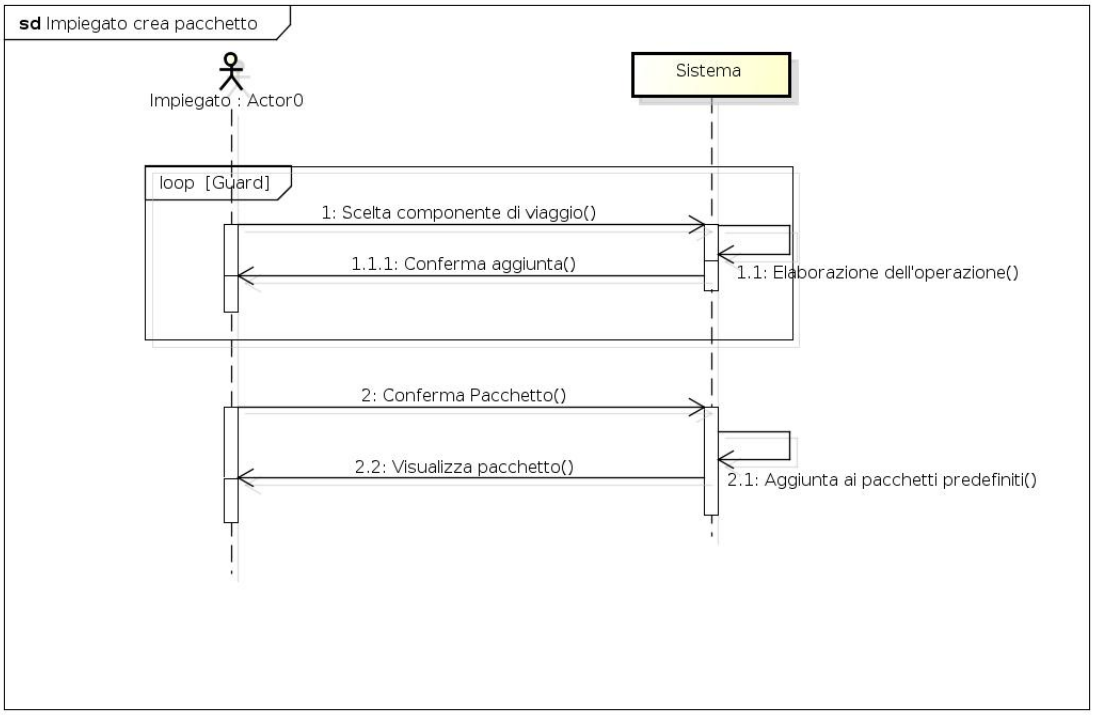

Figura 3.7: Diagramma di sequenza relativo alla creazione di un pacchetto da parte di un impiegato.

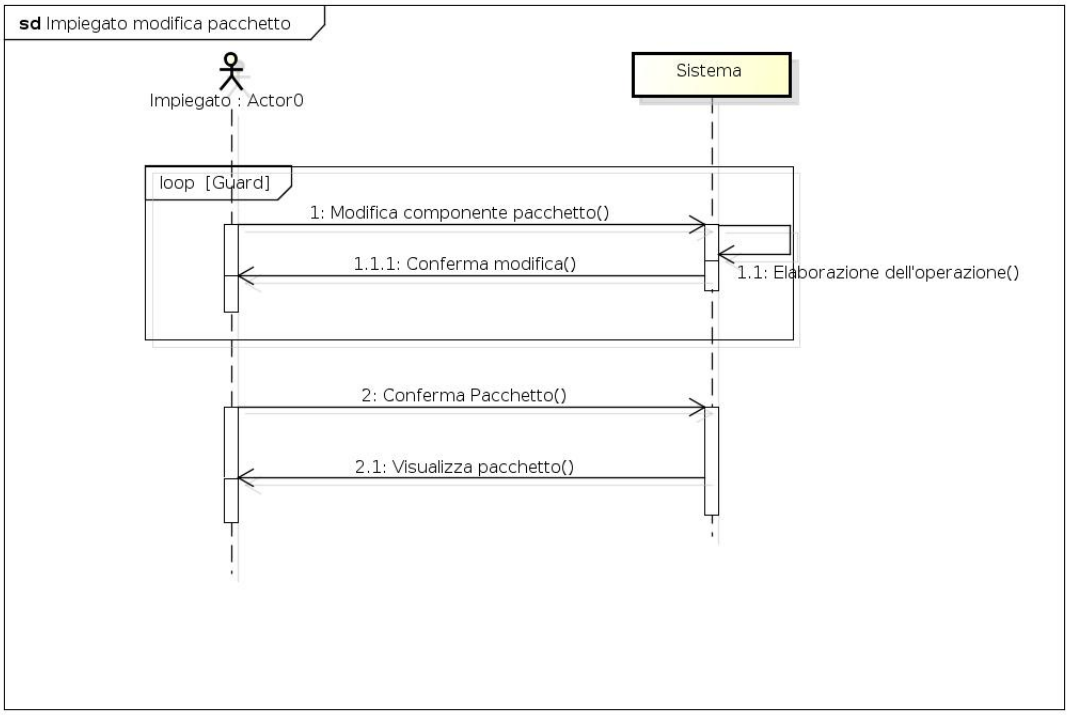

Figura 3.8: Diagramma di sequenza relativo alla modifica di un pacchetto già esistente da parte di un impiegato.

Capitolo 4

Specifiche del sistema

4.1 Diagramma delle classi

Partendo dai requisiti e dalle premesse illustrate nei capitoli precedenti, è stato realizzato un diagramma delle classi con lo scopo, di illustrare, a livello statico, le entità coinvolte e le relazioni tra loro esistenti.

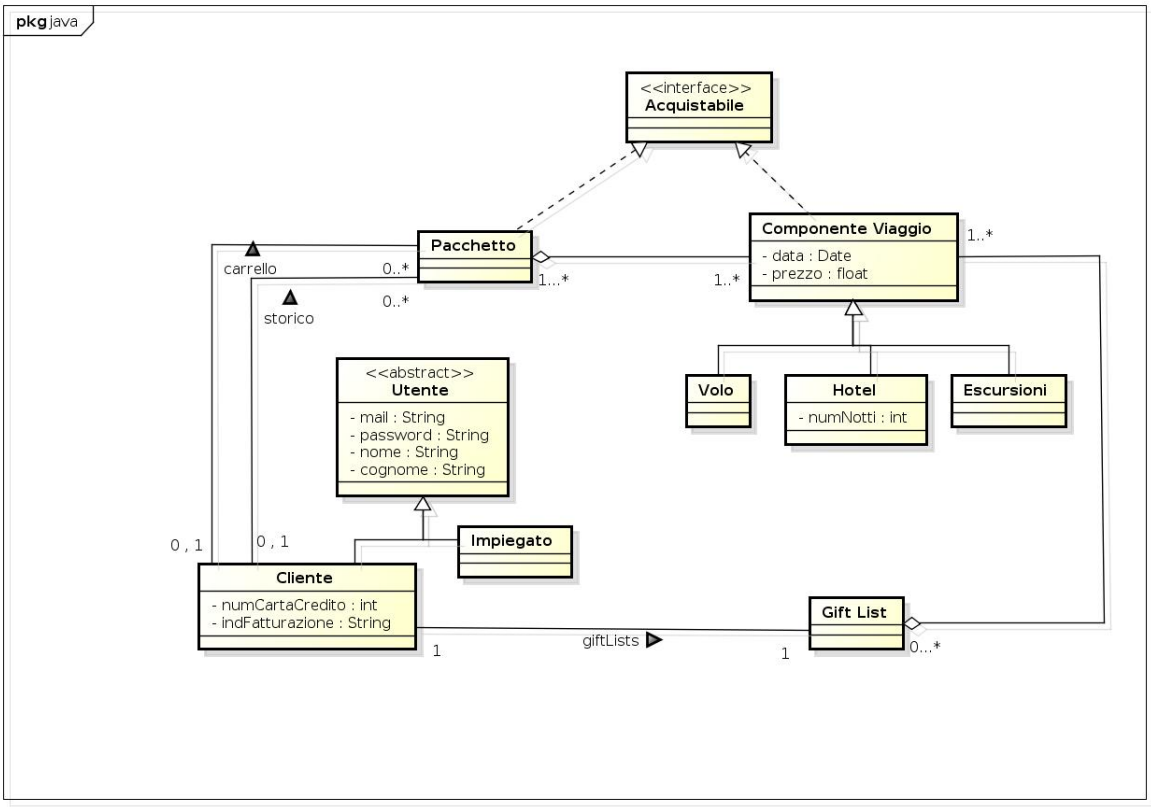

Figura 4.1: Diagramma delle classi

Dalla figura soprastante, si può vedere come l'utente, tramite le relazioni "storico" e "carrello"

tiene conto sia dei pacchetti già comprati e a cui ha preso parte, sia di quelli nuovi, messi in attesa d future modifiche e di una loro eventuale conferma d'acquisto.

4.2 Analisi formale

Partendo dalle considerazioni fatte in precedenza con il diagramma delle classi, e dall'analisi dei vincoli e delle assunzioni descritti di conseguenza, abbiamo realizzato un modello formale del sistema utilizzando la sintassi di Alloy, uno strumento di verifica formale di modelli bounded-scope basato su logica, e ne abbiamo verificato la correttezza utilizzando Alloy Analyzer. Riportiamo di seguito il codice sorgente del modello statico del sistema.

```
module TravelDream

sig Str{}

sig PacchettoPredefinito {
    componentiViaggioPredefiniti:    some ComponenteViaggioPredefinito,
}

sig PacchettoPersonalizzato {
    componentiViaggio: some ComponenteViaggio,
    personalizzatoDa: one PacchettoPredefinito,
}

sig ComponenteViaggio {
    personalizzatoDaComponente: one ComponenteViaggioPredefinito,
}

sig ComponenteViaggioPredefinito{}

sig UtenteNonRegistrato {
    regala: set ComponenteViaggio,
}

abstract sig Utente {
    mail: one Str,
}

sig Impiegato extends Utente {
    crea: set PacchettoPredefinito,
}

sig Cliente extends Utente {
    giftLists: lone GiftList,
    carrello: set PacchettoPersonalizzato,
    storico: set PacchettoPersonalizzato,
    acquista: set ComponenteViaggio,
    invita: set PacchettoPersonalizzato -> Cliente,
    accettaInvito: set PacchettoPersonalizzato -> Cliente,
}
```

```
sig GiftList{
    elementiLista:      some ComponenteViaggio,
}

// Ogni utente deve avere una mail univoca
fact mailUnivoche {
    no u1, u2: Utente | u1 != u2 and u1.mail = u2.mail
}

// Ogni pacchetto personalizzato è nel carrello o nello storico di uno
   ed un solo utente, e o è in un carrello o è in uno storico
fact{
    all p: PacchettoPersonalizzato | one c: Cliente | p in (c.carrello
       + c.storico)
    no p: PacchettoPersonalizzato, c1, c2: Cliente | p in c1.carrello
       and p in c2.storico
}

// Ogni ComponenteViaggio è contenuto in uno ed un solo pacchetto
fact{
    all cv: ComponenteViaggio | one p: PacchettoPersonalizzato | cv in
       p.componentiViaggio
}

// Ogni GiftList appartiene ad uno ed un solo utente
fact{
    all g: GiftList | one c: Cliente | g in c.giftLists }

// Nella GiftList di un cliente, ci possono essere solo componenti
   viaggio di pacchetti (nel carrello o nello storico) di quel cliente
fact{
    all c: Cliente,
    cv: ComponenteViaggio | cv in c.giftLists.elementiLista implies
            cv in c.(carrello + storico).componentiViaggio }

// Se un cliente ha acquistato un pacchetto (è nello storico),allora
   tutti i suoi componenti devono essere stati acquistati (da lui) o
   regalati (da utenti non registrati)
fact{
    all c: Cliente,
    p: PacchettoPersonalizzato | p in c.storico implies
                all cv: ComponenteViaggio | cv in p.componentiViaggio
                   implies (
            one a: UtenteNonRegistrato | cv in a.regala or   cv in c.
               acquista )
}

// Se un cliente ha acquistato un ComponeneteViaggio,questo appartiene
   ad un pacchetto nel suo carrello o nel suo storico
fact{
```

```
    all c: Cliente,
    cv: ComponenteViaggio | cv in c.acquista implies
            cv in c.(carrello + storico).componentiViaggio
}

// Un ComponenteViaggio può essere regalato solo se è presente in una
   GiftList fact{
    all cv: ComponenteViaggio, u: UtenteNonRegistrato | cv in u.regala
       implies
            (some c: Cliente | cv in c.giftLists.elementiLista)
}

// Ogni ComponenteViaggio può essere acquistato al massimo da un
   acquirente (Cliente o UtenteNonRegistrato)
fact{
    no cv: ComponenteViaggio,
    c: Cliente,
    u: UtenteNonRegistrato | cv in c.acquista and cv in u.regala
    all cv: ComponenteViaggio | (one c: Cliente | cv in c.acquista) or
       (one u: UtenteNonRegistrato | cv in u.regala)
}

/** Inviti **/
fact{

// b può accettare l'invito di a solo se a l'ha invitato (su un
   PacchettoPersonalizzato p)
    all a, b: Cliente, p: PacchettoPersonalizzato | a in p.(b.
       accettaInvito) implies b in p.(a.invita)

// non si può invitare se stessi
    no a: Cliente, p: PacchettoPersonalizzato | a in p.(a.invita)

// se a invita b su un pacchetto, a ha il pacchetto nel carrello o
   nello storico all a, b: Cliente,
    p: PacchettoPersonalizzato | p in (a.invita.b) implies p in a.(
       carrello + storico)

// se b acetta l'invito di a su p, allora b avrà una copia del
   pacchetto p nel carrello
    all a, b: Cliente,
    p: PacchettoPersonalizzato | b in p.(a.invita) implies #b.carrello
       > 0
}

/** personalizzazione **/
fact{
// si possono solo sostituire componenti viaggio, quindi il numero di
   componenti non cambia fra il pacchetto predefinito e quello
   personalizzato
    all pe: PacchettoPersonalizzato,
```

```
    pr: PacchettoPredefinito | pr in pe.personalizzatoDa implies
                #(pe.componentiViaggio) = #(pr.
                    componentiViaggioPredefiniti)

// tutti i pacchetti predefiniti devono essere stati creati da qualcuno
    all pr: PacchettoPredefinito | one i: Impiegato | pr in i.crea
}
```

Per ora evitiamo di proporre qui i predicati. Verranno invece presentati i mondi generati di dimensione 10, ovvero contententi al massimo 10 elementi per ogni signature che è stata definita. Inoltre, si sottolinea come dall'analisi dei mondi generati non sono sorte alcun tipo di inconsistenze.

4.2.1 Approssimazioni

Nel realizzare il modello Alloy sono stati omessi o semplificati alcuni vincoli di seguito illustrati:

- all'interno di ciascun pacchetto non si tiene conto di quanti e del tipo dei componenti in esso contenuti (ComponenteViaggio);
- dal punto precedente ne consegue che non si è potuta rappresentare l'ugugalianza a livello di componenti tra due pacchetti personalizzati;
- è stato trascurato che un utente non registrato possa ricevere l'invito di partecipazione ad un pacchetto e registrarsi di conseguenza;

4.2.2 Mondi generati

Abbiamo generato, tramite Alloy Analyzer, alcuni mondi che soddisfano il modello (risultato dell'esecuzione del predicato show, eventualmente inserendo ulteriori vincoli sul numero di elementi per alcuni tipi).

Nota: per semplicità di linguaggio i termini ComponenteViaggio e ComponenteViaggioPredefinto sono stati abreviati in, rispettivamente, CV e CVP.

Mondo 1

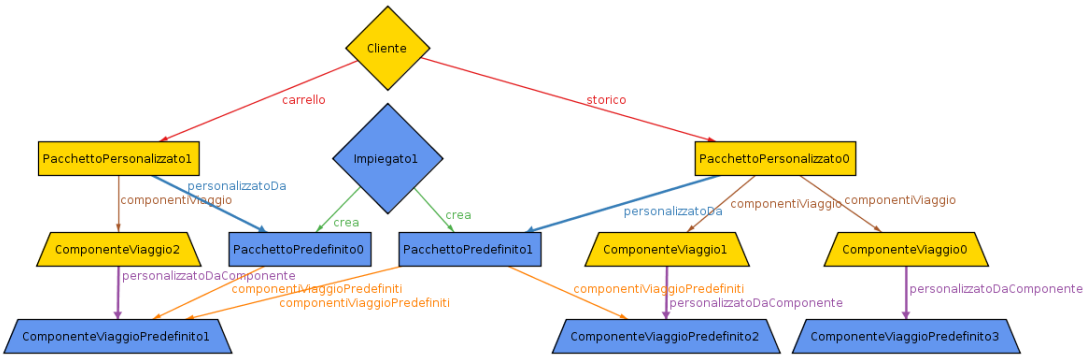

Figura 4.2: Rappresentazione della personalizzazione di pacchetti

Cliente ha nel suo carrello un pacchetto (PacchettoPersonalizzato1) e come si può notare, finché tale pacchetto è nel carrello, è possibile che non tutti i suoi componenti siano acquistati o regalati.

Nello storico, Cliente ha PacchettoPersonalizzato0 che è stato personalizzato a partire da PacchettoPredefinito1; in quest'ultimo erano predisposti due componenti viaggio (CVP1 e CVP2).

Cliente ha modificato il componente CVP2 (ottenendo il componente CV1) e ha invece cambiato il componente CVP1 con il componente CVP3.

Il PacchettoPersonalizzato1 è stato personalizzato a partire dal PacchettoPredefinito0 aggiornandone il suo unico componente.

I pacchetti PacchettoPredefinito0 e PacchettoPredefinito1 sono stati creati dall'utente Impiegato1.

Codice del predicato:

```
pred showPersonalizzazione(){
    #Cliente = 1
    all c: Cliente | #(c.(storico+carrello)) = 2
    #PacchettoPredefinito = 2
    #ComponenteViaggioPredefinito = 4
}

run showPersonalizzazione for 10
```

Mondo 2

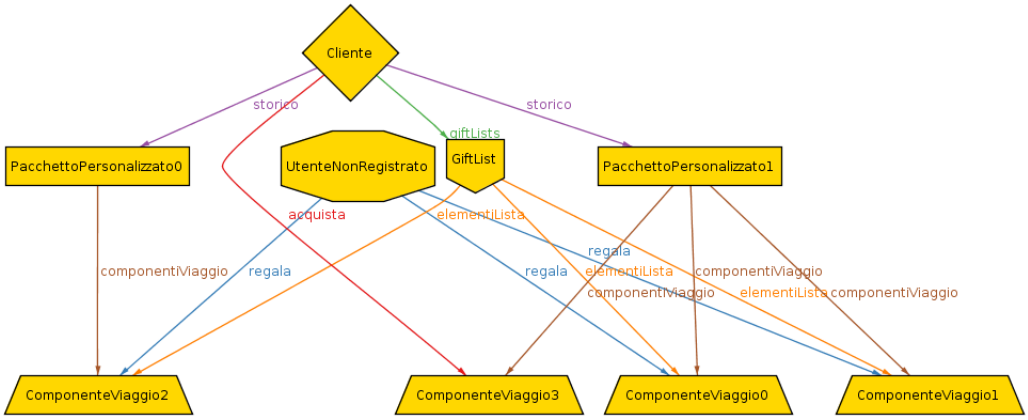

Figura 4.3: Rappresentazione dello storico e della giftlist

Cliente nella sua GiftList ha inserito i componenti viaggio CV0, CV1 e CV2; fortunatamente per Cliente, un utente benefattore UtenteNonRegistrato ha regalato a Cliente tutti i pacchetti presenti nella GiftList.

I pacchetti di Cliente (PacchettoPersonalizzato0 e PacchettoPersonalizzato1) sono nello storico e sono quindi stati acquistati, dal momento che affinchè un pacchetto si trovi nello storico, è necessario che ogni suo componente sia stato acquistato o regalato (come si può vedere dalle relazioni acquista e regala). Per esempio si osservi il caso di PacchettoPersonalizzato1 dove i componenti CV0 e CV1 sono stati regalati da UtenteNonRegistrato e il componente CV3 è invece stato acquistato da Cliente.

Codice del predicato:

```
pred showStoricoEGiftList(){
    #Cliente = 1
    all c: Cliente | #(c.(storico)) = 2
    #GiftList = 1
```

```
    #componentiViaggio = 4
}

run showStoricoEGiftList for 10
```

Mondo 3 e 4

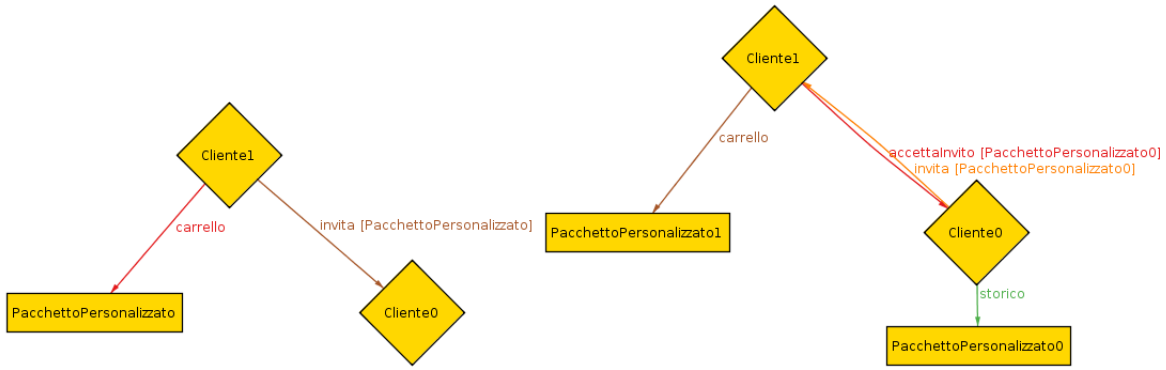

Figura 4.4: Rappresentazione dell'invito alla partecipazione di un pacchetto

Nella prima immagine il cliente Cliente1 invita il cliente Cliente0 a partecipare al pacchetto PacchettoPersonalizzato.

Codice del predicato:

```
pred showInvitoSenzaAccettazione(){
    #Cliente = 2
    #PacchettoPersonalizzato = 1
    #invita > 0
    #accettaInvito = 0
}

run showInvitiSenzaAccettazione for 10
```

Quindi è stato modellato il caso in cui Cliente0 accetti l'invito, ottenendo una copia nel suo carrello di PacchettoPersonalizzato1, che potrà poi personalizzare ulteriormente.

Codice del predicato:

```
pred showInvitoConAccettazione(){
        #Cliente = 2
        #PacchettoPersonalizzato = 2
        #invita > 0
        #accettaInvito > 0
}

run showInvitoConAccettazione for 10
```