POLITECNICO DI MILANO

SCUOLA DI INGEGNERIA DELL'INFORMAZIONE

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

DESIGN DOCUMENT

TRAVEL DREAM

Progetto di ingegneria del software II

Sara Marchesini, Sebastiano Mariani, Riccardo Mastellone

Contents

1 Introduzione

1.1 Scopo del documento

Lo scopo di questo documento è definire un'architettura generale per il sistema da implementare, che dovrà poi essere raffinata in fase di implementazione.

Questa struttura sarà basata sulle specifiche descritte nel documento di analisi dei requisiti.

Si rivolge principalmente al team di svilupatori incaricati di implementare e mantere il software.

1.2 Referenze

- Documento di analisi dei requisiti TravelDream
- Specifiche di progetto "Progetto AA 2013-2014(TravelDream)"

1.3 Definizioni e abbreviazioni

1.3.1 Definizioni

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

1.3.2 Abbreviazioni

Abbreviazione	Definizione
DBMS	Database Management System
JEE	Java Enterprise Edition
API	Application Programming Interface
E-R	Entità-Relazione
EJB	Enterprise Java Bean
HTML	Hypertext Markup Language
HTTP	Hypertext Transfer Protocol
IDE	Integrated Development Environment
JDBC	Java Database Connectivity
UML	Unified Modeling Language
UX	User eXperience
JSF	Java Servlet Faces
MVC	Model View Controller
JPA	Java Persistence API
SMTP	Simple Mail Transfer Protocol
XHTML	eXtensible Hypertext Markup Language
POJO	Plain Old Java Object

1.4 Panoramica

Il documento è così suddiviso

1. Introduzione: breve introduzione contenente lo scopo del documento e la definizione di alcuni termini e abbreviazioni.
2. Descrizione generale: descrizione generale delle tecnologie scelte e delle varie suddivisioni in layer e tier ritenute opportune.
3. Progetto dei dati: descrizione tramite modelli formali(modello E/R, odello logico) della base di dati
4. Progetto del business tier: descrizione della logica applicativa e delle integrazioni tra i componenti mediante diagrammi BCE
5. Progetto del client: descrizione della navigazione tra le varie pagine mediante diagramma UX

2 Descrizione generale

Il sistema TravelDream è stato ideato per aiutare da un lato i dipendenti dell'azienda nella gestione del catalogo dei pacchetti vacanza e nella vendita degli stessi, e dall'altro i clienti nella consultazione del catalogo e nella creazione e acquisto di pacchetti.

L'interfacciamento al sistema avviene esclusivamente via web per rendere il servizio disponibile al maggior numero di persone, e fruibile in maniera estremamente semplice.

Per raggiungere questo scopo il software mette a disposizione dei dipendenti funzioni intuitive per la gestione dei pacchetti come ricerca, cancellazione e aggiunta, mostrando a schermo tutti i campi necessari all'esecuzione della funzione. inoltre fornisce ai clienti strumenti di ricerca arricchiti di filtri per destinazione, data di partenza e tipo di vacanza, per sfogliare il catalogo in maniera molto semplice, e strumenti per la creazione di pacchetti personalizzati, con possibilità di invito di persone esterne.

Gli utenti che vogliono usufruire di tale dservizio devono essere obbligatoriamente registrati.

2.1 Scelte Tecnologiche

Di seguito sono riportate le scelte tecnologiche, effettuate tenendo conto dei vincoli imposti dall'utilizzo della piattaforma JEE, da rispettare nello sviluppo del software.

Tali scelte condizionano fortemente la progettazione del sistema riportata in questo documento. <TODO sistemare>

- Il sistema è basato sulla piattaforma JEE 7. In particolare verranno utilizzate alcune specifiche:
  - EJB 3.2 per lo sviluppo della logica applicativa
  - JSF 2.2 per lo sviluppo dell'applicazione web
  - JavaMail API per interfacciare il sistema con il servizi di posta elettronica
- Come application server viene utilizzato Glassfish 4.0
- come DBMS viene utilizzato MySQL 5.6.14, tale scelta non vincola la struttura del codice dal momento che l'interfacciamento al database avviene tramite JPA, infatti cambiando il DBMS con un altro compatibile con l'application server utilizzato, il codice non varia

Come ambiente di sviluppo è consigliato l'utilizzo di Eclipse (versione Kepler) con i plugin relativi alle componenti sopra specificate.

Per facilitare la collaborzione tra i vari componenti del team di sviluppo è stata attivata una repository git presso google code a questo indirizzo https://code.google.com/p/traveldream-mariani-marchesini-mastellone/

2.2 Descrizione dell'architettura

Si è scelto di utilizzare una architettura multi-tier suddivisa in 4 livelli logici (tier):

- **Client tier:**
  - livello da cui gli utenti accedono all'applicazione tramite un comune browser web. Comunica con il web tier attraverso il protocollo Http inviando al server i dati inseriti dall'utente e si occupa della presentazione delle pagine una volta ricevuto il file XHtml dal server. Inoltre può eseguire codice Javascript che realizza semplici controlli sulla sintassi dei dati immessi (questi controlli andranno effettuati anche lato server per questioni di sicurezza, servono principalmente per ridurre il traffico tra client e server causato da errori di distrazione (es: mancanza della @ nel campo mail)), oppure che realizza effetti grafici all'interno della pagina e nella transizione tra le varie schermate attraverso l'ausilio delle librerie jQuery.

- **Web tier:**
  - Questo livello riceve le richieste del client e in base ai dati ricevuti e al tipo di richiesta(GET, POST ecc..),grazie alla tecnologia JSF, interagisce con il business tier, genera la pagina approriata e la spedisce al client richidente.
- **Business tier:**
  - Questo livello incapsula la logica applicativa dell'intero sistema e comunica direttamente con il database. I dati dell'applicazione, ovvero il modello all'interno del pattern MVC, sono rappresentati da POJO(entity beans) garantiti persistenti grazie alla tecnologia JPA<TODO da riguardare>. Il controller invece è realizzato mediante componenti EJB che implementano la logica applicativa. Inoltre interagisce con un server di post elettronica, utilizzando il protocollo SMTP attraverso le API javaMail, che permette a clienti di inviare inviti e liste desderi.
- **Data tier:**
  - È costituito da un DBMS che realizza la persistenza dei dati e li mette a disposizione dell'application server. Quest'ultimo comunica con il DBMS grazie al driver JDBC.

In questo tipo di arcitettura ogni tier comunica solamente con quello immediatamente adiacente, rendendo di fatto l'applicazione modulare.

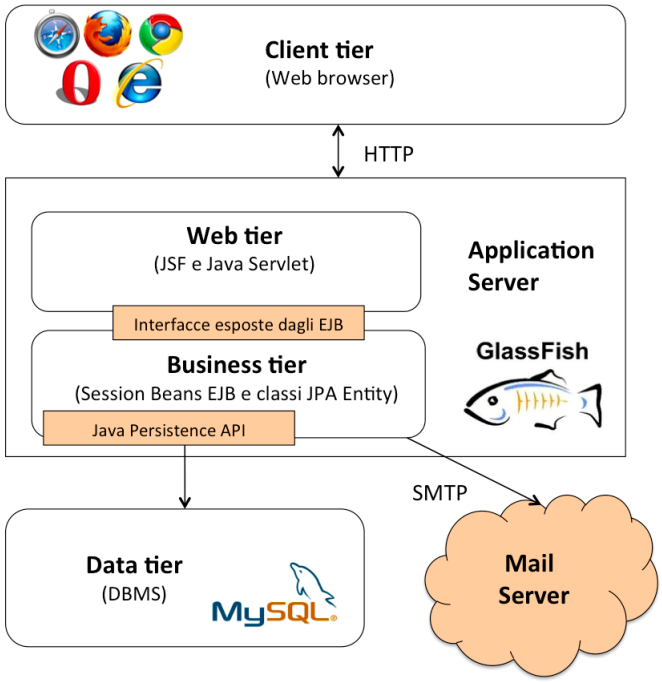

Figure 1: Architettura: tier logici

A livello fisico i 4 tier logici sono suddivisi in 2 layer:

- **Client:**
  - È il terminale da cui l'utente accede all'applicazione, e incapsula il solo client tier.

- **Server:**
  - È il server fisico vero e proprio su è deployato l'application server e in questa prima implementazione anche il DBMS. Questo layer incapsula il web tier, il business tier e il data tier.

Il server mail è esterno al sistema e si appoggia a servizi esterni.

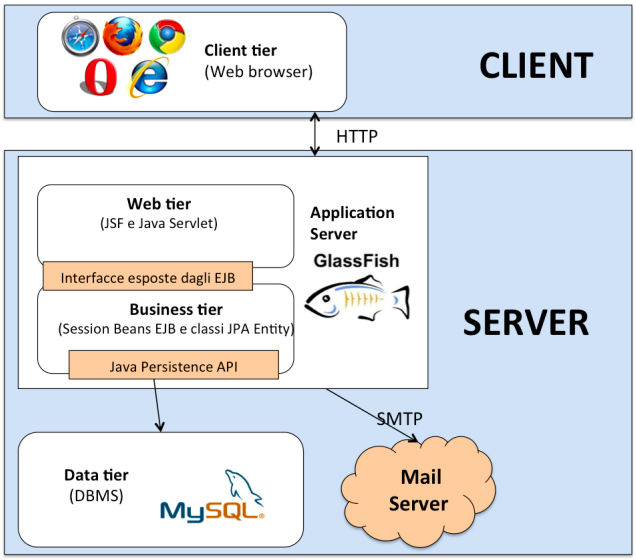

Figure 2: Achitettura: layer fisici

3 Progetto dei dati

La struttura dei dati è basata sul modello relazionale e da implementare su un server MySQL

3.1 Progettazione concettuale

Sono state individuate le entità e le relazioni di interesse per l'applicazione sulla base del diagramma delle classi presente nel documento di analisi dei requisiti.

Di seguito è riportato il diagramma E-R generato da questa analisi.

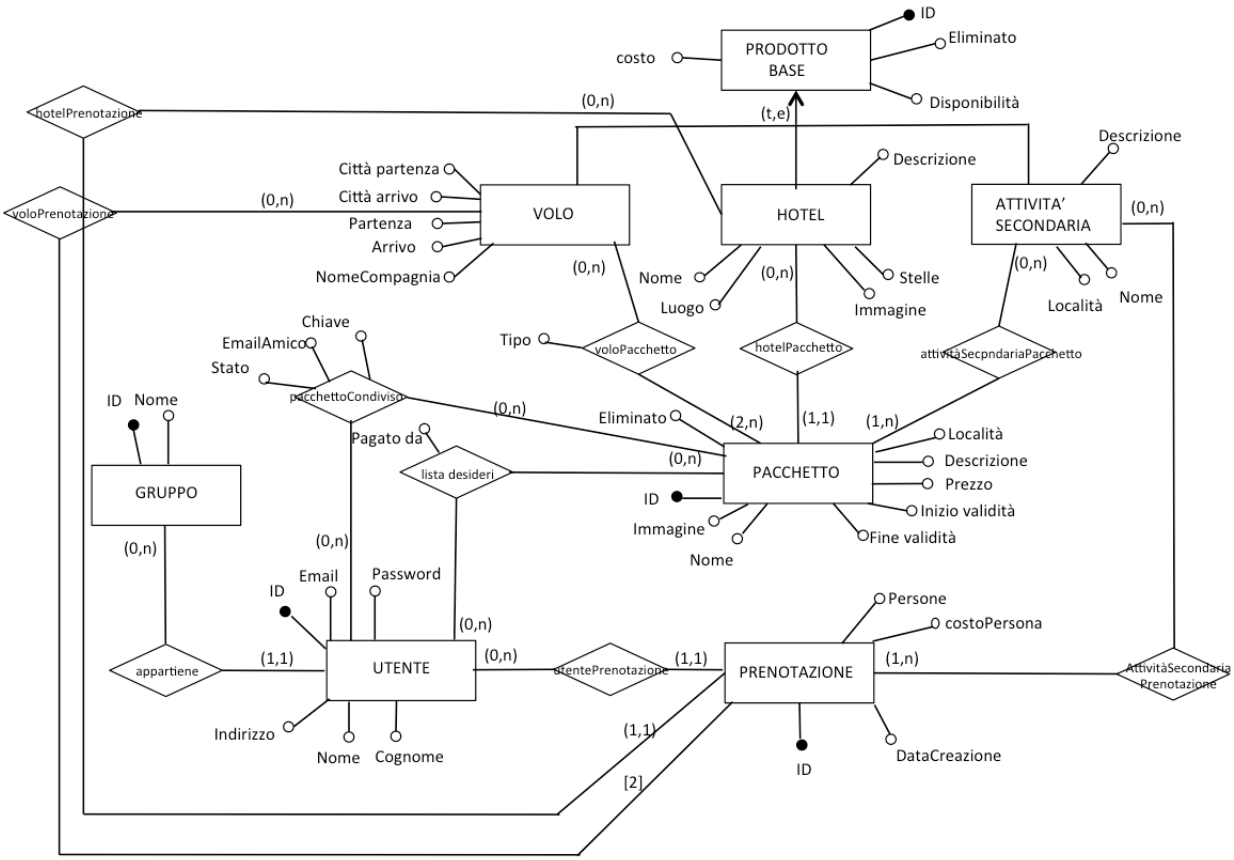

Figure 3: Diagramma E-R

Le entità e le relazioni sono:

- **UTENTE:** rappresenta un utente registrato al sistema, è identificato da un ID univoco autoincrementale, e ha come attributi i dati anagrafici(nome, cognome, indirizzo) e le credenziali di accesso(nome utente coincidente con la mail di registrazione e password). Partecipa alle seguenti relazioni:
  - APPARTIENE: collega ogni utente a uno e un solo gruppo, permettendo così la distinzione tra clienti e dipendenti e amministratore.
  - UTENTEPRENOTAZIONE: presente solo negli utenti appartenenti al gruppo cliente; collega il cliente alle prenotazioni da lui effettuate (zero o più).
  - LISTA DESIDERI: presente solo negli utenti appartenenti al gruppo cliente; specifica per ogni cliente quali pacchetti sono stati aggiunti alla propria lista desideri e attraverso l'attributo "pagato da" quali sono stati pagati e da chi.

  - PACCHETTO CONDIVISO:presente solo negli utenti appartenenti al gruppo cliente; specifica per ogni cliente quali pacchetti sono stati inviati a amici e attraverso gli attributi "emailAmico" e "stato" precisa l'amico invitato e se l'invito è stato accettato o è ancora pendente.

- **GRUPPO:** entità identificata da un ID univoco autoincrementale, e presenta l'attributo nome di tipo enuemrazione che indica il gruppo di appartenenza e quindi di privilegio di un utente. Ammette i seguenti valori: amministratore, con possibilità di scrittura sulla tabella UTENTI per aggiungere o rimuovere dipendenti; dipendente, con possibilità di scrittura sulle tabelle PACCHETTO, VOLO, ATTIVITASECONDARIA, HOTEL per aggiungere, rimuovere o editare le componenti; e cliente con permessi di sola lettura sulle tabelle

- **PACCHETTO:** entità identificata da un ID univoco autoincrementale, e caratterizzata da un nome, costo, località, data inizio e fine validità, immagine e descrizione. Inoltre presenta anche un campo eliminato di tipo eliminato, di default 0, che verrà settato a 1 quando il pacchetto viene eliminato (in modo tale che il pacchetto non venga rimosso irreversibilmente dal database e possa essere comunque mantenuto per prenotazioni già effettuate di quel pacchetto). Presenta le seguenti relazioni:

  - VOLOPACCHETTO: collega al pacchetto tutti i voli di andata e di ritorno (almeno due, uno di andata e uno di ritorno) proposti nel pacchetto e compresi proposti nell'arco di tempo di validità dello stesso
  - HOTELPACCHETTO: collega al pacchetto uno e un solo hotel convenzionato nella vacanza
  - ATTIVITASECONDARIAPACCHETTO: collega al pacchetto una o più attività convenzionata nella vacanza

- **PRODOTTO BASE:** entità che specifica le caratteristiche comuni a tutti i prodotti base(volo, hotel, attività secondarie).È identificato da un ID univoco autoincrementale, e caratterizzata da costo, disponibilità(numero intero decrementato ad ogni acquisto del componente) e un attributo elimato,di default 0, che verrà settato a 1 quando la componente viene eliminata (in modo tale che la componente non venga rimossa irreversibilmente dal database, ma non venga più mostrata) . È generalizzazione totale ed esclusiva di VOLO, HOTEL, ATTIVITA' SECONDARIA.

- **VOLO:** entità caratterizzata da città partenza, città arrivo, partenza (data e ora), arrivo (data e ora) e dal nome della compagnia. Un volo può partecipare alla relazione VOLOPACCHETTO, se questo appartiene ai voli proposti in uno o più pacchetti. Un volo può partecipare alla relazione VOLOPRENOTAZIONE, se questo è prenotato come volo di andata o ritorno all' interno di una o più prenotazioni.

- **HOTEL:** entità caratterizzata da località, nome, numero di stelle, breve descrizione e immagine. È associato tramite la relazione HOTELPACCHETTO a uno o più pacchetti, in quanto lo stesso hotel può essere parte di pacchetti diversi. Inoltre un hotel può partecipare alla relazione HOTELPRENOTAZIONE, se questo è prenotato come alloggio all' interno di una o più prenotazioni.

- **ATTIVITA' SECONDARIA:** entità caratterizzata dalla località in cui si svolge, dal nome, e da una breve descrizione. È associata tramite la relazione ATTIVITASECONDARIAPACCHETTO a uno o più pacchetti, in quanto la stessa attività può essere parte di pacchetti diversi. Un'attività può partecipare alla relazione ATTIVITASECONDARIAPRENOTAZIONE, se questa è prenotata come attività all' interno di una o più prenotazioni.

- **PRENOTAZIONE:** entità identificata da un ID univoco autoincrementale, e caratterizzata da una data di creazione, da un numero di persone e dal costo a persona. Partecipa alle seguenti relazioni:

  - UTENTEPRENOTAZIONE: relazione che specifica l'ID (uno e uno solo) del cliente che ha effettuato la prenotazione
  - HOTELPRENOTAZIONE: relazione che specifica l'ID (uno e uno solo) dell' hotel prenotato

- VOLOPRENOTAZIONE: relazione che specifica gli ID dei voli prenotati (esattamente due, uno d'andata e uno di ritorno)
- ATTIVITASECONDARIAPRENOTAZIONE: relazione che specifica l'ID delle attività secondarie prenotate (una o più)

3.2 Progettazione logica

Lo schema concettuale proposto al punto precedente viene tradotto nel conseguente schema logico, che corrisponde alla reale struttura del database relazionale.

Lo schema del database risultante è il seguente:

- UTENTE (id, nome, cognome, indirizzo, email, password)
- UTENTEGRUPPO (id, utente, gruppo)
- PACCHETTO (id, nome, immagine, località, descrizione, inizioValidità, fineValidità, hotel, eliminato)
- PACCHETTOCONDIVISO (id, utente, pacchetto, emailAmico, chiave, stato)
- VOLO (id, disponibilità, costo, cittaPartenza, cittaArrivo, partenza, arrivo, nomeCompagnia, eliminato)
- VOLOPACCHETTO (id, volo, pacchetto, tipo)
- HOTEL (id, nome, disponbilità, stelle, costoGiornaliero, immagine, luogo, descrizione, eliminato)
- ATTIVITASECONDARIA (id, nome, disponibilità, costo, località, descrizione, eliminato)
- ATTIVITASECONDARIAPACCHETTO (id, pacchetto, attivitàSecondaria)
- ATTIVITASECONDARIAPRENOTAZIONE (id, prenotazione, attivitàSecondaria)
- PRENOTAZIONE (id, persone, costoPersona, utente, voloAndata, voloRitorno, hotel, dataCreazione)
- LISTADESIDERI (id, utente, pacchetto, pagatoDa)

Lo schema di traduzione adottato è quello standard.

Le associazioni uno-a-molti sono state tradotte inserendo, all'interno della tabella che partecipa all'associazione dal lato "uno", un'attributo contenente la chiave primaria della tabella cui è collegata, dichiarando gli opportuni vincoli di chiave esterna (ad esempio: il campo hotel in PACCHETTO è chiave esterna del campo id della tabella HOTEL).

Le relazioni molti-a-molti sono state tradotte invece con una tabella avente come attributi le chiavi delle entità coinvolte (esempio: VOLOPACCHETTO ha come campi volo e pacchetto, chiave esterne degli id delle tabelle, rispettivamente, VOLO e PACCHETTO), più gli eventuali attributi della relazione(esempio: LISTADESIDERI oltre ad avere come attributi le chiavi esterne pacchetto e utente, ha anche pagatoDa).

Per quanto riguarda le generalizzazioni, l'unico caso presente nel diagramma E-R è stato tradotto inserendo nello schema delle entità figlie i campi dell' entità padre; data la diversità e la varietà dei campi delle entità figlie, è sembrato opportuno trattarle come tabelle diverse.

Il grafo completo del progetto logico è riportato nella figura sottostante. Nel diagramma sono rappresentati anche i vincoli di chiave esterna tra le tabelle (con una notazione simile a quella utilizzata per le associazioni tra classi in UML) e i tipi di dato utilizzati per ciascun campo.

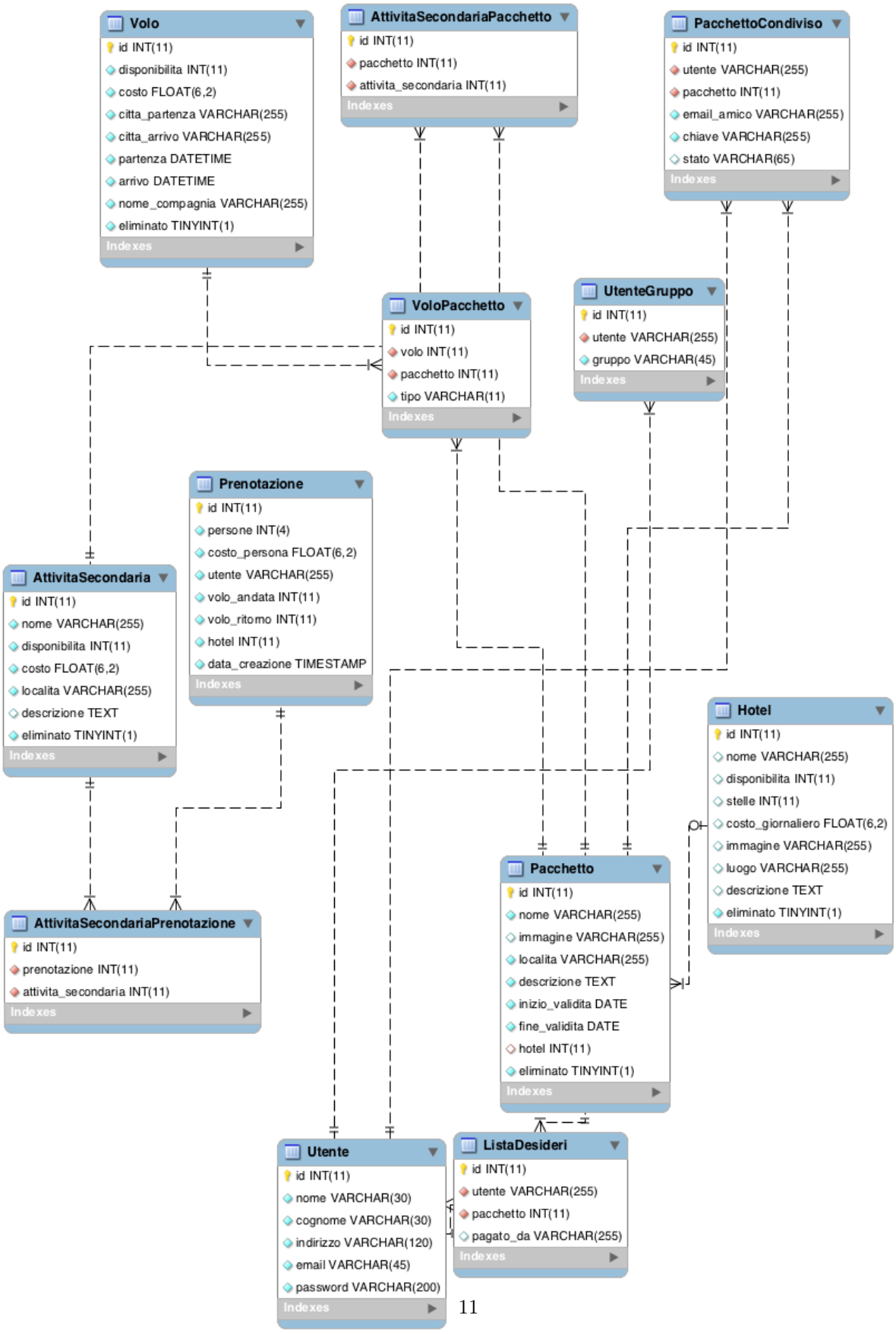

Figure 4: Modello logico del database

Figure 5: Diagramma delle classi

4 Progetto del business tier

Una volta individuati i dati di interesse per il dominio applicativo è stata progettata l'applicazione secondo l'architettura introdotta nella prima sezione e sono state realizzate le componenti software per implementare la logica applicativa.

4.1 Modello (Entity beans)

la struttura dell'insieme delle classi JPA entity(entity beans) astraggono il modello relazionale dei dati basato sul DBMS MySQL. Questa astrazione semplifica notevolmente la gestione dei dati in un linguaggio orientato agli oggetti quale è JEE. Le entity beans rispecchiano esattamente il diagramma UML delle classi riportato di seguito

4.2 Logica Applicativa (EJB)

Per sviluppare la logica applicativa verranno utilizzati soltanto session bean di tipo stateless, in quanto più performanti rispetto a quelli stateful. Questi ultimi infatti non sono necessari all'applicazione in quanto non è importante mantenere lo stato, a livello applicativo, all'interno della sessione di interazione tra utente e sistema ad eccezione del mantenimento dell'autenticazione. Questo compito verrà demandato al web tier. Il compito, invece, di validazione delle credenziali e dell'assegnamento del gruppo di privilegio di appartenenza, è affidato alla logica apllicativa.

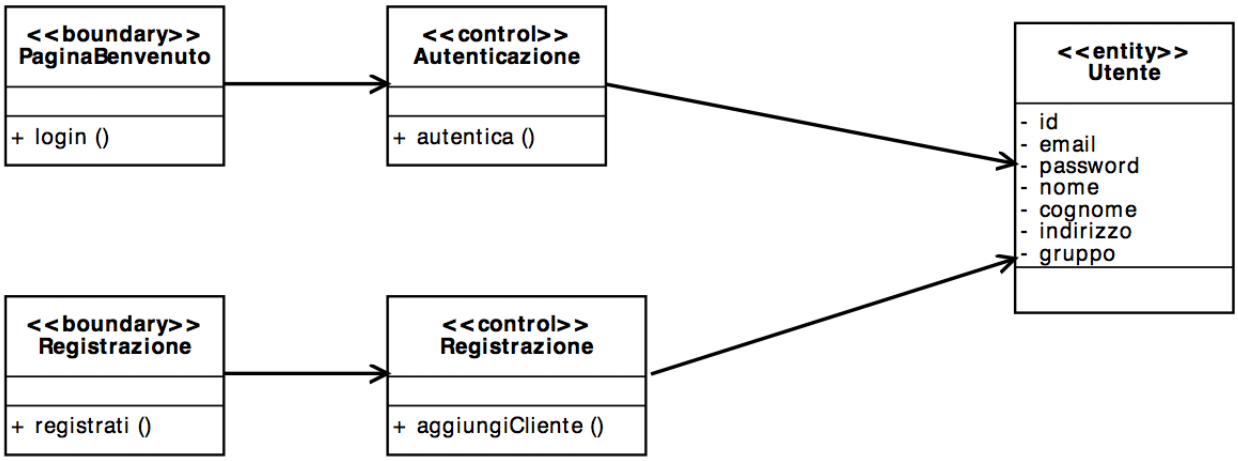

Figure 6: BCE: Utente non registrato o loggato

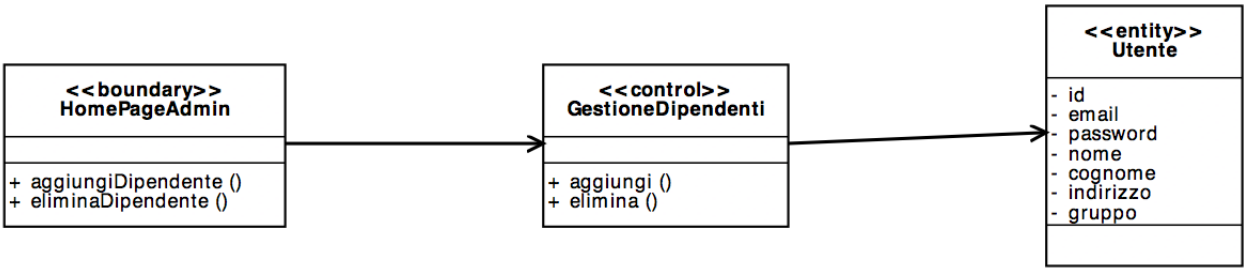

Figure 7: BCE: Amministratore

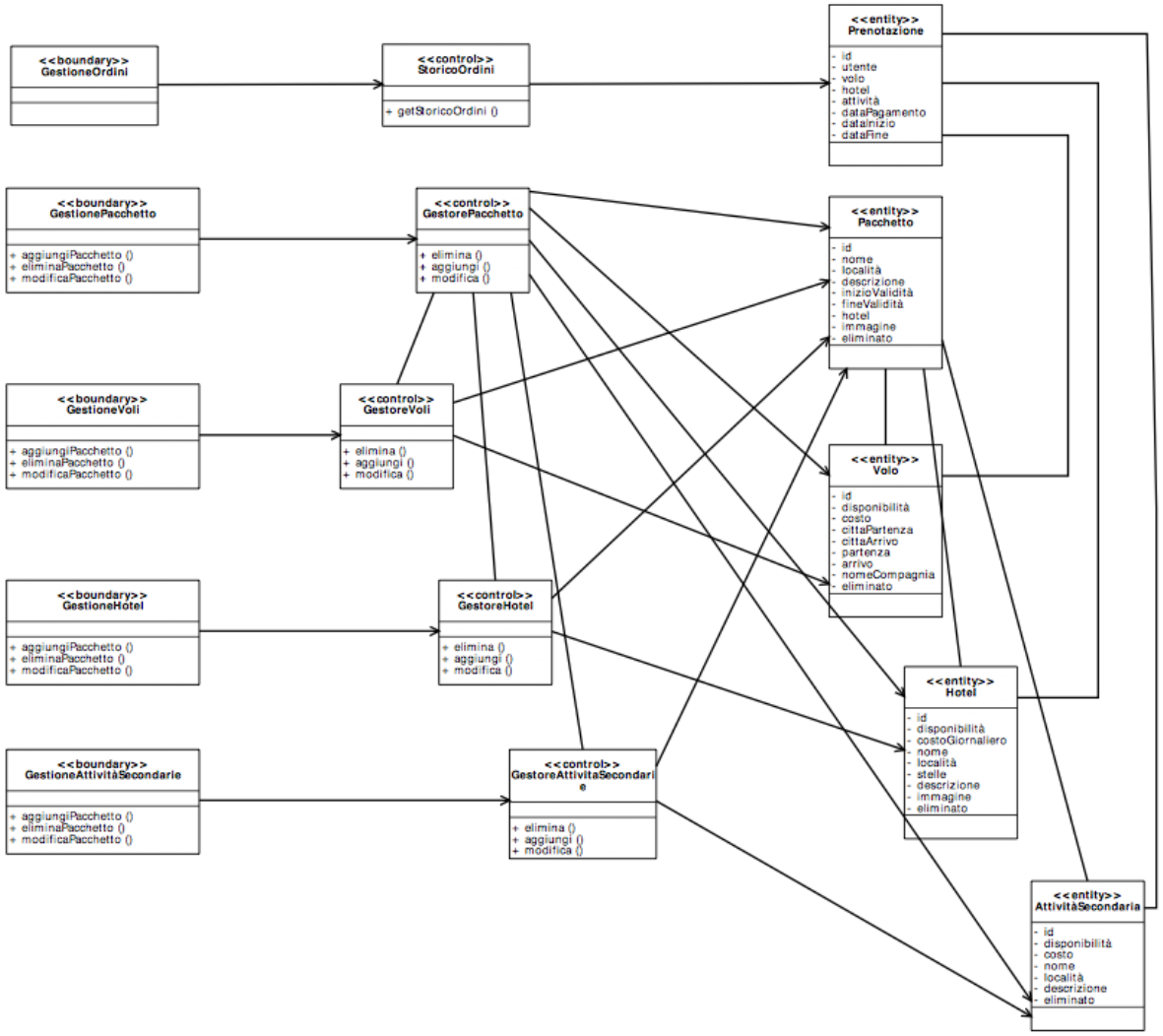

Figure 8: BCE: Dipendente

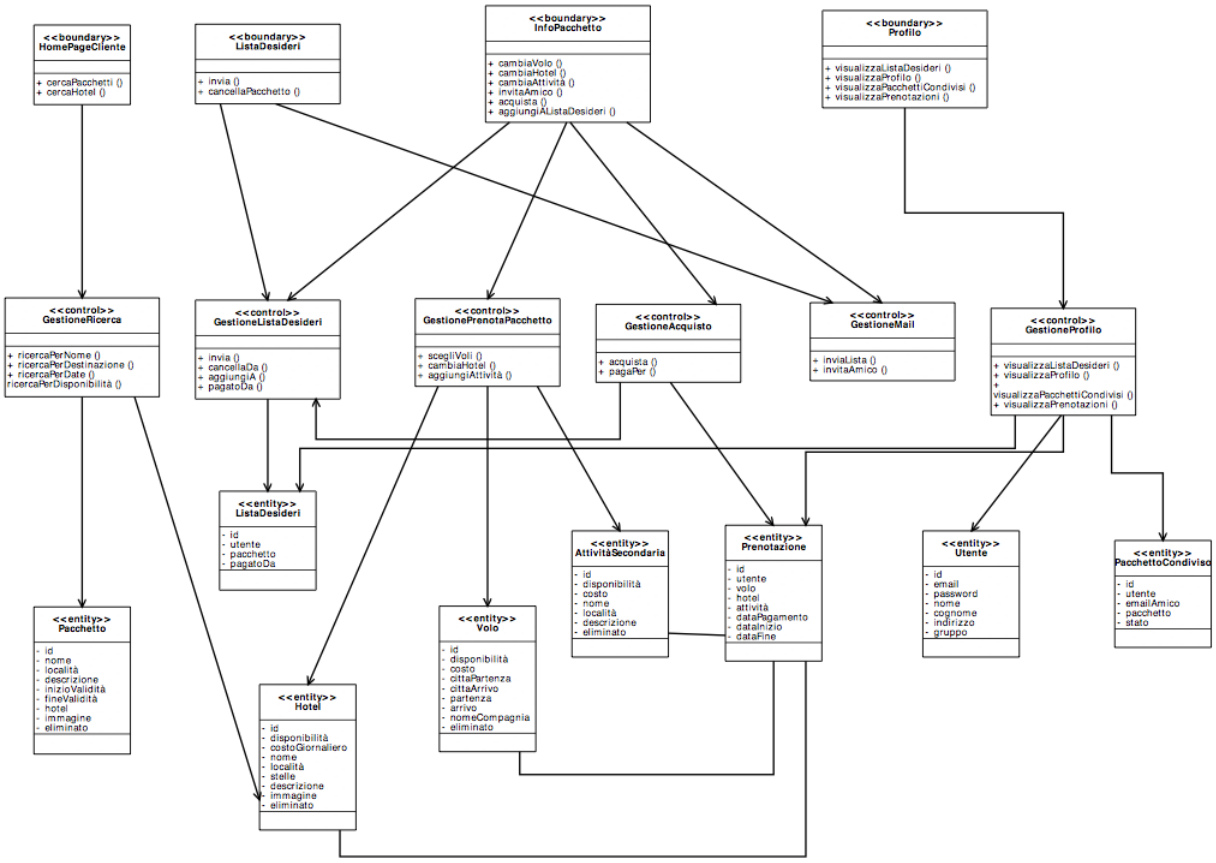

Figure 9: BCE: Cliente

5 Progetto del client

5.1 Progetto della navigazione

Di seguito è trattata la progettazione dell'interfaccia utente e sono mostrati modelli UX per definire i percorsi di navigazione tra le pagine e come queste interagiscono tra loro.

Per non appesantire inutilmente la notazione, i diagrammi presentati di seguito sono separati per tipologia di utente.

Il primo diagramma presentato è relativo alla porzione dell'applicazione accessibile a tutti tramite il bowser: la pagina di benvenuto.

Un utente che accede alla pagina di benvenuto del sito non può fare altro che registrarsi o loggarsi.

In particolare, come rappresentato graficamente nel diagramma, la <<screen>> pagina di benvenuto contiene una <<input form>>, che richiede email e password, per effettuare il login.

Una volta effettuato il login il sistema verifica se le credenziali immesse sono di un cliente registrato, di un dipendente o dell' amministratore, e in base a questo apre schermate appropriate al tipo di utente, rispettivamente HomePageCliente, HomePageDipendente o HomePageAdmin, le cui funzionalità e pagine saranno rappresentate nei diagrammi successivi.

Dalla pagina di benvenuto l'unica altra alternativa è premere il pulsante registrati, in seguito al quale si apre la schermata "registrazione" composta di una form da riempire con i dati. Confermando i dati, in caso di dati validi e quindi di una registrazione avvenuta con successo, si viene reindirizzati alla pagina di benvenuto, dalla quale è ora possibile accedere.

Banalmente, da qualsiasi area o pagina, una volta loggati, è possibile effettuare il logout e tornare alla pagina di benvenuto.

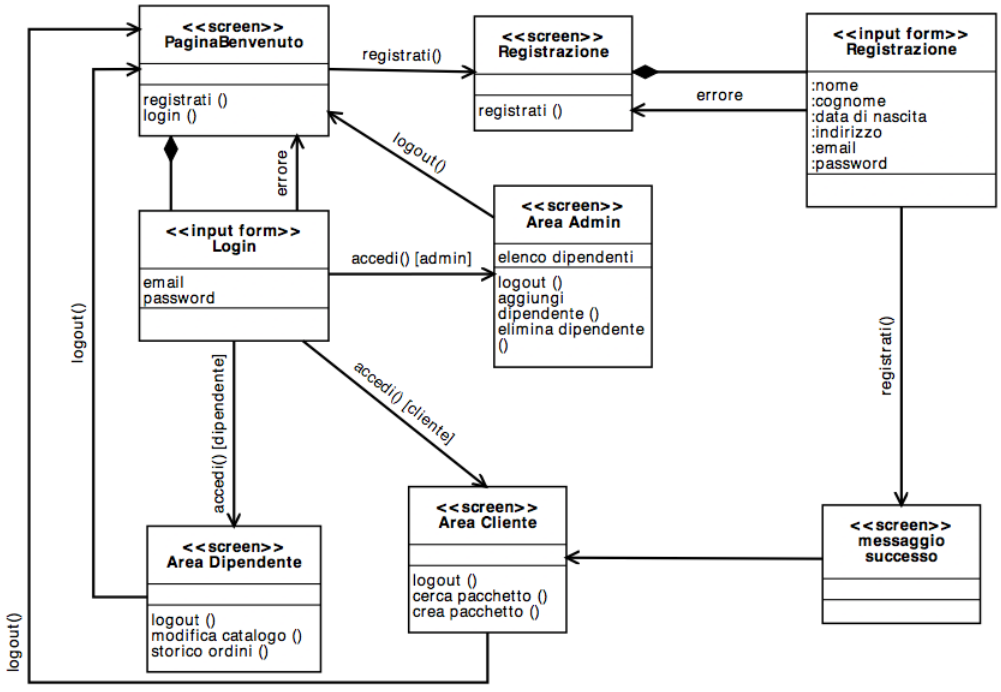

Figure 10: Diagramma UX: funzionalità per utenti non registrati o non loggati

Gli altri diagrammi UX rappresentano le funzionalità dell'applicazione accessibili previa autenticazione.

Il secondo rappresenta in dettaglio l' "AreaAdmin" prima nominata, ossia le funzionalità offerte all'amministratore una volta loggatosi.

La pagina iniziale cui accede l'amministratore è costituita da un elenco di dipendenti.

Le opzioni offerte all' amministratore sono aggiungiDipendente, che apre una pagina con una form da compilare con i dati e le credenziali del nuovo impiegato e in seguito alla conferma ritorna la lista dei dipendenti aggiornata, e eliminaDipendente, che richiede conferma per la cancellazione del cliente.

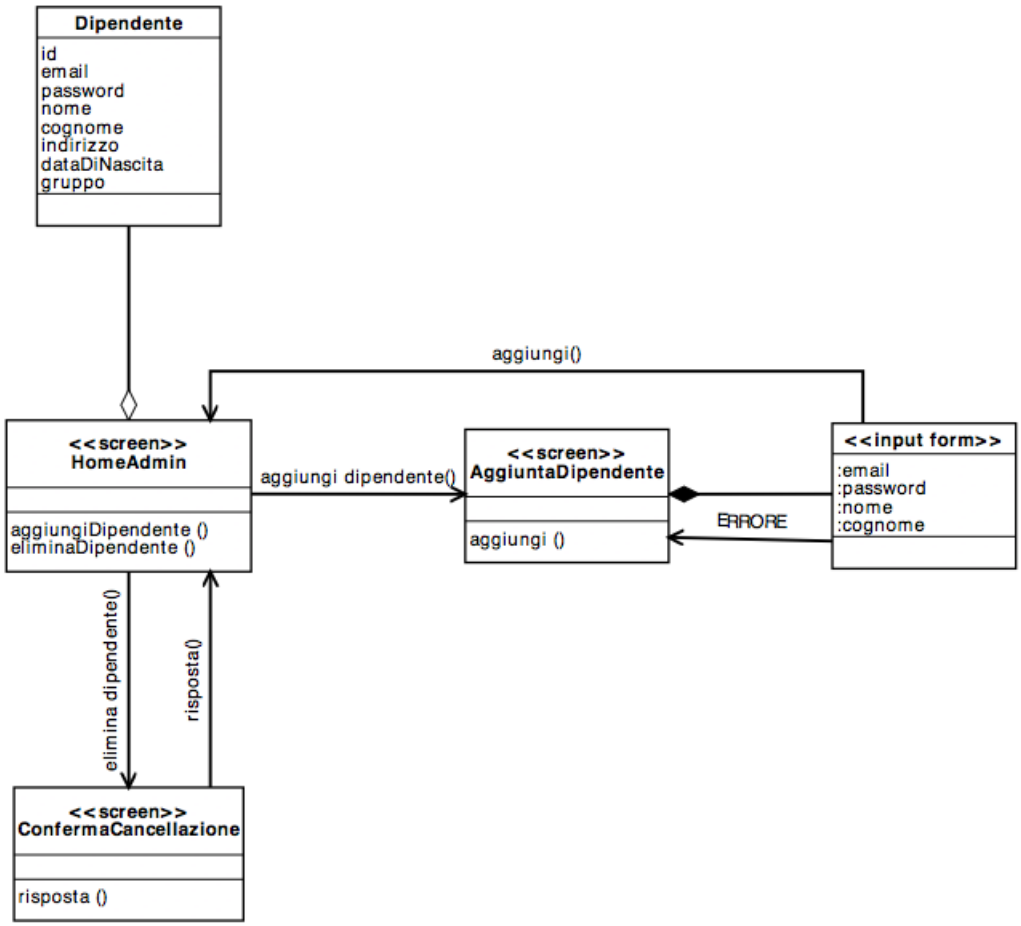

Figure 11: Diagramma UX: AreaAdmin

Il terzo diagramma rappresenta in dettaglio l' "HomePageDipendente" nominata nel primo diagramma, ossia le funzionalità offerte ai dipendenti una volta loggatisi.

Per non appesantire inutilmente la notazione, i diagrammi UX inerenti il lato dipendente sono stati divisi in gestione pacchetti e gestione componenti base.

Ogni pagina dell' area dipendente presenta una toolbar con i seguenti link: home, gestione pacchetti, gestione voli, gestione hotel, gestione attività, gestione ordini. Queste sezioni, proprio perchè potranno essere raggiunte in qualsiasi momento, da qualsiasi pagine, sono caratterizzate dal simbolo $.

Con gestione pacchetti si accede all' elenco dei pacchetti, lista organizzata in una tabella, su cui si possono applicare le operazioni disponibili di aggiunta, modifica, cancellazione.

Ogni riga della tabella (ogni pacchetto), presenta i bottoni per la modifica e la cancellazione del pacchetto.

Premendo il tasto di modifica si accede alla schermata "ModificaPacchetto" costruita in tal modo: stessa tabella di tutti i pacchetti ma con la riga del pacchetto da modificare sovrascrivibile. I campi sono riempiti con i dati del pacchetto attuale, ma possono essere modificati.

Premendo invece il tasto di cancellazione, viene richiesta una conferma.

Entrambe le istruzioni, dopo le esecuzioni, ritornano la pagina con l'elenco aggiornato.

Infine in questa pagina è presente l'opzione "aggiungi pacchetto" che coinvolge una serie di schermate:

"AggiungiInfoGeneraliPacchetto": che permette l'inserimento di nome, descrizione, località, immagine, tutti dati obbligatori per passare alla schermata successiva;

"AggiungiVolo": che obbliga l'inserimento di almeno un volo d'andata e uno di ritorno, scegliendoli tra quelli esistenti oppure inserendoli nuovi;

"AggiungiHotel": che obbliga l'inserimento di un hotel, scegliendolo tra quelli esistenti oppure inserendolo nuovo;

"AggiungiAttività": che obbliga l'inserimento di almeno un' attività, scegliendola tra quelli esistenti oppure inserendola nuovo.

Completando tutte le schermate, e concludendo positivamente il processo di creazione del pacchetto, viene ritornato l' elenco dei pacchetti aggiornato.

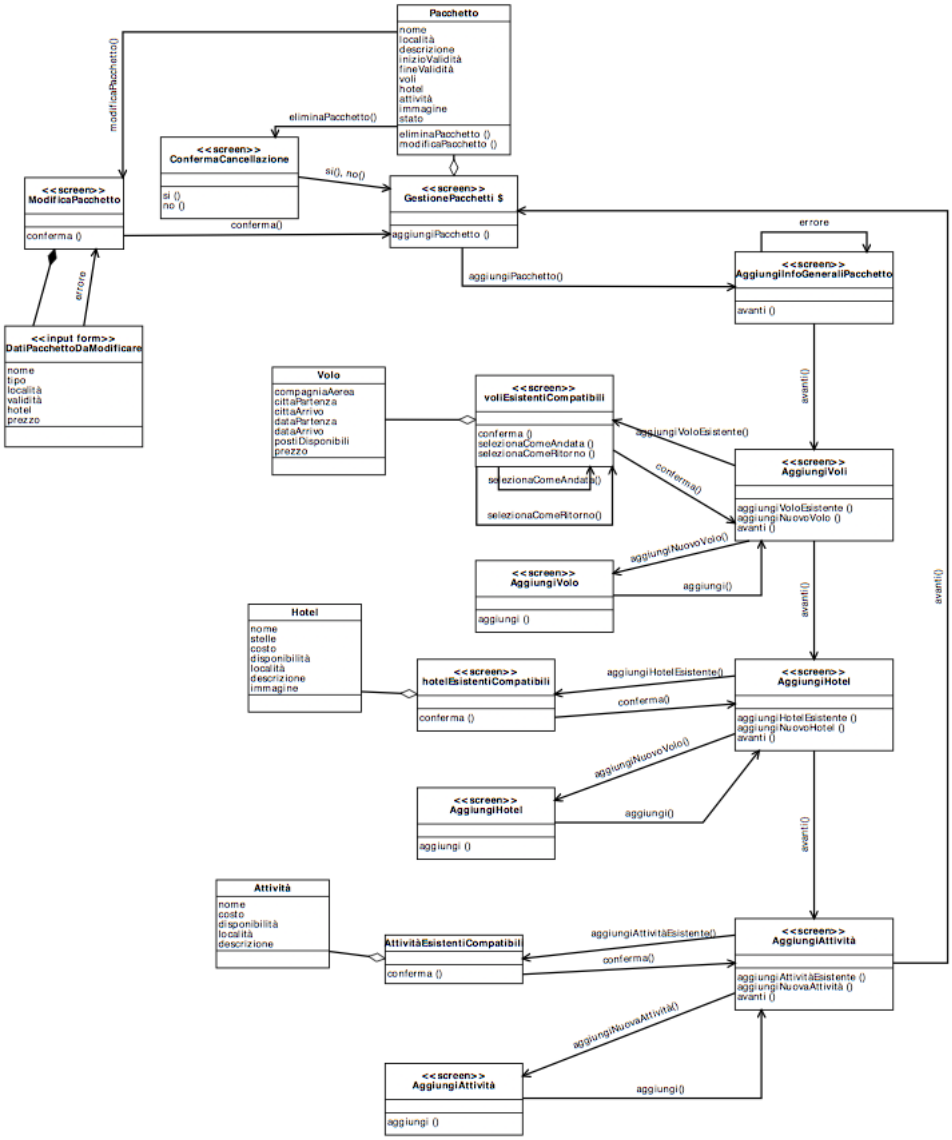

Figure 12: Diagramma UX: AreaDipendente_GestionePacchetti

Il quarto diagramma completa la rappresentazione del sito lato dipendente mostrando la gestione dei componenti base.

In particolare, tralasciando gestione pacchetti già discussa, il dipendente può accedere alle seguenti sezioni e schermate:

"GestioneOrdini": che mostra l' elenco di tutte le prenotazioni che sono state effettuate;

"GestioneVoli": che permette l' inserimento di un nuovo volo tramite la schermata "CreazioneVolo", la cancellazione di un volo esistente, dopo una conferma su "ConfermaCancellazione", e la modifica di un volo esistente attraverso la schermata "ModificaVolo" contentente i campi sovrascribili e modificabili;

"GestioneHotel": che permette l' inserimento di un nuovo hotel tramite la schermata "CreazioneHotel", la cancellazione di un hotel esistente, dopo una conferma su "ConfermaCancellazione", e la modifica di un hotel esistente attraverso la schermata "ModificaHotel" contentente i campi sovrascribili e modificabili;

"GestioneAttivitàSecondarie": che permette l' inserimento di una nuova attività tramite la schermata "CreazioneAttività", la cancellazione di un' attività esistente, dopo una conferma su "ConfermaCancellazione", e la modifica di un' attività esistente attraverso la schermata "ModificaAttività" contentente i campi sovrascribili e modificabili;

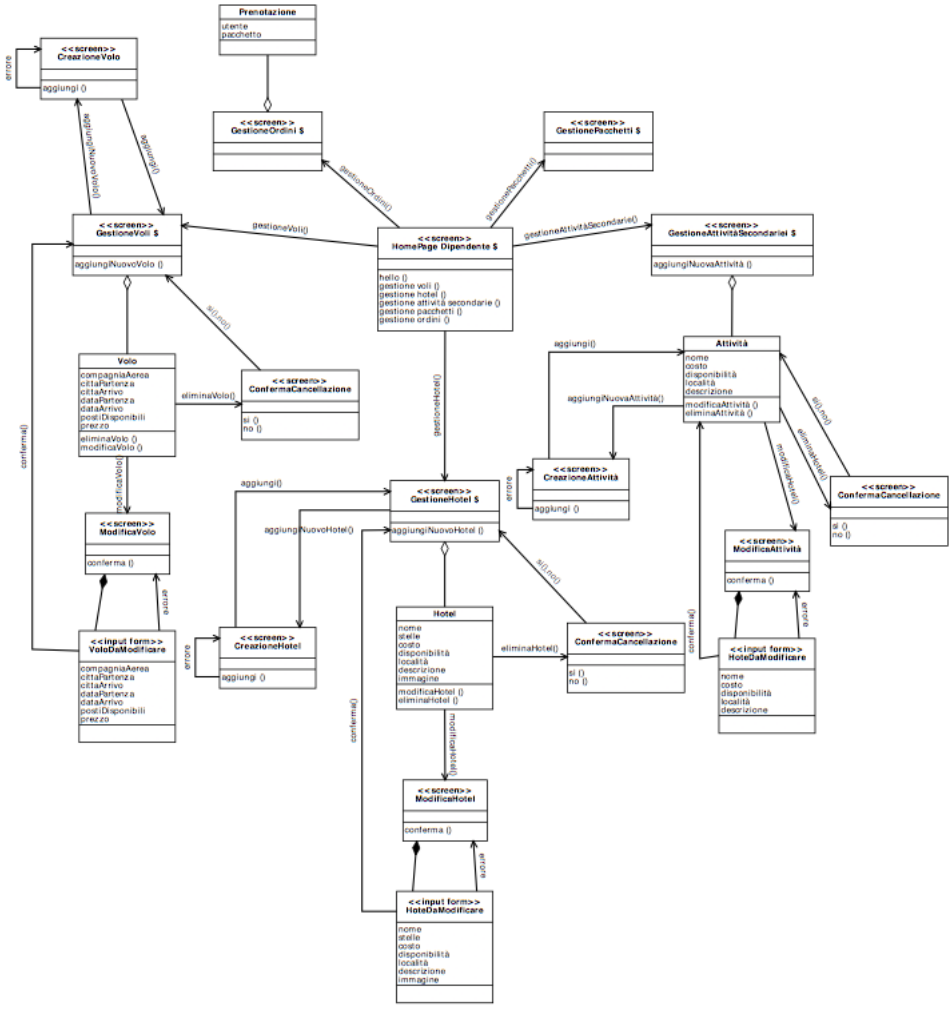

Figure 13: Diagramma UX: AreaDipendente_GestioneComponentiBase

I clienti accedono all'homepage del sito, pagina in cui possono sempre tornare, e che fornisce le operazioni disponibili ai clienti.

Per non appesantire inutilmente la notazione, i diagrammi UX inerenti il lato cliente sono stati divisi in gestione pacchetti e gestione profilo.

Il cliente può accedere alle schermate "ListaHotel" e "listaPacchetti" contenenti tutti gli hotel e tutti i pacchetti, premendo rispettivamente hotels e packages da homepage, oppure può filtrare queste liste attraverso dei motori di ricerca "RicercaHotel" e "RicercaPacchetti" direttamente da hompage, oppure nella rispettiva sezione.

Ottenuta la lista di pacchetti, intera o scremata in base a qualche criterio di ricerca (nome/luogo, date) è possibile premere il pulsante "select" di fianco a qualsiasi pacchetto per avere maggiori informazioni, e accedendo così alla pagina "detailsPacchetto".

Da quest' ultima è possibile effettuare tutte le operazioni sul pacchetto "sendToFriend", "addToWishList"e "bookNow" che permette successivamente di settare le date e il numero di persone, di cambiare l' hotel, aggiungere qualche attività e scegliere tra i voli disponibili.

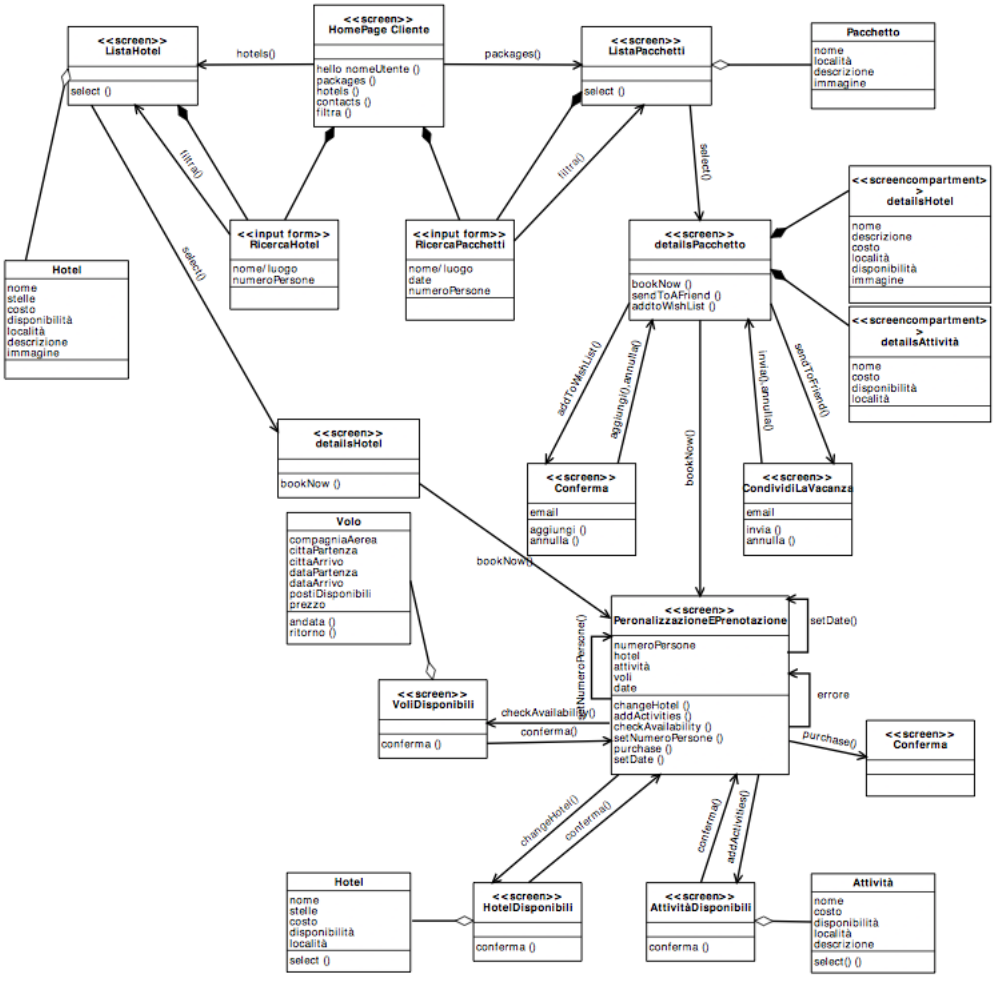

Figure 14: Diagramma UX: AreaCliente_GestionePacchetti

Al cliente è fornita anche una pagina personle, cui può accedere premendo “HelloUtente” nella toolbar presente nell’ header di ogni pagina, e successivamente “yourProfile”.

Dalla schermata “PaginaUtente”, il cliente può visualizzare le sue prenotazioni, premendo “bookings”, visualizzare la sua lista desideri e modificarla, premendo “wish list”, e visualizzare i pacchetti che ha condiviso con amici e lo stato dell’ invito, premendo “shared packages”.

Infine può accedere alla schermata “Contacts”, premendo il link “contacts” presente nella toolbar, e visulizzare i dati dell’ azienda e contattarla in caso di problemi.

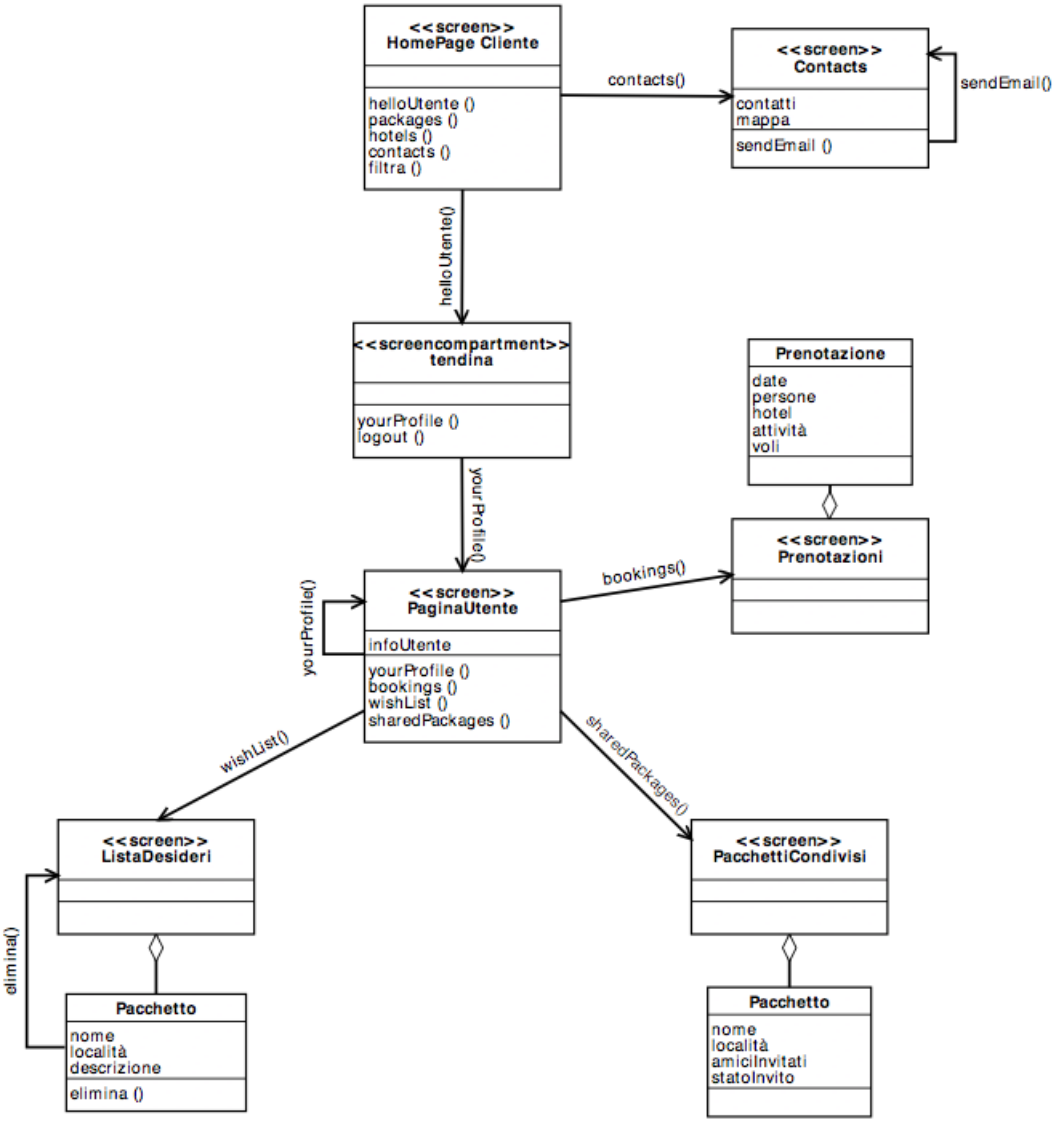

Figure 15: Diagramma UX: AreaCliente_GestioneProfilo