



Laurea Magistrale in informatica-Università di Salerno
Corso di Ingegneria Gestione Evoluzione del Software

Università degli Studi di Salerno

Corso di Ingegneria Gestione Evoluzione del Software



Maintenance Report

2021/2022

Antonio De Matteo

Vincenzo Pecoraro



Sommario

1. INTRODUZIONE	3
2. ARCHITETTURA DEL SISTEMA ATTUALE	3
2.1 Panoramica.....	3
2.2. Decomposizione in sottosistemi.....	4
2.2.1 Architettura del sistema e sottosistemi individuati.....	4
2.2. Gestione dati persistenti	5
3. DESCRIZIONE DELLE MODIFICHE CHANGE REQUEST 1.....	6
3.1 Analisi della richiesta di modifica	6
3.2 Analisi di impatto.....	7
3.3 Metriche di Impact Analisys	19
4. TESTING	Errore. Il segnalibro non è definito.
Approccio	Errore. Il segnalibro non è definito.
1. Testing di unità:.....	Errore. Il segnalibro non è definito.
2. Testing di integrazione:	Errore. Il segnalibro non è definito.
3. Testing di sistema:	Errore. Il segnalibro non è definito.
Testing di Unità	Errore. Il segnalibro non è definito.
Testing di Integrazione	Errore. Il segnalibro non è definito.
Testing di Sistema.....	Errore. Il segnalibro non è definito.
Criteri di ripresa.....	Errore. Il segnalibro non è definito.
Materiale per il testing	Errore. Il segnalibro non è definito.
1. Descrizione del documento.....	Errore. Il segnalibro non è definito.
2. Glossario.....	Errore. Il segnalibro non è definito.



1. INTRODUZIONE

A seguito delle CR presentate al docente e ai tutor, il lavoro di manutenzione effettuato sull'applicativo “WhereDoIEat” si è concentrato nello studio del dominio di analisi e progettazione del sistema attuale, in modo tale da individuare le componenti impattate dalla richiesta di modifica ed eventuali componenti aggiuntive per adattare il nuovo flusso operativo del sistema a seguito delle modifiche che si vogliono apportare.

Per migliorare il sistema, si è ritenuto necessario apportare tre change request:

1. Data questa CR si evince la necessità di semplificare il lavoro dell’utente nel cercare velocemente un ristorante di proprio interesse. Le modifiche che verranno apportate a tal proposito nel sistema saranno:
 - a. Possibilità di cercare un locale filtrando per la categoria dello stesso.
 - b. Inserimento di un locale nella propria sezione “preferiti”.

2. ARCHITETTURA DEL SISTEMA ATTUALE

Al fine di comprendere il funzionamento del sistema prima della modifica sono state riportate le principali caratteristiche della struttura.

2.1 Panoramica

Il sistema attuale, consiste in una web app in cui sono previste diverse funzionalità divisi per due tipi di utenti:

- Ristoratori
- Clienti



nel dettaglio:

- Il ristoratore può registrare la propria attività sulla piattaforma WhereDoIEat e ricevere recensioni da parte degli utenti che la visitano.
- Il cliente può accedere alla piattaforma e valutare le varie recensioni per decidere quale ristorante provare, dopodiché ha la possibilità di prenotare un tavolo, anche se con difficoltà nell'UX

Ogni tipologia di utente registrato avrà, inoltre, una visualizzazione mirata al ruolo ricoperto nel sistema WhereDoIEat.

2.2. Decomposizione in sottosistemi

2.2.1 Architettura del sistema e sottosistemi individuati

Come specificato nella definizione teorica del pattern MVC, la decomposizione del nostro sistema è composta da tre layer aventi ciascuna aspetti e funzionalità differenti:

- **View:** layer di presentazione delle funzionalità del sistema e relativo interfacciamento con l'utenza;
- **Controller:** layer che si occupa della gestione della logica di business del sistema e relativi flussi di controllo dei dati;
- **Model:** layer che si occupa della gestione dei dati persistenti e relativo interscambio degli stessi tra i sottosistemi.

Oggetti **Boundary** individuati nella fase di analisi definiamo un sottosistema view:

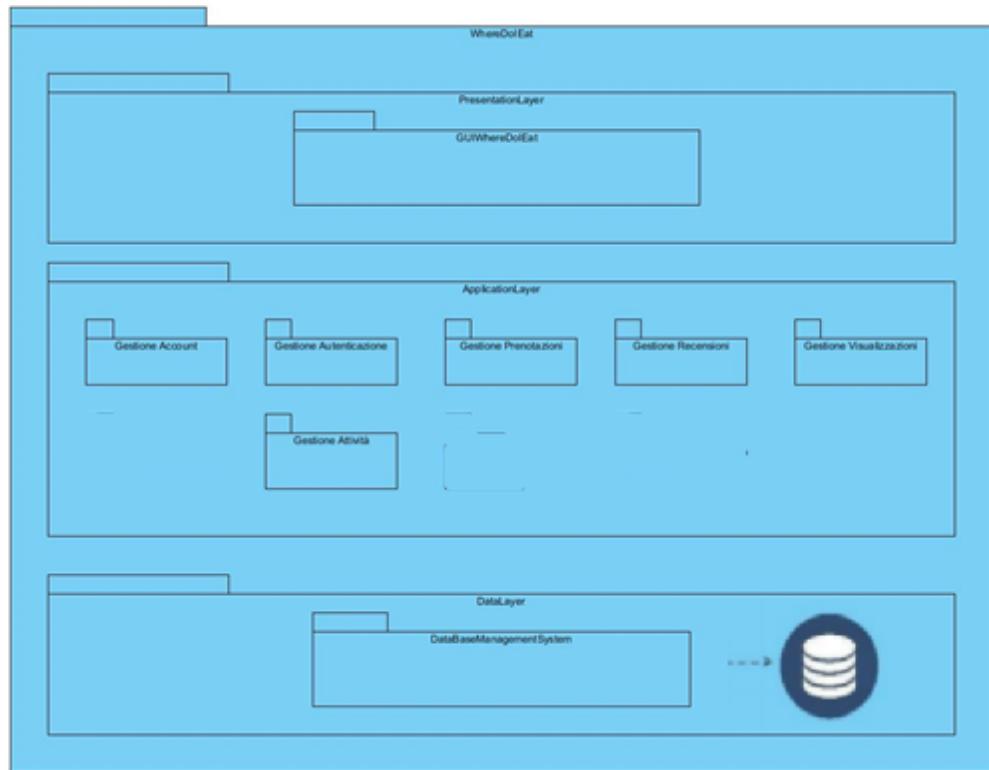
- GUIWhereDoIEat (sottosistema per la gestione dell'interfacciamento con l'utente).

Oggetti **Controller** individuati nella fase di analisi definiamo sei sottosistemi:

- GestioneAccount (sottosistema per la visualizzazione e modifica dei dati dell'utente e modifica della password);
- GestioneAutenticazione (sottosistema per la gestione del login, logout e Sign-in da parte dell'utente);
- GestionePrenotazioni (sottosistema per la gestione della prenotazione);
- GestioneRecensioni (sottosistema per la gestione dell'assegnazione di valutazioni e recensioni da parte dell'utente ad un'attività);
- GestioneVisualizzazioni (sottosistema per la visualizzazione di attività, recensioni e mappa);
- GestioneAttività (sottosistema per la gestione di tutte le operazioni e i dati relativi alle attività)

Da un'analisi degli oggetti **Model** individuati nella fase di analisi definiamo un sottosistema:

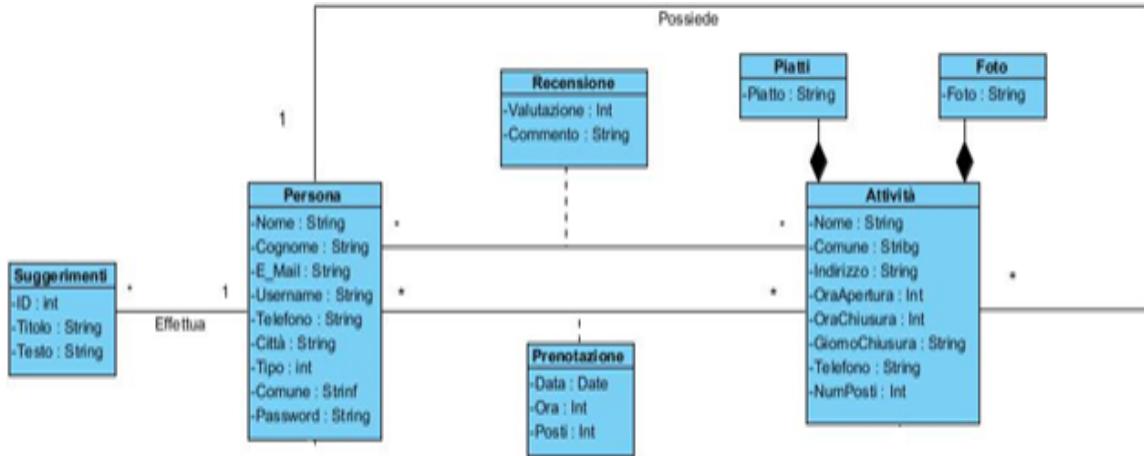
- DataBaseManagementSystem.



2.2. Gestione dati persistenti

Il sistema attuale gestisce la memorizzazione di dati persistenti tramite l'utilizzo di un database relazionale per consentire un accesso veloce e sicuro ai dati e un ampio spazio di archiviazione. Grazie all'uso del DBMS i dati sono protetti quindi diversi utenti con differenti operazioni possono accedere a diverse sezioni del database.

I dati persistenti sono organizzati nel seguente modo:



3. DESCRIZIONE DELLE MODIFICHE CHANGE REQUEST 1

La change request 1 richiede una manutenzione evolutiva, in quanto, introduce una serie di nuove funzionalità al sistema come la ricerca per categoria e la possibilità di aggiungere un'attività ai preferiti.

3.1 Analisi della richiesta di modifica

La prima richiesta di modifica introduce quindi due nuove funzionalità all'interno del sistema che andranno a semplificare al cliente la gestione e la ricerca delle sue attività ristorative.

Quindi, a differenza dell'attuale workflow, un cliente non dovrà più ricercare il ristorante di proprio gradimento scorrendo una lista, potenzialmente molto lunga, di locali ma potrà:

- Selezionare ,nella navbar presente in ogni pagina, la categoria di ristoranti più gradita
- Accedere alla pagina di un ristorante ed aggiungerlo alla lista dei propri ristoranti preferiti premendo l'apposito bottone
- Visualizzare la lista dei ristoranti preferiti

Volendo quindi aggiungere queste nuove funzionalità si rende necessario analizzare le componenti che saranno impattate dalla suddetta modifica e valutarne quanto impatta. Si procederà con l'analisi degli artefatti impattati seguendo un approccio top-down, quindi partendo dalla documentazione fino ad arrivare alle singole classi.



3.2 Analisi di impatto

Per implementare le modifiche al sistema, il lavoro è partito con lo studio del workflow per la prenotazione di un tavolo presso un determinato ristorante.

3.2.1 Caso D'uso

Descrizione	Ricerca per categoria
Attore Principale	Utente L'utente può cercare un ristorante per categoria
Entry Condition	L'utente deve essere loggato al sistema
Exit Condition On Success	Lista dei ristoranti aventi la categoria scelta dall'utente
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO	
1	Cliente: Effettua autenticazione al sistema e vuole ricercare un ristorante per categoria, quindi clicca sul menu dropdown nell'header.
2	Sistema: Mostra un elenco di categorie appartenenti ai ristoranti registrati sul sistema.
3	Cliente: Clicca sulla categoria desiderata
4	Sistema: Mostra una lista dei ristoranti aventi la categoria selezionata



Descrizione	Aggiunta ristorante	
Attore Principale	Utente	
Entry Condition	L'utente deve essere loggato al sistema	
Exit Condition On Success	Il ristorante è aggiunto al sistema	
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Cliente:	Il ristoratore preme sulla voce "Aggiungi attività"
2	Sistema:	Mostra form con dati necessari per l'aggiunta.
3	Cliente:	Inserisci dati, inclusa la categoria
4	Sistema:	Inserisce il ristorante registrato

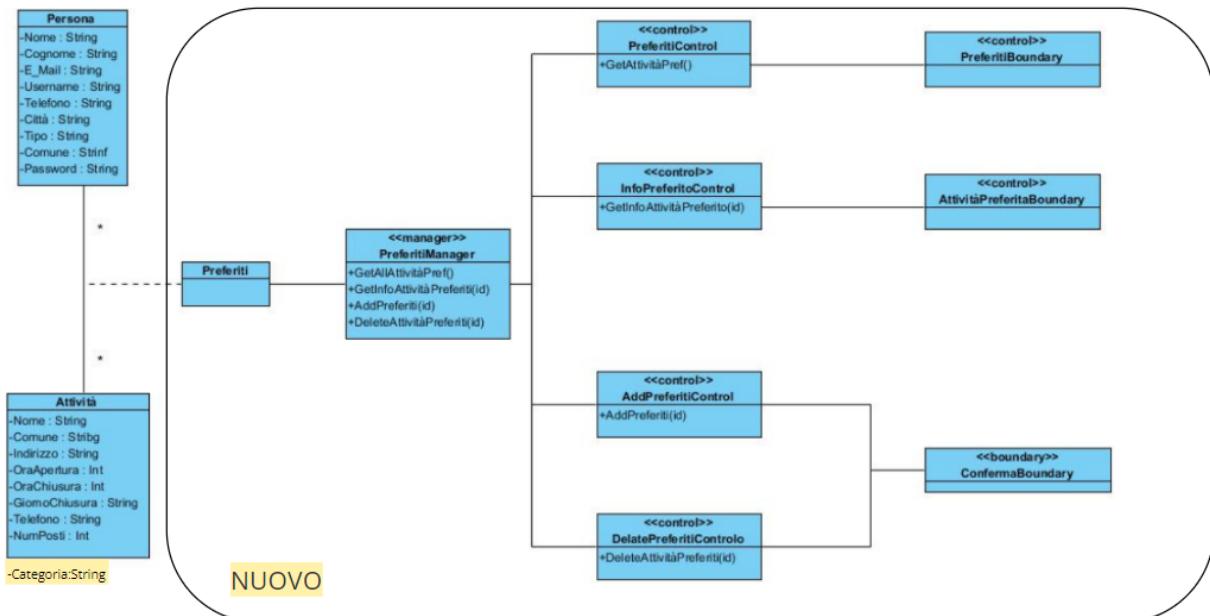


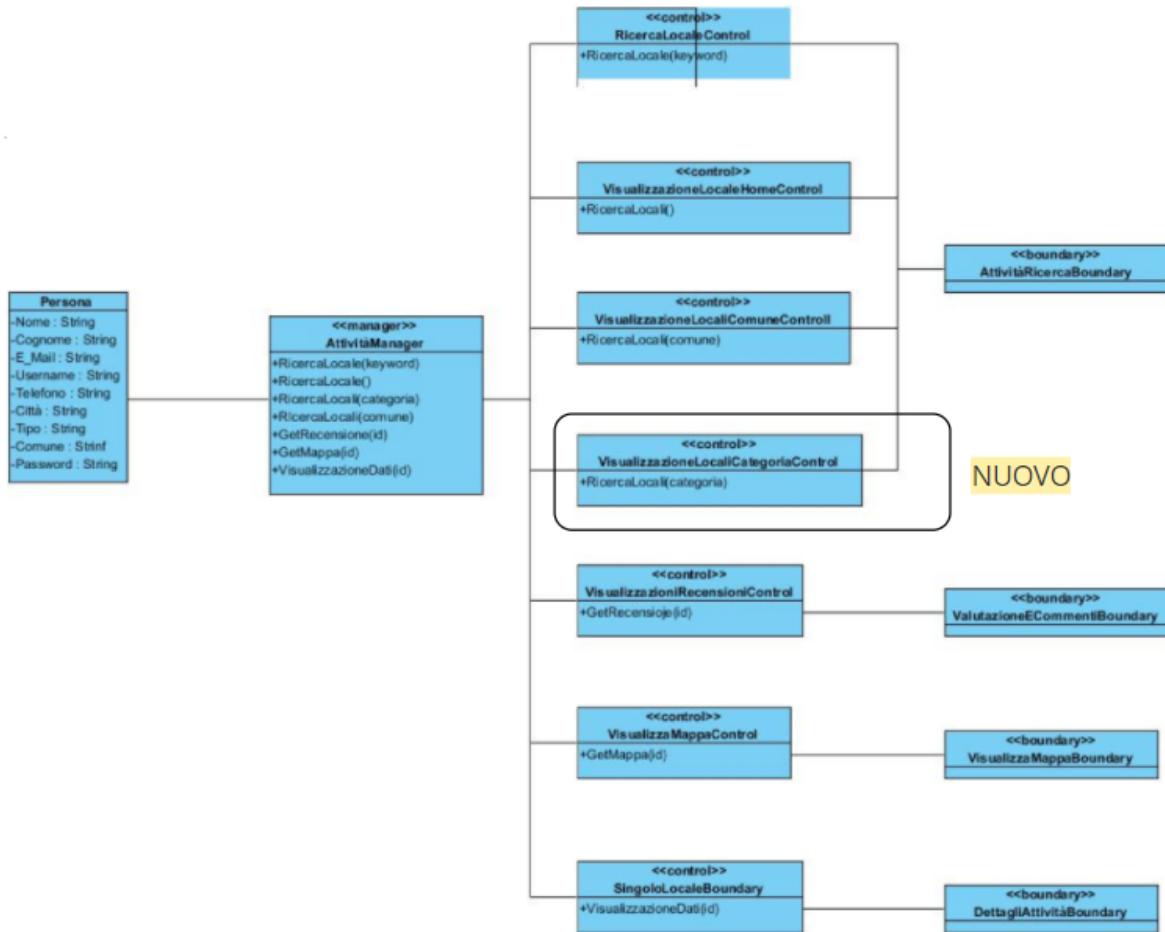
Descrizione		Aggiunta ristorante ai preferiti
Attore Principale		Utente L'utente può inserire un ristorante nella propria lista dei preferiti
Entry Condition		L'utente deve essere loggato al sistema
Exit Condition On Success		Il sistema salva nella lista dei ristoranti preferiti, il ristorante selezionato dall'utente
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Cliente:	Effettua autenticazione al sistema e vuole salvare un ristorante nei propri preferiti. Quindi, cerca il ristorante che gli aggrada di più.
2	Sistema:	Mostra la pagina del ristorante scelto.
3	Cliente:	Clicca sul pulsante di aggiunta ai preferiti
4	Sistema:	Salva nella lista dei preferiti dell'utente il ristorante selezionato.



Descrizione		Visualizzazione Preferiti
Attore Principale		Utente L'utente può inserire un ristorante nella propria lista dei preferiti
Entry Condition		L'utente deve essere loggato al sistema
Exit Condition On Success		L'utente visualizza i propri preferiti
FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO		
1	Cliente:	L'utente clicca sulla voce che si riferisce ai propri preferiti
2	Sistema:	Mostra preferiti precedentemente selezionati

3.2.2 Class Diagram







3.2.3 MOCKUP

Ricerca ristorante - PRIMA

WhereDoIEat

Dove vuoi mangiare? ▾ Cerca Search

Visita i migliori locali

Scopri i migliori locali secondo la nostra clientela, ovvero quelli con le miglior valutazioni in questo momento.

Consigliati per te!

Ricerca ristorante - DOPO

WhereDoIEat

Cosa vuoi mangiare? ▾ Dove vuoi mangiare? ▾ Cerca Search

- Ristorante
- Pizzeria
- Rosticceria
- Pub
- FastFood
- Vegan

Visita i migliori locali

Scopri i migliori locali secondo la nostra clientela, ovvero quelli con le miglior valutazioni in questo momento.

Consigliati per te!



Lista Preferiti - PRIMA

WhereDoIEat

Suggerimenti | Logout | I miei preferiti | Area Personale | Le mie attivita | Aggiungi attivita

Cosa vuoi mangiare? ▾ Dove vuoi mangiare? ▾ Cerca Search

PRODOTTI PREFERITI DI Amedeo

El pampa Sosa

Lista Preferiti – DOPO

WhereDoIEat

Accedi | Registrati ora

Cosa vuoi mangiare? ▾ Dove vuoi mangiare? ▾ Cerca Search

Ristorante

El pocho di Salerno
Salerno



Aggiunta preferiti – PRIMA

WhereDoIEat

Suggerimenti | Logout | I miei preferiti | Area Personale | Le mie attivita | Aggiungi attivita

Cosa vuoi mangiare? ▾ Dove vuoi mangiare? ▾ Cerca Search

El pampa Sosa

13:00-23:00 Chiuso di Mercoledì

I piatti consigliati del locale:
gnocchi al forno: gnocchi cotti al forno per 10 minuti
pizza: pizza cotta nel forno a legna
risotto: Risotto tipico della zona con funghi paesani

Come raggiungere il locale:

prenota

Aggiunta preferiti – DOPO

WhereDoIEat

Suggerimenti | Logout | I miei preferiti | Area Personale | Le mie attivita | Aggiungi attivita

Cosa vuoi mangiare? ▾ Dove vuoi mangiare? ▾ Cerca Search

Aggiungi ai preferiti

El pampa Sosa

13:00-23:00 Chiuso di Mercoledì

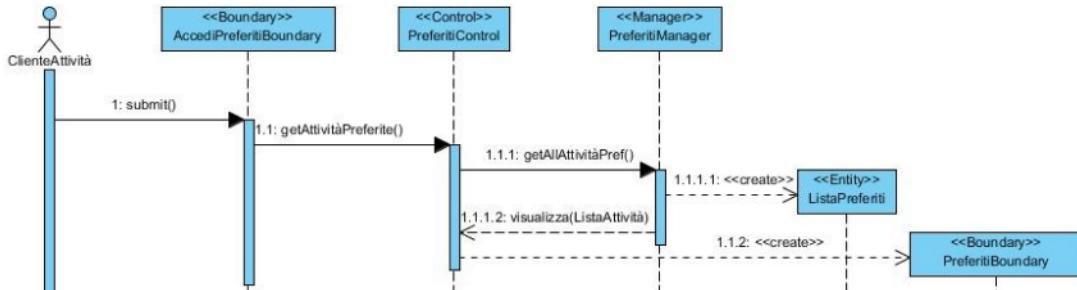
I piatti consigliati del locale:
gnocchi al forno: gnocchi cotti al forno per 10 minuti
pizza: pizza cotta nel forno a legna
risotto: Risotto tipico della zona con funghi paesani

Come raggiungere il locale:

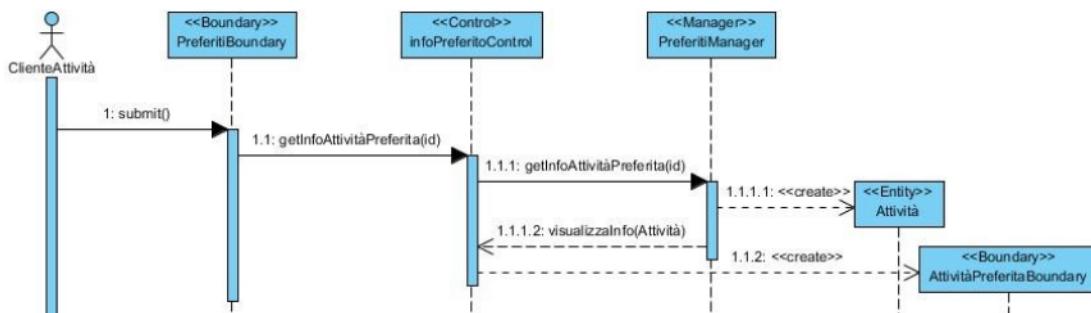
prenota

3.2.4 SEQUENCE DIAGRAM

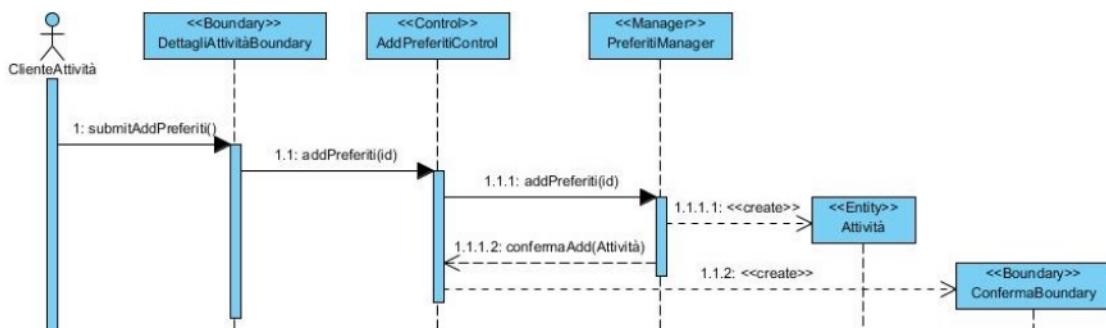
VISUALIZZA PREFERITI



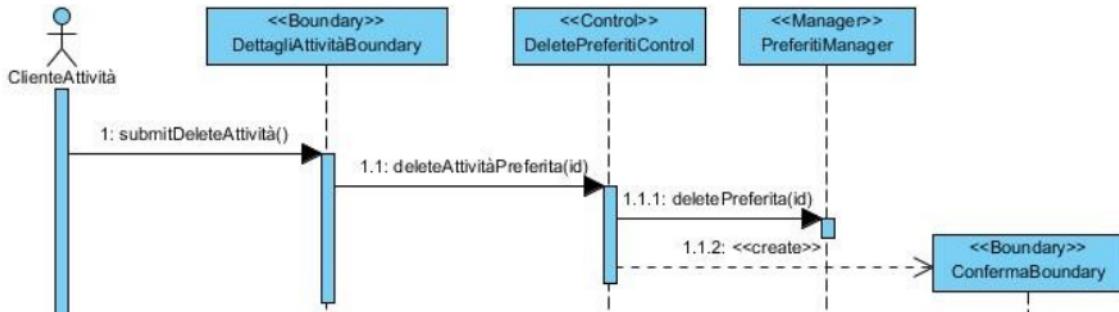
VISUALIZZA DETTAGLI PREFERITI



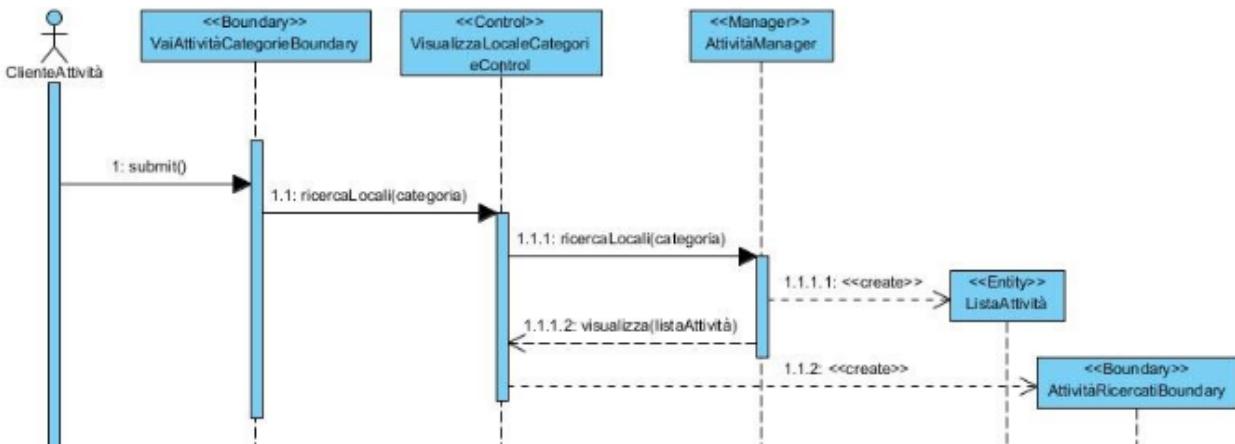
AGGIUNGI ALLA LISTA DEI PREFERITI



ELIMINARE ATTIVITA' DAI PREFERITI



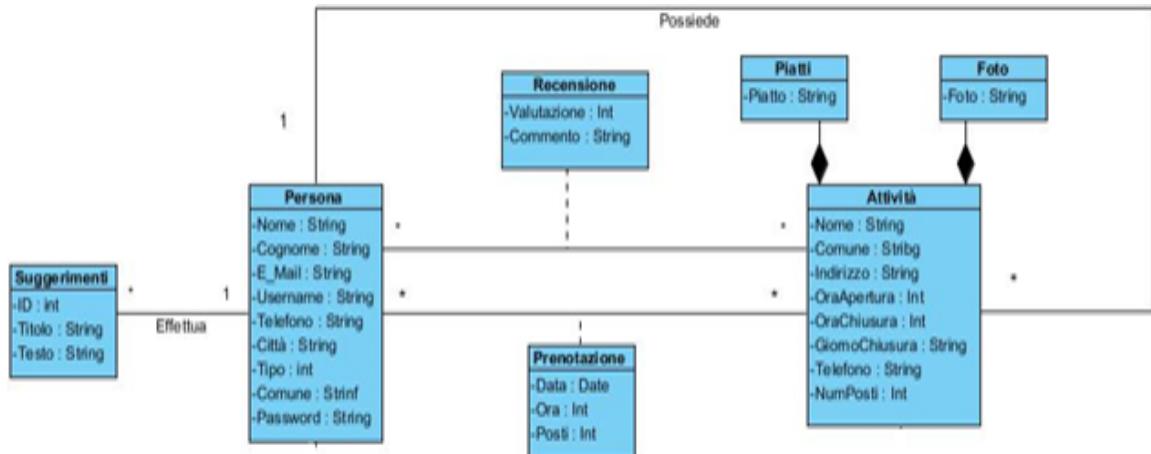
RICERCA ATTIVITA' PER CATEGORIA



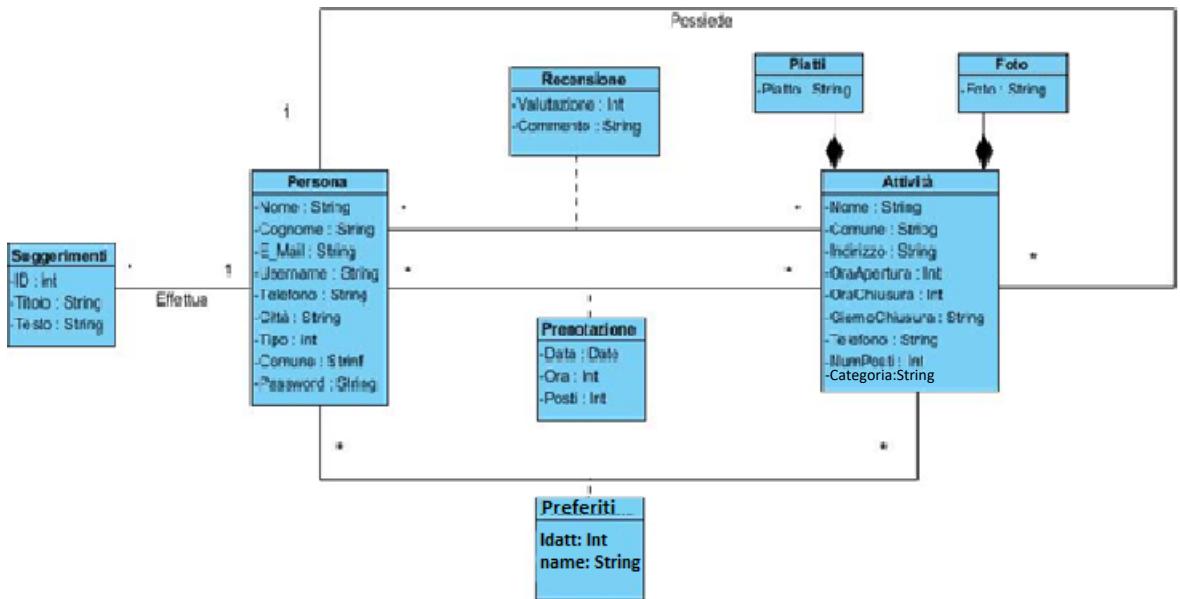
3.2.4 Database

Dopo aver apportato le modifiche agli activity diagram e aver capito i nuovi flussi del sistema, il lavoro è proseguito con l'analisi dell'architettura attuale, partendo dalla strutturazione dei dati persistenti. Dopo aver analizzato il diagramma è stata introdotta una nuova tabella: "Preferiti".

DB prima:



DB dopo:



3.2.5 IMPACT SET

Avendo analizzato la documentazione iniziale ed aver modificato i diagrammi più rilevanti per la richiesta di modifica, si è quindi deciso di redarre lo starting impact set in modo tale da poter poter comprendere cosa modificare per soddisfare la prima change request ad una granularità più alta.



/*Si specifica che lo starting impact set, è stato realizzato con una granularità di classe, in quanto la modifica non solo va ad introdurre nuovi metodi, ma andrà ad impattare metodi già presenti derivanti dal precedente progetto di sviluppo. Con la successiva definizione del Candidate Impact Set si è quindi deciso di adottare un livello di astrazione più basso, considerando i singoli attributi e metodi aggiunti o modificati, avendo a disposizione più dettagli, data l'analisi dei Ripple Effects effettuata tramite il diagramma delle classi*/

STARTING IMPACT SET

- AttivitàDao → aggiunta del metodo doRetriveByCategoria
- AttivitàBean → aggiunta nuovo campo “Categoria:string” all’entità già esistente
- PreferitiBean → creazione della classe bean “PreferitiBean” con getter,setter.
- PreferitiDao → classe con le operazioni CRUD per i preferiti
- ServletPreferiti → creazione della classe “ServletPreferiti”

Segue l’elenco delle JSP che risultano essere candidate per essere aggiunte o modificate, le quali sono state individuate a seguito dell’analisi dei Ripple Effects:

- aggiungiAttività.jsp
- Header.jsp
- visualizzazionePreferiti.jsp

CANDIDATE IMPACT SET

- AttivitàDao → aggiunta di 4 metodi per la ricerca tramite categoria, comune o attività.
- AttivitàBean → aggiunta di 1 attributo “Categoria:string” all’entità già esistente con i rispettivi getter e setter per un totale di 2 metodi.
- PreferitiBean → creazione della classe bean “PreferitiBean” con getter,setter;
 - o Si prevedono gli attributi idPref, personaUsername, attivitàId, quindi un totale di 3 attributi
 - o Ogni attributo sarà accompagnato da un getter e da un setter per un totale di 6 metodi;
- PreferitiDao → classe con le operazioni CRUD per i preferiti e aggiunta di 1 metodo per la ricerca dei preferiti per utente.

Segue l’elenco delle JSP che risultano essere candidate per essere aggiunte o modificate, le quali sono state individuate a seguito dell’analisi dei Ripple Effects:

- aggiungiAttività.jsp
- Header.jsp
- visualizzazionePreferiti.jsp



ACTUAL IMPACT SET

- AttivitàDao → aggiunta del metodo doRetrieveByCategoria, modifica della doSave, doRetrieveByKey, doRetrieveByComune, DoUpdate
- **RicercaControl** → creazione dei metodi get e post per la gestione della ricerca per categoria
- AttivitàBean → aggiunta nuovo campo “Categoria:string” all’entità già esistente con i rispettivi getter e setter
- PreferitiBean → creazione della classe bean “PreferitiBean” con getter,setter;
 - o Si prevedono gli attributi idPref, personaUsername, attivitàId, quindi un totale di 3 attributi
 - o Ogni attributo sarà accompagnato da un getter e da un setter per un totale di 6 metodi;
- PreferitiDao → classe con le operazioni CRUD per i preferiti, doRetrievebyUser
- ServletPreferiti → creazione della classe “ServletPreferiti”, si prevede l’aggiunta della funzionalità doRetrieveByuser
- **aggiungiAttivitàControl** → aggiunta nuovo campo “Categoria:string”

Segue l’elenco delle JSP che risultano essere state effettivamente modificate o aggiunte durante la fase implementativa:

- aggiungiAttività.jsp
- Header.jsp
- visualizzazionePreferiti.jsp
- **attivitàSpecifici.jsp**

A seguito dell’implementazione non sono stati trovati falsi positivi, ma come si può ben notare dall’actual impact set sono state rilevate 3 componenti non incluse nel candidate impact set (componenti evidenziate in grassetto precedentemente).

Summary degli insiemi evinti dall’analisi d’impatto:

CANDIDATE IMPACT SET - 15 componenti impattate

ACTUAL IMPACT SET - 19 componenti impattate

FALSE POSITIVE IMPACT SET - 0 componenti impattate

DISCOVERED IMPACT SET - 0 componenti impattate

3.3 Metriche di Impact Analisys

A seguito delle operazioni di modifica, e avendo individuato gli insiemi di componenti relativi all’analisi d’impatto sono state calcolate ReCall, Precision, Adequacy ed Effectiveness, al fine di valutare la bontà dell’analisi d’impatto effettuata:



Recall: $|CIS \cap AIS| \div |AIS| = 15/18 = 0,83;$

Precision: $|CIS \cap AIS| \div |CIS| = 15/15 = 1;$

Inclusiveness = 0

Come nelle metriche inerenti all'impact analysis della change request n.1, notiamo sia stata trovata una componente impattata non prevista nell'actual impact set. Siccome l'AIS non è, nemmeno in questo caso, un sottoinsieme del CIS, l'inclusiveness sarà pari a 0, quindi non si ricopre una situazione ideale di adeguatezza. D'altra parte, precision e recall forniscono buoni valori siccome sono molto vicini ad 1, il che raffigura una buona situazione per una change request siccome CIS e AIS sono praticamente molto simili tra loro.

È comunque importante ricordare che la granularità degli elementi è molto fine, e che comunque in una situazione reale, difficilmente precision e recall sono effettivamente pari ad 1.