



Relazione del progetto di Web Architetture di Muhammad Waseem Asghar “Universe Shop”

Sommario

Introduzione	2
Requisiti del sistema	2
Attori del sistema.....	2
Requisiti funzionali.....	3
Requisiti non funzionali.....	3
Progettazione e implementazione	3
Strato di persistenza	4
Livello di Business Logic	5
Strato web	6
Diagramma Use-Case	7
Diagramma di sequenze(sequence diagram)	8
Tecnologie utilizzate	11
Lavori futuri	12
INDICE FIGURE.....	12

Introduzione

“Dai la tua opinione sui negozi, supermercati, bar, discoteche ecc... in cui ti rechi abitualmente. Leggi le opinioni di altre persone su negozi in cui stai per fare acquisti! Su questo sito puoi fare tutto questo e molto altro ancora.”

L’obiettivo principale di questa applicazione è quello di raccogliere le “opinioni” sulle attività già presenti nel sistema oppure lasciarle registrare in maniera autonoma. L’ esigenza di condividere l’ opinione sui negozi con altre persone ci ha portati alla creazione di questo sito.

È un’idea che nasce dopo aver sperimentato che ascoltando i consigli di altre persone su qualcosa di cui si sa poco si fanno scelte migliori, almeno ci si prova.

Quindi su questo sito è possibile lasciare i propri commenti su negozi che di solito frequentiamo, magari specificando cosa ci ha colpito di quel locale, se ci ha fatto una buona impressione, se il commesso era gentile o no e quello che ci viene in mente.

Requisiti del sistema

Il sistema ha obiettivo di promuovere le attività dei commercianti, facendo sì che questi ultimi possano utilizzare le opinioni degli utenti che hanno frequentato il locale, per migliorare le loro attività.

Quindi il sistema deve permettere ai commercianti di registrare le loro attività, con una breve descrizione. Una volta che le attività sono inserite nel sistema gli utenti registrati possono lasciare le opinioni o commenti. Per la registrazione si deve fornire una interfaccia grafica, possibilmente semplice.

Attori del sistema

Gli attori sono le entità coinvolte nel sistema e sono:

1. Utenti: Gli utenti registrati, che hanno facoltà di inserire le loro attività o lasciare commenti sulle attività già registrate. Per la registrazione degli utenti sono richiesti dati essenziali.
2. Negozio o attività: posso essere registrate dagli utenti registrati. Le attività possono essere di natura diverse. Nella fase di registrazione è richiesto una breve descrizione ed luogo dove ha la sede l’attività.
3. Opinioni o commenti: le opinioni sono lasciate dagli utenti registrati e danno una valutazione sugli servizi offerti dai locali, dai negozi ecc. Ovviamente le opinioni offensive non verranno pubblicate.
4. Luogo: per il luogo si intende dov’è situato il negozio. In database sono già presenti tutte le regioni con relative province e comuni.

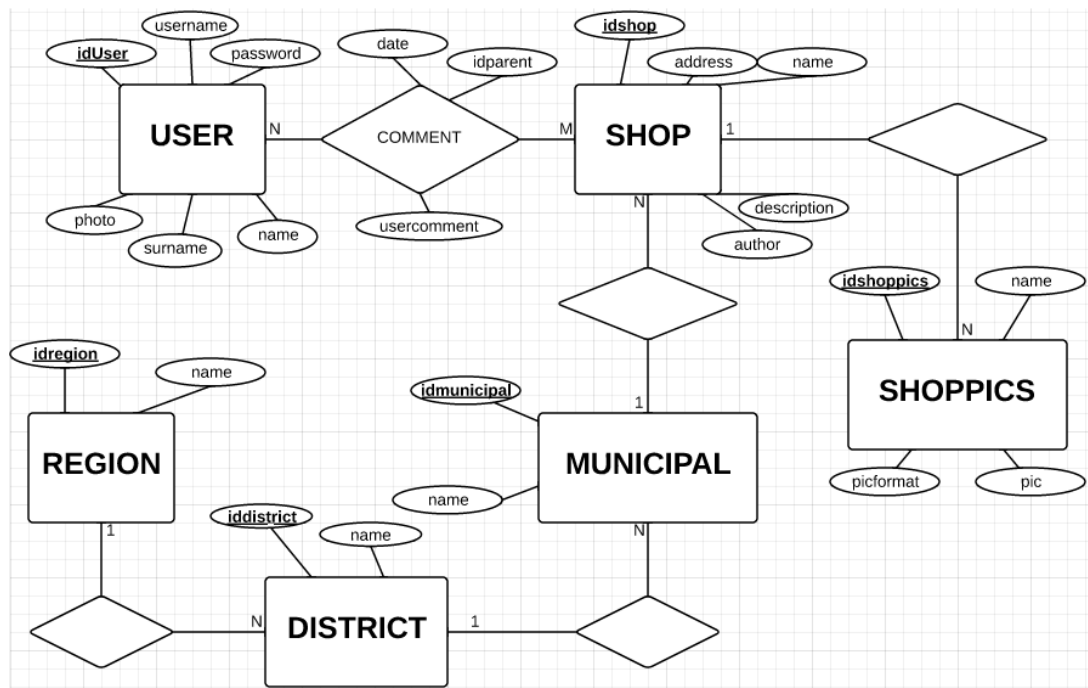


Figure 1 diagramma delle entità

Requisiti funzionali

Il sistema è deve essere molto intuitivo ed ha seguenti requisiti funzionali:

- Visualizzazione dei commenti relativi alle attività presenti nel sistema, secondo il luogo
- Rilascio di una opinione oppure commento di su un attività
- Registrazione di una attività commerciale
- Registrazione di utente/cliente con dati essenziali
- Modifica e/o aggiornamento del profilo di un utente/cliente
- Visualizzazione dei negozi con relativi commenti di un determinato utente
- Autenticazione per rilasciare i commenti

Requisiti non funzionali

Il sistema è stato implemento utilizzando la nuove tecnologie come Primefaces per interfaccia grafica ecc. Questo scelta è dovuto alle di controllo nella compilazione dei campi. Inoltre dà possibilità di utilizzare i temi già predefiniti.

Progettazione e implementazione

La struttura del progetto può essere divisa in tre grandi livelli:

Livello di persistenza: gestisce l'accesso al database, ed è implementato con Hibernate.

Livello di Business Logic: implementa le Enterprise Java Beans e gestisce la relazione tra strato web e livello di persistenza.

Strato web: è la parte di interfaccia grafica, con la quale un utente interagisce con il sistema.

Strato di persistenza

Lo strato di persistenza è il livello il cui compito è quello di fornire un'interfaccia astratta al database. Ad ogni tabella del database corrisponde una entity bean nel progetto. La mappatura tra le tabelle del database e le entities beans viene eseguita con Hibernate. Tutte le impostazioni di Hibernate sono nel file "hibernate.cfg.xml" (nome utente, la password per il database, indirizzo del database, tabelle mappate, ...). Qui sotto nell'immagine si vede il diagramma ER che mostra l'organizzazione del database utilizzato per memorizzare le informazioni relative agli utenti e commenti rilasciati dagli stessi. Ci sono alcune tabelle come ("region", "district" e "municipal") che non vengono aggiornate e contengono dati relativi rispettivamente alle regioni, alle province e ai comuni, rispettando il vincolo di integrità. La tabella "User" contiene le informazioni relative agli utenti registrati, la tabella "Shop" contiene dettagli di una attività, invece la tabella "Comment" è utilizzata per memorizzare le informazioni sulle opinioni/commenti rilasciati dagli utenti. Nella tabella "Comment" il campo "idparent" si riferisce se il commento è una risposta al commento già rilasciato in precedenza, questo non è stato ancora implementato ma in futuro lo sarà. Invece la tabella "shoppics" contiene le immagini dei negozi o locali. Anche questa funzione non è ancora implementata, ma sicuramente verrà considerata in futuro.

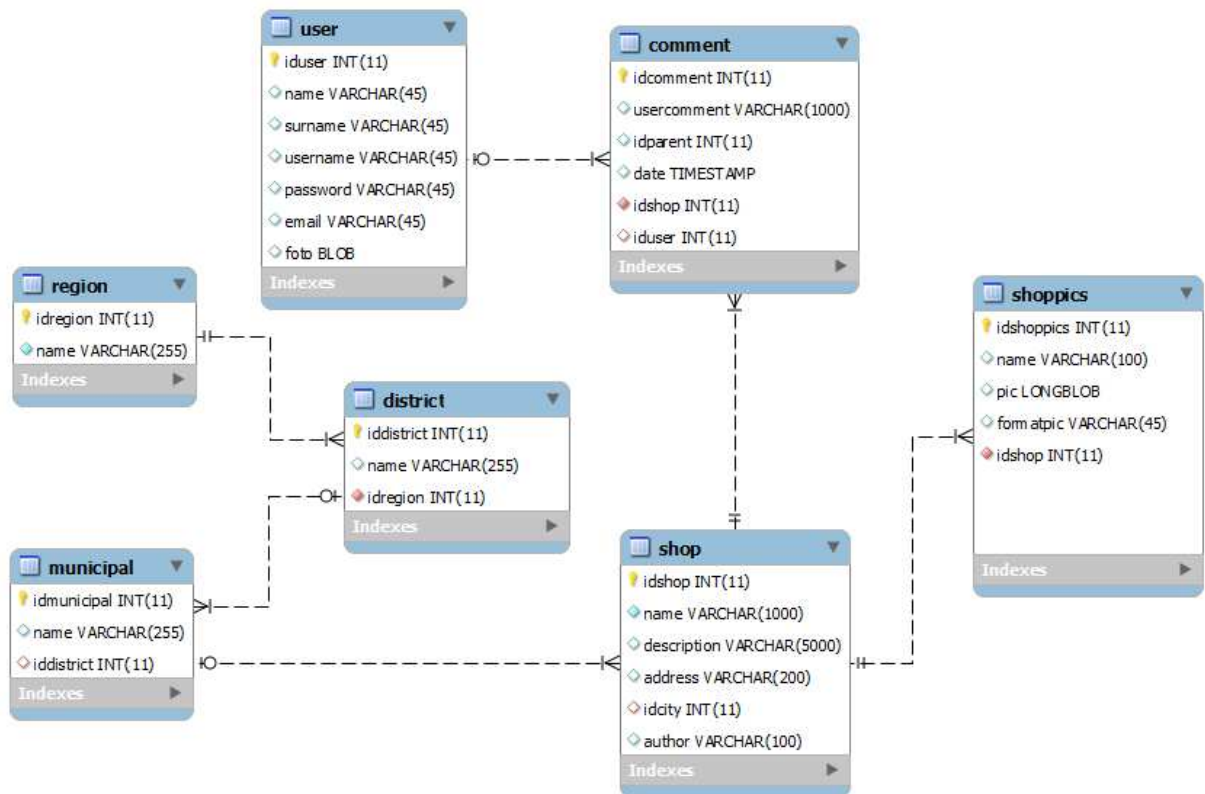


Figure 2 diagramma delle entità

Livello di Business Logic

In questo livello si implementano le Enterprise Java Beans (EJB) per gestire la relazione tra strato web e livello di persistenza. La maggior parte delle EJB sono *stateless*, perché sono principalmente utilizzate per reperire/o salvare le informazioni nel database. Malgrado l'importanza e complessità del progetto non sono riuscito ad individuare qualche *statefull* EJB.

Qui sotto nella figura si è cercato di tracciare un diagramma delle classi, essendo molte classi presenti il diagramma non leggibile a prima vista. Dal lato client non si può accedere direttamente alle EJB ma queste ultime sono gestite da altre Beans che fanno l'interfaccia al client. Quindi preservando così il concetto MVC.

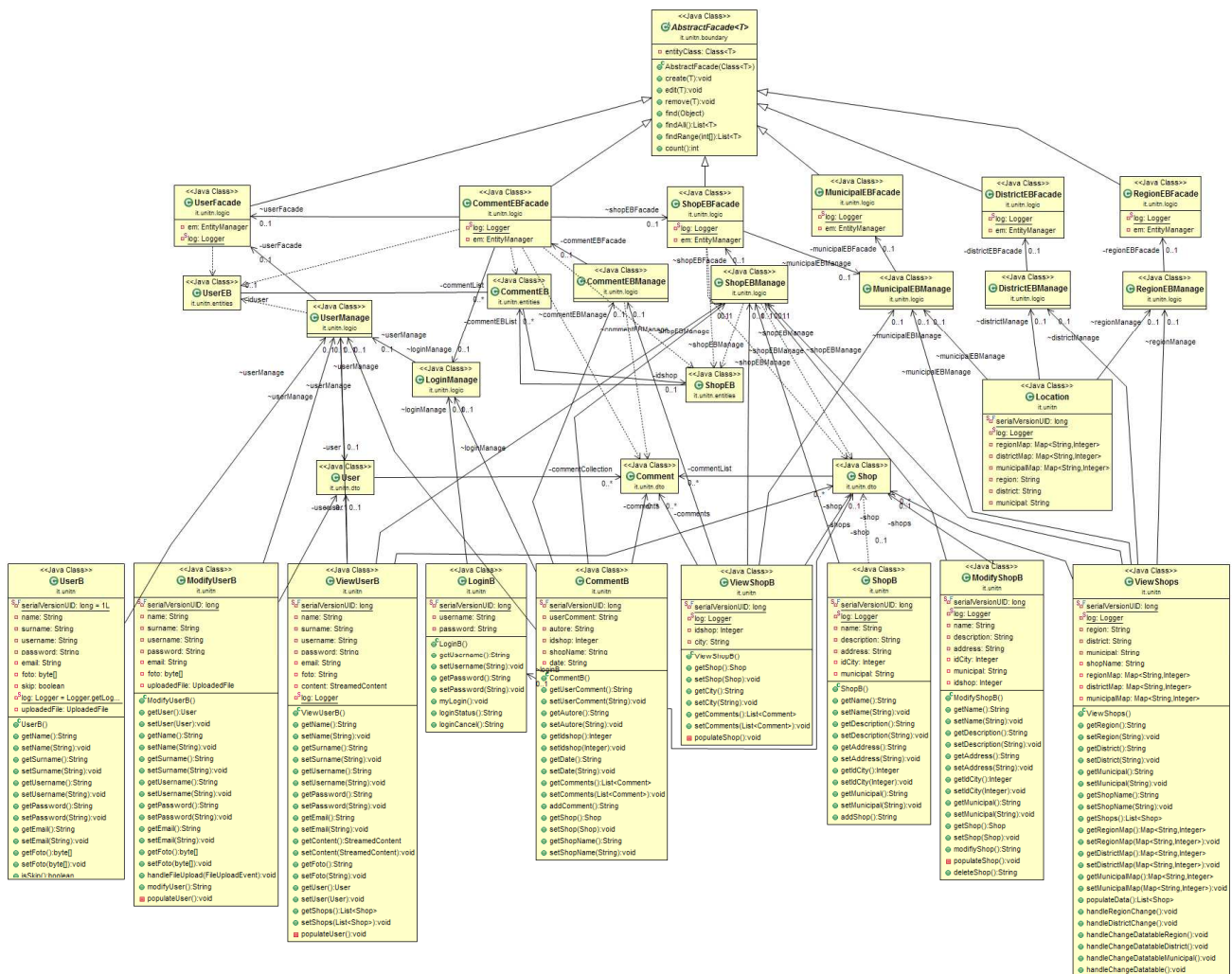


Figure 3 diagramma delle classi

Strato web

Lo strato web è la parte di interfaccia grafica, viene implementata usando la tecnologia JEE (Java Enterprise Edition). In particolare la grafica della applicazione web è stata implementata con la tecnologia **JavaServer Faces(JSF)** . JSF è una tecnologia Java basata sul design pattern architetturale Model-View-Controller (MVC). Inoltre s'è usufruito del framework Primefaces. PrimeFaces è una suite open source di componenti UI avanzati progettata per integrare funzionalità Ajax all'interno di applicazioni Java Server Faces, esattamente come lo è ad esempio il più noto RichFaces. La suite Primefaces offre supporto ad Ajax, grazie ad una integrazione nativa con JQuery (e su questo ci ritorneremo più avanti). Il framework è dotato di più di 100 componenti JSF avanzati.

Primefaces si pone in diretta concorrenza con i ben noti e già maturi RichFaces e IceFaces. Tuttavia sta suscitando un enorme interesse grazie ad alcuni punti di forza che lo rendono assai competitivo nei confronti degli altri progetti.

- Facilità per iniziare:

E' sufficiente una sola libreria e zero configurazione.

- Ajax e JQuery

Supporto Ajax e utilizzo non intrusivo di JavaScript, basato sulla più recente versione di JQuery.

- Maggior numero di componenti:

Come abbiamo detto più di 100 componenti disponibili basati su JQueryUI (HTMLEditor, Dialog, DataTable, Menu, AutoComplete, Charts e molto altro), in più con compatibilità dichiarata con altre librerie di componenti JSF.

- Performance:

PrimeFaces è una libreria leggera, tutte le scelte prese sono basate sul mantenere PrimeFaces il più leggero possibile.

- Facilità d'uso:

I Componenti in PrimeFaces sono sviluppati secondo il seguente principio di design: "Un componente UI buono dovrebbe nascondere la complessità, ma mantenere la flessibilità".

- Documentazione e supporto:

Infine una ricca documentazione e pieno supporto per gli sviluppatori, grazie anche ad una attiva community.

La scelta di JSF e Primefaces è dovuta ai vantaggi descritti sopra.

Diagramma Use-Case

Il diagramma UC dà un'idea delle operazioni che posso essere compiute dall'utente. Quindi molte operazioni l'autenticazione è richiesta, ciò vuol dire che l'utente deve essere registrato ed autenticato prima di effettuare quella operazione. Ad esempio nel diagramma possiamo vedere che per lasciare il commento un utente deve essere autenticato.

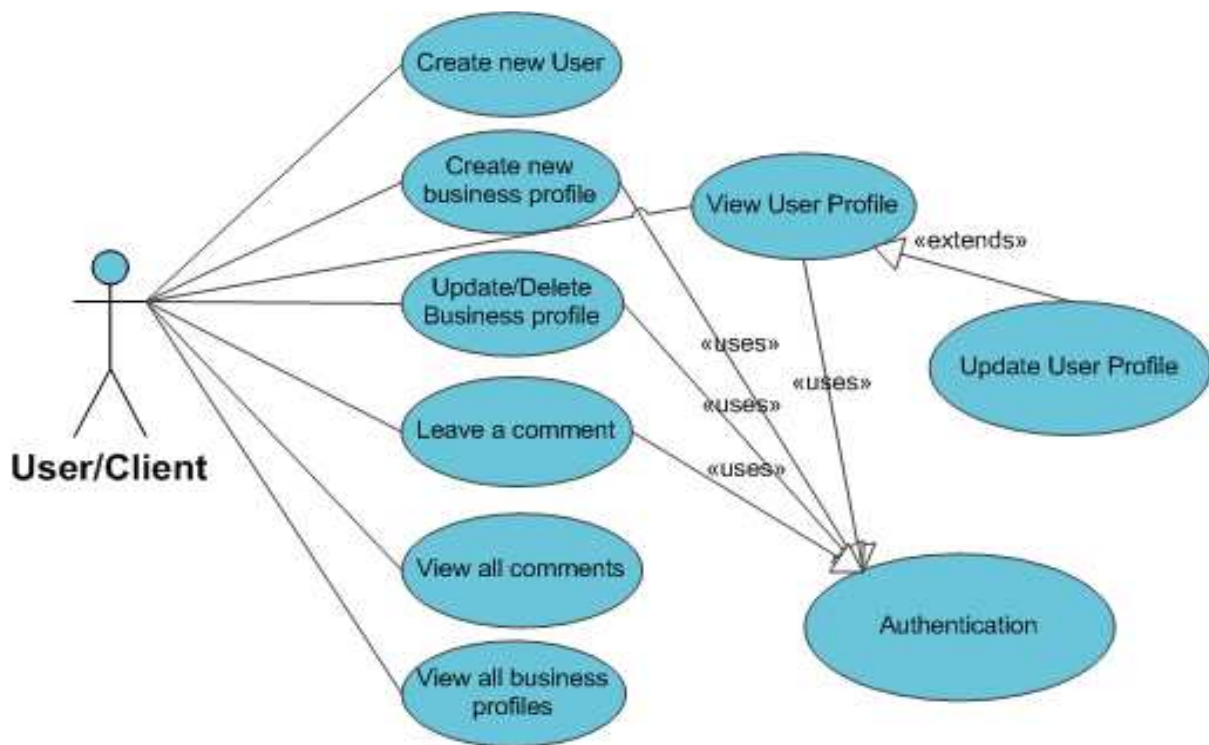


Figure 4 diagramma Use Case

Diagramma di sequenze(sequence diagram)

Il diagramma di sequenze ha lo scopo di mostrare lo scenario considerato in dettaglio. Inoltre descrive le relazioni che intercorrono, in termini di messaggi, tra “Attori”, oggetti di business, oggetti od Entità del sistema che si sta rappresentando.

Di seguito sono riportati alcuni scenari significativi:

Scenario 1: registrazione di un nuovo utente

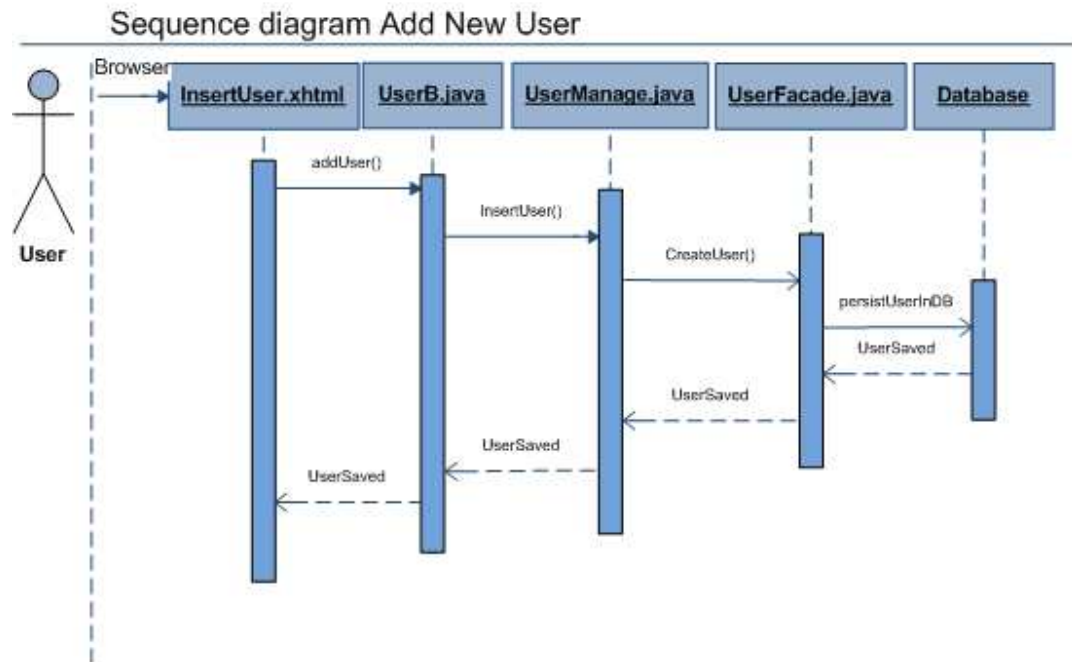


Figure Diagramma di sequenze Scenario 1

Scenario 2: visualizzazione delle attività/negozi

Le attività/negozi posso essere filtrati in base al luogo di sito. Nel primo sotto scenario vengono visualizzati tutti i negozi presenti nel database, quindi senza filtraggio. Invece nella seconda fase del scenario 2, le attività/negozi sono visualizzati in base alla regione, ovviamente il criterio di filtraggio è simile sia per le province che per i comuni, quindi per questo motivo non sono stati mostrati in questo scenari.

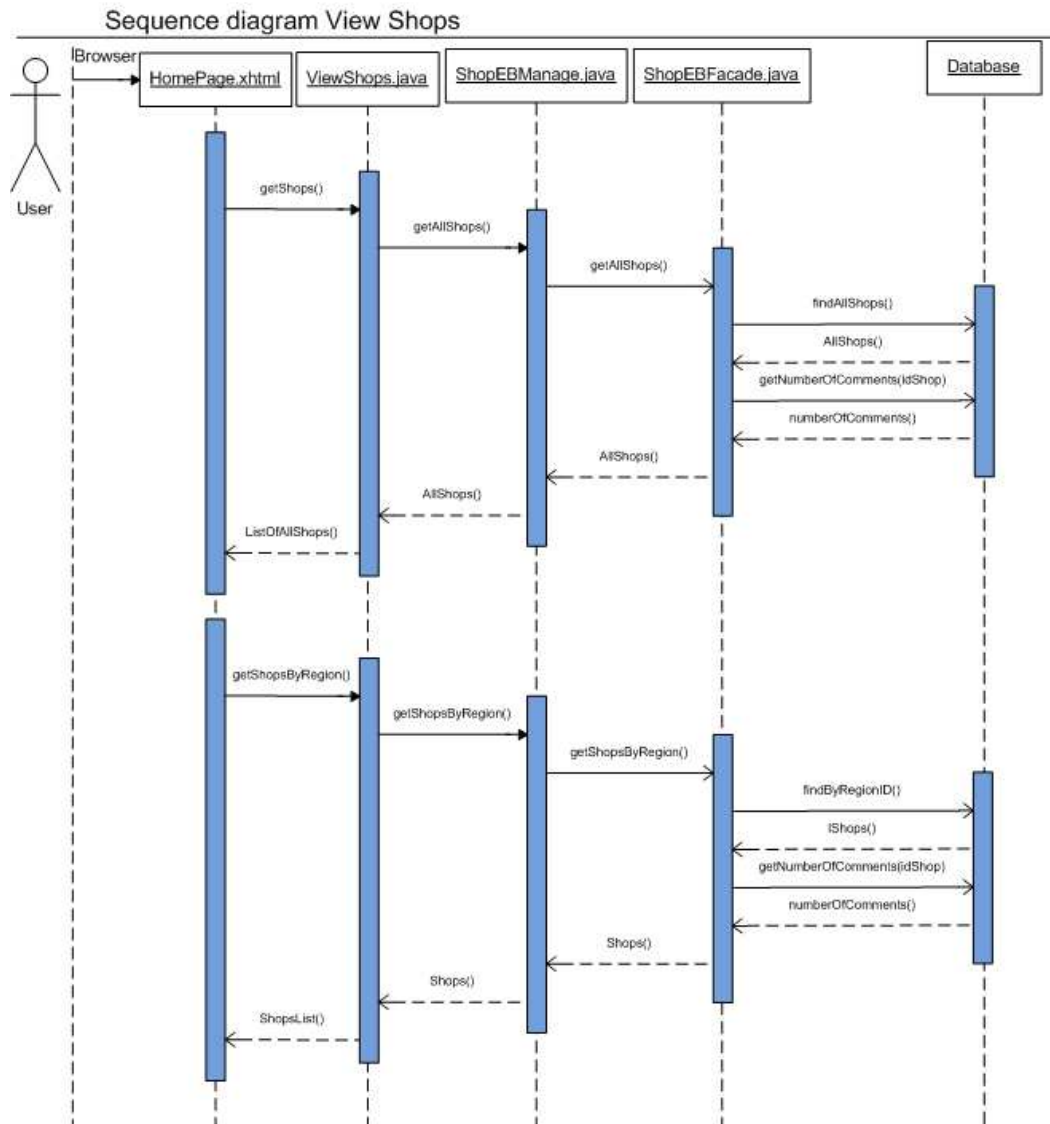


Figure 5 Diagramma di sequenze Scenario 2

Scenario 3: lasciare un commento/opinione su una attività

In questo scenario viene mostrata come lasciare un commento/opinione su una attività già registrata. Visto che per lasciare dei commenti/opinioni l'utente deve essere registrato quindi per questo motivo vengono dimostrati i passi da compiere per l'autenticazione dell'utente. Noi supponiamo per semplicità che i dati di login inseriti dall'utente siano corretti. Una volta che l'autenticazione è avvenuta si passa al rilascio del commento. Dalla "HomePage.xhtml" bisogna scegliere l'attività per il quale si vuole rilasciare il commento. Una volta selezionata l'attività si va sulla pagina "LeaveComment.xhtml" dove si imposta il commento e invia per salvarlo nel database.

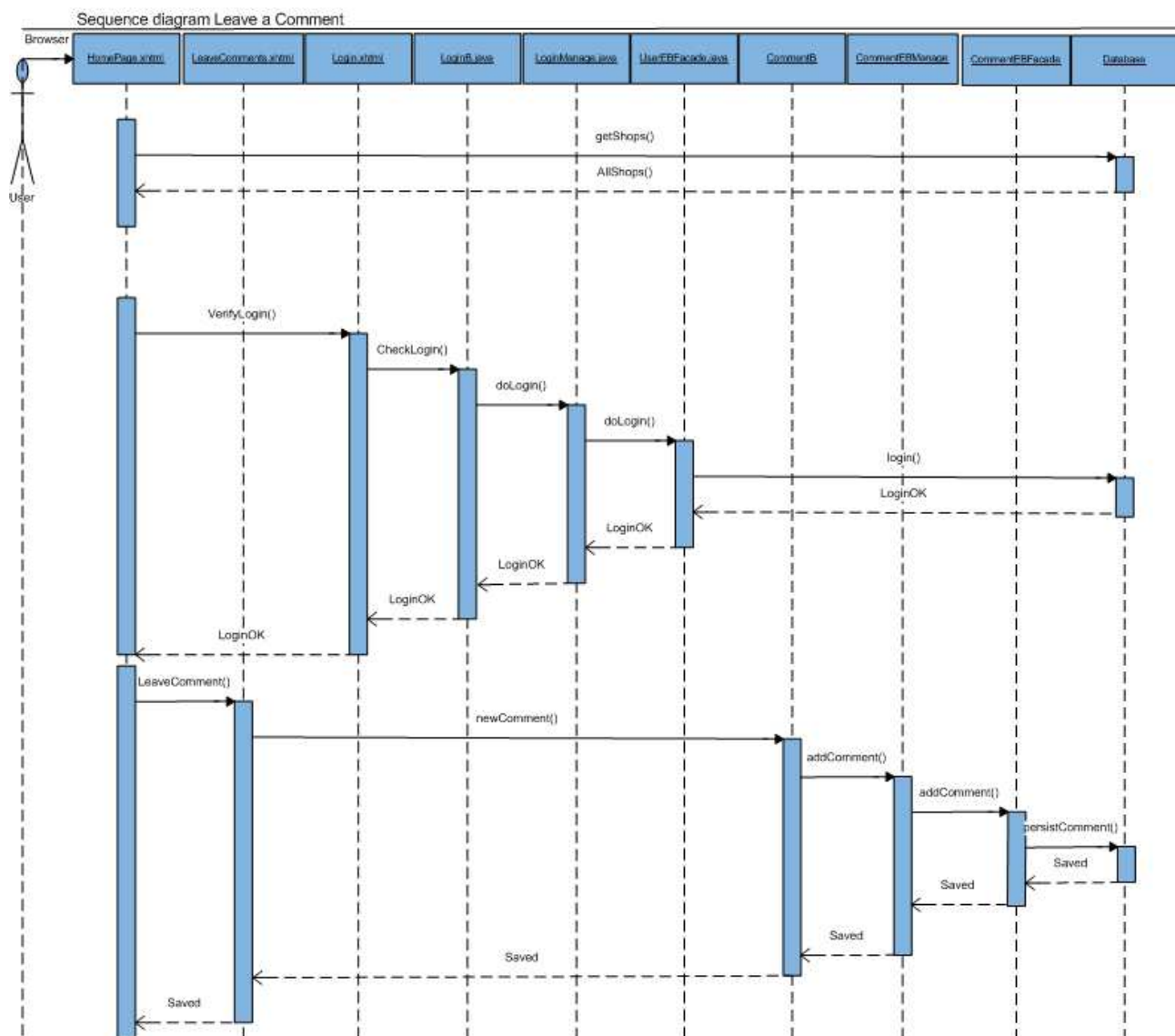


Figure 6 Diagramma di sequenze Scenario 3

Tecnologie utilizzate

Hibernate 4.1.

Framework primefaces

Java Server Faces 2.0

Eclipse Java EE IDE

MySQL

jboss-7.1

Firefox

Lavori futuri

1. Impostare i ruoli per gli utenti (amministratore, utenti commercianti, utenti normali)
2. Creare filtro per i messaggi offensivi, per escludere parole considerate offensive dal pubblico
3. Dare possibilità all'utente di mettere un rating all'attività (tipo mi piace o non mi piace) per poter visualizzare in base a queste informazioni
4. Migliorare nella parte di interfaccia grafica

INDICE FIGURE

Figure 1 diagramma delle entità	3
Figure 2 diagramma delle entità	5
Figure 3 diagramma delle classi	6
Figure 4 diagramma Use Case	8
Figure 5 Diagramma di sequenze Scenario 2	10
Figure 6 Diagramma di sequenze Scenario 3	11