

Tesina di Applicazioni Web Distribuite

Dipartimento di Ingegneria “Enzo Ferrari”

Corso di Laurea in Ingegneria Informatica

magnatron: software gestionale di una mensa aziendale

Elaborato di:

Campagnoli Daniele matr:94197

Indice

Entità del progetto.....	3
amministratore.....	3
impiegato.....	3
cuoco.....	3
piatto.....	3
menu giornaliero.....	3
menu settimanale.....	4
Use case.....	4
1)login.....	4
2)visualizzare la lista delle spese per ogni dipendente.....	4
3)scelta dei piatti nel menu settimanale.....	4
4)eliminazione scelta dei piatti nel menu settimanale.....	4
5)creazione piatti.....	4
6)creazione menu giornaliero.....	4
7)creazione menu settimanale.....	5
8)attivazione menu settimanale.....	5
9)visualizzazioni ordini settimanali.....	5
Architettura del progetto e tecnologie utilizzate.....	5
Weblogic 12C.....	5
MySQL.....	6
Gestione del progetto.....	7
Metodo di sviluppo.....	7
Struttura del software e file principali.....	7
struttura cartella magna tron.....	7
struttura cartella database.....	8
Documentazione aggiuntiva.....	8

Entità del progetto

amministratore

- Visualizza la lista delle spese per ogni dipendente

impiegato

- Selezione dei piatti all'interno di un menu settimanale.

cuoco

- crea i piatti
- crea il menù giornaliero
- crea il menù settimanale
- visualizza la lista degli ordini settimanali

piatto

Piatto servito nella mensa, può essere:

- primo
- secondo
- contorno
- dessert

Un piatto ha un prezzo stabilito dal cuoco.

menu giornaliero

Descrive il menu del giorno.

Un menu è composto da:

- primo
- secondo
- contorno
- dessert

per ogni portata del menù ci devono essere almeno due scelte.

menu settimanale

Ha una data di inizio ed è associato a 6 menù giornalieri.

menu_settimanale.vincolo1: il menu entra in produzione se viene attivato.

menu_settimanale.vincolo2: il menu può essere scelto dai dipendenti nel primo giorno.

Use case

1)login

1. Inserire username e password
2. verrà aperta la schermata corrispondente al ruolo dell'utente

2)visualizzare la lista delle spese per ogni dipendente

1. Impostare intervallo temporale su cui calcolare le spese
2. calcolare le spese

3)sceita dei piatti nel menu settimanale

1. Selezionare un menu settimanale attivo
2. selezionare un menu giornaliero
3. selezionare un piatto disponibile

Questo use case può essere rieseguito fino a che l'utente ha la possibilità di effettuare scelte nel menu in funzione del vincolo temporale(menu_settimanale.vincolo2).

4)eliminazione scelta dei piatti nel menu settimanale

1. Selezionare un menu settimanale attivo
2. selezionare un menu giornaliero
3. eliminare un piatto scelto

Questo use case può essere rieseguito fino a che l'utente ha la possibilità di effettuare scelte nel menu in funzione del vincolo temporale(menu_settimanale.vincolo2).

5)creazione piatti

1. inserire i dati relativi ad un piatto
2. confermare l'inserimento

6)creazione menu giornaliero

1. selezionare i piatti inseriti
2. confermare la selezione

7)creazione menu settimanale

1. Inserire la data di inizio del menu

8)attivazione menu settimanale

1. Creare 6 menu giornalieri con due piatti per ogni portata
2. attivare il menu

9)visualizzazioni ordini settimanali

1. Inserire una data di inizio menu settimanale
2. selezionare un menu settimanale
3. selezionare un giorno dal menu settimanale
4. produzione degli ordini corrispondenti al periodo temporale selezionato

Architettura del progetto e tecnologie utilizzate

Il sistema ha un'architettura multi-tier costituita da due elementi fondamentali:

- Weblogic 12c
- MySQL

Weblogic 12C

Weblogic 12C si occupa di eseguire il livello applicativo e di presentazione in conformità con lo standard javaEE7. In questa implementazione la stessa istanza di Weblogic esegue la presentation logic e la business logic, ma con alcune modifiche al software si possono separare queste due funzionalità in macchine diverse. Il livello applicativo è composto dalle seguenti tecnologie:

- eclipselink
- enterprise java bean
- java server faces

Eclipselink(<http://www.eclipse.org/>) è una implementazione dello standard JPA 2.1, che definisce una serie di tecniche per mappare i dati contenuti all'interno di un database direttamente all'interno di oggetti Java. Questa tecnica è chiamata ORM(Object Relational Mapping) e permette di semplificare la comunicazione con il database. Eclipselink genera in modo automatico il codice, nel nostro caso sql, necessario alla manipolazione del database e fornisce al programmatore un'interfaccia ad oggetti molto semplice e concisa. Questo livello intermedio permette di minimizzare le chiamate al database e di fare diverse attività di ottimizzazione attraverso una serie di parametri di configurazione. Gli oggetti gestiti da eclipselink vengono chiamati entità.

Enterprise java bean è una tecnologia definita dallo standard javaEE 7 e implementano la business logic del sistema. Questo livello si occupa di manipolare le entità attraverso un'interfaccia che può essere utilizzata da processi appartenenti alla stessa jvm(java virtual machine) o ad una jvm remota. In questo sistema si è scelto di utilizzare un'interfaccia locale per semplificare lo sviluppo. Tutti gli enterprise java bean sono stati realizzati sfruttando i servizi offerti dal container per quanto riguarda le transazioni e la comunicazione con eclipselink.

Java Server Faces implementa la presentation logic ed è sostanzialmente il client degli enterprise java bean.

MySQL

MySQL memorizza in modo permanente i dati e li fornisce al livello applicativo. L'integrità dei dati viene garantita attraverso trigger e stored procedure.

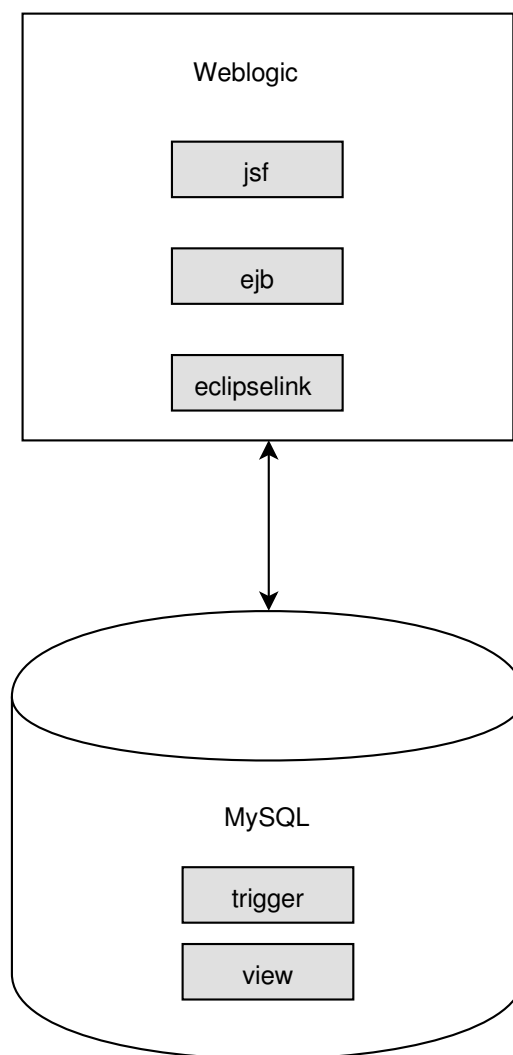


Figura 1: architettura

Gestione del progetto

Il progetto è stato gestito attraverso un software chiamato maven. Questo strumento è stato utilizzato per automatizzare la gestione delle librerie esterne e la creazione della build contenente il

codice del progetto. Maven definisce una struttura standard per il progetto, in questo modo possiamo importarlo in qualsiasi IDE mantenendo le stesse configurazioni.

Metodo di sviluppo

Durante la realizzazione della business logic e della data logic è stato utilizzato il test-driven development, cioè subito dopo la creazione di una funzionalità vengono realizzati dei test che vanno a validare il software appena creato. In questo modo le specifiche iniziali del software vengono tradotte in una serie di test che evidenziano in modo precoce eventuali bug. Per implementare questa tecnica in un'applicazione multi-tier si devono far lavorare più livelli insieme attraverso degli integration test. Il problema di questo approccio nel mondo javaEE è la presenza del container all'interno del server, senza di esso non c'è la possibilità di eseguire il software sviluppato in un ambiente identico a quello di produzione. Se i test vengono eseguiti fuori dal container perdono di efficacia perché diventa impossibile testare il funzionamento dei parametri di configurazione impostati tramite file xml o annotazioni ed è necessario utilizzare tecniche di testing avanzate per avere più moduli nello stesso test. Per risolvere questo problema si possono adottare diverse soluzioni, in questo elaborato sono state utilizzate le seguenti librerie:

- junit
- arquillian
- shrinkWrap

Arquillian(<http://arquillian.org/>) è una libreria che permette di eseguire i test direttamente sul container che ospiterà l'applicazione in produzione. La libreria ha diversi plugin che permettono di gestire il server di destinazione e vengono configurati tramite un file chiamato arquillian.xml. Sul server di destinazione viene eseguito un deployment temporaneo che contiene i test e i moduli da testare, dopodiché arquillian esegue i test e restituisce un report con i risultati.

Junit (<http://junit.org/junit4/>) implementa i test veri e propri che vengono richiamati da arquillian.

shrinkWrap(<https://github.com/shrinkwrap/shrinkwrap>) è una libreria che viene utilizzata per creare le build dei deployment temporanei di arquillian

Struttura del software e file principali

Il software è suddiviso in due cartelle principali:

- magna_tron: contiene il codice java sviluppato
- database: contiene gli script per la creazione del database

struttura cartella magna_tron

Il software java è organizzato secondo la logica imposta dalle configurazioni di default di maven, questo permette a qualunque sviluppatore di comprenderla. All'interno di questa cartella troviamo due sottocartelle:

- main: contiene i sorgenti della build

- test: contiene i test che vengono eseguiti sul software contenuto nella cartella main, non fa parte della build

All'interno della cartella main troviamo le cartelle:

- java: contiene i package del source code
- resources: contiene file di configurazione
- webapp: contiene i file della presentation logic

All'interno della cartella java troviamo i seguenti package:

- magna_tron.controller: contiene la logica dei controller per la presentation logic
- magna_tron.ejb: contiene la business logic
- magna_tron.model: contiene il codice che esegue il mapping tra il database e l'applicazione in stile ORM.

All'interno della cartella test troviamo i test organizzati in modo simile ai package descritti precedentemente.

struttura cartella database

Questa cartella contiene due script che creano ed eliminano il database:

- create_db.sh: crea il database
- drop_tables.sh: elimina il database

Tutti gli altri file sono script sql che definiscono la struttura del database.

Documentazione aggiuntiva

La documentazione dettagliata del software si trova all'interno del percorso :

magna_tron/target/site/apidocs

Gli schemi er si trovano nella cartella: database/documentazione