

# DoveCibo

## **Relazione Progetto “Sistema di recensione e classificazione di ristoranti”**

Giacomo Barbieri - Matricola 171973

Claudio Bizzotto - Matricola 157597

Michael Dolzani - Matricola 173209

Stefano Postal - Matricola 174735

Giorgio Alberti - Matricola 172735

Anno Accademico 2016/2017

## **Premessa**

Lo scopo del progetto è creare un sistema di recensione e classificazione di ristoranti da esporre via WEB e fruibile per dispositivi sia desktop che mobile in modalità responsive. L'applicazione deve essere sviluppata con tecnologia Servlet/JSP/JSTL ed ove fosse conveniente per vostra scelta: WEBServices, CSS e/o Javascript.

## **Indice**

1. Introduzione
2. Struttura del Progetto
  - 2.1 Javascript
  - 2.2 CSS
  - 2.3 Librerie
  - 2.4 Cartelle Extra
  - 2.5 Classi Java
  - 2.6 JSP
3. Scelte progettuali
4. Istruzioni per la creazione del Database
5. ER del Database
6. Conclusioni

## 1. Introduzione

Il progetto fa ampio utilizzo dei linguaggi Java e HTML. Essi vengono integrati attraverso l'utilizzo delle Servlet, con le quali è possibile creare pagine web dinamiche.

## 2. Struttura del Progetto

### 2.1 Javascript

I javascript usati all'interno del progetto sono contenuti nella cartella *DoveCiboGit/web/script*.

- aggiungiRistoranteMap.js
- datatable.js
- datatable2.js
- datatable3.js
- goback\_location.js
- goback\_window.js
- home\_geo.js
- onload.js
- slider.js

### 2.2 CSS

I CSS usati all'interno del progetto sono contenuti nella cartella *DoveCiboGit/web/css*.

- autocompletamento.css
- cercaRistoHomeValue.css
- cercaRistorantiHome.css
- home.css
- modificheRistorante.css
- notifica.css
- ristorante.css
- ristoranteMap.css
- visualizzaRistoranti.css

## 2.3 Librerie

Le librerie usate all'interno del progetto sono contenute nella cartella *DoveCiboGit/web/WEB\_INF/lib*.

- commons-codec-1.10.jar
- commons-codec-1.10-javadoc.jar
- commons-codec-1.10-sources.jar
- commons-codec-1.10-tests.jar
- commons-codec-1.10-tests-sources.jar
- commons-fileupload-1.3.2.jar
- commons-io-2.5.jar
- commons-io-2.5-javadoc.jar
- core-2.2.jar
- cos.jar
- javase-2.2.jar
- javax.mail.jar
- postgresql-9.4.1208.jre6.jar
- qrigen-1.3.jar

## 2.4 Cartelle Extra

- Database - Contiene il backup del database DoveCibo
- ImmaginiCaricate - Contiene le immagini caricate dagli utenti del sito
- LibreriaImmagini - Contiene una libreria di immagini standard pronte all'uso
- Relazione - Contiene la relazione sul progetto
- Sfondi - Contiene le immagini di supporto al sito

## 2.5 Classi Java

### Package "autocompletamento"

**ServletGetAutoC.java** - Prende tutti i valori dal database e li manda alla home.jsp per l'autocompletamento nel textfield della ricerca generale

### Package "database"

**DB\_Coordinate.java** - Gestisce le informazioni sulla posizione di un ristorante nel database

**DB\_CuisineRestaurant.java** - Gestisce le informazioni sulle cucine nel database

**DB\_GestioneRestaurant.java** - Gestisce le informazioni sui ristoranti nel database

**DB\_GestioneUse.java** - Gestisce le informazioni sugli user nel database

**DB\_Manager.java** - Gestisce la connessione con il database

**DB\_Notifica.java** - Gestisce le informazioni sulle notifiche nel database

**DB\_OrariRestaurant.java** - Gestisce le informazioni sugli orari dei ristoranti nel database

**DB\_PriceRestaurant.java** - Gestisce le informazioni sui prezzi dei ristoranti nel database

**DB\_Replies.java** - Gestisce le informazioni sulle risposte ai commenti nel database

**DB\_RestaurantOwner.java** - Gestisce le informazioni sui proprietari di ristoranti nel database

**DB\_RestaurantPhoto.java** - Gestisce le informazioni sulle foto dei ristoranti nel database

**DB\_Reviews.java** - Gestisce le informazioni sui commenti dei ristoranti nel database

### Package "emails"

**SendMail\_Activazione.java** - Spedisce la mail per l'attivazione di un account

**SendMail\_ModificaProfilo.java** - Spedisce la mail con una notifica delle modifiche apportate ad un profilo utente

**SendMail\_RicuperoCredenziali.java** - Spedisce la mail per il recupero delle credenziali

#### Package "notifiche"

**Notifica.java** - Contiene i metodi per la gestione delle notifiche

**Replies.java** - Contiene i metodi per la gestione delle risposte ai commenti

**ServletAccettaReclamo.java** - Gestisce l'assegnamento di un ristorante ad un ristorante

**ServletAccettaRisposta.java** - Gestisce la pubblicazione della risposta del ristorante ad un commento

**ServletAggiungiRecensione.java** - Gestisce la possibilità di aggiungere una recensione ad un ristorante

**ServletNotifiche.java** - Gestisce la visualizzazione delle notifiche

**ServletNotificheBar.java** - Gestisce la visualizzazione delle notifiche in un popup

**ServletRifiutaReclamo.java** - Gestisce il rifiuto da parte dell'Admin di assegnare un ristorante ad un utente

**ServletRifiutaRisposta.java** - Gestisce il rifiuto da parte dell'Admin di pubblicare la risposta di un ristorante ad una recensione di un utente

**comparatorNotifiche.java** - Ordina le notifiche in ordine temporale

#### Package "restaurant\_comments"

**Review.java** - Contiene i metodi per la gestione delle recensioni

**ServletAggiungiCommento.java** - Gestisce l'aggiunta di una recensione e la valutazione di un ristorante

**ServletAggiungiLike.java** - Gestisce l'aggiunta di un like ad una recensione

#### Package "restaurant\_photos"

**Photo.java** - Contiene i metodi per la gestione delle foto

**ServletModificaFoto.java** - Gestisce la rimozione delle foto

**ServletUpload.java** - Gestisce l'upload delle foto

#### Package "restaurants"

**CercaRistorantiHome.java** - Manda a CercaRistorantiHome la lista dei ristoranti da visualizzare secondo i criteri della ricerca generale

**Coordinate.java** - Contiene i metodi per la gestione delle coordinate

**Cusine.java** - Contiene i metodi per la gestione delle tipologie di cucine

**Day\_Hours.java** - Contiene i metodi per la gestione degli orari di apertura dei ristoranti

**Price\_Range.java** - Contiene i metodi per la gestione dei prezzi dei ristoranti

**QRGenerator.java** - Crea il QR di ogni ristorante

**Restaurant.java** - Contiene i metodi per la gestione dei ristoranti

**Riempi.java** - Main per il riempimento veloce dei commenti ai ristoranti

**ServletGetRistorante.java** - Manda a ristorante.jsp le informazioni di un ristorante per poter visualizzare

**ServletGetRistoranteProprietario.java** - Manda a modificaRistorante.jsp i dati da visualizzare

**ServletGetRistorantiHomeCucine.java** - Manda cercaRistoHomeValue.jsp la lista dei ristoranti filtrati per tipologia di cucina e posizione

**ServletGetRistorantiHomeValue.java** - Manda a cercaRistoHomeValue.jsp la lista dei ristoranti filtrati per classifica e posizione

**ServletModificaRistorante.java** - Convalida le modifiche apportate ad un ristorante da modificaRistorante.jsp

**ServletReclamaRistorante.java** - Gestisce il reclamo di un ristorante da parte di un utente

**ServletStampaTuttoRistorante.java** - Inserisce le informazioni di un ristorante nel database

**VisualizzaRistorantiUtente2.java** - Mostra tutti i ristoranti reclamati con successo da un ristorante

**Week\_hours.java** - Contiene i metodi per la gestione dei giorni di apertura settimanali dei ristoranti

### Package "users"

**ExitProfilo.java** - Gestisce il log-out di ogni utente dal sito

**ServletLogin1.java** - Gestisce il log-in di ogni utente dal sito

**ServletRecuperoCredenziali.java** - Gestisce il recupero di credenziali di ogni utente dal sito

**ServletRegistrazione1.java** - Gestisce la registrazione di ogni utente dal sito

**User.java** - Contiene i metodi per la gestione di ogni utente dal sito

**UserUpdate.java** - Gestisce le modifiche dei dati di ogni utente dal sito

## 2.6 JSP

**CercaRistorantiHome.jsp** - Mostra una tabella DataTable dei risultati ottenuti dalla ricerca generale

**VisualizzaRistoranti.jsp** - Mostra i ristoranti inseriti nel database di cui l'utente è proprietario

**aggiungiRistorante.jsp** - Pagina per aggiungere un ristorante al database

**aggiungiRistoranteMap.jsp** - Pagina che contiene le funzioni per localizzare un ristorante durante l'inserimento

**aggiungiUtente.jsp** - Pagina per registrarsi al sito come Utenti Registrato

**autocompletamento.jsp** - Esegue la ricerca generale all'interno del database

**cancellaNotificaPhoto.jsp** - Notifica dell'avvenuta rimozione di una foto dopo una segnalazione

**cercaRistoHomeValue.jsp** - Mostra una tabella DataTable dei risultati ottenuti dalla ricerca per tipologia di cucina e posizione

**confermaDeletePhoto.jsp** - Notifica della rimozione di una foto dopo che è stata segnalata

**confermaReclamo.jsp** - Notifica dell'avvenuto reclamo di un ristorante

**confermaRecupero\_credenziali.jsp** - Notifica dell'invio della mail per il recupero delle credenziali

**confermaSegnalazionePhoto.jsp** - Notifica della segnalazione di una foto non consona all'Admin

**errore.jsp** - Pagina di errore con la possibilità di ritornare alla pagina precedentemente visualizzata

**erroreConnessione.jsp** - Notifica di un possibile errore nella connessione al database

**errorUnhandled.jsp** - Pagina semplice per gli errori non gestiti

**foto\_caricata\_con\_successo.jsp** - Notifica del caricamento con successo di una foto

**home.jsp** - Landing page del sito

**modificaRistorante.jsp** - Pagina per modificare i dati di un ristorante reclamato da un ristorante

**modificaRistoranteMap.jsp** - Pagina che contiene le funzioni per modificare i dati di localizzazione di un ristorante reclamato da un ristorante

**modificheEffettuate.jsp** - Notifica delle modifiche avvenute al profilo utente

**navbar.jsp** - Navbar del sito

**notifiche.jsp** - Pagina che mostra tutte le notifiche presenti

**notificheBar.jsp** - Popup che mostra un'anteprima delle notifiche presenti

**profiloUtente.jsp** - Pagina per modificare i dati del profilo di un utente

**reclamoAccettato.jsp** - Notifica del reclamo di un ristorante terminato con successo

**reclamoRifitutato.jsp** - Notifica del rifiuto del reclamo di un ristorante

**recupero\_credenziali.jsp** - Pagina per il recupero delle credenziali

**registrazioneEffettuate.jsp** - Notifica della registrazione terminata con successo

**repRifiutato.jsp** - Notifica del rifiuto di pubblicare una risposta di un ristorante ad un commento da parte dell'Admin

**rispostaAccettata.jsp** - Notifica dell'accettazione di una risposta di un ristorante ad un commento da parte dell'Admin

**rispostaInviataSuccesso.jsp** - Notifica dell'invio notifica invio risposta al admin per conferma

**rispostaRifiutata.jsp** - Notifica del rifiuto da parte dell'Admin di pubblicare una risposta del ristorante ad un commento

**ristorante.jsp** - Mostra la pagina contenente i dati di un ristorante con i suoi commenti e le sue foto

**ristorante\_inserito\_successo.jsp** - Notifica dell'inserimento di un ristorante nel database avvenuto con successo

**ristorante\_modificato\_successo.jsp** - Notifica delle modifiche fatte ad un ristorante completate con successo

### 3. Scelte progettuali

- Il nickname degli utenti pre-inseriti è uguale al loro cognome e la password è “password” facilitando così l’accesso al sito
- La password non viene scelta dall’utente al momento della registrazione ma viene spedita per email; l’utente potrà poi sostituirla nella propria pagina utente. Ciò è stato fatto per garantire che la prima password di ogni utente sia sempre sicura
- E’ possibile caricare tramite l’upload solo file *jpeg* e *png* (limitazione lato client) di dimensioni non esagerate. Abbiamo usato la libreria Oreilly per gestire l’upload e in particolare il metodo da noi usato ha tra i parametri di input la grandezza massima del file oltre la quale viene lanciata un’eccezione `IOException`. Tuttavia questa eccezione non viene correttamente lanciata e/o catturata e se il file è più grande del limite allora l’errore generato costituisce una perdita di connessione con il server.
- Alcuni bottoni sono stati spostati sulla navbar per facilitare l’utilizzo del sito

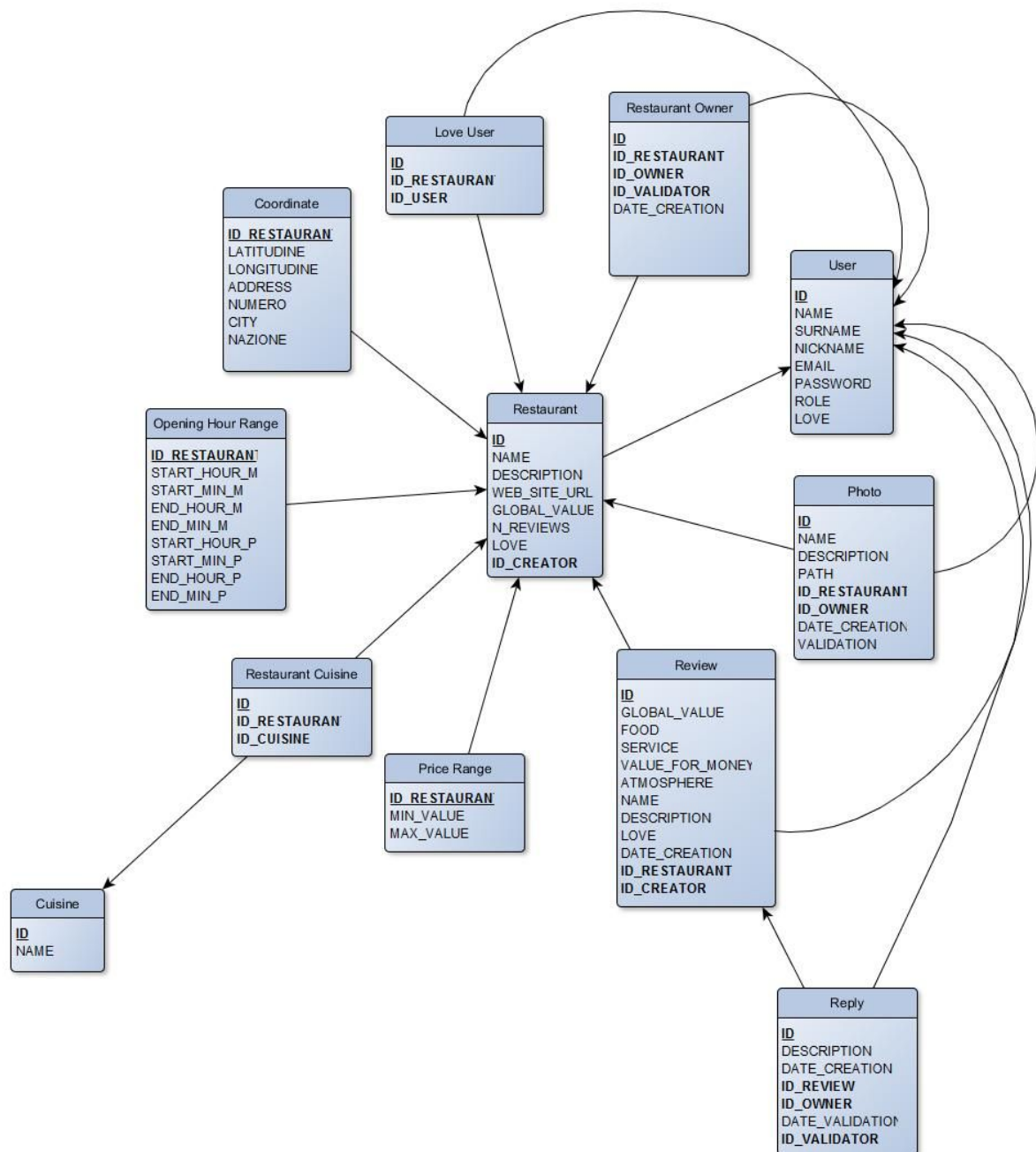
### 4. Istruzioni Database

Per l’implementazione del sito web è stato necessario creare un database che abbiamo creato in locale con PostgreSQL e al quale ci si può accedere con delle query SQL.

Il database si chiama **DoveCibo** (nome utente: postgres e password: postgres) e può essere popolato tramite la funzione Restore di pgAdmin III e il file DoveCiboCompleto80.backup contenuto nella cartella `/DoveCibo/web/database`.



## 5. ER del Database



**Chiave in grassetto e sottolineata** = Primary Key

**Chiave in grassetto** = Foreign Key

## **6. Conclusioni**

Il progetto ci ha permesso di comprendere più a fondo la filosofia che sta alla base della programmazione per il web e in particolare del protocollo MVC.

Siamo inoltre consapevoli che alcune funzionalità si sarebbero potute risolvere in maniera più efficace attraverso l'utilizzo di altre tecnologie (ad esempio JSTL) ma reputiamo fosse al di fuori della portata di questo progetto. La collaborazione all'interno del gruppo si è rivelata assai proficua in quanto le mancanze di uno sono state compensate dalle conoscenze di altri.