Introduzione alla programmazione WEB

AA 2016 – 2017

Docenti: Gino Perna e Stefano Chirico

Progetto finale

"Online Shopping"

Versione documento: 0.4~~2~~ – 2017-05-24~~10~~

SCOPO

Si vuole progettare un sistema di vendita online e classificazione dei venditori 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 a vostra scelta web-services, CSS, javascript

Formatted: Justified

Il sistema prevede:

- Un motore di ricerca che permetta di visualizzare un elenco di oggetti in vendita in base a varie tipologia di scelta tra cui dovranno esserci:
  - Per categoria;
  - Per oggetto;
  - Per venditore [5+]
- Un filtro sulla ricerca per:
  - Zona geografica
  - Per prezzo;
  - Per Customer Review.
- Un sistema di review degli oggetti venduti (da intendere come oggetti, non come qualità~~a'~~ del servizio di vendita, da dettagliare, con stelline);
- Un sistema di classificazione dei venditori che dipende dalla qualità~~a'~~ del servizio di vendita e postvendita ricevuto [5+] (devi tenere traccia che l'oggetto è~~e'~~ arrivato).
- La possibilità, per gli utenti registrati, per ciascun oggetto acquistato, di lasciare una recensione per il venditore. [5+]~~.~~
- La possibilit~~a~~à~~'~~, per gli utenti registrati, di lasciare una review sull'oggetto comperato.
- La possibilità, per i venditori, di rispondere alle recensioni che i clienti lasciano per i propri oggetti... [5+]
- La possibilità, per il venditore, di inserire una o più~~u'~~ di ~~una~~ fotografi~~e~~e legate agli oggetti in vendita;
- Un sistema di georeferenziazione che permette di visualizzare, su una mappa, i negozi che vendono quell'oggetto. [5+]
- L'internazionalizzazione del sito in base alle informazioni ricevute dal browser e/o scelta diretta dell'utente [7] (inteso come localizzazione dei menu e delle informazioni/header~~s~~ tabelle e ~~bottoni~~pulsanti; non devono essere localizzate ovviamente le recensioni e le descrizioni che rimangono nella lingua originale). L'internazionalizzazione è intesa che segua le regole spiegate a lezione o superiori (essendo attivi formalmente uno standard de-facto presentato ed uno standard dell'IANA che codifica le nazioni a due o più lettere).
- Il carrello di fine sessione, prima del pagamento, in cui potro' valutare eventuali cancellazioni di articoli.

- La gestione dell'acquisto, attraverso pagina fittizia di pagamento con carta di credito oppure ritiro diretto presso il negozio, per i soli oggetti con tale possibilitàà' di fruizione: I venditori possono esistere esclusivamente online, oppure avere un negozio fisico con indirizzo classico. Per questa tipologia di negozio, sarà possibile recuperare la merce direttamente in negozio. (da spiegare meglio)
- Un sistema di segnalazione anomalie (es: oggetti non arrivati) che invii al negozio e all'+amministratore la richiesta e che poi sia gestita dall'amministratore:
    - Segnanlao anomalia e viene recapitata a negozioNegozio + Amministratore.
    - L'amministratore decide come gestire
        - Soldi indietro
        - Segnalazione negativa al venditore
        - Non procedo
        - Rigetto l'anomalia

Formatted: Justified

Negozio:

- Nome;
- Descrizione;
- Fotografie pubblicate dal proprietario;
- Link al sito ufficiale del negozio;
- Coordinate geografiche (città, indirizzo, ecc…);
- Orari d'apertura; ??? Tipologia di spedizione? (solo per quelli che permettono ritiro in negozio)
- Valutazione totale (da 0 a 5) (il voto è inteso come voto globale dato al venditore).

Oggetti da vendere

Gli Oggetti classificati almeno da:

- Nome;
- Descrizione;
- Fotografie dell'oggetto (una, se [5+] piùu' di una);
- Revirew dell'oggetto venduto (da completare)
- Negozio che lo vende
- Categoria

UTENTI

Gli utenti saranno classificati almeno da:

- Nome;
- Cognome;
- Email;
- Password.
- AVATAR [7+]

Il portale dovrà prevedere quattro tipologie di utenti:

1. Utente anonimo;
2. Utente registrato;
3. Utente Venditore;
4. Utente amministratore.

UTENTE ANONIMO

Tale utente è quello che accede al sito senza fare login all'interno dello stesso. Tale utente può eseguire le ricerche (come già descritte in SCOPO). L'utente anonimo potrà registrarsi al portale via FORM di registrazione. L'utente anonimo può creare un carrello provvisorio ma non può comperare fino a quando avrà fatto login (o si sia registrato e poi fatto login).

UTENTE REGISTRATO

Tale utente è quello che ha fatto login al sito. L'utente registrato è quello che esegue la maggior parte delle iterazioni con il portale web. Oltre a poter svolgere le attività dell'utente anonimo potrà: comperare, fare recensioni di oggetti e negozi, fare segnalazioni di anomalia.

[Modificare/immettere il proprio avatar che deve essere visualizzato nelle recensioni 7+]

UTENTE VENDITORE

UTENTE che potrà: rispondere alle recensioni degli utenti registrati; modificare i dati relativi al proprio business; Inserire nuovi oggetti da vendere [5+]. Inoltre, potrà eseguire tutte le attività di un utente registrato.

UTENTE AMMINISTRATORE

Tale utente è il gestore del sito web e si occupa di gestire le iterazioni tra utenti registrati e utenti venditori in caso di anomalie. Tale utente gestisce le anomalie (vedi sopra); verrà notificato della richiesta di rimborso di un oggetto per cui dovrà valutarne la validità e, quindi, autorizzare il rimborso. Dovrà, quindi, notificare il venditore e decidere se penalizzarne la reputazione. ~~(da completare).~~

USE CASE (lo ~~o~~ use case è riferito all'intera applicazione, in funzione del numero di persone componenti il gruppo dovete verificare che lo use case sia congruente con quanto richiesto)

Menu principale

Ogni pagina del portale dovrà avere lo stesso menu principale personalizzato per tipologia di utente.

L'***utente anonimo*** avrà accesso al seguente menu principale:

- Pulsante "Registrati": che lo farà accedere alla form di registrazione;
- Pulsante "Accedi": che lo farà accedere alla form di login;
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale[7+]. (dovranno essere presenti almeno due lingue ed un locale di default. Attenzione a fare una traduzione di tutte le keyword~~s~~ (come spiegato a lezione con Google Translator) in modo che sia evidente all'utilizzatore la lingua scelta e non di solo alcune).

L'***utente registrato*** avrà accesso al seguente menu principale:

- Pulsante "Nome e Cognome";
  - Profilo: che permetterà di modificare i dati di profilo dell'utente;
  - Chiedere rimborso oggetto/segnalare anomalia: che permette di chiedere il rimborso di un oggetto da comprato;
  - Esci: che permetterà di fare logout dell'utente.
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale [7+].

L'***utente venditore*** avrà accesso al seguente menu principale:

- Pulsante "Notifiche": visualizzerà l'elenco delle ultime notifiche ricevute più un pulsante per accedere a tutte le notifiche;
- Pulsante "Nome e Cognome";
  - Profilo: che permetterà di modificare i dati di profilo dell'utente;

    - Il vostro negozio: che permetterà di accedere alla pagina di gestione del proprio negozio;
    - Esci: che permetterà di fare logout dell'utente.
    - Sistema di aggiunta/cancellazione oggetti in vendita [5+]

- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale.

L'**utente amministratore** avrà accesso al seguente menu principale:

- Pulsante "Notifiche": visualizzerà l'elenco delle ultime notifiche ricevute più un pulsante per accedere a tutte le notifiche e gestione delle controversie (da spiegare meglio)
- Pulsante "Nome e Cognome";
    - Profilo: che permetterà di modificare i dati di profilo dell'utente;
    - Esci: che permetterà di fare logout dell'utente.
- Pulsante "Lingua": che gli permetterà di modificare l'internazionalizzazione del portale.

Landing Page

Ogni tipologia di utente avrà accesso alla landing page del portale web che sarà composto da:

- Form di ricerca:
    - Text field con auto-completamento: che permetterà di inserire le parole da utilizzare per la ricerca degli oggetti/venditori;
    - Pulsante "Cerca": che avvierà la ricerca.
- Elenco di possibili filtri:
    - Filtro per vicinanza (solo per oggetti con ritiro)
    - Filtro per prezzo
    - Filtro per review

Form di registrazione

L'utente anonimo avrà accesso a tale form che dovrà permettere la registrazione dell'utente e dovrà prevedere:

- Email: che verrà utilizzato anche come username di login;
- Password: che dovrà avere dei meccanismi di scelta di una password solida;
- L'accettazione (anche non immediata, ma prima della prima connessione) alla normativa sulla privacy da accettare;
- Pulsante "Iscriviti": che invierà la richiesta d'iscrizione;
- Pulsante "Annulla": che rimanderà l'utente alla pagina da cui era arrivato.

L'iscrizione al portale dovrà essere validato attraverso il click di un link inviato per email.

Form di login

L'utente anonimo avrà accesso a tale form che dovrà permettere il login dell'utente e dovrà prevedere:

- Username: l'email con cui l'utente si è registrato;
- Password: la password con cui l'utente si è registrato;
- Modalità reset della password se l'utente se l'è dimenticata;
- Pulsante "Accedi": che effettuerà il login dell'utente e ritornerà alla pagina di provenienza;
- Pulsante "Annulla": che rimanderà l'utente alla pagina da cui era arrivato.

Form di ricerca (da vedere se lo stesso di cui alla landing page)

Ogni tipologia di utente avrà accesso a tale form che dovrà permettere la ricerca di oggetti e dovrà prevedere:

- Text field (uno o più consigliati il meno possibile): verranno inserite le parole da ricercare;

- Pulsante “Cerca”: avvierà la ricerca e rimanderà l’utente alla pagina dei risultati.

Il campo di ricerca avrà l’auto completamento che aiuterà l’utente nella scelta delle parole da ricercare.

La ricerca dovrà poter includere almeno: la tipologia di oggetto/oggetto, via/città/regione.

Pagina dei risultati

Ogni tipologia di utente avrà accesso alla pagina dei risultati che dovrà prevedere:

- L’elenco degli oggetti ordinati per prezzo; per ogni oggetto dovrà essere visibile:
  - Nome del venditore con link alla pagina del negozio;
  - Una foto;
  - Il voto totale dell’oggetto (da 1 a 5);
  - Il numero di recensioni ricevute;
  - Link alla mappa (se serve)
  - Prezzo.
- Possibilità di modificare l’ordinamento della classifica:
  - Stelline(review);
  - Fascia di prezzo.

Pagina della mappa

Dovrà visualizzare una mappa di Google centrata sul negozio selezionato [per tutti] e [seguente per 5+] che presenti tutti i negozi presenti in zona che vendono lo stesso oggetto.

Pagina del negozio

Questa pagina dovrà avere almeno le seguenti informazioni:

- Nome;
- Valutazione totale;
- indirizzo
- Almeno una fotografia;
- Elenco delle ultime recensioni;

Pagina delle notifiche

L’utente negoziante visualizzerà, in questa pagina, tutte le notifiche di avvenuta recensione o di anomalie.

Da ogni notifica potrà accedere, rispettivamente, a:

- La pagina di risposta alle recensioni;

L’utente amministratore visualizzerà, in questa pagina, tutte le notifiche di anomalie con il sistema di gestione annesso

Per ogni notifica potrà:

- Rigettarla con motivo;
- Gestire la restituzione del credito all’utente (il credito rimane a disposizione dell’utente per nuovi acquisti) [7+]
- Le decisioni saranno ~~notfictenotificate~~ ad entrambi
- Eventuale dare giudizio negativo al negozio
-

Precompilazione database (da completare)

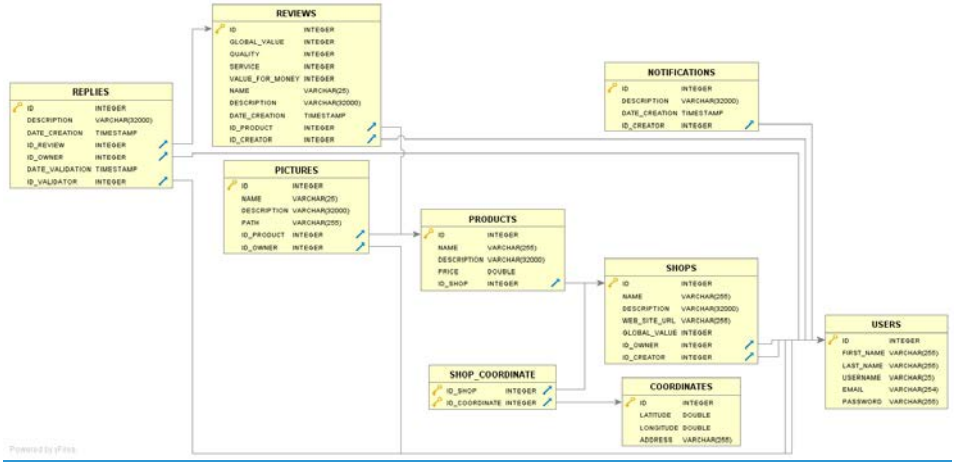
Ciascun progetto (gruppo) dovrà presentare il proprio lavoro con un database già popolato per almeno:

- Quattro regioni amministrative con almeno ciascuna 5 città differenti (anche in differenti stati se volete)
- Per ciascuna città (vista come area estesa attorno alla città) almeno 10 negozi diversi.
- Per ciascuna città la somma delle recensioni (per il totale dei 10 negozi) dovrà essere di almeno 20. Dovranno inoltre essere presenti almeno 4 negozi per città con foto pubblicate da utenti.
-
- 10 utenti minimo già registrati

Formatted: List Paragraph, Justified, Bulleted + Level: 1 + Aligned at: 0.25" + Indent at: 0.5"

Database

Per agevolare chi non avesse ancora completato il corso di DB viene di seguito indicato un possibile diagramma ER di un database che, ragionevolmente, sostiene l'applicazione.



~~DA METTERE STEFANO CHIRICO~~

Figura 1: Diagramma ER database

Il DB copre le ~~intere~~ parte importante delle specifiche descritte precedentemente. Il contenuto delle tabelle è volutamente ridotto al minimo ed è indicativo. Anche la struttura del database stesso è del tutto indicativa e può/deve essere modificata/completata a piacimento secondo le vostre esigenze (ovviamente bisogna saper sostenere le scelte fatte).

Manca ovviamente la parte di tabelle relative al pagamento/spedizione ed alle lamentele sulla spedizione

Hint vari

- String similarity: https://github.com/tdebatty/java-string-similarity. Si tenga presente che per una ricerca efficiente (i.e. motore di ricerca vengono impiegati algoritmi che pre-memorizzano l'indice scelto e quindi ne fanno una sola comparazione al momento della query).
- Geocoding:
  https://developers.google.com/maps/documentation/geocoding/intro#GeocodingRequests e https://developer.mapquest.com/plan_purchase/steps/business_edition/business_edition_free/register + http://www.mapquestapi.com/geocoding/
  Ricordarsi che bisogna pre-memorizzare le coordinate geografiche o box per poi fare la ricerca su numeri.
- Riempite il database con dati plausibili, evitate nomi tipo 'negozio1 negozio2' ecc… cercate di rendere presentabile e credibile l'implementazione (in totale: 2*5* 20 recensioni, suggerimenti per recuperare recensioni/ ricerca: ebay, http://global.rakuten.com/en/, http://www.fromjapan.co.jp/en/
-

Formatted: Justified

Formatted: Justified

Field Code Changed

Raccomandazioni

Il sito deve essere progettato in modalità responsive e dimostrato funzionante per risoluzioni pc/tablet/telefono attraverso il device o la modalità relativa sul browser.

Il progetto dovrà essere mostrato con un database minimo credibile ed intelligentemente popolato per quanto richiesto e nel numero di voci richieste nel presente testo come minimo.

Documentazione

Dovrà essere presentata una documentazione del progetto illustrante le caratteristiche e le scelte effettuate

Ricordarsi che per una buona valutazione:

- all'esame servono due browser per dimostrare le funzionalità
- gli uri di tutti i file devono essere relativi e non assoluti sul file system
- La documentazione presentata deve essere efficace
- Attenzione a chi userà javascript: deve sapere sostenere le scelte implementative *tutti i componenti del gruppo, e non deve violare o rendere inutile il paradigma MVC
- Il database con cui implementare il progetto è a libera scelta (postgres/mysql/derby/~~–~~ect..~~-~~), ricordandosi che nella consegna dovrà essere data una copia anche del database

Regole di presentazione e consegna progetto

~~TBD, indicativamente c~~Consegna progetto 2 giorni prima del giorno dell'appello in cui vorrete sostenere l'esame, da un solo componente del gruppo con chiara indicazione dei partecipanti al gruppo. Consegna progetto elettronica tramite link a strumento di archiviazione cloud a vostra scelta.

Dare un NIKNAME al progetto in modo che sia identificabile facilmente rispetto agli altri

L'esame prevede che:

- Sia Presentato il progetto come sopra
- Il giorno dell'esame si farà~~a'~~ l'appello e stabilito l'ordine di presentazione. Eventuali problemi di orario saranno da concordare (anche in anticipo) via mail nel rispetto dei reciproci impegni e delle liste di partecipanti all'esame
- Dovrà~~a'~~ quindi essere effettuata una presentazione del progetto, da tutti i componenti, che spiegheranno (il relatore del progetto può~~o'~~ essere uno o più~~u'~~ degli appartenenti al progetto), la filosofia di implementa~~n~~zione e dimostreranno le funzionalità~~a'~~.
- Verranno fatte domande sulle funzionalità~~a'~~ o sulle scelte implementative e potrà~~a'~~ rispondere chiunque del gruppo.

Se~~,~~ per qualche motivo, il gruppo non fosse tutto presente, i componenti mancanti dovranno sostenere l'esame come se il resto del non gruppo esistesse, e dovranno dimostrare padronanza di tutto quanto presentato.

Formatted: Heading 3

~~SCHEMA DATABASE~~

DDL Database (versione DERBY)

Formatted: English (United Kingdom)

Formatted: No Spacing

```
CREATE TABLE USERS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    FIRST_NAME VARCHAR(255) NOT NULL,
    LAST_NAME VARCHAR(255) NOT NULL,
    USERNAME VARCHAR(25) NOT NULL,
    EMAIL VARCHAR(254) NOT NULL,
    PASSWORD VARCHAR(255) NOT NULL,
PRIMARY KEY (ID),
UNIQUE (USERNAME),
UNIQUE (EMAIL)
);

CREATE TABLE SHOPS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(255) NOT NULL,
    DESCRIPTION VARCHAR(32000),
    WEB_SITE_URL VARCHAR(255),
    GLOBAL_VALUE INTEGER CONSTRAINT GLOBAL_VALUE_CHECK CHECK (GLOBAL_VALUE >= 0 AND
GLOBAL_VALUE <= 5),
    ID_OWNER INTEGER,
    ID_CREATOR INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID),
FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID)
);

CREATE TABLE COORDINATES (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    LATITUDE FLOAT NOT NULL,
    LONGITUDE FLOAT NOT NULL,
    ADDRESS VARCHAR(255) NOT NULL,
PRIMARY KEY (ID)
);

CREATE TABLE SHOP_COORDINATE (
    ID_SHOP INTEGER NOT NULL,
    ID_COORDINATE INTEGER NOT NULL,
PRIMARY KEY (ID_SHOP, ID_COORDINATE),
FOREIGN KEY (ID_SHOP) REFERENCES SHOPS (ID),
FOREIGN KEY (ID_COORDINATE) REFERENCES COORDINATES (ID)
);

CREATE TABLE PRODUCTS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(255) NOT NULL,
    DESCRIPTION VARCHAR(32000) NOT NULL,
    PRICE FLOAT,
    ID_SHOP INTEGER NOT NULL,
PRIMARY KEY (ID),
```

```
FOREIGN KEY (ID_SHOP) REFERENCES SHOPS (ID)
);

CREATE TABLE PICTURES (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(25) NOT NULL,
    DESCRIPTION VARCHAR (32000),
    PATH VARCHAR(255) NOT NULL,
    ID_PRODUCT INTEGER NOT NULL,
    ID_OWNER INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (ID_PRODUCT) REFERENCES PRODUCTS (ID),
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID)
);

CREATE TABLE REVIEWS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    GLOBAL_VALUE INTEGER NOT NULL CONSTRAINT REVIEW_GLOBAL_VALUE_CHECK CHECK
(GLOBAL_VALUE >= 1 AND GLOBAL_VALUE <= 5),
    QUALITY INTEGER CONSTRAINT REVIEW_QUALITY_CHECK CHECK (QUALITY IS NULL OR (QUALITY >= 1
AND QUALITY <= 5)),
    SERVICE INTEGER CONSTRAINT REVIEW_SERVICE_CHECK CHECK (SERVICE IS NULL OR (SERVICE >= 1
AND SERVICE <= 5)),
    VALUE_FOR_MONEY INTEGER CONSTRAINT REVIEW_VALUE_FOR_MONEY_CHECK CHECK
(VALUE_FOR_MONEY IS NULL OR (VALUE_FOR_MONEY >= 1 AND VALUE_FOR_MONEY <= 5)),
    NAME VARCHAR(25),
    DESCRIPTION VARCHAR(32000),
    DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    ID_PRODUCT INTEGER NOT NULL,
    ID_CREATOR INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (ID_PRODUCT) REFERENCES PRODUCTS (ID),
FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID)
);

CREATE TABLE USERS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(255) NOT NULL,
    SURNAME VARCHAR(255) NOT NULL,
    NICKNAME VARCHAR(25) NOT NULL,
    EMAIL VARCHAR(254) NOT NULL,
    PASSWORD VARCHAR(255) NOT NULL,
PRIMARY KEY (ID),
UNIQUE (NICKNAME),
UNIQUE (EMAIL)
);

CREATE TABLE PRICE_RANGES (
    ID INTEGER NOT NULL,
    NAME VARCHAR(25) NOT NULL,
    MIN_VALUE FLOAT,
    MAX_VALUE FLOAT,
```

Formatted: English (United Kingdom)

Formatted: English (United Kingdom)

Formatted: No Spacing

```
PRIMARY KEY (ID),
UNIQUE (NAME),
CONSTRAINT PRICE_RANGE_CHECK CHECK ((MIN_VALUE IS NULL AND MAX_VALUE IS NOT NULL) OR
(MIN_VALUE IS NOT NULL AND MAX_VALUE IS NULL) OR (MIN_VALUE < MAX_VALUE))
);

CREATE TABLE RESTAURANTS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(255) NOT NULL,
    DESCRIPTION VARCHAR(32000),
    WEB_SITE_URL VARCHAR(255),
    GLOBAL_VALUE INTEGER CONSTRAINT GLOBAL_VALUE_CHECK CHECK (GLOBAL_VALUE >= 0 AND
GLOBAL_VALUE <= 5),
    ID_OWNER INTEGER,
                ID_CREATOR INTEGER NOT NULL,
    ID_PRICE_RANGE INTEGER,
PRIMARY KEY (ID),
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID),
FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID),
FOREIGN KEY (ID_PRICE_RANGE) REFERENCES PRICE_RANGES (ID)
);

CREATE TABLE PHOTOS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(25) NOT NULL,
    DESCRIPTION VARCHAR (32000),
    PATH VARCHAR(255) NOT NULL,
    ID_RESTAURANT INTEGER NOT NULL,
    ID_OWNER INTEGER NOT NULL,
PRIMARY KEY (ID),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID)
);

CREATE TABLE COORDINATES (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    LATITUDE FLOAT NOT NULL,
    LONGITUDE FLOAT NOT NULL,
    ADDRESS VARCHAR(255) NOT NULL,
PRIMARY KEY (ID)
);

CREATE TABLE RESTAURANT_COORDINATE (
    ID_RESTAURANT INTEGER NOT NULL,
    ID_COORDINATE INTEGER NOT NULL,
PRIMARY KEY (ID_RESTAURANT, ID_COORDINATE),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_COORDINATE) REFERENCES COORDINATES (ID)
);

CREATE TABLE OPENING_HOURS_RANGES (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
```

```
    DAY_OF_THE_WEEK INTEGER CONSTRAINT DAY_OF_THE_WEEK_CHECK CHECK (DAY_OF_THE_WEEK
>= 1 AND DAY_OF_THE_WEEK <= 7),
    START_HOUR TIME NOT NULL,
    END_HOUR TIME NOT NULL,
PRIMARY KEY (ID),
CONSTRAINT RANGE_CHECK CHECK (START_HOUR < END_HOUR)
);

CREATE TABLE OPENING_HOURS_RANGE_RESTAURANT (
    ID_RESTAURANT INTEGER NOT NULL,
    ID_OPENING_HOURS_RANGE INTEGER NOT NULL,
PRIMARY KEY (ID_RESTAURANT, ID_OPENING_HOURS_RANGE),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_OPENING_HOURS_RANGE) REFERENCES OPENING_HOURS_RANGES (ID)
);

CREATE TABLE CUISINES (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    NAME VARCHAR(25) NOT NULL,
PRIMARY KEY (ID)
);

CREATE TABLE RESTAURANT_CUISINE (
    ID_RESTAURANT INTEGER NOT NULL,
    ID_CUISINE INTEGER NOT NULL,
PRIMARY KEY (ID_RESTAURANT, ID_CUISINE),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_CUISINE) REFERENCES CUISINES (ID)
);

CREATE TABLE REVIEWS (
    ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,
    GLOBAL_VALUE INTEGER NOT NULL CONSTRAINT REVIEW_GLOBAL_VALUE_CHECK CHECK
(GLOBAL_VALUE >= 1 AND GLOBAL_VALUE <= 5),
    FOOD INTEGER CONSTRAINT REVIEW_FOOD_CHECK CHECK (FOOD IS NULL OR (FOOD >= 1 AND FOOD
<= 5)),
    SERVICE INTEGER CONSTRAINT REVIEW_SERVICE_CHECK CHECK (SERVICE IS NULL OR (SERVICE >= 1
AND SERVICE <= 5)),
    VALUE_FOR_MONEY INTEGER CONSTRAINT REVIEW_VALUE_FOR_MONEY_CHECK CHECK
(VALUE_FOR_MONEY IS NULL OR (VALUE_FOR_MONEY >= 1 AND VALUE_FOR_MONEY <= 5)),
    ATMOSPHERE INTEGER CONSTRAINT REVIEW_ATMOSPHERE_CHECK CHECK (ATMOSPHERE IS NULL OR
(ATMOSPHERE >= 1 AND ATMOSPHERE <= 5)),
    NAME VARCHAR(25),
    DESCRIPTION VARCHAR(32000),
    DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,
    ID_RESTAURANT INTEGER NOT NULL,
    ID_CREATOR INTEGER NOT NULL,
    ID_PHOTO INTEGER,
PRIMARY KEY (ID),
FOREIGN KEY (ID_RESTAURANT) REFERENCES RESTAURANTS (ID),
FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID),
FOREIGN KEY (ID_PHOTO) REFERENCES PHOTOS (ID)
);
```

~~CREATE TABLE REPLIES (~~
~~ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,~~
~~DESCRIPTION VARCHAR(32000) NOT NULL,~~
~~DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,~~
~~ID_REVIEW INTEGER NOT NULL,~~
~~ID_OWNER INTEGER NOT NULL,~~
~~DATE_VALIDATION TIMESTAMP,~~
~~ID_VALIDATOR INTEGER,~~
~~PRIMARY KEY (ID),~~
~~FOREIGN KEY (ID_REVIEW) REFERENCES REVIEWS (ID),~~
~~FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID),~~
~~FOREIGN KEY (ID_VALIDATOR) REFERENCES USERS (ID)~~
~~);~~

~~CREATE TABLE USER_REVIEW_LIKES (~~
~~ID_USER INTEGER NOT NULL,~~
~~ID_REVIEW INTEGER NOT NULL,~~
~~ID_CREATOR INTEGER NOT NULL,~~
~~LIKE_TYPE INTEGER NOT NULL CONSTRAINT LIKE_TYPE_CHECK CHECK (LIKE_TYPE >= 0 AND LIKE_TYPE <= 1),~~
~~DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,~~
~~PRIMARY KEY (ID_USER, ID_REVIEW, ID_CREATOR),~~
~~FOREIGN KEY (ID_USER) REFERENCES USERS (ID),~~
~~FOREIGN KEY (ID_REVIEW) REFERENCES REVIEWS (ID),~~
~~FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID)~~
~~);~~

~~CREATE TABLE REPLIES (~~
~~ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,~~
~~DESCRIPTION VARCHAR(32000) NOT NULL,~~
~~DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,~~
~~ID_REVIEW INTEGER NOT NULL,~~
~~ID_OWNER INTEGER NOT NULL,~~
~~DATE_VALIDATION TIMESTAMP,~~
~~ID_VALIDATOR INTEGER,~~
~~PRIMARY KEY (ID),~~
~~FOREIGN KEY (ID_REVIEW) REFERENCES REVIEWS (ID),~~
~~FOREIGN KEY (ID_OWNER) REFERENCES USERS (ID),~~
~~FOREIGN KEY (ID_VALIDATOR) REFERENCES USERS (ID)~~
~~);~~

~~CREATE TABLE NOTIFICATIONS (~~
~~ID INTEGER NOT NULL GENERATED BY DEFAULT AS IDENTITY,~~
~~DESCRIPTION VARCHAR(32000),~~
~~DATE_CREATION TIMESTAMP NOT NULL DEFAULT CURRENT_TIMESTAMP,~~
~~ID_CREATOR INTEGER NOT NULL,~~
~~PRIMARY KEY (ID),~~
~~FOREIGN KEY (ID_CREATOR) REFERENCES USERS (ID)~~
~~);Progetto finale query tabelle db.ddl~~

~~DA INSERIRE~~

Formatted: English (United States)

Formatted: English (United Kingdom)

Formatted: English (United States)

Formatted: English (United Kingdom)