**Introduction:**

What is your application?

Our application is a restaurant dish review application that allows users to review single dishes at restaurants in contrast to reviewing the restaurant as a whole.

Why do you want to choose this application, instead of others, as the course project (motivation)?

We chose this application instead of others because we love food. Also, we recognized the existence of apps like “Yelp” or “Zomato” which provide rankings for the restaurant as a whole. The importance of an individual dish or food item is lost with these rankings. For example, a wings restaurant may also have good burgers. Differentiating between the individual items is a factor that is overlooked.

What are the key components in your project?

The key components in this project are being able to review dishes from various restaurants with a star rating out of 5 as well as a text description of why the user decided to give that rating.

**Database details:**

How do you design the database?

We designed out database with four main entities. We have users, reviews, menu items, as well as restaurants. In our database, many reviews can come from one user. Also, many reviews can exist for one menu item. A restaurant can sell many items from the menu. Because menu items may have similar names at different restaurants, it is a weak entity with the foreign key set as the restaurant ID because one restaurant will not have two different menu items with the same exact name.

What is your E-R model and relational model?

Diagram

Description automatically generated

Table

Description automatically generated

What are the tables you include in the database?

CREATE TABLE "dishes\_dish" (

"id" integer NOT NULL,

"title" varchar(50) NOT NULL,

"price" decimal NOT NULL,

"description" text NOT NULL,

"numReviews" integer NOT NULL,

"resID\_id" varchar(50) NOT NULL,

PRIMARY KEY("id" AUTOINCREMENT),

CONSTRAINT "menuItem" UNIQUE("title","resID\_id"),

FOREIGN KEY("resID\_id") REFERENCES "restaurants\_restaurant"("name") DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE "reviews\_review" (

"id" integer NOT NULL,

"review\_text" text NOT NULL,

"date\_posted" datetime NOT NULL,

"author\_id" integer NOT NULL,

"dish\_id" integer NOT NULL,

"restaurant\_id" varchar(50) NOT NULL,

"rating" integer unsigned NOT NULL CHECK("rating" >= 0),

PRIMARY KEY("id" AUTOINCREMENT),

FOREIGN KEY("dish\_id") REFERENCES "dishes\_dish"("id") DEFERRABLE INITIALLY DEFERRED,

FOREIGN KEY("author\_id") REFERENCES "auth\_user"("id") DEFERRABLE INITIALLY DEFERRED,

FOREIGN KEY("restaurant\_id") REFERENCES "restaurants\_restaurant"("name") DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE "restaurants\_restaurant" (

"name" varchar(50) NOT NULL,

"totalReviews" integer NOT NULL,

"cuisine" varchar(2) NOT NULL,

PRIMARY KEY("name")

);

CREATE TABLE "profiles\_profile" (

"id" integer NOT NULL,

"image" varchar(100) NOT NULL,

"user\_id" integer NOT NULL UNIQUE,

PRIMARY KEY("id" AUTOINCREMENT),

FOREIGN KEY("user\_id") REFERENCES "auth\_user"("id") DEFERRABLE INITIALLY DEFERRED

);

CREATE TABLE "locations\_location" (

"address" varchar(50) NOT NULL,

"numReviews" integer NOT NULL,

"name\_id" varchar(50) NOT NULL,

PRIMARY KEY("address"),

CONSTRAINT "uniqueLocation" UNIQUE("name\_id","address"),

FOREIGN KEY("name\_id") REFERENCES "restaurants\_restaurant"("name") DEFERRABLE INITIALLY DEFERRED

);

How do you design your database to accommodate functional dependencies?

Are your tables in BCNF or 3NF? How about other constraints?

**Functionality details:**

What are the basic functions and advanced functions in your database and how do you implement them?

**Implementation details:**

What languages and platform you've chosen to do the implementation?

Python 3

HTML5

SQLite

How do you implement the front-end Web interface and the application logic?

The front-end web interface is implemented with HTML and the application logic is implemented with Python.

How does the front-end Web interface interact with the backend database?

The front-end web interface interacts through the Django framework. The HTML web interface passes information to the Python functions which then query our database for the desired results. The query is passed back to our functions in Python and then to HTML to be presented to the user.

Students are encouraged to upload their code to some online repositories, such as  
Dropbox, Github, Bitbucket, and provide the link in the report.

https://github.com/lifeunsubscribe/dopestDish

**Experiences:**

What have you learnt from this project?

Designing a database system with many parts can be challenging. Working in teams during a pandemic is difficult when there is no prior training. As undergraduate students, many of our projects before this past year, we have been able to meet face to face with our respective teams and this allows the team to always be in sync.

How have you solved hard problems in this project?

How to extend your project to more advanced, mature systems in the future?  
  
**References:**

Any resource you have turned to for help during the implementation of your project.