Database Design Document for Food2Go App

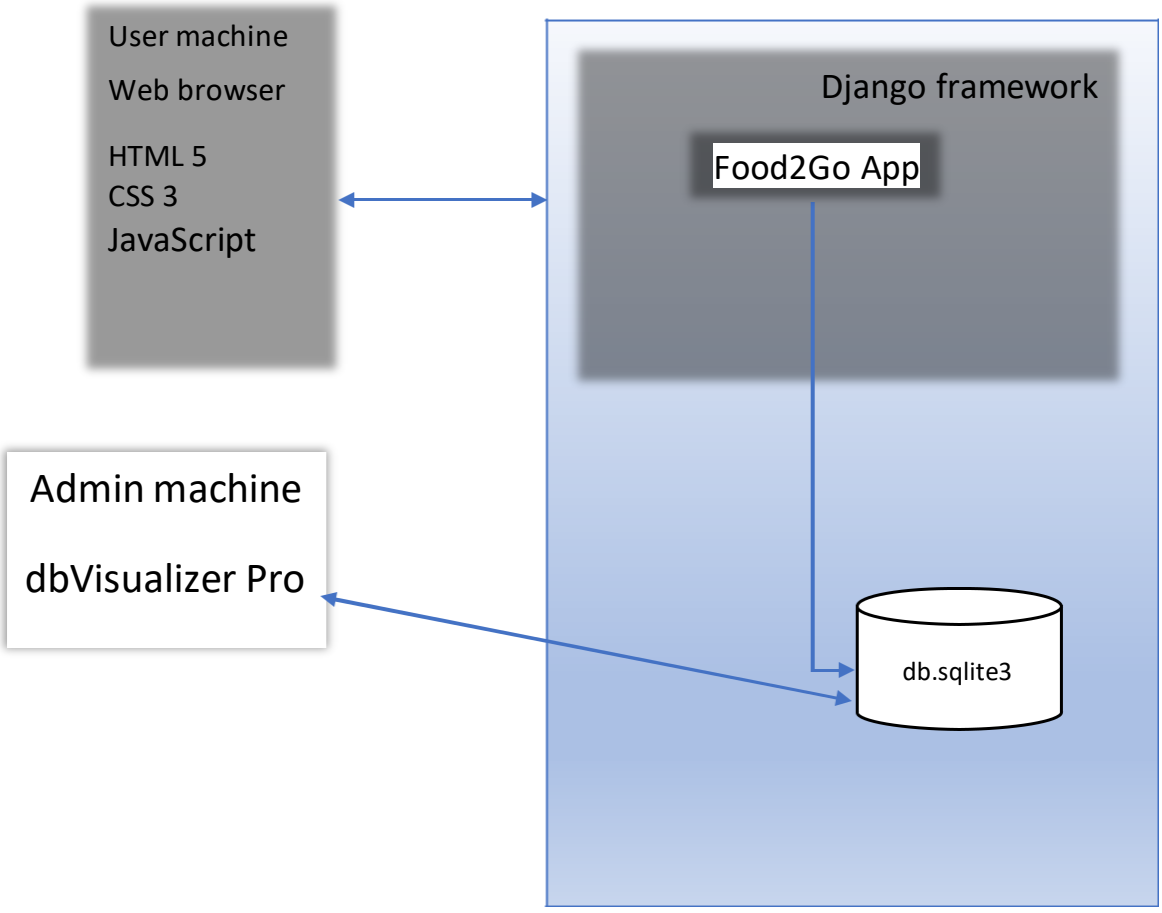
Approved on 11/02/2019

Overview

The Food2Go App is an online food ordering system that enables ease for the customers. It overcomes the disadvantages of the traditional queueing system. The proposed system is a medium to order online food hassle-free from restaurants for pickup and delivery service. The Food2Go App sets up a food menu online, and customers can easily place the order as per their wish. Also, with a food menu, customers can easily track the orders. This app also provides a feedback system in which a user can rate the food items. The payment can be made online or pay-on-delivery system. For more secured separate ordering accounts are maintained for each user by providing them a username and a password.

The Food2Go App will be developed as a web application. Client part runs in a web browser and communicates with the server-side software consisting of a set of python classes built using Django framework to communicate with the database.

The figure below shows the architecture of the Food2Go Application.



Database and Application Software Utilities

Vendor	Product	Version	Comments
SQLite	SQLite3		Database Management System
Django	Django	2.0.8	Framework
Python	Python	3.7	Application Software

Table 1: Application Software Utilities

Database Design Considerations:

Food2Go App uses SQLite to store and retrieve data in a light-weight database on the website.

Some benefits of using SQLite for local storage

✔ SQLite is light-weight and self-contained. It's a code library without any other dependencies. There's nothing to configure.

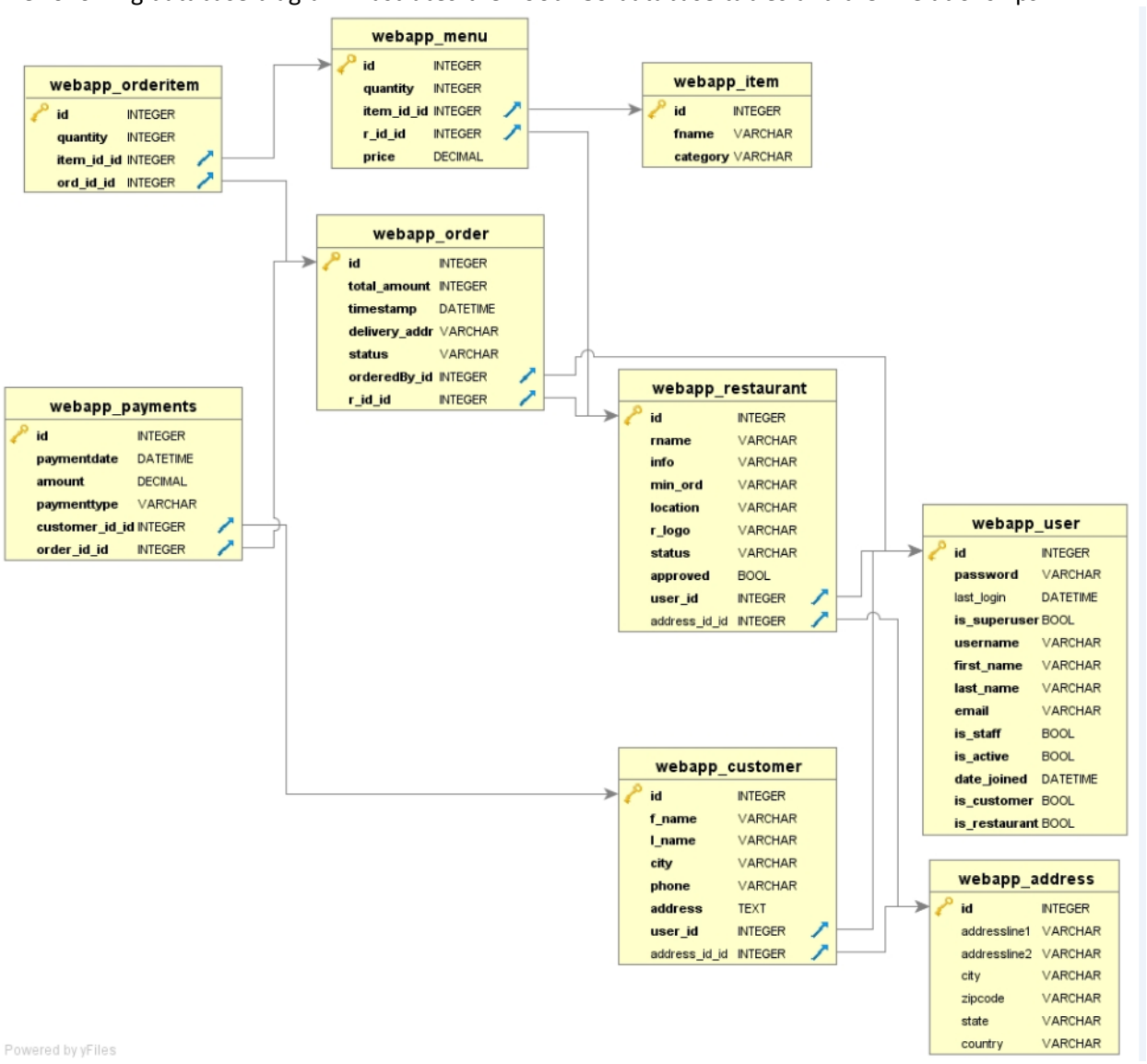
✔ There's no database server. The client and the server run in the same process.

✔ SQLite is in the public domain so you can freely use and distribute it with your app.

✔ SQLite works across platforms and architectures.

ER- Diagram:

The following database diagram illustrates the Food2Go database tables and their relationships.



There are 9 tables in the Food2Go database.

- webapp_user table stores the user details of the app, it can be either customer / partner.
- webapp_customer table stores the details related to the customer who actually creates an order.
- webapp_restaurant table stores the details of the restaurant user who gets access to add/update the menu.
- webapp_address table stores the address details.
- webapp_payments table stores the payment details of the order.

- webapp_menu table stores the details of the menu of a restaurant.
- webapp_order table stores the details of the order details.
- webapp_order table item stores the details of the items in an order.
- webapp_item table store the items in a restaurant.