**Yummy+**

Guanxiaoxiong Liu

<https://github.com/lgxxabc/50900_Project.git>

3/26/2022

**Revision History**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Date** | **Description** | **Author** | **Reviewer** | **Version** |
| 2022/03/18 | Initial release of this document | G.Liu |  | V1.0.0 |

**Table of Contents**

[1. Introduction 4](#_Toc99803299)

[**1.1** **Overview** 4](#_Toc99803300)

[2. Database design 5](#_Toc99803301)

[**2.1** **Relational Database Design** 5](#_Toc99803302)

[**2.2** **Entity Relationship Diagram (ERD) Model** 6](#_Toc99803303)

[**2.3** **To be complete…** 7](#_Toc99803304)

# Introduction

* 1. **Overview**

I have created an online food ordering platform called Yummy+. I need to create a database solution that keeps track of customer data, online shopping cart data, restaurant data, menu item data, and order item data. I’m interested in this data because I need to know who (customer info) is using which cart (cart ID) and which dishes have been ordered (order items) from which restaurant's menu (menu items) (restaurant info). I will manually enter restaurant data and menu item data according to the information provided by the restaurants. While customer data, shopping cart data, and order item data are generated when a customer produces an order, as the business owner, I need to see all the data to understand how my business is doing. The sales team needs to view customer data and order item data to provide the right deals and incentives to encourage customers to use our online ordering platform. They need to look at customer data and order item data to understand which foods are most popular at which restaurants and which are ordered in smaller quantities. Armed with this information, we can appropriately adjust the selection of restaurants and food on the platform or change the price of food to stimulate sales. Website administrators need to understand the behavior of users visiting the website based on customer data to provide support to better meet the needs of users. For example, we can count which users have ordered the most meals from which restaurant on our website, which may indicate that they like this restaurant the most. Then we can recommend new menus to users in a more targeted manner to improve their user experience. In addition, back-end engineers need user data to count which day of the week the website has the most traffic and the highest number of concurrent users, so they can optimize the performance of the website during that period, such as adding servers, doing load balancing, doing caching, and so on. Front-end engineers need restaurant data and menu item data to analyze which web pages are visited the most to determine which pages need to be optimized.

# Database design

* 1. **Relational Database Design**

For my business, I will have the following entities:

1. Customer Info – describes the Email, First Name, Last Name, Password, and Cart ID(FK)
2. Cart Info – describes Cart ID, Date, and Total Price
3. Order Item Info – describes Order Item ID, Quantity, Price, Menu Item ID(FK), and Cart ID(FK)
4. Menu Item Info – describes Menu Item ID, Name, Price, Item’s description, and Restaurant ID(FK)
5. Restaurant Info – describes Restaurant ID, Address, Name, Phone# and Image URL

|  |  |
| --- | --- |
| **Customer Info** | |
| Email (PK) | Varchar (30) |
| First Name | Varchar (20) |
| Last Name | Varchar (20) |
| Password | Varchar (20) |
| Cart ID(FK) | Integer |

|  |  |
| --- | --- |
| **Cart Info** | |
| Cart ID(PK) | Integer |
| Date | Date |
| Total Price | Float (7, 2) |

|  |  |
| --- | --- |
| **Order Item Info** | |
| ID(PK) | Integer |
| Quantity | Integer |
| Price | Float (7, 2) |
| Menu Item ID(FK) | Integer |
| Cart ID(FK) | Integer |

|  |  |
| --- | --- |
| **Menu Item Info** | |
| ID(PK) | Integer |
| Name | Varchar (40) |
| Price | Float (7, 2) |
| Description | Blob |
| Restaurant ID(FK) | Integer |

|  |  |
| --- | --- |
| **Restaurant Info** | |
| ID(PK) | Integer |
| Address | Varchar (60) |
| Name | Varchar (40) |
| Phone | Varchar (20) |
| Image URL | Blob |

* 1. **Entity Relationship Diagram (ERD) Model**

ERD Diagram using Unified Modeling Language (UML):

Diagram

Description automatically generated

ERD Diagram using Crow’s Foot notation:

Diagram

Description automatically generated

* 1. **To be complete…**