Software Design Specification

for

MenuWhiz

Prepared by

Zahin Ahmed

Nafisa Tafshir

Md. Jubaer Khan

28.08.2020

Table of Contents

[1. Introduction 3](#_Toc49525319)

[1.1 Document Conventions 3](#_Toc49525320)

[2. Description of Design Components 4](#_Toc49525321)

[2.1 Database 4](#_Toc49525322)

[2.2 User Interface 4](#_Toc49525323)

[2.3 API 4](#_Toc49525324)

[3. High-Level Component Design 5](#_Toc49525325)

[4. Class diagrams 5](#_Toc49525326)

[4.1 Detailed Class Diagrams 6](#_Toc49525327)

[4.2 Detailed Class Descriptions 6](#_Toc49525328)

[5. Database Design 9](#_Toc49525329)

[5.1 ER Diagram 9](#_Toc49525330)

[5.2 Relational table: 9](#_Toc49525331)

[6. Implementation Plan 9](#_Toc49525332)

# Introduction

The purpose of this Software design Specification (SDS) is to provide a description of the design of MenuWhiz. It includes design components, high level component design, class diagrams and their descriptions, details about ER diagrams to proceed with an understanding of what is to be built and how it is implemented in our project.

## Document Conventions

Main Section Titles

Font: Calibri

Face: Bold

Size: 18

Sub Section Titles

Font: Calibri

Face: Bold

Size: 14

Sub subsection Titles

Font: Calibri

Face: Bold

Size: 13

Subsub subsection Titles

Font: Calibri

Face: Bold

Size: 12

Paragraph text Explanations

Font: Calibri

Face: Normal

Size: 12

# Description of Design Components

In MenuWhiz our basic components are the database, the server-side and API, and the client-side (Android application for customer and Web interface for restaurants).

## Database

One database will be used to store all the information regarding customers, restaurants, menu and order information. MySQL, a Relational Database Management System (RDBMS), will be used for this purpose.

## Server-side and API

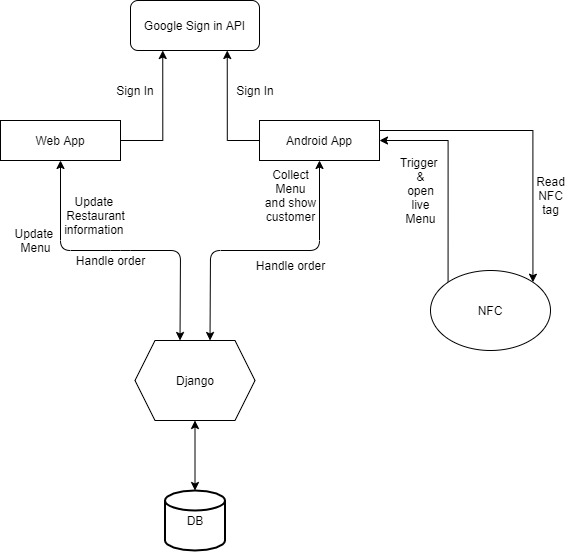
The backend or server-side of both of our android application and website will be the same. The backend will be designed using Django, a framework written using python. The backend will deal with activities such as user authentication, information retrieval from database, updating and writing information to database and processing information.

The android application and website interface will access the backend through an API. We will also use third party APIs such as Google Sign in, and NFC writer/reader to facilitate features of the product.

## Client- Side

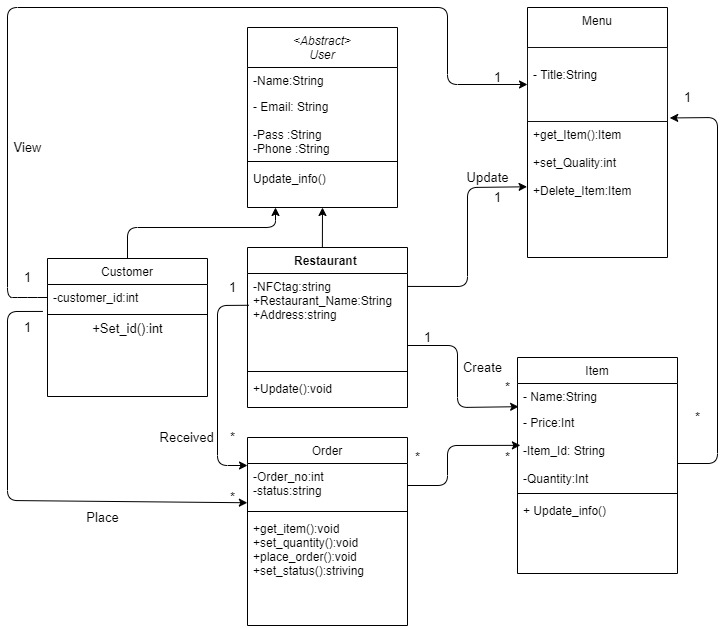
The client-side of our product has two interfaces, the android application and the web interface. The interfaces will be responsible for providing access to the features of our product, and executing the dynamic scripts which are variable from user to user.

# High-Level Component Design

****

# Class diagrams

## Detailed Class Diagrams



## Detailed Class Descriptions

1. **Class Name:** User  
   **Purpose:** Updates basic info about user  
   **Public attributes:** None  
   **Public Method:**   
    **Name:** Update\_info()  
    **Parameters:** None **Return Type:** void  
    **Purpose and description:** Every information of the class will be   
    updated by invoking the method.
2. **Class Name:**  Menu

**Purpose:**  This class will update every available item in the restaurant   
**Public attributes:**  None  
**Public Methods:**   
 **a.** **Name:** get\_item()  
 **Parameters:** None **Return Type:** Item  
 **Purpose and description:** Any items can be added to the menu .

**b.** **Name:** set\_Quantity() **Parameters:** None  
 **Return Type:**  Integer  
 **Purpose and description:**  quantity of the available items will be   
 updated.

**c.** **Name:** Delete\_items() **Parameters:** None  
 **Return Type:** Item  
 **Purpose and description:** Any items can be deleted from the   
 menu.

1. **Class Name:**  Item  
   **Purpose:**  Updates the items information about the menu.  
   **Public attributes:**  None  
   **Public Methods:**   
    **Name:** Update\_info() **Parameters:** None  
    **Return Type:** Void  
    **Purpose and description:** Updates every information of any items.
2. **Class Name:**  Restaurant  
   **Purpose:**  Identifiers of the restaurants will be updated by this class  
   **Public attributes:**    
    **a. Name:** Restaurant\_Name **Type:** String **Purpose:** It updates the name of the restaurant

**b. Name:** Address **Type:**  String **Purpose:** Address of the restaurant will be updated

**Public Methods:**   
  **Name:**  Update() **Parameters:** None  
 **Return Type:** Void  
 **Purpose and description:** Information of the restaurant will be   
 updated.

1. **Class Name:**  Order  
   **Purpose:**  Order is placed through the class  
   **Public attributes:**  None  
   **Public Methods:**   
    **a.** **Name:** get\_item() **Parameters:** None  
    **Return Type:** Void  
    **Purpose and description:** Items are updated when customers select items.

**b.** **Name:** set\_quantity() **Parameters:** None  
 **Return Type:** Void  
 **Purpose and description:** Number of every selected items are   
 updated.

**c.** **Name:** place\_order() **Parameters:** None  
 **Return Type:** Void  
 **Purpose and description:** Orders is paced.

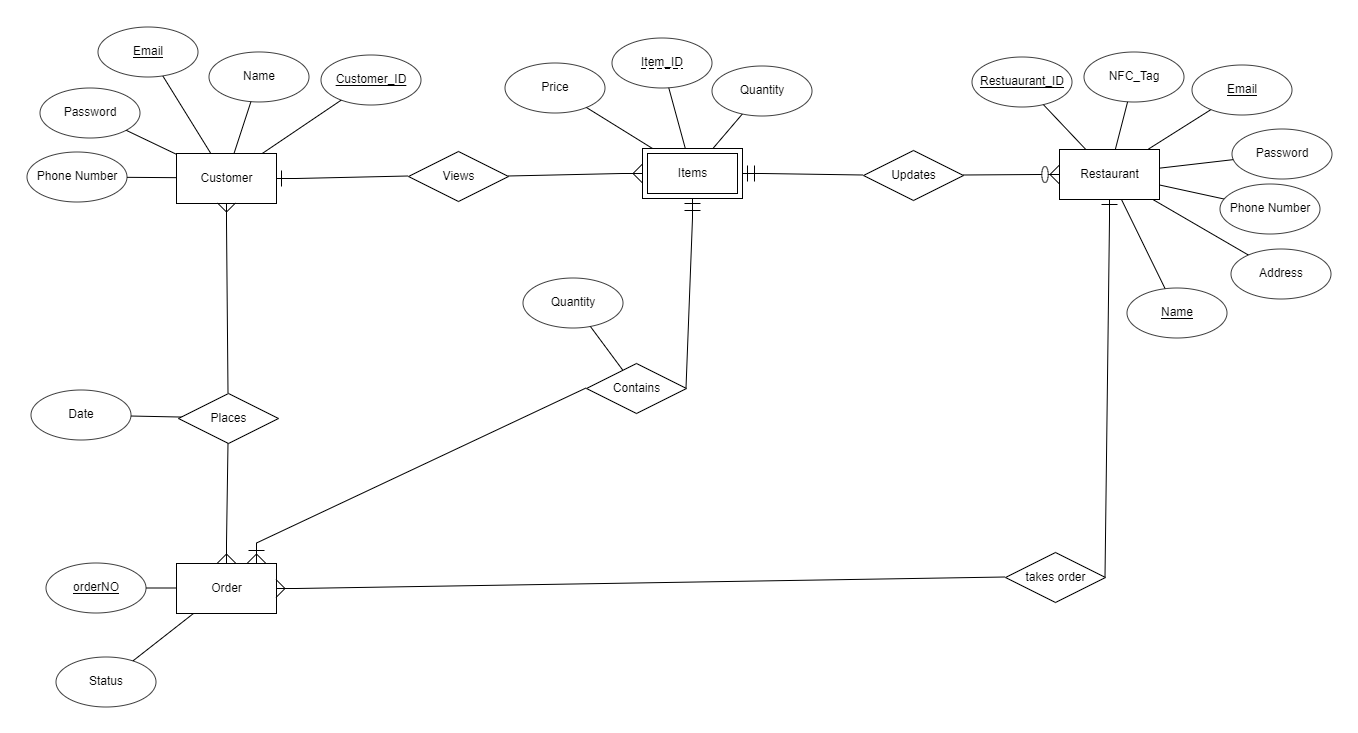
**d.** **Name:** set\_status() **Parameters:** None  
 **Return Type:** String  
 **Purpose and description:** The order status is updated.

1. **Class Name:**  Customer  
   **Purpose:**  People who want to view menu and order from this app.

**Public attributes:**  None  
**Public Methods:**   
  **Name:** Set\_id() **Parameters:** None  
 **Return Type:** Integer  
 **Purpose and description:** Automatic generates a id for customer.

# Database Design

## ER Diagram

****

## Tables

1. Name: Customers

Purpose: Storing the information of customers

Attributes:

1. Name: Customer\_ID

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: No

1. Name: Name

Type: String

Unique: No

Primary key: No

Foreign key: No

1. Name: Email

Type: String

Unique: Yes

Primary key: No

Foreign key: No

1. Name: Phone number

Type: Integer

Unique: No

Primary key: No

Foreign key: No

1. Name: Password

Type: String

Unique: No

Primary key: No

Foreign key: No

1. Name: Restaurant

Purpose: Storing the information of restaurants

Attributes:

1. Name: Restaurant\_ID

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: No

1. Name: Name

Type: String

Unique: Yes

Primary key: No

Foreign key: No

1. Name: Email

Type: String

Unique: Yes

Primary key: No

Foreign key: No

1. Name: Address

Type: String

Unique: No

Primary key: No

Foreign key: No

1. Name: Phone number

Type: Integer

Unique: No

Primary key: No

Foreign key: No

1. Name: Password

Type: String

Unique: No

Primary key: No

Foreign key: No

1. Name: NFC\_Tag

Type: String

Unique: Yes

Primary key: No

Foreign key: No

Dependent table: Items

1. Name: Item\_ID

Type: Integer

Partially Unique: Yes

Partial Primary key: Yes

Foreign key: No

1. Name: Name

Type: String

Unique: Yes

Primary key: No

Foreign key: No

1. Name: Email

Type: Quantity

Unique: No

Primary key: No

Foreign key: No

1. Name: Price

Type: Float

Unique: No

Primary key: No

Foreign key: No

1. Name: Orders

Purpose: Storing the information of current order status

Attributes:

1. Name: orderNO

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: No

1. Name: Status

Type: String

Unique: No

Primary key: No

Foreign key: No

1. Name: Placed\_Orders

Purpose: Storing the information of who placed which orders when.

Attributes:

1. Name: Customer\_ID

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: Yes

1. Name: orderNo

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: Yes

1. Name: Date

Type: Date

Unique: No

Primary key: No

Foreign key: No

1. Name: Order\_Contents

Purpose: Storing the information of which items each order contains

Attributes:

1. Name: orderNo

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: Yes

1. Name: Item\_ID

Type: Integer

Unique: Yes

Primary key: Yes

Foreign key: Yes

1. Name: Quantity

Type: Integer

Unique: No

Primary key: No

Foreign key: No

# Implementation Plan

|  |  |  |
| --- | --- | --- |
| **Target date** | **Landmark** | **Responsibility** |
| 1st September | Sign up, Sign in, Sign out interfaces of Android Application created | Md. Jubaer Khan |
| Sign up (without NFC configuration), Sign in, View Menu and Sign out interfaces created | Nafisa Tafshir |
| Backend of completed interfaces configured | Zahin Ahmed |
| 5th September | NFC Configuration interface created | Nafisa Tafshir |
|  | NFC Configuration enabled in backend | Zahin Ahmed and Md. Jubaer Khan |
| 15th September | Completed interfaces of both Android App and Website | Zahin Ahmed, Md. Jubaer Khan and Nafisa Tafshir |
| 30th September | Full backend configured and testing completed | Zahin Ahmed, Md. Jubaer Khan and Nafisa Tafshir |