**System Design – Digital *Maître d’Hôtel***

**Introduction**

This document will discuss the design process for the Digital Maître d’Hôtel project.

**Database Tables**

The SQL database used in the system will need to include firstly a table of restaurant tables. To avoid any confusion and errors with protected words, the Spanish word *Mesas* will be used as the name of the table. The *Mesas* table will contain every table in the restaurant and information on its booking status. A design will need to be produced to incorporate the status of each booking slot for each table. A TableNumber will be used as the primary key.

A simpler table, Customers, will also be included. This table will simply contain information on customers and their tables. The TableNumber from the *Mesas* table will be used as a foreign key, with an ID used as a primary key.

The Items table will contain all of the food and drink items on offer by the restaurant, with information on item price, stock levels and ingredients. An ID will be used as a primary key.

**Backend Methods**

The following methods will be implemented in the backend of the system in order to interface with the database tables described above, to fulfil functional requirements:

|  |  |  |
| --- | --- | --- |
| **Requirement** | **Method(arguments)** | **Summary** |
| Customer to check table availability / restaurant to check table availability | checkAvailability(int numberOfGuests) | Returns dates and times available for a given number of guests |
| checkAvailability(int numberOfGuests, String date) | Returns times available for a given number of guests on a given date |
| Customer to browse menu / menu to show items on a user interface, greying out meals which are not available | retrieveItems() | Returns items currently in stock |
| displayItems() | Displays relevant items in stock on the UI, displays items not in stock but substitutes image for greyed-out version |
| Customer to book a table | bookTable(int numberOfGuests, String date) | Books table in the restaurant and updates database, calls checkAvailability |
| Restaurant to check food inventory | checkStock() | Returns list of items along with stock levels |
| checkStock(String type) | Returns list of items of a certain category along with stock levels |
| checkStock(String item) | Returns stock level of a given item, calls checkItemExists |
| checkItemExists(String item) | Returns bool true if a given item exists in the database, else returns false |
| Track data on meal orders and table bookings for restaurant | logBooking(String booking) | Prints the information of a booking as a string to a CSV file |
| logOrder(String order) | Prints the information of an order as a string to a CSV file |
| Restaurant to alter menu/stock when necessary | deleteItem(String item) | Deletes a given item from the database |
| updatePrice(String item, float price) | Updates a given item’s price in the database |
| updateStock(String item, int change) | Updates a given item’s stock by a change amount |
| Output orders to a client printer | printOrder(String order) | Prints a given order to the console |
| Customer to pay for meals | payCheque(int ID) | Connects to PayPal to pay the bill associated with an ID |
|  |  |  |
|  |  |  |

**Database Table Headings**

The following table lists the headings for each table which will be present in the SQL database:

|  |  |  |
| --- | --- | --- |
| ***Mesas*** | **Customers** | **Items** |
| * TableNumber (PK)\*\* * Capacity | * ID (PK)\* * FirstName * LastName * TelephoneNumber | * ID (PK) * Name * Cost * Inventory * Allergens |

|  |  |
| --- | --- |
| **Bookings** | **Bills** |
| * ID (PK) * StartTime * FinishTime * CustomerID (FK)\* * MesasID (FK)\*\* | * ID (PK) * Amount * CustomerID (FK)\* |