**Restaurant Database**

Table of Contents

[Introduction 2](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330568)

[Purpose 2](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330569)

[Scope, Approach, and Methods 2](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330570)

[System Overview 2](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330571)

[Acronyms and Abbreviations 2](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330572)

[Points of Contact 3](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330573)

[Quality Assurance Process 3](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330574)

[Database Software Utilities and Developer Tools 3](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330575)

[Database Specifications 4](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330576)

[Assumptions 4](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330577)

[Physical Design 4](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330578)

[Physical Structure 4](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330579)

[Database Design and Functionalities 4](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330580)

[Design & Functional Support 4](file:///C:\Users\superadmin\OneDrive\Documents\Douglas\CSIS%202300%20Database%20I\Project\RESTAURANT-DATABASE\project_spec.docx#_Toc476330581)

# Introduction

The “Restaurant Database” project aims to manage the average restaurant activities and resources such as orders, items, quantities, prices, categories, suppliers, ~~employees~~, phone numbers, and addresses. The database design would provide flexibilities in terms of adding and disabling of a single item and categories as well. The database design would also provide an efficient data system both for reporting and retrieve data derived from discharges into the stored data.

## Purpose

This document provides high-level information on the specifications for the future DB implementation to support the functional requirements for “Restaurant” target database.

## Scope, Approach, and Methods

The Database Design for the “Restaurant” is composed of definitions for database objects derived by mapping entities to tables, attributes to columns, unique identifiers to unique keys and relationships to foreign keys.

Our project scope is limited by the list of functional requirements. During design, these definitions may be enhanced in order to support the requirements of the “Restaurant” application listed in the requirements specifications.

# System Overview

|  |  |  |
| --- | --- | --- |
| # | **System Overview** | **Details** |
| 1 | System name | Restaurant Management Database |
| 2 | System type | Client Server Application |
| 3 | Operational status | In development |
| 4 | Database Name | Restaurant Database |

## Acronyms and Abbreviations

|  |  |  |
| --- | --- | --- |
| # | **Acronym / Abbreviation** | **Meaning** |
| 1 | DB | Database |
| 2 | DBA | Database Administrator |
| 3 | RMD | Restaurant Management Database |
| 4 | TBD | To Be Determined |
| 5 | CR | Change Request |
| 6 | QA | Quality Assurance |
| 7 | DFD | Data Flow Diagram |

## Points of Contact

|  |  |  |  |
| --- | --- | --- | --- |
| # | Role | Name | Email |
| 1 | Project Manager (Instructor) |  |  |
| 2 | Project Coordinator (Student) |  |  |
| 3 | Project Member (Student) |  |  |
| 4 | Project Member (Student) |  |  |
| 5 | Project Member (Student) |  |  |

## Quality Assurance Process

The group members will provide a Quality Assurance tests to ensure DB integrity, functional dependency, and requirements coverage. The QA will involve manual format and content verification by the group members, before submitting the project. QA process will also verify the content for questionable data values, meaning possible outliers or other values, which could be potentially incorrect.

## Database Software Utilities and Developer Tools

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| # | Vendor | Product | Version | Comments |
| 1 | Romain Bourdon | WAMP Server | 3.0.6 (64 bit) | Windows web development environment |
| 2 | Apache Software Foundation | Apache | 2.4.23 | Web server |
| 3 | PHP open source Community | PHP | 5.6.25 | Script Engine |
| 4 | Oracle | MySQL | 5.7.14 | DB engine |
| 5 | Oracle | MySQL Workbench | 6.3.8 (64 bit) | DB visualization tool |
| 6 | CodeLobster | CodeLobster PHP Edition | 5.11.4 | Scripting tool (IDE) |
| 7 | GitHub Inc. | GitHub.com | n/a | Online file repository (<https://github.com/ikostan/RESTAURANT-DATABASE>) |
| 8 | Git open source community | Git GUI client | 0.20.GITGUI | GUI client for distributed version control system |
| 9 | Git open source community | Git client | 2.9.2.windows.1 | Distributed version control system |

# Database Specifications

## Assumptions

* Addresses history will not be preserved.
* One supplier may provide multiple brands.
* Each supplier has one and only one address
* Each supplier may have multiple telephone numbers.
* Suppliers may be from different cities.
* There is a track of prices history per item.
* No updates for customers’ feedbacks.
* One order may have no more than one feedback.
* Item type may have multiple subtypes.
* Each item subtype may have multiple related items.
* Each item must have weight unit.
* In order to make inventory/quantities management more efficient min and max quantities will be preserved in inside the “inventory” table.
* Each item may be related to only one brand.
* No tracks of customers’ data.

## Physical Design

TBD

## Physical Structure

TBD

# Database Design and Functionalities

## Design & Functional Support

The database will be designed in order to meet the below listed functional requirements.

Priorities:

* **High**
* **Medium**
* **Low**

Note: team members are committed to the requirement implementation of high and medium priorities only.

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 1 | **High** | Easy information retrieval | 1.1 Easy retrieval of information from Restaurant Database about ~~clients~~, items, prices, inventory, ~~services, employees~~, addresses ~~(including history~~), orders ~~(including history) and statuses.~~ |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 2 | **High** | Nomenclature consistency | 2.1 Use ~~uppercase~~ lowercase for the table name.  2.2 Use lowercase for key name. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 3 | **High** | Provide orders history | 3.1 Easy retrieval of information on orders from DB (even though~~: a. client address~~  ~~has changed;~~ b. item price has changed). |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 4 | **Medium** | Provide addresses information | ~~4.1 Easy retrieval of information on a billing client address from DB.~~  ~~4.2 Easy retrieval of information on a shipping client address from DB.~~  ~~4.3 Easy retrieval of information on employee address from DB.~~  4.4 Easy retrieval of information on a supplier address from DB. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 5 | **Medium** | Provide telephone information | ~~5.1 Easy retrieval of information on a client’s home telephone number from DB.~~  ~~5.2 Easy retrieval of information on a client’s work telephone number from DB.~~  ~~5.3 Easy retrieval of information on a client’s cell phone number from DB.~~  ~~5.4 Easy retrieval of information on employee’s home telephone number from DB.~~  ~~5.5 Easy retrieval of information on employee’s cellphone number from DB.~~  5.6 Easy retrieval of information on supplier phone number from DB. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 6 | **High** | Able to aggregate data in order to detect item price | 6.1 DB must support the easy detection of the item price according to specified parameters. ~~For example:~~  ~~Delivery price = Delivery type + Delivery weight category~~ |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 7 | **High** | Able to aggregate data in order to calculate total price for the order | 7.1 DB must support the easy calculation of the total price by aggregating all purchased items prices. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 8 | **High** | Possibility to modify the content of the database | 8.1 It will be possible to modify the data in the database using the Edit option available in the Workbench and/or PHP MyAdmin. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 9 | **Low** | Possibility to modify the structure of the database | 9.1 Modification of database structure will be analyzed for costs and impact on the DB. If such modification approved, it will be handled as a Change Request (CR). |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 10 | **High** | Project members to be able to insert data (partial data, or complete annual set) directly  into the database | 10.1 Data insert (partial and/or a complete dataset) to the database is possible via Workbench and/or PHP MyAdmin application.  The data insert operation will be performed by a sequence of INSERT, UPDATE or DELETE operations to the DB. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 11 | **High** | Able to modify previously entered data in the database | 11.1 The users shall be able to edit the previously entered data using the edit option in the web application. This would translate to an update query on the database. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 12 | **Medium** | To be able to include the calculation method for estimated data (e.g. min or max quantities)  when submitting them into the database | 12.1 The client shall be able to specify the minimum and max available quantities according to provided data from the DB.  12.2 The employee shall be able to specify the minimum and max available quantities that must be available in the inventory according to provided data from the DB. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 13 | **Low** | Able to administer user rights (granting/denying actions to other users) | 13.1 The Data Manager has full privileges on the Restaurant database, and able to administer user rights for all the other users of the system.  13.2 Managers should also be able to delegate privileges within their own limited permissions. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 14 | **Medium** | Retrieve a report about which data has been modified, and indicating by whom | 14.1 The restaurant DB will provide the possibility to retrieve the information regarding which data has been modified (date/time).  14.4 The information regarding this can be stored in the system tables provided by SQL Server. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 15 | **High** | New Restaurant Database to easily generate data reports (e.g. tables, figures) on  total quantities for individual items | 15.1 Easy evaluates total numbers for an item according to different parameters obtained from the tables in DB. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 16 | **High** | Easy to create new subcategories for existing items. | 16.1 Creation of the new category or subcategory should not cause to the DB schema alteration. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 17 | **High** | Easy to disable any existing item subtype or category | 17.1 Implement *isActive* key in order to indicate whether the item is enabled or not (BOOLEAN). |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 18 | **High** | Support metric measurement system only | 18.1 All measurements will be according to the metric system: gram, milliliters.  18.2 Min value for gram is 1gr, for milliliter is 1 ml. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 19 | **High** | Distinguish between beverages vs food items in terms of weight and volume | 19.1 All food items measured in grams.  19.2 All beverages volumes measured in milliliters.  19.3 All the above not related to the delivery weight. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| ~~20~~ | **~~High~~** | ~~Easy aggregation of the delivery weight~~ | ~~20.1 Implement weight units in order to provide easy calculation method for the delivery weight.~~  ~~For example:~~  ~~each 100gr = 1 weight unit~~  ~~each 100ml = 1.2 weight unit~~ |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 21 | **Medium** | Customers feedback | 21.1 ~~Registered and unregistered~~ customers must be able to leave their feedback. |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| ~~22~~ | **~~High~~** | ~~Addresses History~~ | ~~22.1 DB must allow managing history for addresses, prices, and products.~~ |

|  |  |  |  |
| --- | --- | --- | --- |
| Requirement | | | |
| ID | Priority | Name | Description |
| 23 | **High** | Prices History | 23.1 DB must allow managing history for prices. |