

<Project  
Name>

Database Design Document (DDD)

Version <version number>

Prepared by: <Author Name>

Revision History

| Date | Version | Description | Author |
|------|---------|-------------|--------|
|      |         |             |        |
|      |         |             |        |

---



# Table of Contents

- 1 INTRODUCTION ..... 1**
  - 1.1 DOCUMENT OBJECTIVES ..... 1
  - 1.2 INTENDED AUDIENCES ..... 1
  - 1.3 REFERENCES ..... 2
- 2 DETAILED DATABASE DESIGN..... 2**
  - 2.1 DYNAMODB DESIGN (NoSQL DATABASE) ..... 2
    - 2.1.1 Object Diagram ..... 3
    - 2.1.2 Data dictionary..... 2
      - 2.1.2.1 Data dictionary for Element: User .....2
      - 2.1.2.2 Data dictionary for Element: Password .....3
      - 2.1.2.3 Data dictionary for Element: File .....3
      - 2.1.2.4 Data dictionary for Element: Area .....5
  - 2.2 SQLITE DATABASE DESIGN (RELATIONAL DATABASE) ..... 4
    - 2.2.1 Conceptual diagram ..... 4
    - 2.2.2 Description..... 4
    - 2.2.3 Purpose of Tables..... 5
      - 2.2.3.1 Purpose of Area Table .....5
      - 2.2.3.2 Purpose of File Table .....7
      - 2.2.3.3 Purpose of User Table .....7
    - 2.2.4 Relations ..... 5
- 3 REFERENCES ..... 6**
- 4 APPENDIX 1 – XML SCHEMA..... 8**

# 1 Introduction

The section introduces Aqua Flow Database.

## 1.1 Document Objectives

- Introduction The purpose of this document is to describe the design of a database for managing administrators and stock data. The system is built to store and manage the credentials of system administrators, as well as track inventory data, including product names, prices, and stock quantities. The database design is intended to provide easy access and manipulation of data through a Database Management System (DBMS), enabling users and software programs to efficiently interact with the stored data. This design outlines the database structure, relationships, and the method of data access, serving as the basis for implementing the system's database. It also provides essential details for software development and future system support.

## 1.2 Intended Audiences

- Designers

Designers will refer to this document to ensure that their design for the database schema and interfaces meets the requirements laid out. They will work on the presentation and integration of database elements into the larger system.

- Programmers

Programmers are responsible for implementing the database design into the application. They must write the necessary code to interface with the database and ensure that data is managed efficiently. Their implementation must adhere to the specifications outlined in this document.

- Testers

Testers are responsible for creating test cases and performing tests to validate that the database functions as expected. They must ensure that all database operations (e.g., insertions, queries, updates, deletions) work correctly, and data is handled according to the requirements set forth in this document.

Designers will refer to this document to ensure that their design for the database schema and interfaces meets the requirements laid out. They will work on the presentation and integration of database elements into the larger system.

## 1.3 References

- Notion – The all-in-one workspace for your notes, tasks, wikis, and databases. (n.d.). Notion.<https://believed-bongo-319.notion.site/CTINFMGL-Project-Specifications19296450aad180aea9ebf27987415f4c>

## 2 Detailed Database Design

This section outlines the Detailed Database Design for the Admin and Stock Management System at varying levels of abstraction, including Conceptual, Internal, Logical, and Physical levels. Each of these levels offers a different perspective on the database design, from high-level requirements to implementation details.

### 2.1.1 Data dictionary

#### 2.1.1.1 Data dictionary for Element: <admin>

| Name             | Data Type    | Constrain | Description                 |
|------------------|--------------|-----------|-----------------------------|
| ID (primary key) | INT          | NOT NULL  | ID of the user              |
| Username         | Varchar (30) | NOT NULL  | Name of the user            |
| Password         | Varchar (30) | NOT NULL  | The Password of the user    |
| City             | Varchar (30) | NOT NULL  | The City of the user        |
| Contact          | Varchar (11) | NOT NULL  | Contact Number of the user. |

|                 |           |                                   |  |
|-----------------|-----------|-----------------------------------|--|
| Account Created | TIMESTAMP | NOT NULL,<br>current_timestamp(), | ,Admin can determine<br>when the account was<br>created. |
|-----------------|-----------|-----------------------------------|--|

#### 2.1.1.2 Data dictionary for Element: <adminUsers>

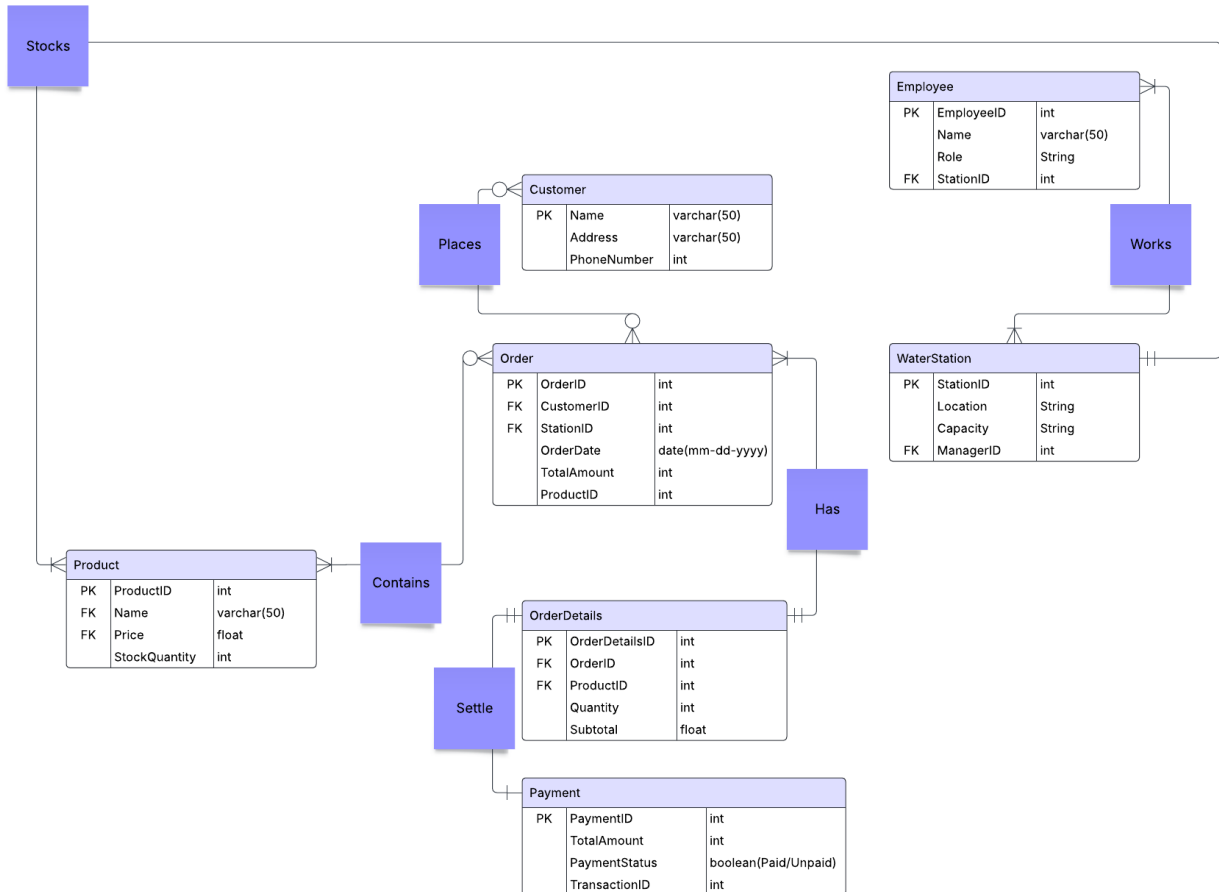
| Name                  | Data Type    | Constrain | Description                    |
|-----------------------|--------------|-----------|--------------------------------|
| Username(primary key) | Varchar (30) | NOT NULL  | ID to identify the<br>password |
| Password              | Varchar (30) | NOT NULL  | The password of the<br>admin   |

#### 2.1.1.3 Data dictionary for Element: <Stock>

| Name                   | Data Type    | Constrain        | Description                          |
|------------------------|--------------|------------------|--------------------------------------|
| Stock ID (primary key) | INT          | NOT NULL         |                                      |
| Product                | Varchar (30) | NOT NULL, UNIQUE | Name of the products.                |
| Price                  | INT          | NOT NULL         | Price of the products.               |
| Stockquantity          | INT          | NOT NULL         | Quantity of the available<br>stocks. |

## 2.2 MySQL database design (Relational database)

### 2.2.1 Conceptual diagram



### 2.2.2 Description

This diagram displays the conceptual model of the SQLite database. This database will be created after the it has imported the user's data. The user session will have the user's details who is currently logged in. The user will have a password and an admin area where he can make changes to his account such as changing password (optional up to user to set it up). Each user will have zero or more files. The local database will only hold the Area and file information. All the data other than the primary and foreign keys will be stored after encryption using user's password. The database contents will be decrypted when user makes request.

## 2.2.3 Purpose of Tables

### 2.2.3.1 Purpose of <user> Table

The admin table's primary purpose is to store essential data about system administrators, facilitating their secure access, management, and interaction with the Admin and Stock Management System. It plays a critical role in user authentication, contact management, activity tracking, and maintaining the integrity of the system by ensuring uniqueness and data consistency.

### 2.2.3.2 Purpose of <adminUsers> Table

The adminUsers table is likely a table that manages administrator-specific data with more detailed control over permissions, roles, and potentially more granular user attributes. It allows for role-based access control, tracking of admin user activity, managing multiple admins with different responsibilities, and ensuring secure authentication. This table complements the admin table by extending its functionality, particularly in systems with varying levels of admin user permissions and roles.

### 2.2.3.3 Purpose of <order> Table

The stock table's primary purpose is to manage and store the details of products within the inventory system. It tracks product information such as name, price, and stock quantity, enabling efficient inventory management, sales tracking, and business operations. By maintaining accurate and up-to-date product data, the stock table ensures smooth operations of the Admin and Stock Management System and supports important functions such as product catalog management, price adjustments, inventory control, and querying product availability.

## 2.2.4 Relations

| From Table | To Table  | Relation                                 |
|------------|-----------|--|
| User       | Stock     | A user can order and see stock.          |
| AdminUser  | User      | Admin can edit user info.                |
| AdminUser  | Stock     | Admin can add stock or adjust the price. |
| Stock      | User      | It is visible for users.                 |
| Stock      | AdminUser | It is visible for users.                 |



### 3 References

Notion – The all-in-one workspace for your notes, tasks, wikis, and databases. (n.d.). Notion.  
<https://believed-bongo-319.notion.site/CTINFMGL-Project-Specifications19296450aad180aea9ebf27987415f4c>