**Solent University**

Advanced Database Systems

**Module Leader**:

Dr. Muhammad Akram

Project with Report

**Student ID**: [Your ID Number]

**Student Name**: [Your Full Name]

**Submission Date**: 4th February 2025

**Word Count**: ???

Table of Contents

**Introduction**

This report describes the construction of a database system for a restaurant. The main aim of the project was to construct a highly efficient and user-friendly database-based application, which can perform all day-to-day activities in the restaurant as smoothly as possible e.g. making table reservations, tracking sales, managing inventory, handling orders, etc.

Managing data in restaurants is still one of the biggest challenges. Most medium and small-scale restaurants use half-manual or not-so-advanced computer applications to manage their data, and thus errors and duplication are not uncommon. Something like this should not happen because it may cause loss of important information about customers i.e. missing out on dollars from opportunities.

To cover up for this gap at an affordable cost we came up with an application, which stores all our needed information and uses easily understandable graphical interfaces to access the results without having strong technical knowledge how the queries should look like or when new search parameter needs to be added

The objectives of this report are to describe the client’s requirements, design an efficient relational database, implement a proof-of-concept system and evaluate its performance for the solution of the problems defined above. SQLite is used as a lightweight but powerful database management system that will be encapsulated by Tkinter, Python’s standard GUI (Graphical User Interface) package. XML will be used for storage and manipulation of arbitrary structured data.

The database is designed with third normal form (3NF) to make data consistency and prevent data for redundancy, relational features like Triggers, Views are also constructed to make system works effectively and efficiently as business wants, as well as usability. And finally the tested results which is performed by users of system was shown to include the context of the functional requirements that have been selected.

In conclusion, our database system for restaurant management shows how using a well-designed database could ease daily operations and save expenses by minimizing errors.

**Part 1: Client Requirements and Database Design**

This section outlines the essential client requirements for the database system, followed by the process of designing and creating the database using SQLite. The goal is to create a system that meets the needs of the restaurant, ensuring it can handle customer data, inventory, employee information, and order details effectively.

**1.1. Business Case for the Chosen Restaurant**

The restaurant is a small-sized bistro specializing in Italian cuisine, with both dine-in and delivery services. It employs approximately 10 staff members and serves around 50 customers daily. Given the nature of the business, efficient management of customer reservations, inventory tracking, and employee scheduling are critical for maintaining smooth operations.

**1.2. Seven Business Requirements**

1. **Customer Management** – The system should allow for storing and retrieving customer information, including personal details like address and phone number.
2. **Inventory Tracking** – The system should track inventory levels for food ingredients and supplies.
3. **Employee Records** – The database should maintain employee information, such as contact details, job title, and payroll information.
4. **Reservation Management** – The system must handle customer reservations, ensuring no double-booking occurs.
5. **Accounting and Financial Records** – The database should track sales, revenue, and expenses, including generating financial reports.
6. **Menu Management** – The system should allow for managing and updating menu items, including prices and availability.
7. **Order Management** – The system should manage customer orders from intake to fulfillment, ensuring timely delivery and accurate billing.

**2. Flat File Creation with Collected Data (Excel)**

For initial data collection, a flat file was created using Microsoft Excel. Below is a screenshot of the flat file created for the 'Customers' table, which includes five records. Similar flat files were created for other tables in the database, including 'Suppliers', 'Reservations’ and 'Staff'. Each file contains sample data to be used in the SQLite database development.

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Customers**

Obraz zawierający tekst, Czcionka, numer, linia

Opis wygenerowany automatycznie

**Suppliers**

Obraz zawierający tekst, zrzut ekranu, Czcionka, numer

Opis wygenerowany automatycznie

**Reservations**

**Staff**

**3. Relational Schema**  **and Database Design**

In this section, we present the **relational schema** for the restaurant management database system, represented through an **Entity-Relationship Diagram (ERD)**. The schema has been carefully designed to fulfill the business requirements, ensure data integrity, and optimize efficiency. By adhering to the principles of **Third Normal Form (3NF)**, the design minimizes redundancy and ensures consistency within the database.

**Obraz zawierający tekst, zrzut ekranu, numer, Czcionka

Opis wygenerowany automatycznie**The ERD below illustrates the key entities, their attributes, and relationships, which collectively address the functional requirements of the system.

**Justification for 3NF Compliance**

To ensure that the database is normalized to Third Normal Form (3NF), the following steps and principles were followed:

1. **First Normal Form (1NF):**
   * Each table contains only atomic (indivisible) values.
   * There are no repeating groups or arrays in any table.
   * Example: In the Orders table, each order is uniquely identified by OrderID, and all attributes are single-valued.
2. **Second Normal Form (2NF):**
   * All non-key attributes are fully dependent on the entire primary key.
   * Partial dependencies were eliminated by breaking down tables.
   * Example: In the OrderDetails table, each Subtotal is dependent on both OrderID and MenuItemID. Attributes that depend on only part of the composite key were moved to appropriate tables.
3. **Third Normal Form (3NF):**
   * There are no transitive dependencies (i.e., non-key attributes are dependent only on the primary key, not on other non-key attributes).
   * Example: In the Customers table, ContactInfo is dependent only on CustomerID, and no unrelated data is stored in this table.

**4. Develop SQLite Server database system**