Auto Repair Shop Management

Overview

The business in focus is auto repair shop that manages services and appointments and stores customer's data. The shop maintains its operational data through access database and uses an Excel-based front-end application to interact with the data. The Auto Repair Shop is designed to efficiently manage customers, services, and appointments. It integrates a Microsoft Access database with an Excel-based front-end interface, utilizing VBA to streamline operations such as fetching service details, loading appointments, and submitting new appointments.

Database Structure

The Access database consists of three main tables:

1. **Customers Table**- This table stores customer information such as customer id, customer name, customer phone number, customer email, registration number of vehicle and model of the vehicle.
2. **Services Table**- This table stores services information such as service id, service name, price, category, warranty details and description of service done.
3. **Appointments Table**- This table stores the appointment which are there for auto repairing which consist of columns like appointment id, customer id, service id, appointment date and service status of vehicle.

Key Queries:

- **Service Details by ID**: Fetches the details of a service based on ServiceID.
- **Upcoming Appointments**: Displays all future appointments sorted by date.

Excel-Based Front-End

The Excel front-end serves as the primary user interface for managing the repair shop's operations.

Appointments Tab: Displays all appointments and colour codes them through VBA on the basis of their service status like red for cancelled service, green for completed service and yellow for their scheduled service. Button is added to run the macro to fetch the records and color code them.

Revenue pivot table- It displays total revenue in respect to service status using pivot table function.

Appointment Form- Appointment form tab is used to insert a new records into database through VBA into appointments table. For columns in excel we have validated the input through functions and added button assigned to macro to add record in database.

Service Details- Service detail tab is used to fetch required record by searching through the service ids and also consists of validation of inputs.

VBA Subroutines

LoadAppointments

The LoadAppointments subroutine pulls appointment records from database and organizes it in sheet. It starts by setting up the database connection and defining a query to fetch details appointment IDs, customer info, service details, and total revenue. Before pulling records, it clears the worksheet for a clean slate and adds column headers. As the data is added row by row, it applies color coding based on the appointment status (e.g., yellow for "Scheduled," green for "Completed"). Once all the data is loaded, the database connection is closed, and a success message confirms the task is complete.

SubmitNewAppointment

The SubmitNewAppointment subroutine allows users to add a new record to the appointments table from an Excel form. It starts by identifying the worksheet and database path and then reads the input values for customer ID, service ID, appointment date, and status from specific cells in the sheet. The inputs are validated to ensure they are correct, such as checking that IDs are numeric, the date is valid, and the status is either "Scheduled," "Completed," or "Cancelled." If any input is invalid, the subroutine shows an error message and exits. Once the inputs are confirmed valid, the subroutine establishes a connection to the Access database and constructs a SQL query to insert the new appointment. The query is executed, and after adding the appointment, the database connection is closed. The subroutine notifies the user of the successful addition with a message and clears the form fields in Excel.

GetServiceDetailsByID

The GetServiceDetailsByID subroutine retrieves details of a specific service from database based on a Service ID entered by the user. It begins by prompting the user to input a Service ID. The input is validated to ensure it's not empty. Afterwards a SQL query is constructed to search for the service in the database. A connection to the Access database is established, and the query is executed. If no matching service id is found, a message is displayed to inform the user, and the process ends. If a service id is found, the details (Service ID, Description of Service, and Price) are written to an Excel worksheet. The Price column is formatted to display as currency with a pound symbol for better readability.

Conclusion

The application could be scaled up for a real business by integrating advanced features and enhancing its current functionality. For example, it can be connected to a cloud database like SQL Server to enable real-time multi-user access. Automated SMS notifications for appointment confirmations, reminders, and status updates could improve customer experience. A mobile app interface could be developed to allow customers to book appointments, view service history, and make payments online.