HAR MS

Home Appliance Repair Management System

Software Requirements Specification (SRS)

Team 5<Herm>

1. Atsloom Alamri
2. Muna Al maashani
3. Doaa al dahab
4. Asma Al alawi

Home Appliance Repair Management System

Purpose

The purpose of this system is to digitize the operations of a medium-sized home appliance maintenance company.
The system will help you:

- Receive and organize customer requests in one place.
- Assign the right technician to each job.
- Track the repair process step by step until completion.
- Generate invoices quickly and accurately.

The system will be used internally through a simple console application, with a basic web prototype to show what the future user interface will look like.

Definitions:

Maintenance Request: The request submitted by the customer to repair a household appliance.
Technician: The person responsible for carrying out maintenance work**.**
Invoice: The financial document that shows the costs of maintenance and spare parts.
Request Status: The stage that the maintenance request goes through (Pending, Assigned, In Progress, Completed, Canceled).

Risk and Assumptions Analysis:

Risk of losing order data in case of system failure.

Delay in technicians' response due to order backlog.

Assumption that customers will provide accurate and complete information.

System Quality Standards:

Performance: System response in less than two seconds per request.

Security: Protecting customer and technician data from unauthorized access.

Ease of Use: A simple interface that supports smooth navigation between screens.

Examples of input and output data:

Input: A maintenance request form containing the customer's name, device type, and problem description.

Output: A report showing the number of requests by status and their distribution by technicians.

Security and Privacy Requirements:

Encrypt customer and technician data in the database.

Restrict data access according to user permissions (even if the system login is not yet implemented, for future readiness).

Commit to protecting customer data in accordance with local and international standards.

- **Scope**

The system includes:

- Customer and technician management
- Repair order creation and tracking
- Automatic invoice generation
- Operational reporting
- Usable console application
- Static HTML/CSS/Bootstrap web UI prototype

- **2. Functional Requirements**
-
    - **Customer Management**

- Add new customer
- Edit customer details
- Delete customer (only if no orders exist)
- View all customers

- **Technician Management**

- Add technician

- Edit technician details
- Delete technician (only if no orders assigned)
- View all technicians

○ **Repair Order Management**

- Create new repair order
- Assign technician to order
- Change order status
- Complete order (enter costs)
- Cancel order
- View all orders

○ **Invoice Management**

- Auto-generate invoice upon order completion
- Calculate total (parts + service)
- View all invoices

○ **Reporting**

- Count of orders by status
- Orders grouped by technician
- Orders filtered by date

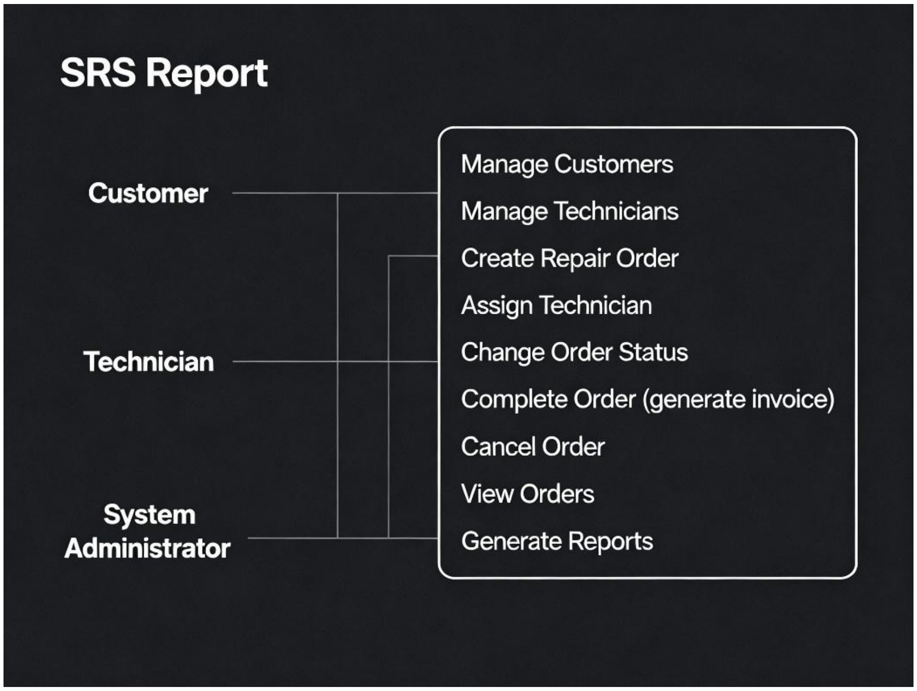
○ **User Interface**

▪ Console Application

- Text-based menus
- Input validation
- Module navigation

- **Web UI Prototype**
- Dashboard
- Customer, technician, order, invoice, and report pages
- Built with HTML5, CSS3, Bootstrap
- ✓ **Use case diagram**

The diagram shows the interaction of three main actors with the system through core functions such as managing customers and technicians, maintenance requests, assigning technicians, completing requests and invoices, and displaying reports. It illustrates the connections between each actor and the functions they use, providing a simple visualization of how the system works.

- 

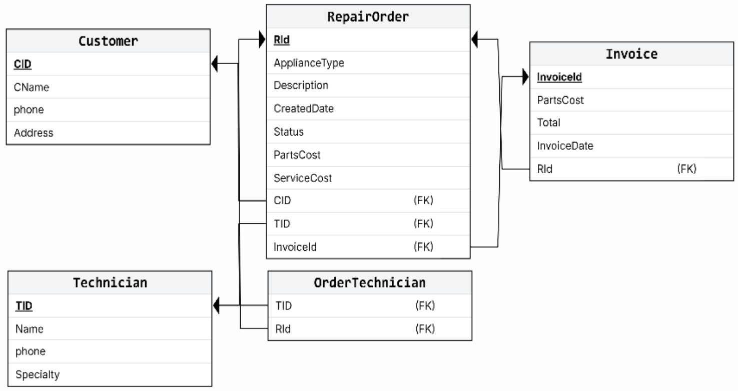
✓ ERD Diagram (Entity-Relationship Diagram)

The schema represents the main entities such as customers, technicians, maintenance requests, and invoices, where each entity stores a set of attributes that describe its data, such as the customer name and phone number in the customers entity.

The relationships between these entities show how they are connected to each other, such as the relationship of maintenance requests with customers and technicians, where each maintenance request is linked to a specific customer and technician.

Primary keys define a unique identifier for each entity, and foreign keys connect different entities.

The schema illustrates the organization of data in the database in a logical way that helps store and retrieve information efficiently and accurately to meet system requirements.



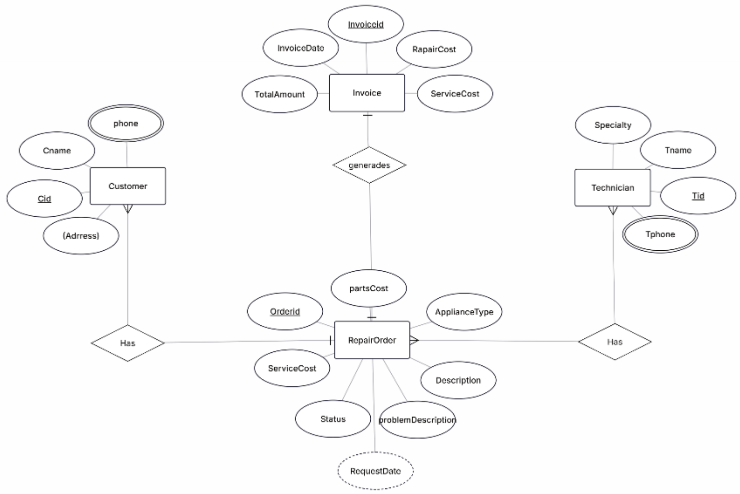
Customer
CID
CName
phone
Address

RepairOrder	
RId	
ApplianceType	
Description	
CreatedDate	
Status	
PartsCost	
ServiceCost	
CID	(FK)
TID	(FK)
InvoiceId	(FK)

Invoice	
InvoiceId	
PartsCost	
Total	
InvoiceDate	
RId	(FK)

Technician
TID
Name
phone
Specialty

OrderTechnician	
TID	(FK)
RId	(FK)



1. Non-Functional Requirements

- EF Code-First with Migrations
- SQL Server LocalDB
- Clean and modular code structure
- User-friendly console menus
- Responsive and professional Bootstrap design
- Well-documented project structure

System Constraints

- No login or user roles required
- Web UI is static and not connected to backend
- Deletion of customers/technicians restricted if linked to orders

Operating Environment

- OS: Windows
- Language: C#
- Database: SQL Server LocalDB
- UI Tools: HTML, CSS, Bootstrap

Console Menu Structure

Main Menu

- Manage Customers
- Manage Technicians
- Manage Repair Orders
- Reports
- Exit

Repair Orders Sub-Menu

- Add new order
- Assign technician
- Change status
- Complete order (generate invoice)
- Cancel order
- List all orders

Reports Sub-Menu

- Orders count by status
- Orders per technician
- Orders by date