

# Wireless Handset (Phone) Repair Service Factory Database

## Table of Contents

<b>Database Topic .....</b>	<b>2</b>
<b>Mission Statement .....</b>	<b>2</b>
<b>Mission Objectives .....</b>	<b>2</b>
<b>Business Requirements .....</b>	<b>3</b>
Transaction History .....	3
Process Control .....	3
Quality Control.....	3
Customer Feedback .....	3
Device Management .....	3
Inventory Management .....	3
<b>Business Problems .....</b>	<b>4</b>
Customer Management:.....	4
Order Tracking:.....	4
Inventory Management: .....	4
Payment Processing: .....	4
Employee Workflow:.....	5
Customer Feedback: .....	5
Quality Control: .....	5
<b>Design Decisions .....</b>	<b>7</b>
<b>Entity Relationships .....</b>	<b>14</b>
<b>Business Flow and Business Rules .....</b>	<b>16</b>

# Database Topic

Wireless Handset (Phone) Repair Service Factory Database

## Mission Statement

To streamline the wireless handset repair and refurbishment process by ensuring comprehensive end-to-end traceability, enabling efficient completion of each stage, and minimizing turnaround time for maximum operational efficiency.

## Mission Objectives

- Ensure each handset can be traced through every step of the repair and refurbishment process, from receiving to shipping back to customer
- Reduce the turnaround time for each handset by optimizing repair, quality check, and shipping processes
- Track and report on quality control to reduce defect rates and ensuring each refurbished handset meets set quality standards before being shipped.
- Manage parts and inventory in real-time to prevent delays caused by shortages, ensuring that repairs can continue without interruption
- Record and maintain a detailed transaction history for each handset throughout the factory process, including timestamps and technician IDs at every stage, to provide full accountability and transparency for repair and refurbishment activities.

# Business Requirements

## Transaction History

Each step in the process (ex: repair, heavy repair, scrap, quality check, shipping) must be recorded with timestamps and employee ID.

## Process Control

## Quality Control

The system must ensure traceability for all quality checks, logging the outcomes of each inspection.

## Customer Feedback

The system must be able to capture and store feedback received from customers via surveys regarding their satisfaction with repair services and replacement device quality.

## Device Management

The system must track all relevant device information, including manufacturer details, serial number, current owner, and repair history.

## Inventory Management

The system must maintain a real-time record of parts inventory, including part availability, supplier information, and re-order thresholds and capacity.

# Business Problems

## Customer Management:

The factory struggles to efficiently manage customer data and track orders. Without a proper system, customer information is scattered, making it difficult to retrieve order histories and update customers on the status of their repairs. This leads to delays and errors in processing orders.

### ***Solution:***

The database centralizes customer information and links it to orders, allowing employees to easily access transaction histories and provide accurate updates on ongoing repairs. This improves order processing efficiency and ensures better communication with customers.

## Order Tracking:

Devices are often misplaced, and there is no reliable system to track their location or repair status. This lack of visibility results in poor communication with customers, who are often left unsure about the progress of their repairs.

### ***Solution:***

The database tracks each device as it moves through the repair process, from receiving to shipping. Real-time updates on device status enable employees to provide customers with accurate information, reducing miscommunication and improving overall service.

## Inventory Management:

The warehouse faces frequent stock shortages because parts and tools are not tracked effectively. This results in delays when repairs cannot be completed due to missing parts or tools, causing frustration for both employees and customers.

### ***Solution:***

The database manages the inventory of parts and tools, keeping track of stock levels and ensuring timely reordering of supplies. This prevents shortages and ensures that necessary parts and tools are available when repairs are scheduled, reducing downtime.

## Payment Processing:

Payment tracking is inconsistent, leading to confusion over which orders are paid, unpaid, or refunded. Without a clear system in place, the factory experiences delays in processing payments and potential disputes with customers over billing.

***Solution:***

The database links each payment to its corresponding order, providing a clear record of payment status. This ensures accurate financial tracking, reduces disputes, and speeds up payment processing, improving overall financial management.

## Employee Workflow:

Employees face bottlenecks and inefficiencies due to a lack of structured task assignments. Some employees may be overloaded while others are underutilized, leading to delays in repairs and inconsistent productivity.

***Solution:***

The database assigns tasks to employees based on their roles and workload, ensuring an even distribution of work. This helps prevent bottlenecks, improves employee productivity, and ensures that repairs are completed more efficiently.

## Customer Feedback:

There is no formal process for collecting and analyzing customer feedback, making it difficult for the warehouse to identify areas where service can be improved. As a result, the factory misses opportunities to enhance customer satisfaction.

***Solution:***

The database includes a customer feedback system through surveys, allowing customers to rate their experience and provide comments after each repair. This feedback is stored and analyzed, enabling the warehouse to improve its services based on real customer input.

## Quality Control:

Devices are sometimes returned to customers without proper inspection, which leads to defective repairs and an increased number of returns. Without a structured quality control process, the warehouse cannot ensure that repairs meet the necessary standards.

***Solution:***

The database tracks quality control checks, ensuring that each device undergoes thorough inspection before being shipped back to the customer. This reduces the likelihood of defective repairs and increases customer satisfaction.

# Design Decisions

## 1. Device

The Device entity includes device information (such as manufacturer, serial number, etc.), status, and owner information.

### *Why the entity is included:*

The Device entity stores detailed information about each device being handled, such as manufacturer, serial number, status, and owner information. This entity is crucial for tracking devices throughout the repair process and ensuring accurate status updates.

### *Entity Relationships with Others:*

The Device entity is related to Order, Customer, and Station through **OrderID**, **CurrentOwnerID**, and **CurrentStationID** respectively. It also tracks which employee worked on the device, the tasks performed, and its shipping status, ensuring comprehensive tracking and reporting.

## 2. Station

The Station can refer to various types of work locations or operational centers, where different processes take place.

### *Why the entity is included:*

The Station entity represents different operational stations where tasks such as repairs, inspections, and quality checks are performed. It helps in managing various stations and assigning employees and managers to specific locations for operational efficiency.

### *Entity Relationships with Others:*

The Station entity is connected to Device, StepTaskAssignment, and Employee through **StationID**. This relationship enables the allocation of employees and tasks to specific stations, ensuring each step in the repair or inspection process occurs at the correct location.

## 3. Employee

The Employee entity describes people who work in the organization.

### *Why the entity is included:*

As a critical entity in the Factory Database and resource management, the Employee entity stores information about the people working in the organization, including their roles, departments, and contact information. It helps manage workforce assignments, track performance, and enforce access control.

***Entity Relationships with Others:***

The Employee entity is associated with Department, Shipping, Station, Warehouse, and **StepTaskAssignment** through **EmployeeID** and **DepartmentID**. This helps assign employees to departments and stations, manage tasks, and oversee their activities and performance.

## 4. Department

The Department entity is used to identify each department and manager 's name.

***Why the entity is included:***

The Department entity helps organize employees by grouping them into functional departments and identifying department managers. Instead of including all employees and department attributes in the Employee entity, this approach reduces the repetition of department information. This structure is critical for managing team assignments, resource allocation, and reporting.

***Entity Relationships with Others:***

The Department entity is associated with the Employee entity. Each employee is assigned to a department, which includes the department name and manager information.

## 5. Warehouse

The Warehouse is a place with a physical location where parts are stored.

***Why the entity is included:***

The Warehouse entity is necessary to track the physical locations where parts are stored. It helps in managing inventory effectively, ensuring that parts are available when needed for device repairs, and avoiding delays caused by shortages.

***Entity Relationships with Others:***

The Warehouse entity is related to PartsInventory, where each warehouse stores multiple parts. The **WarehouseID** acts as a foreign key in the PartsInventory entity, linking each part



to its respective warehouse. Additionally, the Warehouse entity is associated with Employee, with each warehouse being managed by one employee. This helps ensure accountability and proper management of resources.

## 6. StepPartAssignment

This entity tracks the amount of used parts requested.

### *Why the entity is included:*

The StepPartAssignment entity is included to track the usage of parts needed to complete specific repair steps. This ensures proper allocation of parts to different steps, preventing inventory discrepancies and ensuring that parts are available when specific repair steps are being executed.

### *Entity Relationships with Others:*

The StepPartAssignment entity connects the Step and PartsInventory entities, using the **StepID** and **PartID** as foreign keys. Each step may require multiple parts, and each part can be used in multiple steps, establishing a many-to-many relationship. This ensures traceability of parts used in each step and helps in inventory management.

## 7. PartsInventory

The Parts inventory refers to the collection and management of spare parts or components that are kept in stock for use in repair.

### *Why the entity is included:*

The PartsInventory entity is included to maintain an accurate record of all spare parts and components available in the warehouse. This helps ensure that the necessary parts are always in stock and available for device repair, thus minimizing downtime and improving efficiency.

### *Entity Relationships with Others:*

The PartsInventory entity is related to the Warehouse entity, as each part is stored in a specific warehouse, and the **WarehouseID** serves as a foreign key. Additionally, it is linked to the Supplier entity, with the **SupplierID** acting as a foreign key to identify where each part originates from. PartsInventory is also connected to StepPartAssignment, where each part can be assigned to multiple steps to fulfill repair requirements.

## 8. Supplier

The Supplier entity is the organization that supplies parts and materials.

### *Why the entity is included:*

The Supplier entity is necessary to track the vendors providing parts and materials to the warehouse. This helps in managing relationships with suppliers, ensuring that parts are sourced efficiently, and facilitating reorders when stock levels are low.

**Entity Relationships with Others:** The Supplier entity is connected to the PartsInventory entity, with the **SupplierID** serving as a foreign key in PartsInventory. This allows the system to track which supplier provided which parts, helping in supply chain management and ensuring that parts availability can be managed effectively by interacting with the right supplier.

## 9. Order

The Order entity will track customer orders, linked to the devices' repair situation and the payment.

### *Why the entity is included:*

Order entity is where the flow begins. After a customer hand in a device for repairing a row will be created to trace the whole process.

### **Entity Relationships with Others:**

This entity uses FK **CustomerID**, and **DeviceID** to link to a customer and a device. Its PK **OrderID** is also used by multiple entities, such as: Task, Shipping, Survey, and Payment.

## 10. Task

The Task entity is a larger goal waiting to be done. It can be related to multiple smaller tasks (we call them “steps” in this system).

### *Why the entity is included:*

Orders need to be fulfilled. To fulfill them, the related tasks need to be done.

### **Entity Relationships with Others:**

This entity uses FK **OrderID** to link to the order it belongs to. Its PK **TaskID** is used by StepTaskAssignment entity.

## 11. StepTaskAssignment

The entity stores the details of how each step is actually being processed. It can also be used for analytics purpose, like tracking the start and end times of steps and employees' performance.

### *Why the entity is included:*

The system needs to store the details of how the actual work is being processed, so that the work can be traced and analyzed.

### *Entity Relationships with Others:*

This entity is the bridge between the Task entity and Step entity, so it uses **TaskID** and **StepID** as its PK and FK. The FK **EmployeeID** link to the employee who assigned to perform the task, and the FK **stationID** link to the Station where the task will be performed.

## 12. Step

The step entity represents the predefined type of work in the repair process flow (Pre-repair quality control, all kinds of steps for repairing, post-repair quality control etc.) Multiple steps can be grouped as a task.

### *Why the entity is included:*

With the predefined steps, we can design or arrange the production line better.

### *Entity Relationships with Others:*

The same Step can be assigned to different Tasks, so its PK **StepID** is used by StepTaskAssignment as an FK. A Step may need some parts to finish, so its PK **StepID** is also used by StepPartAssignment entity.

## 13. Shipping

The Shipping entity typically represents the data structure related to the shipping process. Its main role is to record information associated with the shipment, which helps track and manage the delivery status of products.

### ***Why the entity is included:***

The existence of this **entity** helps manage the entire shipping process, track the status of goods, and provide detailed shipping information for inquiries or analysis when needed

### ***Entity Relationships with Others:***

The entity records details about customer orders, and **OrderID** acts as a foreign key (FK) from the Order entity. This establishes a relationship where each shipping record corresponds to a specific order, enabling tracking and management of shipments associated with customer orders. The shipping process is typically carried out by employees responsible for handling logistics, so **ShippedBy** is a FK that references the Employee entity.

## **14. Survey**

The Survey entity represents a structured way to gather feedback from customers regarding their opinions. It plays a vital role in assessing customer satisfaction, product feedback, or service evaluations.

### ***Why the entity is included:***

The entity is crucial for collecting and analyzing customer feedback.

### ***Entity Relationships with Others:***

The Survey entity collects feedback that is specifically tied to customer orders, and **OrderID** serves as a FK linking to the Order entity. This relationship allows the survey data to reference the specific order related to the feedback, enabling businesses to analyze customer satisfaction based on individual service.

## **15. Payment**

The Payment entity represents the financial transaction associated with an order. It plays a critical role in tracking how and when customers pay for their purchases.

### ***Why the entity is included:***

The entity is vital for managing financial transactions in a business. By organizing payment data, organizations can effectively track and manage payments, ensure accurate financial reporting.

### ***Entity Relationships with Others:***

The Payment entity records detailed payment information for each customer order, and **OrderID** serves as a FK linking to the Order entity. This relationship allows the entity to associate specific payment records with the corresponding orders, ensuring that all financial transactions are accurately tracked.

## 16. Customer

The Customer entity represents the individuals that purchase services from a business.

### *Why the entity is included:*

The entity is fundamental for managing customer relationships in a business. By organizing customer data, organizations can enhance customer service, improve marketing strategies, and maintain accurate records for order processing.

### *Entity Relationships with Others:*

The relationship between the Customer entity and the Order entity ensures that each customer's information is linked to their corresponding orders. The **CustomerID** serves as a **FK** in the Order entity, allowing the system to associate each order with the specific customer who placed it.

## 17. DeviceModel

Stores information about the manufacturer and model of devices.

### *Why the Entity is Included*

The DeviceModel entity stores information about the manufacturer and model details of each device handled in the repair process. It ensures consistency in device classification and enables tracking of model-specific issues or trends, facilitating quality control and efficient inventory management for spare parts.

### *Entity Relationships with Others*

The DeviceModel entity is connected to the Device entity through **DeviceModelID**, which acts as a primary key in the **DeviceModel** entity and a foreign key in the Device entity. This relationship enables detailed tracking of device-specific issues across different models and ensures accurate reporting of model-related trends.

## 18. Address

Centralized table for address details

### ***Why the Entity is Included:***

The Address entity standardizes the storage of address details across the database. It is used to maintain consistent and accurate address information for customers, employees, and suppliers. This approach minimizes redundancy and ensures a single source of truth for address-related data.

### **Entity Relationships with Others**

The Address entity serves as a centralized table and is related to multiple entities through AddressID, which acts as a primary key in the Address entity and a foreign key in the Customer, Shipping, Employee, and Supplier entities. This structure supports scalability and ensures that updates to address information are applied consistently across the system.

## Entity Relationships

- **Customer - Order:** A single customer can place multiple orders, but each order is linked to only one customer.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Customer – Device:** A single customer can have multiple devices, but each device is linked to only one customer.  
**Relationship with Cardinality:** One-to-(Zero-or-Many).
- **Device – Order:** A device can have multiple orders, but each order is linked to one device.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Order - Payment:** Each order is associated with one payment, and each payment is linked to one order.  
**Relationship with Cardinality:** One-to-(One-to-One).
- **Order - Task:** Each order can have many tasks, similarly, each task can be related to many orders.  
**Relationship with Cardinality:** One-to-(Zero-or-Many).
- **Task - StepTaskAssignment:** Each task can have many steps associated with it, and each step is part of one task.  
**Relationship with Cardinality:** One-to-(One-or-Many).

- **Step - StepPartAssignment:** Each step can require multiple parts, and each part can be required by multiple steps.  
**Relationship with Cardinality:** Many-to-(Many-to-Many).
- **PartsInventory - StepPartAssignment:** Each part can be associated with multiple steps, but each step will have only one part assigned.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Warehouse - PartsInventory:** Each warehouse can store multiple parts, and each part is associated with one warehouse.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Employee – Warehouse:** An employee can manage multiple warehouses, but each warehouse is managed by one employee.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **PartsInventory - Supplier:** Each part is supplied by one supplier, and each supplier can supply multiple parts.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Station - StepTaskAssignment:** Each station can be associated with multiple task assignments, but each task assignment is linked to one station.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Device - Station:**  
A device can be at one station at a time, but a station can have multiple devices.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Station - Employee:** One employee can manage many stations, and each station is managed by one employee.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Employee - StepTaskAssignment:** Each employee can be assigned multiple tasks, and each task may be assigned to multiple employees.  
**Relationship with Cardinality:** One-to-(One-or-One).
- **Shipping - Order:** Each shipping address may have multiple orders, but one order has one shipping address.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Order - Survey:** A survey is associated with one order, and one order will have one survey.  
**Relationship with Cardinality:** One-to-(Zero-or-One)
- **Department – Employee:** Each employee belongs to one department, but a department can have multiple employees.  
**Relationship with Cardinality:** One-to-(Zero-or-Many).

- **DeviceModel – Device:** Each device is associated with one device model, but a single device model can correspond to multiple devices.  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Address – Customer:** Each customer has one address, but multiple customers can share the same address (e.g., family members or group orders).  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Address – Employee:** Each employee has one address, but multiple employees can reside at the same address (e.g., housing provided by the company).  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Address – Supplier:** Each supplier has one address, but multiple suppliers can share the same address (e.g., branches or offices of the same supplier).  
**Relationship with Cardinality:** One-to-(One-or-Many).
- **Address – Warehouse:** Each warehouse has one address, but multiple warehouses can share the same address (e.g., large facilities with multiple storage units).  
**Relationship with Cardinality:** One-to-(One-or-Many).

## Business Flow and Business Rules

1. **Customer places an Order:**
  - a. With the **Device** information for repairing.
  - b. Create the **Payment** that belongs to the **Order**.
    - i. An **Order** should have one **Payment** (Customers must pay when they place an order; payment plans are not allowed).
2. **Can create one or multiple Task(s) to aggregate the Step(s) related to this Order with StepTaskAssignment(s).**
  - a. An **Order** can have zero **Task**.
  - b. We can track where the **Device** is by checking the latest **StepTaskAssignment** of the **Order**.
  - c. Each **StepTaskAssignment** has a relationship with an **Employee** who is responsible for working on the **Step**.
  - d. Each row in the **Step** table represents a pre-defined step to complete a **Task**.
    - i. For example, Pre-repair quality control, all kinds of steps for repairing, post-repair quality control, and shipping could all be a row in the **Step** table.
    - ii. When all the related **StepTaskAssignment(s)** are completed, then we can consider the **Task** is completed.



- e. **StepPartAssignment** uses a **Step** to link its relationship to the **PartsInventory** that are needed to finish the **Step** (using **StepPartAssignment**)
    - i. **PartsInventory** will be assigned to a **Warehouse**.
  - f. A **Station** will assign to a **StepTaskAssignment**, so that an **Employee** can handle the **StepTaskAssignment** at a **Station**.
  - g. A **Customer's Device** can be swapped for a new device (e.g. trade in, refunding), so the current owner might be transferred to the new device.
3. **When all the related Task(s) are done, a Shipping record will be created to ship the Device back to the Customer**
- a. This does not apply to the **Order(s)** end up with refunding.
4. **Customer can submit at most one Survey after a service is completed**
- a. The submission of a **Survey** is not mandatory, so an **Order** could have zero **Survey**.