

SOFTWARE REQUIREMENTS SPECIFICATION

POINT OF SALE SYSTEM

Application Name	Point of Sale System
Team Name	Group 2
Team Member	<ul style="list-style-type: none">- Ryan Hadi Chandra- Sheren Yang- Alif Taufiqurrahman
Task	<ul style="list-style-type: none">- Ryan Hadi Chandra (Product Management, Show Invoice Detail, Report Revenue)- Sheren Yang (Customer Management, Add New Invoice, Export PDF)- Alif Taufiqurrahman (Invoice List, Edit Invoice, Excel Report, Swagger)
Release Date	26 July 2024

Point of Sale System API for Efficient Transaction Management, Customer and Product Handling, and Comprehensive Reporting

2024

Table of Contents

1.	Introduction.....	1
1.1.	Purpose.....	1
1.2.	Scope.....	1
1.3.	Definitions, Acronyms, and Abbreviations	1
1.4.	References.....	2
1.5.	Overview	2
2.	Overall Description.....	2
2.1.	Product Perspective.....	2
2.2.	Product Functions	2
2.3.	Technologies and Tools	3
2.4.	User Characteristics	3
2.5.	Business Flow	3
2.6.	Database Design.....	4
2.7.	Class Diagram.....	4
2.8.	Functional Requirements	5
3.	System Features	7
3.1.	Invoice List	7
3.2.	Add New Invoice	8
3.3.	Show Invoice Detail.....	9
3.4.	Export to PDF	9
3.5.	Edit Invoice.....	9
3.6.	Delete Invoice Products	10
3.7.	Product Management	11
3.8.	Customer Management	12
3.9.	Report.....	13
3.10.	Excel Export.....	14

1. Introduction

The introduction section of this Software Requirements Specification (SRS) provides an overview of the Point of Sale (POS) system, including its purpose, scope, key definitions, references, and a general overview. This document is intended to guide developers, stakeholders, and other parties involved in the system's development and maintenance.

1.1. Purpose

The purpose of this SRS document is to outline the functional requirements of the POS system. The document serves as a comprehensive guide for the development and implementation of a RESTful API to manage key POS functionalities, including invoice management, product management, customer management, and reporting. This SRS aims to ensure that all parties have a clear understanding of the system requirements and to provide a basis for subsequent system development.

1.2. Scope

The POS API system is designed to facilitate key point-of-sale operations through a set of RESTful APIs. The scope includes:

- Invoice Management: Create, read, and update operations for invoices, including the ability to add products, edit invoices within a specified time, and export invoice details in PDF and excel format.
- Product Management: Create, read, update, and delete (CRUD) operations for products, with functionalities to activate or deactivate products and import products from an Excel file.
- Customer Management: CRUD operations for customers, including activation and deactivation functionalities.
- Reporting and Export: Generation of revenue reports and export of invoice lists to Excel based on specified filters.
- Data integrity: Implementing data validation.

This SRS does not cover the user interface design or non-API-related components of the POS system.

1.3. Definitions, Acronyms, and Abbreviations

The following are each standard or typing convention used in this software requirements specification document.

Term	Deskripsi
POS	Point of Sale, the name of this system.
SRS	Software Requirements Specification.
API	Application Programming Interface.
CRUD	Create, Read, Update, Delete.
REST	Representational State Transfer.
JSON	JavaScript Object Notation.
HTTPS	HyperText Transfer Protocol Secure.
PDF	Portable Document Format.

FR-N	Code that represents the numbering of functional requirements.
------	--

1.4. References

This section lists the resources referenced in this development of system and this SRS. The references includes:

- Swagger Documentation for detailed API specifications and usage.
- RESTful API best practices for designing and implementing RESTful APIs.

1.5. Overview

The POS system aims to streamline the management of sales processes by providing an API solution. The system includes functionalities for handling invoices, managing products and customers, and generating reports. The APIs are designed to be consistent with RESTful principles, ensuring easy integration and usability for developers. The system also includes comprehensive API documentation to support developers in understanding and utilizing the API endpoints effectively. This SRS will serve as the foundational document guiding the development, deployment, and maintenance of the POS API system.

2. Overall Description

This section provides a high-level overview of the POS system, detailing its context within the business flow and the functionalities of the system.

2.1. Product Perspective

The POS system is an API component designed to provide standardized interfaces for managing point-of-sale operations. It acts as the backend service to interact with the underlying database and business logic. The system utilizes MySQL as its primary database management system, this relational database supports the efficient storage and retrieval of data related to invoices, products, and customers. The primary goal is to centralize and streamline POS operations, ensuring data consistency and efficient transaction processing.

2.2. Product Functions

The POS system provides a range of functionalities, including:

- a. Invoice Management:
 - Creating, reading, and updating invoices.
 - Searching and filtering invoices by customer details, date, and other criteria.
 - Exporting invoice details in PDF format.
- b. Product Management:
 - CRUD operations for products.
 - Activate or inactivate products.
 - Importing products lists from external sources in excel files.
- c. Customer Management:
 - CRUD operations for customers.
 - Activate or inactivate customers.
- d. Reporting and Export:

- Generating revenue reports based on specified timeframes (day, month, year).
- Exporting invoice lists to excel, filtered by various criteria.

2.3. Technologies and Tools

The POS system is developed using Java Spring Boot, a popular framework for building enterprise-grade applications. Java Spring Boot provides a robust and scalable platform for developing RESTful APIs, with built-in support for various features like dependency injection, security, and data access. This technology choice ensures high performance, ease of development, and strong community support. Key technologies and tools used in this project include:

- Java Spring Boot: For building the backend API services.
- MySQL: As the relational database management system.
- Swagger: For API documentation.

2.4. User Characteristics

The primary users of the POS system include:

User Characteristic	
Category	Description
Primary Stakeholder	
Developers	Develop and integrate the API with systems.
Analysts	Use the reporting and data export features to analyze.
End Users	Interact and uses the features of the application.

2.5. Business Flow

The business flow provides a visual representation of the key processes and interactions within the POS system, illustrating how different components interact.

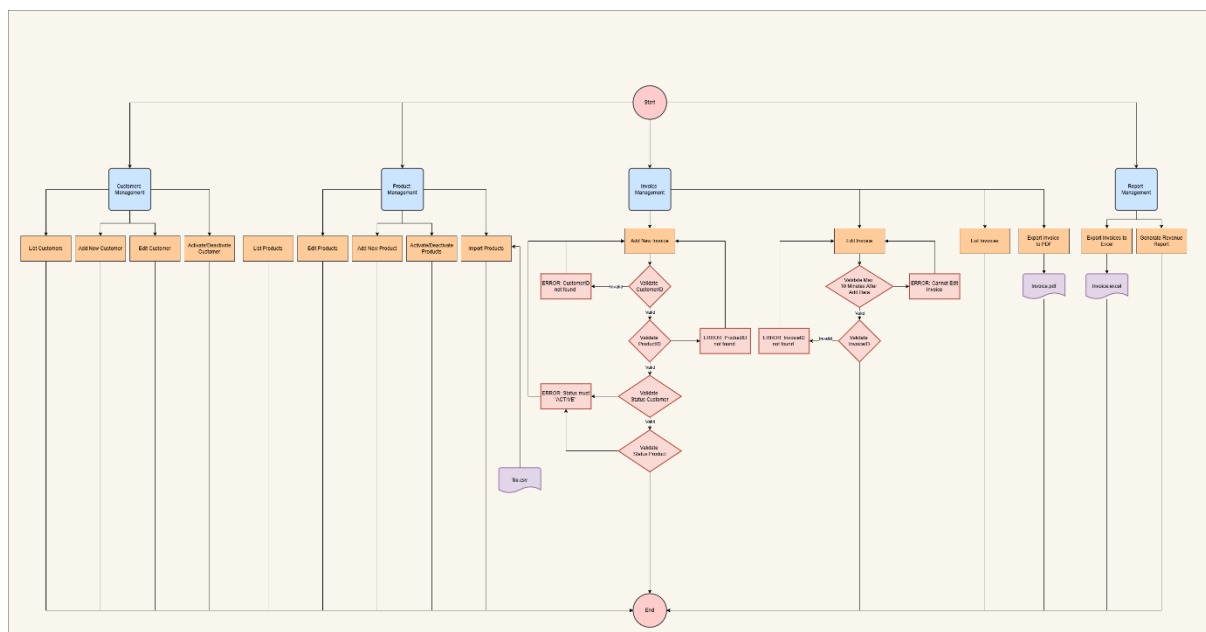


Figure 1. Business Flowchart

The figure 1 outlines the sequence of operations, including steps for customer management, product management, invoice management, and report management. The business flow charts helps stakeholders to understand the overall process.

2.6. Database Design

The database design section presents the structure of the database in the POST system. Includes the tables, relationships, and key attributes required to store and manage data efficiently.



Figure 2. Database diagram

The figure 2 showcases the schema for handling invoices, products, and customer in the database. The table for the system includes customer, product, invoice, and invoice product.

2.7. Class Diagram

The class diagram provides a detailed view of the object-oriented design of the POS system. It depicts the classes, their attributes, methods, and relationships.

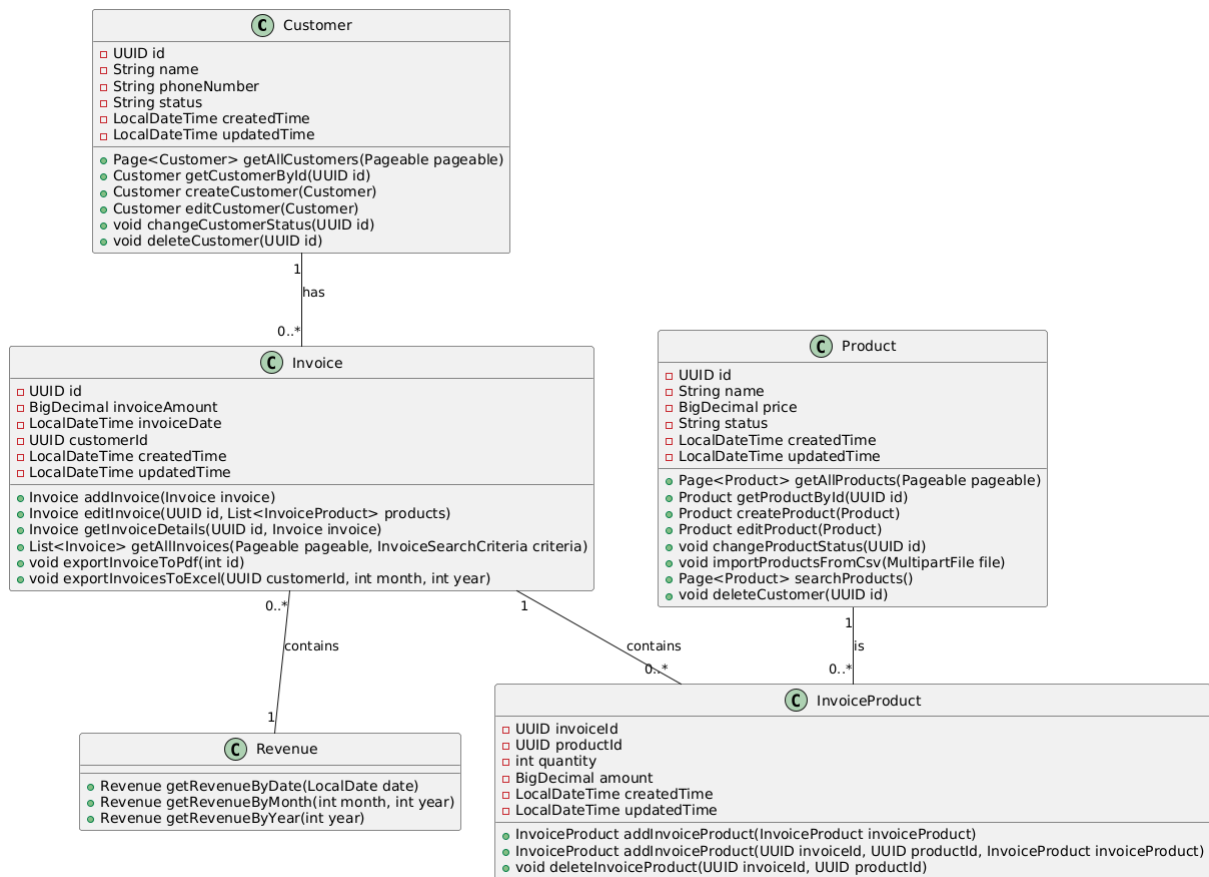


Figure 3. Class Diagram

The figure 3 serves as a blueprint for developers to understand the system's structure and the interactions between different components at the code level.

2.8. Functional Requirements

Functional Requirements		
ID	Feature	Description
FR-01	API Pagination	All list-returning APIs must support pagination with parameters for page number and page size.
FR-02	RESTful API Design	The API must follow RESTful principles using standard HTTP methods and status codes.
FR-03	Parameter Validation	The API must validate input parameters to ensure they meet required formats, types, and constraints, returning errors for invalid inputs.
FR-04	Entity Timestamps	Each entity must include created_time and updated_time fields to track creation and modification times automatically.
FR-05	API Documentation	The API must have documentation using Swagger.
FR-06	Search Invoices by Customer Name	The API must allow searching invoices by customer name using partial matches.
FR-07	Filter Invoices	The API must support filtering invoices by customer ID, date, and month, allowing multiple criteria.

FR-08	Sort Invoices	The API must support sorting invoices by date or amount, with options for ascending or descending order.
FR-09	Invoice Fields	The invoice list response must include ID, amount, customer name, and date.
FR-10	Create New Invoice	The API must enable the creation of a new invoice with customer selection, invoice details, and product management.
FR-11	Show Invoice Details	The API must provide detailed information for a specific invoice, including customer details, invoice details, and product list.
FR-12	Export Invoice to PDF	The API must allow exporting invoice details to a PDF, including all relevant information formatted for print.
FR-13	Edit Invoice	The API must support editing invoices within 10 minutes of creation, with functionality similar to invoice creation.
FR-14	List Products	The API must provide a list of products with ID, name, price, and status, supporting search and sorting.
FR-15	Add New Product	The API must allow adding new products with attributes like name, price, and status.
FR-16	Edit Product	The API must support updating product details, including name, price, and status.
FR-17	Activate/Deactivate Product	The API must enable activation and deactivation of products based on their status.
FR-18	Import Products from Excel CSV	The API must support importing products from an Excel CSV file.
FR-19	List Customers	The API must provide a list of customers with ID, name, and phone number, supporting search and sorting.
FR-20	Add New Customer	The API must allow adding new customers with details like name and phone number.
FR-21	Edit Customer	The API must support updating customer details, including name and phone number.
FR-22	Activate/Deactivate Customer	The API must enable activation and deactivation of customer records.
FR-23	Generate Revenue Report	The API may generate revenue reports for specified periods (day, month, year), showing total revenue.
FR-24	Export Invoice List to Excel	The API may export a filtered list of invoices to an Excel file, including details like ID, customer ID, and product information.

3. System Features

This section outlines the key features of the POS system, detailing the functionalities that the API provides to support point-of-sale operations.

3.1. API Documentation (Swagger)

The API Documentation feature provides documentation for the POS API system using Swagger. This documentation serves as a resource for integrating with the API, ensuring to utilize the API's functionalities efficiently.

Endpoint: GET /swagger-ui/index.html#/ – Access to the web-based interface Swagger UI.

Endpoint: GET /v3/api-docs – Returning the API documentation in JSON format.

The screenshot displays the Swagger UI for the 'Point Of Sale System API Documentation'. At the top, the title 'Point Of Sale System API Documentation' is shown with version 'v1' and 'OAS3' tags. Below the title, there's a 'Servers' section with a dropdown menu showing 'http://localhost:8080 - Generated server url' and an 'Authorize' button. The main content is organized into three sections: 'product-controller', 'invoice-controller', and 'invoice-product-controller'. Each section lists various API endpoints with their corresponding HTTP methods (GET, PUT, DELETE, PATCH, POST) and a lock icon indicating authentication requirements. The endpoints are color-coded: blue for GET, orange for PUT, red for DELETE, green for PATCH, and light green for POST.

Controller	Method	Endpoint	Auth
product-controller	GET	/api/v1/products/{id}	Yes
	PUT	/api/v1/products/{id}	Yes
	DELETE	/api/v1/products/{id}	Yes
	PATCH	/api/v1/products/{id}	Yes
	GET	/api/v1/products	Yes
	POST	/api/v1/products	Yes
	POST	/api/v1/products/import	Yes
	GET	/api/v1/products/search	Yes
invoice-controller	GET	/api/v1/invoices/{id}	Yes
	PUT	/api/v1/invoices/{id}	Yes
	GET	/api/v1/invoices	Yes
	POST	/api/v1/invoices	Yes
	GET	/api/v1/invoices/{id}/pdf	Yes
	GET	/api/v1/invoices/excel	Yes
invoice-product-controller	PUT	/api/v1/invoice-products/{invoiceId}/{productId}	Yes
	DELETE	/api/v1/invoice-products/{invoiceId}/{productId}	Yes
	GET	/api/v1/invoice-products	Yes
	POST	/api/v1/invoice-products	Yes

customer-controller		^
GET	/api/v1/customers/{id}	▼ 🔒
PUT	/api/v1/customers/{id}	▼ 🔒
DELETE	/api/v1/customers/{id}	▼ 🔒
PATCH	/api/v1/customers/{id}	▼ 🔒
GET	/api/v1/customers	▼ 🔒
POST	/api/v1/customers	▼ 🔒

revenue-controller		^
GET	/api/v1/revenue/year	▼ 🔒
GET	/api/v1/revenue/month	▼ 🔒
GET	/api/v1/revenue/day	▼ 🔒

3.2. Invoice List

The Invoice List functionality allows users to retrieve lists of invoices through the API with pagination. Users can search for invoices by customer name. Invoices can be filtered based on customer Id, date, and month. Invoices can be sorted by date or amount. The response includes key invoice details such as ID, invoice amount, customer name, and invoice date within the pagination.

Endpoint: GET /api/v1/invoices

Parameter	Type	Description
page	int	The page number to retrieve, starting from 0.
size	int	The number of invoices per page.
customerName	string	A partial or full customer name to search.
customerId	uuid	The ID of the customer to filter invoices.
invoiceDate	date	A specific date to filter invoices.
month	string	A specific month to filter invoices by, in MM format.
sortBy	string	The field to sort by, either 'date' or 'amount'.
sortAsc	string	Whether to sort in ascending order ('true') or descending order ('false').

Example:

/api/v1/invoices?page=0&size=10&customerName=someone&month=7&sortBy=amount&sortAsc=true

3.3. Add New Invoice

The Add New Invoice feature enables users to create new invoices through the API. The system validates that the products are properly configured and have an 'active' status.

Endpoint: POST /api/v1/invoices

Request Body	Type	Description
customerId	uuid	The ID of the customer
invoiceDate	date	A specific date (yyy-MM-dd)

Example:

```
{
  "customerId": "3f2504e0-4f89-11d3-9a0c-0305e82c3305",
  "invoiceDate": "2024-07-25"
}
```

Endpoint: POST /api/v1/invoice-products

Request Body	Type	Description
invoiceId	uuid	The ID of the invoice.
productid	date	The ID of the product.
quantity	int	The quantity of the product.

Example:

```
{
  "invoiceId": "af11ab1d-6c37-40c8-a452-2932aea03f24",
  "productId": "5f2504e0-4f89-11d3-9a0c-0305e82c3304",
  "quantity": 10
}
```

3.4. Show Invoice Detail

The Show Invoice Detail feature provides detailed information about a specific invoice, including customer and invoice information, and a list of products with quantities, prices, and amounts.

Endpoint: GET /api/v1/invoices/{id}

Parameter	Type	Description
id	uuid	The ID of the invoice to retrieve details for.

Example:

/api/v1/invoices/af11ab1d-6c37-40c8-a452-2932aea03f24

3.5. Export to PDF

The Export to PDF functionality allows users to generate a PDF document containing detailed information about an invoice, formatted for easy readability and printing.

Endpoint: GET /api/v1/invoices/{id}/pdf

Parameter	Type	Description
id	uuid	The ID of the invoice to retrieve invoice to PDF format.

Example:

/api/v1/invoices/af11ab1d-6c37-40c8-a452-2932aea03f24/pdf

3.6. Edit Invoice

The Edit Invoice feature allows users to modify invoices within a 10-minute time limit from the creation time.

Endpoint: PUT /api/v1/invoices/{id}

Parameter	Type	Description
id	uuid	The ID of the invoice to edit.

Example:

/api/v1/invoices/af11ab1d-6c37-40c8-a452-2932aea03f24

Request Body	Type	Description
customerId	uuid	The ID of the customer to edit.
invoiceDate	date	A specific date (yyy-MM-dd) to edit.

Example:

```
{
  "customerId": "3f2504e0-4f89-11d3-9a0c-0305e82c3305",
  "invoiceDate": "2024-07-25"
}
```

Endpoint: PUT /api/v1/invoice-products/{invoiceId}/{productId}

Parameter	Type	Description
invoiceId	uuid	The ID of the invoice to edit.
productId	uuid	The IDE of the product to edit invoice product.

Example:

/api/v1/invoice-products/af11ab1d-6c37-40c8-a452-2932aea03f24/5f2504e0-4f89-11d3-9a0c-0305e82c3302

Request Body	Type	Description
quantity	int	The quantity of the product.

Example:

```
{
  "quantity": 10
}
```

3.7. Delete Invoice Products

The Delete Invoice Products feature allows users to remove an existing invoice product within a 10-minute time limit from the creation time. The API will perform reducing the invoice amounts.

Endpoint: DELETE /api/v1/invoice-products/{invoiceId}/{productId}

Parameter	Type	Description
invoiceId	uuid	The ID of the invoice to delete.
productId	uuid	The ID of the product to delete invoice product.

Example:

/api/v1/invoice-products/af11ab1d-6c37-40c8-a452-2932aea03f24/5f2504e0-4f89-11d3-9a0c-0305e82c3302

3.8. Product Management

The Product Management functionality provides API endpoints for managing products, including listing, adding, editing, and activating or deactivating products. The system also supports importing product data from an Excel file.

Endpoints:

- **GET** /api/v1/products – Retrieve list of products in pagination.
- **POST** /api/v1/products - Add a new product.

Request Body	Type	Description
name	string	The name of the product to create
price	decimal	The price of the product with scale 2 (.00)
status	string	The status of the product, ACTIVE or INACTIVE

Example:

```
{
  "name": "Product A",
  "price": 15.00,
  "status": "ACTIVE"
}
```

- **PUT** /api/v1/products/{id} - Edit an existing product.

Parameter	Type	Description
id	uuid	The ID of the product to edit

Example:

/api/v1/products/9c4bf32e-d80e-4082-825e-a312a38346f9

Request Body	Type	Description
Name	string	The name of the product to edit
Price	decimal	The price of the product with scale 2 (.00)
Status	string	The status of the product, ACTIVE or INACTIVE

Example:

```
{
  "name": "Product A",
  "price": 15.00,
  "status": "ACTIVE"
}
```

- **PATCH** /api/v1/products/{productId}/status - Activate or deactivate a product.

Parameter	Type	Description
id	uuid	The ID of the product to edit the status

Example:

/api/v1/products/9c4bf32e-d80e-4082-825e-a312a38346f9

- **POST** /api/v1/products/import - Import products from an Excel file.

Request Param	Type	Description
file	MultipartFile	The file of the product to import.

CSV data example:

Smartphone A,19.99,active

- **GET** /api/v1/products/search – Search a product in pagination

Parameter	Type	Description
name	string	The name of the product to search
status	string	The status of the product to search, 1 for active and 0 for inactive
page	int	Specify the page number of searched product
size	int	Specify the size of the page
sort	string[]	Sort the field with the order (default = “name, asc”

Example:

/api/v1/products/search?name=producta&status=1

3.9. Customer Management

The Customer Management feature allows users to manage customer information, including listing, adding, editing, and activating or deactivating customers.

Endpoints:

- **GET** /api/v1/customers - List and search customers.
- **POST** /api/v1/customers - Add a new customer.

Request Body	Type	Description
name	string	The name of the customer to create
phoneNumber	string	The phone number of the customer with Indonesia phone format and from 9 to 13 digits.
status	string	The status of the product, ACTIVE or INACTIVE

Example:

```
{
  "name": "Someone",
  "phoneNumber": "+62123456789",
  "status": "ACTIVE"
}
```

- **PUT** /api/v1/customers/{id} - Edit an existing customer.

Parameter	Type	Description
id	uuid	The ID of the customer to edit

Example:

/api/v1/customers/ 8ee4ae4c-67c6-4809-aa4a-c5d30df25115

Request Body	Type	Description
name	string	The name of the customer to edit
phoneNumber	string	The phone number of the customer with Indonesia phone format and from 9 to 13 digits.
status	string	The status of the product, ACTIVE or INACTIVE

Example:

```
{
  "name": "Someone Update",
  "phoneNumber": "+62123456789",
  "status": "ACTIVE"
}
```

- **PATCH** /api/v1/customers/{id} - Activate or deactivate a customer.

Parameter	Type	Description
id	uuid	The ID of the customer to edit the status.

Example:

/api/v1/customers/ 8ee4ae4c-67c6-4809-aa4a-c5d30df25115

- **DELETE** /api/v1/customers/{id} – Delete a customer.

Parameter	Type	Description
id	uuid	The ID of the customer to delete

Example:

/api/v1/customers/ 8ee4ae4c-67c6-4809-aa4a-c5d30df25115

3.10. Report

The Report feature provides optional reporting capabilities, allowing users to generate revenue reports based on specific timeframes, such as daily, monthly, or yearly reports.

Endpoints:

- **GET** /api/v1/revenue/day

Request Param	Type	Description
date	date	The date to report the revenue.

Example:

/api/v1/revenue?date=2024-07-25

- **GET** /api/v1/revenue/month

Request Param	Type	Description
month	int	The month to specify report the revenue.

year	int	The year to specify with month.
------	-----	---------------------------------

Example:

/api/v1/revenue?month=7&year=2025

- **GET** /api/v1/revenue/year

Request Param	Type	Description
year	int	The year to specify the report of revenue.

Example:

/api/v1/revenue?year=2024

3.11. Excel Export

The Excel Export feature allows users to export invoice data to an Excel file based on specified filters, including customer, month, or year.

Endpoint: **GET** /api/v1/invoices/excel

Parameter	Type	Description
customerId	uuid	The ID of the customer to export the invoices to excel.
month	int	The month to filter the invoices.
year	int	The year to filter the invoices.

Example:

/api/v1/invoices/excel?customerId=3f2504e0-4f89-11d3-9a0c-0305e82c3304&month=7&year=2024