Low Level Design Document

Introduction

This Low Level Design (LLD) document outlines the core components and implementation details for **ShopList - Inventory Management API**. The API provides CRUD operations, pagination, and search for managing product inventory in small e-commerce shops, using Node.js, Express, and PostgreSQL.

1. System Components

Component	**Description**	**Key Responsibilities**
API Server	Express.js application	Handle HTTP requests, route mapping
Controller Layer	Request handlers	Validate input, call services
Service Layer	Business logic	CRUD, pagination, search
Data Access Layer	PostgreSQL integration (using pg/ORM)	Query/modify database
Database	PostgreSQL	Store product data

2. Class/Interface Overview

Class/Module	**Description**	**Key Methods/Attributes**
`ProductController`	Express route handlers	`createProduct` , `getProducts` , `getProductById` , `updateProduct` , `deleteProduct`
`ProductService`	Business logic	`create` , `list` , `getById` , `update` , `delete`
`ProductRepository`	DB access (SQL/ORM)	`insert` , `find` , `findById` , `update` , `remove`

Relationships:

- Controller → Service → Repository

Example Method Signatures:

```
// ProductService
create(data)
list({ page, limit, search })
getById(id)
update(id, data)
delete(id)
```

3. Data Structure Overview

Field	Type	Constraints	Description
id	UUID	PK, auto-generated	Product identifier
name	String	Not null	Product name
description	String	Optional	Product description
price	Decimal	Not null, >= 0	Product price
quantity	Integer	Not null, >= 0	Inventory count
created_at	Timestamp	Auto-generated	Creation timestamp
updated_at	Timestamp	Auto-updated	Last update timestamp

---

4. Algorithms / Logic

Product Listing with Pagination & Search (Pseudocode):

```
function list({ page, limit, search }) {
  offset = (page - 1) * limit
  query = "SELECT * FROM products WHERE name ILIKE %search% ORDER BY created_at
  return db.query(query)
}
```

CRUD Flow:

- Validate input
- Call service method
- Service calls repository
- Return result or error

---

5. Error Handling

Scenario	Handling Approach
Invalid input data	400 Bad Request, error message
Product not found	404 Not Found, error message
Database connection/query error	500 Internal Server Error, log details
Duplicate product (if unique)	409 Conflict, error message
Unauthorized access (future)	401 Unauthorized

End of Document