

Pharmacy Management System

POC

Low Level Design (LLD)



Date: [19/05/2022](#)

DOCUMENT APPROVAL

Approvers of this document

Name	Department	Role	Signature	Date

DOCUMENT CHANGE HISTORY

Document Version #	Author	Date	Description
			Pharmacy Management System

Table of Contents

1.0	Document Purpose	5
2.0	Intended Audience	5
3.0	Project Background, Objective(s)	5
3.1	Project Background	5
3.2	Project Objective	5
4.0	Design Pattern	
5.0	Solution Diagram	
6.0	Solution Steps	
7.0	Classes/function name	

Overview:

A pharmacy management system is, especially, to facilitate managing the supply of Drugs needed by a hospital, which makes it easier to treat hospital patients in general. This system involves information technology and databases as a repository of information that is useful in managing the hospital, various tools needed to build a reliable system, and discusses those interests. Compliance with data enhancements and manual drug supply activities being automated completes review. A computer displays information about the Drugs, such as its dosage and expiration date. In large medical stores, manually handling the specifics of all the drugs becomes very tough. We can keep track of all the Drugs by using this pharmacy management system. It is updated with new information as new Drugs are introduced, and it includes an expiration date as well as a search option. When we search the name of a Drug, it displays the Drugs details. The project is totally built on administrative end and thus only the administration is guaranteed the access.

Purpose:

The main purpose of the Pharmacy Management System is to manage the details of drugs, doctor user, orders in a regular manner and to improve the maintenance and manipulation of the drugs in the medicals. The pharmacy management system will be used to minimize the time and resource by maintaining the details of the drug systemically so that the data can be used in possible quickest time.

Scope:

One of the most important responsibilities of pharmacy management is to supervise and manage the pharmacy employees to ensure healthy working relationships and outcomes. Each of these functions is critical to the pharmacy's operation and should be improved. It is more affordable for individual pharmacies to set up some ready-made solutions, Pharmacy Management System will reduce the burden on pharmacist and will make the system efficient by providing the more accurate details about drugs in the medical.

System Requirements

Software Requirements

- Service: Rest API
- Technologies: ASP.NET Core, React JS
- Database: SQL Server

Hardware Requirements

- Processor: Intel i3
- RAM : At least 1 GB
- Hard Disk: 200 GB or More

1.0 Document Purpose

This Document describes the solution architecture for Pharmacy Management System.

2.0 Intended Audience

This document is intended as a reference for the following roles and Pharmaceutical Shops who are interested in the Pharmacy Management System technical architecture.

Role	Nature of Engagement in WB Classics Portal Technical Architecture
Product Owners/SME	Key Pharmaceutical Shops to ensure that the architecture is aligned with business goals.
Business Analysts	Business analysts are one of the stakeholders who are informed with the key architectural decisions.
Enterprise Architects	To enforce Pharmacy Management Platform Architecture is aligned to business goals and architecture, architectural guidelines.
Solution Architects	To ensure solution design and architecture is aligned to business requirements, architectural guidelines.
Developers	Use Technical Architecture Document as the guiding document for detail design and implantation approach to align with Pharmacy management System

3.0 Project Background, Objective(s)

3.1Project Background

Pharmacy management System leads to perform various operation like where admin can add supplier of the drugs and doctor has to order those drugs .

3.2Project Objective

The main aim of our project is the management of the database of the pharmaceutical shop. This is done by creating a database of the available medicines in the shop and we create API for the following services

Requirements:

User API:

For Admin or Doctor Login/Signup: Sign up with basic details name, contact, email and password.

Orders API:

1. **//For Admin**
2. View doctor orders: View orders placed by all the doctors and check if the order is valid or not.
3. Verify orders: On validating the order, he can verify the order.
4. Add order to pickedUp section: If the order has been picked up then add that order to pickedUpOrders section.
5. Views Sales Report.
6. Download/Print sales report.
7. **//For Doctor**
8. View drugs: View and check details of each and every drug available.
9. Buy drugs.

Supplier Inventory:

1. **//For Admin**
2. View suppliers: View and check details of each and every supplier.
3. Add/Edit/Delete supplier.
4. View drugs: View and check details of each and every drug available.
5. Add/Edit/Delete drugs.

Database Schema:

User Database

User Id	Name	Contact	Email	Hash Password	Address
1(key)	Pushkar	54789635	p@gmail.com	Abcd#234	abcxyz
2(key)	Uzma	9875244	uzm@gmail.com	Yujh#234	Deftyu
3(key)	Kalyan	98300457	kalyan@gmail.com	Poku#234	Wysghj

Primary Key

Order Database

Order Table

Order Id	User Id
1	2

Foreign Key Foreign Key

Order Details:

Order Id	Drug Id	Quantity	Payment	Total	Balance
1	BIN00090	2	367	734	0

Primary Key Foreign Key

Pick Up-Orders

Order Id	PickUpDate
1	11/05/2022

Foreign Key

Inventory Database:

Supplier Table:

Supplier Id	Supplier Name	Supplier Email
1	Santhosh	sant@gmail.com

Primary Key

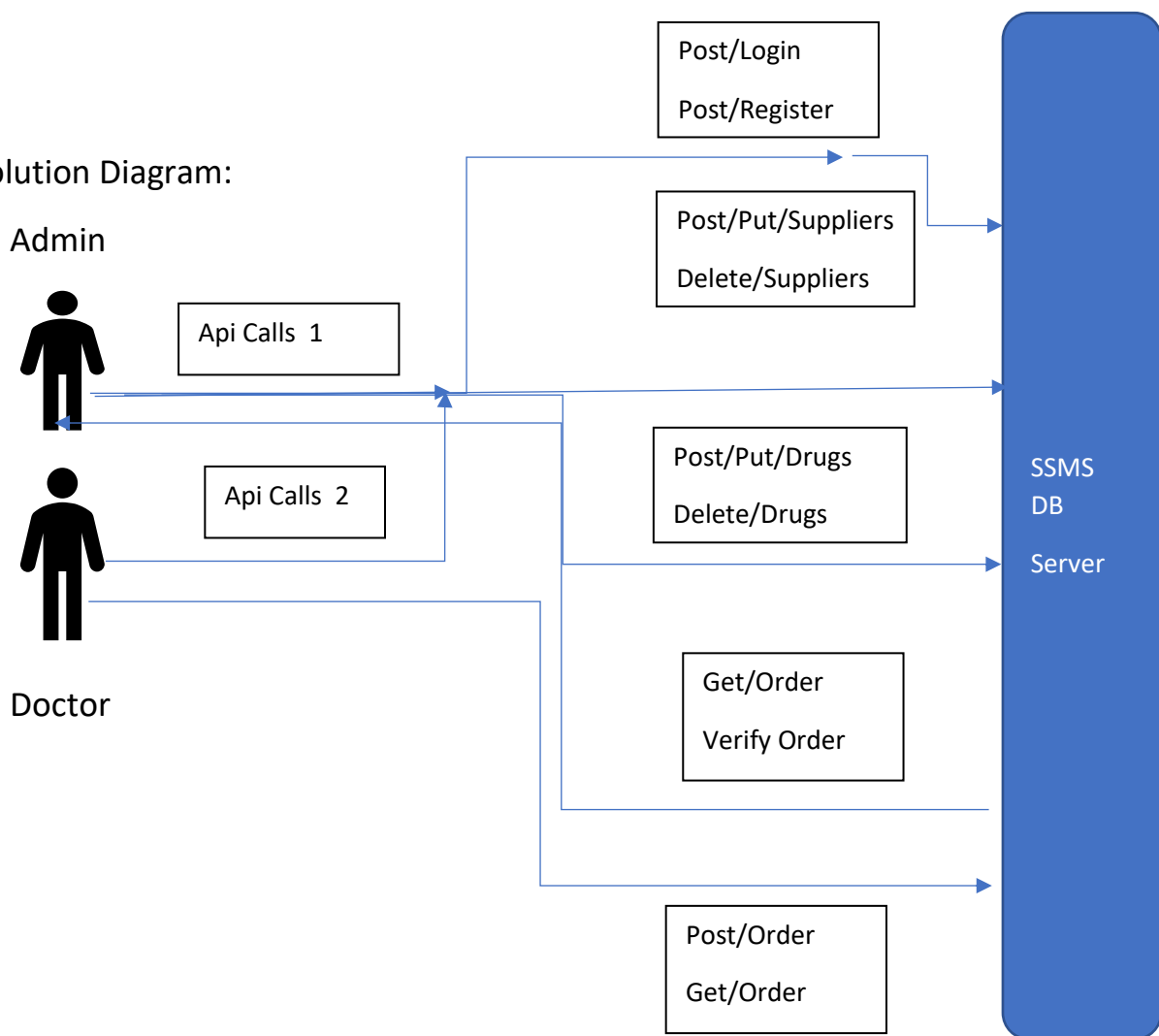
Inventory Drug Table:

Drug Id	Drug Name	Batch Id	Expiry Date	Price	Supplier Id
BIN00090	Rosuvastatin	JKN2312A	19/05/2024	367	1

Primary key

Foreign Key

Solution Diagram:



User Data Base:

User Registration

1. User will enter the required details such as Name, email, phone number, and password and Role admin/doctor click submit button browser directs the request to User registration API (Post)

2. Call reaches the API gateway
3. API gateway does the routing and forwards the request to User Controller and check the client-side validation.
4. Add User () will call the action Method to do the input validation it will have the as argument to perform the validation
 - a. If validation fails, then it will return the error code and error description. with status code
 - b. If validation is successful, then it will post the add User () to store the data in database
5. It sends a response body with HTTP Success response to Add User.
6. This Add User returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

Order Data Base:

Generated Order:

Drugs Listing:

1. Admin wants to get the Drugs details where Doctor add drugs, enters the Order Id in parameter for which drugs wants to see the details, browser directs the request to Drugs List API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to this handle function calls the GetAllOrder ()
4. GetAllOrder () will call the which to fetch the data from database.
5. It sends response body with HTTP Success response code to getAllOrder.
6. getAllOrder returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

Order Details:

Drugs Purchase List:

1. Admin wants to get the Drugs details which Doctor add drugs how many quantity for that browser directs the request to Drugs List API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to this handle function calls the GetAllOrderList ()
4. GetAllOrderList () will call the which to fetch the data from database.
5. It sends response body with HTTP Success response code to GetAllOrderList.
6. GetAllOrderList () returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

Pickup Details:

PickUpOrders:

1. Admin wants to get the Drugs Order which Doctor add drugs how many quantities for that browser directs the request to Drugs List API and if order is add then post order in pickup-order
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to this handle function calls the getAllOrdeList ()
4. Then we have to check if order is added then post this order to PickUpOrder PostPickOrder ()
5. GetAllOrderList () will call the which to fetch the data from database.
6. PostPickOrder () will post the order to our database, and we can get back this.

Inventory DB:

Supplier Info:

Add Supplier

1. Admin will enter the required details such as Supplier Id, Supplier name, click submit button browser directs the request to Add Supplier API (Post)
2. Call reaches the API gateway
3. API gateway does the routing and forwards the request to Supplier Controller and check the client-side validation.
4. AddSupplier () will call the action Method to do the input validation it will have the as argument to perform the validation
 - If validation fails, then it will return the error code and error description. with status code
 - If validation is successful, then it will post the AddSupplier () to store the data in database
8. It sends a response body with HTTP Success response to AddSupplier.
9. This AddSupplier returns JSON Response
10. Success JSON response and HTTP status code 200 with corresponding success message.

Supplier Updation

1. admin wants to update the details enters the supplier and the details which admin wants to update the details. browser directs the request to customer update API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to edit Supplier ().
4. It sends response body with HTTP Success response code to edit Supplier ().
5. edit Supplier returns JSON Response

6. Success JSON response and status HTTP code 200 with corresponding success message.

Supplier Deletion

1. Admin enters the email in parameter for which customer wants to delete the details. browser directs the request to customer deletion API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to remove Supplier ()
4. remove Supplier () will call the delete removeSupplierRepository to delete the data from database.
5. It sends response body with HTTP Success response code to removeDrugs ().
6. removeSupplier () returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

Drug Info:

Drugs Add

1. Admin wants to Add New Drugs the details enter the name, batch id, expiry date, supplier mail, price browser directs the request to Admin Add API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to add Drugs () this handle function
4. It sends response body with HTTP Success response code to add Drugs ().
5. add Drugs () returns JSON Response
6. Success JSON response and status HTTP code 200 with corresponding success message.

Drugs Updation

1. Admin wants to update the details enters the **Drug id** and the details which admin wants to update the details. browser directs the request to customer update API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to edit Drugs () .
4. It sends response body with HTTP Success response code to edit Drugs ().
5. edit Drugs returns JSON Response
6. Success JSON response and status HTTP code 200 with corresponding success message.

Drugs Deletion

1. Admin enters the **Drug id** in parameter for which customer wants to delete the details. browser directs the request to customer deletion API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to removeDrugs ()
4. removeDrugs () will call the delete removeDrugsRepository to delete the data from database.
5. It sends response body with HTTP Success response code to removeDrugs ().
6. removeDrugs () returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

View Drugs

1. Doctor wants to get the Drugs details. enters the Order Id in parameter for which Doctor wants to see the details. browser directs the request to Order List API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to this handle function calls the get Order ()
4. get Order () this is to fetch the data from database.
5. It sends response body with HTTP Success response code to get Order ().
6. get Order () returns JSON Response
7. Success JSON response and HTTP status code 200 with corresponding success message.

Place Order

1. Doctor wants to place the order. enters the name in parameter for which Doctor wants to see the details. browser directs the request to Order List API
2. Call reaches the API gateway.
3. API gateway does the routing and forwards the request to this handle function calls the post Order ()
4. Post User () will call the action Method to do the input validation it will have the as argument to perform the validation
 - If validation fails, then it will return the error code and error description. with status code
 - If validation is successful, then it will post the post Order () to store the data in database
5. post Order () this is to send the data from database.
6. It sends response body with HTTP Success response code to post Order ().
7. pet Order () returns JSON Response
8. Success JSON response and HTTP status code 200 with corresponding success message.

