**Reviewed by:**

**Approved by:**

**Revision / Document History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Ver** | **Date** | **Changed By** | **Modifications** |
| 1.0 | 25/07/23 | Kousil Lakkapragada |  |

**List of Abbreviations**

DFD  **D**ata **F**low **D**iagram

ER  **E**ntity **R**elationship

FHD  **F**unction **H**ierarchy **D**iagram

HLD **H**igh **L**evel **D**esign

LLD  **L**ow **L**evel **D**esign

GUI  **G**raphical **U**ser **I**nterface

IEEE  **I**nstitute of **E**lectricaland **E**lectronic **E**ngineers

S/W **S**oft**w**are

SDL **S**pecification **D**escription **L**anguage

StrD **Str**ucture**d**

Table of Contents

1. Introduction 2

2. Design Scope 2

3. Design Methodology 2

4. Design Notations 2

5. Design Considerations 2

6.Design Overview 2

7.Decomposition 7

DH-1-1 User EO 8

DH-1-3 User Service Impl 8

DH-1-4 Home Controller 8

DH-1-5 Medicine EO 8

DH-1-6 Medicine Repository 9

DH-1-7 Medicine Service 9

DH-1-8 Medicine Service Impl 9

DH-1-9 Home Controller 9

DH-1-10 Discounted Medicine EO 9

DH-1-11 Discounted Medicine Repository 10

DH-1-12 Discounted Medicine Service 10

DH-1-13 Discounted Medicine Service Impl 10

DH-1-14 Home Controller 10

DH-1-15 Report Generation EO 10

DH-1-16 Report Generation Service 11

DH-1-17 Report Generation Impl 11

DH-1-18 Home Controller 11

DH-1-14 Patient EO 11

DH-1-15 Patient Service 12

DH-1-16 Patient Service Impl 12

DH-1-17 Home Controller 12

DH-1-18 Orders EO 12

8.Interface Design 13

8.1 User Interface 13

9.Data design 13

9.2 Data structure (data types, arrays, and structures) 13

10.Reusability 13

11. Design Alternatives 13

13. Additional Hardware and Software required 13

14. Testing Strategy 13

15. Traceability Matrix 13

16. References 14

# Introduction

This Project Aims at developing a Pharmacy Management System, which needs to manage transactions based on the requirements, which includes managing the inventory in the warehouse and keeping a check on the medicines sold and medicines in the stock. This Project also includes raising Invoices to the Retailers and billing to the patients who purchase the Medicines.

This automated system should support them in properly delivering the medicines on time for the patients without making them to wait for longer period

# Design Scope

Object Oriented Analysis and Design (OOAD) methodology has been used for breaking down the specification into functionally independent units.

# Design Methodology

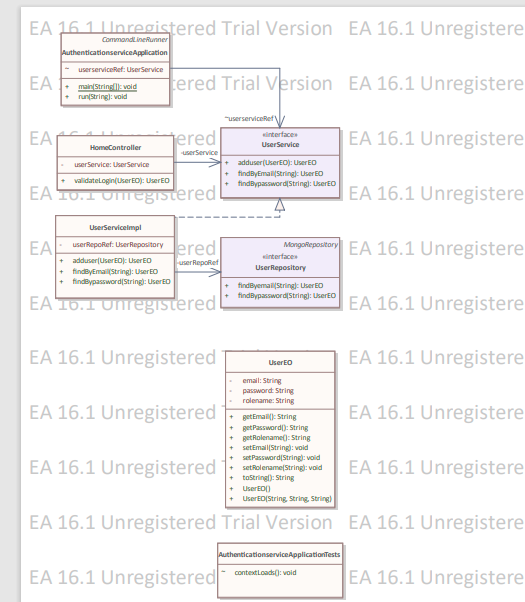
The naming conventions followed conform to Unified Modelling Language (UML) as Object Oriented Analysis and Design (OOAD) is followed.

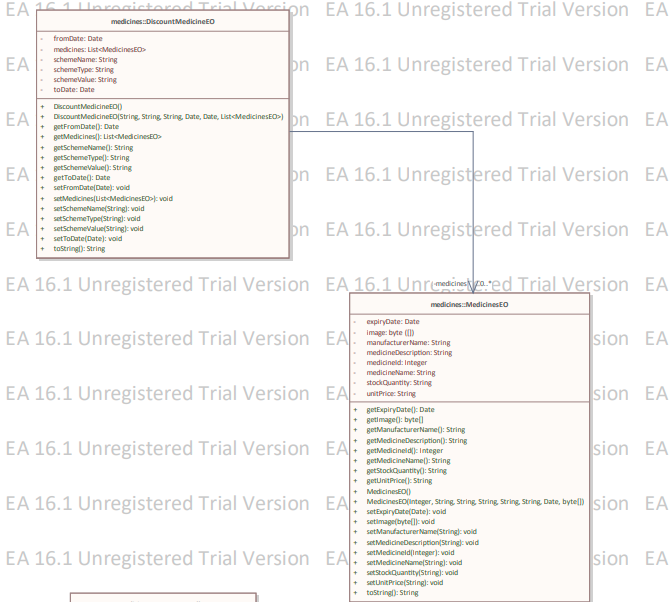
# Design Notations

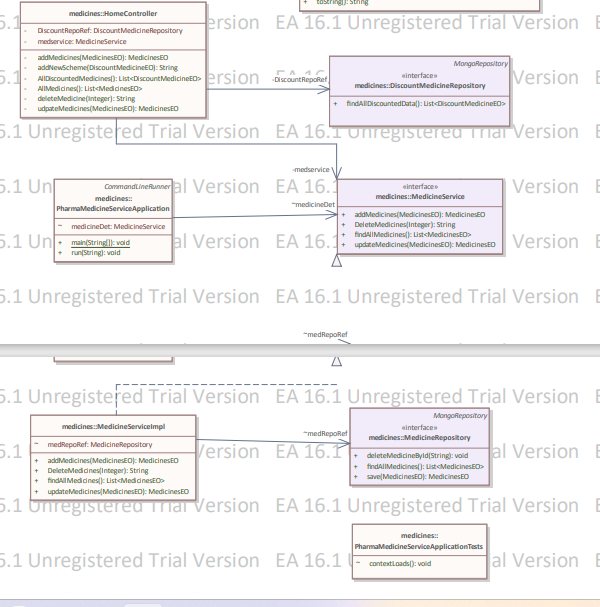
# Design Considerations

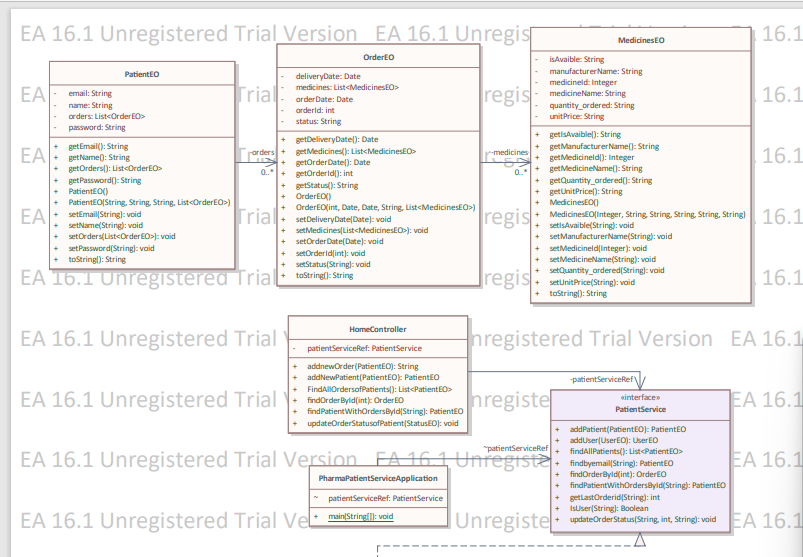
# 6. Design Overview

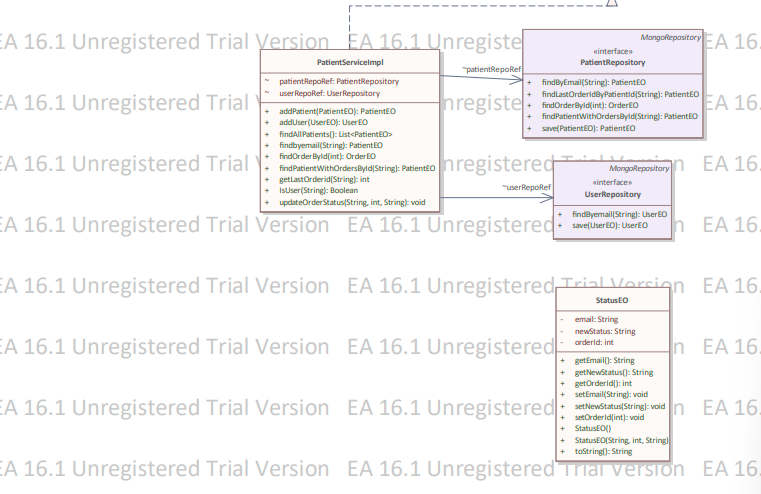
Authentication Service Design

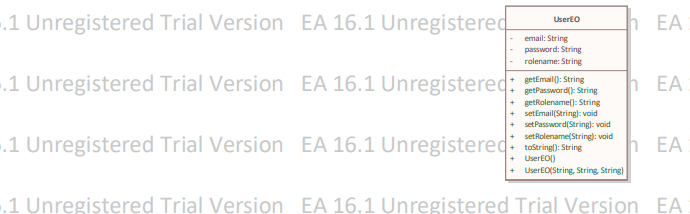


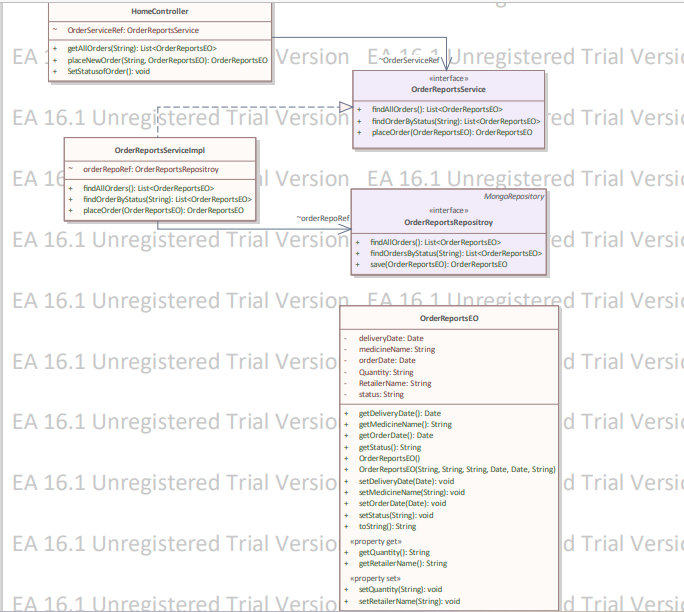












# 7. Decomposition

## **DH-1-1** **User EO**

Inputs:  Sets the user details like Username, Password in details.

Outputs: Gets the user details like Username, Role Name in details.

Scope:   Throughout the Pharmacy Management application

This class is a model class for the database Collection Users.

**DH-1-2 User Repository**

Inputs:  Giving the inputs as of name, password, user EO Object

Outputs: Get the details of Users from USERS Collection and all roles from Users collection

Scope:   Throughout the Pharmacy Management application

This interface extends the Mongo Repository< User EO, String > class.

This class is a model class for the database table USERS.

## **DH-1-3** **User Service Impl**

Inputs:  The details in the USERS be created, deleted & updated.

Outputs: Gets all users details like Username, Password & Role Name and get all role    names also perform validations

Scope:  Throughout the Pharmacy Management application

This class is a model class for the database collection USERS.

## **DH-1-4** **Home Controller**

Inputs:  Giving the data User EO Object

Output:  Get the details of username, password, role name.

Scope:  This class is used for mapping the URL with respect to admin operations.

## **DH-1-5** **Medicine EO**

Inputs:  Giving the inputs as of medicine Id, Medicine Name, Medicine Description, medicine, Manufacturer Name, Stock Quantity, Unit Price, Expiry Date.

Outputs: Get the details of medicines from Medicines collection.

Scope:  Throughout the Pharmacy Management application

## **DH-1-6** **Medicine Repository**

Inputs:  Giving the medicine Id, Medicine Name, Medicine Description, medicine, Manufacturer Name, Stock Quantity, Unit Price, Expiry Date.

Outputs:  Get the details of medicines from Medicines collection

Scope:   Throughout the Pharmacy Management application

This interface extends the Mongo Repository< Medicine EO, Integer> class.

## **DH-1-7** **Medicine Service**

Inputs:  provide username, Medicine Details.

Output:  Get Medicine object all medicine related Data

Scope:  Generic

## **DH-1-8** **Medicine Service Impl**

Inputs:  The details in the medicines be created, deleted & updated.

Outputs: Gets all medicine details like medicine Id, Medicine Name, Medicine Description, medicine, Manufacturer Name, Stock Quantity, Unit Price, Expiry Date.

Scope:  Throughout the Pharmacy Management application

This class is a model class for the database collection medicines.

## **DH-1-9** **Home Controller**

Inputs:  Giving the data medicine EO Object

Output:  Get the details of List of medicines

Scope:  This class is used for mapping the URL with respect to medicine CRUD operations.

## **DH-1-10** **Discounted** **Medicine EO**

Inputs:  Giving the inputs as of medicine Id, Medicine Name, Medicine Description, medicine, Manufacturer Name, Stock Quantity, Unit Price, Discounted Price Expiry Date.

Outputs: Get the details of medicines from Medicines collection.

Scope:  Throughout the Pharmacy Management application

## **DH-1-11 Discounted Medicine Repository**

Inputs:  Giving the medicine Id, Medicine Name, Medicine Description, medicine, Manufacturer Name, Stock Quantity, Unit Price, Discounted Price Expiry Date.

Outputs:  Get the details of medicines from Medicines collection

Scope:   Throughout the Pharmacy Management application

This interface extends the Mongo Repository< Discounted Medicine EO, Integer> class.

## **DH-1-12** **Discounted** **Medicine Service**

Inputs:  provide username, medicine Details.

Output:  Get Medicine object all medicine related Data

Scope:  Generic

## **DH-1-13 Discounted Medicine Service Impl**

Inputs:  The details in the medicines be created, deleted & updated.

Outputs: Gets all medicine details like medicine Id, Medicine Name, Medicine Description, medicine, Manufacturer Name, Stock Quantity, Unit Price, Discounted price Expiry Date.

Scope:  Throughout the Pharmacy Management application

This class is a model class for the database collection Discounted medicines.

## **DH-1-14** **Home Controller**

Inputs:  Giving the data Discounted medicines EO Object

Output:  Get the details of List of medicines

Scope:  This class is used for mapping the URL with respect to medicine CRUD operations.

## **DH-1-15** **Report Generation** **EO**

inputs: Set the details of Order Details like order Id, Medicine Name, Quantity, Date of Delivery.

Output: Get the details of order like owner order Id, Medicine Name, Quantity, Date of Delivery.

Scope: This class is used throughout the Admin Module

## **DH-1-16** **Report Generation Service**

Inputs:  provide all information about Order Details.

Output:  Get List of Order Objects and List of related Data.

Scope:  Throughout the Admin Module

## **DH-1-17 Report Generation Impl**

Inputs:  The details in the orders be created

Outputs: Gets all Order Details like order Id, Medicine Name, Quantity, Date of Delivery.

Scope:  Throughout the Admin Module

This class is a model class for the database collection Report Generation collection

## **DH-1-18 Home Controller**

Inputs:  Giving the data Status of the Order Details.

Output:  Get the details of List of orders according to the status.

Scope:  This class is used for mapping the URL with respect to Report Generation

operations.

## **DH-1-14** **Patient EO**

Inputs: Give the input of username and password.

Outputs: Gets all medicine Details from discounted Medicines and medicines collections.

Scope:  Throughout the Patient/user module in Pharmacy management application

This class is a model class for the database table Patients Collections.

## **DH-1-15** **Patient Service**

Inputs:  Provide username, and Role Name.

Output:  Get Discounted Medicines object all medicine related Data

Scope:   Throughout the Patient/user module in Pharmacy management application

## **DH-1-16** **Patient Service Impl**

Inputs:  provides all medicine details that are added to checked out.

Output: Stores the data in the Orders array in patients Eo and returns success message.

Scope:  Throughout the User module in Pharmacy management application

This class is a model class for the database collection Patients.

## **DH-1-17** **Home Controller**

Inputs:  Giving the data like, Medicines and Status of the Order Details.

Output:  Get the details of List of medicines and orders according to the status.

Scope:  This class is used for mapping the URL with respect to patient

operations.

## **DH-1-18** **Orders EO**

inputs: Set the details of Order Details like order Id, Medicine Name, Quantity, Date of Delivery.

Output: Get the details of order like user order Id, Medicine Name, Quantity, Date of Delivery.

Scope: This class is used throughout the user Module in pharmacy management System

# 8. Interface Design

## 8.1 User Interface

# 9. Data design

<The entity relationship diagram prepared for the database has to be defined here.>

## 9.2 Data structure (data types, arrays, and structures)

Not applicable.

# 10. Reusability

# 11. Design Alternatives

**12. Design Feasibility**

We have used the OOAD approach in this project. This methodology has been chosen based on our analogy of the user requirements, feasibility study and based on the experience of the co-ordinators. It has been seen that several other project groups developing similar projects have chosen the same methodology.

The OOAD assures properties such as reusability, modularity, efficiency.

# Additional Hardware and Software required

This requirement is based on the future stages of development. Therefore as of now this is not applicable

# 14. Testing Strategy

The various stages of testing to be followed for our application includes white unit and integration testing.

We will carry out all such testing in a simulated environment only.

# 15. Traceability Matrix

As per the requirements-HLD tagging shown in the document “Requirement\_Traceability.xls” each of the requirements has been mapped to the appropriate classes. Both the requirements and classes have been tagged according to the tag standards

# 16. References

List of all external sources of information referenced in this document.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sl. No.** | **Description** | **Date** | **Vers.** | **Location** |
| 1. | Software Requirements Specification Document | 29/07/2023 | 1.0 | *SRS.doc* |
| 2. |  |  |  |  |
| 3. |  |  |  |  |

Description, date, and version shall uniquely identify the information source, and the location shall specify where it is to be found.