Reviewed by:
Approved by:

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**lectrical and **E**lectronic **E**ngineers
S/W	**S**oft**w**are
SDL	**S**pecification **D**escription **L**anguage
StrD	**Str**uctur**ed**

TABLE OF CONTENTS

1. Introduction

The purpose of this document is to outline the software requirements for a digital medication management system that helps patients track their prescriptions, dosages, and reminders.. The key goals are to:

- Allow patients to track their medications, dosages, and schedules
- Send reminders and notifications for medication doses and refills
- Integrate with pharmacy systems for real-time updates on medication availability and pricing
- Provide a secure and user-friendly interface for patients to manage their medications
- Offer features for healthcare providers to monitor patient adherence and communicate with patients.

2. Design Scope

The project involves designing, developing, testing, and deploying a digital medication management system that enables patients to track their medications, receive reminders and notifications, and integrate with pharmacy systems for real-time updates, while providing a secure and user-friendly interface and allowing healthcare providers to monitor patient adherence and communicate with patients.

3. Design Methodology

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

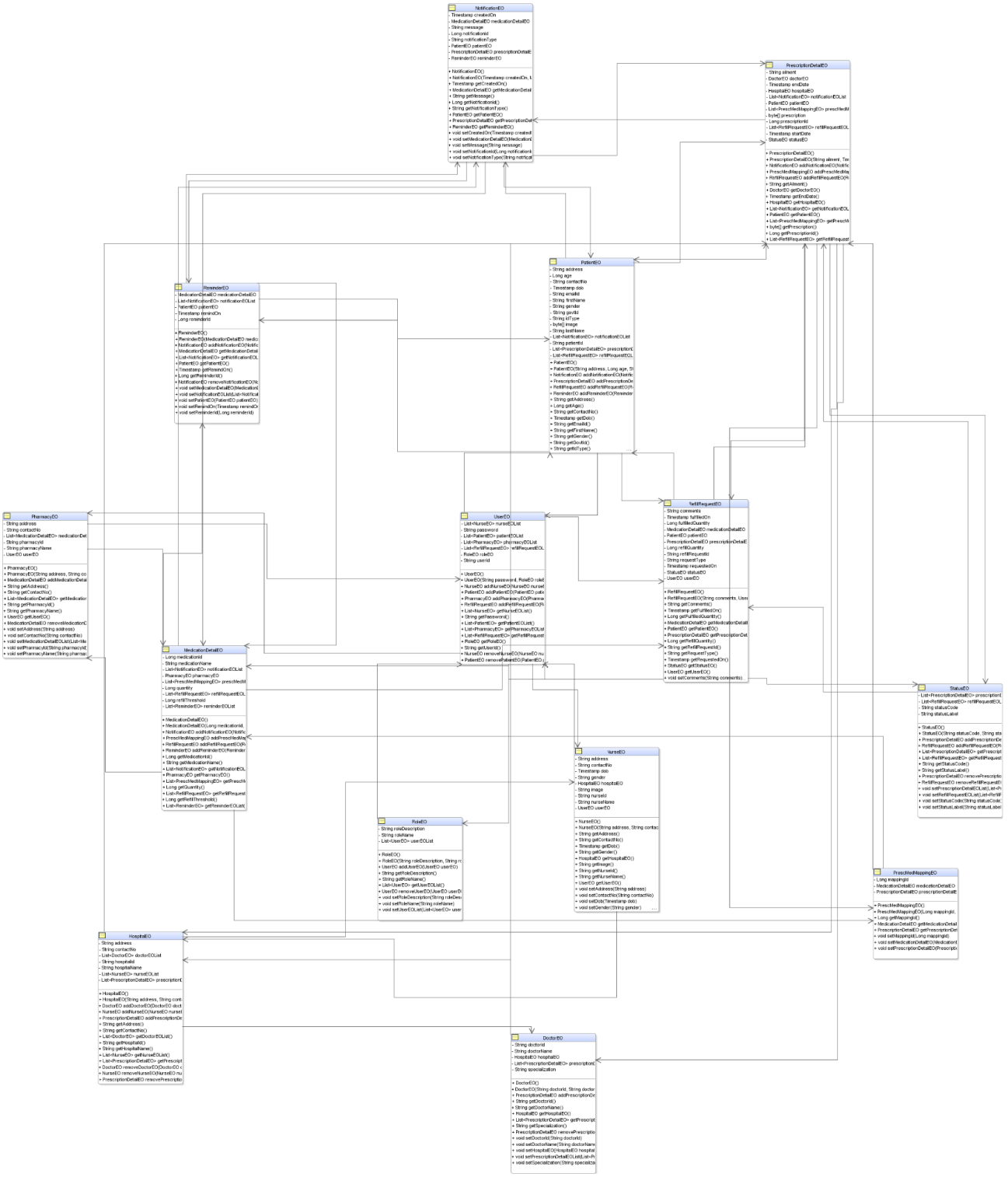
4. Design Notations

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

5. Design Considerations

Not Applicable

6. Design Overview



7. Decomposition

DH-1-1 UserEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-2 RoleEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-3 DoctorEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-4 HospitalEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-5 MedicationDetailEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-6 NotificationEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-7 NurseEO

Inputs: Sets the floor allocation details of staff.
Outputs: Gets the floor allocation details of staff.
Scope: Specific

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

DH-1-8 PatientEO

Inputs: Sets the availability details of supporting staff.
Outputs: Gets the availability details of supporting staff.
Scope: Specific

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

DH-1-9 PharmacyEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-10 PreMedMappingEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-11 PrescriptionDetailEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-12 RefillRequestEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-13 ReminderEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-14 StatusEO

Inputs: Sets the EO in persistence Layer.
Outputs: Gets the EO from Persistence Layer.
Scope: Specific

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

DH-1-15 UserRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of USERS TABLE.

DH-1-16 RoleRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of ROLESTABLE.

DH-1-17 PatientRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of PATIENTS TABLE.

DH-1-18 DoctorRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of DOCTORS TABLE.

DH-1-19 HospitalPharmacyRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides union of HOSPITALS and PHARMACIES TABLE..

DH-1-20 HospitalRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of HOSPITALSTABLE.

DH-1-21 MedicationRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of MEDICATION_DETAILS TABLE.

DH-1-22 NotificationRepossitory

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific
This class provides the JPA CRUD operations of NOTIFICATIONS TABLE.

DH-1-23 PharmacyRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of PHARMACIES TABLE.

DH-1-24 PrescMedMappingReposiory

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of PRESC_MED_MAPPINGS TABLE.

DH-1-25 PrescriptionRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of PRESCRIPTION_DETAILS TABLE.

DH-1-26 RefillRequestRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of REFILL_REQUESTS TABLE.

DH-1-27 ReminderRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific

This class provides the JPA CRUD operations of REMINDERS TABLE.

DH-1-28 NurseRepository

Inputs: Passing all fields to the database table.
Outputs: Returns the CRUD operation results.
Scope: Specific
This class provides the JPA CRUD operations of NURSES TABLE.

DH-1-29 AdminService

Inputs: NA
Outputs: Method declaration of all operations of Admin.
Scope: Specific

This is the interface of Admin.

DH-1-30 AdminServiceImpl

Inputs: Passing variables or objects as parameters.
Outputs: Fetch the data from the database by calling the CRUD Repository.
Scope: Specific

This class creates an implementation of the AdminService interface.

DH-1-31 AdminController

Inputs: Passing path variables, request params, request body as parameters.
Outputs: Calls the Service interface and returns the results to users.
Scope: Specific

This class acts as a medium between front–end user and backend for Admin operations.

DH-1-32 PatientService

Inputs: NA
Outputs: Method declaration of all operations of Patient..
Scope: Specific

This is the interface of Patient..

DH-1-33 PatientServiceImpl

Inputs: Passing variables or objects as parameters.
Outputs: Fetch the data from the database by calling the CRUD Repository.
Scope: Specific

This class creates an implementation of the PatientService interface.

DH-1-34 PatientController

Inputs: Passing path variables, request params, request body as parameters.
Outputs: Calls the Service interface and returns the results to users.
Scope: Specific

This class acts as a medium between front–end user and backend for Patient operations.

DH-1-35 PharmacyService

Inputs: NA
Outputs: Method declaration of all operations of Pharmacy..
Scope: Specific

This is the interface of Pharmacy..

DH-1-36 PharmacyServiceImpl

Inputs: Passing variables or objects as parameters.
Outputs: Fetch the data from the database by calling the CRUD Repository.
Scope: Specific

This class creates an implementation of the PharmacyService interface.

DH-1-37 PharmacyController

Inputs: Passing path variables, request params, request body as parameters.
Outputs: Calls the Service interface and returns the results to users.
Scope: Specific

This class acts as a medium between front–end user and backend for Pharmacy operations.

DH-1-38 ProviderService

Inputs: NA
Outputs: Method declaration of all operations of Provider..
Scope: Specific

This is the interface of Provider..

DH-1-39 ProviderServiceImpl

Inputs: Passing variables or objects as parameters.
Outputs: Fetch the data from the database by calling the CRUD Repository.
Scope: Specific

This class creates an implementation of the ProviderService interface.

DH-1-40 ProviderController

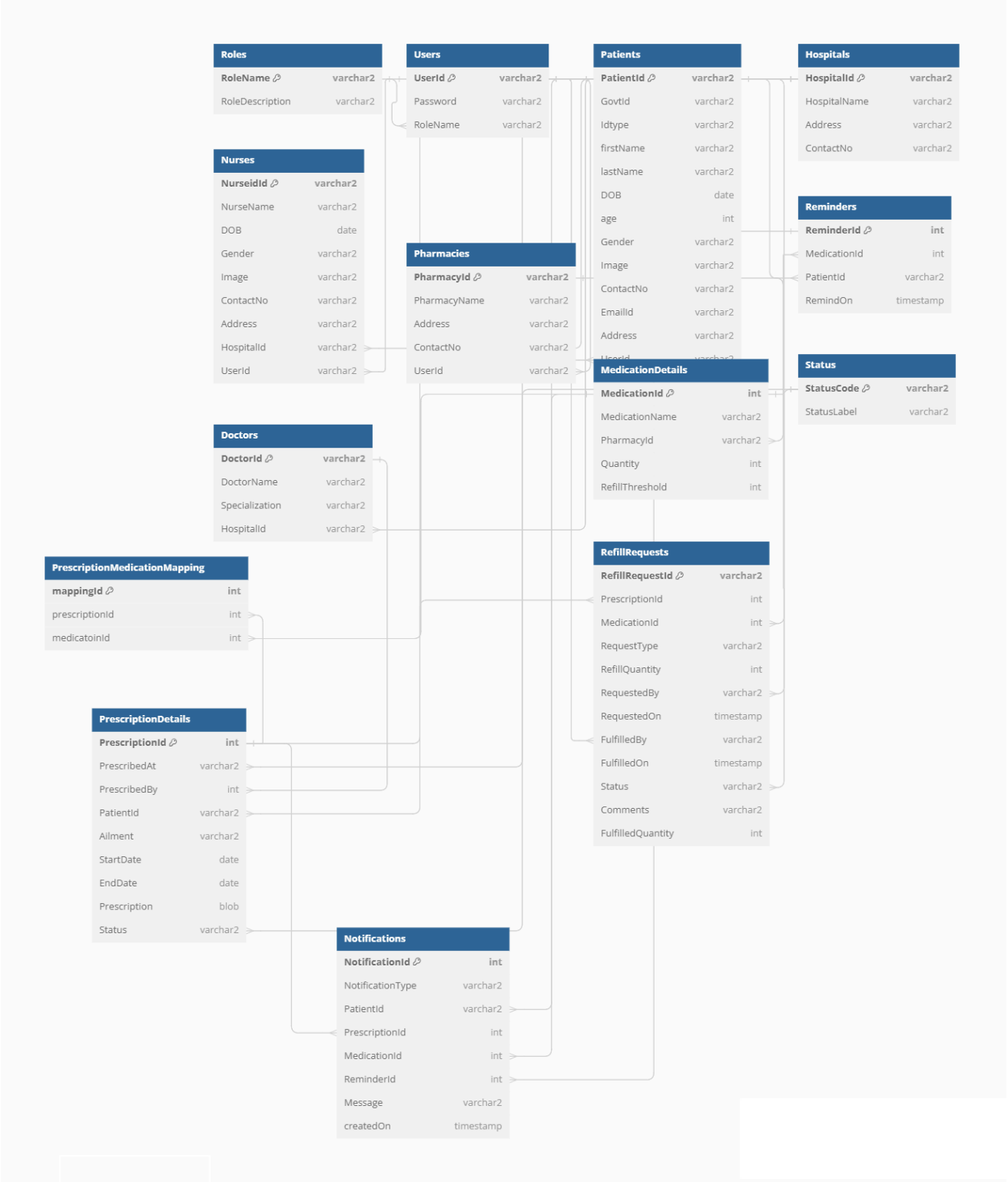
Inputs: Passing path variables, request params, request body as parameters.
Outputs: Calls the Service interface and returns the results to users.
Scope: Specific

This class acts as a medium between front–end user and backend for Provider operations.

8. Interface Design

8.1 User Interface

9. Data design



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

Not applicable.

10. Reusability

- Custom generator

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 coordinators. 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.

13. 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	12/07/2024	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.