PART III: SOFTWARE DESIGN DOCUMENT

Table of Contents

[1. INTRODUCTION 2](#_Toc523574842)

[1.1 Purpose 2](#_Toc523574843)

[1.2 Scope 2](#_Toc523574844)

[1.3 Overview 3](#_Toc523574845)

[1.4 Reference 3](#_Toc523574846)

[1.5 Definitions and Acronyms 3](#_Toc523574847)

[2. SYSTEM OVERVIEW 4](#_Toc523574848)

[2.1. Deployment Diagram 4](#_Toc523574849)

[3. SYSTEM ARCHITECTURE 6](#_Toc523574850)

[3.1. Architectural Design 6](#_Toc523574851)

[3.2. Decomposition Description 8](#_Toc523574852)

[3.3. Design Rationale 8](#_Toc523574853)

[4. DATA DESIGN 8](#_Toc523574854)

[4.1. Data Structure Description 8](#_Toc523574855)

[A no Sql database(Mongo Db) was used to organize the data into five collections. These collections are used to store all data used in the ORMS. They are user, patient records, staff, community and students. 8](#_Toc523574856)

[4.2. Data Dictionary 8](#_Toc523574857)

[**4.2.1 User Collection** 8](#_Toc523574858)

[**4.2.2 Students Collection** 8](#_Toc523574859)

[**4.2.3 Community Collection** 9](#_Toc523574860)

[**4.2.4 Staff Collection** 9](#_Toc523574861)

[**4.2.5 Patient Records Collection** 10](#_Toc523574862)

[5. USER INTERFACE DESIGN 10](#_Toc523574863)

[5.1. Overview of the User Interface 10](#_Toc523574864)

[**5.1.1 Doctor interface** 11](#_Toc523574865)

[**5.1.2 Receptionist interface** 11](#_Toc523574866)

[**5.1.3 Physician interface** 11](#_Toc523574867)

[**5.1.4 Lab Technician interface** 11](#_Toc523574868)

[**5.1.5 Pharmacist interface** 11](#_Toc523574869)

[5.2. Screen Images 12](#_Toc523574870)

[6. Help System Structure 12](#_Toc523574871)

[7. REQUIREMENTS MATRIX 12](#_Toc523574872)

[8. Use Case Realizations 12](#_Toc523574873)

[8.1a Create Patient –Receptionist 12](#_Toc523574874)

[8.1b Update Patient Record –Physician 13](#_Toc523574875)

[8.1c Update Patient Record –Doctor 14](#_Toc523574876)

[8.1d Issue Drugs – Pharmacy 15](#_Toc523574877)

# 1. INTRODUCTION

## 1.1 Purpose

This document contains the complete design description of the Outpatient Records Management System. This includes the architectural features of the system down through details of what operations each code module will perform and the database layout. It also shows how the use cases detailed in the SRS will be implemented in the system using this design.

The primary audiences of this document are the software developers.

This document can also be useful to the users, for the sake of understanding the software, the stakeholders of the project and anyone else who would be interested in knowing how the ORM system works.

## 1.2 Scope

## 1.3 Overview

* Chapter 2 is a Deployment Diagram that shows the physical nodes on which the system resides. This allows a clear explanation of where each design entity will reside. No design unit may straddle two nodes but must have components on each, which collaborate to accomplish the service.
* Chapter 3 is the Architectural Design. This is the heart of the document. It specifies the design entities that collaborate to perform the functionality of the system. Each of these entities has an Abstract Specification and an Interface that expresses the services that it provides to the rest of the system. In turn each design entity is expanded into a set of lower-level design units that collaborate to perform its services.
* Chapter 4 is the basic Data Structure Design, which for this project is a relational database. While it is separated out here for emphasis, it is really the lowest level of the Architectural Design.
* Chapter 5 is on User Interface Design and discusses the methodology chosen, why it was chosen and why it is expected to be effective.
* Chapter 6 describes the structure of the Help System.
* Chapter 7 exhibits the Use Case Realizations. The implementation of each use case identified in the SRS is shown using the services provided by the design objects.

## 1.4 Reference

## 1.5 Definitions and Acronyms

|  |  |
| --- | --- |
| ***Term*** | ***Meaning*** |
| ORM System | Outpatient Records Management System |
| SRS | Software Requirements Specification |
| Deployment diagram | The diagram that displays the functionality of the system with the interconnections between system users. |
| SQL | Structured Query Language |
|  |  |

# 2. SYSTEM OVERVIEW

## 2.1. Deployment Diagram

Web Browser

Hospital Server

Internet

Health Care Staff Processor

Intranet

Database Server

Intranet

Figure - Deployment Diagram

A Health Care staff, say doctor, accesses the Outpatient Records Management System through the Internet using a Web browser (not part of this system although Web pages will run on it). The System resides on a dedicated Hospital server with a permanent Web connection. The Doctor manages all of the record update work on his/her personal computer (Health Care staff processor), communicating with the local database on the Database Server when needed, and uploading completed updates to server when they are ready. Every health care staff Processor contains a local file system.

# 3. SYSTEM ARCHITECTURE

## 3.1. Architectural Design

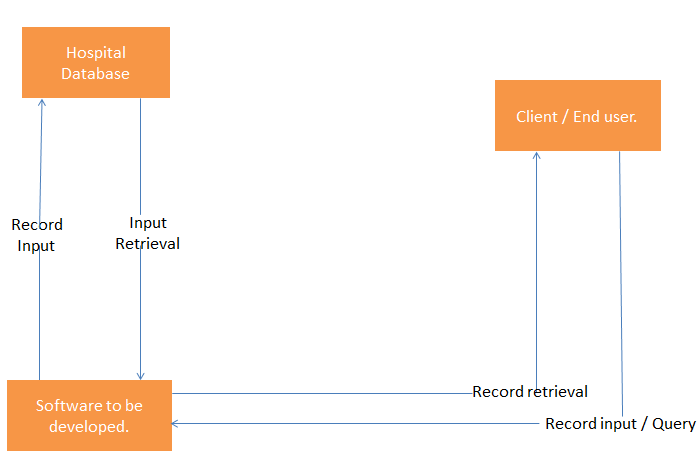


Figure 2: simplified unit design.

## 3.2. Decomposition Description

## 3.3. Design Rationale

# 4. DATA DESIGN

## 4.1. Data Structure Description

We used a NoSql database(Mongo Db) was used to organize the data into five collections. These collections are used to store all data used in the ORMS. They are user, patient records, staff, community and students.

## 

## 4.2. Data Dictionary

A data dictionary is defined as a set of information describing the contents, format, and structure of a database and the relationship between its elements, used to control access to and manipulation of the database.

In Mongo db we used document-oriented database with the following collections.

### **4.2.1 User Collection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Data Format** | **Description** | **Example** |
| userId | String | NNNNNN | Specific identification for | 123425 |
| Name | String | TTTT TTTTT | User’s name | Michael Makali |
| password | String | ntntntnnn | User’s login password | \*\*\*\*\*\*\* |
| staffId | String | NNNNNN | User’s job identification number | 123456 |
| Email | String | TTTTTTT | User’s email |  |
| Role | String | TTTTTTT | User’s role | doctor |
| gender | String | TTTTTTT | User’s gender | female |

### **4.2.2 Students Collection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Data Format** | **Description** | **Example** |
| regNO | String | Ttnn/nnnn/nn or tnn/nnnn/nn | Student’s registration number | S13/09664/15 |
| Name | String | Name name | Student’s name | Michael Makali |
| patientId | String |  | Student’s patient identification number | 123456 |
| gender | String | Tttttttt | student’s gender | female |

### **4.2.3 Community Collection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Data Format** | **Description** | **Example** |
| Id | String | nnnnnnnn | Community member national identification number | 33470928 |
| Name | String | Name name | Community member’s name | Michael Makali |
| patientId | String |  | Community member’s patient identification number | 123456 |
| gender | String | Tttttt | Community member’s gender | female |

### **4.2.4 Staff Collection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Field Name** | **Data Type** | **Data Format** | **Description** | **Example** |
| staffId | String | nnnnn | Staff identification number | 37864 |
| Name | String | Name name | Staff’s name | Michael Makali |
| patientId | String |  | Staff’s patient identification number | 123456 |
| gender | String |  | Staff’s gender | female |

### **4.2.5 Patient Records Collection**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Field Name | Data Type | Format | Description | Example |
| PatientId | String |  | Patient’s identification |  |
| Height | Integer | NN | Patient’s height | 46 |
| Age | Integer | NN | Patient’s age | 22 |
| Weight | Integer | NN | Patient’s weight | 76 |
| Doctor | String | TTTT TTTT | The doctor on duty | Mercy Mokeira |
| Prognosis | String | Long Text | The doctor’s prognosis | There is a need to take these tests. |
| Diagnosis | String | Long Text | The outcome conclusion from the doctor | You suffer from malaria |
| Tests | String | Long Text | The tests taken | Little on no virus |
| Prescription | String | Long Text | The drugs recommendation | Get malaraquine |
| Date | Date | Dd/mm/yyyy | The date of treatment | 13/01/2019 |
| Bmi | Integer |  | Body mass index |  |
| Temperature | Integer | NN | The temperature of the patient. | 37 |

# 5. USER INTERFACE DESIGN

## 5.1. Overview of the User Interface

A user interface is a screen which permits the user to interact with the system.

I n the case of ORMS, the user interfaces are developed with official setup of an hospital system in minds of the developers. Every system user’s home page will be a Login Screen which will require the users to access their pages using their credentials. The Login page will have two text fields which will require the user to enter their staff id and their respective passwords so as to login.

### **5.1.1 Doctor interface**

The doctors’ dashboard after login, will comprise of a list of patients on queue, waiting for the doctor’s attention. There will also be three text areas for; recording prognosis, recording diagnosis and recording prescriptions respectively. There will also be a submit button which the doctor will press in order to update the data into the database. The doctor will be able to view all records of the patient uploaded into the database.

### **5.1.2 Receptionist interface**

After login, the receptionist will be able to capture data for the staff, student or community members using his/her interface. The interface will comprise of Fields requiring the personal details of the aforementioned people. These details comprise of national Identification number in the case of community members, staff identification number in the case of members of the staff and student Registration numbers in the case of students.

### **5.1.3 Physician interface**

After login, the physician will be able to update data for patients using his/her interface. The interface will comprise of Fields requiring the specific patient vitals. These comprise of height, name, weight, age, blood pressure if tested, and temperature. There will be a submit button at the end so that the physician can finally update the patient’s records after their encounter.

### **5.1.4 Lab Technician interface**

After login, the Lab Technician will be able to view the patients that are on the queue requiring tests. He/she will therefore issue the tests manually and later on record the test results using his/her interface. The interface will comprise of Fields requiring him/her to capture the test results. There will be a submit button at the end so that the physician can finally update the patient’s records after their encounter.

### **5.1.5 Pharmacist interface**

After Login, the pharmacist will be in a position to view the doctors’ prescriptions and administer the requested drugs. He/she will thereafter submit the data into the database to ensure track of those patients that have undergone a complete cycle of treatment.

## 5.2. Screen Images

# 6. Help System Structure

# 7. REQUIREMENTS MATRIX

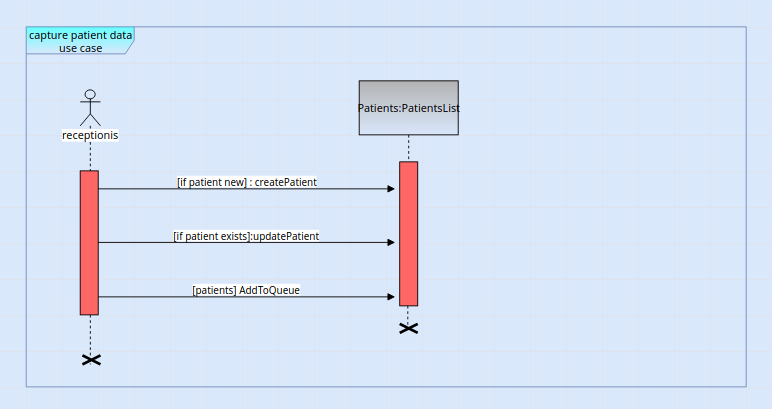
# 8. Use Case Realizations

These section, just as the name suggests, contains diagrams showing the modelling of the use cases in the order of updating patient’s data by different users in the cycle of maintaining records management system.

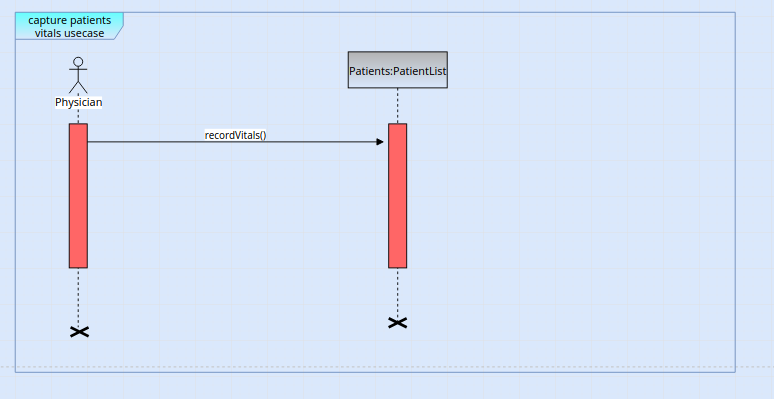
It captures the whole cycle from the point of the patient at the reception until they are treated, issue d drugs and released.

This section is meant to help developers manage the flow of activities and order of patients’ treatment.

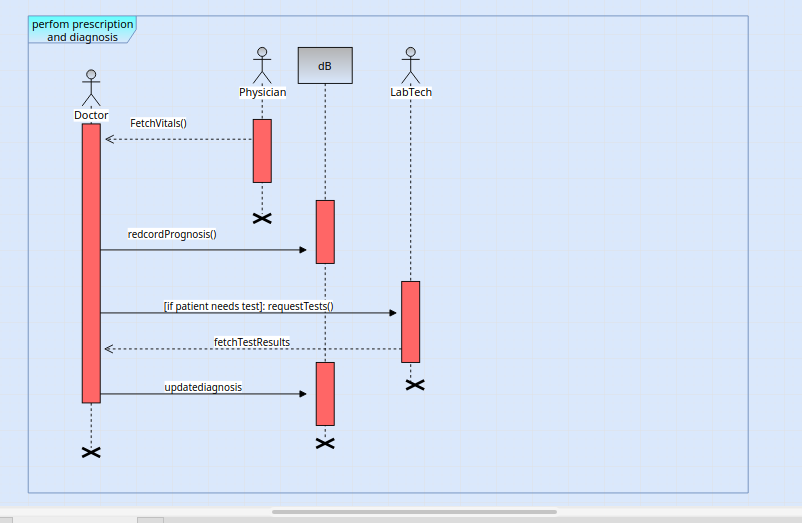
## 8.1a Create Patient –Receptionist



## 8.1b Update Patient Record –Physician



## 8.1c Update Patient Record –Doctor



## 8.1d Issue Drugs – Pharmacy

