Egerton University

Faculty of Science

Department of Computer Science

Comp 362: Object Oriented Systems

Software Requirements Specification

A Healthcare Outpatient Records Management System

S13/20757/13 Innocent Omido

S13/09710/15 Kijedi Wesley

S13/09664/15 Christine Peter

S13/09709/15 Michael Makali

S13/09/15 Erick Ogaroh

Table of contents

[**System Request**](#_2nurtpdi1xrd) **4**

[**1. Introduction**](#_87scma49fh21) **6**

[1.1 Purpose of the document](#_3hhuab90n0xk) 6

[1.2 Scope of Document](#_vstykhgz3fvj) 6

[1.3 Overview of document](#_1492q3v4rieu) 7

[**2. Description of Project**](#_x3lmi6gj1sbe) **8**

[2.1 Project Overview](#_317890s3fmuq) 8

[2.2 Project Functions](#_x1q2sz436itv) 8

[2.3 User Characteristics](#_eo510i30pwwn) 9

[2.4 Constraints to Project Development and Implementation](#_3coy483g5mbx) 10

[2.5 Assumptions and Dependencies](#_xlozdogidchq) 10

[**Specific Requirements of Healthcare Records System**](#_oqjrkardx1lv) **12**

[**3. External interface Requirements**](#_b028c8b4hf6k) **12**

[3.1 User interfaces](#_uwrggakyi95j) 12

[3.2 hardware interfaces](#_xkpzyll3st99) 12

[3.3 software interfaces](#_cf64szpjsbpj) 12

[3.4 Communications interfaces](#_r8r82c56515k) 13

[**4. Functional Requirements of Hospital Digital Record System**](#_gjzgvg9bulwc) **14**

[**5 Non- Functional Requirements of Hospital Digital Record System**](#_gomkmm24kk9p) **15**

[5.1 Performance Requirements](#_8w3vm78qrrs) 15

[5.2 Safety Requirements](#_qi11zbp779oa) 15

[5.3 Security Requirements](#_wq463orrr9x4) 15

[5.4 Hospital Digital Record System Performance](#_akcaz6mbiv4l) 16

[5.5 Software Quality Attributes](#_wjyg81vfehim) 18

[**6. Functional Modelling**](#_np6kzlhbkrsa) **19**

[6.1 Activity Diagrams](#_x4lxerfnv4up) 19

[6.2 Use Cases and use case description](#_mtkklvly9ckg) 22

[**7. Structural Modelling**](#_cmfuhegg8jhq) **28**

[7.1 Crc Cards Used:](#_g4xswdca2lc4) 28

[7.2 Class Diagrams](#_w1okeuhuflcy) 29

[**8. Behavioural Modelling**](#_sj4yu7e22vuu) **30**

[8.1 Sequence Diagrams](#_xydvn72f4slg) 30

[8.2 Behavioral State Machine Diagram](#_jzr4877cr6ob) 32

[**9. Design modelling**](#_z8vysdwyujek) **33**

# System Request

A Records Management System

***Project Sponsor***: Egerton University, Medical Department

***Business need***: This project has been initialized to provide for on demand computerized access to and storage of patients’ medical records.

***Business Requirements***: Using this system, the medical staff should be able to create, update and store patient medical data from their desktop clients. They should be able to store patient records and manage the daily in-flow of patients into the premises. The system should have the following functionality:

* Add a new patient into the hospital records
* Update patient records at the hospital
* Queue patients at the hospital as they come in using the system.
* Record and fetch patient records.
* Archiving and backing up of patient records

***Business Value***: We expect that the cost of acquisition of manual filing material will reduce. Therefore, operational costs are meant to be on the decline after the break-even point. We also expect that customer service will improve due to implementation of the electronic queuing built into the system. The implementation of the system would bring to the department benefits such as:

* Reduced operational costs.
* Space saving due to electronic archiving of patient records.
* Increase in patient satisfaction due to increased system efficiency.
* Easy retrieval of patients’ medical data since the storage is centralized and up-to-date.

***Special issues and Constraints***: The Medical Department views this as a strategic system. This Record Management system will add value to our current business model, and it will also serve as a proof-of-concept for future technical endeavors.

# 1. Introduction

## 1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the “Outpatient Records Management System (ORMS) software. It will illustrate the purpose and complete declaration for the development of system. It will also explain system constraints, interface and interactions with other external applications. This document is primarily intended to be proposed to the medical department for its approval and a reference for developing the first version of the system for the development team.

## 1.2 Scope

The document will identify the software to the developed and the underlying technology to be used during the implementation of the project. This includes the host database management system, the java supported client module and the inputs and outputs of the system. The SRS will show that the use of a secure socket layer in the protocol stack we hope to address the valid security concerns about the networking and transmission of confidential healthcare information.

By completion of this, this document will make clear the design team’s goals of creating value-adding software which not only correctly captures patient health information, but then efficiently stores it, sorts it, retrieves it, and delivers this critical care information where it is needed by healthcare professionals. The benefit of having accurate, complete, and timely health information is that it will inevitably save human lives.

This software is deliberately focused on medical records and the associated diagnostics. It is important to point out that this system which is life critical will not have cross functionality regarding appointment management, billing, or insurance functions, however diagnostic codes sets will be compliant with present legal addresses.

## 1.3 Overview

The Software Requirement Specification will clearly define and illustrate the overall project, its specifications and its requirements- both functional and nonfunctional. In addition the SRS will define the users and their respective characteristics as well as any limitations to development that the team has identified.

The format of the SRS document will address the overall project first- including functions and objectives in an overview. This section will also address how this software interfaces with other legacy systems. Then the subsequent sections will specifically address the components of the larger software system. These sections delineate specifications for every facet of the components design.

# 2. Overall Description

This section will give an overview of the whole system. The system will be explained in its context to show how the system interacts with other systems and introduce the basic functionality of it. It will also describe what type of stakeholders that will use the system and what functionality is available for each type. At last, the constraints and assumptions for the system will be presented.

## 2.1 Product Overview

The System will be a web portal which will require the users to log in, so as to perform any task using it. The first entry part of interacting with the system will be by the receptionist who will be capturing the patient’s data. If the patient is a new patient, the receptionist will open a new record for that patient. If the patient has visited the hospital before and their records have been captured, the receptionist will be required to update the record and add that patient to the queue of that day.

The second person who interacts with the system will be the physician. His/her work will be to record the patient’s vitals; including weight, height, blood pressure, temperature and other related vitals.

The updated data by the physician will be viewable from the Doctor’s endpoint. He/she

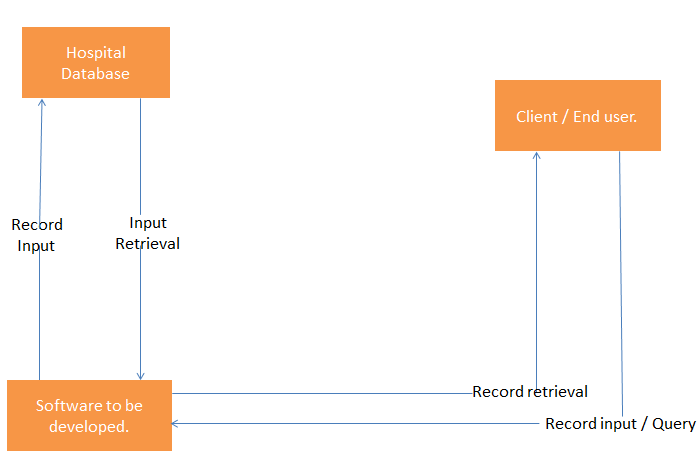
Medical records are an essential part of the medical field. However, these records are not being fully utilized in the case study provided. Often the records are misfiled, inaccurate or duplicated unnecessarily. Recognition of the improvement of data digitization and networking as a constructive force often increases efficiency while lowering costs; it is our view that medical records networking will increase and foster the quality of healthcare offered. The information system, which is a primary link between the healthcare facility, physician and the patient, would be able to capture data, store it and retrieve it efficiently.

This information system is an industry-compliant application, based upon an open architecture, and is designed to function within a standard IEEE compliant Ethernet (10 or 100) Local or Wide Area Network environment. The communications protocol is TCP/IP, and is supported under any routing protocol within an infrastructure (routed or bridged).

The software is based upon standard and emerging web technologies, requiring a workstation to only be capable of running a local desktop client software.

## 2.2 Project Functions

* The software code should be portable between different operating systems such as Linux and Windows.
* The software should be easy to use and should require minimum manual operation.
* The software should have a user-familiar interface so that the system would not pose an additional workload to the users.
  + Note. Interface design would follow generally accepted model conventions for placement of dropdown menus and toolbars.
* The software should allow bidirectional synchronous communication between the user and the data source in real time.
* The software should provide security of operation and confidentiality of information (restricting access to non-privileged users), by FAT32 compression of data and Rijndael (AES) encryption algorithms.
* The software should be able to interface and link all components of system.



## 2.3 User Characteristics

* The primary user will be a healthcare professional like a physician, a nurse, or an emergency medical technician.
* Note. This is a Medical Information System therefore to limit access and ensure integrity of the data only licensed medical personnel have access to input, search, and update functions.
* Nurse Administrators, Physician Office Administrators, System Administrators and/or Therapists will have limited access and information capabilities.
* Note: For the reasons clearly stated in 2.3.1 the System Administrator (or Vendor) will only be able to access data with his Admin access code in combination with the Physician’s code while in the physician’s presence.

## 2.4 Constraints to Project Development and Implementation

* The Health Insurance Portability and Accountability has mandated various standards on security, privacy, transaction and code sets, and unique healthcare identifiers to which this system must adhere.
* Legacy systems in place must be considered and modified to interface with the new system design.
* Both the hospital and physician database will need large storage capabilities and a process to archive outdated data.

(Note. Method and size of Database storage TBD)

## 2.5 Assumptions and Dependencies

* The system relies on a Physician relationship with a hospital system with which he/she is a staff member.
* The SDLC chosen to implement the system will be model driven and based on subsequent versions to insure data integrity and functionality.
* Due to report length constraints imposed by Health regulations will be strictly followed but kept as a stand alone document

# Specific Requirements of Healthcare Records System

# 3. External interface Requirements

This section provides a detailed description of all inputs into and outputs from the system. It also gives a description of the hardware, software and communication interfaces and provides basic prototypes of the

user interface.

## 3.1 User interfaces

Front-end software: javafx

Back-end software: SQL

## 3.2 hardware interfaces

Windows Desktop client.

Ability to run Java desktop apps

## 3.3 software interfaces

Following are the software used for the records management system.

|  |  |
| --- | --- |
| Software used | Description |
| Operating system | We have chosen Windows operating system for its best support and user-friendliness. |
| Database | To save the patient records we have chosen SQL database. |
| Java | To implement the project we have chosen Java language for its more interactive support. |

## 3.4 Communications interfaces

The communication between the different parts of the system is important since they depend on each other. However, in what way the communication is achieved is not important for the system and is therefore handled by the underlying operating systems.

## 

# 4. Functional Requirements of Hospital Digital Record System

* The software must allow input of patient data from the patient and the Physician.
* The software must request username and password for access to data, only after authentication the system will allow access.
* The software must require high levels of error correction and input validation.

*Note. Message box prompts would require a second entry of key data fields including name, DOB, Social Security Number, medications and allergies. Doctor’s inputs will similarly prompt proper code sets for diagnosis.*

* The software must allow browsing by the physician of historical medical information of his/her patients only.
* The software must identify the patient by a unique numeric identifier derived from a function performed on the patient’s birth date.

*Note. Algorithm will be simply TODAY-BIRTHDAY = NUM & Doctor Key & Increment*

*(Increment will be added if duplicate number found in Database.)*

* The software must retrieve, update, and store data from multiple input locations including but not limited to hospital workstations.
* The software must only allow deletions by the vendor and only after archiving data.
* The software to be developed must display the correct patient name.
* The software to be developed shall display the correct time of day in compliance with ISO certification.
* The software to be developed must operate twenty-four hours a day.
* The software must retrieve and sort medical record information and allow for screen and print output of said information.

*Note. Print output will include name, DOB, and requested diagnostic information only.*

# 5 Non- Functional Requirements of Hospital Digital Record System

## 5.1 Performance Requirements

1. ***Normalization***

The basic objective of normalization is to reduce redundancy which means that information is to be stored only once. Storing information several times leads to wastage of storage space and increase in the total size of the data stored.

If a database is not properly designed it can give rise to modification anomalies. Modification anomalies arise when data is added to, changed or deleted from a database table. Similarly, in traditional databases as well as improperly designed relational databases, data redundancy can be a problem. These can be eliminated by normalizing a database.

Normalization is the process of breaking down a table into smaller tables. So that each table deals with a single theme. There are three different kinds of modifications of anomalies and formulated the first, second and third normal forms (3NF) is considered sufficient for most practical purposes. It should be considered only after a thorough analysis and complete understanding of its implications.

## 5.2 Safety Requirements

If there is extensive damage to a wide portion of the database due to catastrophic failure, such as a disk crash, the recovery method restores a past copy of the database that was backed up to archival storage (typically tape) and reconstructs a more current state by reapplying or redoing the operations of committed transactions from the backed up log, up to the time of failure.

## 5.3 Security Requirements

Security systems need database storage just like many other applications. However, the special requirements of the security market mean that vendors must choose their database partner carefully.

* The software interface must follow design conventions which allow for familiar location of drop down menus, help etc.
* Input errors will be returned in red with appropriate message box.
* More than three attempts at login and failure will produce a red flag to system administrator.

## 5.4 Hospital Digital Record System Performance

* The Hospital software should be able to support up to 5 simultaneous users.
* The Hospital software should support an SQL server database.
* 95% of the transactions shall be processed in less than one second.
* Data should be secured and backed up every quarter hour.
* Power supply should have a back up and a disaster recovery plan.
* System should be operable 24 hours a day and accessible in real-time.

In order to use the system

The User:

*Should be logged in to the desktop client*

*Scenario: Successful log-in*

*Given the user wants to log in*

*When the user logs in with his/her account*

*Then the user should be logged in as a restaurant owner*

*Scenario: Retrieve password*

*Given the user wants to log in*

*And has lost the password*

*When the Admin enters the user’s identification number in the “Retrieve password” form*

*And submits the form.*

*The password is then reset*

Feature: Administrator log in

*In order to administer the system*

*An administrator*

*Should be logged in to the web-portal*

*Scenario: Successful log-in*

*Given the administrator wants to log in*

*When the administrator logs in with an administrator account*

*Then the administrator should be logged in as an administrator*

## 5.5 Software Quality Attributes

***Availability***: The system should be available for use at all times.

***Correctness***: The system should contain correct patient data as keyed in the system.

***Maintainability***: The system records should maintain correct patient records as keyed in the system.

***Usability***: The patient records should satisfy a maximum number of customers needs.

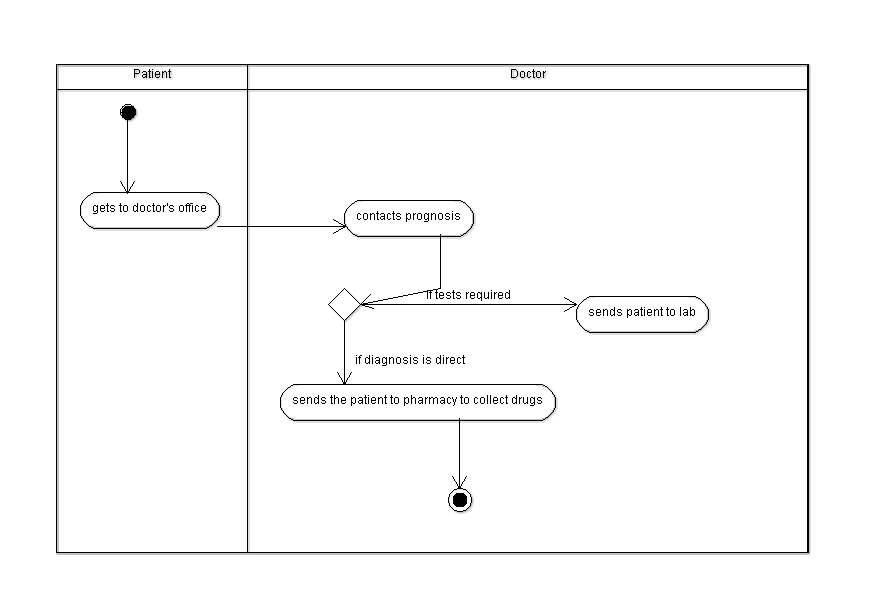
# 6. Functional Modelling

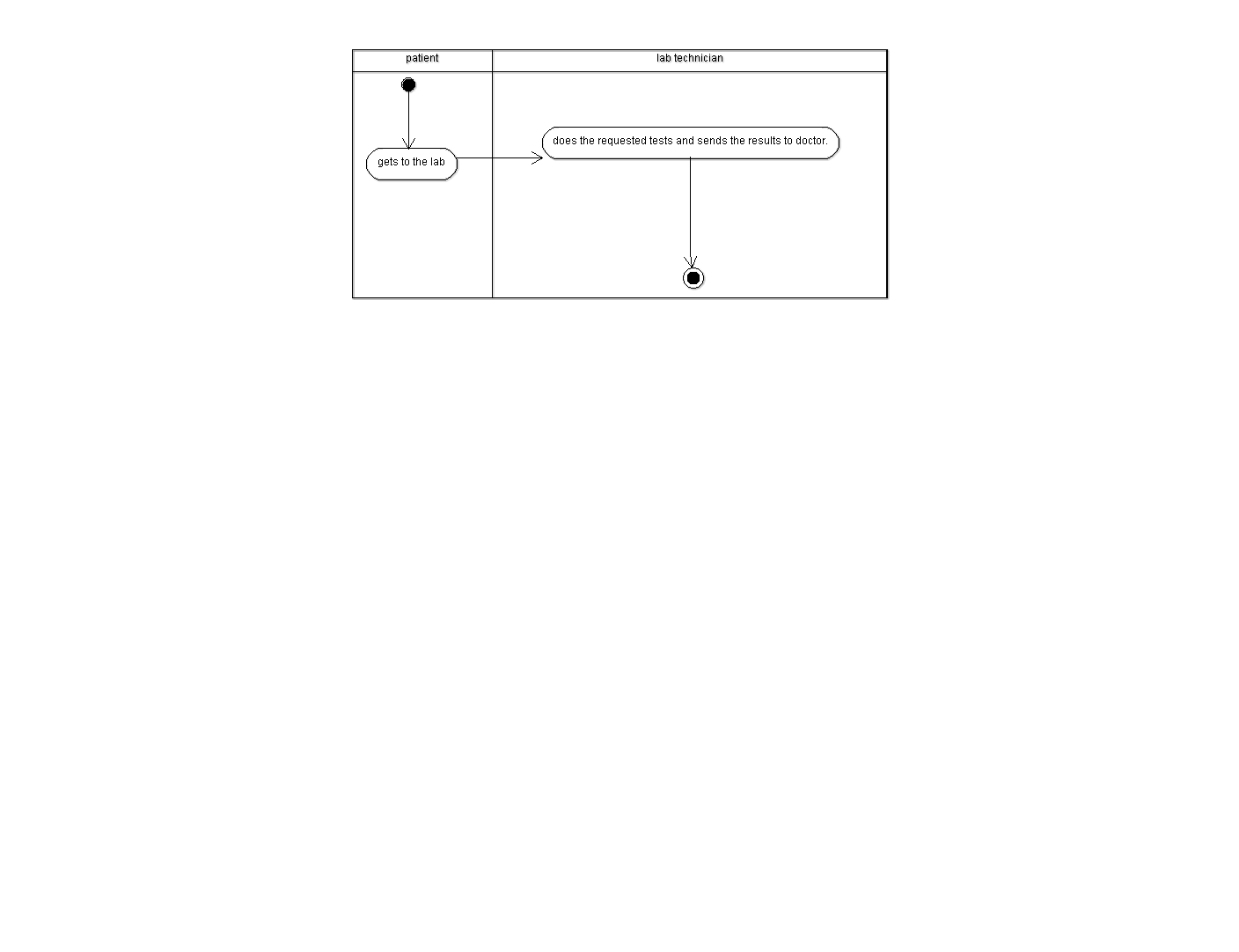
## 6.1 Activity Diagrams

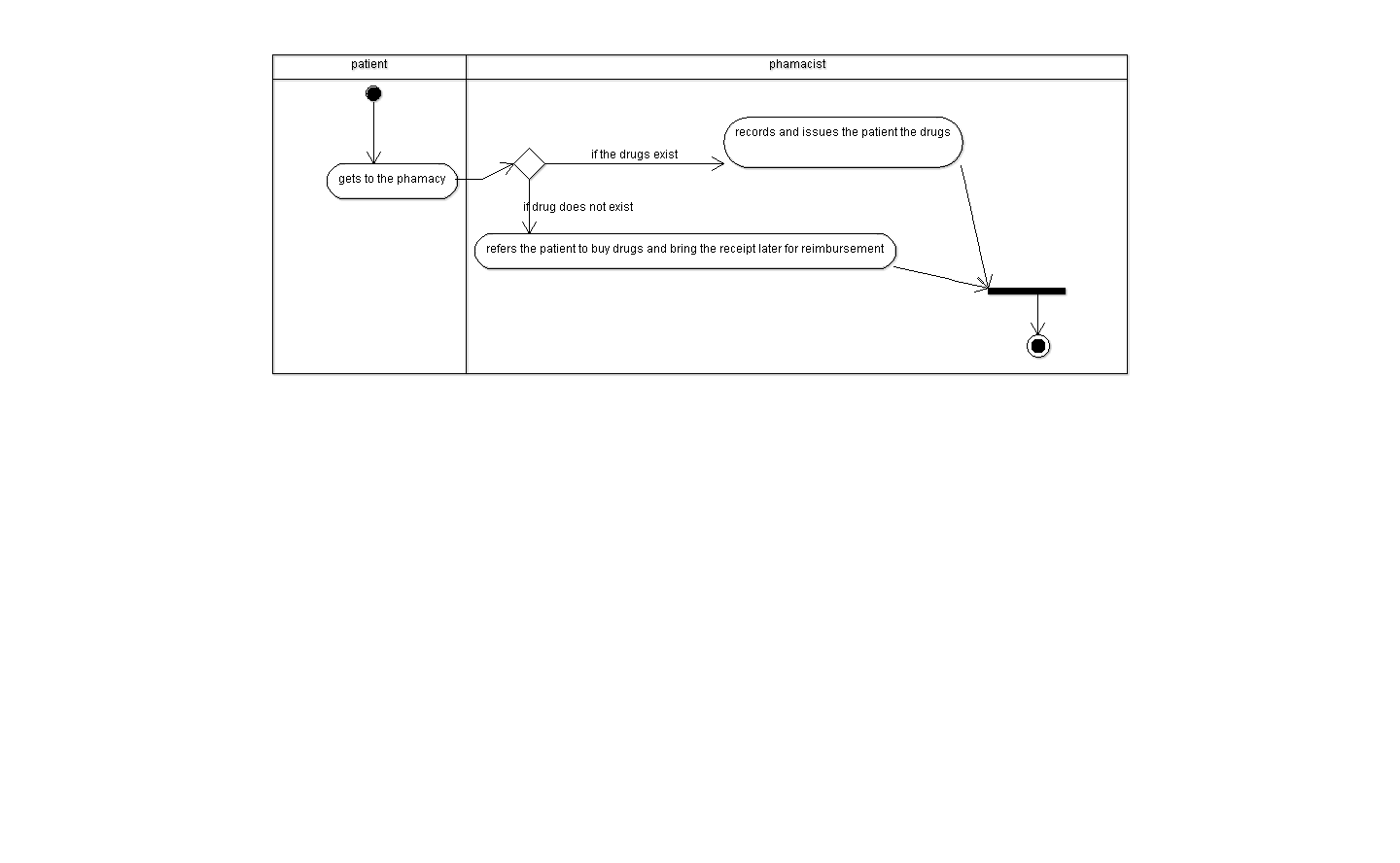
# 

# 

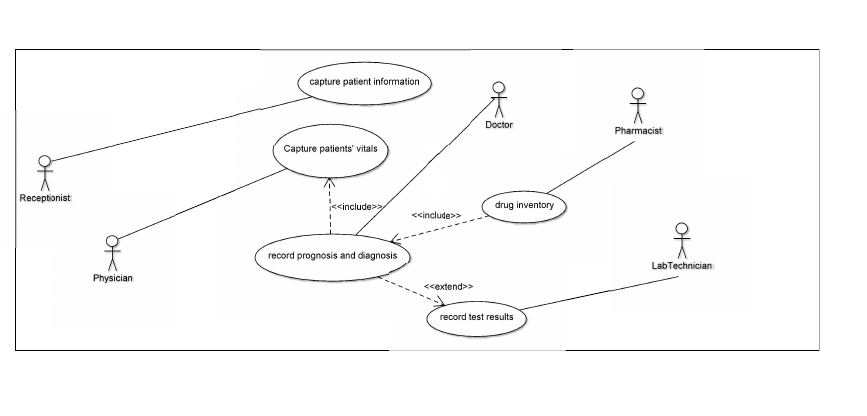
# 







## 6.2 Use Cases and use case description



|  |  |
| --- | --- |
| **Use case 1** | Capture patient information |
| **Description** | The use case helps validate the patient’s details. |
| **Triggers** | A patient comes into the reception. |
| **Actors** | Receptionist(primary), patient(secondary). |
| **Preconditions** | The receptionist must have an account in the system and log in to verify the patient’s details. |
| **Main flows** | 1.The patient provides the receptionist with their identification documents.  2.The receptionist keys in the the identification details of the patient i.e registration number if student, id number if a community member, or staff number if the patient is a staff.  3. After verification the receptionist sends the patient to the physician.  4.exits the system. |
| **Alternate flows** | 1.Fails to identify the patient if not a registered member.  2.Creates a new patient. |
| **Postconditions** | Updates the patient’s record and forwards to physician. |

|  |  |
| --- | --- |
| **Usecase 2** | Capture patient’s vitals |
| **Description** | The use case helps capture the patient’s vitals. This include temperature, weight, height and BMI. |
| **Triggers** | A patient comes into the physician’s office. |
| **Actors** | Physician(primary), patient(secondary). |
| **Preconditions** | The physician must have an account in the system and log in to receive all the patients’ entries from the receptionist. |
| **Main flows** | 1.The physician receives the patient with the entries from the receptionist..  2.He/she takes the patient’s vitals that include temperature, weight, height and BMI.  3.The physician then records the patient’s vitals and sends data to the medical specialist(doctor).  4.exits the system. |
| **Alternate flows** | none. |
| **Postconditions** | Updates the patient’s record and forwards to medical specialist(doctor). |

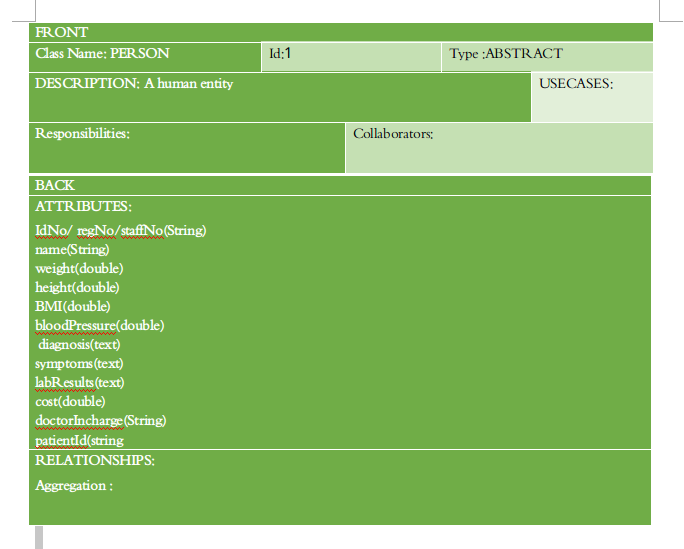
|  |  |
| --- | --- |
| **Usecase 3** | Record patients’ prognosis and diagnosis |
| **Description** | The use case helps capture the patient’s diagnosis and prognosis. |
| **Triggers** | A patient comes into the doctor’s office. |
| **Actors** | doctor(primary), patient(secondary). |
| **Preconditions** | The doctor must have an account in the system and must log in to receive all the patients’ entries from the physician . |
| **Main flows** | 1.The doctor receives the patient with the entries from the physician.  2.He/she conducts a prognosis for the patient.They also conduct a diagnosis thereafter.  3.From the diagnosis, the doctor prescribes the patient’s medicine and directs them to the pharmacy to collect the drugs.  4.exits the system. |
| **Alternate flows** | 1.Incase the patient’s problem cannot be identified from the diagnosis, the patient is send to the lab to take some related tests to help the doctor prescribe the medicine for the patient.  2.The doctor exits the system until he/she receives test results from the lab technician. |
| **Postconditions** | Updates the patient’s record.. |

|  |  |
| --- | --- |
| **Usecase 4** | Record test results |
| **Description** | The use case helps capture the result of the patient’s tests done by the lab technician.. |
| **Triggers** | A patient comes into the laboratory.  Lab technician gets test request from the doctor. |
| **Actors** | Lab technician(primary), patient(secondary). |
| **Preconditions** | The lab technician must have an account in the system and log in to receive all the patients’ entries from the doctor. |
| **Main flows** | 1.The lab technician takes the tests requested by the doctor..  2.He/she then updates the records and forwards them to the doctor.  3.exits the system |
| **Alternate flows** | none. |
| **Postconditions** | Updates the patient’s record .. |

|  |  |
| --- | --- |
| **Usecase 5** | Drug inventory |
| **Description** | The use case helps add new drugs to the system. It also helps in managing the drugs stock i.e to know what drugs are in stock and those that are almost depleted. |
| **Triggers** | A patient comes into the pharmacy.  The pharmacist receives prescription from the doctor. |
| **Actors** | Pharmacist, database. |
| **Preconditions** | The physician must have an account in the system and log in to add or deduct drugs from the database. |
| **Main flows** | 1.The pharmacist receives the drugs when they come to the store.  2. They then do an entry of all drugs to the system.  3. A patient then comes to the pharmacy .At the same time, the pharmacist receives the patient’s description of drugs from the doctor..  4.The pharmacist gives the patient the drugs and deducts the same from the database.  5.exits the system. |
| **Alternate flows** | Incase the prescribed drug does not exist the pharmacist refers the patient to purchase it from the nearby chemist. |
| **Postconditions** | none. |

# 7. Structural Modelling

## 7.1 Crc Cards Used:



## 7.2 Class Diagrams51qq+

## +

## +

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

## 

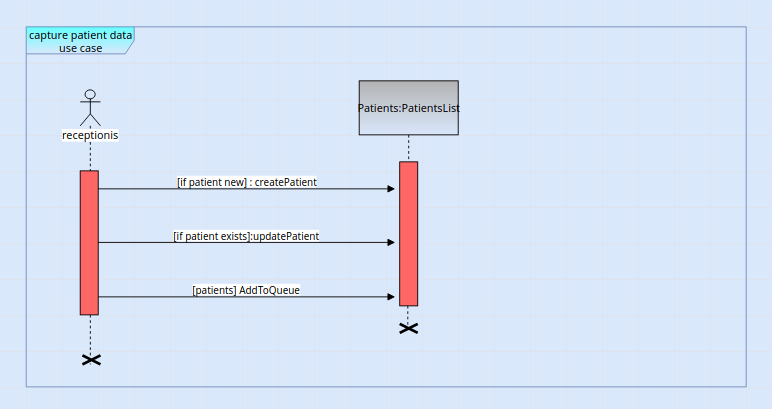
## 

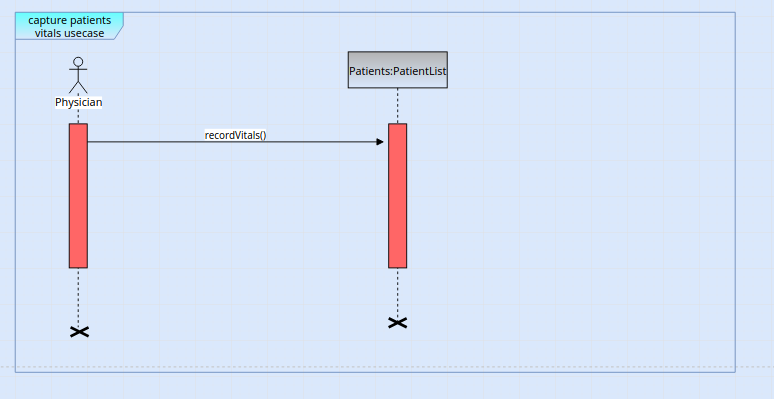
## 

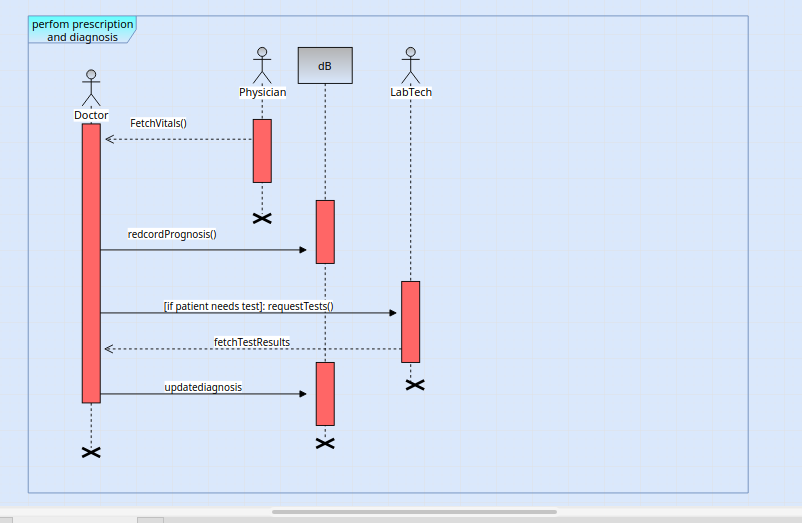
## 55222

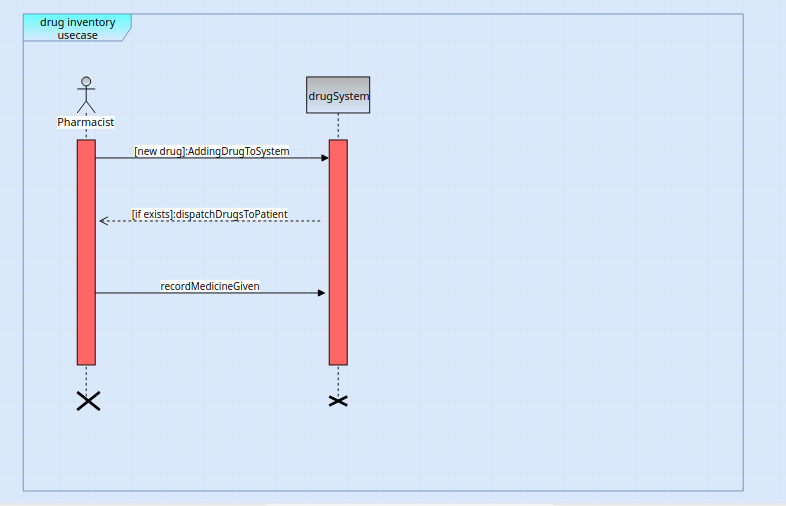
# 8. Behavioural Modelling

## 8.1 Sequence Diagrams

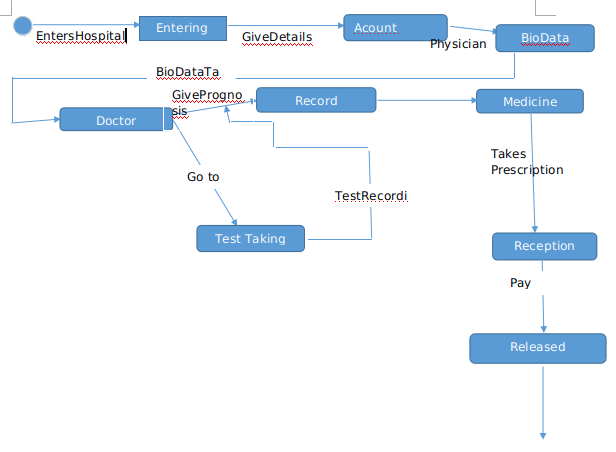








## 8.2 Behavioral State Machine Diagram



# 9. Design modelling

Users to the system who are the main interactors to the system, i.e receptionist, doctor, pharmacist and the lab technicians, have to login in to the system so that they may be able to access their stipulated responsibilities.

