**EGERTON UNIVERSITY**

**SOFTWARE DESIGN DOCUMENT FOR**

**TITLE:SANATORIUM INFORMATION SYSTEM**

**PREPARED BY:**

**STUDENTS NAME: CALVINCE OWUOR**

**REG. NO.: S13/21420/14**

**PROJECT SUPERVISOR:MR. BENJAMIN ODIYO**

**PROJECT COORDINATOR: DR GIKARU**

**REVISION HISTORY:**

|  |  |
| --- | --- |
| **DATE** | **VERSION** |
| 2/07/2018 | 1.0 |
| 4/08/2018 | 2.0 |

**OVERVIEW**

The System Design Document describes the system requirements, operating environment, system and subsystem architecture, files and database design, input formats, output layouts, human-machine interfaces, detailed design, processing logic, and external interfaces.

1. **INTRODUCTION**

The Software Design Document is a document to provide documentation which will be used to aid in software development by providing the details for how the software should be built. Within the Software Design Document are narrative and graphical documentation of the software design for the project including use case models, sequence diagrams, collaboration models, object behavior models, and other supporting requirement information.

* 1. **Purpose**

The purpose of the Software Design Document is to provide a description of the design of a system fully enough to allow for software development to proceed with an understanding of what is to be built and how it is expected to built. The Software Design Document provides information necessary to provide description of the details for the software and system to be built.

* 1. **Scope**

This Software Design Document is for a base level system which will work as a proof of concept for the use of building a system the provides a base level of functionality to show feasibility for large scale production use. This Software Design is focused on the base level system and critical parts of the system. For this particular Software Design Document, the focus is placed on generation of the documents and modification of the documents. The system will be used in conjunction with other existing systems and will consist largely of a document interaction facade that abstracts document interactions and handling of the document objects.

* 1. **Project Executive Summary**

This section provides a description of the project from a management perspective and an overview of the framework within which the conceptual system design was prepared.

* + 1. **System Overview**

The following figure , figure 1 provides the an overview of the system components and how its intends to achieve the business goals of the sanatorium department.

**School Database**

**Hospital Database**

**System User Interface**

***Figure 1: System architecture overview.***

* + 1. **Design Constraints**

Even though there are limited resources set aside for the development and design of the system, the development team are expected to complete the development and deliver the product within the stipulated time and within the budget. The system is expected to be of quality and meets all of the business goal for which it will be deployed to handle. The labor, time and financial constraints should not impact much on the delivery of the product.

* + 1. **Future Contingencies**

The are a number of factors that might arise during the system implementation stage, which may impact the software quality and the intended delivery time. Disagreement on how the users should interact with the system and the interfaces presented to them may prove to be an obstacle. The stakeholders may not like the designed interfaces and thus request for the design of a different user interface. Also as the development progresses, there may be need for additional financial allocation, something that the stakeholders my not be willing to provide.

* 1. **Document Organization**

This design document is divided into sections describing the different design strategies of the system. The Document starts by spelling the overview of the document, the introduction, the system architecture,file and database design, human-machine interface design, detailed design, external interfaces and the system integrity constraints.

* 1. **Point of Contact**

This section provides the organization code and title of the key points of contact (and alternates if appropriate) for the information system development effort. However, for this system there are only three development team members, one developer who is concerned with the all activities that are concerned with the design and development, the project supervisor and the project coordinator.

* 1. **References**

Prior to the delivery of this document, the are other documents that had already been delivered to the project supervisor as well as the coordinator – System Proposal Document and the Software Requirement Specification Document. In addition to the mentioned documents, the following documentation will be of great importance in the writing of this document – XML Legal Documents Utility Software Development Plan Version 1.0, Last Updated on 2007-01-31.

* 1. **Glossary**

|  |  |
| --- | --- |
| **TERM** | **DEFINITION** |
| SDD | System Design Document |
| De-normalized | Normalization of a database is the activity of restructuring the database to avoid data anomalies and inconsistencies by focusing on functional dependencies to help structure the data. |
| Data Objects | Data objects are Java objects with predefined structures capable of holding data in a structure that is quickly and easily accessible by other parts of the software system. They provide also can help provide a convenient abstraction of the data in a database so that it can be retrieved into a format, such as a De-normalized format, that makes access and manipulation of the data easier than if the database had to be called directly. |
| Editable Form Layout | A user interface presentation layout in which the contents of a document are presented to a user in the format of a form predefined editable areas based on the type of document which is being edited. This type of layout allows for changes to be made in a specific manner so that the data used in the form can be reassembled into a structured data format for transfer to other systems and archival. |
| JDBC/ODBC | These two acronyms stand for Java Database Connectivity and Open Database Connectivity API's which allow for standardized database access and interaction from software products. |
| API | Application programming interface |
| Required Field | A critical field is a field in a data set for a document that is required for successful document generation. For example, missing parties in a case, missing county location of court, or other data elements that are required to create a valid legal document. |
| UUID | Universally Unique Identifier. A UUID is an identifier standard in software construction which allows for generating identifiers which do not overlap or conflict with other identifiers which were previously created even without knowledge of the other identifiers. |
| Workflow | The movement of documents through a work process that is structured into tasks with designated persons or systems to perform them and the definition of the order or pathway from start to finish for the work process. |
| XML | eXtensible Markup Language |
| Normalized | Normalization of a database is the activity of restructuring the database to avoid data anomalies and inconsistencies by focusing on functional dependencies to help structure the data |

1. **SYSTEM ARCHITECTURE**

The section describes the system’s overall architecture – it shows how the different components of the system interact to achieve the intended functionality.

* 1. **System Hardware Architecture**

For the system to be able achieve the intended purpose, it is constructed that it interacts with different hardware components that are both locally and remotely. For instance, the Student records are stored in a database that is contained within another hardware, while the patient records remotely in the remote database servers, while the the interfaces are available locally to the user. The lists of hardware components are :

* + 1. Computer Monitor
    2. Servers
    3. Computer Storage

The figure 2: below shows the hardware components that constitutes the system.

**Computer Monitor**

**Web Server**

**Remote Server**

**Computer Memory**

***Figure 2: System hardware architecture .***

* 1. **System Software Architecture**

The Design Overview is section to introduce and give a brief overview of the design. The System Architecture is a way to give the overall view of a system and to place it into context with external systems. The figure **3** below gives the overall software architecture overview.

submitMedication()

<interface>

Pharmacy

getMedication()

<<Class>>

Medication

<interface>

Medication

<<Class>>

Admission

calculateBMI()

submitRecord()

<<Class>>

Student

getStudentRecord()

<<Class>>

Patient Reception

confirmStudent()

recordVisitInfo()

updatePatientRecord()

<interface>

Reception

confirmStudent()

<<Class>>

Visit

<interface>

Trige

<interface>

SchoolRecord

<interface>

Lab

getTest()

submitTest()

<interface>

Visit

recordPatientVisit()

<<Class>>

Patient

<<Class>>

Lab

<<Class>>

Pharmacy

<interface>

Admission

saveAdmissionRecord()

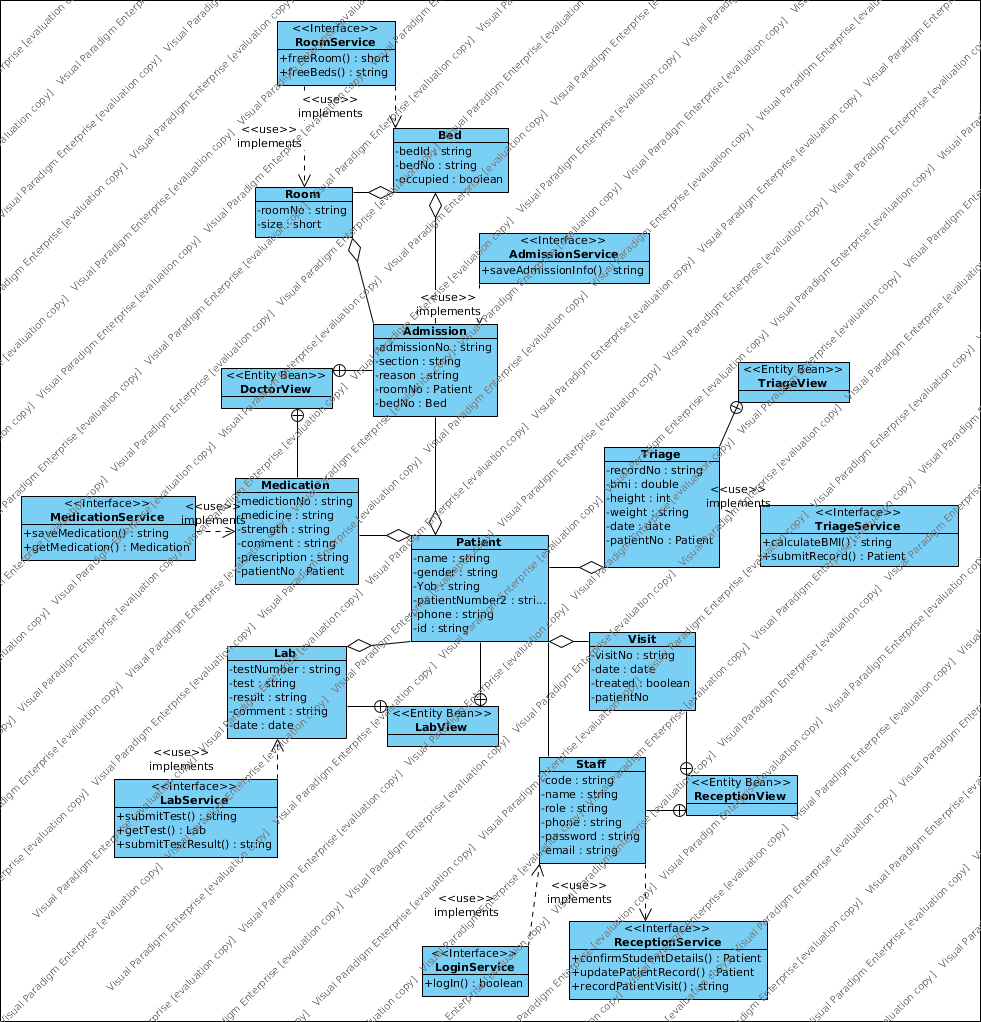
<<Class>>

Triage

* + 1. **Overall software architecture**

***Figure 3: System software architecture.***

* + 1. **Class Diagram**

***Figure 4: The class diagram.***

* 1. **Internal Communications Architecture**

The communication between the different modules of the system will be aided by the presence of internet connection. The connection is presumed be a local area network, which is managed internally by the department. However, for these modules to be feasible, there are different users with specified privileges assigned to each module. The modules are:

* + - * 1. Reception
        2. Triage
        3. Lab
        4. Consultation
        5. Pharmacy

Each of these modules interact to ensure a complete medication process is delivered to the patient, from the point of reception all the way to the treatment and release from the hospital premises.

1. **FILE AND DATABASE DESIGN**

This section provides a detailed design of the database to be used to store the records that will be consumed or exchanged during the normal operations of the system.

* 1. **Database Management System Files**

Bellow are the relations that will be used to hold the data during the operation of the system. The files are detailed independently at this stage.

* + 1. **Admission Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| admission\_number | varchar(15) | NO | PRI | NULL |  |
| date | date | NO |  | NULL |  |
| reason | varchar(300) | NO |  | NULL |  |
| section | varchar(50) | NO |  | NULL |  |
| time | time | NO |  | NULL |  |
| bed | varchar(15) | NO | MUL | NULL |  |
| doc\_number | varchar(15) | NO | MUL | NULL |  |
| patient\_number | varchar(15) | NO | MUL | NULL |  |
| room | varchar(4) | NO | MUL | NULL |  |

* + 1. **Bed Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| record\_number | varchar(15) | NO | PRI | NULL |  |
| bed\_number | varchar(1) | NO |  | NULL |  |
| occupied | bit(1) | NO |  | NULL |  |
| room\_number | varchar(4) | NO | MUL | NULL |  |

* + 1. **Room Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| room\_number | varchar(4) | NO | PRI |  |  |
| Size | Varchar(1) | NO |  | NULL |  |

* + 1. **Illness Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| illness\_number | varchar(15) | NO | PRI | NULL |  |
| date | date | NO |  | NULL |  |
| illness | varchar(55) | NO |  | NULL |  |
| admission\_number | varchar(15) | NO | MUL | NULL |  |
| patient\_number | varchar(15) | NO | MUL | NULL |  |

* + 1. **Staff Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| employee\_number | varchar(15) | NO | PRI | NULL |  |
| email | varchar(100) | NO |  | NULL |  |
| first\_name | varchar(15) | NO |  | NULL |  |
| last\_name | varchar(15) | NO |  | NULL |  |
| gender | varchar(15) | NO |  | NULL |  |
| national\_id | varchar(8) | NO |  | NULL |  |
| phone | varchar(10) | NO |  | NULL |  |
| Role | varchar(25) | NO |  | NULL |  |

* + 1. **Lab Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| lab\_test\_number | varchar(15) | NO | PRI | NULL |  |
| comment | varchar(350) | NO |  | NULL |  |
| date | date | NO |  | NULL |  |
| result | varchar(350) | NO |  | NULL |  |
| test | varchar(150) | NO |  | NULL |  |
| patient\_number | varchar(15) | NO | MUL | NULL |  |

* + 1. **Medication Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| medicine\_number | varchar(15) | NO | PRI | NULL |  |
| comment | varchar(350) | NO |  | NULL |  |
| date | date | NO |  | NULL |  |
| medicine\_name | varchar(150) | NO |  | NULL |  |
| mode | varchar(50) | NO |  | NULL |  |
| description | varchar(250) | NO |  | NULL |  |
| strength | varchar(155) | NO |  | NULL |  |
| illness\_number | varchar(15) | NO | MUL | NULL |  |
| doc\_number | varchar(15) | NO | MUL | NULL |  |
| patient\_number | varchar(15) | NO | MUL | NULL |  |

* + 1. **Patient Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| patient\_number | varchar(15) | NO | PRI | NULL |  |
| address | varchar(100) | NO |  | NULL |  |
| first\_name | varchar(15) | NO |  | NULL |  |
| last\_name | varchar(15) | NO |  | NULL |  |
| gender | varchar(15) | NO |  | NULL |  |
| national\_id | varchar(8) | NO |  | NULL |  |
| phone | varchar(10) | NO |  | NULL |  |
| YoB | varchar(12) | NO |  | NULL |  |

* + 1. **Triage Table**

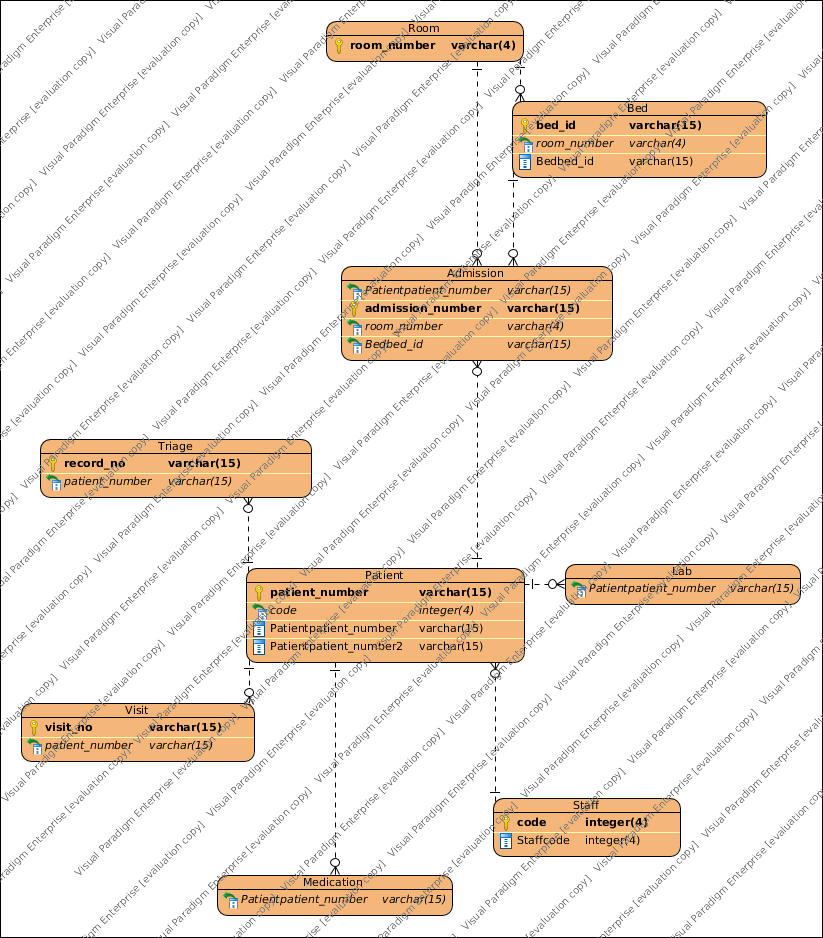
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| record\_number | varchar(15) | NO | PRI | NULL |  |
| bmi | varchar(5) | NO |  | NULL |  |
| date | date | NO |  | NULL |  |
| height | varchar(3) | NO |  | NULL |  |
| pressure | varchar(10) | NO |  | NULL |  |
| weight | varchar(5) | NO |  | NULL |  |
| time | time | NO |  | NULL |  |
| patient\_number | varchar(15) | NO | MUL | NULL |  |

* + 1. **Visit Table**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Field** | **Type** | **Null** | **Key** | **Default** | **Extra** |
| visit\_number | varchar(15) | NO | PRI | NULL |  |
| date | date | NO |  | NULL |  |
| time | time | NO |  | NULL |  |
| treated | bit(1) | NO |  | NULL |  |
| patient\_number | varchar(15) | NO | MUL | NULL |  |

* 1. **Database Relations**

The figure 5 below describes the relationship between the tables – the relations are done such that insertion and deletion anomalies are avoided during the operations of the system.



1. **HUMAN – MACHINE INTERFACE**

This section provides the detailed design of the system and subsystem inputs and outputs relative to the user/operator.

* 1. **Inputs**

There are a number of input screens that allows the user to feed information into the system.

* + 1. **Login form**

To start off with is the login page, that allows the user to input credentials for authentication. Figure 5 below shows the latter.

Password

**Login**

User code

User code

***Figure 5: Login interface***

If the user authentication is successful, the user will be directed into another interface where the tasks corresponding to the assigned privileges can be accessed.

* + 1. **Bmi form**

Once the user is logged in to the system as a nurse of the triage section an interface where the data relating to the bmi recordings is shown below.

**BMI MEASUREMENTS**

Height

Pressure

Weight

Patient. No

Name

BMI

**CALCULATE BMI**

**SUBMIT**

***Figure 5: BMI calculation interface***

* + 1. **Admission form*.***

When the user want to record the Admission details for the patient, he/she is presented with and interface that resemble the once shown in figure 6 below.

Time

Date

Name

Patient. No

**SUBMIT**

Reason

Bed

Room

Section

**Admission Form**

***Figure 6: Admission details entry interface***

* + 1. **Medication form*.***

When the user want to prescribe the medication for the patient, he/she is presented with and interface that resemble the once shown in figure 7 below.

Medicine

Illness

Strength

Prescription

**SUBMIT**

Patient. No

Name

Comment

Mode

**Medication Form**

***Figure 7: Medication details entry interface***

* + 1. **Test entry form**

When the doctor wants to input the tests to be done on the patient, the following interface is leveraged, see figure 8.

**TEST SUBMISSION**

Test

Patient. No

Name

**SUBMIT**

**CLOSE**

***Figure 8: Test submission interface***

* + 1. **Test result submission form**

When the physician wants to submit the test results done on the patient, the following interface is leveraged, see figure 9.

**LAB TEST RESULT SUBMISSION**

Result

Comment

Test

Patient. No

Name

**TEST NUMBER**

**TEST NAME**

Date

**SUBMIT**

***Figure 9: Lab test result submission interface.***

* + 1. **Patient Reception form*.***

When the user want to undertake the patient reception action, he/she is presented with and interface that resemble the once shown in figure 10 below.

Input registration number

**SEARCH**

**STUDENT RECEPTION**

***Figure 5: Student Reception interface***

* 1. **Outputs**

There are a number of output screens that allows the user to view the result of the system’s actions.

* + 1. **Student Records view**

When the system fetches the student records from the school database the information fetched is displayed in an interface that resembles the one shown in the figure 11 below.

S13/21450/14

Reg.No

Name

Calvince Owuor

Faculty

Science

Dept

Gender

Male

07217845771

Phone

Computer Science

**OK**

***Figure 11: Student Records interface***

* + 1. **Patient Records view**

When the system fetches the patient records the information fetched is displayed in an interface that resembles the one shown in the figure 12 below

07217845771

**CONFIRM**

YES

**TREATED**

11:50 am

**TIME OF VISIT**

**DATE OF VISIT**

14/5/218

**VISISIT INFO**

Phone number

Date Registered

24/5/2018

Gender

Male

Calvince Owuor

Name

Patient Number

PA599BJHJJ8799A

**PATIENT RECORDS**

***Figure 12: Patient Record interface***

* + 1. **Consultation view**

When the doctor clicks on the open consultation view tab on the doctors view the information is displayed in an interface that resembles the one shown in the figure 13 below.

***Figure 13: Patient Consultation interface***

**ADM. NO**

NGGJ27537AA7

**SERIAL**

Oroda

**MEDICATION**

**DATE**

24/1/2018

**PREVIOUS MEDICATIONS**

SJAGJDHJ231A

**PREVIOUS ILLNESS**

Medical Care

**SECTION**

**DATE**

2/3/2018

**PREVIOUS ADMISSIONS**

1

**ROOM**

**BED**

1

ASHXZXXX3551A

**TEST NUMBER**

Thromboin

**TEST NAME**

A- clots

**RESULT**

**DATE**

24/1/2018

**PREVIOUS LAB RESULTS**

**CONSULTATION OPENED FOR**

Calvince Owuor

SGMHGN2646BA

**SERIAL**

Malaria

**ILLNESS**

**DATE**

24/1/2018

When the user is clicks on the “*end consultation*” a confirm dialog is shown ,that requires the user to acknowledge the ending of the consultation process for the particular patient. See the figure 14 below.

Ending the consultation process..?

**CANCEL**

**OK**

* + 1. **Doctor Board View**

When the user logs into the system an a doctor he is presented with an output screen that shows the relevant actions that the doctor can perform as shown in figure 12 below.

GENDER

**ADMIT**

**MEDICATION**

**ILLNESS**

**TEST**

**To Do List**

**profile**

ROLE

YOB

NAME

**User**

**CONSULTATION**

**DOCTOR**

***Figure 12: Doctor Board interface***

If the user is clicks on the “*consultation*” tab a window with two options are shown. One option is to initiate the consultation process and another to end the consultation process. Initiating the process enables the doctor to view the patient that is at the attendance. Ending the process is done after the doctor treated the patient. Figure 13 below shows the interface.

**CONSULTATION OPTION**

**OPEN CONSULTATION**

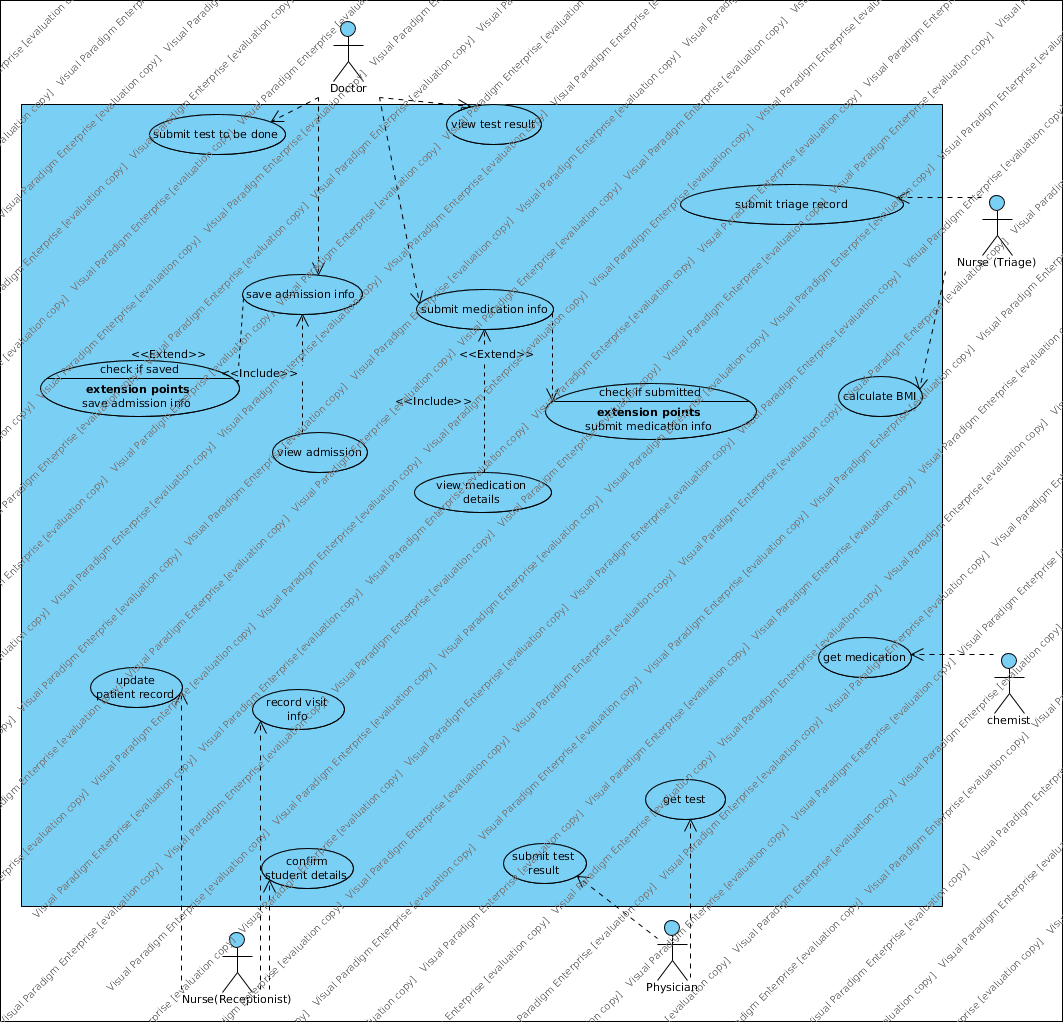
**END CONSULTATION**

***Figure 13: Consultation option interface***

1. **DETAILED DESIGN**

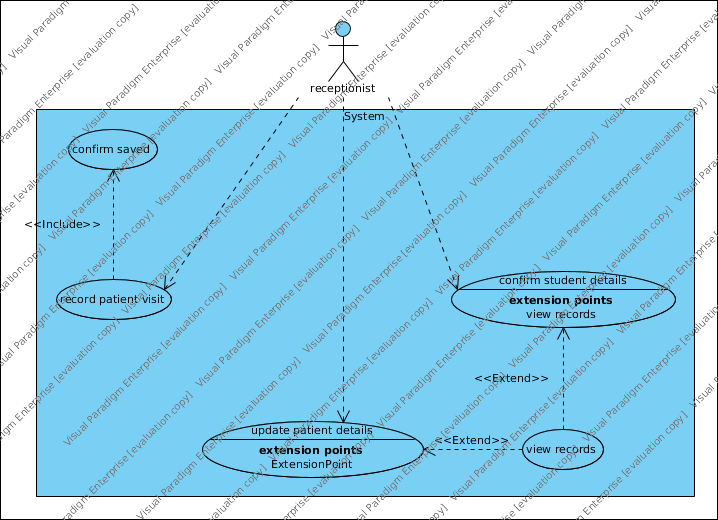
This section provides the information needed for a system development team to actually build and integrate the hardware components, code and integrate the software modules, and interconnect the hardware and software segments into a functional product. Additionally, this section addresses the detailed procedures for combining separate COTS packages into a single system.

* 1. **Software Detailed Design**
     1. **Use cases**
        + **System Use case**

The figure 14 below presents the overall use case of the system. It shows the expected actors of the system and how they interact with the system.

Due to the time and resource limits, each only four use cases will be described and illustrated to both the user and the developers of the system who will be using this system.

* + - * **Receptionist Use case**

The figure 15 below presents the receptionist use cases for the interaction with the system.

***Figure 15: receptionist use cases***

* + - * **Receptionist Use cases –** *confirm student record*

**Use case name:** *confirm student record*

**ID:** *RCS*

**Priority:** *High*

**Primary actor***: Receptionist*

**Use case type:** *Business*

**Level***:Overview*

**Interested Stakeholders***: Nurse*

**Brief description:**

*This use case describes the reception of the students who came seeking for the services of the sanatorium – reception module is one of the key functions of the system. In this use case the main target of the receptionist is to confirm who the student is claiming to be.*

**Goal:**

*The successful completion of student identification.*

**Success Measurement:**

*The student record matching the provided id is extracted and the record is presented to the receptionist.*

**Precondition:**

* *The receptionist has successfully passed through Authentication and Authorization*
* *There is a student available for reception and has provided his/her student id car of letter of offer.*

**Trigger:**

*The nurse enters the registration number of the student and presses the search button.*

**Relationships***:*

*Include: Extend: Depends on:*

**Typical flow of events***:*

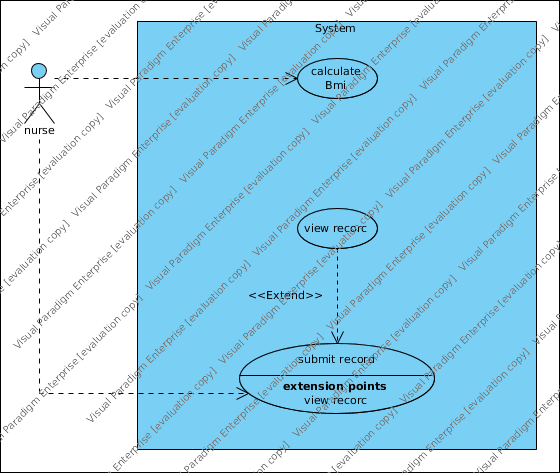
* + - 1. *The receptionist enters the student registration of student.*
      2. *The nurse then clicks the submit button.*
      3. *The system fetches the records matching the provided registration number*
      4. *The data fetched is presented to the receptionist who confirms them against the patient’s card or letter of offer.*
      5. *Once the receptionist is satisfied with the records presented, he/she clicks on the “OK” button.*
      6. *Another wok flow is initiated.*

**Alternative flow of events:**

1. *Records matching the student registration number is not found.*
2. *The system halts and displays nothing.*

**Assumptions** :

1. *It is assumed that work-flows will be carried out internally or within the reception section of the sanatorium.*
2. *It is assumed that the student will provide a valid registration number and that the student is still in session.*
3. *It is assumed that the receptionist will input the registration number in a recommended format.*
   * + - **Nurse (Triage) Use case**

*The figure 16 below presents the nurse (at triage section) use cases for the interaction with the system.*

***Figure 16: receptionist use cases***

* + - * **Nurse ( Triage) Use cases –** *calculate patient bmi*

**Use case name:** *calculate patient bmi*

**ID:** *CPB*

**Priority:** *medium*

**Primary actor***: Nurse*

**Use case type:** *Business*

**Level***:Overview*

**Interested Stakeholders***: Nurse*

**Brief description:**

*This use case describes the calculation of the BMI of the patient after the doctor has requested for the same. This feature is available at the triage section of sanatorium – triage module is one of the key functions of the system. In this use case the main target of the nurse is to measure different aspects of the patient body and then calculate the resulting BMI.*

**Goal:**

*The successful calculation of th patient BMI.*

**Success Measurement:**

*The patient BMI of the patient is calculated and the result displayed to the nurse to confirm.*

**Precondition:**

* *The nurse has successfully passed through Authentication and Authorization*
* *There is a request from the doctor that requires that the patient BMI recordings be done and sent.*
* *There is a patient whose BMI is to be measured.*

**Trigger:**

*The nurse the presses the calculate button on the form.*

**Relationships***:*

*Include: Extend: Depends on:*

**Typical flow of events***:*

* + - 1. *The nurse enters the height,weight,and blood pressure of the patient.*
      2. *The nurse then clicks the calculate button.*
      3. *The system then calculates the BMI based on the inputs.*
      4. *The resulting result of the calculation is presented to the user who confirms it before sending to the doctor.*
      5. *Once the receptionist is satisfied with the record presented, he/she clicks on the “Submit” button.*
      6. *Another wok flow is initiated.*

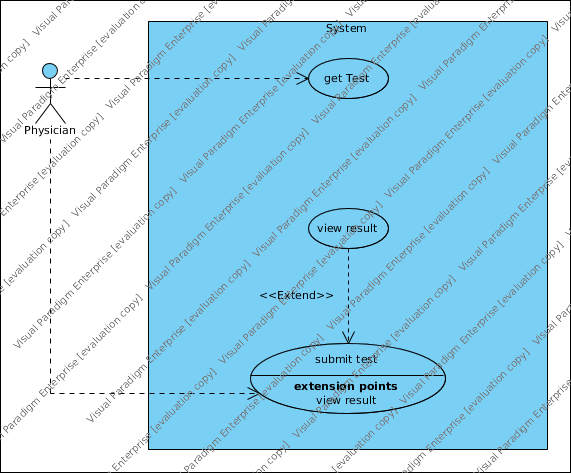
**Alternative flow of events:**

1. *User forgets to provide either of the inputs.*
2. *The system displays a message requesting for missing input.*
3. *The system halts and does nothing.*

**Assumptions** :

1. *It is assumed that work-flows will be carried out internally or within the triage section of the sanatorium.*
2. *It is assumed that the nurse will provide a valid inputs and leaves neither of the inputs blank.*
3. *It is assumed that the patient is present and have already been confirmed and the doctor has opened the consultation with him/her.*
   * + - **Physician Use case**

*The figure 17 below presents the Physician (at lab section) use cases for the interaction with the system.*

***Figure 17: Physician use cases***

* + - * ***Physician* ( Lab) Use cases –** *submit test result*

**Use case name:** *submit test result*

**ID:** *SBTR*

**Priority:** *medium*

**Primary actor***: Physician*

**Use case type:** *Business*

**Level***:Overview*

**Interested Stakeholders***: Physician*

**Brief description:**

*This use case describes the submission of the patient test result after the test has been done and the results are ready for recoding in the system – after the doctor has requested for the same. This feature is available at the lab section of sanatorium – lab module is one of the key functions of the system. In this use case the main target of the physician is to record the test results.*

**Goal:**

*The successful submission of the patient test results.*

**Success Measurement:**

*The patient test results are submitted and the success message displayed to the user.*

**Precondition:**

* *The physician has successfully passed through Authentication and Authorization*
* *There is a request from the doctor that requires that the patient be tested and the test results be sent.*
* *There is a patient whose whose test results results are to be recorded.*

**Trigger:**

*The physician presses the submit button on the form.*

**Relationships***:*

*Include: Extend: Depends on:*

**Typical flow of events***:*

* + - 1. *The physician fills the inputs resulting from the test results.*
      2. *The nurse then clicks the submit button.*
      3. *The system first checks if the test has already been submitted.*
      4. *If the test has not been submitted, the system submits the test results.*
      5. *Successful submission is followed by a message that alerts the user of the outcome of the submission process.*
      6. *Another wok flow is initiated.*

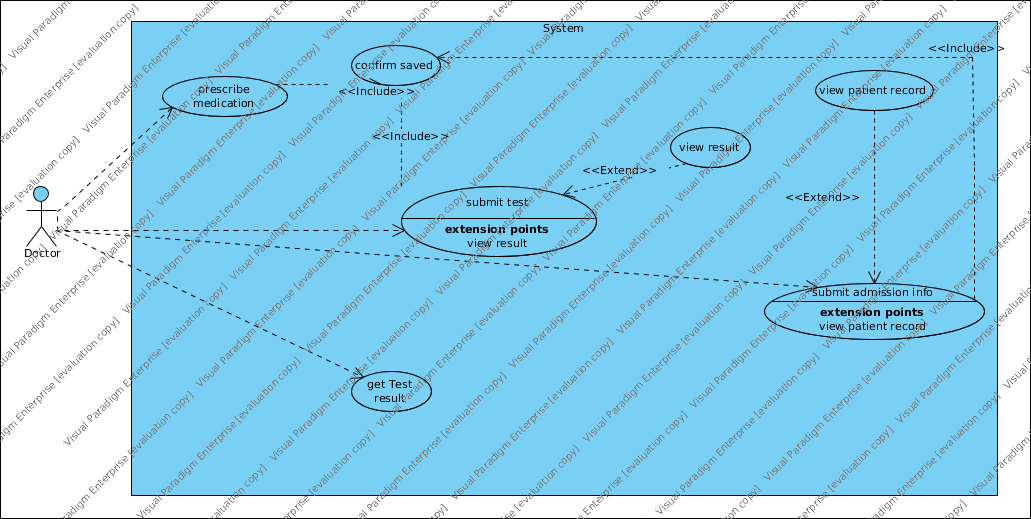
**Alternative flow of events:**

1. *User forgets to provide either of the inputs.*
2. *The system displays a message requesting for missing input.*
3. *The system halts and does nothing.*

**Assumptions** :

1. *It is assumed that work-flows will be carried out internally or within the lab section of the sanatorium.*
2. *It is assumed that the physician will provide a valid inputs and leaves neither of the inputs blank.*
3. *It is assumed that the doctor had sent the tests to be done on the patient before sending him/her to the lab.* 
   * + - **Doctor Use case**

*The figure 18 below presents the doctor’s use cases for the interaction with the system.*

***Figure 18: Doctor use cases***

* + - * **Doctor Use cases –** *prescribe medication*

**Use case name:** *prescribe medication*

**ID:** *RPM*

**Priority:** *high*

**Primary actor***: Doctor*

**Use case type:** *Business*

**Level***:Overview*

**Interested Stakeholders***: Doctor*

**Brief description:**

*This use case describes the submission of the patient prescribed medications – after the doctor has identified the illnesses that the patient might have been suffering from. This feature is available at the consultation section of sanatorium – doctor module is one of the key functions of the system. In this use case the main target of the doctor is to record the relevant medications for the identified illnesses.*

**Goal:**

*The successful prescription of the patient medications.*

**Success Measurement:**

*The patient medications are submitted and the success message displayed to the user.*

**Precondition:**

* *The Doctor has successfully passed through Authentication and Authorization*
* *There is a patient illnesses has already been identified and the doctor has decided that the patient doesn’t require close monitoring and therefore should be released from the hospital premises.*

**Trigger:**

*The doctor presses the submit button on the form.*

**Relationships***:*

*Include: Extend: Depends on:*

**Typical flow of events***:*

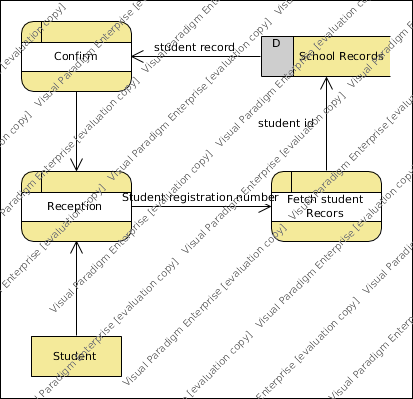
* + - 1. *The doctor fills the inputs resulting from the test results.*
      2. *The doctor then clicks the submit button.*
      3. *The system first checks if the medication has already been recorded.*
      4. *If the test has not been recorded, the system submits the prescribed medications.*
      5. *Successful submission is followed by a message that alerts the user of the outcome of the submission process.*
      6. *Another wok flow is initiated.*

**Alternative flow of events:**

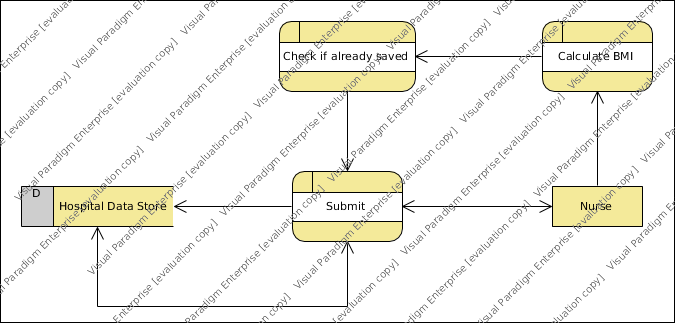
1. *User forgets to provide either of the inputs.*
2. *The system displays a message requesting for missing input.*
3. *The system halts and does nothing.*

**Assumptions** :

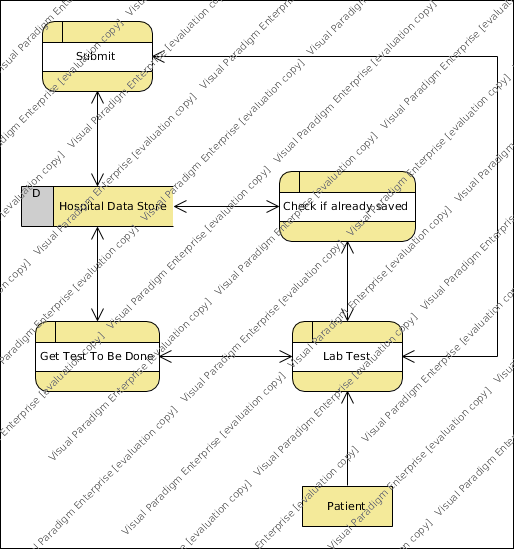
1. *It is assumed that work-flows will be carried out internally or within the consultation section of the sanatorium.*
2. *It is assumed that the doctor will provide a valid inputs and leaves neither of the inputs blank.*
3. *It is assumed that the doctor had already identified what the patient is suffering from and recorded the same.*
   * 1. **Dynamic Model**
        + **Data Flow Diagrams**
          1. **Confirm student details data flow diagram**

The figure 15 below presents the data-flow diagram of student reception procedure at the reception phase.

* + 1. **Calculate patient BMI data flow diagram**

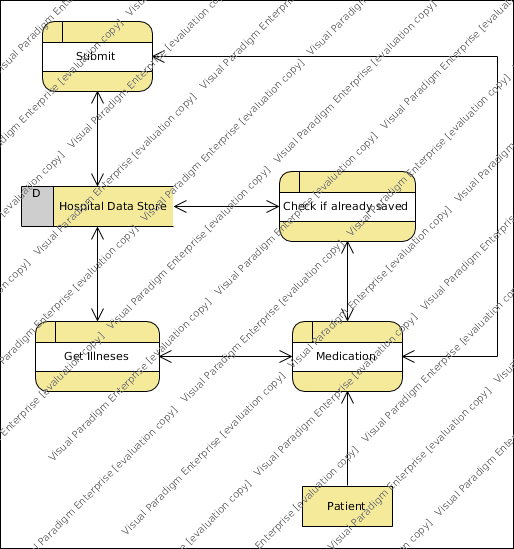
The figure 16 below presents the data-flow diagram of bmi calculation procedure at the triage section. The figure 16 below presents the data-flow diagram of bmi calculation procedure at the triage section.

* + 1. **Test submission data flow diagram**

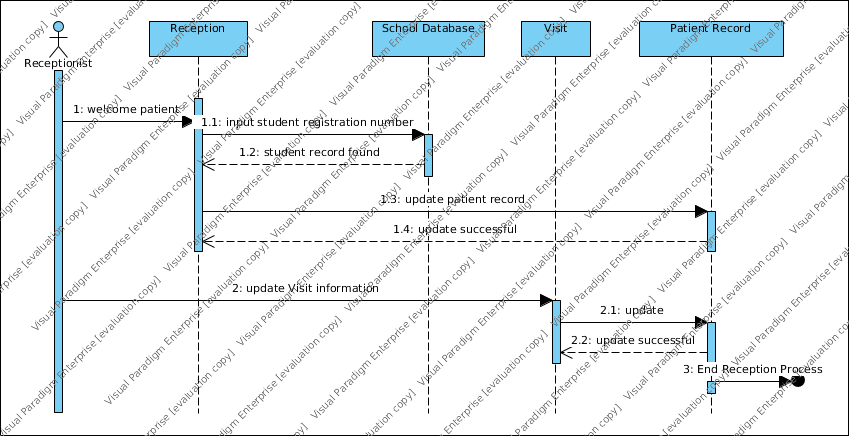
The figure 17 below presents the data-flow diagram of test submission procedure at the lab section.

* + 1. **Medication record submission data flow diagram**

The figure 18 below presents the data-flow diagram of medication record submission procedure at the consultation section.

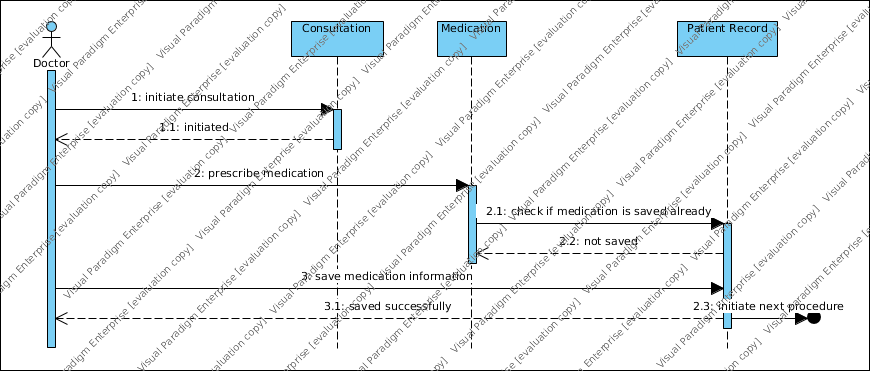


* + - * **Sequence Diagrams**
        1. **Confirm student sequence diagram**

The figure 19 below presents the data-flow diagram of student reception procedure at the reception phase.

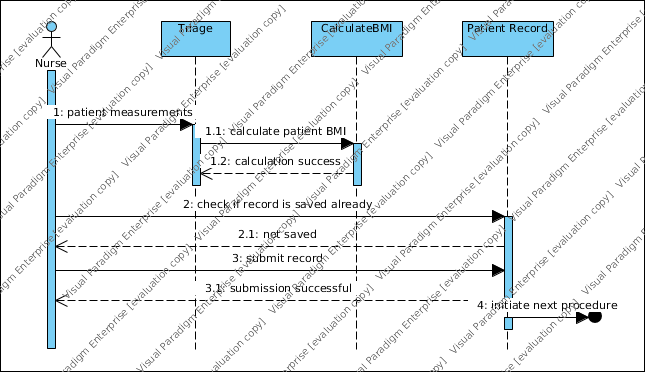
* + 1. **Confirm student sequence diagram**

The figure 20 below presents the sequence diagram of patient medication record entry procedure at the consultation section.

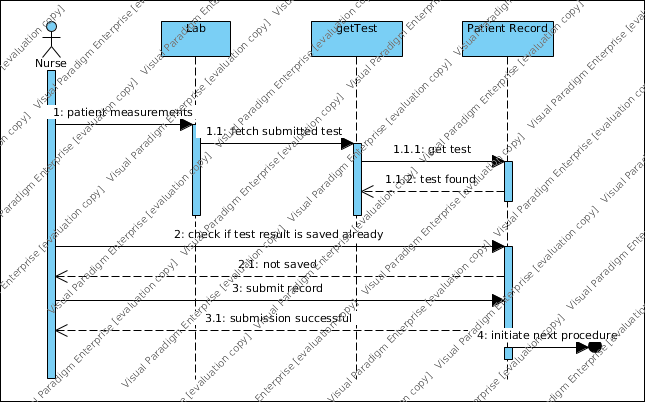


* + 1. **BMI calculation sequence diagram**

The figure 20 below presents the sequence diagram of patient BMI calculation and submission procedure at the triage section.



* + 1. **Test Result submission sequence diagram**

The figure 21 below presents the sequence diagram of patient test result submission procedure at the lab section.

1. **EXTERNAL INTERFACES**
   1. **Interface Design**

The interface for the system will allow the user to easily generated documents, search for documents, and modify documents. The user should be presented with all main functions on the first user interface page to allow for the user to select the function to use without the need to navigate inward to find it. The interface will need to use tab focus marks to allow for navigation using a keyboard as much as possible to alleviate stress on users' arms and hands caused by changing constantly from keyboard to mouse. It will be accessible through a web interface to allow for centralized hosting and use by various operating systems.

The software will need to interface with a case management system to pull data from it and push data updates to it. The connection will be a standard database connection using JDBC or ODBC

It is assumed the certain documents used within a sanatorium and with closely partnered agencies can be standardized and held stable enough in structure that the supporting structures of an XML schema for the XML data set, an XSL Stylesheet, a classification of the data elements used for the document for security applications, and element update screens can be created and held reasonably stable to avoid a churn of constant modifications to the system and the supporting elements for the documents.

1. **SYSTEM INTEGRITY CONTROLS**

Sensitive systems use information for which the loss, misuse, modification of, or unauthorized access to that information could affect the conduct of State programs, or the privacy to which individuals are entitled. Developers of sensitive State systems are required to develop specifications for the following minimum levels of control:  Internal security to restrict access of critical data items to only those access types required by users  Audit procedures to meet control, reporting, and retention period requirements for operational and management reports  Application audit trails to dynamically audit retrieval access to designated critical data  Standard Tables to be used or requested for validating data fields  Verification processes for additions, deletions, or updates of critical data Ability to identify all audit information by user identification, network terminal identification, date, time, and data accessed or changed.