

A thick dark grey vertical bar runs down the left side of the page. An orange arrow points to the right from this bar, containing the date '1/1/2015'. Below the bar, several thin, curved lines in black and grey sweep upwards and to the right.

1/1/2015

# Medical Lab Automation System (MLAS)

Software Analysis (SA) and Software  
Design (SD) Document

R Naresh (13CS30024)

Barnopriyo Barua (13CS30009)

SOFTWARE ENGINEERING ASSIGNMENT

## TABLE OF CONTENTS

---

1	Feasibility study.....	3
1.1	Understanding the problem.....	3
1.2	Scope of the problem.....	3
1.3	Analyzing stakeholders .....	3
1.4	Defining alternatives .....	3
1.4.1	Connection between various supervisors (Management) and the software .....	3
1.4.2	Hardware Infrastructure .....	4
1.4.3	Software Infrastructure.....	4
1.4.4	Technology used .....	4
1.4.5	Security .....	4
1.5	Defining criteria to evaluate .....	4
1.6	Assessment of unusual circumstances.....	5
1.7	Evaluation of alternatives .....	5
1.7.1	Connection between resident and city corporation offices .....	5
1.7.2	Connection between various supervisors and the software .....	5
1.7.3	Hardware infrastructure .....	5
1.7.4	Software infrastructure.....	5
1.7.5	Technology used .....	5
1.7.6	Security .....	6
1.8	Report .....	6
2	Requirement analysis.....	6
2.1	Functional Requirements.....	6
2.1.1	Data flow .....	6
2.1.2	Structure Chart.....	8
3	Detailed Design .....	8
3.1	Global System Architecture .....	8
3.2	Platform .....	8
3.3	Software Architecture.....	9
3.4	Database Design.....	9
3.5	Design I/O.....	9
3.6	Procedures & Users.....	9

3.7 Interfaces ..... 10

3.8 Detailed Class Design ..... 10

3.9 Report ..... 10

# 1 FEASIBILITY STUDY

---

## 1.1 UNDERSTANDING THE PROBLEM

In this section, we try to understand the purpose of the software. The MLAS (Medical Lab Automation System) is intended to automate various medical lab management activities which were earlier done manually. The MLAS will help the management in a feasible generation of various test reports and bills and will help them to keep track of various equipments like test tubes, syringes etc. The Management has the authority to add/edit/remove tests and their normal values. More specifically, the goal of this software is to allow the management to carry out their jobs in a very simplistic manner. It helps the management to easily keep track of bills and reports and equipments. Since everything is done by the computer it is easy to keep a record, to schedule inventory re-stock etc.

## 1.2 SCOPE OF THE PROBLEM

In this section, we scope the problem and find the various functions the software needs to perform.

- Addition of new tests along with their normal values and charges entered by the Management.
- Editing/Removal of already existing tests and their various attributes (normal values, test charges etc.)
- Generate a bill with a unique id when a patient comes for a test (The bill will contain data like name of doctor and patient, date of collection of test report etc.)
- Aid in generation of the test report associated to a particular bill.
- Listing of various tests and their details as and when required by the Management.

## 1.3 ANALYZING STAKEHOLDERS

The stakeholders are –

- Management – Power to edit/add/remove test, list tests, and place order for new stocks, Generate a Bill when a Patient takes a test and Generate Test Report when the patient comes to collect the test report, check stocks and notify Management when shortage of stocks.
- Patient – Power to take test and collect test report. He receives a bill after taking the test and he has to present that bill on or after the specified date and time to collect the test report.

## 1.4 DEFINING ALTERNATIVES

### 1.4.1 Connection between various supervisors (Management) and the software

- The supervisors can list the tests existing in the database and view their details.

- The supervisors can add new tests and their normal values along with the test charge into the database.
- The supervisors can generate a bill with all relevant details and a unique id upon the arrival of a patient who is seeking a test.
- The supervisors can get the generated test report upon the production of the associated bill after the date of collection of test report has arrived.
- The supervisors can edit/remove already existing tests.

#### 1.4.2 Hardware Infrastructure

- The software is designed to run on all Operating systems which support the recent version of java. (JDK 1.7 or a newer release)
- In addition to the java environment support, the central system (one containing all the databases) should also have a proper installation and support for sql database.
- The software is designed both for 32-bit and 64-bit systems.

#### 1.4.3 Software Infrastructure

- The database system used for MLAS can use SQL server, Access, MySQL or Oracle for managing the database.

#### 1.4.4 Technology used

- A 2-tier architecture which comprises only of the client and the database without any interference of the server has been employed. Compatibility with 3-tier architecture which comprises of the client, the server and the database also exists. In case, we can compromise on the graphical aspect of the MLAS, we can use programming languages such as C, C++ instead of Java.

#### 1.4.5 Security

- The MLAS can be designed for various levels of security for different supervisors.
- Instead of using password protection for login into the system, we can also incorporate Face recognition, a Bar Code scanning system, or a thumb impression recognition system for login of the important stakeholders.

### 1.5 DEFINING CRITERIA TO EVALUATE

The primary criteria which are to be kept in mind while evaluating the alternatives are:

- Cost of technology
- Cost of infrastructure
- Lifetime of technology
- Stability of technology

## 1.6 ASSESSMENT OF UNUSUAL CIRCUMSTANCES

- In case the MLAS uses a 3-tier architecture, it need to be hack proof by designing it using the concepts of special algorithms, data encryption and server-based cryptography.
- The design should take care of the fact that the data is not lost in any case, be it software or a hardware failure, system going down or any unusual circumstances that might intervene in between the smooth functioning of the MLAS. For this, we can design a MASTER system which stores the backup of all the data which is fed into the MLAS time to time. This master system would allow us to retrieve data at any point of time and restore the database to its original state.

## 1.7 EVALUATION OF ALTERNATIVES

### 1.7.1 Connection between resident and city corporation offices

If the Patient is given the option to get the medical report from his work place or home, the server has to adopt a 3-tier architecture and incorporate more security modules for guarding against privacy of generated reports. In our problem, we have not been instructed to give the patient this option. Hence, this alternative has lesser priority.

### 1.7.2 Connection between various supervisors and the software

If we use multiple machines, we have to use server and as mentioned above we will have security concerns which in turn will increase the cost of technology. Also, multiple machines will directly increase the cost of infrastructure.

### 1.7.3 Hardware infrastructure

Using an external hard disk to save the database will indirectly imply a backup of the database along with the master system which can be retrieve the data at any point of time desired. Thus, this is better than using the system hard disk to store the data. This also ensures that the software does not consume any space on the system hard disk. The MLAS can be designed for a 32-bit system as well as a 64-bit system. Designing the MLAS for a 64-bit system should be preferred for the graphical interface concerns.

### 1.7.4 Software infrastructure

Any database system can be used for the managing the database of MLAS and all have equal priority.

### 1.7.5 Technology used

If we use the 3-tier system for the implementation of MLAS, we will have security concerns and hence will affect the cost of technology. Thus, this alternative should have less priority and 2-tier system has high priority.

Since we do not deal with algorithm intensive techniques in MLAS, we need not go for programming languages like C++ and Java would suffice.

#### 1.7.6 Security

If the MLAS uses security systems like thumb impression recognition, the cost of technology increases but the security increases. As the privacy of all patients is a major concern for the MLAS design, security should be given a high priority if the 3-tier version is implemented.

### 1.8 REPORT

In the feasibility study, we went through the complete details of the problem. The objectives of the MLAS have been laid out and the various scopes have been discussed in detail. Firstly we understood the complete problem and found the various functions that the software performs such as processing employee's instructions (Bill generation, test report generation etc.), processing and addition of various tests details, updating medical lab inventory. Then, various cases were analyzed with the help of use case diagrams along with the brief description followed by a step-by- step description of each use case. Each of the use cases were supported by the use case diagrams for ease of understanding. Then, the various alternatives were developed keeping in mind the cost and the lifetime of the components the alternative brings with it and hence the advantages and disadvantages were highlighted. These alternatives included the hardware, software, technology, security and many other aspects which form an integral part of the software and which could be incorporated in the MLAS, if desired. The primary criteria for evaluation were expected lifetime, cost, stability, security and stability of the technology. The unusual circumstances like loss of data due to hardware or software failure or hacking were taken care of by certain concepts of data backup, cryptography etc. At last, all the alternatives proposed earlier were analyzed in depth and their advantages and boon to the MLAS were clearly mentioned. A very vivid comparison was made between the MLAS development without the alternatives and with the alternatives if incorporated in the software.

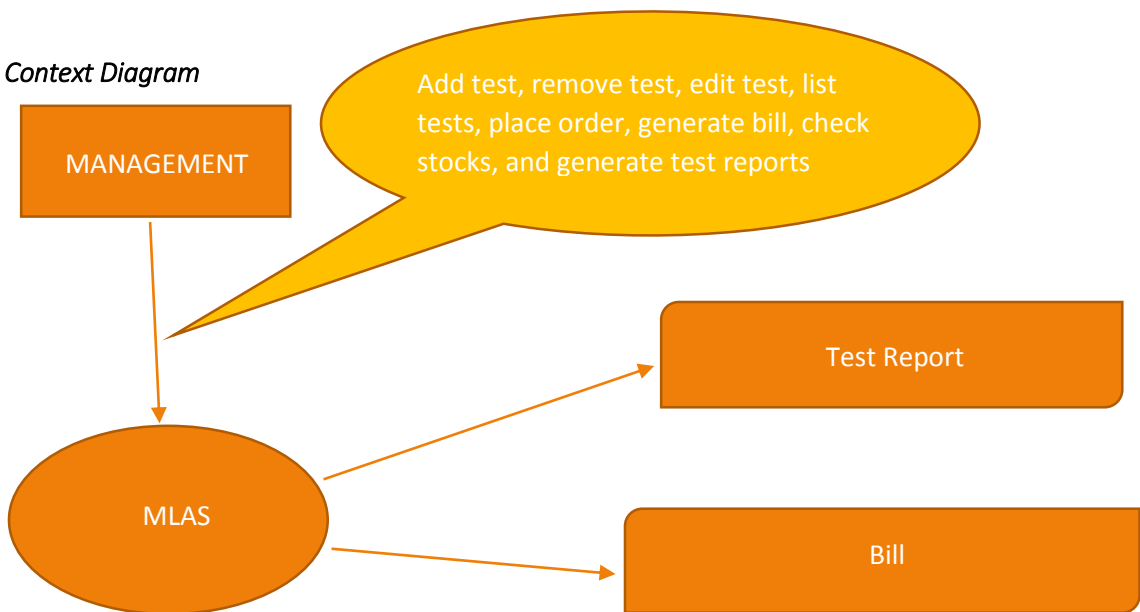
## 2 REQUIREMENT ANALYSIS

---

### 2.1 FUNCTIONAL REQUIREMENTS

#### 2.1.1 Data flow

#### 2.1.1.1 Context Diagram



##### 2.1.1.1.1 Context Diagram

Here we have Management as the external entity. The whole of MLAS is shown as the process.

##### 2.1.1.1.2 Inputs

- Add Test
- Remove Test
- Edit Test
- List Test
- Notify Management
- Check Stocks
- Generate Bill
- Check Stocks

##### 2.1.1.1.3 Process

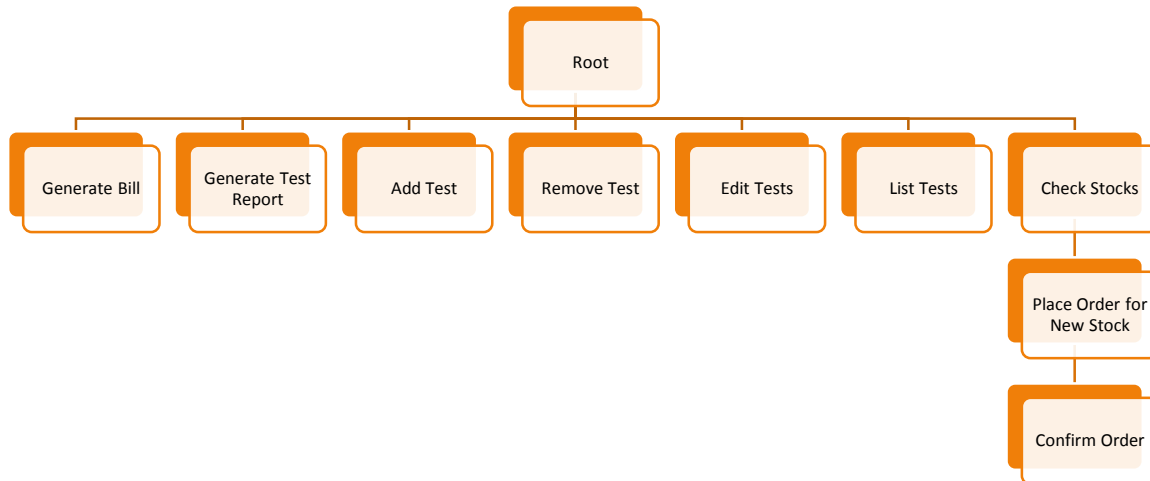
MLAS (Medical Lab Automation Software)

##### 2.1.1.1.4 Outputs

- Test reports
- Bill
- Place order



### 2.1.2 Structure Chart



## 3 DETAILED DESIGN

### 3.1 GLOBAL SYSTEM ARCHITECTURE

The overall system architecture is a 2-tier architecture which includes client at one end and the database at the other. There is no server based middle tier in the software being designed.

### 3.2 PLATFORM

- Minimum system requirements:
  - Hardware requirements

Operating system	Windows XP or later versions, Linux, OS X
Processor	At least Pentium II processor or equivalent
Free Hard Disk space	500 MB
RAM	512 MB

- Software requirements:

MySQL, JDK 1.7 or above

- Recommended system requirements
  - Hardware requirements

Operating system	Windows 7 or later versions, Linux, OS X
Processor	400MHz Pentium III processor or above

Free Hard Disk space	500 MB
RAM	1 GB

### 3.3 SOFTWARE ARCHITECTURE

Object-oriented architecture forms the basis of the MLAS. In this style data representations and their associated primitive operations are encapsulated in an abstract data type or object. The components of this style are the objects—or instances of the abstract data types. Objects interact through function and procedure invocations. Two important aspects of this style are:

- A. That an object is responsible for preserving the integrity of its representation (usually by maintaining some invariant over it)
- B. That the representation is hidden from other objects. Thus the aspects of OOA mentioned justify our choice.

### 3.4 DATABASE DESIGN

The core table of the database is Test. This tracks the bulk of information about the various medical tests. Each field holds a single piece of information about a medical test. A secondary table deals with all the bills. The primary id is the bill id, and each bill is associated with the details of the patient to whom it was issued. A third table deals with the medical lab inventory and keeps tracks of the various items and their numbers and their required quantity.

### 3.5 DESIGN I/O

The Management will be provided with a portal for the addition/editing/removal/listing of various tests. Another portal will be dedicated to inventory management. Initial input will be provided in the form of initial medical lab stocks and the medical tests. Input for the medical tests will be provided in two ways, via a properly formatted file and also via entry of all details pertaining to the test manually. Employees will have to supply input pertaining to the patients in various JTextAreas. The primary output for LMAS is bill and test report. Both of them will be displayed via a dedicated JOptionPane and an option for print will be provided.

### 3.6 PROCEDURES & USERS

- Users:  
The different user classes are:
  - Management – Power to edit/add/remove test, list tests, place order for new stocks, Power to Generate a Bill when a Patient takes a test and Generate Test Report when the patient comes to collect the test report, check stocks and notify Management when shortage of stocks.

- Patient – Power to take test and collect test report. He receives a bill after taking the test and he has to present that bill on or after the specified date and time to collect the test report.
- Procedures:  
Refer to Sequence Diagrams.

### 3.7 INTERFACES

The central system runs the database application, for example, SQL\*Plus, that accesses database information and interacts with a user. A server can be incorporated as an interface to the queries from the client side to the database. The user interface comprises of java swing components. The interface will be mainly in the form of jButtons.

### 3.8 DETAILED CLASS DESIGN

Refer to Class Diagram.

### 3.9 REPORT

Under the detailed design section of the software design, the global system architecture was discussed. The MLAS has a 2-tier architecture comprising of the client and the database with no server. Then the platform requirements for the MLAS was discussed in terms of the operating system, the processor required, the minimum and recommended hard disk space and RAM requirements, etc. The software architecture of the MLAS was later stated to be of the object-oriented type using JAVA as the core technology. The important aspects of OOP used for the MLAS are data abstraction and the preservation of integrity of the software.