Software Requirements Specification

For

Diagnostic's Medical Test Information System

VERSION 1.5

Course title: Object Oriented Software Development
Course code: SWE 331

Prepared By:

S. Rayhan Kabir (ID: 133-35-561)
Pankoz Nandi (ID: 133-35-574)
Sharmin Sultana (ID: 123-35-335)
Mahdi Bin Shahjahan Talukder (ID: 131-35-371)

Supervised by:

MD. MAHMUDUL HASAN
Lecturer
Department of Software Engineering
Daffodil International University

Date of Submission: **27.06.2016**

Department of Software Engineering

Contents

1. Introduction

"Diagnostic's Medical Test Information System" is a simple software which will keep the detailed information of the patient's medical test of Diagnostic medical center. This article is about medical tests for diagnostics. In medicine, a diagnostic's medical test is any kind of medical test performed to aid in the diagnosis or detection of disease. A diagnostic test is a procedure performed to confirm, or determine the presence of disease in an individual suspected of having the disease, usually following the report of symptoms, or based on the results of other medical tests. So, we've tested this for Diagnostic Center and we hope *"Software of Diagnostic Medical Information System"* will helpful for collect medical test at any Diagnostic Center. That's why; we think it will make some revolution to our real life.

1.1 Purpose

The purpose of this document is to collect, analyze and define the high-level needs and features of the "Diagnostic's Medical Test Information System". It focuses on the capabilities and facilities needed by the stakeholder who are work in Diagnostic Center and target users the Doctors, Receptionist, Director, Chairman etc. of the Diagnostic Center. The details of what all are the needs Diagnostic's Information *System* fulfils these needs are detailed in the use-case and supplementary specifications.

1.2 Product Scope

In this section we are going to list the functionalities which will be provided and will not be provided after the project completion.

The "*Diagnostic Medical Test Information System"* that is to be developed provides the Patient, Doctors, and Worker of the Diagnostic Center with detail information about Medical Test. The System is supposed to have the following features. The system provides medical test result or information service to the Patient, Doctor and some worker who are related with work of medical test. The system provides the stuffs with the option to check their medical test whenever needed all through the day. The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criteria for the system are based in the level up to which the features described in this document are implemented in the system

1.3 References

- Ian Sommerville., Software Engineering :6th Edition ,Pearson Education.
- IEEE Standard 830-1993, "IEEE Recommended Practice for Software

- Requirements Specifications"
- Phillips, D.: The Software Project Manager's Handbook, IEEE Computer Society, 2000
- Pressman, Roger S., software Engineering: A Practitioner's Approach_,4th Ed, McGraw-Hill.

2. Overall Description

This section gives overview of current systems available in the tannery industry and the product perspective of the "Diagnostic's Medical Test Information System" (DMTS) when released.

2.1 Product Perspective

"Diagnostic's Medical Test Information System" (DMTS) is a medical test management systems which depend on patient by recording medical information. The features that are described in this document are used in the future phases of the software development cycle. The features described here meet the needs of all the users. The success criteria for the system are based in the level up to which the features described in this document are implemented in the system.

2.2 Product Functions

2.2.1 Administrators:

- The member should be provided with the updated information about the software catalog.
- Can accept or reject a new user to the organization policy.
- Add and edit medical test information and arrange patients by medical test categories and patient's code number.

2.2.2 Normal Users:

- The member should be provided with the updated information about the medical test catalog.
- Members have the ability to search through medical test by code.
- The owner, manager, technologist, chemist, super visor users can monitoring it.

2.2.3. Medical Test Counting and Management

In this Software project we include counting of medical test in every process step, start to end. For this reason user and owner can easily manage the medical test information.

2.2.4 Patient's Medical test Input & Output

In this system different patient's different medical tests input by patient ID and get output by searching patient's ID.

2.2.5 User to User Data Entry

Which employee or user edit, update, delete or entry data or use this software, that all include this *software*.

2.2.6 Development & Environment

This subsection describes the necessary methods tools and technology used in this project. The following table shows the environment used in this project in different milestones and its purpose.

2.3 User Classes and Characteristics

Administrator users can use this product; there is no specific user hierarchy level among the administrator to access it. This software product supports user level security to access mails. There are two levels of authentication used software.

At the first level of authentication user has to give his/her login and password so that he/she will be able to access.

At the second level of authentication, user has to give authentication details about the mail account/accounts so that E-Mail by software admin.

User Classes:

- Manager Admin.
- Assistant of Doctors or Technologist.
- Receptionist.
- Computer Operator.

2.4 Operating Environment

This DMTS software can run in Windows 8-10, Linux based Operating System (OS). This software need at least duel core computer. A MySQL server need install in this system or computer. This I is a desktop based software. So there no need any internet connection.

2.4.1 Hardware

- Processor: Intel based 166 MHz
- RAM: 128 MB
- Hard-disk space: 4 Gb

- GUI support needed

.

2.4.1 Software

- Operating System: Windows or *NIX.
- MySQL Server

2.5 Design and Implementation Constraints

- The information of all patient, all medical test must be stored in a database that is accessible by the software.
- MySQL Server will be used as SQL engine and database.
- This DMTS Software is running 24 hours a day.
- Users may access DMTS from one computer.
- Users must have their correct usernames and passwords to enter into the software.

2.6 Assumptions and Dependencies

- The product needs the following third party products.
- MySQL server to store the database.
- Java programming language to develop the Product.
- Existence of JRE &JDK service runs this software program.
- This Software interface must be friendly and easy-to-use.
- The search mechanism should be simple and fast.

3. External Interface Requirements

In this section, the general factors that affect the product will be described in general. Here the requirements will be presented as a miscellaneous concept. Later these rough details will be analyzed, specified, organized and then categorized into functional and non-functional requirements.

3.1 Identification of Stakeholders

The stakeholders of DMTIS involve those who are directly or indirectly benefitted to the system. We have identified the following list of suspects who can contribute to this fact.

The stakeholders for this system are:

- Chairman.
- Directors.
- Manager Admin.
- Doctors.
- Medical Technologist.
- Chemist
- Patient.
- Assistant of Doctors or Technologist.
- Receptionist.
- Computer Operator.

3.2 Collaborating multiple viewpoint

- Admin panel should be provided for checking the uploaded information.
- Individual and full site statistics will be provided for all kind of searching and welfare of all the stakeholders
- Whether the codes of the projects should be open source or not is the decision of the developers. We are not going to give any concern on this. If the developers think to provide projects code or other information, they may upload zip file of all the things.

3.3User Interfaces

User or admin can use this software by GUI (Graphical User Interface). But in this documentation CMD interface. We will show this in our software product.

Designing the visual composition and temporal behavior of a GUI is an important part of software application programming in the area of human–computer interaction. Its goal is to enhance the efficiency and ease of use for the underlying logical design of a stored program, a design discipline named usability. Methods of user-centered design are used to ensure that the visual language introduced in the design is well-tailored to the tasks.

- **This is Main Menu**

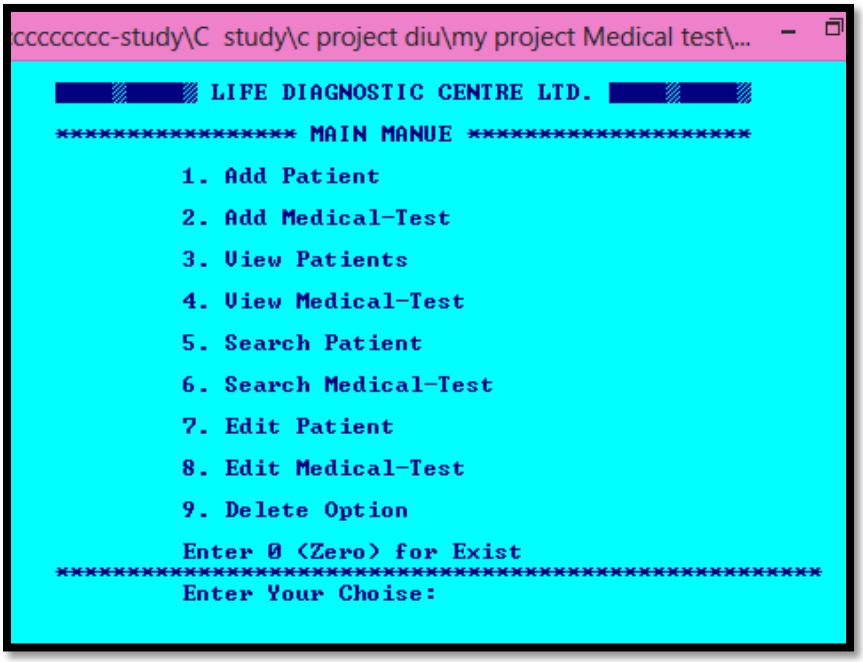

Figure 1: Main menu

- **This is Add Medical Test**

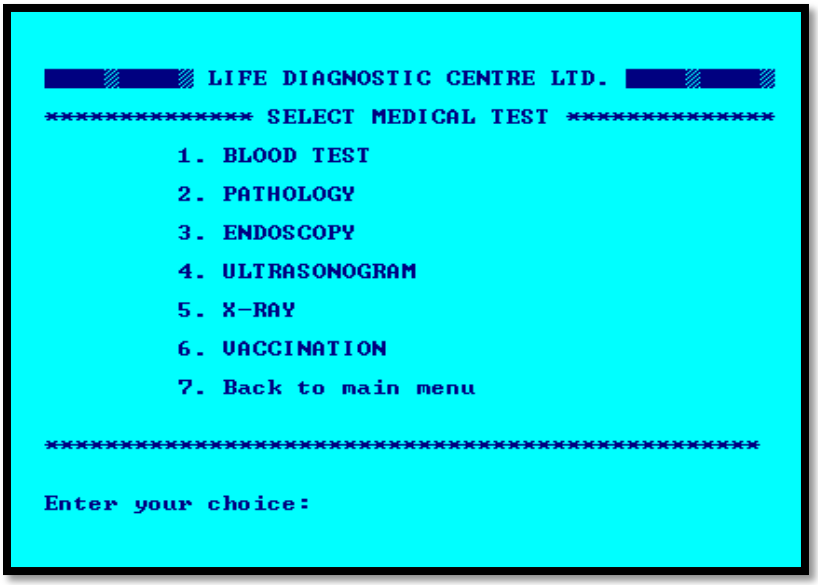

Figure 2: Add Medical Test

- **This is Add Pathology Test**

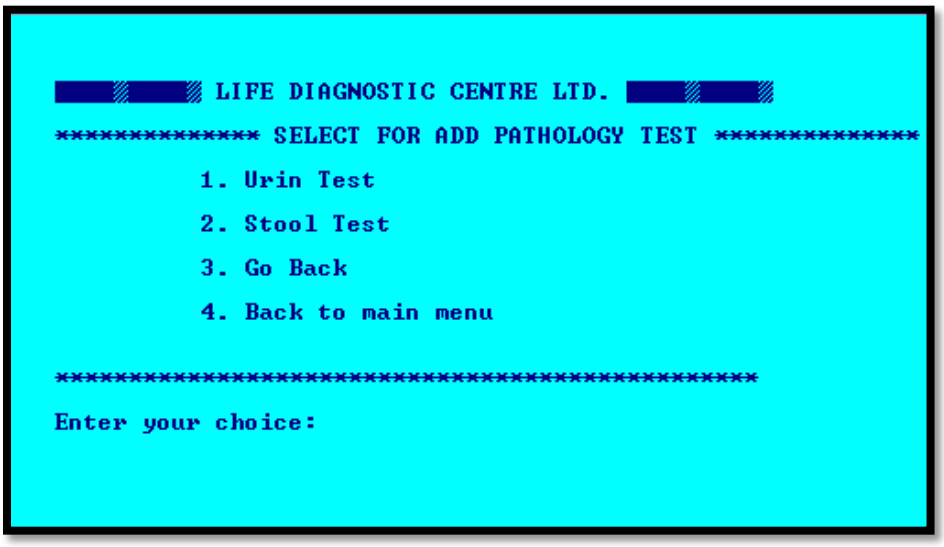

Figure 3: Add Pathology Test

- **This is View Medical Test**

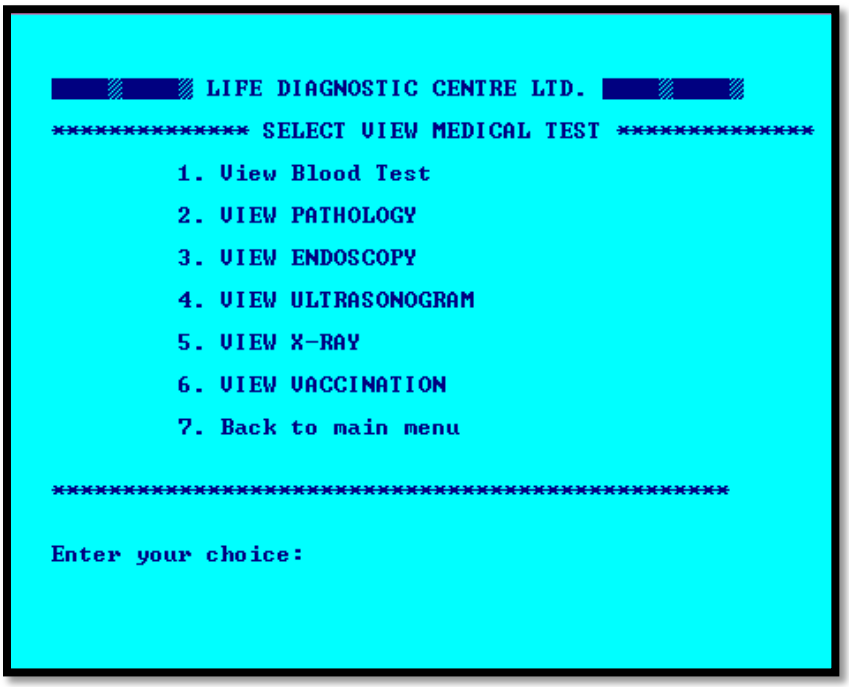

Figure 4: View Medical Test

- **This is Search Option**

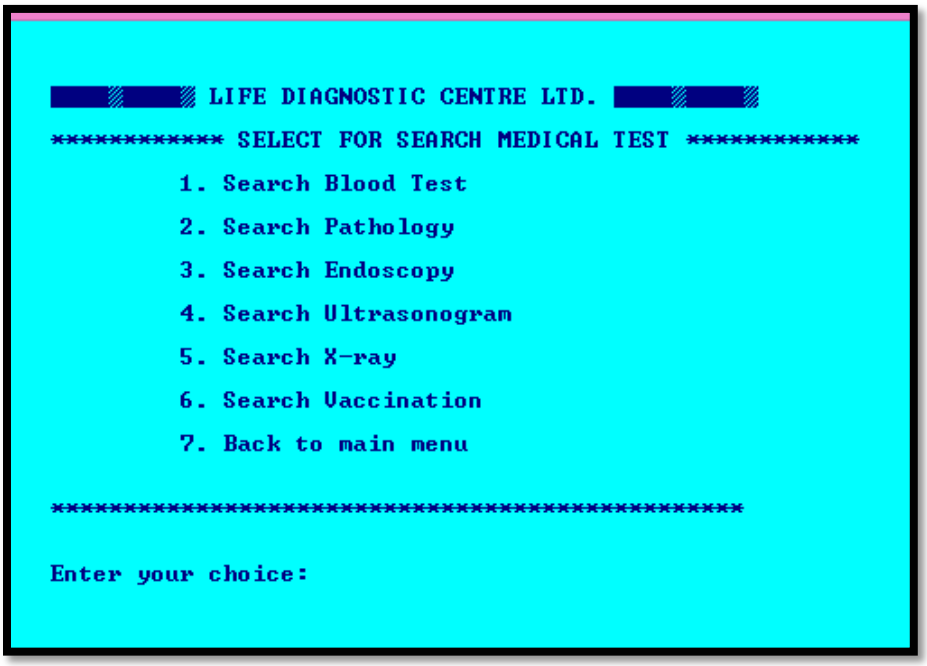

Figure 5: Search medical Test

- **This is Edit Medical Test**

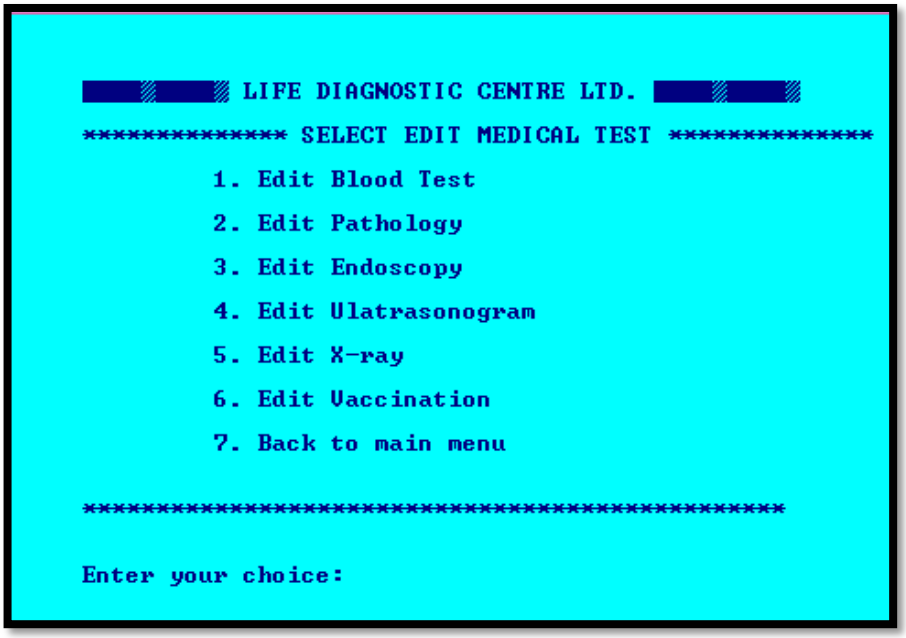

Figure 6: Edit Medical Test

3.4 Hardware Interfaces

This subsection describes the logical and physical characteristics of each interface between the software product and the hardware components.

The basic hardware components necessary to run this product are –

- Computers with available web service
- A MySQL server to host the complete Database of This software
- Basic hardware configuration at the server side to run SQL database 2015 such as 1GB ram, 2 GHz processor etc.

3.5 Software Interfaces

This subsection lists the tools, libraries and any other software components required for the project.

The following language tools, libraries and protocols are required to design and develop the project:

- Java FX 8: In order to design theSoftware.
- Java programming Language: To develop the website and the business logic of the product.
- MySQL server 2015: To keep the database of all the projects and the users.

3.6 Functional Requirement

1. Add Patient
2. Add Medical Test
3. Blood Test
4. Pathology Test
5. Urine Test
6. Stool Test
7. Endoscopy
8. Ultrasonogram
9. Vaccination
10. X-ray Report
11. View Patient
12. View Medical Test
13. Search Patient
14. Search Medical Test
15. Edit Patient
16. Edit Medical Test

17. Delete Option

3.7 Non Functional Requirement

- **Speed**: Take a little time to execute the full program.
- **Platform compatibility**: Can work on any OS (operating environment) like Windows, Linux and Mac with Code Block, Turbo C etc.
- **Usability:** For being simple and easy coding it becomes understandable to all especially to the students. They can easily able to use this system and got best feedback.
- **Backup:** This system is till now under processing and for any kind of disaster attack we keep a backup of our system for recovery.
- **Operation:** It can able to save passenger's information and admin panel data.
- **Effectiveness:** If any one give any input this system quickly execute all the data and give expected result.
- **Efficiency:** It can work on any platform and it can be handled easily by anyone.
- **Extensibility:** We can able to add any features and any more option at any time with our system.

4. Development & Environment

This section describes the necessary methods tools and technology used in this project. The following table shows the environment used in this project in different milestones and its purpose.

Table 1: Development & Environment

Item	Applied for
Methods	
Use Case	Requirements capturing
Sequence Diagram	Requirements capturing & Software Architecture
Tools	
Rational Rose	Design
Languages	
UML	Design
Java	Interface & Program Coding
C	…
SQL	Database

5. Quality Assurance

1. Design, develop and support user interface testing application and regression testing software.

2. Test software applications for reliability and stability.

3. Develop test plans, QA processes and test cases for product management and software development teams.

4. Implement testing programs to perform quality assurance on database applications including negative testing and usability.

5. Lead projects to perform quality assessments that expose security flaws, reveal defects and identify areas of optimization.

6. Design automated testing scripts with oversight for deployment of tests by QB analysts.

7. Create quality assurance documentation and reports using a variety of diagnostic tools.

8. Develop software testing policies, best practices and guidelines.

9. Maintain database with product defects, user reviews, survey information and functional improvements.

10. Coordinate with product development teams and software engineers to recommend solutions to maximize performance and efficiency.

6. Security Aspect

Component-structured software is composed from components which are independently created, combined, and deployed. The high number of principals is a reason for more subtle security risks than in monolithic programs. In order to solve this problem we will develop a formal security model for component-structured in this software. Moreover, we are developing methods and tools for securing components and applications against hostile attacks.

7. Analysis Models

For design purposes, some models are given below which will provide a clear view of the software to be designed. Therefore, each model is traceable the SRS's requirements.

7.1 Use Case Diagram

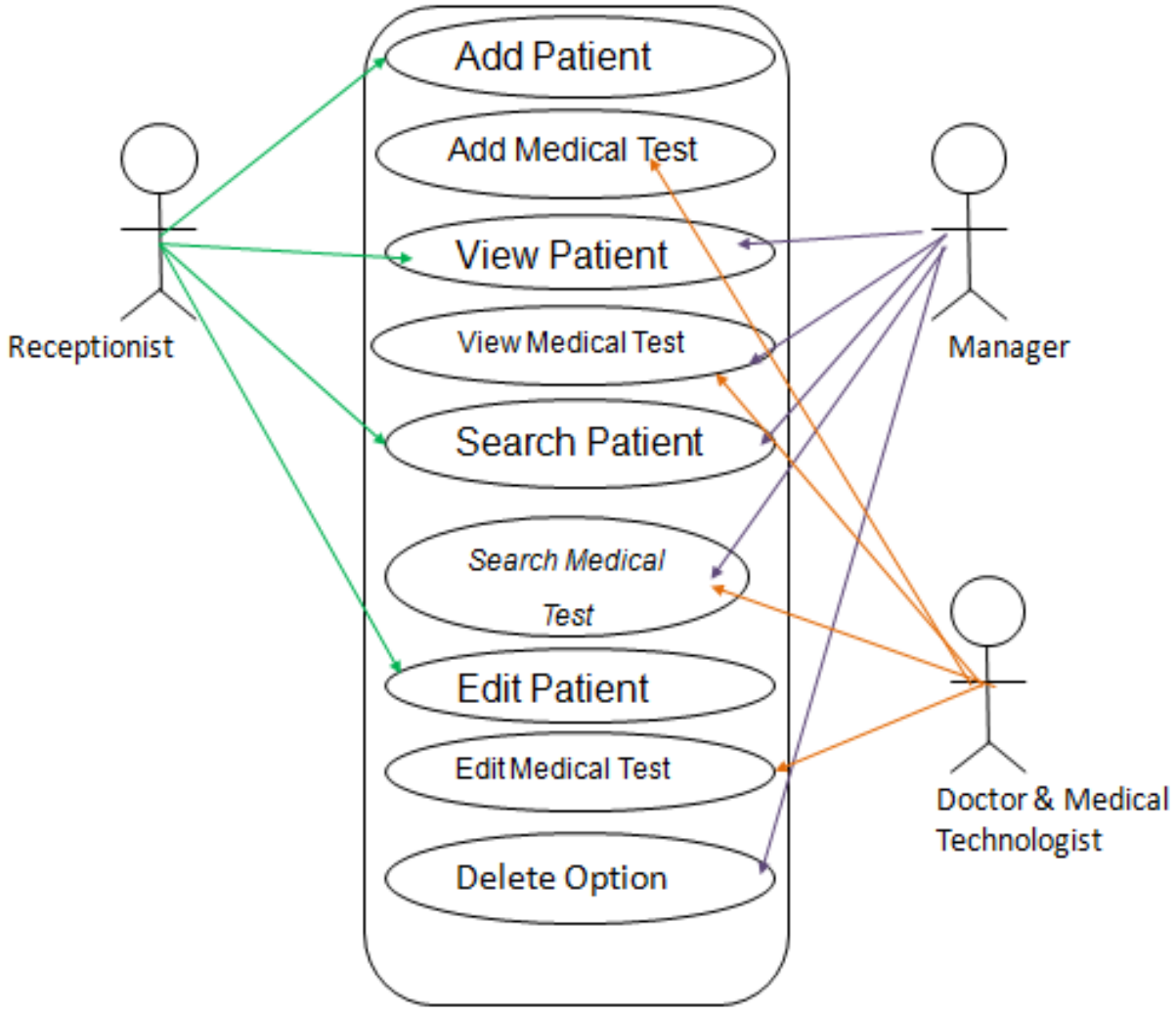

Figure 7: Use Case Diagram

7.2 Activity Diagram

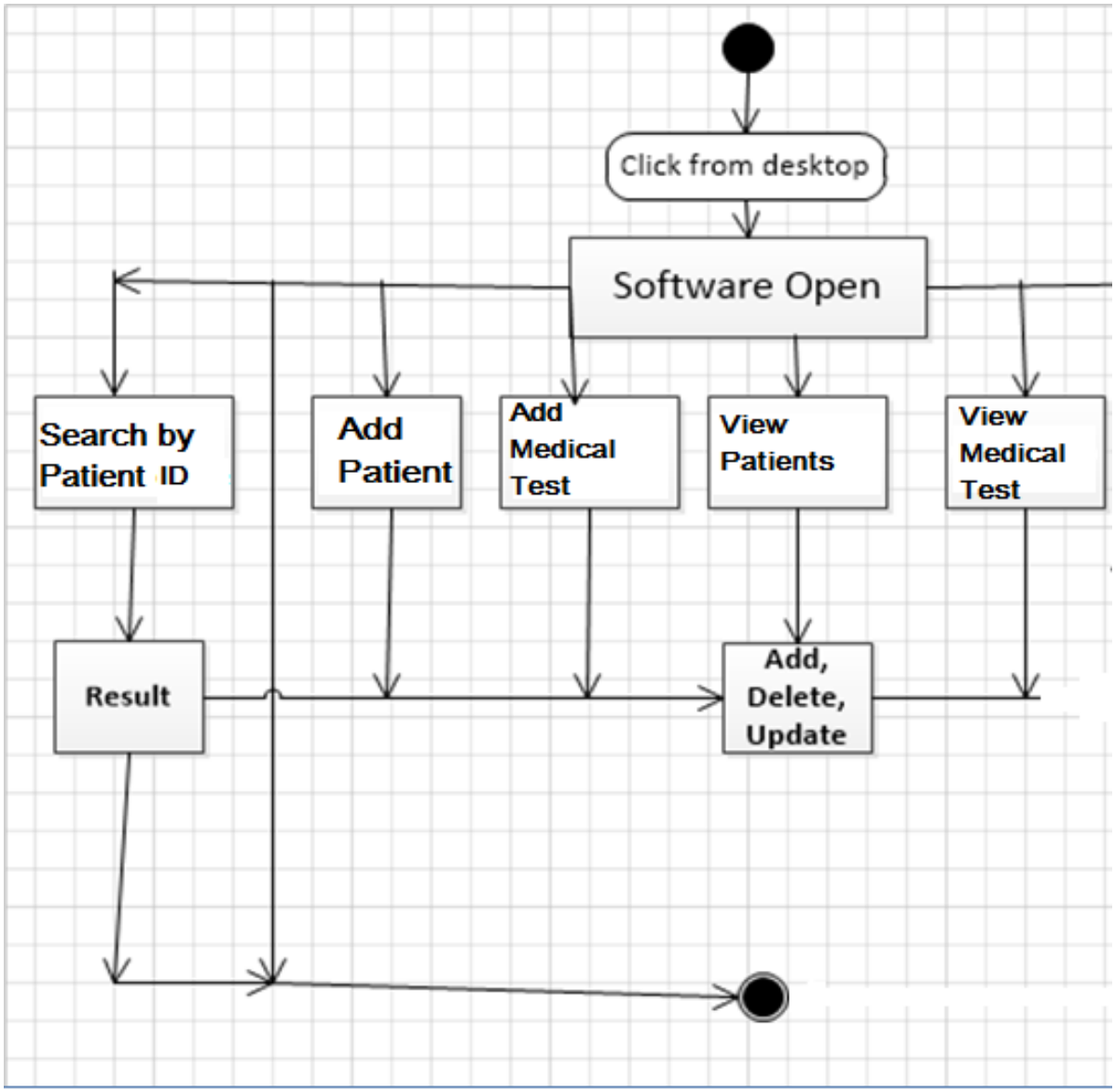

Figure 8: Activity Diagram

7.3 Data Flow Diagram

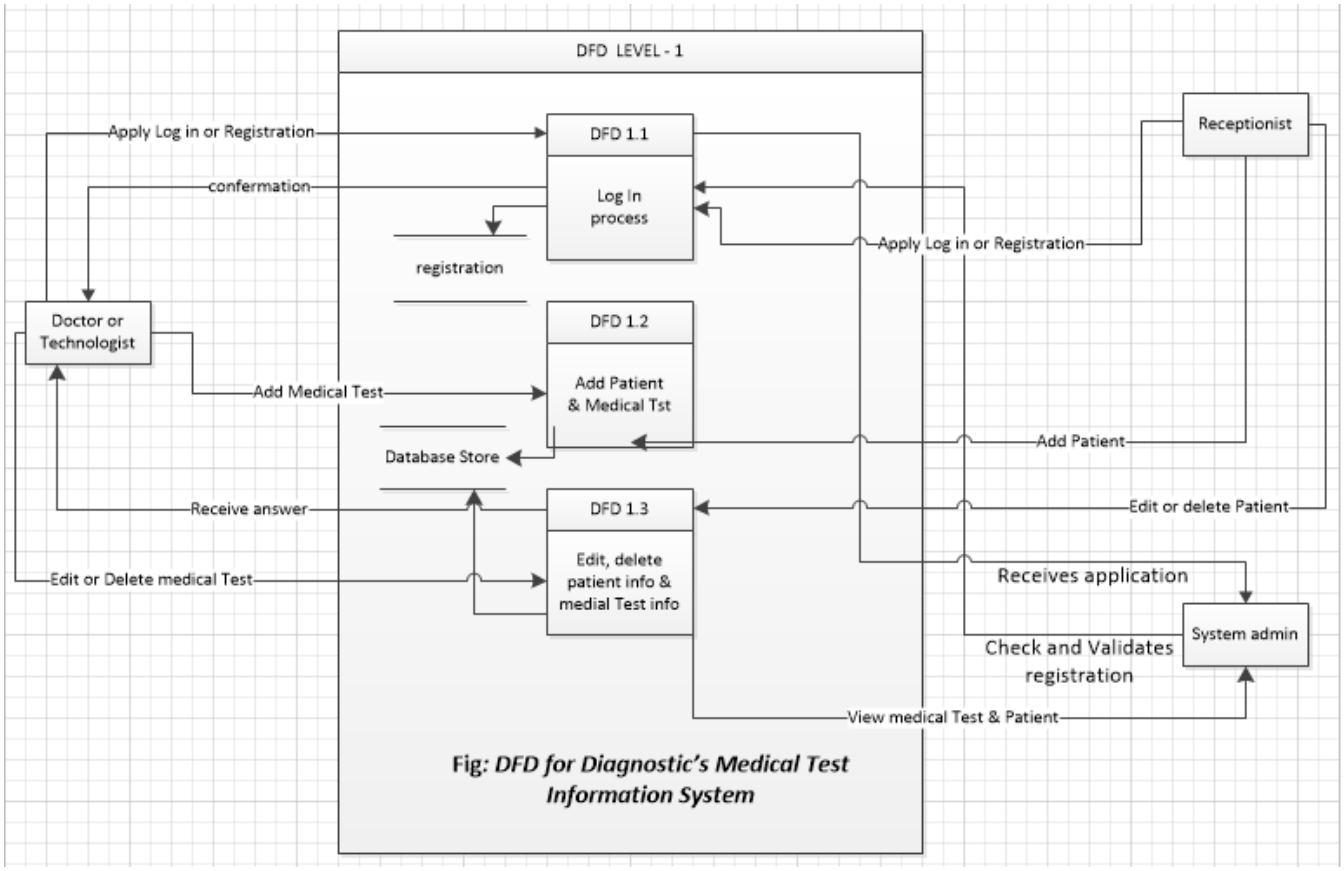

Figure 8: Data Flow Diagram

8. Change Management Process

Identify and describe the process that will be used to update the SRS, as needed, when project scope or requirements change. Who can submit changes and by what means, and how will these changes be approved.

9. References

[1] K. K. Breitman, M. A. Casanova och W. Truszkowski, Semantic Web: Concepts, technologies and applications. London: Springer, 2007.

[2] W. Terkaj, T. Tolio och A. Valente, "Focused flexibility in production Systems," in Changeable and Reconfigurable Manufacturing Systems, H. A. ElMaraghy, Ed. London: Springer, 2009. pp. 47-66.

[3] Y. Zhong, B. Shirinzadeha, G. Alicib and J. Smith, "An autowave based methodology for deformable object simulation," Computer-Aided Design, vol. 38, no 7, pp. 740-754, July 2006.

A. Appendices

Appendices may be used to provide additional (and hopefully helpful) information. If present, the SRS should explicitly state whether the information contained within an appendix is to be considered as a part of the SRS's overall set of requirements.

Example Appendices could include (initial) conceptual documents for the software project, marketing materials, minutes of meetings with the customer(s), etc.