Diagnostic's Medical Test Information System

VERSION 1.5

Diagnostic's Medical Test Information System

Submitted 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:

S. Rayhan Kabir
Project Manager of this Project
Student
Department of Software Engineering
Daffodil International University

Submitted To:

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

Daffodil International University

Department of Software Engineering

Date of Submission: **22.06.2016**

DECLARATION

We declare that this project “C.I.M.S” has been submitted to Software Engineering Department of Daaffodil International Uiversity to fulfill the requirements for the Object Oriented Software Development (OOSD)program in 9th Semester. This project work has been done under the guidance of MD. Mahmudul Hasan, Lecturer and Professor Dr. Touhid Bhuiyan , Department of Software Engineering, Daffodil International University.Though we may extend the project in future, this is certified that we did this project under the course SWE-331 Object Oriented Software Development and it has not been submitted elsewhere for the requirement of any degree or any other purposes.

Project Members

S. Rayhan Kabir	133-35-542	——————————-
Pankoz Nandi	133-35-543	——————————-
Sharmin Sultana	133-35-563	——————————-
Mahdi Bin Shahjahan Talukder	133-35-567	——————————-

Signature and Date

Project Supervisor

_____

MD. Mahmudul Hasan
Lecturer of SWE Deapartment

Approved By

——————————-

Dr. Touhid Bhuiyan
Head of SWE Department

Acknowledgement

We express our heartfelt thanks to MD. Mahmudul Hasan and Dr. Touhid Bhuiyan for providing us all the support during the development procedure of D.M.T.I.S. The patience that they showed with us really helped us a lot. Without them the application shouldn't be residing in its present state. The project idea was completely a new one and she kept her trust over us.We are grateful to MD.Mahmudul Hasan, Course Coordinator SWE-331 Design Patterns, who helped us by sharing his development knowledge throughout out the whole way of developing the system. Especially during the bug resolving procedure we were not able to do much without his help and finally we are grateful to all the Faculty Members and Class Mates of Software Engineering Department whose feedbacks have helped us to make our system better

Abstraction:

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.

Interpretation of diagnostic tests should always take sources of inaccuracy and imprecision into account. Sources of inaccuracy and imprecision of diagnostic tests may broadly be categorized as:

- Physical sources within the diagnostic test taking itself
- Interpretational sources of the resultant data in relation to the target condition. Such sources include conversion of continuous values to binary ones (creating artificially binary values), such as designating a blood test for prostate specific antigen as "positive" when having reached a certain cutoff value, which is generally less accurate than considering the value itself.

Proper evaluation of a diagnostic test involves the use of statistical analysis. In this context the test is referred to as a classification rule for a binary classifier. The test is then compared to a gold standard test to assess the quality.

CONTENTS

Chapter 1
PROJECT PROPOSOAL

1.1 Introduction

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 Background of Proposed Project

The purpose of this document is to describe the collect and include of the information of the Medical Test of the patient. The Document captures high-level requirements and design constraints, which gives the reader an understanding of the Information of Medical Test Handling System of Patients. Requirements Specification defines and describes the operations, interfaces, performance, and quality assurance requirements of the Medical Test Information System. The document also describes the nonfunctional requirements such as the user interfaces. It also describes the design constraints that are to be considered when the system is to be designed.

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

The stakeholders for this system are:

- Chairman.
- Directors.
- Manager Admin.
- Doctors.
- Medical Technologist.
- Chemist
- Patient.

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

1.3 Objectives of the Project:

Diagnostic Center a kind of organization where find medical test of the patient. Diagnostic organization only finds and identifies the medical report or result of patient. The software system is like a spiral system and their service.

- 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 themedical test information.
- 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.
- User to User Data Entry
  Which employee or user edit, update, delete or entry data or use this software, that all include this *software*.
- **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.

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	Web interface
C++	…
SQL	Database

The medical profession requires the ability to provide diagnostic medical imaging utilizing critical thinking skills to make judgments in the process. Medical professionals who must possess high level skills in diagnostic techniques under the guidance of a licensed physician. A medical software is responsible for providing excellent patient care and gathering adequate data necessary for diagnoses to be determined.

1.4 Expected Outcomes

Most diagnostic tests are conducted on the living; however, some of these tests can also be carried out on a dead person as part of an autopsy. Some of the diagnostic tests are parts of a simple physical examination which require only simple tools in the hands of a skilled practitioner, and can be performed in an office environment. Some other tests require elaborate equipment used by medical technologists or the use of a sterile operating theatre environment. Some tests require samples of tissue or body fluids to be sent off to a pathology lab for further analysis. Some simple chemical tests, such as urine pH, can be measured directly in the doctor's office.

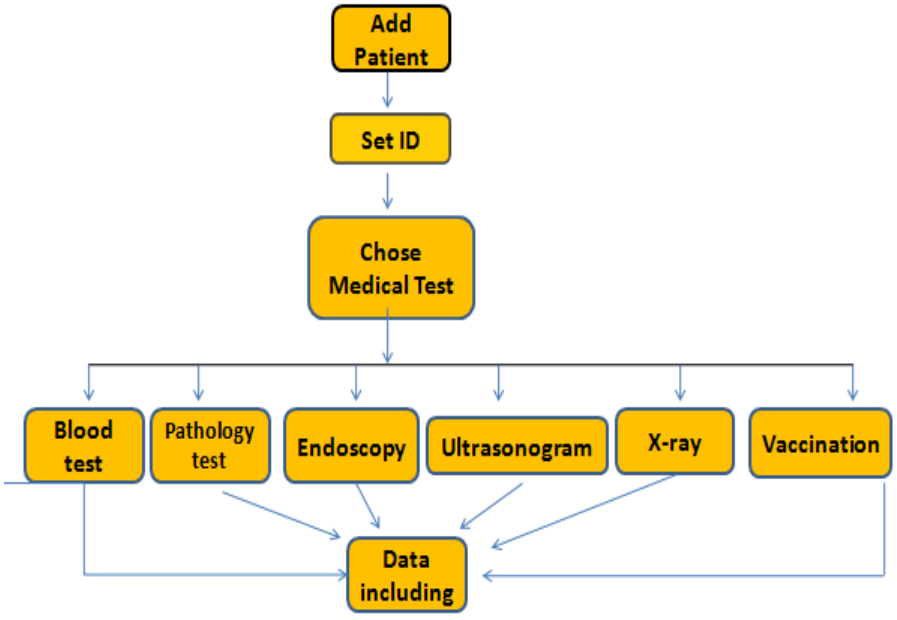

Figure 1: Expected outcome process

The medical profession requires the ability to provide diagnostic medical imaging utilizing critical thinking skills to make judgments in the process. Medical professionals who must possess high level skills in diagnostic techniques under the guidance of a licensed physician. A medical software is responsible for providing excellent patient care and gathering adequate data necessary for diagnoses to be determined.

Graduates of the project will be able to perform, at minimum, the following objectives:

Demonstrate critical thinking skills during the performance of medical procedures to provide optimum diagnostic services in a information system.

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

1.5 Proposed Activities

This document listed all the capabilities of count the Information and Result of Medical Test in the Diagnostic Center.

- **Expected Main Menu:**
  In this figure we just show this option of main menu will include in our project.

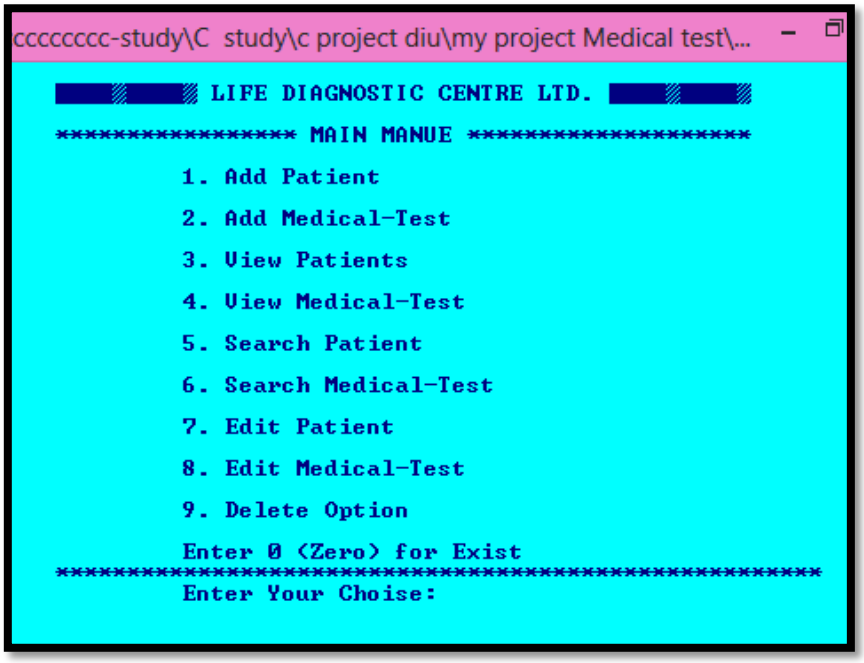

Figure 2: Expected Main Menu

1.6 Estimative Budget

This subsection proposes an approximate budget for this software project. Budget is proposed for each milestone by considering different aspects. Final amount is the total budget for this project. The budget is shown in the table below:

Table 2: Budget

Category	Budget for Period in kUS$					
	M0-M1	M1-M2	M2-M3	M3-M4	M4-M5	M5-M6
Human Resources (internal)	10000	10000	10000	20000	30000	8000
Human Resources (external)	8000	8000	8000	10000	10000	5000
Purchases (COTS)						
Equipment					5000	
Premises						
Tools					5000	
Travel costs						
Training						
Review activities						
Other						
Total	18000	18000	18000	30000	40000	13000

1.7 Project Timelines

This section described the project timeline. It's working approach, its working selection process for development etc. An approximate budget is also proposed in this chapter section.

- **Work Breakdown Structure**

  Work breakdown is the decomposition is the whole project in several parts. In this software project is divided into several millstones.

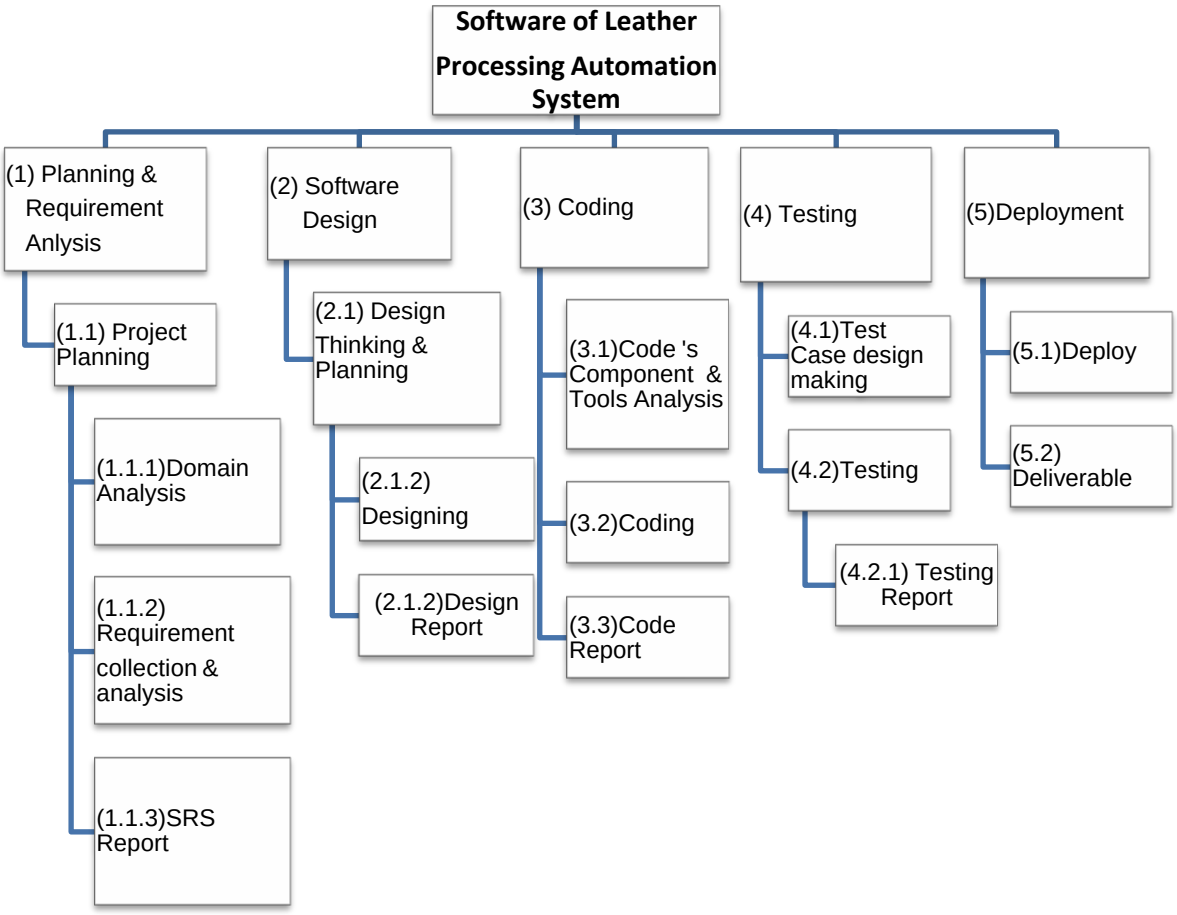

Figure 3: Work Break-down Structure

- **Schedule & Milestone**

Achieving the above the full work the project is divided into several milestones. A general view of this project milestone is given in the table below:

Table 5: Schedule and Milestone

Milestones	Description	Milestone Criteria	Planned Date
M0	Start Project	Budget Release	5 -6-2016
	Software Project goals and scope defined	PRS or SRS reviewed Stakeholders identified Impl. Proposal reviewed	7-3 -1016
M1	Start Planning		7 -6 -2016
	Life Cycle of Tannery industry defined	Scope and concept described	10 -6 -2016
M2	Start Execution		10-7 -2016
	Software Requirements Defined	Requirements agreed, project plan reviewed, resources committed	15-7 -2016
M3	Confirm Execution		15-8-2016
	Software Architecture & design	Architecture reviewed and stable	25-8-2016
M4	Start Introduction		25-8-2016
	Software Coding & Development	Coding of new functionality finished, Draft documentation	26-8-2016
M5	Release Product		26-8-2016
	Software Testing & Implementation & others	Product system tested, documentation reviewed	30-8-2016
M6	Close Project		30-8-2016

1.8 Summery

We believe such system will help to the Diagnostic Center organization for take care the Information of medical test of patient. So A Diagnostic's Medical Test Information System is a kind of medical software procedure performed to detect, diagnose, or monitor diseases, disease processes, susceptibility, and determine a course of treatment.

Bibliography

The Top Technical Resources for Java Software Developers Larger Cover Java software Development Bibliography.

O'Reilly Media, 2011.
Java Application Development For Dummies. Histon Wiley / For Dummies, 2015.
saf [2011] bar [2015]

Chapter 2

SOFTWARE REQUIREMENT SPECIFICATION

2.1 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.

2.1.1 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.

2.1.2 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.

2.1.3 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.

2.2 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.

2.2.1 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.

2.2.2 Product Functions

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.

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.

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.

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.

User to User Data Entry

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

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, Linux, Mac OD.
- 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.

2.7 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.

2.7.1 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.

2.7.2 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.

2.7.3 User 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**

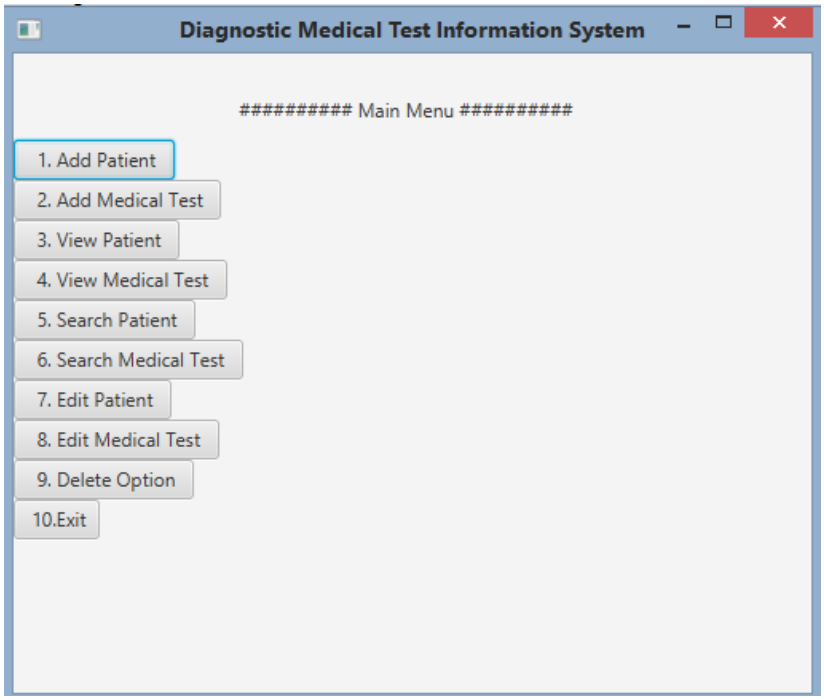

Figure 4: Main menu

- **This is Add Medical Test**

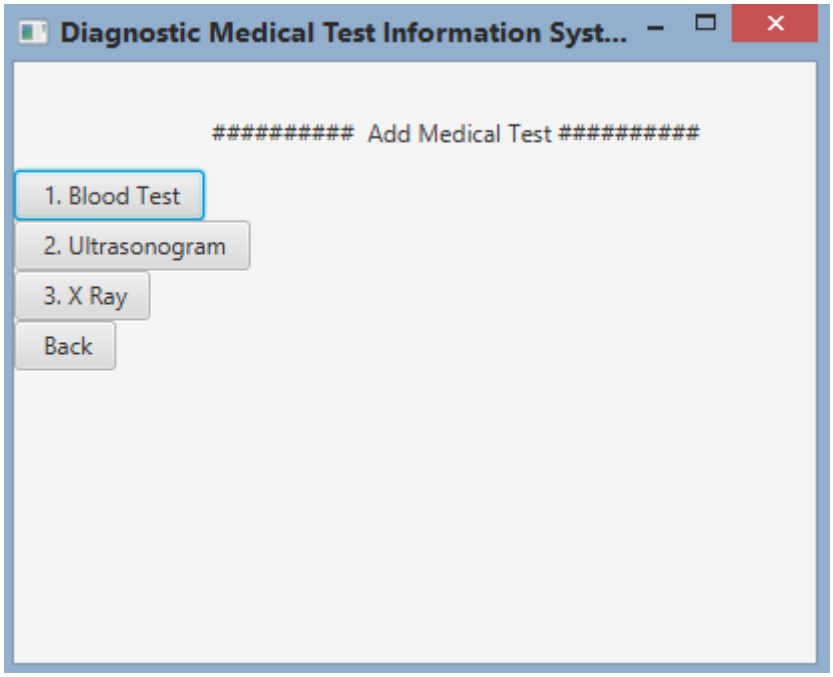

Figure 5: Add Medical Test

- **This is Add Blood Test**

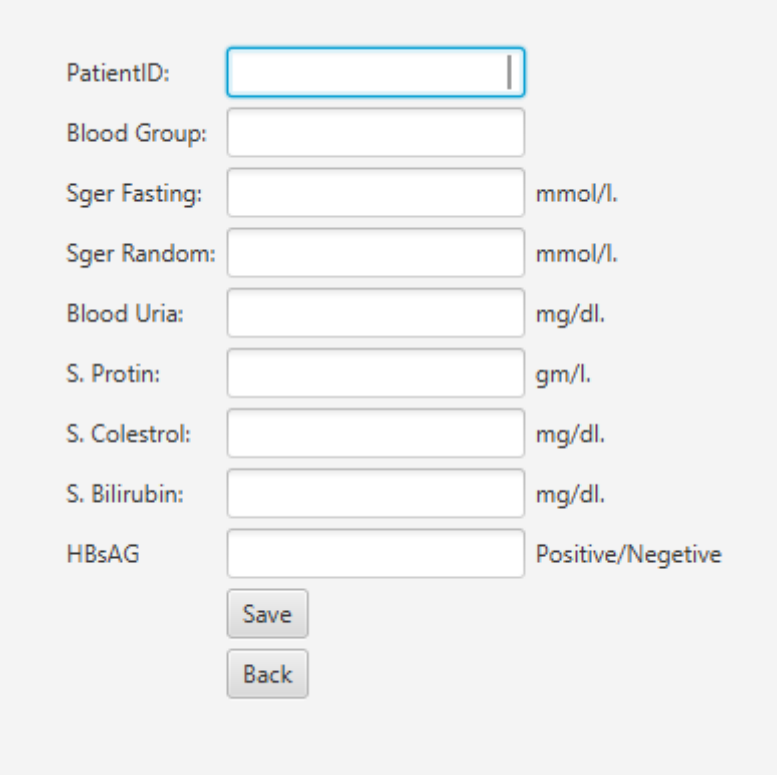

Figure 6: View Medical Test

- **This is patient information part**

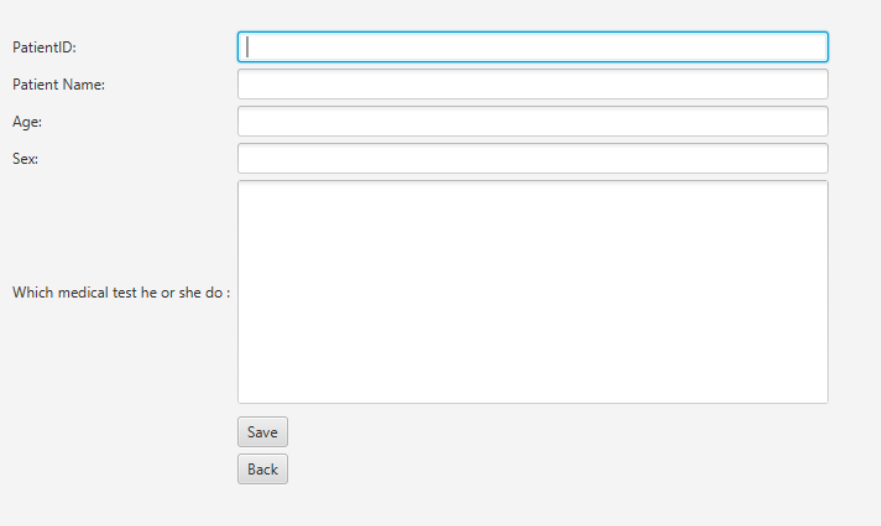

Figure 7: Add Patient Information Part

2.7.4 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.

2.7.8 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.

2.8 Functional Requirement

18. Add Patient
19. Add Medical Test
20. Blood Test
21. Pathology Test
22. Urine Test
23. Stool Test
24. Endoscopy
25. Ultrasonogram
26. Vaccination
27. X-ray Report
28. View Patient
29. View Medical Test
30. Search Patient
31. Search Medical Test
32. Edit Patient

33. Edit Medical Test
34. Delete Option

2.9 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.

2.10 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

Item	Applied for

2.11 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.

2.12 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.

Chapter 3

DESIGN AND IMPLEMENTATION

3.1 Introduction

The goal of this chapter is to provide visual representation of the whole structure, system or components to the developers in a diverse way. The requirements specification document was followed to create those diagrams and provide components. Later, the designs were rectified by design engineers. These designs were then followed by developers to provide the software.

3.2 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.

3.2.1 Use Case Diagram

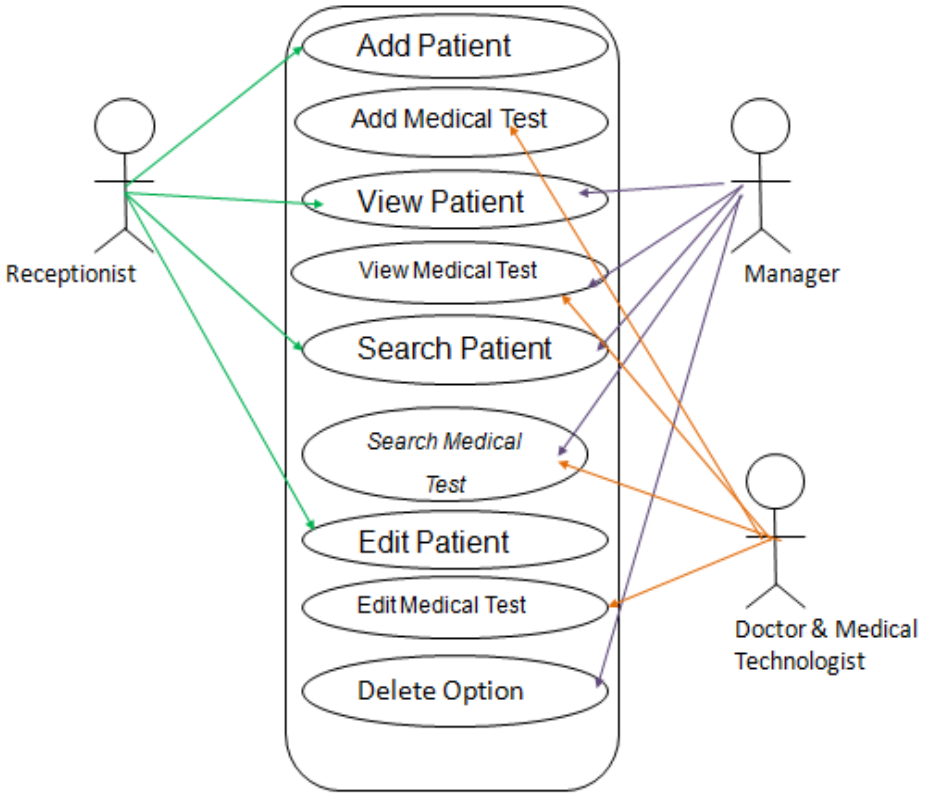

Figure 8: Use Case Diagram

3.2.2 Activity Diagram

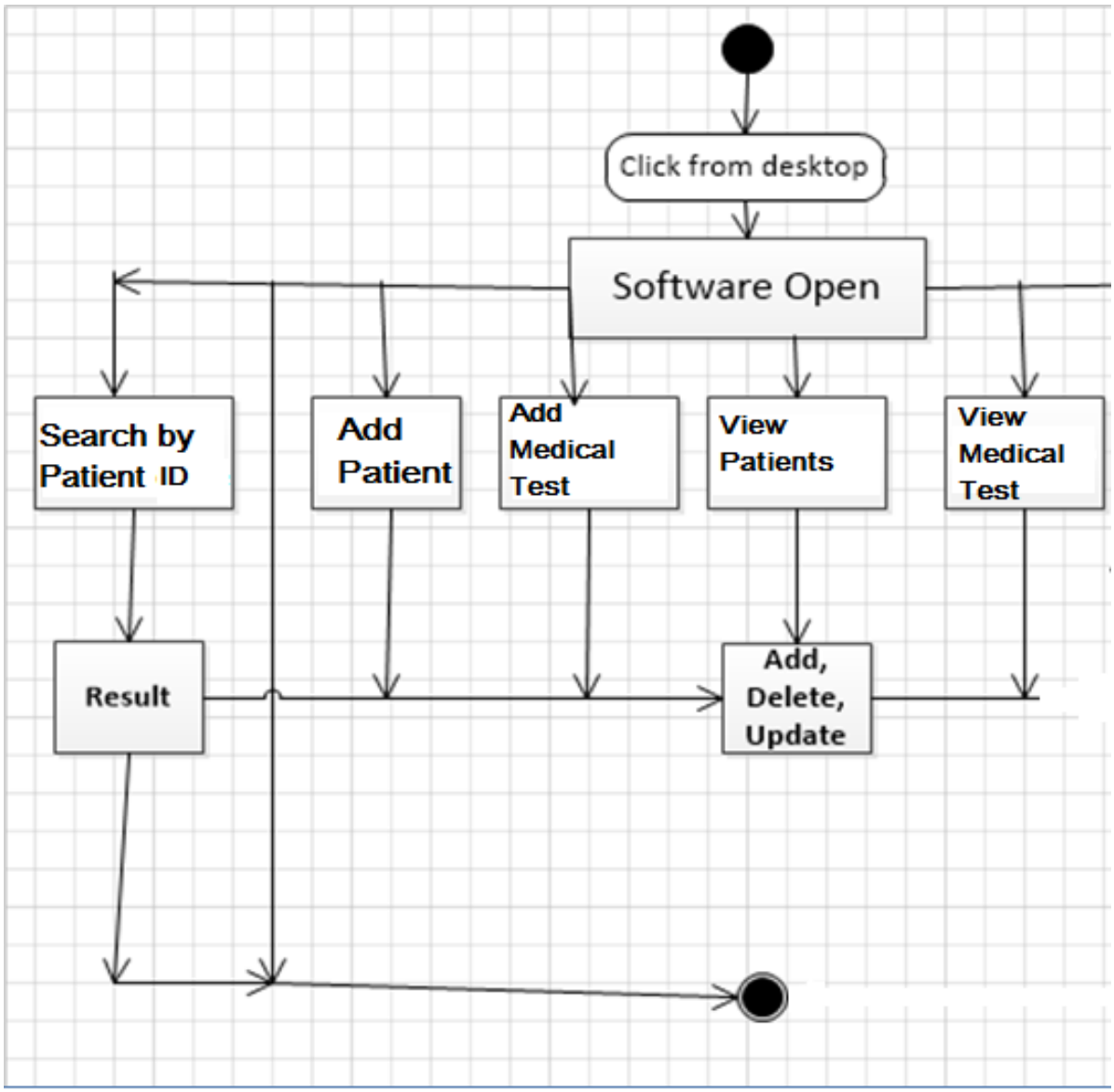

Figure 9: Activity Diagram

3.2.3 Data Flow Diagram

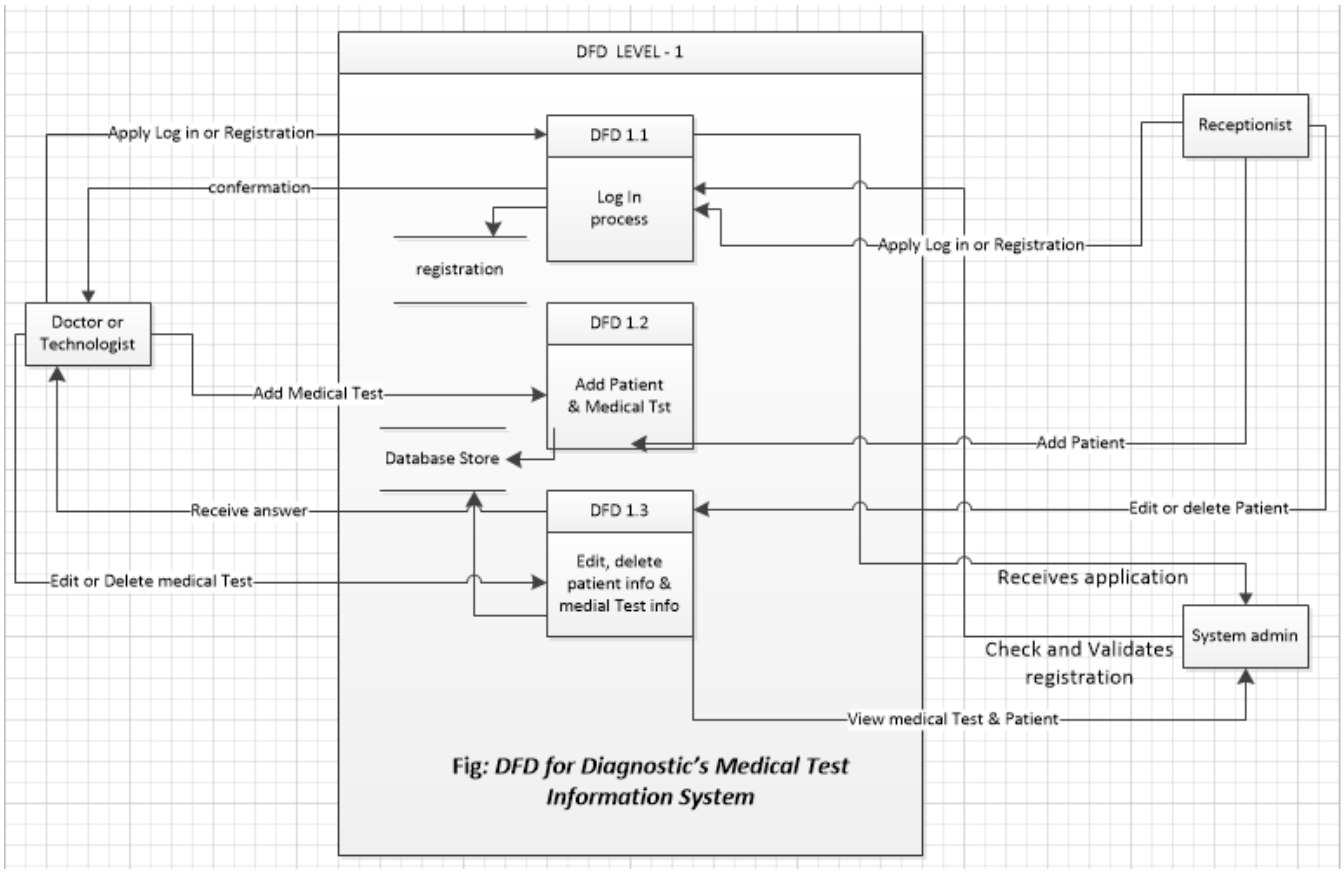

Figure 10: Data Flow Diagram

3.3 How to DMTS Works

Most diagnostic tests are conducted on the living; however, some of these tests can also be carried out on a dead person as part of an autopsy. Some of the diagnostic tests are parts of a simple physical examination which require only simple tools in the hands of a skilled practitioner, and can be performed in an office environment. Some other tests require elaborate equipment used by medical technologists or the use of a sterile operating theatre environment. Some tests require samples of tissue or body fluids to be sent off to a pathology lab for further analysis. Some simple chemical tests, such as urine pH, can be measured directly in the doctor's office.

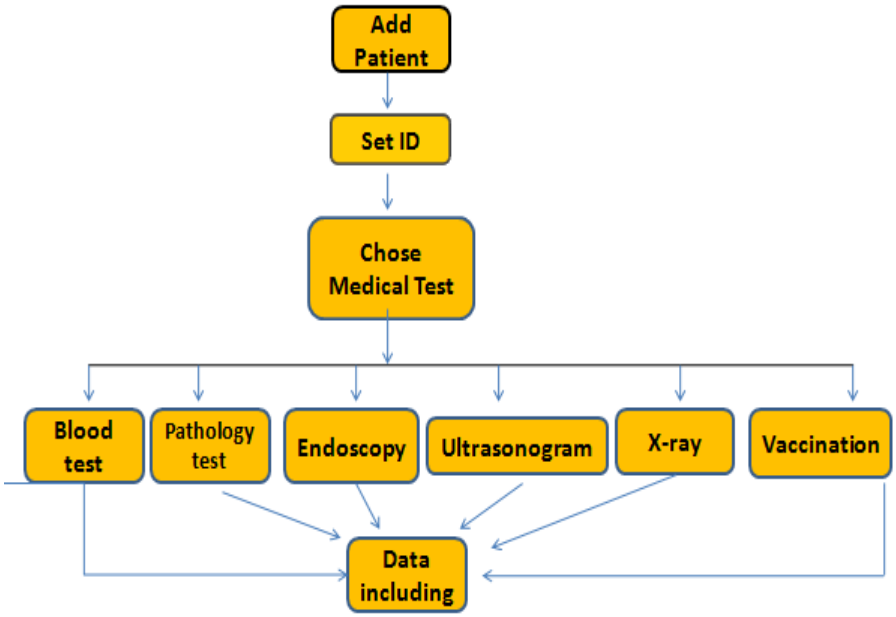

Figure 11: Expected outcome process

The medical profession requires the ability to provide diagnostic medical imaging utilizing critical thinking skills to make judgments in the process. Medical professionals who must possess high level skills in diagnostic techniques under the guidance of a licensed physician. A medical software is responsible for providing excellent patient care and gathering adequate data necessary for diagnoses to be determined.

Graduates of the project will be able to perform, at minimum, the following objectives:

Demonstrate critical thinking skills during the performance of medical procedures to provide optimum diagnostic services in a information system.

35. Add Patient
36. Add Medical Test
37. Blood Test
38. Pathology Test
39. Urine Test
40. Stool Test
41. Endoscopy
42. Ultrasonogram
43. Vaccination
44. X-ray Report
45. View Patient

46. View Medical Test
47. Search Patient
48. Search Medical Test
49. Edit Patient
50. Edit Medical Test
51. Delete Option

3.4 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.

Chapter 4
Testing

4.1 Introduction

This test approach document describes the appropriate strategies, process, workflows and methodologiesused to plan, organize, execute and manage testing of DMTIS

4.1.1 Scope

The DMTIS test plan defines the necessary testing approaches, i.e. unit, integration, system, and client acceptance

- Testing of all fulfilled requirements, including but not limited to functional, nonfunctional requirements, security and other requirements which are provided in software requirement document.
- Testing of all interfaces that work with the system.

4.1.2 Objective

Primary Objective:
At the end of software development life cycle, the user should find that the software has fulfilled or exceeded all his mentioned requirements, including non-functional requirements, otherwise defined as quality requirements. Any changes, modifications, additions or deletions to the requirements provided in the software requirement specification document will be attempted to be full filled provided that they are made within the remaining time of the software development and
are within the abilities of the developers whereas it is provided that the newly modified requirements are feasible to be implemented.

Secondary Objective:
The secondary objective of this document is to provide a guideline to identify and expose all issues, and possible problems, as well as associated risks to the project team so that these can be solved within the remaining time of the software development cycle.

4.2 Roles and Responsibilities

Considering the limited size of the team, the members of the software development team will have to play multiple roles, and thus will have to carry many responsibilities. The roles and their responsibilities are given below:

Developer

- Develop the System/Software/components
- Develop the use cases, and related UML diagram.

Tester

- Conduct unit, integration, and system level testing.
- Support user acceptance testing.

Tester manager

- Monitor and manage testing process 3Testing on DSEDroid
- Assign roles to Testers
- Co-ordinate testing among testers.

4.3 Assumptions

There are several assumptions concerning the testing process. These are:

- The developer team has completed the requirements specified in the requirement specification document, including quality requirements.
- The user acceptance testing will be conducted by end users.
- Developers, testers will be obliged to Tester Manager to follow instructions to provide maximum efficiency to complete the testing phase.
- Developers will support testing based on priorities.
- Developers will help provide testing facilities if required.
- User acceptance testing cannot be conducted by any of the development team.

4.4 Definitions

Bugs:
Any error or defects that cause the software/application or hardware to malfunction. That is also included in the requirements and does not meet the required workflow, process or function point.

Enhancement:

- Any alteration or modification to the existing system for better workflow and process.
- An error or defect that causes the software/application or hardware to malfunction.
- Where 1) and 2) is NOT included in the requirements can be categorized as an enhancement.
- Enhancement can be added as a new requirement after appropriate Change Management process.

4.5 Test Methodology

The purpose of the Test Plan is to achieve the following:

- Define testing strategies for each area and sub-area to include all the functional and quality (nonfunctional) requirements.
- Divide Design Spec into testable areas and sub-areas (do not confuse with more detailed test spec). Be sure to also identify and include areas that are to be omitted (not tested) also.
- Define bug-tracking procedures.
- Identify testing risks.
- Identify required resources and related information.
- Provide testing Schedule.

4.5.3 Usability Testing

The purpose of usability testing is to ensure that the new components and features will function in a manner that is acceptable to the customer. Development will typically create a non-functioning prototype of the UI components to evaluate the proposed design. Usability testing can be coordinated by testing, but actual testing must be performed by non-testers (as close to end-users as possible). Testing will
review the findings and provide the project team with its evaluation of the impact these changes will have on the testing process and to the project as a whole.

4.5.4 Unit Testing (Multiple)

Unit testing is conducted by the Developer during code development process to ensure that proper functionality and code coverage have been achieved by each developer both during coding and in preparation for acceptance into iterations testing.

4.5.5 Iteration/Regression Testing

During the repeated cycles of identifying bugs and taking receipt of new builds (containing bug fix code changes), there are several processes which are common to this phase across all projects. These include the various types of tests: functionality, performance, stress, configuration, etc. There is also the process of communicating results from testing and ensuring that new drops/iterations contain stable fixes (regression). The project should plan for a minimum of 2-3 cycles of testing (drops/iterations of new builds).

4.5.6 Final Release Testing

Testing team with end-users participates in this milestone process as well by providing confirmation feed back on new issues uncovered, and input based on identical or similar issues detected earlier. The

intention is to verify that the product is ready for distribution, acceptable to the customer and iron out potential operational issues. Assuming critical bugs are resolved during previous iterations testing- Throughout the Final Release test cycle, bug fixes will be focused on minor and trivial bugs (severity 3 and 4). Testing will continue its process of verifying the stability of the application through regression testing (existing known bugs, as well as existing test cases). The milestone target of this phase is to establish that the application under test has reached a level of stability, appropriate for its usage (number users, etc.), that it can be released to the end users

4.5.7 Testing Completeness Criteria

Release for production can occur only after the successful completion of the application under test throughout all of the phases and milestones previously discussed above. The milestone target is to place the release/app (build) into production after it has been shown that the app has reached a level of stability that meets or exceeds the client expectations as defined in the Requirements, Functional Specification.

4.6 Test Levels

Testing of an application can be broken down into three primary categories and several sub-levels. The three primary categories include tests conducted every build (Build Tests), tests conducted every major milestone (Milestone Tests), and tests conducted at least once every project release cycle (Release Tests). The test categories and test levels are defined below:

Chapter 5
Conclusion

5.1 Conclusion

We believe such system will help to the Diagnostic Center organization for take care the Information of medical test of patient. So A Diagnostic's Medical Test Information System is a kind of medical software procedure performed to detect, diagnose, or monitor diseases, disease processes, susceptibility, and determine a course of treatment

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.