MEDICAL LABORATORY MANAGEMENT SYSTEM

By

Asfand Yar
(Reg No: 01161811003)

INSTITUTE OF INFORMATION TECHNOLOGY

QUAID-I-AZAM UNIVERSITY

ISLAMABAD

2020

MEDICAL LABORATORY MANAGEMENT SYSTEM

This Documentation submitted in partial fulfillment of the requirements for the degree of

Masters of Science

In

Information Technology

By

Asfand Yar

(Reg No: 01161811003)

Supervised

By

Prof: Dr. Khurram Gulzar Rana

INSTITUTE OF INFORMATION TECHNOLOGY

QUAID-I-AZAM UNIVERSITY

ISLAMABAD

2020

DEDICATION

Every challenging work needs self-efforts as well as the guidance and support of the elders especially the ones who are close to our hearts.

My humble efforts are dedicated

TO

My parents who support me financially and give me courage.

CONTENTS

ACKNOWLEDGEMENT ..... **VI**

ABSTRACT ..... **VII**

PROJECT IN BRIEF ..... **VIII**

CHAPTER 1 ..... **1**

1.1 INTRODUCTION 1

CHAPTER 2 ..... **4**

REQUIREMENT ANALYSIS **4**

2.2 Requirement Elicitation: 4
2.3 Requirement Analysis: 5
2.4 Requirement Specification: 5
2.5 Requirement Validation: 5
2.8 SYSTEM REQUIREMENTS: 7
2.8.1 User Interface Requirements: 7
2.8.2 Software Requirements: 7
2.9 Hardware Requirements 7

CHAPTER 3 ..... **8**

ANALYSIS AND DESIGN **8**

3.1 SYSTEM DESIGN OVERVIEW: 8
3.2 ANALYSIS OF EXISTING SYSTEM: 8
3.3 Use Case Model: 8
3.2 ACTIVITY DIAGRAMS 10
ACTIVITY DIAGRAM: 12
3.3 SEQUENCE DIAGRAM 13
3.4 CLASS DIAGRAM 15
3.5 ENTITY RELATIONSHIP DIAGRAM 16

CHAPTER 4 ..... **17**

TOOLS AND TECHNOLOGIES **17**

CHPATER 5 ..... **19**

INTERFACES AND DESIGN **19**

5.1 LOGIN PAGE 19

CHAPTER 6 ..... **28**

IMPLEMENTATION **28**

6.4 THREE TIER ARCHITECTURE: 29
User interface: 29
Logic: 29
ARCHITECTURE LAYERS: 29
Presentation tier: 29
Application tier: 29
Data tier: 29

6.5 SOFTWARE PROCESS MODEL 30
6.5.1 Agile Software Development 30
6.5.2 Why Agile? 30
Agile Process Models 30
Key activities: 31

CHAPTER 7 32

TESTING AND DEBUGGING 32

7.1 INTRODUCTION: 32
7.2 SOFTWARE TESTING: 32
GOALS: 33
TESTING LEVELS: 33
1. Unit Testing: 34
2. Component Testing: 34
3. Integration Testing: 34
4. Component Integration Testing: 34
5. System Testing: 34
6. Acceptance Testing: 34
7. Alpha testing: 34
7.5 TESTING TECHNIQUES: 34
7.5.1 Black box testing. 35
7.5.2 White box testing 36
7.6 Gray box testing 37
A. TEST CASES 37

FUNCTIONS TO BE TESTED: 38

CHAPTER 8 41

FUNCTIONALITY AND COMMANDS 41

8.1 FUNCTIONALITES: 41

8.1.1 Select Test 41
8.1.1 Print Form 41
8.1.3 Date and Time 41
8.2 Maintenance 41
8.3 System Documentation 42
8.4 Over View 42
8.5 Data Base Specification 42

CHAPTER 9 43

CONCLUSION AND FUTURE WORK 43

ACKNOWLEDGEMENT

Foremost, I am grateful to the Allah for the good health and well-being that were necessary to complete this project. I am taking the opportunity to express my sincere gratitude to my supervisor **Prof. Dr Khurram Gulzar Rana** for the continuous support of my master‟s study and Project, for his patience, motivation, enthusiasm, and immense knowledge and also thanks to **Mr. Safiullah** for his collaboration throughout in the project. His guidance helped me in all the time of my project and writhing this thesis. The door to his office was always open whenever I got into a trouble or had questions about my project. I could not have imagined having a better supervisor and mentor for my final year project. This project could not have been accomplished without his support and guidance.

I also place on record, my gratitude to all of those who directly or indirectly supported me throughout my education career and on this project. Last but not the least, I must express my deepest gratitude to my parents and my family members for providing me with unfailing support and encouragement throughout my years of study and supporting me spiritually throughout my life. This accomplishment would not have been possible without them. Thank you.

Asfand Yar

ABSTRACT

The Laboratory Management System belongs to the class of application software intended for storage and management of information obtained in the course of work. The systems are used to control and manage samples, test result, laboratory, and staff and work flow automation. Integration of Medical laboratory management systems with the enterprise administration.

This project is a desktop based application and maintains a centralized repository of all related information. The objective of this project is to develop a system that automates the processes and activities of a medical laboratory. The purpose is to design a system using which one can perform all operations related to patients. In the present system many work are manual based. This often requires a lot of time and effort. The purpose is to design a system using which one can perform all operations related to Medical Laboratory.

It is very easy to get this software we will create for you according to your demand, it is very easy to use and handle all the work involving in medical laboratory, this management software is actually digital medical lab, just get this software and manage your medical laboratory system

PROJECT IN BRIEF

Project title:	Medical Laboratory Management System
Undertaken by:	Asfand Yar
Session:	2018-20
Internal supervisor:	Dr. Khurram Gulzar Rana
Technologies :	C#,SQL
Tool:	Visual Studio Code
Documentation:	word office 2013
Operating System:	Windows 10pro
System Used:	Lenovo core i7 , 4^{th} **Gen**

CHAPTER 1

INTRODUCTION

1.1 INTRODUCTION

Medical Laboratory Technology presents the development in the medical laboratory science. In many countries, there are two main types of labs that process the majority of medical specimens. Hospital laboratories are attached to a hospital, and perform tests on patients. Private (or community) laboratories receive samples from general practitioners, insurance companies, clinical research sites and other health clinics for analysis. For extremely specialized tests, samples may go to a research laboratory. A lot of samples are sent between different labs for uncommon tests. It is more cost effective if a particular laboratory specializes in a rare test, receiving specimens (and money) from other labs, while sending away tests it cannot do.

1.2 Purpose

The medical laboratory management system is built for the purpose of effective and clear data saving and manipulating. The medical laboratory management system project highly minimize time and resource by which, searching the reports and data you can get the data in quickest time. And almost the resources are wise used since most actions are done on the medical laboratory management system. Some of the resources minimize the manpower and paper work. The feature use to storing data in a secure way.

1.3 Scope

The application can be used in any Hospital that has medical/laboratory within the hospital for maintaining patients‟ tests and test report detail by customizing some of the features. There are many modules available in this medical laboratory management system and you will not find anywhere else only because of the lead features and modules. Medical laboratory management system contains all the main features which are difficult to handle physically but now you can run medical laboratory easily.

1.4 Objectives

It is the user friendly application which reduces the burden and helps to manage all sections of laboratory like patient reports and Billing etc., which improve the processing efficiency. It deals with the automating tasks of maintaining of Bills, it is

the key process. Including safe data store about patient reports as well as fast searching, delete and update of reports. The laboratory management system is easy for use so the user can do actions without ambiguities. The main Objectives of the MLMS is making the Laboratory organizations computerized by creating neat work through minimizing or eliminating wasting of time as well as removing the resources such as papers for data saving since knows a days is paper based.

1.5 Definitions, Acronyms and abbreviation

1.5.1 User Friendly

Is the way that the built system is not ambiguous which is clear for using the created software interface for manipulating actions or tasks? In the other way the proposed system is designed for human likable components in color, font and other related things.

1.5.2 Manual Based System

The system that uses was paper based and arranged on the shelf through functionality of documents. Everything that is arranged, searched, updated and deleted is through humans only. In general manual based system is uncomputerized system which is tedious in its data arrangement for efficient work.

1.5.3 Medical Laboratory

The place where done the different test which recommended by the doctor for the purpose of known the disease. In which you can handle multi types of work within seconds, medical laboratory management system contains all the main features which are difficult to handle physically but now you can medical laboratory easily.

1.5.4 Management System

A system in which manage, organize, formulate data‟s through a technical data structure arrangement.

1.5.5 Billing

The way in which generating paper which store information about some specific data containing details explanation.

1.5.6 MLMS

Medical laboratory management system

1.5.7 C#

Technology use for coding.

1.5.8 My SQL

Technology use for Data Base.

Overview

The Laboratory management system is built in order to replace manual based system to computerize. If you want to get this software we will create for you according to your demand, it is very easy to use and handle all the work involving in medical laboratory, this management software is actually digital medical lab, just get this software and manage your medical laboratory system.

Laboratory Practice in The Past

This gives an overview of events that have occurred in Laboratory practice through several ages both in the past and present, and also indicates possible events of the future.

Synopsis of This Project:

In the First chapter we discuss about our project, existing system, proposed system, goals. In Second Chapter we describe our functional and non-functional requirements. In Third Chapter we discuss about analysis of existing system and the design. UML and ER diagram are also discussed. In Fourth Chapter we discuss about the tools and technologies that we used in this project. In Chapter Five we add the snapshots of the existing system. And in Chapter 6 we discuss about the implementation of the existing system. In Seventh Chapter Testing and debugging are discussed. In Chapter 8 we discuss about the functionalities and commands use in the existing system. In Last Chapter we discuss about the conclusion and future work.

CHAPTER 2

REQUIREMENT ANALYSIS

2.1 INTRODUCTION

The software requirement analysis determines user expectations for the product being developed. It is also called requirement engineering. The requirements must be relevant, detailed and quantifiable. These requirements are called functional specifications. It captures complete descriptions about how the system is expected to perform. In the requirement analysis complete information about the system that what end user wants is described like the system inputs and the expected output from the system. Software requirements analysis is an important aspect of the project. Requirements analysis involves frequent communication with system users to determine specific feature expectations, resolution of conflict or ambiguity in requirements as demanded by the various users or groups of users, avoidance of feature creep and documentation of all aspects of the project development process from start to finish. All efforts are put to conform that the final product is up to the client‟s expectations. Software requirement is a functional or non-functional need to be implemented in the system.

Objective

The main goal of requirement analysis and gathering is to clear the ambiguity of requirements of the system how the system should behave what the system should do and what the system should not do. Another thing is to clarify the requirements and analyse them that how the requirements would help in the implementation of the project with the actual expectations of the ends user.

2.2 Requirement Elicitation:

It consist of collection of information, understanding the stakeholder needs. It includes interviews, questionnaires, user observations, workshops, brain storming, use case and prototyping.

2.3 Requirement Analysis:

The analysis activity examines the high level requirements and determines if they are clear complete and free of contradictions, and then define the strategy to address these issues. The main goal of requirement analysis and gathering is to clear the ambiguity of requirements of the system how the system should behave what the system should do and what the system should not do.

2.4 Requirement Specification:

A specification is a document that specifies, in a complete, verifiable manner, the requirements, design, behavior, other characteristics of the system and often the procedure for determining whether these provisions have been satisfied.

2.5 Requirement Validation:

Requirement Validation is the process of confirming the completeness of the user requirements.

- ✓ Achieved stated business objective.
- ✓ Meet the need of stakeholders.
- ✓ Clear and understood by the developers.

Validation is important to identification of missing requirements meet certain quality characteristics.

Requirements are the description of services which a software will provide to the end user.

Types of Software Requirements

1. Functional Requirements
2. Non-Functional Requirements

2.6 Functional Requirements

Functional is actually what a system do. These requirements are related to the function of the system. So functional requirements are the description of services which as software provides to the end user. So the functional requirements are the operations of the software does.

Following are the functional requirements of this project.

1. Login In

To perform all the operations Admin/Receptionist will required to be logged in.

2. Add Receptionist

After successful log in to the system the admin can successfully add receptionist.

3. Add Tests

After adding receptionist admin can add new tests.

4. Add Patient

Receptionist can add patient and select test for patient.

5. Modify Information

Only admin can modify the information. Receptionist can only modify patient information

6. Generate Report

Receptionist can generate test report of Patient.

7. Total Collection

Receptionist can generate collection report and also will view the report.

2.7 Non-Functional Requirement:

Non-functional requirements are the behaviour of the software while performing the operations. Like how the system should behave while performing the operations. These requirements are not related to the functional aspect of the software. Non-functional requirements are how the system will do something not what the system will do.

Non Functional requirements are the behavior of the software while performing the operations. Like how the system should behave while performing the operations. These requirements are not related to the functional aspect of the software. Nonfunctional requirements are how the system will do something not what the system will do. They describe the overall quality and attributes of the proposed system. These requirements place restrictions on the product being developed. Nonfunctional requirements include safety, security, usability, reliability and performance requirements.

1. Usability

Any familiar in using windows operation can operate the system since it have user friendly user interface. Which have the instruction menu‟s how to

use it which self-directive application then can be used the system without ambiguity.

2. User Interface

The user interface is friendly which is easy to use. And having attractive frame structure which is prepared in assumption with other related systems.

3. Operation

The Medical Laboratory management system is operated and controlled by the lab Receptionist for safe work.

4. Support-ability

This Medical Laboratory management system operates in any version of windows operating system. Such as windows 10, windows xp, windows 8, windows 7 and other related versions. The system can be easily maintained by admin of the laboratory system by using the p documents of the system for easy maintenance. Other ways it is maintained by the system developers.

2.8 System Requirements:

2.8.1 User Interface Requirements:

- ✓ GUI (Graphic User Interface) along with meaningful frames and interfaces.
- ✓ Reports will be generated as per requirements.

2.8.2 Software Requirements:

- ✓ Windows 2007, 2008, 2010 windows XP, Vista and above.
- ✓ Platform C#
- ✓ Visual Studio 2015

2.9 Hardware Requirements

Recommended computer hardware requirements:

- ✓ 2.2 GHz or higher CPU
- ✓ 4 GB or more RAM
- ✓ 1280x1024 of Display

CHAPTER 3

ANALYSIS AND DESIGN

3.1 System Design Overview:

In describing the system design, diagrams will be sketched to make clear on how the project is carried out. The diagrams included are use-case diagram, activity-flow diagram, class diagram, entity relationship diagram etc.

System design is the process of defining the architecture, components, modules, interfaces and data for a system to satisfy specified requirements through system modeling.

3.2 Analysis of Existing System:

Before we analyses the design of the proposed system, we need to carefully highlight the problems of the existing system so as to avoid recurrence. Every day, millions of blood specimens are analyzed as part of routine clinical work in hospitals and medical practices throughout the world. Laboratory tests performed are analyzed in clinical laboratories and are usually recorded in computer-based laboratory information systems.

- Significant amount of time is allocated for writing the order as the Lab Receptionist needs to go through a rough amount to order based figures.
- Record in File based system maintain is difficult, it will overcome the load on them.
- Generating report within a specified period of time.
- Reducing the employee‟s workload.

3.3 Use Case Model:

Use case diagram is a behaviour or dynamic diagram in UML. It is the graphical or pictorial representation of user interaction with the system that shows the relationship between the users and the different use cases in which the user is involved. Use case diagrams consists of actors, use cases and relationships among them. Actors

are human and external systems. The use case model is used to model the system and its subsystems of a software application.

The use case diagram has four parts:

1. Actors, actor in a use case perform certain functions in the system. It interacts with the use cases.

2. Boundary, the boundary defines the scope of the system. It limitizes the system.

3. The relationship among the actors and the use cases is actually dependency among them.
   Relationship types are: Extend, Generalization and Include etc.

4. The use cases, in a use case diagram are visual representation of different functionalities.

Use Case:

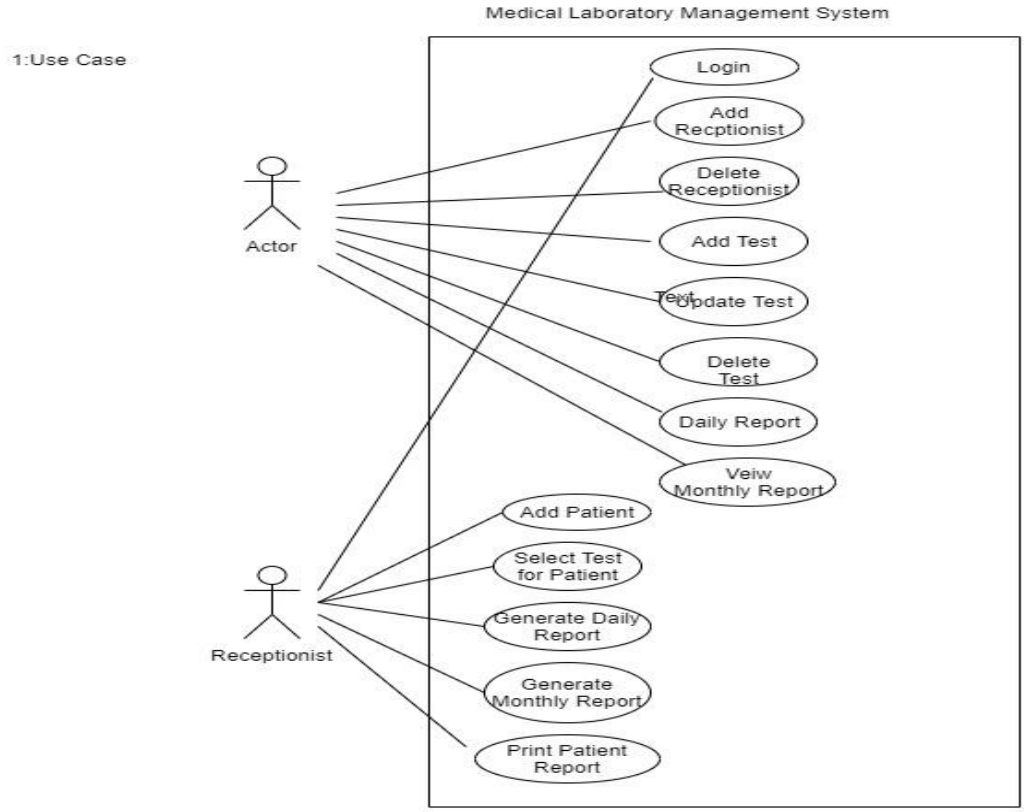

Figure 3.1 Use Case Diagram

Actors:

Admin, Receptionist

Flow of Event:

1. The user or receptiveness initiates the system.
2. The system displays the first page.
3. The first page consists of login Menu‟s.
4. The Admin/Receptionist enters user name and password.
5. The system displays Main Admin page/Receptionist Main page
6. Admin can Add New Tests, Update, and Delete Tests.
7. Receptionist enter the information of patient.
8. Save the information.
9. Receptionist can select tests for patient.
10. Receptionist can generate daily and monthly report.
11. Admin can view the daily and monthly report.
12. Admin can also view the Total collection.

3.2 Activity Diagrams

Activity Diagrams describe the dynamic aspects of the system. It graphically represents the flow from one activity to another activity, it is basically a flowchart. These activities which are shown in graphical flow are the basic operation of the system.

Following are the notations which are used for the creation of activity diagrams:

- ✓ **Activity:**

  The rounded rectangle represents the activity.

- ✓ **Initial Node:**

  The filled in circles is the starting point of the diagram.

- ✓ **Final Node:**

  The circle which is filled and having border around is the ending point of the activities.

- ✓ **Fork:**

  A black bar with one flow going into it and several leaving it.

- ✓ **Join:**

  A black bar with many entering in it and one leaving it.

- ✓ **Decision:**

  A diamond with one flow entering and several leaving.

- ✓ **Merge:**

  A diamond with several flows entering it and one leaving it.

- ✓ **Flow Final:**

  The circle with the X through it. This indicates that the pro cess stops at this point.

Activity Diagram:

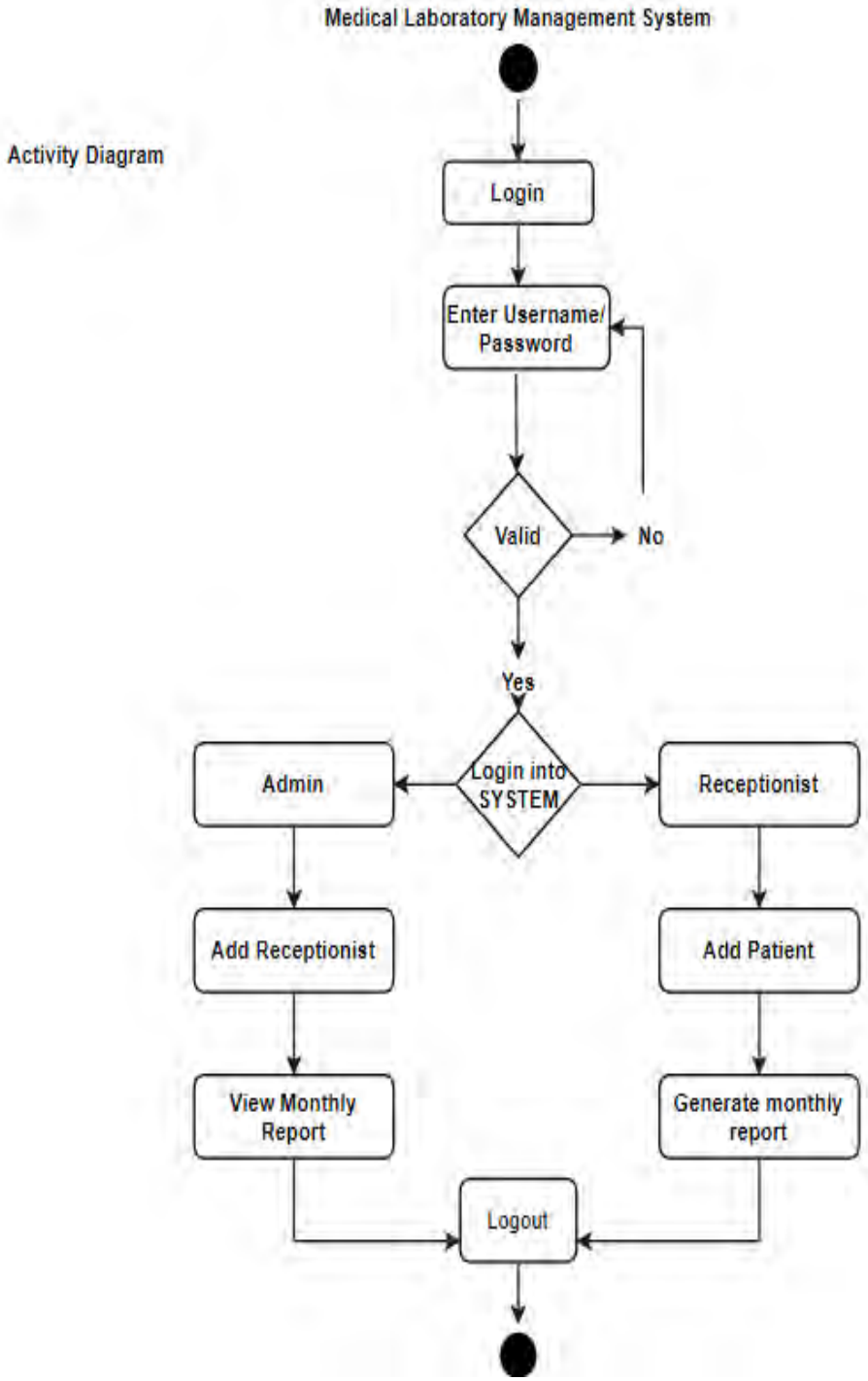

Figure 3.2 Activity Diagram

3.3 Sequence Diagram

A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario.

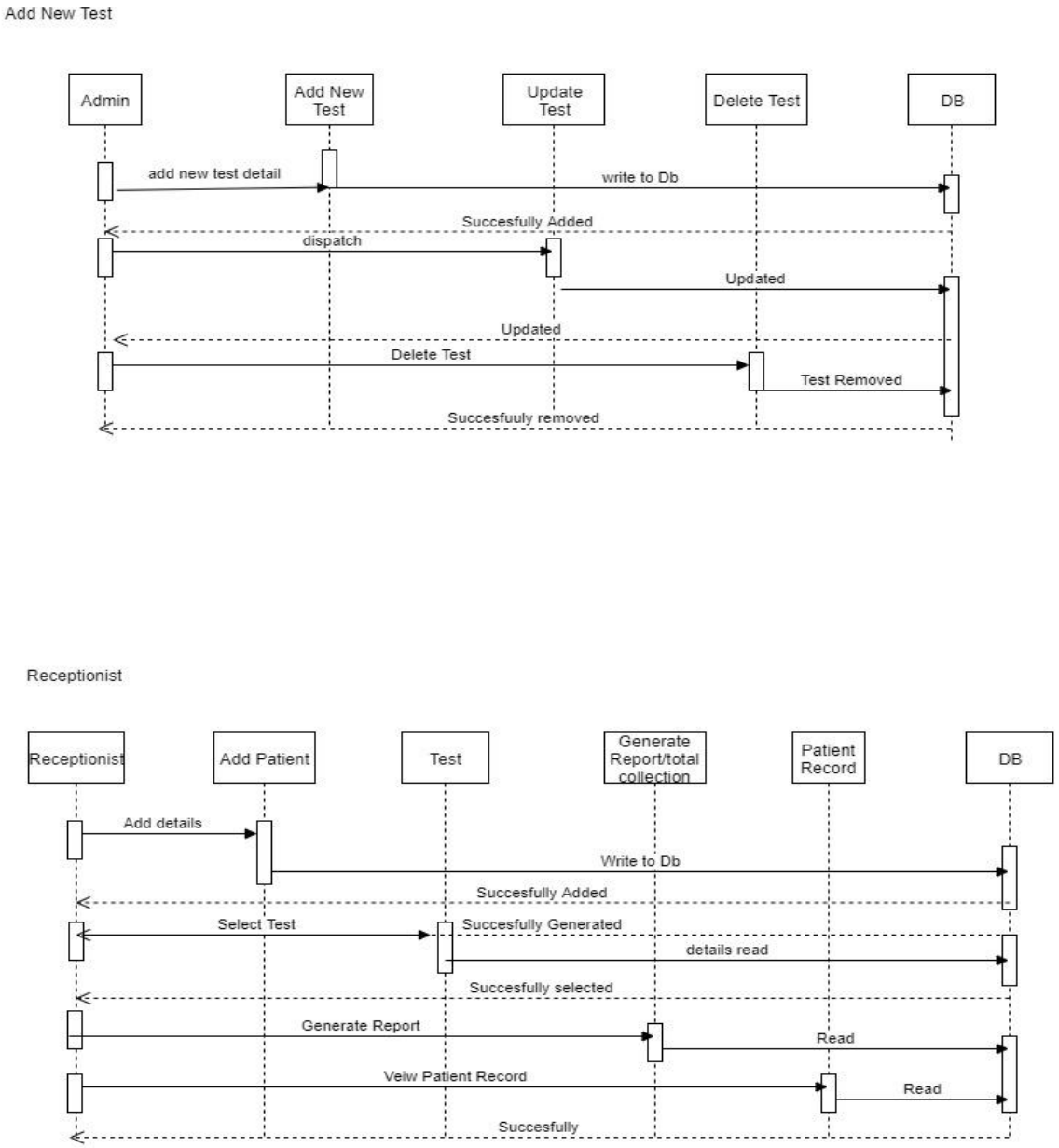

Figure 3.3 Sequence Diagram

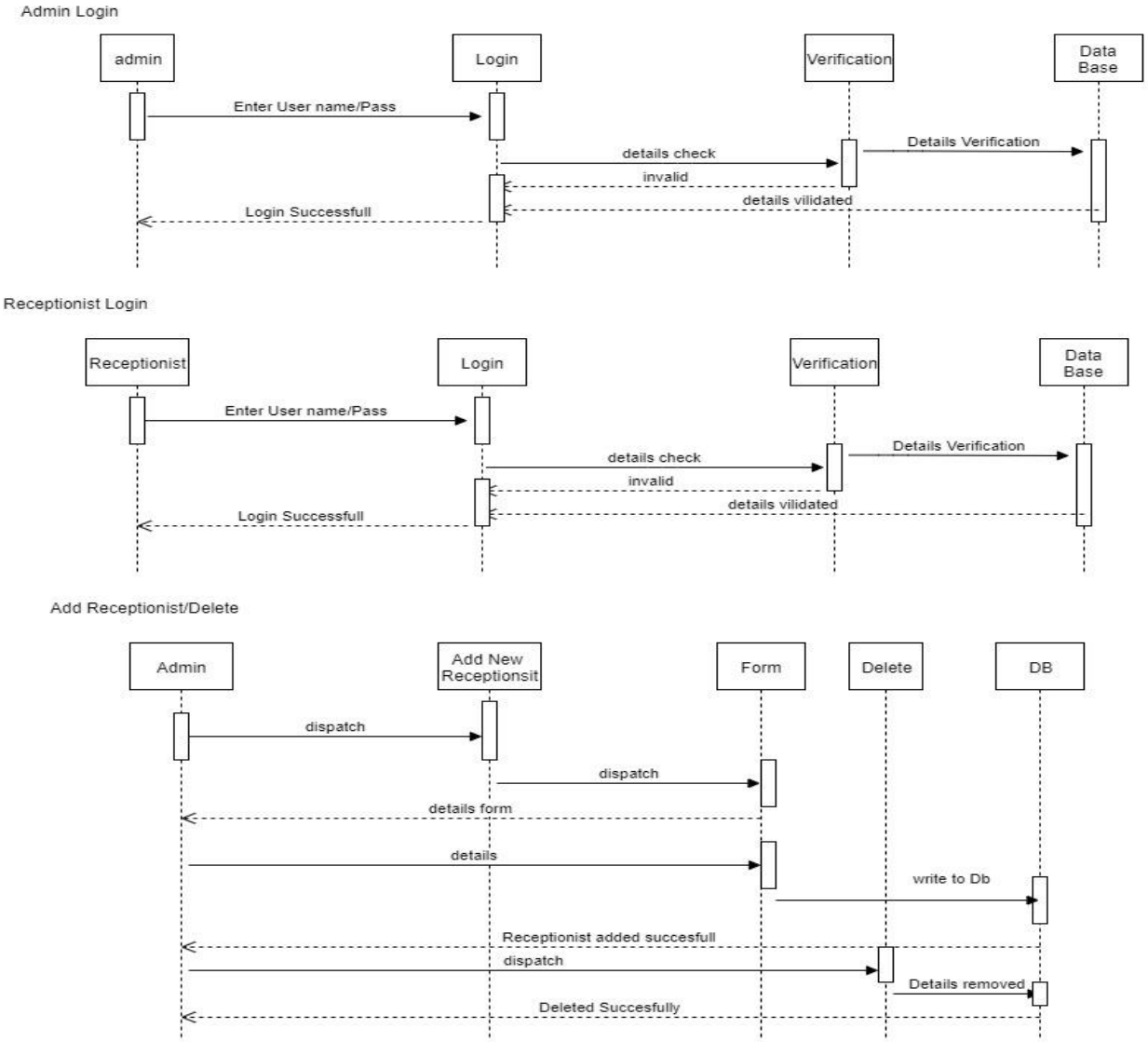

Figure 3.4 Sequence Diagram

3.4 Class Diagram

A class diagram describes the structure of a system by showing the system's classes, their attributes, operations, and the relationships among objects.

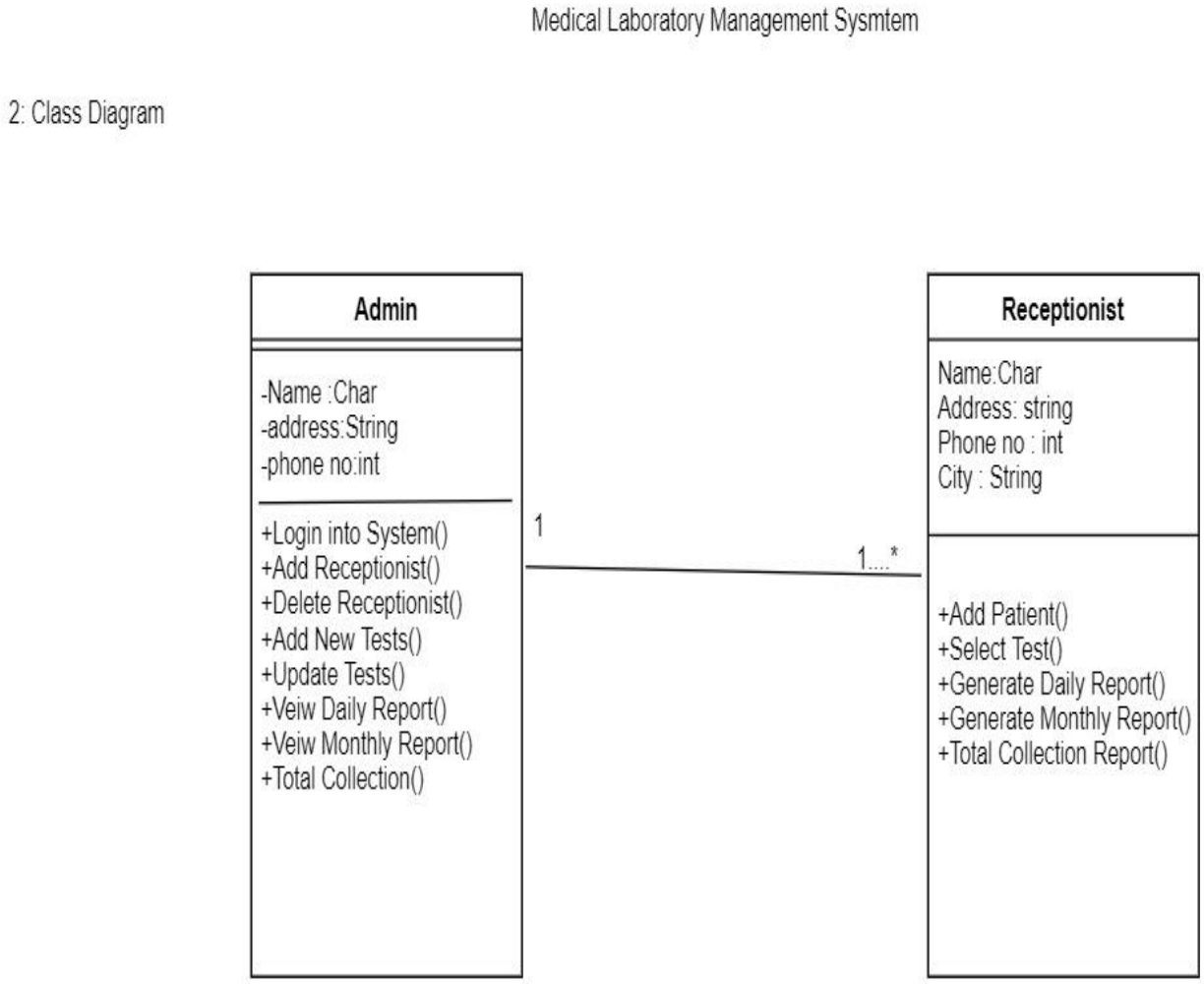

Figure 3.5 Class Diagram

3.5 Entity Relationship Diagram

Database is the integral part of the software system. To fully utilize ER Diagram in database engineering guarantees you to produce high quality database design to use in database creation, management, and maintenance. An ER model also provides a mean of communication.

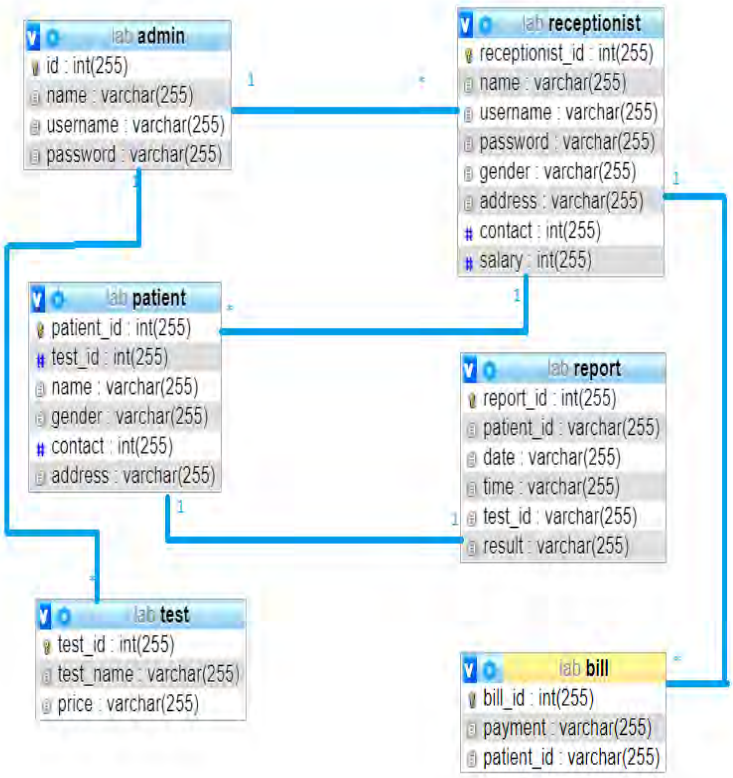

Figure 3.6 ER Diagram

CHAPTER 4

TOOLS AND TECHNOLOGIES

4.1 INTRODUCTION

The mainstream tools used in the development of Medical Laboratory Management System are Visual Studio 2015 for proposed Desktop application. Following is an introduction to these tools which contains programming languages that have their own programming language.

4.1.1 Visual Studio Software

When making a suggestion to someone to try something new, it is important **t**o suggest your case with a strong evidence base. This allow your listener to make an informed decision as to whether or not they will take up your offer. A colleague or friend may suggest that you consider.

Users of visual studio have the ability to select different views when creating a Desktop application. A „code view" show only C# code for the application. A „design view" gives an indication of how the page will appear in a desktop. A „Split view" as the name implies, allows you to see the code view and design view at the same time. Both views are updated automatically as changes are being made and the user can immediately see the effects of any changes.

4.1.2 Language Use for Programming

C# is pronounced as C-sharp is an object-oriented programming language of Microsoft that aims is to combine the computing power of C++ with the programming ease of Visual Basic. C# is based on C++ and contains features similar to those of Java. C# is designed to work with Microsoft Net platform.

C# is designed to work with Microsoft's .Net platform. Microsoft's aim is to facilitate the exchange of information and services over the Web, and to enable developers to build highly portable applications. C# simplifies programming through its use of Extensible Markup Language and Simple Object Access Protocol which allow access to a programming object or method without requiring the programmer to write additional code for each step. Because programmers can build on existing code, rather

than repeatedly duplicating it, C# is expected to make it faster and less expensive to get new products and services to market.

Microsoft is collaborating with ECMA, the international standards body, to create a standard for C#. International Standards Organization recognition for C# would encourage other companies to develop their own versions of the language. Companies that are already using C# include Apex Software, Bunka Orient, Component Source, devSoft, FarPoint Technologies, LEAD Technologies, ProtoView, and Seagate Software.

4.1.3 My SQL for Data Base

To overcome the problems of manual records stored, I have developed "Desktop based Medical Laboratory Management System". Medical Laboratory Management System is based on web server, which can be implemented on any computer. In This application, PHP is server side language, MySQL and PHP is used as back-end design and C# are used as front-end tools. The system communicates with database residing on a remote server. It calculates automatically, the record of students without any manual paper-based work. The system facilitates the end users with interactive design and automated processing of attendance management

CHPATER 5

INTERFACES AND DESIGN

5.1 Login Page

Admin and Receptionist can Login to system by providing username and password to perform activities in the system.

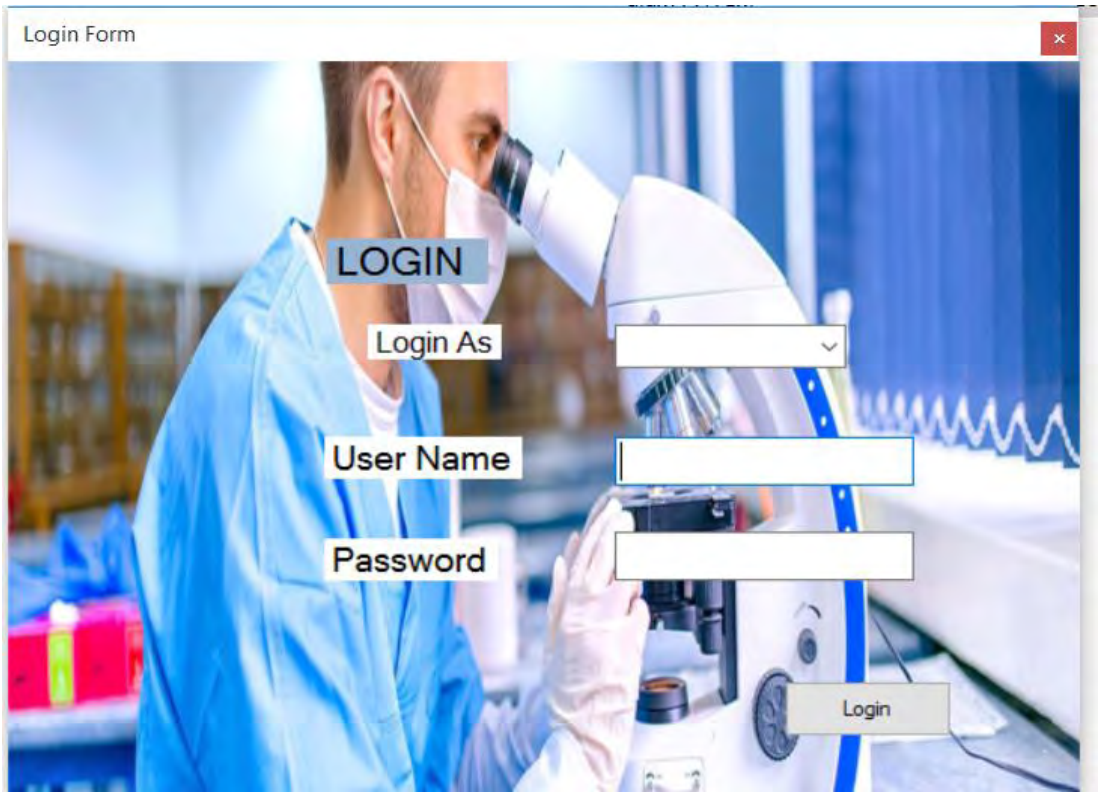

Figure 5.1 Login Form

5.2 Admin Dashboard:

This screen appears when admin login. Admin can perform different operation e.g. add receptionist, add new Tests, View Monthly Report.

Figure 5.2 Admin Dashboard

5.3 Receptionist Dashboard:

When receptionist Login this screen appears, through a receptionist can add patient, update patient, generate patient report, view tests details etc.

Figure 5.3 Receptionist Dashboard

5.4 Add Receptionist:

When admin click on add receptionist this page appears. In this admin add information about receptionist and give receptionist user name and password, through this receptionist will be successfully login to the system**.**

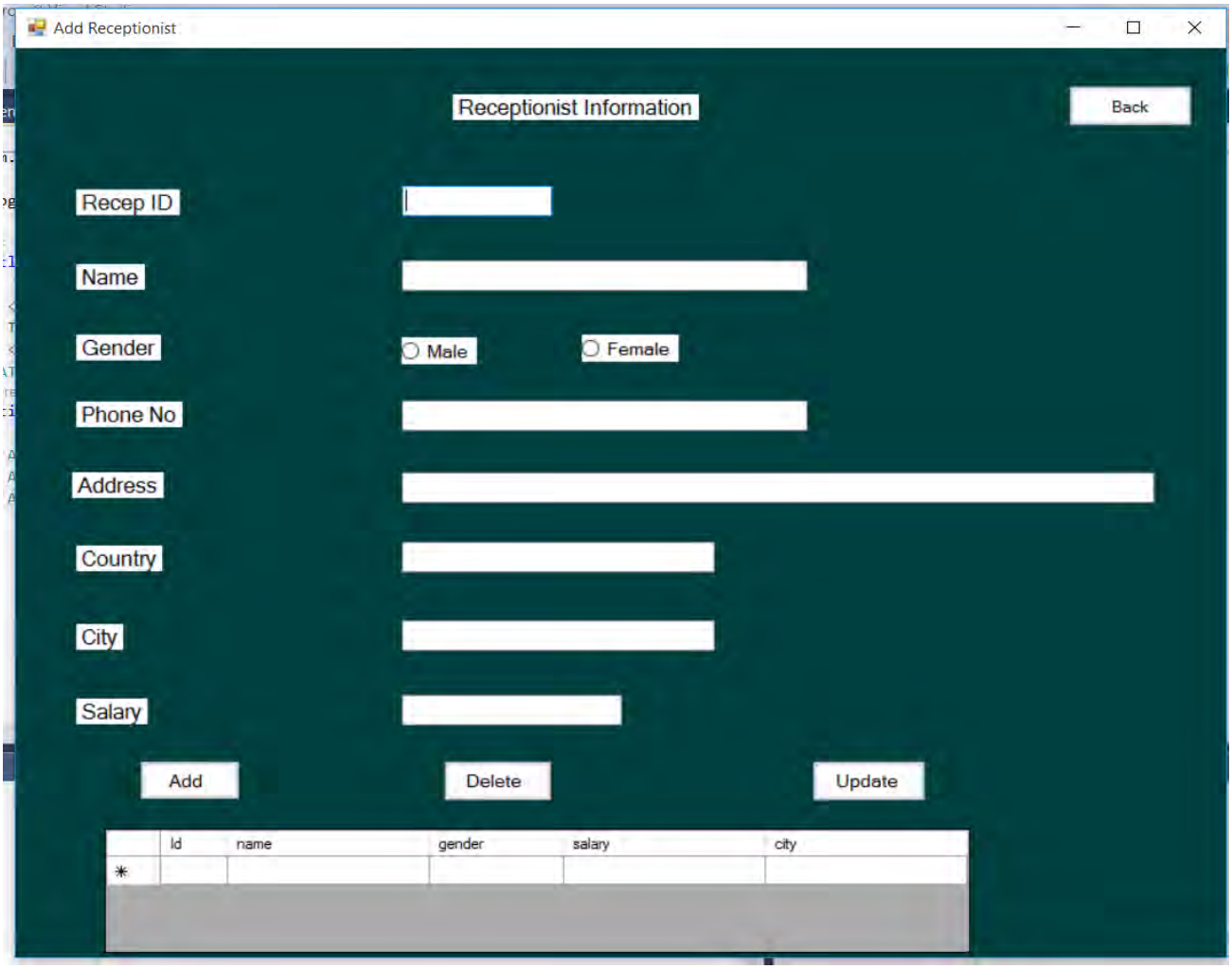

Figure 5.4 Add Receptionist

5.5 Add Tests Page:

Admin can add new tests, and then this tests details will be available for receptionist. Receptionist can assign tests to the patient.

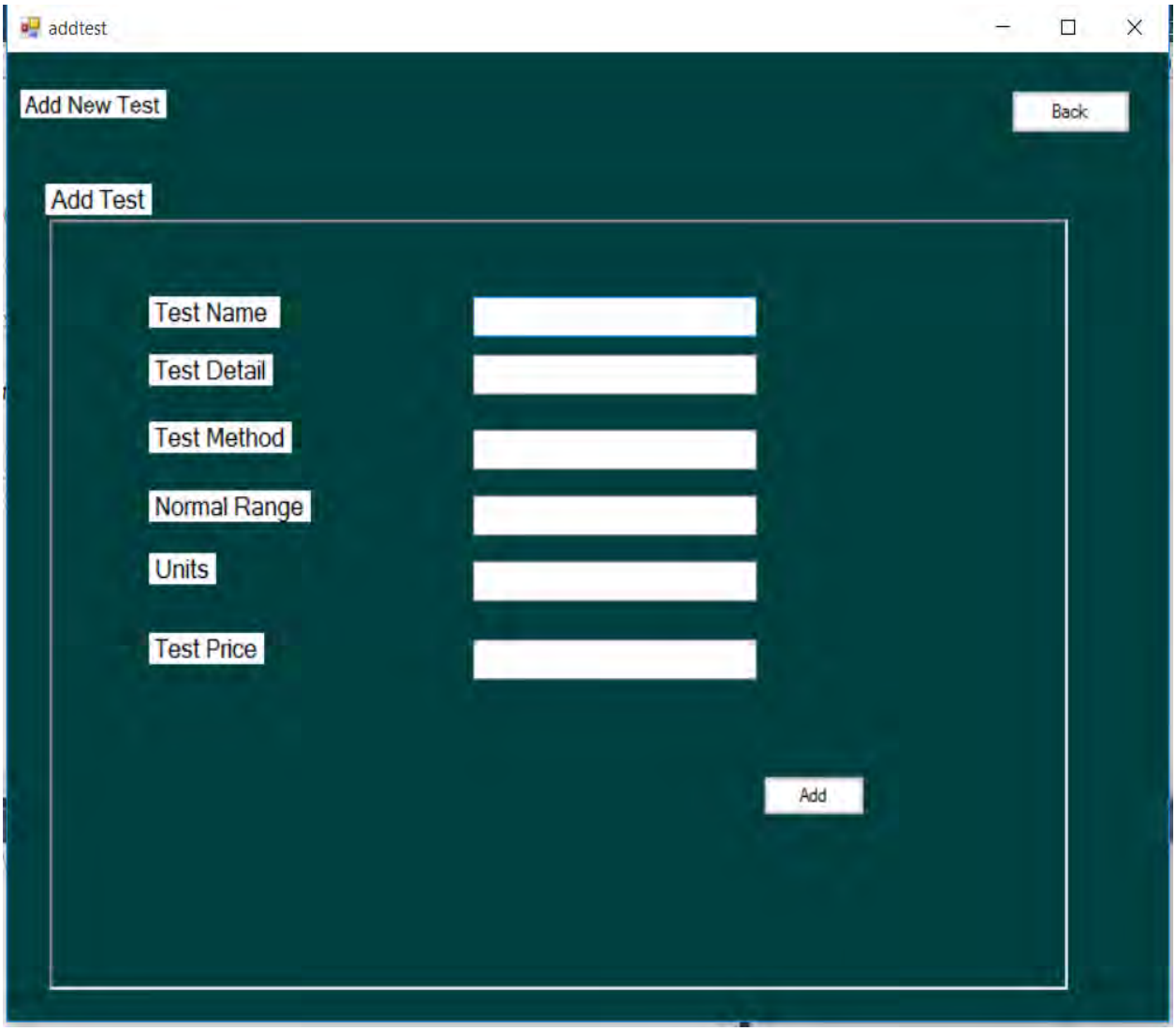

Figure 5.5 Add New Tests

5.6 Add Patient:

This page is appears when receptionist can click on add patient button. Receptionist can add new patient and also assign tests to the Patient.

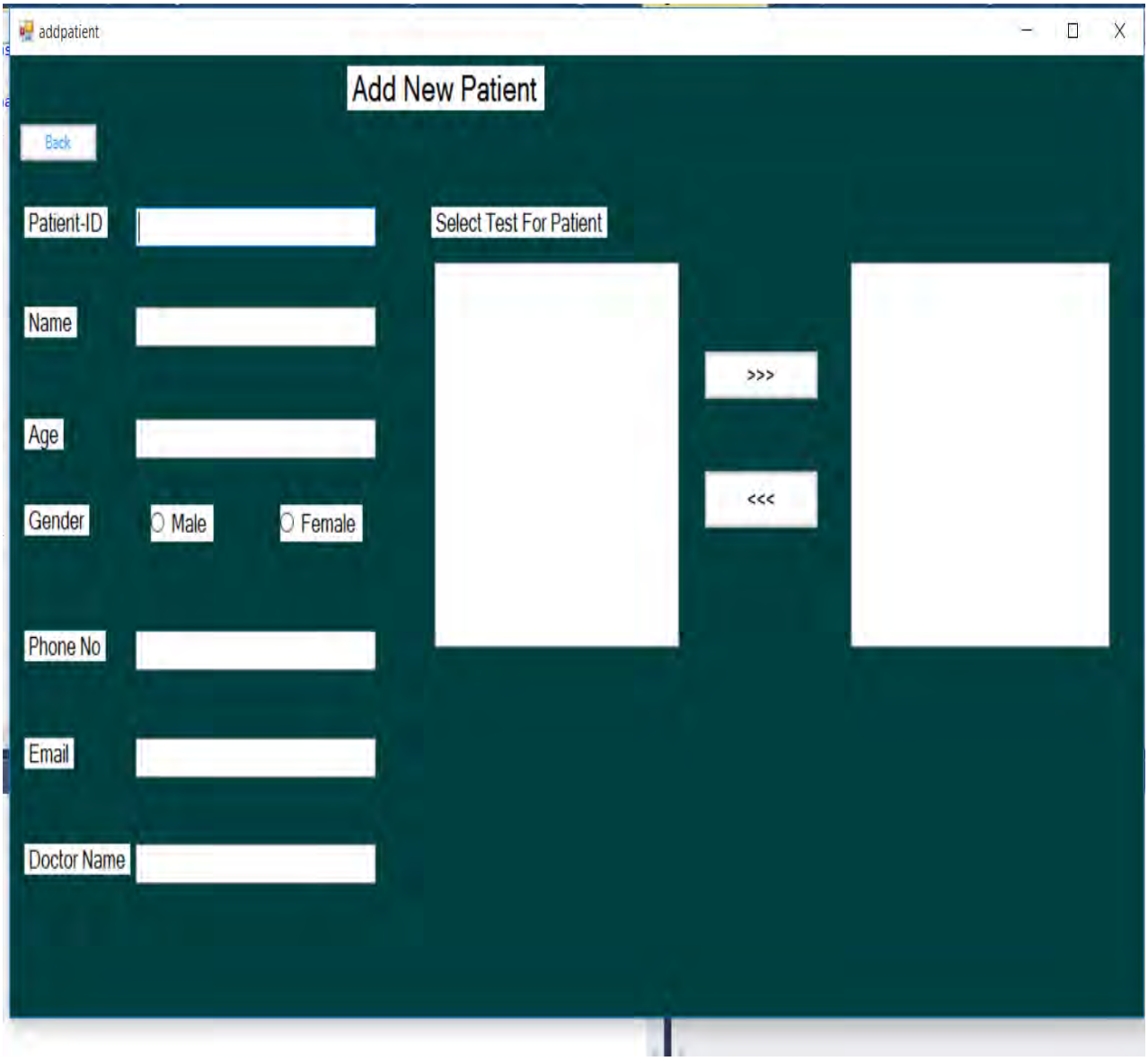

Figure 5.6 Add Patient

5.7 Patient Report:

This page appears when to generate the patient report. In patient report details provided to the patient about tests.

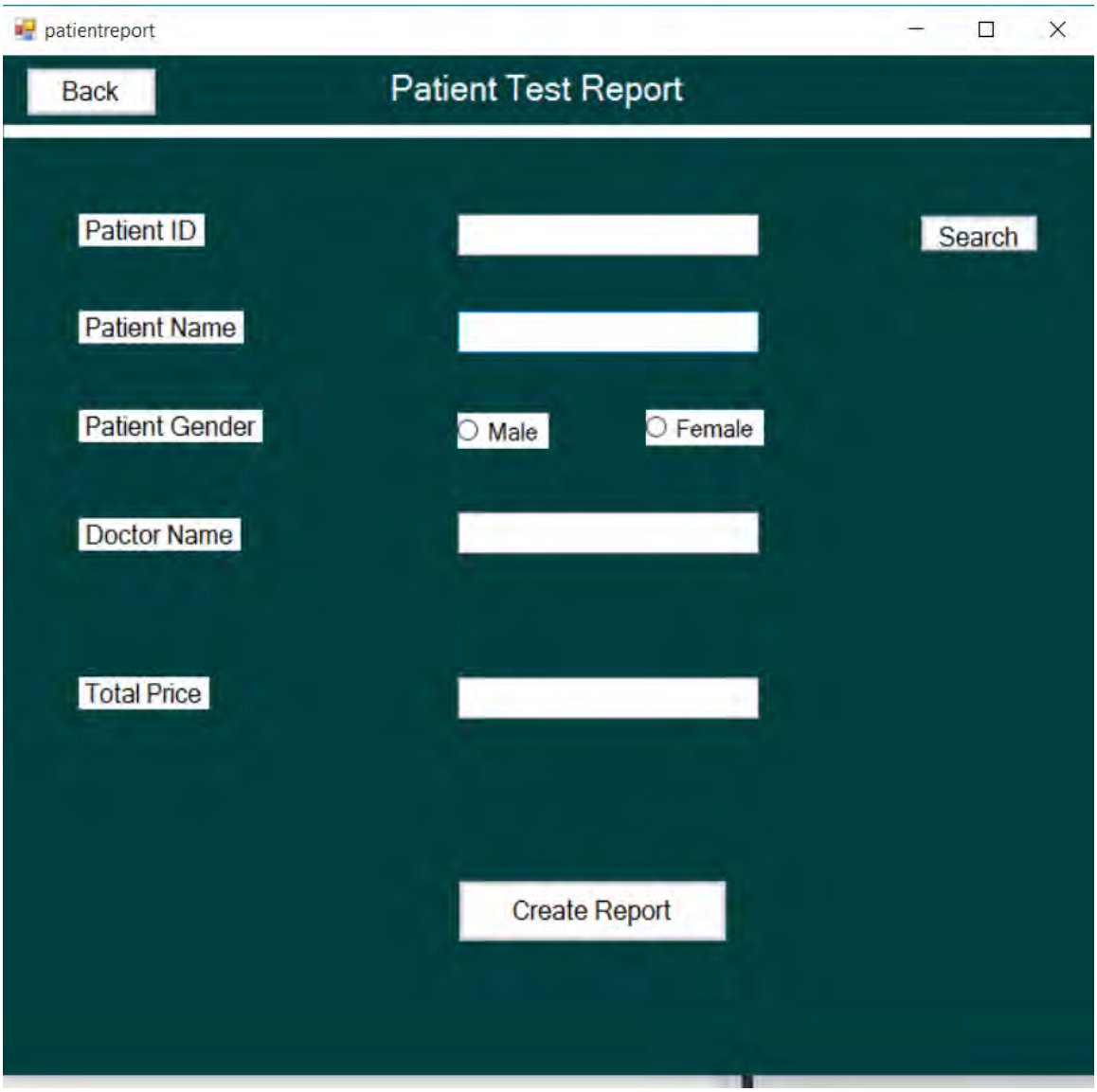

Figure 5.7 Patient Report

5.8 Test Details Page:

Show information about tests that are added by admin. Receptionist can view the tests details, through this it will assign tests to the patient.

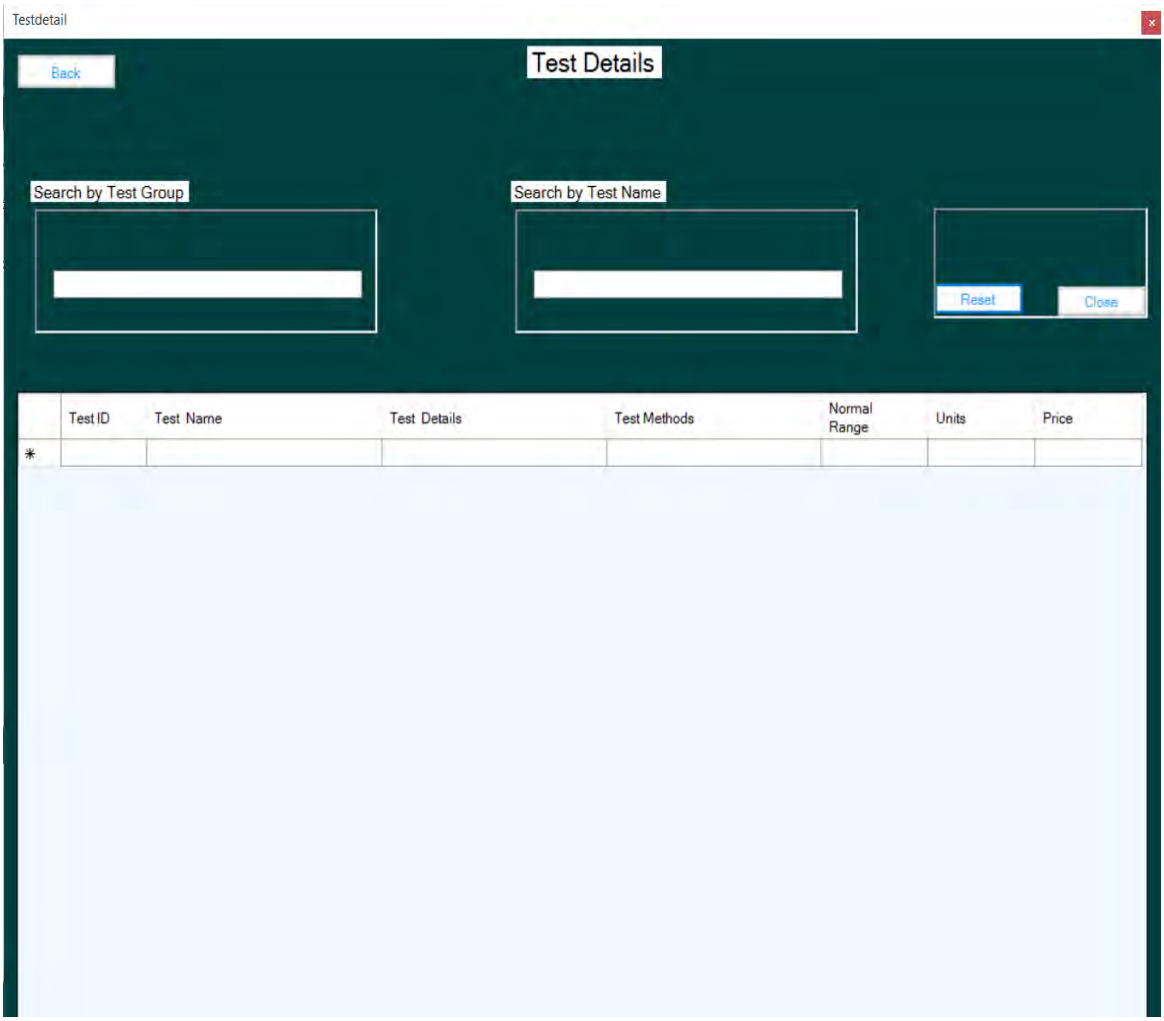

Figure 5.8 Tests Details

5.9 Coding Section:

C# coding snap shots that are used in our system.

```
entreport.cs [Design]    staff.cs [Design]    addrec.cs    patientreport.cs    addpatient.cs [Design]
loginform                                    loginform.admin
20          {
21
22          }
23
            0 references
24          private void admin_Load(object sender, EventArgs e)
25          {
26
27          }
28
            0 references
29          private void button1_Click(object sender, EventArgs e)
30          {
31              addrec newForm = new addrec();
32              newForm.Show();
33              this.Hide();
34          }
35
            0 references
36          private void button2_Click(object sender, EventArgs e)
37          {
38              addtest newForm = new addtest();
39              newForm.Show();
40              this.Hide();
41          }
```

Figure 5.9

```
ientreport.cs [Design]    staff.cs [Design]    addrec.cs    patientreport.cs    addpatient.cs [Design]    admin.cs    Program.cs
loginform                                      loginform.addrec                 textBox7_TextChanged_1(object sender, K
108
109         }
110
            1 reference
111         private void textBox5_TextChanged(object sender, KeyPressEventArgs e)
112         {
113             if (!char.IsControl(e.KeyChar) && !char.IsLetter(e.KeyChar))
114             {
115                 e.Handled = true;
116
117             }
118         }
119
            1 reference
120         private void textBox3_TextChanged(object sender, EventArgs e)
121         {
122
123         }
124
            1 reference
125         private void textBox3_KeyPress(object sender, KeyPressEventArgs e)
126         {
127             if (!char.IsControl(e.KeyChar) && !char.IsDigit(e.KeyChar))
128             {
129                 e.Handled = true;
130
131             }
```

Figure 5.10 Coding

CHAPTER 6

IMPLEMENTATION

6.1 INTRODUCTION

When the design of the system is completed then the development phase of the system begins to transfer the system design into an executable form. As this project is a web and mobile application based so it is important to specify that how they are implemented what tools and technologies are used to implement them. The system implementation and system deployment are the part in which the system is implemented and deployed into real world to be used. This chapter is about the step by step process to implement the system.

When all the necessary stuffs, tools installation is done then the next step is the configuration and creation of the table entities of the database. The database is created on the bases of the system design to store the data. The development of the system is begun with creation to user pictorial interfaces.

6.2 Language Selection:

The selection to language to develop an application is always been a critical job. A good language and tool support helps to avoid upcoming difficulties as we move forward in the development phase. Language use to develop this project is C#. Mostly this project is based on C# language and for data base we use xamp server. SQL queries is used to connect database.

The Desktop application…..

6.3 Desktop based implementation:

The Desktopapp is developed in C#. An application that runs stand-alone in a desktop or laptop computer.

- Windows File Explorer.
- Microsoft Office applications.
- Web browsers (Chrome, Firefox)
- Adobe Photoshop.

6.4 Three Tier Architecture:

I will use three tier architecture in my development. In this architecture there are three layers.

User interface:

Presentation layer shows at the side of client

Logic:

Server contains logic

Architecture Layers:

There are three layers in architecture.

- ✓ Presentation tier
- ✓ Application tier
- ✓ Data tier

Presentation tier:

This is the most visible part of the application which is visible to the user the user interacts through this with the application. It provides interface to the user to input data and there is no logic required. It is dependent on the platform like web application, windows application or android application etc.

Application tier:

It is the middle layer of three tier architecture. This tier perform calculations and operations between I/O requirements. This layer also known as application server. This layer responds to all queries and requests from the user and respond sent through the presentation tier.

Data tier:

The last layer of three tier architecture. It ensures all operations with data, i.e. database management system and basic database operations for functional storage, selection, processing, storage and integrity. All database and tables are stored in this tier.

6.5 Software Process Model

Software process models represent the activities of software development that in which order they are undertaken. It describes the sequence in which the phases of the software lifecycle will be performed. Software process models are used to develop software in a systematic way.

Different software models are used in different scenarios. There are a lot of different software developments process models that developers are using every day fit into a handful of process models: waterfall , spiral, iterative, Prototyping, incremental development etc. but for the development of this system the agile software development is used.

6.5.1 Agile Software Development

Agile is the ability to create and respond to change. It is a way of dealing with, and ultimately succeeding in, an uncertain and turbulent environment. Agile software development is an umbrella term for a set of frameworks and practices based on the values and principles expressed in the Manifesto for agile software development.

6.5.2 Why Agile?

Agile is an iterative approach features are delivered incrementally and the product is continuously developed. It is used to develop software very quickly. It has the ability to create and respond to change.

Agile Process Models

- ✓ Extreme Programming (XP)
- ✓ Adaptive Software Development (ASD)
- ✓ Dynamic Systems Development Method (DSDM)
- ✓ Scrum.
- ✓ Crystal.
- ✓ Feature Driven Development (FDD)
- ✓ Agile Modeling (AM)

These models have their own advantages based on different scenarios. The chosen approach for this project is Extreme Programming (XP).

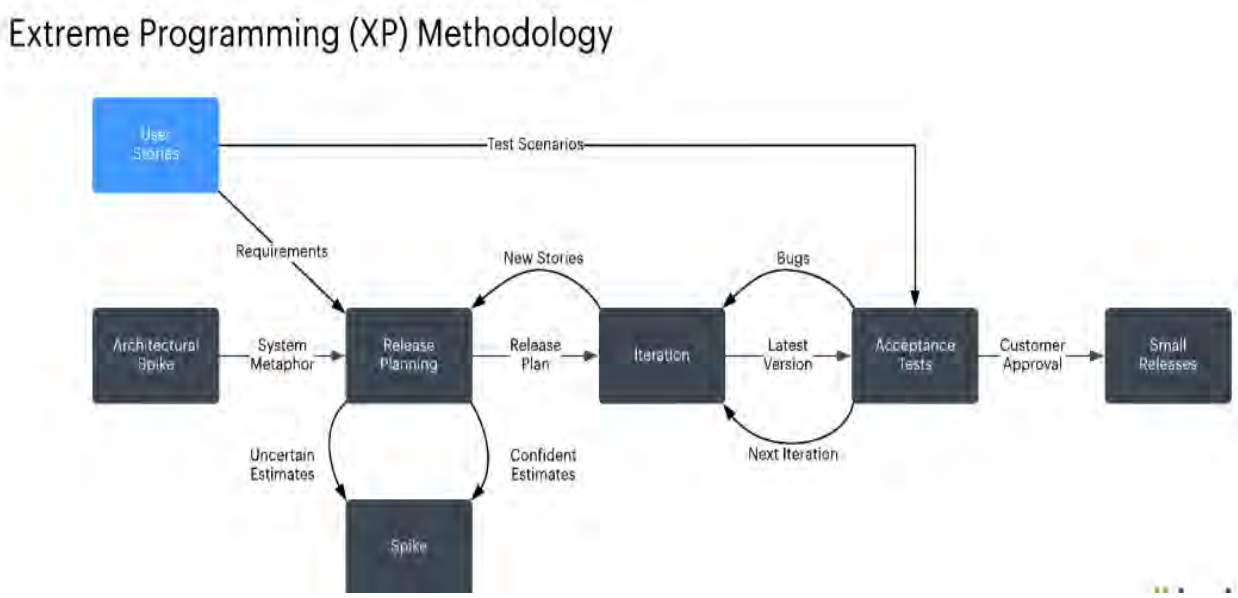

Figure 6.1 Methodology

Key activities:

- ✓ Planning
- ✓ Design
- ✓ Coding
- ✓ Testing

CHAPTER 7

TESTING AND DEBUGGING

7.1 INTRODUCTION:

Testing is aimed at finding errors in the test object and giving confidence in its correct behavior by executing the test object with selected input values. After the development of the system the process of system testing must be performed in order to test if the system is free of bugs. In the system testing behavior of whole system is tested as defined by the scope of the development project or product. It may include tests based on the risks, requirement specifications, business process, use cases, or other high level description of the system behavior, interactions with the operating systems and system resources. Software testing can also provide an objective, independent view of the software to allow the business to appreciate and understand the risks of software implementation. System testing should investigate both functional and non-functional requirements of the testing. System testing is most often the final test to verify that the system to be delivered meets the specification and its purpose. During the system testing, if bugs or errors are detected, they should be corrected immediately.

There are few types of testing that must be carried out:

- Unit testing
- Integration testing
- System testing
- Acceptance testing.

7.2 Software Testing:

Software testing is the process of ensuring the good quality of the software. Testing is an evaluating process which is conducted by the testing team which evaluates whether it meets their requirements or not. It verifies and validates the software to avoid any problems in future.

7.3 Verification:

Verification is the process of evaluating products of a development phase to find out whether they meet the specified requirements

7.4 Validation:

Validation is the process of evaluating software at the end of the development process to determine whether software meets the customer expectations and requirements.

Goals:

Software testing is the process of knowing the expected results and finding out the Bugs and errors. It is the most important part of the project. It is done to ensure that the software is always bug free and easily maintainable. Its basic purpose is to make sure that it is up to the end users expectations? It ensures that we are building the right product which meets the requirements gathered during the requirement elicitation. If a bug stays undetected for a long time after deployment it will eventually show itself after harming the system. It basically covers all the aspects for which we are testing the system before deployment. System testing is most important part of the project. Without this we cannot get confidence to fully deploy it.

The main goals are:

- ✓ To identifying the bugs as early as possible.
- ✓ To find defects in the system.
- ✓ To prevent defects in the system.
- ✓ To make sure that the end results meet the given requirements and specifications.
- ✓ To gain the confidence of the customer to provide best product.

Testing Levels:

Testing levels are done to find out missing areas and to prevent overlapping and repetition between the development life cycle phases. There are different levels of testing which are carried out during system testing:

- ✓ Unit testing
- ✓ Component testing
- ✓ Integration testing
- ✓ Component integration testing
- ✓ System integration testing
- ✓ System testing

- ✓ Acceptance testing
- ✓ Alpha testing
- ✓ Beta Testing

1. **Unit Testing:**

This is done by developers/individuals person to make insure that their code is working fine and their output. In which function, procedure, and classes checks.

2. Component Testing:

The main difference between the unit testing and the component testing is that, in unit testing the developer test their piece of code and in component testing the whole component is tested.

3. Integration Testing:

System testing is done when two modules are integrated. There are few integration testing.

- ✓ Big bang integration testing or top down.
- ✓ Bottom up.
- ✓ Functional incremented.

4. Component Integration Testing:

When modules and components are integrated this testing is called component integration testing.

5. System Testing:

In this the tester basically test compatibility of the application with the system.

6. Acceptance Testing:

Acceptance testing is done and give the results whether the system meets the requirements or not.

7. Alpha testing:

It is done at the developer side. It is done at the end of development process.

7.5 Testing techniques:

There are some testing techniques are used to test the system. Which are as follows.

- ✓ Black box testing.

- ✓ White box testing.
- ✓ Gray box testing.

7.5.1 Black box testing.

In black box testing we do not test data structure and variables. We just give input to the system and checks its output if the output is similar to the expected results then the test is successful. Otherwise unsuccessful. In black box testing we discover interface errors and termination errors. In this, tester does not have access to the source code.

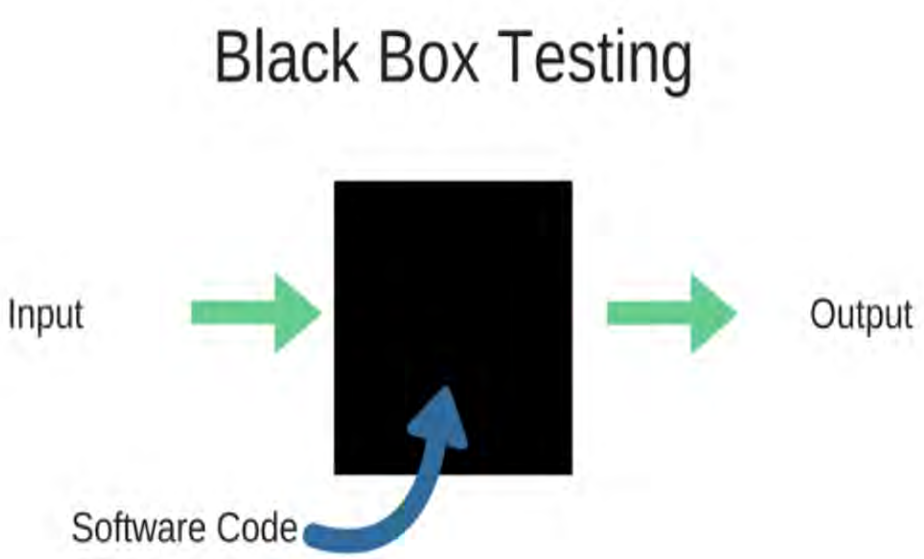

Figure 7.1 Black Box Testing

Advantages

- ✓ Well suited and efficient for large code segments.
- ✓ Code access is not required.
- ✓ Large number of moderately skilled testers can test the application with no knowledge of implementation, programming language, or operating systems.

Disadvantages

- ✓ Inefficient testing, due to the fact that the tester only has limited knowledge about an application.
- ✓ Limited coverage, since only a selected number of test scenarios are actually performed.

✓ Blind coverage, since the tester cannot target specific code segments or error-prone areas.

7.5.2 White box testing

White box testing is the detail investigation of internal structure code and logic. White box testing is also called glass testing or open box testing. In this, the tester have to access to the internal structure of the system.

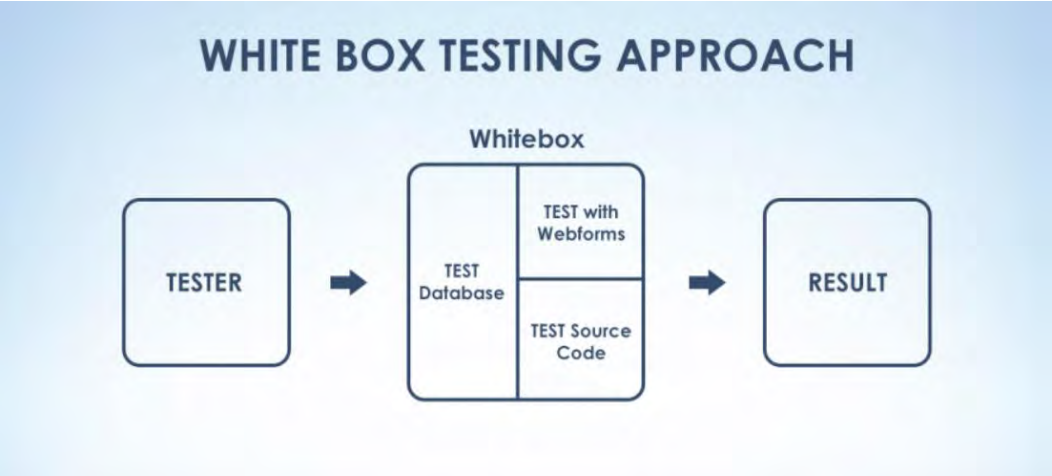

Figure 7.2 White Box Testing

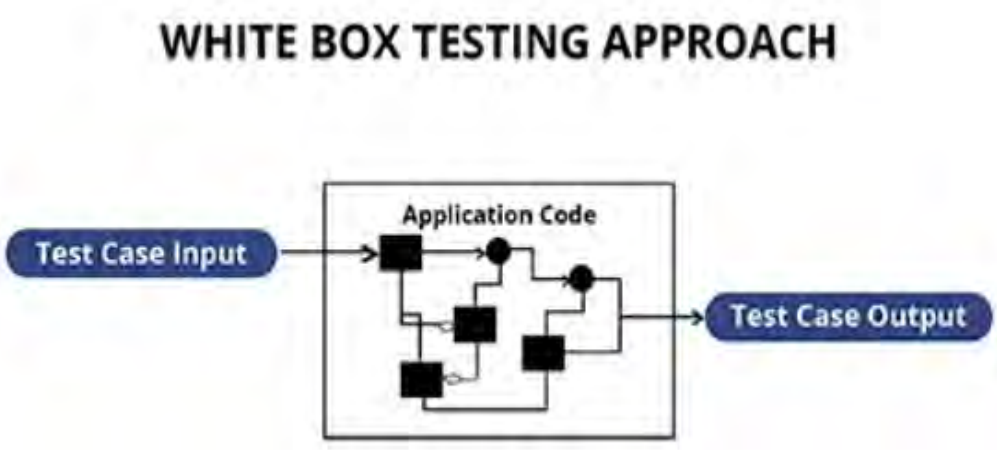

Figure 7.3 White Box Testing Approach

Advantages:

- ✓ It helps in optimizing the code.
- ✓ Extra lines of code can be removed which can bring in hidden defects.
- ✓ As the tester has knowledge of the source code, it become very easy to find out which type of data can help in testing the application.

Disadvantages

- ✓ Sometimes it is impossible to look into every corner to find out hidden errors that may create problems, as many paths will go untested.
- ✓ It is difficult to maintain white-box testing, as it requires specialized tools like code analyzers and debugging tools.

7.6 Gray box testing

Gray box testing is a technique to test the application with having a limited knowledge of the internal workings of an applications. Gray-box testing is a combination of white-box testing and black-box testing. The aim of this testing is to search for the defects if any due to improper structure or improper usage of applications.

Figure 7.4 Gray Box Testing

a. Test Cases

Test cases are described in tabular form. A test case is a set of conditions or variables on the basis of which a tester determines that the requirements on an application is fully satisfied or not. A test has components that describe an input, action or event and an expected response to determine if a feature of an application is working correctly.

FUNCTIONS TO BE TESTED:

- ✓ Login
- ✓ Add Receptionist
- ✓ Add Tests
- ✓ Add Patient
- ✓ Generate Report
- ✓ Generate Monthly Report
- ✓ Print Report

Test Cases of Application

Test no.	Function to Test	Testing approach/White Box
1	Login	Black box testing/White Box
2	Add Receptionist	Black box testing/White Box
3	Add Tests	Black box testing/White Box
4	Add Patient	Black box testing/White Box
5	Generate Report	Black box testing/White Box
6	Generate Monthly Report	Black box testing /White Box
7	Print Report	Black Box testing/Whit Box

Table 1: Test Plan

Table 2: Test case for Login

Test case ID	01
Tester	Asfand Yar
Test type	Black box testing/White Box
Test case name	Login
Description	Validation of username and password.
Expected result	If valid username/password address than user will be logged in.
Actual result	Error showed if not a valid username or password.
Status	Successful.

Table 2: Test Plan

Table 3: Test case for Add Receptionist

Test case ID	02
Tester	Asfand Yar
Test type	Black box testing/White Box
Test case name	Add Receptionist
Description	Receptionist can be added by admin.
Procedure	Receptionist Detail will be provided.
Expected result	Successfully Added.
Actual re-sult	Error if information is not provided.
Status	Successful.

Table 3: Test Plan

Table 4: Test case for Add New Tests

Test case ID	03
Tester	Asfand Yar
Test type	Black box testing/White Box
Test case name	Add New Tests
Description	Add Tests has been done by the Admin
Procedure	Admin should add new tests that is provided.
Expected result	Successfully done
Actual re-sult	False Values
Status	Successful.

Table 4: Test Plan

CHAPTER 8

FUNCTIONALITY AND COMMANDS

8.1 FUNCTIONALITES:

The program should firstly be started on the Desktop Systems; the initial mode of the program is Login since this program aims at making a secure assistant program. After the program has been started, the admin should add receptionist. The main work will done by the receptionist and he/she will add new patient, maintain their record.

Receptionist can also generate patient report, daily /monthly report of the patient. Admin can also view the patient report on the GUI Dashboard. Patient‟s records are kept in database. Patient can search by their names and generate report in print form.

8.1.1 Select Test

The Receptionist can select test for the patient, that recommended by the doctor.

8.1.1 Print Form

Patient get the test details in the printed form, with medical laboratory detail.

8.1.3 Date and Time

Current date and time are available on admin/receptionist dashboard.

8.2 Maintenance

After the program is completed, the program still needs long term maintenance to make it available and stable to execute. The program will be test after a certain period of time and debug each of the function and possible bugs, whenever a potential bug is detected; the program needs to be refined to a better design.

Meanwhile, there will update and add more data to the database to increase the database capacity. Depending on the new keywords, responses, relevant data found that could be applied in this program; the database will always be improved and can handle more and more cases.

8.3 System Documentation

System documentation is a crucial aspect of implementation process. It describes the working of components and serves as a method of communication between application developers and users. It also helps future analysis of application either by the same or different system analysts and developers.

To setup the system, there must be visual basic 15 software installed on the computer before it can work.

8.4 Over View

The medical laboratory management system provides functions on identify good test usages instruction, minimize human errors in medication safety, facilitate accessibility of test‟ information and information management among employees, providing optimal tests movement in laboratory unit, enable reports with in significantly short period of time, despite simultaneous usage of database for the purpose stated above.

The system will solve the problem of the current system by minimizing time wastage and reduce resources which simply change manual based system to computerized system.

8.5 Data Base Specification

A database is a single file which consists of structured data and records which are stored in minimum or no duplication of data. It is therefore a constructed, consistent and controlled pool of data. A good database must be common to all users and independent of the programs which use it to generate output.

However, My SQL was used as the database application tool for designing the database management system. The database management system is limited only to database administrator (Management). Whilst the system designer / developer / programmer is responsible for maintaining and upgrading of the database and the whole software.

CHAPTER 9

CONCLUSION AND FUTURE WORK

'Our imagination is the only limit to what we can hope to have in the future.'

Charles F. Kettering

In this thesis, the problem of Medical laboratory related which are faced were addressed. And the main contribution of this project is to provide the authentic information to the Owner regarding patients and also about total collection. Adding new receptionist, new tests, add patient and add generate report like functionalities have been achieved. It ensures the privacy of the patient data and secures its access from unauthorized users. This project is not launched yet. Due to lack of time many things have been left and many things will be added and improved in future, but a better user interface with attractive styles have been achieved

Due to lack of time many things have been left for future. The project will be enhanced with the passage of time and according to the needs.

More things to add in future:

1. Doctors record will be added
2. Patient Report can be sent through Email which can be used in future.
3. In future we can make further changes to the interface.