CSE — 180007,

Diagnostic Center Bill Management System

A Project Report Submitted By

Md. Hafizur Rahman
ID: CSE1402002010

Esrat Jahan Eity
ID: CSE1501004007

Kanij Farjana Rupa
ID: CSE1501004018

B. M. Hanif
ID: CSE1501004021

Akramul Haque Shawon
ID: CSE1501004025

In Partial Fulfillment of the Requirements for the degree of
Bachelor of Computer Science and Engineering

Department of Computer Science and Engineering

Faculty of Science and Engineering

Sonargaon University (SU).

Submission Date: 15th September, 2018

Declaration

We are Md. Hafizur Rahman, Esrat Jahan Eity, Kanij Farjana Rupa, B. M. Hanif and Akramul Haque Shawon, students of Department of Computer Science and Engineering, Sonargaon University (SU), declaring that this project paper on the stated topic has only been prepared for the fulfillment of CSE-400 Project/Thesis, as the partial fulfillment of "Bachelor of Computer Science and Engineering" degree.

It has not been prepared for any other purposes, rewards, or presentation.

hafiz 15.09.18

Md. Hafizur Rahman
ID: CSE1402002010

Esrat 15.9.18

Esrat Jahan Eity
ID: CSE1501004007

Kanij farzana 15.9.18

Kanij Farjana Rupa
ID: CSE1501004018

15.09.18

B. M. Hanif
ID: CSE1501004021

Shawon 15. 9. 18

Akramul Haque Shawon
ID: CSE1501004025

Acceptance

This is to certify that Project report on **"Diagnostic Center Bill Management System"** has been carried out by Md. Hafizur Rahman (CSE1402002010), Esrat Jahan Eity (CSE1501004007), Kanij Farjana Rupa (CSE1501004018), B. M. Hanif (CSE1501004021) and Akramul Haque Shawon (CSE1501004025), students of Department of Computer Science and Engineering, Sonargaon University (SU) as a partial fulfillment of the requirement of CSE-400 Project/Thesis. The report has been prepared under my guidance and is a record of the bona fide work carried out successfully. To the best of my knowledge and as per their declaration, no parts of this report has been submitted anywhere for any degree, diploma or certificate.

Now they are permitted to submit the report. I wish their all success in their future endeavors.

Project Supervisor

Arif 15.09.18

Arifur Rahaman

Lecturer,

Department of Computer Science and Engineering,

Sonargaon University (SU)

Acknowledgement

In the name of Allah most Gracious, most Merciful.

First of all, we sincerely like to thank **Arifur Rahaman**, Lecturer, Department of Computer Science and Engineering, Sonargaon University, for giving me the opportunity to complete our project and report. We would have been able to make this report effectively and properly only for his right direction. Beside that we again like to thank him to give us an opportunity to submit this report.

We would also like to thank and we are very appreciative to **Mohammad Naderuzzaman**, Head of Department of computer Science and Engineering, Sonargaon University for his unrelenting direction and sustain throughout the semester.

It is our pleasure to be grateful to our families, friends for their mental support throughout this work. Lastly, we would also like to thank our Vice Chancellor **Professor M. A. Razzaque** who has given us the opportunity to study in such a decent environment.

Our success could not turn into reality without these people who helped me in different ways.

Abstract

Diagnostic Center Bill Management System (DCBMS) is a web based online project. The main aim of the project is to store the billing information of the test details in a hospital or a clinic.

DCBMS is an intranet based private web application. The objective of the project work is to develop effective software for maintaining the information relating to the bill details. This project is completely menu driven and user friendly.

It gives details about various tests for patients and purchase section. In addition, it has a provision for generating reports in different formats and hence reports are useful in strategic and tactic decision-making .The ultimate aim of implementing this project is to automate the Bill Management System of the concern. Totally this project is towards inducing complete Billing through computer with maximum user-interaction and error-free information.

There are outpatient department, inpatient department, nursing stations, billing counter, pharmacies, central drug stores, diagnostic center, ambulance, intensive care units, operation theatres, emergency ward, mortuary, patient room management, staff management and training departments. Each department has its own manager and few staff given to it to manage its operations.

All staff are trained to use the healthcare software provided by the management. The staff can never afford to forget data entry as otherwise they will not be able to fulfill their duties with appropriate inputs.

List of Figures

Table of Contents

Chapter 1: Introduction

1.1 Introduction of project

Diagnostic center Bill Management system (DCBMS) will manage billing activities for a small diagnostic center. Generally small diagnostic center offers several types of tests like X-ray, Urine test, CBC, blood test etc. Bill processing of those type of tests require paper work and is very error prone. So this application will process the bill automatically from their side.

This project work is to develop useful software for maintaining bill generate and store all billing information for future requirement. This application is very easy to operate and user friendly.

1.2 Aim of Project

This web application will manage billing activities for a small diagnostic center. Generally, small diagnostic offers several types of tests like Complete Blood Count, Lipid Profile, Urine C/S-200, X-Ray LS Spine, USG Lower abdomen etc. Since bill processing for these types of tests requires paper work and is very error prone, we are offering this application to process it automatically from their side.

1.3 System Study & Analysis

System analysis is an explicit formal inquiry carried out to help someone identify a better course of action and make a better decision than he might otherwise have made. Systems analysis is a problem solving technique that decomposes a system into its component pieces for the purpose of the studying how well those component parts work and interact to accomplish their purpose. System analysis is a process of gathering and interpreting facts, diagnosing problems and the information to recorded improvements on the system. It is a problem solving activity that requires intensive communication between the system user and system developer. System analysis or study is an important phase of any system development process. A detailed study of the process must be made by various techniques like interviews, questionnaires etc. The data collected by these sources must be scrutinized to arrive to conclusion every synthesis is built upon the results of a preceding analysis and every analysis requires a subsequent. The characteristic attributes of a problem situation where systems analysis is called upon are complexity of the issue and uncertainty of the outcome of any course of action that might reasonably be taken. That part or aspect of systems analysis that concentrates on finding out wherever and indeed course of scion violates any constraints is referred to as feasibility analysis [1].

1.4 Methodology of our project

Methodology is the systematic, theoretical analysis of the methods applied to a field of study. It comprises the theoretical analysis of the body of methods and principles associated with a branch of knowledge. Typically, it encompasses concepts such as paradigm, theoretical model, phases and quantitative or qualitative techniques [6].

A methodology does not set out to provide solutions - it is, therefore, not the same as a method. Instead, a methodology offers the theoretical underpinning for understanding which method, set of methods, or best practices can be applied to specific case, for example, to calculate a specific result.

The methodology is the general research strategy that outlines the way in which research is to be undertaken and, among other things, identifies the methods to be used in it. These methods, described in the methodology, define the means or modes of data collection or, sometimes, how a specific result is to be calculated. Methodology does not define specific methods, even though much attention is given to the nature and kinds of processes to be followed in a particular procedure or to attain an objective.

Methodology and method are not interchangeable. In recent years however, there has been a tendency to use methodology as a "pretentious substitute for the word method" Using methodology as a synonym for method or set of methods leads to confusion and misinterpretation and undermines the proper analysis that should go into designing research. There are many types of Methodology; we choose AGILE method for our project.

1.4.1 Agile

Agile is a time boxed, iterative approach to software delivery that builds software incrementally from the start of the project, instead of trying to deliver it all at once near the end.

It works by breaking projects down into little bits of user functionality called user stories, prioritizing them, and then continuously delivering them in short two week cycles called iterations.

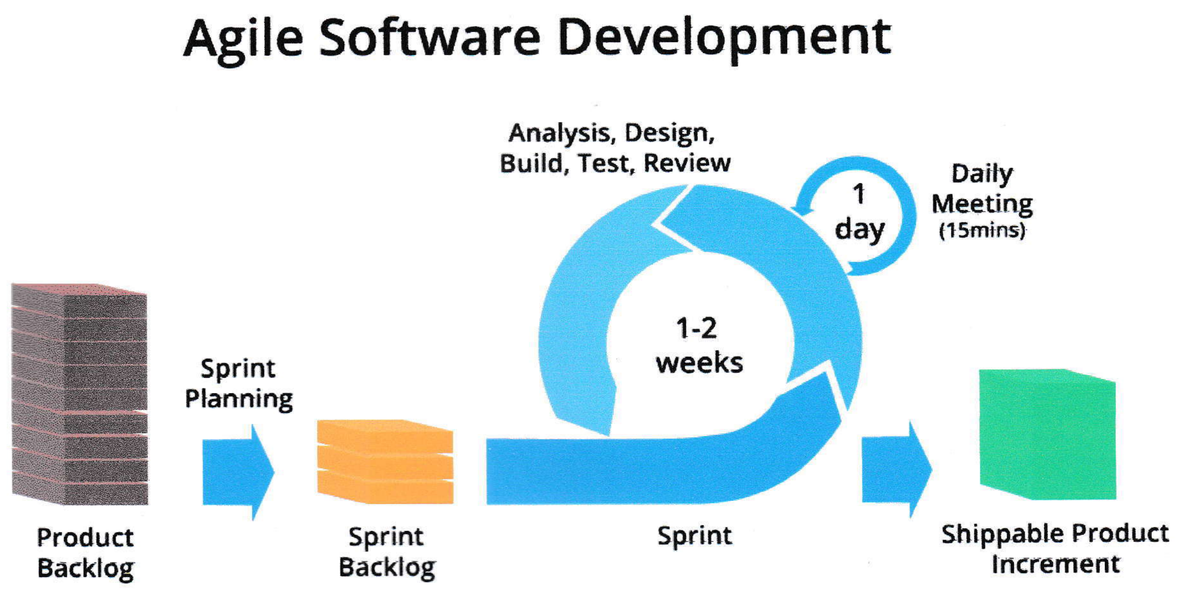

Agile Software Development
Analysis, Design, Build, Test, Review
1 day
Daily Meeting (15mins)
1-2 weeks
Sprint Planning
Product Backlog
Sprint Backlog
Sprint
Shippable Product Increment

Figure 1.1: Agile Project management methodology

1.4.2 Cause of using agile methodology

Agile Methodology is a type of project management process. The agile method anticipates change and allows for much more flexibility than traditional methods. Clients can make small objective changes without huge amendments to the budget or schedule. The process involves breaking down each project into prioritized requirements, and delivering each individually within an iterative cycle. Iteration is the routine of developing small sections of a project at a time. Each iteration is reviewed and assessed by the development team and client. The insights gained from the assessment are used to determine the next step in development. Clients come to prescheduled regular meetings to review the work completed the previous iteration, and to plan work for the upcoming iteration. Detailed goals are set in each iteration meeting such as; expected changes, time estimates, priorities and budgets [6].

The agile method is based on giving high priority to customer participation, from the very beginning of the development cycle. The objective is to keep the client involved at every step so that they have a product that they are happy with at the end. This method saves the client money and time because the client tests and approves the product at each step of development. If there are defects or challenges, then changes can be made during production cycles to fix the issue. Traditional models of project management would not find defects as early because they do not test as often. Typically (in traditional methods of production) defects that are not discovered at the different stages can find their way into the final product. This can result in increased overhead prices and client dissatisfaction.

Businesses have proven this model of project management with their increased client satisfaction rate. The values for businesses that use this model include:

- Lower Cost
- Enables clients to be happier with the end product by making improvements and involving clients with development decisions throughout the process.
- Encourages open communication among team members, and clients.
- Providing teams with a competitive advantage by catching defects and making changes throughout the development process, instead of at the end.
- Speeds up time spent on evaluations since each evaluation is only on a small part of the whole project.
- Ensures changes can be made quicker and throughout the development process by having consistent evaluations to assess the product with the expected outcomes requested.
- It keeps each project transparent by having regular consistent meetings with the clients and systems that allow everyone involved to access the project data and progress.

Businesses use this model of project management to ensure that throughout the process customers save time, money, and have the flexibility to make changes anytime during the development process.

1.4.3 Advantages of Agile model

- Customer satisfaction by rapid, continuous delivery of useful software.
- People and interactions are emphasized rather than process and tools. Customers, developers and testers constantly interact with each other.
- Working software is delivered frequently (weeks rather than months).
- Face-to-face conversation is the best form of communication.
- Close daily cooperation between business people and developers.
- Continuous attention to technical excellence and good design.
- Regular adaptation to changing circumstances.
- Even late changes in requirements are welcomed.

1.4.4 Listed below are a few disadvantages of Agile

- In case of some software deliverables, especially the large ones, it is difficult to assess the effort required at the beginning of the software development life cycle.
- There is lack of emphasis on necessary designing and documentation.
- The project can easily get taken off track if the customer representative is not clear what final outcome that they want.
- Only senior programmers are capable of taking the kind of decisions required during the development process. Hence it has no place for newbie programmers, unless combined with experienced resources.

Agile Developments also fails at times due to Unrealistic Expectations – Agile actually is and what it can help you achieve. Agile is commonly believed to be a set a practices, processes and tools, when in fact, Agile is really more of a mind-set and culture.

Chapter 2: Existing System

2.1 Process of Existing System

In the existing system, the small diagnostic center have only manually like paper- pen based system. The system is given below:

- They have no software or online system to control their system and management database.
- They are using Microsoft Office Excel or a ledger book for their database. Every year, to store their billing information they are using new Excel sheet.
- As it is a manual process for this, they need many staff to manage their billing system.

2.2 Problems with Existing System

After analyzing the process of existing system and market different kinds of software, we get the following improvements and problems, they are given below:

- There is no software for bill management.
- Every year they need a new Microsoft Office Excel Sheet or a ledger book.
- The patient registration form are stored in hard copy. That's why they have facing many problems to find out the patient information. Possibility to loss of information.
- Patient do different type of tests at a time, the tests fee are count manually or with help of a calculator.
- There is no way to find the type wise or test wise billing report.
- Too hard to find the specific some days billing report and unpaid report.
- Lack of security of data.
- Need many staff.

Chapter 3: Proposed System & Design

3.1: Aim of Proposed System

Diagnostic center Bill Management system is a web based online project. The aim of the project is store the information of the test details in a hospital or a clinic. It gives details about various tests for patients and purchase section. In addition to that it has a provision for generating reports in different formats and hence reports are useful in strategic and tactic decision-making. The ultimate aim of implementing this project is to automate the Bill Management System of the concern. Totally this project is towards inducing complete Billing through computer and error-free information.

3.1.1 Advantage of the Proposed System

When it comes to using a DCBMS system, the prime reason is to log and manage billing system. The administrators can easily add the tests type in the system. When a patient comes to do diagnostic tests, the administrator can take the patient test requests and generate a patient bill copy. After that it's an option to pay the test fees, as it is a small diagnostic center, partial payment is available for the system. Administrators can find and generate the monthly, day wise report to manage the total bill.

3.2 System Feasibility Study

The system feasibility study is an important tool in engineering, management and business sector. System feasibility study is made to see if the project on completion will serve the purpose of the organization for the amount of work, effort and the time that spend on it. Feasibility study helps the developer to their future of the project. Feasibility is a formal report the documents decisions concluding a choice of one from two or more alternatives. A feasibility study of a system proposal is according to its workability which is the impact on the organization to their ability to meet their user needs and effective use of resources [1].

In the system feasibility study, the problem is carefully described and one or more potential solutions to the problem are considered in some depth. Three category of system feasibility, Technical Feasibility, Economical Feasibility and Operational Feasibility are given below:

3.2.1 Technical Feasibility

The system must be evaluated from the technical point of view first. In our project, we use HTML, CSS, JavaScript, C# programming language and Microsoft SQL server 2012. For our project, the hardware part is simple like, processors core i3, motherboard Gigabyte 61, RAM 4 GB etc. and software part is operating system Windows 8 / Windows 10, Notepad++, Microsoft SQL server 2012, Visual Studio 2013 etc. All the technology which is mention above is ready to use. So, that our project is technically feasible [1].

3.2.2 Economical Feasibility

The developing system must be justified by cost and benefit. Criteria to ensure that effort is concentrated on project, which will give best, return at the earliest. By investing our resource including time, money, hardware and software renouncement, we are economic feasibility. It will also less time consuming. So, it is intangible benefit. So that we are economic feasibility.

3.2.3 Operational Feasibility

Our system is user friendly by using textboxes, button and so on. So that our project system is operational feasible. In our system, there is nothing complexity and all thing are fully friendly usable. No need for tutorial and user guide line. So that our system is operationally feasible.

3.3 Proposed System Design

There are two design part of proposed system, Logical Design & Physical Design. Design of proposed system is given below:

3.3.1 Logical Design

The logical flow of a system and define the boundaries of a system. It includes the following steps:

- Reviews the current physical system, data flows, file content, other content etc.
- Prepares output specifications, determines the format, content and other content of reports.
- Prepare input specifications, format, content and most of the input functions.
- Prepares edit, security and control specifications.
- Specifies the implementation plan.
- Prepares a logical design walk through of the information flow, output, input, controls and implementation plan.
- Reviews benefits, costs, target dates and system constraints.

3.3.2 Physical Design

Physical system produces the working systems by define the design specifications that tell the programmers exactly what the candidate system must do. It includes the following steps.

- Design the physical system.
- Specify input and output media.
- Design the database and specify backup procedures.
- Design physical information flow through the system and a physical design.
- Plan system implementation.
- Determine training procedures, courses and timetable.
- Update benefits, costs, and conversion date and system constraints.

3.4 Use Case Diagram

Admin/Staff Functionality:

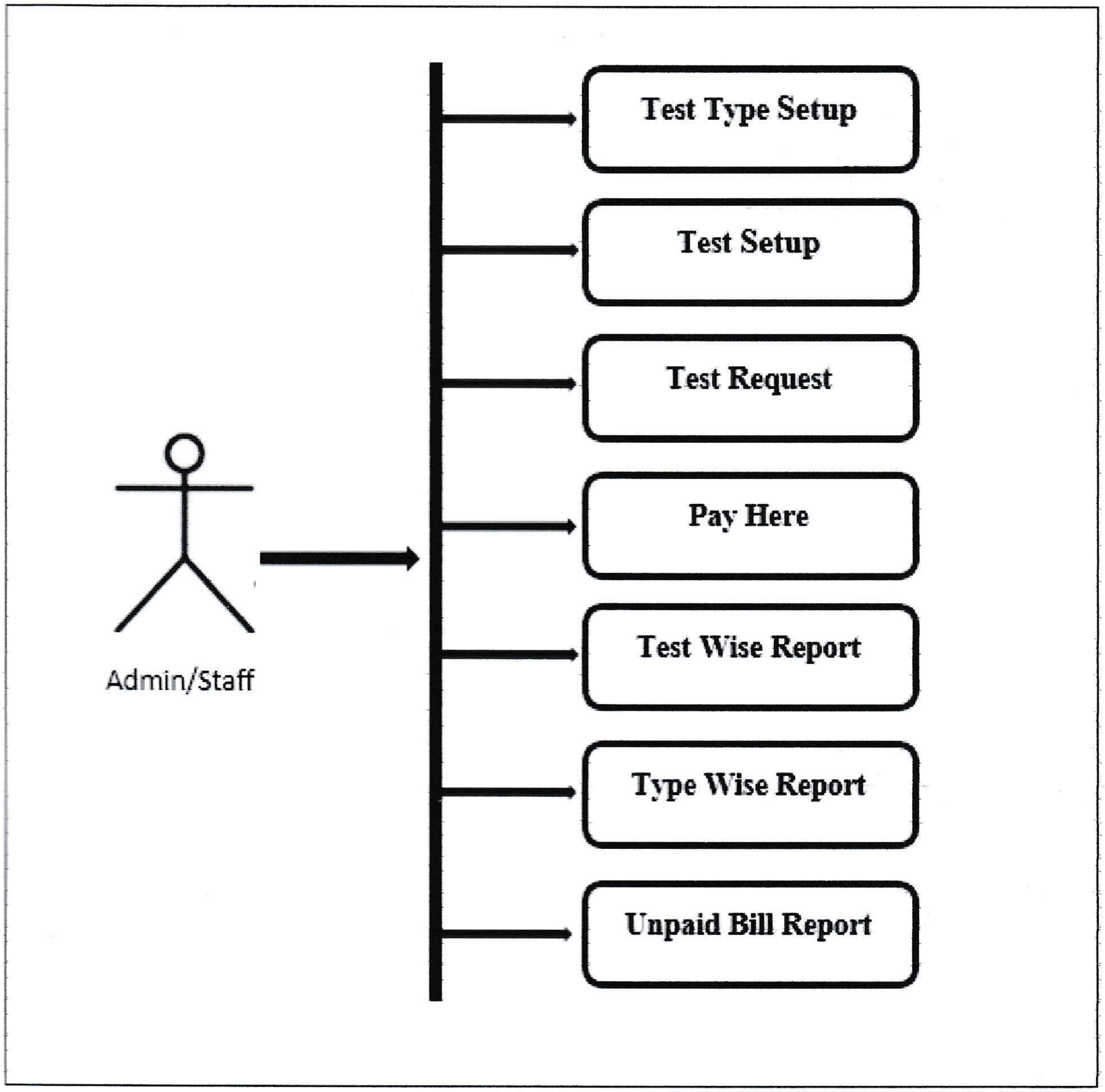


Figure 3.4.1: Use Case Diagram of DCBMS

Chapter 4: System Requirement and Cost Estimation

4.1 System Requirements & Cost

4.1.1 Hardware Requirements

Hardware Requirements of our project are given below:

Mother Board: Gigabyte 61 chipset

Processor: Intel Core i3, 3.0 GHz Clock speed

RAM: 4 GB or more

Hard Disk: 512 GB or more

Monitor: LED

Keyboard: 104 Keys

Mouse: 3 buttons

Network: Local Network

4.1.2 Software Requirements

Software Requirements of our project are given below:

Operating System: Windows 7 / Windows 10

Front-end: HTML, CSS

Back-end: Microsoft SQL Server 2012

Framework: Dot NET 4.5

Language: C#, SQL

IDE/Editor: Visual Studio 2013, Notepad++

4.1.3 Project cost Estimation

Cost estimation describes the expense that needs to spend during project development. Project Cost Estimation is given below:

Personal Cost:

Persons Required: 4

Post: Coder & UI/UX Design

Salary: 50.00/h

Working Duration: 8 Months

Weeks: 32

Total working hours = 32x3x4= 384 hours

Total salary = 50x384=19200.00

Total Salary of Four persons= 76800.00

Hardware Cost:

Cost of computer	35000.00	Cost of Printer	7000.00
Computer life	1 year	Printer life	1 year
Computer usage	8 months	Printer usage	8 months
Computer cost (Depreciation cost)	(35000/12)x8= 23333.33	Printer cost (Depreciation cost)	(7000.00/12)x8= 4,666.00
3 person computer cost	23333.33x 3=70000.00	1 person printer cost	4,666.00x1= 4,666.00

Total Hardware Cost **= (70000.00+4666.00)**

= 74666.00

Software cost:

Description	Cost	Depreciation expense	Depreciation Cost
Microsoft windows 10	10,000.00	(10,000.00/12)*8	6666.00
Microsoft office 2010	10,000.00	(10,000.00/12)*8	6666.00
Dream waver DC	1,500.00	(1,500.00/12)*8	1000.00
		Total=	14332.00
		For Three person software cost=	42998.00

Total Project Cost =Personal cost + Hardware Cost+ Software cost

=76800.00+74666.00+42998.00

=194464.00(BDT)

Chapter 5: Risk Engineering

5.1 Risk Management

A risk is a serious problem that might or might not happen. It is necessary to analyze the potential risks in a project. If the risks of a software project are not properly analyzed and estimated, many problems can plague the software project [11]. Risk analysis and management are a series of steps that help a software team to understand and manage uncertainty. To establish a risk management model the following phases are followed:

Identification: Risk identification is the process of detecting potential risks or hazards through data collection. A range of data collection and manipulation tools and techniques exists. The team is using both automated and manual techniques to collect data and begin to characterize potential risks to Web resources. Web crawling is one effective way to collect information about the state of Web pages and sites.

Classification: Risk classification is the process of developing a structured model to categorize risk and fitting observable risk attributes and events into the model. The team combines quantitative and qualitative methods to characterize and classify the risks to Web pages, Web sites, and the hosting servers.

Assessment: Risk assessment is the process of defining relevant risk scenarios or sequences of events that could result in damage or loss and the probability of these events. Rosenthal describes the characteristics of a generic standard for risk assessment as "transparent, coherent, consistent, complete, comprehensive, impartial, uniform, balanced, defensible, sustainable, flexible, and accompanied by suitable and sufficient guidance.

Analysis: Risk analysis determines the potential impact of risk patterns or scenarios, the possible extent of loss, and the direct and indirect costs of recovery. This step identifies vulnerabilities, considers the willingness of the organization to accept risk given potential consequences, and develops mitigation responses. Implementation: Risk management implementation defines policies, procedures, and mechanisms to manage and respond to identifiable risks. The implemented program should balance the value of assets and the direct and indirect costs of preventing or recovering from damage or loss [3].

To take comprehensive care of a web based system we must consider the following points:

1. Hardware and software environment including any upgrades to the operating system and Web server, the installation of security patches, the removal of insecure services, use of firewalls, etc.
2. Administrative procedures such as contracting with reputable service providers, renewing domain name registration, etc.
3. Network configuration and maintenance including load balancing, traffic management, and usage monitoring.
4. Backup and archiving policies and procedures including the choice of backup media, media replacement interval, number of backups made and storage location.
5. Physical location of the server and its vulnerability to fire, flood, earthquake, electric power anomalies, power interruption, temperature fluctuations, theft, and vandalism.

There are different categories of risks that should be considered in any software project. The following categories of risks have been considered in this software project.

1. Project risks: These risks threaten the project plan. If these risks become real, it is likely that the project schedule will slip and that costs will increase. Project risks identify potential budgetary, schedule, personnel, resource, customer and requirement problems and their impact on the software project.

2. Technical risks: These risks threaten the quality and timeliness of the software to be produced. If a technical risk becomes a reality, implementation may become difficult or impossible. Technical risks identify potential design, implementation, interface, verification and maintenance problems. Moreover, specification ambiguity, technical uncertainty, technical obsolescence are also risk factors.

3. Business risks: These risks threaten the viability of the software to be built. The business risks can be market risks, building a system that no one really wants. Strategic risks, building a system that no longer fits into the overall business strategy for the company. Management risks, losing the support of senior management due to a change in focus or a change in people. Budget risks, losing budgetary or personnel commitment.

5.2 The RMMM plan

1. Risk Mitigation: Proactive planning for risk avoidance.

2. Risk Monitoring: Assessing whether predicted risks occur or not, ensuring preventive steps are being properly applied, collect information for future risk analysis, attempt to determine which risks caused which problem.

3. Risk Management: Actions to be taken in the event that mitigation steps have failed and the risk has become a live problem.

Type of Impact: Catastrophic (1), Marginal (2), Tolerable (3), Critical (4).

Type of Probability: very low (<10%), low (10–25%), moderate (25–50%), high (50–75%), very high (>75%).

Project Risk (P01)	Date: 18-02-2018
Name	Changes the requirements
Probability	Low (15%)
Impact	Marginal (2)
Description	Company may change their requirements
Mitigation & Monitoring	Requirements are redefined by the company due to time or business needs. Meeting will be held with the company regularly. This insures that the product we are producing solves a problem.
Management	Emergency meeting between both parties to identify new project requirements and goals.
Status	Not occur

Project Risk (P02)	Date: 05-04-2018
Name	Poor Quality Documentation
Probability	Low (25%)
Impact	Marginal (2)
Description	Poor quality documentation of the members
Mitigation & Monitoring	Meeting will be held routinely to offer documentation suggestions and topics. The progress on documentation will also have a monitor in each meeting.

Management	The addition of new topics or removal of unnecessary topics into the documentation will assigned to responsible person.
Status	Not occur

Business Risk (B01)	Date: 20-05-2018
Name	Insufficient Budget
Probability	Moderate (35%)
Impact	Marginal (3)
Description	If the budget is low project may not complete.
Mitigation & Monitoring	The project needs streaming server that is costly to set-up. We find several alternative streaming services to reduce the budget risk.
Management	Refinement in project goal. A new plan for regulate the budget.
Status	Problem resolved.

Business Risk (B02)	Date: 27-06-2018
Name	End Users Accept System
Probability	Low (20%)
Impact	Critical (4)
Description	The system fails to gain user's faith
Mitigation & Monitoring	In order to prevent this from happening, the software will develop with the end user in mind. The user-interface will design in a way to make use of the program convenient and pleasurable.
Management	Training the users to familiarize them with the new system. Releasing patches/bug fixes for greater user satisfaction.
Status	The risk has not been arisen yet.

Technical Risk (T01)	Date: 11-07-2018

Name	Lack of Experience
Probability	Low (20%)
Impact	Tolerable (3)
Description	Lack of members experience
Mitigation & Monitoring	The development cost of the software may increase by 20%. Consult with the System Analyst during the system analysis, design and testing phase of the software project.
Management	Though the development cost is increased by 20%, the project is still feasible. Set appointment for formal meeting with the System Analyst to solve different problems of each of the phases.
Status	The risk has not been arisen yet.

Technical Risk (T02)	Date: 05-08-2018
Name	Poor Comments in Code
Probability	Low (15%)
Impact	Catastrophic (1)
Description	Poor comment in the coding part.
Mitigation & Monitoring	Proper coding grammar is followed to make sure that the codes are easily understandable and reusable.
Management	In emergency a refinement in code commenting should be done. This may slow down the development process but it will help the developers in long run.
Status	The risk has not been arisen yet.

Technical Risk (T03)	Date: 11-09-2018
Name	Computer Crash
Probability	High (60%)
Impact	Tolerable (3)
Description	Computer can be crash.
Mitigation & Monitoring	We should take proper follow up of computers. We also take regular data backup every day, We can use IPS to stop

	unexpected shutdown.
Management	If our computer has been crashed then we will restore backup.
Status	We are not facing such kind of problem yet.

Chapter 6: System Design & Implementation

6.1 System Implementing & Testing

Implementation is the stage of the project where the theoretical design is turned into a working system. It can be considered to be the most crucial stage in achieving a successful new system gaining the users confidence that the new system will work and will be effective and accurate.

Software Testing is the process of executing software in a controlled manner, in order to answer the question like; does the software behave as specified? Software testing is often used in association with the terms verification. Software testing is just one kind of verification, which also used techniques such as reviews, analysis, inspections and walk through. Software testing should not be confused with debugging. Debugging is the process of analyzing and localizing bugs when software does not behave as expected. Although the identification of some bugs will be obvious from playing with the software. Testing is a process of executing a program with intend of finding an error. A good test case is one that has high possibility of finding an undiscovered error. A successful test is one that uncovers an undiscovered error. If a testing is conducted successfully according to the objectives as started above, it would uncovered errors in the software also testing demonstrate that the software faction appear to be working according to specification, that performance requirement appear to have been met [6].

6.1.2 Project Diagram

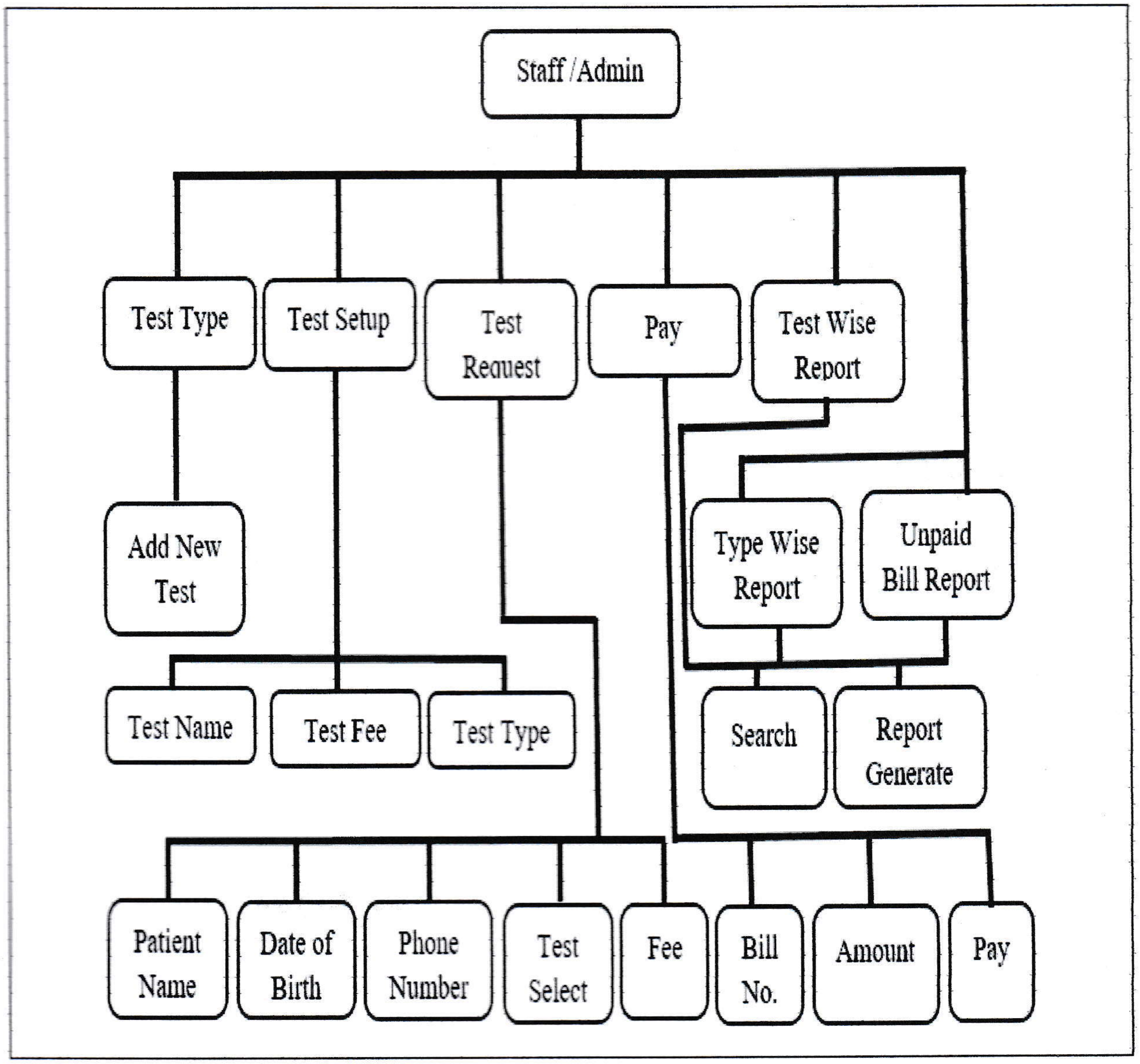


Figure 6.1.1: Project Diagram Structure

Layered design with a database class:

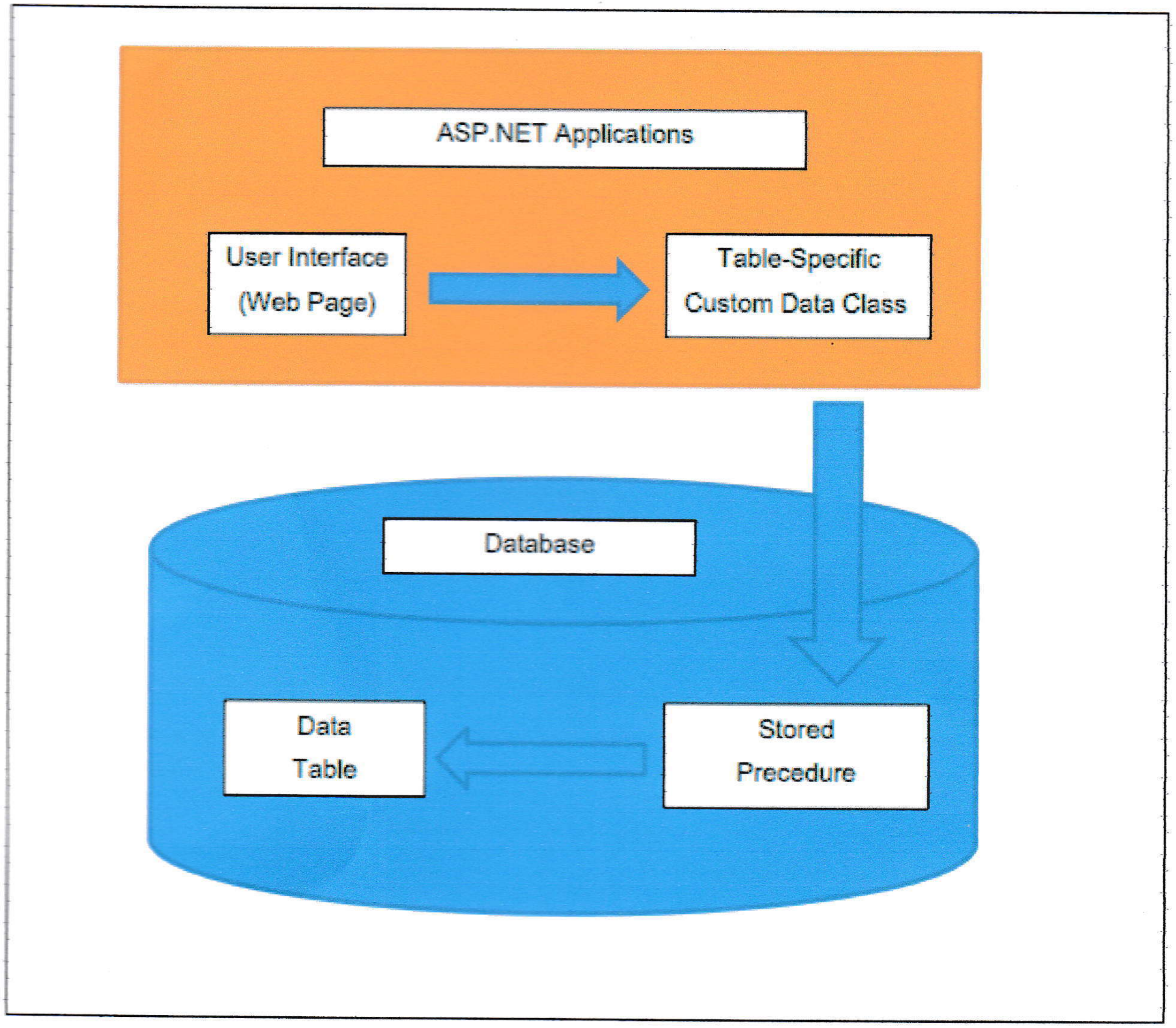
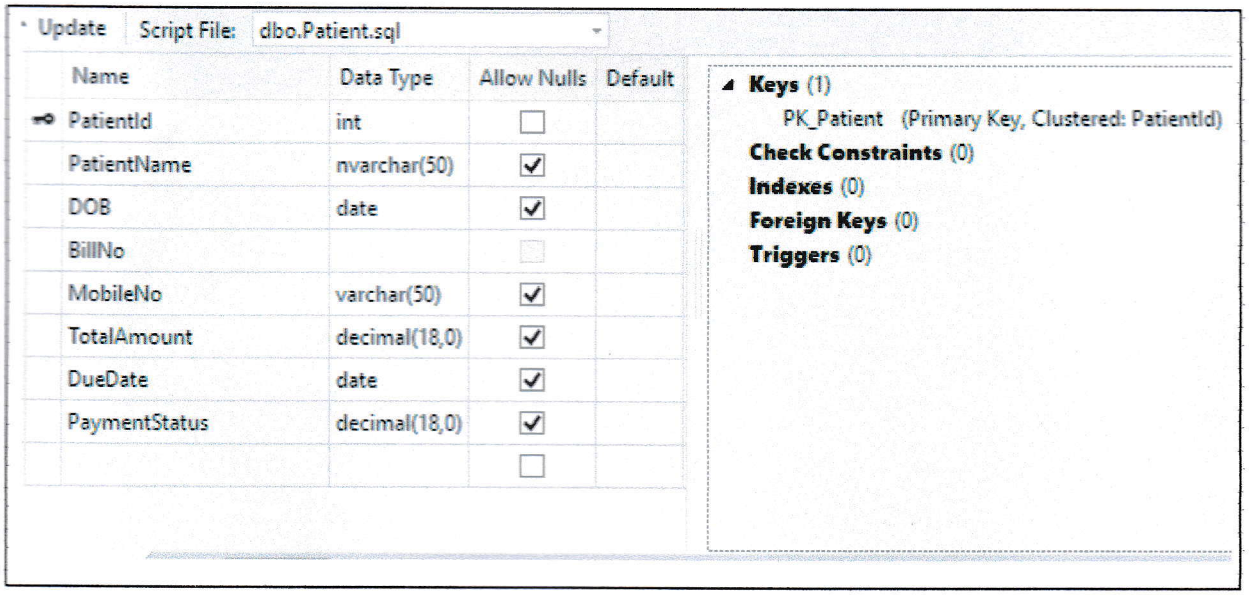


Figure 6.1.2: Layered design with a database class

6.2 Database Table Structure

Table structure of Patient details given below:



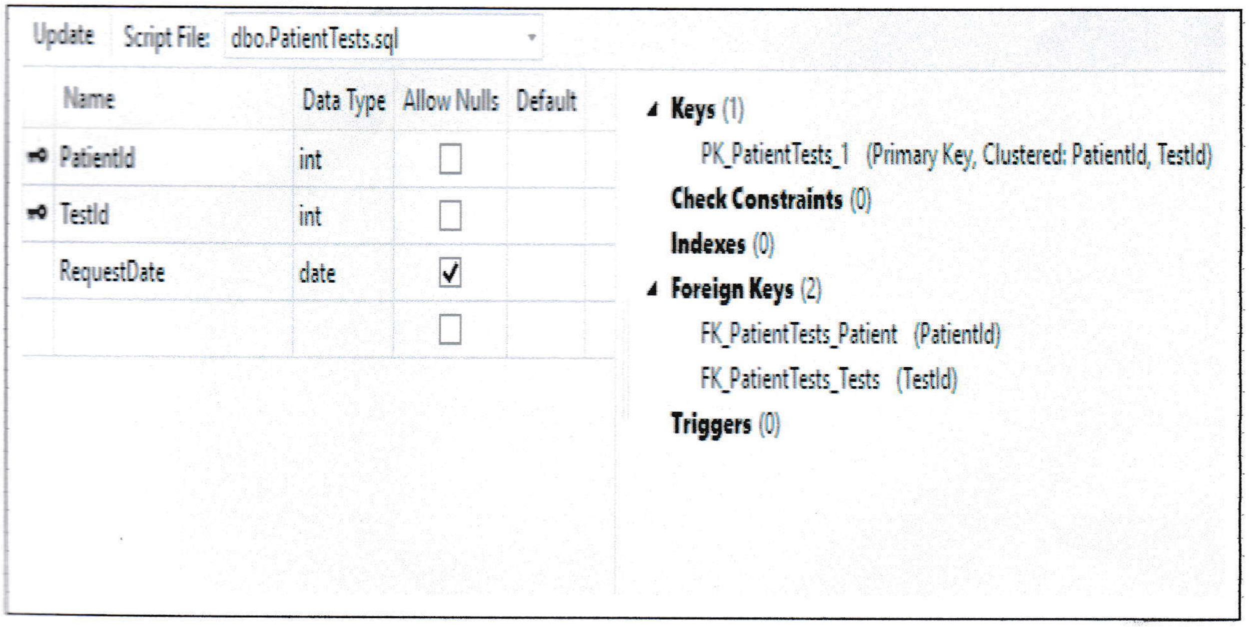
Update   Script File: dbo.Patient.sql

	Name	Data Type	Allow Nulls	Default
🔑	PatientId	int	□	
	PatientName	nvarchar(50)	☑	
	DOB	date	☑	
	BillNo		□	
	MobileNo	varchar(50)	☑	
	TotalAmount	decimal(18,0)	☑	
	DueDate	date	☑	
	PaymentStatus	decimal(18,0)	☑	
			□	

- **Keys** (1)
  - PK_Patient (Primary Key, Clustered: PatientId)
- **Check Constraints** (0)
- **Indexes** (0)
- **Foreign Keys** (0)
- **Triggers** (0)

Figure 6.2.1: Database Patient Table Structure Part

Table structure of Patient Test details given below:



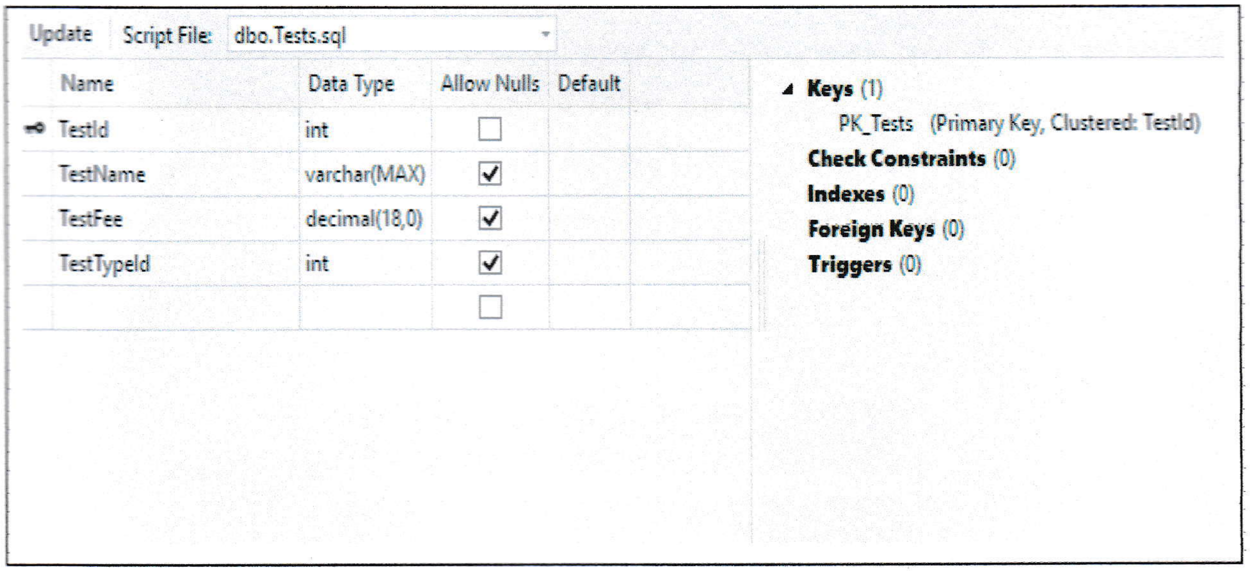
Update   Script File: dbo.PatientTests.sql

	Name	Data Type	Allow Nulls	Default
🔑	PatientId	int	□	
🔑	TestId	int	□	
	RequestDate	date	☑	
			□	

- **Keys** (1)
  - PK_PatientTests_1 (Primary Key, Clustered: PatientId, TestId)
- **Check Constraints** (0)
- **Indexes** (0)
- **Foreign Keys** (2)
  - FK_PatientTests_Patient (PatientId)
  - FK_PatientTests_Tests (TestId)
- **Triggers** (0)

Figure 6.2.2: Database Test Table Structure Part

Table structure of Test details given below:



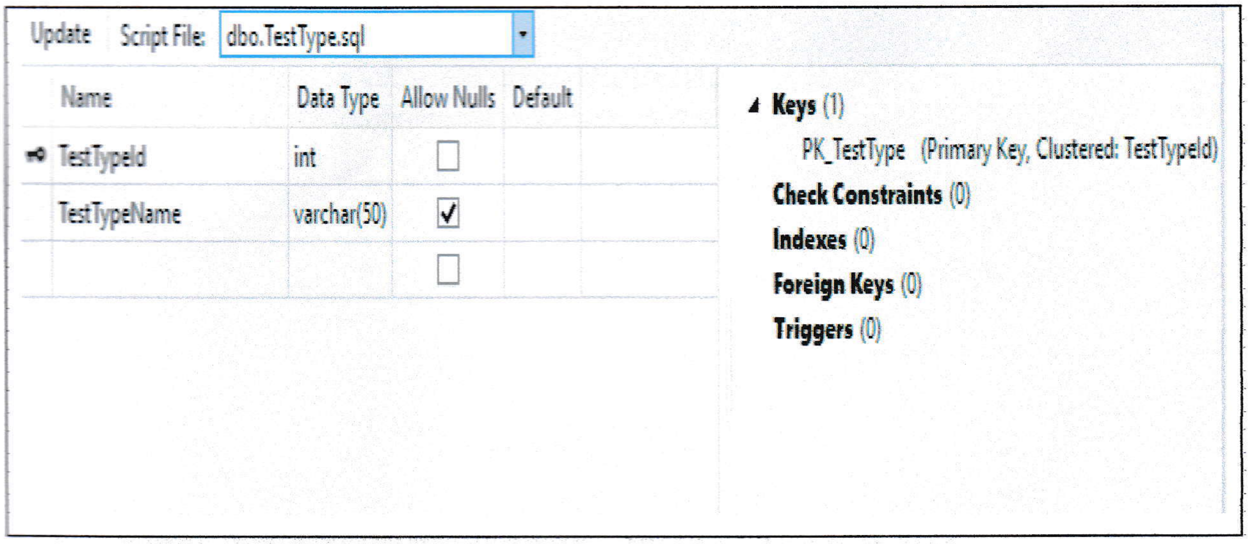
Update Script File: dbo.Tests.sql

	Name	Data Type	Allow Nulls	Default
🔑	TestId	int	□	
	TestName	varchar(MAX)	☑	
	TestFee	decimal(18,0)	☑	
	TestTypeId	int	☑	
			□	

- **Keys** (1)
  - PK_Tests (Primary Key, Clustered: TestId)
- **Check Constraints** (0)
- **Indexes** (0)
- **Foreign Keys** (0)
- **Triggers** (0)

Figure 6.2.3: Database Test Table Structure Part

Table structure of Test Type details given below:



Update Script File: dbo.TestType.sql

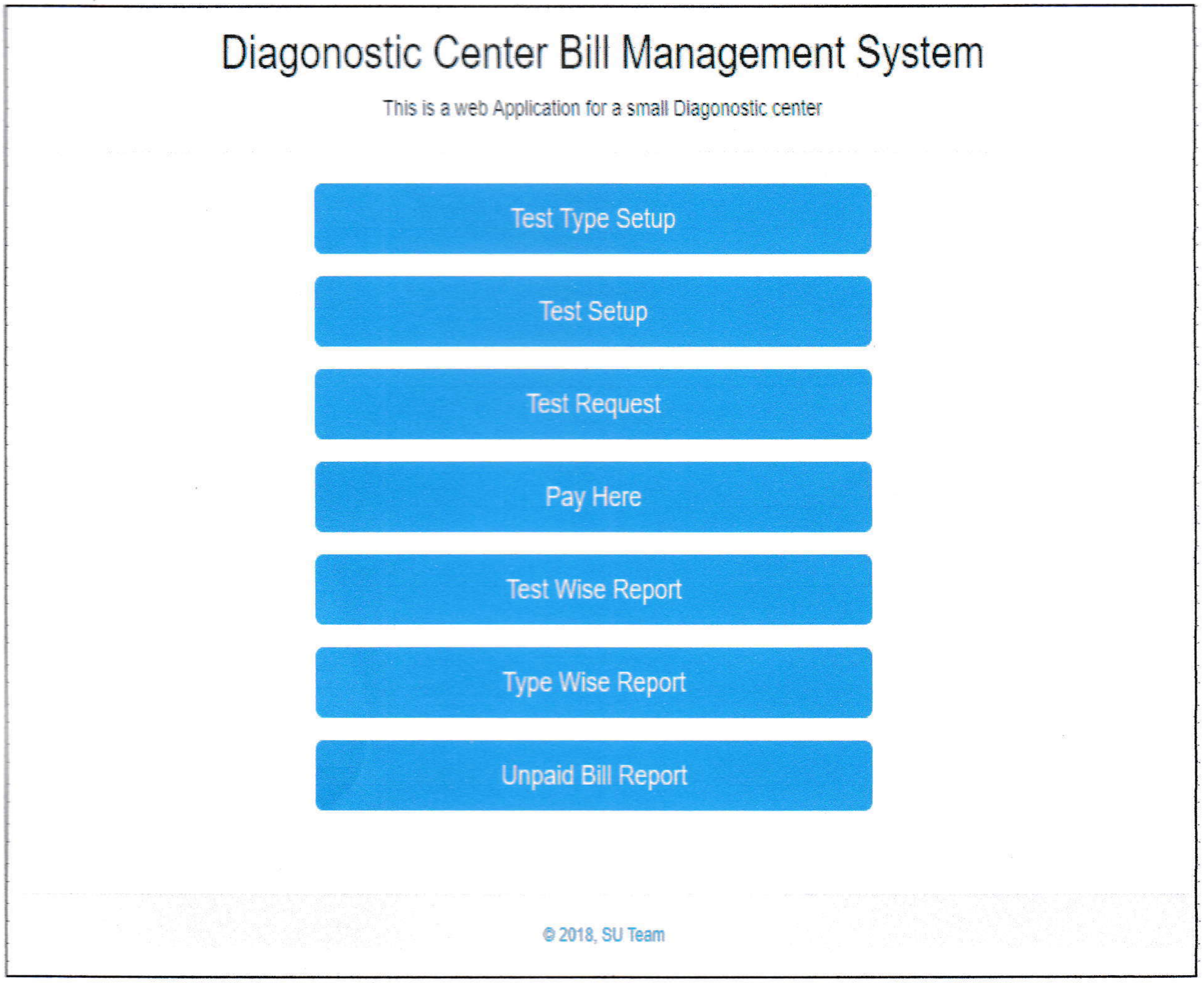
	Name	Data Type	Allow Nulls	Default
🔑	TestTypeId	int	□	
	TestTypeName	varchar(50)	☑	
			□	

- **Keys** (1)
  - PK_TestType (Primary Key, Clustered: TestTypeId)
- **Check Constraints** (0)
- **Indexes** (0)
- **Foreign Keys** (0)
- **Triggers** (0)

Figure 6.2.4: Database Test Type Table Structure Part

6.3 Screenshots

Home page: Home page for Diagnostic Center Bill Management System. It is the starting page of the application. In here admin/ staff can add test type, test setup, new patient test request. Also pay bill, see the billing report detail.



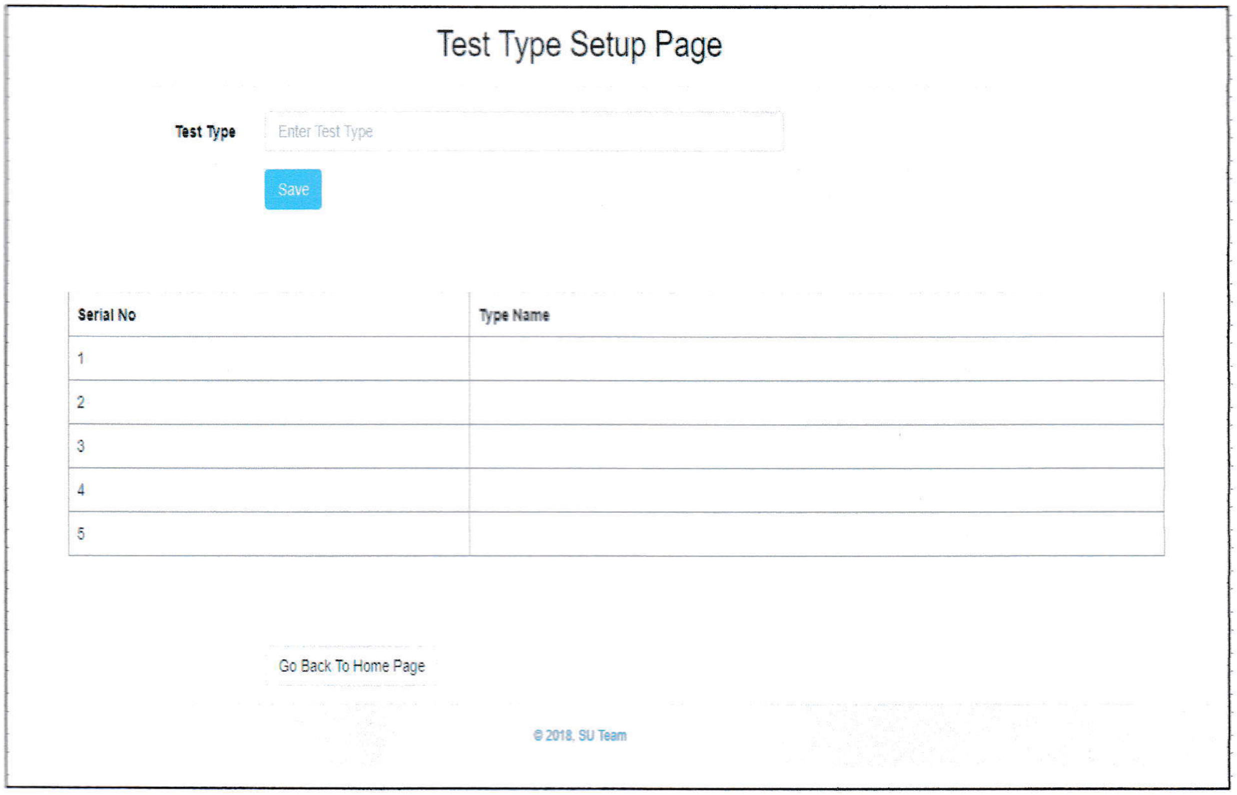
Diagonostic Center Bill Management System

This is a web Application for a small Diagonostic center

© 2018, SU Team

Figure 6.3.1: Home Page

Test Type Setup page: User will enter some basic tests (like blood, urine, ECG, X-ray) using UI. Type name will be unique. Type list will be ordered by type name.



Test Type Setup Page

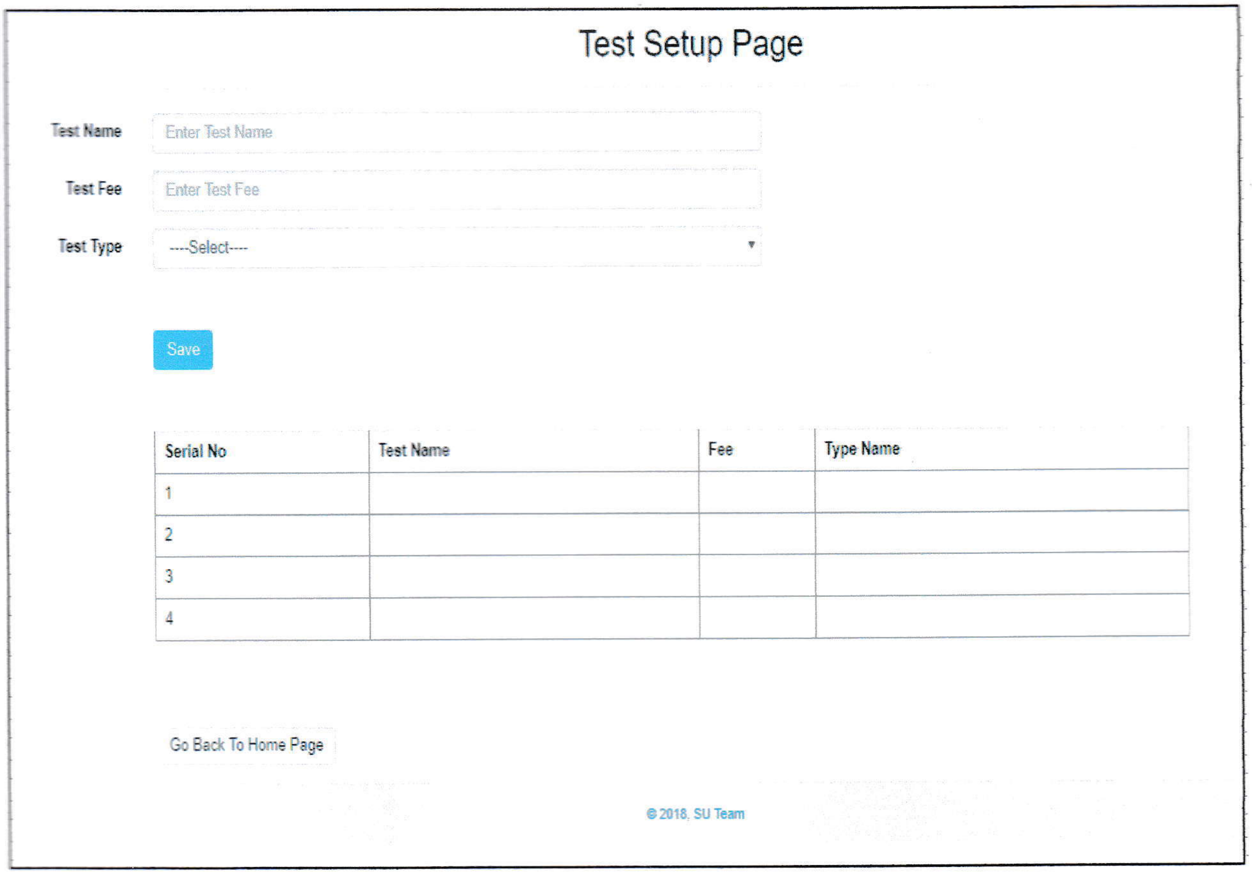
Test Type: Enter Test Type

Serial No	Type Name
1	
2	
3	
4	
5	

© 2018, SU Team

Figure 6.3.2: Test Type Page

Test Setup page: User will enter some basic test information (like whole abdomen, Urine C/S-200, spine X-Ray etc.) with test fee under the test type using this UI. Test name will be unique. Test list be ordered by type name.



Test Setup Page

Test Name: Enter Test Name

Test Fee: Enter Test Fee

Test Type: ----Select----

Serial No	Test Name	Fee	Type Name
1			
2			
3			
4			

Figure 6.3.3: Test Setup Page

Test Request Setup page: When a patient comes for some tests, user of the diagnostic will enter all test information and patient details. When 'Add' button will be clicked, entered information will be displayed in grid. Finally when user will click 'Save' button all information will be saved in the database and a PDF will be opened which contain unique bill number, billing date and patient details.

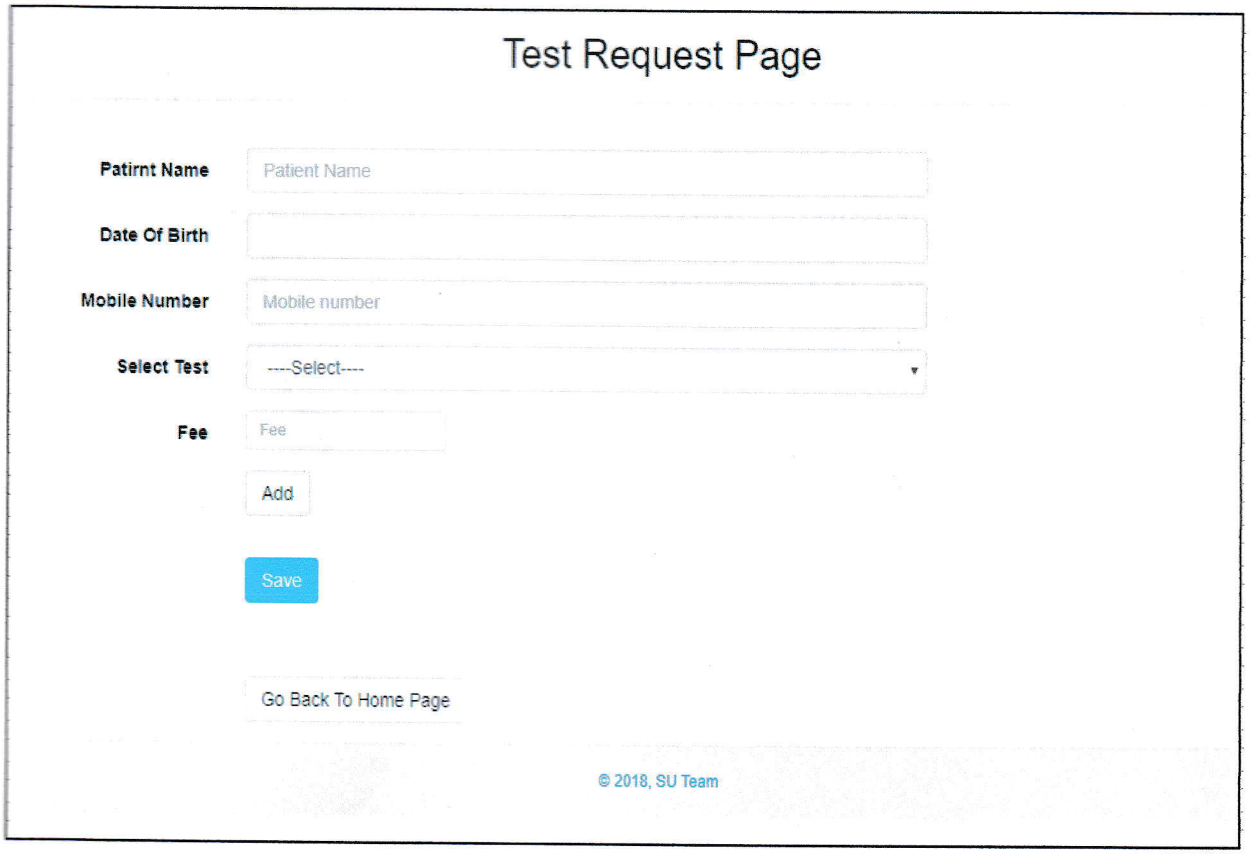


Figure 6.3.4: Test Request Page

Payment page: User of this UI will use payment option when patient comes to pay the bill. User will enter the bill no or phone no and click the search button. Total amount and due amount data against the bill no will be displayed. After that user will take amount from patient and click pay button to update bill payment status.

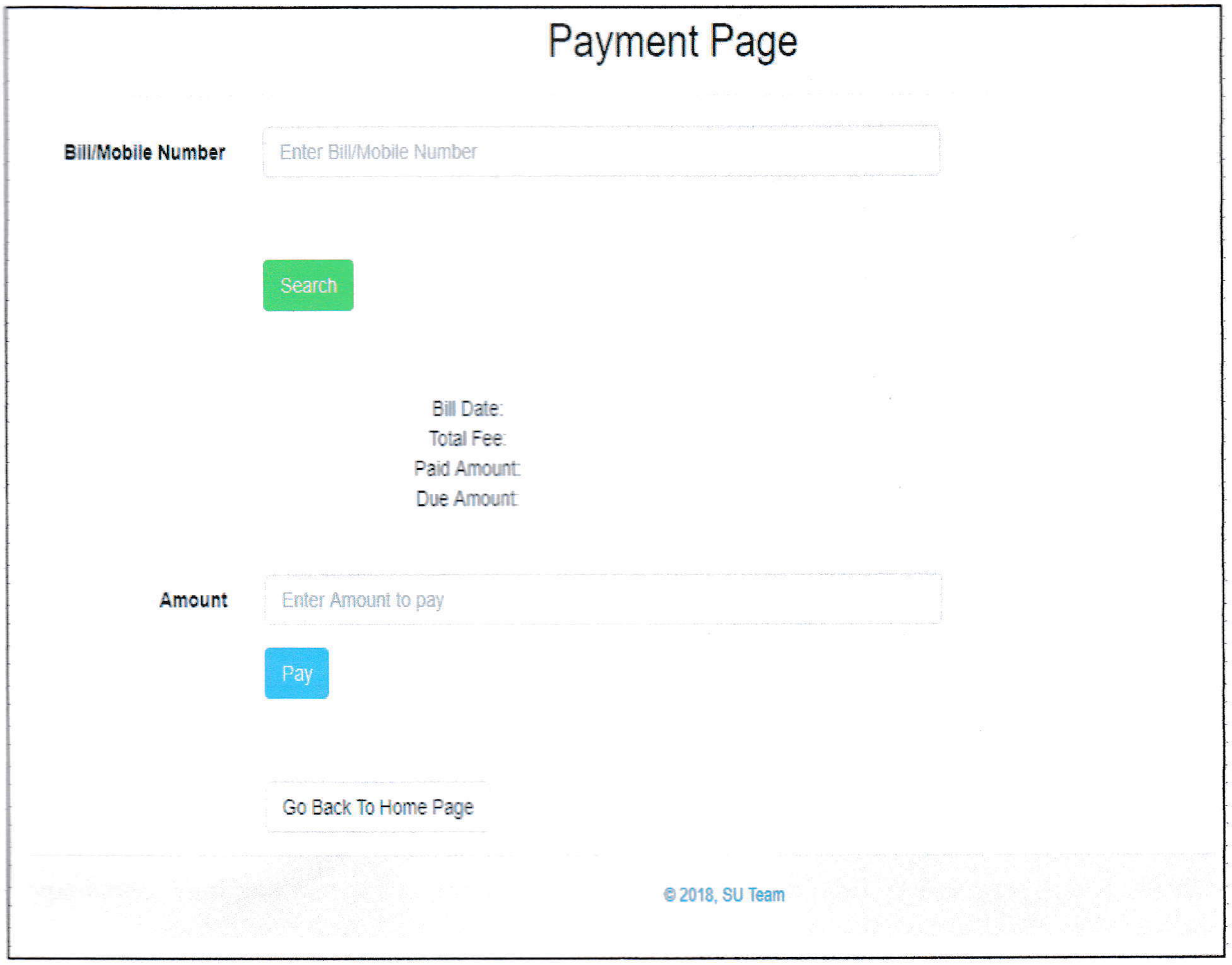


Figure 6.3.5: Payment Page

Test Wise Report page: User will select from and to date and click show button, test name, total number of test and total amount will be displayed. If any test has no test request for selected date between, those will also be displayed with their number of test and total amount will be zero.

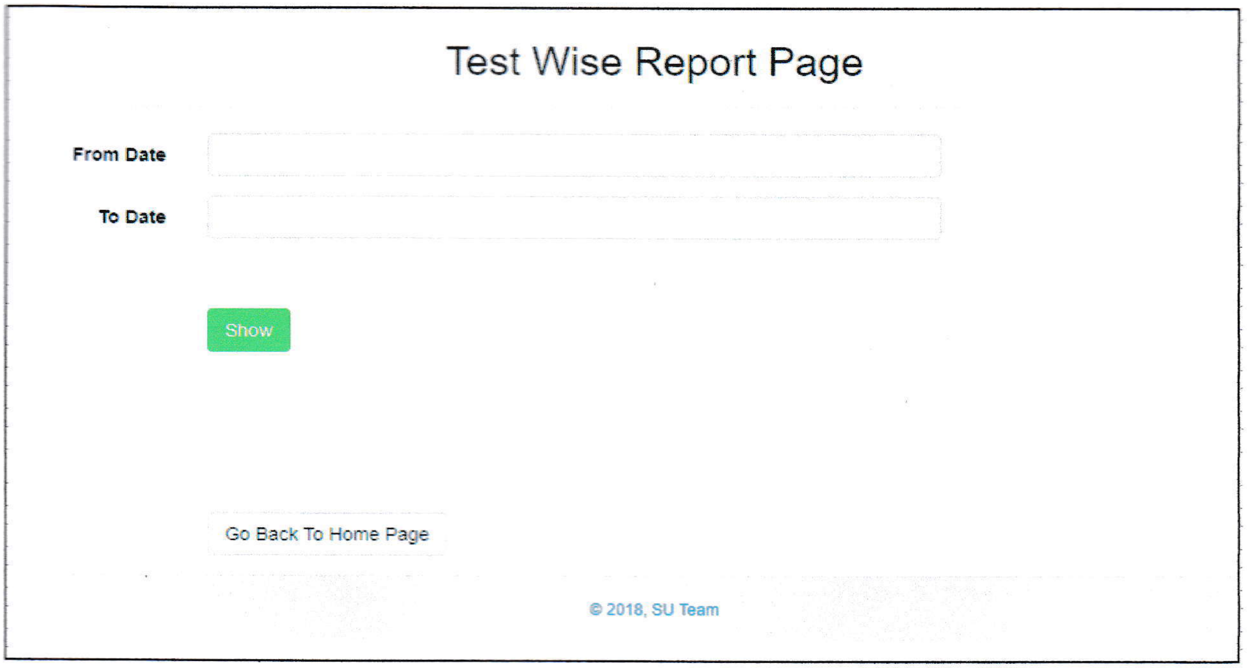


Figure 6.3.6: Test Wise Report page

Test Type Wise Report page: User will select from and to date and click show button, test type name, total number of test and total amount will be displayed. If any type of test has no test request for selected date between, those will also be displayed with their number of test and total amount will be zero.

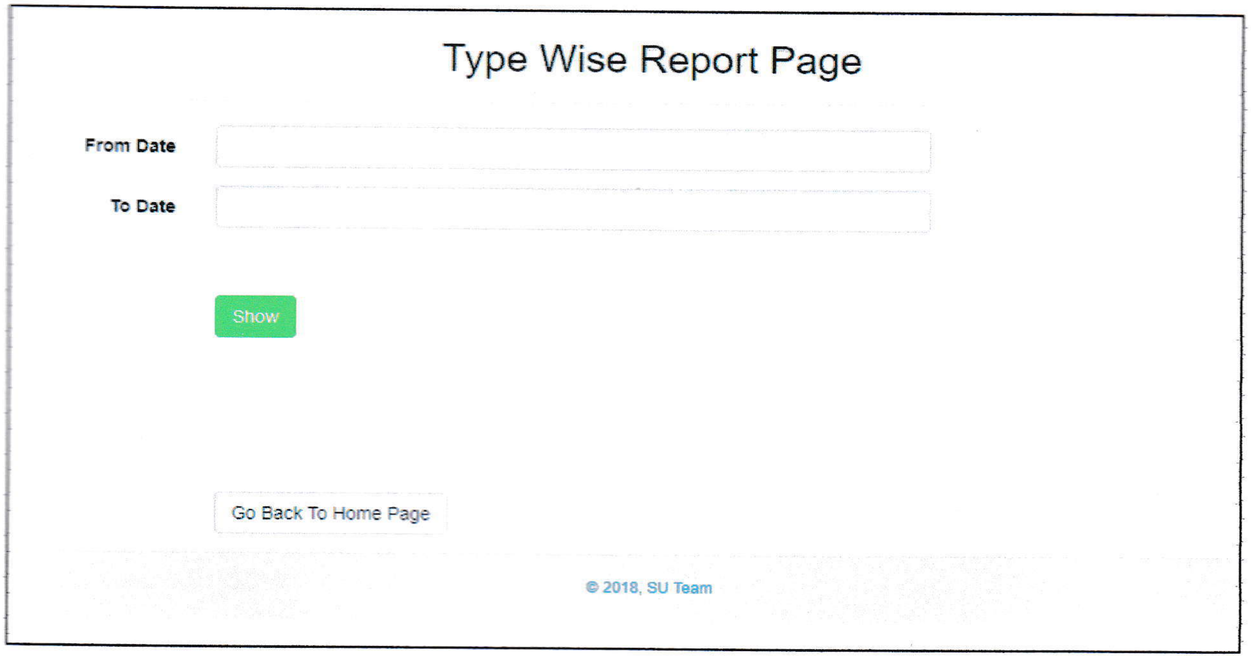


Figure 6.3.7: Type Wise Report Page

Unpaid Bill Report page: An unpaid bill information will be displayed between two selected date.

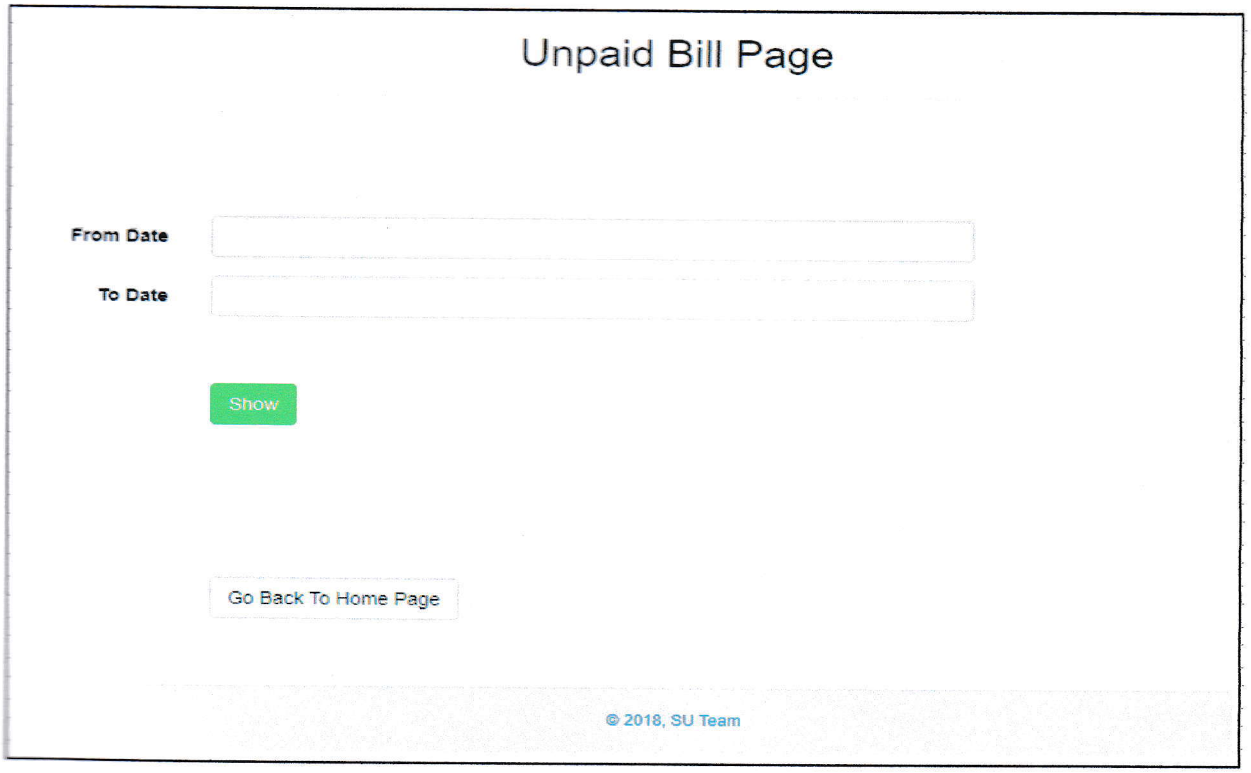
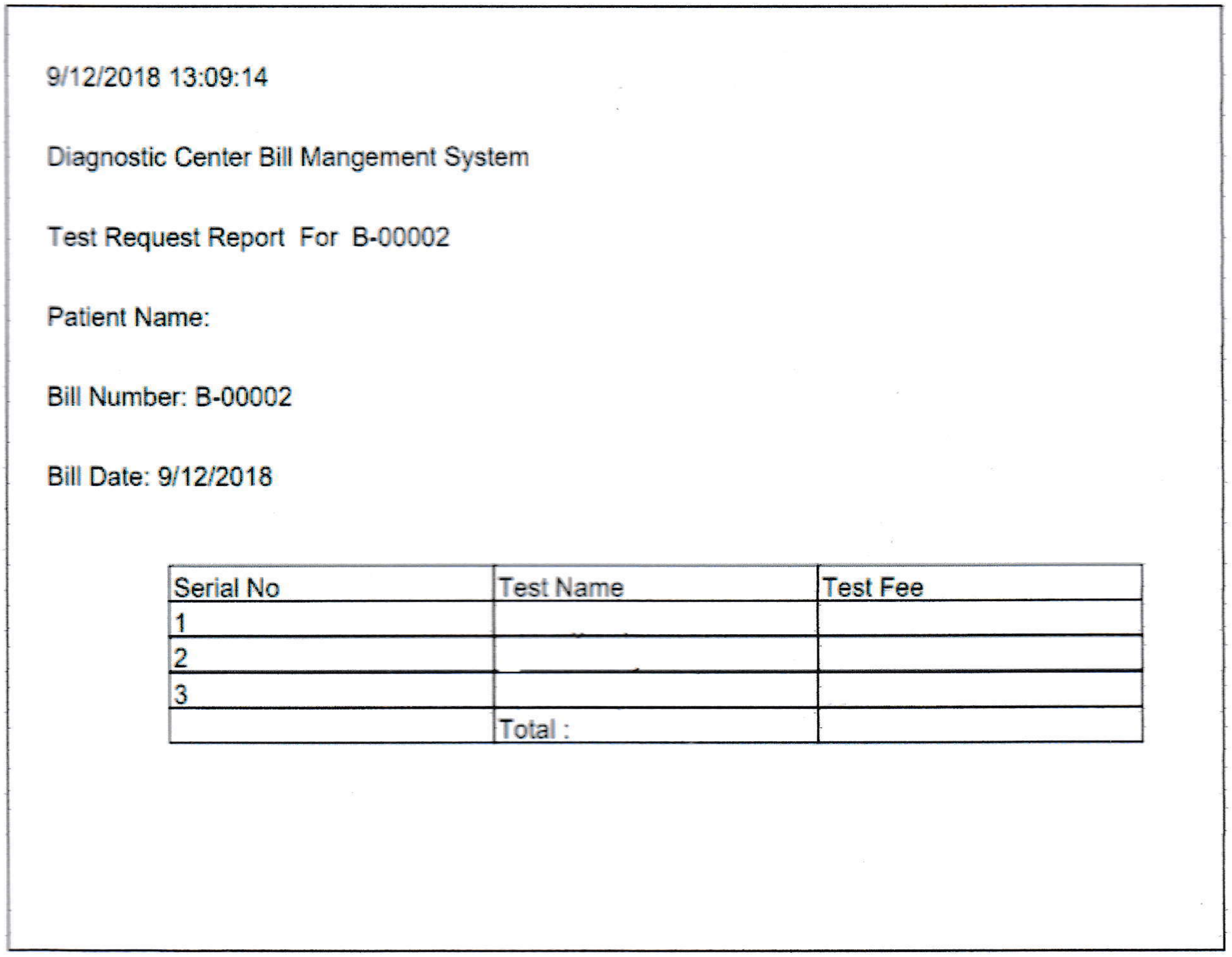


Figure 6.3.8: Unpaid Bill Page

Test Request Report Generation page: If UI user submit test request then system automatically generate a PDF test request report form.



9/12/2018 13:09:14

Diagnostic Center Bill Mangement System

Test Request Report For B-00002

Patient Name:

Bill Number: B-00002

Bill Date: 9/12/2018

Serial No	Test Name	Test Fee
1		
2		
3		
	Total :	

6.3.9: Test Request Report Generation page

Chapter 7: System Quality Assurance

7.1 System Quality Management

A quality management software system that is automated and connects all departments is essential for a regulated. A total quality management system can connect each phase in a products development lifecycle with every department in a company. This gives everyone an opportunity to provide feedback. By building quality into products as opposed to forcing. Quality Assurance to bear the burden of the responsibility, engineering, manufacturing, sales and marketing. The quality of software is assessed by a number of variables. These variables can be divided into external and internal quality criteria. External quality is what a user experiences when running the software in its operational mode. Internal quality refers to aspects that are code- dependent. External quality is critical to the user, while internal quality is meaningful to the developer only [6].

There are mainly two types of quality, they are given below:

1. External Quality
2. Internal Quality

External Quality:

- Features
- Speed
- Network usage
- Stability
- Robustness
- Easy-of-use
- Determinism
- Security
- Power consumption

Internal Quality:

- Testability
- Maintainability
- Conciseness
- Portability

- Documentation
- Scalability
- Thread-safeness

7.2 Quality Assurance

Quality Assurance is crucial for a successful software development effort. Quality Assurance encompasses all stage of the software life cycle and ensures that the processes are structured and carried out in a way that will secure proper quality level of the software products that are being developed or customized. The essence of Quality is to satisfy customer's needs [6].

7.3 White Box Testing

Focuses specifically on using internal knowledge of the software to guide the selection of test data. White box testing involves the testing by looking at the internal structure of the code. White box testing is the detailed investigation of internal logic and structure of the code. In white-box testing an internal perspective of the system, as well as programming skills, are used to design test cases [4]

7.4 Black Box Testing

Black box is testing that ignores the internal mechanism of a system or component and focuses solely on the outputs generated in response to selected inputs and execution conditions. When black box testing is applied to software engineering, the tester would only know the "legal" inputs and what the expected outputs should be, but not how the program actually arrives at those outputs [4].

Chapter 8: Conclusion

8.1 The Experience

Working in this project was a big opportunity for us. We have learned a lot about the technical stuff of software development. The following will indicate our learning from this project.

1. The designing and development strategy of a web application.
2. Database Management.
3. The analysing strategy of software.
4. Hands-on experience about system security management.

8.2 Future Work

DCBMS is an intranet based private web application. This web application will manage billing activities for a small diagnostic center. Now we are doing very initial version of the big software. In future we want to develop hole diagnostic center bill and diagnosis report management system. As our limited time of project we develop the core features of our system but in future it can be possible to add more features.

8.3 Conclusion

The DCBMS is the complete project which keeps the day by day billing data record. It gives details about various tests for patients and purchase section. A small diagnostic center owner starts business with a small capital, they did not interested to use billing application because high cost of application implementation and maintenance. As it is a small web based private application, its cost not very high. So the small diagnostic center can easily use this application to manage their billing system with low cost.

8.4 Acronyms

CSE – Computer Science and Engineering

CSS – Cascading Style Sheets

ERM – Entity Relationship Model

GB – Giga Byte

HTML – Hypertext Markup Language.

DCBMS – Diagnostic Center Bill Management System

LED – Light Emitting Diode

QA – Quality Assurance

RAM –Random Access Memory

RMMM – Risk Mitigation, Monitoring & Management

SU – Sonargaon University

SQL – Structured Query Language

UI – User Interface

UX – User Experience

WWW – World Wide Web

8.5 References

[1] S. Pressman, Roger, Software Engineering a Practitioner's Approach, 6th edition, 2008

[2] Pressman, Roger S. Software Engineering: A Practitioner's Approach. 5thed. Boston: McGraw Hill, 2004.

[3] Kendall, E. Kendall. System Analysis and Design. 4thed. New Delhi: Prentice Hall. 1999.

[4] Hank Meyne, Developing Web Applications with ASP.NET and C#. 1st Edition

[5] By Erik Reitan, Getting Started with ASP.NET 4.5 Web Forms and Visual Studio 2013

[6] Apress Beginning. ASP.Net.4.5 in Csharp Aug-2012

[7] [online] http://www.onlineclassnotes.com/2013/01/what-is-evolutionary- software-process.html

[8] [online] https://sites.google.com/site/softeng2012/

[9] [online] http://www.slideshare.net/koolkampus/quality-management-in-software-engineering-se24?qid=53be2f26-7a75-43ea-87c7- cf2bf898327b&v=&b=&from_search=2

[10] [online] http://www.slideshare.net/EmertxeSlides/introduction-to-uml-37999777

[11] [online] https://www.projectsmart.co.uk/12-tips-for-accurate-project- estimating.php

[12] [online] http://www.informit.com/articles/article.aspx?p=19807&seqNum=6

[13] [online] http://www.functionpointmodeler.com/fpm-infocenter/index.jsp?topic=%2Fcom.functionpointmodeler.fpm.help%2Fditafiles%2Fconcepts%2Fcon-60.html

[14] [online] http://symphony.okdhs.org/symphony/OKDHS/guidances/guidelines/external_inquiry_EA51287C.html

[15] [online] http://alvinalexander.com/FunctionPoints/node12.shtml

[16] [online] http://www.inf.ed.ac.uk/teaching/courses/sapm/2011- 2012/slides/estimation.pdf

[17] [online] http://go.microsoft.com/fwlink/?LinkId=169433

[18] [online] https://stackoverflow.com/

Appendix

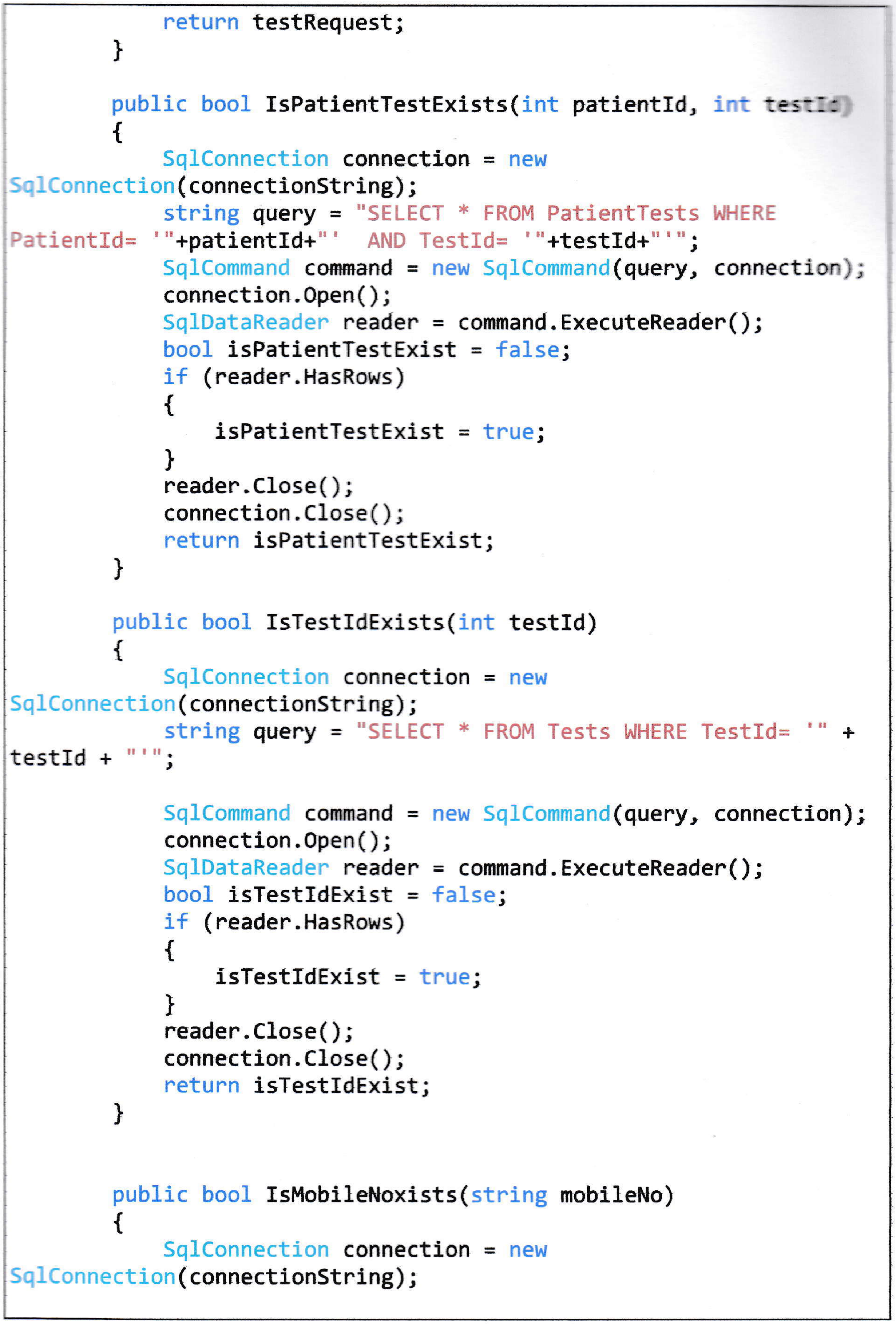
Home page code

```
<%@ Page Language="C#" AutoEventWireup="true"
CodeBehind="Index.aspx.cs"
Inherits="DiagnosticCenterApplication.Index" %>

<!DOCTYPE html>

<html xmlns="http://www.w3.org/1999/xhtml">
<head runat="server">
    <meta charset="UTF-8" />
    <meta name="viewport" content="width=device-width, initial-
scale=1" />
    <meta name="viewport" content="width=device-width, initial-
scale=1, maximum-scale=1, user-scalable=no" />

    <title>Diagonostic Center Bill Management System</title>
    <link href="../Content/bootstrap.css" rel="stylesheet" />
    <link href="../Content/bootstrap-datepicker.css"
rel="stylesheet" />
    <link href="../Content/jquery-ui.css" rel="stylesheet" />
</head>
<body>
    <div class="page-header" style="text-align: center">
        <h2>Diagonostic Center Bill Management System</h2>
        <p>This is a web Application for a small Diagonostic
center</p>
    </div>
    <div class="container-fluid">
        <div class="row">
            <div class="ccol-md-6 col-md-offset-4">
                <form id="form1" runat="server" class="form-
horizontal" style="align-content: center">
                    <div align="center" class="form-group">
                        <div class="col-sm-6" align="center">
                            <asp:Button ID="testTypeSetupPageButton"
runat="server" Text="Test Type Setup" CssClass="btn btn-primary btn-
lg btn-block" OnClick="testTypeSetupPageButton_Click" />
                        </div>
                    </div>
                    <div align="center" class="form-group">
                        <div align="center" class="col-sm-6">
                            <asp:Button ID="testSetupPageButton"
runat="server" Text="Test Setup" CssClass="btn btn-primary btn-lg
btn-block" OnClick="testSetupPageButton_Click" />
                        </div>
                    </div>
                    <div align="center" class="form-group">
                        <div align="center" class="col-sm-6">
                            <asp:Button
ID="testRequestEntryPageButton" runat="server" Text="Test Request"
CssClass="btn btn-primary btn-lg btn-block"
OnClick="testRequestEntryPageButton_Click" />
                        </div>
                    </div>
                    <div align="center" class="form-group">
                        <div align="center" class="col-sm-6">
                            <asp:Button ID="paymentPageButton"
runat="server" Text="Pay Here" CssClass="btn btn-primary btn-lg btn-
block" OnClick="paymentPageButton_Click" />
                        </div>
                    </div>
                    <div align="center" class="form-group">
                        <div align="center" class="col-sm-6">
                            <asp:Button
ID="testWiseReportPageButton" runat="server" Text="Test Wise Report"
CssClass="btn btn-primary btn-lg btn-block"
OnClick="testWiseReportPageButton_Click" />
                        </div>
                    </div>
                    <div align="center" class="form-group">
                        <div align="center" class="col-sm-6">
                            <asp:Button
ID="typeWiseReportPageButton" runat="server" Text="Type Wise Report"
CssClass="btn btn-primary btn-lg btn-block"
OnClick="typeWiseReportPageButton_Click" />
                        </div>
                    </div>
                    <div align="center" class="form-group">
                        <div align="center" class="col-sm-6">
                            <asp:Button ID="unpaidBillPageButton"
runat="server" Text="Unpaid Bill Report" CssClass="btn btn-primary
btn-lg btn-block" OnClick="unpaidBillPageButton_Click" />
                        </div>
                    </div>

                </form>
            </div>
        </div>
    </div>
    <br/>
    <br/>

    <footer class="panel-footer">
        <div style="text-align: center">
```

```
            <h6 class="text-info" style="text-align: center;">&copy;
2018, SU Team</h6>
        </div>
    </footer>

</body>
</html>
```

Data access layer code for Test request gateway

```
using System;
using System.Collections.Generic;
using System.Data.SqlClient;
using System.Linq;
using System.Web;
using System.Web.Configuration;
using DiagnosticCenterApplication.Models;
using DiagnosticCenterApplication.Models.ViewModels;

namespace DiagnosticCenterApplication.DAL
{
    public class TestRequestGateway
    {
        string connectionString =

WebConfigurationManager.ConnectionStrings["DiagnosticCenterConnectio
nString"].ConnectionString;

        public int SaveTestRequest(TestRequestEntry testRequest)
        {
            SqlConnection connection = new
SqlConnection(connectionString);
            string query = "INSERT INTO
Patient(PatientName,DOB,MobileNo,TotalAmount,DueDate,PaymentStatus)
VALUES ('" + testRequest.PatientName + "','" + testRequest.Dob +
"','" + testRequest.MobileNo + "','" + testRequest.TotalAmount +
"','" + testRequest.DueDate + "', '"+testRequest.PaymentStatus+"')";

            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();

            int rowAffacted = command.ExecuteNonQuery();
            connection.Close();
            return rowAffacted;
        }
        public int SavepatientTests(int patientId, int testId,
string requestDate)
        {
            SqlConnection connection = new
SqlConnection(connectionString);

            string query = "INSERT INTO
PatientTests(PatientId,TestId,RequestDate) VALUES('" + patientId +
"', '" + testId + "', '" + requestDate + "')";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            int rowAffected = command.ExecuteNonQuery();
            connection.Close();
            return rowAffected;
        }

        public TestRequestEntry GetPatientByMobileNo(string
mobileNo)
        {
            SqlConnection connection = new
SqlConnection(connectionString);

            string query = "SELECT * FROM Patient WHERE MobileNo =
'" + mobileNo + "'";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();

            TestRequestEntry testRequest = null;

            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    testRequest = new TestRequestEntry();

                    testRequest.PatientName =
reader["PatientName"].ToString();
                    testRequest.MobileNo =
reader["MobileNo"].ToString();
                    testRequest.BillNo =
reader["BillNo"].ToString();
                    testRequest.PatientId =
Convert.ToInt32(reader["PatientId"].ToString());

                }
                reader.Close();
            }
            connection.Close();
```



```
            return testRequest;
        }

        public bool IsPatientTestExists(int patientId, int testId)
        {
            SqlConnection connection = new
SqlConnection(connectionString);
            string query = "SELECT * FROM PatientTests WHERE
PatientId= '"+patientId+"'  AND TestId= '"+testId+"'";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            bool isPatientTestExist = false;
            if (reader.HasRows)
            {
                isPatientTestExist = true;
            }
            reader.Close();
            connection.Close();
            return isPatientTestExist;
        }

        public bool IsTestIdExists(int testId)
        {
            SqlConnection connection = new
SqlConnection(connectionString);
            string query = "SELECT * FROM Tests WHERE TestId= '" +
testId + "'";

            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            bool isTestIdExist = false;
            if (reader.HasRows)
            {
                isTestIdExist = true;
            }
            reader.Close();
            connection.Close();
            return isTestIdExist;
        }


        public bool IsMobileNoxists(string mobileNo)
        {
            SqlConnection connection = new
SqlConnection(connectionString);
```

```
            string query = "SELECT * FROM Patient Where MobileNo =
'" + mobileNo + "'";

            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            bool isMobilenoExists = false;
            if (reader.HasRows)
            {
                isMobilenoExists = true;
            }
            reader.Close();
            connection.Close();
            return isMobilenoExists;
        }

        public TestRequestEntry SearchByBillorMobileNo(string
billNo, string mobileNo)
        {
            SqlConnection connection = new
SqlConnection(connectionString);

            string query = "SELECT * FROM Patient WHERE BillNo = '"
+ billNo + "' OR MobileNo = '" + mobileNo + "'";

            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            SqlDataReader reader = command.ExecuteReader();
            TestRequestEntry testRequest = null;
            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    testRequest = new TestRequestEntry();

                    testRequest.PatientId =
Convert.ToInt32(reader["PatientId"].ToString());
                    testRequest.PatientName =
reader["PatientName"].ToString();
                    testRequest.MobileNo =
reader["MobileNo"].ToString();
                    testRequest.TotalAmount =
Convert.ToDecimal(reader["TotalAmount"].ToString());
                    testRequest.DueDate =
Convert.ToDateTime(reader["DueDate"].ToString());
                    testRequest.PaymentStatus =
Convert.ToDecimal(reader["PaymentStatus"].ToString());

                }
                reader.Close();
            }
            connection.Close();
            return testRequest;
        }

        public List<SearchView> GetAllBillInfo(string billNo, string
mobileNo)
        {
            SqlConnection connection = new
SqlConnection(connectionString);
            string query = "SELECT * FROM SearchView WHERE BillNo =
'"+billNo+"' OR MobileNo = '"+mobileNo+"'";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();

            SqlDataReader reader = command.ExecuteReader();
            List<SearchView> searches = new List<SearchView>();

            if (reader.HasRows)
            {
                while (reader.Read())
                {
                    SearchView search = new SearchView();
                    search.PatientId =
Convert.ToInt32(reader["PatientId"].ToString());
                    search.BillNo = reader["BillNo"].ToString();
                    search.MobileNo = reader["MobileNo"].ToString();
                    search.TestName = reader["TestName"].ToString();
                    search.TestFee
=Convert.ToDecimal(reader["TestFee"].ToString());
                    search.Amount =
Convert.ToDecimal(reader["PaymentStatus"].ToString());
                    searches.Add(search);
                }
                reader.Close();
            }
            connection.Close();
            return searches;
        }

        public int UpdatePaymentStatus(string billNo, string
mobileNo, decimal amount)
        {
            SqlConnection connection = new
SqlConnection(connectionString);
```

```
            string query = "UPDATE Patient SET 
PaymentStatus='"+amount+"' WHERE BillNo = '" + billNo + "' OR 
MobileNo = '" + mobileNo + "'";
            SqlCommand command = new SqlCommand(query, connection);
            connection.Open();
            int rowAffected = command.ExecuteNonQuery();
            connection.Close();
            return rowAffected;
        }


    }
}
```

End