**HOSPITAL RECORD MANAGEMENT SYSTEM PROJECT DOCUMENTATION**

**Name : Alfred Koomson**

**Course: Software Engineering 1-3**

* **Introduction**
* **Problem Definition**
* **Objectives**
* **Analysis**
* **Feasibility Study**
* **Design**
* **Requirement Analysis**
* **Limitations**
* **Conclusion**

**INTRODUCTION**

**Introduction:**

Human Body is a very complex and sophisticated structure and comprises of millions of functions. All these complicated functions have been understood by man him, part-by part their research and experiments. As science and technology progressed, medicine became an integral part of the research. Gradually, medical science became an entirely new branch of science. As of today, the Health Sector comprises of Medical institutions i.e. Hospitals, HOSPITALs etc. research and development institutions and medical colleges. Thus the Health sector aims at providing the best medical facilities to the common man.

**Medical Institutions in Ghana**

Still being a developing nation Ghana has seen a tremendous growth of the Health sector in the field of research as well as in the field of development of numerous large and small scale Hospital institutions still lacking in inter-structure facilities. Government of Ghana has still aimed at providing medical facilities by establishing hospital. The basic working of various hospitals in Ghana is still on paper as compared to hospitals in European countries where computers have been put in to assist the hospital personnel’s work. The concept of automation of the administration and management of hospital is now being implemented in Ghana also, with large hospitals like Korle Bu Teaching Hospital and Ridge Hospital in Accra.

Our project is based on the above concept i.e. automation of Administration and Records Management of Hospital. The project has been developed keeping in-view the following aspects: -

1. Working environment of the Hospital.
2. The thought-process and attitude of Ghanaian people.
3. The literacy rate of Ghanaian.
4. The Existing system, being used in the majority of Hospitals.
5. The availability of Infra-structural facilities likes finance, skilled personals, and working environment.

**PROBLEM DEFINITION**

Since HOSPITAL is associated with the lives of common people and their day-to-day routines so I decided to work on this project.

The manual handling of the record is time consuming and highly prone to error. The purpose of this project is to automate or make online, the process of day-to-day activities like Admission of New Patient, Discharge of Patient, Assign a Doctor, and finally compute the bill etc.

I have tried my best to make the complicated process **Hospital Record Management System** as simple as possible using Structured & Modular technique & Menu oriented interface. I have tried to design the software in such a way that user may not have any difficulty in using this package & further expansion is possible without much effort. Even though I cannot claim that this work to be entirely exhaustive, the main purpose of my exercise is perform each Hospital’s activity in computerized way rather than manually which is time consuming.

I am confident that this software package can be readily used by non-programming personal avoiding human handled chance of error.

**AIMS & OBJECTIVES**

**Objective:**

Hospital are the essential part of our lives, providing best medical facilities to people suffering from various ailments, which may be due to change in climatic conditions, increased work-load, emotional trauma stress etc. It is necessary for the hospitals to keep track of its day-to-day activities & records of its patients, doctors, nurses, ward boys and other staff personals that keep the hospital running smoothly & successfully.

But keeping track of all the activities and their records on paper is very cumbersome and error prone. It also is very inefficient and a time-consuming process Observing the continuous increase in population and number of people visiting the hospital. Recording and maintaining all these records is highly unreliable, inefficient and error-prone. It is also not economically & technically feasible to maintain these records on paper.

Thus keeping the working of the manual system as the basis of our project. We have developed an automated version of the manual system, named as “ADMINISTRATION DASHBOARD”.

The main aim of our project is to provide a paper-less hospital up to 90%. It also aims at providing low-cost reliable automation of the existing systems. The system also provides excellent security of data at every level of user-system interaction and also provides robust & reliable storage and backup facilities.

**AIM:**

The aim of the study to fully related with Hospital Management system.

* The Software is for the automation of Hospital Management System.
* It maintains two levels of users:-

\_ Administrator Level

\_ User (doctors etc.) Level

* The Software includes:-

i. Preserve records throughout their life cycle.

ii. Simplify the activities, systems, and processes of records maintenance and use.

iii. Save time and reduce costs.

iv. Identify what records exist by records inventory.

v. To provide a user-friendly GUI, that will enable the user to work on clicks and data entry that makes the software easy to use.

vi. Easy accessibility regardless of the location.

vii. Maintaining Patient details.

**Features of the of system;**

The admin has the following capabilities;

a. Login and Log out securely.

b. Add and manage new patients by assigning id etc.

c. Add and manage new doctors.

d. Add new subject and manage subjects.

e. Change user password.

Doctors Capabilities;

a. Read only the details of patients.

b. Access the diagnosis page.

c. Print out the diagnosis summary of patients.

## SYSTEM ANALYSIS

**PRINCIPLES OF SYSTEM ANALYSIS:**

**PRINCIPLES:**

* Understand the problem before you begin to create the analysis model.
* Develop prototypes that enable a user to understand how human machine

Interaction will occur.

* Record the origin of and the reason for every requirement.
* Use multiple views of requirements like building data, function and behavioral models.
* Work to eliminate ambiguity

System Analysis is a separation of a substance into parts for study and their implementation and detailed examination.

Before designing any system it is important that the nature of the business and the way it currently operates are clearly understood. The detailed examination provides the specific data required during designing in order to ensure that all the client's requirements are fulfilled. The investigation or the study conducted during the analysis phase is largely based on the feasibility study. Rather it would not be wrong to say that the analysis and feasibility phases overlap. High-level analysis begins during the feasibility study. Though analysis is represented as one phase of the system development life cycle (SDLC), this is not true. Analysis begins with system initialization and continues until its maintenance. Even after successful implementation of the system, analysis may play its role for periodic maintenance and up gradation of the system. One of the main causes of project failures is inadequate understanding, and one of the main causes of inadequate understanding of the requirements is the poor planning of system analysis.

Analysis requires us to recall the objectives of the project and consider following three questions:

* + What type of information is required?
  + What are the constraints on the investigation?
  + What are the potential problems that may make the task more difficult?

Keeping the above questions in mind and considering the survey conducted to determine the need of the system; the total system was designed and can be described as under:

**The three major parts of the system are:**

* **Providing Information:**

The system is effectively used to provide large variety of information to the interested customer. The major purpose of the site is to easily provide access to records of various Job seekers & users of matrimonial such as resume & profile of boys and girls those who want to search a life partner with quick update to latest modifications in the records. This thing is not at all possible in printed material, which are updated only once a few weeks. It also gives information about the general usage of the system for first time visitors. The system itself works as an information provider for company & life partner seekers.

**Preliminary Investigation**

System development, a process consisting of two major steps of system analysis and design, start when management or sometimes system development personnel feel that a new system or an improvement in the existing system is required. The system development life cycle is classically thought of as the set of activities that analysts, designers and users carry out to develop and implement an information system. The system development life cycle consists of the following activities:

* Preliminary investigation
* Determination of system requirements
* Design of system
* Development of software
* System testing
* Implementation, evaluation, and maintenance

**A request to take assistance from information system can be made for many reasons, but in each case someone in the organization initiates the request is made, the first system activity the preliminary investigation begins. This activity has three parts:**

1. Request clarification
2. Feasibility study
3. Request approval

**Request clarification**: Many requests from employees and users in the organizations are not clearly defined, therefore it becomes necessary that project request must be examined and clarified properly before considering systems investigation.

**FEASIBILITY STUDY:**

The feasibility study proposes one or more conceptual solution to the problem set of the project. In fact, it is an evaluation of whether it is worthwhile to proceed with project or not.

1. Evaluation of feasibility of such solutions. Such evaluation often indicates shortcomings in the initial goals. This step is repeated as the goals are adjusted and the alternative solutions are evaluated.

Feasibility analysis usually considers a number of project alternatives, one that is chosen as the most satisfactory solution. These alternatives also need to be evaluated in a broad way without committing too many resources. Various steps involved in feasibility analysis are:

1. To propose a set of solution that can realize the project goal. These solutions are usually descriptions of what the new system should look like.

Four primary areas of interest in feasibility study are:

**Economic Feasibility**: An evaluation of development cost weighed against the ultimate income of benefit derived from the development system of product. In economic feasibility, cost benefit analysis is done in which expected cost and benefits are evaluated.

**COST AND BENEFIT ANALYSIS:**

Developing an IT application is an investment. Since after developing that application it provides the organization with profits. Profits can be monetary or in the form of an improved working environment. However, it carries risks, because in some cases an estimate can be wrong. And the project might not actually turn out to be beneficial.

Cost benefit analysis helps to give management a picture of the cost, benefits and risks. It usually involves comparing alternate investments.

Cost benefit determines the benefits and savings that are expected from the system and compares them with the expected costs.

In performing cost and benefit analysis it is important to identify cost and benefits factors. Cost and benefits can be categorized into the following categories:

1. **Development Costs –** Development costs is the costs that are incurred during the development of the system. It is one time investment.
2. **Operating Costs –** Operating Costs are the expenses required for the day to day running of the system. Examples of Operating Costs are Wages, Supplies and Overheads.
3. **Hardware/Software Costs –** It includes the cost of purchasing or leasing of computers and its peripherals. Software costs involves required S/W costs.
4. **Personnel Costs –** It is the money spent on the people involved in the development of the system.
5. **Facility Costs –** Expenses that are incurred during the preparation of the physical site where the system will be operational. These can be wiring, flooring, acoustics, lightning, and air-conditioning.
6. **Supply Costs –** These are variable costs that are very proportionately with the amount of use of paper, ribbons, disks, and the like.

 **BENEFITS**

We can define benefits as

**Profit or Benefit = Income – Costs**

Benefits can be accrued by:

* Increasing income, or
* Decreasing costs, or
* Both

**TECHNICAL FEASIBILITY:**

Technical Feasibility includes existing and new H/W and S/W requirements that are required to operate the project using C#.NET. The basic S/W requirement isC#.NET in which the front end of the hospital record management project has been done. The basic entry forms are developed in C# and the data is stored in the MSSQL.

**OPERATIONAL FEASIBILITY:**

Operational feasibility is mainly concerned with issues like whether the system will be used if it is developed and implemented. Whether there will be resistance from users that will affect the possible application benefits? The essential questions that help in testing the technical feasibility of a system are following:

* Does management support the project?
* Are the users not happy with current business practices? Will it reduce the time considerably? If yes, then they will welcome the change and the new system.
* Have the users involved in the planning and development of the project? Early involvement reduced the probability of resistance towards the new system.
* Will the proposed system really benefit the organization? Does the overall response increase? Will accessibility of information be lost? Will the system affect the customers in considerable way?

**Legal Feasibility**:

A determination of any infringement, violation, or liability that could result from development of the system. Legal feasibility tells that the software used in the project should be original purchased from the legal authorities and they have the license to use it or the software are pirated.

**Alternatives**:

An evaluation of alternative approaches to the development of system or product.

**SYSTEM LIFE CYCLE:**

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system.

System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the System Analysis and Design terminology, the system development life cycle means software development life cycle.

Following are the different phases of software development cycle:

* System study
* Feasibility study
* System analysis
* System design
* Coding
* Testing
* Implementation
* Maintenance

The different phase of software development life cycle is shown below.

34

**PRELIMINARY**

**INVESTIGATION**

**DETERMINATION OF**

**REQUIREMENTS**

**REVIEW RUNNING**

**SYSTEM AND SYSTEM**

**MAINTENANCE**

**SYSTEM**

**IMPLEMENTATIONN**

**SYSTEM TESTING**

**SYSTEM TESTING**

**DEVELOPMENT OF**

**SOFTWARE**

**AND CODING**

**DESIGN OF SYSTEM**

**DEVELOPMENT OF**

**PROTOTYPE SYSTEM**

SYSTEM

ANALYSIS

50

%

DESIGN OF

SYSTEM 30%

CODING

%

20

20

%

**FIG: SHOWING GENERAL LIFE CYCLE PROCESS AND PERCENTAGE OF TIME DEVOTED**

**FACT FINDING TECHNIQUES:**

The functioning of the system is to be understood by the system analyst to design the proposed system. Various methods are used for this and these are known as fact-finding techniques. The analyst needs to fully understand the current system.

The analyst needs data about the requirements and demands of the project undertaken and the techniques employed to gather this data are known as fact-finding techniques. Various kinds of techniques and the most popular among them are interviews, questionnaires, record views, case tools and also the personal observations made by the analyst himself.

 **Interviews**

Interview is a very important data gathering technique as in this the analyst directly contacts system and the potential user of the proposed system.

One very essential aspect of conducting the interview is that the interviewer should first establish a rapport with the interviewee. It should also be taken into account that the interviewee may or may not be a technician and the analyst should prefer to use day to day language instead of jargon and technical terms.

The advantage of the interview is that the analyst has a free hand and the he can extract almost all the information from the concerned people but then as it is a very time consuming method, he should also employ other means such as questionnaires, record reviews, etc. This may also help the analyst to verify and validate the information gained. Interviewing should be approached, as logically and from a general point of view the following guides can be very beneficial for a successful interview:

1. Set the stage for the interview.
2. Establish rapport; put the interview at ease.
3. Phrase questions clearly and succinctly.
4. Be a good listener; a void arguments.
5. Evaluate the outcome of the interview.

The interviews are of the two types namely **structured** and **unstructured**.

**I. Structured Interview**

Structured interviews are those where the interviewee is asked a standard set of questions in a particular order. All interviews are asked the same set of questions. The questions are further divided into two kinds of formats for conducting this type if interview.

**II. Unstructured Interview**

The unstructured interviews are undertaken in a question-and-answer format. This is of a much more flexible nature than the structured and can be very rightly used to gather general information about the system.

 **Questionnaires:**

Questionnaires are another way of information gathering where the potential users of the system are given questionnaires to be filled up and returned to the analyst.

Questionnaires are useful when the analyst need to gather information from a large number of people. It is not possible to interview each individual. Also if the time is very short, in that case also questionnaires are useful. If the analyst guarantees the anonymity of the respondent then the respondent answers the questionnaires very honestly and critically.

The analyst should sensibly design and frame questionnaires with clarity of its objective so as to do just to the cost incurred on their development and distribution.

 **Record Reviews**

Records and reports are the collection of information and data accumulated over the time by the users about the system and its operations. This can also put light on the requirements of the system and the modifications it has undergone. Records and reports may have a limitation if they are not up-to-date or if some essential links are missing. All the changes, which the system suffers, may not be recorded. The analyst may scrutinize the records either at the beginning of his study which may give him a fair introduction about the system and will make him familiar with it or in the end which will provide the analyst with a comparison between what exactly is/was desired from the system and its current working.

 **On-Site Observation**

On-site observations are one of the most effectively tools with the analyst where the analyst personally goes to the site and discovers the functioning of the system. As an observer, the analyst can gain first-hand knowledge of the activities, operations, processes of the system on-site, hence here the role of an analyst is of an information seeker. This information is very meaningful as it is unbiased and has been directly taken by the analyst. This exposure also sheds some light on the actual happenings of the system as compared to what has already been documented, thus the analyst gets closer to system.

This technique is also time-consuming and the analyst should not jump to conclusions or draw inferences from small samples of observation rather the analyst should be more patient in gathering the information. This method is however less effective for learning about people’s perceptions, feelings and motivations.

 **Research**

Research is the systematic investigation into and study of a well define project through the use of internet in order to establish fact and draw new conclusions for problem solving.  
This fact finding method was one of the methods used to fully understand the current problem with the hospital record management system in Ghana as a country.

**SYSTEM CONFIGURATION**

**HARDWARE REQUIREMENTS**

PROCESSOR : INTEL PENTIUM 4 (OR) HIGHER

RAM : 512 MB & ABOVE

HARD DISK DRIVE : 500 MB FREE SPACE OR ABOVE

PRINTER : INK-JET PRINTER

PEN DRIVE : 512MB.

**SOFTWARE REQUIREMENTS**

PROGRAMMING LANGUAGE : Visual C#.NET

BACKEND : MSSQL SERVER-2018

FRONT END : Visual C#.NET

OPERATING SYSTEM : WINDOWS 7 & HIGHER VERSION

**Software Specification**

##### VISUAL STUDIO. NET

Visual Studio .NET is a complete set of development tools for building

* ASP Web applications
* XML Web services
* desktop applications
* mobile applications

Visual Basic .NET, Visual C++ .NET, and Visual C# .NET all use the same integrated development environment (IDE), which allows them to share tools and facilitates in the creation of mixed-language solutions.

**Visual Basic**

Visual Basic has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, and custom attributes. In addition, Visual Basic supports multithreading. Multithreading is the ability to assign individual tasks to separate processing threads.

**C#**

Visual C#, pronounced C sharp, is a new object-oriented programming language that is an evolution of C and C++, providing a simple and type-safe language for developing applications.

**C++**

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework. Attributes, like C++ keywords, are used in your source files and interpreted by the compiler. Attributes are designed to provide a quick and efficient method to simplify COM programming with Visual C++.

**JScript**

JScript has been updated to be a class-based, object-oriented scripting language that maintains full backwards compatibility with previous versions of JScript. JScript now provides class-based objects, typed variables, true compiled code, and cross-language support through Common Language Specification (CLS) compliance. The primary role of JScript is development of Web sites with ASP.NET and customization of applications with Script for the .NET Framework.

**Types of forms used in .Net**

**Web Forms**

Web Forms are an ASP.NET technology that you use to create programmable Web pages. Web Forms render themselves as browser compatible HTML and script, which allows any browser on any platform to view the pages. Using Web Forms, you create Web pages by dragging and dropping controls onto the designer and then adding code, similar to the way that you create Visual Basic forms

**Windows Forms**

Windows Forms is the new platform for Microsoft Windows application development, based on the .NET Framework. This framework provides a clear, object-oriented, extensible set of classes that enables you to develop rich Windows applications. Additionally, Windows Forms can act as the local user interface in a multi-tier distributed solution.

**The .NET Framework**

##### Overview

The .NET Framework is a new computing platform that simplifies application development in the highly distributed environment of the Internet. The .NET Framework is designed to fulfill the following objectives:

* To provide a consistent object-oriented programming environment whether object code is stored and executed locally, executed locally but Internet-distributed, or executed remotely.
* To provide a code-execution environment that minimizes software deployment and versioning conflicts.
* To provide a code-execution environment that guarantees safe execution of code, including code created by an unknown or semi-trusted third party.
* To provide a code-execution environment that eliminates the performance problems of scripted or interpreted environments.
* To make the developer experience consistent across widely varying types of applications, such as Windows-based applications and Web-based applications.
* To build all communication on industry standards to ensure that code based on the .NET Framework can integrate with any other code.

The .NET Framework is a multi-language environment for building, deploying, and running XML Web services and applications.

* **Common Language Runtime** Despite its name, the runtime actually has a role in both a component's runtime and development time experiences. While the component is running, the runtime is responsible for managing memory allocation, starting up and stopping threads and processes, and enforcing security policy, as well as satisfying any dependencies that the component might have on other components. At development time, the runtime's role changes slightly; because it automates so much (for example, memory management), the runtime makes the developer's experience very simple, especially when compared to COM as it is today. In particular, features such as reflection dramatically reduce the amount of code a developer must write in order to turn business logic into a reusable component.
* **Unified programming classes** The framework provides developers

With a unified, object-oriented, hierarchical, and extensible set of class libraries (APIs). Currently, C++ developers use the Microsoft Foundation Classes and Java developers use the Windows Foundation Classes. The framework unifies these disparate models and gives Visual Basic and JScript programmer’s access to class libraries as well. By creating a common set of APIs across all programming languages, the common language runtime enables cross-language inheritance, error handling, and debugging. All programming languages, from JScript to C++, have similar access to the framework and developers are free to choose the language that they want to use.

* **ASP.NET** ASP.NET builds on the programming classes of the .NET Framework, providing a Web application model with a set of controls and infrastructure that make it simple to build ASP Web applications. ASP.NET includes a set of controls that encapsulate common HTML user interface elements, such as text boxes and drop-down menus. These controls run on the Web server, however, and push their user interface as HTML to the browser. On the server, the controls expose an object-oriented programming model that brings the richness of object-oriented programming to the Web developer. ASP.NET also provides infrastructure services, such as session state management and process recycling that further reduce the amount of code a developer must write and increase application reliability. In addition, ASP.NET uses these same concepts to enable developers to deliver software as a service.

##### Debugging

Visual Studio .NET provides a single integrated debugger for all Visual Studio languages, including Visual Basic and C#. A new, unified interface combines features of the Visual C++ and Visual Basic 6.0 debuggers, as well as many new features.

* **Cross-Language Debugging**

Using the integrated debugger, you can debug projects that are part of the same solution but are written in different languages. For example, you can debug a solution that contains a project consisting of a Visual Basic or Visual C# user interface application and a Visual C++ server application, and you can step back and forth between these projects, for example from Managed Extensions for C++ to Visual Basic and back to Managed Extensions for C++.

* **Attaching to a Running Program**

You can attach the debugger to a program that is already running and debug the program. Attaching to a running program works the same way whether the program is running on a host machine or a remote machine. The program does not need to be launched in the Visual Studio .NET integrated development environment (IDE).

* **Remote Debugging**

You can attach to and debug a process that is running on a different computer from the one on which you are running Visual Studio. For example, if you are debugging a Windows Forms client application and an XML Web service it uses, you can run Visual Studio on the client and then attach to the server that is providing the XML Web service. You typically perform this kind of debugging for Web Forms projects or ASP.NET Web Service projects.

* **Debugging Multithreaded Applications**

New for Visual Basic users is the ability to write and debug multithreaded applications. The Threads window can be used to view the threads that are running and to switch context.

* **Debugging Multiple Programs**

You can debug multiple programs by attaching to running programs or by launching multiple programs from the Visual Studio IDE.

* **Debugging ASP.NET Web Applications**

Configuring debugging for ASP.NET Web applications has been significantly improved. In most cases you can simply choose Start from the Debug menu and the debugger will automatically attach to the ASP.NET worker process for debugging Web Forms, even if the ASP.NET worker process is running on a different server. Security configuration has been made easier by the addition of a debugger users group.

* **.NET Framework Classes for Debugging and Code Tracing**

You can include instrumentation code in your Microsoft .NET application. Several .NET Framework classes are available for use both in debugging and in instrumenting your code.

**SYSTEM DESIGN**

**THE SYSTEM DESIGN PROCESS:**

System design develops the architectural detail required to build a system or product.

**The system design process encompasses the following activities:**

* Partition the analysis model into subsystems.
* Identify concurrency that is dictated by the problem.
* Allocate subsystems to processors and tasks.
* Develop a design for the user interface.
* Choose a basic strategy for implementing data management.
* Identify global resources and the control mechanisms required to access them.
* Design an appropriate control mechanism for the system, including task management.
* Consider how boundary conditions should be handled.
* Review and consider trade-offs.

##### Input design

Input design is a part of overall system design, requires the very careful analysis of the input data items. The goal of the input design is to make the data entry easier, logical and free from errors. The user controls input data.

The commonly used input, output devices are mouse, keyboard and the visual display unit. The well designed, well organized screen formats are used to acquire the inputs. The data accepted is stored on database file.

My system is classified into subsystem such as

* Administrator
* Doctors Details
* Patient Details
* Diagnosis

##### Output Design

Output is the most important and direct source of information the user.

Efficient & intelligent output design improves the system relationships with the users and helps in decision-making. The output is collected in order to help the user to make a wise decision.

***PATIENT***

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **CONSTRAINTS** |
| patid | int | Primary key |
| patfirstname | varchar(50) | Not null |
| patlastname | varchar(50) | Not null |
| pataddress | varchar(50) | Not null |
| patphone | varchar(50) | Not null |
| patdob | date | Not null |
| patage | int | Not null |
| patgender | varchar(50) | Not null |
| patblood | varchar(50) | Not null |
| patdisease | varchar(50) | Not null |
| image | image | Not null |

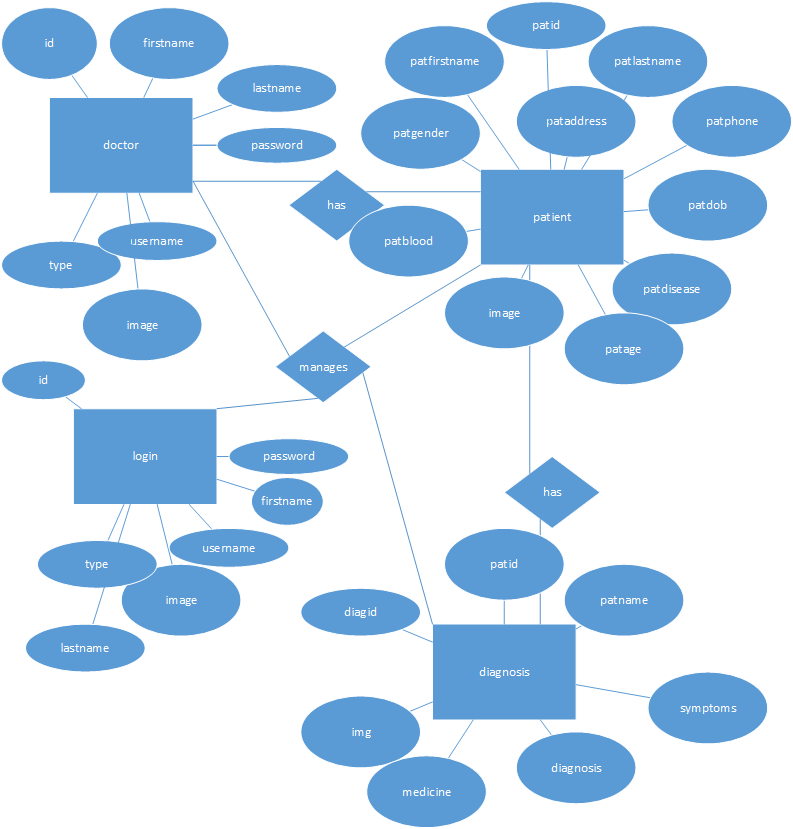
***DIAGNOSIS***

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **CONSTRAINTS** |
| diagid | Int | Primary key |
| patid | Int | Not null |
| patname | Varchar(50) | Not null |
| patlastname | Varchar(50) | Not null |
| Symptoms | varchar(80) | Not null |
| diagnosis | Varchar(150) | Not null |
| medicines | varchar(50) | Not null |
| image | image | Not null |

***ADMIN/DOCTOR***

|  |  |  |
| --- | --- | --- |
| **NAME** | **TYPE** | **CONSTRAINTS** |
| id | int | Primary key |
| firstname | nvarchar(50) | Not null |
| lastname | nvarchar(50) | Not null |
| username | nvarchar(50) | Not null |
| password | nvarchar(50) | Not null |
| type | nvarchar(50) | Not null |
| image | image | Not null |

**DATAFLOW DIAGRAM**

****

### SYSTEM TESTING

#### System Testing

System Testing is an important stage in any system development life cycle. Testing is a process of executing a program with the intention of finding errors. The importance of software testing and its implications with respect to software quality cannot be overemphasized. Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and coding. A good test case is one that has a high probability of finding a yet undiscovered error.

Testing is the set of activities that can be planned in advance and conducted systematically. Different test conditions should be thoroughly checked and the bugs detected should be fixed. The testing strategies formed by the user are performed to prove that the software is free and clear from errors. To do this, there are many ways of testing the system’s reliability, completeness and maintainability.

**Unit Testing:**

In the unit testing the analyst tests the program making up a system. The software units in a system are the modules and routines that are assembled and integrated to perform a specific function. In a large system, many modules on different levels are needed.

Unit testing can be performed from the bottom up starting with the smallest and lowest level modules and proceeding one at a time. For each module in a bottom-up testing, a short program executes the module and provides the needed data.

**Integration Testing:**

Integration testing is a systematic technique for constructing the program structure while conducting test to uncover errors associate with interfacing. Objectives are used to take unit test modules and built program structure that has been directed by design.

The integration testing is performed for this Hospital Management System when all the modules where to make it a complete system. After integration the project works successfully.

**Validation Testing:**

Validation testing can be defined in many ways, but a simple definition is that can be reasonably expected by the customer. After validation test has been conducted, one of two possible conditions exists.

* The functions or performance characteristics confirm to specification and are accepted.
* A deviation from specification is uncovered and a deficiency list is created.

Proposed system under consideration has been tested by using validation testing and found to be working satisfactorily.

For example, in this project validation testing is performed against inpatient search module. This module is tested with the following valid and invalid inputs for the field pat name.

##### White Box Testing

White box testing, sometimes called glass-box testing is a test case design method that uses the control structure of the procedural design to derive test cases. Using white box testing methods, the software engineer can derive test cases that

* Guarantee that all independent paths with in a module have been exercised at least once.
* Exercise all logical decisions on their true and false sides.
* Execute all loops at their boundaries and within their operational bounds and
* Exercise internal data structure to assure their validity.

For example in this project white box testing is performed against inpatient module. Without entering text if we apply it displays the message “First add record then save it” else it should be saved.

##### Black Box Testing

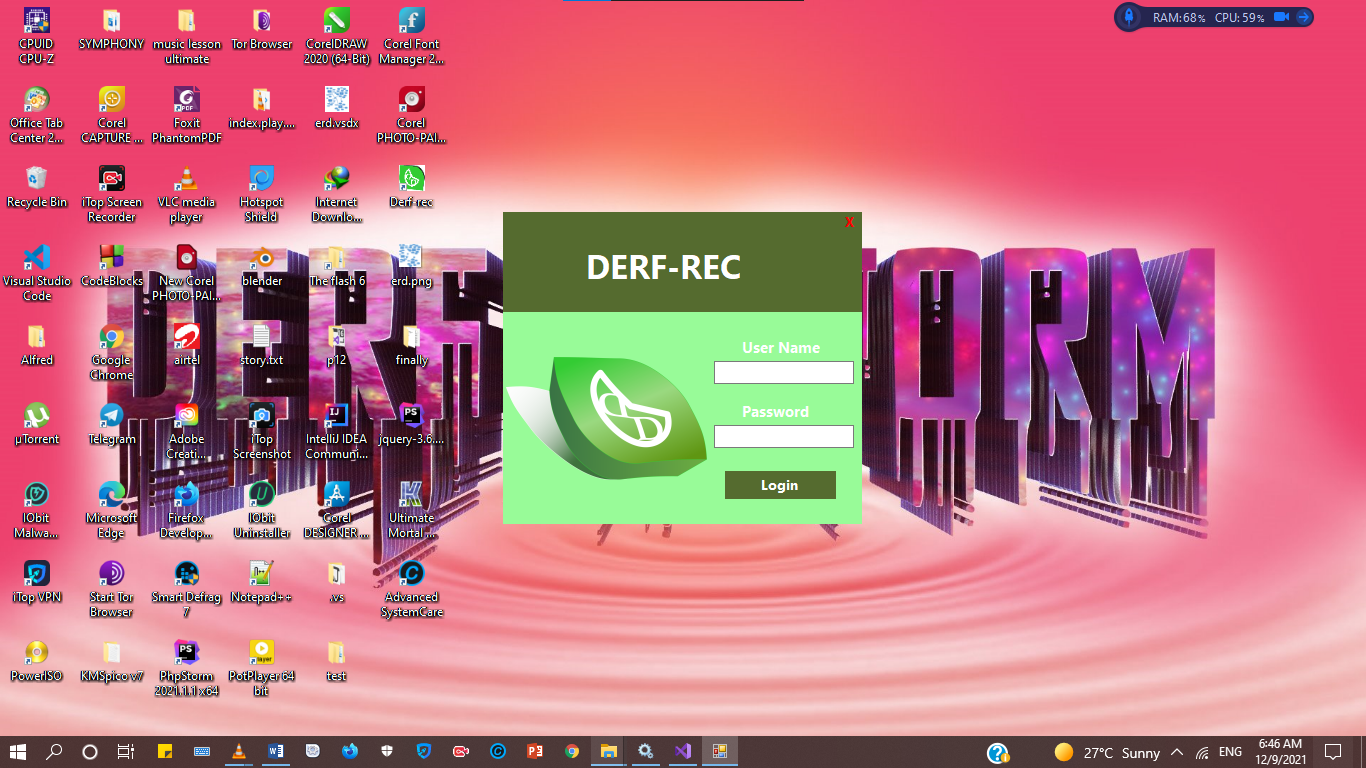
This method treats the coded module as a black box. The module runs with inputs that are likely to cause errors. Then the output is checked to see if any error occurred. This method cannot be used to test all errors, because some errors may depend on the code or algorithm used to implement the module.

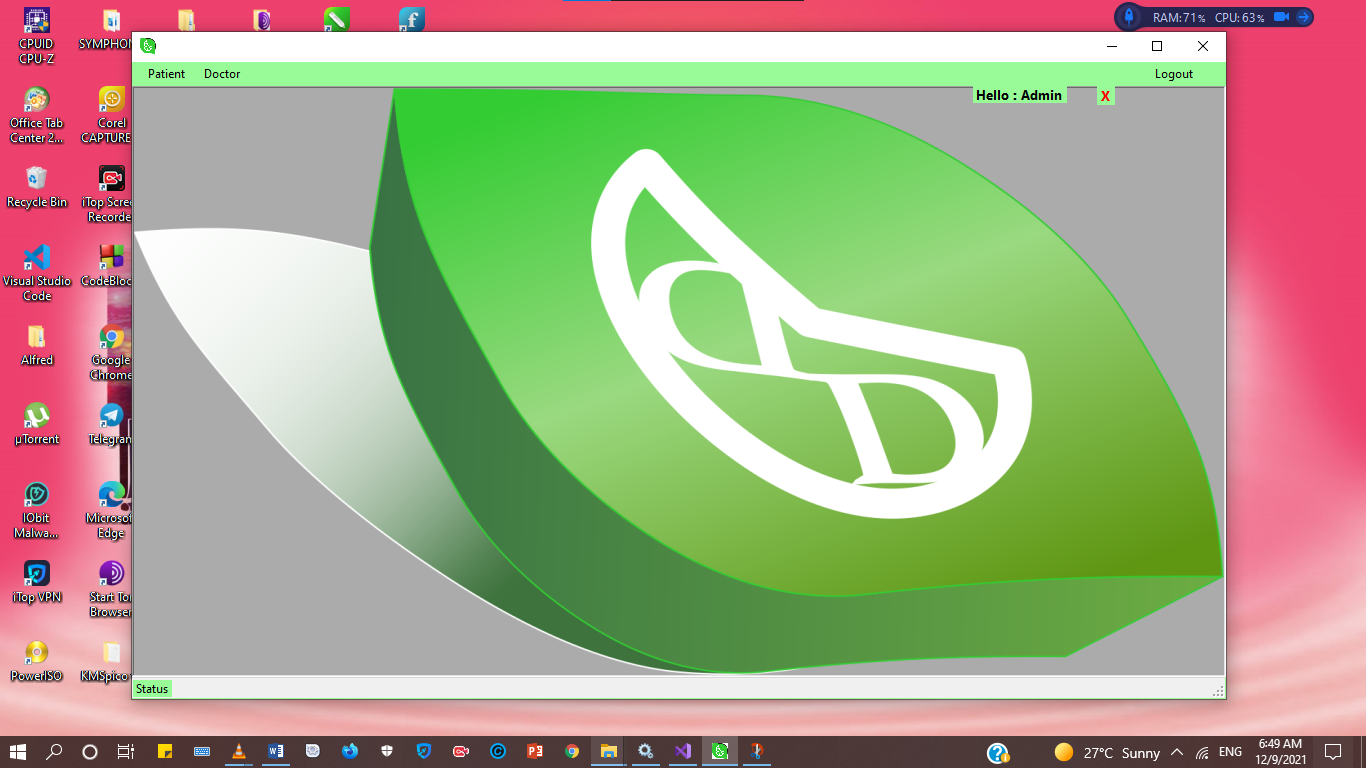
##### System Implementation

Implementation is the process of having system personal check out and provides new equipment into use, train the users to install a new application and construct any files of data needed to use it. There are three types of implementation.

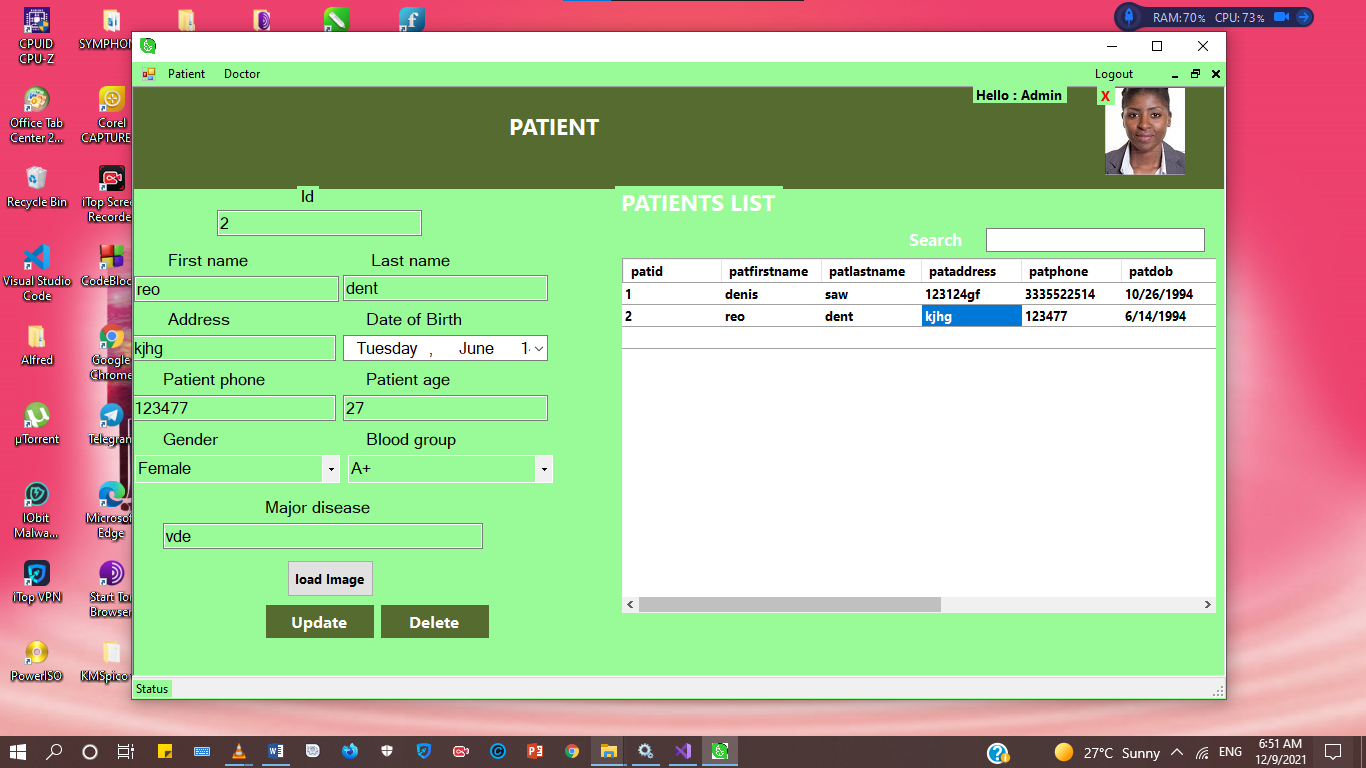
* Implementation of computer system to replace a manual system. The problems encountered are covering files, training users, creating accurate files and verifying print outs for integrity.
* Implementation of a new computer system to replace an existing one. This is usually difficult conversion. If not properly planned, there can be many problems. So large computer systems may take as long as a year to convert.
* Implementation of a modified application to replace the existing one using the same computer. This type of conversion is relatively easy to handle, usually there are no major changes in the file.

**LOGIN FORM**

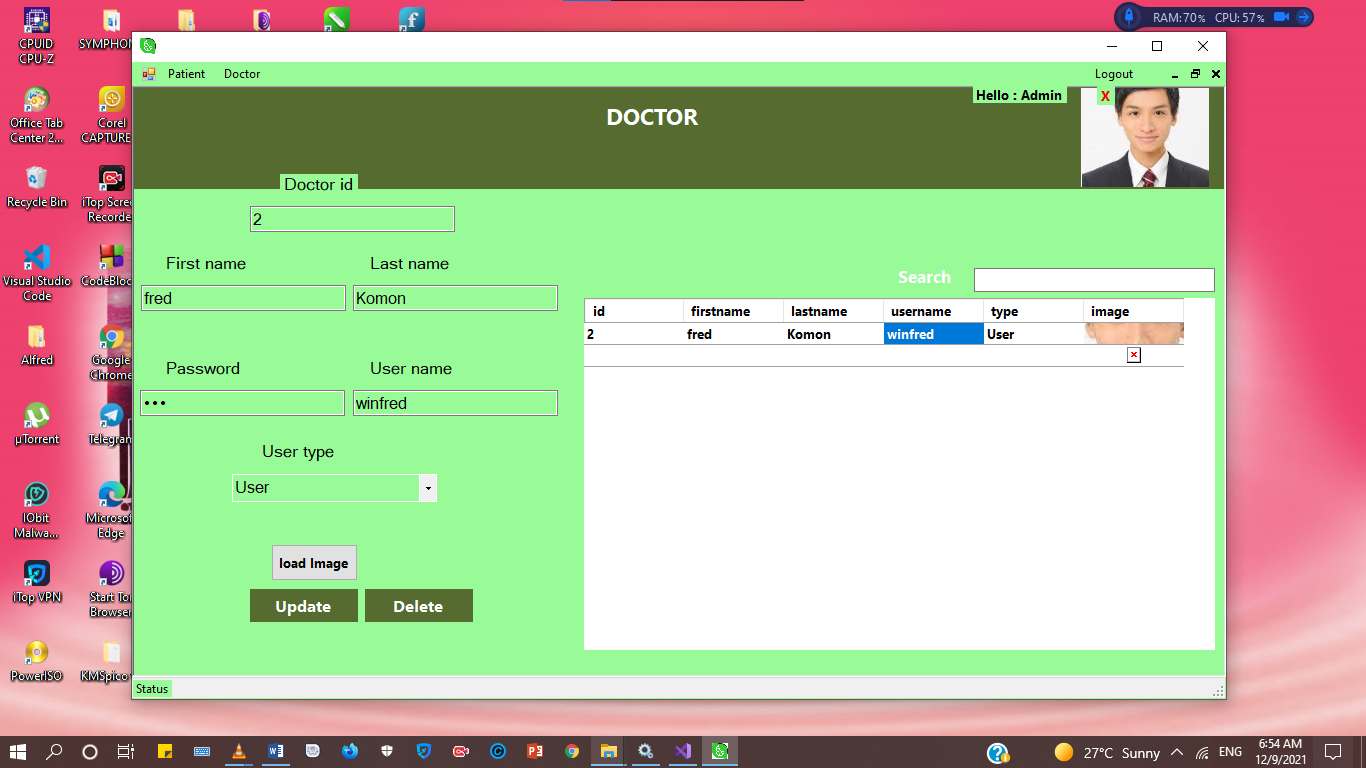


**ADMIN DASHBOARD**

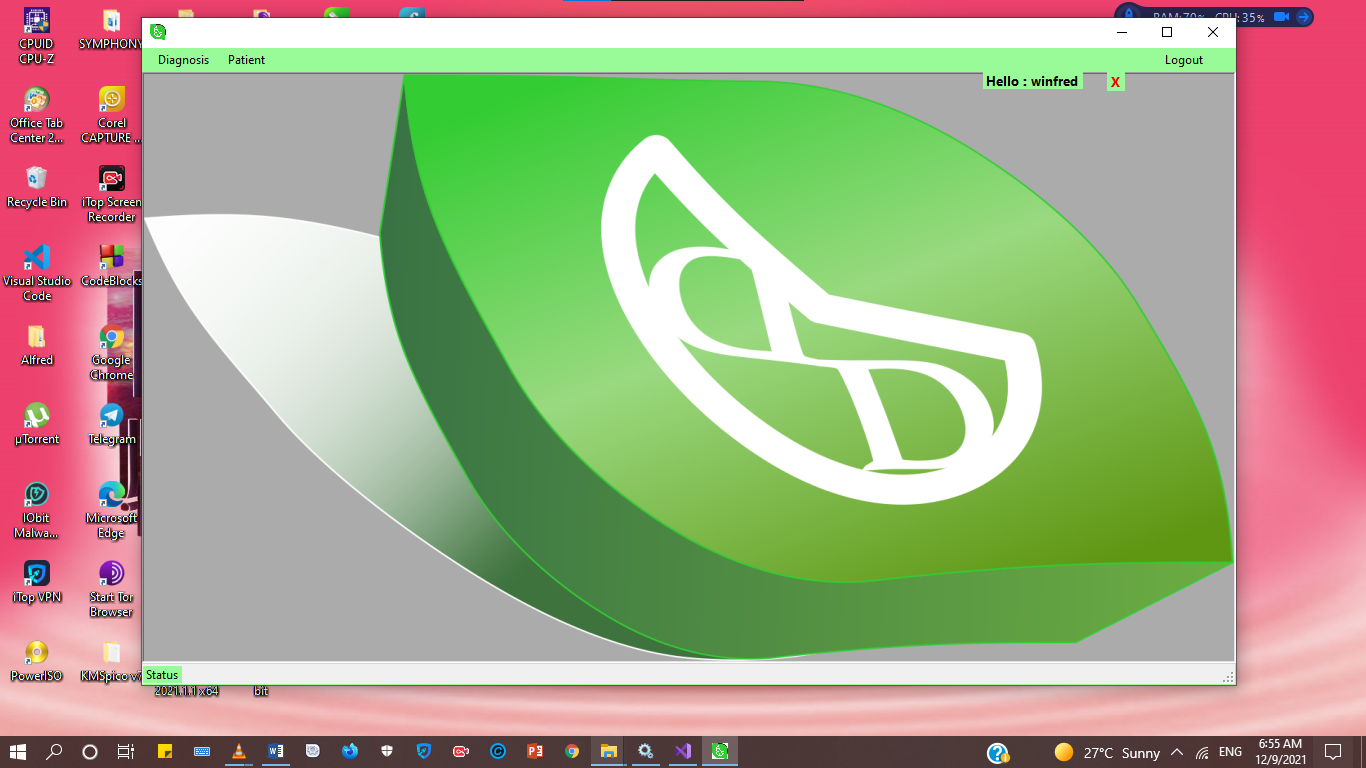
**PATIENT REGISTRATION FORM**



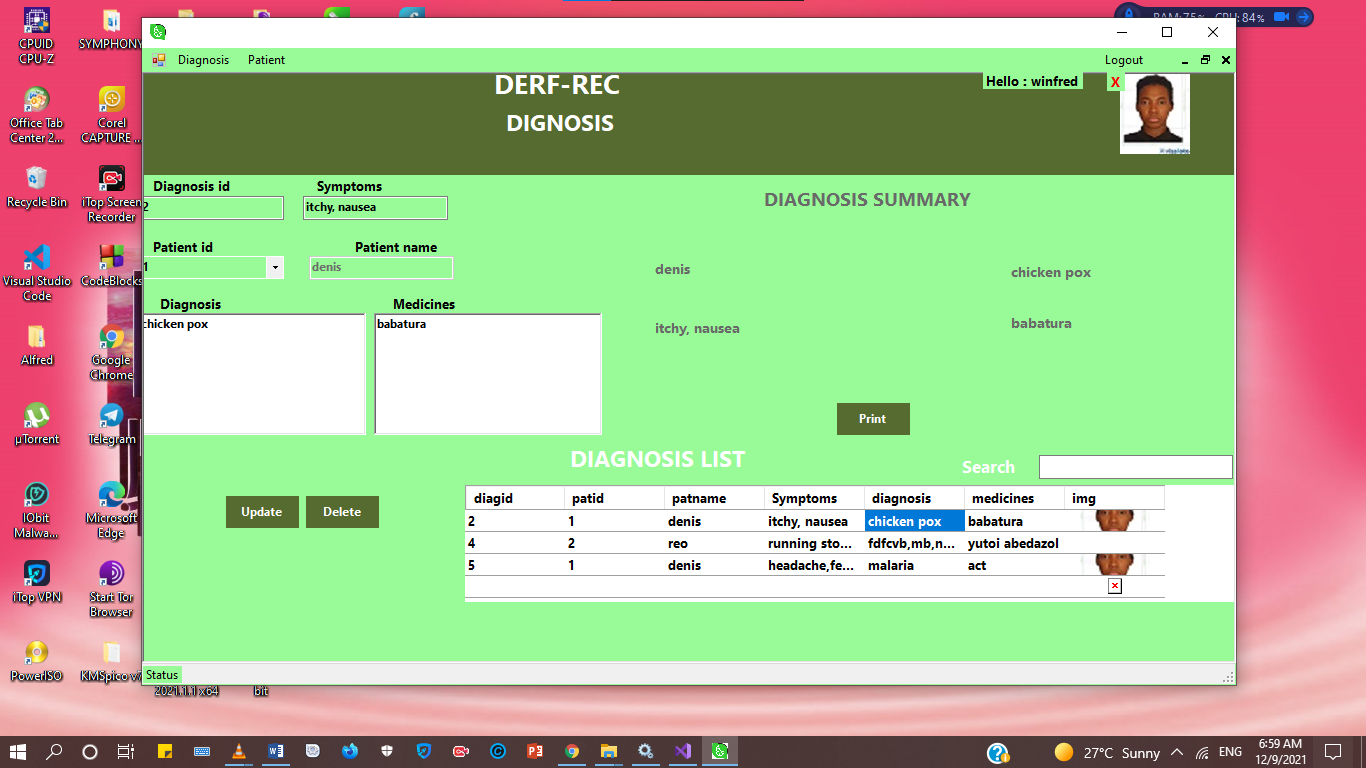
**DOCTORS REGISTRATION FORM**

****

**DOCTORS DASHBOARD**

****

**DOCTORS DIAGNOSIS FORM**

****