DNA PCR ANALYSIS SOFTWARE

By

PC Muhammad Ikram

PC Anam Rafique

PC Anaum Tariz

NC Zainab Ikram

Submitted to the Faculty of Department of Computer Software Engineering

National University of Sciences and Technology, Islamabad

in partial fulfillment for the requirements of a B.E Degree in

Computer Software Engineering

AUGUST 2011

ABSTRACT

PCR is one of the latest developed techniques in Medical Sciences, so there is a lack of PCR tools and equipment. Moreover, no recognized software is available to carry out the complete PCR tests from start to the diagnosis phase. Our software team is working in collaboration with Maj. Gen Suhaib Ahmed for the development of a Genetic Software for the PCR endpoint analysis and Real Time PCR.

The hardware portion of the device developed in AFIP is done by Maj. Gen Suhaib whereas its software portion is entirely done by our software team. The main aim of this project is to develop a diagnostic utility within the country and within the available resources which is purely meant for virology. It will perform qualitative as well as quantitative analysis on DNA images and generate a report specifying the presence of a certain virus in the patient's blood sample. This product is completely developed locally which reflects the research of one of our best pathologists and also the dynamic work of software engineering students. This collaborative work also meets standards in all respects.

The software takes the images of the test tubes in the presence of different lights and filters. The software then crops out the required portion of a DNA image and calculates the values of the control and analysis dye respectively which are used later on for the diagnosis based on some threshold and some positive or negative control. After the diagnosis the reports are generated by the software and the graphical representation of the result is also shown for complete analysis.

ACKNOWLEDGEMENTS

There is no success without the will of ALLAH. We are grateful to ALLAH, who has given us guidance, strength and enabled us to accomplish this task. Whatever we have achieved, we owe it to Him, in totality.

We are also grateful to our parents and family and well-wishers for their admirable support.

We would like to thank our supervisor Dr Asif Masood and our co-supervisor Maj Gen Suhaib Ahmed for their motivation throughout the course of our project. Without his help we would have not been able to accomplish anything. He provided us with the opportunity to polish our technical skills and guided us in this area of learning.

Our project group would also like to acknowledge the administration of CS Dept, Military College of Signals for their support and assistance. We are also extremely thankful to all the instructors who imparted knowledge on us during the course of our engineering studies at Military College of Signals, NUST.

PREFACE

The discovery of DNA, its structure, and function was probably the most significant biological discovery of the 20th century. It has had a tremendous impact on science and medicine. Now a days various techniques are used in pathology for the diagnosis of diseases and to detect certain genetic predispositions to disease. PCR is one of the latest developed technique on which doctors and researchers are working on. This is a medical diagnosis software which has been developed with the help of a prestigious institution of Army Medical College i.e. AFIP. The support of one of the classified pathologist of AFIP and the current Dy Cmdt Dr. Gen Suhaib in hardware of this project is remarkable. The hardware of this project is completely built at AFIP and using the local resources which was led by Dr. Gen Suhaib. The software part was the most important section of this project which was developed by us as our Final Year Degree project. The overall coding part and designing of the software is done by PC Muhammad Ikram and PC Anam Rafique. The interface and Database work is performed by PC Anaum Tariq and NC Zaineb Ikram.

Table of Contents

Chapter 1

Introduction

1.1 Introduction

The discovery of DNA, its structure, and function was probably the most significant biological discovery of the 20th century. It has had a tremendous impact on science and medicine. From identifying genes that lead to the development of diseases, to producing pharmaceuticals to treat them, identifying and analyzing genes has led to extraordinary breakthroughs that have changed the face of the future of science forever.

In the field of modern medicine and genetic research, the discovery of DNA has allowed for the improved ability to diagnosis disease, detect genetic predisposition to disease, create new drugs to treat disease, use gene therapy as treatment, and design "custom drugs" based on individual genetic profiles. These breakthroughs now offer hope for patients who suffer from what were once untreatable diseases.

The effects that the discovery of DNA have had on medicine are truly remarkable, but the impact crosses over into all aspects of our society. From cloning, to paternity cases, to determining the guilt or innocence of a suspect in a crime, to identifying victims, to breeding disease-resistant farm animals and growing more nutritious produce, the classification, analysis and manipulation of genes has transformed our world.

Now a days various techniques are used in pathology for the diagnosis of diseases and to detect certain genetic predispositions to disease. PCR is one of the latest developed technique on which doctors and researchers are working on.

1.1.1 Discovery of PCR

PCR was invented by Kary Mullis. At the time he thought up PCR in 1983, Mullis was working in Emeryville, California for Cetus, one of the first biotechnology companies. There, he was charged with making short chains of DNA for other scientists. Mullis has written that he conceived of PCR while cruising along the Pacific Coast Highway 128 one night on his motorcycle. He was playing in his mind with a new way of analyzing changes (mutations) in DNA when he realized that he had instead invented a method of amplifying any DNA region. Mullis has said that before his motorcycle trip was over, he was already savoring the prospects of a Nobel Prize. He shared the Nobel Prize in chemistry with Michael Smith in 1993.

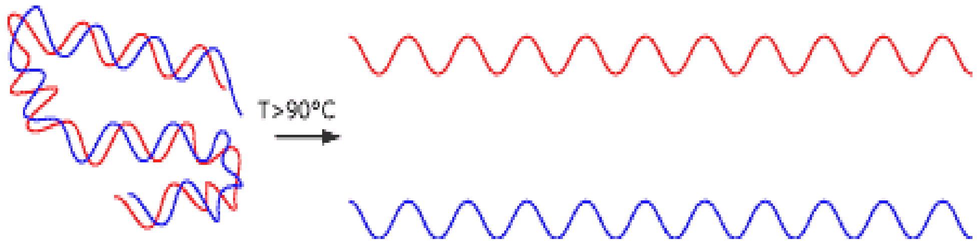
As Mullis has written in the Scientific American: "Beginning with a single molecule of the genetic material DNA, the PCR can generate 100 billion similar molecules in an afternoon. The reaction is easy to execute. It requires no more than a test tube, a few simple reagents, and a source of heat."

1.1.2 What is PCR ?

The PCR (Polymerase Chain Reaction) is a technique for copying a piece of DNA a billion-fold. As the name suggests, the process creates a chain of many pieces, in this case the pieces are nucleotides, and the chain is a strand of DNA. PCR is an enzyme-mediated reaction, and as with any enzyme, the reaction must occur at the enzyme's ideal operating temperature. The enzymes that are used for the PCR are DNA-dependent DNA polymerases (DDDP) derived from thermophilic (heat-loving) bacteria. These DNA polymerases operate at 60-75°C, and can even survive at temperatures above 90°C. This is important because a part of the PCR requires that the reaction reaches ~95°C. Apart from the DNA polymerase, PCR needs a DNA template to copy, and a pair of short DNA sequences called "primers" to get the DNA polymerase started. Three major steps are involved in a PCR.

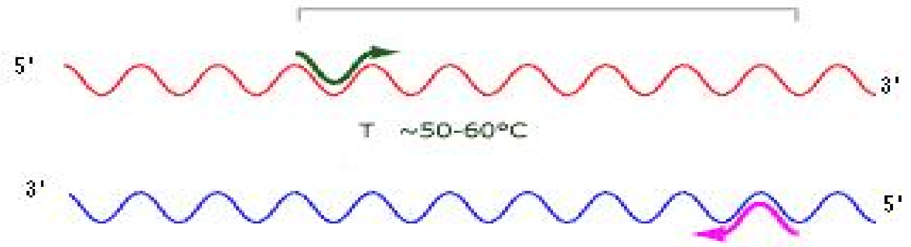
Denaturation

At temperatures above 94°C (201.2 F), double-stranded DNA denatures or "melts". That means the weak hydrogen bonds that usually hold the two complementary strands together at normal temperatures are disrupted resulting in two single stranded DNA strands. (shown below in an idealized form).



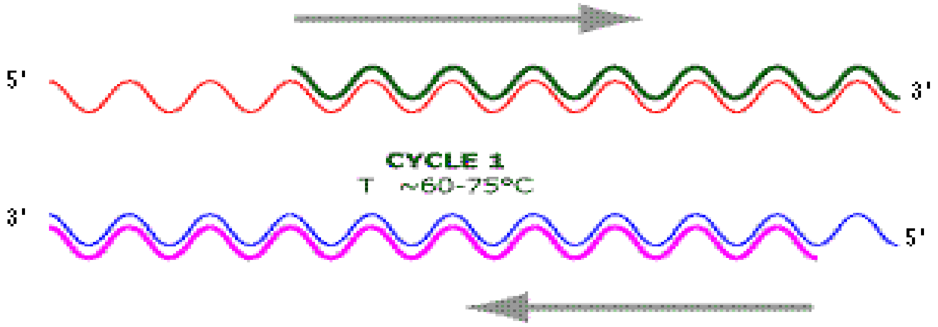
Annealing

At medium temperatures, temp>50, the primers pair up (anneal) with the single-stranded "template" (The template is the sequence of DNA to be copied.) On the small length of double-stranded DNA (the joined primer and template), the polymerase attaches and starts copying the template.



Extension

At 72 C (161.6 F), the DNA polymerase binds to the hybridised primer and begins to add complementary nucleotides (i.e. every time the polymerase reads a "G" on the template strand, its adds a "C"; an "A" for a "T"; a "G" for a "C" and a "T" for an"A"), chemically binding each new addition to the last to form a growing chain. The process only occurs in one direction. In our example, the green primer is binding to its complementary template sequence and is facing toward the right (this is called the 5' (five-prime) to 3' (three prime) direction. Extension occurs in the direction that the primer faces. The result is a new double-stranded PCR product we usually call an "amplicon". An amplicon can be defined as an amplified molecule of a single type, in this case, an exact replicate of the original template.



1.1.3 Purpose and Importance of PCR

PCR permits early diagnosis of malignant diseases such as leukemia and lymphomas, which is currently the highest developed in cancer research and is already being used routinely. PCR also permits identification of non-cultivatable or slow-growing microorganisms such viruses. The basis for PCR diagnostic applications in microbiology is the detection of infectious agents and the discrimination of non-pathogenic from pathogenic strains by virtue of specific genes.

Viral DNA can likewise be detected by PCR. The primers used need to be specific to the targeted sequences in the DNA of a virus, and the PCR can be used for diagnostic analyses or DNA sequencing of the viral genome. The high sensitivity of PCR permits virus detection soon after infection and even before the onset of disease. Such early detection may give physicians a significant lead in treatment. The amount of virus in a patient can also be quantified by PCR-based DNA quantization techniques.

PCR's versatility has been astounding; scientists have produced new contexts and new uses with stunning regularity. These uses have opened new avenues of research, which have in turned proved amenable to new uses of PCR. In less than a decade, PCR has become simultaneously an absolutely routine component of every molecular biology laboratory and a constantly changing tool whose potential has shown no signs of leveling off.

1.1.4 Project Purpose

As PCR is one of the latest developed techniques in Medical Sciences so there is lack of PCR tools and equipment and also no recognized software is available to carry out the complete PCR tests from the start to the diagnosis phase. Engineers, doctors, researchers and students all over the world are working in collaboration to develop a generic tool kit along with complete integrated software for PCR so that it can be readily used in the pathology labs for diagnosis purpose.

Armed Forces Institute of Pathology (AFIP), Rawalpindi is a premier reference diagnostic laboratory of the Pakistan Armed Forces. Maj Gen. Suhaib (Dy cmdt M.H) is carrying out a research on PCR endpoint analysis and Real Time PCR. Our software team is working in collaboration with Maj Gen Suhaib in the development of a Genetic Software for the PCR endpoint analysis and Real Time PCR. The core idea and working on the device being developed in AFIP is done by Maj Gen Suhaib whereas its software portion will be entirely done by our software team. The main aim of this project is to develop a diagnostic utility within the country and within the available resources which is purely meant for virology.

It will perform qualitative as well as quantitative analysis on DNA images and generates a report specifying the presence of a certain virus in the patient’s blood sample. This product will be completely developed locally which will reflect the research of our one of the best pathologist and also the dynamic work of software engineering students. This collaborative work will also meet the standards in all respects

1.1.5 Project Scope

PCR deals with the DNA and its structure so the device being developed works on the medical images of the billions of DNA and based on these images the diagnosis will be done. As the diagnosis results have to be taken care and are used by time to time so there is a need to keep the record of all the results. Also this will be a complete software kit so the main focus will also be on the user interface. So in terms of the Software field this projects lies in the following domains.

1. Image Processing.
2. Databases.
3. Human Computer Interface.

As the proposed system is completely developed from scratch so all the aspects of the Software Project Management will be covered as well which includes the following

1. Software requirement Engineering.
2. Software Design and Architecture.
3. Software Quality Assurance

1.1.6 Intended Audience

This document is intended for

a. **Project Supervisor**

   In order to be sure that the development of the project fulfills the requirement provided in the document.

b. **Project Team**

   In order to be sure that the team is developing the right project that fulfills requirements provided in this document

c. **Project Panel**

   In order to analyze and evaluate the progress of project

d. **Users**

   In order to get familiar with the idea of the project and suggest other features that would make it even more functional.

e. **System administrators**

In order to know exactly what they have to expect from the system, right inputs and outputs and response in error situations.

1.1.7 Document Conventions

As this is a medical diagnosis project is to be used in highly sensitive and confidential environments (i.e. Hospitals). The requirements have to be fully satisfied in order to guarantee the proper use of the software.

It is important to mention here that the requirements for our project keep on changing. The requirements will be modified as the project will progress as per demand of the potential user of the system.

First an overall view about the expected product (SAAZI-GIS) is presented and then all features and functions are explained in brief.

1.1.8 Overall Product Description

Our expected product is divided into two phases. In first phase we will develop software for the End Point Static analysis of PCR. In the second phase we will develop software for the Real Time PCR and this will conclude our complete product.

1.2 Static End Point Analysis of PCR

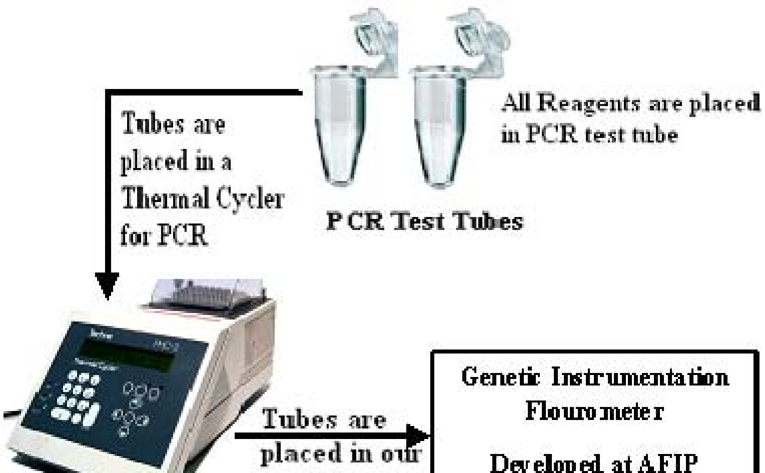
1.2.1 Product Perspective

Our product will be a follow-on member from the family of flourometers. A fluorometer or fluorimeter is a device used to measure parameter of fluorescence. The endpoint analysis of PCR needs to calculate these parameter values to do the diagnosis for different viruses. These parameters include value of a green dye i.e. known as the control dye and value of a yellow dye that is the basically the main analysis dye. The PCR is designed in such a way that in the test tubes the DNA sample the enzymes and the primer dyes and other reagents are placed inside the test tube. This test tube undergoes the 3-step PCR process in a device called Thermal cycler (also known as a Thermo cycler, PCR Machine or DNA Amplifier). . The thermal cycler raises and lowers the temperature of the block in discrete, pre-programmed steps.

If the DNA sample contains the virus then that portion of the DNA is amplified and the dyes also become active at each cycle. As at every cycle the DNA sample is amplified exponentially so the dyes activeness ratio is also increased exponentially at each cycle. Once this step is done then the test tube is taken to the Genetic Instrumentation Hardware where it is placed in a sophisticated hardware assembly which is specially designed for the endpoint analysis. This hardware is connected to the computer where the expected product (SAAZI-GIS) is installed. The hardware functions are controlled by microcontroller.

The software takes the images of the test tubes in the presence of different lights and different filters. The software then crops out the required portion of DNA image and calculates the values of the control and analysis dye respectively which are used later on for the diagnosis based on some threshold and some positive or negative control. After the diagnosis the reports are generated by the software and the graphical representation of the result is also done by the software for complete analysis.

The requirements gathered from the user are completely displayed in the form of a diagram below.


All Reagents are placed in PCR test tube
Tubes are placed in a Thermal Cycler for PCR
PCR Test Tubes
Genetic Instrumentation Flourometer
Developed at AFIP
Tubes are placed in our

1.2.2 Product Features

Broadly our product will have the following major features:

1. The SAAZI-GIS will allow the users to capture images of sample tubes via a UI (User Interface), once they have passed through the thermal cycler.
2. The software will perform the calculations for the fluorescent dyes from the captured images.

3. The software then performs diagnosis on the calculated values based on some threshold provided by the doctors.
4. After classifying the DNA samples as Positive or Negative the software makes a backup of the result in a Database and generates the reports.
5. For detailed analysis for the doctors the software also presents the result in a graphical view as well

1.2.3 Operating Environment

1.2.3.1 Operating System

All 32-bit MS Windows (95/98/NT/2000/XP),Win2K,WinXP, Microsoft Windows Server 2003

1.2.3.2 Database

The complete database of SAAZI-GIS will be made in MYSQL. So complete MYSQL 2.0 database must be installed and fully functional

1.2.4 Development environment

The UI (User Interface) and all the basic logic of the software will be implemented in C# (.NET). Therefore, Visual studio 2008 should be completely functional in all respects on the target computer machines.

1.2.5 PCR- EPA hardware

The self made PCR endpoint analysis hardware (developed at AFIP) should be in working condition and connected to the target computers so that SAAZI-GIS will detect it and connect to the device to perform its respective functions

1.2.6 Design Constraints

i. The UI is designed to be used in the field by medical professionals, and as such must meet their needs to the fullest, and must be easy for them to use.

ii. In the beginning the requirements are not clear because the hardware is also being developed side by side so the requirements keep on changing as the product is developed and tested at regular intervals in accordance with the hardware.

iii. As the requirements change as the implementation proceeds so we cannot make complete design in the start of project.

iv. We have to start the implementation with initial design but when the requirements are completed then we have to remove the overheads and redesign the system again.

1.2.7 Implementation constraints

1. As SAAZI-GIS will be medical diagnosis software so it has to be tested for every new implemented module which requires extra testing (time) to ensure the safety and reliability.

2. This product will be used for official medical tests at pathology labs and hence must conform to a certain level of reliability and availability.

3. The hardware is developed at the same time so during testing it’s a major issue to identify that a certain problem is due to hardware failure or software failure.

1.2.8 Assumptions and Dependencies

1. It’s hard to judge the usability and reliability without testing the product in the pathology labs in collaboration with the doctors.
2. Software assumes that inputs it takes from the device are always valid.

1.3 System Features

1.3.1 New Batch

1.3.1.1 Description

It is the first thing a user must do to begin using SAAZI-GIS. Its main function is to activate all the controls on the interface.

1.3.1.2 Stimulus/Response sequence

1.3.1.2.1 Basic Data Flow

1. User opens SAAZI-GIS.
2. User selects Start->New Batch.
3. The new batch window opens, displaying the basic controls.

1.3.1.3 Alternative Data Flows

1.3.1.3.1 Alternative data flow1

1. User selects Exit
2. Exit from AAZI-GIS

1.3.1.3.2 Alternative data flow2

1. User selects open.
2. Browse for folder window opens

1.3.2 Open Batch

1.3.2.1 Description

This feature allows the user to open an existing batch.

1.3.2.2 Stimulus/Response sequence

1.3.2.2.1 Basic Data Flow

1. User opens AAZI-GIS.
2. User selects Open.
3. Browse for folder window opens.
4. User navigates through his folders.
5. User selects the intended folder.
6. The batch containing the required images opens.

1.3.2.3 Alternative Data Flow

1.3.2.3.1 Alternative data flow1

1. User selects a type of folder non suitable for database, a message of “file not found” appears
2. User selects another folder

1.3.2.3.2 Alternative data flow2

1. User chooses exit.
2. Exit from AAZI-GIS.

1.3.3 Capturing DNA Images

1.3.3a Connecting to camera devices

1.3.3a.1 **Description:**

The system detects all the connected cameras and lets the user select the PCR device camera.

1.3.3a.2 Stimulus/Response Sequences:

1.3.3a.2.1 Basic Data Flow:

1. After creating a new batch, the actifvate option is enabled.
2. User selects Device->Activate.
3. The system detects all connected cameras.

4. User selects the PCR device camera.

5. The live frames of the DNA images are captured and shown in the form of live video in the picture box.

1.3.3a2.2 Alternative Data Flows:

1.3.3a2.2.1 Alternative Data Flow1:

1. User selects Start->Exit.
2. The system exits.

1.3.3a2.2.2 Alternative Data Flow 2:

1. No camera connected.
2. An error message appears.

1.3.3b Capturing the required portion of the DNA image:

1.3.3b.1 Description:

This feature captures the images of control dye and the analysis dye automatically as the user presses the key from the PCR device.

1.3.3b.2 Stimulus/Response Sequences:

1.3.3b2.1 Basic Data Flow:

1. After creating a new batch, the activate option is enabled.
2. User selects Device->Activate.
3. The system detects all connected cameras.
4. User selects the PCR device camera.

1.3.3b2.2 Alternative Data Flows:

1.3.3b.2.2.1 Alternative Data Flow 1:

1. User selects Start->Exit.
2. The system exits.

1.3.3b.2.2.2 Alternative Data Flow 2:

1. No camera connected.
2. An error message appears.

1.3.4 Calculations for the color Dye

1.3.4.1 Description:

This feature calculates the control dye fluorescence values for the DNA sample once the images are captured. Calculations may be performed when the user presses the calculate button or calculations may be performed automatically. Calculations can only be performed for a captured image.

1.3.4.2 Stimulus/Response Sequences:

1.3.4.2.1 Basic Data Flow:

1. The user captures the images by pressing at the capture button.
2. Once the images are captured, the user presses the calculate button.
3. The system calculates the green dye value for a selected image.

1.3.4.2.2 Alternative Data Flows:

1.3.4.2.2.1 Alternative Data Flow 1:

1. The user captures the images.
2. The system automatically calculates the dye value for the captured image.

1.3.4.2.2.2 Alternative Data Flow 2:

1. No image captured.
2. The user presses the calculate button.
3. The system generates an error message.

1.3.5 Perform Diagnoses:

1.3.5.1 Description:

This feature calculates the control dye fluorescence values for the DNA sample once the images are captured. Calculations may be performed when the user presses the calculate button or calculations may be performed automatically. Calculations can only be performed for a captured image.

1.3.5.2 Stimulus/Response Sequences:

1.3.5.2.1 Basic Data Flow:

1. The user selects a sample.
2. **Selection of negative or positive control:**

   The user sets a negative control value.
3. **Setting a threshold value:**

The user sets a threshold value based on a specified criteria for a disease.

4. The user presses the diagnoses button.
5. **Classification based on negative or positive control:**

The system then classifies the selected sample to be positive or negative and diagnoses the presence of a certain disease.

1.3.5.3 Alternative Data Flows:

1.3.5.3.1 Alternative Data Flow 1:

1. The user selects a sample.
2. The user presses the diagnoses button.
3. The system prompts the user to set a negative control value and a threshold.

1.3.5.3.2 Alternative Data Flow 2:

1. The user directly presses the diagnoses button.
2. The system displays an error message.

1.3.5.3.3 Alternative Data Flow 3:

1. The user selects a sample.
2. The user sets a negative control value.
3. The user presses the diagnoses button.

4. The system prompts for a threshold.

1.3.6 Save Batch:

1.3.6.1 Description:

When a user wants to save a batch which includes the captured images along with their respective calculations then save would provide the ability to perform this function.

1.3.6.2 Stimulus/Response Sequences:

1.3.6.2.1 Basic Data Flow:

1. User opens SAAZI-GIS.
2. User selects Start->New Batch.
3. User selects Device->Activate.
4. After capturing images, the user again selects the Start-> Save Batch, a small dialogue box will open to enter the batch name.
5. After entering name, user selects save button.
6. This will save a complete batch.

1.3.6.2.2 Alternative Data Flows:

1.3.6.2.2.1 Alternative Data Flow 1:

1. User enters an invalid batch name.
2. An error message appears.

1.3.6.2.2.2 Alternative Data Flow 2:

1. User chooses cancel.
2. Exit from dialogue box.

1.3.6.2.2.3 Alternative Data Flow 3:

1. User enters an existing file name.

2. Message appears “replace existing file” or “keeping both files”.

1.3.7 Generate Report:

This feature enable user to view to generate a complete report of the patient.

1.3.7.1 Description:

Generate report provide the functionality to generate a report of the results obtained from the analysis done by the system, which would then be viewed on the system as well as the user could take a computer generated print of it as required. User can generate a report once calculation and diagnoses has been done. User can also generate a report out of an existing batch as well.

1.3.7.2 Stimulus/Response Sequence:

1.3.7.2.1 Basic Data Flow:

1. User selects view file
2. User selects start->open
3. User opens the file of a patient whose report has to be generated.
4. User select generate report, this will generate the report of the respective patient.

1.3.7.2.2 Alternative Data Flows:

1.3.7.2.2.1 Alternative Data Flow 1:

1. User chooses cancel.
2. Exit from view file.

1.3.8 Generate Graph:

1.3.8.1 Description:

This feature generates a graph for a selected sample and shows a visual of the results of the test.

1.3.8.2 Stimulus/Response Sequences:

1.3.8.2.1 Basic Data Flow:

1. The user selects a sample by a clicking a single time over it.
2. The user sets a threshold value.
3. The user presses a generate graph button.
4. The system displays the graph for a selected sample.

1.3.8.2.2 Alternative Data Flows:

1.3.8.2.2.1 Alternative Data Flow 1:

1. The user presses the generate graph button.
2. The system prompts the user to select a sample first.

1.3.8.2.2.2 Alternative Data Flow 2:

1. The user selects a sample.
2. The user presses the generate graph button.
3. The system prompts the user to select a threshold value first.

1.4 External Interface Requirements

User Interfaces

User interface is used to provide communication between users and system. SAAZI-GIS will have two main user interfaces.

1. One user interface will be used to connect to the PCR devices, captures the images and saves it to the database in form of a batch along with calculations.

2. The second interface is used to access saved batches (images along with their calculated results). This interface will also allow the users to select a particular negative or positive control out of the saved images. It will also give an option to enter a particular threshold value on the basis of which diagnosis of a particular virus will be done by the software. It will also provide an option to save the batch of images with the new diagnosed results in database.

Hardware Interfaces

1. A USB cable will connect the PCR device camera to the computer so that the software can detect the camera and connects to it on user’s request.

2. Another USB cable will connect the image capturing button on device to the computer. The image will be directly captured by the software as soon as the user presses the button on device. This button uses the click functionality of mouse.

Software Interfaces

Same as 1.2.1.3

Communications Interfaces

None.

1.5 Other Nonfunctional Requirements

1.5.1 Performance Requirements

The system should have a mechanism for maintaining a reasonable level of performance. This is particularly important while performing live video streaming. The system requires high performance that contains all the features necessary for pathologist to perform prompt and accurate diagnoses, ensuring the highest level of patient care.

1.5.2 Safety Requirements

A safety requirement for a medical system is to have greater error resistance and improved patient safety. Records should be saving manually as the system does not automatically save the records in order to avoid any loss of data.

1.5.3 Security Requirements

The software should include an authentication system so that only valid users can use the software in order to secure the software from any unauthorized user. Only system administrator has the right to change system parameters, such as access rights etc. Users need to be authenticated before having access to any confidential data.

1.5.4 Software Quality Attributes

Reliability

The system should have a low failure rate and a high level of service availability. As this is a diagnoses system so it must be reliable.

Efficiency:

Algorithms used must be so efficient that it should take as less time as possible .

Usability:

As it is a desktop application its interface must be such that anyone could use it without any prior training.it shall provide an easy-to-use graphical user interface.

Robustness:

The system should be robust and able to generate accurate result of the captured image.

Portability:

As it is a desktop based application so it should be easily portable to be shifted from one system to another without any problem.

Interoperability:

System shall minimize the effort required to couple it to another system.

Maintainability:

When the system is put to use, new requirements may emerge. When these requirements are emerged, the system should be changeable to accommodate these requirements for maintaining the usefulness of the system. If the system is not maintainable, then the system cannot be modified for new requirements. In this situation, a new system should be developed to provide new requirements.

Chapter 2

System Design and Architecture

2.1 Introduction

This document contains a structure for a software design specification. Most software projects fail because of the flaws in the design, so the design phase can be referred to as a very crucial stage in the software development lifecycle. The purpose of writing this document is to describe the design of the system in detail. This document contains a detailed description of the design of the system, and helps in clearing any doubts which might have been left in the specification of the requirements of the system. Misunderstanding between the users of the system and those developing it can further be clarified if the user can see the design of the system. So this document will help in clarifying the requirements as well as design of the proposed system. Some of the intended audiences (readers) of this document include system customer, project manager, system engineer, system test engineer and system maintenance engineer.

2.2System Overview

This part of the document describes the design relationship between the system, its components and the external environment of the system. There are three main modules of the software system DNA Image Acquisition module, Data Access Module and Diagnosis Module. The DNA Image Acquisition module acquires and process images of the patients blood samples. The Diagnosis Module performs diagnosis over the processed images. The Data Access Module accesses the backend database for data storage and retrieval.

2.3 System Architecture

Architectural Diagrams shows the basic architectural layout of the system being designed. The following figure shows the system architecture of our software. The architectural pattern which we have followed in our project is the traditional MVC (Model View Controller) pattern.

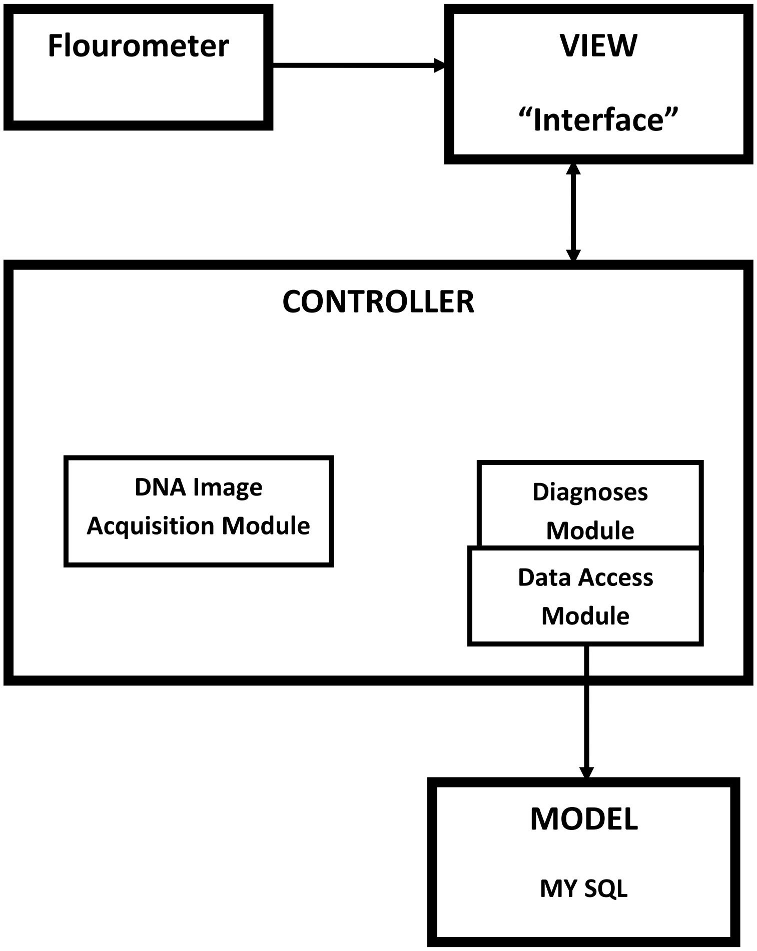


Fig 2.1 Architecture Design of SAAZI-GIS

2.4 Decomposition Description

2.4.1 DNA Image Acquisition Module

First of all a connection to the camera is made (the camera resides within the hardware device) which results in the live streaming of the video. On every user click, a green and a yellow image is acquired. Odd number of clicks results in green images whereas even number of clicks results in yellow images. The software then crops the centre portion of every image and displays the cropped image in a picturebox on the user interface. It then passes the batch to the Data Access Module which saves the batch in the backend database.

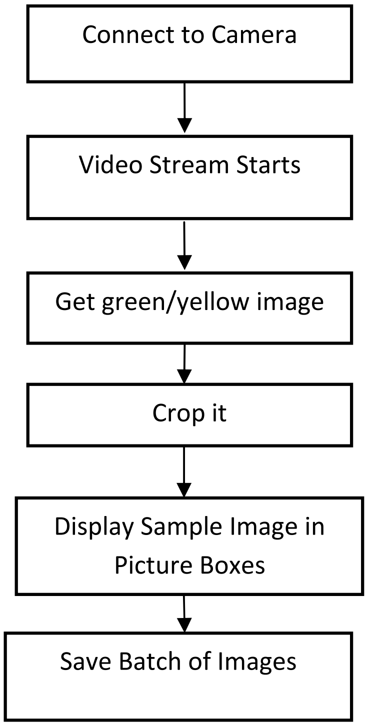
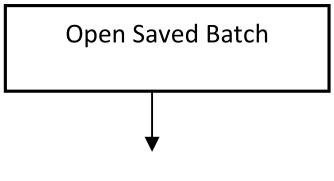


Fig 2.2 Architecture Design of Image Acquisition Module

2.4.2 Diagnosis Module

The Diagnosis Module opens up a saved batch which it gets from the Data Access Module. It extracts the saved images within the batch and opens up diagnosis tools. The user then selects the reference sample image on the basis of which the further diagnosis process will take place. The module sets the image as the reference. The user then inputs the threshold values. Based on the input, the module then performs diagnosis on the blood samples and outputs the results.



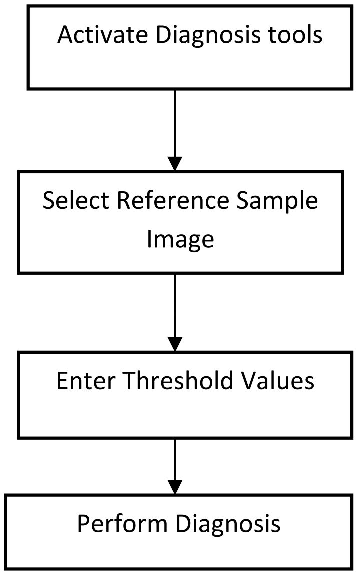
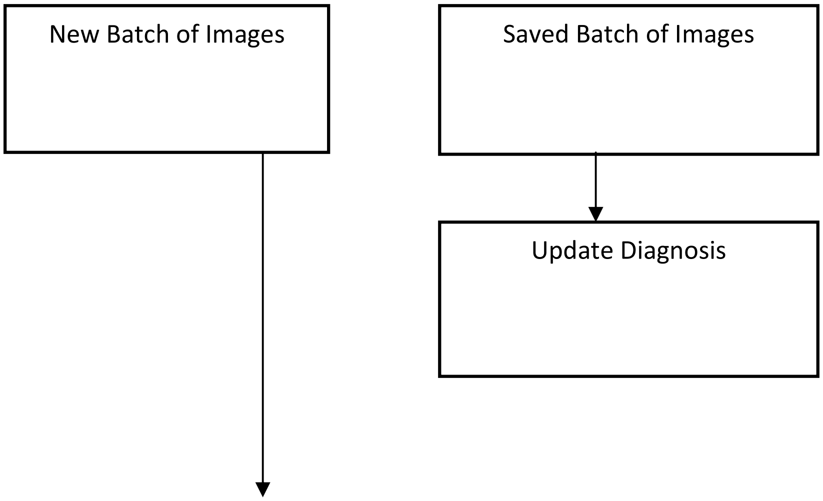


Fig 2.3 Architecture Design of Diagnosis Module

2.4.3 Data Access Module

The Data Access Module works by accessing a backend MySql database. It works either by saving a new batch to the database or by updating an already saved batch of images. While saving a new batch, it first saves the batch on the drive and then to the backend database. While updating an already saved batch, it first accesses the saved batch and then updates it. The module then saves the batch first on the drive and then in the backend database.



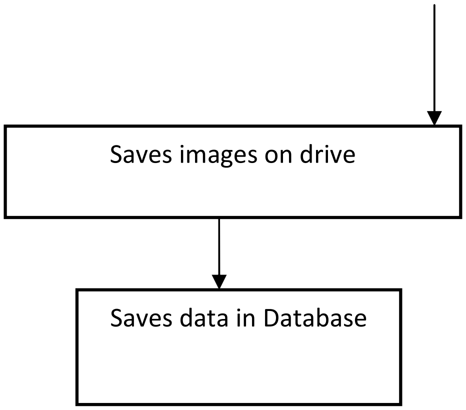
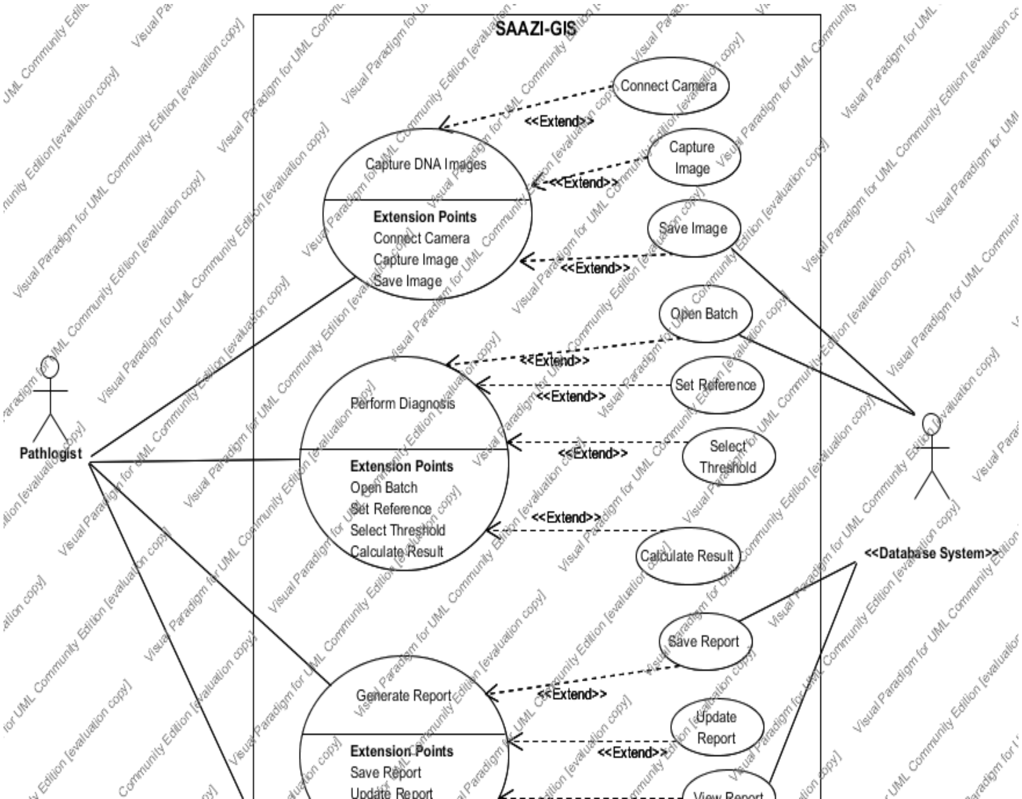


Fig 2.4 Architecture Design of Data Access Module

2.5 Use - Case Diagram

The following diagram shows the use case diagram of our project. There is one primary actor i.e. the Pathologist and two secondary actors' i.e. the Database System and the Printer. There are four main use cases (all four use cases are extended use cases). The first use case i.e. 'Capture DNA Images' is extended by three use cases i.e. 'Connect Camera', 'Capture Image' and 'Save Image'. The second use case i.e. 'Perform Diagnosis' is extended by four use cases i.e. 'Open Batch', 'Set Reference', 'Select Threshold' and 'Calculate Result'.



2.5.1 Fully Dressed Use-cases and Sequence Diagrams

2.5.1.1 Use case UC1: Capture DNA Images

1) **Primary Actor**: Pathologist

2) **Preconditions**: Camera is connected.

3) **Success Guarantee (Post conditions):** image is acquired and shown in a picture box.

4) **Main success scenario / Basic flow:**

    1. User connects to the camera.
    2. User clicks the image acquirer button.
    3. System captures the image.
    4. System crops the image.
    5. System displays the image in a picture box.

5) **Extensions/ Alternative flow:**

    1a. Camera not connected.

    System generates a warning that camera is not connected.

6) **Special requirements**: Image/video should be noise free and Camera calibrated.

7) **Technology and Data Variations List**: No technology and data variations.

Sequence Diagram of UC1: Capture Image

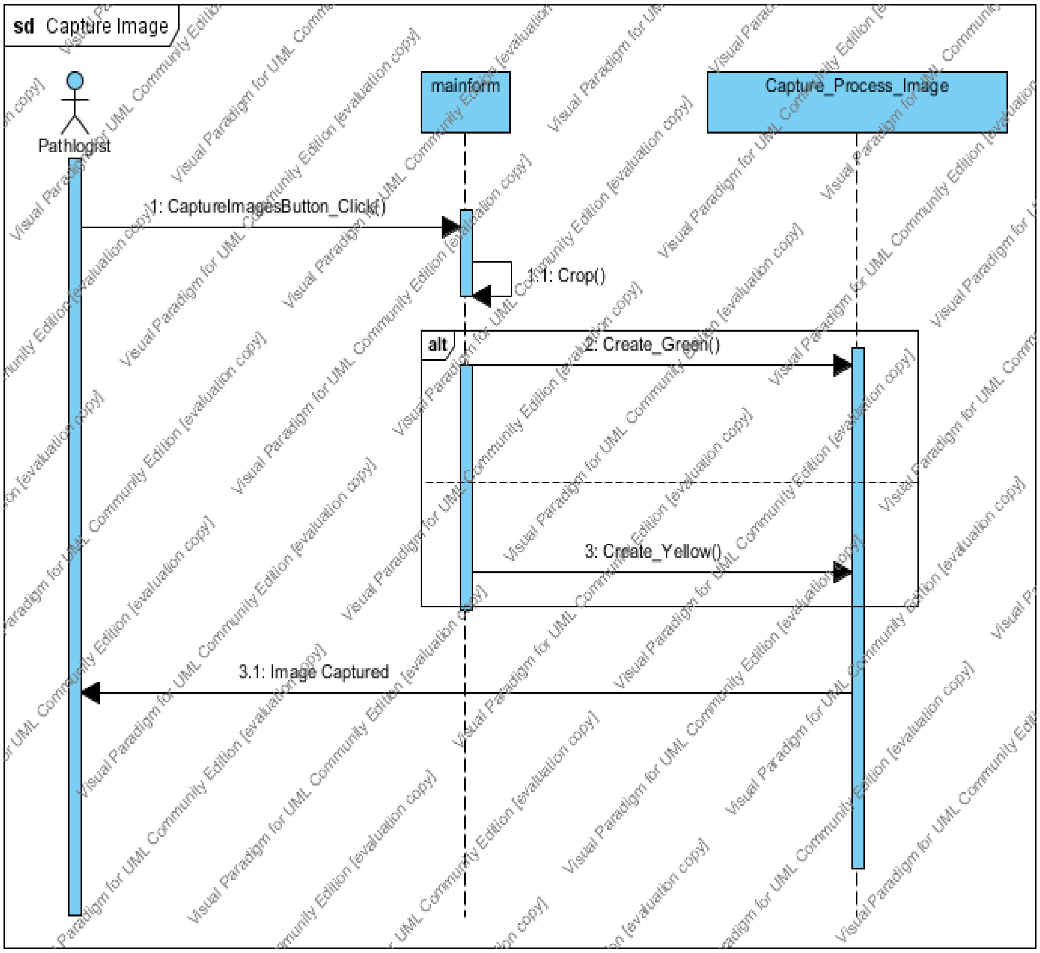


Fig 2.1 Sequence diagram of UC1

2.5.1.2 Use case UC2: Perform Diagnosis

1) **Primary Actor**: Pathologist.

2) **Preconditions**: A saved batch should be opened, a reference should be set and green and yellow thresholds should be selected.

3) **Success Guarantee (Post conditions):** Image classification to be positive or negative.

4) **Main success scenario / Basic flow:**

    1. The user opens a saved batch.
    2. The user selects a reference image.
    3. The user selects the green and yellow thresholds.
    4. The user clicks on the diagnosis button.
    5. The system performs the diagnosis.
    6. The system displays the result by classifying every image in the batch to be positive or negative.

5) **Extensions/ Alternative flow:**

    2a. The user does not select a reference image.

    The system generates an error to select a reference image first.

    3a. The user does not select a green or a yellow threshold. The system generates an error to select a threshold first.

6) **Special requirements**: No special requirement

7) **Technology and Data Variations List**: No technology and data variations.

Sequence Diagram UC2: Perform Diagnosis

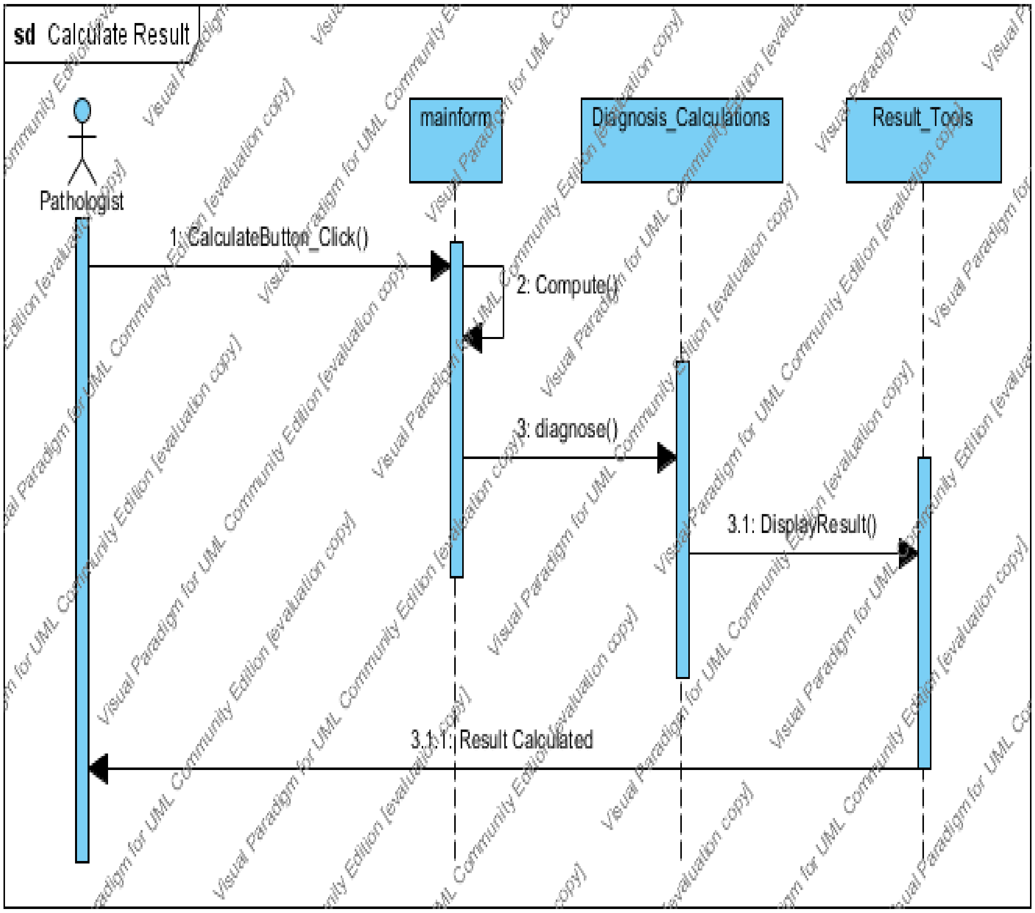


Fig 2.2 Sequence diagram of UC2

2.5.1.3 Use case UC3: Generate Report

1) **Primary Actor**: Pathologist
2) **Preconditions**: A saved batch should be open and results should have been calculated for it.
3) **Success Guarantee (Post conditions):** the system generates a complete report of the diagnosed blood samples.
4) **Main success scenario / Basic flow:**
    1. The pathologist clicks the generate report button.
    2. The system generates the report.
5) **Extensions/ Alternative flow:**

    There are no alternative flows for this particular use case.

6) **Special requirements**: No special requirement

7) **Technology and Data Variations List**: No technology and data variations.

Sequence Diagram of UC3: Generate Report

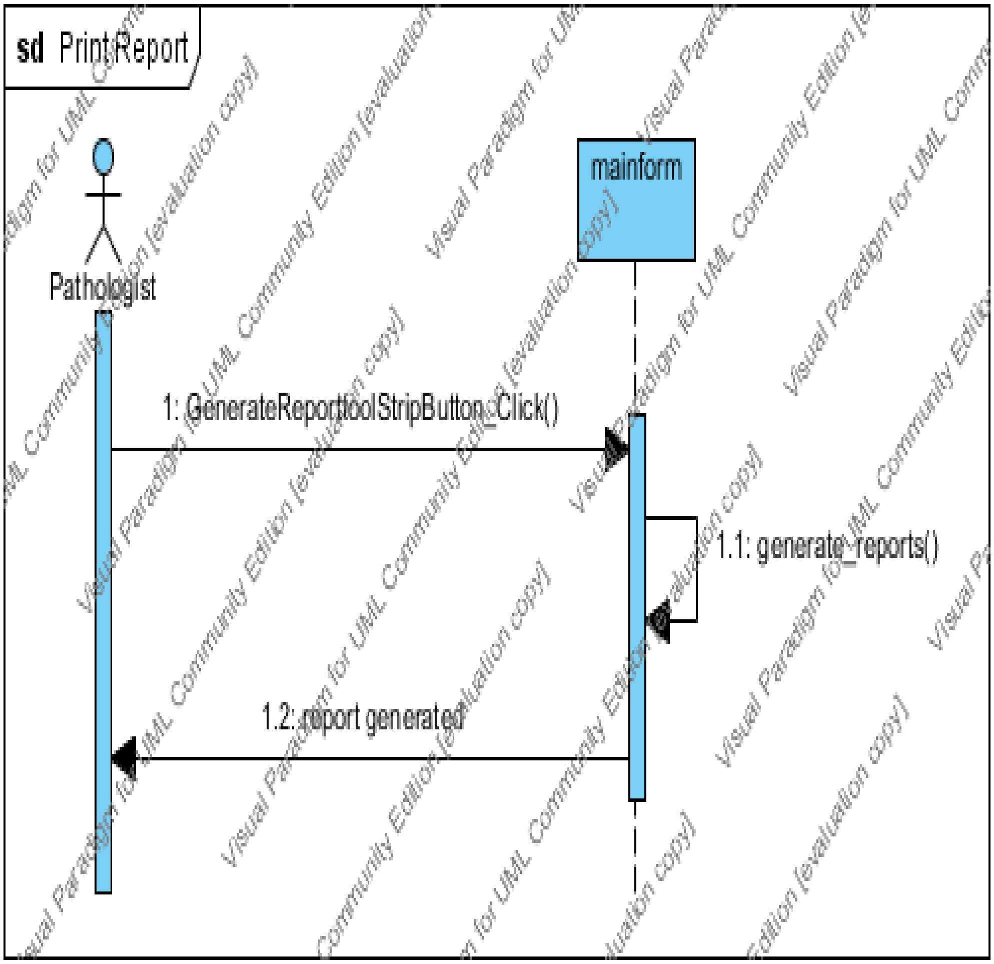


Fig 2.3 Sequence diagram of UC3

2.5.1.4 Use case UC4: Change Camera Settings

1) Primary Actor: Pathologist

2) Preconditions: Camera should be connected.

3) Success Guarantee (Post conditions): picture brightness and contrast should be changed according to the user’s desire.

4) Main success scenario / Basic flow:

1. The pathologist connects to the camera.
2. The pathologist changes the brightness or contrast values.
3. The system makes the changes according to the user’s desires.

5) Extensions/ Alternative flow:

1a.The pathologist does not connect to the camera.

The system generates an error to connect to the camera.

6) Special requirements: No special requirement

7) Technology and Data Variations List: No technology and data variations.

Sequence Diagram of UC4: Examine/Track

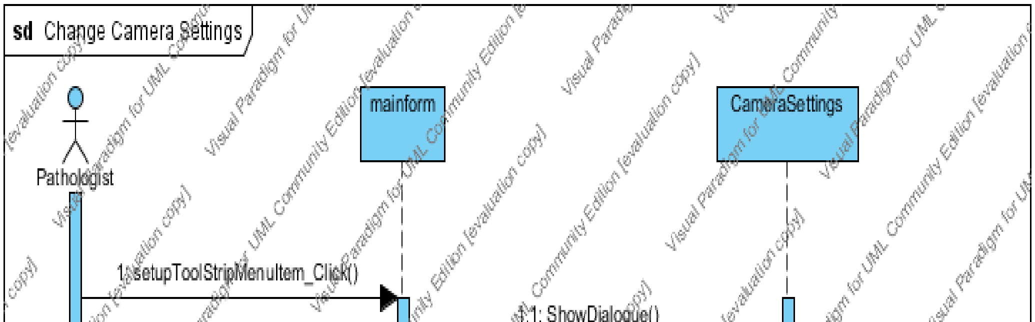


Fig 2.4 Sequence diagram UC4

2.6 Class Diagram

s

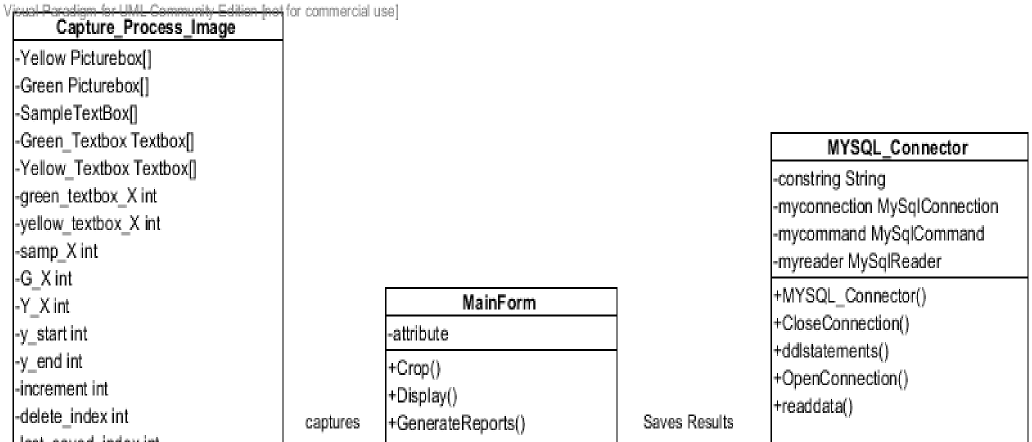


Fig 2.5 Class diagram of our system

2.7 System Sequence Diagram

Figure 2.6 shows the system sequence diagram of our software. It gives the sequence of operations between the pathologist and the system. The pathologist first opens up a new batch, takes images of the blood samples, saves the batch, opens the saved batch, performs diagnosis and then generates the report.

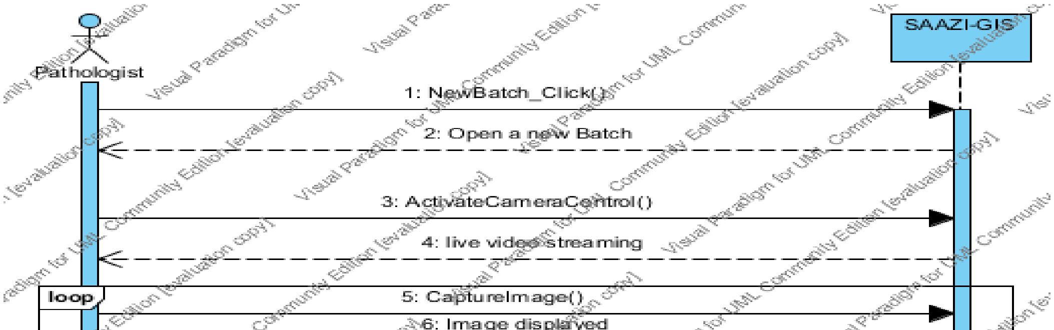


Fig 2.6 System Sequence diagram of our system

2.8 Deployment Diagram

Figure 2.8 shows the deployment diagram of our system. It consists of a hardware device, a PC on which our software is deployed and a printer.

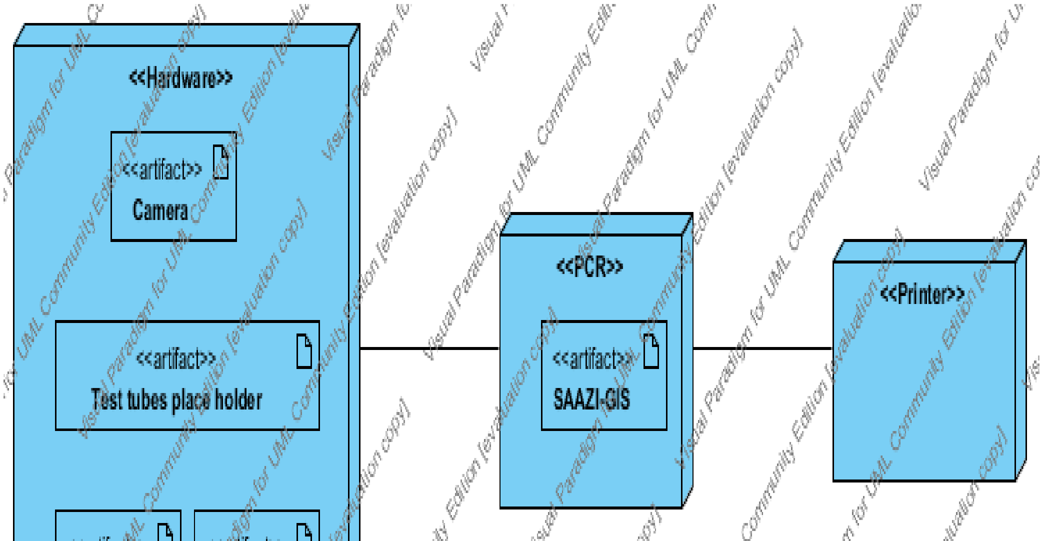


Fig 2.7 System Sequence diagram of our system

Chapter 3

Implementation

5.1 Introduction

The implementation of the software is done using the spiral model and using the iterative development because the requirements were changing as the hardware was developed simultaneously. The main requirements of the software were developed in the start and were verified by the user as soon as the requirement is developed and tested by the team.

Once the implementation is verified by the user the implementation of the next requirement is started and if there are any bugs it is first fixed and then new implementation is started. This approach was used to make sure that none of the functional requirement miss out because this is a medical diagnosis project and all risks have to be considered at every step.

5.2 Classes used

The important classes used in the software are as below

Main_form.cs

In this class all the global functions are defines which are called behind the windows form i.e. User Interface

Capture_Process_Image.cs

This class is mainly responsible for acquiring the video from the device and separates out the images and after doing respective cropping makes the image ready for the processing and diagnosis.

Diagnosis_Calculation.cs

The functionality of this class is to perform the diagnosis bases on the criteria set by the users for a particular batch of blood samples. This class is responsible for making the final decision of presence of any diseases in the particular sample.

MYSQL_Connector.cs

The main functionality of this class is to perform all database related operation i.e connection making, insertion of any data, updating any data and deleting old data.

Result_Tools.cs

This class is responsible for activating all the controls on the User Interface specially once the user is in diagnosis mode of software.

5.3 Most Important functions

ActivateCameraControl()

This function is implemented in Main_Form.cs. This function contain the implementation of the connecting the software with the camera of the PCR hardware and shows the live video from the camera on the software interface in a picture box.

crop()

Once the video is started this function is used to crop the respective yellow and green images and saves in the array of picture box separately.

generate_reports()

Once all the computation and diagnosis is done the user can use this function by clicking a button on UI to print and generate reports for a particular batch of blood samples.

compute()

This function is mainly responsible for the computation of green and yellow value of an image as soon as the image is cropped from the video. These values are used later on for the diagnosis purpose.

Diagnose()

This function is implemented in diagnosis_calculation.cs This the most important function of the whole software as it perform the final diagnosis bases on the pre calculated green and yellow values and also using the threshold and reference image values for diagnosing the negative and positive samples and saves the data for future use and report generation.

Chapter 4

Testing and Results

5.1 Testing

Software Testing is the process of executing a program or system with the intent of finding errors. It involves any activity aimed at evaluating an attribute or capability of a program or system and determining that it meets its required results. Software is not unlike other physical processes where inputs are received and outputs are produced. Where software differs is in the manner in which it fails. Most physical systems fail in a fixed set of ways. By contrast, software can fail in many bizarre ways. Detecting all of the different failure modes for software is generally infeasible.

Unlike most physical systems, most of the defects in software are design errors, not manufacturing defects. Software does not suffer from corrosion, wear and tear generally it will not change until upgrades, or until obsolescence. So once the software is shipped, the design defects or bugs will be buried in and remain latent until activation.

Software bugs will almost always exist in any software module with moderate size, not because programmers are careless or irresponsible, but because the complexity of software is generally intractable and humans have only limited ability to manage complexity. It is also true that for any complex systems, design defects can never be completely ruled out.

5.1.1 Types of testing

Following techniques have been followed for testing;

Black box testing

Internal system design is not considered in this type of testing. Tests are based on requirements and functionality.

White box testing

This testing is based on knowledge of the internal logic of an application’s code. Also known as Glass box Testing. Internal software and code working should be known for this type of testing. Tests are based on coverage of code statements, branches, paths, conditions.

Unit testing

Testing of individual software components or modules. Typically done by the programmer and not by testers, as it requires detailed knowledge of the internal program design and code.

Incremental integration testing

Bottom up approach for testing i.e. continuous testing of an application as new functionality is added; Application functionality and modules should be independent enough to test separately. Done by programmers or by testers.

Integration testing

Testing of integrated modules to verify combined functionality after integration.

Functional testing

This type of testing ignores the internal parts and focus on the output is as per requirement or not.

System testing

Entire system is tested as per the requirements. Black-box type testing that is based on overall requirements specifications, covers all combined parts of a system.

End-to-end testing

Similar to system testing, involves testing of a complete application environment in a situation that mimics real-world use, such as interacting with a database, using network communications, or interacting with other hardware, applications, or systems if appropriate.

Acceptance testing

Normally this type of testing is done to verify if system meets the customer specified requirements. User or customer does this testing to determine whether to accept application.

Usability testing

User-friendliness check. Application flow is tested, Can new user understand the application easily, Proper help documented whenever user stuck at any point. Basically system navigation is checked in this testing.

Install/uninstall testing

Tested for full, partial, or upgrade install/uninstall processes on different operating systems under different hardware, software environment.

Comparison testing

Comparison of product strengths and weaknesses with previous versions or other similar products.

Alpha testing

In house virtual user environment can be created for this type of testing. Testing is done at the end of development. Still minor design changes may be made as a result of such testing.

Beta testing

Testing typically done by end-users or others. Final testing before releasing application for commercial purpose.

5.2 Results

The proposed software is implemented as complete software using object oriented programming techniques with c-sharp (.net). This chapter presents the output generated by the software on various test cases selected to judge its performance.

5.2.1 TEST CASES

Test case id: 1

Unit to test: Calculate Result

Assumptions: The sample has been captured and saved in the required directory

Test data: Green values, Yellow values

Steps to be executed:

1. Open the application

1.1 Open a new batch and connect to the camera.

1.1.1 Capture green and yellow images of the sample.

1.1.2 Save the batch in a folder

1.1.3 Open view and analysis window.

1.1.4 Open the desired batch and set the reference

1.1.5 Now press the calculate button

1.2 Open view and analysis window.

1.2.1 Open existing folder saved in a directory

1.2.2 Press the button “Calculate”.

Expected result: 414

Actual result: 400

Pass/Fail: Pass

Comments: None

Test case id: 2

Unit to test: Open Batch

Assumptions: The sample has been captured and saved in the required directory

Test data: samples (or samples along with their values) stored in the database.

Steps to be executed:

1. Open the application.
2. Open view and analysis window.
3. Click “open folder” button.
4. Select required folder saved in a directory

Expected result: required batch

Actual result: required batch

Pass/Fail: Pass

Comments: None

Test case id: 3

Unit to test: Generate Report

Assumptions: The sample has been captured and saved in the required directory

Test data: sample id along with their values stored in the database.

Steps to be executed:

1. Open the application.
2. Select “view and analysis” view.
3. Click the open folder button
4. Select the batch whose report has to be generated.
5. Click “generate report” button.

Expected result: generation of report.

Actual result: generation of report.

Pass/Fail: Pass

Comments: None

Test case id: 4

Unit to test: Save Batch

Assumptions: The sample has been captured.

Test data: samples (or samples along with their values).

Steps to be executed:

1. Open the application.
2. Open a new batch and connect to the camera.
3. Capture green and yellow images of the sample.

4. Click save button.
5. Enter the batch name.
6. Click “ok”.

Expected result: batch saves in the database.

Actual result: batch saves in the database.

Pass/Fail: Pass

Comments: None

Test case id: 5

Unit to test: Capture DNA images

Assumptions: The application has been opened and hardware is connected

Test data: green and yellow images

Steps to be executed:

1. Open the application.
2. Connect the hardware.
3. Click “new batch” button.
4. Click “connect to the camera” button.

5. Click “capture image” button to create respective green and yellow images of the sample.

Expected result: green and yellow images.

Actual result: green and yellow images.

Pass/Fail: Pass

Comments: None

Test case id: 6

Unit to test: Print Report

Assumptions: The sample has been captured and saved in the required directory

Test data: sample id along with their values stored in the database.

Steps to be executed:

1. Open the application.
2. Select “view and analysis” view.
3. Click the open folder button
4. Select the batch whose report has to be generated.
5. Click “generate report” button.
6. Click “Print Report” button.

Expected result: Printout of the report generated.

Actual result: Printout of the report generated.

Pass/Fail: Pass

Comments: None

Chapter 5

Future Work

This project is basically performing end point analysis of PCR and is meeting all the requirements of the static end point PCR analysis. The next step in this project is to work on the Real Time PCR analysis. In real time PCR analysis the diagnosis, calculation and computation is done during the PCR and real time information is captured. If the real time machine can be developed locally with the local resources it will be a great achievement and then the pathology labs of our country can be self-dependent in terms of PCR equipment. As both these products will be locally manufactured its maintenance will be easy and also these products will be very cost effective.

Conclusion

DNA Real-Time PCR Analysis Software is a desktop application that does all processing in run time. Our aim was to develop a medical diagnostic utility within the available resources which will be used in pathology labs to help assist in the diagnosis of diseases like cancer, hepatitis A, B, C,HIV. No recognized software was available to carry out complete PCR tests from the start to the diagnosis phase. It performs quantitative as well as qualitative analysis on DNA images and generates a report specifying the presence of a certain virus in the patient's blood sample.

PCR permits early diagnosis of malignant diseases such as leukemia and lymphomas, which is currently the highest developed in cancer research and is already being used routinely. PCR also permits identification of non-cultivatable or slow-growing microorganisms such viruses. The basis for PCR diagnostic applications in microbiology is the detection of infectious agents and the discrimination of non-pathogenic from pathogenic strains by virtue of specific genes.

This product is completely developed locally which reflects the research of our one of the best pathologist and also the dynamic work of software engineering students. This collaborative also meets the standards in all respects.

As PCR deals with DNA and its structure so the device developed deals with medical images of billions of DNA and the diagnoses is done based on these images. This is a complete software kit and importance has been given to the user interface through the development lifecycle.