Rajajinagar Education Society's

SRI AUROBINDO COLLEGE

#6/C, V Main, Dr. M. C. Modi Road, Mahalakshmipuram, Bengaluru-560086

www.sriaurobindocollege.ac.in **080-23492449**

Permanently Affiliated to Bangalore University JB Campus & Accredited by NAAC with B+ Grade

DEPARTMENT OF COMPUTER SCIENCE

V SEMESTER BCA PROJECT REPORT

ON

BUG MANAGEMENT SYSTEM

SUBMITTED FOR THE V SEMESTER EXAMINATION OF BANGALORE UNIVERSITY PRACTICAL EXAMINATIONS

MARCH-2022

Submitted by

GOWTHAM H S **19UCSB7005**

Rajajinagar Education Society's

SRI AUROBINDO COLLEGE

#6/C, V Main, Dr. M. C. Modi Road, Mahalakshmipuram, Bengaluru-560086

www.sriaurobindocollege.ac.in 080-23492449

Permanently Affiliated to Bangalore University JB Campus & Accredited by NAAC with B+ Grade

DEPARTMENT OF COMPUTER SCIENCE

V SEMESTER BCA PROJECT REPORT

CERTIFICATE

This is to certify that *Mr.Gowtham H S* has satisfactorily completed the project on BUG MANAGEMENT SYSTEM using VB.NET Technology and MySQL prescribed by the Bangalore University for the V-semester BCA course 2021-2022.

External Examiner **Internal Examiner**

Principal **Head of the Department**

DECLARATION

I, *Mr.Gowtham H S* hereby declare that the project entitled "**BUG MANAGEMENT SYSTEM**" is bona fide work carried out by us under the guidance of Sushmitha V S. This project, as presented in this report, is our original work and has not been presented for any other University award. This project has been submitted as partial fulfillment of requirements for the degree of Bachelor of Computer Application of Bangalore University.

Project associate:

GOWTHAM H S [19UCSB7005]

ACKNOWLEDGEMENT

Before introducing our work, we would like to thank the people without whom the success of this project would have been only a dream. We express our deep sense of gratitude and indebtedness to ***HOD of BCA Department Mr. Vinod Annigere*** & the project guide ***Ms.Sushmitha V S*** for their constructive information and useful suggestion which helped us to complete our project successfully on time. We thank our project incharge ***Mr. Vinod Annigere*** for her great support and guidance till completion of project. It is with great pleasure; we extend our gratitude and thanks to our ***Principal Dr. Shailaja*** for constant support and encouragement. We would express our gratitude to ***Mrs. Megha*** , ***Mrs. Anuradha*** and ***Mrs. Divya***, Lecturers of BCA Department for their kind support and guidance to carry out this project. We would like to thank the members of teaching and non-teaching staff of our department for their constant support & co-operation. We feel short of words to express our heartfelt thanks to all our family members and friends and all those who have directly or indirectly helped us to complete our project on time.

BUG MANAGEMENT SYSTEM

ABSTRACT

Bug-Management mechanism is employed only is some of the large software development houses. Most of the others never bothered with bug Management at all, and instead simply relied on shared lists and email to monitor the status of defects. This procedure is error-prone and tends to cause those bugs judged least significant by developers to be dropped or ignored.

Bug-Management System is an ideal solution to track the bugs of a product, solution or an application. Bug Management System allows individual or groups of developers to keep track of outstanding bugs in their product effectively. This can also be called as Defect Management System.

The Bug Management System can dramatically increase the productivity and accountability of individual employees by providing a documented work flow and positive feedback for good performance.

Features:

- Product and Component based
- Creating & Changing Bugs at ease
- Query Bug List to any depth
- Reporting & Charting in more comprehensive way
- User Accounts to control the access and maintain security
- Simple Status & Resolutions
- Multi-level Priorities & Severities
- Targets & Milestones for guiding the programmers
- Attachments & Additional Comments for more information
- Robust database back-end

Table of Contents

List of Figures

List of Photos

CHAPTER 1

1. Introduction

1.1 Purpose

The main objective of this system is develop flawless system, which is access real time information from anywhere in the world, 24 hours a day 365 days in a year. Another aim is that manage hundred of projects in multiple locations or just a few. The another main objective of this system is track the all the defects or bugs in the project and make the project user friendly and bugs free system.

1.2 Existing System

In any software development bugs are inevitable. Let it be in any kind of product bugs arise at any phase of development. One has to take a great care in the proper maintenance and resolution of the bugs. In the Existing system the bugs are not properly maintained and they are simply relied on shared lists and email to monitor the bugs.

In this type of system it becomes difficult to track a bug if a bug is over looked then it may cause tremendous errors in the next phase and can improve the cost of project whatever necessary effort spent on the bug maintenance may not be worthy. So bug history has to be maintained properly. And there is no efficient search technique.

One has to search the whole database for the details of particular bug which might have occurred sometime earlier. It is both time consuming and error prone. And it is very difficult to share the bug among several users as there is no proper maintenance of the bugs. In order to have an efficient product bugs must be maintained properly and should be resolved in time both to reduce time and money spent on the development.

1.3 Proposed System

- This system maintains the products, Bugs and bug Management. It has advantage of maintaining bug history it stores all the details from bug origin to bug resolution.

- Each product can have versions for easy maintenance of the product and all the user of the product is stored in the database. It provides the advantage of maintaining users to the bugs and resolutions provided by them.
- Our System provides the searching based on status, priority, and operating system.
- It provides with user and bug hierarchy, which would be helpful in knowing the relation between bugs and users allotted to the bug.
- It is provided with a fully authenticated system with password encryption. And has the facility for storing attachments for a bug.
- One can keep a track of the bug in a product with much lower cost and effort.
- The most advantage of this system is maintaining log records which are helpful in knowing any errors or misuse of the system by other users.

CHAPTER 2

2. Software and Hardware Requirements

2.1 Software Requirements

A set of programs associated with the operation of a computer is called software. Software is the part of the computer system which enables the user to interact with several physical hardware devices.

The minimum software requirement specifications for developing this project are as follows:

- Windows XP/ 10 operating system.
- Microsoft Office package.
- Microsoft Visual Basic 10 express

2.2 Hardware Requirement Specification

The Collection of internal electronic circuits and external physical devices used in building a computer is called Hardware.

The minimum hardware requirement specification for developing this project is as follows:

- Computer with either Intel Pentium processor or AMD processor.
- 128MB DDR RAM
- 40GB hard disk drive

CHAPTER 3

3. Literature Survey

Bug Management System is an ideal solution to track the bugs of a product, solution or an application. Bug Management System allows individual or groups of developers to keep track of outstanding bugs in their product effectively. This can also be called as Defect Management System.

The Bug Management System can dramatically increase the productivity and accountability of individual employees by providing a documented work flow and positive feedback for good performance.

For many years, bug-Management mechanism is employed only in some of the large software development houses. Most of the others never bothered with bug Management at all, and instead simply relied on shared lists and email to monitor the status of defects. This procedure is error-prone and tends to cause those bugs judged least significant by developers to be dropped or ignored.

In any software development bugs are inevitable. Let it be any kind of product bugs arise to any phase of development. One has to take a great care in proper maintenance and resolution of the bugs. In the Existing system the bugs are not properly maintained and they are simply relied on shared lists and email to monitor the bugs. In this type of system it becomes difficult to track a bug if a bug is over looked then it may be cause tremendous errors in the next phase and can improve the cost of project what ever necessary effort spent on the bug maintenance may not be worthy. So bug history has to be maintained properly. And there is no efficient search technique. One has to search the whole database for the details of particular bug which might have occurred some time earlier. It is both time consuming and error prone. And it is very difficult to share the bug among several users as there is no proper maintenance of the bugs.

In order to have an efficient product bugs must be maintained properly and should be resolved in time both to reduce time and money spent on the development.

CHAPTER 4

4. Software Requirements Analysis

4.1 Overview

The main focus of the analysis phase of Software development is on "What needs to be done". The objects discovered during the analysis can serve as the framework or Design. The class's attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution.

4.2 Problem Description

One has to take a great care in the proper maintenance and resolution of the bugs. In the Existing system the bugs are not properly maintained and they are simply relied on shared lists and email to monitor the bugs.

In this type of system it becomes difficult to track a bug if a bug is over looked then it may cause tremendous errors in the next phase and can improve the cost of project whatever necessary effort spent on the bug maintenance may not be worthy. So bug history has to be maintained properly. And there is no efficient search technique.

One has to search the whole database for the details of particular bug which might have occurred sometime earlier. It is both time consuming and error prone. And it is very difficult to share the bug among several users as there is no proper maintenance of the bugs. In order to have an efficient product bugs must be maintained properly and should be resolved in time both to reduce time and money spent on the development.

4.3 Solution

This system maintains the products, Bugs and bug Management . It has advantage of

maintaining bug history it stores all the details from bug origin to bug resolution. Each product can have versions for easy maintenance of the product and all the user of the product is stored in the database. It provides the advantage of maintaining users to the bugs and resolutions provided by them. Our System provides the searching based on status, priority, and operating system. It provides with user and bug hierarchy, which would be helpful in knowing the relation between bugs and users allotted to the bug. It is provided with a fully authenticated system with password encryption. And has the facility for storing attachments for a bug. One can keep a track of the bug in a product with much lower cost and effort. The most advantage of this system is maintaining log records which are helpful in knowing any errors or misuse of the system by other users.

4.4 Modules

1. Developers

2. Manager

3. Team Lead

4.5 Module Description

Developer

Employees are of two types, developers and testers. Developers are used to develop program and open bugs where as testers resolve the bugs and save to the database.

Manager

Manager takes care of recruitment of employees and the management of employees in the project and monitors the completion of project

Team Lead

Team Lead is a person who will take care of all registration status, acceptance of new bugs, and many other tasks to reduce burden on employee.

CHAPTER 5

5. Software Design

The main focus of the analysis phase of Software development is on "What needs to be done". The objects discovered during the analysis can serve as the framework or Design. The class's attributes, methods and association identified during analysis must be designed for implementation language. New classes must be introduced to store intermediate results during the program execution.

Emphasis shifts from the application domain of implementation and computer such as user interfaces or view layer and access layer. During analysis, we look at the physical entities or business objects in the system, that is, which players and how they cooperate to do the work of the application. These objects represent tangible elements of the business.

During the Design phase, we elevate the model into logical entities, some of which might relate more to the computer domain as people or employees. Here his goal is to design the classes that we need to implement the system the difference is that, at this level we focus on the view and access classes, such as how to maintain information or the best way o interact with a user or present information.

Design process:

During the design phase the classes identified in object-oriented analysis Must be revisited with a shift focus to their implementation. New classes or attribute and Methods must be an added for implementation purposes and user interfaces.

The following are some of the vies of software design life cycle. They are

- Data Flow Diagrams
- UML Diagrams
- Data Base Design

5.1 Data Flow Diagram

A Data Flow Diagram (DFD) is a graphical representation of the "flow" of data through an information system. It can also be used for the visualization of data processing (structured design).

There are two types of DFDs. They are:

- Context Level DFD
- Top Level DFD

5.1.1 Context Level DFD

In the Context Level the whole system is shown as a single process.

- No data stores are shown.
- Inputs to the overall system are shown together with data sources (as External entities).
- Outputs from the overall system are shown together with their destinations (as External entities).

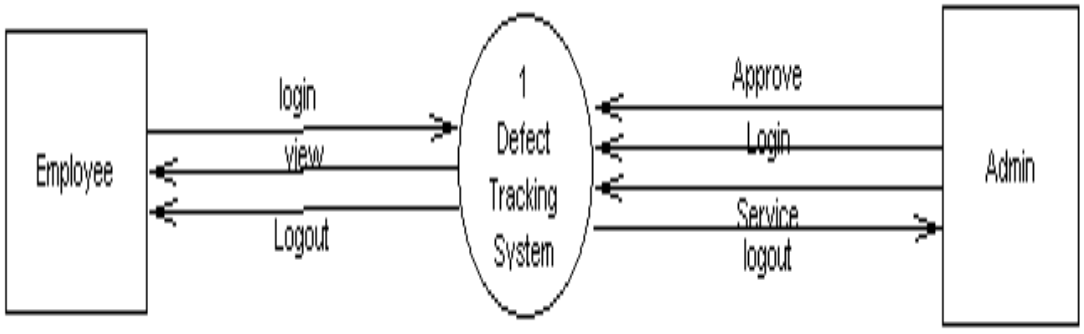


Fig. 5.1 Context Level DFD

5.1.2 Top Level DFD

The Top Level DFD gives the overview of the whole system identifying the major system processes and data flow. This level focuses on the single process that is drawn in the context diagram by 'Zooming in' on its contents and illustrates what it does in more detail.

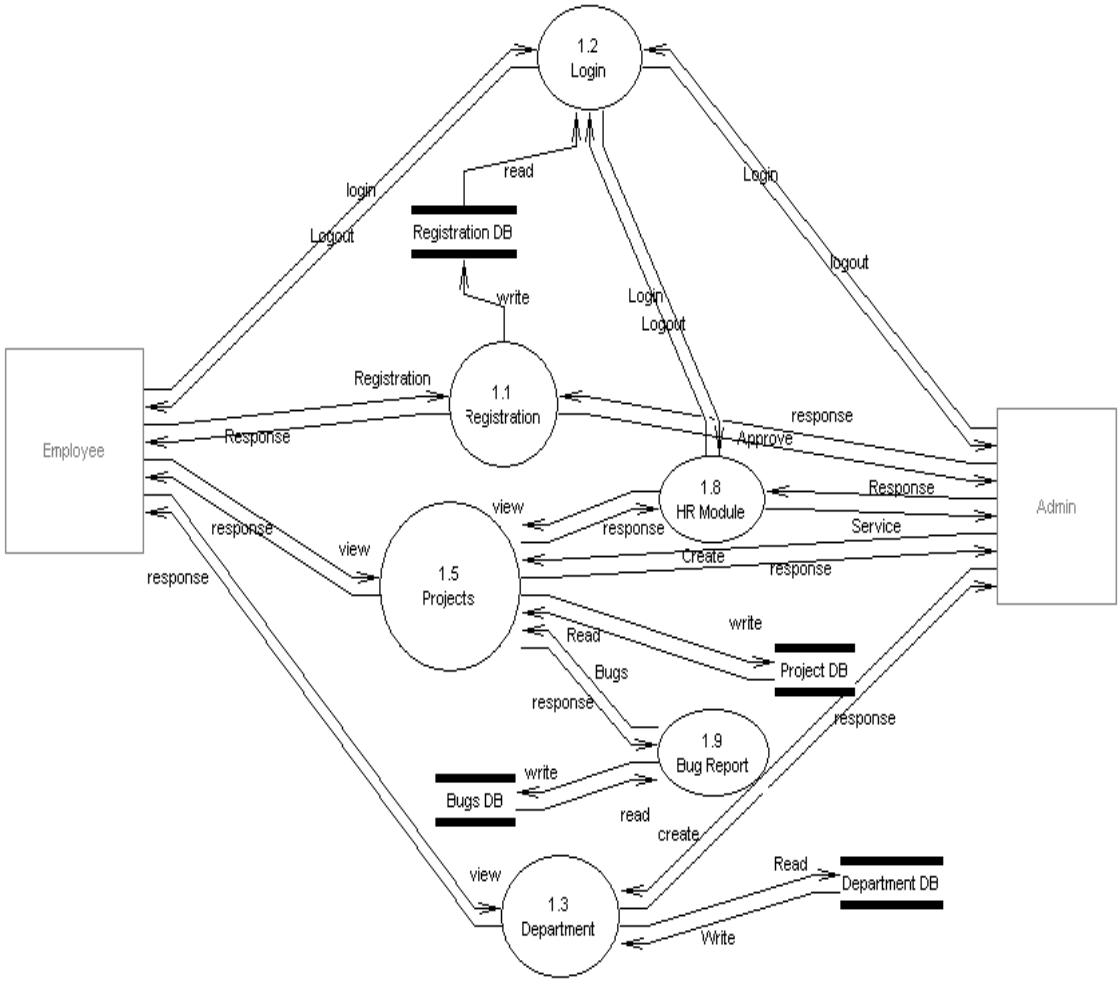


Fig. 5.2 Top Level DFD

5.2 UML Diagrams

Unified Modeling Language

The Unified Modeling Language allows the software engineer to express an analysis model using the modeling notation that is governed by a set of syntactic semantic and pragmatic rules.

This UML diagrams must include the following:

- Class diagram
- Interaction Diagram
- Use case Diagram
- Activity Diagram
- Component Diagram
- Deployment Diagram

Class Diagrams

The class diagram is the main building block in object oriented modeling. They are being used both for general conceptual modeling of the systematic of the application, and for detailed modeling translating the models into programming code.

The classes in a class diagram represent both the main objects and or interactions in the application and the objects to be programmed. In the class diagram these classes are represented with boxes which contain three parts:

- The upper part holds the name of the class
- The middle part contains the attributes of the class, and
- The bottom part gives the methods or operations the class can take or undertake

An Activity Diagram shows the flow from activity to activity.
An activity is an ongoing non- atomic execution within a state machine.

5.2.1Class Diagrams

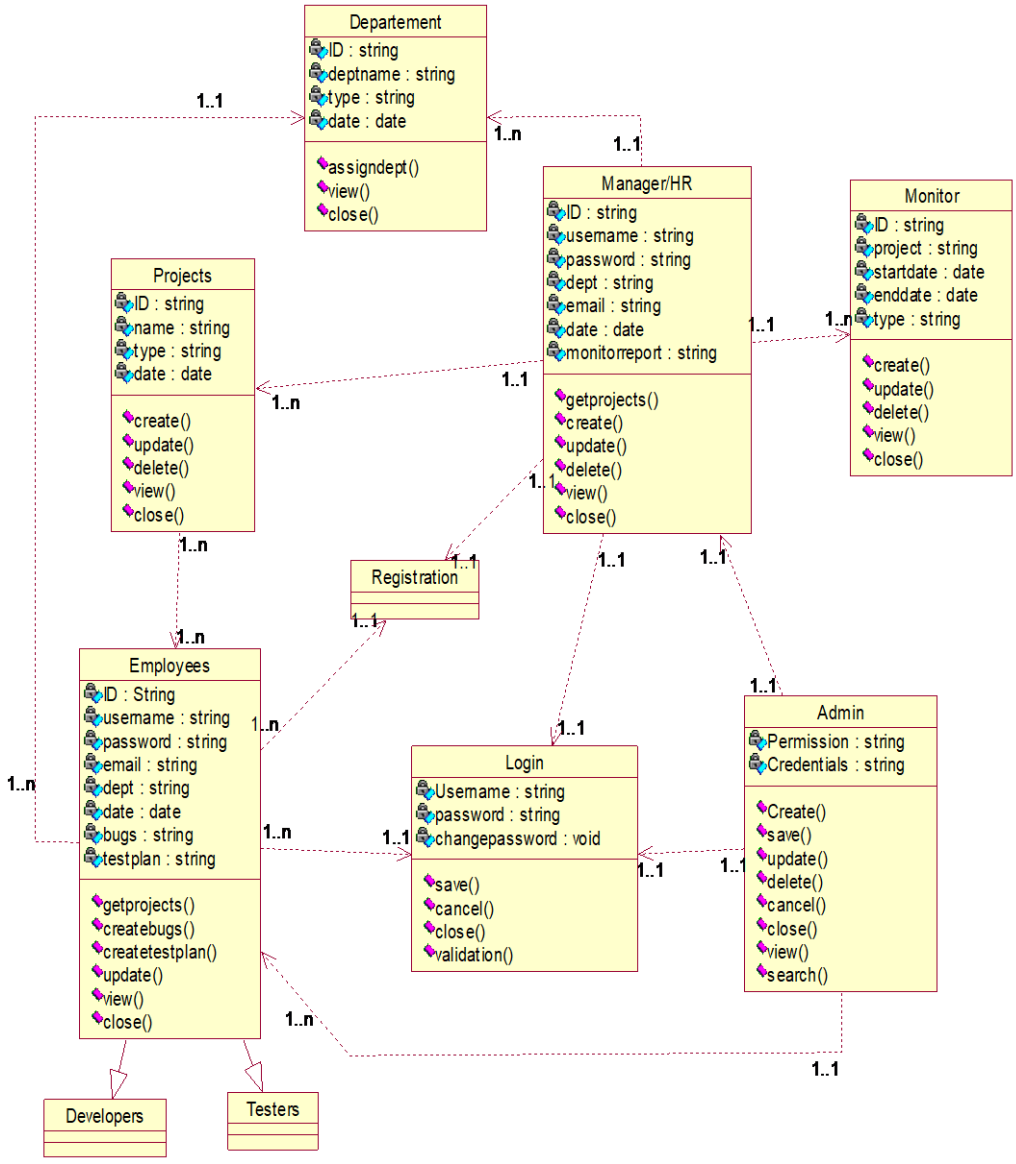


Fig 5.3 Class Diagram

5.2.2 Interaction Diagram

Interaction Diagrams

An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.

A sequence diagram is an interaction diagram that emphasizes the time ordering of messages. Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.

A Collaboration is a society of classes, interfaces, and other elements that work together to provide some cooperative behavior that's bigger than the sum of all its parts.

5.2.2.1 Sequence Diagram

- An interaction diagram shows an interaction, consisting of a set of objects and their relationships, including the messages that may be dispatched among them.
- A sequence diagram is an interaction diagram that emphasizes the time ordering of messages.
- Graphically, a sequence diagram is a table that shows objects arranged along x-axis and messages, ordered in increasing time, along the y-axis.

Contents

- Sequence diagrams commonly contain the following:
    - Objects
    - Links
    - Messages

Like all other diagrams, sequence diagrams may contain notes and constrains.

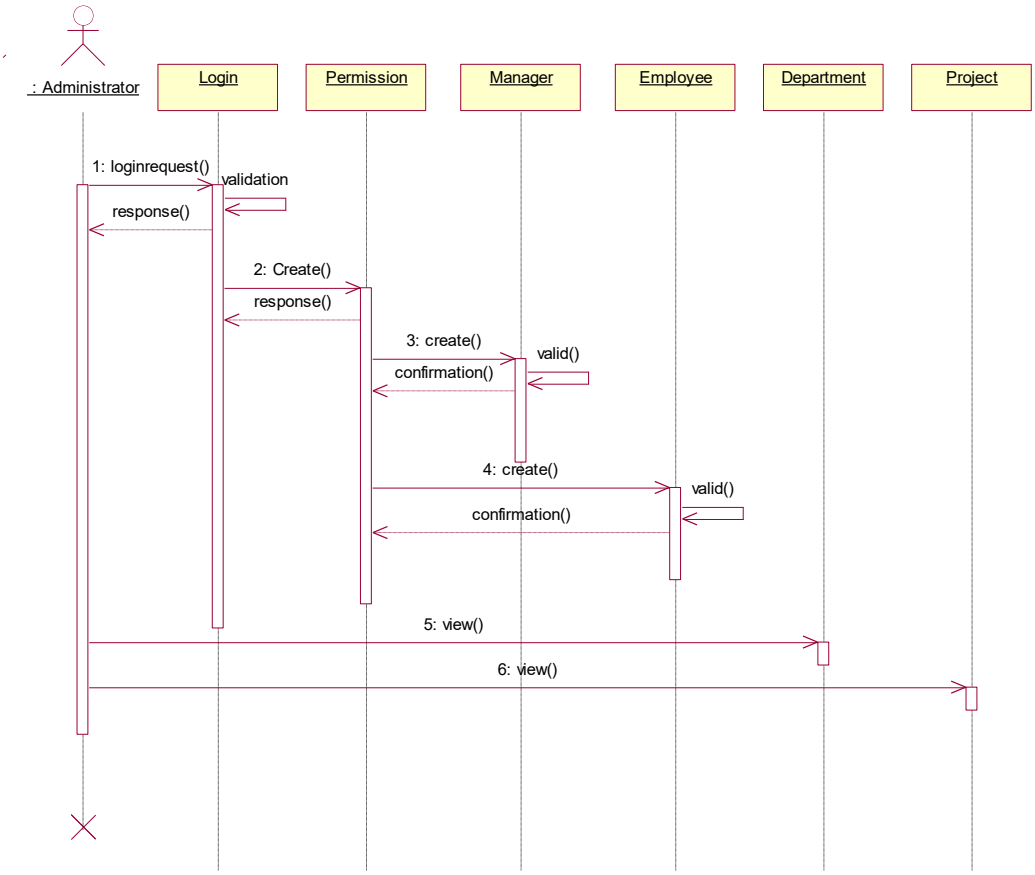


Fig 5.4 Administration Sequence Diagram

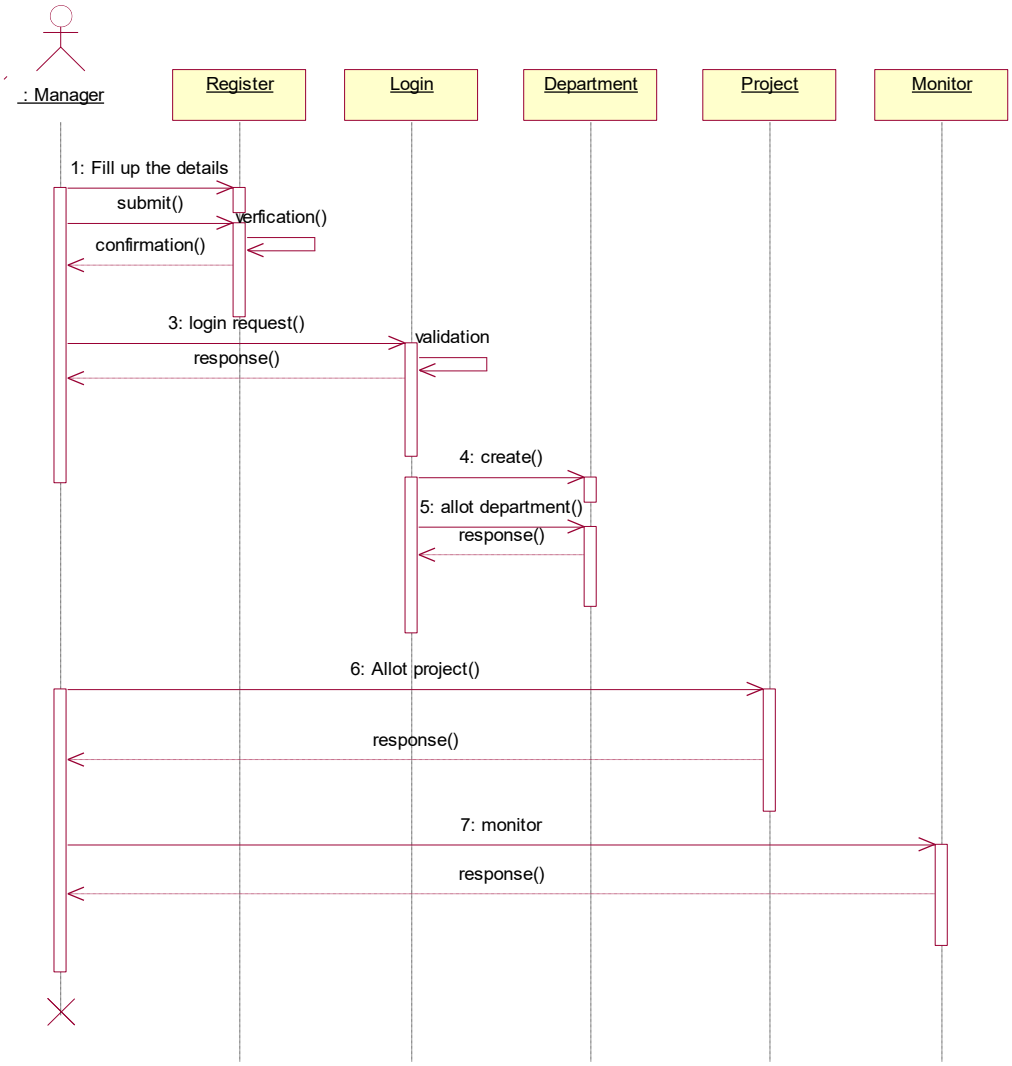


Fig 5.5 Manager Sequence

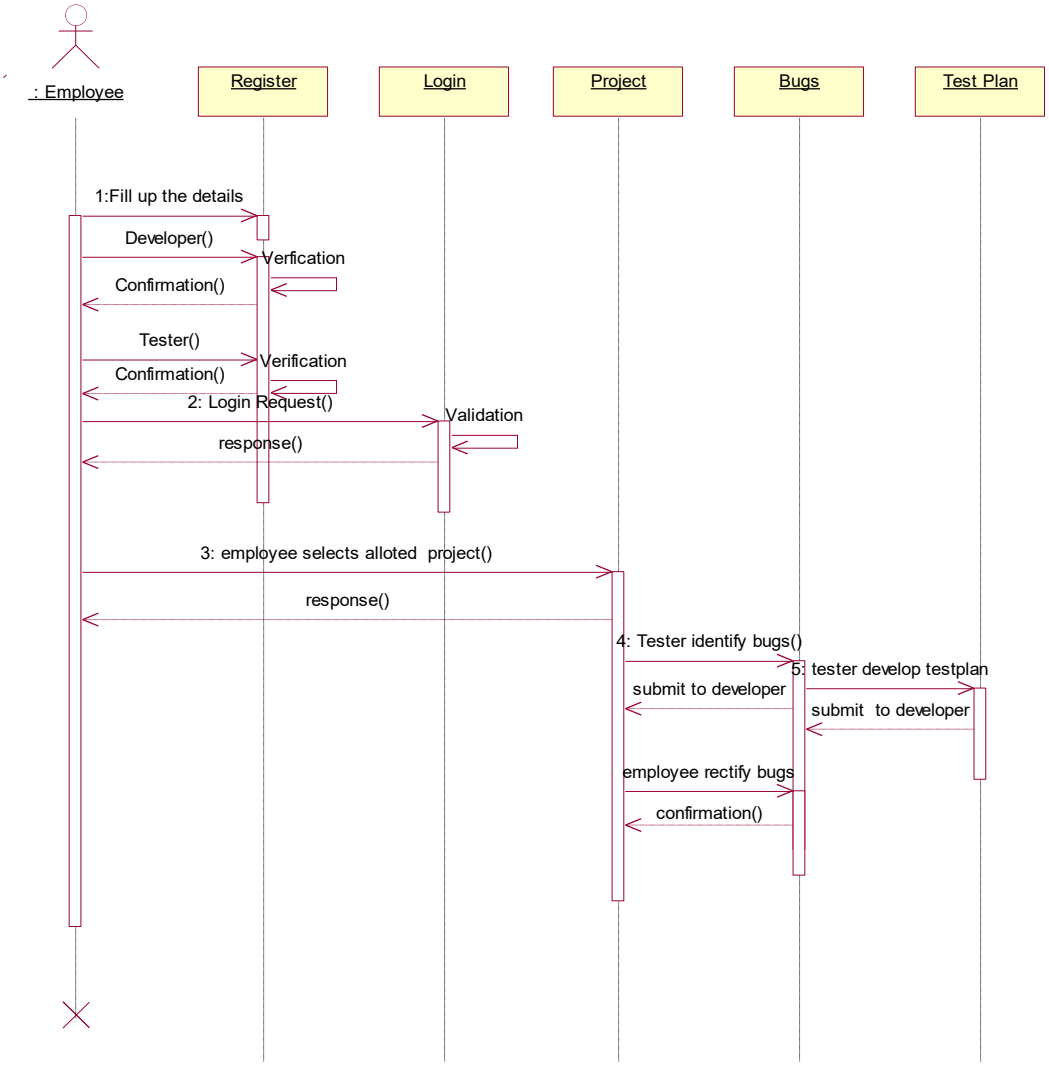


Fig 5.6 Employee Sequence

5.2.2.2 Collaborations Diagram

Collaboration is a society of classes, interfaces, and other elements that work together to provide some cooperative behavior that's bigger than the sum of all its parts.

Collaboration is also the specification of how an element, such as a classifier or an operation, is realized by a set of classifiers and associations playing specific roles used in a specific way

Contents

Collaboration diagrams commonly contain the following:

- Objects
- Links
- Messages

Like all other diagrams, sequence diagrams may contain notes and constrains.

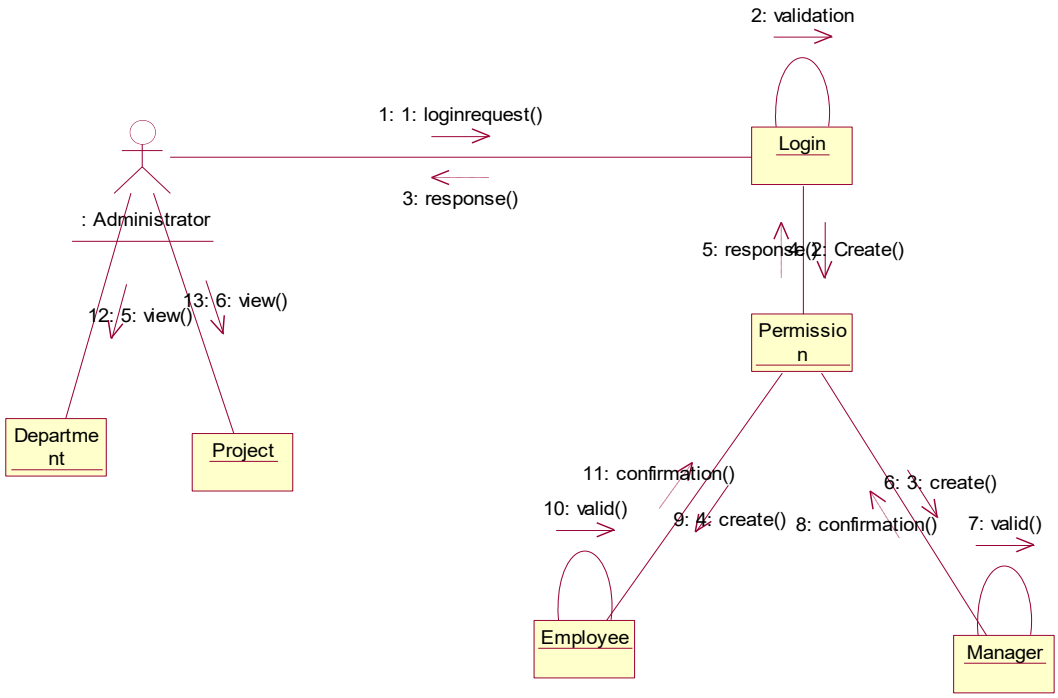


Fig 5.7 Administrative Collaboration

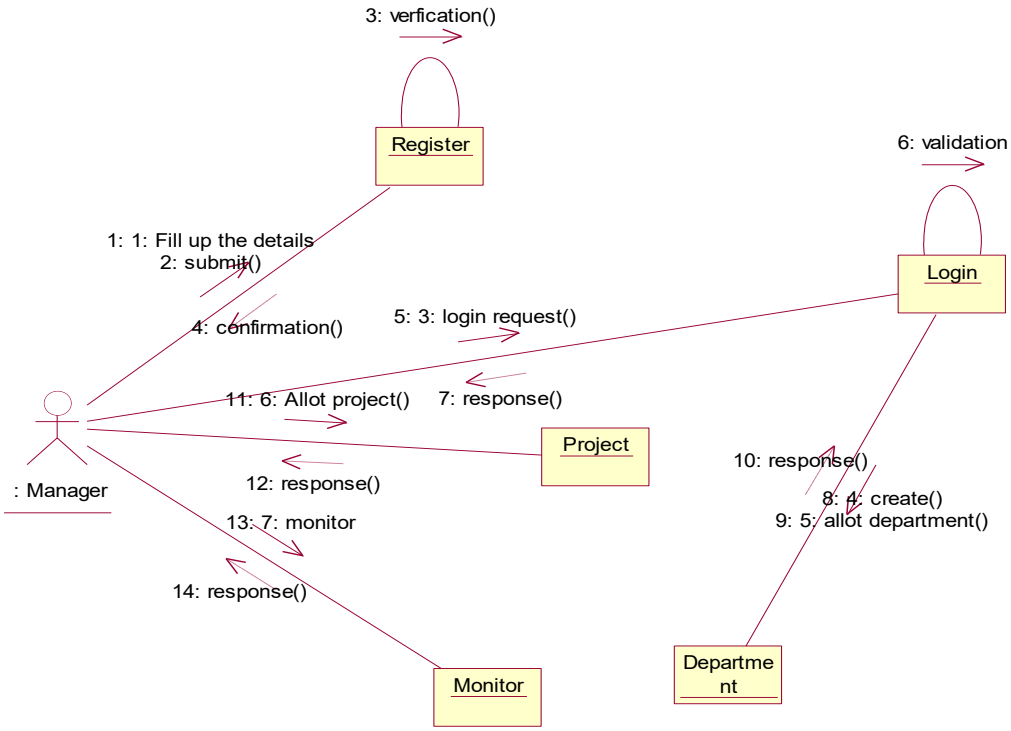


Fig 5.8 Manager Collaboration

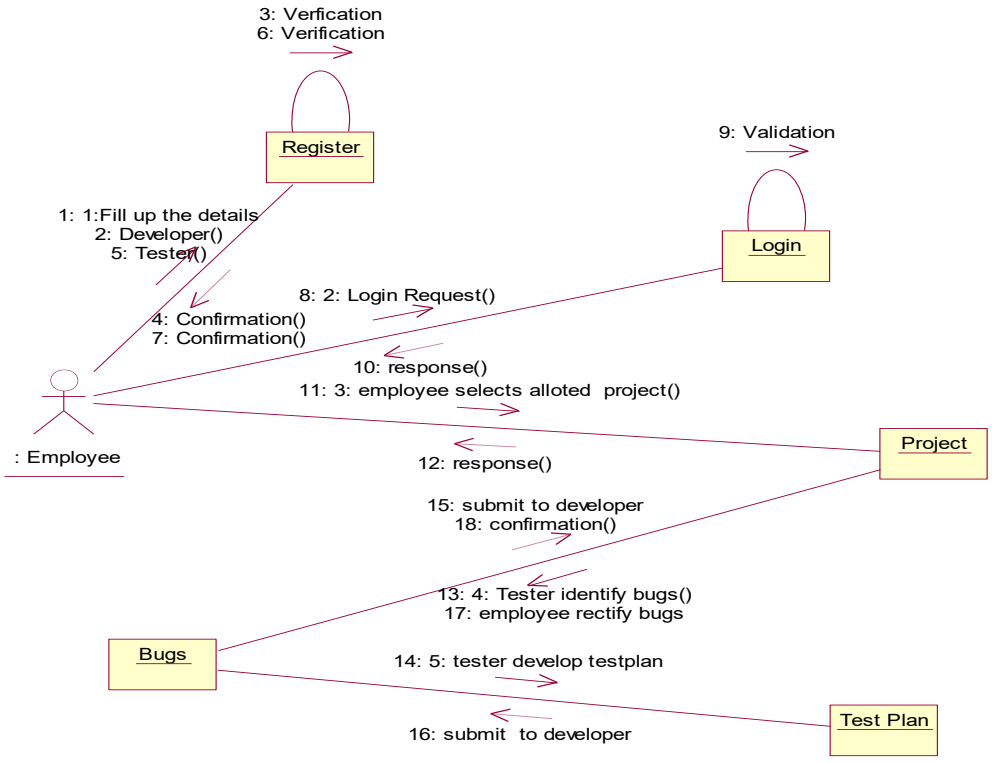


Fig 5.9 Employee collaboration

5.2.3 Use case Diagram

A use case diagram is a diagram that shows a set of use cases and actors and relationships.

Use case Diagrams represent the functionality of the system from a user's point of view. Use cases are used during requirements elicitation and analysis to represent the functionality of the system .Use cases focus on the behavior of the system from external point of view.

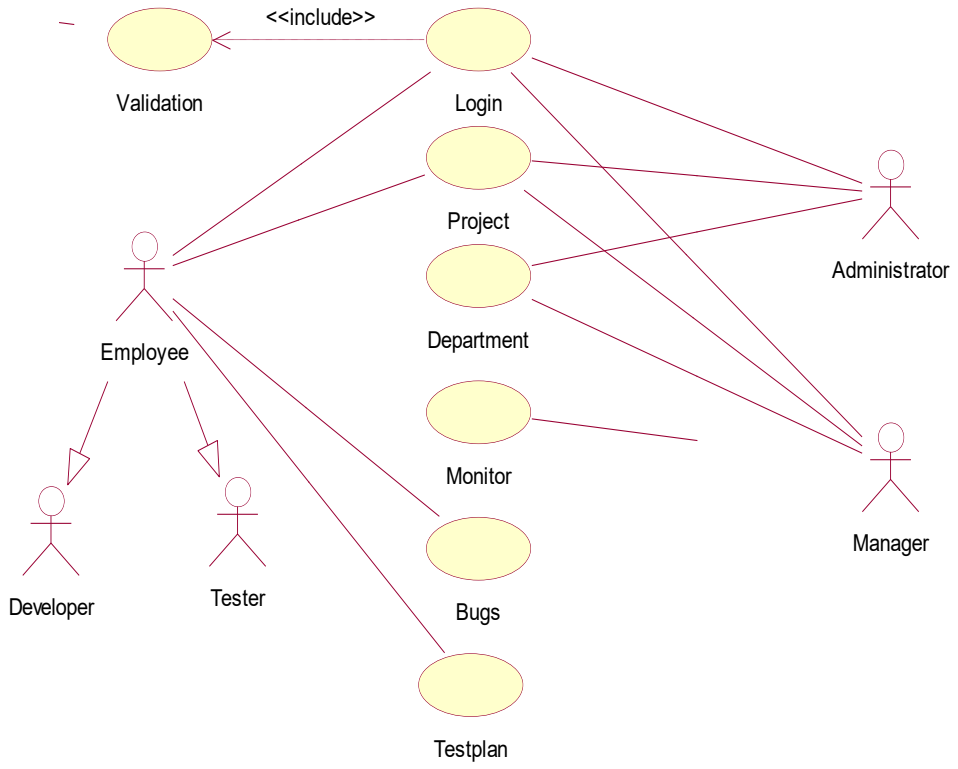


Fig 5.10 Overall Use case Diagram

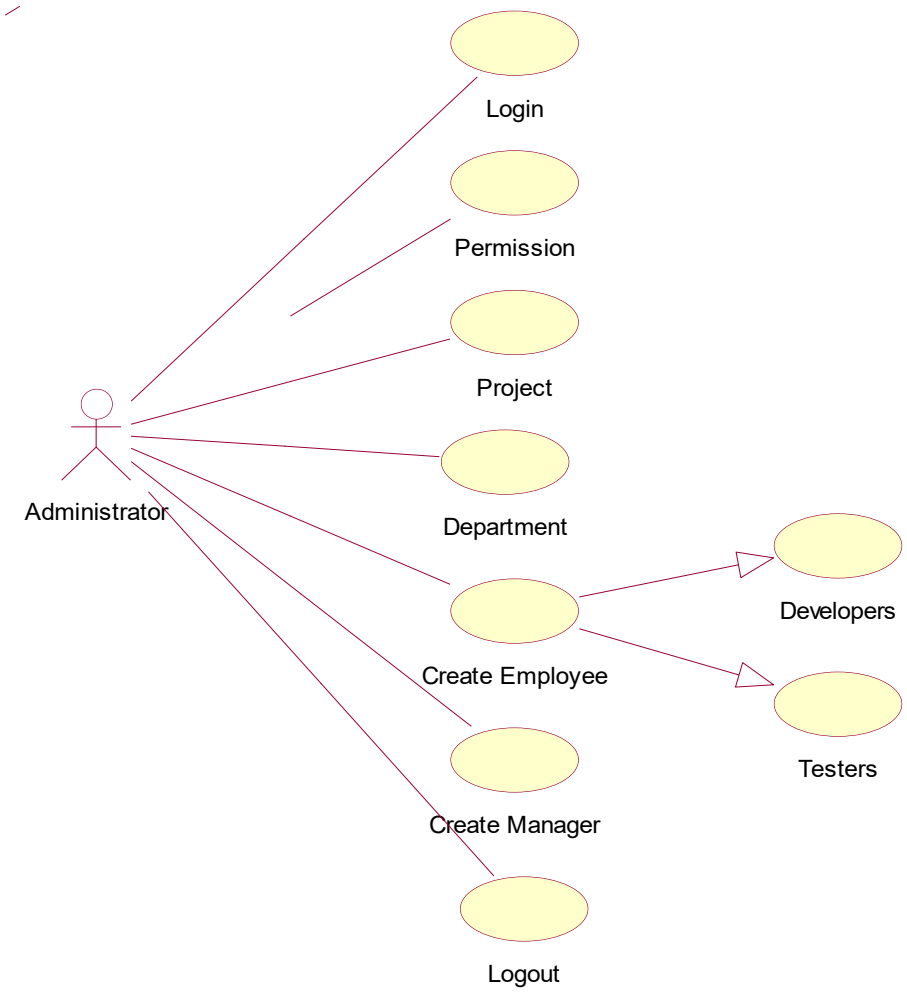


Fig 5.11 Administrator Use case

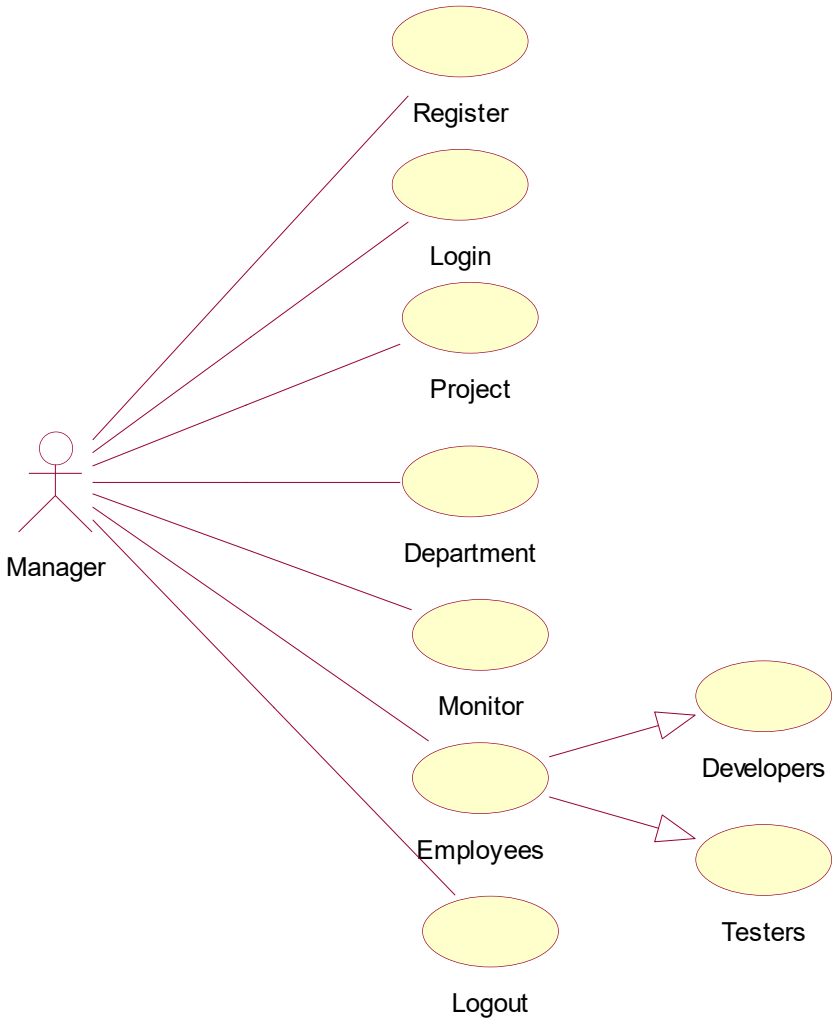


Fig 5.12 Manager Use case

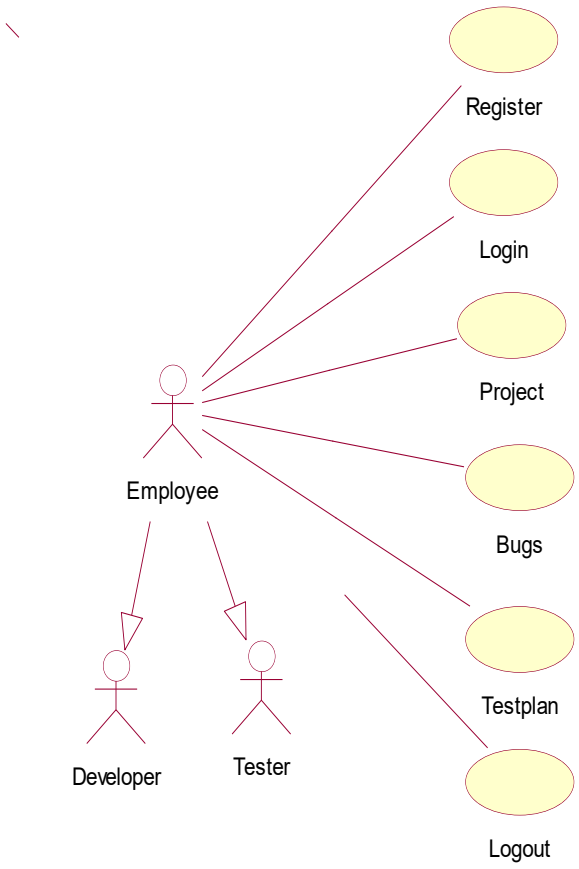


Fig 5.13 Employee use case

5.2.4 Component Diagram

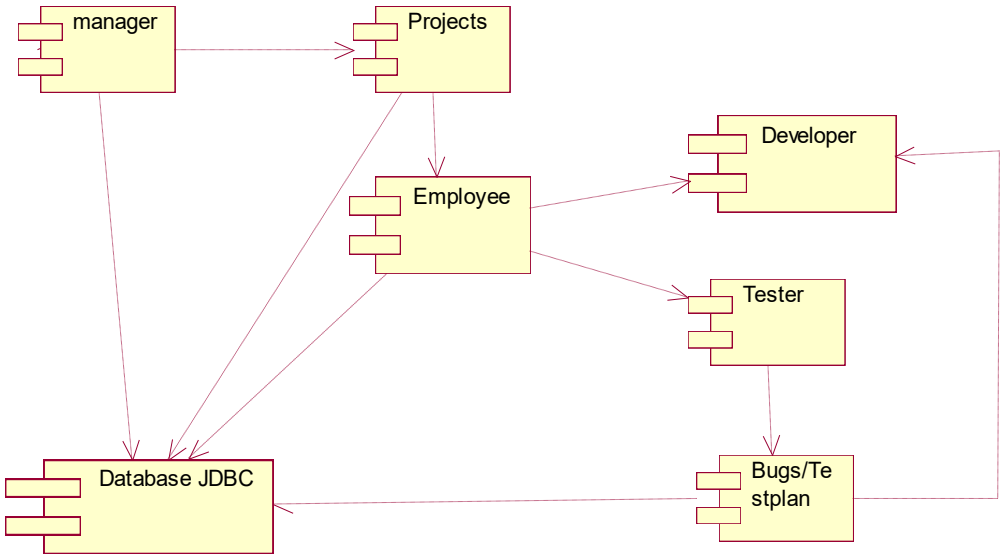


Fig 5.14 Component Diagram

5.2.5 Deployment Diagram

A deployment diagram is a diagram that shows the configuration of run time processing nodes and the components that live on them.

Graphically, a deployment diagram is collection of vertices and arcs.

Contents

- Deployment diagram commonly contain the following things:
- Nodes
- Dependency and association relationships
- Like all other diagrams, deployment diagrams may contain notes and constraints.
- Deployment diagrams may also contain components, each of which must live on some node.

- Deployment diagrams may also contain packages or subsystems, both of which are used to group elements of your model into larger chunks.

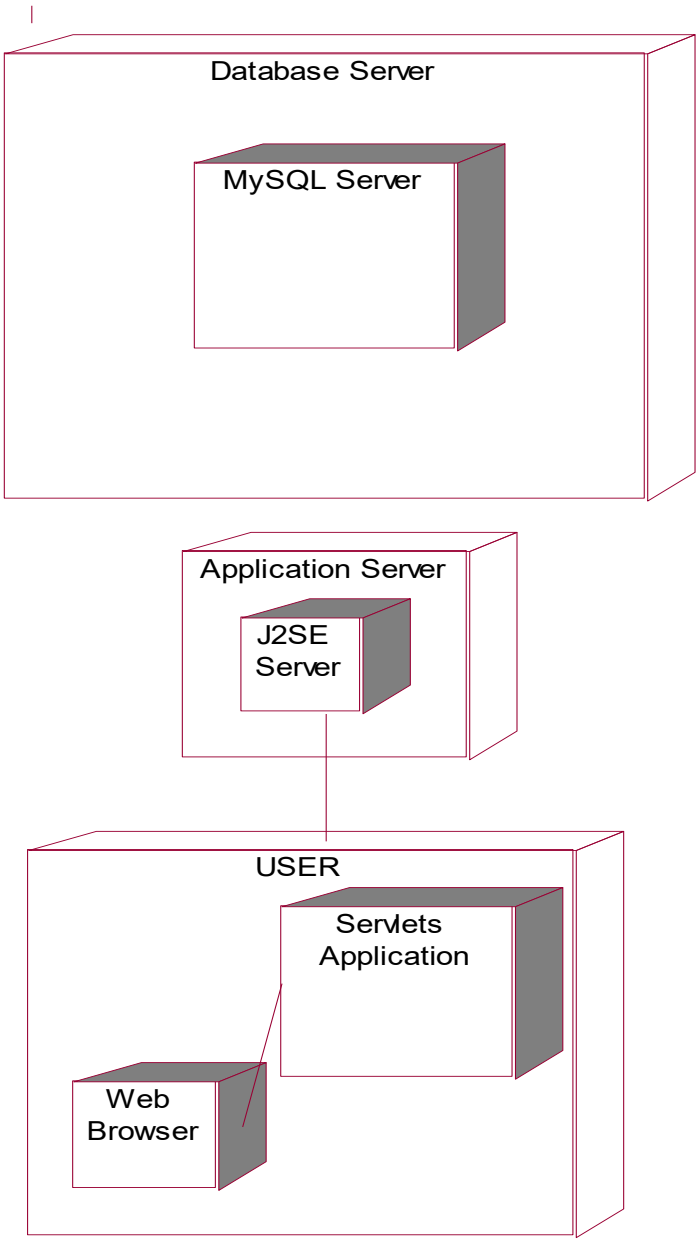


Fig 5.15 Deployment Diagram

5.3 Control Flow diagrams

5.3.1 Activity Diagram

- An activity diagram shows the flow from activity to activity. An activity is an ongoing non-atomic execution within a state machine.
- Activities ultimately result in some action, which is made up of executable atomic computations that result in a change in state of the system or the return of a value.

Activity diagrams commonly contain

- Activity states and action states
- Transitions
- Objects

Like all other diagrams, activity diagrams may contain notes and constrains

Login Process

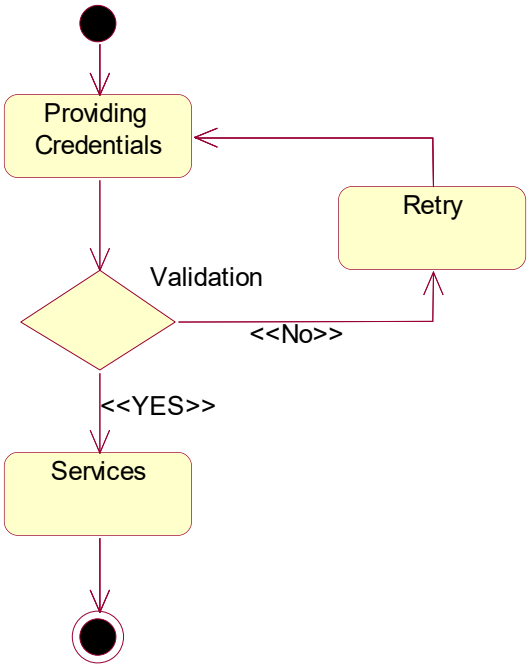


Fig 5.16 Login Activity Diagram

Registration Process

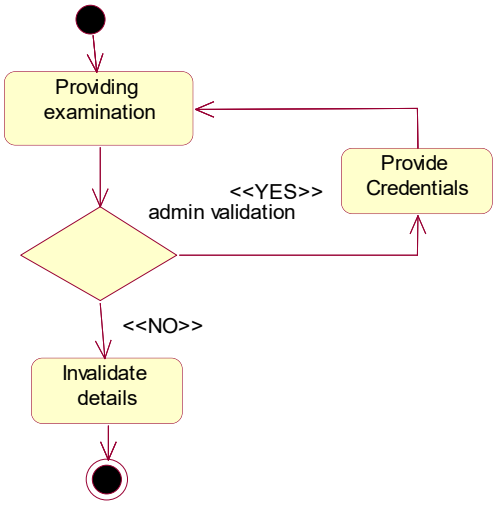


Fig 5.17 Registration Activity Diagram

Manager Process

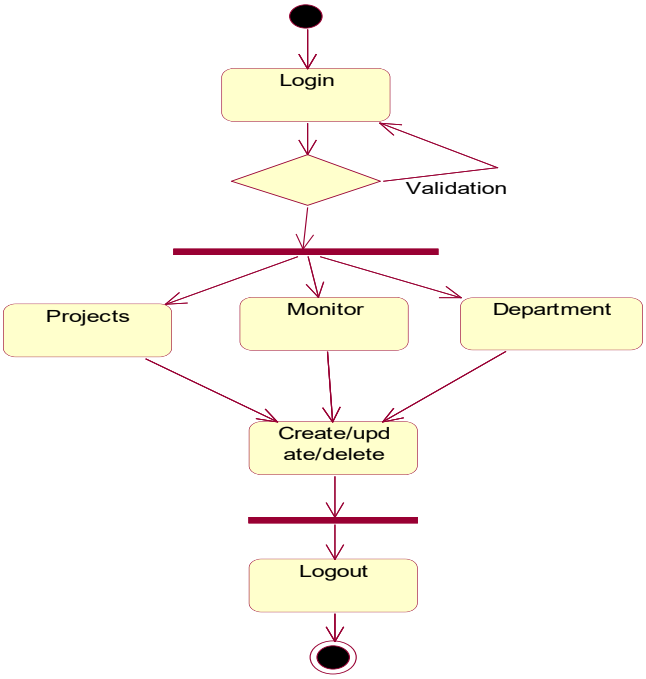


Fig 5.18 Manager Activity Diagram

Administrator Process

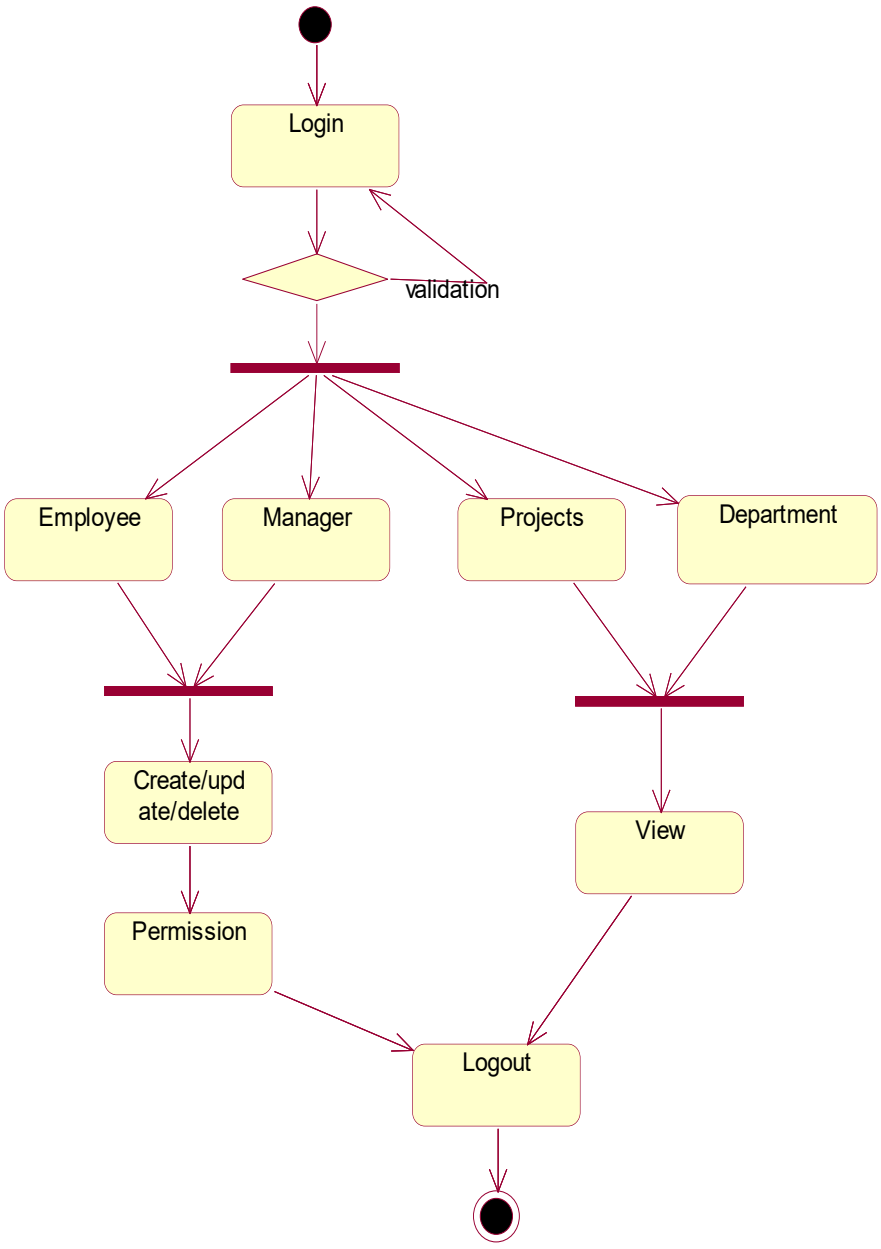


Fig 5.19 Administrator Process

Employee Process

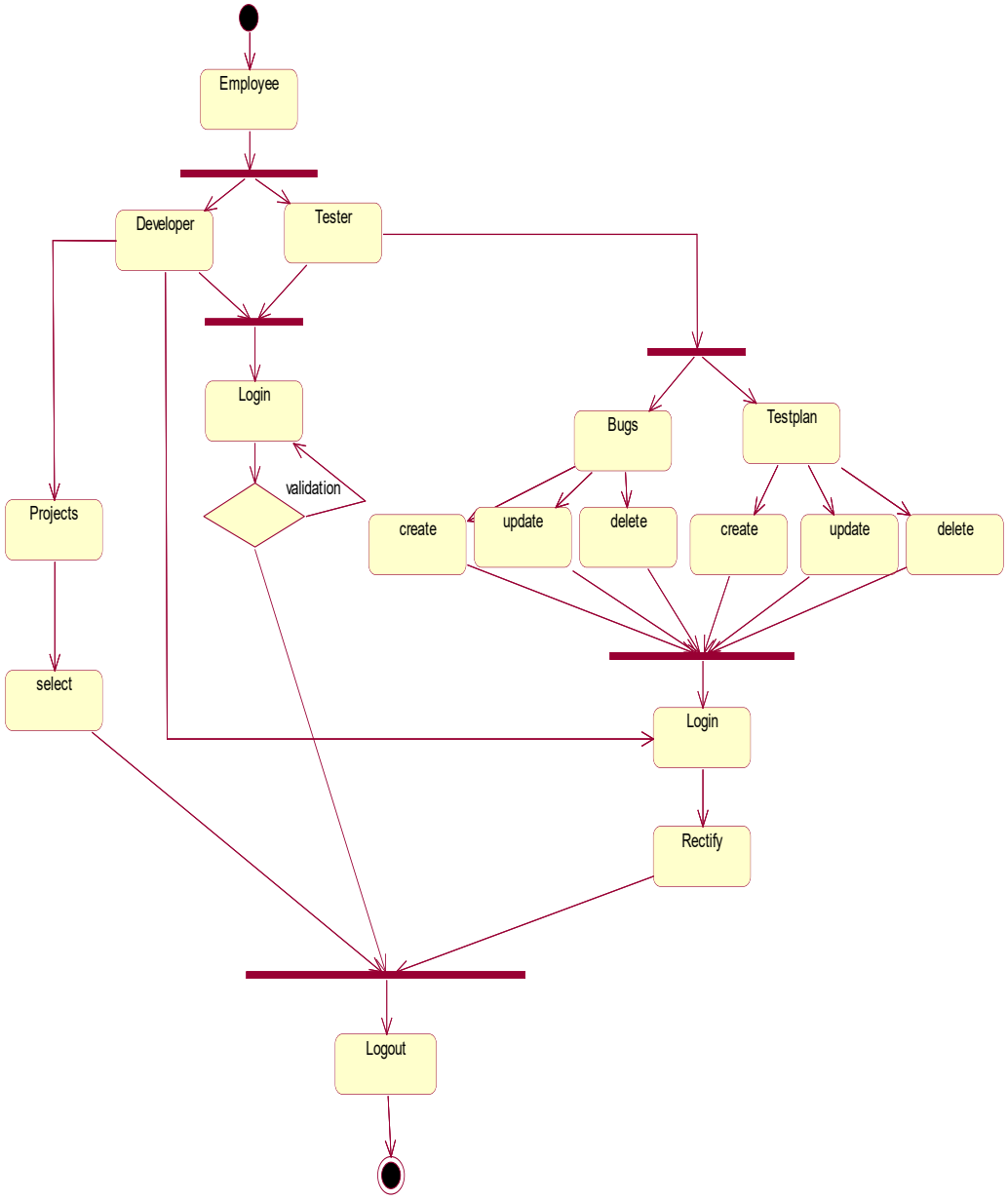


Fig 5.20 Employee Process

5.4 Database Design

Database design is the process of producing a detailed data model of a database. This logical data model contains all the needed logical and physical design choices and physical storage parameters needed to generate a design in a Data Definition Language, which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design can be used to describe many different parts of the design of an overall database system. Principally, and most correctly, it can be thought of as the logical design of the base data structures used to store the data. In the relational model these are the tables and views.

In an object database the entities and relationships map directly to object classes and named relationships. However, the term database design could also be used to apply to the overall process of designing, not just the base data structures, but also the forms and queries used as part of the overall database application within the database management system (DBMS).

5.4.1 ER-Diagrams

Entity Relationship Diagrams (ERDs) illustrate the logical structure of databases. An entity-relationship (ER) diagram is a specialized graphic that illustrates the interrelationships between entities in a database.

ER diagrams often use symbols to represent three different types of information. Boxes are commonly used to represent entities. Diamonds are normally used to represent relationships and ovals are used to represent attributes.

Entity relationship diagrams are a way to represent the structure and layout of a database. It is used frequently to describe the database schema. ER diagrams are very useful as provide a good conceptual view of any database, regardless of the underlying hardware and software.

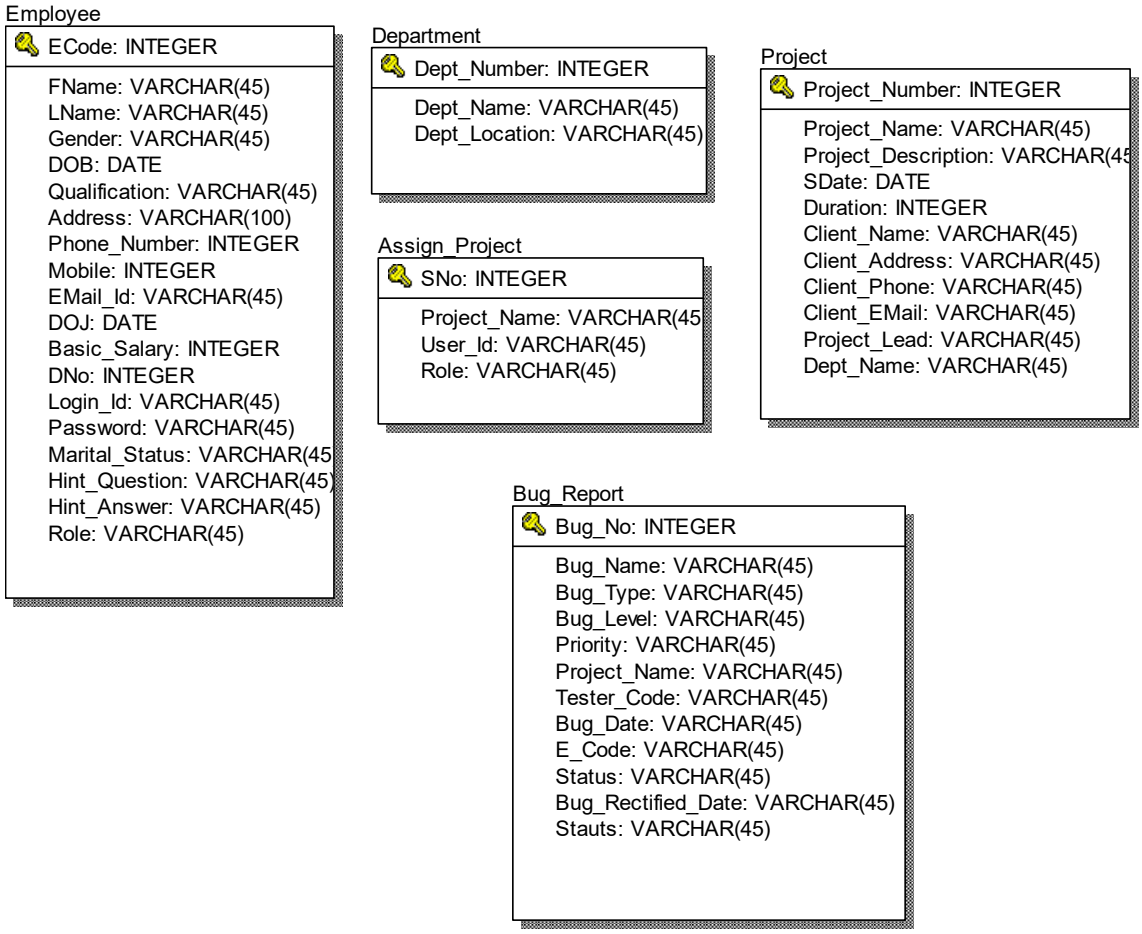


Fig 5.21 ER-Diagram

I. .

Tables

Manager Login:

mloginTab: Query...IN\DEBUG\BUG.MDF) | AffixNewBug.vb [Design]

	username	password
▶	Harshitha	harsh@25
*	NULL	NULL

TeamLeadLogin:

teamloginTab: Qu...IN\DEBUG\BUG.MDF) | AffixN

	username	password
	gowtham	gowtham
▶*	NULL	NULL

Developer Login:

AnalyseTab: Quer...IN\DEBUG\BUG.MDF) | teamloginTab: Qu...IN\DEBUG\BUG.MDF) | regTab: Query(de...IN\DEBUG\BUG.MDF) | reg1tab: Query(d...IN\DEBUG\BUG.MDF)

	UserId	Password	Domain	Email
	gowtham	gowtham	Java and ...	gowtha...
	sushmitha	123	Java and ...	sushmith...
	prem	prem@1...	DOT.NET	prem@g...
	shanmuk...	shan@123	DOT.NET	shanmuk...
▶*	NULL	NULL	NULL	NULL

Register:

AnalyseTab: Quer...IN\DEBUG\BUG.MDF) teamloginTab: Qu...IN\DEBUG\BUG.MDF) regTab: Query(de...IN\DEBUG\BUG.MDF)

devname	domain	email	password	dob	address	phno
admin	Select yo...	admin@...	admin	23 Januar...	12 1st m...	8748057...
gowtham	Java and ...	gowtha...	123	05 Febru...	banglore	9876543...
fka	Java and ...	abc2@g...	adia	05 Febru...	manglore	1234567...
surya	DOT NET	surya@g...	456	05 Febru...	mysure	8748057...
rudresh	DOT NET	rudresh...	789	05 Febru...	bagalkot	0987654...
prem	Java and ...	abc@gm...	avc@123	14 Febru...	gulbarga	6361434...
rudra	Java and ...	belaw92...	edf@456	13 Febru...	raichur	7892172...
Rinku	Select yo...	rinkuseer...	rinku@831	08 Augu...	bidar	8149256...
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

AnalyseTab

AnalyseTab: Quer...IN\DEBUG\BUG.MDF) teamloginTab: Qu...IN\DEBUG\BUG.MDF) reg

Summary	Des	Product	Platform	Imp	Id	Date
testing	1	internal	Java and ...	High	1	02-02-20...
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

TraigeTab

TraigeTab: Query(...IN\DEBUG\BUG.MDF) AnalyseTab: Quer...IN\DEBUG\BUG.MDF) teamloginTab: Qu...IN\DEBUG\BUG.MDF)

Summary	Des	Product	Platform	Imp	Id	Date	Status	Developer	Devname
Testing	2	Java and ...	Syntax Er...	Medium	2	24-02-20...	*NULL*	Java and ...	fka
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*		*NULL*
NULL	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*	*NULL*

:

CHAPTER 6

Coding

1.MDI Parent

```
Private Sub HomeToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles HomeToolStripMenuItem.Click
        Me.Show()
    End Sub


    Private Sub AbstractToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles AbstractToolStripMenuItem.Click
        Abstract.MdiParent = Me
        Abstract.Show()

    End Sub

    Private Sub ManagerLoginToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles ManagerLoginToolStripMenuItem.Click
        Abstract.Hide()

        ManagerLogin.MdiParent = Me
        ManagerLogin.Show()
    End Sub

    Private Sub TeamLeadLoginToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles TeamLeadLoginToolStripMenuItem.Click
        TeamLeaderLogin.MdiParent = Me
        TeamLeaderLogin.Show()
    End Sub

    Private Sub DeveloperLoginToolStripMenuItem_Click(ByVal sender As System.Object,
ByVal e As System.EventArgs) Handles DeveloperLoginToolStripMenuItem.Click
        DeveloperLogin.MdiParent = Me
        DeveloperLogin.Show()
    End Sub

    Private Sub RegistrationToolStripMenuItem_Click(ByVal sender As System.Object, ByVal
e As System.EventArgs) Handles RegistrationToolStripMenuItem.Click
        Register.MdiParent = Me
        Register.Show()
    End Sub

    Private Sub ExitToolStripMenuItem_Click(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles ExitToolStripMenuItem.Click
        Me.Close()
```

```
    End Sub

    Private Sub MdiParent1_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        Dim SAPI As Object
        SAPI = CreateObject("SAPI.spvoice")
        SAPI.speak("Welcome to bug management system")

        TXT = "Welcome To Bug Management System"
        LL = Len(TXT)
        II = 1
        PP = 1
    End Sub


    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        Label4.Text = Label4.Text + Mid(TXT, II, 1)
        If II > LL Then
            II = 0
            Label4.Text = ""
        End If
        II = II + 1
    End Sub
End Class
```

2.Manager

2.1 Login

```
Imports System.Data.SqlClient

Public Class ManagerLogin

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Then
            MsgBox(" Username required")
            Exit Sub
        End If
        If TextBox2.Text = "" Then
            MsgBox(" password required")
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd0 As New SqlCommand("select * from mloginTab where username='" &
UCase(TextBox1.Text) & "' and password= '" & TextBox2.Text & "'", conn)
        Dim D1 As SqlDataReader = cmd0.ExecuteReader()
        If D1.HasRows Then
            Me.Hide()
            Manger_Main_Page.Show()

            If conn.State = ConnectionState.Open Then conn.Close()
        Else
            MsgBox("Username or password is not corret please Check!!!")
```

```
        End If
        TextBox1.Text = ""
        TextBox2.Text = ""
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        MdiParent1.Show()
        Me.Hide()
    End Sub
```

2.2.RecuritDeveloper

```
Imports System.Data.SqlClient

Public Class Recruit_Developer
    Dim pkvar

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        pkvar = DataGridView1.CurrentRow.Cells(0).Value
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim Cmd0 As New SqlCommand("Select * from regTab where devname='" & pkvar & "'",
conn)
        Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
        If D1.HasRows Then
            D1.Read()
            'TextBox1.Text = D1(0).ToString
            'TextBox2.Text = D1(1).ToString
        Else
            'TextBox1.Text = ""
            'TextBox2.Text = ""
        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim q1var, q2var
        If TextBox1.Text = "" Then
            MsgBox("Please Enter The Necessary Details")
            Exit Sub
        End If

        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd0 As New SqlCommand("Select UserId from reg1Tab where UserId='" &
(TextBox1.Text) & "'", conn)
        Dim D1 As SqlDataReader = cmd0.ExecuteReader()
        If D1.HasRows Then
            MsgBox("This Developer Is Already Been Approved")
            If conn.State = ConnectionState.Open Then conn.Close()
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        q1var = "Insert into reg1Tab("
```

```
        q2var = "Values("
        q1var = q1var & "UserId" & " , "
        q2var = q2var & "'" & (TextBox1.Text) & " ', "
        q1var = q1var & "Password" & " , "
        q2var = q2var & "'" & (TextBox4.Text) & " ', "
        q1var = q1var & "Domain" & " , "
        q2var = q2var & "'" & (TextBox2.Text) & " ', "
        q1var = q1var & "Email" & ")"
        q2var = q2var & "'" & (TextBox3.Text) & " ') "

        Dim cmd1 As New SqlCommand(q1var & q2var, conn)
        cmd1.ExecuteNonQuery()
        If conn.State = ConnectionState.Open Then conn.Close()
        'disRecords()

        MsgBox("Approved Successfully")
        TextBox1.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox2.Text = ""
    End Sub

    Private Sub DataGridView2_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView2.CellContentClick
        pkvar = DataGridView2.CurrentRow.Cells(0).Value
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim Cmd0 As New SqlCommand("Select * from reg1 where UserId='" & pkvar & "'",
conn)
        Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
        If D1.HasRows Then
            D1.Read()
            'TextBox1.Text = D1(0).ToString
            'TextBox2.Text = D1(1).ToString
        Else
            'TextBox1.Text = ""
            'TextBox2.Text = ""
        End If
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd2 As New SqlCommand("Delete from regTab where devname='" & pkvar & "'",
conn)
        cmd2.ExecuteNonQuery()
        If conn.State = ConnectionState.Open Then conn.Close()
        disRecords1()


        disRecords()
        disRecords1()
    End Sub

    Private Sub Recruit_Developer_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        disRecords()
```

```
    End Sub

    Private Sub disRecords()
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim DS1 As New DataSet
        Dim adp As New SqlDataAdapter("select devname as 'Developer Name',domain as
'Domain',email as 'E-Mail',password as 'Password',dob as 'Date of Birth',address as
'Address',phno as 'Phone Number' From regTab order by devname", conn)
        adp.Fill(DS1)
        DataGridView1.DataSource = DS1.Tables(0)
        If conn.State = ConnectionState.Open Then conn.Close()
    End Sub

    Private Sub disRecords1()
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim DS1 As New DataSet
        Dim adp As New SqlDataAdapter("select UserId, Domain, Email  From reg1Tab order
by UserId", conn)
        adp.Fill(DS1)
        DataGridView2.DataSource = DS1.Tables(0)
        If conn.State = ConnectionState.Open Then conn.Close()
    End Sub

    Private Sub DataGridView1_CellMouseClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DataGridView1.CellMouseClick
        TextBox1.Text = DataGridView1.Rows(e.RowIndex).Cells(0).Value
        TextBox2.Text = DataGridView1.Rows(e.RowIndex).Cells(1).Value
        TextBox3.Text = DataGridView1.Rows(e.RowIndex).Cells(2).Value
        TextBox4.Text = DataGridView1.Rows(e.RowIndex).Cells(3).Value
    End Sub
End Class
```

2.3.Trace History

```
Imports System.Data.SqlClient

Public Class Trace
    Dim pkvar


    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        pkvar = DataGridView1.CurrentRow.Cells(0).Value
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim Cmd0 As New SqlCommand("Select * from TraigeTab where summary='" & pkvar &
"'", conn)
        Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
        If D1.HasRows Then
            D1.Read()
            'TextBox1.Text = D1(0).ToString
            'TextBox2.Text = D1(1).ToString
        Else
```

```
            'TextBox1.Text = ""
            'TextBox2.Text = ""
        End If
    End Sub

    Private Sub Trace_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        disrecord()
    End Sub

    Private Sub disrecord()
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim DS1 As New DataSet
        Dim adp As New SqlDataAdapter("select id as 'Bug ID',Date as 'Updated
on',Des,Status,Summary as 'Summary' From TraigeTab order by Id", conn)
        adp.Fill(DS1)
        DataGridView1.DataSource = DS1.Tables(0)
        If conn.State = ConnectionState.Open Then conn.Close()
    End Sub

End Class
```

3.TeamLead

3.1.Login

```
Imports System.Data.SqlClient

Public Class TeamLeaderLogin

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Then
            MsgBox(" Username required")
            Exit Sub
        End If
        If TextBox2.Text = "" Then
            MsgBox(" password required")
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd0 As New SqlCommand("select * from teamloginTab where username='" &
UCase(TextBox1.Text) & "' and password= '" & TextBox2.Text & "'", conn)
        Dim D1 As SqlDataReader = cmd0.ExecuteReader()
        If D1.HasRows Then
            TeamLeaderMainPage.Show()
            Me.Hide()
            If conn.State = ConnectionState.Open Then conn.Close()
        Else
            MsgBox("Username or password is not corret please Check!!!")
        End If
        TextBox1.Text = ""
        TextBox2.Text = ""
    End Sub
```

```
    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        MdiParent1.Show()
        Me.Hide()
    End Sub
```

3.2.AffixNewBug

```
Imports System.Data.SqlClient

Public Class AffixNewBug
    Dim Count As Integer
    Dim LL, II, PP As Integer
    Dim TXT As String

    Private Sub AffixNewBug_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        TXT = "Affix New Bug Report"
        LL = Len(TXT)
        II = 1
        PP = 1

        autogenerate_id()
    End Sub

    Private Sub autogenerate_id()
        Dim number As Integer
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd As New SqlCommand(number, conn)
        cmd.CommandText = "SELECT MAX(Id) FROM AnalyseTab"
        If IsDBNull(cmd.ExecuteScalar) Then
            number = 1
            TextBox2.Text = number
        Else
            number = cmd.ExecuteScalar + 1
            TextBox2.Text = number
        End If
        cmd.Dispose()
    End Sub

    Private Sub Timer1_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer1.Tick
        Label6.Text = Label6.Text + Mid(TXT, II, 1)
        If II > LL Then
            II = 0
            Label6.Text = ""
        End If
        II = II + 1
    End Sub

    Private Sub Timer2_Tick(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Timer2.Tick
        Dim i As Integer
        TextBox4.Text = Today & vbNewLine & TimeOfDay
        i = i + 1
```

```
        If i > 5 Then
            Timer2.Enabled = False
            TextBox4.Show()
        End If
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim q1var, q2var
        If TextBox1.Text = "" Then
            MsgBox("Please Enter The Necessary Details")
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd0 As New SqlCommand("Select des from AnalyseTab where des='" &
(TextBox3.Text) & "'", conn)
        Dim D1 As SqlDataReader = cmd0.ExecuteReader()
        If D1.HasRows Then
            MsgBox("This Record Is Already Present In The DataBase")
            If conn.State = ConnectionState.Open Then conn.Close()
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        q1var = "Insert into AnalyseTab("
        q2var = "Values("
        q1var = q1var & "summary" & " , "
        q2var = q2var & "'" & (TextBox1.Text) & " ', "
        q1var = q1var & "des" & " , "
        q2var = q2var & "'" & (TextBox2.Text) & " ', "
        q1var = q1var & "product" & " , "
        q2var = q2var & "'" & (TextBox3.Text) & " ', "
        q1var = q1var & "platform" & " , "
        q2var = q2var & "'" & (ComboBox1.Text) & " ', "
        q1var = q1var & "imp" & " , "
        q2var = q2var & "'" & (ComboBox2.Text) & " ', "
        q1var = q1var & "Id" & " , "
        q2var = q2var & "'" & (TextBox2.Text) & " ', "
        q1var = q1var & "date" & ")"
        q2var = q2var & "'" & (TextBox4.Text) & " ') "

        Dim cmd1 As New SqlCommand(q1var & q2var, conn)
        cmd1.ExecuteNonQuery()
        If conn.State = ConnectionState.Open Then conn.Close()
        'disRecords()
        autogenerate_id()

        MsgBox("Bug Reported Successfully")
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        ComboBox2.Text = "Select Importance"
        ComboBox1.Text = "Select Platform"

    End Sub

End Class
```

3.3.Analyse Bug Report

```
Imports System.Data.SqlClient

Public Class AnalyseBugReport
    Dim pkvar

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        pkvar = DataGridView1.CurrentRow.Cells(3).Value
        MsgBox(pkvar)
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim Cmd0 As New SqlCommand("Select * from AnalyseTab where platform='" & pkvar &
"'", conn)
        Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
        If D1.HasRows Then
            D1.Read()
            a = D1(3).ToString

            'TextBox1.Text = D1(0).ToString
            'TextBox2.Text = D1(1).ToString
        Else
            'TextBox1.Text = ""
            'TextBox2.Text = ""
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd3 As New SqlCommand("select UserId from reg1Tab where domain='" & a & "'",
conn)
        Dim adapter As New SqlDataAdapter(cmd3)
        Dim table As New DataTable()

        adapter.Fill(table)
        ComboBox1.DataSource = table
        ComboBox1.DisplayMember = "UserId"
        ComboBox1.ValueMember = "UserId"
    End Sub

    Private Sub AnalyseBugReport_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        disrecord()

    End Sub

    Private Sub disrecord()
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim DS1 As New DataSet
        Dim adp As New SqlDataAdapter("select Id as 'Bug ID',summary as 'Summary',des as
'Description',Platform as 'Platform',Product as 'Product',Imp as 'Critical',Date From
AnalyseTab order by Id", conn)
        adp.Fill(DS1)
        DataGridView1.DataSource = DS1.Tables(0)
        If conn.State = ConnectionState.Open Then conn.Close()
    End Sub
```

```
    Private Sub DataGridView1_CellMouseClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DataGridView1.CellMouseClick
        TextBox1.Text = DataGridView1.Rows(e.RowIndex).Cells(0).Value
        TextBox2.Text = DataGridView1.Rows(e.RowIndex).Cells(1).Value
        TextBox3.Text = DataGridView1.Rows(e.RowIndex).Cells(2).Value
        TextBox4.Text = DataGridView1.Rows(e.RowIndex).Cells(3).Value
        TextBox5.Text = DataGridView1.Rows(e.RowIndex).Cells(4).Value
        TextBox6.Text = DataGridView1.Rows(e.RowIndex).Cells(5).Value
        TextBox7.Text = DataGridView1.Rows(e.RowIndex).Cells(6).Value
    End Sub

    Private Sub TextBox5_Click(ByVal sender As Object, ByVal e As System.EventArgs)
Handles TextBox5.Click



    End Sub

    Private Sub TextBox5_TextChanged(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles TextBox5.TextChanged

    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim q1var, q2var
        If TextBox1.Text = "" Then
            MsgBox("Please Enter The Necessary Details")
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd0 As New SqlCommand("Select des from TraigeTab where des='" &
(TextBox3.Text) & "'", conn)
        Dim D1 As SqlDataReader = cmd0.ExecuteReader()
        If D1.HasRows Then
            MsgBox("This Bug Is Already Assigned")
            If conn.State = ConnectionState.Open Then conn.Close()
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        q1var = "Insert into TraigeTab("
        q2var = "Values("
        q1var = q1var & "summary" & " , "
        q2var = q2var & "'" & (TextBox2.Text) & " ', "
        q1var = q1var & "des" & " , "
        q2var = q2var & "'" & (TextBox3.Text) & " ', "
        q1var = q1var & "product" & " , "
        q2var = q2var & "'" & (TextBox4.Text) & " ', "
        q1var = q1var & "platform" & " , "
        q2var = q2var & "'" & (TextBox5.Text) & " ', "
        q1var = q1var & "imp" & " , "
        q2var = q2var & "'" & (TextBox6.Text) & " ', "
        q1var = q1var & "Id" & " , "
        q2var = q2var & "'" & (TextBox1.Text) & " ', "
        q1var = q1var & "date" & " , "
```

```
        q2var = q2var & "'" & (TextBox7.Text) & " ', "
        q1var = q1var & "Developer" & ","
        q2var = q2var & "'" & a & "',"
        q1var = q1var & "Devname" & ")"
        q2var = q2var & "'" & ComboBox1.Text & "')"

        Dim cmd1 As New SqlCommand(q1var & q2var, conn)
        cmd1.ExecuteNonQuery()
        If conn.State = ConnectionState.Open Then conn.Close()
        'disRecords()


        MsgBox("Traige Assigned Successfully")
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        ComboBox1.Text = "Select Developer"
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
        MsgBox(a)

    End Sub
End Class
```

3.4.Data Reduction

```
Imports System.Data.SqlClient

Public Class DataReduction
    Dim pkvar


    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        FilterData(TextBox1.Text)
    End Sub

    Private Sub DataReduction_Load(ByVal sender As System.Object, ByVal e As
System.EventArgs) Handles MyBase.Load
        FilterData("")
        disRecords()
    End Sub

    Private Sub FilterData(ByVal valueToSearch As String)
        'SELECT * From Users WHERE CONCAT(fname, lname, age) like '%F%'
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()

        Dim searchQuery As String = "SELECT * From TraigeTab WHERE summary like '%" &
valueToSearch & "%'"
```

```
        Dim command As New SqlCommand(searchQuery, conn)
        Dim adapter As New SqlDataAdapter(command)
        Dim table As New DataTable()

        adapter.Fill(table)

        DataGridView1.DataSource = table
    End Sub

    Private Sub disRecords()
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim DS1 As New DataSet
        Dim adp As New SqlDataAdapter("select Id as 'Bug ID',summary as
'Summary',Status,Devname From TraigeTab order by Id", conn)
        adp.Fill(DS1)
        DataGridView1.DataSource = DS1.Tables(0)
        If conn.State = ConnectionState.Open Then conn.Close()
    End Sub

    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        pkvar = DataGridView1.CurrentRow.Cells(0).Value
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim Cmd0 As New SqlCommand("Select * from TraigeTab where Id='" & pkvar & "'",
conn)
        Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
        If D1.HasRows Then
            D1.Read()
            'TextBox1.Text = D1(0).ToString
            'TextBox2.Text = D1(1).ToString
        Else
            'TextBox1.Text = ""
            'TextBox2.Text = ""
        End If

    End Sub
End Class
```

4.Developer

4.1.Login

```
Imports System.Data.SqlClient

Public Class DeveloperLogin

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button2.Click
        MdiParent1.Show()
        Me.Hide()
```

```
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        If TextBox1.Text = "" Then
            MsgBox(" Username required")
            Exit Sub
        End If

        If TextBox2.Text = "" Then
            MsgBox(" password required")
            Exit Sub
        End If
        If ComboBox1.Text = "" Then
            MsgBox(" Please Select your Domain")
            Exit Sub
        End If
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim cmd0 As New SqlCommand("select * from  reg1tab where UserId='" &
TextBox1.Text & "' and Password= '" & TextBox2.Text & "'", conn)
        Dim D1 As SqlDataReader = cmd0.ExecuteReader()
        If D1.HasRows Then
            MsgBox(TextBox1.Text)
            MsgBox(Module1.a = TextBox1.Text)

            Module1.developername = TextBox1.Text
            MDIParent2.Show()
            Me.Hide()

            If conn.State = ConnectionState.Open Then conn.Close()
        Else
            MsgBox("Username or password is not corret please Check!!!")
        End If
        TextBox1.Text = ""
        TextBox2.Text = ""
    End Sub
```

4.2.Traige Box

```
Imports System.Data.SqlClient

Public Class Traige
    Dim pkvar As String

    Private Sub Traige_Load(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles MyBase.Load
        displayRecord()

    End Sub

    Private Sub displayRecord()
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
```

```
        Dim DS1 As New DataSet
        Dim adp As New SqlDataAdapter("select Id as 'Bug ID',summary as 'Summary',des as
'Description',product as 'Product',platform as 'Platform',imp as 'Critical',date From
TraigeTab where DeveloperName = '" & developername & "' order by Id", conn)
        adp.Fill(DS1)
        DataGridView1.DataSource = DS1.Tables(0)
        If conn.State = ConnectionState.Open Then conn.Close()

    End Sub
    Private Sub DataGridView1_CellContentClick(ByVal sender As System.Object, ByVal e As
System.Windows.Forms.DataGridViewCellEventArgs) Handles DataGridView1.CellContentClick
        pkvar = DataGridView1.CurrentRow.Cells(0).Value
        If conn.State = ConnectionState.Open Then conn.Close()
        conn.Open()
        Dim Cmd0 As New SqlCommand("Select * from TraigeTab where developername='" &
pkvar & "'", conn)
        Dim D1 As SqlDataReader = Cmd0.ExecuteReader()
        If D1.HasRows Then
            D1.Read()

        Else
            'TextBox1.Text = ""
            'TextBox2.Text = ""
        End If
    End Sub
    Private Sub DataGridView1_CellMouseClick(ByVal sender As Object, ByVal e As
System.Windows.Forms.DataGridViewCellMouseEventArgs) Handles DataGridView1.CellMouseClick
        On Error Resume Next
        TextBox1.Text = DataGridView1.Rows(e.RowIndex).Cells(0).Value
        TextBox2.Text = DataGridView1.Rows(e.RowIndex).Cells(1).Value
        TextBox3.Text = DataGridView1.Rows(e.RowIndex).Cells(2).Value
        TextBox5.Text = DataGridView1.Rows(e.RowIndex).Cells(3).Value
        TextBox4.Text = DataGridView1.Rows(e.RowIndex).Cells(4).Value
        TextBox6.Text = DataGridView1.Rows(e.RowIndex).Cells(5).Value
        TextBox7.Text = Today
    End Sub

    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs)
Handles Button1.Click
        Dim q1var, q2var As String
        For Each r As DataGridViewRow In DataGridView1.Rows
            If conn.State = ConnectionState.Open Then conn.Close()
            conn.Open()
            Dim cmd2 As New SqlCommand("Delete from TraigeTab where Id='" &
r.Cells(0).Value & "'", conn)
            cmd2.ExecuteNonQuery()
            If conn.State = ConnectionState.Open Then conn.Close()

            If r.Cells(0).Value = "" Then Exit For
            q1var = "insert into TraigeTab("
            q2var = " values("
            q1var = q1var & "Id" & ","
            q2var = q2var & "" & (TextBox1.Text) & ","
            q1var = q1var & "des" & ","
            q2var = q2var & "'" & (TextBox3.Text) & "',"
            q1var = q1var & "product" & ","
            q2var = q2var & "'" & (TextBox4.Text) & "',"
            q1var = q1var & "platform" & ","
```

```
            q2var = q2var & "'" & (TextBox5.Text) & "',"
            q1var = q1var & "imp" & ","
            q2var = q2var & "'" & (TextBox6.Text) & "',"
            q1var = q1var & "summary" & ","
            q2var = q2var & "'" & (TextBox2.Text) & "',"
            q1var = q1var & "date" & ","
            q2var = q2var & "'" & (TextBox7.Text) & "',"
            q1var = q1var & "Status" & ","
            q2var = q2var & "'" & (ComboBox3.Text) & "',"
            q1var = q1var & "developername" & ")"
            q2var = q2var & "'" & Module1.developername & "')"

            If conn.State = ConnectionState.Open Then conn.Close()
            conn.Open()
            'MsgBox(q1Var & q2Var)
            Dim cmd1 As New SqlCommand(q1var & q2var, conn)
            cmd1.ExecuteNonQuery()
            If conn.State = ConnectionState.Open Then conn.Close()
            displayRecord()

        Next
        MsgBox("Updated Successfully")
        TextBox1.Text = ""
        TextBox2.Text = ""
        TextBox3.Text = ""
        TextBox4.Text = ""
        TextBox5.Text = ""
        TextBox6.Text = ""
        TextBox7.Text = ""
        ComboBox3.Text = "Select Status"
    End Sub
End Class
```

CHAPTER 7

Software Testing is evaluation of the software against requirements gathered from users and system specifications. Testing is conducted at the phase level in software development life cycle or at module level in program code. Software testing comprises of Validation and Verification.

SOFTWARE VALIDATION:

Validation is process of examining whether or not the software satisfies the user requirements. It is carried out at the end of the SDLC. If the software matches requirements for which it was made, it is validated.

- Validation ensures the product under development is as per the user requirements.
- Validation answers the question – “Are we developing the product which attempts that entire user needs from this software?”
- Validation emphasizes on user requirements.

SOFTWARE VERIFICATION:

Verification is the process of confirming if the software is meeting the business requirements, and id developed adhering to the proper specifications and methodologies.

- Verification ensures the product being developed is according to design specifications.
- Verification answers the question- “Are we developing this product by firmly following all design specifications?”
- Verification concentrates on the design and system specifications.

Targets of the test are:

- Errors- These are actual coding mistakes made by developers. In addition, there is a difference in output of software and desired output, is considered as an error.

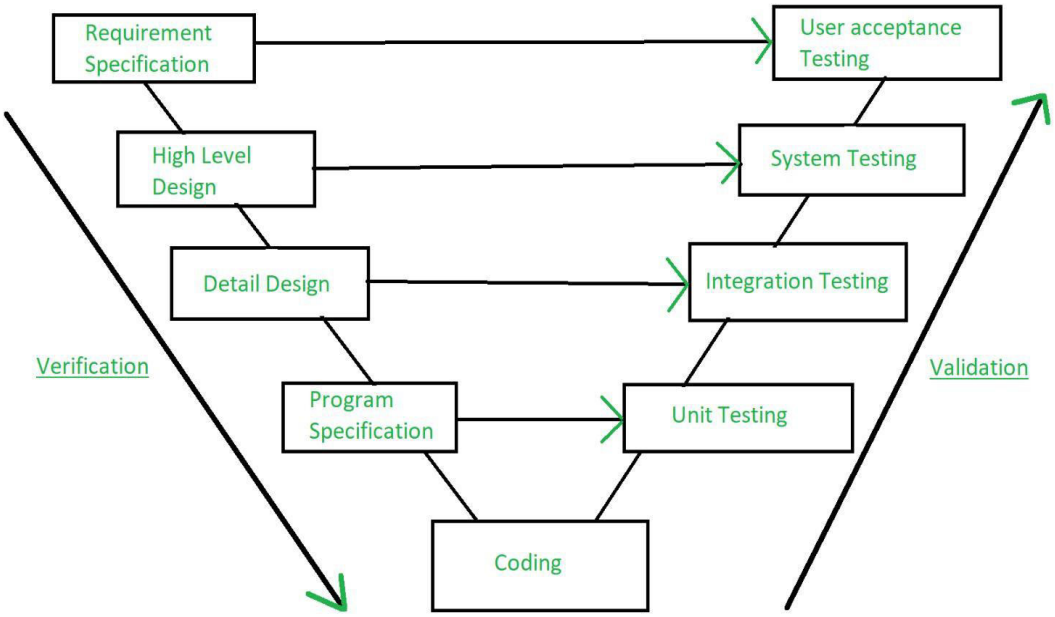
- Fault-When error exists fault occurs. A fault, also known as a bug, is a result of an error which can cause system to fail.
- Failure- Failure is said to be the inability of the system to perform the desired task. Failure occurs when fault exists in the system.

7.1Validation testing:

The process of evaluating software during the development process or at the end of the development process to determine whether it satisfies specified business requirements. Validation Testing ensures that the product actually meets the client's needs. It can also be defined as to demonstrate that the product fulfills its intended use when deployed on appropriate environment.

7.1.1 Workflow:

Validation testing can be best demonstrated using V-Model. The Software/product under test is evaluated during this type of testing.



7.1.2 Classification ms of validation testing:

- Unit Testing
- Integration Testing
- System Testing
- User Acceptance Testing

7.1.3 Advantages of Validation:

ADVANTAGES OF VALIDATION TESTING:

- The users of the V-Model participate in the development and maintenance of the V Model. Change control board publicly maintains the V-Model. The change control board meets anywhere from every day to weekly and processes all change requests received during system development and test.
- The V-Model provides concrete assistance on how to implement an activity and its work steps, defining explicitly the events needed to complete a work step: each activity schema contains instructions, recommendations and detailed explanations of the activity.

7.1.4 Limitation of Validation:

- The placing of contracts for services is not regulated.
- The organization and execution of operation, maintenance, repair and disposal of the system are not covered by the V-Model. However, planning and preparation of a concept for these tasks are regulated in the V-Model.
- The V-Model addresses software development within a project rather than a whole organization.

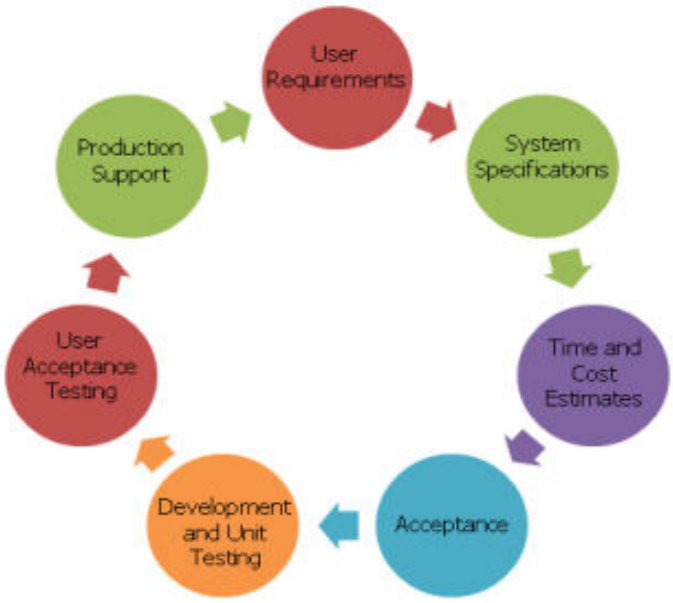
7.2 Unit Testing

Unit testing, a testing technique using which individual modules are tested to determine if there are any issues by the developer himself. It is concerned with functional correctness of the standalone modules. The main aim is to isolate each unit of the system to identify, analyze and fix the defects.

7.2.1 Advantages of Unit Testing:

- Reduces Defects in the newly developed features or reduces bugs when changing the existing functionality.
- Reduces Cost or Testing as defects are captured in very early phase.
- Improves design and allows better refactoring of code.
- Unit Tests, when integrated with build gives the quality of the build as well.

7.2.2 Life Cycle of Unit Testing:



Unit Testing Techniques:

Black Box Testing-Using which the user interface, input and output are tested.

White Box Testing-used to test each one of those functions behavior is tested.

7.2.3 Black Box Testing:

Black box testing treats the software as a "black box"—without any knowledge of internal implementation. Black box testing methods include: equivalence partitioning,

boundary value analysis, all-pairs testing, fuzz testing, model-based testing, traceability matrix, exploratory testing and specification-based testing.

Specification-based testing: Specification-based testing aims to test the functionality of software according to the applicable requirements. Thus, the tester inputs data into, and only sees the output from, the test object. This level of testing usually requires thorough test cases to be provided to the tester, who then can simply verify that for a given input, the output value (or behavior), either "is" or "is not" the same as the expected value specified in the test case.

Specification-based testing is necessary, but it is insufficient to guard against certain risks.

Advantages and disadvantages:

- The black box tester has no "bonds" with the code, and a tester's perception is very simple: a code *must* have bugs.
- Using the principle, "Ask and you shall receive," black box testers find bugs where programmers do not.
- But, on the other hand, black box testing has been said to be "like a walk in a dark labyrinth without a flashlight," because the tester doesn't know how the software being tested was actually constructed.
- As a result, there are situations when (1) a tester writes many test cases to check something that could have been tested by only one test case, and/or (2) some parts of the back-end are not tested at all.

- Therefore, black box testing has the advantage of "an unaffiliated opinion," on the one hand, and the disadvantage of "blind exploring," on the other.

Black box testing techniques:

EQUIVALENCE CLASS- The input is divided into similar classes. If one element of a class passes the test, it is assumed that all the class is passed.

BOUNDARY VALUES- The input divided into higher and lower end values. If these values pass the test it is assumed that all values in between, may pass too.

CAUSE-EEFECT GRAPHING- In both previous methods, only one input value at a time is tested. Cause (input)-Effect (output) is a testing technique where combination of input values is tested in a systematic way.

PAIR-WISE TESTING- The behavior of software depends on multiple parameters. In pair wise testing, the multiple parameters are tested pair-wise for their different values.

STATE-BASED TESTING- The system changes state on provision of input. These systems are tested based on their states and input.

7.2.4 White box testing:

White box testing is when the tester has access to the internal data structures and algorithms including the code that implement these.

Types of white box testing:

- **Control flow testing**: In computer science, control flow (or flow of control) is the order in which individual statements, instructions or function calls of an imperative program are executed or evaluated. The emphasis on explicit control flow distinguishes an imperative programming language from a declarative programming language.

- **Data flow testing:** Data flow testing is a family of test strategies based on selecting paths through the program's control flow in order to explore sequences of events related to the status of variables or data objects.
- **Branch testing:** Branch Testing is defined as a testing method, which has the main goal to ensure that each one of the possible branches from each decision point is executed at least once and thereby ensuring that all reachable code is executed.
- **Statement coverage:** Statement coverage is a white box testing technique, which involves the execution of all the statements at least once in the source code.
- **Decision coverage:** Decision coverage or Branch coverage is a testing method, which aims to ensure that each one of the possible branch from each **decision** point is executed at least once and thereby ensuring that all reachable code is executed.

Advantages: White-box testing is one of the two biggest testing methods used today. It has several major advantages:

- Side effects of having the knowledge of the source code are beneficial to thorough testing.
- Optimization of code becomes easy as inconspicuous bottlenecks are exposed.
- Gives the programmer introspection because developers carefully describe any new implementation.
- Provides traceability of tests from the source, thereby allowing future changes to the source to be easily captured in the newly added or modified tests.
- Easy to automate.
- Provides clear, engineering-based rules for when to stop testing.

Disadvantages: Although white-box testing has great advantages, it is not perfect and contains some disadvantages:

- White-box testing brings complexity to testing because the tester must have knowledge of the program, or the test team needs to have at least one very good programmer who can understand the program at the code level. White-box testing requires a programmer with a high level of knowledge due to the complexity of the level of testing that needs to be done.

- On some occasions, it is not realistic to be able to test every single existing condition of the application and some conditions will be untested.

- The tests focus on the software as it exists, and missing functionality may not be discovered.

7.3 Integration Testing:

Integration testing is the phase in software testing in which individual software modules are combined and tested as a group. Integration testing is conducted to evaluate the compliance of a system or component with specified functional requirements. It occurs after unit testing and before validation testing. Integration testing takes as its input modules that have been unit tested, groups them in larger aggregates, applies tests defined in an integration test plan to those aggregates, and delivers as its output the integrated system ready for system testing.

7.3.1 Integration Strategies:

I. Big-Bang Integration
II. Top Down Integration
III. Bottom Up Integration
IV. Hybrid Integration

Big -Bang Integration: Big bang approach integrates all the modules in one go i.e. it does not go for integrating the modules one by one. It verifies if the system works as expected or not once integrated. If any issue is detected in the completely integrated module, then it becomes difficult to find out which module has caused the issue.

Top-Down Integration: This technique starts from the topmost module and gradually progress towards the lower modules. Only the top module is unit tested in isolation. After this, the lower modules are integrated one by one. The process is repeated until all the modules are integrated and tested.

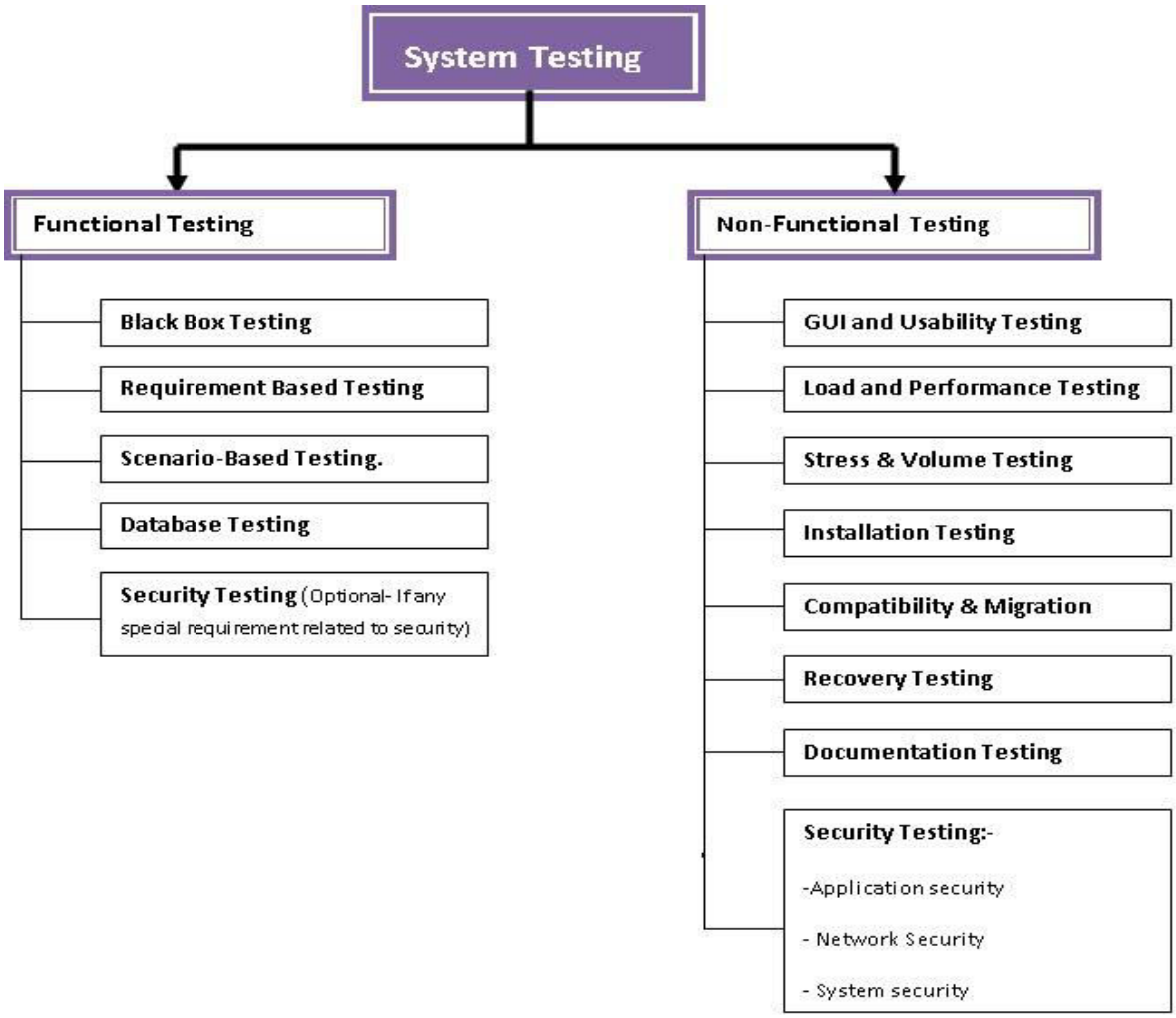
Bottom-Up Integration: Bottom-up testing, as the name suggests starts from the lowest or the innermost unit of the application, and gradually moves up. The Integration testing starts from the lowest module and gradually progresses towards the upper modules of the application. This integration continues till all the modules are integrated and the entire application is tested as a single unit.

Hybrid Integration: Hybrid Testing can be adopted if the customer wants to work on a working version of the application as soon as possible aimed at producing a basic working system in the earlier stages of the development cycle.

7.4 System Testing

System testing is testing conducted on a complete integrated system to evaluate the system's compliance with its specified requirements.

System testing takes, as its input, all of the integrated components that have passedintegration testing. The purpose of integration testing is to detect any inconsistencies between the units that are integrated together (called assemblages). System testing seeks to detect defects both within the "inter-assemblages" and also within the system as a whole. The actual result is the behavior produced or observed when a component or system is tested.



7.4.1 Types Of System Testing:

- Usability testing: To test that the user interfaces are user friendly and easy to operate and use.
- Documentation testing: To test that the user manual and system guide is correct and complete.
- Functionality Testing: To test that the system functionalities are behaving as expected and specified in software requirements document.

- ❖ Inter-operability testing: To test that the system compatibility with other third-party software products.
- ❖ Performance testing: To test that the Performance of the system and to make sure system does not break while operating with available resources.
- ❖ Scalability Testing: To test that the system is scalable enough in terms of user, geographic, resources, etc.
- ❖ Stress Testing: To test that the system behavior when it is operated under stress and to mark its break point.
- ❖ Load & Stability Testing: To test that the system is capable enough to withstand expected load without breaking down.
- ❖ Reliability Testing: To test how long the system can operate without developing any issue or error.
- ❖ Regression Testing: To make sure that new functionalities added into the system does not break the existing functionalities.
- ❖ Compliance & Regulatory Testing: To test that the system specification and operation compiles well with required regulators.
- ❖ Security Testing: To test that the system is secured enough to protect it from unintended users.
- ❖ Recoverability Testing: To test how well the system can recover after breakdown or outage without impacting the business.

7.5 Acceptance Testing:

Acceptance testing, a testing technique performed to determine whether or not the software system has met the requirement specifications. The main purpose of this test is to evaluate the system's compliance with the business requirements and verify if it is has met the required criteria for delivery to end users.

There are various forms of acceptance testing:

- User acceptance Testing
- Business acceptance Testing

User acceptance testing:

This may include factory acceptance testing (FAT), i.e. the testing done by a vendor before the product or system is moved to its destination site, after which site acceptance testing (SAT) may be performed by the users at the site.

Business Acceptance Testing:

BAT is used to determine whether the product meets the business goals and purposes or not. BAT mainly focuses on business profits which are quite challenging due to the changing market conditions and new technologies so that the current implementation may have to being changed which result in extra budgets.

Alpha Testing:

Alpha testing is used to determine the product in the development testing environment by a specialized tester's team usually called alpha testers.

Beta Testing:

Beta testing is used to assess the product by exposing it to the real end-users, usually called beta testers in their environment. Feedback is collected from the users and the defects are fixed. Also, this helps in enhancing the product to give a rich user experience.

7.6 Test Testing:

A test case is a set of conditions or variables under which a tester will determine whether a system under test satisfies requirements or works corks correctly.

The process of developing test cases can also help find problems in the requirements or design of an application.

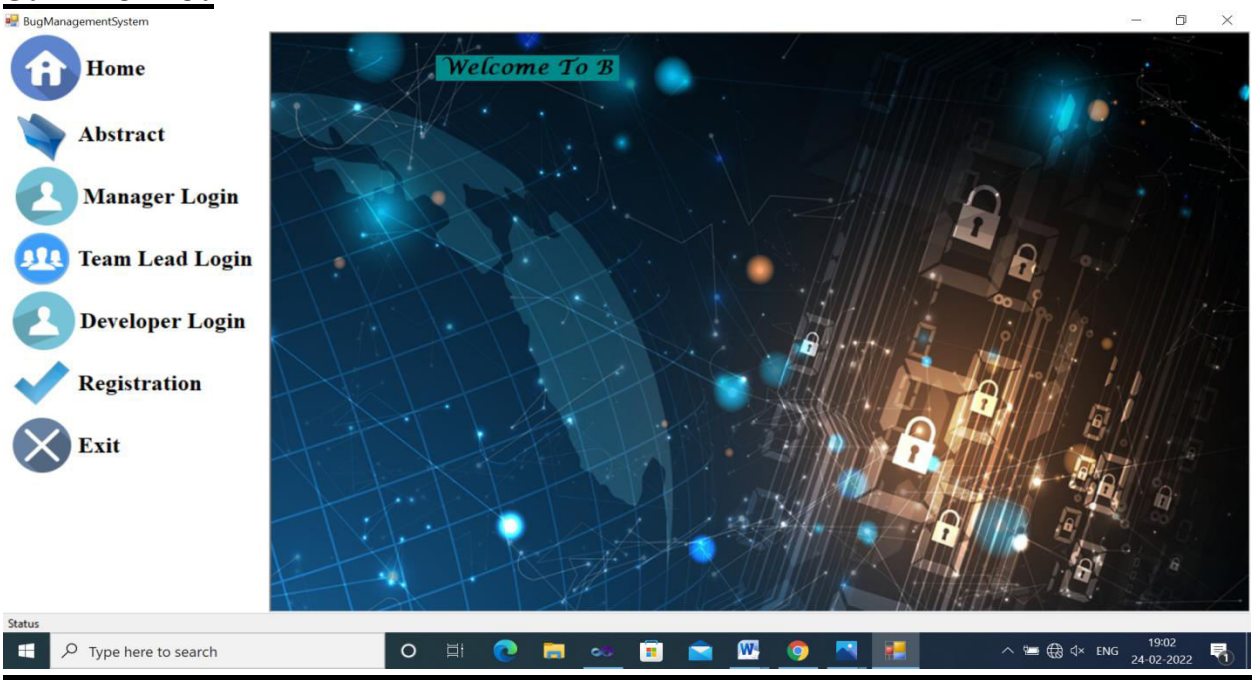
A test case is a specification of the inputs, execution conditions, testing procedure, and expected results that defines a single test to be executed to achieve a particular software test objective, such as to exercise a particular program path or to verify compliance with specific equipment. Test cases underlie testing that is methodical rather than haphazard. A battery of test cases can be built to produce the desired coverage of the software being tested. Formally defined test cases allow the same tests to be run repeatedly against successive versions of the software, allowing for effective and consistent regression testing

CHAPTER 8

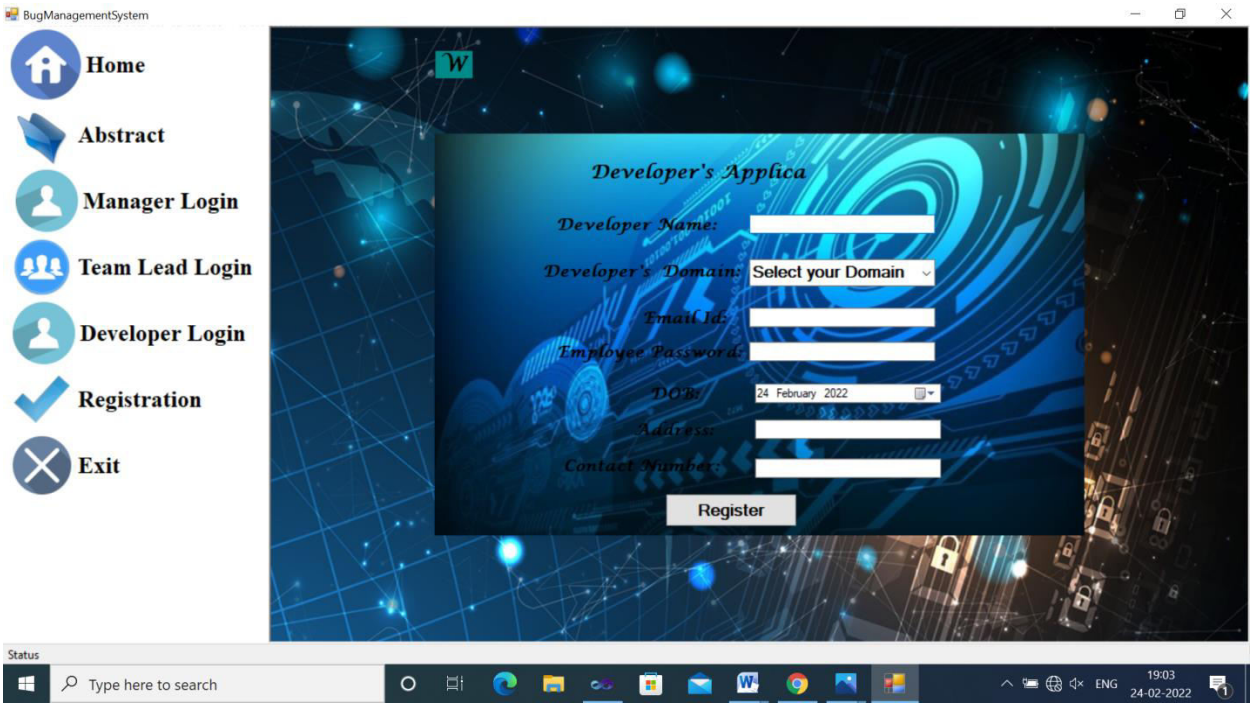
8. GRAPHICAL USER INTERFACE

8.1 WELCOME SCREEN:

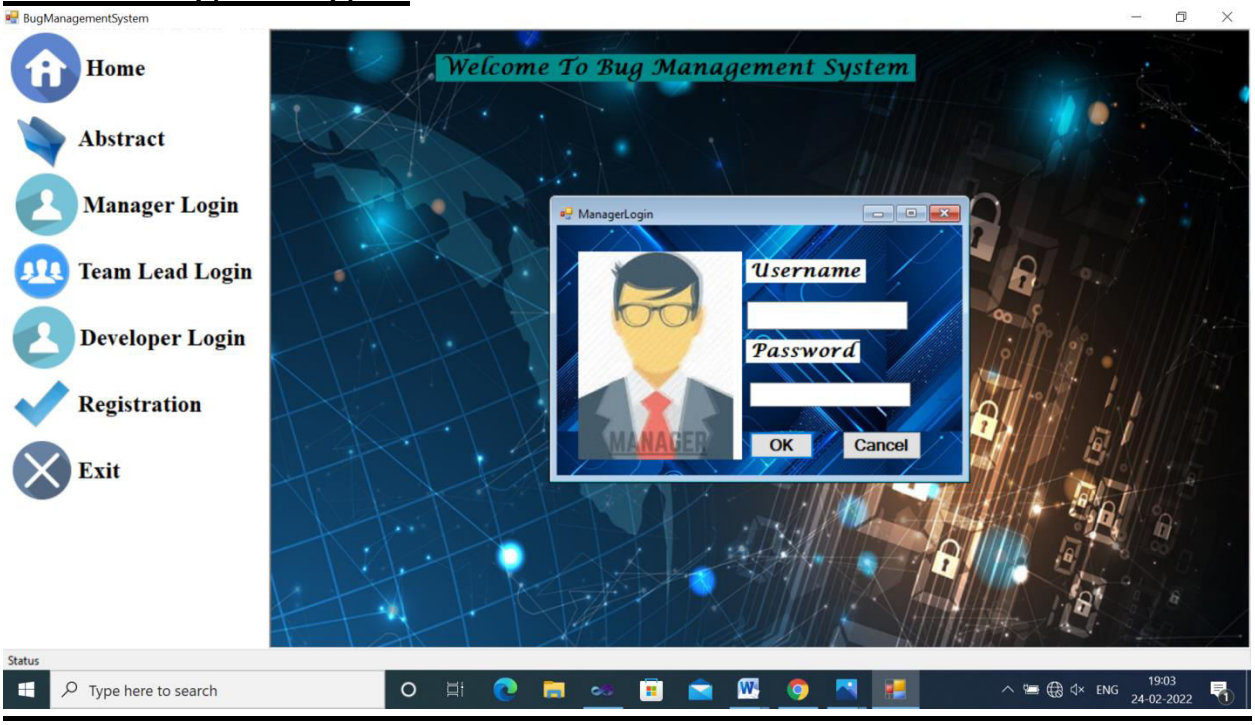
8.2 Home:



8.3 Registration:

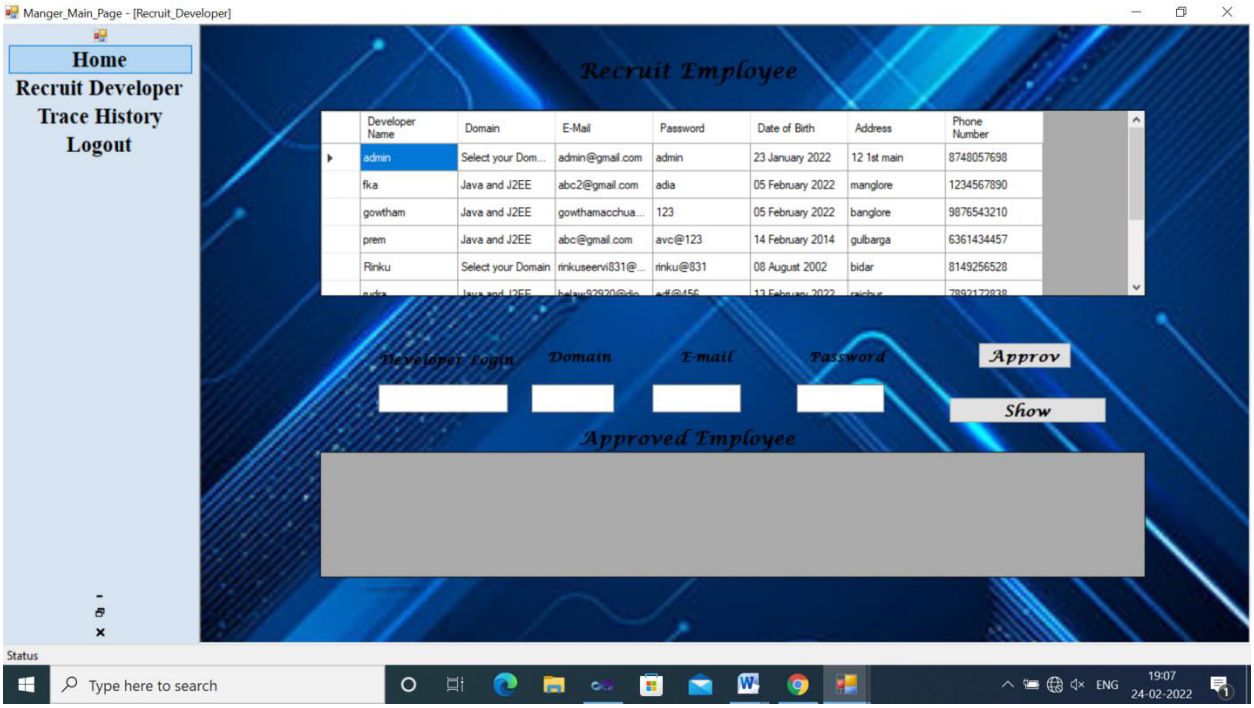


8.4 Manager Login:

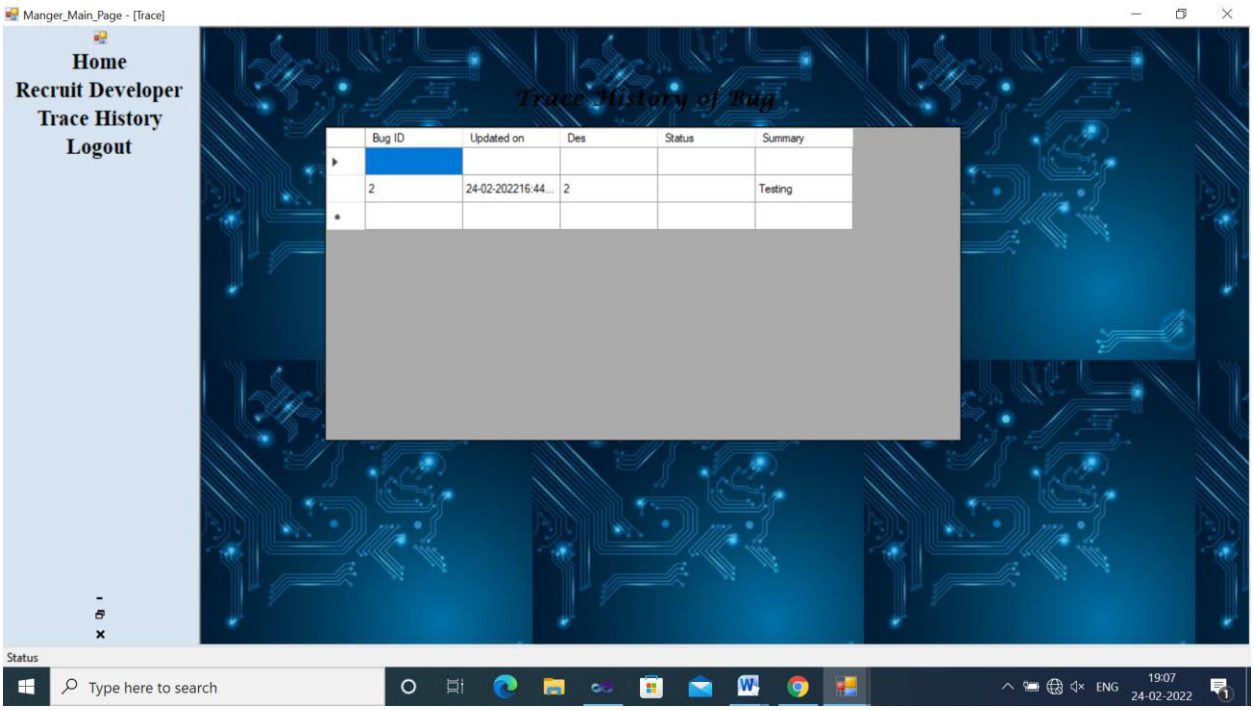


8.4.1 Manager Main Page:

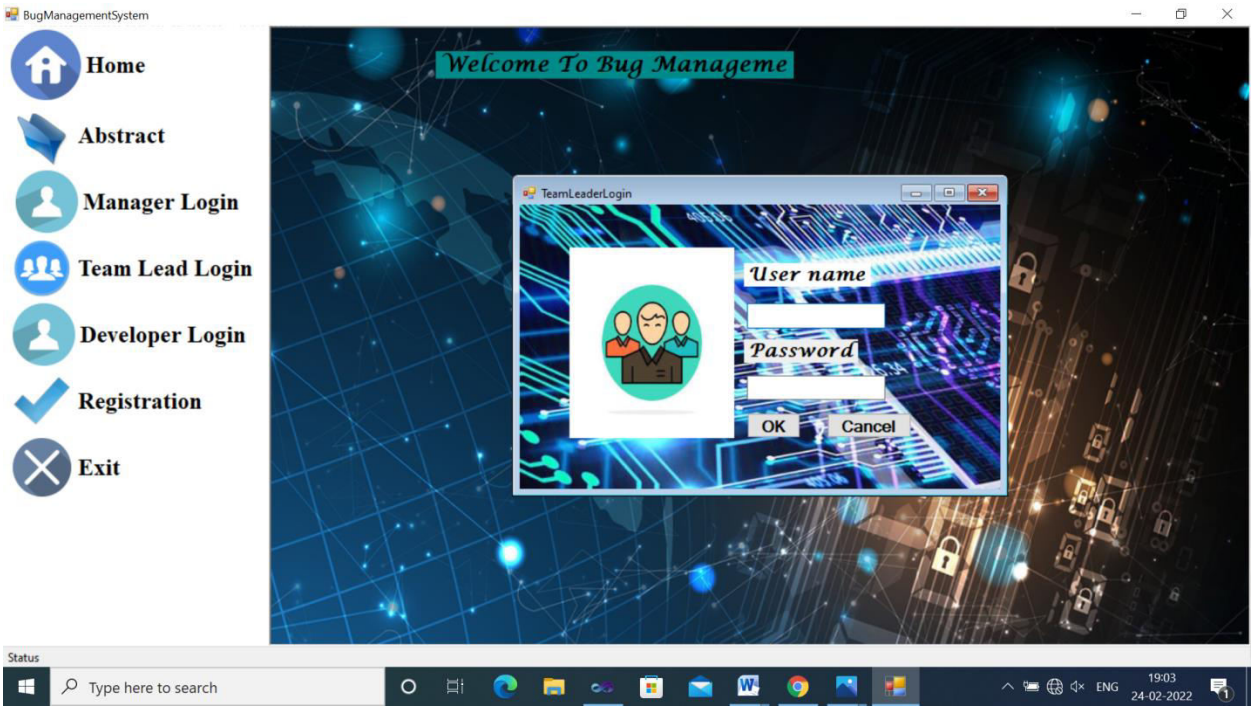
8.4.2 Recurit Developer:



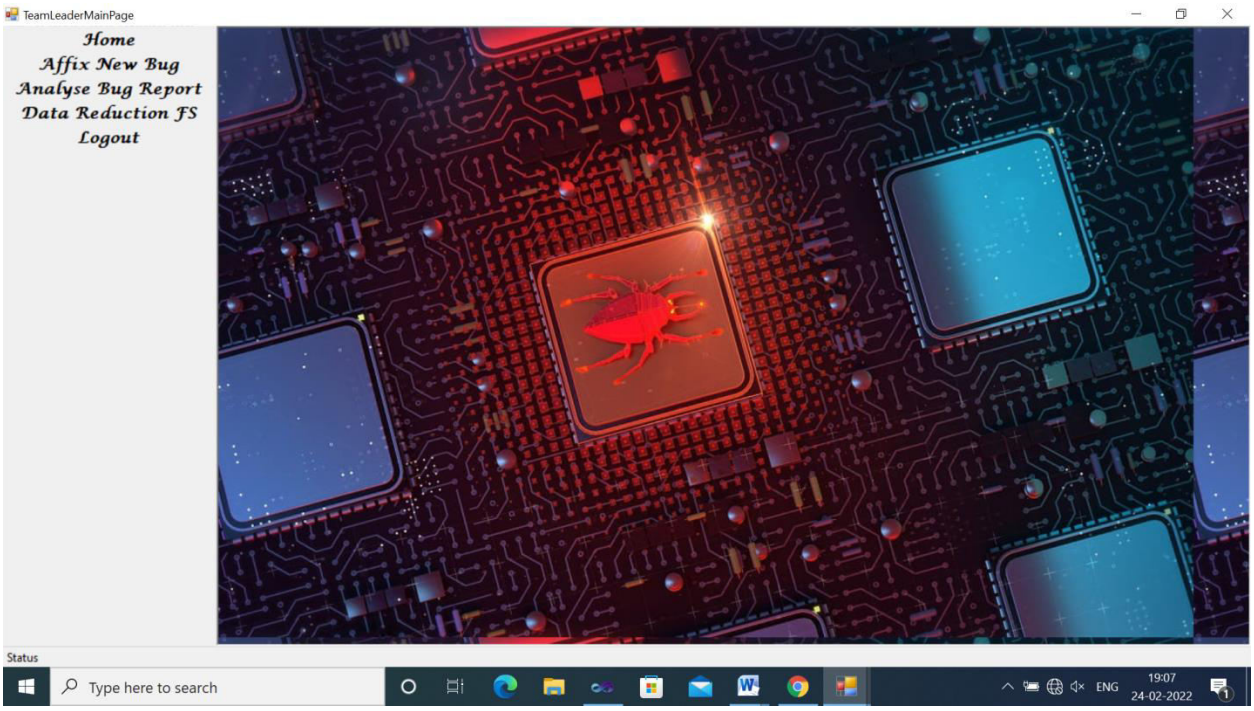
8.4.3 Trace History:



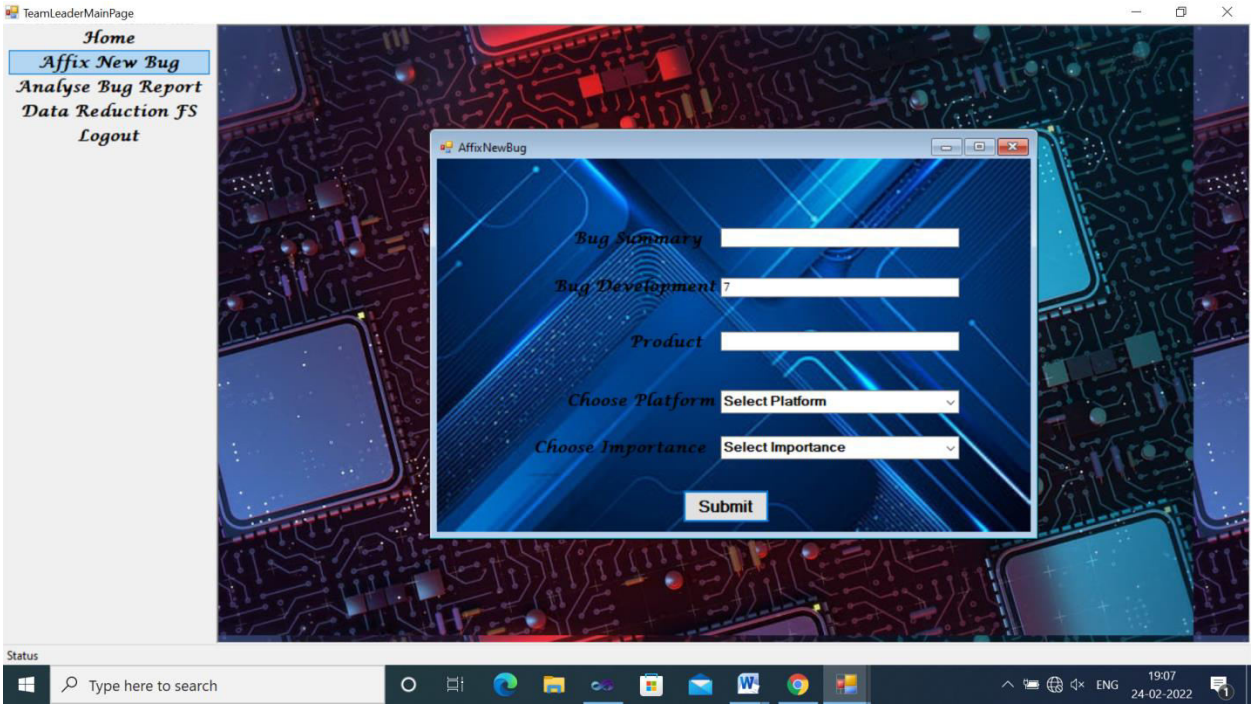
8.5 Team Lead Login:



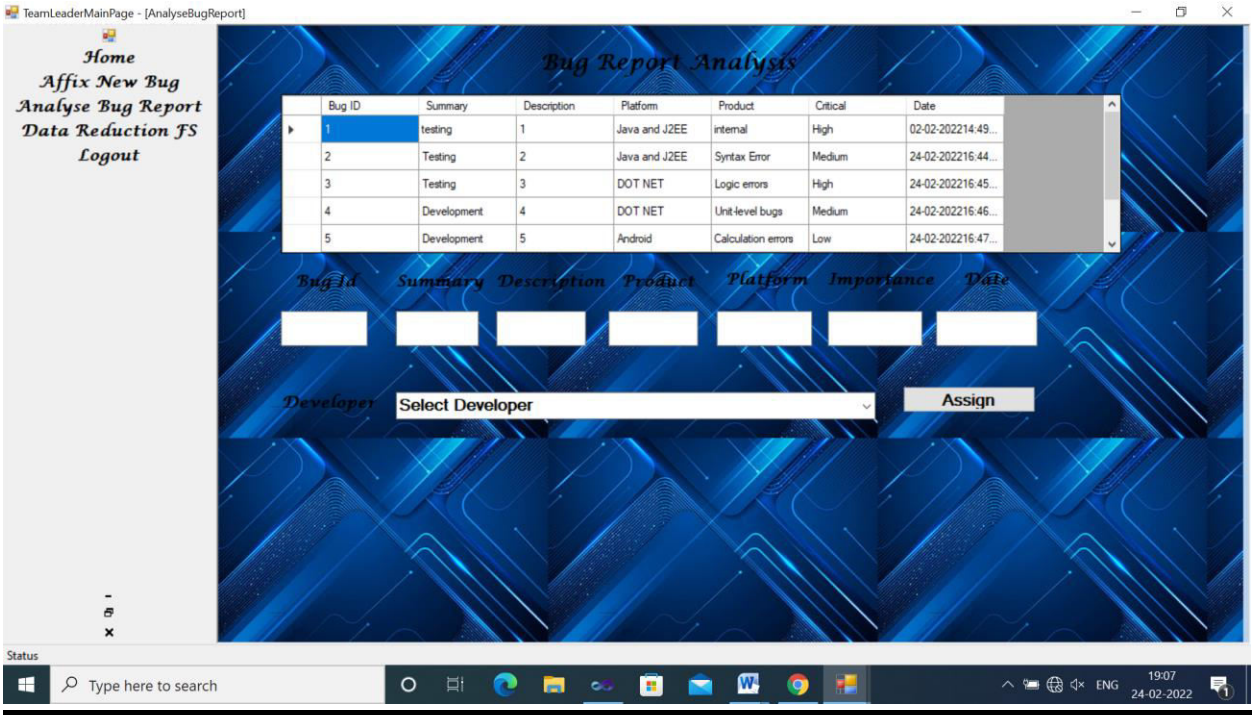
8.5.1 Team Lead MainPage



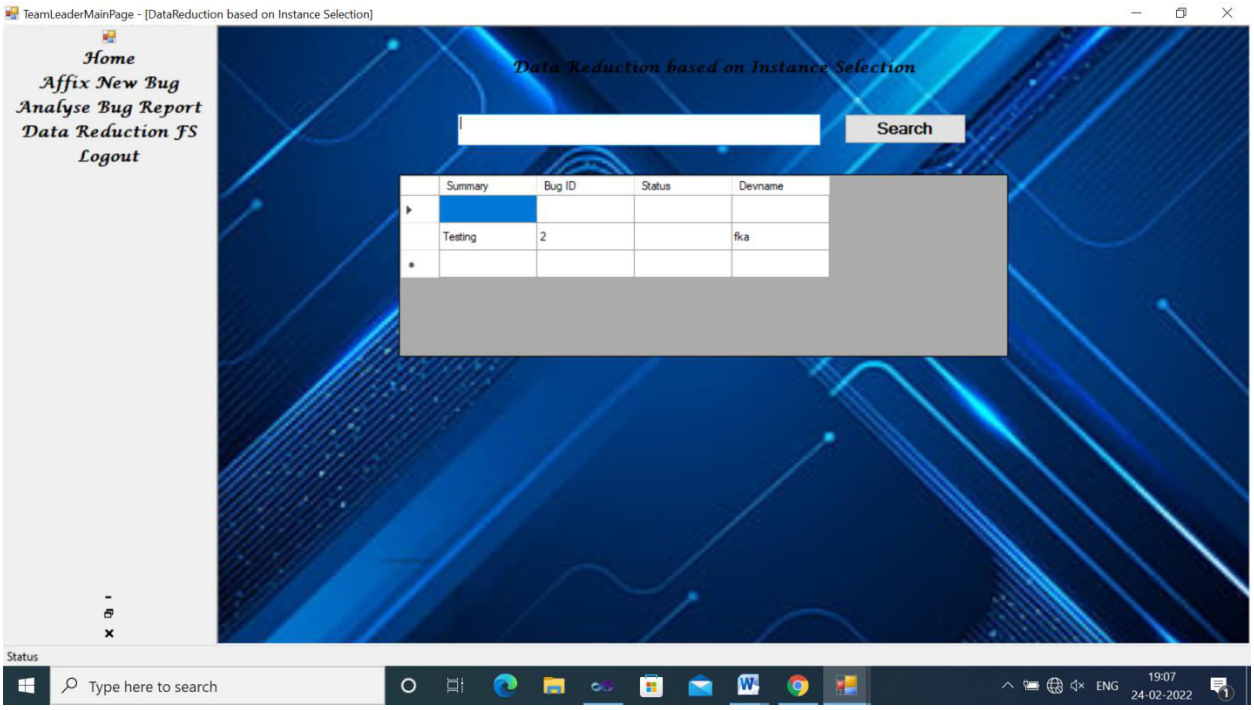
8.5.2 Affix New Bug:



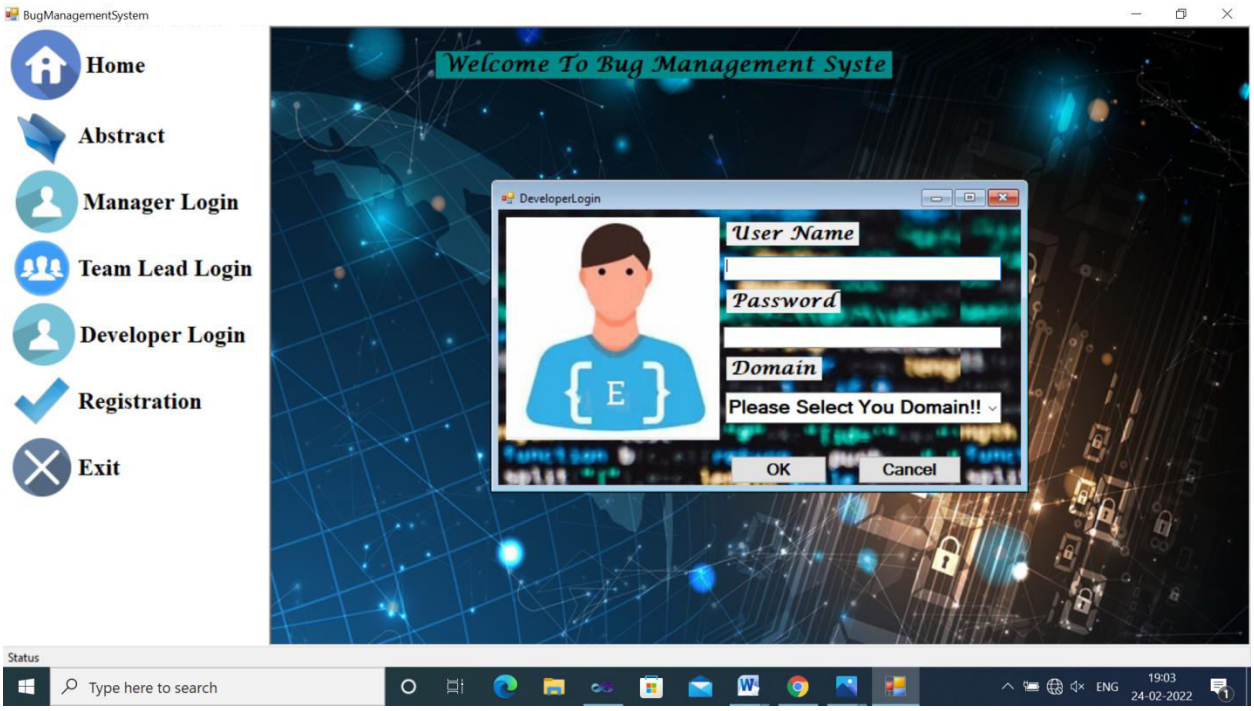
8.5.3 Analyse Bug Report:



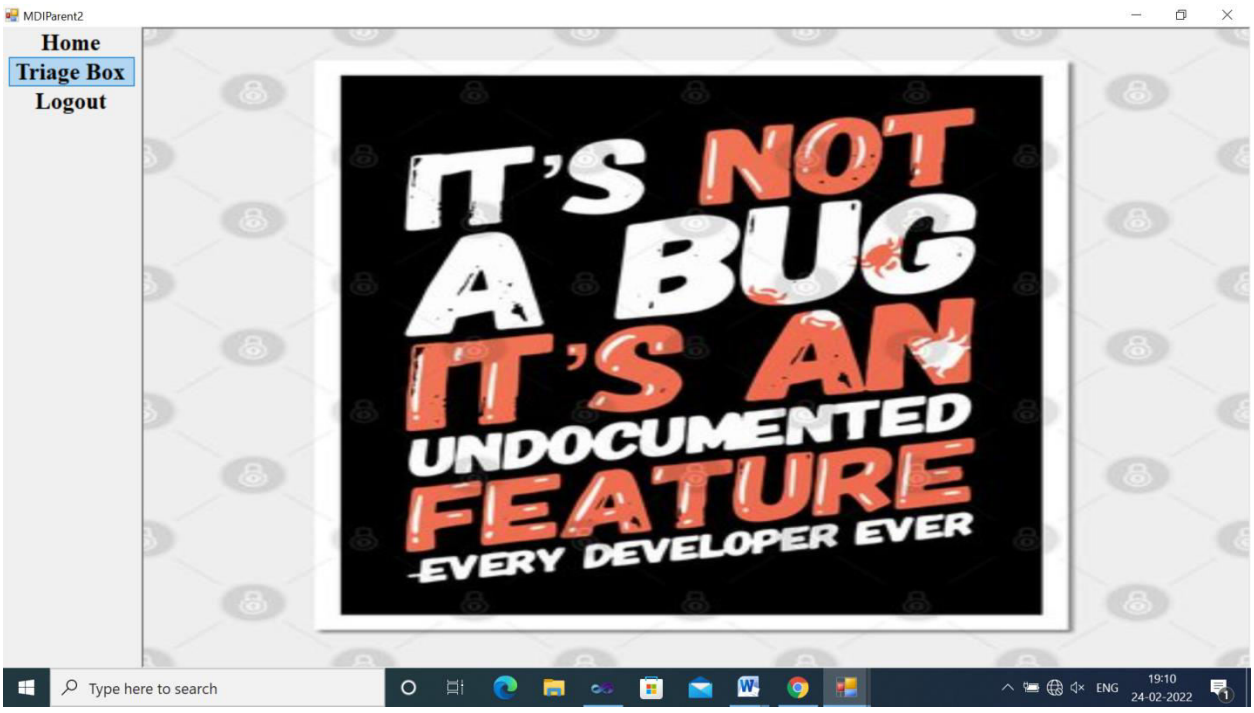
8.5.4 Data Reductoin Using Instance Selection:



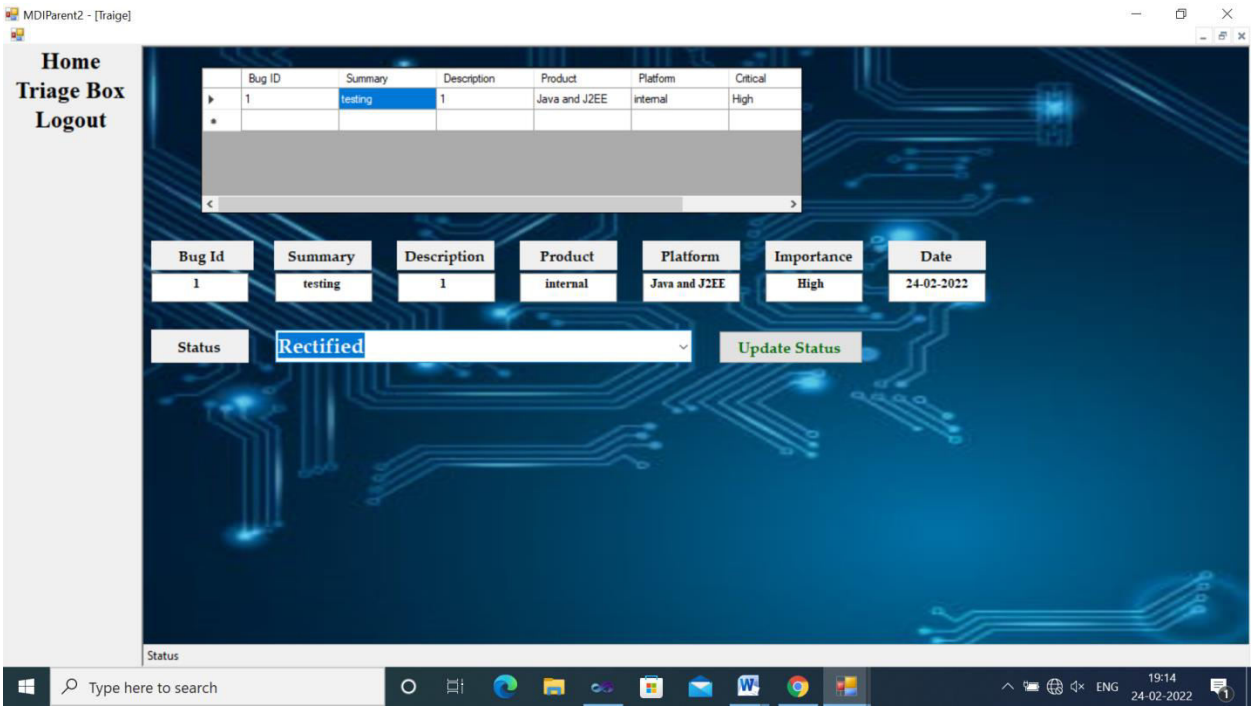
8.6 Developer Login



8.6.1Developer MainPage



8.6.2 Traige Box



CHAPTER 9

Conclusion

It has been a great pleasure for us to work on this project. We started this project with the intention of giving our best and doing something interesting and we believe that we have successfully achieved that. If we were able to complete this project, this is largely due to the collective efforts.

This project proved good for us as it provided practical knowledge of not only programming in Visual Basic.NET application and know some extent windows application and SQL server.

This DBMS package is very efficient and user friendly and way to store the details that comes to existence. Although all features are not provided the important ones need to make this software self-sufficient are present. The project has been successful in that manner.

The tables that are created for this software have been designed in such a way that complete information is given in a concise manner. Few of the functions provided by this software are new records, saved and modified.

The software is portable and flexible for future enhancements. Thus, this project helps in retrieving the details as single go using data base concept.

It is concluded that the objective of the project work entitled " BUG MANAGEMENT SYSTEM" filled and desired result has been obtained. Each program in the project are functioning normally and reliable to user

.

CHAPTER 10

BIBLIOGRAPHY

Books:

- Chitra Ravi "Database Management System" Subhash publications.
- Ashwini S Divakar "Software Engineering" Skyward publications.

Online References:

- www.tutorialpoint.com
- www.wikipedia.com
- www.google.com
- www.slideshare.com