ABSTRACT

Bug-Tracking mechanism is employed only is some of the large software development houses. Most of the others never bothered with bug tracking 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-Tracking System is an ideal solution to track the bugs of a product, solution or an application. Bug Tacking System allows individual or groups of developers to keep track of outstanding bugs in their product effectively. This can also be called as Defect Tracking System.

The Bug Tracking 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 Tracking. 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:

Designing frontend	:	JSP and Servlets
Backend	:	My SQL
Scripting	:	Java Script
UML	:	Rational Rose
IDE	:	My Eclipse
Web Server	:	Tomcat

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:

Processor	:	Pentium IV
RAM	:	512MB RAM
Hard Disk	:	10GB

CHAPTER 3

3. Literature Survey

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

The Bug Tracking 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-tracking mechanism is employed only in some of the large software development houses. Most of the others never bothered with bug tracking 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 tracking. 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. Employee

2. Manager

3. Administrator

4.5 Module Description

Employee

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

Administrator

Administrator 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).

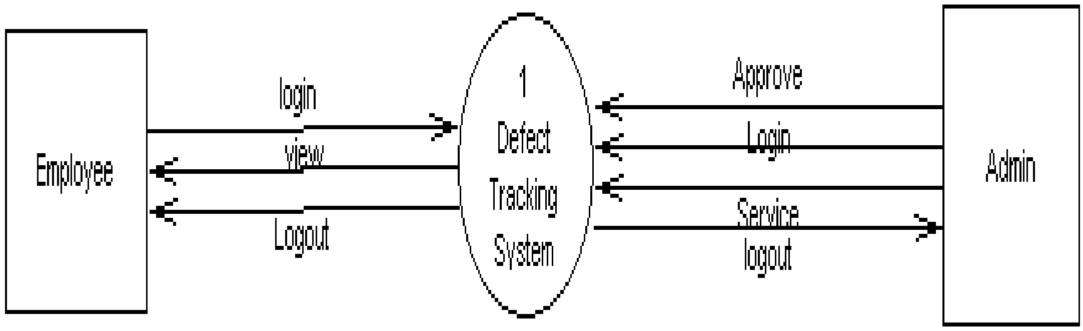


Fig. 5.1 Context Level DFD

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.

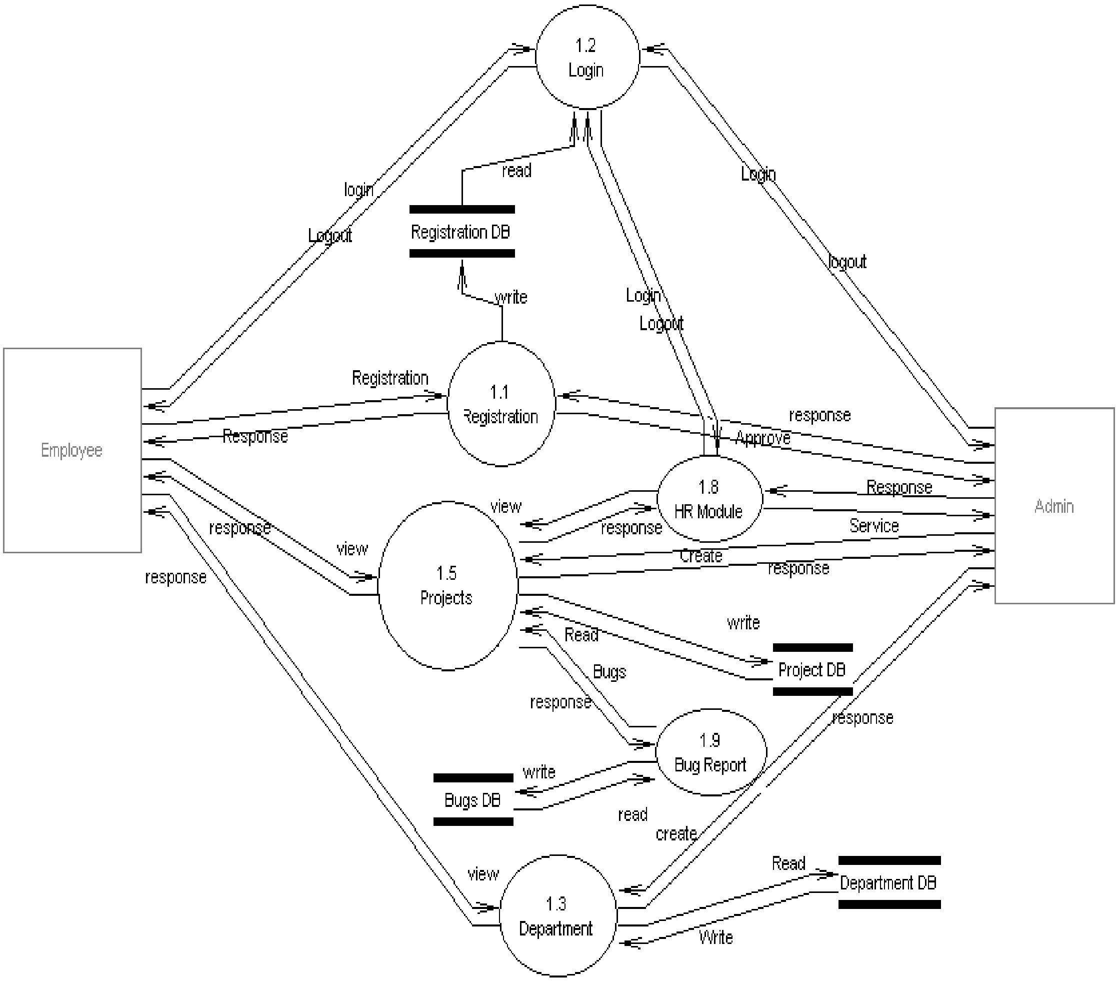


Fig. 5.2 Top Level DFD

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

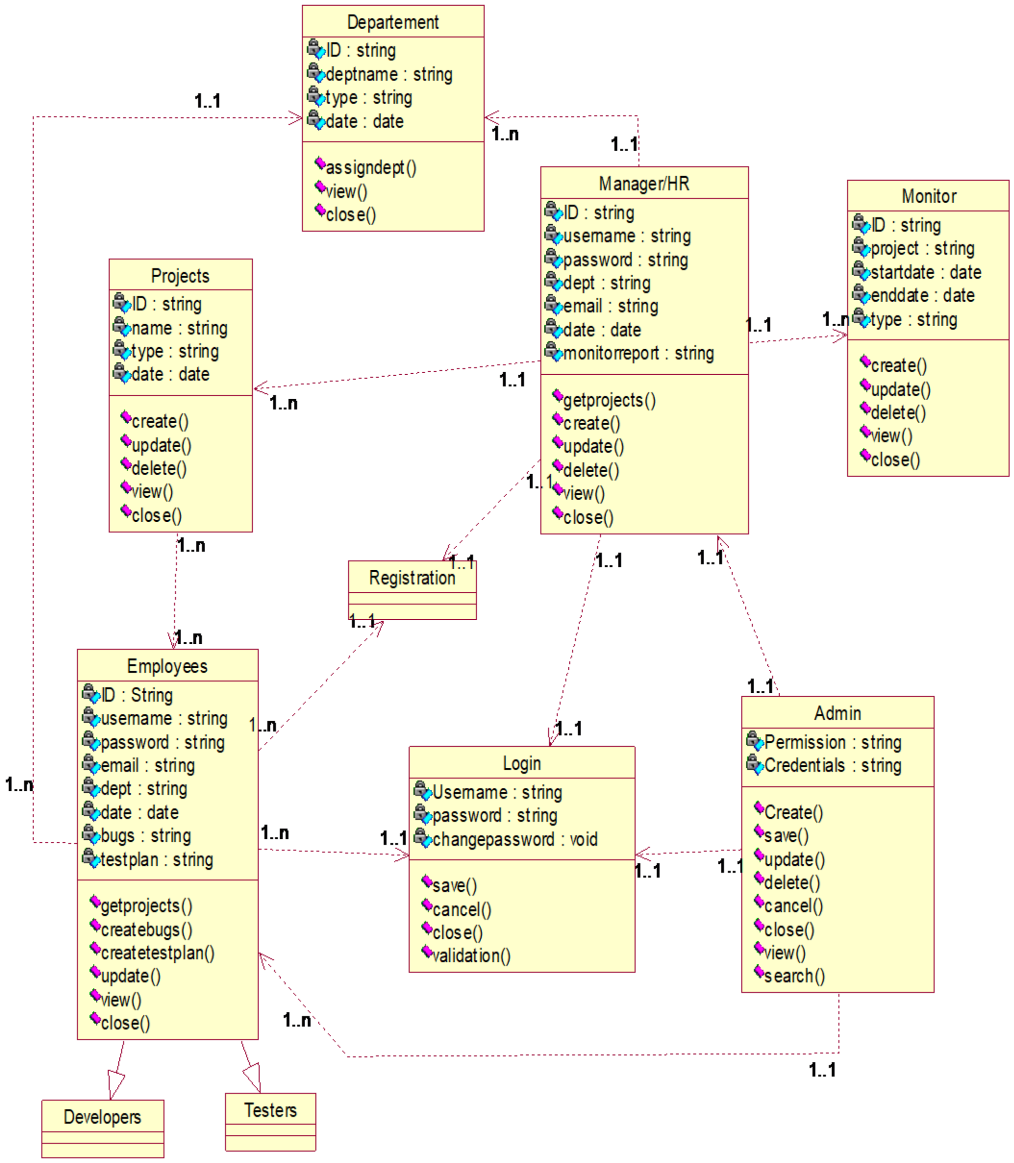


Fig 5.3 Class Diagram

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.

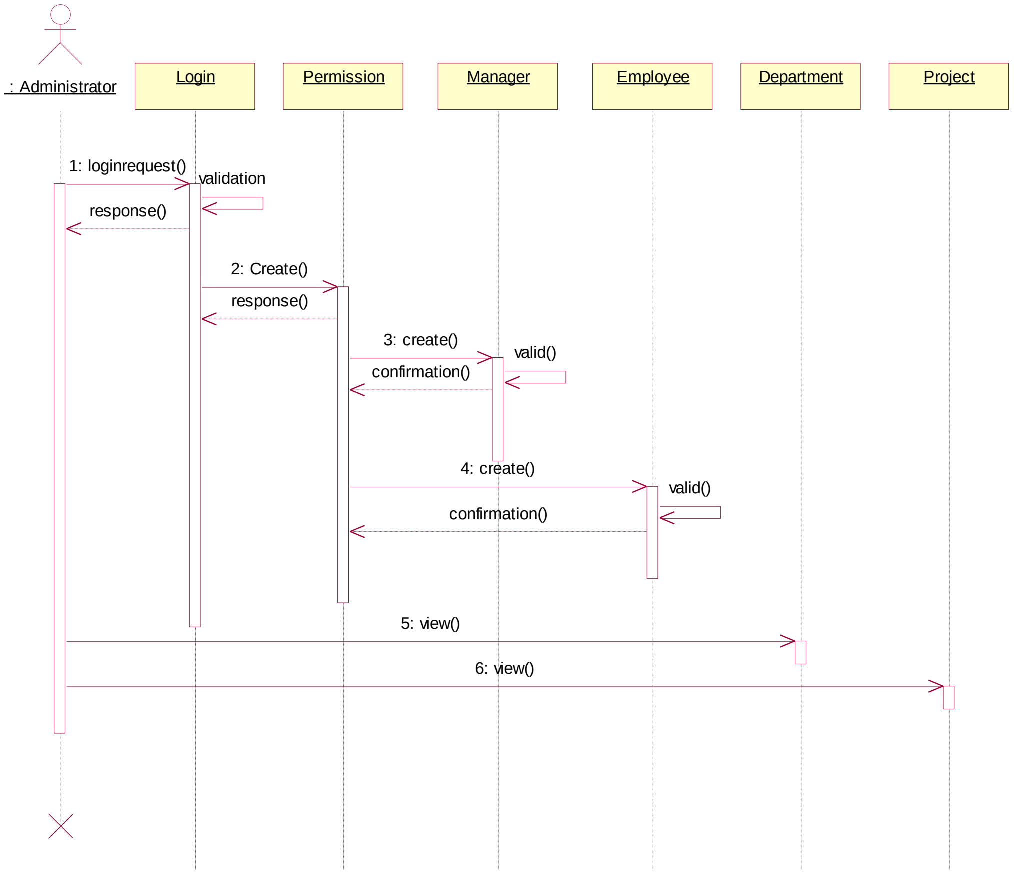


Fig 5.4 Administration Sequence Diagram

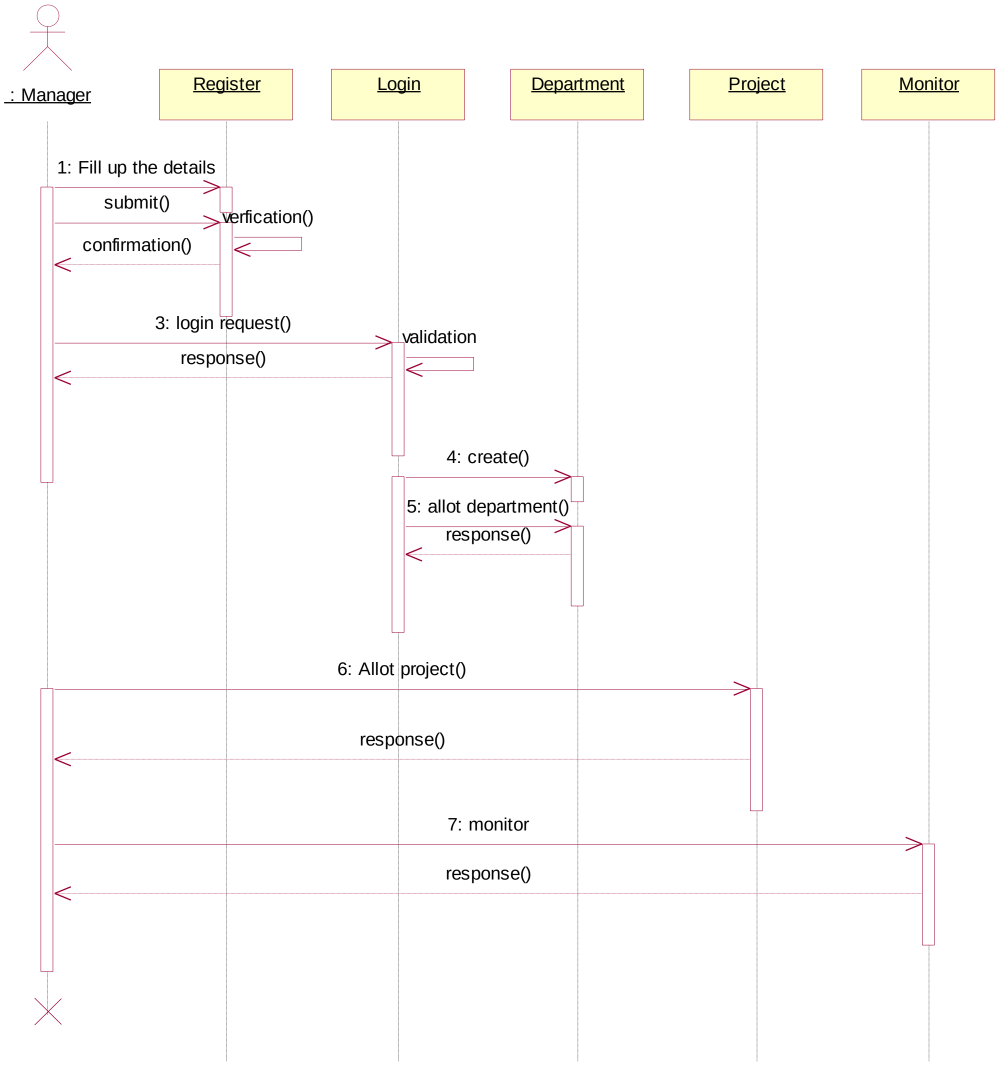


Fig 5.5 Manager Sequence

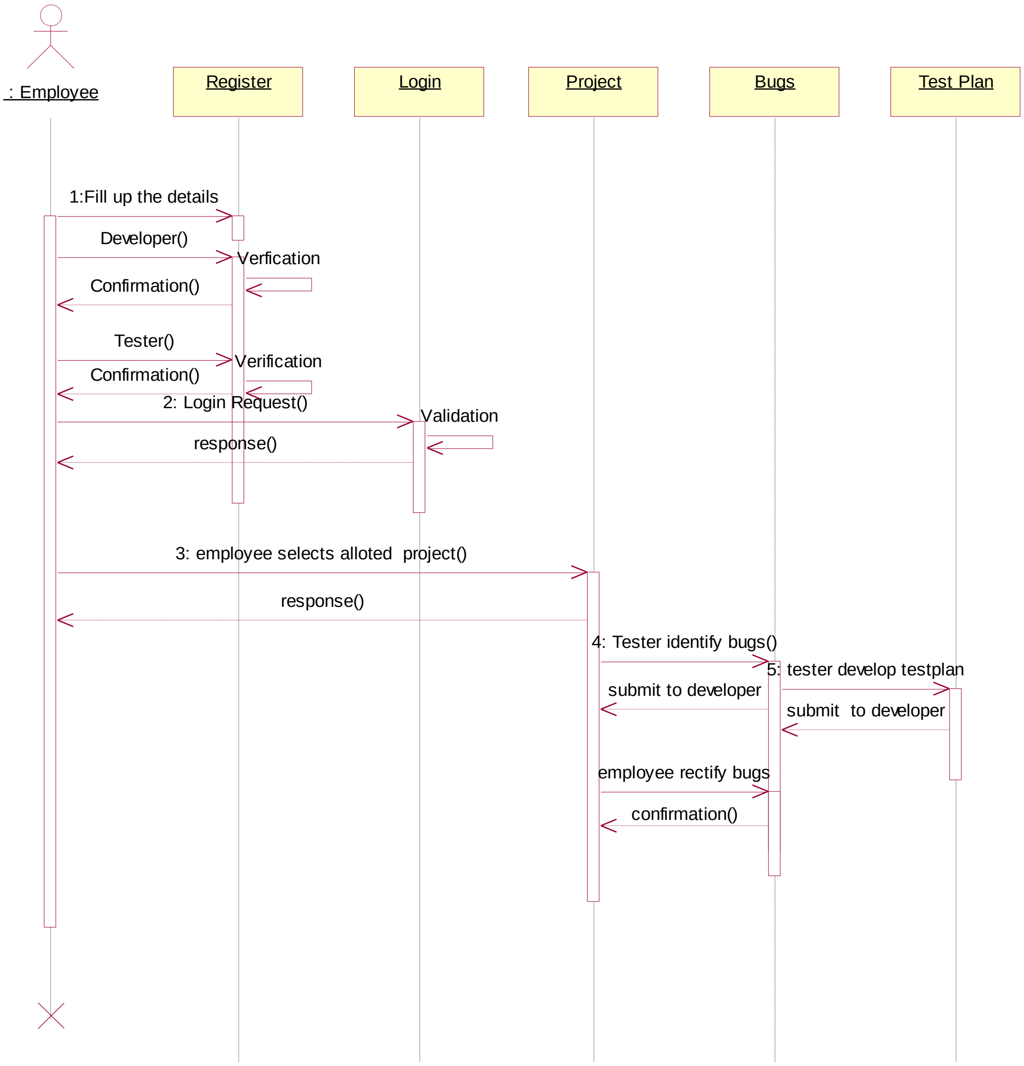


Fig 5.6 Employee Sequence

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.

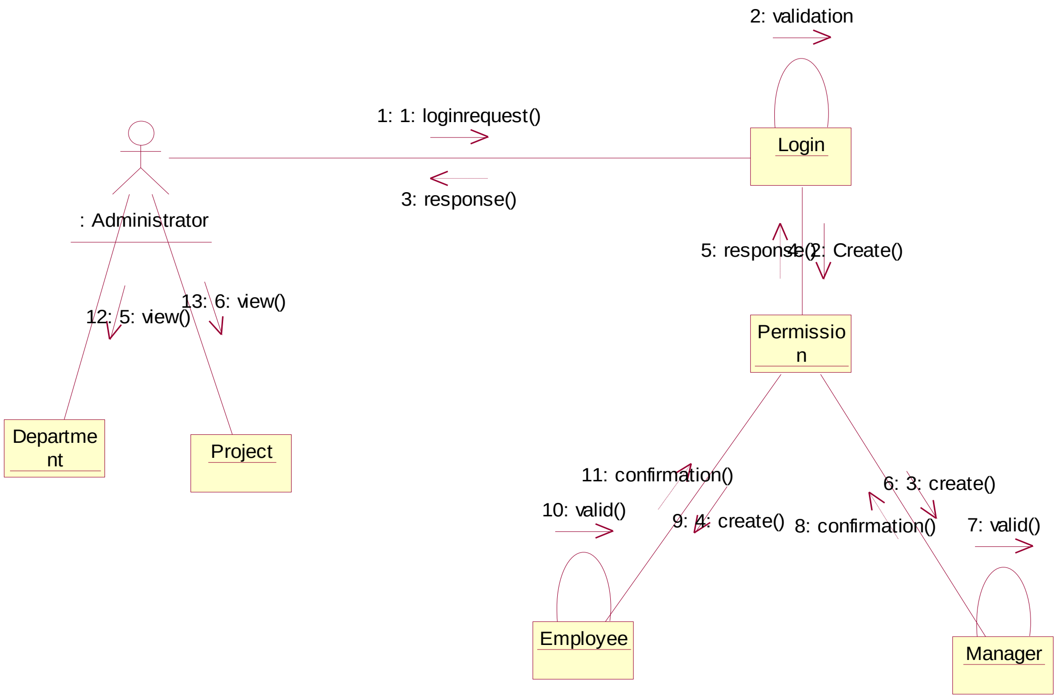


Fig 5.7 Administrative Collaboration

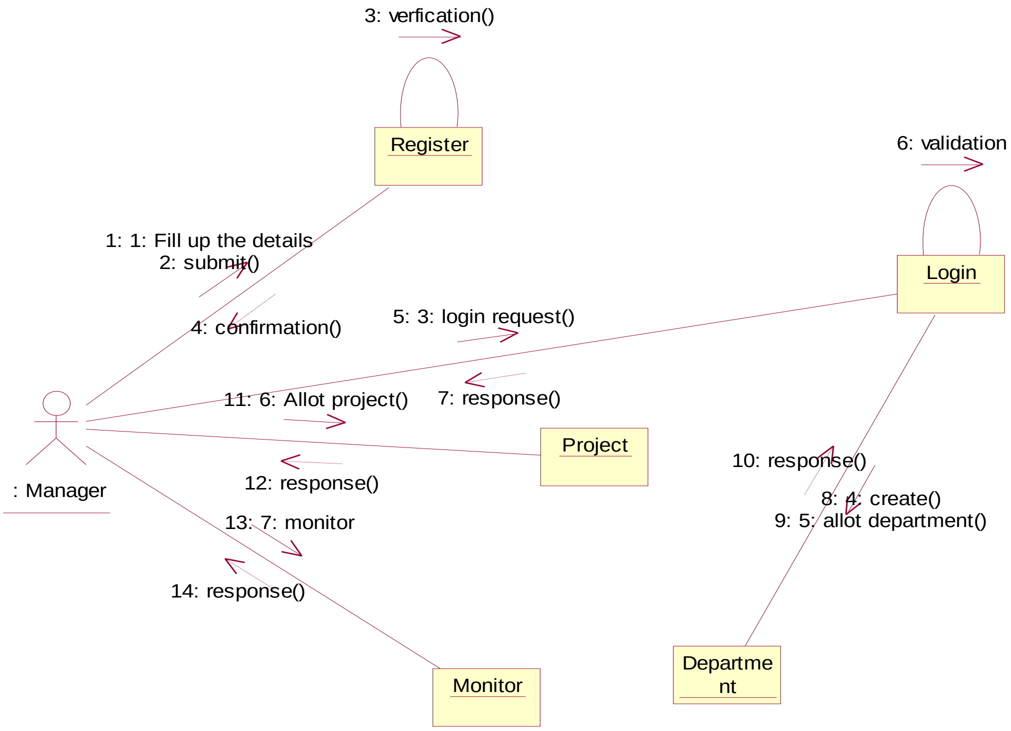


Fig 5.8 Manager Collaboration

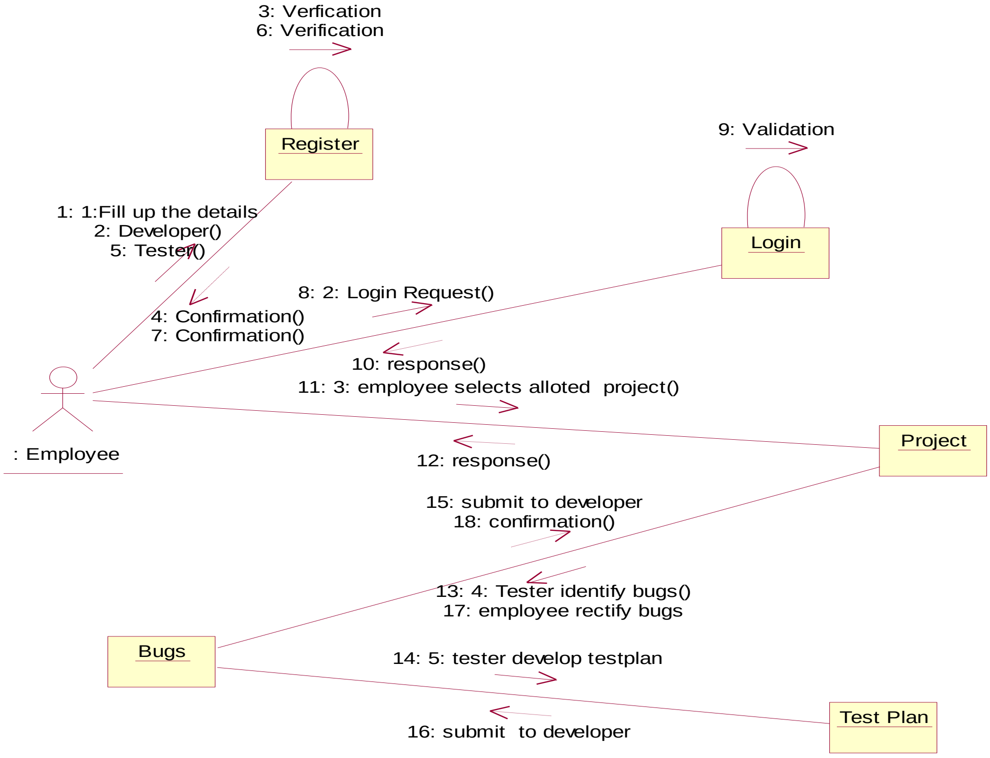


Fig 5.9 Employee collaboration

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.

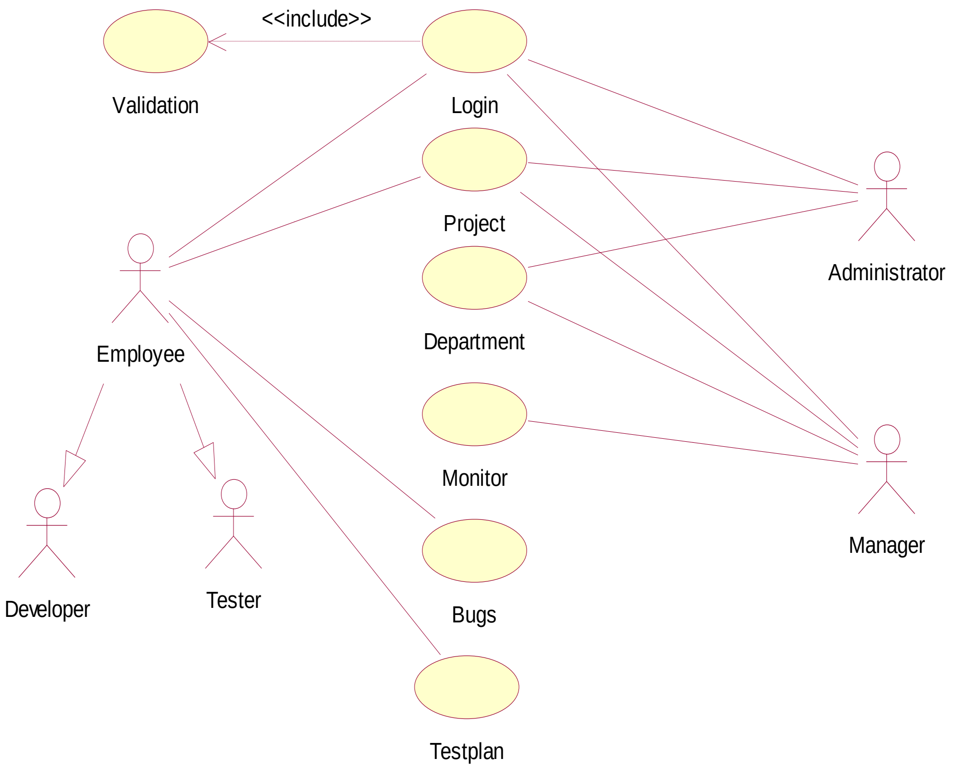


Fig 5.10 Overall Use case Diagram

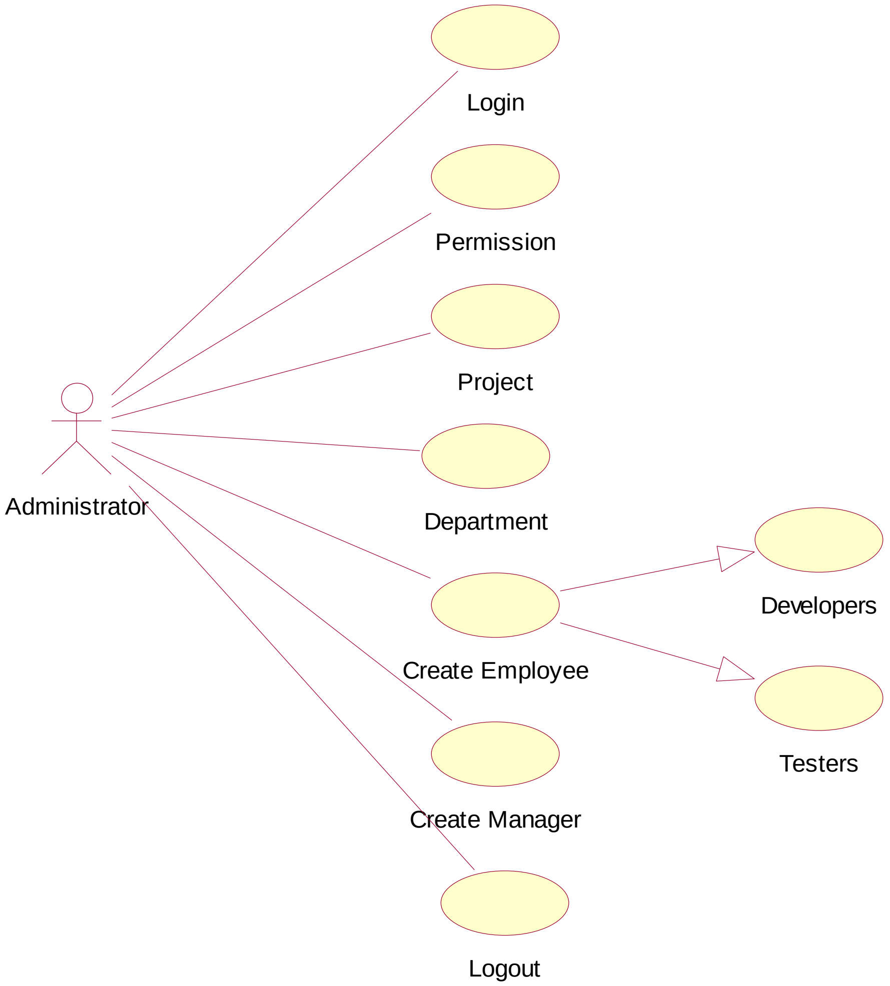


Fig 5.11 Administrator Use case

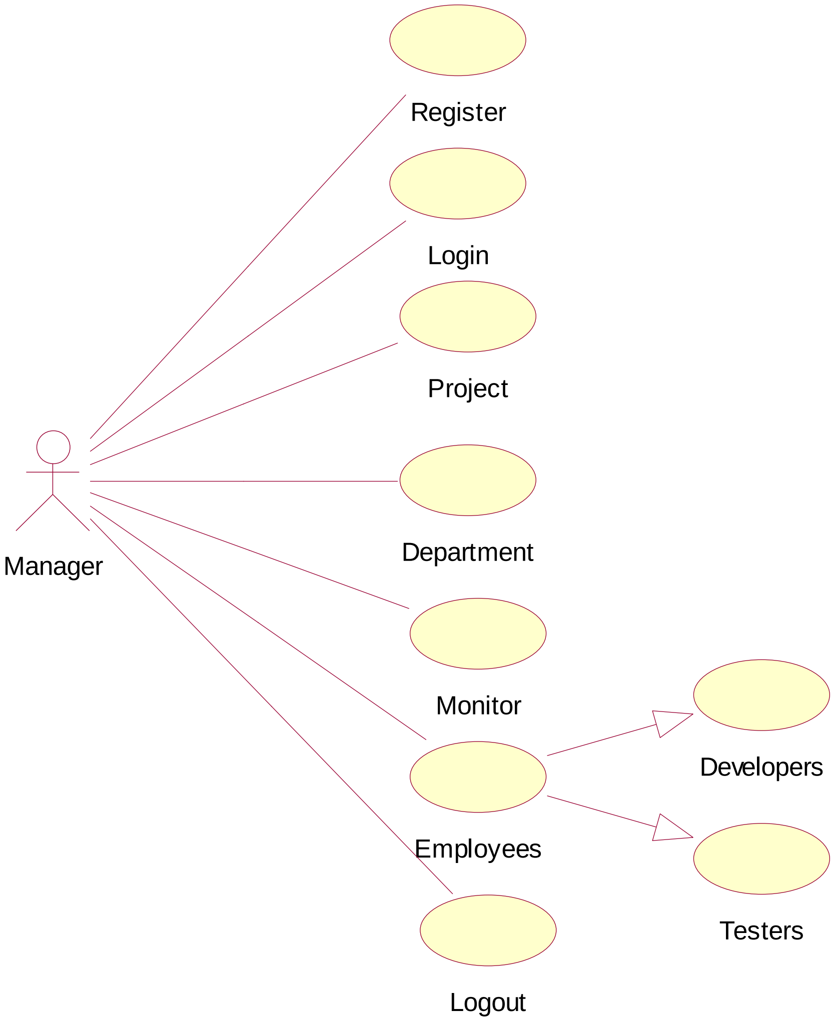


Fig 5.12 Manager Use case

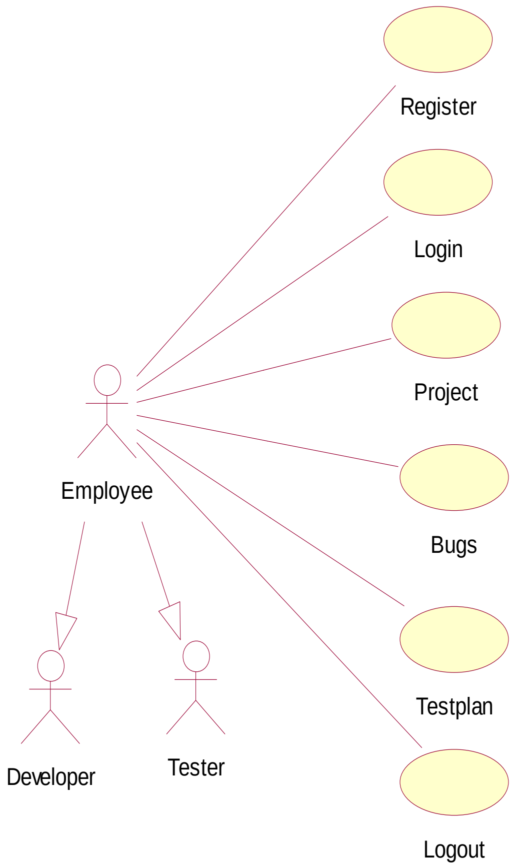


Fig 5.13 Employee use case

5.2.4 Component Diagram

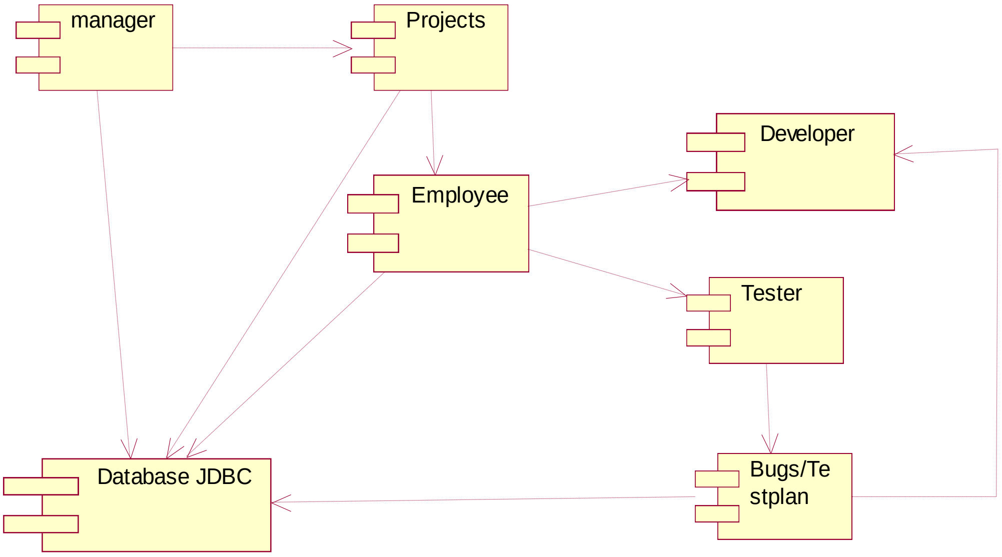


Fig 5.14 Component Diagram

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.

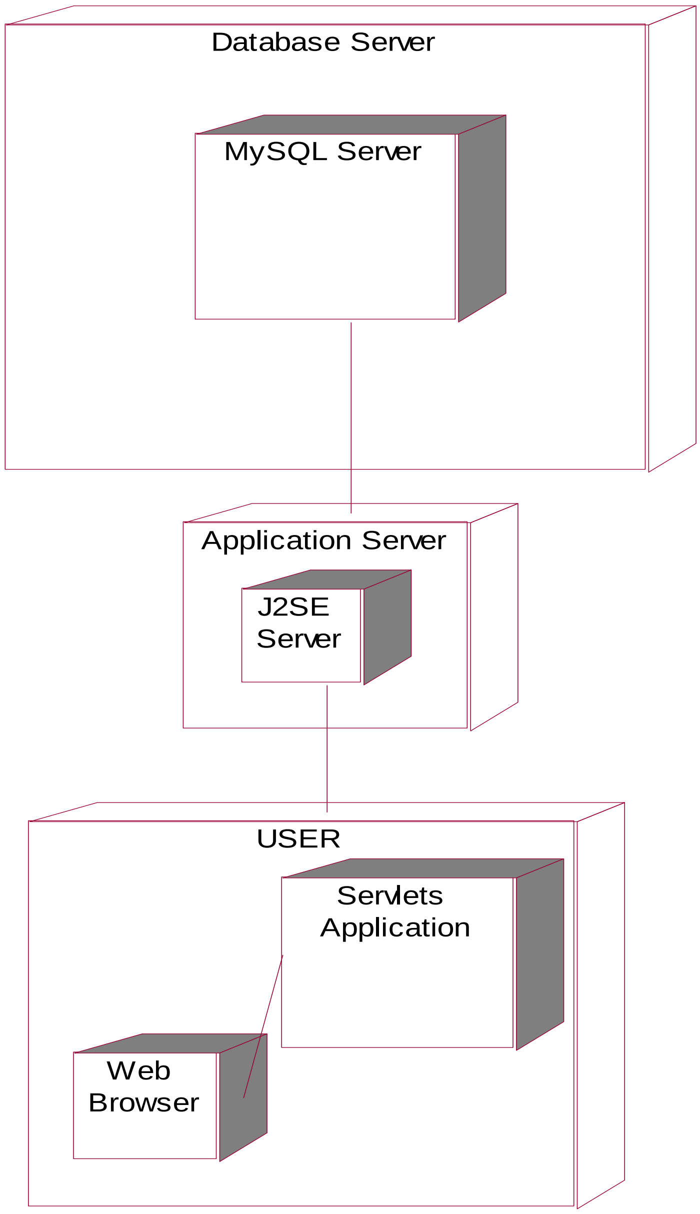


Fig 5.15 Deployment Diagram

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

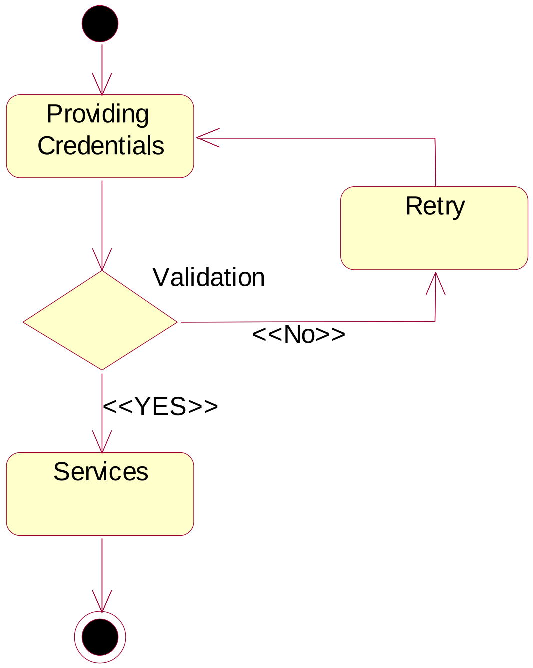


Fig 5.16 Login Activity Diagram

Registration Process

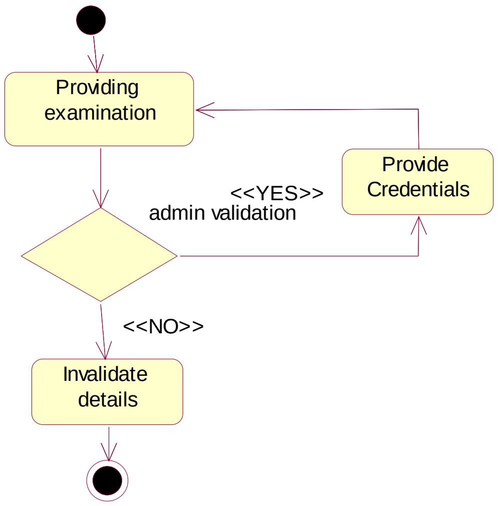


Fig 5.17 Registration Activity Diagram

Manager Process

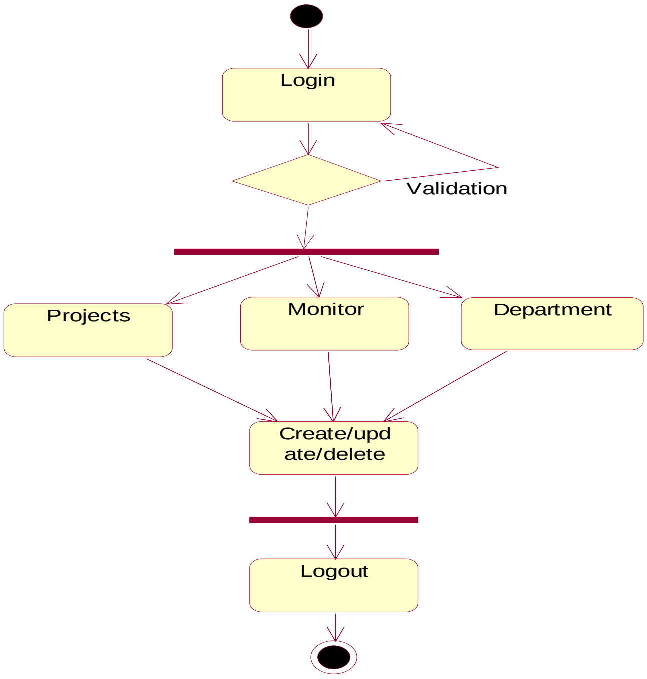


Fig 5.18 Manager Activity Diagram

Administrator Process

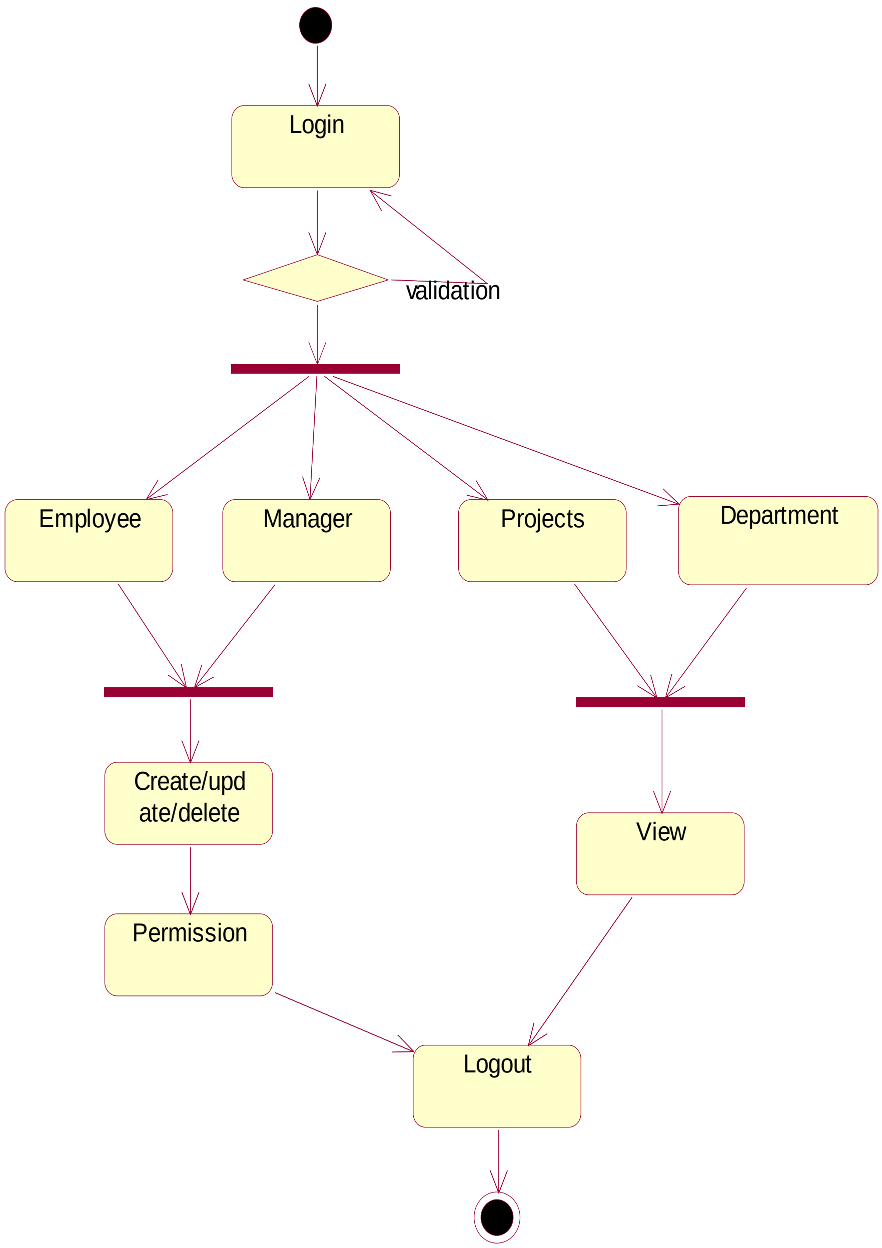


Fig 5.19 Administrator Process

Employee Process

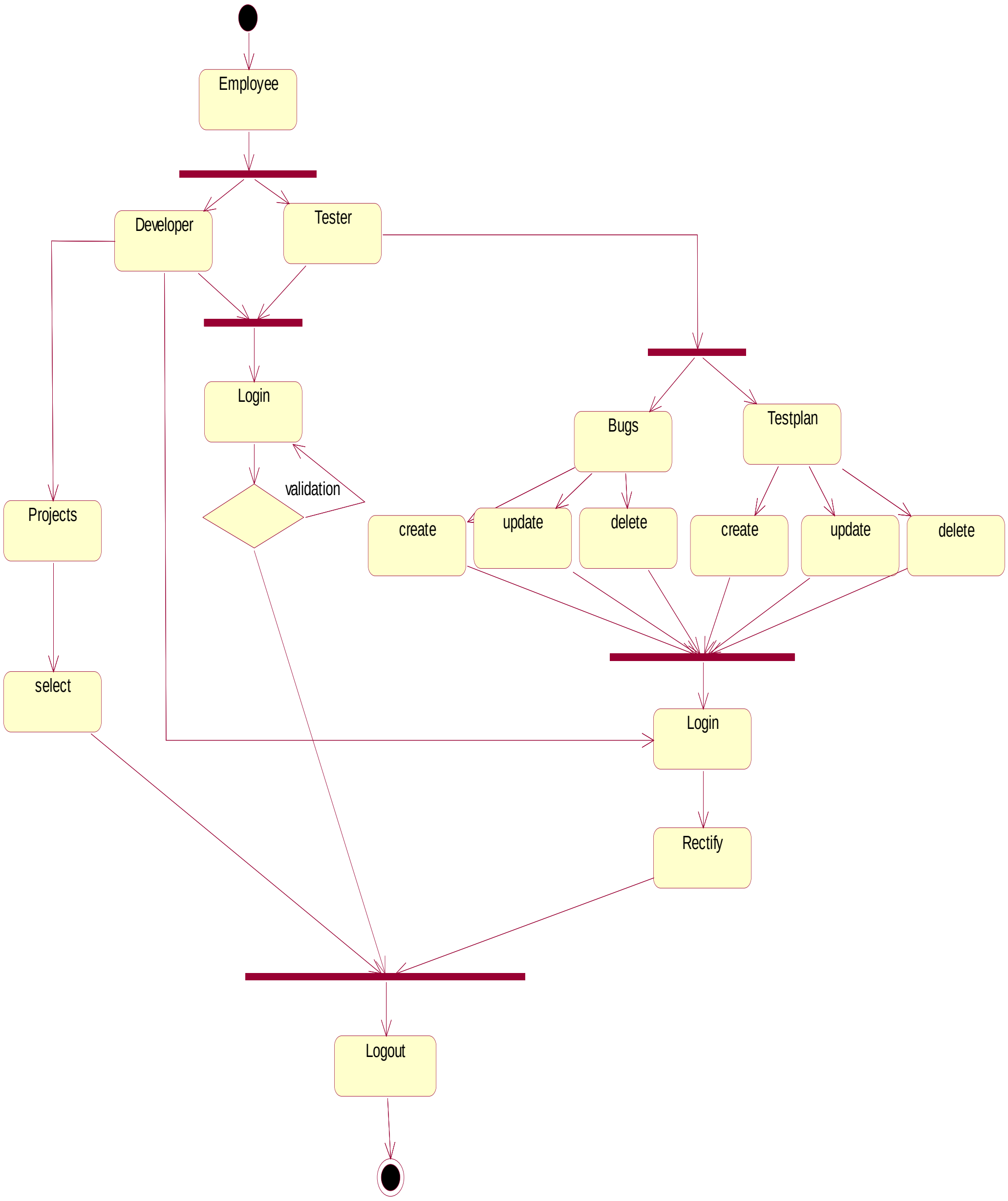


Fig 5.20 Employee Process

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.

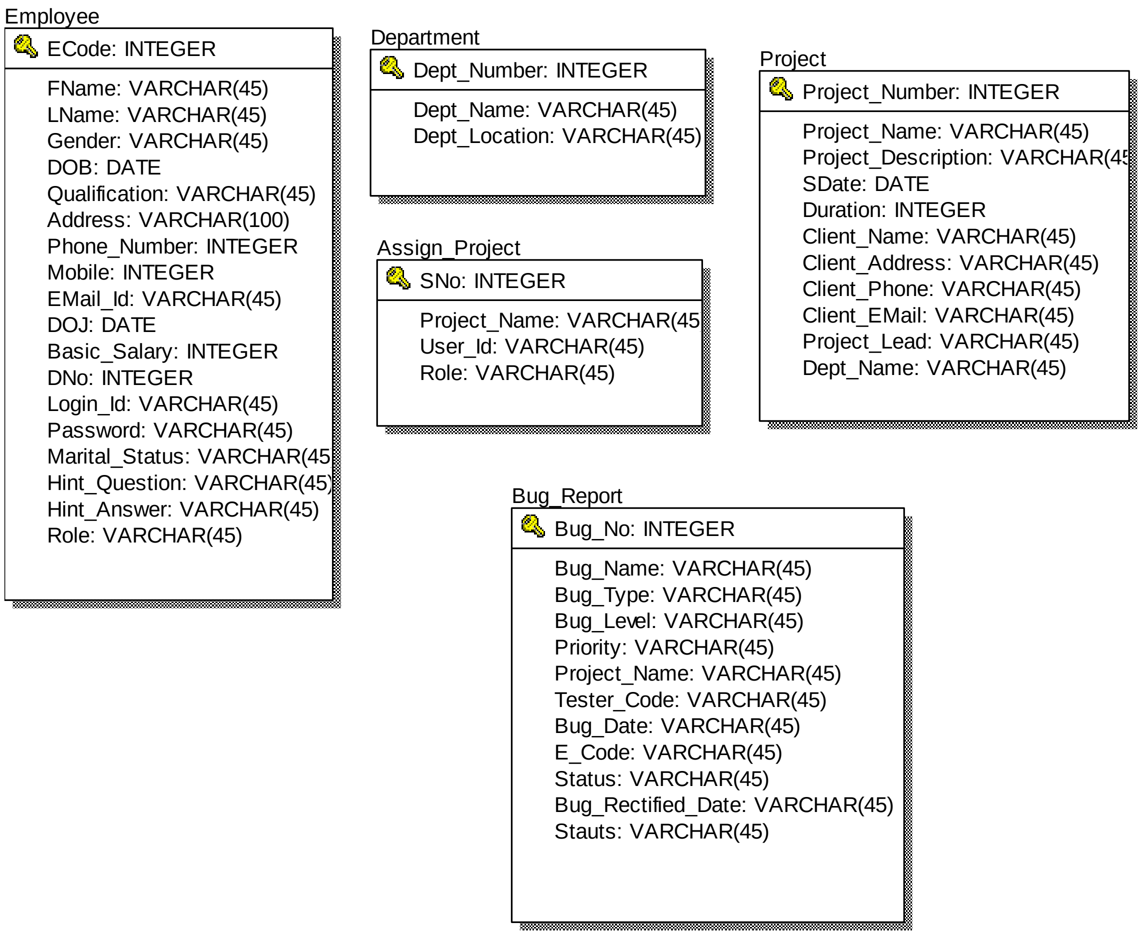


Fig 5.21 ER-Diagram

CHAPTER 6

6. CODING

6.1 General

A programming tool or software tool is a program or application that software developers use to create, debug, maintain, or otherwise support other programs and applications. The term usually refers to relatively simple programs that can be combined together to accomplish a task. The Chapter describes about the software tool that is used in our project.

- **6.2. Java Server Pages**

JSP not only enjoys cross-platform and cross-Web-server support, but effectively melds the power of server-side Java technology with features of static HTML pages.
JSP pages typically comprise of: Static HTML / XML components.
-Special JSP tags.
-Optionally, snippets of code written in the java programming language called "script lets."

- **JSP Advantages**

Separation of static from dynamic content: In JSP, the logic to generate the dynamic content is kept separate from the static presentation templates by encapsulating it within external Java beans components. When a page designer makes any changes to the presentation template, the JSP page is automatically recompiled and reloaded into the web server by the JSP engine.
Write Once Run Anywhere: JSP technology brings the "Write Once, Run anywhere" paradigm to interactive Web pages.
Dynamic content can be served in a variety of formats: There is nothing that mandates the static template data within a JSP page to be of a certain format.

- **JSP Architecture:**

The purpose of JSP is to provide a declarative, presentation-centric method of developing servlets. JSP pages are subject to a translation phase and a request-processing phase. The translation phase is carried phase is carried out only once, unless the JSP page changes, in which case it is repeated. The JSP engine itself typically carries out the translation phase, when it receives a request for the JSP page for the first time.

Life Cycle of A JSP: Life cycle of a JSP consists of the following three methods:

_jspInit

_jspService

_jspDestroy

- **JSP Syntax**

- **Directives**

JSPs can define information for the container with directives. Here is what directives look like in a general form:

```
<%@ directive attribute="some Value" attribute="another Value" ... %>
```

There are three directives:

<%@ page ... %> specifies information that affects the page

- <%@ include ... %> includes a file at the location of the include directive (parsed)
- <%@ taglib ... %> allows the use of custom tags in the page

<%@ page language="java" %> will always be the first line of every JSP file.

- **Declarations**

Declarations are used to specify supplemental methods and variables. You can think of these are the page's private functions; they can only be called by the JSP where they are defined, or by another JSP that includes it (using the <@ include > directive).

Here is a sample declaration:

```
<%! // this integer can be used anywhere in this JSP page
private int myVariable = -1; // this function can be called from anywhere in this JSP page
public boolean isPositive() {
   return ( myVariable > 0 );
}
%>
```

- **Scriptlets**

Scriptlets are bits of Java code. They can do anything but they will most likely concentrate on generating HTML code or setting up variables to be part of later expressions.

- **Expressions**

Expressions are special-purpose mini-Scriptlets used for evaluating expressions. This could be something as simple as outputting the value of a variable, or a more complicated Java expression, like calling a function and outputting the result. <%= counter %> Note that counter is defined as an int, but we do not need to explicitly convert it to a string.

6.3 Servlets

A servlet is a java programming language class that is used to extend the capabilities of servers that host applications access via a request-response programming mode. Servlets are Java technology's answer to Common Gateway Interface (CGI) Programming. They are programs that run on a Web server, acting as middle layer between request coming from a Web browser or other HTTP client and databases of applications on the HTTP server.

Read any data sent by the user: This data usually entered in a form on a Web page, but could also come from a java applet or a custom HTTP client program.

Look up any other information about the request that is embedded in the HTTP request: This information includes details about browser capabilities, cookies, the host name of the requesting client, and so froth.

Generate the results: This process may require talking to a database, executing an RMI or CORBA call, invoking a legacy application, or computing the response directly.

Format the results inside a document: In most cases, this involves embedding the information inside an HTML page.

Set the appropriate HTTP response parameters: This means telling the browser what type of document is being returned (e.g.HTML), setting cookies and caching parameters, and other such tasks.

Send the document back to the client: This document may be sent in text format (HTML), binary format (GIF images), or even in a compressed format like gzip that is layered on top of some other underlying format.

The Javax.servlet and javax.servlet.http packages provide interfaces and classes for writing servlets. All servlets must implement the Servlet interface, which defines life-cycle methods. When implementing a generic service, you can use or extend the GenericServlet class provided with the java Servlet API. The HttpServlet classes provide methods, such as doGet and do Post, for handling HTTP-specific services.

To be a servlet, a class should extend HTTPServlet and override doGet or do Post (or both), depending on whether the data is being sent by GET or by POST. These methods take two arguments: An HttpServletRequest and an HttpServletResponse.The HttpServletRequest have methods that let you find out about incoming information such as FORM data, HTTP request headers, and the like. Finally, note that doGet and do Post are called by the service method, and sometimes you may want to override service directly.

Servlet Life Cycle: The life cycle of a servlet is controlled by the container in which the servlet has been deployed. When a request is mapped to a servlet, the container performs the following steps.

1. If an instance of the servlet does not exist, the Web container:

Loads the servlet class.

Creates an instance of the Servlet class.

Initializes the servlet instance by calling the init method.

2. Invokes the service method, passing request and response objects.
If the container needs to remove the servlet, it finalizes the servlet by calling the servlet's destroy method.

- **Cookies**

Cookies are small bits of textual information that a Web server sends to a browser and that the browser returns unchanged when visiting the same Web site or domain later

Browsers generally only accept 20 cookies per site and 300 cookies total, and each cookie is limited to 4KB, cookies cannot be used to fill up someone's disk or launch other denial of service attacks.

- **The Servlet Cookie API**

To send cookies to the client, a servlet would create one or more cookies with the appropriate names and values via new Cookie (name, value), set any desired optional attributes via cookie.setXxx,and add the cookies to the response headers via response.addCookie(cookie).To read incoming cookies, call request.getCookies(), which returns an array of Cookie objects.

Session Management

Many applications require that a series of requests from a client be associated with one another. Sessions are represented by an Http Session object. A session cab be accessed by calling the get Session () method of a request object. This method returns the current session associated with this request, or, if the request does not have a session, it creates one. The timeout period can be accessed by using a session’s [get\set] Max Inactive Interval methods.

Session Tracking

A Web container can use several methods to associate a session with a user, all of which involve passing an identifier between the client and the server. The identifier can be maintained on the client as a cookie, or the Web component can include the identifier in every URL that is returned to the client.

In fact, on many servers, they use cookies if the browser supports them, but automatically revert to URL-rewriting when cookies are unsupported or explicitly disabled.

- **The Session Tracking API**

Using sessions in servlets is quite straightforward, and involves looking up the session object associated with the current request, creating a new session object when necessary, looking up information associated with a session, storing information in a session, and discarding completed or abandoned sessions.

CHAPTER 7

7. TESTING

Software Testing is the process used to help identify the correctness, completeness, security and quality of developed computer software. Testing is a process of technical investigation, performed on behalf of stakeholders, that is intended to reveal quality-related information about the product with respect to the context in which it is intended to operate. In general, software engineers distinguish software faults from software failures. Our project" Defect Tracking System" is tested with the following testing methodologies.

The test process begins by developing a comprehensive plan to test the general functionality and special features on a variety of platform combinations. Strict quality control procedures are used. The process verifies that the application meets the requirements specified in the system requirements document and is bug free. The following are the considerations used to develop the framework for developing the test methodologies.

The process of executing a system with the intent of finding an error. Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.

Testing

- **Testing Methodologies**
  - Black box Testing:
  - White box Testing.
  - Gray Box Testing.

- **Levels of Testing**
  - Unit Testing.
  - Module Testing.

  - Integration Testing.
  - System Testing.
  - User Acceptance Testing.

- **Types Of Testing**
  - Smoke Testing.
  - Sanitary Testing.
  - Regression Testing.
  - Re-Testing.
  - Static Testing.
  - Dynamic Testing.
  - Alpha-Testing.
  - Beta-Testing.
  - Monkey Testing.
  - Ext….
- **TCD (Test Case Documentation)**
- **STLC**
  - Test Planning.
  - Test Development.
  - Test Execution.
  - Result Analysis.
  - Bug-Tracing.
  - Reporting.

- **Microsoft Windows – Standards**
- **Manual Testing**
- **Automation Testing (Tools)**
  - Win Runner.
  - Test Director

Testing:

- The process of executing a system with the intent of finding an error.
- Testing is defined as the process in which defects are identified, isolated, subjected for rectification and ensured that product is defect free in order to produce the quality product and hence customer satisfaction.
- Quality is defined as justification of the requirements
- Defect is nothing but deviation from the requirements
- Defect is nothing but bug.
- Testing --- The presence of bugs
- Testing can demonstrate the presence of bugs, but not their absence
- Debugging and Testing are not the same thing!
- Testing is a systematic attempt to break a program or the AUT
- Debugging is the art or method of uncovering why the script /program did not execute properly.

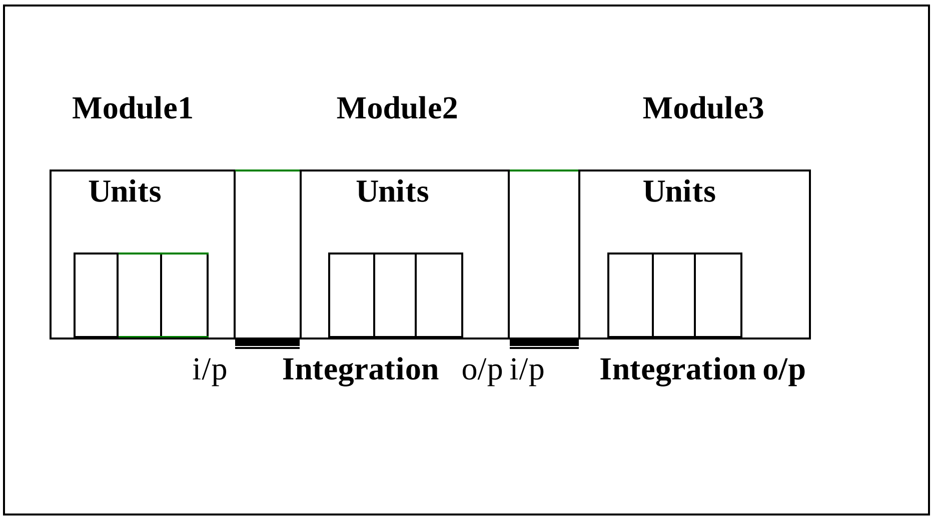
Testing Methodologies:

- **Black box Testing**: is the testing process in which tester can perform testing on an application without having any internal structural knowledge of application.
  Usually Test Engineers are involved in the black box testing.

- **White box Testing**: is the testing process in which tester can perform testing on an application with having internal structural knowledge.
  Usually The Developers are involved in white box testing.

- **Gray Box Testing**: is the process in which the combination of black box and white box techniques are used.

Levels of Testing



System Testing: Presentation + business +Databases

UAT: user acceptance testing

STLC (SOFTWARE TESTING LIFE CYCLE)

Test Planning: **1.**Test Plan is defined as a strategic document which describes the procedure how to perform various testing on the total application in the most efficient way.

2. Objective of testing,

3. Areas that need to be tested,

4. Areas that should not be tested,

5. Scheduling Resource Planning,

7. Areas to be automated, various testing tools used

Test Development: **1.** Test case Development (check list)

2. Test Procedure preparation. (Description of the test cases)

Test Execution: **1.** Implementation of test cases. Observing the result.

Result Analysis: **1.** Expected value: is nothing but expected behavior Of application.

2. Actual value: is nothing but actual behavior of the application

Bug Tracing: Collect all the failed cases, prepare documents.

Reporting: Prepare document (status of the application)

TCD (Test Case Document)

Test Case Document Contains

- **Test Scope (or) Test objective**
- **Test Scenario**
- **Test Procedure**
- **Test case**

This is the sample test case document for the Academic details of student project:

Test scope

- Test coverage is provided for the screen " Academic status entry" form of a student module of university management system application
- Areas of the application to be tested

Test Scenario

- When the office personals use this screen for the marks entry, calculate the status details, saving the information on student's basis and quit the form.

Test Procedure

- The procedure for testing this screen is planned in such a way that the data entry, status calculation functionality, saving and quitting operations are tested in terms of Gui testing, Positive testing, Negative testing using the corresponding Gui test cases, Positive test cases, Negative test cases respectively

Test Cases

- Template for Test Case

T.C.No	Description	Exp	Act	Result

Guidelines for Test Cases

1. **GUI Test Cases**

- Total no of features that need to be check
- Look & Feel
- Look for Default values if at all any (date & Time, if at all any require)
- Look for spell check

Example for GUI Test cases

T.C. No	Description	Expected value	Actual value	Result
1	Check for all the features in the screen	The screen must contain all the features		
2	Check for the alignment of the objects as per the validations	The alignment should be in proper way		

2. **Positive Test Cases**

- The positive flow of the functionality must be considered
- Valid inputs must be used for testing

- Must have the positive perception to verify whether the requirements are justified.

Example for Positive Test cases

T.C. No	Description	Expected value	Actual value	Result
1	Check for the date Time Auto Display	The date and time of the system must be displayed		
2	Enter the valid user id into the user id field	It should accept		

3. Negative Test Cases

- Must have negative perception.
- Invalid inputs must be used for test.

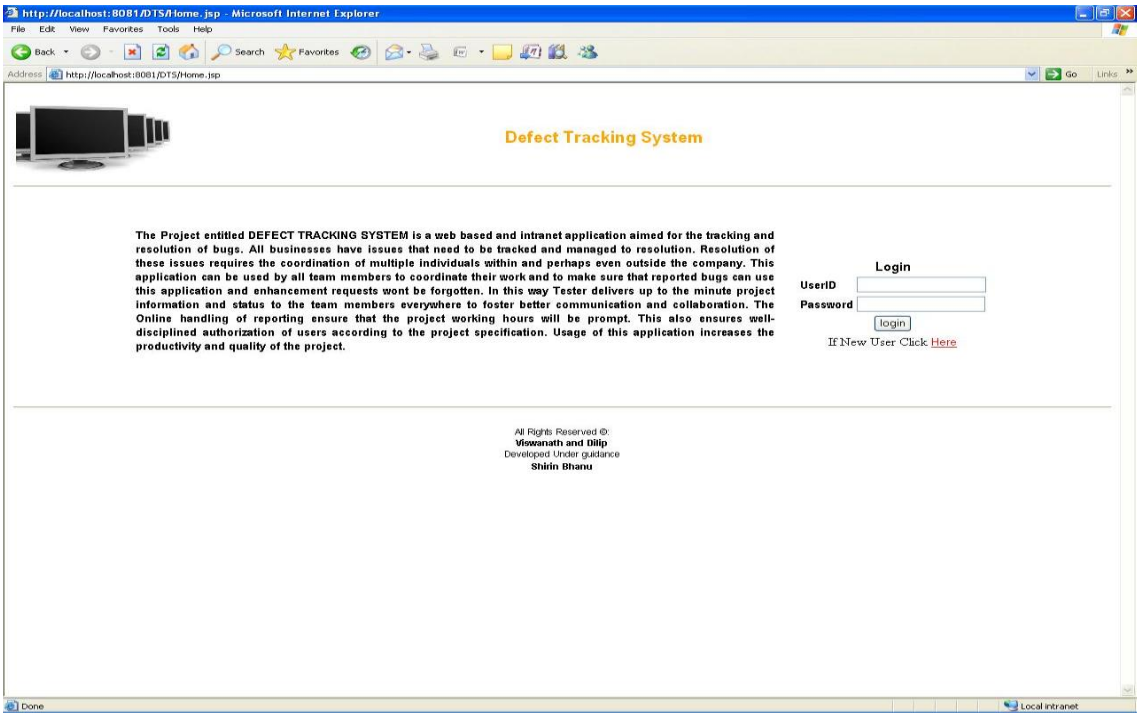
Example for Negative Test cases

T.C. No	Description	Expected value	Actual value	Result
1	Try to modify the information in date and time	Modification should not be allow		
2	Enter invalid data in to the login details form, click on Save	It should not accept invalid data, save should not allow		

CHAPTER 8

8. GRAPHICAL USER INTERFACE

8.1 Home Page



http://localhost:8081/DTS/Home.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/DTS/Home.jsp
Go Links
Done
Local intranet

Defect Tracking System

The Project entitled DEFECT TRACKING SYSTEM is a web based and intranet application aimed for the tracking and resolution of bugs. All businesses have issues that need to be tracked and managed to resolution. Resolution of these issues requires the coordination of multiple individuals within and perhaps even outside the company. This application can be used by all team members to coordinate their work and to make sure that reported bugs can use this application and enhancement requests wont be forgotten. In this way Tester delivers up to the minute project information and status to the team members everywhere to foster better communication and collaboration. The Online handling of reporting ensure that the project working hours will be prompt. This also ensures well-disciplined authorization of users according to the project specification. Usage of this application increases the productivity and quality of the project.

Login

UserID

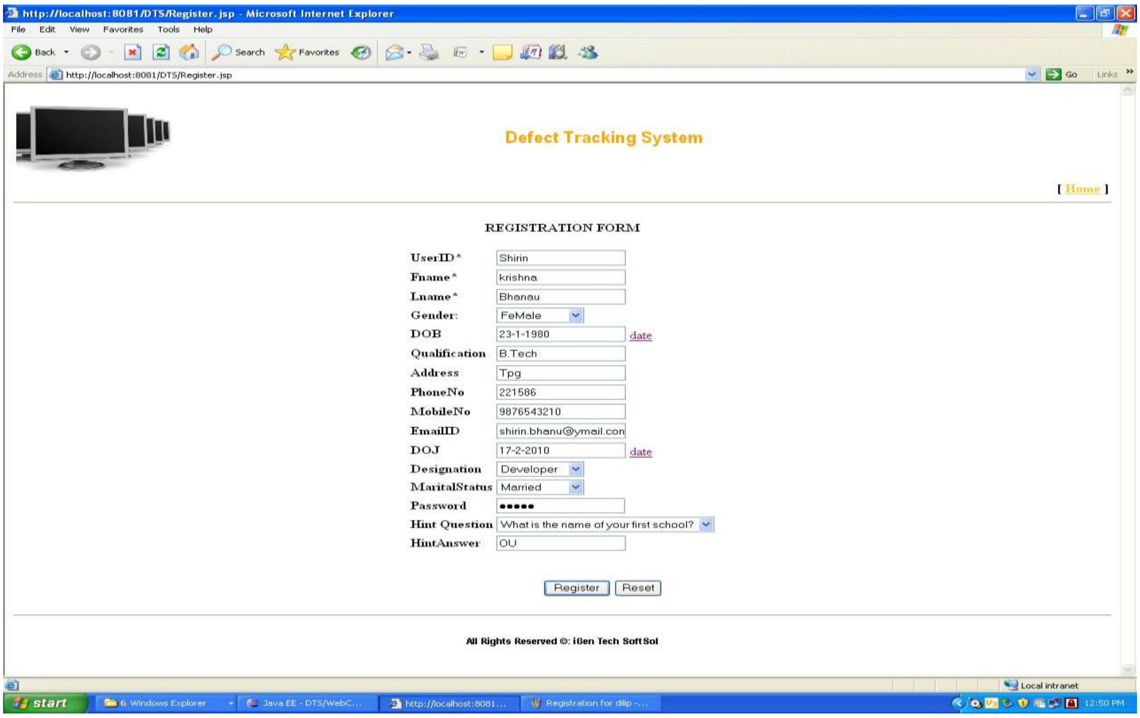
Password

login

If New User Click Here

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

8.2 Registration Form



http://localhost:8081/DTS/Register.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/DTS/Register.jsp
Go Links
Local intranet
start
6 Windows Explorer
Java EE - DTS/WebC...
http://localhost:8081...
Registration for dilip -...
12:50 PM

Defect Tracking System

[Home]

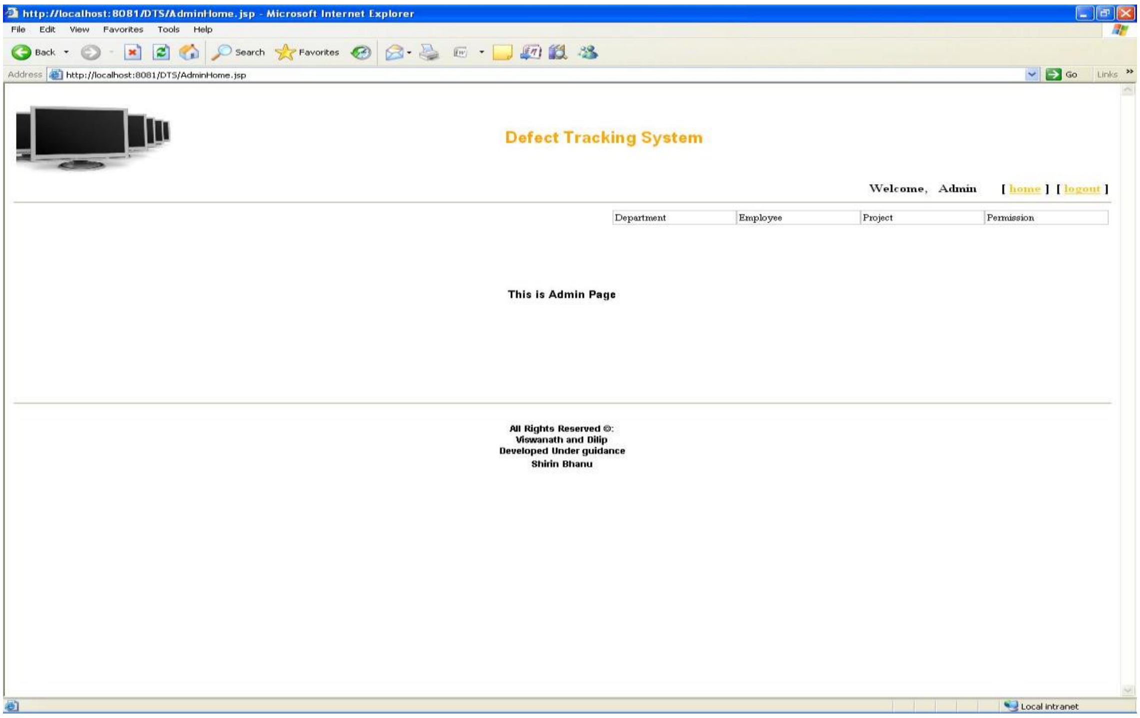
REGISTRATION FORM

UserID*	Shirin	
Fname*	krishna	
Lname*	Bhanau	
Gender:	FeMale	
DOB	23-1-1980	date
Qualification	B.Tech	
Address	Tpg	
PhoneNo	221586	
MobileNo	9876543210	
EmailID	shirin.bhanu@ymail.con	
DOJ	17-2-2010	date
Designation	Developer	
MaritalStatus	Married	
Password	●●●●●	
Hint Question	What is the name of your first school?	
HintAnswer	OU	

Register Reset

All Rights Reserved ©: iGen Tech SoftSol

8.3 Admin home



http://localhost:8081/DTS/AdminHome.jsp - Microsoft Internet Explorer

Address http://localhost:8081/DTS/AdminHome.jsp

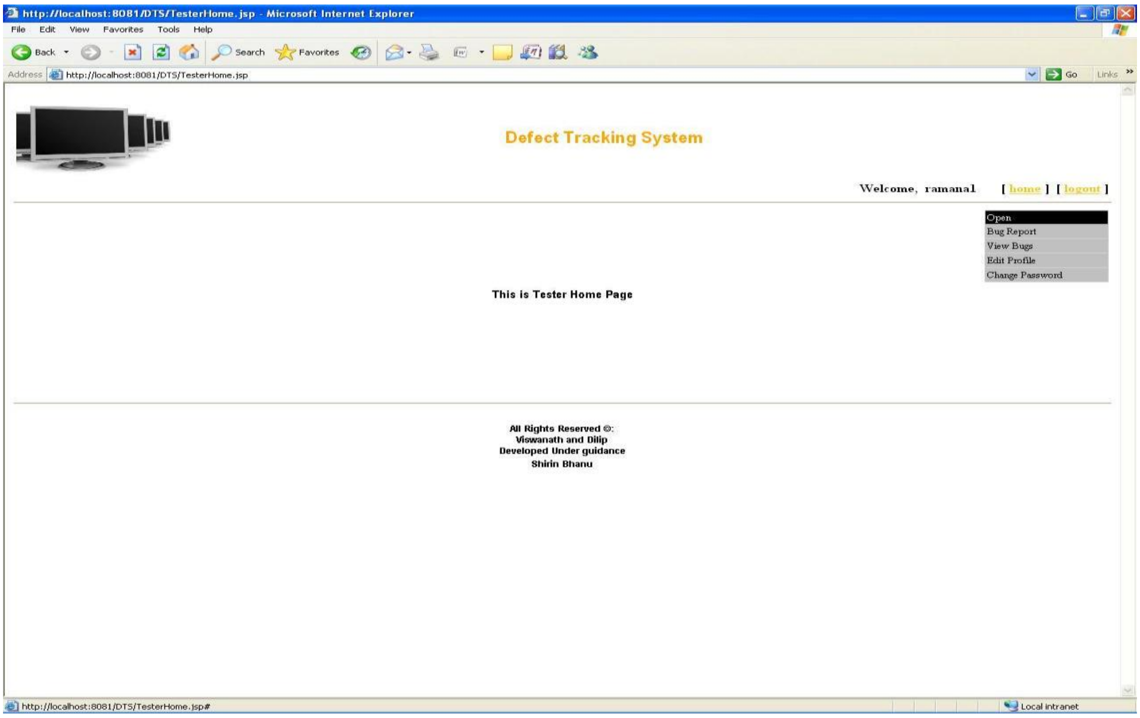
Defect Tracking System

Welcome, Admin [home] [logout]

This is Admin Page

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

8.4 Tester Home



http://localhost:8081/DTS/TesterHome.jsp - Microsoft Internet Explorer

Address http://localhost:8081/DTS/TesterHome.jsp

Defect Tracking System

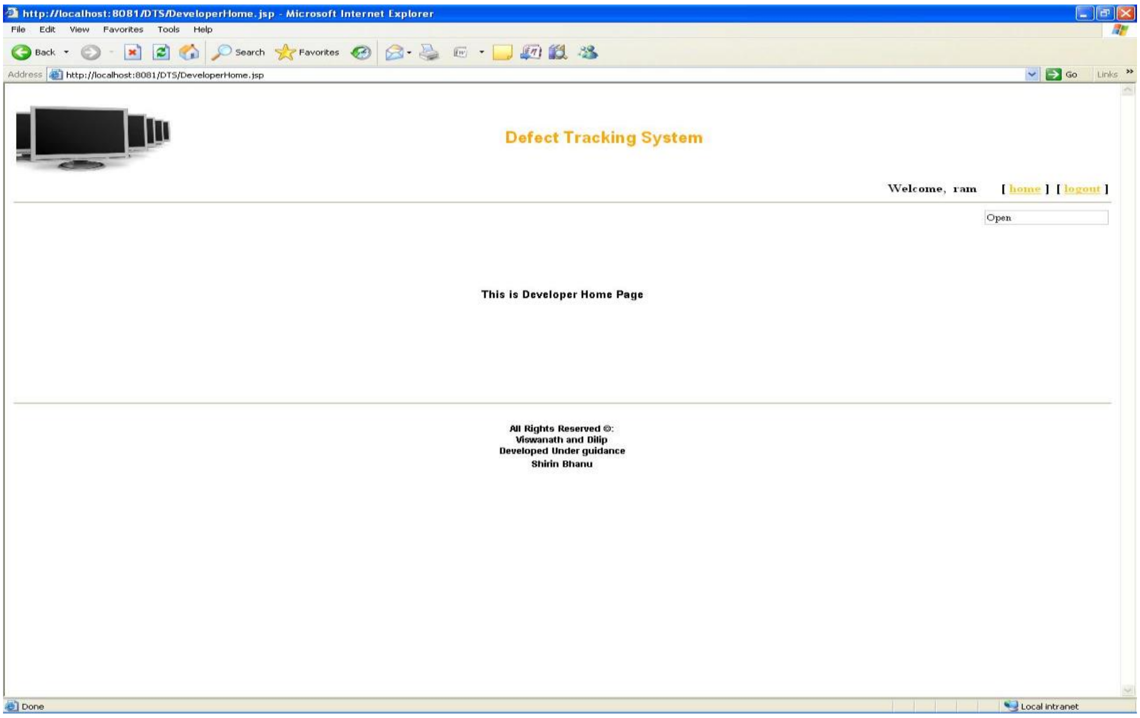
Welcome, ramanal [home] [logout]

This is Tester Home Page

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

http://localhost:8081/DTS/TesterHome.jsp#

8.5 Developer home


http://localhost:8081/DTS/DeveloperHome.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back
Search
Favorites
Address
Go
Links
Done
Local intranet

http://localhost:8081/DTS/DeveloperHome.jsp

Defect Tracking System

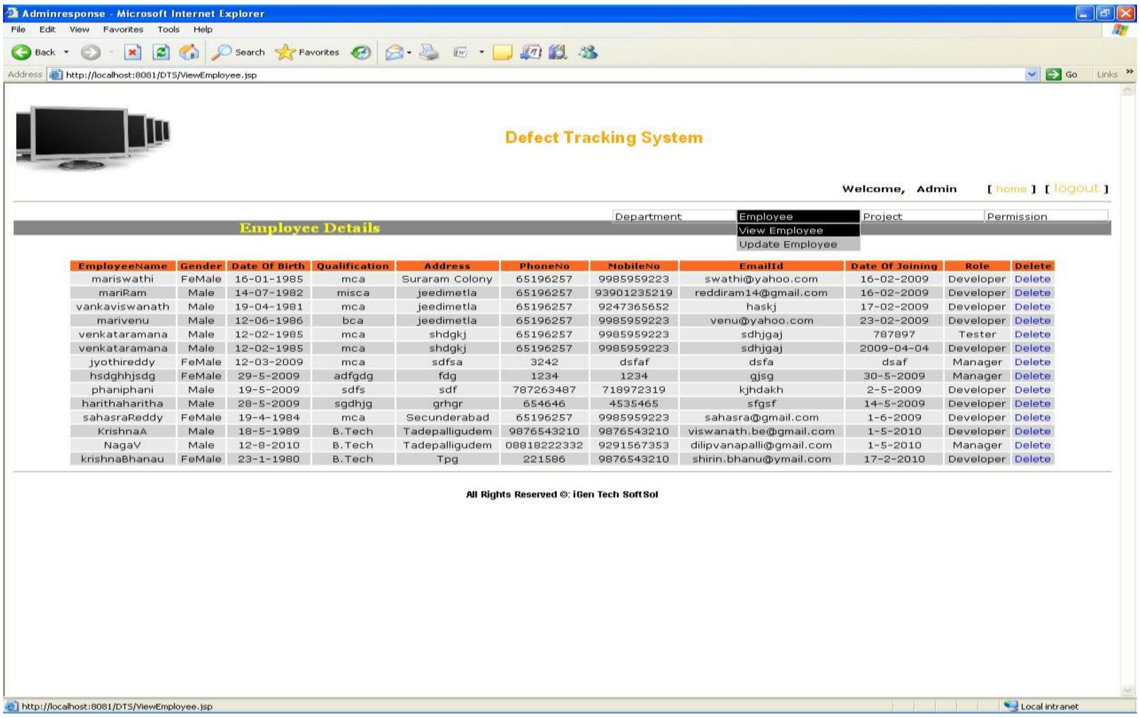
Welcome, ram [home] [logout]

Open

This is Developer Home Page

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

8.6 Admin view employee



http://localhost:8081/DTS/ViewEmployee.jsp

Defect Tracking System

Welcome, Admin [home] [logout]

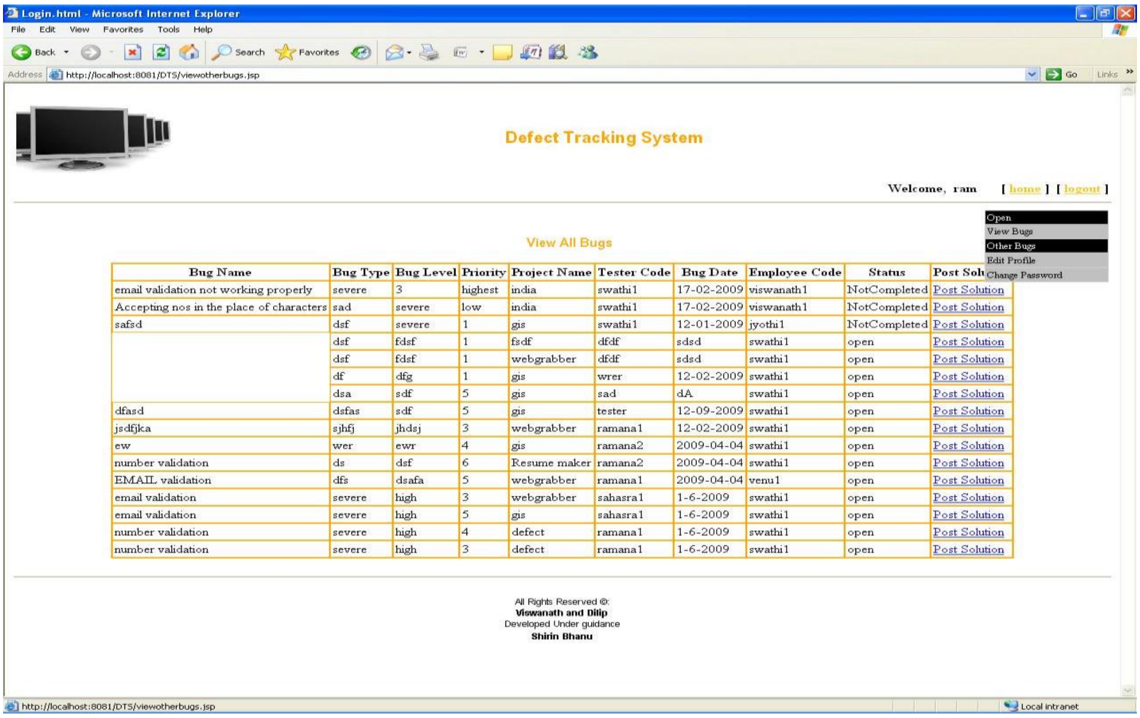
Employee Details

EmployeeName	Gender	Date Of Birth	Qualification	Address	PhoneNo	MobileNo	EmailId	Date Of Joining	Role	Delete
mariswathi	FeMale	16-01-1985	mca	Suraram Colony	65196257	9985959223	swathi@yahoo.com	16-02-2009	Developer	Delete
mariRam	Male	14-07-1982	misca	jeedimetla	65196257	93901235219	reddiram14@gmail.com	16-02-2009	Developer	Delete
vankaviswanath	Male	19-04-1981	mca	jeedimetla	65196257	9247365652	haskj	17-02-2009	Developer	Delete
marivenu	Male	12-06-1986	bca	jeedimetla	65196257	9985959223	venu@yahoo.com	23-02-2009	Developer	Delete
venkataramana	Male	12-02-1985	mca	shdgkj	65196257	9985959223	sdhjgaj	787897	Tester	Delete
venkataramana	Male	12-02-1985	mca	shdgkj	65196257	9985959223	sdhjgaj	2009-04-04	Developer	Delete
jyothireddy	FeMale	12-03-2009	mca	sdfsa	3242	dsfaf	dsfa	dsaf	Manager	Delete
hsdghhjsdg	FeMale	29-5-2009	adfgdg	fdg	1234	1234	gjsg	30-5-2009	Manager	Delete
phaniphani	Male	19-5-2009	sdfs	sdf	787263487	718972319	kjhdakh	2-5-2009	Developer	Delete
harithaharitha	Male	28-5-2009	sgdhjg	grhgr	654646	4535465	sfgsf	14-5-2009	Developer	Delete
sahasraReddy	FeMale	19-4-1984	mca	Secunderabad	65196257	9985959223	sahasra@gmail.com	1-6-2009	Developer	Delete
KrishnaA	Male	18-5-1989	B.Tech	Tadepalligudem	9876543210	9876543210	viswanath.be@gmail.com	1-5-2010	Developer	Delete
NagaV	Male	12-8-2010	B.Tech	Tadepalligudem	08818222332	9291567353	dilipvanapalli@gmail.com	1-5-2010	Manager	Delete
krishnaBhanau	FeMale	23-1-1980	B.Tech	Tpg	221586	9876543210	shirin.bhanu@ymail.com	17-2-2010	Developer	Delete

All Rights Reserved ©: iGen Tech SoftSol

http://localhost:8081/DTS/ViewEmployee.jsp

8.7 Tester view bugs



Login.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/DTS/viewotherbugs.jsp
Go Links

Defect Tracking System

Welcome, ram [home] [logout]

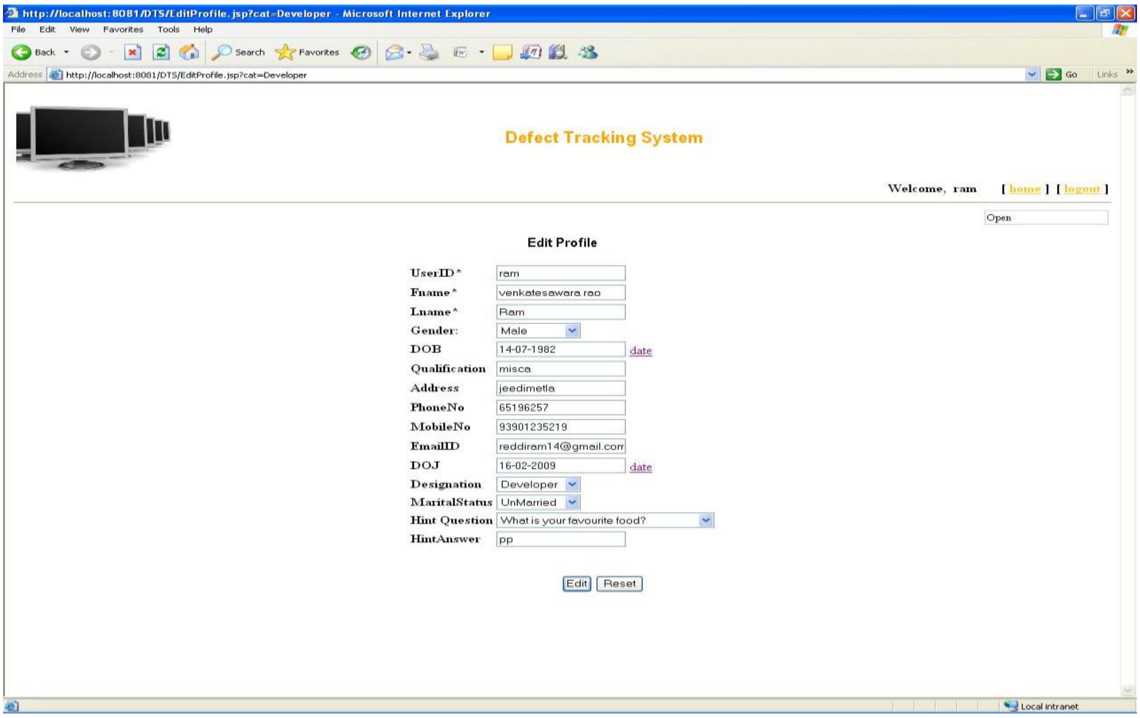
Open
View Bugs
Other Bugs
Edit Profile
Change Password

View All Bugs

Bug Name	Bug Type	Bug Level	Priority	Project Name	Tester Code	Bug Date	Employee Code	Status	Post Solu
email validation not working properly	severe	3	highest	india	swathi1	17-02-2009	viswanath1	NotCompleted	Post Solution
Accepting nos in the place of characters	sad	severe	low	india	swathi1	17-02-2009	viswanath1	NotCompleted	Post Solution
safsd	dsf	severe	1	gis	swathi1	12-01-2009	jyothi1	NotCompleted	Post Solution
	dsf	fdsf	1	fsdf	dfdf	sdsd	swathi1	open	Post Solution
	dsf	fdsf	1	webgrabber	dfdf	sdsd	swathi1	open	Post Solution
	df	dfg	1	gis	wrer	12-02-2009	swathi1	open	Post Solution
	dsa	sdf	5	gis	sad	dA	swathi1	open	Post Solution
dfasd	dsfas	sdf	5	gis	tester	12-09-2009	swathi1	open	Post Solution
jsdfjka	sjhfj	jhdsj	3	webgrabber	ramana1	12-02-2009	swathi1	open	Post Solution
ew	wer	ewr	4	gis	ramana2	2009-04-04	swathi1	open	Post Solution
number validation	ds	dsf	6	Resume maker	ramana2	2009-04-04	swathi1	open	Post Solution
EMAIL validation	dfs	dsafa	5	webgrabber	ramana1	2009-04-04	venu1	open	Post Solution
email validation	severe	high	3	webgrabber	sahasra1	1-6-2009	swathi1	open	Post Solution
email validation	severe	high	5	gis	sahasra1	1-6-2009	swathi1	open	Post Solution
number validation	severe	high	4	defect	ramana1	1-6-2009	swathi1	open	Post Solution
number validation	severe	high	3	defect	ramana1	1-6-2009	swathi1	open	Post Solution

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

8.8 Edit profile



http://localhost:8081/DTS/EditProfile.jsp?cat=Developer - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/DTS/EditProfile.jsp?cat=Developer
Go Links

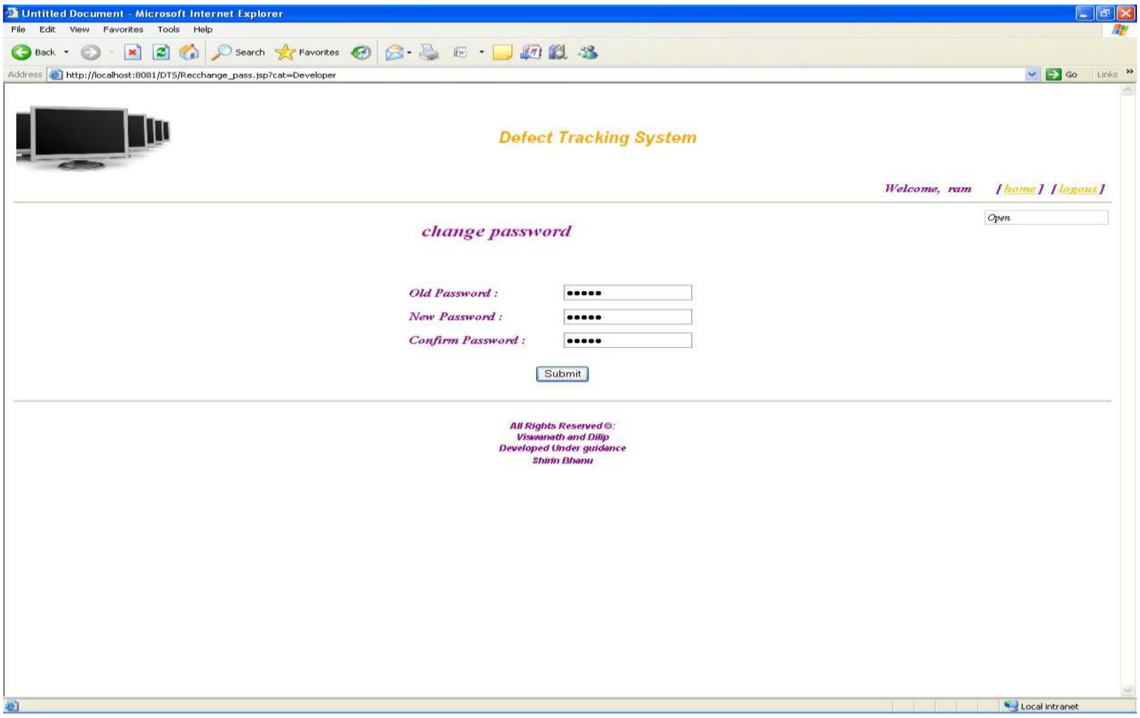
Defect Tracking System

Welcome, ram [home] [logout]

Edit Profile

UserID*	ram	
Fname*	venkatesawara rao	
Lname*	Ram	
Gender:	Male	
DOB	14-07-1982	date
Qualification	misca	
Address	jeedimetla	
PhoneNo	65196257	
MobileNo	93901235219	
EmailID	reddiram14@gmail.com	
DOJ	16-02-2009	date
Designation	Developer	
MaritalStatus	UnMarried	
Hint Question	What is your favourite food?	
HintAnswer	pp	

8.9 Change password


Untitled Document - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Local intranet

http://localhost:8081/DTS/Recchange_pass.jsp?cat=Developer

Defect Tracking System

Welcome, ram *[home] [logout]*

Open

change password

Old Password : •••••

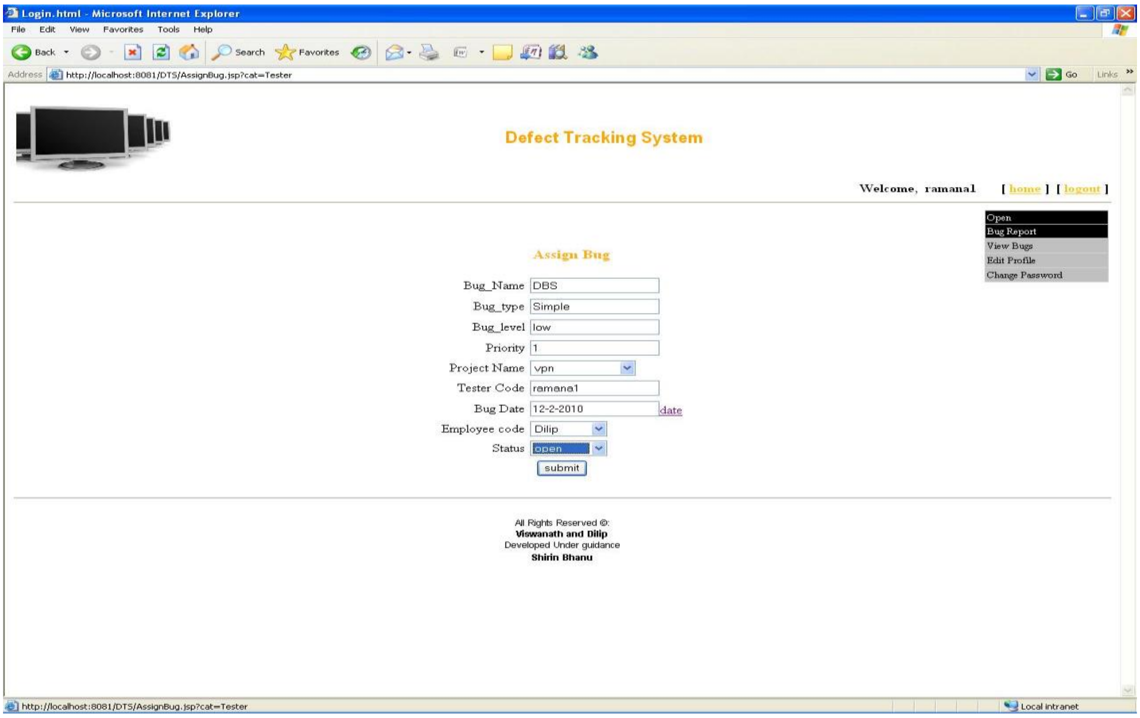
New Password : •••••

Confirm Password : •••••

Submit

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

8.10Developer bug report


Login.html - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address
Go Links
Local intranet

http://localhost:8081/DTS/AssignBug.jsp?cat=Tester

Defect Tracking System

Welcome, ramana1 [home] [logout]

Open
Bug Report
View Bugs
Edit Profile
Change Password

Assign Bug

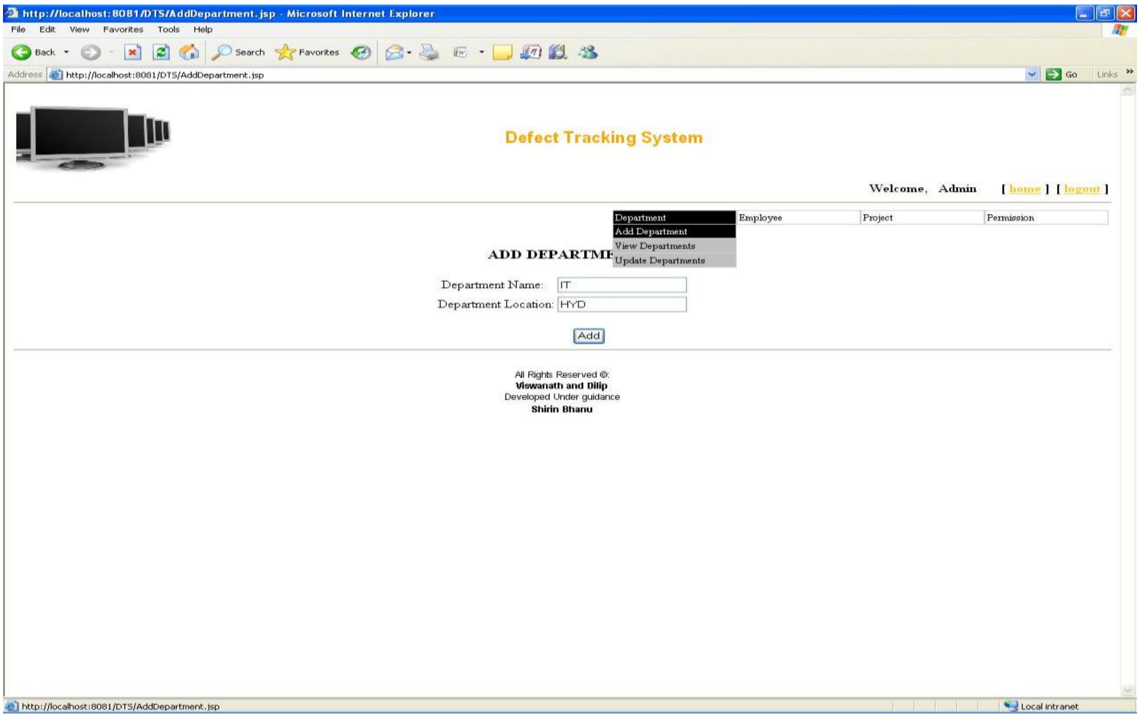
Bug_Name	DBS
Bug_type	Simple
Bug_level	low
Priority	1
Project Name	vpn
Tester Code	ramana1
Bug Date	12-2-2010 date
Employee code	Dilip
Status	open

submit

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

http://localhost:8081/DTS/AssignBug.jsp?cat=Tester

8.11 Admin add department


http://localhost:8081/DTS/AddDepartment.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/DTS/AddDepartment.jsp
Go Links
Department
Add Department
View Departments
Update Departments
Employee
Project
Permission
http://localhost:8081/DTS/AddDepartment.jsp
Local intranet

Defect Tracking System

Welcome, Admin [home] [logout]

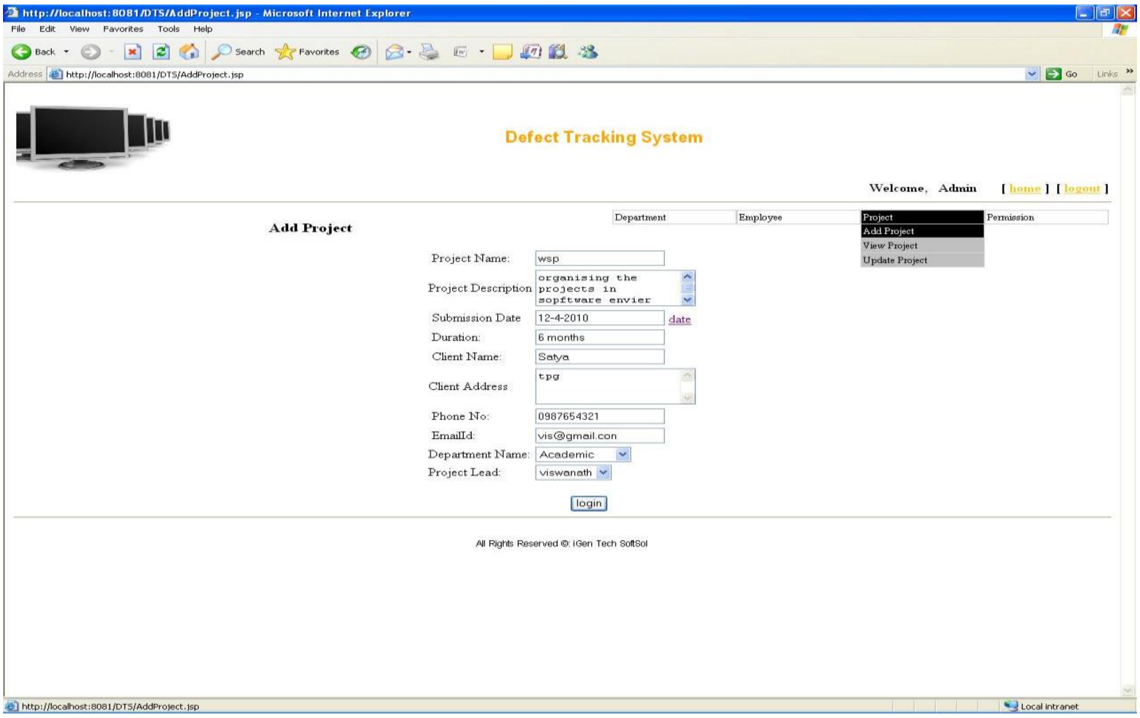
ADD DEPARTME

Department Name: IT
Department Location: HYD

Add

All Rights Reserved ©:
Viswanath and Dilip
Developed Under guidance
Shirin Bhanu

8.12 Admin add project


http://localhost:8081/DTS/AddProject.jsp - Microsoft Internet Explorer
File Edit View Favorites Tools Help
Back Search Favorites
Address http://localhost:8081/DTS/AddProject.jsp
Go Links
Department
Employee
Project
Add Project
View Project
Update Project
Permission
http://localhost:8081/DTS/AddProject.jsp
Local intranet

Defect Tracking System

Welcome, Admin [home] [logout]

Add Project

Project Name: wsp
Project Description: organising the projects in sopftware envier
Submission Date: 12-4-2010 date
Duration: 6 months
Client Name: Satya
Client Address: tpg
Phone No: 0987654321
EmailId: vis@gmail.con
Department Name: Academic
Project Lead: viswanath

login

All Rights Reserved ©: iGen Tech SoftSol

9. CONCLUSION

User comes to the search engine and makes a query, typically by giving key words, the engine looks up the index and provides a listing of best-matching web pages according to its criteria, usually with a short summary containing the document's title and sometimes parts of the text.

Most search engines employ methods to rank the results to provide the “best” results first. How a search engine decides which pages are the best matches, and what order the results should be shown in, varies widely from one engine to another.

Search engine is technically the software and algorithms used to perform a search, the term have become synonymous with the website itself.

10. BIBLIOGRAPHY

- **System Analysis And Design**

  By William S. Davis, David C. Yen

- **An Analysis and Design of Information Systems**

  By Simon Bell

- **Software Engineering**

  -By Schach

- **Database Management System**

  By Jason Price

- **JSP: A Beginner's Guide**

  -By Robert Brunner

- **Java Server Pages Programming: A visual professional guide for Design**

  By Carsten Thomsen