1. INTRODUCTION

1.1 Purpose

The purpose of the project is to provide detailed information about application. It includes the Functional and Non-Functional Requirements.

1.2 Scope of Project

It is a web-based application which is developed to solve programmers or developers errors in programming and also solving other related problems while programming.

1.3 Intended Audience

The project is being developed for the programmers and others like students, computer analyst software employees etc.

1.4 Overview

This section of the project describes about the functionality of the application and other requirements which are utilized in the development of application. Other technical modules which are present in the system.

2.LITERATURE SURVEY

2.1Existing System

The present system is ineffective in logging and maintaining all the details of defects and errors in a convenient manner. The traditional way of maintaining these details causes errors and results in slow processing. In IT industry, Many developers come across a lot of problems while developing any project in which defects and errors are the most common problems.

Employee has to manually solve his errors either by referring books or confessing with colleagues. Where working in a company doesn’t involve in showcasing ones skills and abilities for helping another employee which is a very time taking procedure and it may also result in delaying of work and sometimes even search engines also doesn’t provide required solutions and eventually this may lead to the delay in fulfilling the domain

Disadvantages

- Difficulty in finding solutions
- Should filter a huge list of results.
- No guarantee for getting a solution.
- Each time he has to waste lot of time searching for solutions.

2.2 Proposed System

The proposed system is a web-based application whose main goal is to take the responsibilities for providing fast and accurate solutions of the java problems to the Java and project developers. And users can create his profile through which he can post his error along with description on our proposed application. Our Application will take the responsibility for providing the rightful solution for the user within a short time so that the user need not be waiting for a long time in search of solutions. And he would be assigned with a guaranteed solution until the problem gets solved.

Advantages

- Ease in getting solutions.
- User gets very least number of solutions.
- Full time guarantee for solutions.
- Solutions are provided with very short tim

3.OVERALL DESCRIPTION

3.1Product Perspective

This website is developed for all purposes in various perspectives. It is developed as a website using web designing technologies and java technologies.

3.2Modules Description

Number of Modules

1. Administrator Module
2. User Module
3. Manager Module

Administrator Module

Administrator is a kind of manager who can add, delete and edit the user information. He has the functionality of the user. He also maintains the details of all bugs and takes the help from other managers so that each user will received appropriate solutions.

User Module

Users can post their errors and can view solutions to his relative errors.Users can also view and edit any queries with other users. Each user can have his own profile and is allowed to update the profile details

.

Manager Module

Manager interacts with the administrator to maintain the details of errors and defects .They also communicate with the user to find the status of bugs.

3.3 User Characteristics

- The user should register on the website and should posses with email id and password every time he login into the website.
- He should do some background work regarding the errors before posting it.

Manager Characteristics

- The manager should also have his own profile on the website and this manager profile is generated by the administrator.
- He posts about the errors for the users so that he can provide a solution.

4. SPECIFIC REQURIMENTS

This section describes about the technologies and other requirements that are used in application development.

4.1 Interface Requirements

4.1.1 Software Requirements

IDE: Netbeans

OS: Windows 7

Database: MySql Server

Server: Tomcat Apache

Programming Language: Java JDK 1.8

Web Design: HTML, CSS, and

JavaScript

4.1.2. Hardware Requirements

- Processor: Intel I3 processor
- Speed: 2.10 Ghz
- Ram: 1 GB
- Monitor, Mouse and Keyboard

4.2 Functional Requirements

User Registration

Purpose: The purpose of registration is to store the details of the user with personal and professional details.

Inputs: The user will enter details in the registration form according to required fields, some of the fields include are username

Processing: On the server side the servlet is used to capture the details of the user and then store the details in the database.

Output : After the registration the user will be directed to the main home.

4.3 Non-Functional Requirements

- Performance: The Performance is high only when valid things are provided
- Security: Only authorized users can download or view certain files.
- Database: It is used to store the details of the user and his credentials.

4.4 Design Constraints

- The Application is tested only on Windows 8. But, It will perform all the features in other Operating Systems as expected.
- The User should not expect for quota or Storage Space from the application because it is not intended to provide additional quota to the users.

4.5Key Functionalities

The following are the key functionalities in this application

The user has the facility to:

- Create and delete of accounts
- Updating of personal data
- Retrieval of password and other information
- Restricted list of download documents and other data

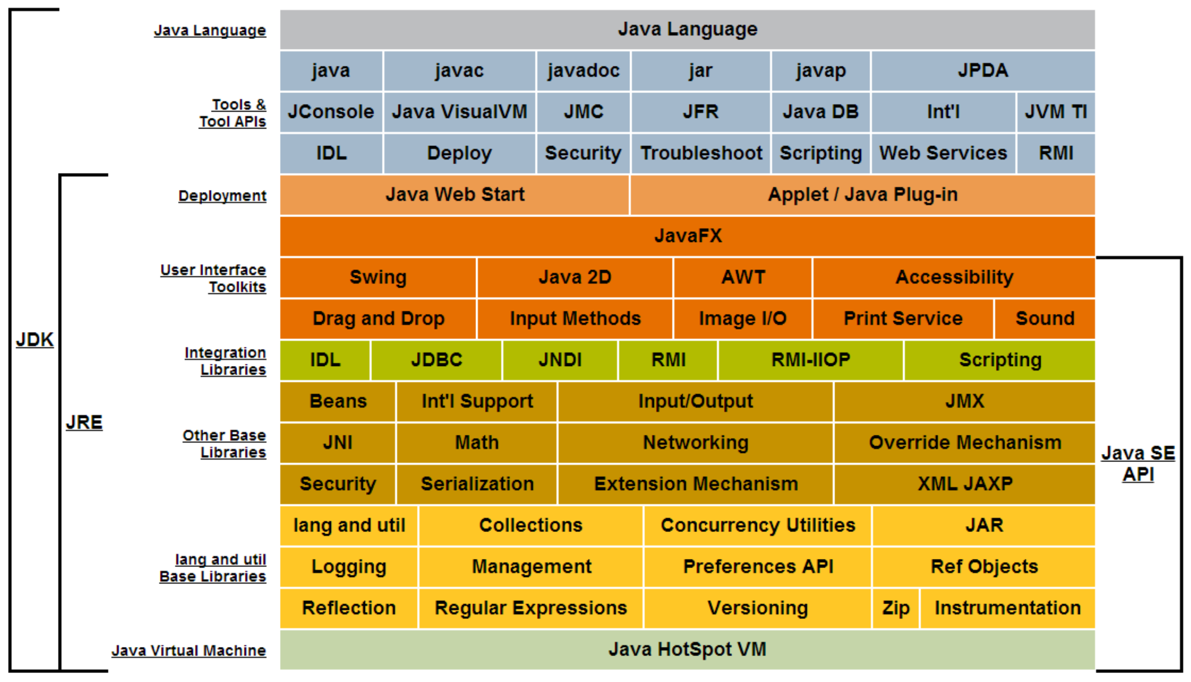
5. TECHNOLOGIES USED

5.1 Java and Java Architecture

5.1.1 About Java

Java is a general-purpose computer programming language that is **concurrent, class-based, object-oriented** and specifically designed to have as few implementation dependencies as possible. It is intended to let application developers "**write once, run anywhere**" (WORA) meaning that compiled Java code can run on all platforms that support Java without the need for compilation. Java applications are typically compiled to **byte code** that can run on any **Java virtual machine** (JVM) regardless of computer architecture. As of 2015, Java is one of the most popular programming languages in use particularly for client-server web applications, with a reported 9 million developers. Java was originally developed by **James Gosling** at **Sun Microsystems** (which has since been acquired by **Oracle Corporation**) and released in **1995** as a core component of Sun Microsystems Java platform.

5.1.2Architecture



5.2HTML:

HyperText Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. Along with CSS, and JavaScript, HTML is a cornerstone technology, used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

Web browsers can read HTML files and render them into visible or audible web pages. HTML describes the structure of a website semantically along with cues for presentation, making it a markup language, rather than a programming language.

HTML elements form the building blocks of all websites. HTML allows images and objects to be embedded and can be used to create interactive forms. It provides a means to create structured documents by denoting structural semantics for text such as headings, paragraphs, lists, links, quotes and other items.

The language is written in the form of HTML elements consisting of tags enclosed in angle brackets (like <html>). Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page.

HTML can embed scripts written in languages such as JavaScript which affect the behavior of HTML web pages. Web browsers can also refer to Cascading Style Sheets (CSS) to define the look and layout of text and other material. The World Wide Web Consortium (W3C), maintainer of both the HTML and the CSS standards, has encouraged the use of CSS over explicit presentational HTML since 1997.

HTML markup consists of several key components, including those called tags (and their attributes), character-based data types, character references and entity references. HTML tags most commonly come in pairs like <h1> and </h1>, although some represent empty elements and so are unpaired, for example <img>. The first tag in such a pair is the start tag, and the second is the end tag (they are also called opening tags and closing tags).Another important component of the HTML document type declaration, which triggers standards mode rendering.The Document Type Declaration <!DOCTYPE html> is for HTML5. If a declaration is not included, various browsers will revert to "quirks mode" for rendering.

HTML documents imply a structure of nested HTML elements. These are indicated in the document by HTML tags, enclosed in angle brackets thus: <p>. In the simple, general case, the extent of an element is indicated by a pair of tags: a "start tag" <p> and "end tag" </p>. The text content of the element, if any, is placed between these tags.

Tags may also enclose further tag markup between the start and end, including a mixture of tags and text. This indicates further (nested) elements, as children of the parent element. The start tag may also include attributes within the tag. These indicate other information, such as identifiers for sections within the document, identifiers used to bind style information to the presentation of the document, and for some tags such as the <img> used to embed images, the reference to the image resource.

Some elements, such as the line break <br>, do not permit any embedded content, either text or further tags. These require only a single empty tag (akin to a start tag) and do not use an end tag.

Many tags, particularly the closing end tag for the very commonly-used paragraph element <p>, are optional. An HTML browser or other agent can infer the closure for the end of an element from the context and the structural rules defined by the HTML standard. These rules are complex and not widely understood by most HTML coders.

The general form of an HTML element is therefore: <tag attribute1="value1" attribute2="value2">content</tag>. Some HTML elements are defined as empty elements and take the form <tag attribute1="value1" attribute2="value2">. Empty elements may enclose no content, for instance, the <br> tag or the inline <img> tag. The name of an HTML element is the name used in the tags. Note that the end tag's name is preceded by a slash character, "/", and that in empty elements the end tag is neither required nor allowed. If attributes are not mentioned, default values are used in each case.

Line breaks:<br>. The difference between <br> and <p> is that "br" breaks a line without altering the semantic structure of the page, whereas "p" sections the page into paragraphs. Note also that "br" is an empty element in that, although it may have attributes, it can take no content and it may not have an end tag.

<p>This<br> is a paragraph <br> with <br> line breaks</p>

This is a link in HTML. To create a link the <a> tag is used. The href= attribute holds the URL address of the link.

<a href="https://www.wikipedia.org/">A link to Wikipedia!</

Attributes:

Most of the attributes of an element are name-value pairs, separated by "=" and written within the start tag of an element after the element's name. The value may be enclosed in single or double quotes, although values consisting of certain characters can be left unquoted in HTML (but not XHTML) Leaving attribute values unquoted is considered unsafe. In contrast with name-value pair attributes, there are some attributes that affect the element simply by their presence in the start tag of the element,like the ismap attribute for the img element.

There are several common attributes that may appear in many elements :

The id attribute provides a document-wide unique identifier for an element. This is used to identify the element so that stylesheets can alter its presentational properties, and scripts may alter, animate or delete its contents or presentation. Appended to the URL of the page, it provides a globally unique identifier for the element, typically a sub-section of the page.

The class attribute provides a way of classifying similar elements. This can be used for semantic or presentation purposes. For example, an HTML document might semantically use the designation class="notation" to indicate that all elements with this class value are subordinate to the main text of the document. In presentation, such elements might be gathered together and presented as footnotes on a page instead of appearing in the place where they occur in the HTML source. Class attributes are used semantically in microformats. Multiple class values may be specified; for example class="notation important" puts the element into both the "notation" and the "important" classes.

An author may use the style attribute to assign presentational properties to a particular element. It is considered better practice to use an element's id or class attributes to select the element from within a stylesheet, though sometimes this can be too cumbersome for a simple, specific, or ad hoc styling.

The title attribute is used to attach subtextual explanation to an element. In most browsers this attribute is displayed as a tooltip.

The lang attribute identifies the natural language of the element's contents, which may be different from that of the rest of the document.

Data types:

HTML defines several data types for element content, such as script data and stylesheet data, and a plethora of types for attribute values, including IDs, names, URIs, numbers, units of length, languages, media descriptors, colors, character encodings, dates and times, and so on. All of these data types are specializations of character data.

HTTP:

The World Wide Web is composed primarily of HTML documents transmitted from web servers to web browsers using the Hypertext Transfer Protocol (HTTP). However, HTTP is used to serve images, sound, and other content, in addition to HTML. To allow the web browser to know how to handle each document it receives, other information is transmitted along with the document. This meta data usually includes the MIME type (e.g. text/html or application/xhtml+xml) and the character encoding (see Character encoding in HTML).

In modern browsers, the MIME type that is sent with the HTML document may affect how the document is initially interpreted. A document sent with the XHTML MIME type is expected to be well-formed XML; syntax errors may cause the browser to fail to render it. The same document sent with the HTML MIME type might be displayed successfully, since some browsers are more lenient with HTML.

The W3C recommendations state that XHTML 1.0 documents that follow guidelines set forth in the recommendation's Appendix C may be labeled with either MIME Type.XHTML 1.1 also states that XHTML 1.1 documents should be labeled with either MIME type.

5.3 CSS:

Cascading Style Sheets (CSS) is a style sheet language used for describing the presentation of a document written in a markup language. Although most often used to set the visual style of web pages and user interfaces written in HTML and XHTML, the language can be applied to any XML document, including plain XML, SVG and XUL, and is applicable to rendering in speech, or on other media. Along with HTML and JavaScript, CSS is a cornerstone technology used by most websites to create visually engaging webpages, user interfaces for web applications, and user interfaces for many mobile applications.

CSS is designed primarily to enable the separation of document content from document presentation, including aspects such as the layout, colors, and fonts.This separation can improve content accessibility, provide more flexibility and control in the specification of presentation characteristics, enable multiple HTML pages to share formatting by specifying the relevant CSS in a separate .css file, and reduce complexity and repetition in the structural content, such as semantically insignificant tables that were widely used to format pages before consistent CSS rendering was available in all major browsers. CSS makes it possible to separate presentation instructions from the HTML content in a separate file or style section of the HTML file. For each matching HTML element, it provides a list of formatting instructions. For example, a CSS rule might specify that "all heading 1 elements should be bold", leaving pure semantic HTML markup that asserts "this text is a level 1 heading" without formatting code such as a <bold> tag indicating how such text should be displayed.

This separation of formatting and content makes it possible to present the same markup page in different styles for different rendering methods, such as on-screen, in print, by voice (when read out by a speech-based browser or screen reader) and on Braille-based, tactile devices. It can also be used to display the web page differently depending on the screen size or device on which it is being viewed. Although the author of a web page typically links to a CSS file within the markup file, readers can specify a different style sheet, such as a CSS file stored on their own computer, to override the one the author has specified. If the author or the reader did not link the document to a style sheet, the default style of the browser will be applied. Another advantage of CSS is that aesthetic changes to the graphic design of a document (or hundreds of documents) can be applied quickly and easily, by editing a few lines in one file, rather than by a laborious (and thus expensive) process of crawling over every document line by line, changing markup.

The CSS specification describes a priority scheme to determine which style rules apply if more than one rule matches against a particular element. In this so-called cascade, priorities (or weights) are calculated and assigned to rules, so that the results are predictable.

The CSS specifications are maintained by the World Wide Web Consortium (W3C). Internet media type (MIME type) text/css is registered for use with CSS by RFC 2318 (March 1998). The W3C operates a free CSS validation service for CSS documents

5.3.1 Syntax:

CSS has a simple syntax and uses a number of English keywords to specify the names of various style properties.A style sheet consists of a list of rules. Each rule or rule-set consists of one or more selectors, and a declaration block.

5.3.2. Selector:

In CSS, selectors are used to declare which part of the markup a style applies to by matching tags and attributes in the markup itself.

Selectors may apply to:

- **:** all elements of a specific type, e.g. the second-level headers h2
- : elements specified by attribute, in particular:
- id: an identifier unique within to the document
- class: an identifier that can annotate multiple elements in a document
- **:** elements depending on how they are placed relative to others in the document tree.

Classes and IDs are case-sensitive, start with letters, and can include alphanumeric characters and underscores. A class may apply to any number of instances of any elements. An ID may only be applied to a single element.Pseudo-classes are used in CSS selectors to permit formatting based on information that is not contained in the document tree.

Advantages:

Separation of content from presentation:

CSS facilitates publication of content in multiple presentation formats based on nominal parameters. Nominal parameters include explicit user preferences, different web browsers, the type of device being used to view the content (a desktop computer or mobile Internet device), the geographic location of the user and many other variables.

Site-wide consistency:

When CSS is used effectively, in terms of inheritance and "cascading", a global style sheet can be used to affect and style elements site-wide. If the situation arises that the styling of the elements should need to be changed or adjusted, these changes can be made by editing rules in the global style sheet. Before CSS, this sort of maintenance was more difficult, expensive and time-consuming.

5.3.3Bandwidth:

A stylesheet, internal or external, specifies the style once for a range of HTML elements selected by class, type or relationship to others. This is much more efficient than repeating style information inline for each occurrence of the element. An external stylesheet is usually stored in the browser cache, and can therefore be used on multiple pages without being reloaded, further reducing data transfer over a network.

5.3.4. Page reformatting:

With a simple change of one line, a different style sheet can be used for the same page. This has advantages for accessibility, as well as providing the ability to tailor a page or site to different target devices. Furthermore, devices not able to understand the styling still display the content.

5.3.5 Accessibility

Without CSS, web designers must typically lay out their pages with techniques such as

HTML tables that hinder accessibility for vision-impaired users (see Tableless web design#Accessibility).

5.3.6 CSS frameworks:

CSS frameworks are pre-prepared libraries that are meant to allow for easier, more standards-compliant styling of web pages using the Cascading Style Sheets language. CSS frameworks include Foundation, Blueprint, Bootstrap, Cascade Framework and Materialize. Like programming and scripting language libraries, CSS frameworks are usually incorporated as external .css sheets referenced in the HTML <head>. They provide a number of ready-made options for designing and laying out the web page. Although many of these frameworks have been published, some authors use them mostly for rapid prototyping, or for learning from, and prefer to 'handcraft' CSS that is appropriate to each published site without the design, maintenance and download overhead of having many unused features in the site's styling.

5.4 JAVA:

Java is a set of computer software and specifications developed by Sun Microsystems, later acquired by Oracle Corporation, that provides a system for developing application software and deploying it in a cross-platform computing environment. Java is used in a wide variety of computing platforms from embedded devices and mobile phones to enterprise servers and supercomputers. While less common, Java applets run in secure, sandboxed environments to provide many features of native applications and can be embedded in HTML pages.

Writing in the Java programming language is the primary way to produce code that will be deployed as byte code in a Java Virtual Machine (JVM); byte code compilers are also available for other languages, including Ada, JavaScript, Python, and Ruby. In addition, several languages have been designed to run natively on the JVM, including Scala, Clojure and Groovy. Java syntax borrows heavily from C and C++, but object-oriented features are modeled after Smalltalk and Objective-C. Java eschews certain low-level constructs such as pointers and has a very simple memory model where every object is allocated on the heap and all variables of object types are references. Memory management is handled through integrated automatic garbage collection performed by the JVM.

Platform:

The Java platform is a suite of programs that facilitate developing and running programs written in the Java programming language. The platform is not specific to any one processor or operating system, rather an execution engine (called a virtual machine) and a compiler with a set of libraries are implemented for various hardware and operating systems so that Java programs can run identically on all of them. There are multiple platforms, each targeting a different class of devices:

Java Card:

A technology that allows small Java-based applications (applets) to be run securely on smart cards and similar small-memory devices.

Java ME (Micro Edition):

Specifies several different sets of libraries (known as profiles) for devices with limited storage, display, and power capacities. Often used to develop applications for mobile devices, PDAs, TV set-top boxes, and printers.Java SE (Standard Edition):

For general-purpose use on desktop PCs, servers and similar devices.

Java EE (Enterprise Edition):

Java SE plus various APIs useful for multi-tier client–server enterprise applications.

The Java platform consists of several programs, each of which provides a portion of its overall capabilities. For example, the Java compiler, which converts Java source code into Java bytecode (an intermediate language for the JVM), is provided as part of the Java Development Kit (JDK). The Java Runtime Environment (JRE), complementing the JVM with a just-in-time (JIT) compiler, converts intermediate bytecode into native machine code on the fly. An extensive set of libraries are also part of the Java platform.

The essential components in the platform are the Java language compiler, the libraries, and the runtime environment in which Java intermediate bytecode executes according to the rules laid out in the virtual machine specification.

Java Virtual Machine:

The heart of the Java platform is the concept of a "virtual machine" that executes Java bytecode programs. This bytecode is the same no matter what hardware or operating system the program is running under. There is a JIT (Just In Time) compiler within the Java Virtual Machine, or JVM. The JIT compiler translates the Java bytecode into native processor instructions at run- time and caches the native code in memory during execution.

The use of bytecode as an intermediate language permits Java programs to run on any platform that has a virtual machine available. The use of a JIT compiler means that Java applications, after a short delay during loading and once they have "warmed up" by being all or mostly JIT-compiled, tend to run about as fast as native programs.[citation needed] Since JRE version 1.2, Sun's JVM implementation has included a just-in-time compiler instead of an interpreter. Although Java programs are cross-platform or platform independent, the code of the Java Virtual Machines (JVM) that execute these programs is not. Every supported operating platform has its own JVM.

Java Development Kit:

The Java Development Kit (JDK) is a Sun product aimed at Java developers. Since the introduction of Java, it has been by far the most widely used Java software development kit (SDK).It contains a Java compiler, a full copy of the Java Runtime Environment (JRE), and many other important development tools.

5.5. Java Scripts:

Java script is a high-level, dynamic, untyped, and interpreted programming language. Alongside HTML and CSS, it is one of the three essential technologies of World Wide Web content production; the majority of websites employ it and it is supported by all modern web browsers without plug-ins. JavaScript is prototype-based with first-class functions, making it a multi-paradigm language, supporting object-oriented, imperative, and functional programming styles. It has an API for working with text, arrays, dates and regular expressions, but does not include any I/O, such as networking, storage or graphics facilities, relying for these upon the host environment in which it is embedded.

Despite some naming, syntactic, and standard library similarities, JavaScript and Java are otherwise unrelated and have very different semantics. The syntax of JavaScript is actually derived from C, while the semantics and design are influenced by the Self and Scheme programming languages.

JavaScript is also used in environments that are not web-based, such as PDF documents, site-specific browsers, and desktop widgets. Newer and faster JavaScript virtual machines (VMs) and platforms built upon them have also increased the popularity of JavaScript for server-side web applications. On the client side, JavaScript has been traditionally implemented as an interpreted language, but more recent browsers perform just-in-time compilation. It is also used in game development, the creation of desktop and mobile applications, and server-side network programming with runtime environments such as Node.

5.5.1. Features:

Imperative and structured:

JavaScript supports much of the structured programming syntax from C (e.g., if statements, while loops, switch statements, do while loops, etc.). One partial exception is scoping: JavaScript originally had only function scoping with var. ECMAScript 2015 adds a let keyword for block scoping, meaning JavaScript now has both function and block scoping. Like C, JavaScript makes a distinction between expressions and statements. One syntactic difference from C is automatic semicolon insertion, which allows the semicolons that would normally terminate statements to be omitted.

Dynamic:

Dynamic typing: As in most scripting languages, types are associated with values, not with variables. For example, a variable x could be bound to a number, then later rebound to a string. JavaScript supports various ways to test the type of an object, including duck typing.

Object-based: JavaScript is almost entirely object-based. JavaScript objects are associative arrays, augmented with prototypes (see below). Object property names are string keys. They support two equivalent syntaxes: dot notation (obj.x = 10) and bracket notation (obj['x'] = 10). Properties and their values can be added, changed, or deleted at run- time. Most properties of an object (and those on its prototype inheritance chain) can be enumerated using a for...in loop. JavaScript has a small number of built-in objects such as Function and Date.

Run-time evaluation: JavaScript includes an eval function that can execute statements provided as strings at run-time.

Functional:

First-class functions:

Functions are first-class; they are objects themselves. As such, they have properties and methods, such as .call() and .bind(). A nested function is a function defined within another function. It is created each time the outer function is invoked. In addition, each created function forms a lexical closure: the lexical scope of the outer function, including any constants, local variables and argument values, becomes part of the internal state of each inner function object, even after execution of the outer function concludes.JavaScript also supports anonymous functions.

Prototype-based object-oriented programming:

Prototypes:

JavaScript uses prototypes where many other object-oriented languages use classes for inheritance. It is possible to simulate many class-based features with prototypes in JavaScript.

Functions as object constructors:

Functions double as object constructors along with their typical role. Prefixing a function call with new will create an instance of a prototype, inheriting properties and methods from the constructor (including properties from the Object prototype). ECMAScript 5 offers the Object.create method, allowing explicit creation of an instance without automatically inheriting from the Object prototype (older environments can assign the prototype to null). The constructor's prototype property determines the object used for the new object's internal prototype. New methods can be added by modifying the prototype of the function used as a constructor. JavaScript's built-in constructors, such as Array or Object, also have prototypes that can be modified. While it is possible to modify the Object prototype, it is generally considered bad practice because most objects in JavaScript will inherit methods and properties from the Object prototype and they may not expect the prototype to be modified.

Functions as methods:

Unlike many object-oriented languages, there is no distinction between a function definition and a method definition. Rather, the distinction occurs during function calling; when a function is called as a method of an object, the function's local this keyword is bound to that object for that invocation.

Implicit and explicit delegation:

JavaScript is a delegation language.

Functions as Roles (Traits and Mixins): JavaScript natively supports various function-based implementations of Role patterns like Traits and Mixins. Such a function defines additional behavior by at least one method bound to the this keyword within its function body. A Role then has to be delegated explicitly via call or apply to objects that need to feature additional behavior that is not shared via the prototype chain.

Object Composition and Inheritance: Whereas explicit function-based delegation does cover composition in JavaScript, implicit delegation already happens every time the prototype chain is walked in order to, e.g., find a method that might be related to but is not directly owned by an object. Once the method is found it gets called within this object's context. Thus inheritance in JavaScript is covered by a delegation automatism that is bound to the prototype property of constructor functions.

Miscellaneous:

Run-time environment:

JavaScript typically relies on a run-time environment (e.g., a Web browser) to provide objects and methods by which scripts can interact with the environment (e.g., a webpage DOM). It also relies on the run-time environment to provide the ability to include/import scripts (e.g., HTML
<script> elements). This is not a language feature per se, but it is common in most Java Script implementations.

JavaScript processes messages from a queue one at a time. Upon loading a new message, JavaScript calls a function associated with that message, which creates a call stack frame (the function's arguments and local variables). The call stack shrinks and grows based on the function's needs. Upon function completion, when the stack is empty, JavaScript proceeds to the next message in the queue. This is called the event loop, described as "run to completion" because each message is fully processed before the next message is considered. However, the language's concurrency model describes the event loop as non-blocking: program input/output is performed using events and callback functions. This means, for instance, that JavaScript can process a mouse click while waiting for a database query to return information.

Variadic functions:

An indefinite number of parameters can be passed to a function. The function can access them through formal parameters and also through the local arguments object. Variadic functions can also be created by using the bind method.

Array and object literals:

Like many scripting languages, arrays and objects (associative arrays in other languages) can each be created with a succinct shortcut syntax. In fact, these literals form the basis of the

JSON data format.

Regular expressions:

JavaScript also supports regular expressions in a manner similar to Perl, which provide a concise and powerful syntax for text manipulation that is more sophisticated than the built-in string functions.

6. SYSTEM DESIGN ARCHITECTURE

System design is transition from a user oriented document to programmers or data base personnel. The design is a solution, how to approach to the creation of a new system. This is composed of several steps. It provides the understanding and procedural details necessary for implementing the system recommended in the feasibility study. Designing goes through logical and physical stages of development, logical design reviews the present physical system, prepare input and output specification, details of implementation plan and prepare a logical design walkthrough.

6.1System Architecture

- First , A User enters the website and logins with his credentials
- Second, He selects his desired forum for posting his selected problem.
- Once the user selects his forum, now he can enter the details of the problem he is facing regarding the code.
- If any solution is already stored in the database, the solution is directly given to user.
- If the solution is not existed then the user has to wait for a little time until the developers post the solution..

Hence, By using this the user can easily obtain any solutions to his problem during coding and also save a lot of time searching for solutions.

6.2UML Diagrams

Unified Modelling Language-UML

UML is a standard language for writing software blueprints. It is a language which provide various vocabulary and rules for communication that focus on conceptual and physical representation of the system.

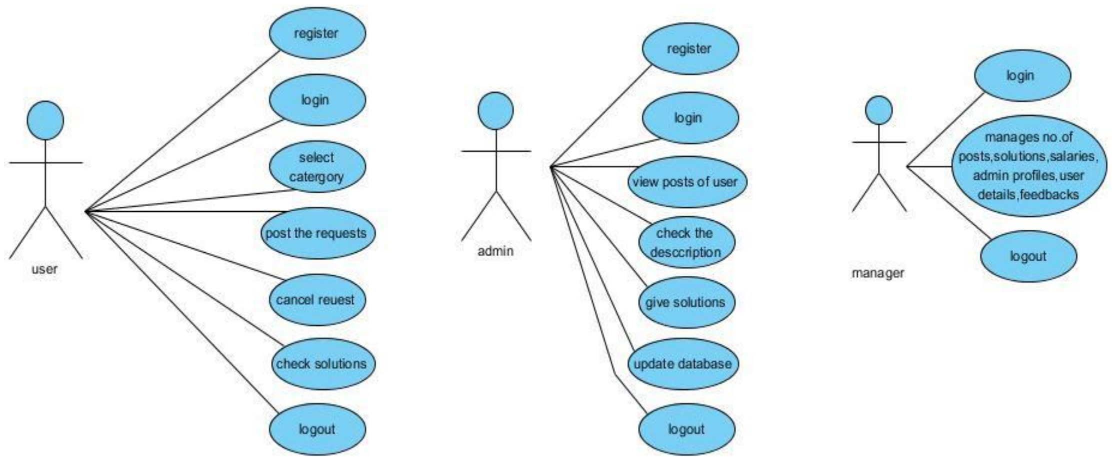
It is used to visualize, specify, construct, and document the artifacts of a software-intensive system.

This Design Documentation consists the following diagrams from UML to represent the system flow

- Use-case Diagrams
- Class Diagrams
- Sequence Diagrams
- Activity Diagrams
- Component Diagrams
- Deployment Diagrams

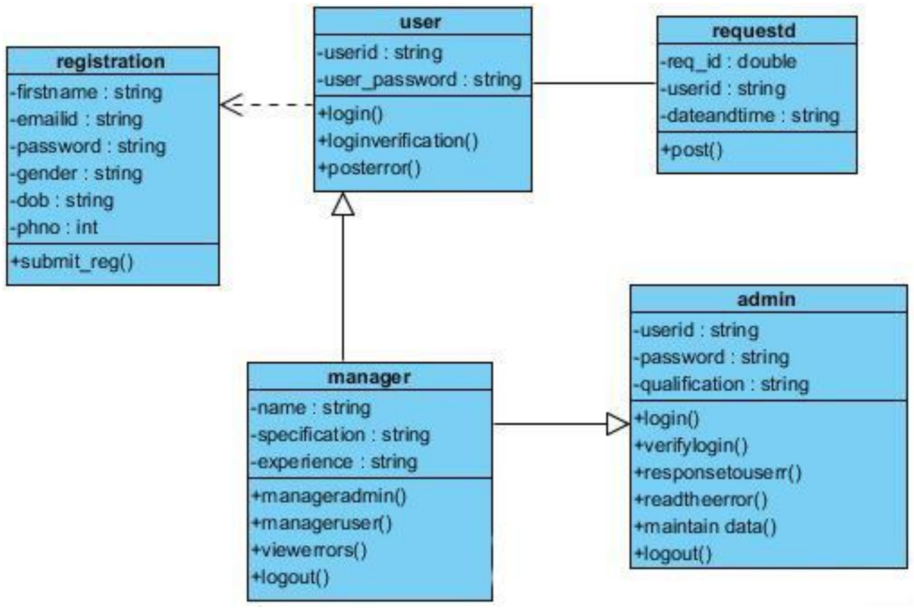
Use-Case Diagrams

A Use-case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use- cases in which the user is involved. A use-case diagram can identify the different types of users of a system and the different use cases and will often be accompanied by other types of diagrams as well.



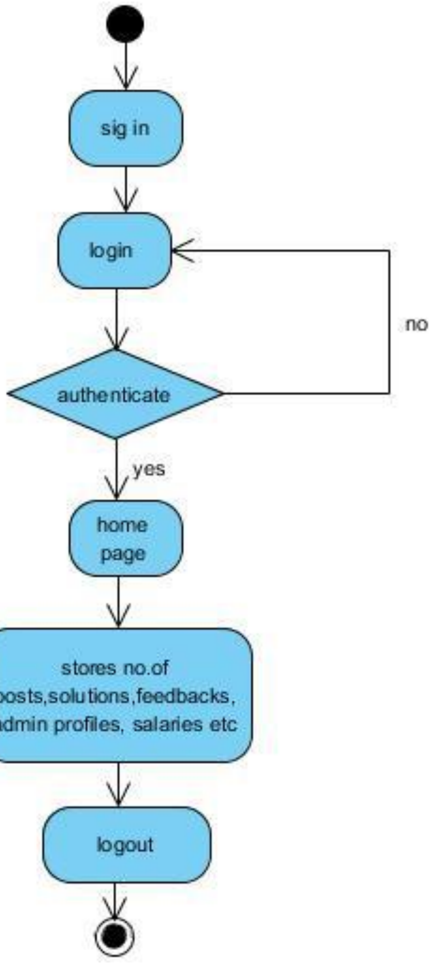
Class Diagram

Class Diagram is a static structure diagram that describes the structure of a system by showing the systems classes, their attributes, operations and the relationships between them.



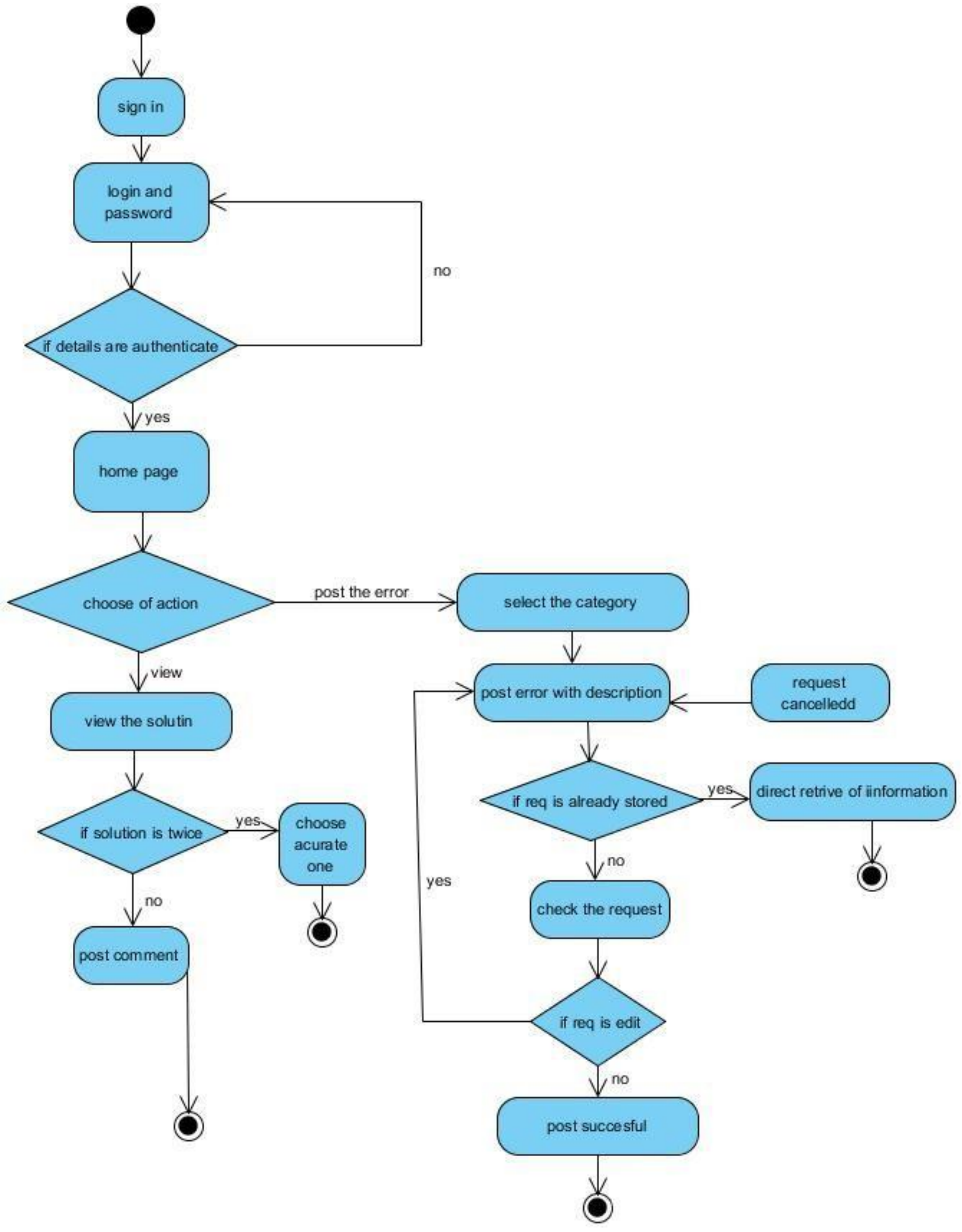
Activity Diagram

Activity Diagrams are graphical representations of workflow of stepwise activities and action with support to choice, iteration and concurrency. It shows the overall flow of control. This are intended to model both computational and organisational processes in the system.

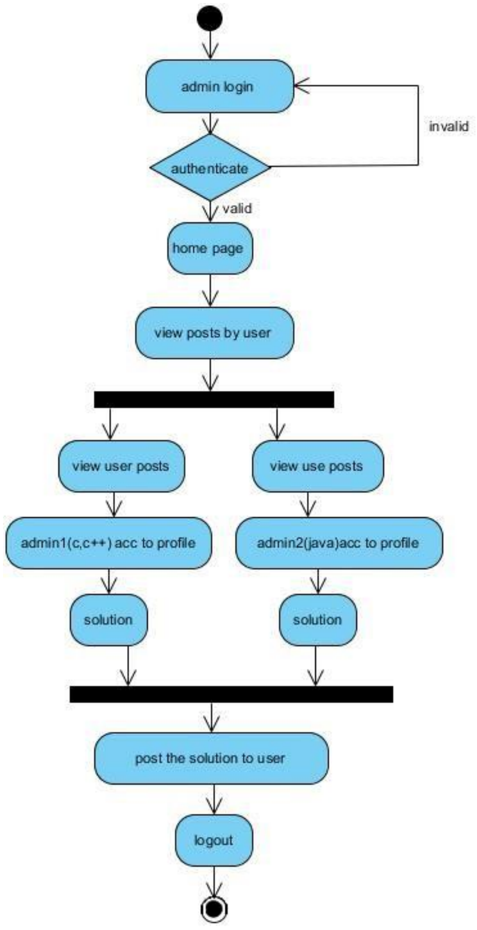

sig in
login
no
authenticate
yes
home page
stores no.of posts,solutions,feedbacks, admin profiles, salaries etc
logout

Activity diagram of manager

Activity diagram of user

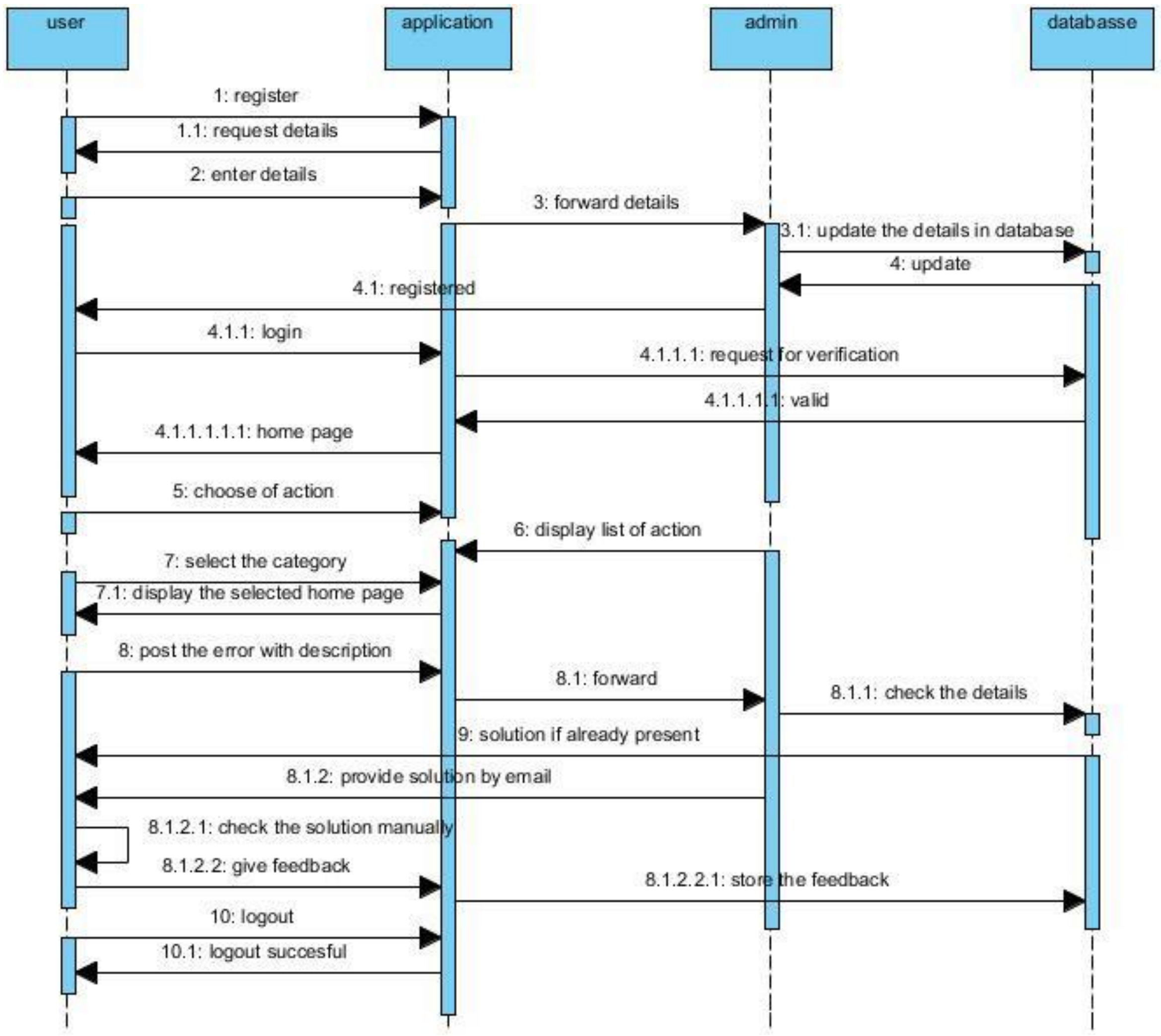


Activity diagram of admin



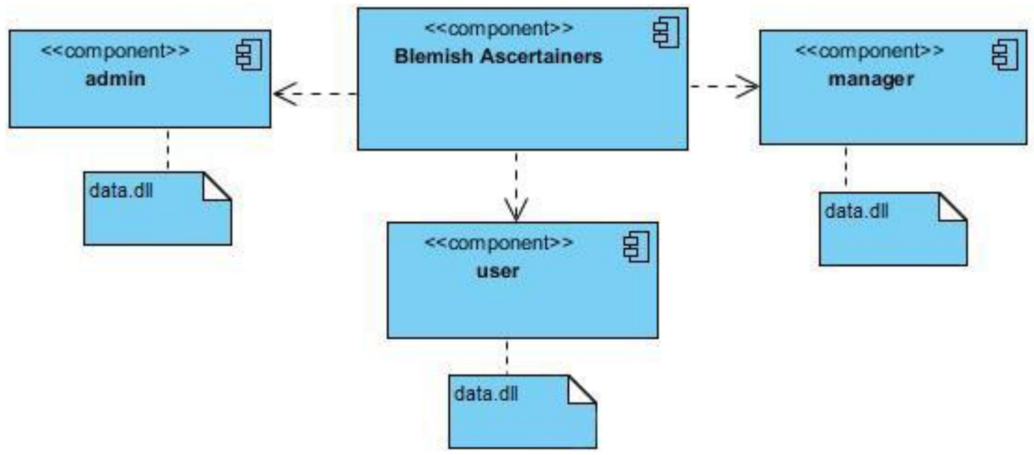
Sequence Diagrams

Sequence Diagram is an interaction diagram that shows how processes operate with one another and in what order. It shows the object interaction arranged in time sequence.



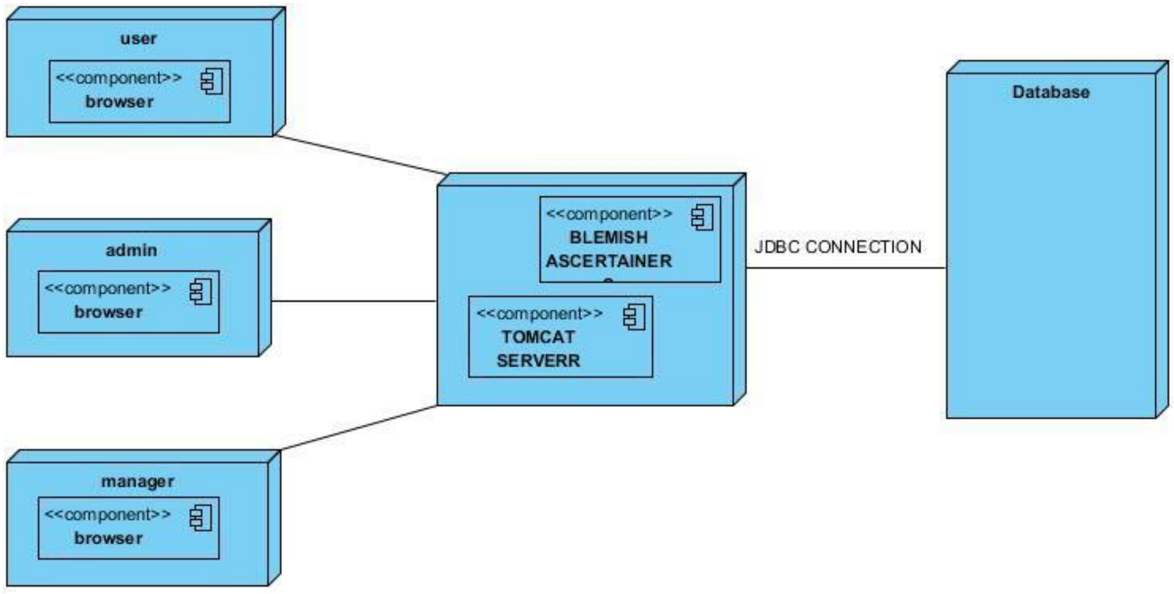
Component Diagram

Component diagrams depicts how components are wired together to form large components and or software systems. Component is something that is required to execute a function like database tables, library, executables etc.



Deployment Diagram

Deployment Diagram is a structure diagram which shows architecture of the system as deployment of software artifacts. These artifacts represent the concrete elements of the physical world that are result in the development process.



7. TESTING

Software testing is a critical element of software quality assurance and represents the ultimate review of specification, design and code generation.

7.1 Testing Objectives:

- To ensure that during operation the system will perform as per specification.
- To make sure that system meets the user requirements during operation
- To make sure that during the operation, incorrect input, processing and output will be detected
- To see that when correct inputs are fed to the system the outputs are correct
- To verify that the controls incorporated in the same system as intended

Testing is a process of executing a program with the intent of finding an error
A good test case is one that has a high probability of finding an as yet undiscovered error
The software developed has been tested successfully using the following testing strategies and any errors that are encountered are corrected and again the part of the program or the procedure or function is put to testing until all the errors are removed. A successful test is one that uncovers an as yet undiscovered error.

Note that the result of the system testing will prove that the system is working correctly. It will give confidence to system designer, users of the system, prevent frustration during implementation process etc.

7.2 Test Case

White box testing:

White box testing is a testing case design method that uses the control structure of the procedure design to derive test cases. All independents path in a module are exercised at least once, all logical decisions are exercised at once, execute all loops at boundaries and within their operational bounds exercise internal data structure to ensure their validity. Here the customer is given three chances to enter a valid choice out of the given menu. After which the control exits the current menu.

Black Box Testing:

Black Box Testing attempts to find errors in following areas or categories, incorrect or missing functions, interface error, errors in data structures, performance error and initialization and termination error. Here all the input data must match the data type to become a valid entry.

The following are the different tests at various levels:

Unit Testing:

Unit testing is essentially for the verification of the code produced during the coding phase and the goal is test the internal logic of the module/program. In the Generic code project, the unit testing is done during coding phase of data entry forms whether the functions are working properly or not. In this phase all the drivers are tested they are rightly connected or not.

Integration Testing:

All the tested modules are combined into sub systems, which are then tested. The goal is to see if the modules are properly integrated, and the emphasis being on the testing interfaces between the modules. In the generic code integration testing is done mainly on table creation module and insertion module.

Validation Testing:

This testing concentrates on confirming that the software is error-free in all respects. All the specified validations are verified and the software is subjected to hard-core testing. It also aims at determining the degree of deviation that exists in the software designed from the specification; they are listed out and are corrected.

System Testing:

This testing is a series of different tests whose primary is to fully exercise the computer-based system. This involves:

- Implementing the system in a simulated production environment and testing it.
- Introducing errors and testing for error handling.

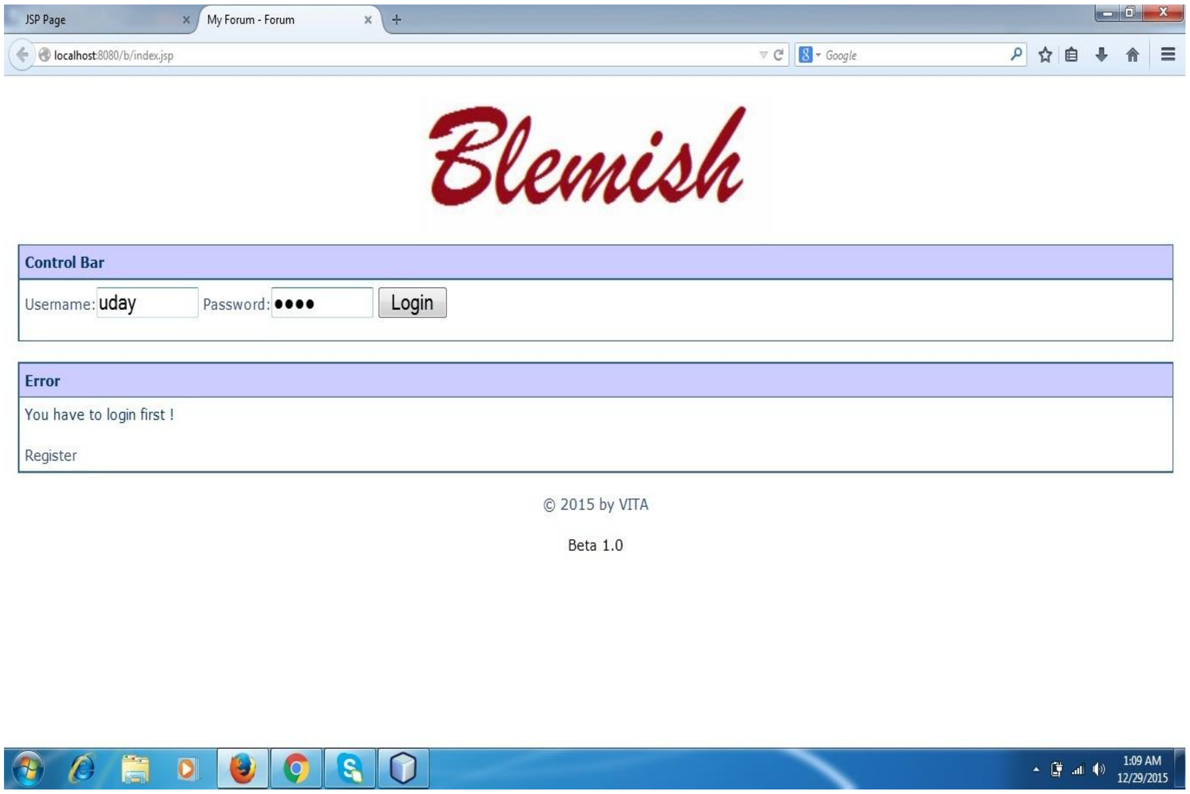
7.3 Test Case Design:

Test Case	Check Item	Test case Objective	Test Data / Input	Expected Result	Actual Result	Results	Remarks
TC-001	**Log-in Page**	**Leave all fields as blank and click Log-in button**		**By leaving all fields as blank and on click Log-in button then mandatory symbol (*) should appear in front of Username**	**PASS**	**PASS**	**PASS, since no field data is entered**
TC-002	**Username**	**Enter Invalid Username**	**usernam e:Udya**	**By entering invalid Username then an error message should appear as " Please Enter Valid Username "**	**PASS**	**PASS**	**PASS By adding invalid user name it does not allow to**
TC-003	**Username**	**Enter valid Username**	**Usernam e :Uday**	**It should allow the user to proceed**	**PASS**	**PASS**	**PASS userna me is checked in the DB. If present the login procedu re takes**
TC-004	**Password**	**Enter your password**	**Input:** ****	**The password field should display the encrypted format of the text typed as (****)**	**PASS**	**PASS**	**PASS ’cause text type is in encrypt ed**

TC-005	**Password**	**Enter wrong password**	**Password : *****	**By entering invalid password then an error message should appear as " Please Enter Correct Password "**	**PASS**	**PASS**	**PASS, after entering wrong password the message has**
TC-006	**Password**	**Enter Correct password**	**Password : *********	**It should allow the user to proceed**	**PASS**	**PASS**	**PASS Allows to login**
TC-007	**Log-in button**	**Correct Inputs**	**Log-in:”Button”**	**It should lead the user to the respect page**	**PASS**	**PASS**	**PASS Sucessfully sent to respective home**
TC-008	**Forgot Password**	**Check hyperlink on Forgot Password label**	**-----------**	**while mouse over of the label an hand icon should display**	**FAIL**	**FAIL**	**FAIL since no such link availabl**
TC-009	**Forgot Password**	**----------**	**--------------**	**User can recover the password using the “Forgot Password” link**	**FAIL**	**FAIL**	**FAIL No such link present**
TC-010	**Registration**	**Check hyperlink on Registration label**	**Moving the mouse on the registration label**	**while mouse over of the label an hand icon should display**	**PASS**	**PASS**	**PASS, since hand icon is appeared on mouseo**
TC-011	**Registration**	**Clicking on the Registration button**	**Registration button**	**On click " Registration " page should redirect to the User Registration page**	**PASS**	**PASS**	**PASS Redirected to the registration**

8.SCREENSHOTS

8.1.User and Admin login



Control Bar

Username: uday Password: ●●●● Login

Error

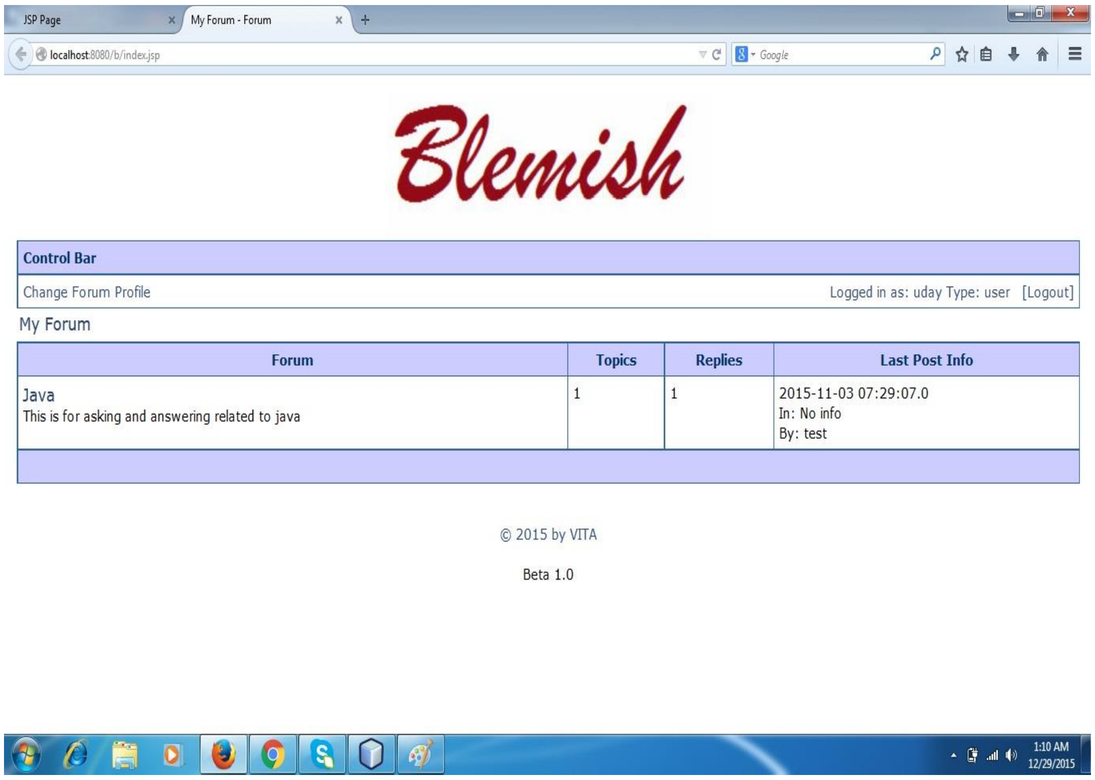
You have to login first !

Register

© 2015 by VITA

Beta 1.0

User Profile-Forum page



Control Bar	
Change Forum Profile	Logged in as: uday Type: user [Logout]

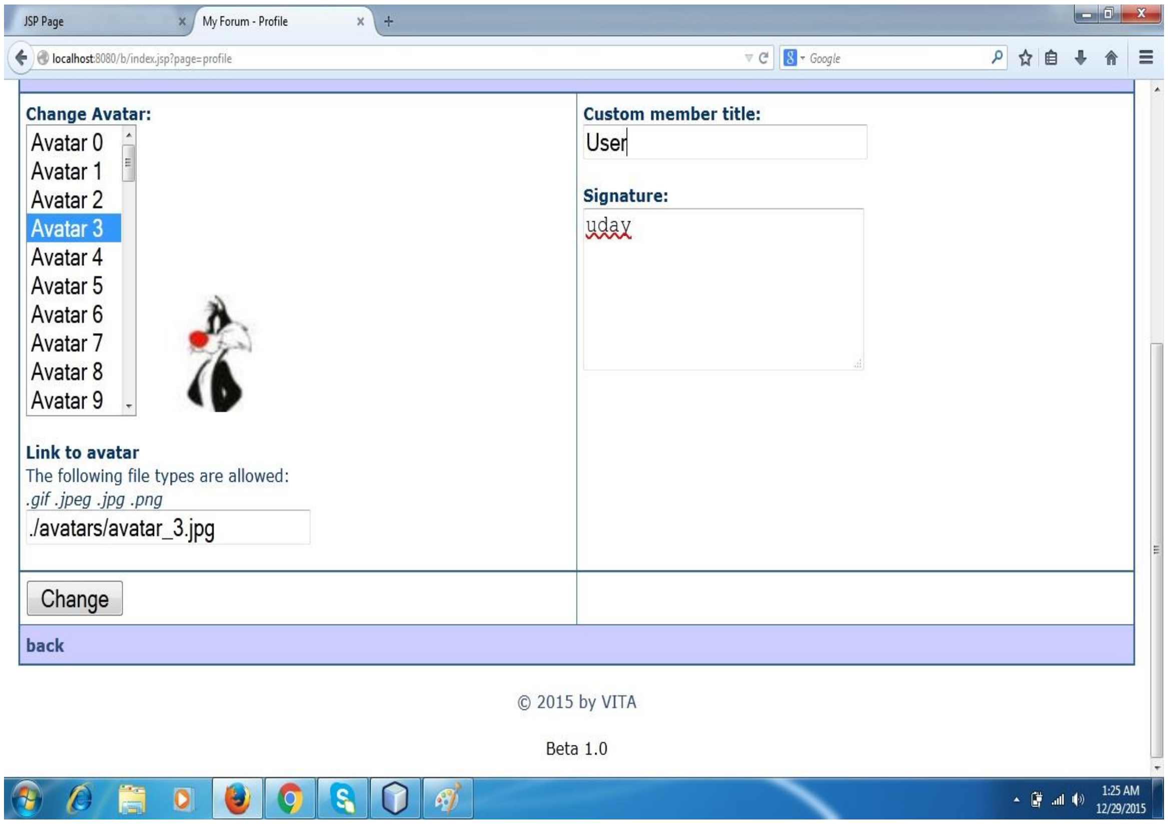
My Forum

Forum	Topics	Replies	Last Post Info
Java This is for asking and answering related to java	1	1	2015-11-03 07:29:07.0 In: No info By: test

© 2015 by VITA

Beta 1.0

User Profile-Edit Profile



Change Avatar:

Avatar 0
Avatar 1
Avatar 2
Avatar 3
Avatar 4
Avatar 5
Avatar 6
Avatar 7
Avatar 8
Avatar 9

Link to avatar
The following file types are allowed:
.gif .jpeg .jpg .png
./avatars/avatar_3.jpg

Custom member title:
User

Signature:
uday

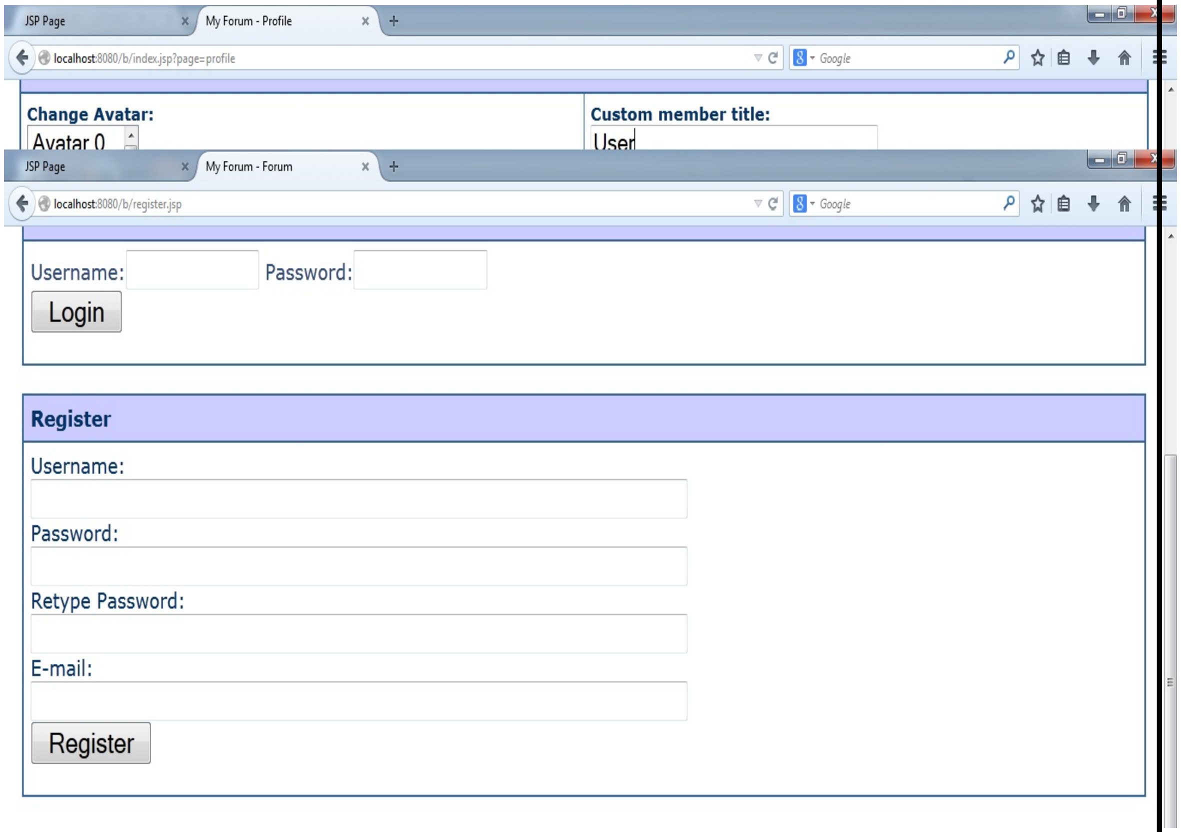
Change

back

© 2015 by VITA

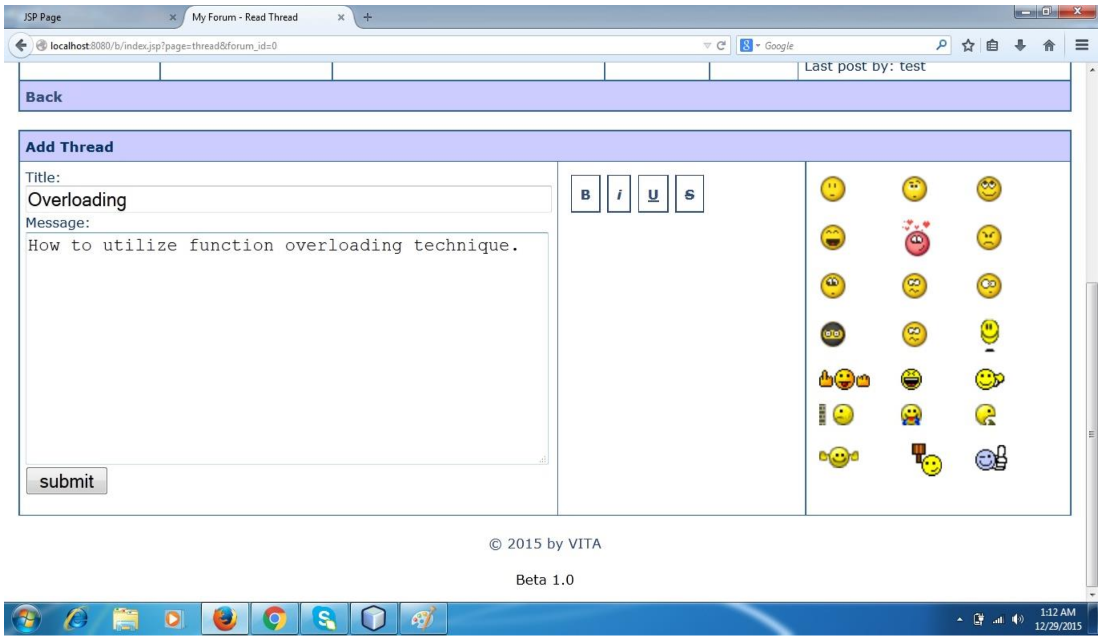
Beta 1.0

User Register


JSP Page
My Forum - Profile
localhost:8080/b/index.jsp?page=profile
Change Avatar:
Avatar 0
Custom member title:
User
JSP Page
My Forum - Forum
localhost:8080/b/register.jsp
Username: Password:
Login

Register
Username:
Password:
Retype Password:
E-mail:
Register

User-Add thread

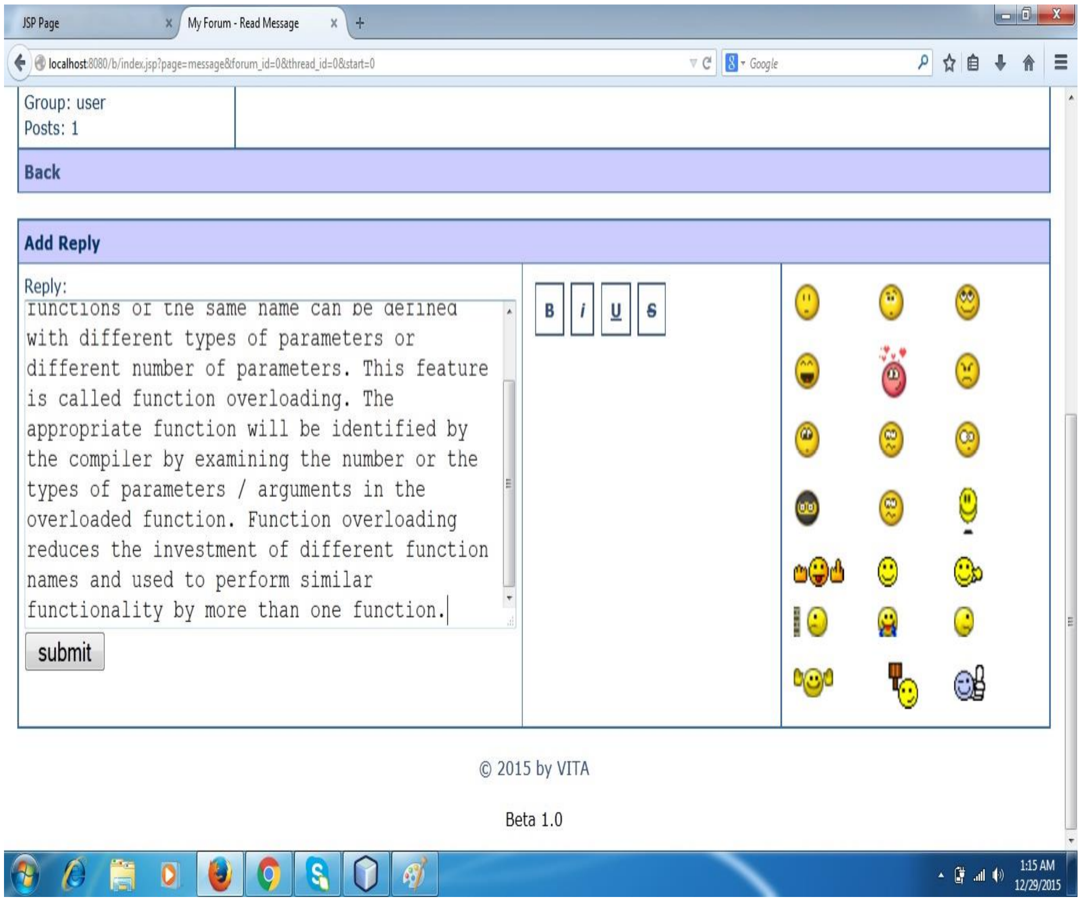


Add Thread

Title: Overloading

Message: How to utilize function overloading technique.

Admin-Add reply


JSP Page
My Forum - Read Message
localhost:8080/b/index.jsp?page=message&forum_id=0&thread_id=0&start=0
Google
B
i
U
S
submit
1:15 AM
12/29/2015

Group: user
Posts: 1

Back

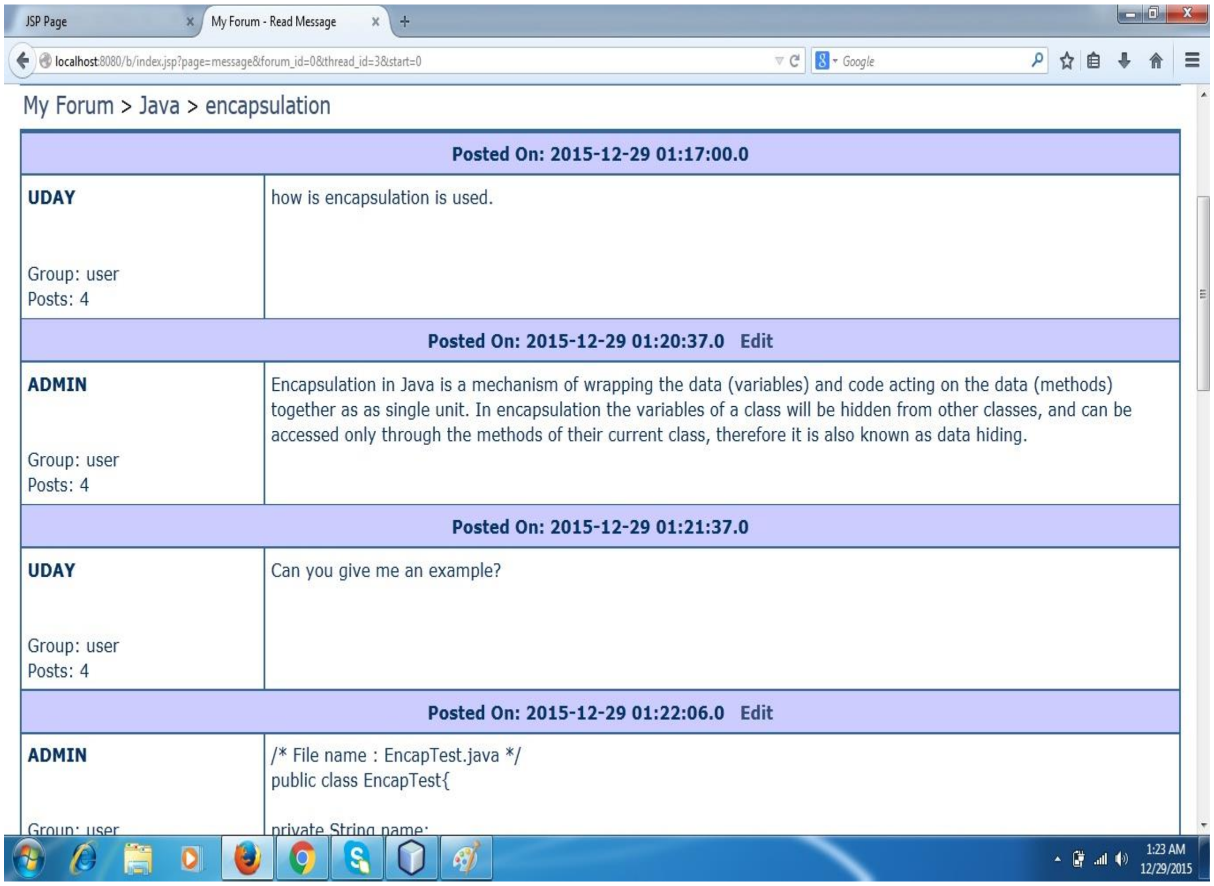
Add Reply

Reply:

functions of the same name can be defined with different types of parameters or different number of parameters. This feature is called function overloading. The appropriate function will be identified by the compiler by examining the number or the types of parameters / arguments in the overloaded function. Function overloading reduces the investment of different function names and used to perform similar functionality by more than one function.

© 2015 by VITA

Beta 1.0



JSP Page | My Forum - Read Message

localhost:8080/b/index.jsp?page=message&forum_id=0&thread_id=3&start=0

My Forum > Java > encapsulation

Posted On: 2015-12-29 01:17:00.0	
UDAY Group: user Posts: 4	how is encapsulation is used.
Posted On: 2015-12-29 01:20:37.0 Edit	
ADMIN Group: user Posts: 4	Encapsulation in Java is a mechanism of wrapping the data (variables) and code acting on the data (methods) together as as single unit. In encapsulation the variables of a class will be hidden from other classes, and can be accessed only through the methods of their current class, therefore it is also known as data hiding.
Posted On: 2015-12-29 01:21:37.0	
UDAY Group: user Posts: 4	Can you give me an example?
Posted On: 2015-12-29 01:22:06.0 Edit	
ADMIN Group: user	/* File name : EncapTest.java */ public class EncapTest{ private String name;

1:23 AM 12/29/2015

Solved.

9.CONCLUSION

It is a web-based application which is developed to solve programmers or developers errors in programming and also solving other related problems while programming. Our Application will take the responsibility for providing the rightful solution for the user within a short time so that the user need not be waiting for a long time in search of solutions. And he would be assigned with a guaranteed solution until the problem gets solved

Bibliography

References

a. Java Complete Reference by Herbert Schildt

b. Head First Servlets and JSP by O'Reilly

c. http://www.ijcea.com/wp-content/uploads/2014/11/03Gaurav-Kakariya-et- al..pdf

d. S. Bugiel, S. Nurnberger, A. Sadeghi, and T. Schneider.

e. Thinking In Java By Bruce Eckel

f. HTML and CSS: Design and Build Websites by Jon Duckett

g. JavaScript and JQuery: Interactive Front-End Web Development by Jon Duckett

h. Learning Web Design: A Beginner's Guide to HTML, CSS, JavaScript

i. http://www.w3schools.com/sql/

ii. http://net.tutsplus.com/tutorials/html-css-techniques/30-html-best- practices-for-beginners/

iii http://www.learn-javascript-tutorial.com/