

CHAPTER 1

INTRODUCTION

1.1 PROBLEM DEFINITION

Various Assessment applications are currently being used in the process of recruiting the employees for an organization and they are paying a huge amount for that to the Third party vendor. Assessment Application is a web service based one, which is having further more features than the third party vendor applications, and also in which it supports customization based on the organization's present needs which will be beneficial to the organization in long run.

1.2 OBJECTIVES

To develop an Assessment Application which provides

- Organisation based customization.
- Low cost.
- Upgradability with upcoming database technologies.
- Lightweight processing.
- Helps in easy recruitment and organization development process.
- Concurrency.
- High availability or disaster recovery
- Storage and compression
- Ease of integration with external libraries

- Full sql support vs partial support
- Pricing
- Integration with reporting and archive tools
- Backup option

1.3OVERVIEW

Assessment Application is an application used for the purpose of conducting assessments which helps in understanding the knowledge gained by the users using it. It will be much more useful in the recruitment process of any organization and the specialty of using it is customization based on the present needs of the organization and it leads to the productive development of the organization. The organization's objectives can be achieved and much more efficiency will lead to higher production and much beneficial to the organization.

CHAPTER 2

REQUIREMENT SPECIFICATION

2.1 SYSTEM REQUIREMENT

To be used efficiently all computer software needs certain hardware components or other software resources to be present on a computer. These requirements are known as (computer) system requirements. Most software defines two set of requirements: minimum and recommended. With increasing demand for higher processing power and resources in newer version of software, system requirements tend to increase over time. Industry analysts suggest that this trend plays a bigger part in driving updates to existing computer system than technological advancements. System requirements are all of the requirements at the *system level* that describe the functions which the system as a whole should fulfill to satisfy the stakeholder needs and requirements

2.2 FUNCTIONAL AND NON FUNCTIONAL REQUIREMENT

2.2.1 Functional Requirement

Functional requirement as the product capabilities are things that a product must do for its user. Functional requirements define how software behaves to meet user needs. A functional requirement is a requirement that, when satisfied, will allow the user to perform some kind of function. In Software engineering and systems engineering, a functional requirement defines a function of a system or its component. A function is described as a set of inputs, the behavior, and outputs.

2.2.2 Non Functional Requirement

Nonfunctional requirements as the quality attributes, design and implementation constraints, and external interfaces which a product must have. Quality attributes are often affectionately called the “ilities” because the names of many of them end in ility. Examples of quality attributes include availability, maintainability, performances, portability, reliability, robustness, security, scalability, testability, usability and others. Many nonfunctional requirements are global in nature; they applied to an entire system.

2.3 HARDWARE AND SOFTWARE SPECIFICATION

2.3.1 Hardware Requirements

The most common set of requirements defined by any operating system or software application is the physical computer resources, also known as hardware. A hardware requirement list is often accompanied by a Hardware Compatibility List (HCL), especially in case of operating systems. An HCL lists tested, compatible and sometimes incompatible hardware devices for a particular operating system or application.

- ✓ PROCESSOR :Intel core 2 duo and advance
- ✓ SPEED :2.0 GHz
- ✓ MEMORY :2 GB RAM and above
- ✓ HARD DISK DRIVE : 100 GB and above.

2.3.2 Software Specification

Software requirements deal with defining software resource requirements that need to be installed on a computer to provide optimal functioning of an application. These requirements are generally not included in the software

installation package and need to be installing separately before the software is installed.

- ✓ PLATFORM : Windows 7
- ✓ LANGUAGE USED : JAVA, HTML, CSS(Bootstrap),Java script,
Jquery, Angular JS.
- ✓ DATABASE USED : Apache derby
- ✓ TOOLS USED : Eclipse Kepler

CHAPTER 3

SYSTEM ANALYSIS

3.1 SYSTEM ANALYSIS

Systems are created to solve problems. One can think of the systems approach as an organized way of dealing with a problem. In this dynamic world, the subject system analysis and design, mainly deals with the software development activities. A collection of components that work together to realize some objectives forms a system. Basically there are three major components in every system, namely input, processing and output.

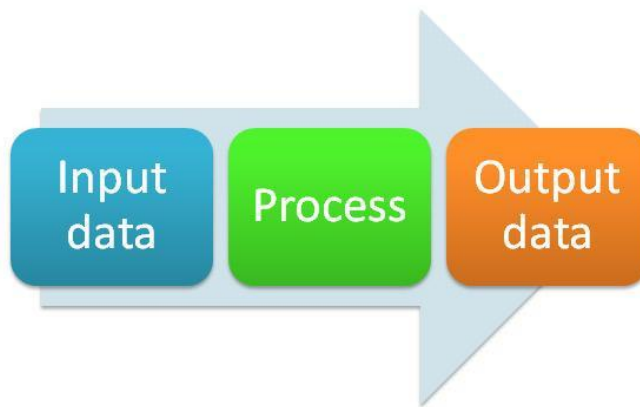


FIGURE: 3.1 SYSTEM PROCESS

In a system the different components are connected with each other and they are interdependent. The objective of the system demands that some output is produced as a result of processing the suitable inputs. A well designed system also includes an additional element referred to as ‘control’ that provides a way feedback to achieve desired objectives of the system.

System analysis is a process of collecting factual data, understand the process involved, identifying problems and recommending feasible suggestions for

improving the system functioning. This involves studying the business processes, gathering operational data, understand the information flow, finding out bottlenecks and evolving solutions for overcoming the weaknesses of the system so as to achieve the organizational goals. System analysis also includes subdividing of complex process involving the entire system, identification of data store and manual processes. The major objectives of system analysis are to find answers for each business processes:

What is being done, How is it being done, Who is doing it, When is he doing it, Why is it being done and How can it be improved? It is more of a thinking process and involves the creative skills of the system analyst.

It attempts to give birth to a new efficient system that satisfies the current needs of the user and has scope for future growth within the organizational constraints. The result of this process is a logical system design, System analysis is an iterative process that continues until a preferred and acceptable solution emerges.

3.1.1 System Life Cycle

System life cycle is an organizational process of developing and maintaining systems. It helps in establishing a system project plan, because it gives overall list of processes and sub-processes required for developing a system. System development life cycle means combination of various activities. In other words we can say that various activities put together are referred as system development life cycle. In the system analysis and design terminology, the system development life cycle also means software development life cycle. The different phase of system development life cycle is shown in this diagram.

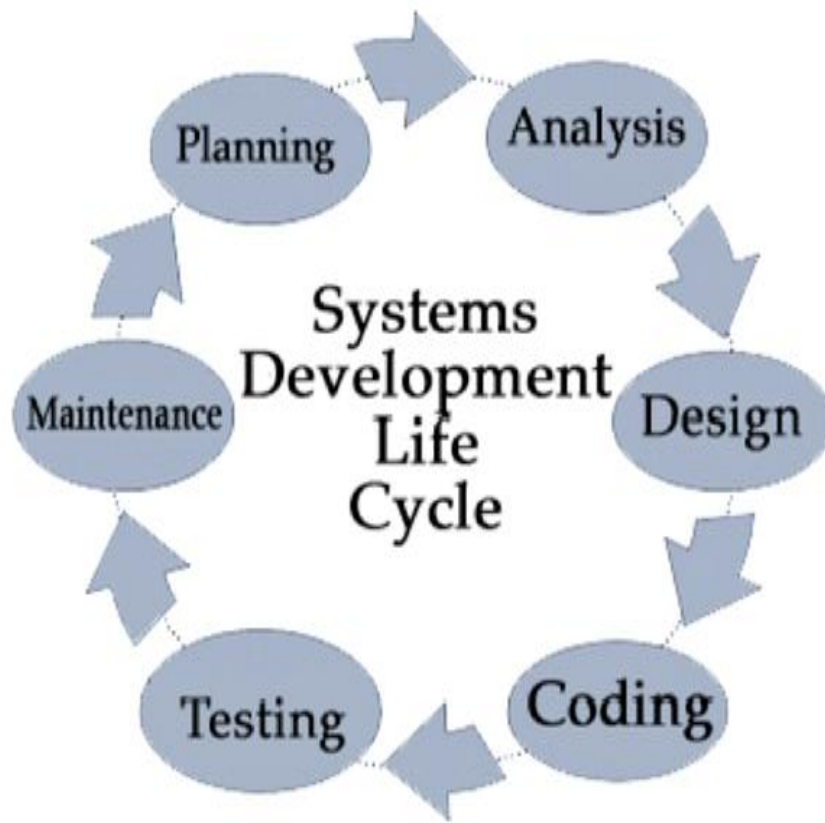


FIGURE: 3.2 SYSTEM LIFE CYCLES

3.2 EXISTING SYSTEM

The existing thirdparty applications are designed in a fixed way so that the organizations needs to pay alone for their customized designing and any specialization that need to be wished by the organization should be done by the vendor providing it ,any changes in the application must be carried out only with the help and support of the vendor providing the application.

3.2.1 DRAWBACKS OF EXISTING SYSTEM

- **PRICE:**

The costs of the third party vendor applications are cost wise higher and even though the organization gets the service, these organizations cannot be able to work independently on its own pace. Any modifications in those applications can be done only with the vendor support.

- **INABILITY TO EXPAND:**

There are several applications which uses SQL database and it has the inability to withstand when the number of records exceeds certain limits, the ability to retrieve gets slow and as it is web service based application the user may have a slower response as the database gets loaded.

- **Migration to other database:**

There are so many third party applications which doesn't supports No SQL databases , it cannot be further scaled up to a higher data contented database and it will not be useful for the organization in a long term use.

3.3 PROPOSED SYSTEM

The proposed system is based on completely supportable to the organization in all cases. The proposed system supports No SQL database as it uses java based spring REST web services. The customization of organization based assessment can be done with the proposed system. The proposed system is lightweight and it doesn't get slow over loading of data.

3.3.1 FEATURES OF PROPOSED SYSTEM

- Lightweight: Spring is lightweight when it comes to size and transparency. The basic version of spring framework is around 2MB.

- Inversion of control:

Loose coupling is achieved in spring using the technique Inversion of Control. It minimizes the amount of code in your application.

- Dependency Injection:

You do not create your objects but describe how they should be created. You don't directly connect your components and services together in code but describe which services are needed by which components in a configuration file.

- Container:

Spring contains and manages the life cycle and configuration of application objects.

3.4 DATABASE

A database is an organized collection of data. The data is typically organized to model relevant aspects of reality (for example, the availability of rooms in hotels), in a way that supports processes requiring this information (for example, finding a hotel with vacancies). A general purpose database management system (DBMS) is a software system design to allow the definition, creation, querying, update and administration of databases. Well known DBMSs include Oracle database.

A database transaction is a sequence of actions that are treated as a single unit of work. These actions should either complete entirely or take no effect at all. Transaction management is an important part of and RDBMS oriented enterprise applications to ensure data integrity and consistency. The concept of transactions can be described with following four key properties described as **ACID**:

- **Atomicity:** A transaction should be treated as a single unit of operation which means either the entire sequence of operations is successful or unsuccessful.
- **Consistency:** This represents the consistency of the referential integrity of the database, unique primary keys in tables etc.
- **Isolation:** There may be many transactions processing with the same data set at the same time, each transaction should be isolated from others to prevent data corruption.
- **Durability:** Once a transaction has completed, the results of this transaction have to be made permanent and cannot be erased from the database due to system failure.

A real RDBMS database system will guarantee all the four properties for each transaction. The simplistic view of a transaction issued to the database using SQL is as follows:

- Begin the transaction using *begin transaction* command.
- Perform various deleted, update or insert operations using SQL queries.
- If all the operation are successful then perform *commit* otherwise *rollback* all the operations.

Spring framework provides an abstract layer on top of different underlying transaction management APIs. The spring's transaction support aims to provide an alternative to EJB transactions by adding transaction capabilities to POJOs. Spring supports both programmatic and declarative transaction management. EJBs

requires an application server, but Spring transaction management can be implemented without a need of application server.

3.4.1 LOCAL VS GLOBAL TRANSACTIONS

Local transactions are specific to a single transactional resource like a JDBC connection, whereas global transactions can span multiple transactional resources like transaction in a distributed system.

Local transaction management can be useful in a centralized computing environment where application components and resources are located at a single site, and transaction management only involves a local data manager running on a single machine. Local transactions are easier to be implemented.

Global transaction management is required in a distributed computing environment where all the resources are distributed across multiple systems. In such a case transaction management needs to be done both at local and global levels. A distributed or a global transaction is executed across multiple systems, and its execution requires coordination between the global transaction management system and all the local data managers of all the involved systems.

3.4.2 PROGRAMATIC VS DECLARATIVE

Spring supports two types of transaction management:

- **Programmatic transaction management:** This means that you have managed the transaction with the help of programming. That gives you extreme flexibility, but it is difficult to maintain.

- Declarative transaction management: This means you separate transaction management from the business code. You only use annotations or XML based configuration to manage the transactions.

Declarative transaction management is preferable over programmatic transaction management though it is less flexible than programmatic transaction management, which allows you to control transactions through your code. But as a kind of crosscutting concern, declarative transaction management can be modularized with the AOP approach. Spring supports declarative transaction management through the Spring AOP framework.

3.4.3 DERBY

Apache **Derby** (previously distributed as IBM Cloudscape) is a relational **database** management system (RDBMS) developed by the Apache Software Foundation that can be embedded in Java programs and used for online transaction processing. It has a 2.6 MB disk-space footprint.

The Derby network server increases the reach of the Derby database engine by providing traditional client server functionality. The network server allows clients to connect over TCP/IP using the standard DRDA protocol.

Derby comes with Java 7 and has been branded as "JavaDB" but is exactly the same bit-for-bit as Derby is. For developers wanting to use Java 6, they can still download Derby as before, but for developers requiring JRE 7 or later, Derby is included in the Java API.

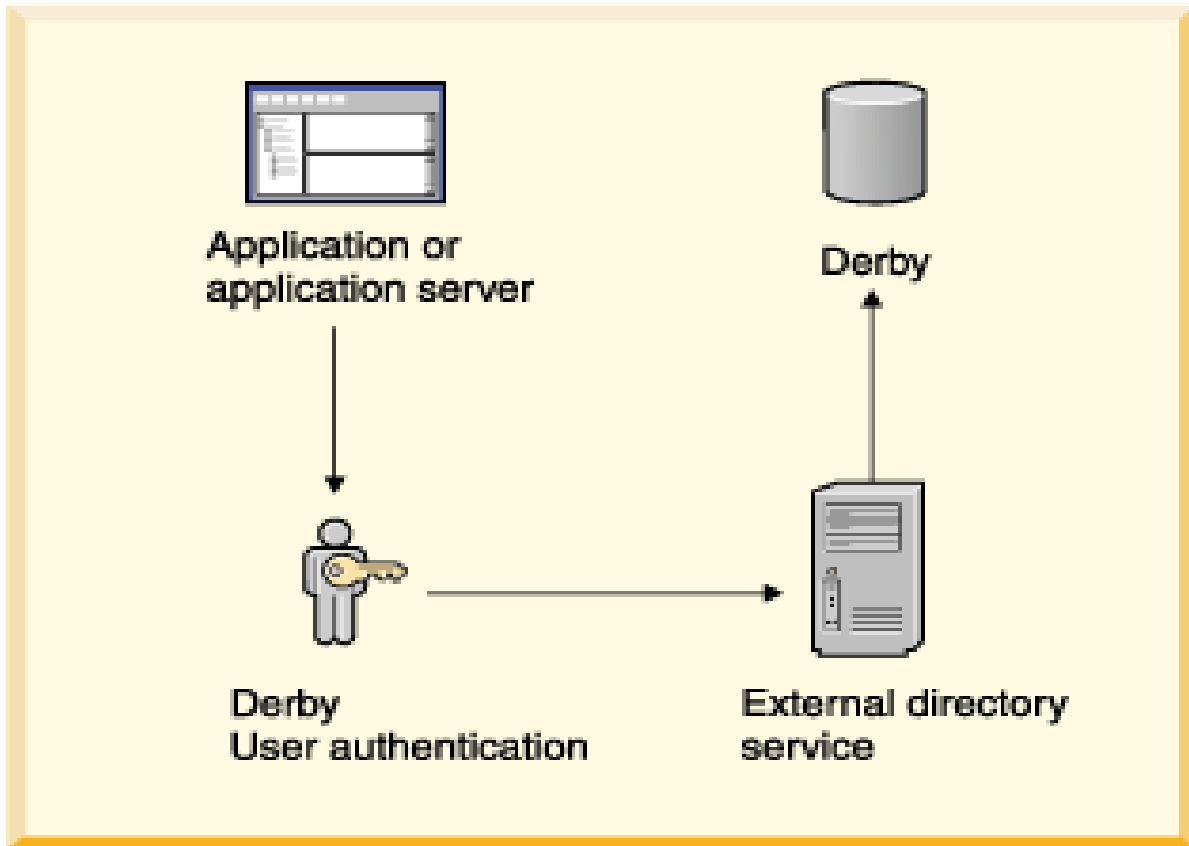


FIGURE: 3.3 DERBY DATABASE

3.5 SYSTEM FEATURES

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 recompilation. 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 merged into Oracle Corporation) and released in 1995 as a core component of Sun Microsystems' Java platform. The language derives much of its syntax from C and C++, but it has fewer low-level facilities than either of them.

Java has gained enormous popularity since it first appeared. Its rapid ascension and wide acceptance can be traced to its design and programming features, particularly in its promise that you can write a program once, and run it anywhere. Java was chosen as the programming language for network computers (NC) and has been perceived as a universal front end for the enterprise database. As stated in Java language white paper by Sun Microsystems: "Java is a simple, object-oriented, distributed, interpreted, robust, secure, architecture neutral, portable, multithreaded, and dynamic."

The advantages of Java are as follows:

- Java is easy to learn.
- Java was designed to be easy to use and is therefore easy to write, compile, debug, and learn than other programming languages.
- Java is object-oriented.
- Java allows you to create modular programs and reusable code.
- Java is platform-independent.

One of the most significant advantages of Java is its ability to move easily from one computer system to another. The ability to run the same program on many different systems is crucial to World Wide Web software, and Java

succeeds at this by being platform-independent at both the source and binary levels.

Hyper Text Markup Language, commonly referred to as HTML, is the standard markup language used to create web pages. It is written in the form of HTML elements consisting of tags enclosed in angle brackets (like `<html>`). HTML tags most commonly come in pairs like `<h1>` and `</h1>`, although some tags represent empty elements and so are unpaired, for example ``. The first tag in a pair is the start tag, and the second tag is the end tag (they are also called opening tags and closing tags).

Web browsers can read HTML files and compose them into visible or audible web pages. Browsers do not display the HTML tags and scripts, but use them to interpret the content of the page. 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. It 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, encourages the use of CSS over explicit presentational HTML.

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog, Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Class path and it runs without problems under Iced Tea.

CHAPTER 4

SOFTWARE PROFILE

4.1 SPRING MVC

The spring web MVC framework provides model-view-controller architecture and ready components that can be used to develop flexible and loosely coupled web applications. The MVC pattern results in separating the different aspects of the application (input logic, business logic, and UI logic), while providing a loose coupling between these elements.

- The **Model** encapsulates the application data and in general they will consist of POJO.
- The **View** is responsible for rendering the model data and in general it generates HTML output that the client's browser can interpret.
- The **Controller** is responsible for processing user requests and building appropriate model and passes it to the view for rendering.

Following is the list of few of the great benefits of using Spring Framework:

- Spring enables developers to develop enterprise-class applications using POJOs. The benefit of using only POJOs is that you do not need an EJB container product such as an application server but you have the option of using only a robust servlet container such as Tomcat or some commercial product.
- Spring is organized in a modular fashion. Even though the number of packages and classes are substantial, you have to worry only about ones you need and ignore the rest.

- Spring does not reinvent the wheel instead; it truly makes use of some of the existing technologies like several ORM frameworks, logging frameworks, JEE, Quartz and JDK timers, other view technologies.
- Testing an application written with Spring is simple because environment-dependent code is moved into this framework. Furthermore, by using Java Bean-style POJOs, it becomes easier to use dependency injection for injecting test data.
- Spring's web framework is a well-designed web MVC framework, which provides a great alternative to web frameworks such as Struts or other over engineered or less popular web frameworks.
- Spring provides a convenient API to translate technology-specific exceptions (thrown by JDBC, Hibernate, or JDO, for example) into consistent, unchecked exceptions.
- Lightweight IoC containers tend to be lightweight, especially when compared to EJB containers, for example. This is beneficial for developing and deploying applications on computers with limited memory and CPU resources.
- Spring provides a consistent transaction management interface that can scale down to a local transaction (using a single database, for example) and scale up to global transactions (using JTA, for example).

Spring Session provides an API and implementations for managing a user's session information. It also provides transparent integration with:

Http Session - allows replacing the Http Session in an application container (i.e. Tomcat) neutral way. Clustered Sessions - Spring Session makes it trivial to support clustered sessions without being tied to an application container specific solution. Multiple Browser Sessions - Spring Session supports managing multiple users' sessions in a single browser instance. RESTful APIs - Spring Session allows providing session ids in headers to work with RESTful APIs. Web Socket - provides the ability to keep the Http Session alive when receiving Web Socket messages

4.1.1 SPRING – IOC CONTAINERS

The spring container is at the core of the Spring Framework. The container will create the objects, wire them together, configure them, and manage their complete lifecycle from creation till destruction. The spring container uses dependency injection (DI) to manage the components that make up an application. These objects are called Spring Beans which we will discuss in next chapter.

The container gets its instructions on what objects to instantiate, configure, and assemble by reading configuration metadata provided. The configuration metadata can be represented either by XML, Java annotations, or Java code. The following diagram is a high-level view of how spring works. The Spring IoC container makes use of Java POJO classes and configuration metadata to produce a fully configured and executable system or application.

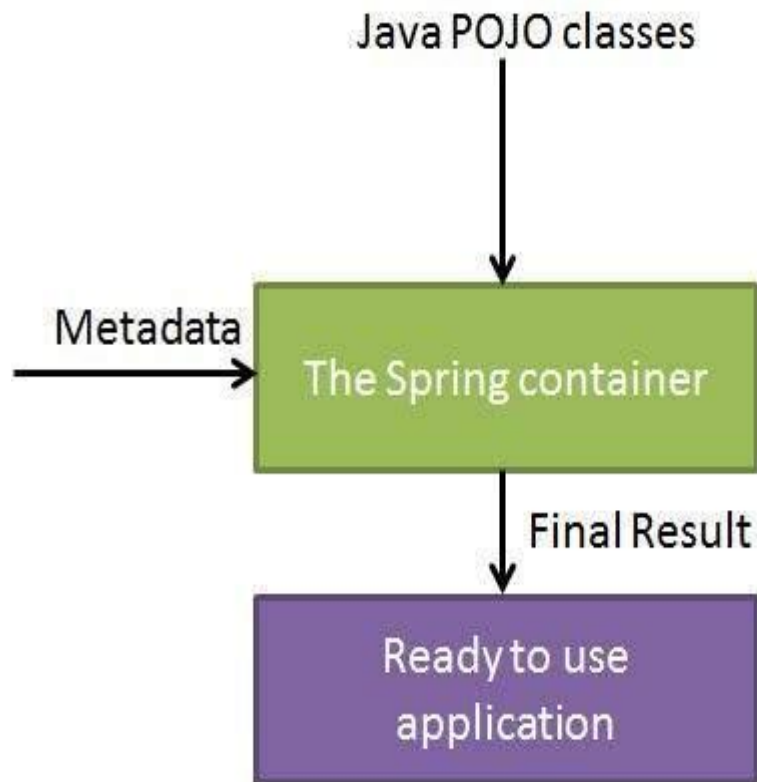


FIGURE: 4.1 SPRING IOC

4.1.2 DEPENDENCY INJECTION (DI)

The technology that spring is most identified with is the **Dependency Injection (DI)** flavor of Inversion of Control. The Inversion of Control (IoC) is a general concept, and it can be expressed in many different ways and Dependency Injection is merely one concrete example of Inversion of Control.

When writing a complex Java application, application classes should be as independent as possible of other Java classes to increase the possibility to reuse these classes and to test them independently of other classes while doing unit

testing. Dependency Injection helps in gluing these classes together and same time keeping them independent.

What is dependency injection exactly? Let's look at these two words separately. Here the dependency part translates into an association between two classes. For example, class A is dependent on class B. Now, let's look at the second part, injection. All this means is that class B will get injected into class A by the IoC.

Dependency injection can happen in the way of passing parameters to the constructor or by post-construction using setter methods.

4.1.3 ASPECT ORIENTED PROGRAMMING (AOP):

One of the key components of spring is the **Aspect oriented programming (AOP)** framework. The functions that span multiple points of an application are called **cross-cutting concerns** and these cross-cutting concerns are conceptually separate from the application's business logic. There are various common good examples of aspects including logging, declarative transactions, security, and caching etc.

The key unit of modularity in OOP is the class, whereas in AOP the unit of modularity is the aspect. Whereas DI helps you decouple your application objects from each other, AOP helps you decouple cross-cutting concerns from the objects that they affect.

The AOP module of Spring Framework provides aspect-oriented programming implementation allowing you to define method-interceptors and point cuts to cleanly decouple code that implements functionality that should be separated.

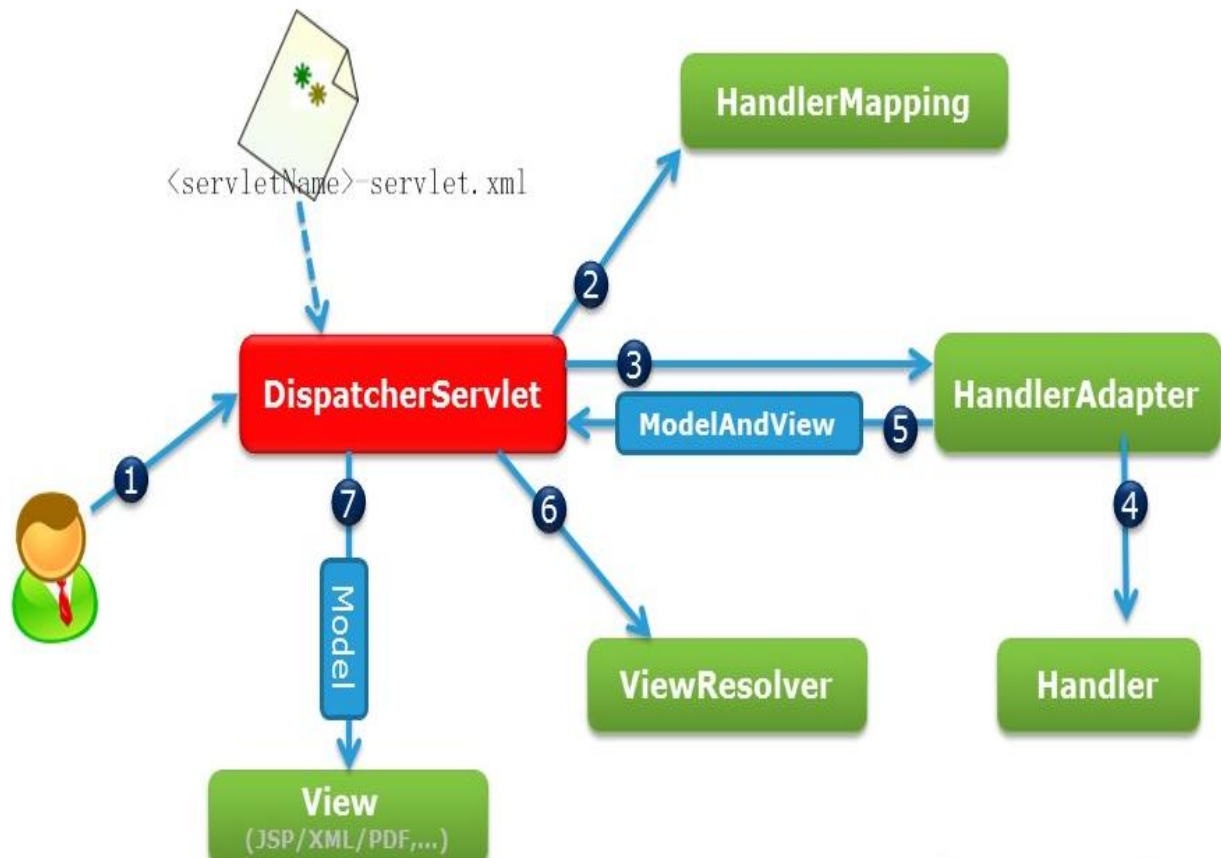


FIGURE: 4.2 SPRING MVC

4.1.4 DISPATCHERSERVLET

The Spring Web model-view-controller (MVC) framework is designed around a DispatcherServlet that handles all the HTTP requests and responses. The request processing workflow of the Spring Web MVC DispatcherServlet is illustrated in the following diagram:

Following is the sequence of events corresponding to an incoming HTTP request to DispatcherServlet:

- After receiving an HTTP request, DispatcherServlet consults the handler mapping to call the appropriate *Controller*.
- The *Controller* takes the request and calls the appropriate service methods based on used GET or POST method. The service method will set model data based on defined business logic and returns view name to the DispatcherServlet.

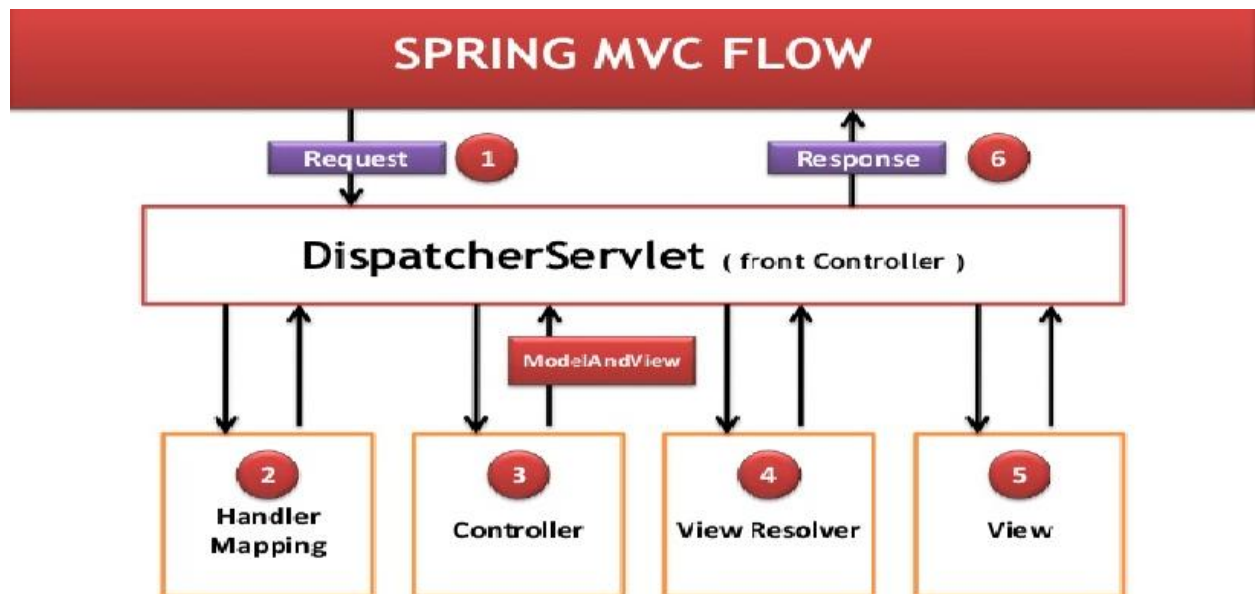


FIGURE: 4.3 DISPATCHERSERVLET

- The DispatcherServlet will take help from ViewResolver to pickup the defined view for the request.
- Once view is finalized, The DispatcherServlet passes the model data to the view which is finally rendered on the browser.

4.1.5 SPRING JDBC FRAMEWORK

Spring JDBC provides several approaches and correspondingly different classes to interface with the database. I'm going to take classic and the most

popular approach which makes use of **JdbcTemplate** class of the framework. This is the central framework class that manages all the database communication and exception handling. Instances of the JdbcTemplate class are thread safe once configured. So you can configure a single instance of a JdbcTemplate and then safely inject this shared reference into multiple DAOs.

The Core Container consists of the Core, Beans, Context, and Expression Language modules whose detail is as follows:

- The **Core** module provides the fundamental parts of the framework, including the IoC and Dependency Injection features.
- The **Bean** module provides BeanFactory which is a sophisticated implementation of the factory pattern.
- The **Context** module builds on the solid base provided by the Core and Beans modules and it is a medium to access any objects defined and configured. The Application Context interface is the focal point of the Context module.
- The **SpEL** module provides a powerful expression language for querying and manipulating an object graph at runtime.

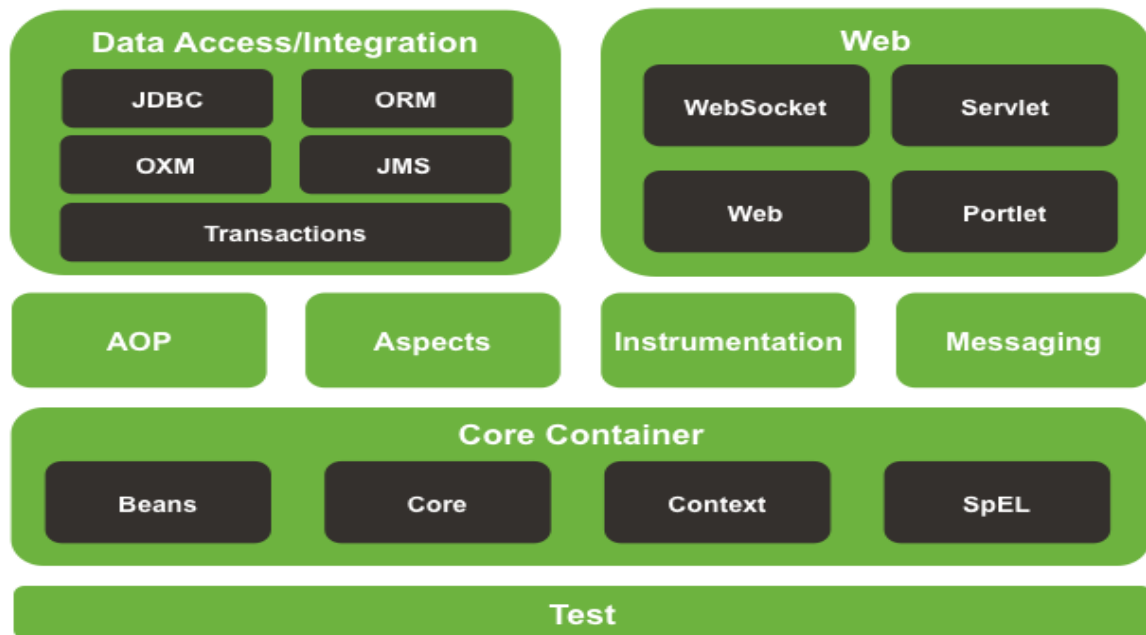


FIGURE: 4.4 SPRING JDBC FRAMEWORK

4.2 WHAT IS JSON?

JSON (JavaScript Object Notation) is a lightweight data-interchange format. It is easy for humans to read and write. It is easy for machines to parse and generate. It is based on a subset of the JavaScript Programming Language, Standard ECMA-262 3rd Edition - December 1999. JSON is a text format that is completely language independent but uses conventions that are familiar to programmers of the C-family of languages, including C, C++, C#, Java, JavaScript, Perl, Python, and many others. These properties make JSON an ideal data-interchange language. JSON uses JavaScript syntax, but the JSON format is text only, just like XML. JSON is easy to read and write. It is a lightweight text-based interchange format. JSON is language independent.

JSON is built on two structures:

- A collection of name/value pairs. In various languages, this is realized as an *object*, record, struct, dictionary, hash table, keyed list, or associative array.
- An ordered list of values. In most languages, this is realized as an *array*, vector, list, or sequence.

4.3 ANGULARJS

AngularJS let you build properly structured web applications very easily.

AngularJS is a MVC framework that defines numerous concepts to properly organize your web application. Your application is defined with modules that can depend from one to the others. It enhances HTML by attaching directives to your pages with new attributes or tags and expressions in order to define very powerful templates directly in your HTML. It also encapsulates the behavior of your application in controllers which are instantiated thanks to dependency injection. Thanks to the use of dependency injection, AngularJS helps you structure and test your JavaScript code very easily.

Angular is a JS framework which implements Model/View/Controller concept in the UI and separates the UI data and the UI representation of data. This division is very important because it now decouples the data from the visual representation. The maximum amount of code people spend writing in frontend, is the translation of data to the DOM and vice versa. Angular completely makes this binding very seamless hence increasing productivity massively.

4.4 BOOTSTRAP

Bootstrap is a free and open-source collection of tools for creating websites and web applications. It contains HTML- and CSS-based design templates for typography, forms, buttons, navigation and other interface components, as well as optional Java Script extensions. It aims to ease the development of dynamic websites and web applications.

Bootstrap is a front end web framework, that is, an interface for the user, unlike the server-side code which resides on the "back end" or server. Bootstrap comes with several JavaScript components in the form of jQuery plugins. They provide additional user interface elements such as dialog boxes, tooltips, and carousels. They also extend the functionality of some existing interface elements, including for example an auto-complete function for input fields. In version 2.0, the following JavaScript plugins are supported: Modal, Dropdown, Scroll spy, Tab, Tooltip, Popover, Alert, Button, Collapse, Carousel and Type ahead.

4.4.1 ADVANTAGES OF BOOTSTRAP

Ease of Use

Bootstrap provides less files for those who are accustomed to CSS preprocessing but if you don't know it or don't want to use it, you can have plain CSS files as well. All you need to do is download the files from Bootstrap's website, unzip them and include them in the head of your HTML files. This gives you the access to the entire framework and you can start using the pre defined bootstrap classes with our markup. Before you start converting your designs to markup it is very important that the designs use the default bootstrap framework grid.

Highly Flexible

Bootstrap gives the developers the flexibility to develop. It's a CSS framework with predefined classes for layout using its grid system, various CSS components and JavaScript functions. These are all included and the developer has the flexibility and freedom of using only those classes that are needed in the markup. This makes it highly flexible as only those objects that are needed in the markup are used by the developer leaving the rest.

Responsive Grid

This is the strongest part of the bootstrap framework. Bootstrap offers a 12 column grid system. The grid system is responsive, that is it adjusts itself depending on the device resolution of the client. These grids have further classes that have been defined in sync with the device resolution that they represent. These grids have classes xs, sm, md and lg each representing a device resolution. All the developer needs to do is include these classes while defining the visibility of an element in the markup and hence come up with a responsive website. The responsive grid makes developing responsive websites really easy using bootstrap.

Comprehensive List of Components

Bootstrap has all the components you would require for your website. It includes Drop Down Menus, Navigation Bar, Progress Bar, Alerts, Labels, Badges etc. It comprehensively includes all the components you may require and the developer can easily include them in the markup. This saves a lot of time as the developer would not have to develop these from scratch and worry about cross browser and cross device compatibility. All that is needed to get them working is to include the correct classes in your markup and things would fall into place.

Leveraging JavaScript Libraries

Bootstrap has over a dozen JavaScript plug-in pre-developed for the developers to be used. There are various functions that are most often used in most of the websites that have been included in the JavaScript libraries of bootstrap. These include Tabs, Scroll Spy, Tooltips, Pop Over's, Buttons, Alerts and lots more. Again using and customizing them is really easy.

Frequent Updates

Bootstrap releases more updates than any other framework. The bootstrap development team as soon as encounters any problem, starts working on the solution. With Bootstrap updates released consistently and frequently, you can rest assured that you are working with the latest tools. This also ensures a wider range of cross browser and cross device compatibility.

Detailed Documentation and Vast Community

Bootstrap has a very detailed documentation and a vast community supporting it. Even if a developer is new to Bootstrap, the documentation provides a great support in learning it without any hassles. The documentation includes examples and demos which help understanding the concepts and getting accustomed to Bootstrap in a short span of time. Despite of such a detailed documentation even if a developer gets stuck somewhere there is a vast community and a lot of forums that help answer questions.

Consistency

Bootstrap was developed with an idea to give developers a centralized development code. Bootstrap provides you the end result which is uniform across platforms. You don't have to worry about compatibility issues with Internet

Explorer, Google chrome or Firefox, it gives you the same performance independent of the platform.

4.5 Apache Maven

Maven is a build automation tool used primarily for Java projects. The word *maven* means "accumulator of knowledge" in Yiddish. Maven addresses two aspects of building software: First, it describes how software is built, and second, it describes its dependencies. Contrary to preceding tools like Apache Ant, it uses conventions for the build procedure, and only exceptions need to be written down. An XML file describes the software project being built, its dependencies on other external modules and components, the build order, directories, and required plug-ins. It comes with pre-defined targets for performing certain well-defined tasks such as compilation of code and its packaging. Maven dynamically downloads Java libraries and Maven plug-ins from one or more repositories such as the Maven 2 Central Repository, and stores them in a local cache. This local cache of downloaded artifacts can also be updated with artifacts created by local projects. Public repositories can also be updated.

Maven can also be used to build and manage projects written in C#, Ruby, Scala, and other languages. The Maven project is hosted by the Apache Software Foundation, where it was formerly part of the Jakarta Project. Maven is built using a plug-in based architecture that allows it to make use of any application controllable through standard input. Theoretically, this would allow anyone to write plug-ins to interface with build tools (compilers, unit test tools, etc.) for any other language. In reality, support and use for languages other than Java has been minimal. Currently a plug-in for the .NET framework exists and is maintained and a C/C++ native plug-in is maintained for Maven.

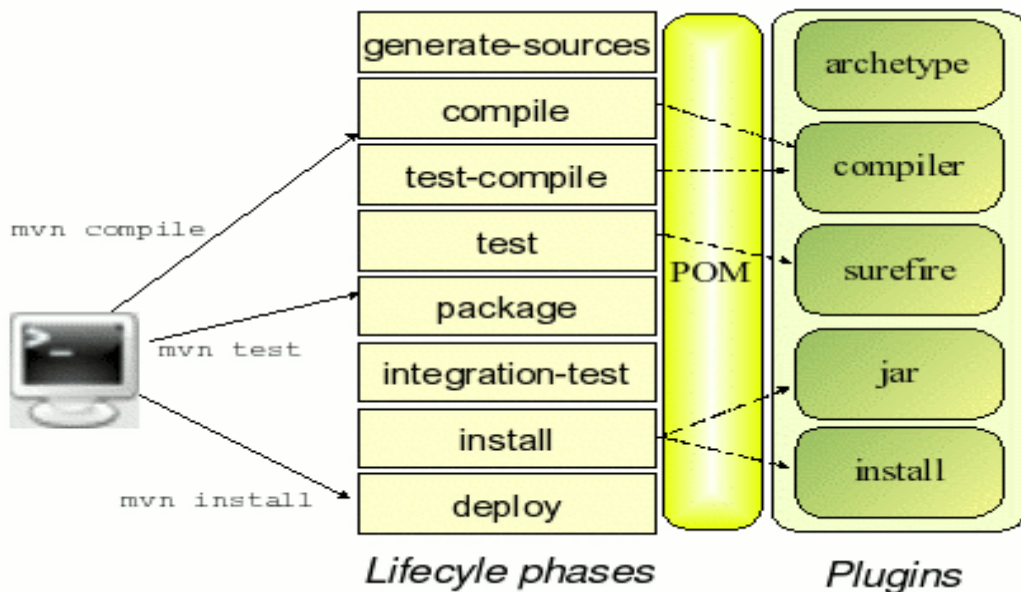


FIGURE: 4.5 MAVEN PHASES

4.6 Apache Tomcat 7.0.53

Apache Tomcat is an open source implementation of the Java Servlet and Java Server Pages technologies. The Java Servlet and Java Server Pages specifications are developed under the Java Community Process.

It is developed in an open and participatory environment and released under the Apache License version 2. Apache Tomcat is intended to be collaboration of the best-of-breed developers from around the world. Apache Tomcat powers numerous large-scale, mission-critical web applications across a diverse range of industries and organizations. It also provides by default a HTTP connector on port 8080, which helps it to work as a HTTP server.

Apache Tomcat, Tomcat, Apache, the Apache feather, and the Apache Tomcat project logo are trademarks of the Apache Software Foundation. Tomcat is one specific open source collaboration from a larger group of open source projects that are collectively famous as the Apache Jakarta Project. Tomcat is an

application server from the Apache Software Foundation that executes Java servlets and renders Web pages that comprise Java Server Page coding. Described as a “reference implementation” of the Java Servlet and the Java Server Page specifications, Tomcat is the result of an open involvement of developers and is accessible from the Apache Web site in both binary and source versions. Tomcat can be used as either a separate product with its own internal Web server or mutually with other Web servers, including Apache, Netscape Enterprise Server, Microsoft Internet Information Server (IIS), and Microsoft Personal Web Server. Tomcat requires a Java Runtime Enterprise Environment that conforms to JRE 1.1.

Tomcat 7 introduced a number of new features as well as enhancements to existing features. Several articles list the new Tomcat 7 features, but most don't explain them in detail, critique them, or provide working code examples. Rather than just list the new features, this article will identify the seven most notable Tomcat 7 features and enhancements, critique them, and present examples of working code that you can use to get a better understanding of each feature/enhancements.

The release of Apache Tomcat 7 has made great strides in improving the overall security and general robustness of the world's most popular application server. In fact, over 450 improvements and issues have been resolved in this latest stable release. While these changes range from small to significant, what is notable is the mature architecture of Apache Tomcat has remained intact as we have seen little problems thus far in the back portability of the application

4.7 JDK 1.7

The Java Development Kit (JDK) is an implementation of either one of the Java SE, Java EE or Java ME platforms released by Oracle Corporation in the form of a binary product aimed at Java developers on Solaris, Linux, Mac OS X or Windows.

The JDK includes a private JVM and a few other resources to finish the recipe to a Java Application. Since the introduction of the Java Platform, it has been by far most widely used Software Development Kit (SDK). On 17 November 2006, sun announced that it would be released under the GNU General Public License (GPL), thus making it free software.

Following is the list JDK 1.7 Features added, of daily importance.

- String in Switch Expression
- Underscores Between Digits in Numeric Literals
- Integral Types as Binary Literals
- Handling multiple exceptions in a single catch block
- Try-with-resources Statement
- Automatic Type Inference in Generic object instantiation

4.8 Eclipse Kepler

In computer programming, Eclipse is an integrated development environment (IDE). It contains a base workspace and an extensible plug-in system for customizing the environment. Written mostly in Java, Eclipse can be used to develop applications. By means of various plug-ins, Eclipse may also be used to develop applications in other programming languages: Ada, ABAP, C, C++, COBOL, Fortran, Haskell, JavaScript, Lasso, Lua, Natural, Perl, PHP, Prolog,

Python, R, Ruby (including Ruby on Rails framework), Scala, Clojure, Groovy, Scheme, and Erlang. It can also be used to develop packages for the software Mathematica. Development environments include the Eclipse Java development tools (JDT) for Java and Scala, Eclipse CDT for C/C++ and Eclipse PDT for PHP, among others.

The initial codebase originated from IBM Visual Age. The Eclipse software development kit (SDK), which includes the Java development tools, is meant for Java developers. Users can extend its abilities by installing plug-ins written for the Eclipse Platform, such as development toolkits for other programming languages, and can write and contribute their own plug-in modules.

Released under the terms of the Eclipse Public License, Eclipse SDK is free and open source software (although it is incompatible with the GNU General Public License). It was one of the first IDEs to run under GNU Class path and it runs without problems under Iced Tea.

CHAPTER 5

SYSTEM TESTING:

5.1 TESTING

Testing a product is relatively independent of the development method used to develop it. The test activity are normally divided into

- Verification
- Validation

Verification checks whether the results agree with the specification. However this alone does not guarantee customer's satisfaction. Validation checks whether its results obtained are actually what was wanted.

There are many types of testing like:

- Unit Testing
- Functional Testing
- System Testing
- Stress Testing
- Performance Testing
- Usability Testing
- Regression Testing
- Black Box Testing
- White Box Testing

5.2 UNIT TESTING:

Unit testing is the testing of an individual unit or group of related units. It falls under the class of white box testing. It is often done by the programmer to test that the unit he/she has implemented is producing the expected output against given input.

5.3 FUNCTIONAL TESTING:

Functional testing is the testing to ensure that the specified functionality required in the system requirements works. It falls under the class of black box testing.

5.4 SYSTEM TESTING:

System testing is the testing to ensure that by putting the software in different environments it still works. System testing is done with full system implementation and environment. It falls under the class of black box testing.

5.5 STRESS TESTING:

Stress testing is the testing to evaluate how the system behaves under unfavorable conditions. Testing is conducted at beyond limits of the specifications. It falls under the class of black box testing.

5.6 PERFORMANCE TESTING:

Performance testing is the testing to assess the speed and effectiveness of the system and to make sure it is generating results within a specified time as in the performance requirements. It falls under the class of black box testing.

5.7 USABILITY TESTING:

Usability testing is performed to the perspective of the client, to evaluate how GUI is user-friendly? How easily can the client learn? After learning how to use, how proficiently can the client perform? How pleasing is it to use its design? This falls under the class of black box testing

5.8 REGRESSION TESTING:

Regression testing is the testing after the modification of a system, component, or group of related units to ensure that the modification is working correctly and not damaging or imposing other modules to produce unexpected results. It falls under the class of black box testing.

5.9 BLACKBOX TESTING:

Black box testing is a testing technique that ignores the internal mechanisms of the system and focuses on the input and the output generated against any input and execution of the system. It is also called as functional testing.

5.10 WHITEBOX TESTING:

White box testing is a testing technique that takes into account the internal mechanisms of a system. It is also called as structural testing and glass box testing. Black box testing is often used for validation and white box testing is often used for verification.

CHAPTER 6

CONCLUSION

6.1 CONCLUSION

Organization based customization on the assessment of employee is available in the proposed system, which is not available in the existing system. Generation of reports on the performance of the each and every individual employee over the whole working period on the organization is available in the proposed system.

6.2 FUTURE ENHANCEMENTS

Presently in the proposed system, it uses Apache Derby database for the data storing purpose. As spring supports Big Data, further enhancement can be done on any no SQL database without affecting the performance of the proposed system and report generation can be done with the same efficiency even though there is a big pool of data.

APPENDIX

SCREENSHOTS

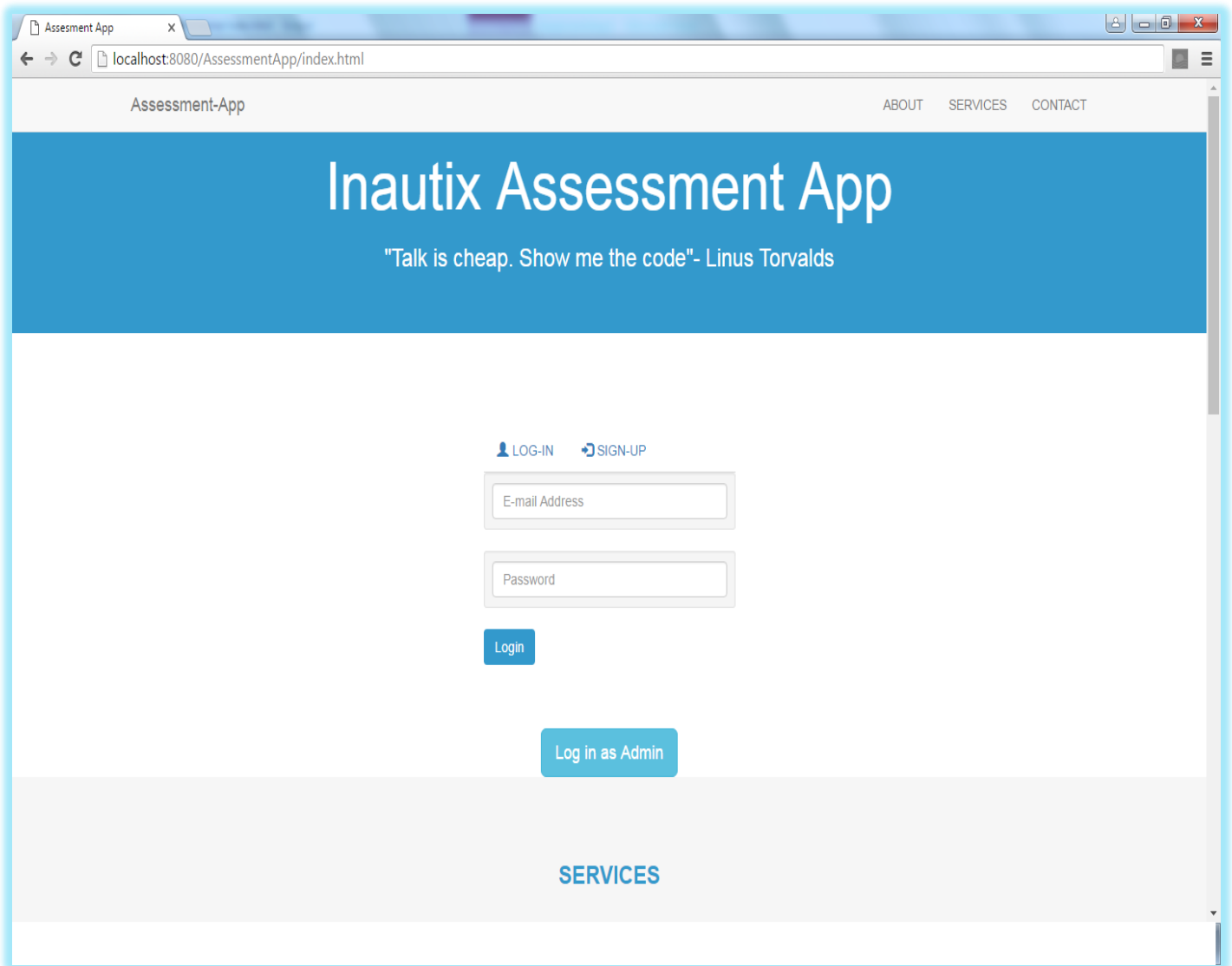


FIGURE: 6.1 INDEX PAGE

Admin Signup Screen for Authentication:

- The admin signup with their name, emailid, employeeid and password.
- The application identifies the admin using their employeeid, other than admin non can't signup.
- The admin details are stored in **Apache Derby** database.

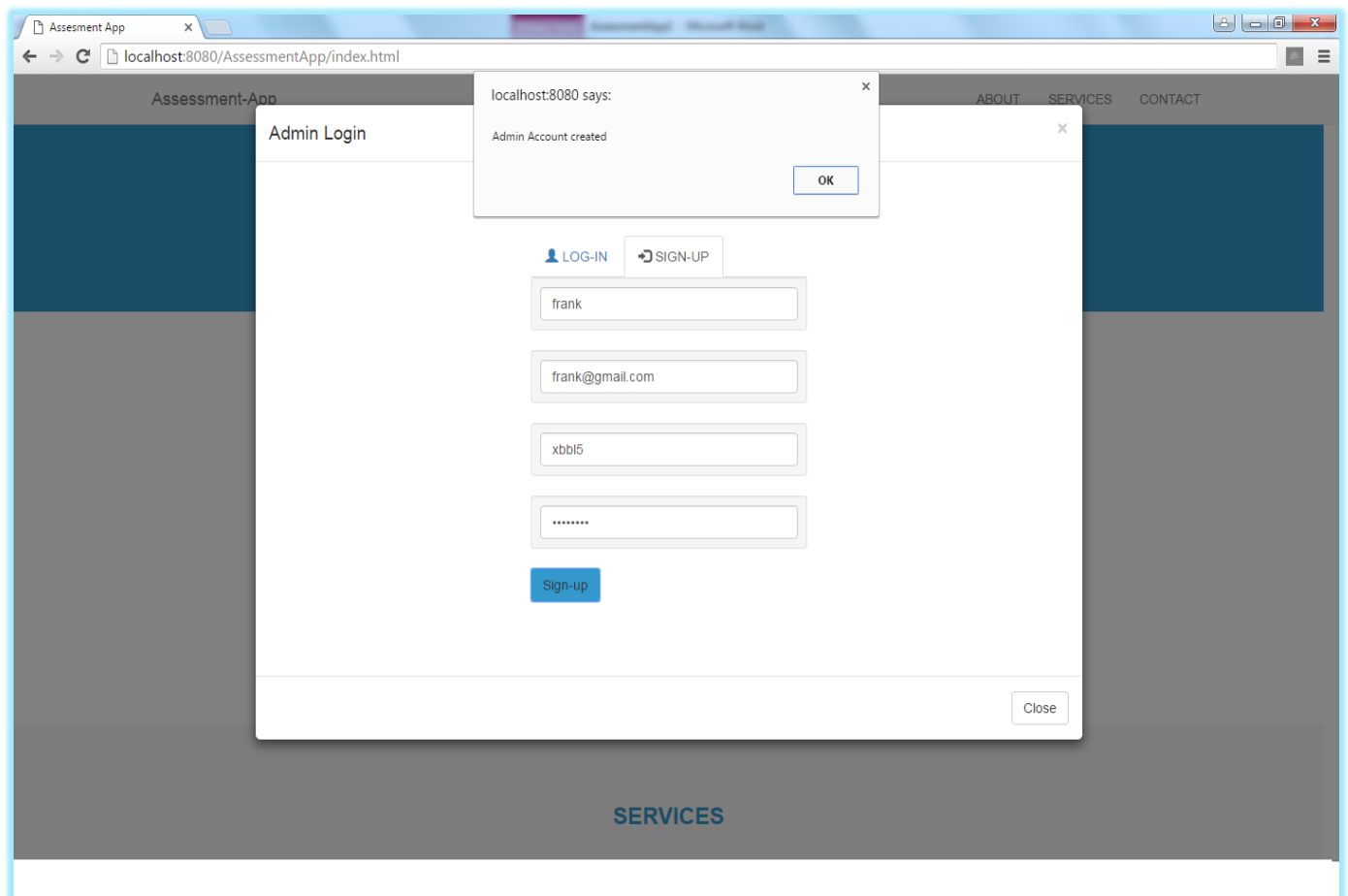


FIGURE: 6.2 ADMIN SIGNUP

Admin Login Screen for Authentication:

- The admin login with their secure emailid and password.
- Using **AngularJS** by sending **AJAX** request and response to valid the admin emailid and password.

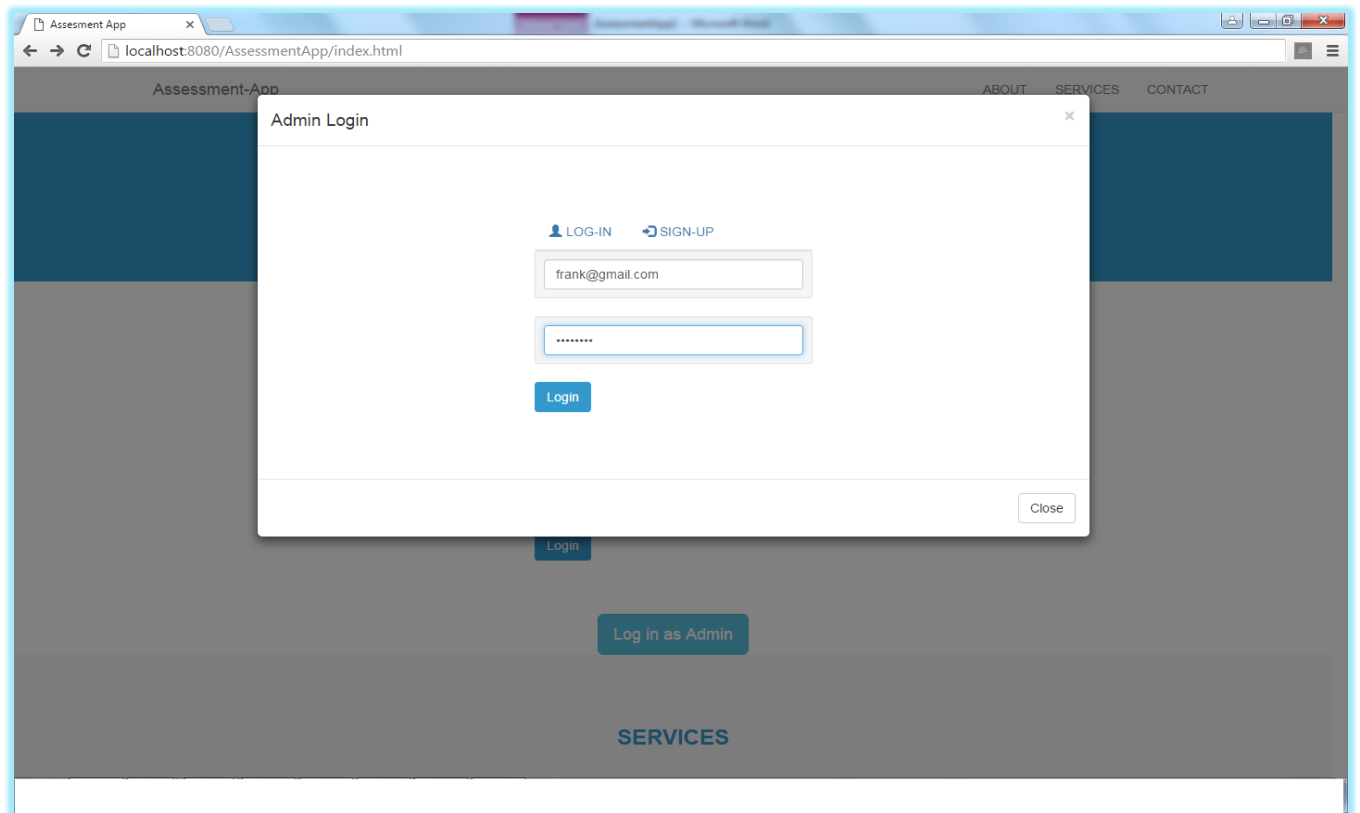


FIGURE: 6.3 ADMIN LOGIN

- Using **AngularJS** by sending **AJAX** request and response to valid the admin emailid and password.
- The application valid the admin emailid and password , if it is invalid the application will show an **Alert message**.

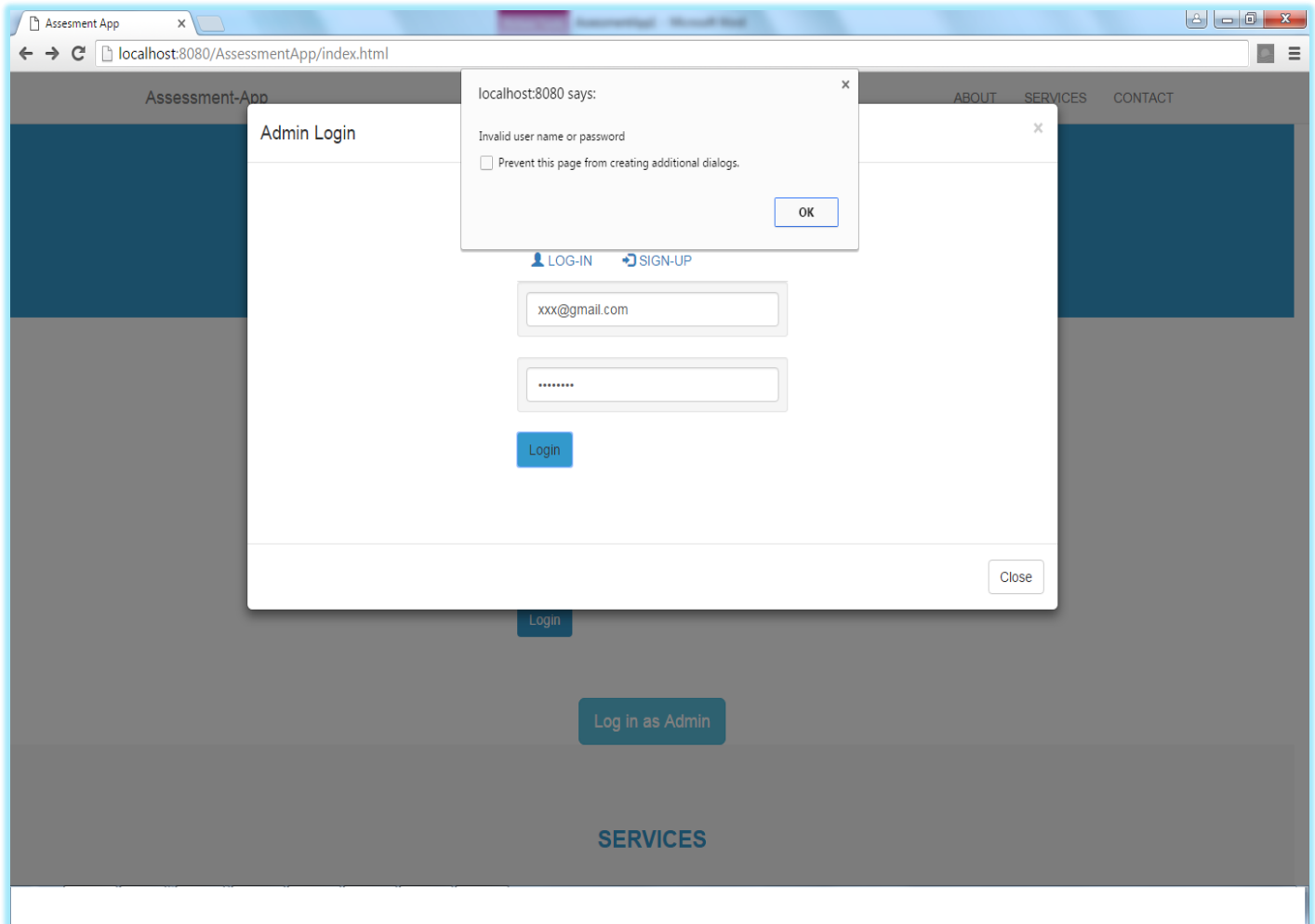


FIGURE: 6.4 INVALID LOGIN

Admin Page:

- This is a page used to create a test and view already existing test details.
- On clicking **MY TEST**, it will show the already existing test created by the respective admin.
- On clicking **CREATE TEST** to create new test like on multiple choice questions(MCQs) and Programming task.

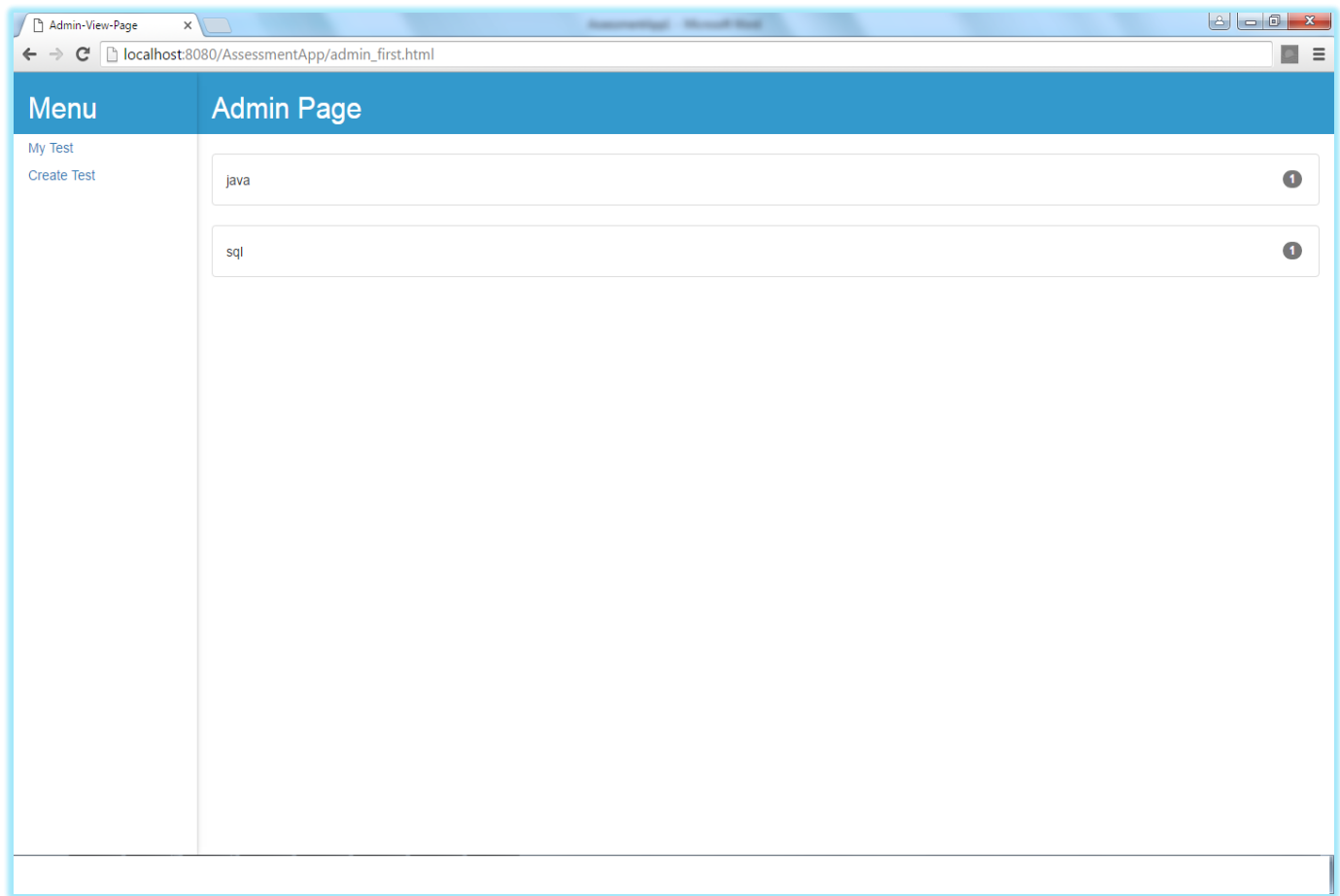


FIGURE: 6.5 ADMIN PAGE

MY TEST:

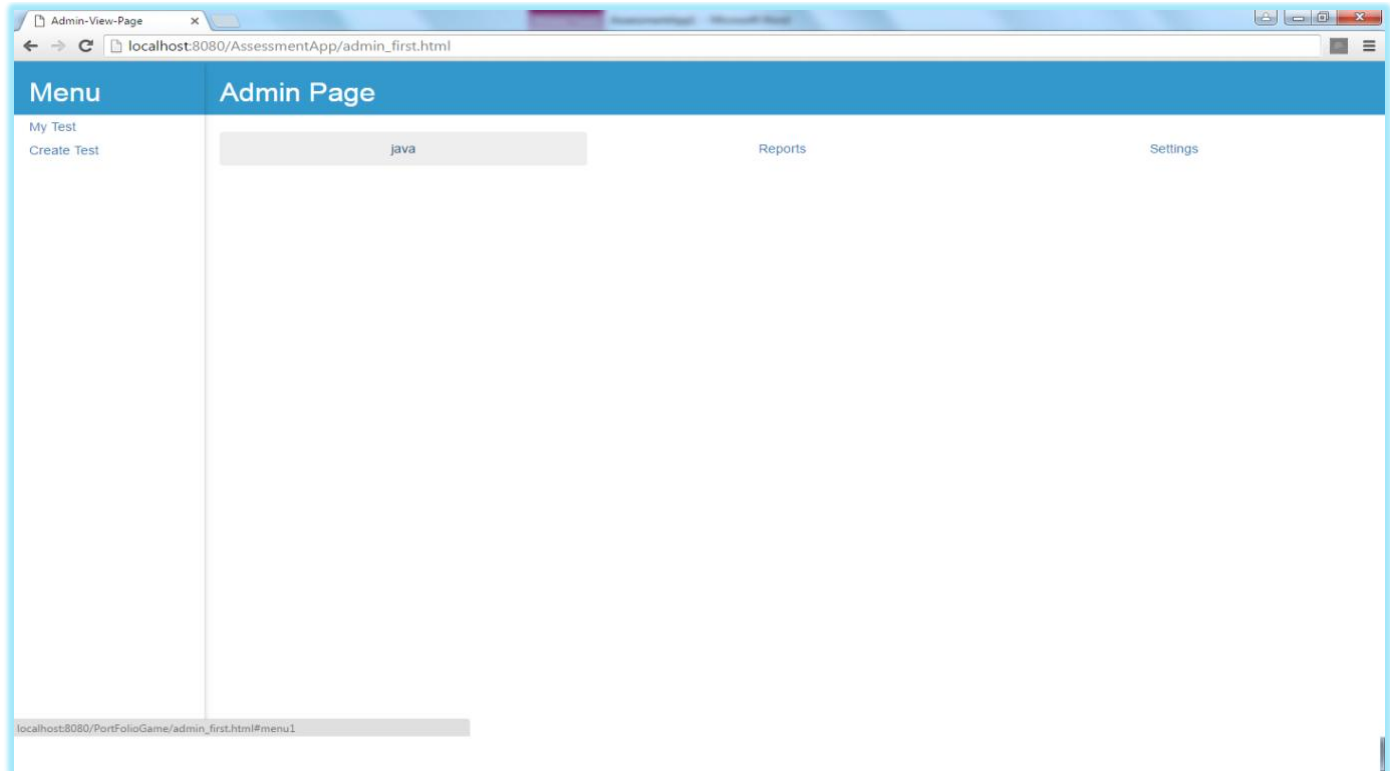
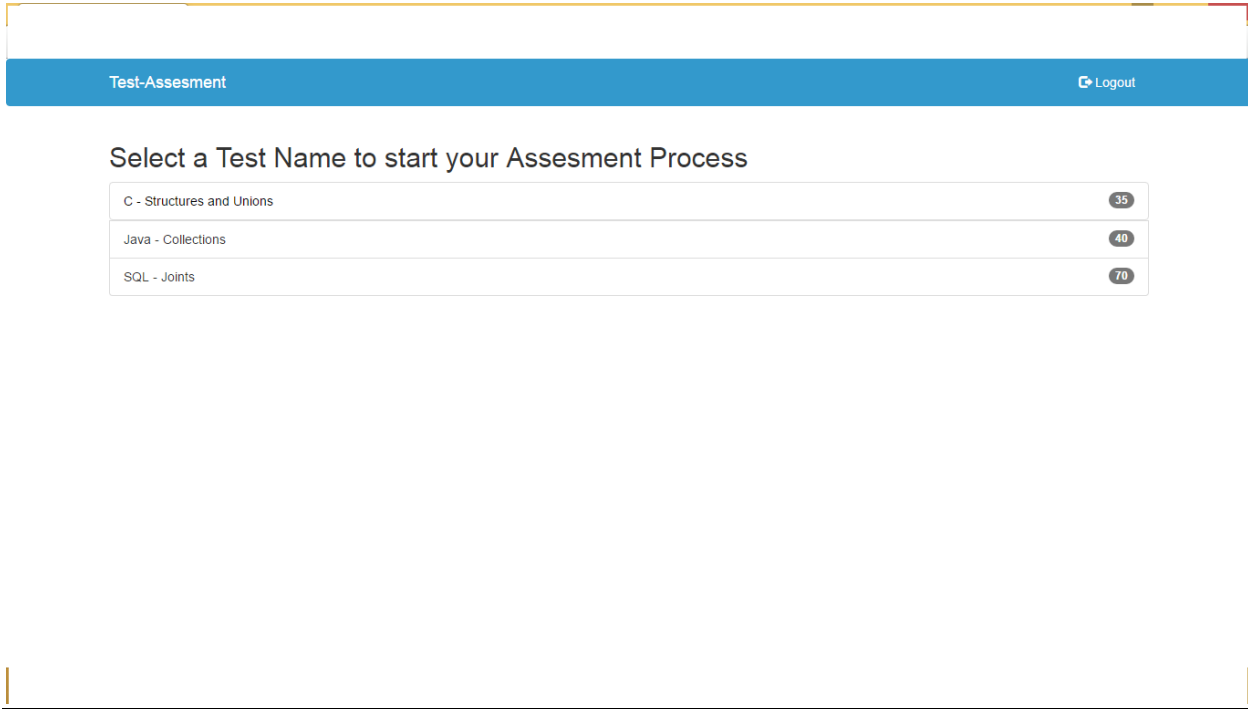


FIGURE: 6.6 MY TEST

USER TAKE TEST:

- The user can click the domain name to select the test.
- After completing the test, the result will be display to the user profile.



The screenshot shows a web interface for a test assessment. At the top, there is a blue header bar with the text "Test-Assesment" on the left and a "Logout" button on the right. Below the header, the main content area has the heading "Select a Test Name to start your Assesment Process". Under this heading is a table with three rows, each representing a different test category. The first row is "C - Structures and Unions" with a score of 35. The second row is "Java - Collections" with a score of 40. The third row is "SQL - Joins" with a score of 70. Each row is a clickable button.

Select a Test Name to start your Assesment Process	
C - Structures and Unions	35
Java - Collections	40
SQL - Joins	70

FIGURE: 6.7 USER SELECT TEST

USER PROGRAM TEST:

- In this programming section, the user get question based on the domain names they selected.
- The question will contain explanation along with two sample inputs and outputs .
- This makes the user solve the program efficiently and quickly.

Programming Section

Logout

1. Write a program to find the input two strings are Anagrams or not ?

Explanation :

For Example: Two strings AA and BB are called anagrams if they consist same characters, but may be in different orders.
So the list of anagrams of CAT are "CAT", "ACT" , "TAC", "TCA", "ATC" and "CTA". Given two strings, print "Anagrams" if they are anagrams, print "Not Anagrams" if they are not. The strings may consist at most 50 english characters, the comparison should NOT be case sensitive. This exercise will verify that you are able to sort the characters of a string, or compare frequencies of characters.

Sample Input 1

anagram
margana

Sample Output 1:

Anagrams

Sample Input 2

anagramm
marganaa

Sample Output 2:

Not Anagrams

Write your code here ..

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     Boolean retValue = false;
6     if(A != null && B != null) {
7         char [] arrayA = A.toLowerCase().toCharArray();
8         char [] arrayB = B.toLowerCase().toCharArray();
9         Arrays.sort(arrayA);
10        Arrays.sort(arrayB);
```

FIGURE: 6.8 USER TEST

anagram
margana

Sample Input 2
anagram
marganaa

Write your code here ..

```
1 import java.io.*;
2 import java.util.*;
3
4 public class Solution {
5     Boolean retValue = false;
6     if(A != null && B != null) {
7         char [] arrayA = A.toLowerCase().toCharArray();
8         char [] arrayB = B.toLowerCase().toCharArray();
9         Arrays.sort(arrayA);
10        Arrays.sort(arrayB);
11        retValue = Arrays.equals(arrayA, arrayB);
12    }
13 }
```

☐ Run against user input

RunSubmit

Sample Output 2:
Not Anagrams

Your Output :
Anagrams

Expected Output
Anagrams

Correct Output

FIGURE: 6.9 TEST RESULT

Sample Input 1

anagram
margana

Sample Input 2

anagramm
marganaa

Sample Output 1:

Anagrams

Sample Output 2:

Not Anagrams

Write your code here ..

2 import java.util.*;
3
4 public class Solution {
5 Boolean retValue = false;
6 if(A != null && B != null) {
7 char [] arrayA = A.toLowerCase().toCharArray();
8 char [] arrayB = B.toLowerCase().toCharArray();
9 Arrays.sort(arrayA);
10 Arrays.sort(arrayB);
11 retValue = Arrays.equals(arrayA, arrayB)
12 }
13 return retValue;
14 }

C

☐ Run against user input

Run

Submit

Your Output :

Error Occurred :

Error at line 11 : missing ; statement

Expected Output

Anagrams

FIGURE: 6.10 TEST RESULT

49

REFERENCES:

BIBLIOGRAPHY

1. AGIC. 2009. Subbasin (8-digit HUC) Boundaries – NHD. Retrieved from <http://agic.az.gov/portal/dataList.do?sort=theme&dataset=415>
2. Beard, G. R., W. A. Scott, and J. K. Adamson. 1999. The value of consistent methodology in long-term environmental monitoring. *Environmental Monitoring and Assessment* 54: 239 Bonham, C. D. 1989.
3. American insects: A Handbook of the Insects of America North of Mexico. Van Nostrand Reinhold, New York.

Wibliography

- [1] http://www.tutorialspoint.com/spring/spring_web_mvc_framework.html
- [2] http://www.tutorialspoint.com/maven/maven_eclipse_ide.html
- [3] <http://www.codejava.net/java-se/jdbc/connect-to-apache-derby-java-db-via-jdbc>
- [4] <http://www.eclipse.org/articles/article.php?file=Article-EclipseDbWebapps/index.html>
- [5] https://db.apache.org/derby/integrate/plugin_help/start_toc.html
- [6] http://www.w3schools.com/angular/angular_controllers.asp
- [7] http://www.w3schools.com/bootstrap/bootstrap_panels.asp