# Online Insurance Portal

**Synopsis**

This project is aimed to develop “**Online Insurance Portal”.** It is web-based application, which is developed mainly for the agents of the insurance company to take the policy for his clients electronically. By the request from his clients, he processes and maintains the policy details through the system.

**Online Insurance Portal** System is an integrated insurance system which links up all the channels within the insurance industry. It is a revolutionized insurance solution that can facilitate online processing and services to the insurance partners, agents and customers through the Internet.

Insurance can be broadly divided into two types.

1. Motor

2. Non-motor

**Modules:**

* 1. **Motor Insurance**

Motor Insurance is insurance purchased for cars, trucks, and other road vehicles. Its primary use is to provide protection against physical damage resulting from traffic collisions and against liability that could also arise there from.

The policy covers the following types of vehicle:

* **Private Vehicles**

Private Car, Private Van & Four Wheel Drive

(Private Use)

* **Goods Carrying Vehicles**

Commercial Goods Carrying Vehicles also Petroleum, Diesel and Bitumen Tankers, LPG Long Trailer etc.

* **Other Types of Vehicle**

Taxis, Buses, Motorcycles, Tractor, Mobile Crane etc.

**1. Study on** **Proposed System**

The Online Insurance Portal provides a numerous policy for the customers, so that the customer can view all the policy, complete details and benefits about the policy. The online assistance will help the customer to choose the policy he/she wants and the premiums calculator are been use to calculate the premium amount of their own policies.

The customer can buy their policies through online, and the payments can be made through credit/debit cards. The premium payment and policy renewal date can be intimated to the customers through emails. Policy premium payments and renewal of the policies can also be done through online.

**2.** **Defining the Problem**

1. Records are maintained manually.
2. The customer should visit the insurance office to know the policy details.
3. The premium calculations are done manually.
4. The policies are purchased directly visiting the insurance office.
5. The premium period of the policy will be intimated to the customer through post.
6. Payment is done in person.
7. New policy arrivals are not intimated to the customer in time.

**3. Developing solution strategies**

1. All the manual process that is done is automated and provided an online service where the customer can buy any policy at anytime from anywhere.
2. Providing a user-friendly interface to the agent.
3. Payments can be done through online by their credit cards and the receipts can be printed out immediately after the payment is done.
4. Premium calculations are done online by the customers to know their premium value.
5. Premium periods and the policy renewal dates are intimated to the customers through email.

**4. System Specification**

Every insurance company needs an agency suite to make their applications in efficient way. To achieve this, the project should develop in a flexible manner. Our target application is to build an online insurance portal as web based application through Java, J2EE and with the oracle 10g database. Our proposed system, the Online insurance portal suite helps to provide the efficient access for the agents / clients under the single roof.

**5. Process Description**

**Dataflow Diagram:-**

**Context-0**



Doc New Business Printed document

**Context-I**



Login\_DB

Invalid Login

Verify New User

Valid Login

**Context-II**





Store/retrieve









Store/retrieve







Store/retrieve







Store/retrieve







Store/retrieve 





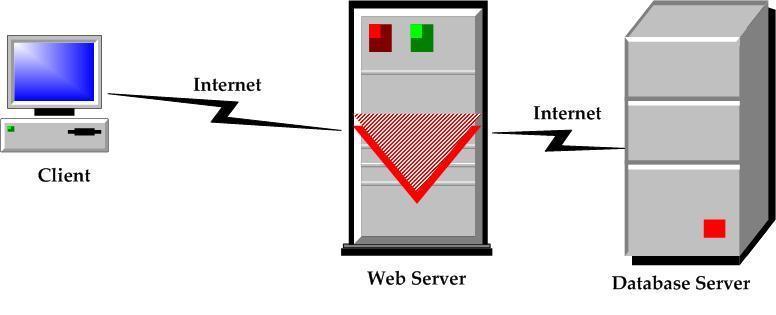


Store/retrieve 

**s**

Receipt

1. **Application Specification**



The Insurance Management system uses 3-tier application is a program, which is organized into three major disjunctive tiers. These tiers are

* Presentation Tier (Front end)
* Business Tier (Logical)
* Data Tier (Backend)

**7. Network Specification**

The development and availability of Internet technology has resulted in an upsurge of Intranets within these Organizations. It is now relatively easy for someone with an understanding of the technology and HTML to create web pages, and implement a server to host them. As their experience increases they may also become able to produce graphics to enhance the site, and more complicated and functional navigation.

**8. Hardware Specification**

**Server**

Processor : Intel Xeon 7400

Hard Disk Drive : 500 GB

RAM : 4 GB

Monitor : 21’’ TFT Monitor

Keyboard : 108 keys Multimedia keyboard

Mouse : Optical

**Client**

Processor : Intel Pentium III

RAM : 128 MB

Hard Disk : 20GB

Monitor : 15” Color Monitor

Keyboard : 104 keys

Mouse : Optical

**Development Environment**

Processor : Amd Athlon 2.02GHz

RAM : 1 GB

Monitor : 15’’Color Monitor

Hard Disk : 40 GB

Keyboard : 104 keys

Mouse : Optical

1. **Software Environment**

The application should be able to be used with the following software:

**Server**

Operating System : Windows server 2008

Database Server : Oracle 10g

Web Browser : Internet Explorer 6.0

Web Server : Apache Tomcat

Reader : Adobe Acrobat Reader 6.0

**Client**

Operating System : Windows xp

Web Browser : Internet Explorer 6.0

Reader : Adobe Acrobat Reader 6.0

**Development Environment**

Operating System : Windows xp

Database Server : Oracle 10g

Web Browser : Internet Explorer 6.0

Web Server : Apache Tomcat

Reader : Adobe Acrobat Reader 6.0

IDE : Jboss Eclipse 2.0.0

Frame Work : Apache Struts

**HTML:**

HTML, an initialize of Hyper Text Markup Language, is the predominant markup language for Web pages. It provides a means to describe the structure of text-based information in a document by denoting certain text as links, headings, paragraphs, lists, and so on and to supplement that text with interactive forms, embedded images, and other objects. HTML provides the user with a consistent interface and provides developers a highly effective medium for presenting Information.HTML is the base of a webpage. HTML presence is dominating the web – it is the most widely accepted language for web design.

**JAVA SCRIPT:**

JavaScript is a Scripting language that can be included on web pages to make them more interactive. It is used to check or modify the contents of forms, change images, open new windows and write dynamic page content. Using the features available in JavaScript, the designer can decide to have dynamically placed text at run time.

JavaScript also has the feature of validating data submitted at the client level. This helps in saving the processing time of the server because JavaScript initially creates the validation on the client side. JavaScript is an excellent solution to implement when validating input forms on the client side. Client side JavaScript is embedded inside HTML this embedded JavaScript is used along with DOM (Document Object Model) for control over the browser by means of objects.

**JSP:**

Java server Pages (JSP) technology enables Web developers and designed to rapidly develop and easily maintain information-rich, dynamic web pages that leverage existing business systems. As part of the java technology family, JSP technology enables rapid development of Web-based applications that are platform independent. JSP technology separates the user interface from content generation, enabling designers to change the overall page layout without altering the underlying dynamic content.

The main features of JSP technology are

* A language for developing JSP pages, which are text-based documents that describe how to process a request and construct a response

* Constructs for accessing server-side objects

* Mechanisms for defining extensions to the JSP language

JSP pages can be moved easily across platforms, and across web servers, without any changes. JSP page is automatically recompiled and reloaded into the web server by the JSP engine. JSP uses simplified scripting language based syntax for embedding HTML into JSP.

**EJB:**

Enterprise JavaBeans (EJB) technology is the server-side component architecture for Java Platform, Enterprise Edition (Java EE). EJB technology enables rapid and simplified development of distributed, transactional, secure and portable applications based on Java technology. Enterprise JavaBeans contain the application’s business logic and live business data.

The EJB specification intends to provide a standard way to implement the back-end business code found in enterprise applications. Such code was frequently found to reproduce the same types of problems, and it was found that solutions to these problems are often repeatedly re-implemented by programmers. Enterprise JavaBeans were intended to handle such common concerns as persistence, transactional integrity, and security in a standard way, leaving programmers free to concentrate on the particular problem at hand.

**AJAX:**

AJAX stands for Asynchronous JavaScript and XML. AJAX is a type of programming made popular in 2005 by Google (with Google Suggest). AJAX is not a new programming language, but a new way to use existing standards. With AJAX you can create better, faster, and more user-friendly web applications. AJAX is based on JavaScript and HTTP requests.

With AJAX, our JavaScript can communicate directly with the server, using the JavaScript XMLHttpRequest object. With this object, our JavaScript can trade data with a web server, without reloading the page.

**Web Standards and Technologies Used In AJAX**

* JavaScript
* CSS
* XMLHttpRequest

**Advantages**

* Emerges as a viable RIA technology (Rich Internet Application)
* Several toolkits and frameworks are emerging
* AJAX-enabled JSF Component libraries
* Standardization of XMLHttpRequest
* Better browser support
* Better and Standardized Framework support
* More best practice guidelines in the programming model

**ORACLE 10g:**

Oracle is one of the most powerful database management system based on Relational Database Model, provided by Oracle Corporation along with fully integrated database application development and administration tools. It uses Structured Query Language (SQL) for database access and its own proprietary procedural language PL/SQL for application development along with Java programming support.

**Oracle 10g New Features**

* The SYSAUX Table space
* Automated Storage Management
* Statistics Collection
* New Table-Monitoring Behaviors
* Flushing the Buffer Cache
* Scheduler Changes
* User-Configurable Default Table spaces
* Table space Groups and Multiple Default Temporary Table spaces
* Dropping Databases
* Larger LOBs
* Sorted Hash Clusters

**APACHE STRUTS:**

Apache Struts is a free open-source framework for creating Java web applications.

Web applications differ from conventional websites in that web applications can create a dynamic response. Many websites deliver only static pages. A web application can interact with databases and business logic engines to customize a response.

Web applications based on Java Server Pages sometimes commingle database code, page design code, and control flow code. In practice, we find that unless these concerns are separated, larger applications become difficult to maintain.

One way to separate concerns in a software application is to use Model-View-Controller (MVC) architecture. The *Model* represents the business or database code, the *View* represents the page design code, and the *Controller* represents the navigational code. The Struts framework is designed to help developers create web applications that utilize MVC architecture.

The framework provides three key components:

* A "request" handler provided by the application developer that is mapped to a standard URI.
* A "response" handler that transfers control to another resource which completes the response.
* A tag library that helps developers create interactive form-based applications with server pages.

The framework's architecture and tags are buzzword compliant. Struts works well with conventional REST applications and with nouveau technologies like SOAP and AJAX.

## MVC (Model View Controller)

## The term "MVC" originated with the Smalltalk Model-View-Controller framework. Under MVC, an application is seen as having three distinct parts. The problem domain is represented by the Model. The output to the user is represented by the View. And, the input from the user is represented by Controller.

**MODEL:**

The *Model* portion of an MVC-based system can be often be divided into two major subsystems -- the **internal state** of the system and the **actions** that can be taken to change sthat state.

In grammatical terms, we might think about state information as **nouns** (things) and actions as **verbs** (changes to the state of those things).

Many applications represent the internal state of the system as a set of one or more JavaBeans. The bean properties represent the details of the system' state. Depending on your application's complexity, these beans may be self contained (and know how to persist their own state), or they may be facades that know how to retrieve the system's state from another component. This component may be a database, a search engine, an Entity Enterprise Java Bean, a LDAP server, or something else entirely.

**VIEW:**

The View portion of a Struts-based application is most often constructed using Java Server Pages (JSP) technology. JSP pages can contain static HTML (or XML) text called "template text", plus the ability to insert dynamic content based on the interpretation (at page request time) of special action tags. The JSP environment includes a set of standard action tags, such as <jsp: use Bean> whose purpose is described in the [Java Server Pages Specification.](http://java.sun.com/products/jsp/download.html) In addition to the built-in actions, there is a standard facility to define your own tags, which are organized into "custom tag libraries."

**CONTROLLER:**

Struts provide the Controller portion of the application. The Controller is focused on receiving requests from the client (typically a user running a web browser), deciding what business logic function is to be performed, and then delegating responsibility for producing the next phase of the user interface to an appropriate View component. The primary component of the Controller in the framework is a servlet of class ActionServlet. This servlet is configured by defining a set of ActionMappings. An ActionMapping defines a path that is matched against the request URI of the incoming request and usually specifies the fully qualified class name of an Action class. All Actions are subclassed from [org.apache.struts.action.Action].

**APACHE TOMCAT:**

**Apache Tomcat** (or **Jakarta Tomcat** or simply **Tomcat**) is an [open source](http://en.wikipedia.org/wiki/Open_source) [servlet container](http://en.wikipedia.org/wiki/Java_Servlet#Servlet_containers) developed by the [Apache Software Foundation](http://en.wikipedia.org/wiki/Apache_Software_Foundation) (ASF). Tomcat implements the [Java Servlet](http://en.wikipedia.org/wiki/Java_Servlet) and the [JavaServer Pages](http://en.wikipedia.org/wiki/JavaServer_Pages) (JSP) specifications from [Sun Microsystems](http://en.wikipedia.org/wiki/Sun_Microsystems), and provides a "pure [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29)" [HTTP](http://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol) [web server](http://en.wikipedia.org/wiki/Web_server) environment for [Java](http://en.wikipedia.org/wiki/Java_%28programming_language%29) code to run.

Tomcat should not be confused with the [Apache web server](http://en.wikipedia.org/wiki/Apache_HTTP_Server), which is a [C](http://en.wikipedia.org/wiki/C_%28programming_language%29) implementation of an HTTP web server; these two web servers are not bundled together. Apache Tomcat includes tools for configuration and management, but can also be configured by editing [XML](http://en.wikipedia.org/wiki/XML) configuration files.

**10. Cost Estimation and Scheduling**

Software cost is related to many variables such as Human, Technical, Environment and Effort applied to develop it. The estimates of cost depend, on our ability to estimate and evaluate several factors, given below.

* Number of user inputs
* Number of user output
* Number of files
* Number of inquiries
* Experience and ability of the project personnel
* The quality of software development environment
* The degree to which software components can be reused

The productivity of a CMM Level 4 company is 2.0 and we assumed the productivity of our team as 2.0

Based on the size and Productivity the man days and man hours are calculated as shown below,

Man Days = 180.0 FP/2.0

= 90days

Man Hours = 90 \* 8 = 720 hr

**Man Power Estimation**

|  |  |
| --- | --- |
| **Phase** | **Time Taken** |
| User Requirement Specification | 50 |
| Project Planning | 45 |
| System Requirement Specification | 50 |
| Architectural Design Document | 70 |
| Detailed Design Document | 95 |
| Coding | 200 |
| Unit Testing | 40 |
| Integration Testing | 40 |
| System Testing | 40 |
| User Acceptance Testing | 40 |
| Installation | 50 |
| Total | 720 |

**11. Final Outline of the Proposed System**

As a result of the detailed study, various objectives were formulated. They are given below,

* Performance is increased for the system with the latest technology
* Provide a one-stop place where all agents of the insurance company can perform all transactions
* Provide a convenient set of communications services, which are web-based
* Support a single log-on to obtain authentication and authorization to all information resources and applications
* New features are added to the existing one in order to give endless service
* 24 hours worldwide protection
* High security and reliability
* Performance increased by adding the search option for all existing details