YCompany-Electronic Insurance Policy

DAR Document



Nagarro Software Pvt. Ltd.

Isha Gupta

|  |  |  |  |
| --- | --- | --- | --- |
| **Revision History** | | | |
| Version | Date | Author/Contributor | Comments |
| V1.0 | 6th May 2017 | Isha Gupta |  |
| V1.1 | 11th May 2017 | Isha Gupta |  |
| V1.2 | 19th May 2017 | Isha Gupta |  |
|  |  |  |  |
|  |  |  |  |

Contents

[1 Introduction 4](#_Toc482346413)

[1.1 Objective and scope of document 4](#_Toc482346414)

[2 Requirements at a Glance 5](#_Toc482346415)

[3 Available tools 5](#_Toc482346416)

[3.1 Data providers 5](#_Toc482346417)

[3.1.1 SQL Server 5](#_Toc482346418)

[3.1.2 PostgreSQL 5](#_Toc482346419)

[3.2 Object-Relational Mappers (ORMs) 6](#_Toc482346420)

[3.2.1 Entity Framework 6 6](#_Toc482346421)

[3.2.2 Nhibernate 4 6](#_Toc482346422)

[3.3 Alert & Notification Framework 6](#_Toc482346423)

[3.3.1 A Windows Service & Using SignalR & SQL Dependency for database notifications 6](#_Toc482346424)

[3.3.2 Using SQL Server Job Scheduler for automated email notifications 6](#_Toc482346425)

[3.4 Front-end technologies 6](#_Toc482346426)

[3.4.1 Angular 1.5 6](#_Toc482346427)

[3.4.2 Angular 2.0 6](#_Toc482346428)

[4 Comparison Analysis 7](#_Toc482346429)

[4.1 ORMs 7](#_Toc482346430)

[4.2 Frontend technologies 7](#_Toc482346431)

[5 Recommendation 8](#_Toc482346432)

[6 Assumptions 9](#_Toc482346433)

[7 Risks 10](#_Toc482346434)

[8 Appendix 11](#_Toc482346435)

[8.1 References 11](#_Toc482346436)

# Introduction

This document has detailed evaluation of various tools and technologies which can be used in designing the electronic Insurance Policy. The document also has a detailed explanation of available features and pricing model for the various technologies available.

## Objective and scope of document

This document describes in detail the various tool & technology options available for development of ePolicy Portal. It will begin with presenting all the options available in different areas of software development, bringing out a detailed analysis & comparison between different tools & framrworks around various factors. In the end recommendations are presented.

# Requirements at a Glance

The application will consist of the following features:

* Web application – this web based application will be available to authorized users such as administrator, underwriter, customer, auditor, billing administrator view the live feed, reporting etc.
  + Agent Portal
  + Customer Portal
  + Internal Portal
* Alert and notification system – this sub-system will be analyzing the time to resolve for any incident.
* Reporting
  + Reconciliation report should be generated for premium collection for policies.
  + Reports for Auditor
  + Reports for Agents
  + Reports for Internal Users

# Available tools

## Data providers

Available data providers for such an application are SQL Server, Oracle, PostgreSQL, MySQL, SQLite, or DB2.

### SQL Server

#### Features

1. It is supported by Microsoft.
2. Integrated with Microsoft data access platform like Entity Framework.
3. Visual studio provides build-in tools for working with SQL Server.
4. Database deployment features are also available within Visual studio to facilitate deploying SQL server databases for web projects.
5. It is a closed source for corporate/enterprise applications.
6. It offers a high degree of control over transactions and procedures
7. Offers advanced security to in-database analytics.

However, SQL Server Developer edition will be used for development environment and the full edition for the production database.

### PostgreSQL

#### Features

* Open-source
* Adheres well to current SQL standards, and easier to learn as a result
* Large footprint makes it ill-suited for read-heavy operations
* Advanced business/location analytics features
* Rich variety of data and character types
* Fully ACID-compliant
* Designed for reliability and data-integrity; developer-focused
* Full-text search, support for powerful server-side procedural languages
* Full support for advanced SQL features such as table expressions and window functions
* Can efficiently join large numbers of tables
* Replication is poorly-implemented
* Not well-suited for low-concurrency projects
* Works best with Java.

## Object-Relational Mappers (ORMs)

Asp.Net provides many ways for storing, retrieving and displaying data. Which one is appropriate is not always obvious. ORMs (Object Relational Mapper) make data access so straight forward that there is a possibility of having data access logic/predicates scattered all across the application.

### Entity Framework 6

#### Features

It is Microsoft’s primary data access platform in the .NET Framework. It is an open source object-relational mapping (ORM) framework for ADO.NET. Here are some features introduced in Entity Framework 6.0 RC1. Smaller API i.e. less knobs and levers.

* Easier to learn
* Customizing Code First Conventions.
* Logging of database commands.
* Stored Procedure Mapping.
* Asynchronous Queries and Save support.
* Code based configuration support.
* EF Power Tools (not new to EF6).

### Nhibernate 4

#### Features

* Allows greater specification of loading strategies.
* Configurable eager loading between parent and child collection objects.
* Larger API hence comes a steeper learning curve

## Front-end technologies

### Angular Js

### It is a javascript library. It extends HTML with ng-directives.

#### Features:

* No mobile support

### Angular 2

### AngularJs rewritten is angular2. Here are some of its features:

#### Features:

* Mobile oriented
* Better Performance
* Partial loading
* Improved dependency injection model
* Based on Typescript

## Continuous Integration

Continuous integration software tools can be used to automate the testing and build a document trail.

### Jenkins

It open source automation server written in Java. Forked from Hudson, by Sun Microsystems. It is free and open source. It can work with many common .NET version control systems, can run MSBuild scripts, and has a very active plug-in development community. It is widely used and well documented. It can be integrated with a wide variety of plugins to add different functionality. Sonarqube can be integrated with Jenkins build for code coverage analysis & inspecting code vulnerabilities in the build.

### Bamboo

It is continuous integration server by Atlassian. It possesses all the features provided by Jenkins. It is licensed tool and purchased with other Atlassian products.

### Team City

It is a continuous integration server based on Java. It possesses all the features provided by Jenkins, in addition, it has many .net tools integrated such code coverage analysis & several testing frameworks. It is licensed.

# Comparison Analysis

## Data Providers

|  |  |  |
| --- | --- | --- |
| Feature | SQL Server | Postgre |
| Open Source | **** | **** |
| Relational DBMS | **** | **** |
| In-memory capabilities | **** | **** |
| XML support | **** | **** |
| Entity framework support | **** | **** |

## ORMs

Comparison table

|  |  |  |
| --- | --- | --- |
| Feature | Entity Framework 6 | Nhibernate 4 |
| Open Source | **** | **** |
| Custom Eager Loading | **** | **** |
| Linq Support | **** | **** |
| Code first Mapping | **** | **** |
| Wider database provider support | **** | **** |
| Seeding of Initial database | **** | **** |
| Support of Asynchoronours Operations | **** | **** |
| Connection Resiliency | **** | **** |

## Frontend Technologies

|  |  |  |
| --- | --- | --- |
| Feature | AngularJS | Angular 2 |
| Two way binding | **** | **** |
| Mobile oriented | **** | **** |
| Component based | **** | **** |
| Better Performance | **** | **** |
| Partial Loading | **** | **** |
| ES6 Support | **** | **** |

## Continuous Integration

|  |  |  |  |
| --- | --- | --- | --- |
| Feature | Jenkins | Bamboo | TeamCity |
| Open source | **** | **** | **** |
| Plugin Support | **** | ** (but expensive)** | **** |
| Automatic Platform | **** | **** | **** |
| JIRA Support | **** | **** | **** |
| Ease of usage | **** | **** | **** |
| Repository Support | **** | **** | **** |
| Wide Online Support | **** | **** | **** |
| Handle multiple languages | **** | **** |  |
| Documentation | **** | **** | **** |
| Parallel build in multiple branches | **** | **** | **** |

# Recommendation

Since we are dealing with Microsoft stack here in the application, we are using SQL Server for this application. Another reason to propose it would be that system would be dealing with critical user information and it would easily scale up to critical business intelligence requirements. Hence, Entity Framework 6 becomes the obvious choice for ORMs over Nhibernate. For Frontend this document recommends using Angular 2. For Continuous Integration, Jenkins is recommended.

# Assumptions

1. Resourcing of these tools and the engineers to work would be available.
2. Deployment environment would support the installation & configuration of continous integration servers.

# Risks

Resourcing for the tool, framework selected is important and if not done then things need to be revisited to accommodate the changes.

# Appendix

## References

1. http://techbrij.com/database-change-notifications-asp-net-signalr-sqldependency
2. https://www.aspsnippets.com/Articles/Automated-Email-Notifications-using-SQL-Server-Job-Schedular.aspx
3. <https://msdn.microsoft.com/en-us/library/ms178359(v=vs.110).aspx#orm>
4. <https://www.devbridge.com/articles/entity-framework-6-vs-nhibernate-4/>
5. https://xhalent.wordpress.com/2011/02/01/nhibernate-vs-entityframework-experience-from-the-real-world/#comment-158
6. <https://www.libertycenterone.com/blog/oracle-vs-mysql-vs-sql-server-vs-postgresql-which-dbms-is-the-best-choice-for-you/>
7. https://codeship.com/continuous-integration-essentials