

|  |
| --- |
|  |
| Promotion Case Study |
|  |

Assessment for Team Lead

**ePolicy**

Electronic Insurance Policy V1.0

Copyright © 2016 Nagarro Software Pvt. Ltd.

All rights reserved. This document is confidential and proprietary to Nagarro Software Pvt. Ltd. and no part of it should be reproduced, published, transmitted or distributed in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, or stored in any information storage or retrieval system of any nature nor should it be disclosed to third parties without the prior express written authorization of Nagarro Software Pvt. Ltd.

**Contents**

[1 Introduction 2](#_Toc476136439)

[2 Problem Statement 2](#_Toc476136440)

[3 Project Requirements 3](#_Toc476136441)

[4 ePolicy Application 4](#_Toc476136442)

[4.1 Web Application 4](#_Toc476136443)

[4.2 Alert and Notification System 4](#_Toc476136444)

[5 Design Requirements 4](#_Toc476136445)

[5.1 Functional Requirements 5](#_Toc476136446)

[5.2 Non-Functional Requirements 5](#_Toc476136447)

[6 Artifacts 5](#_Toc476136448)

[7 Assignment 6](#_Toc476136449)

[8 Guidelines 8](#_Toc476136450)

# Introduction

“YCompany” is one of the renowned name in Insurance providers in industry providing team and life insurance to customers in US. The company's product offerings include term and life insurance.

This document describes a problem that “YCompany” is currently facing. The company is planning to modernize the application intake process, underwriting process, policy issue and policy maintenance process.

This case study is technology agnostic. It is expected that you to go through the problem and based on your understanding propose solutions to various questions at the end of this document.

# Problem Statement

“YCompany” server more than 200 million customers across various geographies. YCompany being a key player in its business area has been facing challenges with the manual process currently in place.

YCompany sells its policy through agent, there is no direct sale to the end customer.

* End Customer reach out to Agent looking for term and life policy.
* Agent provide details with respect to the various product offering from YCompany.
* The prospective customer then provides details to the Agent, which include personal details, immediate dependent details.
* The information is collected and filled in paper application by the Agent,
* Agent then gets the signature on the paper form.
* The paper form is then sent through snail mail to YCompany.
* On receiving the application in paper format, the application is first scanned and stored in a document management system and an application id is generated.
* This application is then manually keyed into the system where a link is also set up with the document management system id.
* The application then goes through the **underwriting process**, which is currently done manually by an underwriter, where,
  1. based on the face value of the policy the premium is decided for the policy and policy is accepted or rejected.
* Post underwriting, the application is keyed in to mainframe system which runs a nightly batch process to validate all the data points in the policy against the business rules, post all validation an application becomes a policy.
* The policy is issued and is again mailed by snail mail to the customer address. This also adds to the cost of the policy issue.
* Policy is issued and
  1. the policy becomes in effect only when the first premium is paid for the policy,
  2. the premium can only be paid be check by the customer which are sent to YCompany, which also is inefficient as this caused delay in the premium collection.
  3. For any update or change required to the policy like address change or correspondence change or premium collection change from monthly to quarterly or semi annually, these are all done by calling up the billing department of the company. There is no self service portal for the customer.

# Project Requirements

“YCompany” is willing to build customer self service portal, agent portal along with an internal portal accessible to the internal staff. YCompany has collaborated with 3rd party to provide underwriting service or them and they expose service to be consumed by YCompany for any application coming it.

* Customer Self Service Portal
  + Customer can create a login using their policy details
  + Customer should be able to change the correspondence address
  + Customer should be able to change the billing cycle
  + Customer should be able to make electronic payment against policy premium.
* Agent Portal
  + Agents should be able to login to the Agent Portal. The login for Agents are created when they register with the company for the first time.
  + Agent should be able to submit the policy electronically.
  + Agent should be able to get eSignature of the customer
  + Agent should be able to look at the current status of the application.
  + Agent should be able to see report of the applications submitted and converted to policy
  + Agent should be able to see the total premium for the policy issued by them.
* Internal Portal (Underwriter/Admin/ Billing Department /Auditor)
  + Internal user should be able to login to the portal.
  + Internal user should be able to look at the incoming application.
  + Internal user should be able to update incoming application if any rectifications are required.
  + Internal user should be able to look at the underwriting results coming from a third party and should be able to edit or update the underwriting result if required.
  + Underwriting data should be stored for further auditing.
  + Internal User should be able to generate reports on the total volume of application coming in and conversion to policy.
  + Internal User should be able to generate report for policy defaulting on payment.
* Process Automation
  + Intake process of application to send data to **mainframe** and send policy data to 3rd Party for Underwriting.
  + Receive underwriting results and post received the data is automatically processed in mainframe in the batch cycle.
  + Any rejects from mainframe cycle should be notified.
  + Integration with 3rd party payment provider should be done for premium collection.

# ePolicy Application

The proposed ePolicy System will essentially provide the following:

* 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

## Web Application

Web application provided by proposed ePolicy system will allow legitimate users to access application and perform their desired tasks. Some desired features/functionalities have been mentioned above

## Alert and Notification System

Customer and Agents should be notified (via SMS/Email) as soon as policy status changes or premium deduction date or premium default happens. Similarly, alerts should be sent to the billing department if the policy defaults.

# Design Requirements

Below are some design requirements for ePolicy

* Design for evolution, flexibility and reuse
* Componentize functionality as services
* ePolicy environment should be up and running 24 x 7. In case of any crashes system should again restart on its own.
* System should be reliable.
* System should have enough logging to help in debug any error condition.

## Functional Requirements

1. The proposed system shall be based on well-defined data and interface standards.
2. Proposed system should utilize an identity management system with role-based user authentication and authorization.
3. Actions the different roles can perform within the system must be configurable without code changes.
4. The proposed system should be able to store, backup, and recover system data in a distributed environment.
5. The system should enforce encryption strategies for sensitive data at rest. Security of confidential business data should be in compliance with the industry standards.

## Non-Functional Requirements

1. System should have flexible design and it should be able to handle increased load in future.
2. It should support both on premise and cloud-based deployment with auto-scaling to match demand.
3. The offered solution must complete 99% of provided services in less than 5000 milliseconds, over both the peak and non-peak hours
4. Proposed solution will have the ability to monitor and maintain the system, and include testability, configurability and upgradeability.
5. Proposed solution must provide guard against various security threats (OWASP Top 10 etc.). System components should utilize industry-proven security standards and protocols.
6. Systems is only in English with no multi-lingual requirements.

# Artifacts

Following artifacts should be used to complete the assignment.

1. Estimation  
   <http://coe/portfolio/estimation-template-v2>
2. Solution Approach Design Document   
   <http://coe/portfolio/high-level-design-document-template>
3. DAR Template

****

# Assignment

Following is the overall tasks for the assignment of the project.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Sr.** | **Question** | **Technology** | **Details** | **Marks** |
| 1 | Create a Solution Approach Document containing high level solution design.  ***Note****: Use Solution Approach template for this. Pull out NFR in a single page as you think appropriate. You also need to provide the source files (Visio) of the diagrams you have created.* |  | Your solution approach must have following concepts:   1. Detailed solution architecture covering multi-layer design and approach 2. Non Functional Requirements Coverage 3. Technology Stack 4. Performance and Scalability 5. Assumptions & Scope (In Scope/Out Scope) 6. References / Appendix (if any) | 50 |
| 2 | Create an estimation for the above mentioned selected technology.  ***Note****: Use Estimation Sheet Template. Also provide the tentative schedule & resource plan for the project implementation.* |  | Cover all stages of below phases in excel and give final numbers.   1. Requirement Specifications and System Design 2. Implementation 3. Testing 4. UAT 5. Documentation, Go-Live, and Transition to operations 6. Project Management and Coordination | 30 |
| 3 | Create a DAR to define the  technology adoption for live workflow, eSignature and NFR  ***Note****: Your document should have detailed comparison of tools / technologies and your recommendation out of those.* |  | DAR Document containing:  1. Identification and explanation of tools for comparison  2. Explanation of each attribute of comparison in detail  3. Comparison Matrix  4. Recommended Tool  5. Risks  6. Assumptions  7. Pricing  8. References | 10 |
| 3 | A minimal working POC showcasing the key features of the proposed solution. | MEAN/PHP/.Net/J2EE (or any language of your choice) | The evaluation criteria will be focused on below key concepts rather than solely on coding and full fledge implementation:   * Design considerations of the application * Layers, tiers and modules of the application * Appropriate project folder structures for various layers like presentation, business, data access etc. | 10 |

# Guidelines

* Take relevant assumptions during the application architecting/designing wherever either requirements are not clear to you or the information required is not provided in this case study.

Following guidelines must be adhered when you create your deliverables.

1. Your deliverables such as documentations etc. must be ***Client Ready*** presentable format. It means that your deliverables should not have any grammatical mistakes, formatting issues etc.
2. Please try to stick with the format of the templates provided in the section 5.
3. For diagrams, you can use or take hints from AG – architecture design guidelines which you can find at: <http://coe/portfolio/diagram-preparation-guidelines-best-practices>
4. Your POC solution must be zipped after cleanup of all referenced assemblies if there is any. Your solution should be able to build/compile and run after unzipping.