# John Hancock Project Coeus Frontend Assembly Specification

# Project Overview

## System Description

Currently, various JH business units within the US Division have their own risk assessment process. Some business units are more advanced than others. The process involves assessing the risks of many functions and reporting on results. This siloed approach leads to a lack of consistency in regards to data input, templates, frequency of assessments and reporting at the US Division Operations level. Furthermore, there is no tool or means of capturing an overall inventory of controls.

This project will build a tool to capture and assess risks and controls data at the function level across various Business Units within US Division Operations

This architecture covers both backend and frontend implementations.

This assembly will implement Web Admin app using HTML5 and AngularJS. It will be responsible for integrating backend REST API. **This assembly is responsible for implementing all other pages which are not implemented in Frontend1 assembly.**

## Competition Task Overview

A complete list of deliverables can be found in the TopCoder Assembly competition Tutorial at:

<http://apps.topcoder.com/wiki/display/tc/Assembly+Competition+Tutorials>

*Note: Extensive implementation notes are provided at method documentation on UML Class Diagrams. Please follow them for implementation.*

*Note: Please read the whole Application Design Specification first. All the details not mentioned in this specification are provided in that document.*

### Front End Classes

This assembly is responsible for implementing AngularJS services/controllers defined in Frontend Class Diagram. It will also be responsible for integrating backend REST API with frontend pages. **This assembly is responsible for implementing all other pages which are not implemented in Frontend1 assembly.**

## *Home page*

Prototype: Home page(Owners and Approvers)

Controller: homeCtrl

* $scope.searchAssessment can be used to search assessments.
* If user clicks any assessment in search result, it will be redirected to Assessment Details page.
* If user clicks "Draft"/"Awaiting"/"Approved" button, it will be redirected to Assessments page.
* The approver name in search result should be the latest approver/rejecter of the assessment.
  + if assessment.Rejecter is not null, it should be assessment.Rejecter (FirstName + Lastname).
  + else if assessment.DivisionRiskManangementApprover is not null, it should be assessment.DivisionRiskManangementApprover(FirstName + Lastname).
  + else if assessment.BURiskManagementApprover, it should be assessment.BURiskManagementApprover(FirstName + Lastname).
  + else if assessment.BUFunctionalApprover is not null, it should be assessment.BUFunctionalApprover(FirstName + Lastname).

## *Assessments page*

Prototype: Assessments page(Owners and Approvers)

Controller: assessmentsCtrl

* When user clicks "Draft Assessment"/"Awaiting Approval"/"Approved"/"Rejected" link, $scope.getDraftAssessment/getAwaitingApprovalAssessments/getApprovedAssessments/getRejectedAssessments will be called.
* If user clicks any assessment in search result, it will be redirected to Assessment Details page.

## *Assessment Details page*

Prototype: Assessment Details page(Owners and Approvers)

Controller: assessmentsCtrl

* $scope.assessment will be used to render the page.
* $scope.getOverallRiskRatingReport will be used to get overall risk rating report.
* $scope.update will be used to update the assessment. Only draft assessment can be update. Admin can update all assessments.
* $scope.approve/reject will be used to approve/reject the assessment. This is only applicable for approvers.
* $scope.export will be used to export an assessment.
* The approver name in search result should be the latest approver/rejecter of the assessment.

## Technology overview

* Windows Server 2008+ / Azure with .NET Framework 4.5 installed
* SQL Server 2012
* Microsoft IIS 8.5
* C# 5.0
* .NET Framework 4.5
* ASP.NET Web API 2.2
* HTML5
* Unity 3.5 <http://unity.codeplex.com/>
* Log4net 1.2.13 <http://logging.apache.org/log4net/>
* AngularJS 1.2.28 <https://angularjs.org>
* Microsoft OpenXML SDK 2.0 <http://msdn.microsoft.com/en-us/office/ee358824.aspx>
* ClosedXML 0.76.0 <https://closedxml.codeplex.com>
* jqplot 1.0.9: <http://www.jqplot.com/>
* Flee 0.9.26.0: <http://flee.codeplex.com/>

## Existing Documents

* Class Diagrams
* Sequence Diagrams
* Application Design Specification
* Assembly Specification
* REST API Specification
* ERD