# Angular Technical

## Background

There is a new requirement to implement a view for the user. The user should be allowed to trigger the calculation in an actuarial model. The model is still in the development phase but the programming interface has already been agreed. The model is independent from the Web application but both access single database to read and write input parameters and calculation results.

## Task

The task is to implement Angular application prototype that allows the user to do the following:

* Prepare the input
  + Main Limit (monetary, greater than 0, usually entered in millions of £, e.g. £2,000,000)
  + Main Retention (as for Main Limit)
  + Type of Execution (available values: Simple, Complex)
* Trigger Calculation
  + Button to trigger the calculation (you should mock this calculation)
* View the results (can be between 5 and 20 entries) in a tabular format
  + 3 columns
    - Reference Date (date in the following format: 22 Jan 2017)
    - Benchmark 1 (percentage value, e.g. 22.4%)
    - Benchmark 2 (as for Benchmark 1)

## Expected Solution

* Must have
  + Angular 2/4 application (in 6)
  + HTML/CSS template with all required inputs, calculation button and table with results – appropriate presentation as for analytical systems
  + Loaded data to be executed
  + Persisted results
  + Integrate with a web API developed using C#/ASP.Net core and SQL server database.
  + Database interactions should be via the ORM Dapper and use SQL.vbgh
  + Number and date formatting for the UI
  + Restricting/validating user inputs
  + Disable the button to trigger the calculation if the input is incomplete or invalid
  + Angular application built using Angular CLI
  + Use of Angular Material module
  + Interactive frontend with results shown only after the calculation is triggered
  + Simulating the execution with a random artificial delay 1-5 seconds for calculations.
  + Add extra loading animation to improve UX
  + Suitable levels of automated testing

The code implemented should be well structured and easy to maintain.