The problem is solved in two ways in two projects:

* FamousQuoteQuiz.Web – using MVC
* FamousQuoteQuiz.Web.WebForms – using Web Forms

There are also several joint projects:

* FamousQuoteQuiz.Models – contains entity models – Author and Question
* FamousQuoteQuiz.Data – contains the connection with the database. They are used two design patterns: Repository pattern – in Repositories folder; Unit of work pattern – implemented in IFamousQuizData and FamousQuizData. I used Code First Approach to build the database.
* FamousQuoteQuiz.Common – contains global constants, variables and methods.

FamousQuoteQuiz.Web (MVC project):

The project contains two controllers for each page – MainController and SettingsController.

Settigns page – user can change the mode of the quiz. Every change of radio buttons’ state is sending ajax request to method ChangeMode in MainController and it is changing the value of GlobalVariable.BinaryMode. – depends on it its loading Binary mode or Multi choice mode

Main page: we pick a random question from database using RandomFactory (folder Infrasctructure/Factory)

* If user select ‘Binary Mode’ partial view \_BinaryModeView loaded – we pick random author from database.
* If user select ‘Multi choice mode’ we add right answer (author id) in HashSet<int> and pick two random authors’ ids from database

When user click ‘Yes’ or ‘No’ (in Binary mode) or select some author (in Multi choice mode) an ajax request is sending with parameters questionId, authorId and choosen answer (if Binary mode). The ajax request calling method IsUserChoiceCorrect, which returns partial view (\_ResultView) with context, depends on the fact is user’s answer correct or no.

Used packages: Entity Framework 6.1.1, AutoMapper, Ninject, Bootstrap