***Документация***

Проектът “Exam Question” представлява проста система client/server, в която сървърът се явява потребителя-изпитващ,а клиентът представлява потребителя, който ще бъде изпитан.Основната идея на проекта е изпитваща система, при която сървърът задава броя въпроси и времето за което трябва да се реши теста, а потребителят (или потребителите) решава теста , като след приключване разбира оценката си.

**Namespace ExamQuestions съдържа :**

* **Client** – windows form application – служи за избразяване на самия тест, получаване на информация за изпитващия и резултат от проведения тест.
* **Server** – windows form application – служи за изпращане на информация към клиента (брой въпроси и време с което разполага всеки потребител), добавяне на въпроси към теста и получаване на информация относно изпитващите и техните оценки от теста.
* **CreateXml** – console application – служи за дефиниране на класа Question, както и за създаване на xml файл и писането в него.
* **XmlReader** – class libraly - служи за четене от предварително създаден xml файл и създаване на контейнер от въпроси , който да послужи за взаимодействието на client/server.

**Console application CreateXml съдържа:**

* **Класа Question** – който описва въпросите от теста. Променливата textQuestion описва текста на въпроста, answer представлява 4те възможни отговори за съответния въпрос и numberOfRightQuestion е номерът на верния отговор (от 1 до 4 вкл.)
* **Класа XmlFile** – дефинира xml файл

**-public void FillXml(List<Question> quizzes)** – запълва дефинирания xml файл с въпросите подадени като параметър

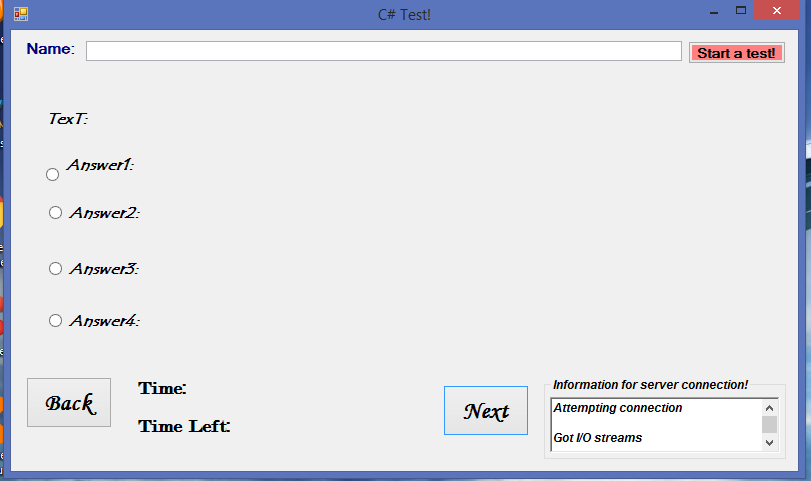
**-public void CreateNode(Question q)** – създава един връх от xml документа.

* **FirstXmlFile.cs** – създава xml файл именуван testC#.xml и го запълва с 30 въпроса с по 4 възможни отговори

**Class libraly XmlReader съдържа:**

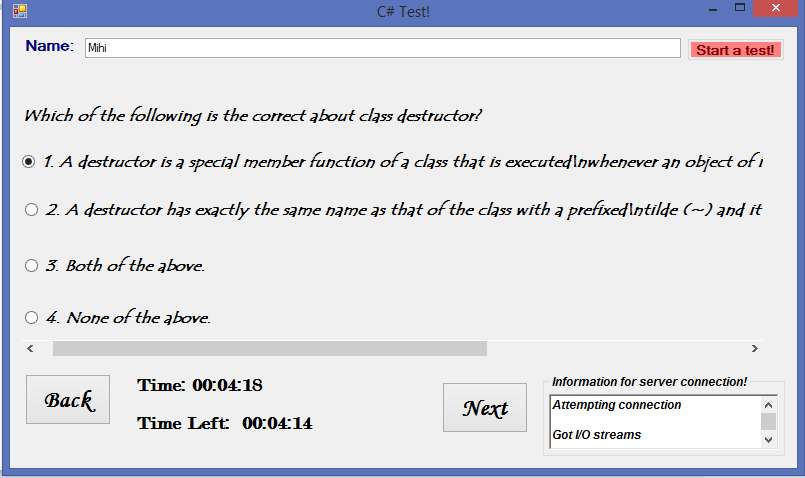
* private List<Question> quiz – променлива описваща самият тест.
* private void FillQuiz() – чете xml файл и запълва quiz с прочетените въпроси
* public void InsertInXml(Question userQuestion) – отваря се xml файла и се добавя нов въпрос, подаден като параметър, като не се нарушава структурата на хml документа.
* public void Shuffle() – след като сме запълнили пременливата quiz въпросите в нея се разбъркват, за да се подадат на потребителя в произволен ред.

**Windows form application Client**

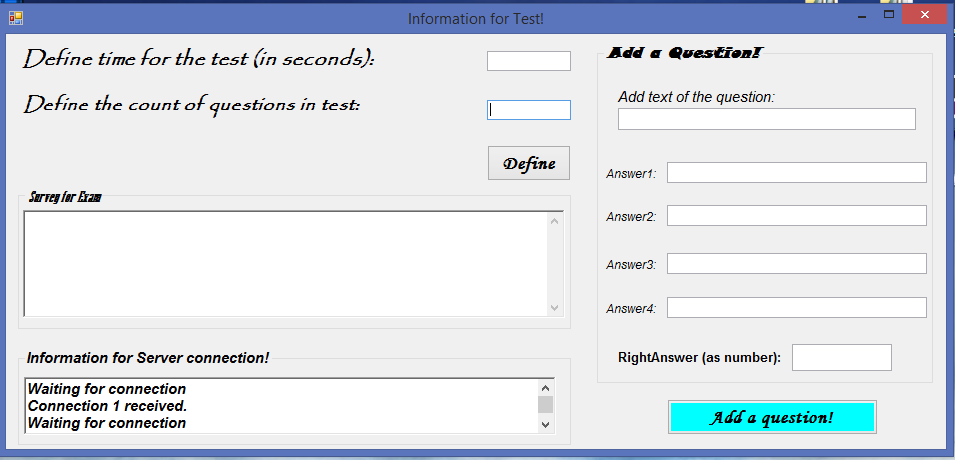


* **Text field after label name** – тук всеки потребител (client) въвежда името си.
* **Button Start a test!** – стартира теста като запълва панела (използвала съм панел , тъй като е възможно скролване при по големи текстове на въпросите или отговорите) съдържащ самия въпрос и 4те възможни отговори, също така стартира и времето, с което разполага потребителя за теста (получава се от сървъра)
* **Button Back** – връща потребителя на предишния въпрос, ако такъв съществува.
* **Button Next** – зарежда се следващия въпрос , ако потребителят е на последния въпрос текста на бутона се преименува на Submit, при натискане на бутона Submit потребителят се информира за оценката, и информацията , включваща името на потребителя и негова оценка се изпраща на сървъра
* **Text field information for server connection** – информира потребителя дали е установена връзка със сървъра.

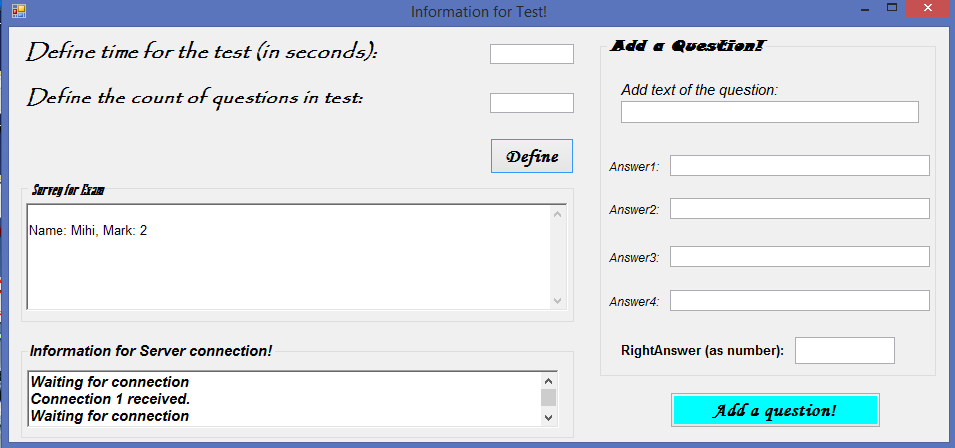
След натискане на бутона Start a test! , той става забранен , т.е. не може да се натиска повече и формата изглежда така:



**Windows form application Server**



* **First text field** – тук сървърът задава времето, за което потребителят (потребителите) трябва да решат теста
* **Second text field** - тук сървърът задава броя на въпросите за теста , като ако във xml файла , който съдържа въпросите няма достатъчно , програмата подканва сървъра да добави още въпроси.
* **Group box Add a question** – добавя въпрос към xml файла , който съдържа теста.
* **Text field information for server connection** – информира сървъра за установена връзка с клиент приложението.
* **Button define** – при натискане на този бутон, сървърът опакова информацията за времето за теста и броя въпроси в стринг, който се изпраща към клиента (клиентите)
* **Text field survey for exam** – информира сървъра за имената на клиента (клиентите) и оценката, която е получил на изпита :



***Изготвил: Михаела Чуренска***

***Спец. Компютърни науки,***

***Курс: 3 , група: 6, ФН: 81126***

**Използвана литература:**

* <https://www.dotnetperls.com/xmlwriter>
* <http://csharp.net-informations.com/xml/how-to-read-xml.htm>
* <http://codepolicy.com/skill-test/c-oops-skill-test/>
* <https://tests4geeks.com/c-sharp>
* <http://www.withoutbook.com/OnlineTestStart.php?quizId=71>