Test 1:

* Design the database (create tables, indexes, primary keys, foreign keys) for two tables
  + USERS
    - Will contain a list of users
    - Will have a username and a password field.
  + Roles
    - Will contain the list of roles for the users
  + Create a foreign key between the two
  + Create sql statements to insert data into the two tables (two users, three roles) of your choosing
* Create a C# Windows forms application for a login function
  + The form will have a username and a password field, plus a login button
  + When clicking on the login button will verify in the database the match with the data in the USERS table
  + Will return on a message box success or fail, if the login was successful or not

You must use object oriented design and programming (create classes for USER / ROLE)

               The classes must have CRUD methods (Create, Read, Update, Delete) methods, and other methods as you see fit (for example the USER class may need a Login method). Also you can use store procedures for queries(not mandatory but desirable)

It is recommended to use an n-tier architecture

               -DAL (Data Access Layer) for database connectivity

               - BLL (Business Access Layer) that uses DAL classes for CRUD operations but it does not connect directly to the database

- the app.config will contain any configuration parameters (such as sqlConnection strings and such)

What you need to provide is sql scripts to create the database structure and insert any data into it (Microsoft SQL Server version 2012, you can download and install the Express version from Microsoft, it is free)

And also a visual studio solution (visual studio solution, version 2010 min)

You also have to provide detailed configuration documentation so that I can configure and run the application in my development environment.

Test 2

1. We want to develop a software (Java or C#) with the next requirements:

* This app has to generate a compressed file (zip) that contains a text file with “.meta” extension and a folder named with the current day (Monday, Tuesday, etc…). Inside this folder you have to insert the images which are in a route to specify.
* Inside the text file (.meta) you have to generate a line per each image, which is formed the next way: separated by one character to choose from the configuration file.
  + ID: is formed with the current date in Julian format (YYJJJ) plus a consecutive of 5 digits which can’t be repeated all day long (it has to be reset daily).
  + Image creation date: (format YYYY/MM/DD HH:MM:SS).
  + Image route inside zip file.

Example:

ID | Creation date |Image route

1102700001|2010/01/27 10:35:10| thursday /image1.png  
1102700002|2010/01/27 10:35:11| thursday /image2.tif  
1102700003|2010/01/27 10:35:12| thursday /image3.gif

Where “11027” is the Julian day and “00001” is the consecutive

* The name of the created compressed file has to be in the next format: YYYY\_MM\_DD\_hh\_mm\_ss.zip where:
  + YYYY: Actual year
  + MM: Actual month
  + DD: Actual day
  + hh: Actual hor
  + mm: Actual minute
  + ss: Seconds
* You have to use an XML configuration file, this file structure it is at your discretion but at least it should contain the next information:
  + Image routes from where to obtain the images to insert in the zip file.
  + Delimiter character (for example “|” or coma “,”).

1. The second part consists in develop another software that reads the file generated by the first application created in the first part of this test, and it should do the following:

* Unzip the file.
* Read the .meta file and insert a record in a SQl Server database with all the information contained on the file. Insertion should be done using a stored procedure
* Key field must be the ID that should not be repeated.
* Prove that the image of reference actual exists physically.

You have to send the code with both applications and the SQL Server data base.

Generate scripts for database tables, stored procedures and data

\*archive it with a password if email client does not let you send it