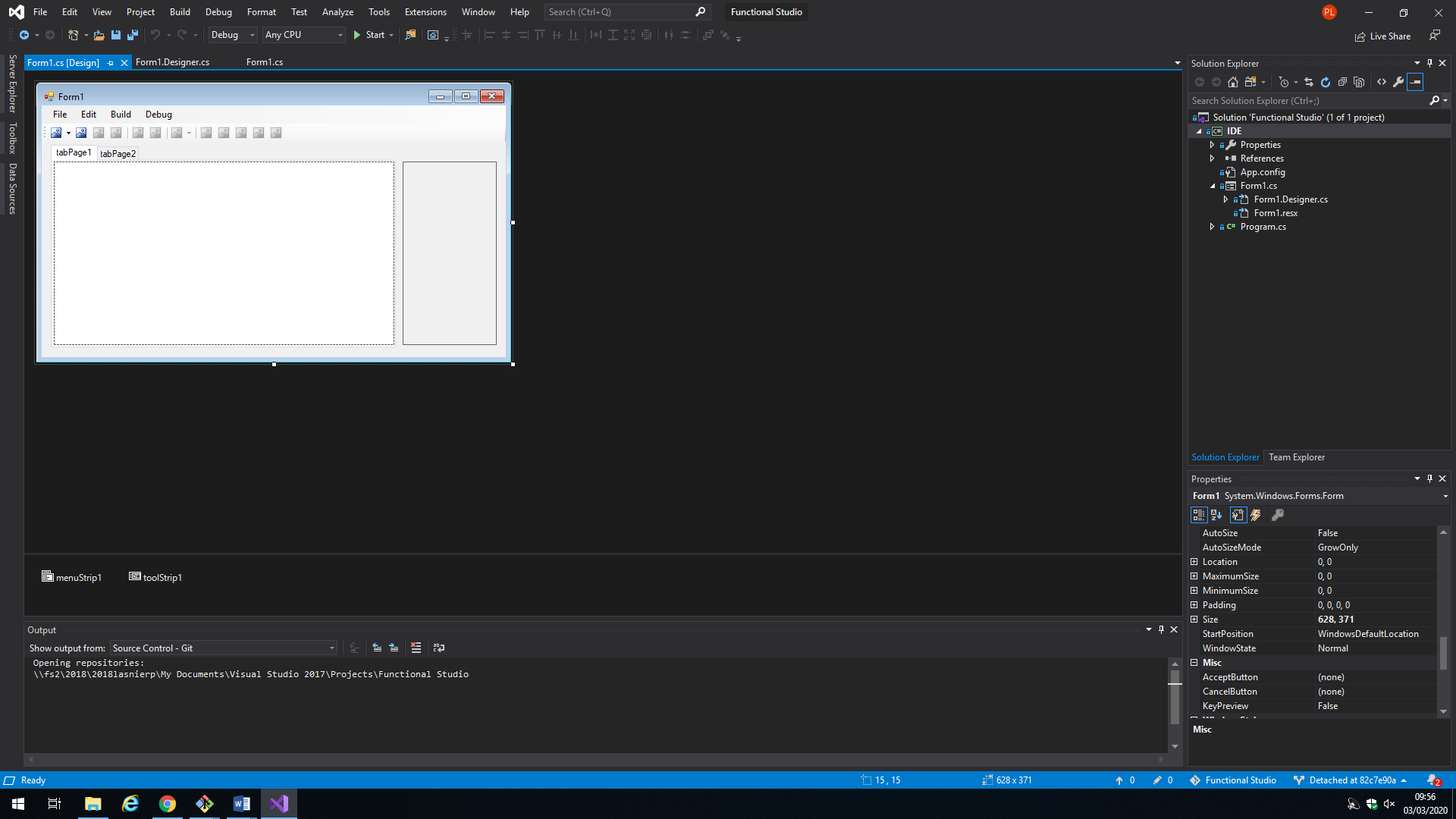
# Design

## Diagram information:

|  |  |  |
| --- | --- | --- |
| Class definition | | Class  +Property (get only?) +Method() +(return type) Function() -Private Processes() |
| Function/method description | | Function()/method()  return type  description |
| Associations | Inheritance |  |
| Aggregation |  |
| Composition |  |
| Method/function |  |
| Event |  |
|  | |  |

## First UI attempts

No code as of yet, only skeleton of UI:

**Form1.Designer.cs controls**

private System.Windows.Forms.MenuStrip menuStrip1;

private System.Windows.Forms.ToolStripMenuItem fileToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem saveToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem saveAsToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem editToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem buildToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem debugToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem undoToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem redoToolStripMenuItem;

private System.Windows.Forms.ToolStrip toolStrip1;

private System.Windows.Forms.ToolStripSplitButton toolStripSplitButtonNew;

private System.Windows.Forms.ToolStripMenuItem newFileToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem newProjectToolStripMenuItem;

private System.Windows.Forms.ToolStripButton toolStripButtonOpen;

private System.Windows.Forms.ToolStripButton toolStripButtonSave;

private System.Windows.Forms.ToolStripButton toolStripButtonSaveAs;

private System.Windows.Forms.ToolStripSeparator toolStripSeparator1;

private System.Windows.Forms.ToolStripButton toolStripButtonRedo;

private System.Windows.Forms.ToolStripSeparator toolStripSeparator2;

private System.Windows.Forms.ToolStripSplitButton toolStripSplitButtonStart;

private System.Windows.Forms.ToolStripMenuItem startToolStripMenuItem;

private System.Windows.Forms.ToolStripMenuItem startDebugToolStripMenuItem;

private System.Windows.Forms.ToolStripButton toolStripSplitButtonUndo;

private System.Windows.Forms.TabControl tabControl1;

private System.Windows.Forms.TabPage tabPage1;

private System.Windows.Forms.TabPage tabPage2;

private System.Windows.Forms.ToolStripSeparator toolStripSeparator3;

private System.Windows.Forms.ToolStripButton toolStripButtonPause;

private System.Windows.Forms.ToolStripButton toolStripButtonStop;

private System.Windows.Forms.ToolStripButton toolStripButtonStepOver;

private System.Windows.Forms.ToolStripButton toolStripButtonStepInto;

private System.Windows.Forms.ToolStripButton toolStripButtonStepOutof;

private System.Windows.Forms.Panel panel1;

## Adding text editor functionality

## Section 1

**Data modelling**

* You must include as a minimum:
  + A flow chart in a sensible level of detail (you are trying to show how your program meets the original objectives – you may need more than one for a complex program)
  + A set of tables for any data storage (this could be hard code or database) showing variables possible – by procedure/database table would be a good way to do order this. A possible outline of minimum headings is here, but you can add others:

|  |  |  |  |
| --- | --- | --- | --- |
| VarName | VarType | Validation | Brief description of use |

* Include at least 2 complex algorithms in structured English or Pseudocode – commented where necessary.
* If you are proposing to use a database as part of your solution:
* ERD (Entity Relationship Diagram) for tables
* Table design including primary and foreign keys
* Samples of SQL algorithms
* Explain how this meets the objectives.
* If you are proposing a fully OOP solution (note, marks for a simple OOP/partial implementation would probably not gain any more than a good fully implemented event driven program):
* Class description
* Class diagram
* Links between applications/hardware – ie between your program and a Word Processor/text file/printer:
  + Draw an overview of how the systems link
  + Explain, in some detail, how this set of links helps the end user and solves your objectives.

## Section 2

**The interface:**

* Draw out the main interface – you can use screenshots if necessary, but in either case, the key is:
  + How does this interface help the user?
  + What features work well?
  + Have you demonstrated the planned use of a range of programming techniques?
  + How does it meet the objectives?
* Possible things to think about in terms of the interface design:
  + How does this match with 1 or more of the following interface theories:
  + Shneidermann’s 8 golden rules?
  + KLM model?
  + Fitt’s law?

## Section 3

* Security
* Backup?
* Saving?
* Threats from external sources? (ie hack, logging on – can this be easily viewed via a textfile?)
* How does this meet the objectives?

## Section 4

Look back at your design so far.

Does it?

* Show the skills that you are planning to use
* Explain, in detail, why these techniques/skills meet the objectives

If necessary, add a paragraph or two to explain how you think it meets the objectives (this could be at the start or the end of the design section)