Technology:

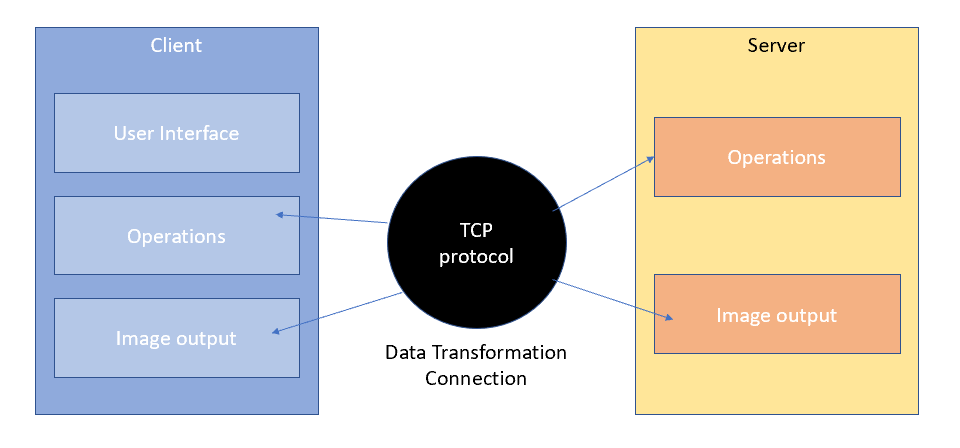
* IDE: Qt Creator 4.1.1
* Language: C++
* Library: Built Library in QtCreator
* Connection: QLocalSocket@Qt library

Functional Requirement:

* Image process: combinations of flip horizontal and vertical, rotate +/- n degrees, convert to grayscale, resize, generate a thumbnail, rotate left or right
* Cloud-based
  + In order to make it legit, product owner essentially does not require to modify Server, but replacement of the QLocalSocket in Client to corresponding TCP address is necessary.
* Quick respond
  + Per request of each image process.

Design pattern:

Architecture was built under Client-Server and Object-orientated architecture pattern, and Client was implemented as event based. The components interact under the TCP protocol with request as Request/Response.



* Client
  + All the features are implemented in Client class inherited from QDialog class
  + Instantiate widgets: buttons, labels and entry boxes. Hook up with the signals and slots in order to trigger events.
  + Instantiate the QLocalSocket—the core object of the data transformation—and connect the connected() signal and transImage() functions in order to receive and parse the handled data from the Server.
  + As long as the socket transforms data in multiple times, it is implemented send all the operations and uploaded image as an object in order to maintain efficiency of the server side: uploaded image will be packed as jpg and the operations are connected as an appending string and encapsulated as a QString object. TCP encoding as follow
    - Start from a 32bit int (cmd), 0 means uploading image without operations:, 1 means operation included. If(cmd==0), send the image size in 32bit and upload image data, If(cmd==1), send the operation string as Qstring
    - Operations:
      * 1 = flip, and the following character represents argument of horizon(0) or vertical(1)
      * 2 = rotate, the following characters represents parameter of the angle of rotation and ends with #
      * 3 = convert to gray scale, no parameter
      * 4 = resize, the following string represents the parameter of width and height, each of them ends with # for separation
      * 5 = generate thumbnail, no parameter
* Server
  + The initialization of Server is similar to Client, except the signal of newConnection() are connected to sendFortune() function and get new connected socket, in order words, the new connection from client, via nextPandingConnection method within, and connect signal of readyRead() on Client to the receiveImage() on Server. Long story short, Server will execute receiveImage() when Client is ready to receive image.
  + Handling image
    - Include 5 operations: flip, rotate, convert to gray scale, resize and genThumbnail
    - Filp: handled by QImage::mirrored(horizontal flip, vertical flip), 0 = false, 1 = true
    - Rodate: instantiate a matrix by QMatrix::rotate and call QImage::transformed() in order to apply the rotated matrix to the image
    - Resize: handled by QImage::scaled(width,height)
    - genThumbnail: handled by QImage::scaled but fixed the width as 200 and height as 150.