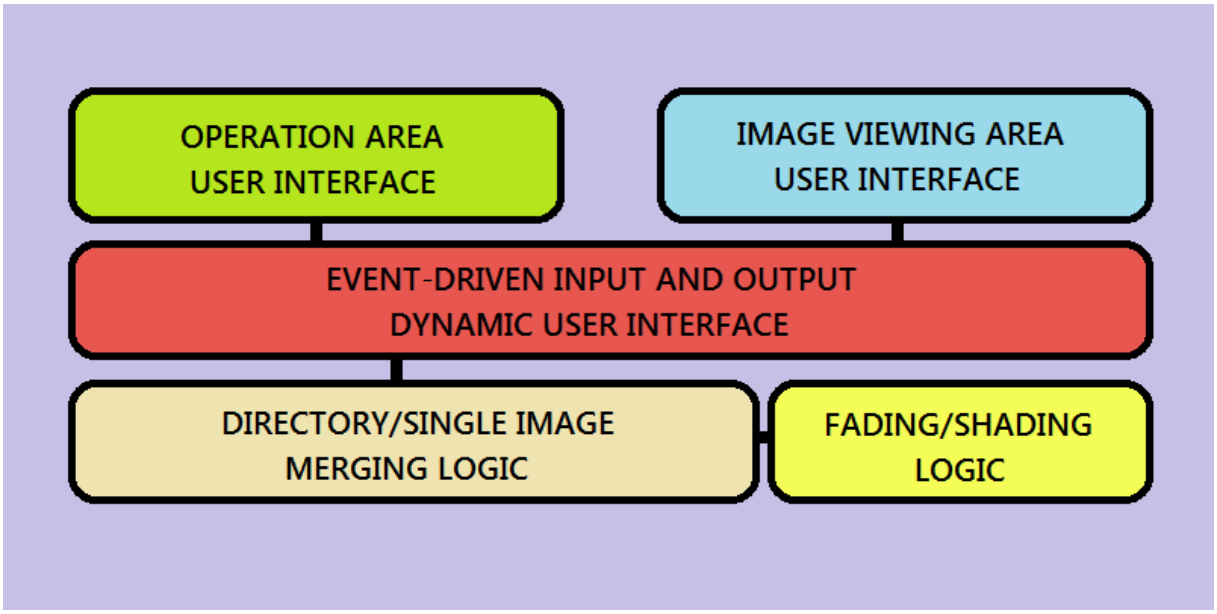
Application architecture

1. General overview

The Micromerge application is composed of 3 layers: the static GUI, the input/output control and the image manipulation logic. The GUI consists of the Operation Area and the Image Viewing Area, as it is specified in the project assignment. The image manipulation logic is divided into the Image Merging part and the Fading/Shading part.

Such partition of the application determined the assignment of tasks during the development.



2. Description of layers

- **Operation Area** - the layer functions as a display area for the various merging options. Also it allows the user to freely browse, view and choose or remove the images and directories meant for merging.
- **Image Viewing Area** - this layer lets the user view, browse and delete merged images. Eventually user can save the merged images, choosing suitable options and file properties. User can quickly switch between area by using the tabs at top of the GUI. The areas also share a menu bar, where auxiliary options can be found: clearing the whole application or exiting.

- **Event-driven Input/Output** - the layer connecting user interface and merging logic. Responsible for dynamically loading input and output images to GUI, directory browsing, initializing the merging and saving processes with user chosen or default options.
- **Image Merging Logic -** initialized by any input operations, implements the application main purpose, that is image merging. Allows to modify the basic merging process with additional bitwise operations on pixels, shading/fading and image saving.
- **Fading/Shading Logic** - supports the image merging with options of gradual shading and fading. Not directly exposed to the input/output operations.

3. Relation of architecture to the classes

The intialization of user interface areas and input/output events are implemented in the StartWin class. In hindsight, it was not the best idea, as the class became less readable and more prone to errors.

The image merging operations have been implemented in ImageMerger class, which could be further partitioned to be more in line with object oriented design.

The fading/shading options have been implemented in FaderShader class, which acts as an optional helper to the ImageMerger class .
Class diagram below: