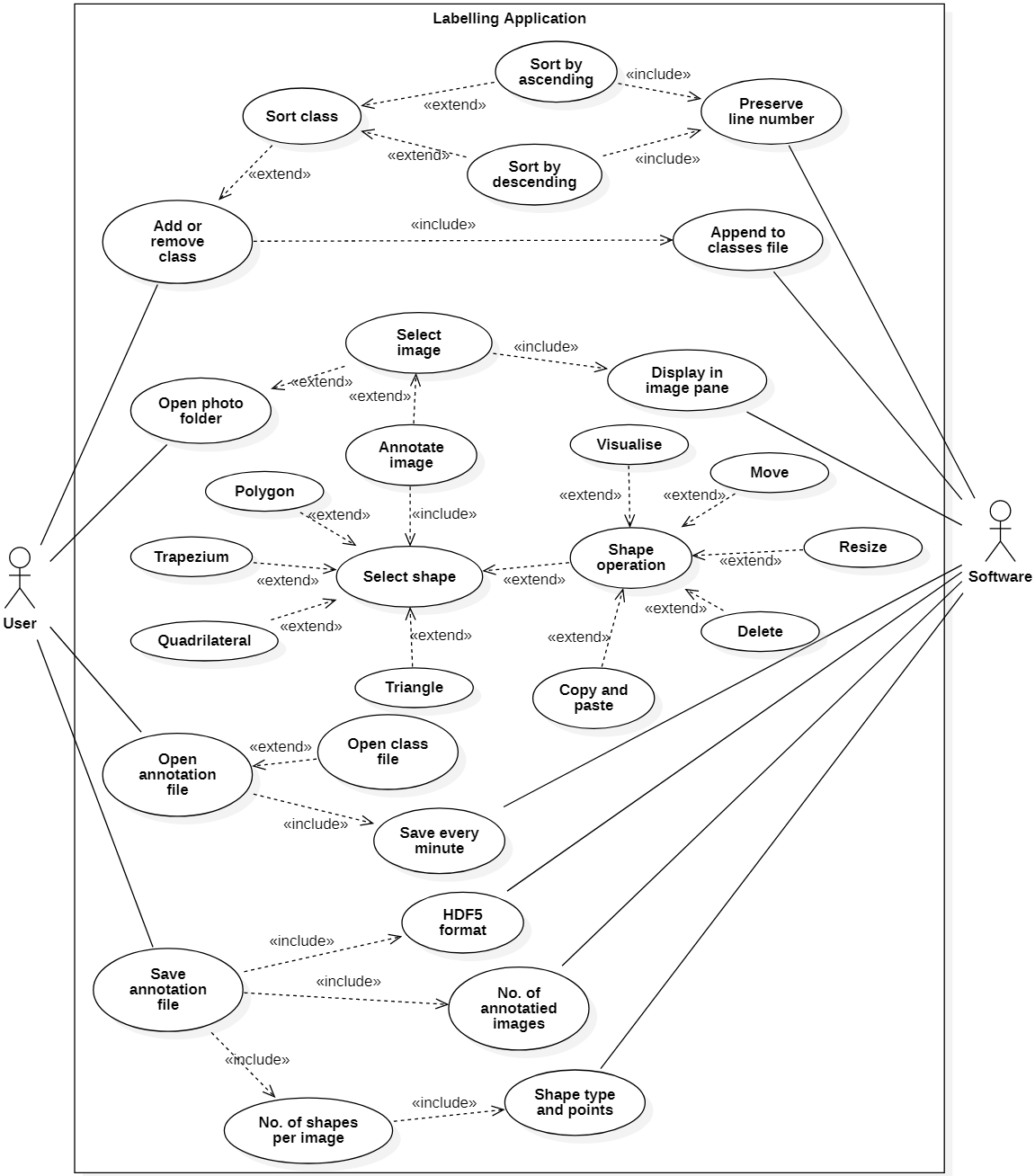
# Software Architecture

## UML Diagrams

### Use Case Diagram



The user can add remove a class in which the classes file will always be updated. There is an option of sorting by ascending or descending order but while the software still preserving the initial line number.

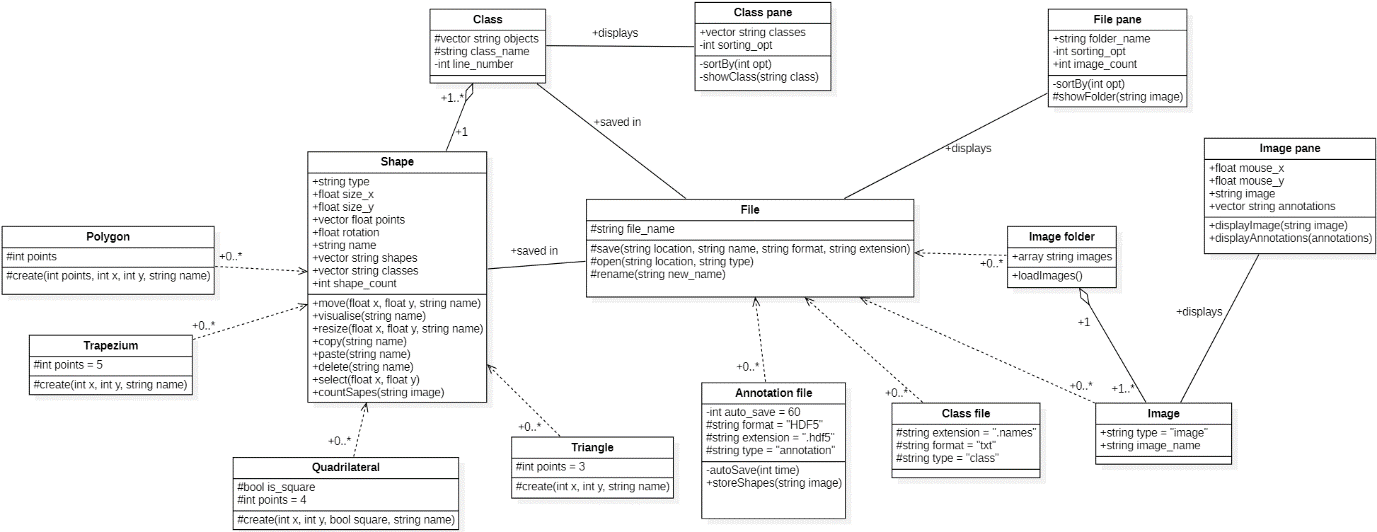
The user can open a photo folder allowing the selection of images to be displayed in the image pane.

The selected image can be annotated with shapes (including: polygon, trapezium, quadrilateral, triangle) and these have operations (including: visualise, move, resize, delete, copy/paste).

The user can open an annotation file with a class file. The annotation file will be saved every minute by the software.

The user can save an annotation file in which the software will combine the number of shapes (including: types, points) and the number of annotated images. This will be in HDF5 format.

### Class Diagram

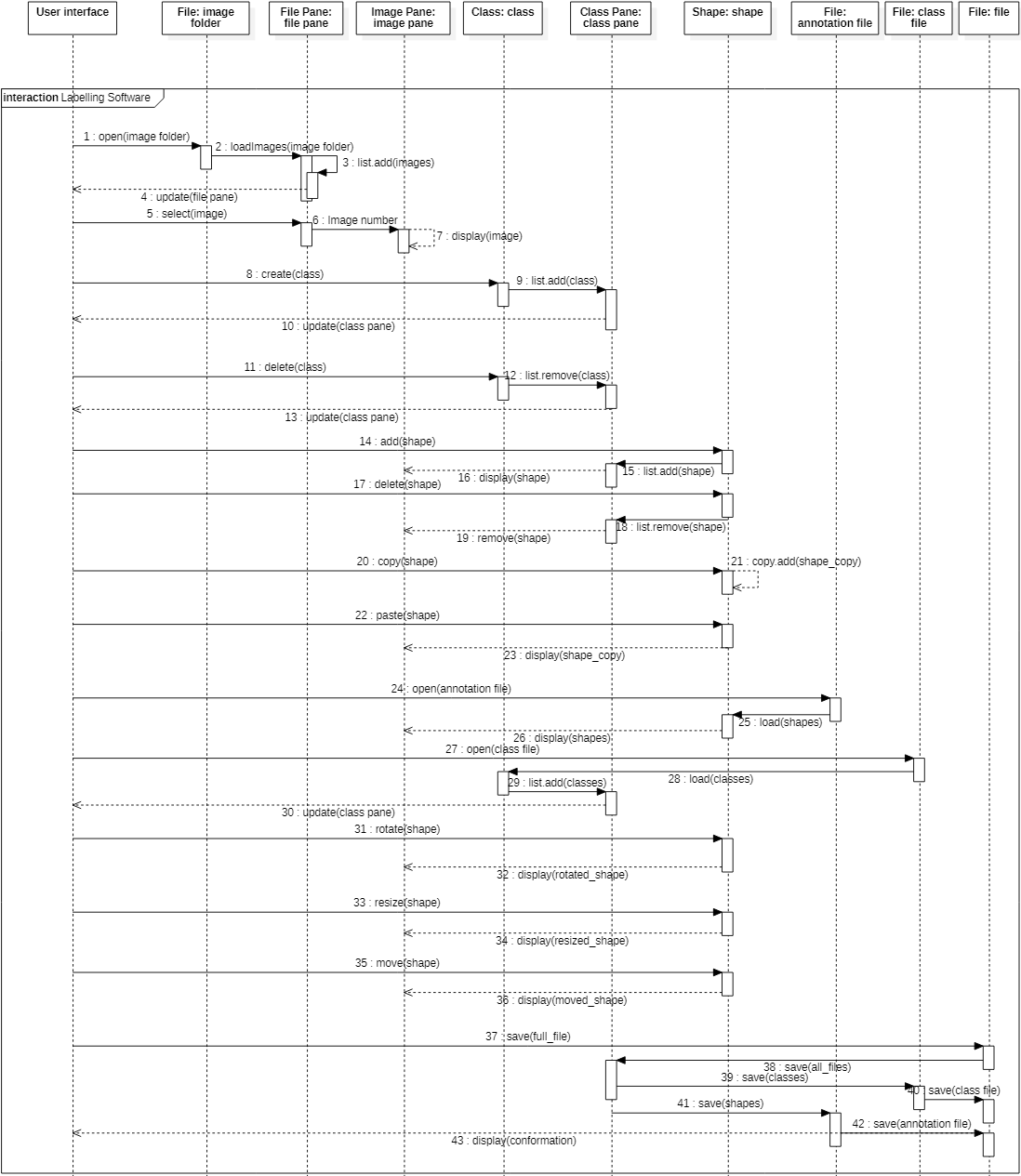


The shape (or annotation) class has many subclasses being the different shape options (polygon, trapezium, quadrilateral, triangle) in which infinite shapes can exist of any type. The shapes can then be assigned to 1 class each which is displayed in the class pane.

The file class included the subclasses: annotation, class and image file in which many images can exist but only 1 of the rest. Many images can be loaded from an image folder and one can be displayed in the image pane.

The file pane class can display many images.

### Sequence Diagram



An image folder is opened by the UI and loaded by the image folder sub-class into a list. Then the pane is updated.

An image is selected and the file pane passes the image number to the image pane class. The UI then displays that image.

A class is created and added to the list by the class and the class pane is updated.

A class is deleted and removed from the list by the class and the class pane is updated.

A shape is added by the shape class and added to the class pane. The shape is then displayed by the image pane.

A shape is deleted by the shape class and removed from the class pane. The shape is then removed by the image pane.

A shape is copied by the shape class and duplicated in the list.

A shape is pasted and the class pane is updated and then the image pane.

An annotation file is opened and the annotation file class reads the file and the shape class loads the shapes into the list. The image pane then displays the shapes.

An class file is opened and the class file class reads the file and the class loads the classes and then the class pane loads them into the list. The pane is then updated.

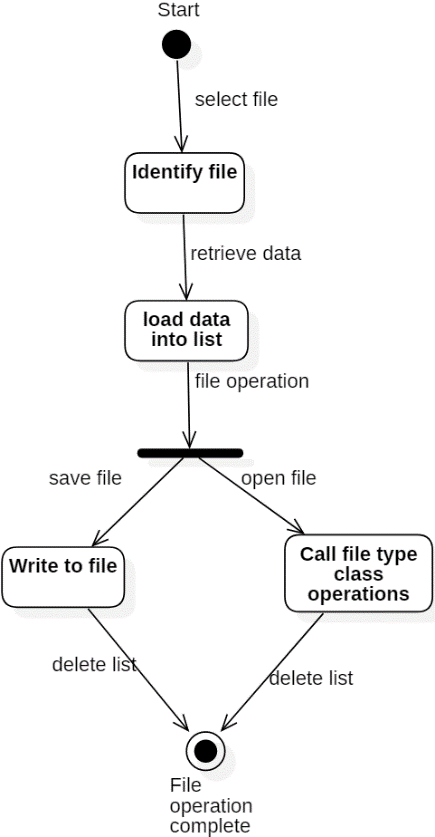
A shape is rotated by the shape class and image pane is updated.

A shape is resized by the shape class and image pane is updated.

A shape is moved by the shape class and image pane is updated.

The full file is saved by the file class by calling the class pane class to save a class file then the shape class to save the annotation file.

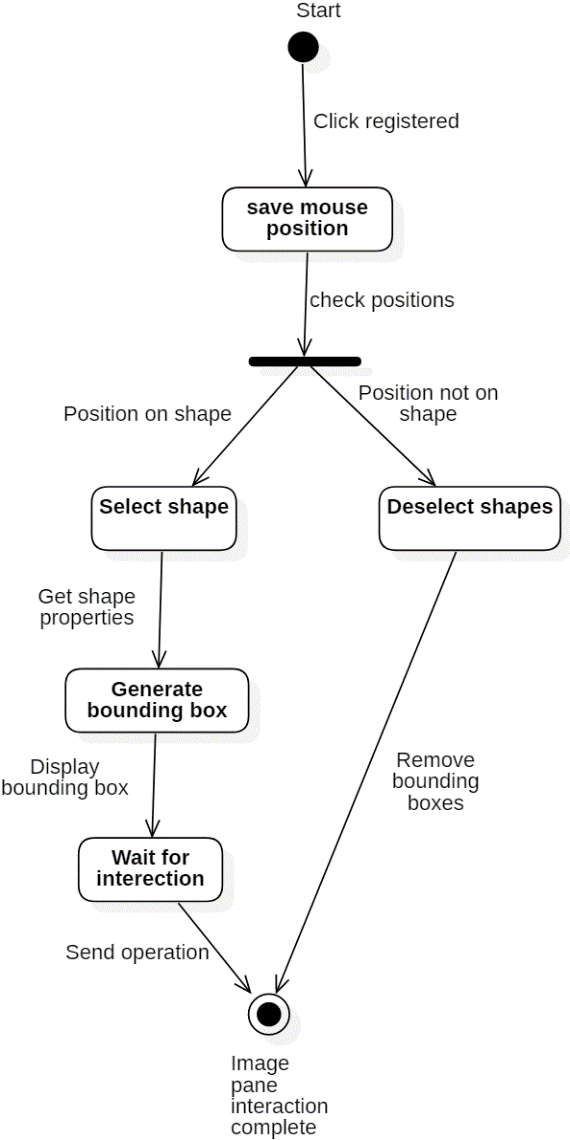
### State Diagrams



The file class will be given a file, identify it, and then load the retrieved data into a list.

If the operation was to save the file then the data will be written to the file.

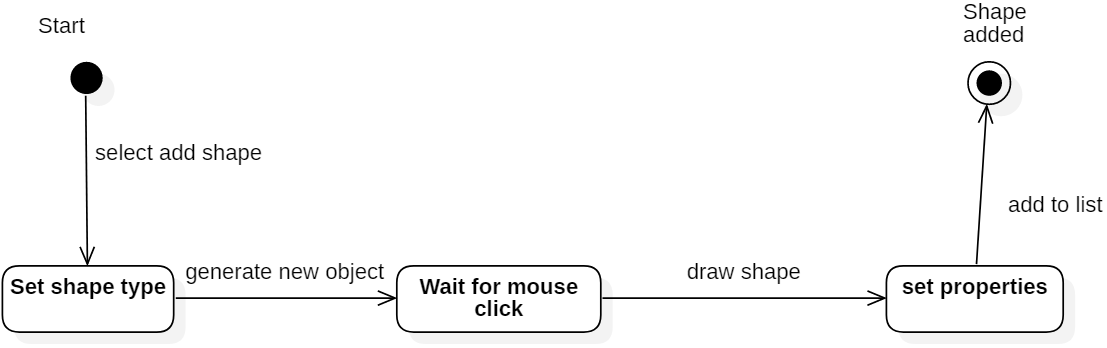
If the operation was to open the file data will be passed to the file type class.



The image pane class registers a click then saves the mouse position.

If a shape was clicked, the shape is selected so generating a bounding box for the shape to be adjusted. It then waits for an interaction and provide the shape class with the operation.

If no shape was clicked, deselect shapes by removing bounding box/es.



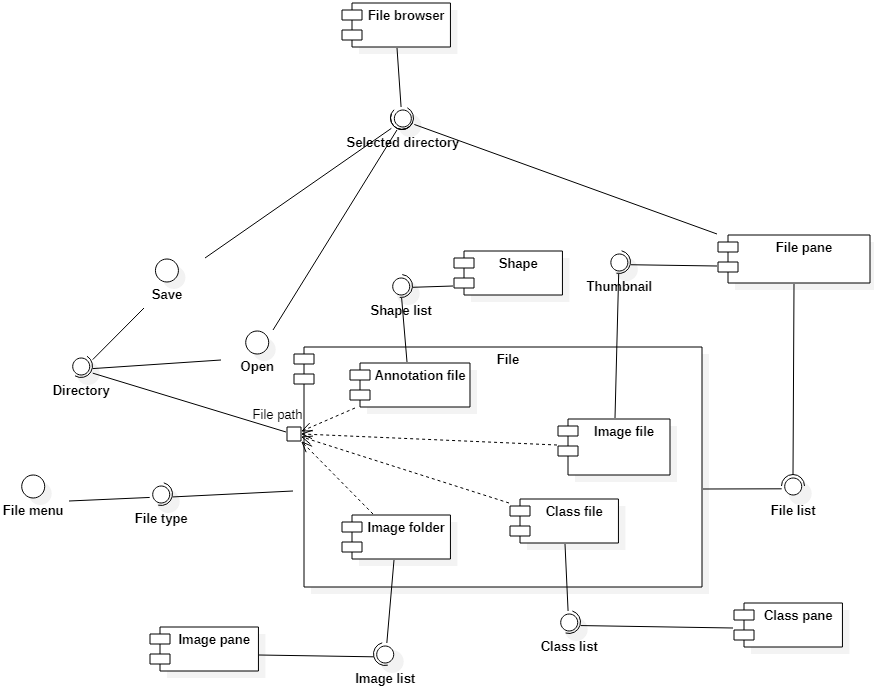
The shape class will set shape type by the selected option.

It will then generate a new shape object and wait for a mouse click.

Once received it will draw a shape with the properties defined by the user.

It will then add the drawn shape to the list of shapes.

### Component Diagram



The file component requires the file type interface provided by the file menu interface and can provide a file list interface.

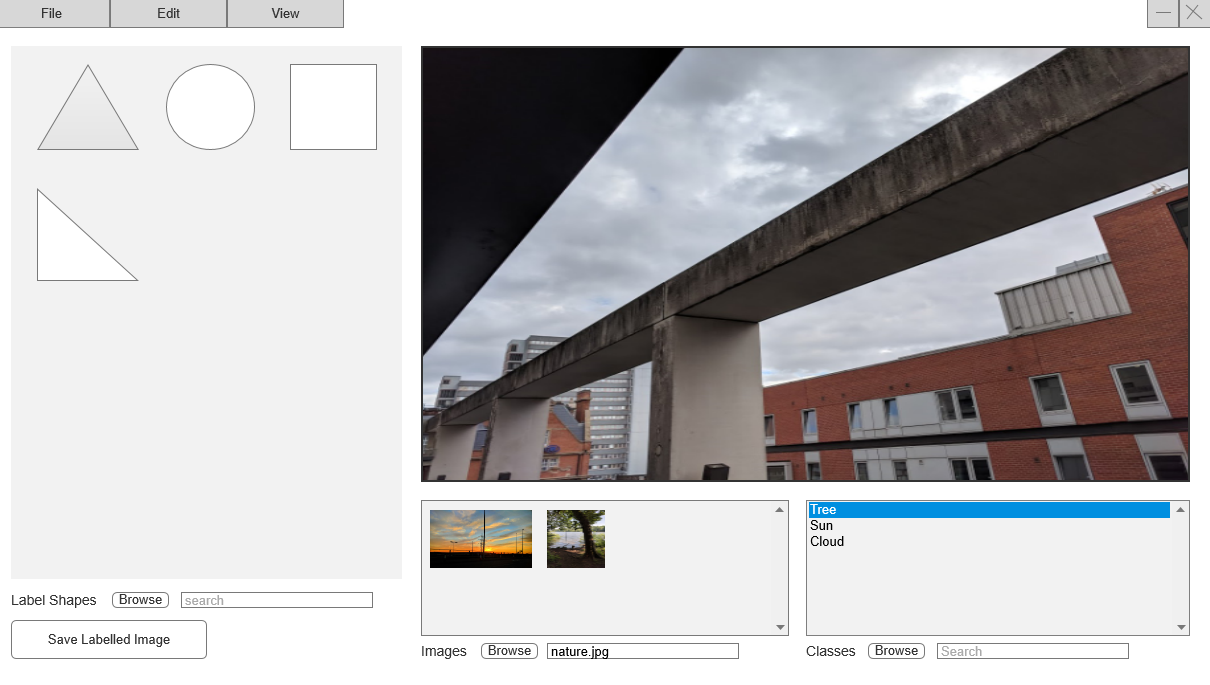
Each file sub-component requires the directory interface and can provide an object list interface in which the image pane, class pane, file pane and shape component require.

The file pane requires the file list, thumbnail and selected directory interfaces.

The file browser provides the selected directory interface.

The save and open interfaces provide the directory interface and require the selected directory.

## GUI Snapshot



## Tools and Libraries

|  |  |
| --- | --- |
| Tool/Library | Reasoning |
| Visual Studio 2017 | Good c++ support/compiler  Group members have previous experience  Good project/file organisation features |
| Microsoft Foundation Class Library (MFC) | Included in Visual Studio  Ease of use  C++ library  Lots of documentation/support |
| StarUML | Free non-commercial use  Large set of UML diagram templates and support  Auto code generation |
| OpenCV | Easy to learn (lots of support)  C++ primary interface  Focussed on real-time processing  Completely free |
| Git (with GitHub) | Completely free  Group previous experience  Integrated into Visual Studio  Easy to handle merging and branching |
| Slack | Free  Easy and organised commination  Helpful tools (integrated apps) |