|  |  |  |  |
| --- | --- | --- | --- |
| Task Number | Requirement Description | Requirement Implication | Task(s) |
| 1 | Must be able to present functionality of the system through a GUI | Making use of existing c++ frameworks, such as Qt, to be able to create and render objects that the user can use to interact with the system.  The GUI will be easy to navigate and read information from. | Research how to use Qt or a similar framework to make a functioning GUI.  Research good UI design practices. |
| 2 | Must be able to allow the user to specify a target folder for photos | Devising and implementing a method for the user to select a directory on their computer system’s storage, from which the system will read photos from. | Develop code that allows the user to select a directory.  Develop code such that the system reads compatible image files from the specified. |
| 3 | Must have buttons to browse the files within the target folder | The GUI must have buttons to allow the user to navigate through the photos within the target folder. These buttons should include options to navigate to files and open them. | Implement buttons in the GUI to allow for navigating files within the target folder. |
| 4 | Must have a pane within the GUI that lists compatible files within the target folder | Implementing a system that passes a list of compatible files to the GUI to output on the screen | Develop source code to handle displaying compatible files in a section of the GUI. |
| 5 | Must be able to sort the files within the pane with the following options:   * Ascending by file name * Descending by file name * Ascending by file date * Descending by file date | Using an appropriate sorting algorithm that can effectively sort the names and dates of file objects into ascending and descending order. | Research sorting algorithms  Implement a sorting algorithm to handle sorting relevant files |
| 6 | Must have a class selector with a browse button | Implementing a pop up window that allows the user to navigate through their directories to select class files. This window should be launches as the user selects the “browse” button for classes | Implement a system for browsing files that can open classes files |
| 7 | Must use class files which are plain text files with the extension “.names”, where each line corresponds to a class | The system should be able to read and write to plain text files under the custom extension “.names” . The system should be able to interpret each line in the text file as a class. | Develop code for reading and writing to files with custom extensions, where these files are in the structure that has been specified. |
| 8 | Should list all classes in a classes pane | Implementing a system that takes a list of all the relevant classes and lists them in a pane. | Develop source code to handle putting classes into a list and displaying them in the GUI |
| 9 | Must allow users to add and remove classes | Allowing the user to be able to create new classes to which they can annotate sections of the image. Classes also need to be removed, which should also remove annotations of that class. | Create a container of classes where classes can be created or removed. |
| 10 | Must append new and removed classes to the classes file | Appending new classes to the end of the classes file in addition to being able to find and remove classes that the user wishes to delete. | Develop code to modify the classes files to add and remove classes. |
| 11 | Must allow users to select and use one of the following shapes options to annotate classes:   * Triangle * Square/rectangle * Trapezium * Polygon (with up to 8 points) | Make buttons in the GUI to select different shape options, with only one shape option being able to be selected at the same time.  Defining each shape as an arrangement of coordinates, where the polygon shape allows the user to freely place up to 8 coordinates. The space encapsulated within these coordinates should be recognisable as a class. | Develop a set of radio buttons for the GUI.  Ensuring coordinates that represent the vertices of a shape can be put onto the image. |
| 12 | Must only allow the use of the provided shapes for annotating images | No other shape options should be available to the user. | Ensure that the set of radio buttons for selecting shapes is the only method to select shapes. |
| 13 | Must allow the user to draw the shape on top of the image | Making use of the mouse or other pointer input to be able to scale and place the preset shapes at the position of the cursor. | Develop code that places the shape or coordinate at the cursor position. |
| 14 | Should only show the borders of the shape | Displaying only the edges of the shape that defines a class whilst ensuring the rest of the shape is transparent. Thus, the border of the shape should only be displayed as a line of a few pixels wide. | Ensure the edges of the shape are displayed as a line, where the shape itself has no fill. |
| 15 | Should display annotations on the image | Existing annotations should be persistently displayed onto the Image within the GUI. | Ensure that the annotations are constantly rendered by the GUI |
| 16 | Must have an annotations file and filename selector | A pop up window similar to that of the classes selector should appear that allows the user to select annotations files. | Implement a system for which the user can browse and select annotations files. |
| 17 | The filename selector must be able to open and load annotations files | Selected files should be passed to the system to be read, where file contents are displayed correctly on the GUI as they were stored. | Ensure that the contents of the selected annotations files are loaded into the system correctly. |
| 18 | Must allow users to change the name of existing annotations files | Within the file selector, file names should be able to be selected and renamed.  Other file contents should remain the same.  Files should not be renamed to names that are identical to existing files | Include code that allows annotations files to be renamed.  Ensure that files cannot be renamed to the same name as existing files. |
| 19 | Annotations files must follow the hierarchical data format 5 standard | Annotations files must be written in a way that follows the HDF5 standard. | Research the HDF5 standard for files.  Ensure the annotations follow this standard. |
| 20 | Must store the following data in each annotation file   * Number of annotated images   For each image:   * Image file name * Number of shapes per image   For each shape:   * Shape type * Coordinates for each point | The data must be written to each annotation file in a way such a way that it can be read again when the file is opened. | Develop code for the system to be able to store the state of the user’s annotations in a way that can be read again, which includes the specified details of the data that the file should include. |
| 21 | Should display the selected image in the image pane | The image should be continuously displayed in a section of the GUI. | Ensure the relevant image is constantly rendered to a specific section of the GUI. |
| 22 | Must allow user to increase the size of shapes using the mouse | The cursor should be able to interact with shape, where if the mouse button is held and the cursor moves, the size of the shape should scale respective to the movement of the cursor. | Ensure that the vertices of shapes are interactable within the GUI.  Develop code that scales the size of the shapes when selected with the cursor. |
| 23 | Must allow user to move the vertices of polygons using the mouse | The coordinates of the vertices of a shape must change respective to the position of the cursor when selected by the cursor. | Obtain the coordinates of the cursor.  Change the coordinates of selected vertices with the cursor. |
| 24 | Must allow the user to delete shapes using the mouse | Create an option to delete a shape using the cursor if the right mouse button is clicked whilst the cursor is over the shape. Annotations associated with the shape should also be deleted. | Use coordinates of the cursor along with developing source code to delete all data associated with specified shapes. |
| 25 | Must allow the user to copy and paste shapes using the mouse | Create an option to copy and/or paste a shape using the cursor if the right mouse button is clicked whilst the cursor is over the shape. | Ensure shapes can be duplicated through interaction with the cursor. |
| 26 | Must allow the user to visualise the name of the class on top of the shape using the mouse | Class names should appear over a shapes with annotations when the cursor appears over the shape. | Ensure that the GUI displays the class name on top of its associated shape when appropriate. |
| 27 | Should automatically save the annotation file every minute, using threads | A method of tracking time should be devised.  The contents of the current working annotations file should be rewritten every minute, being implemented using threading. | Develop code that tracks the runtime of the application.  Make use of this code to have the system automatically save the file every minute. |
| 28 | Must use data structures for storing data in memory | Using learned data structures for the system storing data in memory. | Research and implement an appropriate data structures for storing data. |
| 29 | Must use a sort and a search algorithm | Using learned sorting and searching algorithms for sorting and searching for files for the system to interact with. | Research and implement an appropriate sorting algorithms . |