

# SE 3XA3: Software Requirements Specification

Group #10

Lab: L03

Aamina Hussain, hussaa54

Jessica Dawson, dawsor1

Fady Morcos, morcof2

# Contents

<b>1</b>	<b>Project Drivers</b>	<b>1</b>
1.1	The Purpose of the Project . . . . .	1
1.2	The Stakeholders . . . . .	1
1.2.1	The Client . . . . .	1
1.2.2	The Customers . . . . .	1
1.2.3	Other Stakeholders . . . . .	1
1.3	Mandated Constraints . . . . .	1
1.4	Naming Conventions and Terminology . . . . .	2
1.5	Relevant Facts and Assumptions . . . . .	2
1.5.1	Facts . . . . .	2
1.5.2	Assumptions . . . . .	2
<b>2</b>	<b>Functional Requirements</b>	<b>3</b>
2.1	The Scope of the Work and the Product . . . . .	3
2.1.1	The Context of the Work . . . . .	3
2.1.2	Work Partitioning . . . . .	3
2.1.3	Individual Product Use Cases . . . . .	4
2.2	Functional Requirements . . . . .	4
<b>3</b>	<b>Non-functional Requirements</b>	<b>6</b>
3.1	Look and Feel Requirements . . . . .	6
3.1.1	Appearance Requirements . . . . .	6
3.1.2	Style Requirements . . . . .	6
3.2	Usability and Humanity Requirements . . . . .	7
3.2.1	Ease-Of-Use Requirements . . . . .	7
3.2.2	Accessibility Requirements . . . . .	7
3.2.3	Learning Requirements . . . . .	7
3.3	Performance Requirements . . . . .	7
3.3.1	Speed and Latency Requirements . . . . .	7
3.3.2	Reliability and Availability Requirements . . . . .	8
3.4	Operational and Environmental Requirements . . . . .	8
3.4.1	System Requirements . . . . .	8
3.4.2	Productization Requirements . . . . .	9
3.5	Maintainability and Support Requirements . . . . .	9
3.5.1	Maintenance Requirements . . . . .	9
3.5.2	Support Requirements . . . . .	9

3.6	Security Requirements . . . . .	9
3.7	Cultural and Political Requirements . . . . .	9
3.8	Legal Requirements . . . . .	9
3.9	Health and Safety Requirements . . . . .	10
<b>4</b>	<b>Project Issues</b>	<b>10</b>
4.1	Open Issues . . . . .	10
4.2	Off-the-Shelf Solutions . . . . .	10
4.3	New Problems . . . . .	10
4.4	Tasks . . . . .	10
4.5	Migration to the New Product . . . . .	11
4.6	Risks . . . . .	11
4.7	Costs . . . . .	11
4.8	User Documentation and Training . . . . .	12
4.9	Waiting Room . . . . .	12
4.10	Ideas for Solutions . . . . .	12
<b>5</b>	<b>Appendix</b>	<b>13</b>
5.1	Symbolic Parameters . . . . .	13

## List of Tables

1	Revision History . . . . .	iii
2	Work Partitioning Events . . . . .	3
3	Work Partitioning Summaries . . . . .	3

## List of Figures

1	Work Context Diagram . . . . .	3
2	Use Case Diagram . . . . .	4

Table 1: **Revision History**

<b>Date</b>	<b>Version</b>	<b>Notes</b>
Feb 9, 2022	0.0	Initial Document; Completed Section 1, 2.1.1, 3.1, 3.2
Feb 11, 2022	0.1	Completed Section 2.1.2, 2.1.3, 2.2, 3.3-3.9, 4, 5
Apr 11, 2022	1.0	Updated FRs, Added Fit Criterion for NFRs

This document describes the requirements for the Abstract Art Generator. The template for the Software Requirements Specification (SRS) is a subset of the Volere template.

# **1 Project Drivers**

## **1.1 The Purpose of the Project**

The purpose of the proposed project is to enhance an existing art generator. The existing Abstract Art Generator can create a wide array of different images with many customization options. The project will add to these options by increasing the ways the art can be generated and manipulated. The finished product should allow for even more variation in art generation.

## **1.2 The Stakeholders**

### **1.2.1 The Client**

The project's clients will be Dr. Asghar Bokhari as the instructor of SFWRENG 3XA3 and the TAs of the course. They are interested in expanding the Abstract Art Generator and will be determining whether the requirements have been met.

### **1.2.2 The Customers**

The customers of the project are all those who are interested in generating abstract art.

### **1.2.3 Other Stakeholders**

The developers of the project are also stakeholders as they will be applying their skills to the project. The developer of the original Abstract Art Generator, Burak Unutmaz, is also a stakeholder as they have an interest in seeing how their project expands.

## **1.3 Mandated Constraints**

1. The project must run on Windows 10 or newer, or Linux Ubuntu 20.04.

2. The project must be completed according to the schedule included in Tasks (Section 4.4).

## 1.4 Naming Conventions and Terminology

- **Color Palette:** Up to eight different colors that complement each other that the generated abstract art is made up of.
- **Layer:** The Abstract Art Generator contains layers. Each layer is made up of a specified shape laid out in a specified pattern. The colors of these shapes are from the chosen color palette. The layers are then stacked on top of each other to create the finished art piece.

## 1.5 Relevant Facts and Assumptions

### 1.5.1 Facts

- N/A

### 1.5.2 Assumptions

- Users have access to a computer
- Users know how to use a computer
- Users know how to download files
- Users can read English

## 2 Functional Requirements

### 2.1 The Scope of the Work and the Product

#### 2.1.1 The Context of the Work

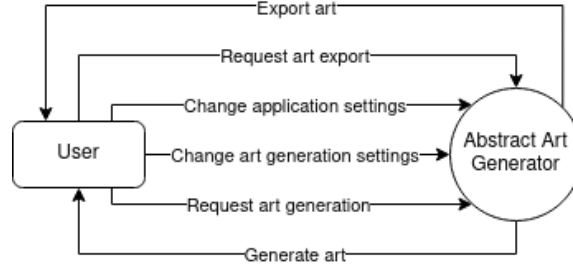


Figure 1: Work Context Diagram

#### 2.1.2 Work Partitioning

Event Number	Event Name	Input	Output
1	Generating a picture	Mouse	Generated Picture
2	Exporting a picture	Keyboard/Mouse	Image File
3	Opening the help overlay	Mouse	Help Overlay
3	Opening the settings menu	Mouse	Settings Menu

Table 2: Work Partitioning Events

Event Number	Summary
1	The user can, through mouse input, choose to manipulate and change generation settings and generate an image. The image will then be displayed to the user in the UI.
2	The user, through mouse and keyboard input, can specify a resolution and file location to export an image file to.
3	The user can open and read the help menu by using mouse input
3	The user can open and manipulate the applications' setting by using mouse input

Table 3: Work Partitioning Summaries

### 2.1.3 Individual Product Use Cases



Figure 2: Use Case Diagram

The above diagram specifies the product use cases. Generating a picture is the most complex as it involves changing generation parameters. Exporting a picture requires a resolution and file location to be specified. Opening the help overlay and changing the UI theme just involves the user accessing a **HELP** button or **THEME** button.

## 2.2 Functional Requirements

FR1. The system shall have at least 30 unique color palette options.



- Rationale: The user will have a wider range of color options to choose from.

FR2. The system shall allow the user to choose the background color of the art from the chosen color palette.

- Rationale: The user may want to choose the exact background color instead of it being randomly chosen from the color palette.

FR3. The system shall display which color the user chose as the background.

FR4. The system shall allow the user to have three generated layers.

- Rationale: Currently, the program only allows for two generated layers. This will increase the variety of art the user can generate.

FR5. The system shall allow the user to change the transparency of the layers.

- Rationale: This will increase the variety of art the user can generate.

FR6. The system shall allow the user to have text on top of the art.

- Rationale: This will increase the variety of art the user can generate.

FR7. The system shall allow the user to change the font of the text on top of the art.

FR8. The system shall allow the user to change the size of the text on top of the art.

~~FR9. The system shall allow the user to change the color of the text on top of the art.~~

##COMMENT: We decided to not implement this requirement and allow the color of the text to be randomly chosen from the color palette of the current generated art image. This is because in order to allow the user to choose the text color, we would need to include a long list of colors, and it is likely that most colors would clash with the generated art. The art will look more cohesive using a color from the current color palette.

FR9. The system shall allow the user to add a border to the art.

- Rationale: This will increase the variety of art the user can generate.

FR10. The system shall display the border options.

FR11. The system shall display the instructions for using the program.

- Rationale: This will promote the system's usability.

~~FR12. The system shall display the display settings.~~

FR12. The system shall allow the user to switch between different UI themes, such as dark mode and light mode.

- Rationale: This will increase accessibility and allow users to adjust the UI theme to their comfort level.

## 3 Non-functional Requirements

### 3.1 Look and Feel Requirements

#### 3.1.1 Appearance Requirements

LF1. The menu of the program must be formatted neatly and not be cluttered.

- Rationale: The user should find the appearance of the program menu attractive to further their interest in using the program.
- Fit Criterion: The different sections of the menu will be in-line with each other and have the same width.

#### 3.1.2 Style Requirements

LF2. The program background must be a simple and solid color.

- Rationale: If the program background is not simple, it may clash with the generated art and end up distracting the user.
- Fit Criterion: The program background will be one single color.

## 3.2 Usability and Humanity Requirements

### 3.2.1 Ease-Of-Use Requirements

UH1. The options of the program shall be easy for the user to access via the program menu.

- Rationale: The user should be able to select any option efficiently and should not struggle to find the options.
- Fit Criterion: All of the program options will be reasonably sized, positioned, and displayed on the program menu.

### 3.2.2 Accessibility Requirements

UH2. The UI text shall be legible for the average user.

- Rationale: The average user cannot make full use of the features if they find it difficult to read the instructions or option descriptions.
- Fit Criterion: The UI text size will be 12 pt.

### 3.2.3 Learning Requirements

UH3. The instructions for the program shall be clearly available to the user via the program menu.

- Rationale: The instructions are necessary for the user to be able to use the program.
- Fit Criterion: All of the program instructions can be accessed by the user through a **HELP** option in the program menu.

## 3.3 Performance Requirements

### 3.3.1 Speed and Latency Requirements

PR1. The system shall respond to the user's parameter input or selection within a reasonable time frame.

- Rationale: The user will not appreciate having to wait too long for the system to respond to their change.
- Fit Criterion: The system's response time to the user's input should be no longer than MAX\_PARAM\_TIME.

PR2. The system shall generate an image within a reasonable time frame.

- Rationale: The user will not appreciate having to wait too long for the system to respond to their change.
- Fit Criterion: The time taken for the system to generate an image should be no longer than MAX\_GENERATION\_TIME.

### 3.3.2 Reliability and Availability Requirements

PR3. The program shall be available for the user to run whenever they wish.

- Rationale: The program does not require any external resources since it is downloaded and run on the user's machine. So the program should be designed such that as long as the user's PC is working, they should be able to run the program.
- Fit Criterion: The program launches 99% of the time when the user's PC is functional.

## 3.4 Operational and Environmental Requirements

### 3.4.1 System Requirements

OE1. The program shall be able to run on most modern operating systems.

- Rationale: The program should be available to user's with common operating systems.
- Fit Criterion: The program shall be able to run on Windows 10 or newer, or Linux Ubuntu 20.04.

### 3.4.2 Productization Requirements

OE2. The program shall be available as an executable file for the user to download.

- Rationale: Having the program like this will ensure it is not difficult for the user to install.

## 3.5 Maintainability and Support Requirements

### 3.5.1 Maintenance Requirements

MS1. The source code shall be fully documented and consistent.

- Rationale: This will make it simpler and more straightforward for the current and future developers to understand.
- **Fit Criterion: The source code shall be fully documented using doxygen comments.**

### 3.5.2 Support Requirements

MS2. The program's repository will be public.

- Rationale: A public repository for the program will allow users to raise issues and access the most updated information regarding the program's development.

## 3.6 Security Requirements

N/A

## 3.7 Cultural and Political Requirements

CP1. The program instructions or feature descriptions shall not have any culturally offensive or politically sensitive words, phrases, or symbols.

## 3.8 Legal Requirements

LR1. The program shall adhere to and not violate any Canadian or US laws.

### **3.9 Health and Safety Requirements**

N/A

## **4 Project Issues**

### **4.1 Open Issues**

- Issue 1. Currently all the code for the original Abstract Art Generator program is written and contained within one single module composed of 1400 lines of code. This creates an issue for understandability and maintainability. Having poor modularity makes future adjustments and modification difficult. The next developer to try and update or make adjustments to the program may run into trouble comprehending it and thus have a hard time altering the program.

### **4.2 Off-the-Shelf Solutions**

There are many websites and applications which also generate random art similar to what our project does (for example, [randomart.co.uk](http://randomart.co.uk)). However, there are not many open-source random art generators, and it is unlikely we will be using any other sources of code. Most of the adapted code will be coming from the original Abstract Art Generator we are enhancing.

### **4.3 New Problems**

N/A

### **4.4 Tasks**

1. Begin modularizing the code.
2. Add additional layers (allow for more than two generated art layers).
3. Conduct first code review and debugging.
4. Add text overlay feature to the abstract art generator.
5. Add feature that allows user to set preferred background colour.

6. Add more color palette options.
7. Change color palette implementation to allow color palettes of various lengths.
8. Conduct second code review and debugging.
9. Add feature that allows user to change the UI theme.
10. Conduct testing of the code.
11. Conduct third code review and debugging.
12. Conduct an analysis of the program against the functional and non-functional requirements.

For a more detailed overview refer to the project [Gantt Chart](#)

## 4.5 Migration to the New Product

N/A. The new product will be a standalone version of the original Abstract Art Generator.

## 4.6 Risks

There is minimal risk associated with this project since it is an independent program that does not rely on any external resources. The one main risk is that since the code was not well managed originally, it may be difficult to understand and work with. This could cause delays and difficulty modifying the program.

## 4.7 Costs

There is no monetary cost related to this project. This is because we are working on an open sourced project and the tools we are using are all free of cost. The only cost associated with this project is the time and effort our team will be spending on brainstorming, development, and testing, which is roughly 5-8 hours weekly for each team member.

## 4.8 User Documentation and Training

The program will include a help tab that the user can select to access the instructions. The GUI will be kept clean and simple so the user should not have any issue navigating the program menu.

## 4.9 Waiting Room

Possible future features to be added:

1. Create a more appealing GUI (i.e. better choice of fonts, more attractive front-end, etc.).
2. Add layer effects like distortion and noise.
3. Add a feature that allows the program to design and generate 3-dimensional abstract art pieces.
4. Change the overlay/border section to have a scrolling dialogue box, which will allow more overlay and border options to be displayed.
5. Add an option to allow the user to change the size of the UI text.

## 4.10 Ideas for Solutions

Issue 1. A restructuring of the code may help in modularizing it to allow for better understandability, maintainability, and modification.



## 5 Appendix

N/A

### 5.1 Symbolic Parameters

MAX\_PARAM\_TIME: 1 sec

MAX\_GENERATION\_TIME: 10 sec