Software Requirements Specification

Baltariu Ionuț-Alexandru
Beldiman Vladislav
Nistor Paula-Alina
Rusu Iulian

"Gheorghe Asachi" Technical University of Iași
Faculty of Automatic Control and Computer Engineering

Contents

1 Introduction

1.1 Purpose

The purpose of this document is to give a detailed description of the requirements for the SeeSharper application. It will explain the purpose and features of the program, the interfaces of the program and what the program does.

1.2 Scope

SeeSharper is a simple program that people can use for creating drawings and minimally editing images. The application provides basic functionality for drawing and painting like different types of brushes, brush sizes and a color palette.
SeeSharper also provides shaped stencils and line tools, making the application useful for diagram development. After the drawing is ready, the user has the possibility to save the image in a specific format.
SeeSharper develops an efficiently easy to use interface. This application wants to be a simplified version of Microsoft Paint and Paint.Net.

1.3 Overview

The next chapter, the Overall Description section of this document, gives an overview of the functionality of the application. It describes expected user characteristics and general constraints.
The third chapter, Specific Requirements section of this document, describes in technical terms the details of the functionality of the application and describes the graphical user interface.

2 Overall Description

2.1 Product Perspective

2.2 Product Functions

2.3 User Characteristics

2.4 General Constraints

3 Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

This section describes the graphical user interface (GUI) features and constraints.

1. The main window of the application should have a large canvas, initially with a white background.

2. A drop-down menu with options will be located in the top left corner (Open, Save, About, Exit etc.).

3. To the right of this menu, a toolbar that extends across the entire top side of the window will display all the tools used for drawing. The tools menu will provide several types of tools, grouped logically (Brush Types, Colors, Brush Sizes etc.).

4. The drawing color must offer at least 5 different colors, should ideally offer the possibility to specify any color in the RGB spectrum. Optionally, other color spaces may be selectable.

5. Clicking on the Open/Save buttons will open a separate dialog window where the user will be able to open or save their drawing.

6. Clicking on the About button will open a separate message box with a description of the application.

3.1.2 Software Interfaces

The application must provide a way to communicate with the underlying Operating System's API to delegate the saving/loading of drawings as image files. There are no other software interface constraints as the application is self-contained.

3.2 Functional Requirements

3.3 Performance Requirements

This section covers the requirements that concern the performance of the application in response to user interaction. It is aimed to describe general use case scenarios, as well as time performance constraints.

1. Usage of the toolbar menu.
   The graphical interface must provide a menu that must be easily accessed at the top of the main application window in one click.

2. Usage of drawing tools.
   All drawing tools must be easily selectable from a unified menu of drawing tools. The user should have the option to change the color, size and other parameters of a selected tool easily.

3. The drawing process.
   Drawing must feel smooth and be responsive to user input. The application may provide a way to undo/redo changes from history, without noticeably slowing down performance.

4. File system interaction.
   All drawings must have the ability to be saved on the disk in a specific image format (`png, jpeg, bmp`). The application may provide a way to load images into the canvas and draw on them. The process of saving and loading should not take longer than doing so in other applications that interact with the file system.

5. General timing constraints.
   Any user input should be processed without a noticeable delay. In the event that a more complex computation is required, the user must be notified and the graphical interface must remain responsive for the whole duration. Additionally, the interface may provide status messages to notify the user.

3.4 Design Constraints

This section describes the limitations imposed on the application by software or hardware characteristics of the environment.

1. Disk space usage.
   The whole application must not occupy more than 50 MB of disk storage. Ideally, the application should fit into 25 MB.

2. Memory usage.
   The application's RAM usage must not exceed 50 MB. Ideally, the application should not use more than 25 MB of RAM.

3.5 Attributes

This section describes different software system attributes, metrics and requirements for them.

1. System reliability
   Reliability refers to the system's capacity to correctly respond to user input. The application must correctly load, save and draw images 100% of the time.

2. Maintainability.
   The application should be easily maintainable and extendable. The code should allow easy testing and should be open for future extensions and new features.