Software Requirements Specification

for

Embroid-It

Version 1.5

Prepared by Nathan Owens
Chris Parks
Trae Rawls

EWU Senior Capstone 2016

March 8, 2016

Table of Contents

Revision History

Name	Date	Reason For Changes	Version
Nathan Owens	2/8/16	Created SRS rough draft	1.0 – 1.1
Chris Park	2/21/16	Draft 1.1 revision	1.2
Nathan Owens	2/25/16	Draft 1.2 revised – added use cases	1.3
Chris Park	2/27/16	Draft 1.3 revision	1.4
Nathan Owens	3/7/16	Final revision for Spring Quarter Submission – changed formatting	1.5

1. Introduction

1.1 Purpose

Embroid-It is an interactive application that allows the user to create and edit embroidery project files, including import/export functionality for use with the *Brother* brand PES file format.

1.2 Document Conventions

In each section of this document, requirements and specifications are listed from highest-priority to lowest-priority. Unless otherwise stated, it is assumed that higher-level requirements are inherited by lower-level detailed requirements.

1.3 Intended Audience and Reading Suggestions

This document is intended for use and reference by the developers and stakeholders involved in creating the *Embroid-It* software application. Section 1 identifies the scope of this document, the conventions followed during documentation, and the purpose of the software. Section 2 gives an overall description of the functionality of the software, as well as user characteristics. Section 3 defines all external interfaces associated with the *Embroid-It* application. Section 4 is intended for use primarily by the developers of *Embroid-It.* and details the use cases and system features of the software. Section 5 will identify and describe any other nonfunctional requirements. Finally, section 6 will include any other software requirements not previously mentioned. Appendices are included at the end of this document, containing additional resources.

1.4 Product Scope

Both home and commercial use embroidery machines use embroidery stitch files that provide the machine with necessary information for proper design output. An input file typically provides information such as stitch color, placement, length, and type. There are several different file formats that embroidery machines accept as proper input, depending on the machine make and model. The *Embroid-It* application will allow the user to open, create, edit, and save embroidery project files, as well as import/export to and from PES file format. This software will to be open-source. The goal of the *Embroid-It* application is to provide a free, easy-to-use environment for the user to edit and create embroidery designs.

1.5 References

IEEE. *IEEE Std 830-1998 IEEE Recommended Practice for Software Requirements Specifications.*

IEEE Computer Society, 1998.

2. Overall Description

2.1 Product Perspective

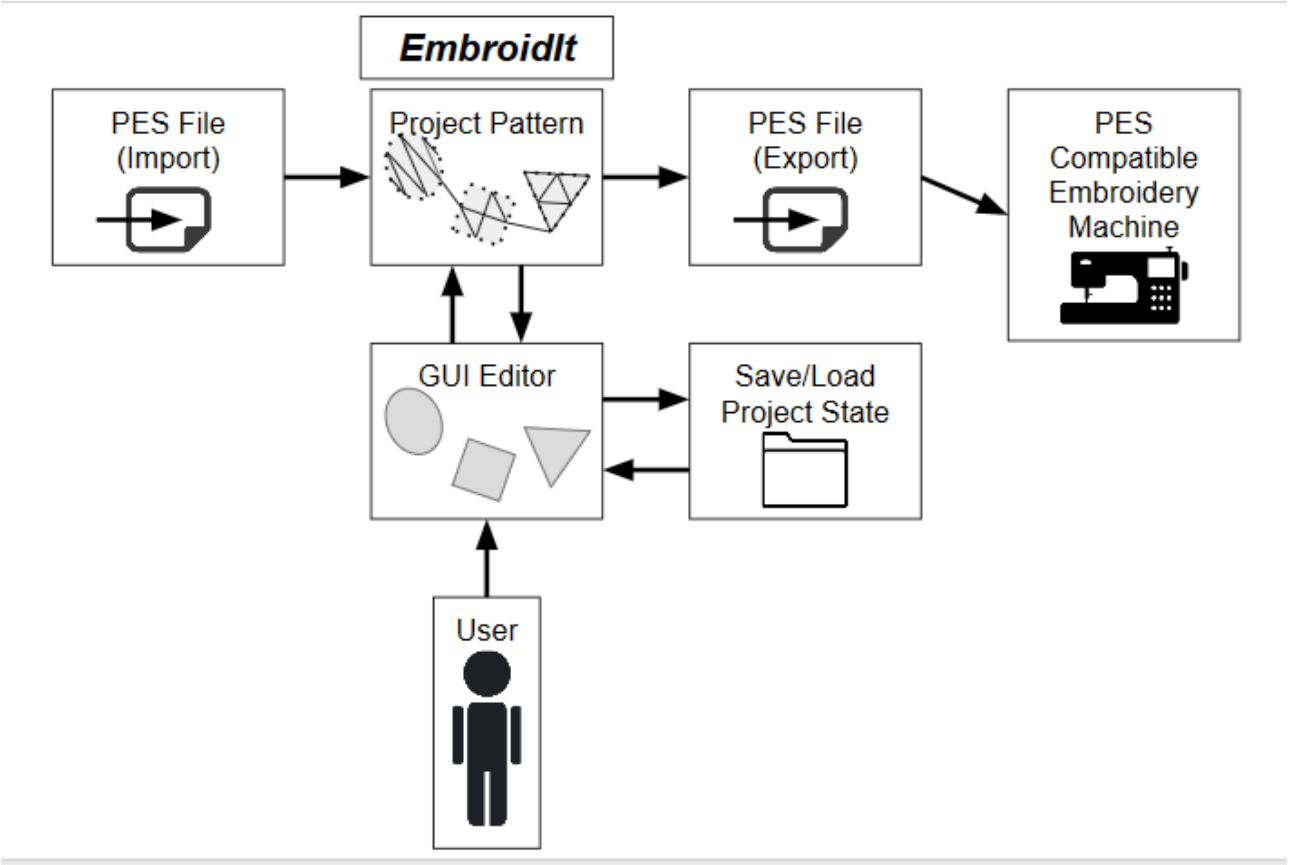

Figure 1 – System Environment

The *Embroid-It* software is designed as a stand-alone desktop application that assists the user in creating and editing embroidery designs. Aside from working with its own project files, the application will import and export PES formatted files. An EEI (embroidery editing interface) will be presented, allowing the user to see a visual rendering of the current design, and manipulate different aspects of the design, such as stitch color and length. The EEI shall also present the user with tools to select all or portions of the project, draw basic shapes, overlay stitches, and other features described throughout this document.

2.2 Product Functions

At minimum, the functions that *Embroid-It* will perform are as follows:

- Create a new embroidery project.
- Save the current project, or load a different one.
- Create and edit basic shapes (Ellipse, Rectangle, Line).

- Fill shape areas with stitches.
- Edit fill stitch length, angle, and color.
- Import/Export PES format files for editing.
- Provide a visual representation of the project.
- Provide basic operational information through help menus and tool tips.

See Section 4 of this document for a more detailed breakdown of program operations and user interaction.

The software will preserve all information contained in an embroidery project file that is opened and saved by the application, except where information is explicitly deleted or changed by the user prior to saving. Information that must be preserved in the project file includes, but is not limited to: stitch colors, length, fill type, and shape layout.

2.3 User Classes and Characteristics

Users of this product are assumed to have a basic knowledge and background of the embroidery process, but are not expected to learn any special set of skills or commands beyond the scope of the EEI. Since the purpose of this product is to provide an easy-to-use environment, the application will be intuitive and provide the user with necessary information to perform basic editing operations. The user is assumed to be literate in their current operating environment, having a basic knowledge of the file systems and desktop applications involved in operation. This software is intended only as a stand-alone application to assist the user in editing an embroidery design and will not provide the user with the knowledge to use and/or run an embroidery machine.

2.4 Operating Environment

Embroid-It will at minimum operate on the following systems and environments:

- Windows

The product will be a stand-alone desktop application. Once installed, it will not require network access.

2.5 Design and Implementation Constraints

The initial release of *Embroid-It* will import/export only PES formatted embroidery files. If the intent is to use a PES file created by the software, the user will be limited to that functionality which is supported by the software, and only those models of embroidery machines that are compatible with the PES file format.

2.6 User Documentation

Embroid-It will contain a help/getting started section, accessed through a drop down menu within the user interface. It will provide information about the commands and interactions available to the user.

2.7 Assumptions and Dependencies

No assumptions or dependencies have been identified at this time.

3. Specific Requirements

3.1 External Interface Requirements

3.1.1 User Interfaces

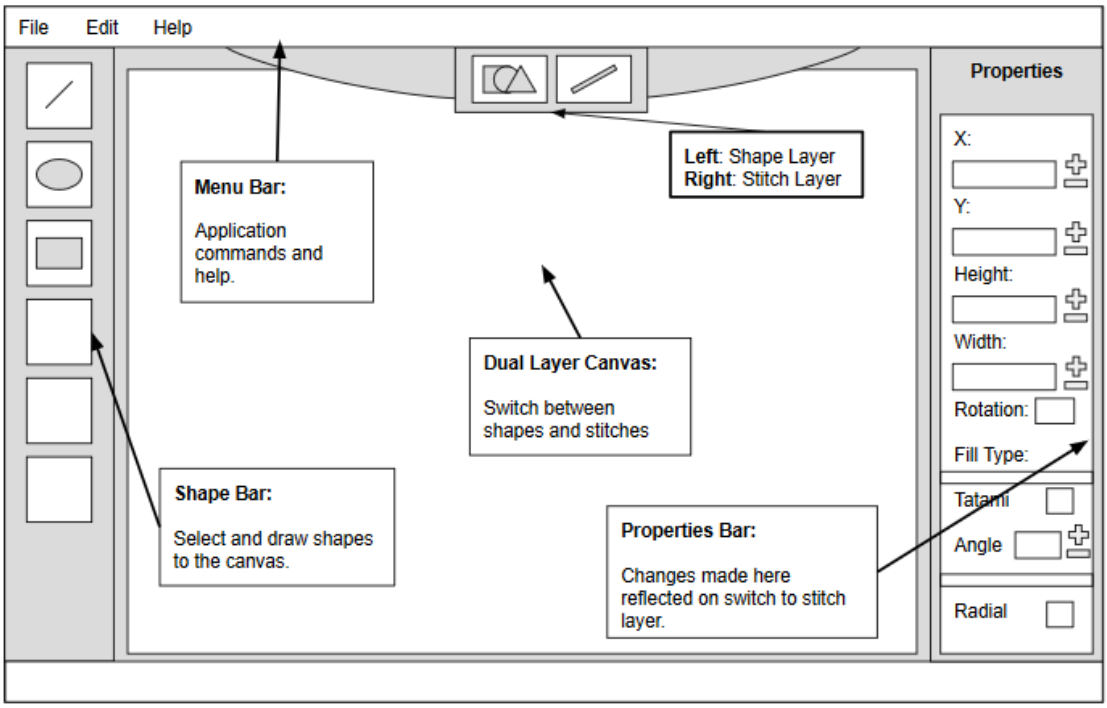

Figure 2 – EEI Layout

When run, *Embroid-It* will present an EEI (Embroidery Editing Interface) for user interaction. This layout for the interface will contain the following:

- Menu bar:
  - File: New, Open, Save, Save As, Import, Export
  - Edit: Undo, Redo
  - Help: About, Getting Started, Developer Documentation.

- Shape creation bar:
  - Ellipse, Rectangle, Line

- Shape properties bar:
  - Position, Height, Width, Rotation, Fill Type

- Dual layer canvas editor that renders shapes and stitches on separate visual layers.

3.1.2 Hardware Interfaces

None

3.1.3 Software Interfaces

Embroid-It will utilize:

- System calls to the file management system in order to:
  - Save and/or Open a project.
  - Import and export PES files.

- JavaFX shape and media packages for use in:
  - Shape definition and manipulation.

- The ewu.embroidit package library for use in:
  - GUI functionality.
  - Pattern storage
  - Shape and stitch manipulation
  - Importing and exporting PES files.

3.1.4 Communications Interfaces

None

3.2 EEI Functional Requirements

This sections describes the functional requirements for the Embroidery Editing Interface (EEI).

3.2.1 EEI Internal Data

The EEI shall store the past sequence of edit actions, in order to provide a multi-level undo. The EEI will allow the user to undo an entire sequence of edits, back to the last saved state of the design.

3.2.2 User Commands

The EEI will make the following commands available to the user. These commands will be provided by whatever interface mechanisms are appropriate (e.g. pull-down menus, toolbar buttons, keyboard shortcuts, etc.). If usability tests suggest, a combination of such mechanisms may be used. The following interactions will be made available via the EEI:

- **New:** initializes the canvas. This command must give the user a warning if the previous design has not been saved since the last edit, and give the user an opportunity to cancel this operation.

- **Open:** Allows the user to browse the file directory to select a serialized project file to open for editing. This command must give the user a warning if the previous design has not been saved since last edit, and give the user an opportunity to cancel this operation.

- **Save:** Saves the state of the current design to the most recent serialized project file associated with this design. If the current project has been initialized and not yet saved to a file, this command will function the same as the 'Save As' command.

- **Save As:** This command will prompt the user for a file name and directory to save the current project to (as a serialized project file). This command will validate the file name and directory. If a file name exists, user will be prompted to confirm whether to overwrite the existing file.

- **Import PES:** Allows the user to browse the file directory to select a PES file to import for editing, and will show the user an error message if the file is in the wrong format or unreadable. This command must also warn the user if the previous design has not been saved since last edit, and give the user an opportunity to cancel this operation.

- **Export PES:** This command will prompt the user for a file name and directory to export the project to (in a PES formatted file).

- **Draw shape:** Initializes a new shape object. The user will be provided with the tools to draw the following shapes; ellipse, rectangle, and line. This command will initialize a shape object based on the shape the user has selected, and allow the user to draw the shape to the canvas using the drag and drop features of the mouse.

- **Edit shape:** This allows the user to manipulate a selected shape in the current design. When a shape is selected, its properties will be visible in the properties panel to the right side of the EEI. The panel will provide location, width, height, fill type, and rotation adjustments.

- **Undo:** Allows the user to undo a previous edit action.

- **Redo:** Allows the user to 'undo' an undo command. This command should be enabled immediately after an undo command has been used, and disabled immediately after any other editing actions are performed.

- **Exit:** allows the user to exit the application. The user will be prompted to save the current project, if it hasn't been saved already.

4. Other Nonfunctional Requirements

4.1 Safety Requirements

No specific safety requirements have been identified

4.2 Security Requirements

There are no specific security requirements

4.3 Software Quality Attributes

One of the big focal points of this application project is usability. A user should be provided with an editing interface that is simple to use and easy to understand. Therefore, the EEI will be designed with that usability focus in mind. The tools contained in the interface will have simple icons that communicate function intuitively to the user. The EEI layout will avoid clutter and aim to present a clean organization that does not overwhelm the user.

Appendix A: Glossary

Canvas – Draw-able surface within the EEI on which an embroidery design is created.

EEI (Embroidery Editing Interface) – User interface allowing for visual (non-command line) interaction between the user and the *Embroid-It* software.

PES – Computer aided manufacturing file that contains instructions for creating embroideries with a sewing/embroidery machine; contains codes such as “stop,” “jump,” and “trim;” also includes colors for the design from the supported *Brother* color palette. Mainly used by *Brother* embroidery machines, but also compatible with *Babylock* and *Bernina* machines.

Appendix B: Analysis Models

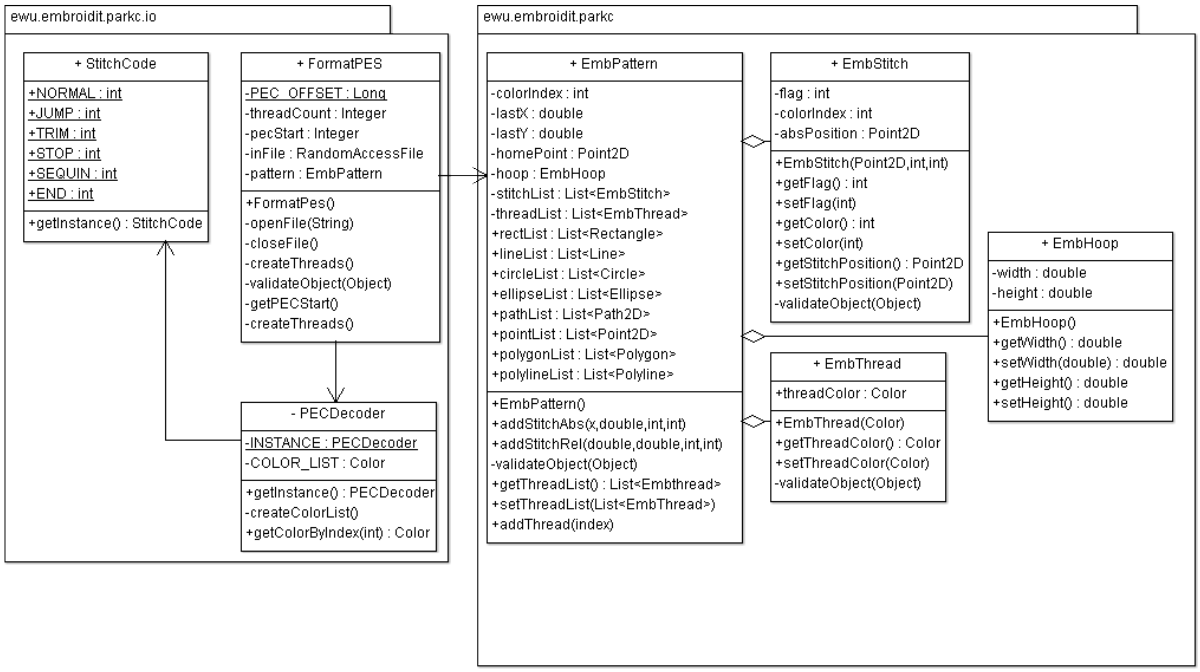

Figure 3 – I/O & Pattern UML diagram

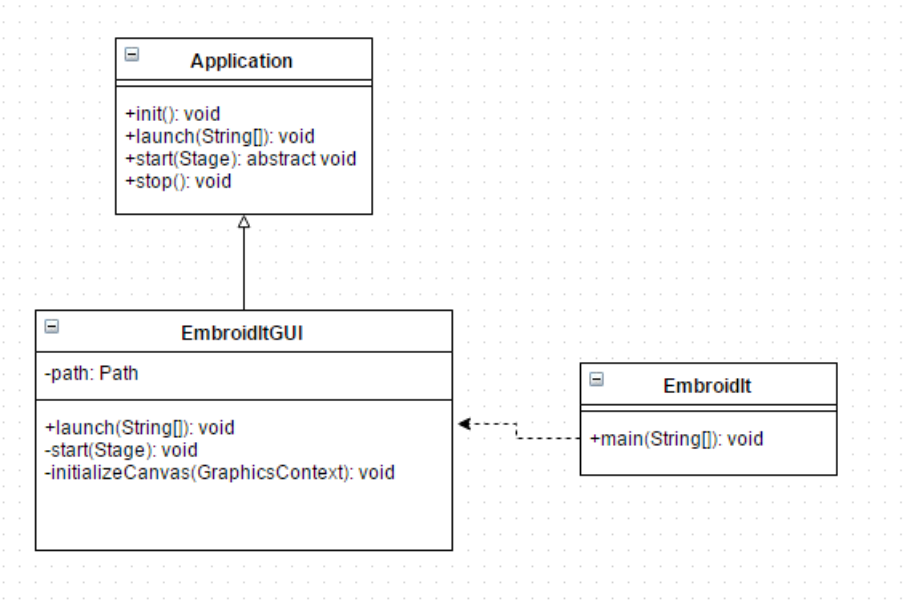

Figure 4 – EEI UML diagram

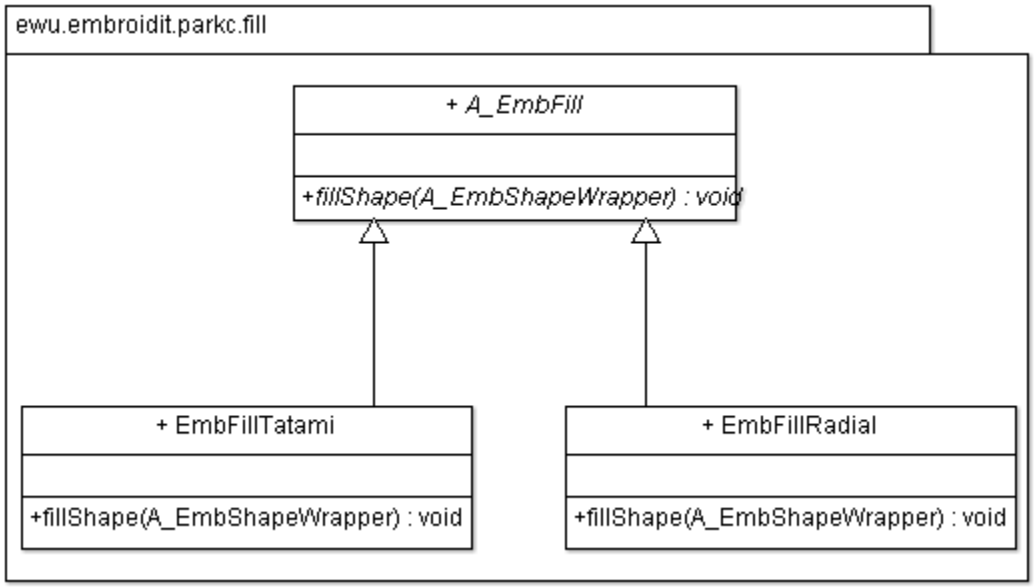

Figure 5 – Fill UML

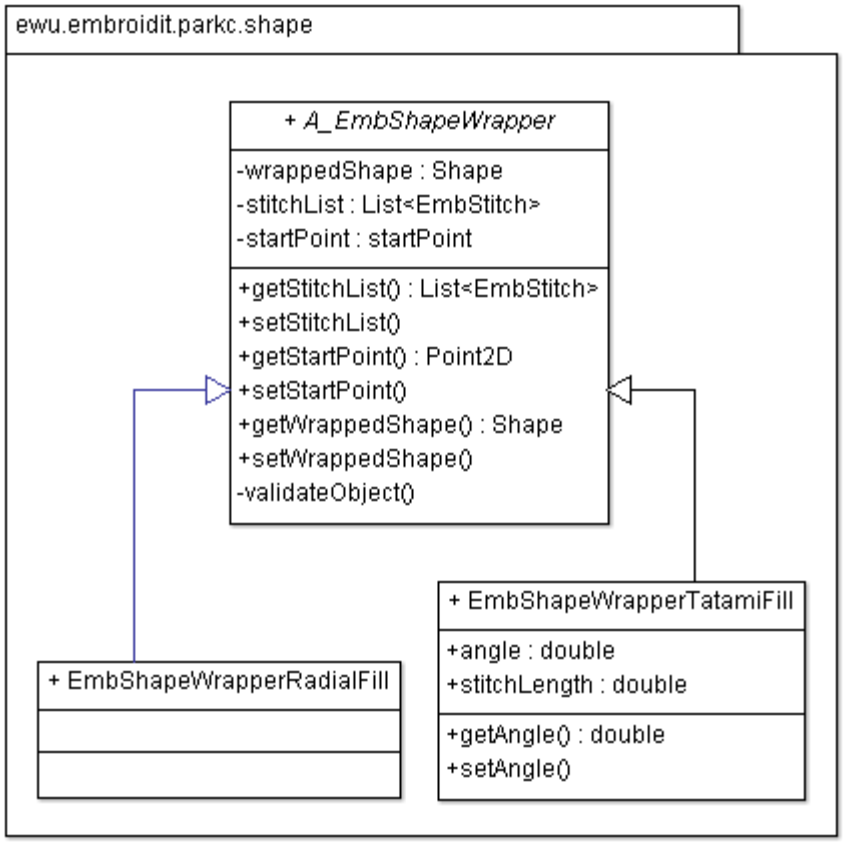

Figure 6 – Shape UML