Software Requirements Specification (SRS)
Graphics Editor Software Package

- **Introduction**
  - **Purpose**

    The purpose of this document is to provide a detailed description of the Graphics Editor Software Package, outlining the functional and nonfunctional requirements, design constraints, and interfaces of the system. This SRS aims to be used by developers, testers, and stakeholders to ensure that the software meets the intended needs and expectations.

  - **Scope**

    The Graphics Editor Software Package(GraphyX) is designed to provide a comprehensive suite of tools for creating, editing, and managing digital graphics. The software will support various image formats, offer essential editing features, and provide a userfriendly interface for both novice and experienced users.

  - **Definitions, Acronyms, and Abbreviations**
    - **Graphics Editor**: The software application designed for creating and editing digital images.
    - **UI**: User Interface
    - **UX**: User Experience
    - **API**: Application Programming Interface

  - **References**
    - [ISO/IEC 25010:2011 Systems and Software Engineering – Systems and Software
    - Quality Requirements and Evaluation (SQuaRE) – System and Software Quality Models]
    - [User Interface Guidelines]

  - **Overview of Developer's Responsibilites**
    - **Development**: Design, implement, and test the Graphics Editor Software Package according to the requirements outlined in this document. Ensure that the software meets all specified functional and non-functional requirements, and adheres to best practices in software development.

    - **Installation**: Provide a straightforward installation process for the end-users. Develop installation packages for various operating systems (Windows, macOS, Linux) and ensure that the installation process is smooth, with minimal user intervention required.

    - **Training**: Create comprehensive user documentation and training materials to assist users in understanding and utilizing the software effectively. This includes user manuals, online help systems, and tutorials. Optionally, offer training sessions or webinars to guide users through the software's features.

    - **Maintenance**: Offer ongoing maintenance and support for the software. This includes addressing bug reports, releasing updates to fix issues or improve functionality, and ensuring compatibility with new operating system versions. Provide a support channel for users to report problems and request assistance.

        - **Quality Assurance**: Implement robust testing procedures to ensure the software is reliable, secure, and performs well under expected usage conditions. Conduct regular testing and quality assurance checks to maintain high standards of software performance.

        - **Documentation**: Maintain clear and thorough documentation for both users and developers. This includes technical documentation for future development and maintenance, as well as end-user documentation to support ease of use.

- **Overall Description**
    - **Product Perspective**
        - The Graphics Editor Software Package will be a standalone application compatible with major operating systems (Windows, macOS, Linux).
        - It will interact with the user through a graphical user interface and will be capable of exporting images in multiple formats.

    - **Product Functions**
        - **Basic Graphical Shapes**: Tools to create lines, polygons, ellipses, and other fundamental shapes.
        - **Text Editing**: Create and edit text elements within the graphics.
        - **Graphic Selection and Manipulation**: Select, rotate, reposition, and manipulate existing graphics, including adjusting shape, size, location, fill color, line color, line width, and line style.
        - **Clipboard Management**: Clipboard with 10 slots for copying and pasting objects.
        - **Object Management**: Functions for copying, pasting, deleting, and highlighting selected objects.
        - **File Management**: Save graphics to disk with userspecified names and load previously created files.
        - **Image Import**: Import bitmap images into the interface.
        - Zoom and Pan: Features to zoom into a selected rectangular area, fit graphics to the screen, and pan across the workspace.
        - **Grouping**: Grouping functionality for handling complex drawings, where groups can contain other groups and be managed as independent entities.

    - **User Classes and Characteristics**
        - **Novice Users**: Individuals with minimal experience in graphic design, requiring simple and intuitive tools.
        - **Intermediate Users**: Users with some experience, needing more advanced features and customization options.
        - **Professional Users**: Experienced designers requiring extensive tools, advanced editing features, and high performance.

    - **Operating Environment**
        - **Hardware**: Minimum system requirements include a modern multicore processor, 4GB RAM, and a graphics card with at least 1GB VRAM.
        - **Software**: Compatible with Windows 10 and later, macOS 10.14 and later, Linux distributions with GTK3 or later.

    - **Design and Implementation Constraints**
        - **Performance**: The application must perform efficiently with images up to 50MP in size.

- **Security**: Must ensure user data is protected, especially when handling file imports and exports.

- **Assumptions and Dependencies**
  - The software assumes that the user has basic familiarity with graphic design concepts.
  - The software depends on thirdparty libraries for image processing and rendering.

- **Specific Requirements**
  - **Functional Requirements**
    - **Creating Basic Graphical Shapes**

      Introduction: The system shall provide tools for creating lines, polygons, ellipses, and other basic shapes.

      Input: Shape Type, Dimensions(eg., Height, Width, Radius, etc.), position(Coordinates on the Canvas) and Optional Attributes (eg., Fill Color, Line Color, etc.)

      Processing: The user selects the desired shape tool, specifies the dimensions and position for the shape and the system renders the shape on the canvas according to the specified attributes.

      Outputs: A graphical shape is created and displayed on the canvas. It can be interacted with for gurther manipulation.

    - **Creating and Editing Text**

      Introduction: The system shall allow users to create and edit text elements within the graphics. Users shall be able to modify font, size, color, and other text properties.

      Inputs: Text Contents, Font type and Size, Text Color, Position(Coordinates on the Canvas), Optional Text Effects(eg., Bold, Italics)

      Processing: The user selects the text tool and inputs text content. Then they customizes the font, size, and other text properties. The system renders the text on the canvas at the specified position with the given attributes.

      Outputs: Text is added and displayed on the canvas. Users can see and interact with the text for further editing.

    - **Selecting and Manipulating Graphics**

      Introduction: The Selecting and Manipulating Graphics function allows users to interact with existing graphics for adjustments and modifications.

      Inputs: Selection of Graphic(via mouse Click), Manipulation commands(eg., rotate, resize, reposition), Attribute Modification(eg., fill color, line color, line width, line style, etc.)

      Processing: The user clicks on a graphic to select it. The system highlights the selected graphic. The user applies manipulation commands or attribute changes. The system updates the graphic on the canvas according to the modifications.

Outputs: The selected graphic is highlighted. The graphic is updated with the applied changes.

- **Copying, Pasting, and Deleting Objects**

  Introduction: The Copying, Pasting, and Deleting Objects function enables users to duplicate or remove graphics on the canvas.

  Inputs: Selected object(s) to copy or delete, Destination location for pasting (for copy operation), Clipboard slot (for storing copied objects).

  Processing: The user selects an object and copies it to the clipboard. They pastes the object to a new location or deletes the selected object. The system updates the canvas based on the action performed.

  Outputs: The object is copied and displayed in the new location or removed from the canvas.

- **Saving and Loading Files**

  Introduction: The Saving and Loading Files function allows users to save their work and reload previously saved graphics.

  Inputs: File Name(For Saving), File Path(For Loading), Graphics Data(For saving).

  Processing: The user specifies a name and location to save the file or selects a previously saved file to load. The system saves or loads the graphics data to/from the specified file path.

  Outputs: The graphics are saved to disk or loaded from disk, and the canvas is updated accordingly.

- **Importing Bitmap Images**

  Introduction: The system shall support importing bitmap images (e.g., PNG, JPEG) into the graphics editor interface.

  Inputs: Bitmap image file (e.g., PNG, JPEG), Import settings (e.g., size adjustments).

  Processing: The user selects a bitmap image file for import. The system processes the image file and integrates it into the graphics editor interface.

  Outputs: The bitmap image is displayed on the canvas and is available for further manipulation.

- **Zooming and Panning**

  Introduction: The Zooming and Panning function allows users to adjust the view of the graphics workspace.

  Inputs: Selection of a rectangular area (for zooming) or Pan commands (e.g., drag or arrow keys)

Processing: The user selects a rectangular area to zoom into or activates the fit screen function. The system adjusts the zoom level and view to accommodate the selection or fit the screen. The user pans across the workspace as needed.

Outputs: The canvas view is updated to reflect the zoomed area or fit screen adjustments. Users can navigate the workspace by panning.

- **Grouping Functionality**

  Introduction: The Grouping Functionality allows users to group objects together for easier management of complex drawings.

  Inputs: Selection of objects to group, Grouping commands (e.g., create group, add to group).

  Processing: The user selects multiple objects and creates a group or adds objects to an existing group. The system manages the group as a single entity for operations like delete, copy, and select.

  Outputs: The selected objects are grouped together and can be manipulated as a single entity.

- **Non-Functional Requirements**
  - **Performance**

    The application shall respond to user actions within 1 second for most operations.
    The application shall handle images up to 50MP without significant performance degradation.

  - **Usability**

    The user interface shall be intuitive and easy to navigate.
    The application shall provide a comprehensive help section and user manual.

  - **Reliability**

    The application shall not crash under normal usage conditions.
    The application shall provide autosave functionality to prevent data loss.

  - **Security**

    The application shall ensure that files are not corrupted during import/export operations.
    The application shall include measures to prevent unauthorized access to user data.

  - **Portability**

    The application shall be compatible with Windows, macOS, and Linux operating systems.

  - **Maintainability**

    The codebase shall be well documented and modular to facilitate future updates and maintenance.