**Number ID:** UC-001
**Application:** Shape Drawing
**Application Use Case Name:** Save and Load Shapes
**Use Case Description:** This use case allows users to save all shapes on the canvas to a file and reload them later. The user initiates the save process to store shape data and can load the data back to restore the shapes on the canvas.

**Primary Actor:** User

**Precondition:** The application runs, and the canvas contains at least one shape. The file system must be accessible for saving and loading.

**Trigger:** The user selects the "Save" or "Load" option from the menu.

## Basic Flow

### Save Shapes:

1. The user selects the "Save" option.
2. The system opens a file (`shapes.txt`) in write mode.
3. The system iterates through all shapes on the canvas and retrieves their string representations.
4. The system writes each shape's data to the file.
5. The file is closed, and the system confirms the save action.

### Load Shapes:

1. The user selects the "Load" option.
2. The system opens the file (`shapes.txt`) in read mode.
3. The system reads each line of the file and parses the data to recreate shapes.
4. The recreated shapes are added to the canvas.
5. The file is closed, and the system confirms the load action.

## Alternate Flows

### File Not Found:

- The system displays an error message if the file does not exist during the load process.

### Corrupted File:

- If the file contains malformed data, the system skips the corrupted lines and only loads valid shapes.

Empty Canvas (Save): If the canvas is empty during the save process, the system creates an empty file.

System Error: If the system encounters an error while reading or writing, an error message is displayed, and the process is halted.

**Number ID:** UC-002
**Use Case Name:** Add New Shape
**Use Case Description:** This use case allows users to add a new shape to the canvas. The user selects a shape type (e.g., Circle, Rectangle, Line), specifies its properties, and places it on the canvas. The system validates the input and updates the canvas to include the new shape. Additionally, a shape is saved to the shapes.txt file every time it is created. When the program loads, shapes are loaded from the `shapes.txt` file into the shape container.

**Primary Actor:** User

**Precondition:** The application is running, the user has accessed the drawing canvas, and previously saved shapes from `shapes.txt` have been loaded into the shape container.

**Trigger:** The user selects the "Add Shape" option and chooses a shape type.

## Basic Flow

1. The user navigates to the shape selection menu.
2. The user selects a shape type (e.g., Circle, Line).
3. The system prompts the user to specify shape properties (e.g., dimensions, color).
4. The user provides the required properties and confirms the addition.
5. The system validates the input.
6. The system adds the shape to the canvas and updates the display.
7. The system saves the new shape to the `shapes.txt` file.
8. The user sees the new shape on the canvas.

## Alternate Flows

### Invalid Input:

- If the user provides invalid data (e.g., negative dimensions), the system displays an error message and prompts for correction.