REVISION HISTORY

|  |  |  |  |
| --- | --- | --- | --- |
| **Date** | **Version** | **Description** | **Author** |
| 22/11/2018 | 0.1 | Edited Software Architecture Overview and Software Design Description | Arman Hasanzade,  Güneş Büyükgönenç |
| 23/11/2018 | 0.2 | Edited the Software Design Description 3.2 | Arman Hasanzade |
| 23/11/2018 | 0.3 | Edited the Software Design Description 3.3 | Arman Hasanzade |
| 23/11/2018 | 0.4 | Edited Project References 1.1.1 | Kaan Kırlı |
| 23/11/2018 | 1.0 | Edited the Table of Contents and Introduction and finalized the paper | Arman Hasanzade, |
| 27/11/2018 | 1.1 | Final revision & edits for the paper | Yazan Shehab, Kaan Yıldız |

**TABLE OF CONTENTS**

[**1 Introduction 3**](#_Toc531366405)

[**1.1 References 3**](#_Toc531366406)

[***1.1.1* Project References 3**](#_Toc531366407)

[**2 Software Architecture overview 3**](#_Toc531366408)

[**3 Software design description 3**](#_Toc531366409)

[**3.1 GUI 3**](#_Toc531366410)

[**3.1.1 Component interfaces 3**](#_Toc531366411)

[**3.1.2 Component design description 3**](#_Toc531366412)

[**3.1.3 Workflows and algorithms 4**](#_Toc531366413)

[**3.1.4 Software requirements mapping 4**](#_Toc531366414)

[**3.2 Operations 4**](#_Toc531366415)

[**3.2.1 Component interfaces 4**](#_Toc531366416)

[**3.2.2 Component design description 5**](#_Toc531366417)

[**3.2.3 Workflows and algorithms 5**](#_Toc531366418)

[**3.2.4 Software requirements mapping 5**](#_Toc531366419)

[**3.3 Database 6**](#_Toc531366420)

[**3.3.1 Component interfaces 6**](#_Toc531366421)

[**3.3.2 Component design description 6**](#_Toc531366422)

[**3.3.3 Workflows and algorithms 6**](#_Toc531366423)

[**3.3.4 Software requirements mapping 6**](#_Toc531366424)

[**4 COTS Identification 6**](#_Toc531366425)

# Introduction

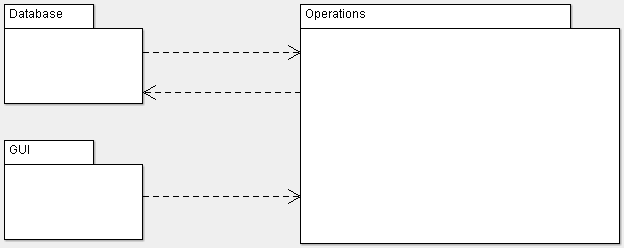
This document describes the design of the SUP software system.

## References

### Project References

|  |  |  |
| --- | --- | --- |
| # | Document Identifier | Document Title |
| [SRS] | SUP-SRS-1 | SUP Software Requirements Specifications |
| [R1] | MSPain3D | <https://www.microsoft.com/tr-tr/p/paint-3d/9nblggh5fv99#activetab=pivot:overviewtab> |
| [R2] | Gimp | <https://www.gimp.org/> |

# Software Architecture overview



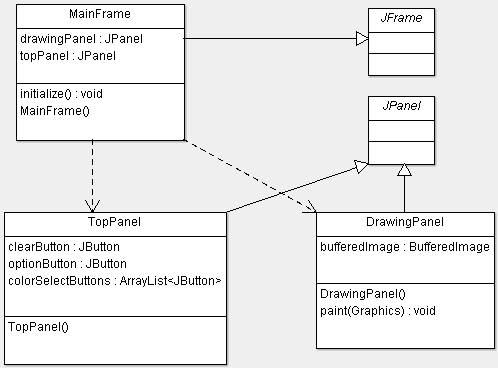
# Software design description

## GUI

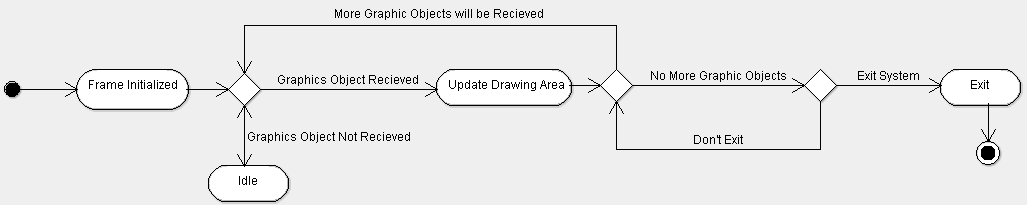
### Component interfaces

* initialize() : void -> Used for initializing frame with specified values like size, resizability, and etc.
* paint(Graphics) : void -> Takes Graphics object as input and changes the drawing panel accordingly.
* setOptionPanel() : void -> Sets the panel on top of the drawing panel where the user will be able to pick brush colors, sizes and etc.
* setDrawingPanel() : void -> Sets the center panel which will be used for displaying drawings
* This component extends JFrame/JPanel/JButton/JLabel/JTextField classes

### Component design description



### Workflows and algorithms



### Software requirements mapping

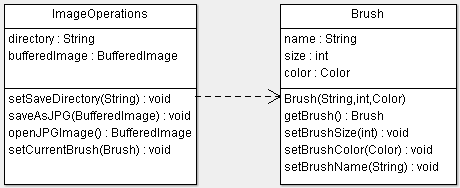
* **SRS-SUP-001**
* **SRS-SUP-009**
* **SRS-SUP-010**

## Operations

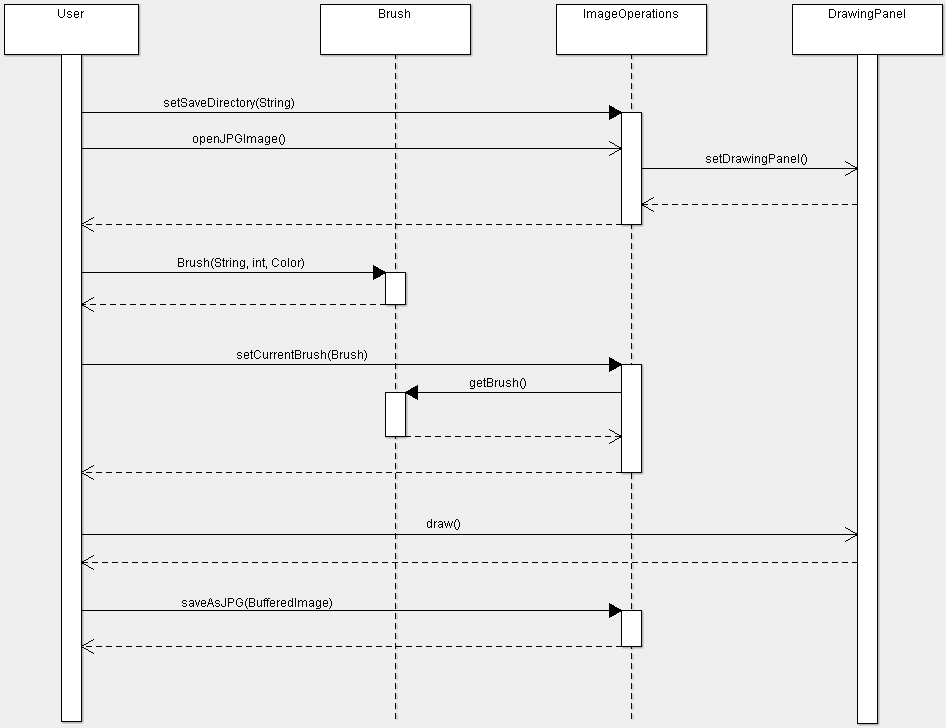
### Component interfaces

* setSaveDirectory(String) : void -> Sets the directory in which the user’s files will be accessed.
* saveAsJPG(BufferedImage) : void -> Saves the drawing in drawing area to the user’s computer.
* openJPGImage() : BufferedImage -> Opens an existing JPG file from user’s computer and displays it on the drawing area.
* getBrush() : Brush -> Returns the selected brush from the Brush list.
* createBrush(String, int, Color) -> Creates a custom brush with parameters name, size, and color.
* setCurrentBrush(Brush) : void -> Sets the current brush with its attributes which will affect the drawing area.

### Component design description



### Workflows and algorithms



### Software requirements mapping

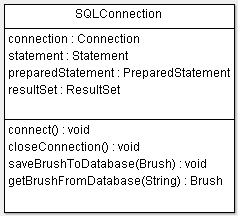
* **SRS-SUP-002**
* **SRS-SUP-003**
* **SRS-SUP-005**

## Database

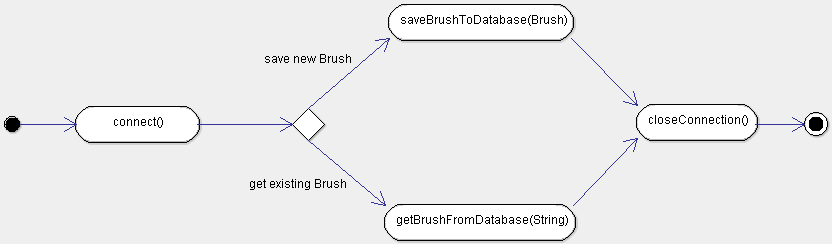
### Component interfaces

* connect() : void -> creates a SQL Database connection to our Database URL
* closeConnection() : void -> safely closes the Database connection
* shaveBrushDatabase(Brush) : void -> saves the brush to Database
* getBrushFromDatabase(String) : Brush -> Returns the brush with the given name if it exists in the Database

### Component design description



### Workflows and algorithms



### Software requirements mapping

* **SRS-SUP-004**
* **SRS-SUP-006**
* **SRS-SUP-011**

# COTS Identification

**No External libraries will be used.**

**MySQL Workbench will be used for viewing and editing the data to spot any possible errors.**