



## **Department of Computer Science & Information Technology**

---

University of Sargodha

## **Sketch an Image**

Version 3.0

---

# Functional Specifications Document

---

## Sketch an Image

### **Project Supervisor:**

Mr. Israr Ahmed

### **Project Manager:**

Dr. Qaisar Abbas

### **Project Leader:**

Muhammad Arbain

### **Project Team:**

Dawood Ahmad	16-uglc-1007
Muhammad Arbain	16-uglc-1029

### **Submission Date:**

30<sup>th</sup> of December, 2019

---

Project Manager's Signature

## Revision History

Date	Version	Description of Change	Author
October 25, 2019	1.0	Introduction	Arbain Rahat, Dawood Ahmad
November 29, 2019	2.0	Software Requirement Specification	Arbain Rahat, Dawood Ahmad
December 30, 2019	3.0	Functional Specification Document	Arbain Rahat, Dawood Ahmad

## Distribution List

Recipient	Role / Designation	Office	Contact Details
Qaiser Abbas	Assistant Professor	Department of Computer Science & Information Technology	qaiser.abbas@uos.edu.pk

## Definition of Terms, Acronyms and Abbreviations

Term	Description
SAI	Sketch an Image
DFD	Data Flow Diagram
UML	Unified Modelling Language

## Table of Contents

---

<b>1.</b>	<b>Introduction.....</b>	<b>7</b>
	1.1 Purpose of Document.....	7
	1.2 Project Overview .....	7
	1.3 Scope .....	8
	1.4 Objectives.....	8
	1.5 High Level Project Plan.....	9
<b>2.</b>	<b>Overall System Description.....</b>	<b>9</b>
	2.1 User characteristics .....	10
	2.2 Operating environment .....	10
	2.3 System constraints.....	10
<b>3.</b>	<b>External Interface Requirements .....</b>	<b>11</b>
	3.1 Hardware Interfaces .....	11
	3.2 Software Interfaces .....	11
<b>4.</b>	<b>Functional Requirements .....</b>	<b>12</b>
<b>5.</b>	<b>Non-functional Requirements.....</b>	<b>12</b>
	5.1 Performance Requirements.....	12
	5.2 Extensibility.....	12
	5.3 Maintainability.....	13
	5.4 Usability Requirement .....	13
<b>6.</b>	<b>Assumptions and Dependencies.....</b>	<b>13</b>
<b>7.</b>	<b>System Architecture.....</b>	<b>14</b>
	7.1 System Overview .....	14
	7.2 Context Level DFD .....	15
<b>8.</b>	<b>Use Cases.....</b>	<b>16</b>
	8.1 Use Case Diagrams .....	16
	8.2 Use Case Description .....	17
	8.3 UML Diagrams .....	17
<b>9.</b>	<b>Graphical User Interfaces .....</b>	<b>22</b>

<b>10.</b>	<b>References .....</b>	<b>25</b>
------------	-------------------------	-----------

# 1. Introduction

This chapter details the software requirement specifications and functional specification of sketch an image (SAI), which include introduction in Section 1 (purpose of document, project overview, scope). System Description describes about user activities, constraints and system environment in which SAI will work in Section 2. After section 2 external interface requirements are described that include hardware and software interfaces in Section 3.

Section 4 and Section 5 about the functional and non-functional requirements of SAI respectively. Assumptions and dependencies of SAI are written in Section 6. System architecture describes the architecture of app and zero level DFD in Section 7. Section 8 describes the design diagrams of SAI which includes use case and UML diagrams. Graphical User Interface section include pictures (screen shots) of SAI in Section 9. References write at the last of this document in Section 10.

## 1.1 Purpose of Document

The purpose of this document is to present a detailed description and detail design of the application. It will explain the purpose and features of the application and what the system will do. This document will be useful for stakeholders.

## 1.2 Project Overview

The goal of this project is to make an android app that will provide free of cost services to users. SAI will work offline. So, internet connection will not be required for perform task.

Sketch an image will use edge detection phenomena. The process of classifying and placing sharp discontinuities in an image is called the edge detection [1]. SAI will detect the edge boundaries of the image which then will be used to convert the image to digital sketch. Edge detection works by detecting edges and find the differences between brightness of image. SAI works by measuring the intensity of light. SAI is used edge detection; Edge detection will analyze the image by using image segmentation.

Image segmentation is necessary step in image analysis. Segmentation separates an image into its component parts or objects. The level to which the separation is carried depends on the problem being solved. When the objects of interest in an application have been inaccessible the segmentation must stop. Segmentation algorithms for images generally based on the discontinuity and similarity of image intensity values [2].

Discontinuity approach is to partition an image based on abrupt changes in intensity and similarity is based on partitioning an image into regions that are similar according to a set of predefined criteria. There are many different algorithms of edge processing that are being used worldwide which are Sobel, Canny, Prewitt, Roberts and Fuzzy logic [3].

These are all part of AI (robotics vision). The aim of these algorithms is to find the edges of the images by checking at which parts the brightness drastically changes. SAI transforms original image into the edged image of sudden changes of the brightness and uses discontinuities. Edges occur on the boundary between two regions.

SAI may assist those who interested in sketching. This will make learning easier because as a contrast for them, they will have a digital drawing. This will also make people enjoy and entertain people who want to enjoy watching or framing their sketches.

### **1.3 Scope**

#### **In Scope**

- SAI will make sketches of the image based on the edges found in the image
- SAI will save the image
- SAI will access the internal storage
- SAI will be able to use other applications for sharing images
- SAI will be able to make different type of sketches by using different algorithms
- SAI will not made for organization
- User will be able to discard a sketch

#### **Not in Scope**

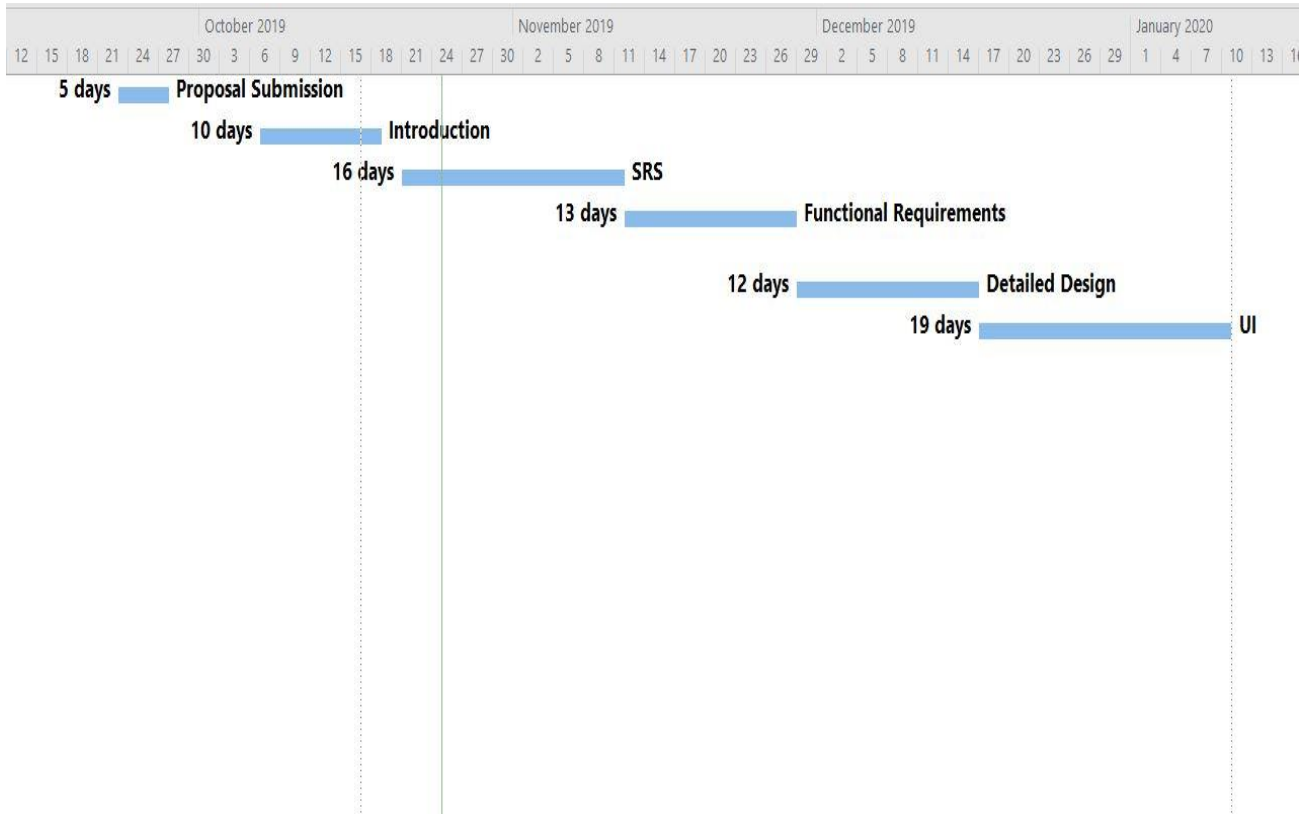
- SAI will not allow the user to crop the pictures
- SAI will not allow the user to edit the pictures
- SAI will not provide step by step guidance to the user
- SAI will not make sketch of blurred images
- SAI will not have any servers
- SAI will not able to share images if there are no sharing apps are available in mobile devices
- SAI will not be able to delete any image from the gallery

### **1.4 Objectives**

- making the complete design of SAI
- A working layout for the user to perform actions
- Adding the capability to access an image from the storage
- Methods for applying algorithms for edge detection
- Adding the capability to make an image file
- capability of saving the image
- Adding capability to share the image with sharing apps
- testing the app



## 1.5 High Level Project Plan



## 2. Overall System Description

SAI will be developed in the windows (OS) using android studio. SAI will run in android mobile devices. Only android mobile users will be able to use it. The user will be able to take images from the internal storage and then make sketches of those images with the SAI.

Then the user will be able to save the image and share the image with the help of some external app that is available in the device in which SAI will be installed. There is no need for any particular database. The data, in the present case, images will be stored directly in the storage of the device based on the permissions given by the user to the app.

## ***2.1 User characteristics***

Users of the SAI will be able to access images from the storage of phone that he/she has given the permission to access the storage of phone. User will be able to convert an image into sketch. User will also be able to save the image into phone storage. User will also be able to discard the sketch. User will also be able to share the image.

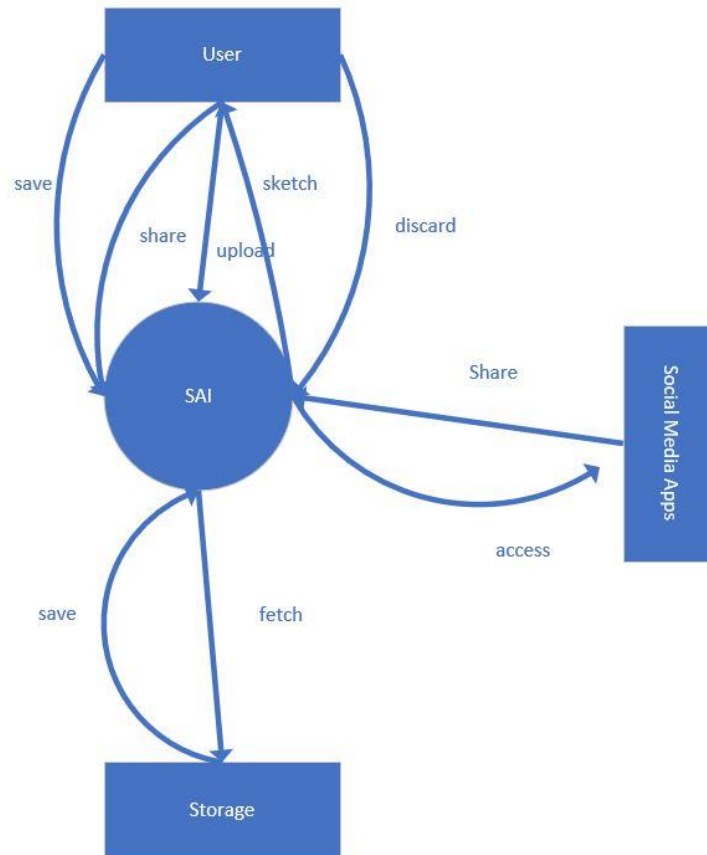
## ***2.2 Operating environment***

The SAI will work on android OS. Android Studio will be used as a platform for developing SAI. The hardware will be android mobile phones and tabs. There will be no database in it but internal storage will be used to directly save the data into it. SAI will have to coexist with apps that can share images over the internet.

## ***2.3 System constraints***

The SAI will work offline. SAI will only run on android OS and only on android mobile devices. Only phone storage will be used and phone storage must require permission for saving images. No database will be used for saving images.

### 3. External Interface Requirements



**Figure: Context Diagram**

Context Diagram describe the system level functionality of SAI. User, Storage and Social media apps are external objects that will access the application. And application represents in circle.

#### 3.1 Hardware Interfaces

The image data will be taken from the storage of the device and then the data will be used for all the processing that is to be done to make a sketch. Afterwards if required this data will be saved back to the storage of the device. Any data sharing capability will not be within SAI itself rather SAI will interact with other apps which then will interact with their respective servers.

#### 3.2 Software Interfaces

The library for graphics manipulation will be used. OS will be Android. There will be no database required but a direct image storage. Libraries for algorithms such as sobel and canny will be used.

Following are the software used for the SAI application.

Software used	Description
Operating system	We have chosen Android operating system for its best support and user-friendliness.
Storage	To save an image, we have chosen internal storage of phone.

## 4. Functional Requirements

- **Access Images**  
SAI will access the internal storage of phone for selecting an image to convert it into a sketch.
- **Image Processing**  
SAI will convert the image into a sketch. Application will provide different types of edge detection methods.
- **Save Image**  
After converting image into sketch, user could also save the image in the internal storage of phone.
- **Discard Image**  
If user do not want to save image then he/she can discard the image.
- **Share Image**  
SAI will be able to share the image by using social media apps.

## 5. Non-functional Requirements

### 5.1 Performance Requirements

Performance of SAI will measure in processing time of converting image into sketch. Processing time will be less than 1 minute. Processing time is the time that require by a application for convert an image into sketch.

### 5.2 Extensibility

SAI will be capable for extend its features in future. Application could be able for adopt changes. SAI will be extendable without affect the core functionality of application.

### **5.3 Maintainability**

*Application will be easily modifiable in the case of any error occur. If error occurs and in the case of operation failure SAI will be easily repairable.*

### **5.4 Usability Requirement**

SAI will be easily useable for end users. Interface and working of application will be easily understandable for end users for use SAI.

## **6. Assumptions and Dependencies**

### **Assumptions**

1. the gallery is accessible on user's device
2. image is not distorted
3. image is not blurry
4. user has given us permission to access storage
5. user has given permission to store the data on storage
6. user's device has enough processing power
7. external sharing apps should be able to handle bitmaps/images

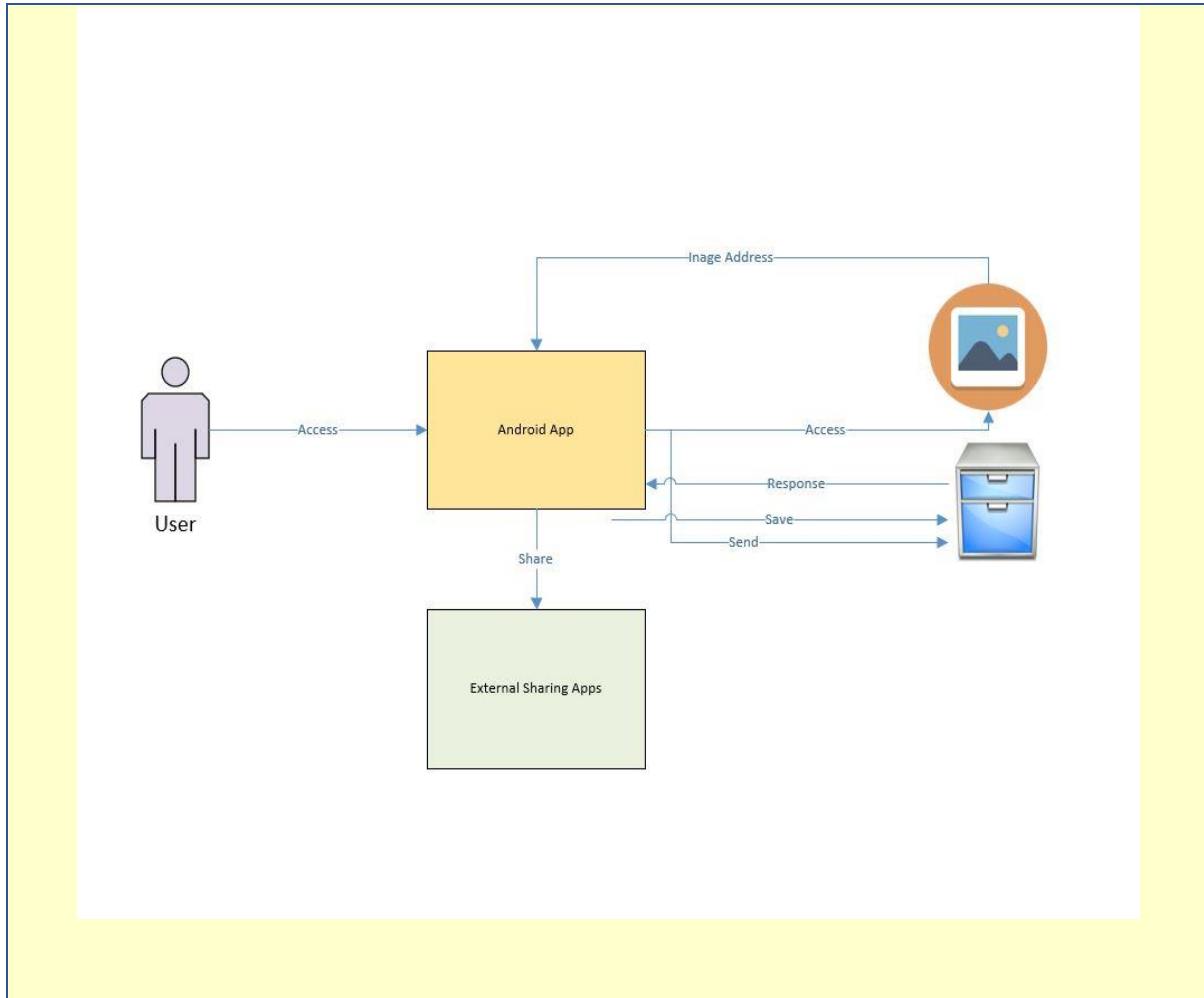
### **Dependencies**

1. the accessibility of the image from gallery
2. user to give us permission to store and retrieve data from storage
3. user needs to be able to understand the hints given in app
4. External Sharing apps

## 7. System Architecture

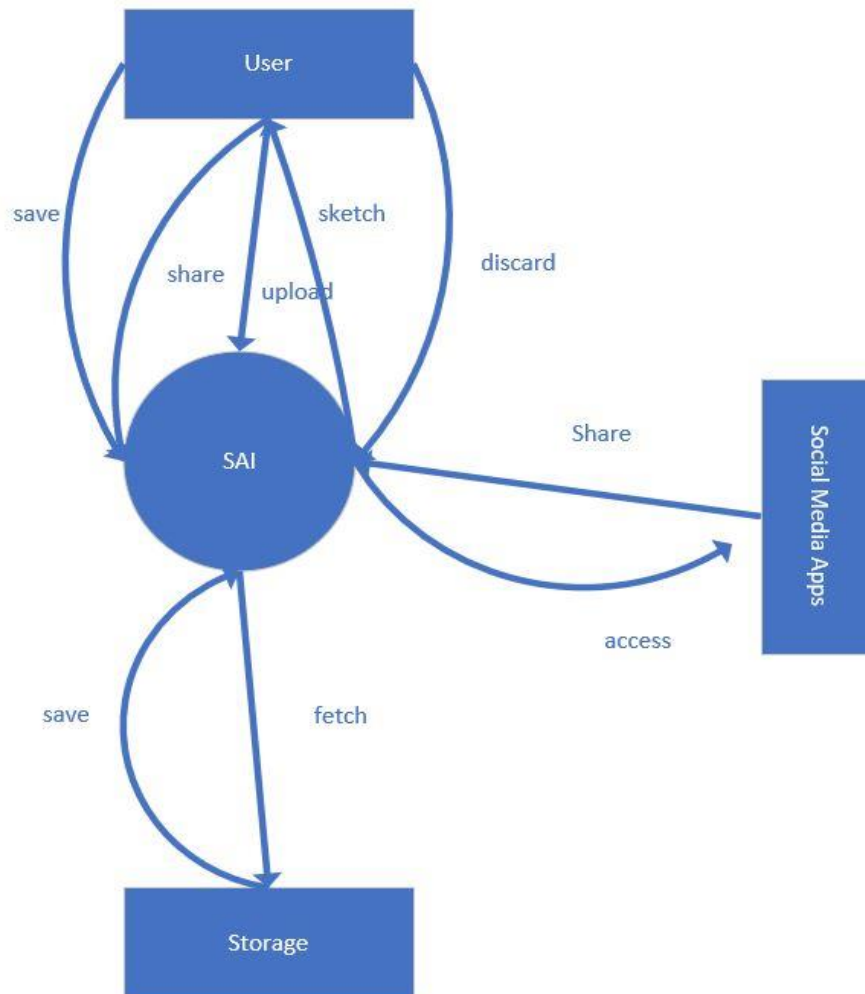
System architect represents the higher-level design and view of system. And how different module will interact with each other.

### 7.1 System Overview



## 7.2 Context Level DFD

Context Diagram is also called 0-level DFD. Context Diagram describe the system level functionality of SAI. User, Storage and Social media apps are external objects that will access the application. And application represents in circle.

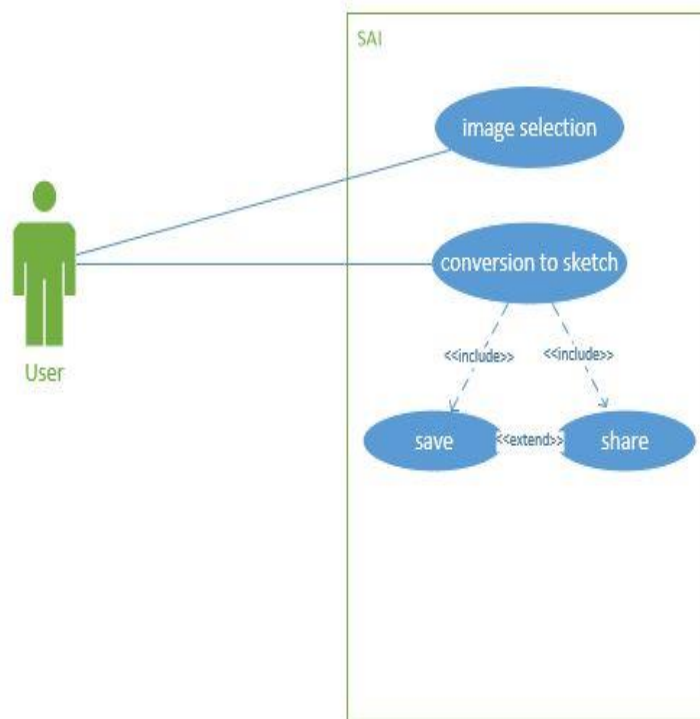


## 8. Use Cases

### 8.1 Use Case Diagrams

Use case diagram describes the interaction of user application. This use describes the function that user can perform. These functions include image selection, conversion, save and share.

**Actor:** USER\_An external object that interact with system.





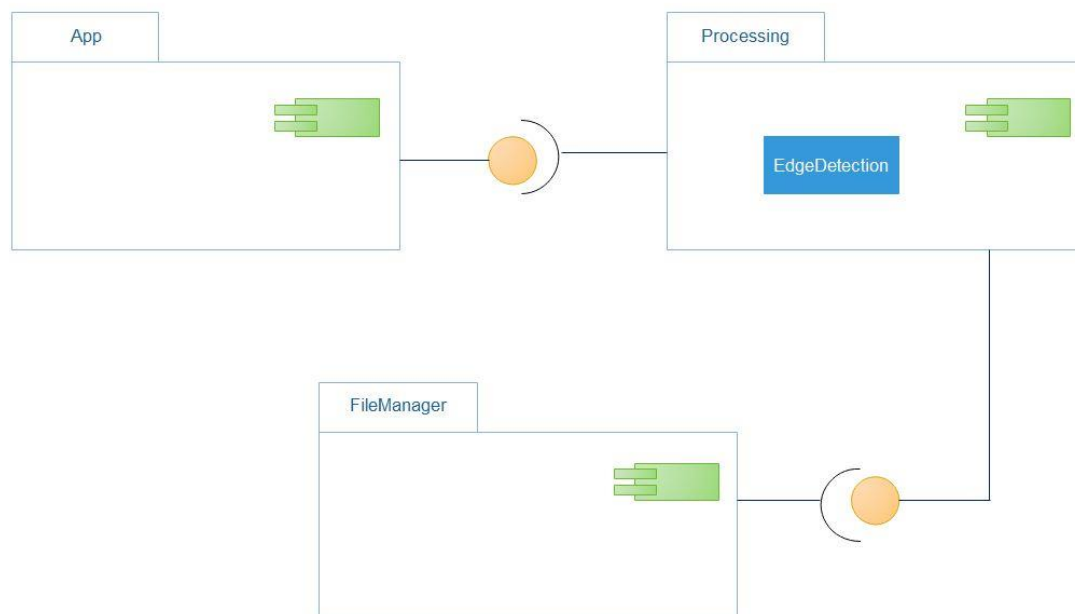
## 8.2 Use Case Description

USER		
Actors: <i>User</i>		
Use case Id:	<i>1</i>	
Pre-condition:	<i>User have images in gallery</i>	
Scenarios		
Step#	Action	Software Reaction
1.	<i>Selection Image</i>	<i>Show sketch to user</i>
2.	Conversion	
3.	Save	
4.	Share	

## 8.3 UML Diagrams

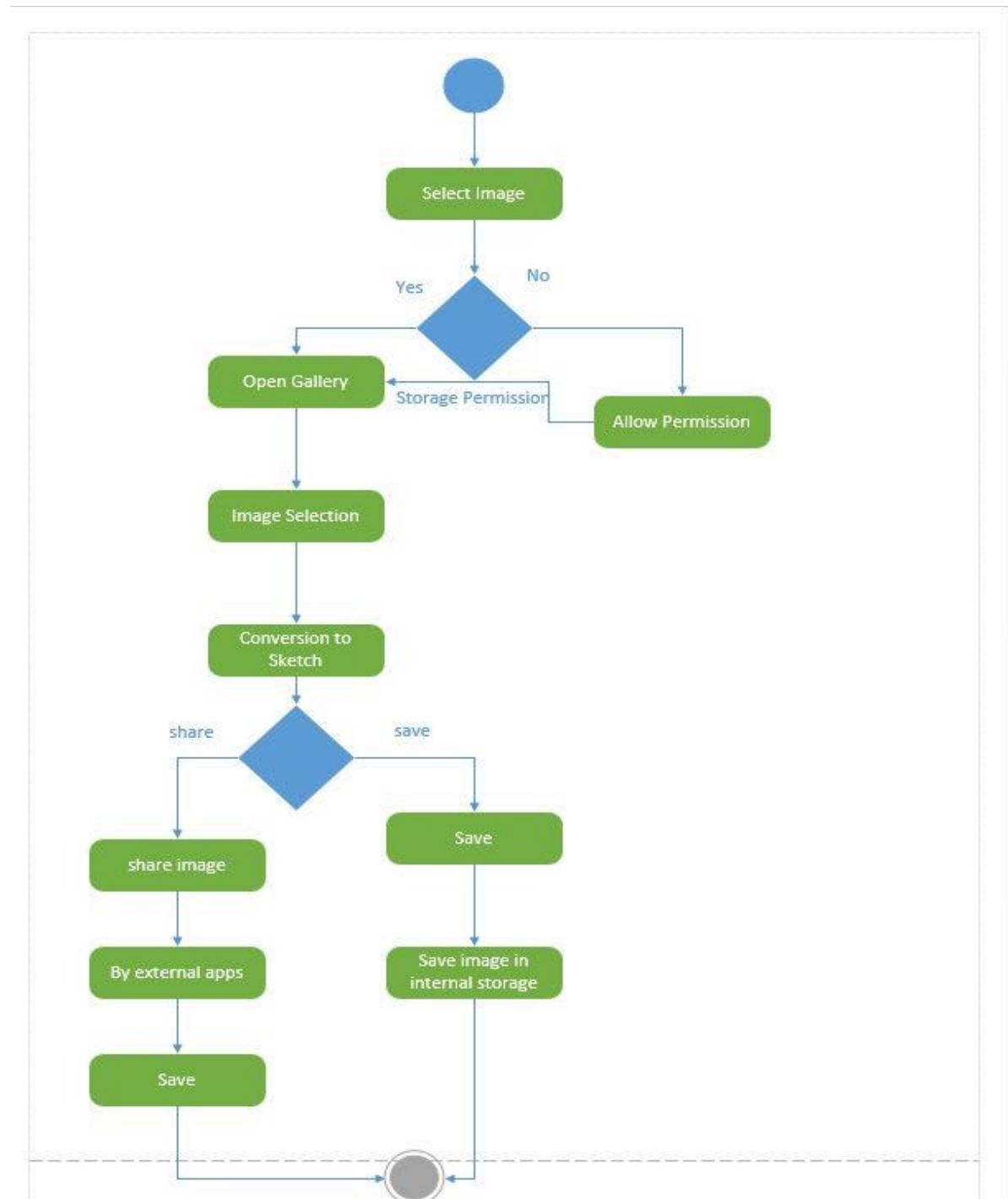
### 8.3.1 Component Diagram

Component diagram describes the relationship between components of application. This diagram also represents how data share between components. One component sends or receive data from another component for processing.



### 8.3.2 Activity Diagram

Activity diagram describes the behavior of application. It describes the flow of events in system. An activity diagram portrays the control flow from a start point to a finish point showing the various decision paths that exist while the activity is being executed.

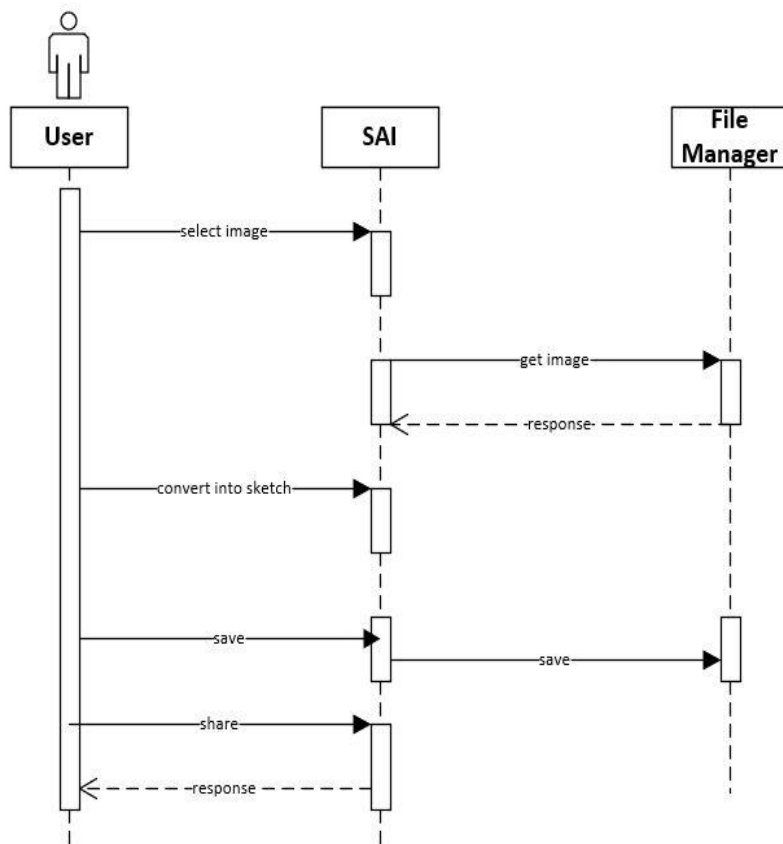


### 8.3.3 Sequence Diagram

Sequence diagram depicts the interaction between the objects in sequential order.

**Objects:** User, SAI and File manger

These objects interact with each other and messages exchanged between the objects needed to carry out the functionality of system.

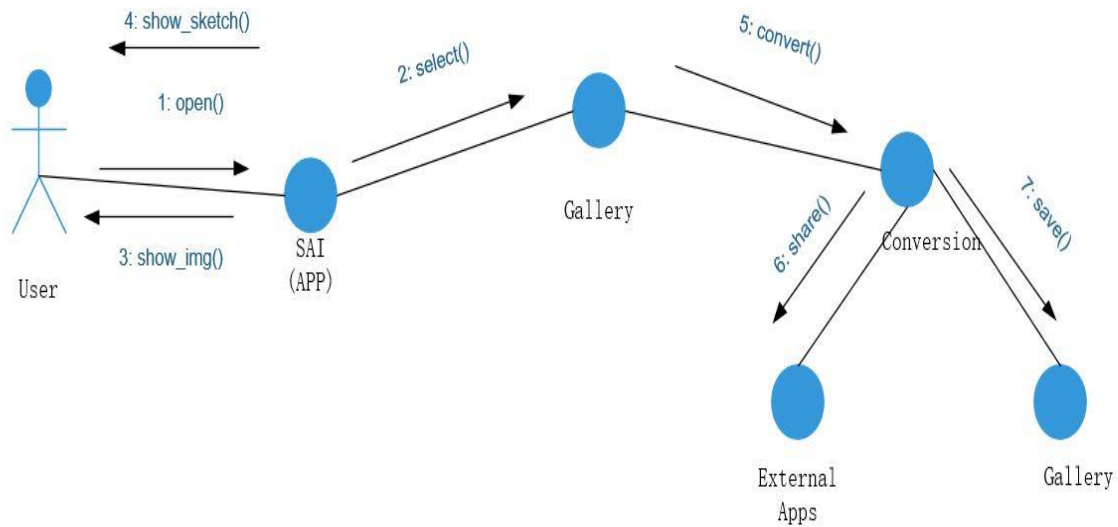


:

### 8.3.4 Collaboration Diagram

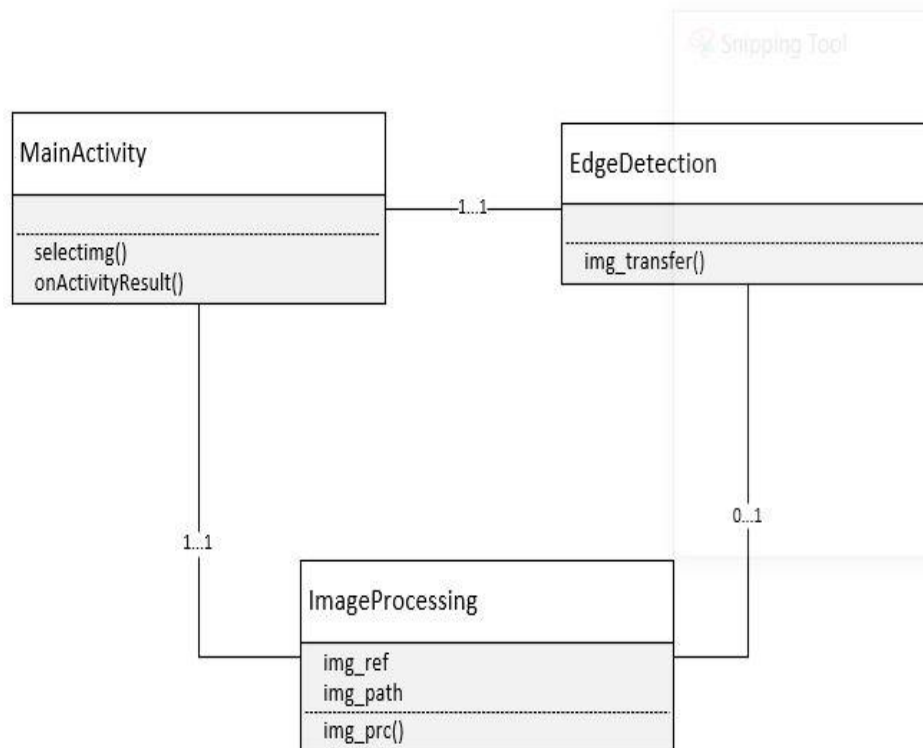
Collaboration diagram is also called communication diagram, it describes the relationship and interaction between the object of application. In this diagram, user perform different action at different objects of application.

**Functions:** open, select, convert, save, share and show.

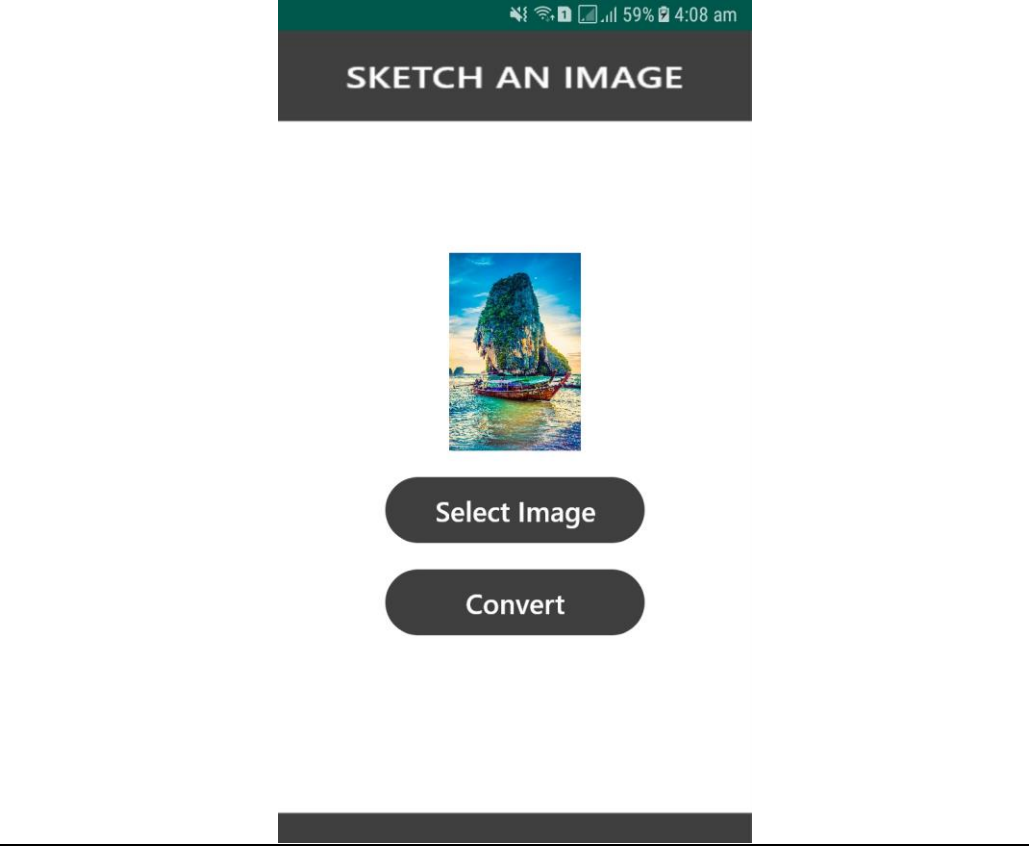


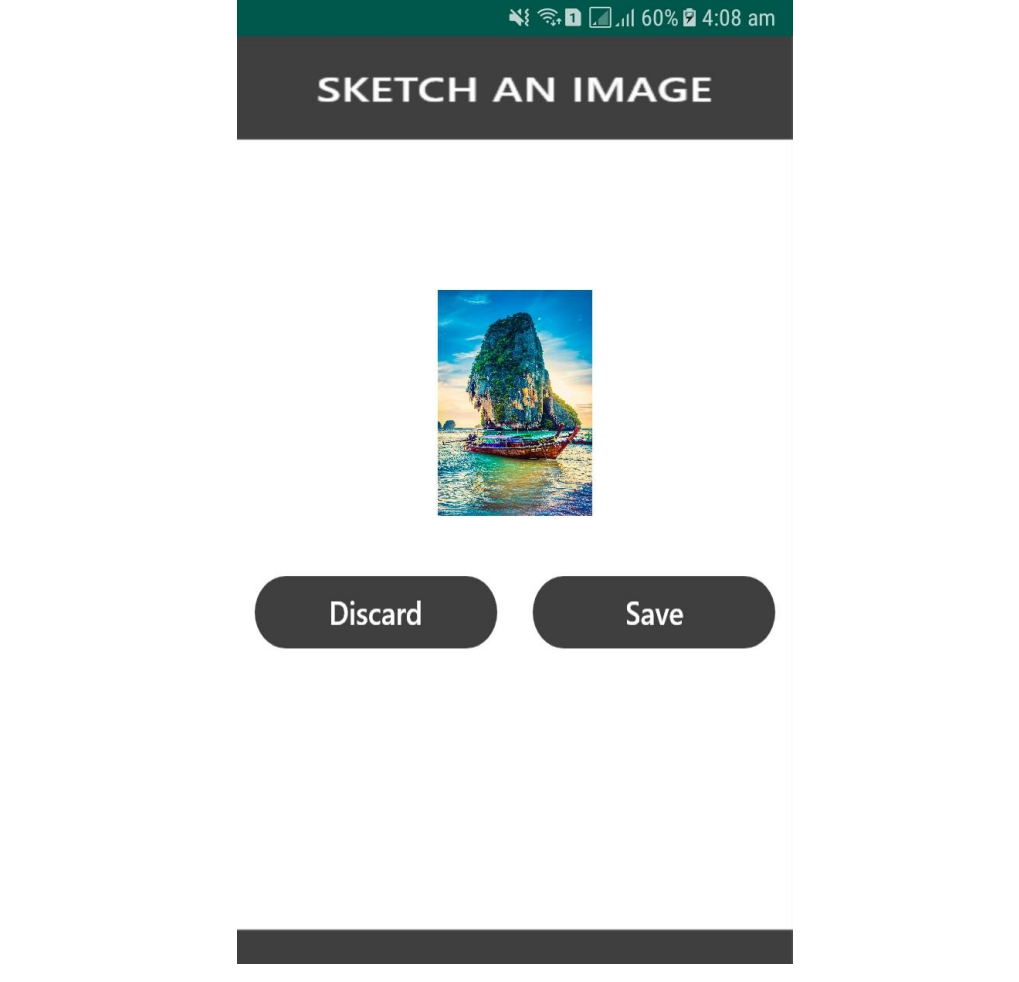
### 8.3.5 Class Diagram

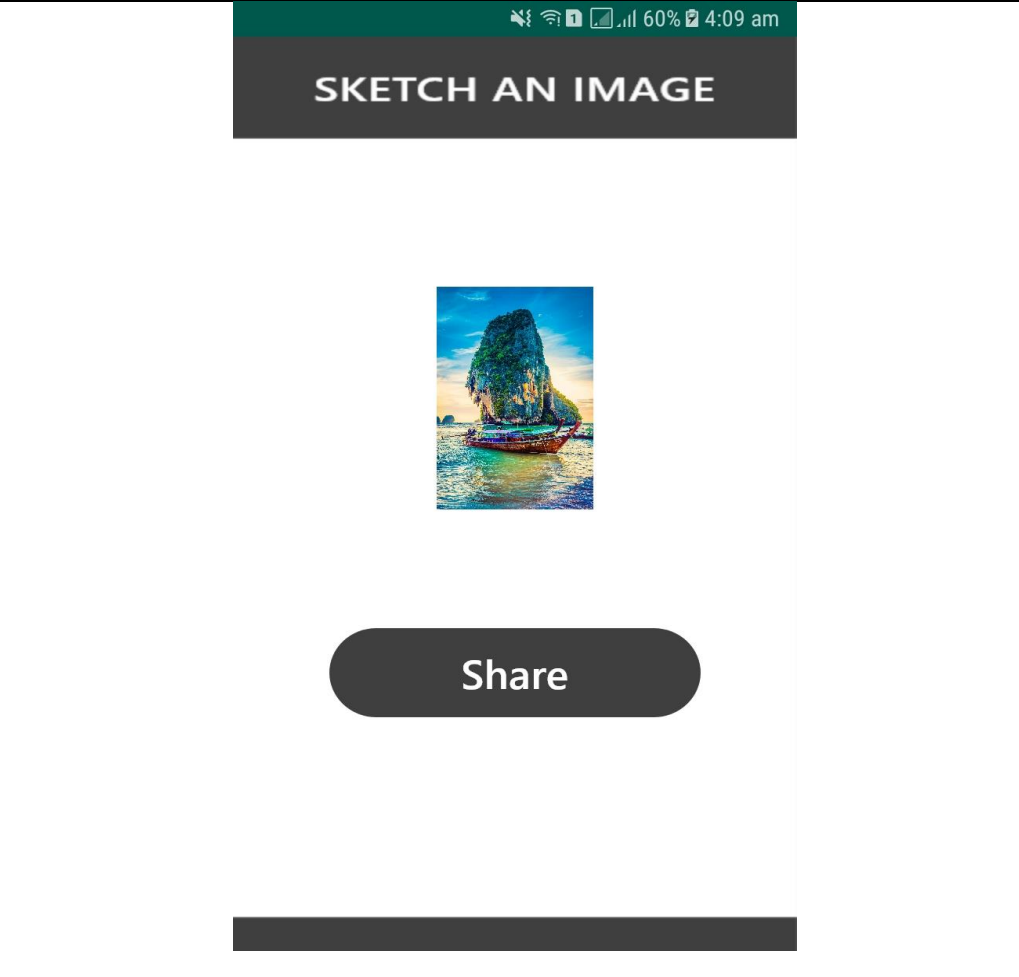
Class diagram describes the static structure of application. It describes the structure of a application by showing the classes, objects, methods and association between them. MainActivity, EdgeDetection and ImageProcessing class have different attributes and methods and also interact with each other.



## 9. Graphical User Interfaces

1.0: Image Selection Screen	
Interface Id.	1.0
Use case Reference	1
Snapshot	
 A screenshot of a mobile application interface titled "1.0: Image Selection Screen". The screen has a white background. At the top, there is a dark green status bar with white icons for signal, Wi-Fi, battery, and time (4:08 am). Below the status bar is a dark grey header bar with the text "SKETCH AN IMAGE" in white. In the center of the screen, there is a square image of a boat on water. Below the image are two dark grey buttons with white text: "Select Image" and "Convert". At the bottom of the screen, there is a dark grey bar.	

2.0: Save Screen	
Interface Id.	2.0.
Use case Reference	1
Snapshot	
	

3.0: Share Screen	
Interface Id.	3.0.
Use case Reference	1
Snapshot	
	



## 10. References

Ref. No.	Document Title	Date of Release/ Publication	Document Source
1	ALGORITHM AND TECHNIQUE ON VARIOUS EDGE DETECTION	June 2013	An International Journal (SIPIJ) Vol.4, No.3
2	EDGE DETECTION TECHNIQUES FOR IMAGE SEGMENTATION	Dec 2011	International Journal of Computer Science & Information Technology (IJCSIT) Vol 3, No 6
3	Edge Detection Methods	Jan 20, 2018	<a href="https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e">https://medium.com/@nikatsanka/comparing-edge-detection-methods-638a2919476e</a>