Cairo University

Faculty of Computers and Information

SCS351

Software Modeling

FCI image Toolbox

Software Requirements Specifications

Team Names

Nada Mohamed 20186031

Ayaat Hany 20186007

Sarah Khaled 20186008

Mark Rofaeel 20186043

Month & Year

March 2020

SCS351: Phase 2-Corona Team
Project: FCI image Toolbox

Software Design Specification

Contents

SCS351: Phase 2-Corona Team
Project: FCI image Toolbox

Software Design Specification

Team

ID	Name	Email	Mobile
20186031	Nada Mohamed	nadanody60083@gmail.com	+20 111 847 5294
20186007	Ayat Hany	ayathany26@gmail.com	+20112 642 0204
20186008	Sarah Khaled	sarahkhaledd2000@gmail.com	+20 112 519 5800
20186043	Mark Rofaeel	mark.rofaeel@gmail.com	+20 122 187 3225

Document Purpose and Audience

The purpose of this document is to give an in-depth insight description of FCI image Toolbox. It will provide an overview of our software product by collecting and analyzing all assorted ideas that have come up to define such product, and its detailed requirements with respect to customers. It defines how our client, team and audience see the product and its functionality.

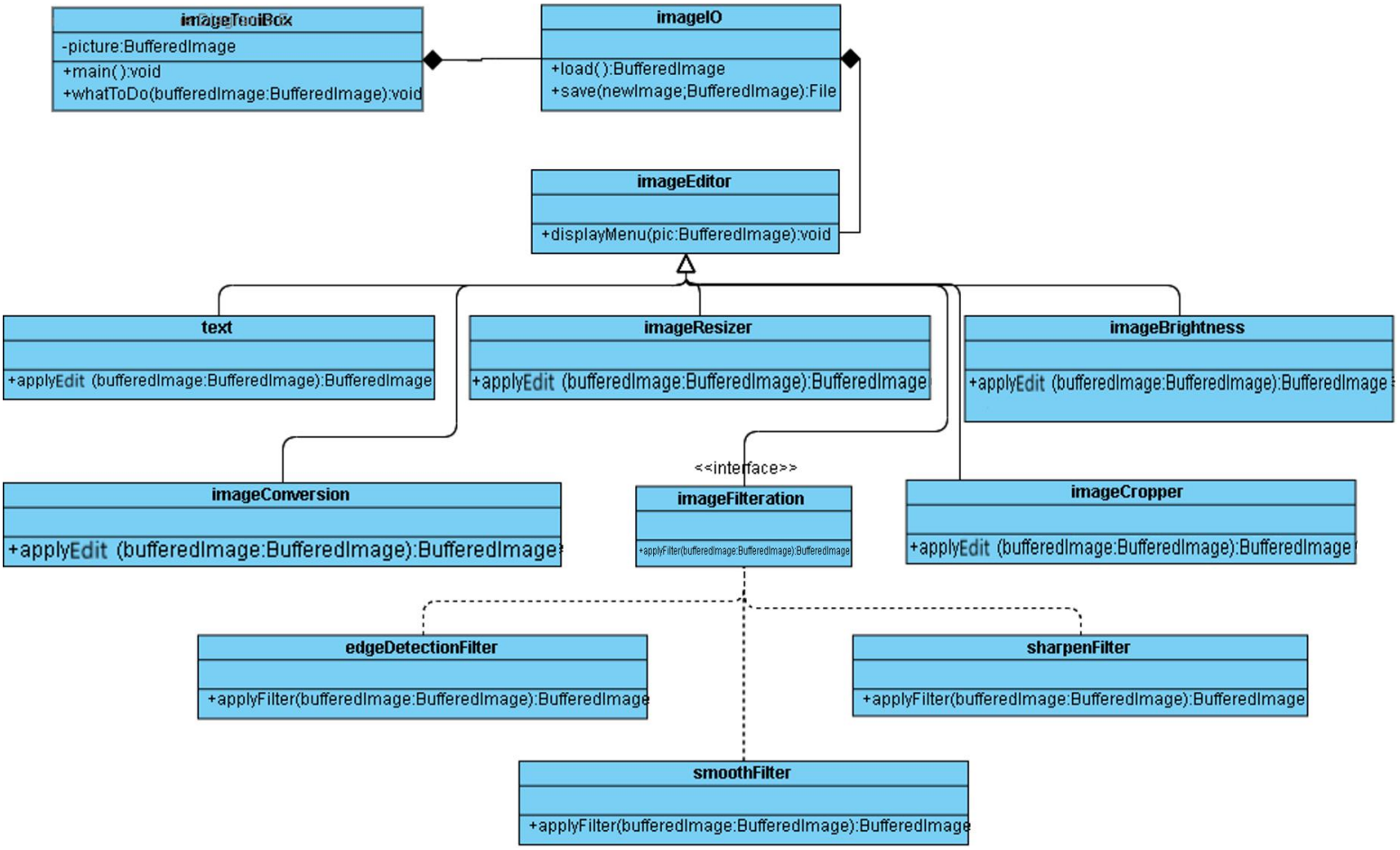
This document is intended for the stakeholders, mainly for the **customers**, **designers** and **developers** of the software.

SCS351: Phase 2-Corona Team
Project: FCI image Toolbox

Software Design Specification

System Models

I. Class diagram:



Software Design Specification

II. Class diagram table:

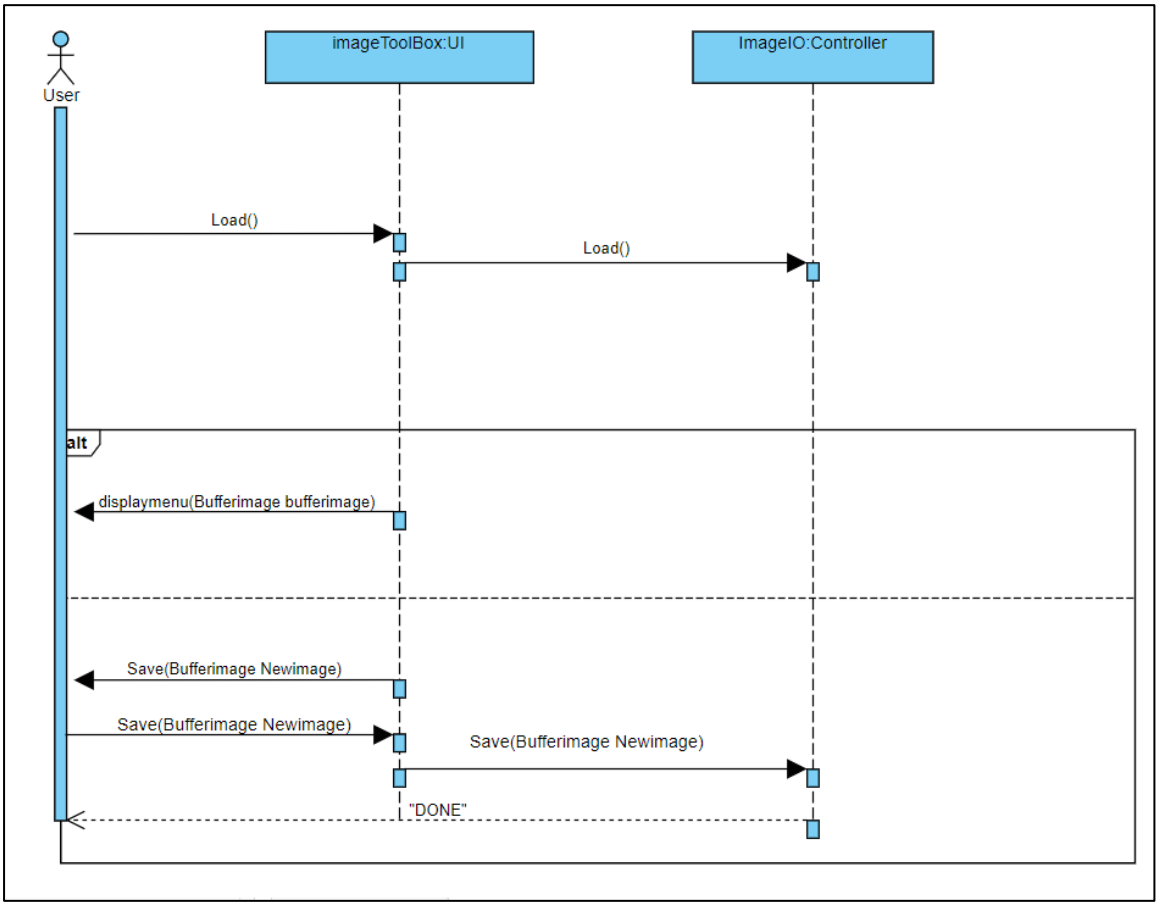
Class Name	Description & Responsibility
imageToolBox	This is the **main** class (UI).
imageIO	This is a controller class of **image** class that controls the main operations of this class. This class is also responsible for **saving or loading PNG images from the user.**
imageEditor	This class is responsible for **displaying the menu** to the user. Also, there is a **composition** relationship with **imageIO** class.
imageBrightness	This class inherits from **imageEditor** class. This class is responsible for **brightening images**.
imageCropper	This class inherits from **imageEditor** class. This class is responsible for **cropping images** to certain size entered by the user.
text	This class inherits from **imageEditor** class. This class is responsible for **adding text** to images.
imageResizer	This class inherits from **imageEditor** class. This class is responsible for **resizing images** to certain length and width.
imageConversion	This class inherits from **imageEditor** class. This class is responsible for **converting colored images into black and white one**.
imageFiltration	This class is **interface** class.
sharpenFilter	This class implements from **imageFiltration** class and responsible for one function “applyFilter(bufferedImage:BufferedImage):BufferedImage”

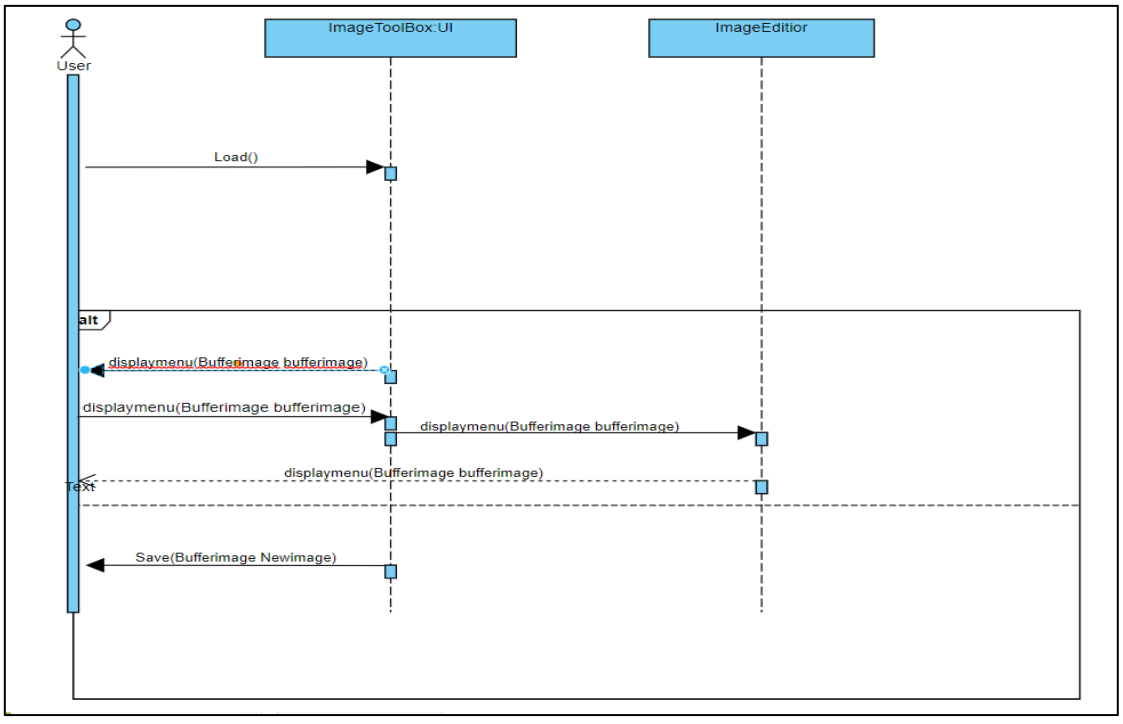
Software Design Specification

Class Name	Description & Responsibility
smoothFilter	This class implements from **imageFiltration** class and responsible for one function "applyFilter(bufferedImage:BufferedImage):BufferedImage"
edgeDetectFilter	This class implements from **imageFiltration** class and responsible for one function "applyFilter(bufferedImage:BufferedImage):BufferedImage"

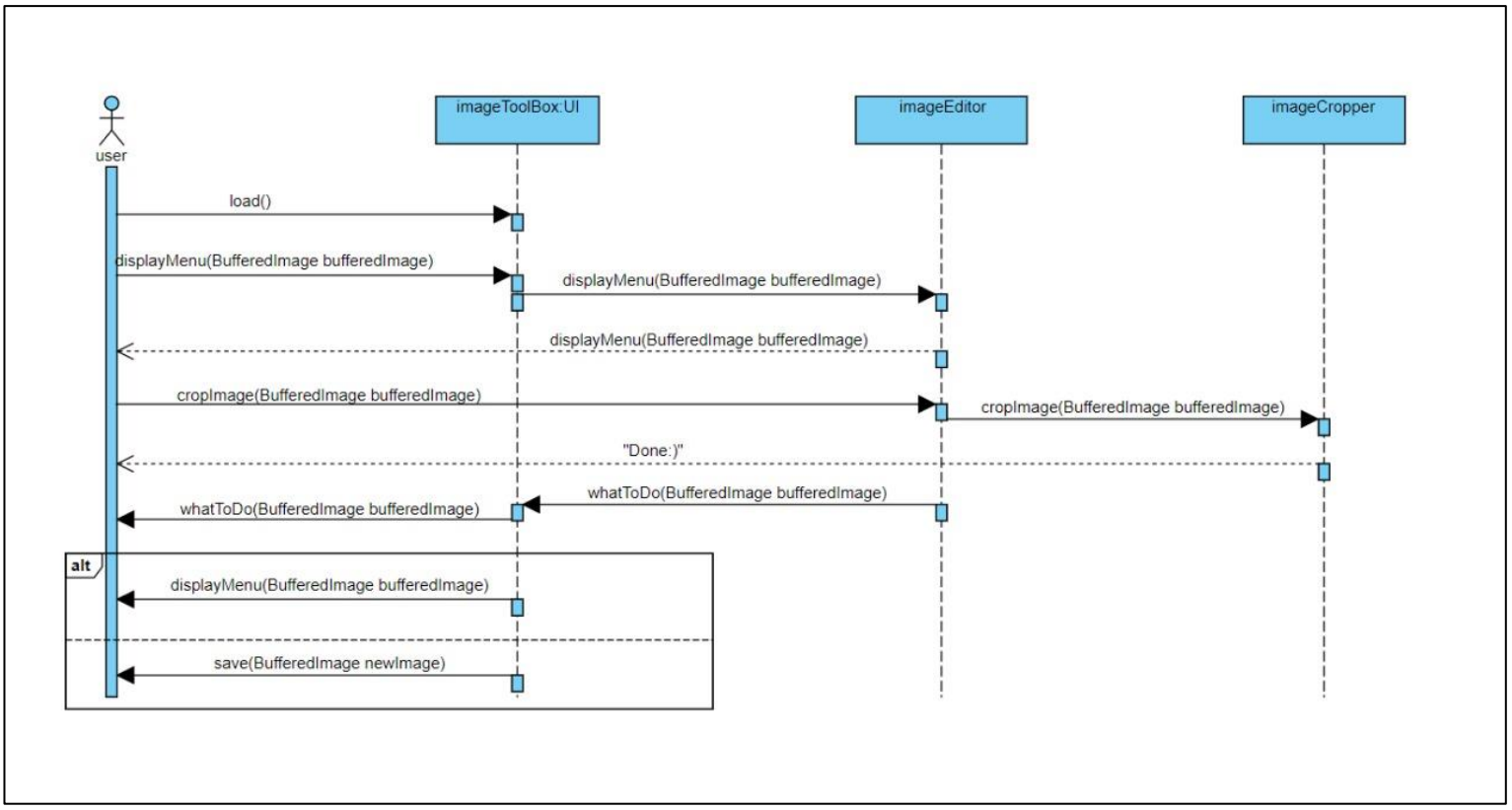
Software Design Specification

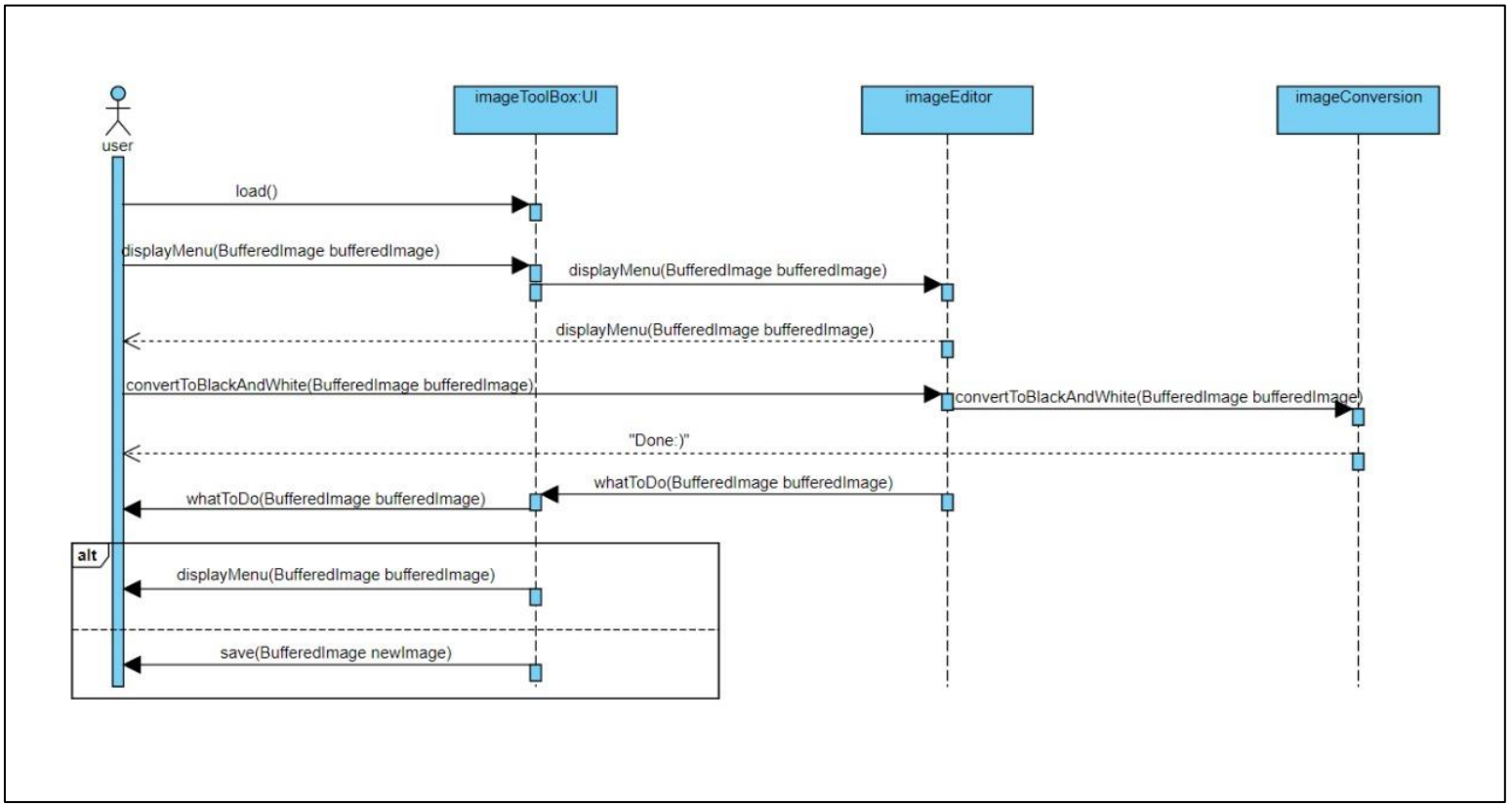
II. Sequence diagrams:



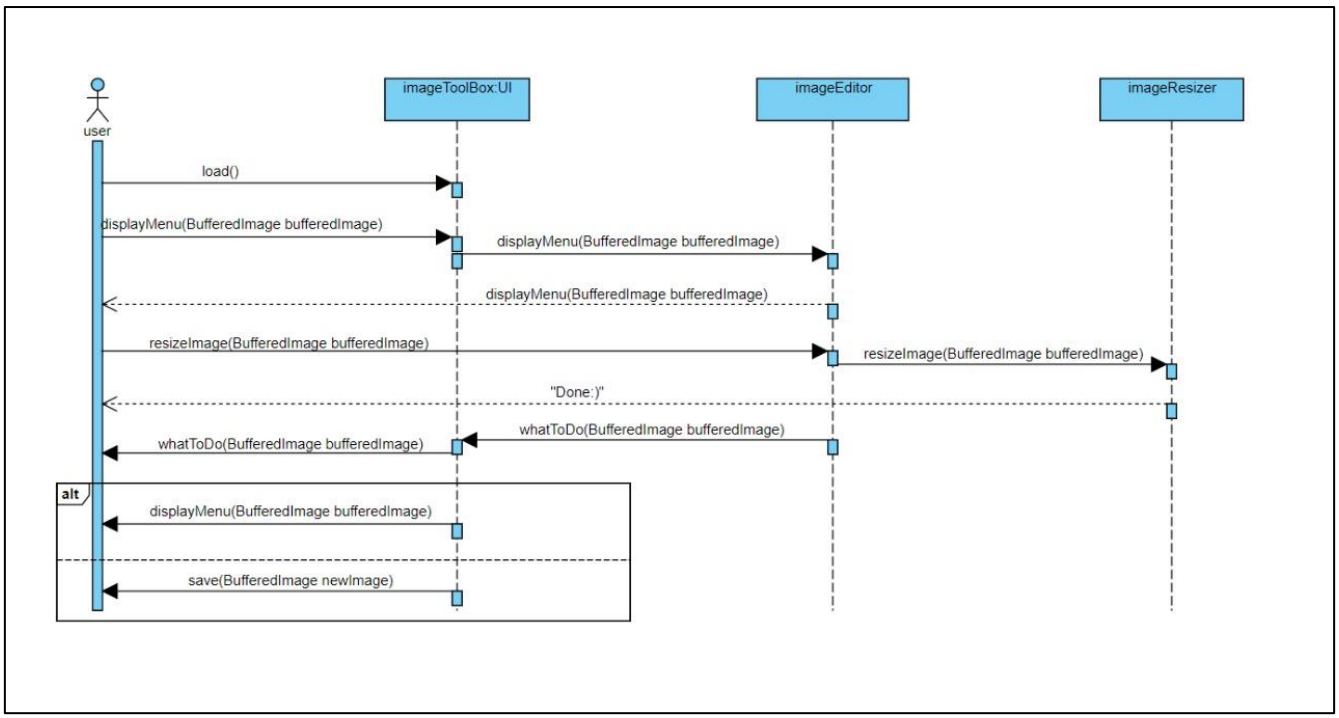


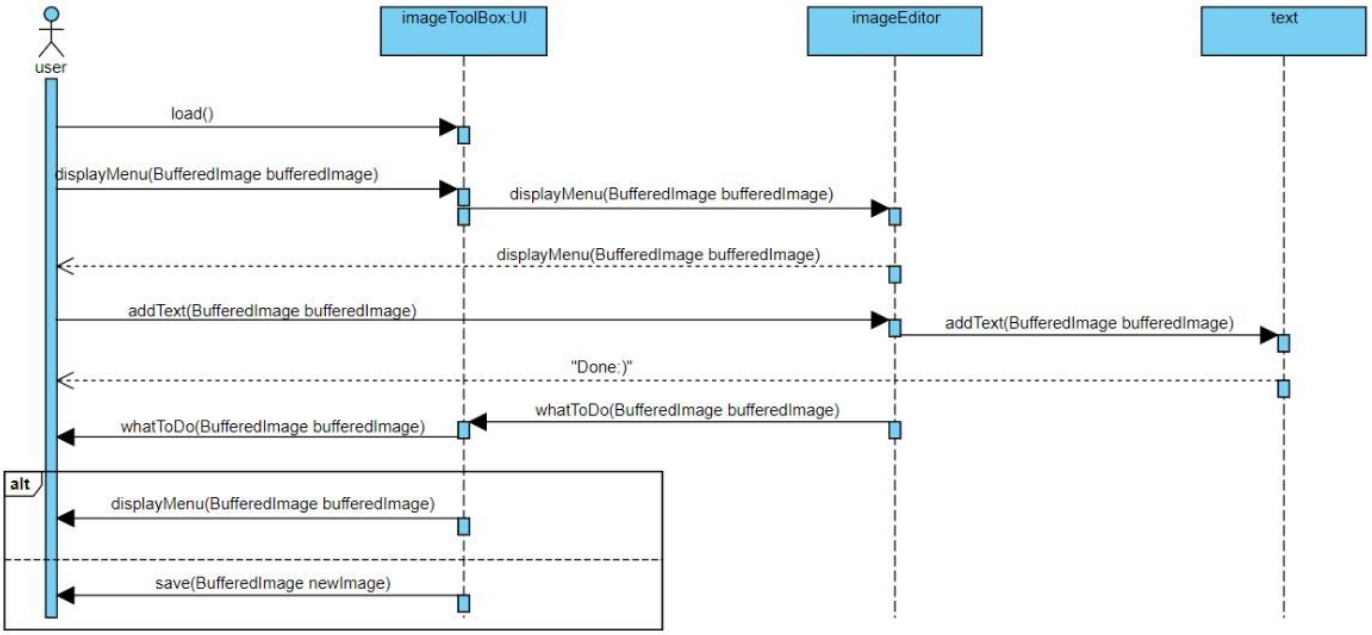
Software Design Specification



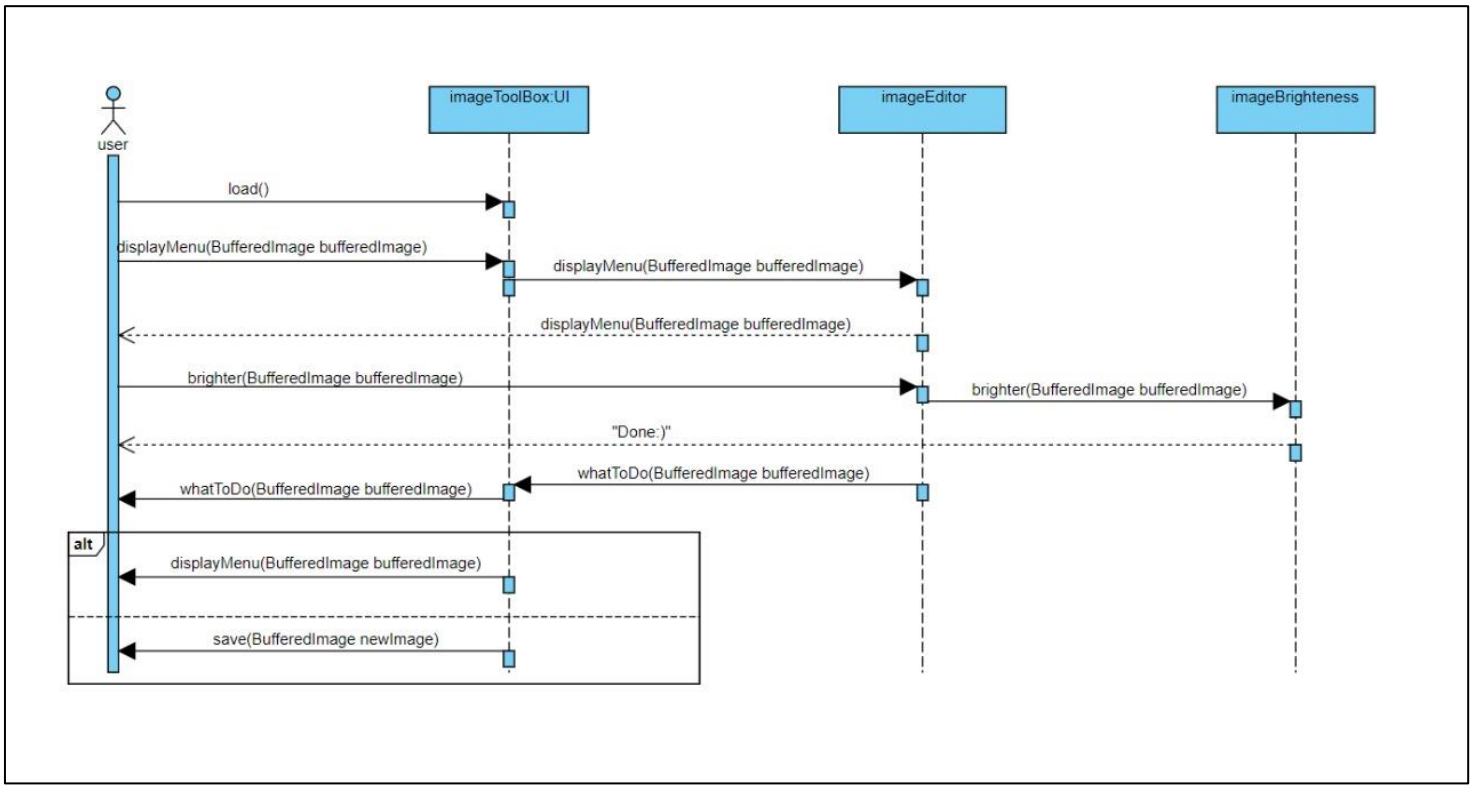


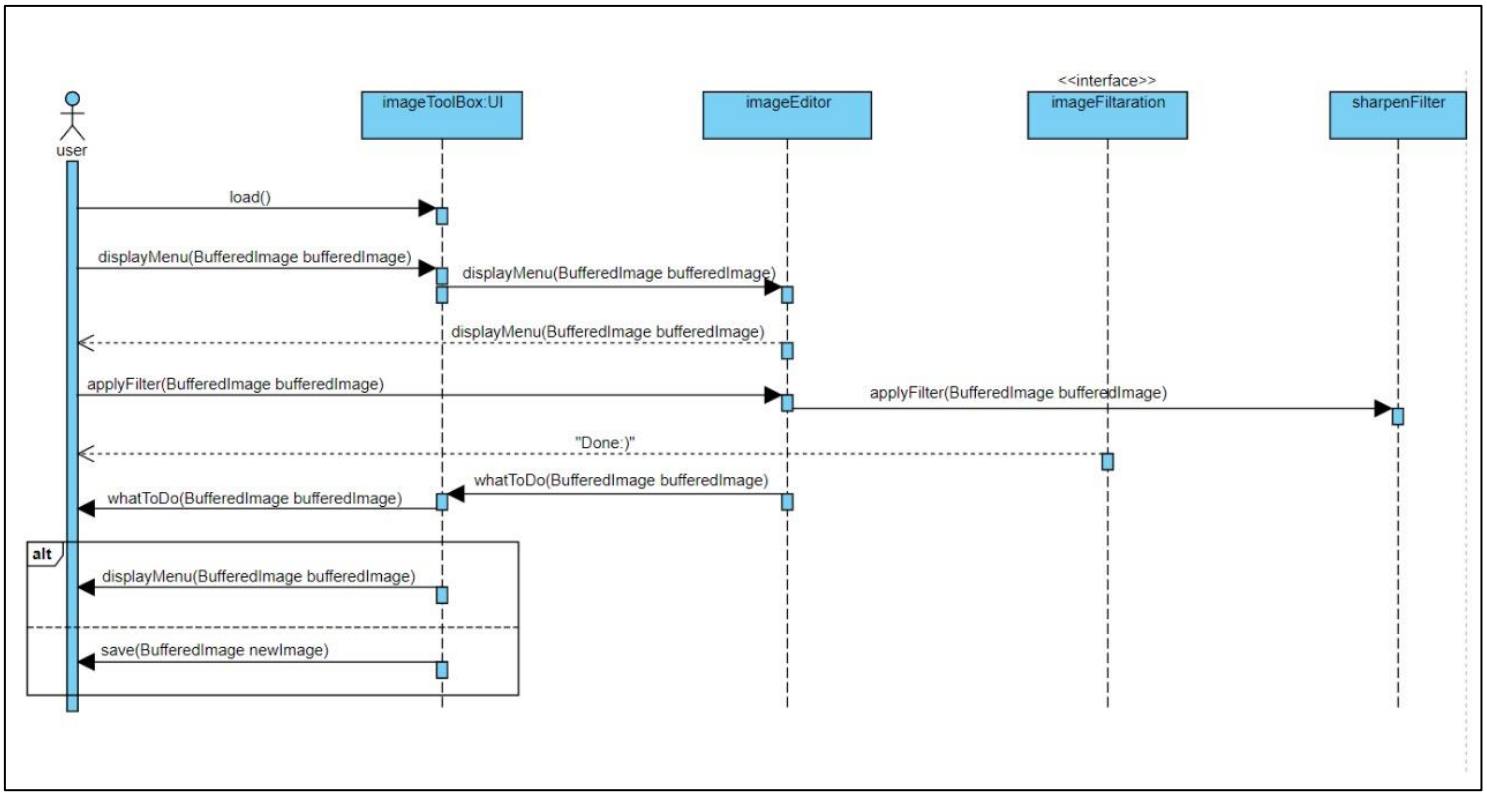
Software Design Specification



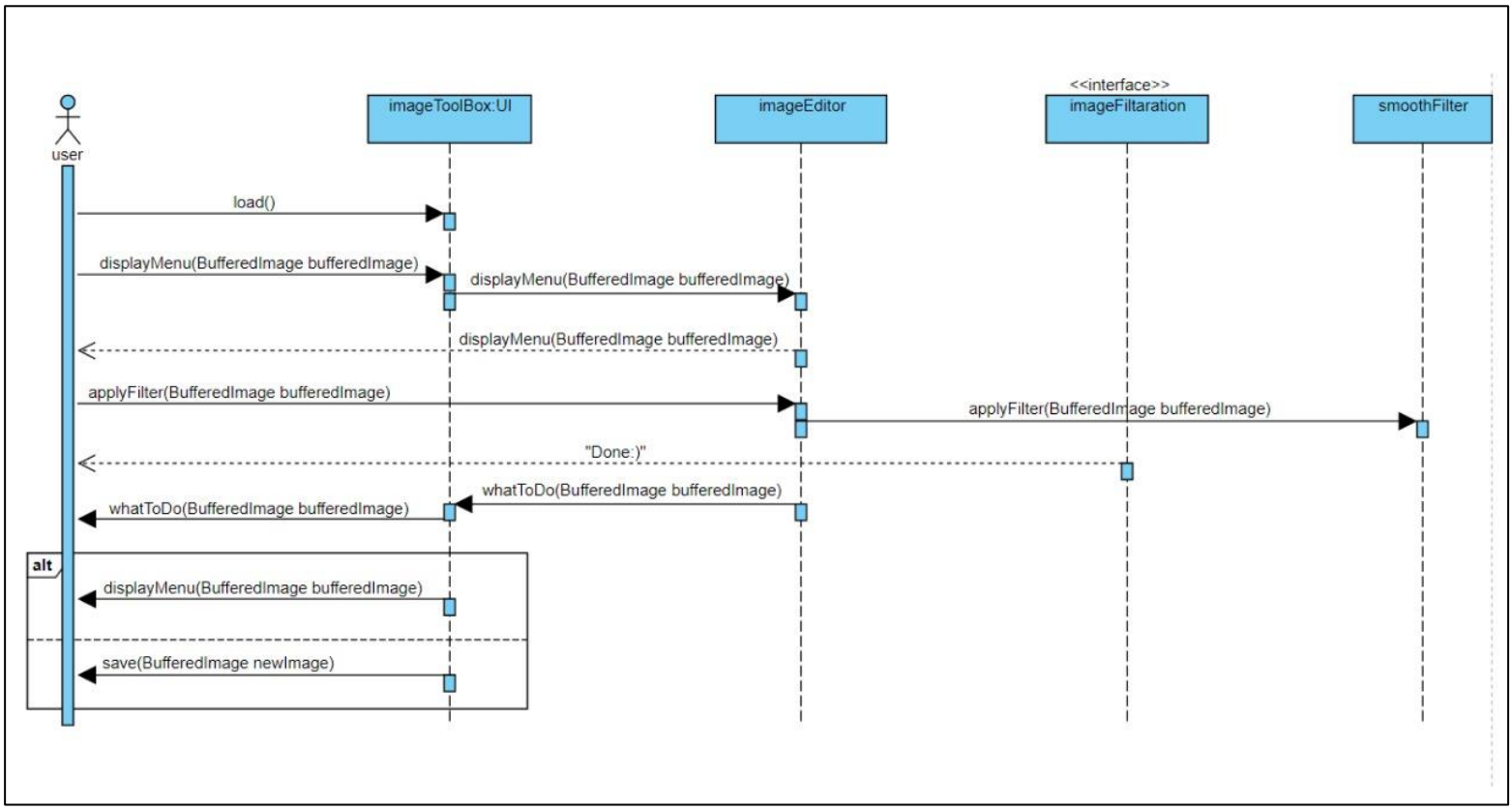


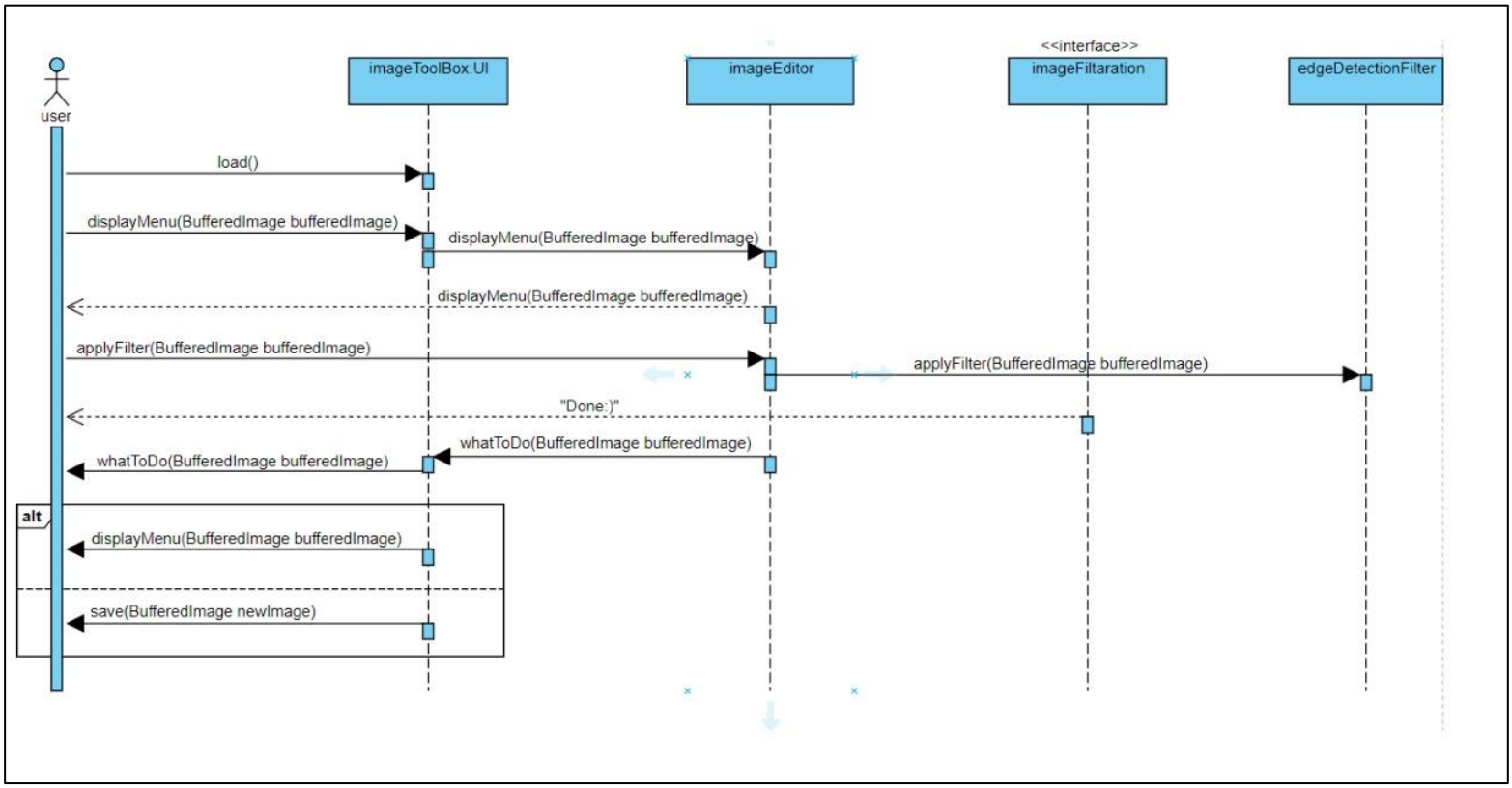
Software Design Specification





Software Design Specification





Software Design Specification

Class - Sequence Usage Table:

Class Name	Sequence Diagrams	Overall used methods
imageToolbox imageIO	1	main():void whatToDo(bufferedImage:BufferedImage):void load():BufferedImage save(newImage:BufferedImage):File
imageToolbox imageEditor	2	displayMenu(pic:BufferedImage):void save(newImage:BufferedImage):File
imageCropper	3	applyEdit(bufferedImage:BufferedImage):BufferedImage
imagconversion	4	applyEdit(bufferedImage:BufferedImage):BufferedImage
imageResizer	5	applyEdit(bufferedImage:BufferedImage):BufferedImage
text	6	applyEdit(bufferedImage:BufferedImage):BufferedImage
imageBrightness	7	applyEdit(bufferedImage:BufferedImage):BufferedImage
sharpenFilter	8	applyFilter(bufferedImage:BufferedImage):BufferedImage
smoothFilter	9	applyFilter(bufferedImage:BufferedImage):BufferedImage
edgeDetectionFilter	10	applyFilter(bufferedImage:BufferedImage):BufferedImage

Ownership Report:

Everything is done by all team members.