

SWEN Prixos Design Document

Isabella Sturm

Ivan Kovacevic

Srdan Lazarevic

22 April 2017

Class Table

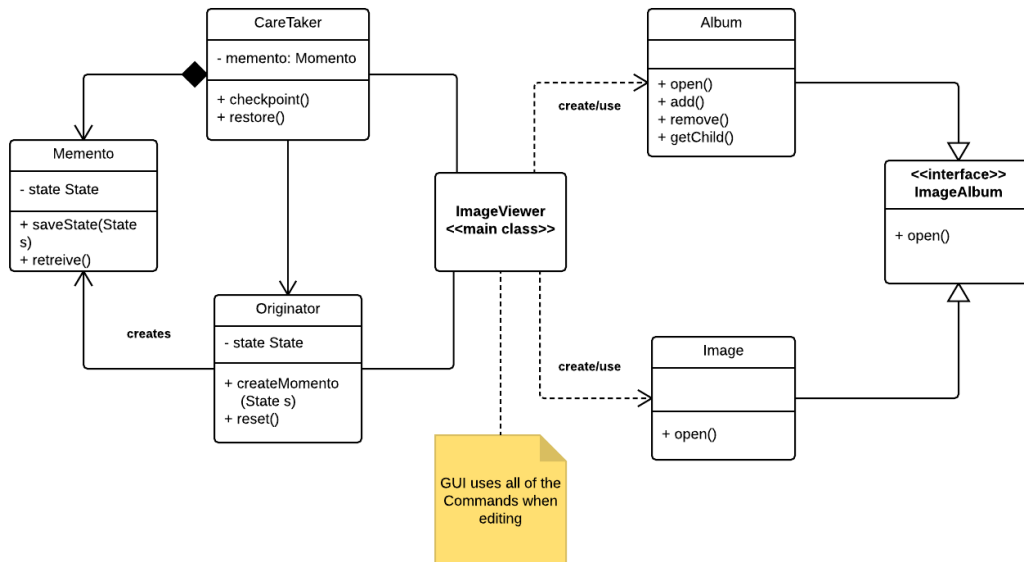
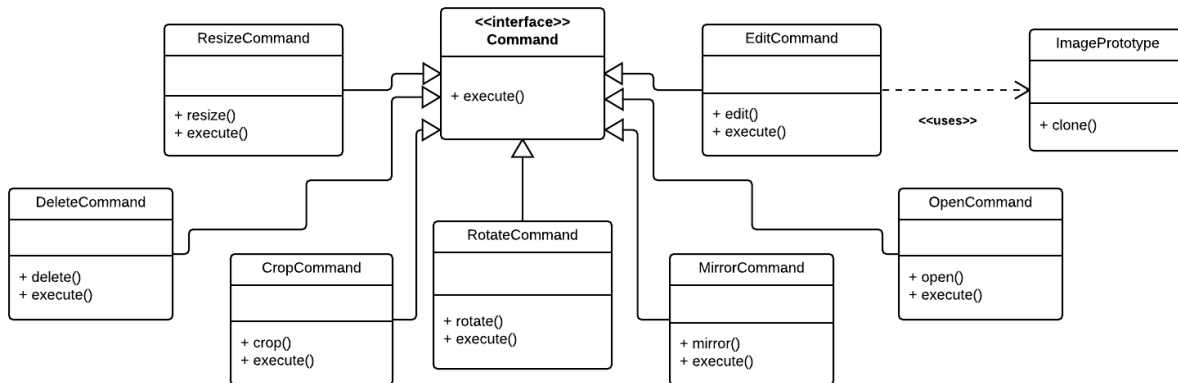
Class Name	Class Description	Uses	Used By	Methods
ImageViewer	Main class; GUI. This class provides the user interface of the program.	ResizeCommand, CropCommand, RotateCommand, MirrorCommand, CareTaker, Originator		main() ImageViewer()
Command	Interface for commands that will control the functionalities allowed by the user		ResizeCommand, CropCommand, MirrorCommand, RotateCommand, EditCommand	execute()
ResizeCommand	Implements Command interface; when called by user, resize command prompts user to change size of image	Command interface	ImageViewer	resize() execute()
CropCommand	Implements Command interface; when called, user can crop image	Command interface	ImageViewer	crop() execute()
RotateCommand	Implements Command interface, when called upon, user can chose to rotate image by custom number of degrees or by the standard 90 or -90 degrees	Command interface	ImageViewer	rotate() execute()

MirrorCommand	Implements Command interface, when called, user can mirror original image vertically or horizontally	Command interface	ImageViewer	mirror() execute()
EditCommand	Implements Command interface; when user wants to edit page, EditCommand calls prototype to make copy of original photo before editing	Command interface, ImagePrototype	ImageViewer	edit() execute()
DeleteCommand	Implements Command interface; allows users to delete images and albums when called, but also can “cancel” the editing of an image, which will delete the cloned (edit) image	Command interface	ImageViewer	delete() execute()
OpenCommand	Implements Command interface; opens albums/ images. If the user opens an album, it will reveal the contents of the album in a 2x2 grid; if the user opens an	Command interface, ImageAlbum/ Album/ Image	ImageViewer	open() execute()

	image, the image will be displayed on its own.			
ImageAlbum	Interface for images and album/ labels (Composite Pattern)		Album, Image	open()
Album	Implements ImageAlbum; "Directory" level; Array or ArrayList to store images in a specific album	ImageAlbum interface	ImageViewer, Command	open() add() remove() getChild()
Image	Implements ImageAlbum; "File" level;	ImageAlbum interface	ImageViewer, Command	open()
CareTaker	Stores mementos and can restore (redo/undo) to another state of the image	Memento	ImageViewer	checkpoint() restore()
Originator	Creates and saves mementos of the image	Memento	ImageViewer	createMemento() reset()
Memento	Keeps state of mementos of image when image is edited		CareTaker, Originator	save(State s) retrieve()
ImagePrototype	Provides functionality to clone image that user wants to edit so the original can remain and the new		EditCommand	clone()

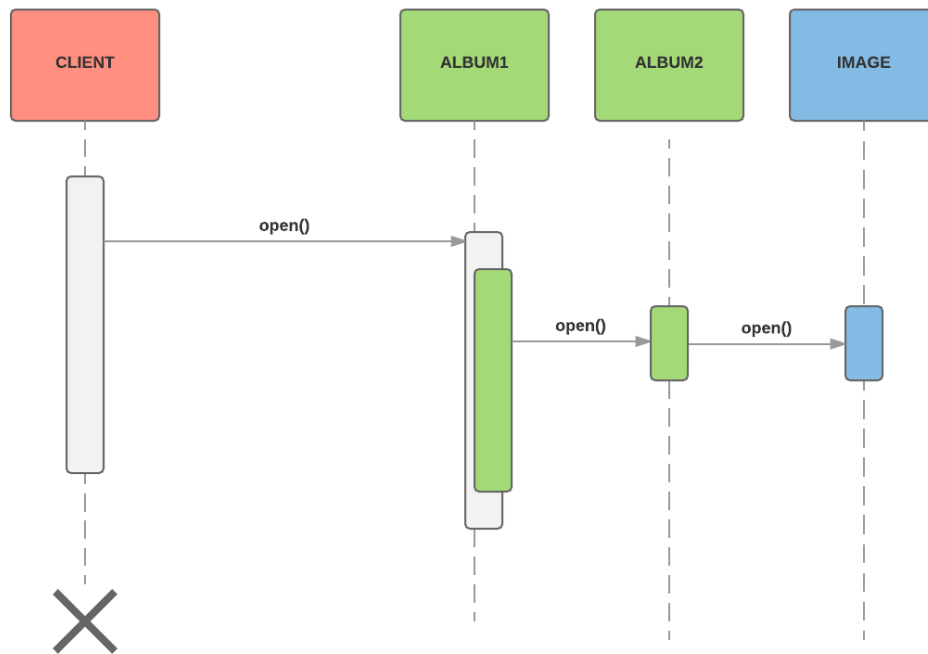
	edited image can be created			
--	--------------------------------	--	--	--

UML Diagrams

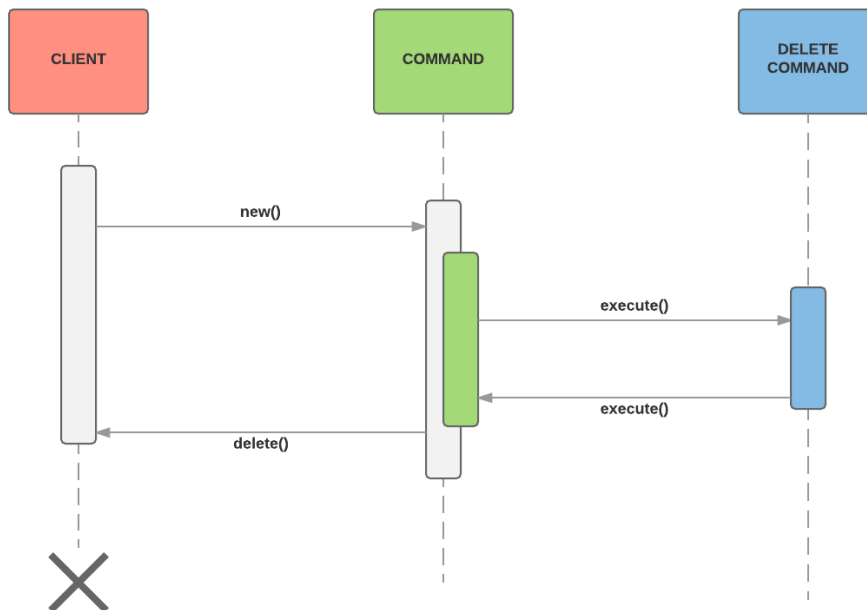


Sequence Diagrams

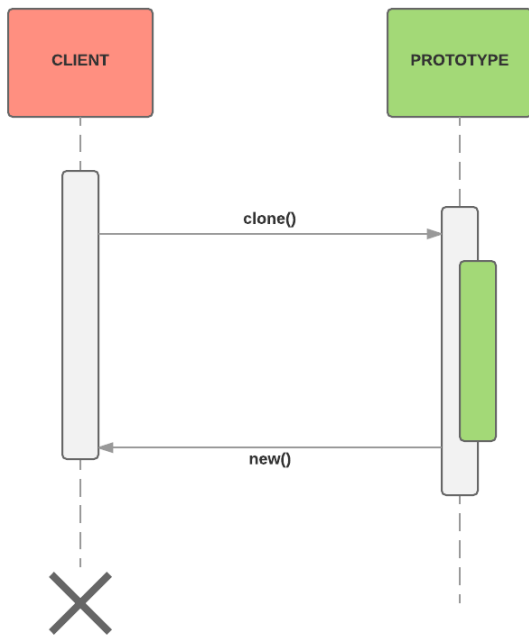
ImageAlbum/ Album/ Image -- Composite Pattern



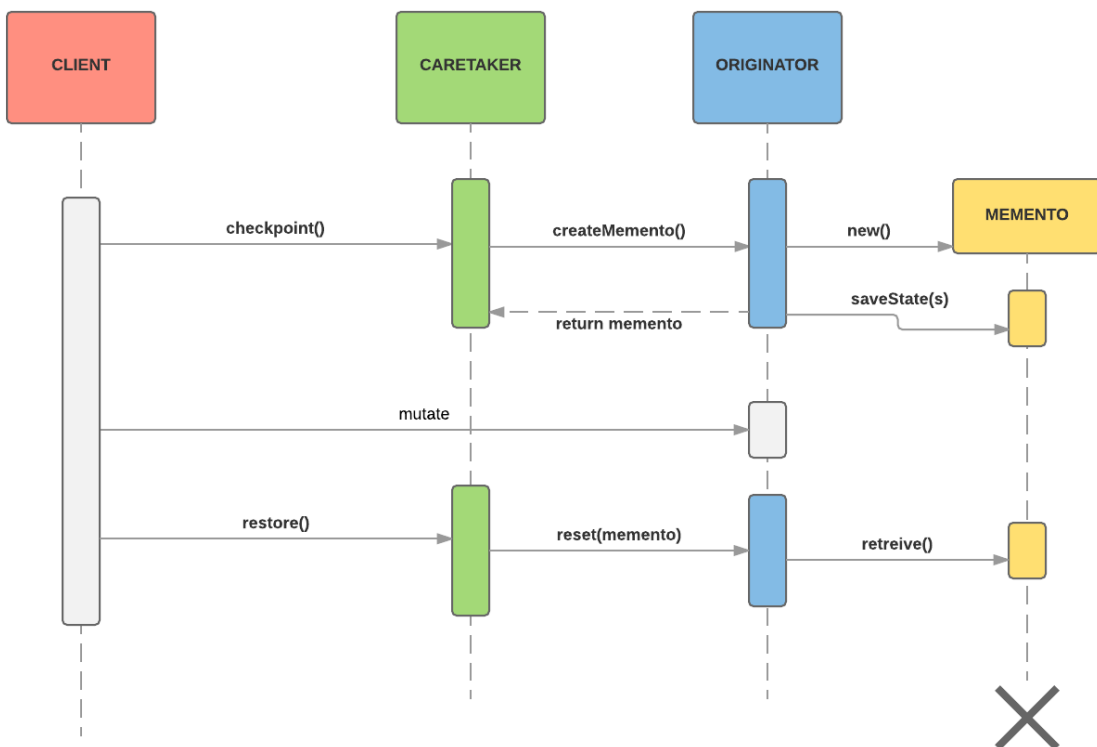
Command (DeleteCommand)



Prototype



Memento



Design Patterns

Memento: Handles Undo and Redo functionality when editing images

Prototype [+ Factory]: Clones (makes copy) of original image and creates new image for editing

Command: Handles functionality of JButtons in GUI

Composite: Organizes into and allows user to go through hierarchical collection of albums and images inside.

Patterns that we are under consideration:

MVC - On linking the GUI with Command and adding a Model.

Virtual Proxy: may having something to do with editing image or with displaying

Facade: We will probably use this if we ever run into any trouble with a lot of interfaces in our project. So for example if we want to provide our user with the cleanest experience we might consider using this, however at the current point we do not know if we will use so many interfaces.