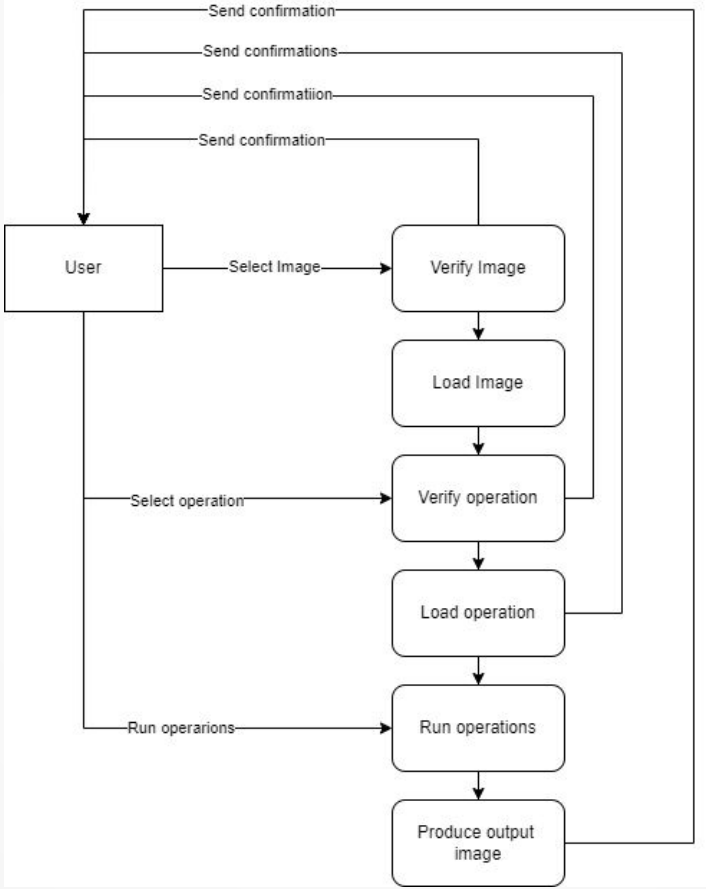
Photoformer *the tiny image editor*

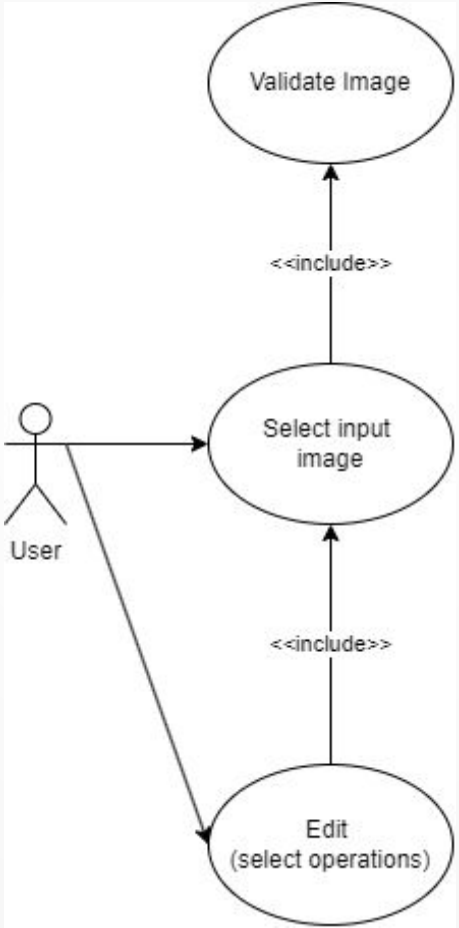
By Guillermo Ramos
and Alejandro López

Content

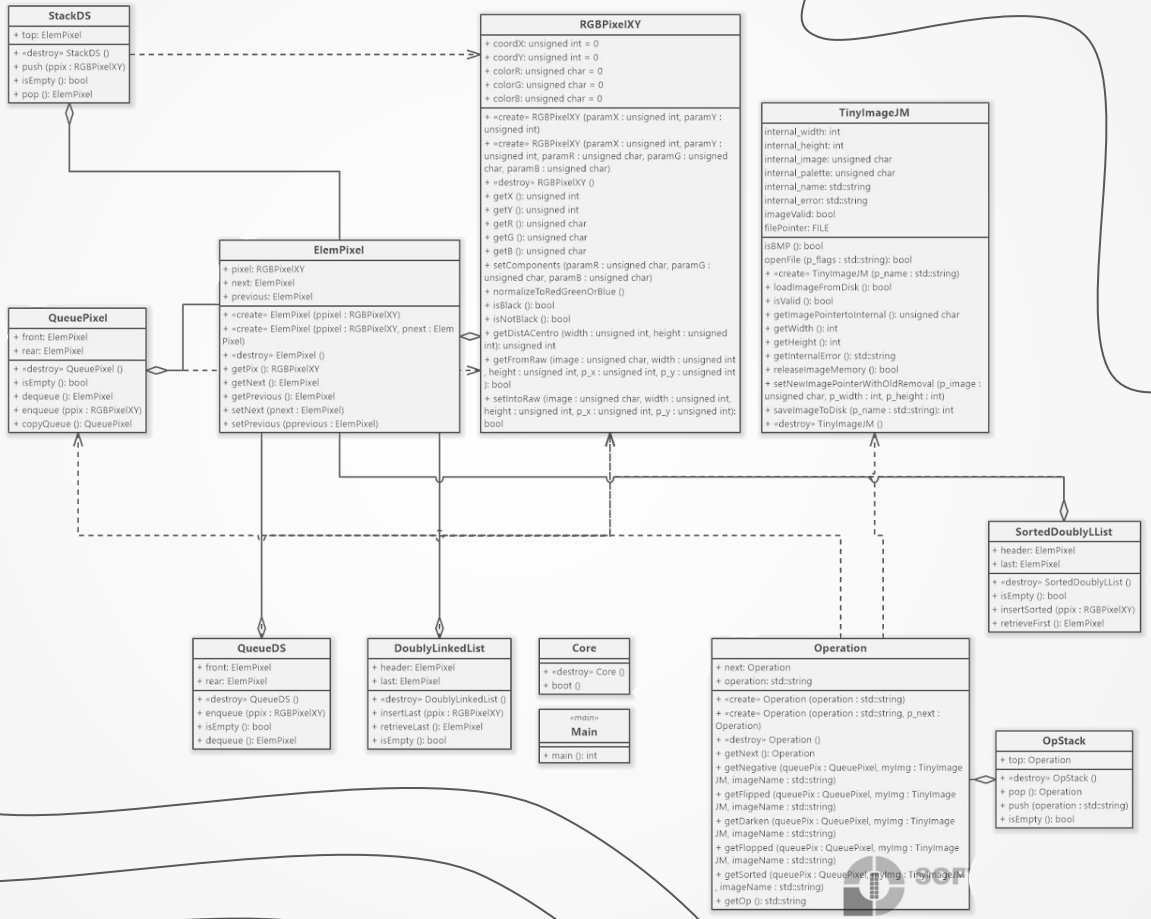
Data flow diagram



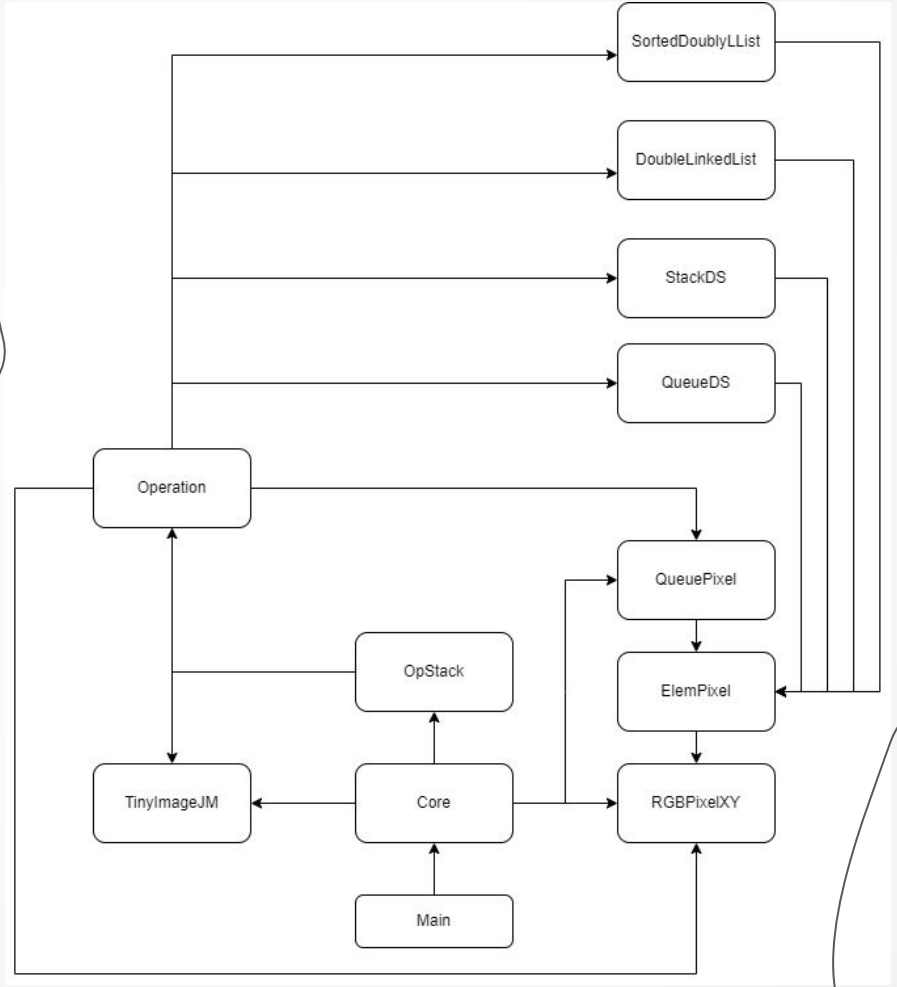
Use case diagram



Class diagram



Source file diagram



ADTs REASONING

getNegative - Queue

getDarken - Queue

getFlipped - Stack

getFlopped - Doubly Linked List

getSorted - Sorted Doubly Linked List

ADTs ADAPTATIONS: QUEUEPIXEL

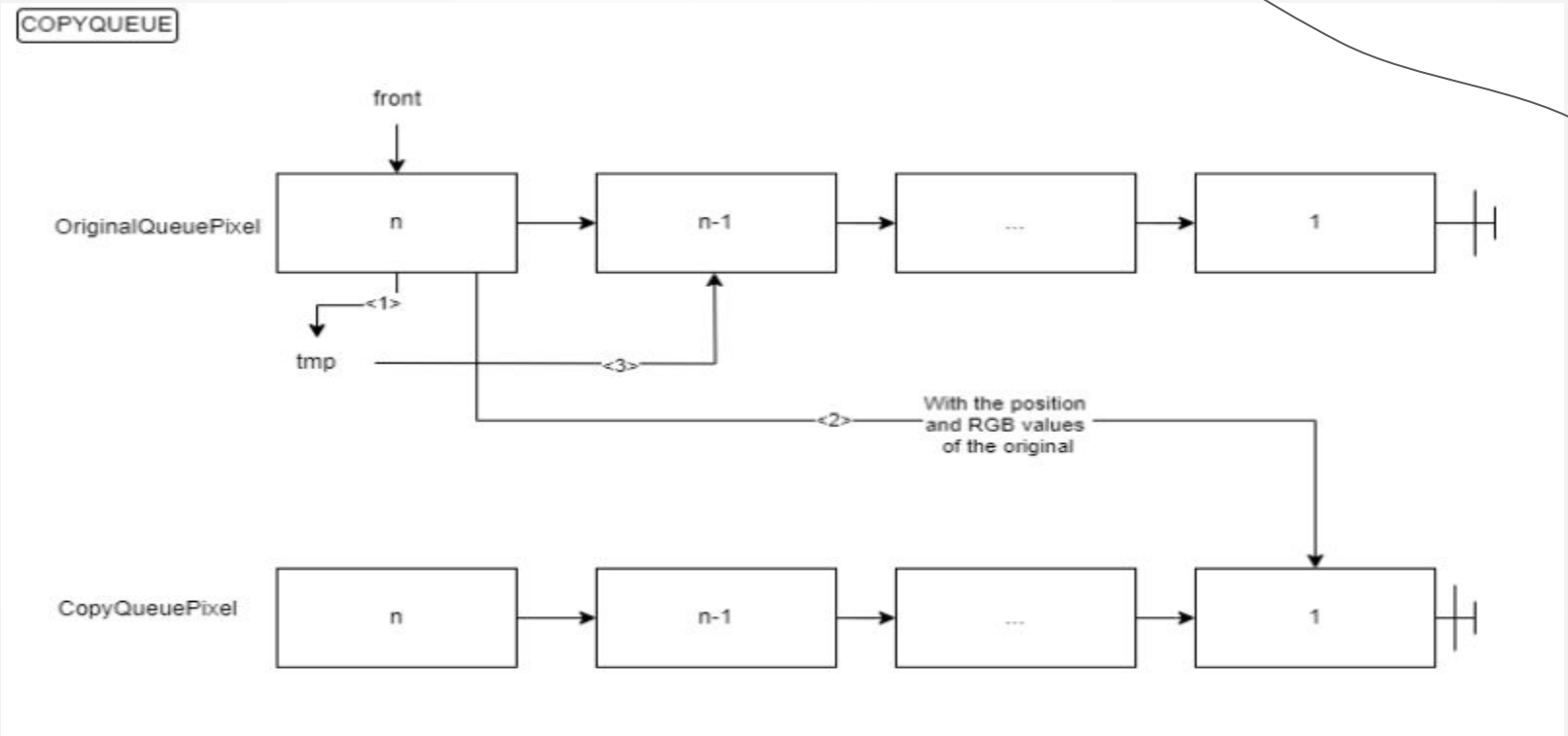
```
SPEC QUEUE[ITEM]
        GENRES queue, item
        OPERATIONS:
                enqueue: queue item -> queue
                dequeue: queue -> item
                front: queue -> item
                makenull: queue -> queue
                empty: queue -> boolean

ENDSPEC
```

```
SPEC QUEUE[ITEM]
        GENRES queue, item
        OPERATIONS:
                enqueue: queue item -> queue
                dequeue: queue -> item
                empty: queue -> boolean
                copyQueue: queue -> queue

ENDSPEC
```

ADTs ADAPTATIONS: QUEUEPIXEL



ADTs ADAPTATIONS: QUEUEDS

```
SPEC QUEUE[ITEM]
        GENRES queue, item
        OPERATIONS:
                enqueue: queue item -> queue
                dequeue: queue -> item
                front: queue -> item
                makenull: queue -> queue
                empty: queue -> boolean

ENDSPEC
```



```
SPEC QUEUE[ITEM]
        GENRES queue, item
        OPERATIONS:
                enqueue: queue item -> queue
                dequeue: queue -> item
                empty: queue -> boolean

ENDSPEC
```

ADTs ADAPTATIONS: OPSTACK

```
SPEC STACK[ITEM]
        GENRES stack, item
        OPERATIONS:
                push: stack item -> stack
                pop: stack -> item
                top: stack -> item
                makenull: stack -> stack
                empty: stack -> bool
ENDSPEC
```

```
SPEC STACK[ITEM]
        GENRES stack, item
        OPERATIONS:
                push: stack item -> stack
                pop: stack -> item
                empty: stack -> bool
ENDSPEC
```

ADTs ADAPTATIONS: STACKDS

```
SPEC STACK[ITEM]
        GENRES stack, item
        OPERATIONS:
                push: stack item -> stack
                pop: stack -> item
                top: stack -> item
                makenull: stack -> stack
                empty: stack -> bool
ENDSPEC
```

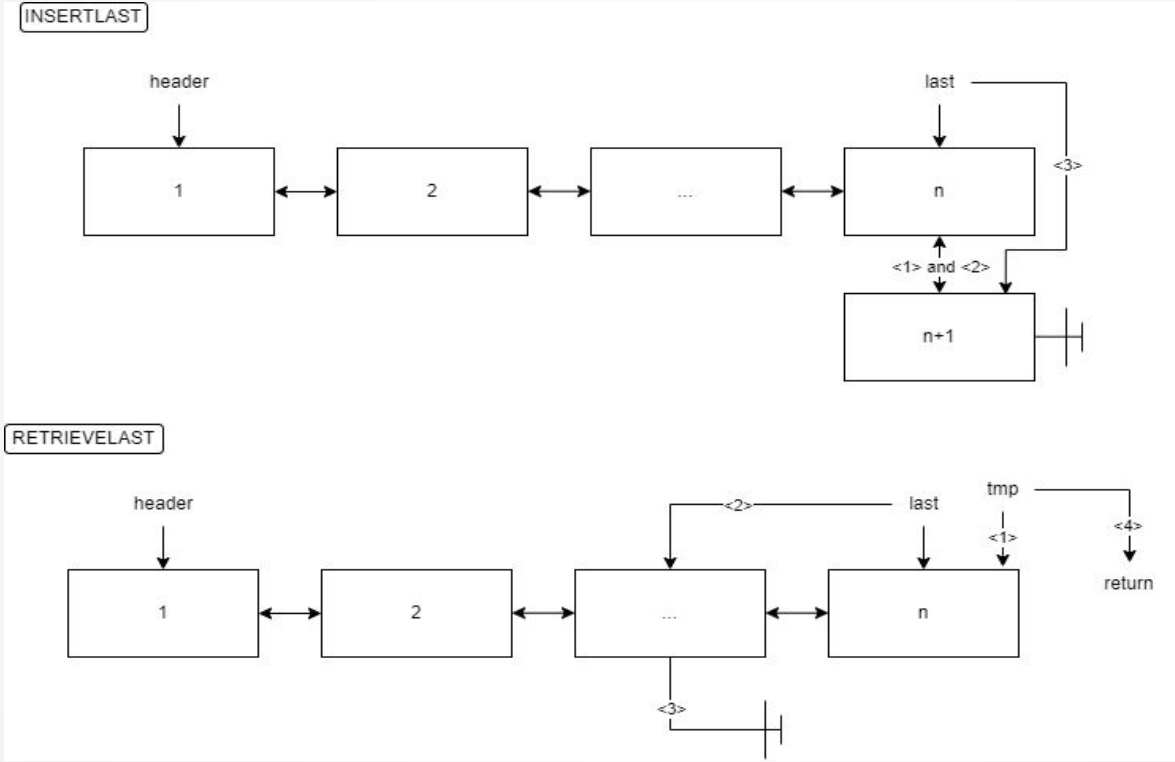
```
SPEC STACK[ITEM]
        GENRES stack, item
        OPERATIONS:
                push: stack item -> stack
                pop: stack -> item
                empty: stack -> bool
ENDSPEC
```

ADTs ADAPTATIONS: DOUBLYLINKEDLIST

```
SPEC LIST[ITEM]
        GENRES list, item, position
        OPERATIONS:
                insert: item position list -> list
                delete: position list -> list
                locate: item list -> position
                retrieve: position list -> item
                next: position list -> item
                previous: position list -> item
                makenull: list -> list
                empty: list -> bool
ENDSPEC
```

```
SPEC LIST[ITEM]
        GENRES list, item
        OPERATIONS:
                insertLast: item list -> list
                retrieveLast: list -> item
                empty: list -> bool
ENDSPEC
```

ADTs ADAPTATIONS: DOUBLYLINKEDLIST

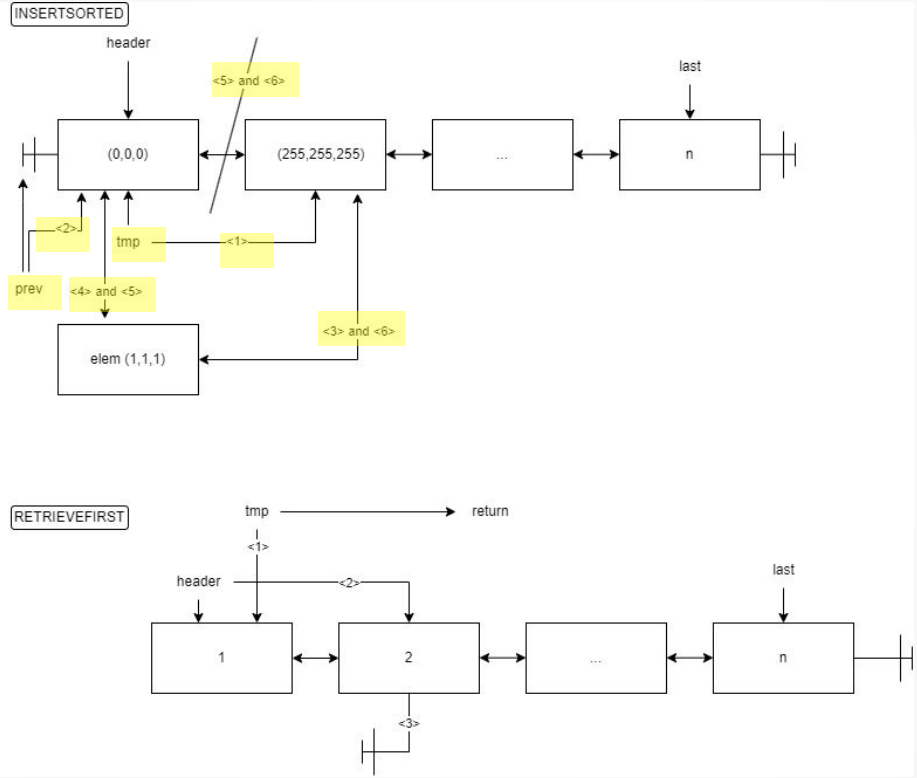


ADTs ADAPTATIONS: SORTED DOUBLYLINKEDLIST

```
SPEC LIST[ITEM]
        GENRES list, item, position
        OPERATIONS:
                insert: item position list -> list
                delete: position list -> list
                locate: item list -> position
                retrieve: position list -> item
                next: position list -> item
                previous: position list -> item
                makenull: list -> list
                empty: list -> bool
ENDSPEC
```

```
SPEC LIST[ITEM]
        GENRES list, item
        OPERATIONS:
                insertSorted: item list -> list
                retrieveFirst: list -> item
                empty: list -> bool
ENDSPEC
```

ADTs ADAPTATIONS: SORTED DOUBLYLINKEDLIST



CLASSES EXPLANATION

Core:
- void boot()

ElemPixel

Operation:
- void getNegative()
- void getFlipped()
- void getDarken()
- void getFlopped()
- void getSorted()

OpStack

QueueDS

QueuePixel:
- QueuePixel* copyQueue()

StackDS

DoublyLinkedList

SortedDoublyLList:
- void insertSorted()

RUNNING TIME EXPLANATION

LOADING PHASE

Pixel queue creation -> O(1)

Validation of the image format -> O(1)

Pixel enqueuing -> O(N)

USER INTERACTION PHASE

Insert operations in OpStack -> O(1)

Where N is the total number of pixels in the image

RUNNING TIME EXPLANATION

DATA PROCESSING PHASE

getNegative() -> O(N + N + N) -> O(3N) -> O(N)

getFlipped() -> O(N + N + N + N) -> O(4N) -> O(N)

getFlopped() -> O(N + N + N + N) -> O(4N) -> O(N)

getDarken() -> O(N + N + N + N) -> O(4N) ->O(N)

getSorted() -> O(N + N*N + N) -> O(2N + N^2) -> O(N^2)

CONCLUSIONS