5조

# Another Photo Editor

201411257 강정모
201411307 이한강
201611284 이유진

# Functional & non-Functional Requirements

추가 및 변경된 내용 : 파란 bold체로 서술함

[Functional Requirements] ⇒ **Text 기반 UI 에서 GUI로 변경. (Prototype Illustration 참고)**

**1. 해상도 조절**

**1.1. 자유 선택 : 사용자가 직접 가로 해상도와 세로 해상도를 지정하여 image의 해상도를 변경하고, 변경된 image와 변경 전 image를 함께 화면에 출력한다.**

**1.2. 비율 유지**

**1.2.1. 가로 해상도 지정 : 사용자가 직접 가로 해상도를 지정하면 자동으로 image의 가로 세로 해상도 비에 따라 세로 해상도가 정해지고 image의 해상도가 변경된다. 그리고 변경된 image와 변경 전 image를 함께 화면에 출력한다.**

**1.2.2. 세로 해상도 지정 : 사용자가 직접 세로 해상도를 지정하면 자동으로 image의 가로 세로 해상도 비에 따라 가로 해상도가 정해지고 image의 해상도가 변경된다. 그리고 변경된 image와 변경 전 image를 함께 화면에 출력한다.**

2. 회전

2.1. 왼쪽으로 회전

2.1.1. 90도 단위로 반복 : 입력을 줄 때마다 왼쪽으로 90도 → 180도 → 270도 → 360도(처음) 순으로 반복하여 image를 회전시키고 회전된 image를 변경 전 image와 함께 화면에 출력한다.

2.2. 오른쪽으로 회전

2.2.1. 90도 단위로 반복 : 입력을 줄 때마다 오른쪽으로 90도 → 180도 → 270도 → 360도(처음) 순으로 반복하여 image를 회전시키고 회전된 image를 변경 전 image와 함께 화면에 출력한다.

**2.3. 원하는 방향으로 0°~45° 각도 회전 : 사용자가 왼쪽과 오른쪽 중 원하는 방향을 선택 후 0°~45° 내에서 원하는 각도를 입력하여 image를 회전시키고, 회전된 image를 변경 전 image와 함께 화면에 출력한다.**

3. 대칭

3.1. 상하 대칭 : image를 세로 해상도의 중간값을 기준으로 대칭시키고 대칭된 image와 변경 전 image를 함께 화면에 출력한다.

3.2. 좌우 대칭 : image를 가로 해상도의 중간값을 기준으로 대칭시키고 대칭된 image와 변경 전 image를 함께 화면에 출력한다.

**4.DRM**

**4.1. AES256 암호화 : 사용자로부터 최대 256Byte의 string data를 입력 받아 sha256으로 key를 생성하고, 생성한 key를 가지고 AES256 암호화 알고리즘을 사용하여 현재 image를 암호화하여 저장한다.**

**4.2. AES256 복호화 : 사용자로부터 최대 256Bytes의 string data를 입력 받아 sha256으로 key를 생성하고, 생성한 key를 가지고 암호화된 image를 복호화하여 저장한다.**

5. 변환

5.1. jpg to bmp : jpg format의 image를 해상도와 컬러/흑백 속성을 유지하며 bmp format으로 변환한다.

5.2. jpg to png : jpg format의 image를 해상도와 컬러/흑백 속성을 유지하며 png format으로 변환한다.

5.3. png to bmp : png format의 image를 해상도와 컬러/흑백 속성을 유지하며 bmp format으로 변환한다.

5.4. png to jpg : png format의 image를 해상도와 컬러/흑백 속성을 유지하며 jpg format으로 변환한다.

5.5. bmp to png : bmp format의 image를 해상도와 컬러/흑백 속성을 유지하며 png format으로 변환한다.

5.6. bmp to jpg : bmp format의 image를 해상도와 컬러/흑백 속성을 유지하며 jpg format으로 변환한다.

**6. 잘라내기**

**6.1. 원하는 영역 자르기 : left-top pixel과 right-bottom pixel을 주어 image 내의 원하는 image 영역만 잘라내고, 잘라낸 image 영역과 기존의 image를 함께 화면에 출력한다.**

**7. 확대 및 축소**

**7.1. 확대 : image를 단계적으로 150% → 200% → 300% → 400% → 500% 까지 확대하여 화면에 출력한다.**

**7.1.1.화면 이동 : image를 확대한 경우 출력 창 크기를 초과하면 화면을 이동하며 확대된 image를 살펴볼 수 있게 해준다.**

**7.1.2. 축소 : image를 단계적으로 확대한 경우 다시 단계적으로 축소하여 image를 화면에 출력한다.**

8. 저장 및 불러오기, **실행 취소 및 재실행**

8.1. 저장 : 사용자가 원하는 파일을 원하는 file name 및 file format으로 원하는 위치에 저장한다.

8.2. 불러오기 : 사용자가 원하는 파일을 지정된 위치로부터 불러와 화면에 출력한다.

**8.3. 실행 취소 : 사용자가 실행한 동작을 취소시키고 한 단계 전으로 돌아가고, 해당 단계의 image를 화면에 출력한다.**

**8.4. 재실행 : 사용자가 실행 취소한 동작을 재실행 하고, 해당 단계의 image를 화면에 출력한다.**

[Non-functional requirements]

1. Performance

⇒ **사용자가 원하는 기능을 입력한 후 최대 5초 안에 결과를 출력한다.**

2. Usability

⇒ **그림판과 같은 단축 아이콘을 사용한다.**
**또 해당 아이콘에 마우스를 올리면 제공하는 기능에 관한 메시지를 띄워 단축 아이콘 및 기능에 대한 사용자의 이해를 돕고 편의성을 제공한다.**

**[Resource]**

**Pyqt 혹은 tkinter와 같은 GUI library 사용 예정.**

# Architecture Diagram

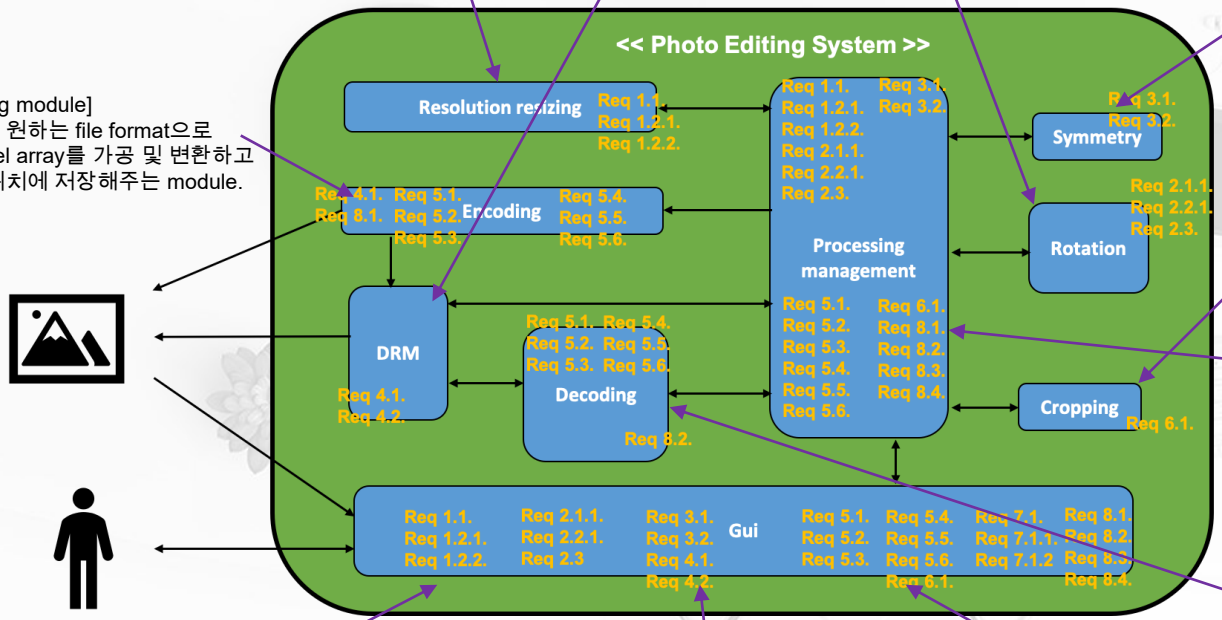
[DRM module]
~~sha-256 hash function을 사용해 입력 된 string 으로부터 256 Bit hash key를 생성하고~~
~~생성한, 혹은 입력 받은 Hash key를 이용해서 RGB pixel 값들에 noise를 넣거나,~~
~~반대로 Hash key를 이용해서 RGB pixel 값들의 noise를 filtering 하는 기능을 수행하는 module.~~
→ pixel noising이 아닌 암호화 사용으로 결정. 입력값은 string data로 한정.
내부적으로 sha256으로 입력값을 hash key로 바꾸어서 AES256 암호화 알고리즘 사용할 예정.

[Resolution resizing module]
입력받은 해상도로 RGB pixel array의 size를
변환하여 해상도를 바꿔주는 module.

[Rotation module]
RGB pixel array와 방향, 각도를 받아 회전시키는 module.

[Symmetry module]
RGB pixel array와 방향을 받아 대칭 처리하는 module.

[Encoding module]
사용자가 원하는 file format으로
RGB pixel array를 가공 및 변환하고
지정한 위치에 저장해주는 module.




[Cropping module]
Left-top pixel index와
right-bottom pixel index를 받아
RGB pixel array 중 일부를 잘라내는 module.

[Processing management module]
GUI를 통해 사용자의 입력을 받은 후 사용자의
입력과 RGB pixel array를 적절한 처리 module로
넘겨주고, 다시 그 결과를 받아 관리하는 module.
전체적인 process의 history도 관리함.
⇒ GUI module을 추가하여
user 입력을 담당하는 기능을 분리함.

[Decoding module]
입력한 image의 확장자에 따라 header를 제거하고,
압축된 data를 RGB pixel 값들로 바꾸는 Module.

[Enlarging module]
~~사용자의 입력에 따라 image를 화면에~~
~~정해진 배율로 확대해주는 module.~~
→ GUI에서 처리해야 하는 사항 ⇒ 삭제
→ mapping이 Gui module로 변경

[Shrinking & moving module]
~~확대된 image를 다시 축소시키거나, 이동할 수 있게 해주는 module.~~
→ GUI에서 처리해야 하는 사항 ⇒ 삭제
→ mapping 이 Gui module로 변경

[Screen module]
~~RGB pixel array를 화면에 출력해주는 Module.~~
⇒ GUI module로 변경

# System Testcases

Testcase Set :

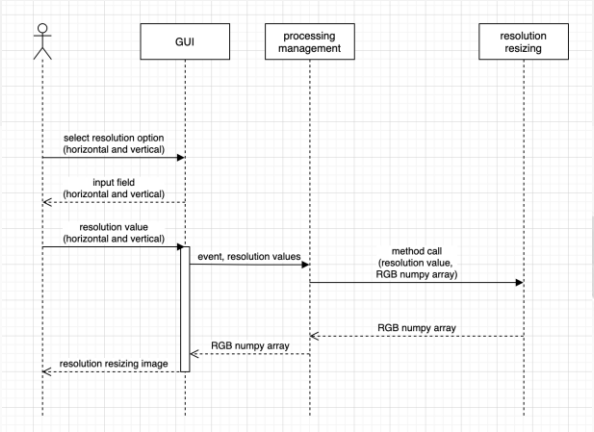
(1) 800 * 600 jpg color image
(2) 1280 * 1660 jpg color image
(3) 64 * 64 png color image
(4) 1920 * 1080 png color image
(5) 735 * 564 bmp gray-scale image
(6) 234 * 639 bmp color image

| Spec | input | Expected Output |
|---|---|---|
| 1.1 | (3), 가로 520, 세로 230 해상도 | 우측 출력 창에 520 * 230 해상도를 가진 png color image 출력 |
| 1.2.1. | (2), 가로 1600 해상도 | 우측 출력 창에 1600 * 2152 해상도를 가진 jpg color image 출력 |
| 1.2.2. | (4), 세로 539 해상도 | 우측 출력 창에 958 * 539 해상도를 가진 png color image 출력 |
| 2.1. | (6), 왼쪽 회전 1번 | 우측 출력 창에 왼쪽으로 90° 회전된 639 * 234 bmp color image 출력 |
| 2.2. | (5), 오른쪽 회전 3번 | 우측 출력 창에 오른쪽으로 270° 회전된 564 * 735 bmp gray-scale image 출력 |
| 2.3. | (4), 왼쪽 회전 27° | 우측 출력 창에 왼쪽으로 27° 회전된 1131 * 636 png color image 출력 |
| 3.1. | (1), 상하 대칭 | 우측 출력 창에 상하 대칭된 800 * 600 jpg color image 출력 |
| 3.2. | (5), 좌우 대칭 | 우측 출력 창에 좌우 대칭 된 735*564 bmp gray-scale image 출력 |
| 4.1. | (2), hello, world! 입력 | 다른 image editor에서는 열리지 않는 암호화된 (2) image file |
| 4.2. | 4.1.의 expected output, hello, world! 입력 | 다른 image editor에서도 열리는 복호화된 (2) image file |
| 5.1. | (1), jpg → bmp 선택 | 800 * 600 해상도의 bmp color image |
| 5.2. | (2), jpg → png 선택 | 1280 * 1660 해상도의 png color image |

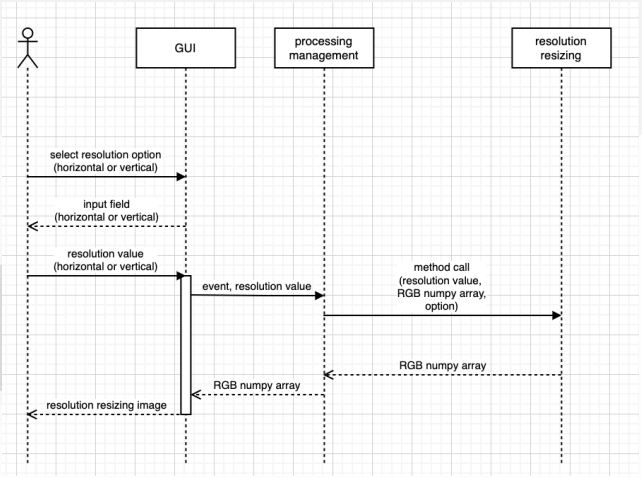
| Spec | input | Expected Output |
|---|---|---|
| 5.3. | (3), png → bmp 선택 | 64 * 64 해상도의 bmp color image |
| 5.4. | (4), png → jpg 선택 | 1920 * 1080 해상도의 jpg color image |
| 5.5. | (5), bmp → png 선택 | 735 * 564 해상도의 png gray-scale image |
| 5.6. | (6), bmp → jpg 선택 | 234 * 639 해상도의 jpg color image |
| 6.1. | (1), (1) 내부를 마우스 커서로 drag 한 영역 | 우측 출력 창에 Drag한 영역 만큼의 해상도를 가지는 (1) 내부의 jpg color image 출력 |
| 7.1. | (3)을 띄운 출력 창 선택 3번 확대 | 현재 선택된 출력 창에 (3)이 300% (3배) 확대된 image 출력 |
| 7.1.1. | 키보드 → 화살표 | 확대된 image가 출력 창 크기를 초과하는 경우 확대된 (3)의 오른쪽 부분으로 이동하여 출력 |
| 7.1.2. | (3)을 3번 확대한 출력 창 선택 1번 축소 | 현재 선택된 출력 창에 (3)이 200% (2배) 확대된 Image 출력 (확대 단계 1단계 감소) |
| 8.1. | 4.2. expected output, 원하는 directory, file name, file extension | 4.2.의 expected output을 지정한 directory에 지정한 file extenstion format과 file name으로 저장 |
| 8.2. | (6), (6) file이 있는 directory 경로 | (6) Image file을 우측 출력 창에 출력 |
| 8.3. | 6.1. expected output | 좌측 출력 화력 6.1. expected output, 우측 출력 화면 2.3. (1) 출력 |
| 8.4. | 8.3. expected output | 좌측 출력 화면 2.3. (1) 출력 우측 출력 화면 6.1. expected output |

# System Sequence Diagram (1)

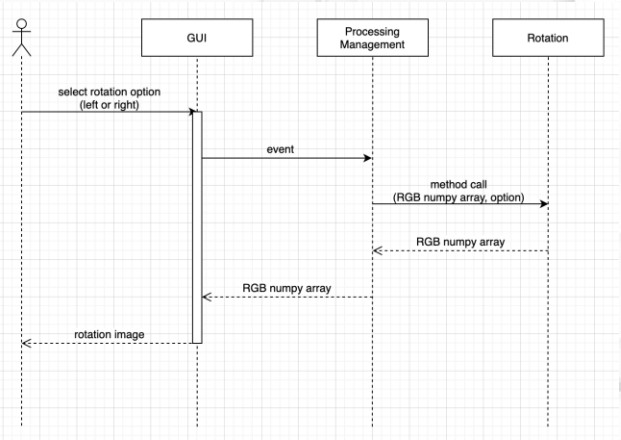
Req 1.1.



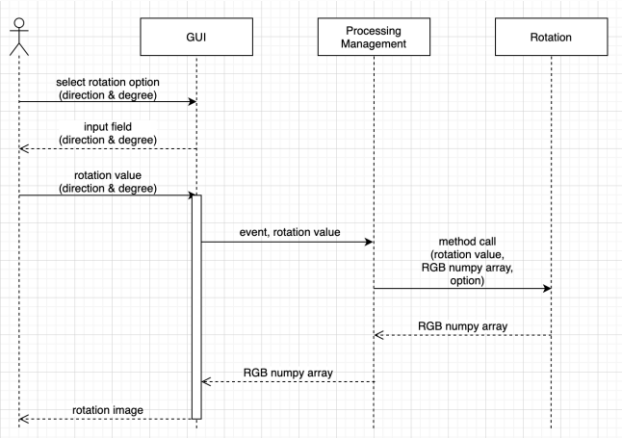

Req 1.2.1.
1.2.2.




Req 2.1.
2.2.

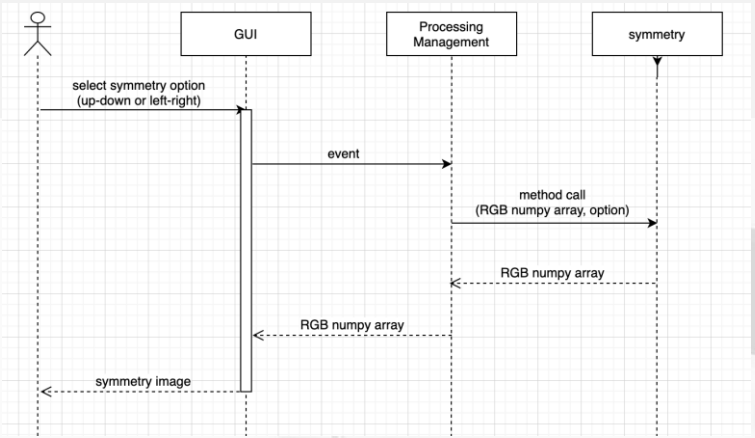



Req 2.3.

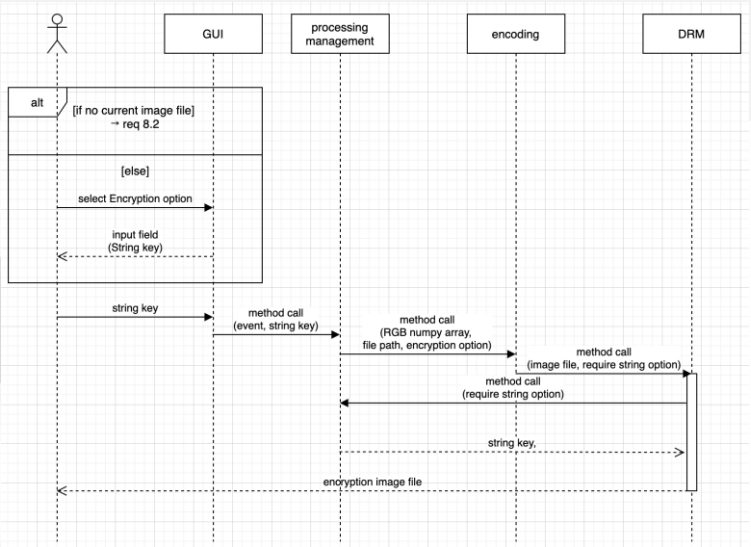


# System Sequence Diagram (2)

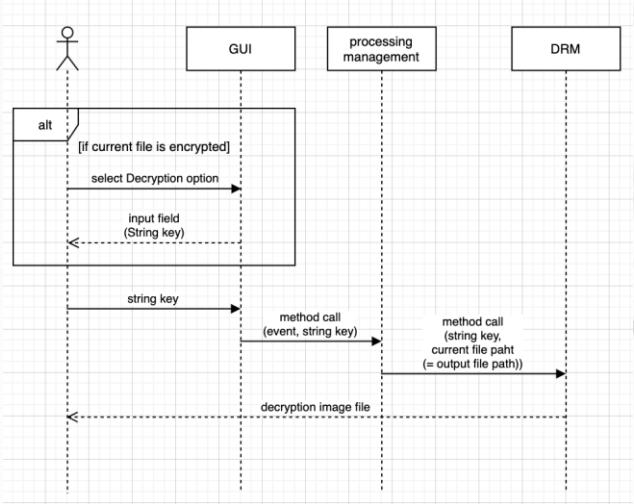
Req 3.1.
3.2.




Req 4.1.

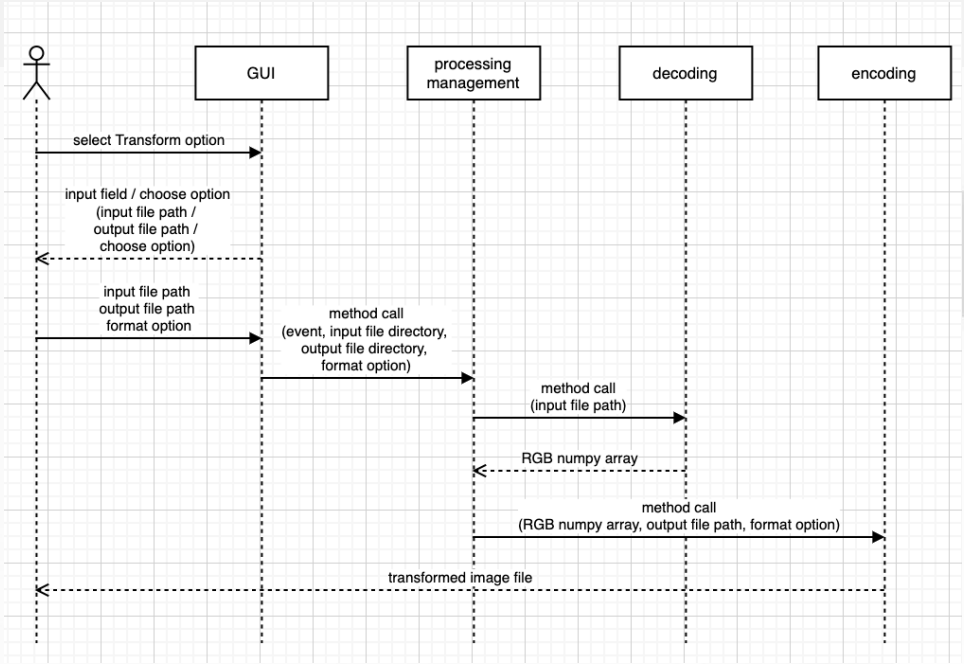



Req 4.2.

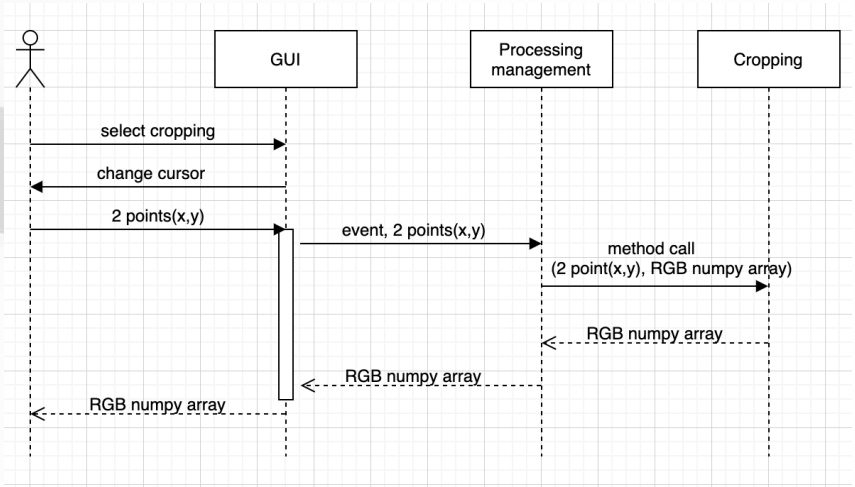


# System Sequence Diagram (3)

Req 5.1. ~ 5.6.

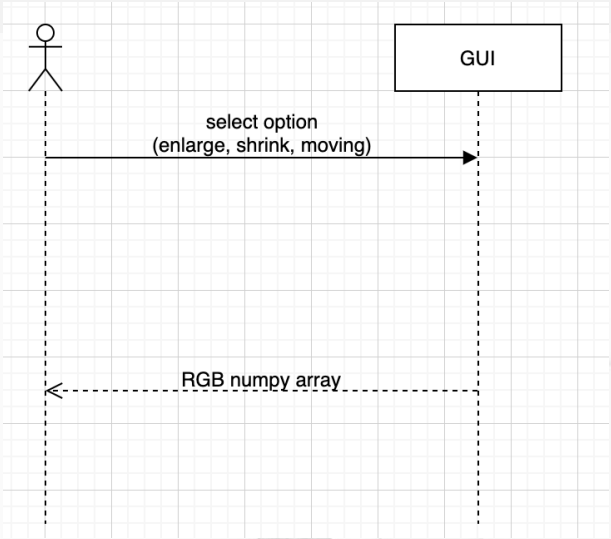



Req 6.1.

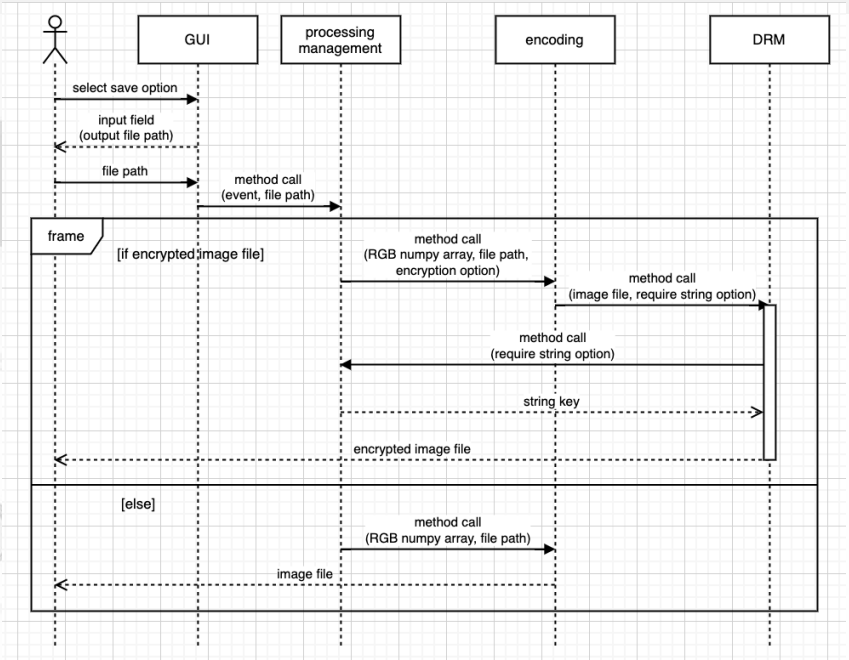


# System Sequence Diagram (4)

Req 7.1.

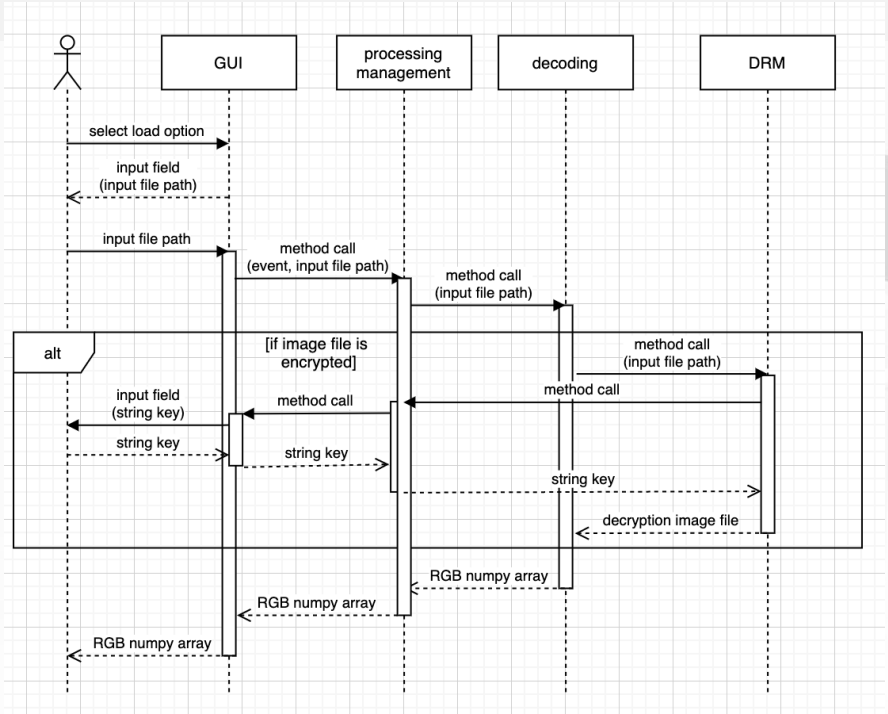



Req 8.1.

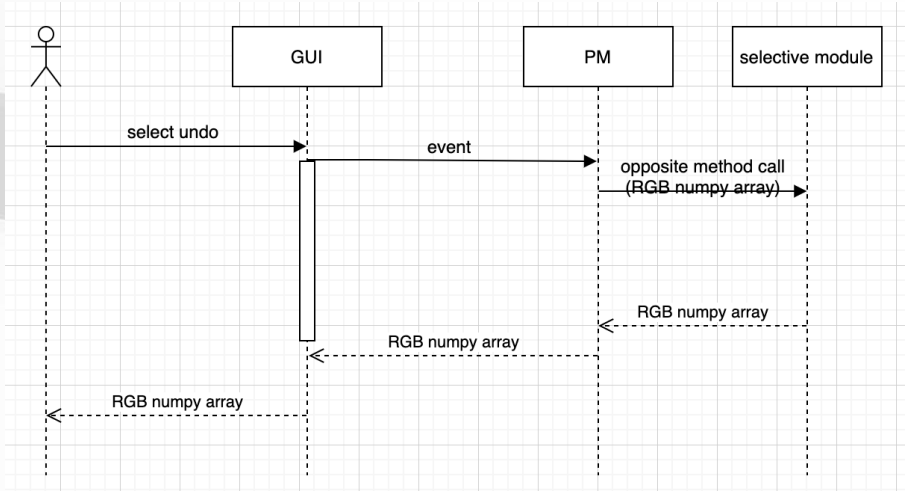


# System Sequence Diagram (5)

Req 8.2.

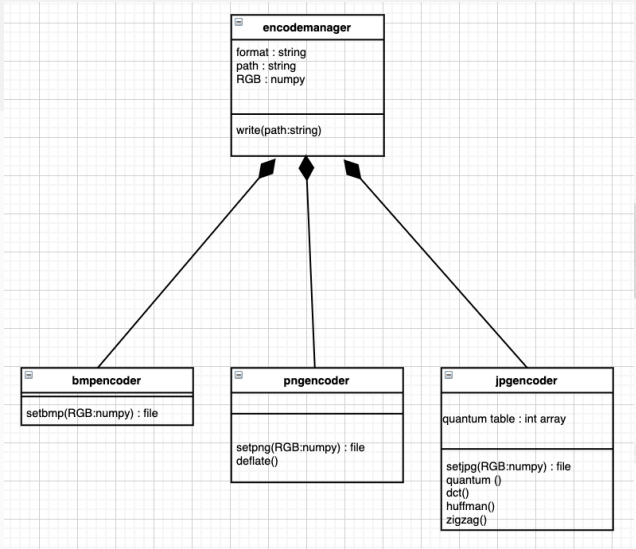



Req 8.3.
8.4.

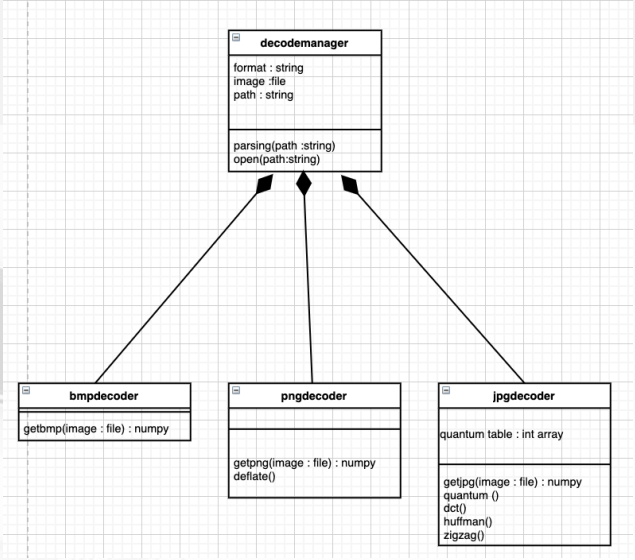


# Class Diagram (1)

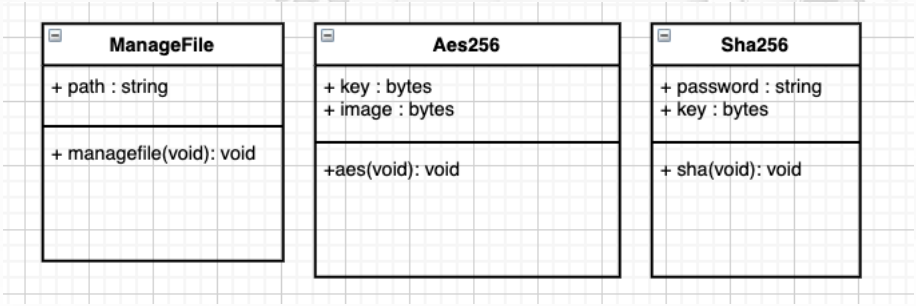
**encoding**



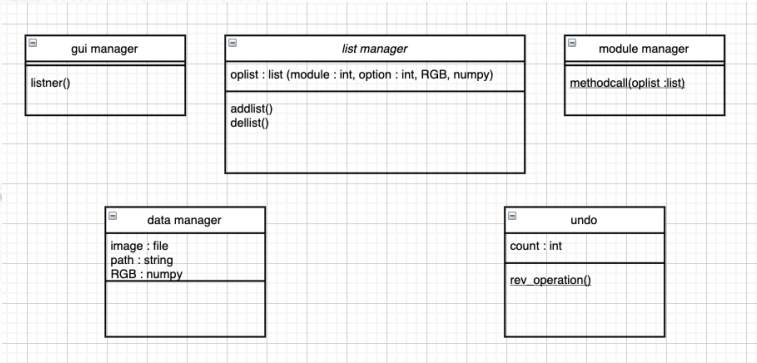

**decoding**




**DRM**

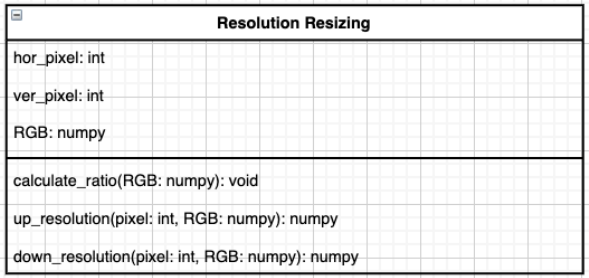



PM

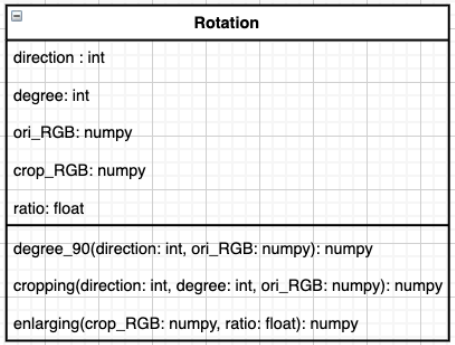


# Class Diagram (2)

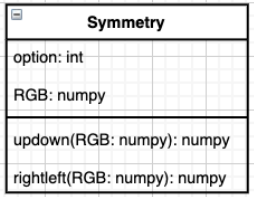
resolution



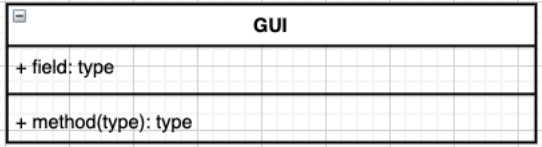

rotation



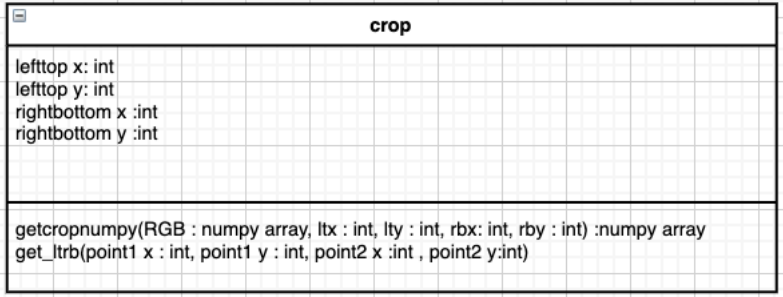

symmetry




GUI




cropping



# Traceability Matrix (1)

| seq | Operation | Method | Class |
|---|---|---|---|
| 1.1 free resolution | Select_resolution_option | Calculate_ratio(RGB: numpy): void | Resolution |
| 1.2 selected resolution | Method_call() | Up_resolution(pixel: int, RGB: numpy) : numpy | |
| 2.1 selected rotation | Method_call() | Down_resolution(pixel: int, RGB: numpy): numpy | |
| 2.3 free rotation | Select_resolution_option | Degree_90(direction: int, ori_RGB: numpy): numpy | Rotation |
| 3 symmetry | Method_call() | Cropping(direction: int, degree: int, ori_RGB: numpy): numpy | |
| | Method_call() | Enlarging(crop_RGB: numpy, ratio:float): numpy | |
| | Select_rotation_option | Updown(RGB: numpy): void | Symmetry |
| | Method_call() | Rightleft(RGB: numpy): numpy | |
| | Method_call() | Listner(): void | GUI manager |
| | Select_rotation_option | | |
| | Method_call() | | |
| | Method_call() | | |
| | Select_rotation_option | | |
| | Method_call() | | |
| | Method_call() | | |
| | Select_symmetry_option | | |
| | Method_call() | | |
| | Method_call() | | |

# Traceability Matrix (2)

| seq |
| --- |
| 4.1 encryption |
| 4.2 decryption |
| 5 transform |
| 6.1 cropping |
| 7.1 enlarging |
| 8.1 file store |
| 8.2 file load |
| 8.3 undo |

| Operation - input |
| --- |
| Select_encruption_option |
| Method_call() |
| Method_call() |
| Method_call() |
| Method_call() |
| Select_decryption_option |
| Method_call() |
| Method_call() |
| Select_transform_option |
| Method_call() |
| Method_call() |
| Method_call() |
| Select_cropping_option |
| Method_call() |
| Method_call() |
| Select_save_option |
| Method_call() |
| Method_call() |
| Method_call() |
| Method_call() |
| Select_load_option |
| Method_call() |
| Method_call() |
| Method_call() |
| Method_call() |
| Method_call() |
| Select_undo_option |
| Method_call() |
| Reverse_method_call |

| Method | Class |
| --- | --- |
| Aes(): void | AES256 |
| Managefile(): void | ManageFile |
| Sha(): void | Sha256 |
| Listener(): void | GUI manager |
| Addlist(): void | List manager |
| Dellist(): void | |
| Methodcall(oplist: list): void | Module manager |
| Rev_operation(): void | undo |
| getcropnumpy(RGB : numpy array, ltx : int, lty : int, rbx: int, rby : int) :numpy array | crop |
| get_ltrb(point1 x : int, point1 y : int, point2 x :int , point2 y:int): void | |
| Parsing(path: string): void | Decode manager |
| Open(path: string): void | |
| Getbmp(image: file): numpy | bmp decoder |
| Getpng(image: file): numpy | Png decoder |
| Deflate(): void | |
| Getjpg(image: file): numpy | jpg decoder |
| Quantum(): void | |
| Dct(): void | |
| Huffman(): void | |
| Zigzag(): void | |
| Write(path: string): void | Encode manager |
| Setbmp(RGB: numpy): file | Bmp encoder |
| Setpng(RGB: numpy): file | Png encoder |
| Deflate(): void | |
| Setjpg(RGB: numpy): file | Jpg encoder |
| Quantum(): void | |
| Dct(): void | |
| Huffman(): void | |