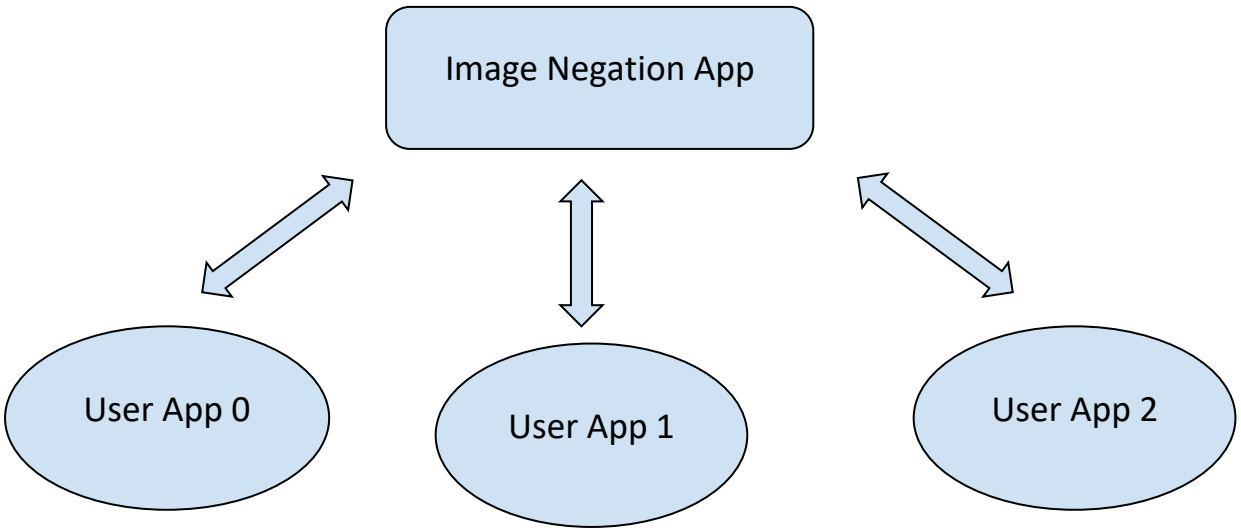
Implement a Real Time Image Negation Application

Constraints:

- Multiple users (other applications) will send image buffer to the application and the application will process and send back the processed output.
- Two or more users should be able to communicate with the image negation application at a same time (proper locking mechanism should be deployed).
- Use only greyscale image



Milestones:

1) Basic Image Negation Application
   a. Read image from disk
   b. Process
   c. Store the output image on disk
2) Image Negation Application responding to one user application
   a. User will read image from disk and send to Image Negation Application
   b. Image Processing done by Image Negation Application
   c. User should receive the processed output and store it on disk
3) Image Negation Application responding to multiple user application
   a. User will read image from disk and send to Image Negation Application
   b. Image Processing done by Image Negation Application. User Applications should be synchronized across them while accessing the Image Negation application.
   c. User should receive the processed output and store it on disk

Things will be considered on priority order:

- **Output Correctness**
- **Data Transfer Speed**
- **Performance** - How fast the Image Negation Application can receive / process / send
- **Race Condition Avoidance**
- **Operational Bandwidth -** How many users can be handled at a time

Note :

Use STB Image Library for reading and writing images as follows:

```
#define STB_IMAGE_IMPLEMENTATION
#include "stb_image.h"
#define STB_IMAGE_WRITE_IMPLEMENTATION
#include "stb_image_write.h"

int width, height, channel;
/* reading image from disk as greyscale */
unsigned char* image_data = stbi_load( "myimage.png", &width, &height, &channel, 1);

/* stroring the image on disk */
stbi_write_png("output_image.png", width, height, channel, image_data, width*channel);

stbi_image_free( rgb );
```