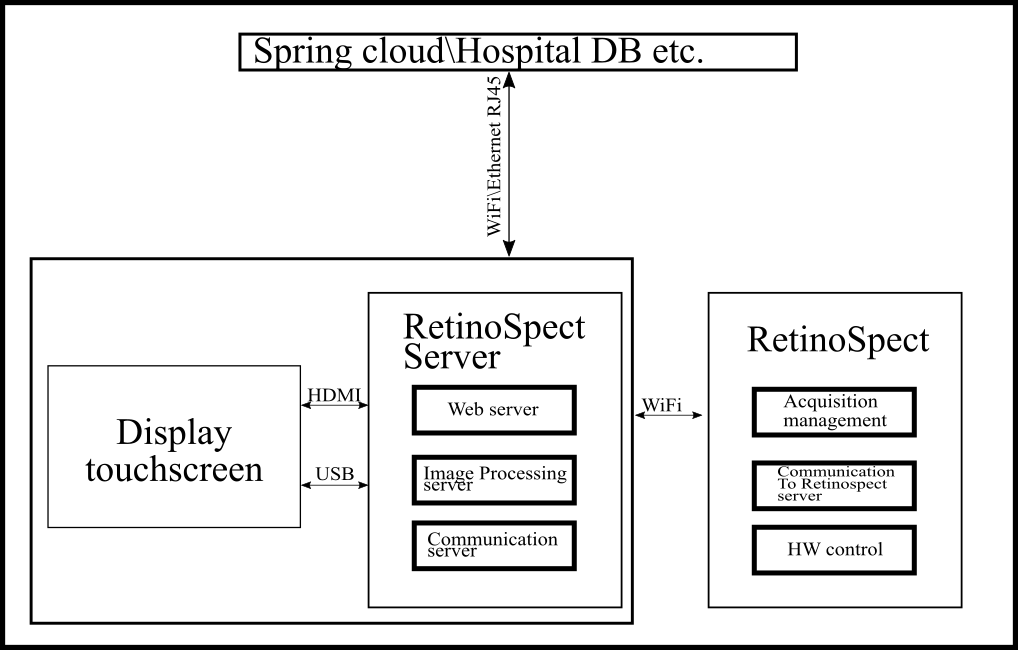
*SRS 001 Retino Spect*

*Author: Avihu Gamliel*

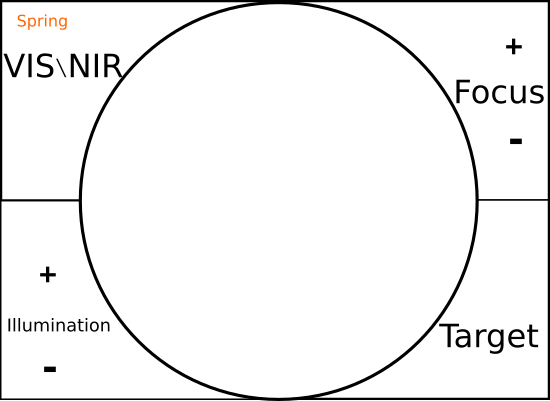
*28.1.2020*

***RetinoSpect* – Software Requirements Document**

SBV Multi-Spectral Retinal Imager should be part of the ophthalmic equipment used by Ophthalmologists and placed on their desk beside slit-lamp etc (We refer the reader to Apendix A).

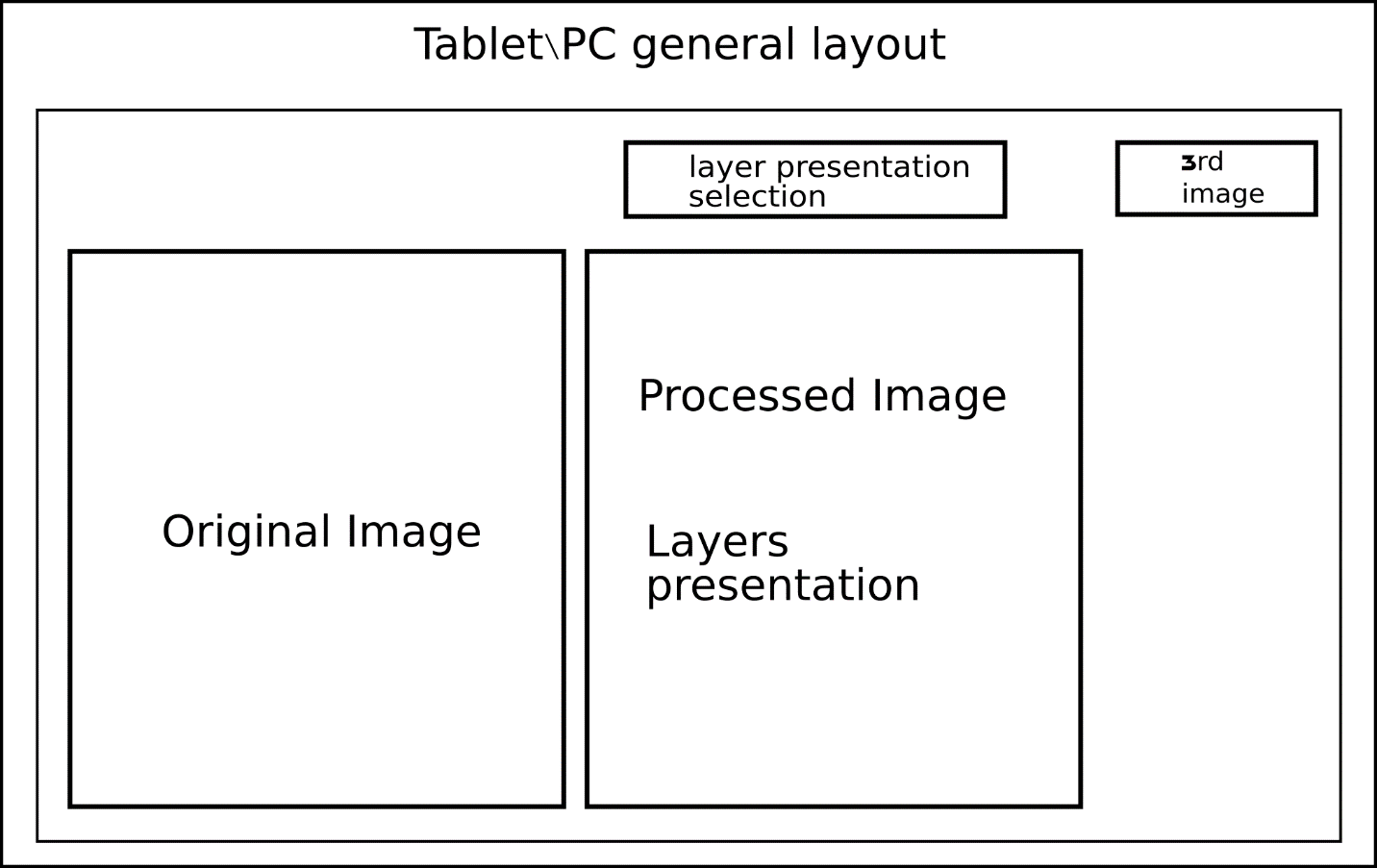
In order to understand SW System general layout we refer to the following diagram which will be detailed hereinafter. 

1. RetinoSpect - SVB Retinal Imager
   1. General description – SVB’s retinal imaging unit is the Fundus camera of SVB system. The unit is a wireless unit that communicate with external devices for display, image processing and storing information. The unit placed on the Ophthalmologist examination desk and has thin profile (in order to fit space aside Slitlane\Slit Lamp and other examination devices). The RetinSpect unit captures the images and send them for processing and display at external devices (Retino Server unit or PC).
   2. Components –
      1. Algorithms – Since RetinoSpect is actually a fundus camera, several algorithms are implemented at the unit processor for RT use, such as: auto focus, Auto illumination, stabilizers, etc.
         1. Auto focus, auto illumination calibration – C++ OpenCV – implementation of algorithms. Focus measuring and focus adjustments (moving focal point forward and backward).
            1. Focus measuring – local-contrast method – stable method.
            2. Illumination measuring – histogram entropy – this method found to have false measuring while used as focus ref, however it is one of the best to evaluate optimal illumination via contrast measuring.
            3. Focus algorithm – already implemented in C# - should be implemented in C++
         2. Partial auto focus mode – half button press – starts as auto focus, waits for user manual capturing trigger.
         3. Manual focus and illumination mode – starts as auto focus – enables manual fine tune.
      2. HW control – used to control illumination\flash intensity and duration, focus step motor, camera physical triggering (and acquiring process triggering), illumination modes memory, battery status, communication status, fixation target physical control, capturing button etc.
         1. Illumination control – driver with API in C++ – illumination control will adjust IR, VIS and spectral illumination intensity, duration and color.
         2. Focus Control – driver with API in C++ and python – two general focus modes (auto, manual) will be available with several properties. The focal point can be set manually (physically adjustment) as well as automatically. Focal wavelength\channel (RGB imaging) can be set as well.
         3. Image Capturing control – camera API in C++ – setting frame rate, exposure, disabling automatic properties such as AGC, WB, BLC, etc. image capture triggering, etc.
         4. Communication control – configuring and triggering external listening devices (RetinoSpect server\PC see schema 1), Wi-Fi\Ethernet\Bluetooth communication configuring.
         5. Battery indicator – charging indicator, alerts, etc.
         6. Technician Bluetooth identifier – opens technician screen and controls.
         7. Communication with HW controller – C++ - via serial port – RS232, I2C optional.
         8. Please refer to full functionality at HW specifications document, and briefAPI excel.
      3. UI – the user interface will consist of displaying live image, control buttons and menus, inner target display etc. Since local display screen is small (3.5’’) it affects the UI and display resolution, as well as layout.
         1. General layout (functionality will be detailed hereinafter):



* + - 1. Focus – focus adjustment – mode and direction
      2. Illumination –adjustment - intensity and duration.
         1. Upon central click on illumination button – illumination intensity
         2. Otherwise – illumination duration.
      3. VIS\NIR - illumination mode
      4. Target – display fixation target for the patient.

1. Local Display\ PC – Display and PC has similar functionality, both are configured as web application, hence they are cross-platform, and saves double programming. While on local display (that is connected directly to RetinoSpect Server) the application will be presented in browser’s kiosk view, on the PC the user will open the application as a browser site.
   1. Operating system – Ubuntu Linux –
      1. On RetinoSpect Server – aarch64 (arm64)
      2. On SVB Server – x64
      3. Regarding SVB implementation - the difference between OS is compilation mode only.
   2. Web application
      1. Schematic layout:



* + 1. Offered design pattern – MVVM family.
    2. Any displayed image can be modified – zoom in\out, translation, brightness.
    3. 3rd image auto-registration with previous image – using Spring’s registration algorithm.
    4. All displayed images respond simultaneously to dragging and zooming in\out operations, and their combinations.
       1. Providing reset option to default view prior user’s modifications.
    5. Option of image focusing:
       1. Double click option to highlight one of the images and view it in full screen mode. Double clicking on screen again return to previous\default layout (i.e. 2-3 images layout).
    6. Layers presentation – blood vessels, blood leaks, abnormalities, etc. derived by the algorithm components.
    7. Patient’s details input –
       1. UI input – virtual\physical keyboard input
       2. External devices input – magnetic card reader, 2D barcode reader, etc.
       3. External communication input – external server\PC input.
    8. Other system adjustments:
    9. Memory management –
       1. predefined buffer size
       2. synchronizing data with server
       3. after verifying that data uploaded to server\local server\PC deleting it from local unit\s including Imager (i.e. send command to imager).
    10. Parameters and algorithm parameters adjustment –
        1. Only service authorized can modify algorithm parameters.
        2. Authorization via service code or Bluetooth\NFC identifier etc.
        3. System parameters – specific authorization –
           1. Base illumination and focus parameters
    11. System configuration and communication –
        1. Dialog for communication configuration with capturing device.
        2. Dialog for communication configuration with local server\DB
        3. Alert communication failure in semi-RT.

1. Web server – Node JS – currently chosen platform – can run on arm64 – Ubuntu – cross platform, easy installation and usage.
2. Processing Server – The processing server is logic unit that provides the following processing functionalities:
   1. Blood vessels extraction – given spectral image based on SVB algorithm the blood vessels map is extracted.
      1. C++ openCV implementation.
      2. openCV –CUDA optimization – wherever applied.
      3. We should check an option of 3rd party production of openCV – Cuda compatible methods for basic image processing methods such as: convolution, morphological methods (erode\dilate), etc.
      4. Currently implementation is as singleton classes (private constructor and static pointer to class object) that can be easily modified to standard class structure.
   2. Oximetry extraction - – given spectral image based on SVB algorithm the oximetry map is extracted.
      1. Same configuration as at 3.1.
   3. Image registration – whenever we would like to present either older patient acquired data (such as traditional Angiography, medical history images, etc.) or older images that we captured, aligned together we need to register the images.
      1. SimpleElastix – C – open source library.
   4. Other feature extraction – such as blood leaks detection, pathology analysis and AI
      1. Same as 3.1
   5. The processing server may be used to analyze older images and blood vessels extraction of older images by request.