University of Massachusetts, Amherst
College of Engineering

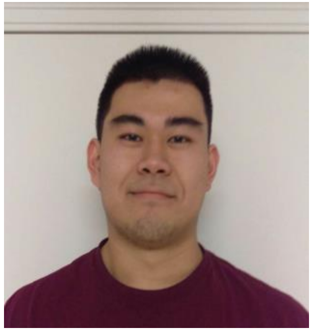
ITS LIT

Mid-Year Design Review
Senior Design Project ‘17

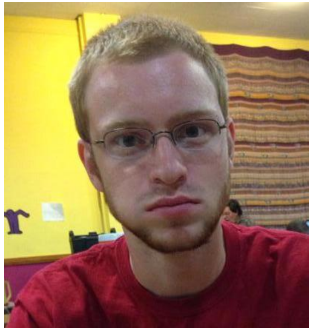
Department of Electrical and Computer Engineering

Meet The Team

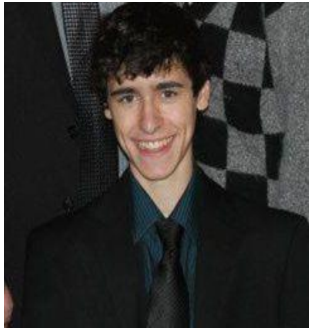
Advisor:
Professor David McLaughlin



Tommy Zhen
CSE



Michael Polin
CSE

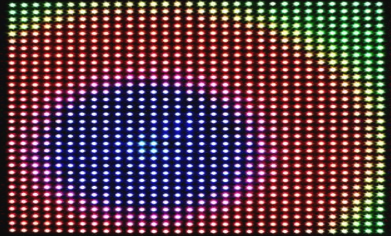


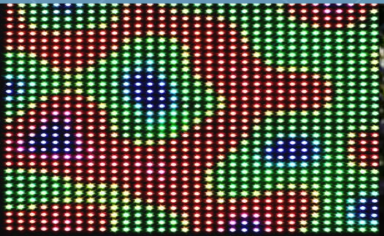
Patrick Browne
EE

Varun Menon
EE

Emma Bryce
EE

ITS LIT

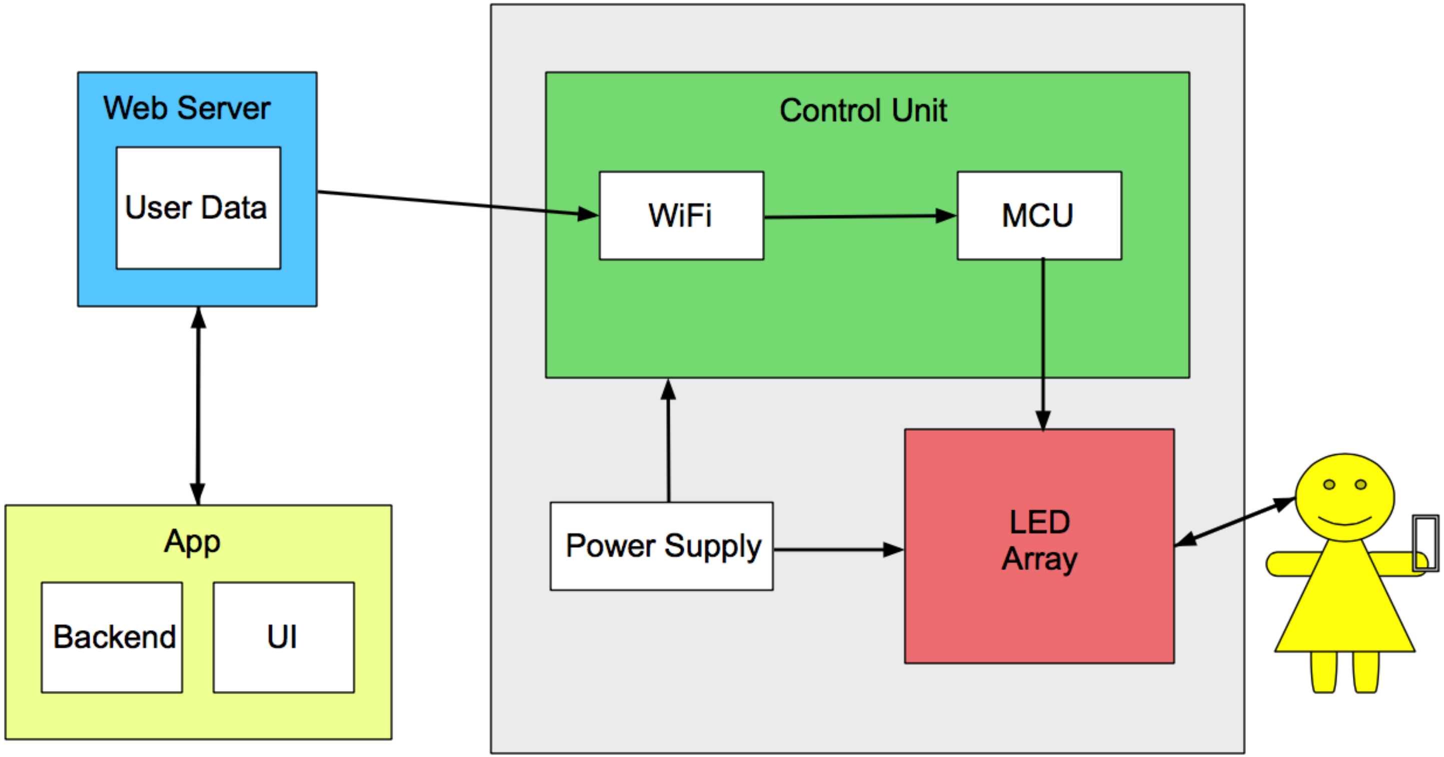




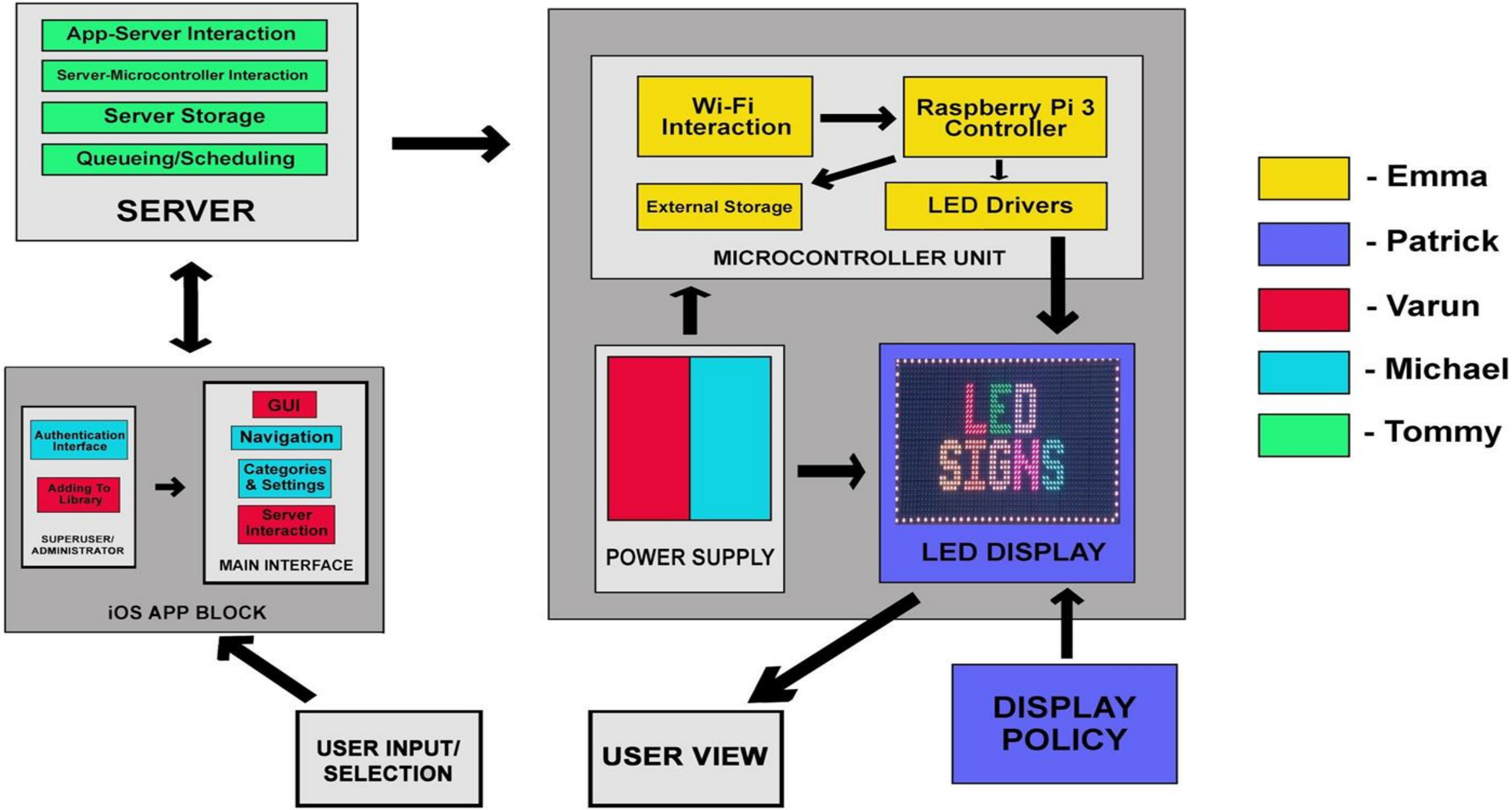
Top Level Requirements

- Visible by many people simultaneously (~100's)
    - Visibility out to 10's of meters at night (stretch goal 60m)
    - Show a selection of ~10 images (stretch goal 30)
- Short-term deployment on a UMass structure
    - Inside installation
- Users able to update the display via handheld app
- Overall power consumption of < 300 watts
    - Plug into AC

Previous Block Diagram



Redesigned Block Diagram



LED Display

Requirements:

- Refresh rate on display can support changing of images

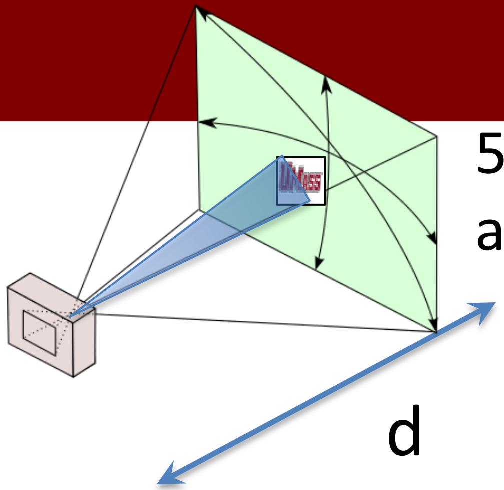
MDR Deliverables:

- Create temporary sample display array of LEDs on a board
- Interact with microcontroller

LED Display

- Alternatives: HDTV Screen, Projector, and custom LED PCB
- Custom PCB will house RGB LEDS
  - 1m x 1m display
  - 30mm Pitch
  - 1000 LED Pixels
- Optimal Viewing distance 10m-30m for low contrast imagery
- WS2812 chosen as RGB LEDS

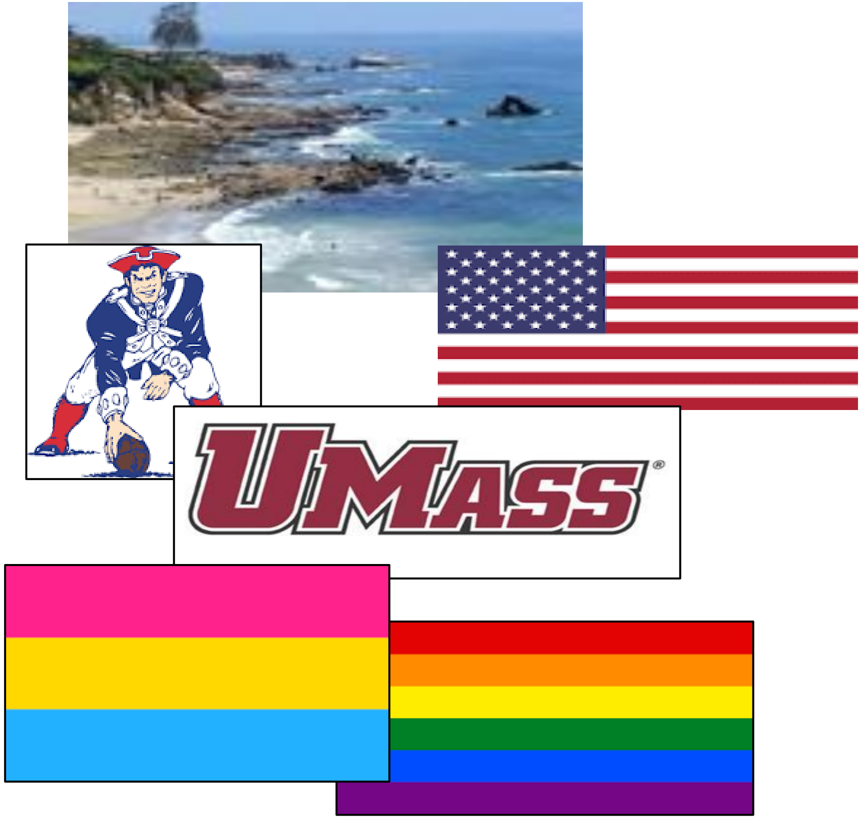
55 degree visual attention FOV



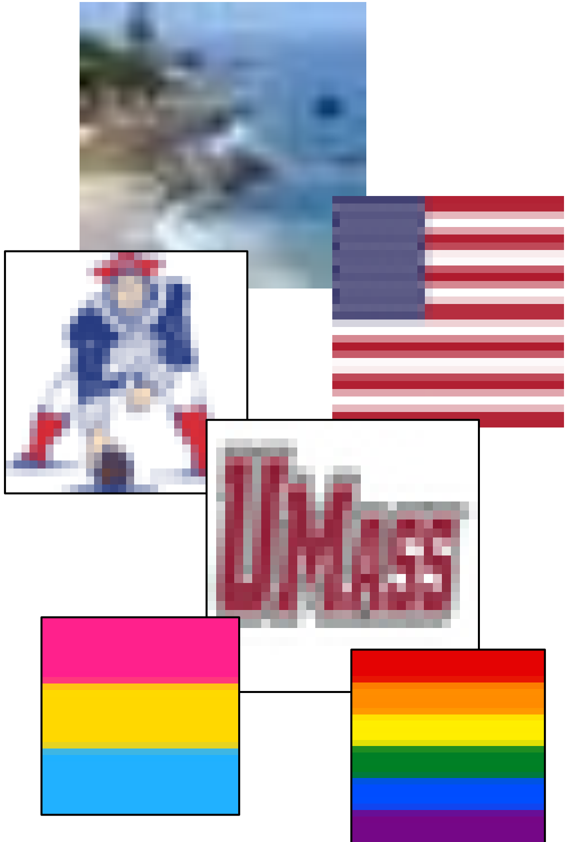
- Human cone of visual attention is ~ 55 degrees wide
- Linear field of view (FOV) at distance d:
  - FOV = 2*tan(27.5)*d = 1.04 d ~ d meters

- What fraction of the FOV is illuminated by the 1 meter display?
  - 1/3 FOV @ d=3 m range
  - 1/10 @ d=10 m range
  - 1/60 FOV @ d=60 m range

Original 400x180 pixels



Resampled 30x30 pixels



Visibility at 3 meters

Visibility at 10 meters

Visibility at 60 meters

Microcontroller and Peripherals

Requirements:

- Still images
- Accepts and crops generic image files from server via WiFi to correct aspect ratio
- Downsamples to fit number of pixels in display
- Hexadecimal to RGB conversion
- Outputs serial neopixel protocol
- Refresh picture in ~seconds

Microcontroller and Peripherals

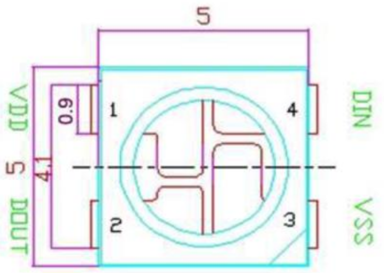
Implementation alternatives:

-Raspberry Pi, Beaglebone, or similar
-Raspberry Pi chosen due to availability of drivers for ws2812b neopixels
-Arduino being used in test and development stage

Microcontroller

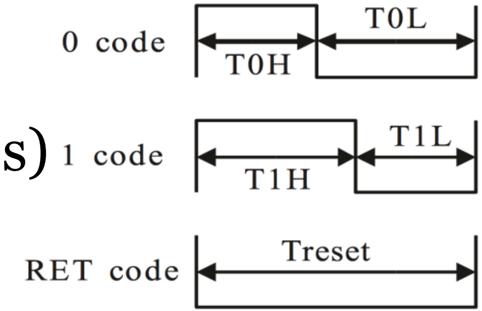
MDR Deliverables:

- Microcontroller drives sample display



Accomplished:

- Microcontroller drives sample display (6 LEDs)
- Arduino used for proof of concept
- Neopixel communication protocol fulfilled

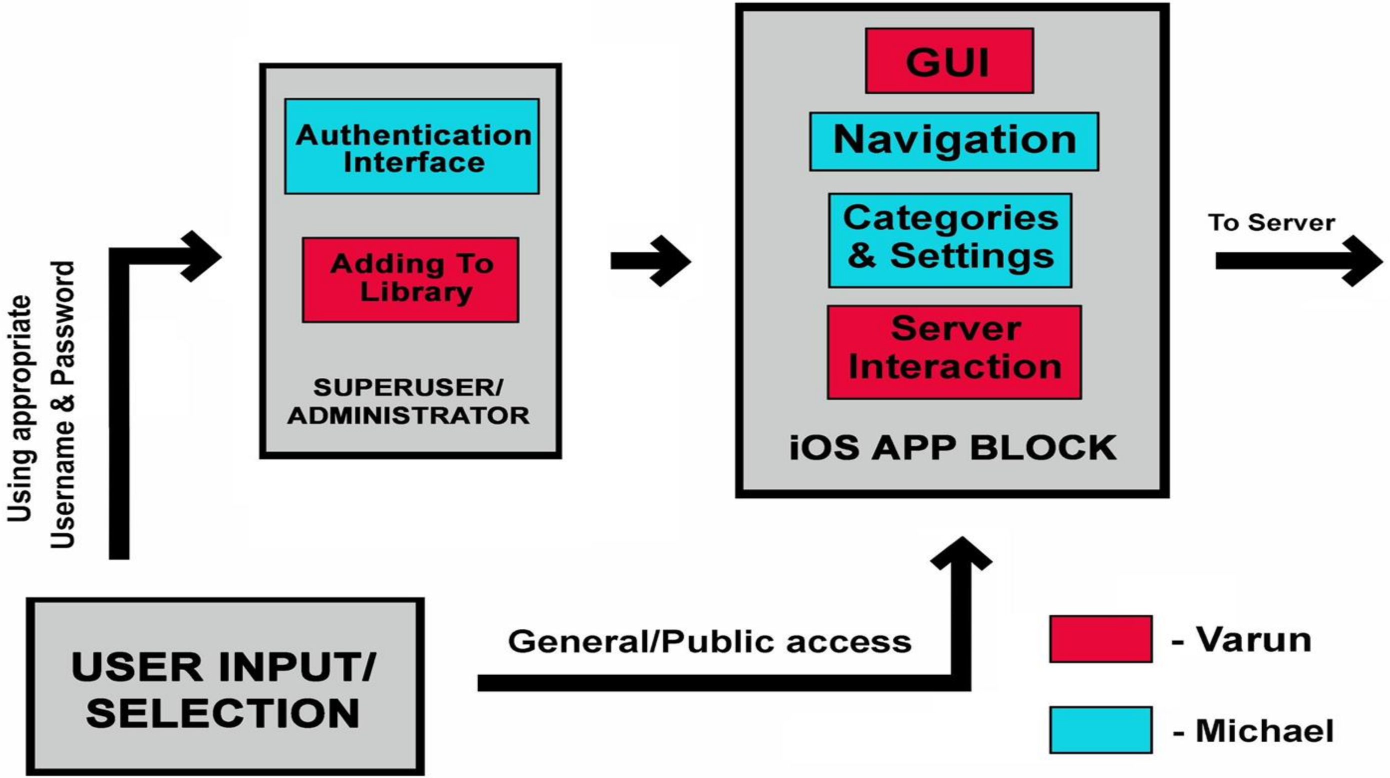


iOS App

System Requirements:

- An easy-to-navigate app interface where users can pick an image to be displayed on the LED display panel
- Users can choose from a catalogue of preset images
- App receives user input and transmits data to the server (real time communication)
- Superuser/administrator
    - Authentication (username/password)
    - Privileges (uploading new images, creating new categories etc.)

App - Block Diagram



App - MDR Deliverables

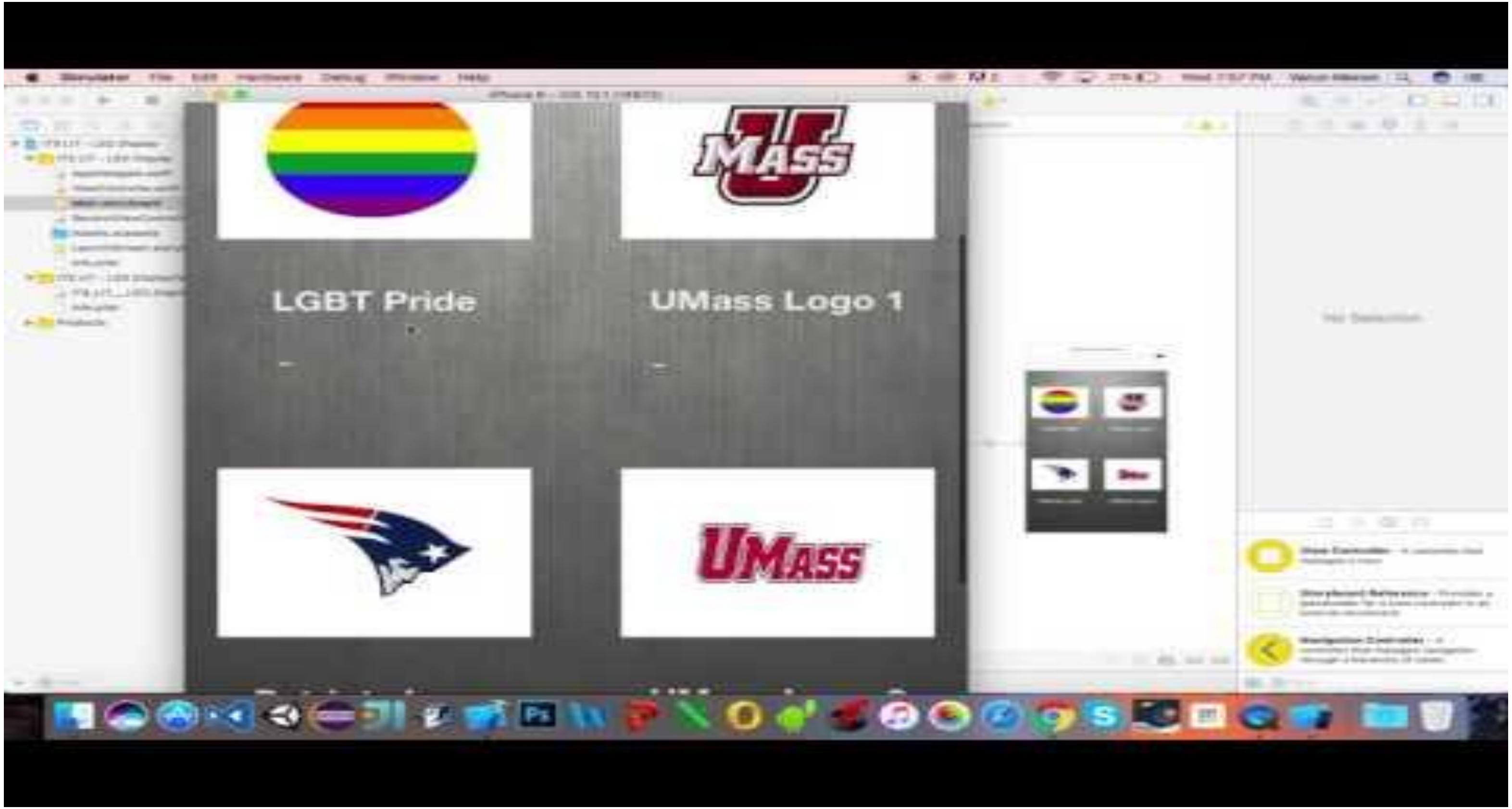
Deliverables:

- Basic app GUI, layout and navigation
- User input tested

Accomplished:

- Home screen and sample preset images (basic graphics and navigation)
- User clicks tested

App - Demo


Simulator File Edit Hardware Debug Window Help
LGBT Pride
UMass Logo 1
UMASS
UMASS

App - Superuser/Administrator

Authentication:

- Single superuser account with unique username and password
- Login credentials will be encrypted and decrypted

Privileges:

- Ability for superuser to create new categories
- Ability for superuser to add and delete images from new and/or existing categories
- App will add new images to the local library as well as the server library

App - Interface & Server Interaction

App Interface:

- Categories will be decided in accordance with display policy
- Scroll view for categories
- Presets will be sampled before being added to library

Server Interaction:

- App makes HTTP requests to web server and parses the response
- RestKit for iOS: Framework that integrates with core data and a simple set of networking primitives for mapping HTTP requests and responses

App - Location Access

- Not a driving requirement
- This approach may be used as an alternative to a Wi-Fi fence/perimeter
- App asks user for location access while running
- Location of LED display panel hardcoded into app
- Checks proximity between user and display, and decides whether to send request or not (Eg. proximity perimeter of 120 ft.)

Server

Requirements

- Act as storage between app and microcontroller
- Queueing for multiple requests
- Response time (~5 seconds)

MDR Deliverables:

- Server is created and set up
- Microcontroller interaction tested

Server

Apache

- HTTP server (TCP/IP protocol)
- Hosted on laptop and accessed through IP address

Alternatives

- Host on cloud (Amazon Web Services, Microsoft Azure)

Demonstrate:

- Apache web server created and hosted
- Microcontroller can access and download server files

Server Demo

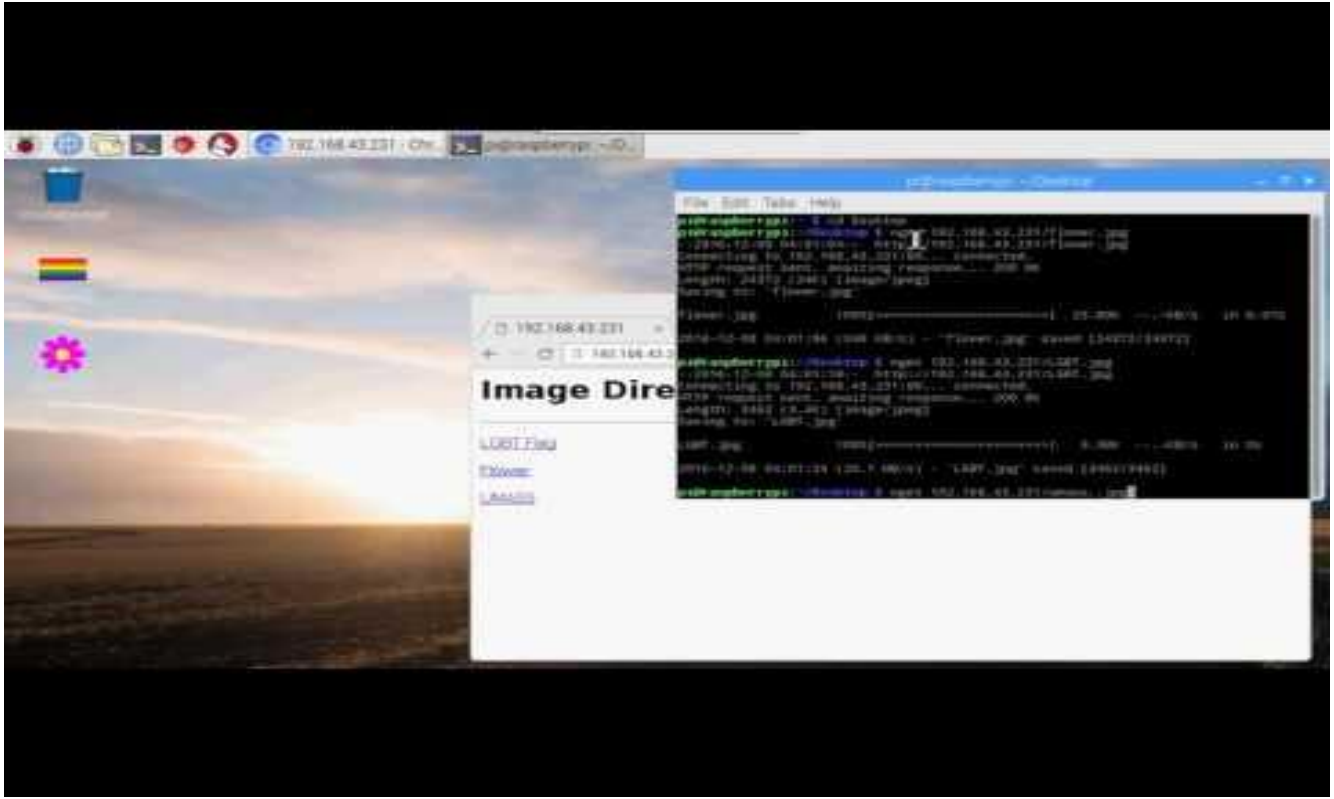


Image Dire

LGBT Flag

Flower

[illegible]

```
pi@raspberrypi:~ $ cd Desktop
pi@raspberrypi:~/Desktop $ wget 192.168.43.231/flower.jpg
--2016-12-08 [illegible]--  http://192.168.43.231/flower.jpg
Connecting to 192.168.43.231:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: [illegible] [image/jpeg]
Saving to: 'flower.jpg'

flower.jpg          100%[===================>]  [illegible]    --.-KB/s   in [illegible]

2016-12-08 [illegible] ([illegible]) - 'flower.jpg' saved [illegible]

pi@raspberrypi:~/Desktop $ wget 192.168.43.231/LGBT.jpg
--2016-12-08 [illegible]--  http://192.168.43.231/LGBT.jpg
Connecting to 192.168.43.231:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: [illegible] [image/jpeg]
Saving to: 'LGBT.jpg'

LGBT.jpg            100%[===================>]  [illegible]    --.-KB/s   in [illegible]

2016-12-08 [illegible] ([illegible]) - 'LGBT.jpg' saved [illegible]

pi@raspberrypi:~/Desktop $ wget 192.168.43.231/[illegible].jpg
```

Display Policy

Requirements:

- Implement display policy

MDR Deliverable:

- Begin vetting with school or faculty member

Accomplished:

- Started a dialogue with Professor Carolina Aragon from landscape Architecture

Proposed CDR/FPR Deliverables

PJ

- PCB design & display policy complete for CDR - Feb
- Display complete for FPR - April

Tommy

- Server and app transmit/receive data for CDR - Feb
- Server and microcontroller full functionality and queuing for FPR - April

Emma

- Raspberry Pi computer controls a 2-dimensional display for CDR and renders at least one image type from server - Feb
- Able to render downsampled images of different file types for FPR - April

Mike

- Multiple views for categories/settings complete for CDR - Feb
- Power supply & superuser authentication process complete for FPR - April

Varun

- App is able to send, receive requests and interact with server for CDR - Feb
- Power supply & superuser privileges complete for FPR - April

Questions?