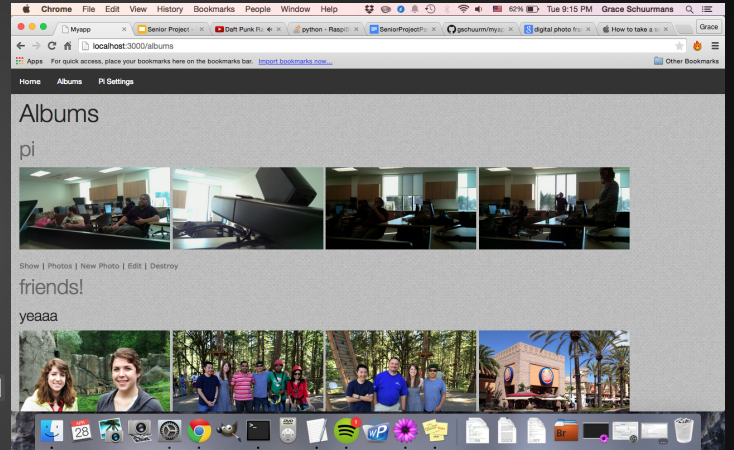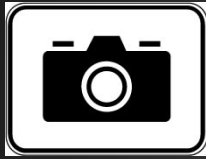# Camera with Photo Website and Display

Grace Schuurmans, BS in CS

Display and Website Communication
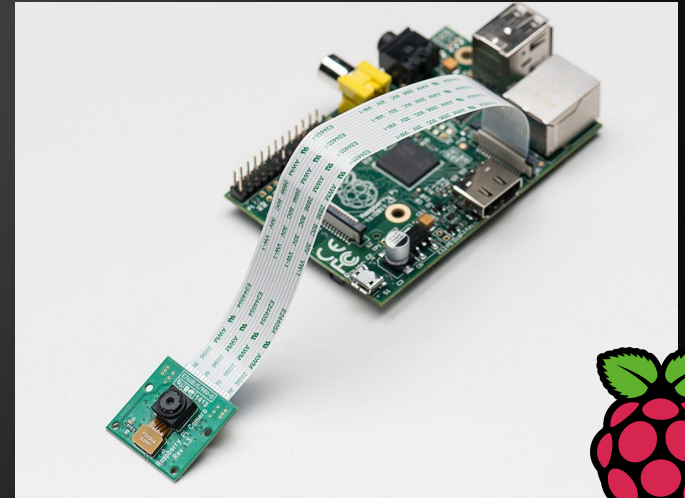
Technologies used

camera and display program:

    -Raspberry Pi (+RaspiCam)

    -Python 2.7

        *Tkinter - (PIL and PhotoImage)

        *requests (HTTP)

website: Ruby on Rails v.4

    -Bootstrap (CSS)

    -HAML (HTML)

    -Rspec (Test Suite)

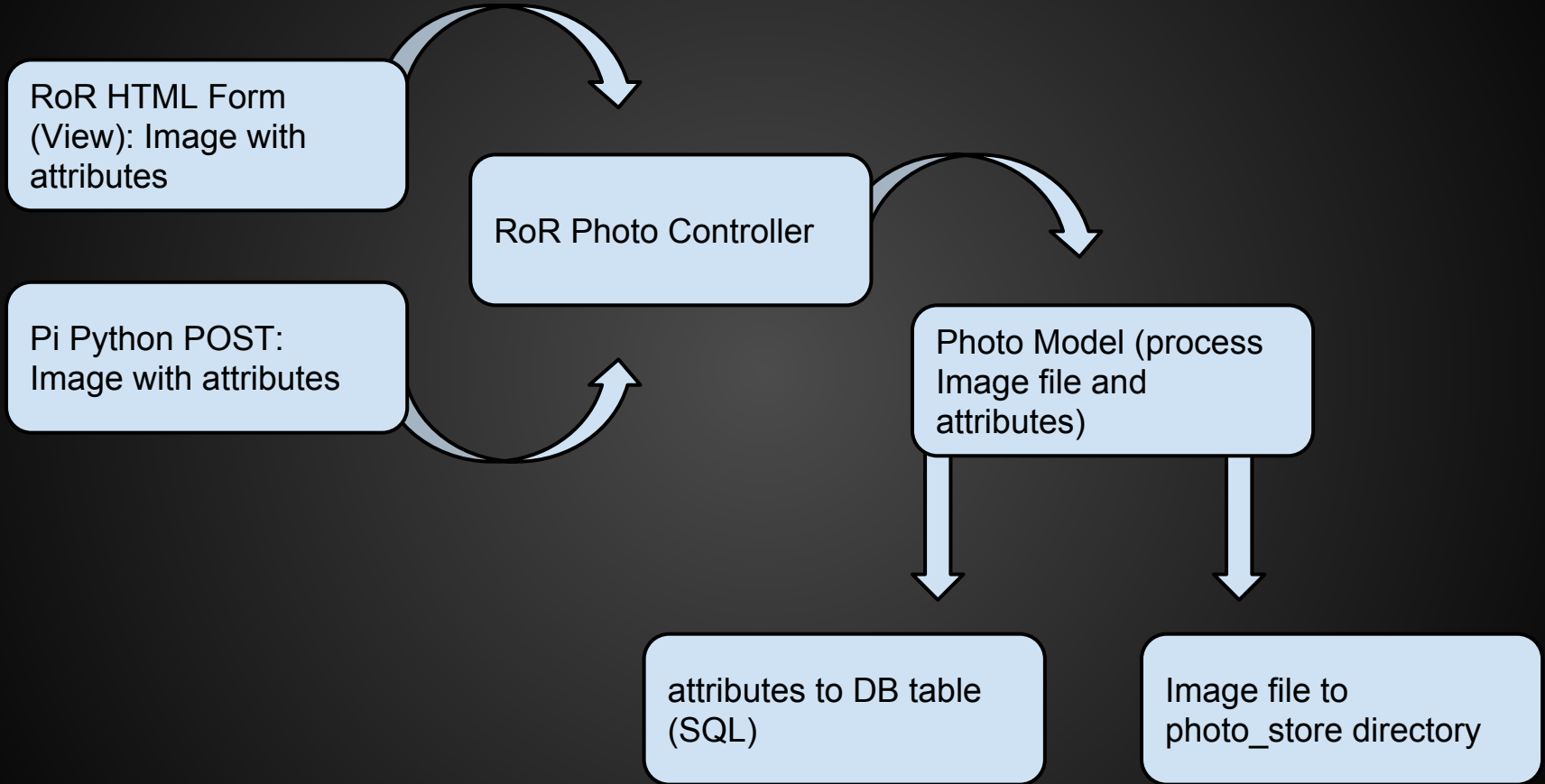GIT version control (helps a lot)

Points of Interest

RESTful URIs

    "albums/1/photos/23"

Python modules

    Tkinter/image display

    requests send json and file in HTTP POST

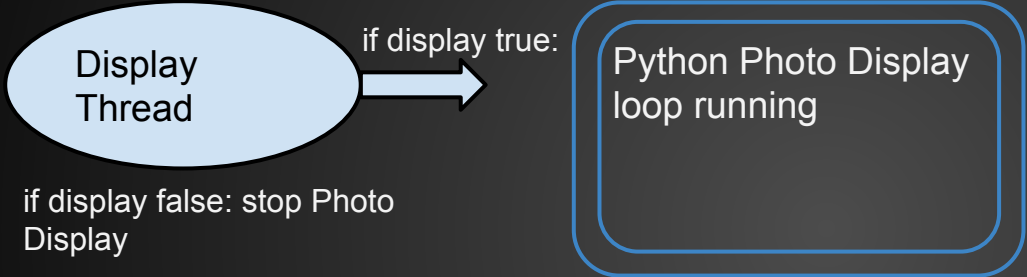# Creating Photos on Website Flow

RoR HTML Form (View): Image with attributes

Pi Python POST: Image with attributes

RoR Photo Controller

Photo Model (process Image file and attributes)

attributes to DB table (SQL)

Image file to photo_store directory

Python on Pi
Photos to Website
and Display Flow

Get-Settings Thread*

GET every 10 sec

RoR Pi Settings webpage
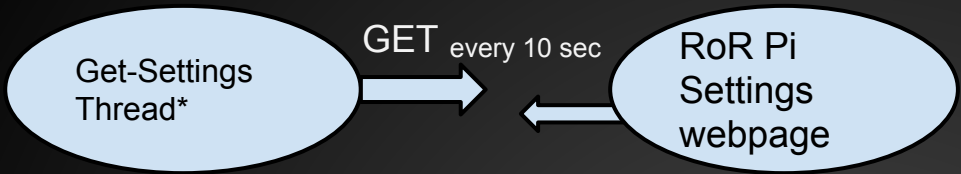
*display and capture vars updated

Display Thread

if display true:

Python Photo Display loop running

if display false: stop Photo Display

Capture Thread

if capture true:

take photo and POST
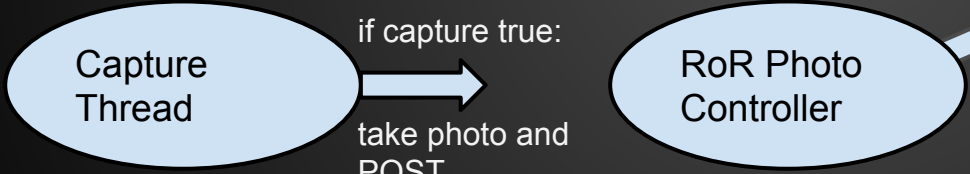
if capture false: stop capturing new photos

RoR Photo Controller

save attributes and photo file from request
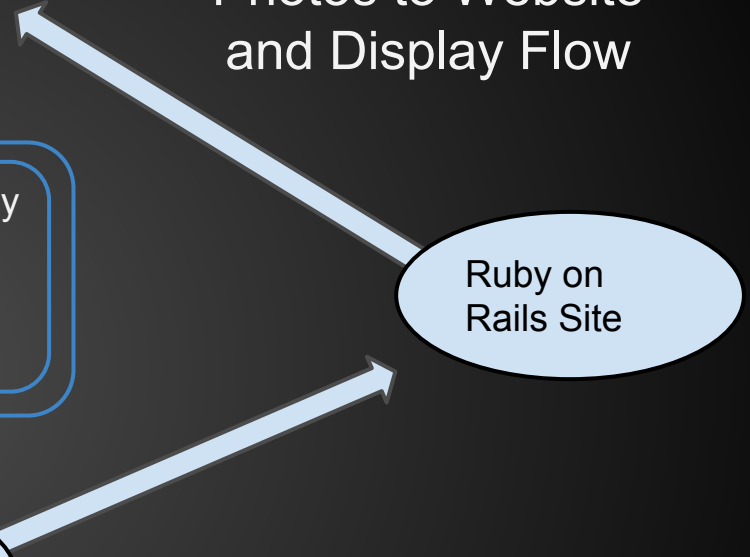
Ruby on Rails Site

My Rubric

Feature	Points
Raspberry Pi Python Program:	
Capture mode: captures photos from Pi camera as image objects every **5 minutes** (or other configured time)	15
adds newly captured photos to queue of photos to display on a screen when in Display mode	15
switches the Capture and Display modes on and off from site interface	20
displays photos from a USB/flash drive for Display mode**...**	10
*shows when photo is being taken- screen flash countdown 5 seconds prior	5
*inserts recently-taken photo into "next" photo in queue [with Display mode on]	5**
*downloads photos to Display queue from website	10

RoR Photo Website:	
Basic site functionality working for easy usability (RESTful URL design, Bootstrap framework for instant CSS design on all devices, **displaying photos from database on web page**)	30
pick up photos taken by Pi camera (automatically pushed from Pi when they're taken or initiated by website) and add to database	30
upload non-Pi photos to website from a browser	5
edit, delete, and add photo info (title, description) and ability to delete photos	5
TDD with Rspec, HAML instead of HTML	10** - 1/2
*authentication for accessing site	5
*show **photo timestamp** [and title/description] with photo on web page	3
*automatically organize Pi photos into "Party Cam" album	5
*organize other photos into albums	5

A	B	C	D	F
140	120	100	80	<80

Total: 143