IoT Smart Calendar

Advisor: Dr. Malinowski

Jason Morris, Cole Lindeman

Presentation Contents

- Introduction
- Prior Work
- Functional Requirements
  - System Level Diagram
  - Subsystem Diagrams
- Efforts Completed
- Parts List
- Schedule for Completion
- Future Discussion
- References

Introduction

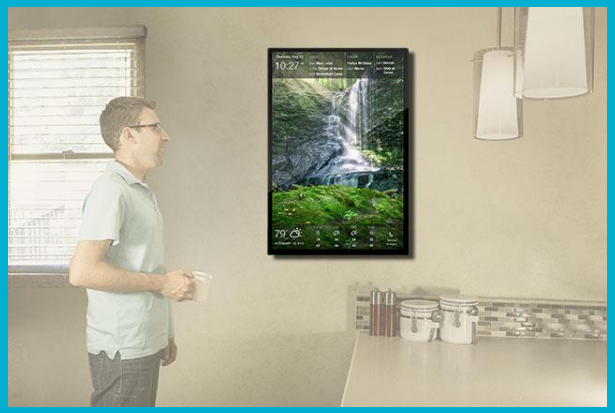
An Internet of Things Smart Calendar

- Wall mounted Smart Calendar placed outside of the Professor's office
  - Displays Calendar for people who stop by
  - Displays advertisements for people walking by
  - Allow users to leave messages
- Interface with sensors
  - Track and record motion
  - Sense if office door is open
  - Electronically open office door*
- Communicate with the Internet
  - Gather GPS data
  - Retrieve urgent announcements
  - Send messages, alerts and tracking data

Prior Work

Raspberry Pi Framed Informational Display

- Powered by Raspberry Pi 2
- Displays Google Calendar and local weather
- Turns off display at night
- Buttons on side toggle Google Calendar between monthly, weekly, and daily views



DAKboard

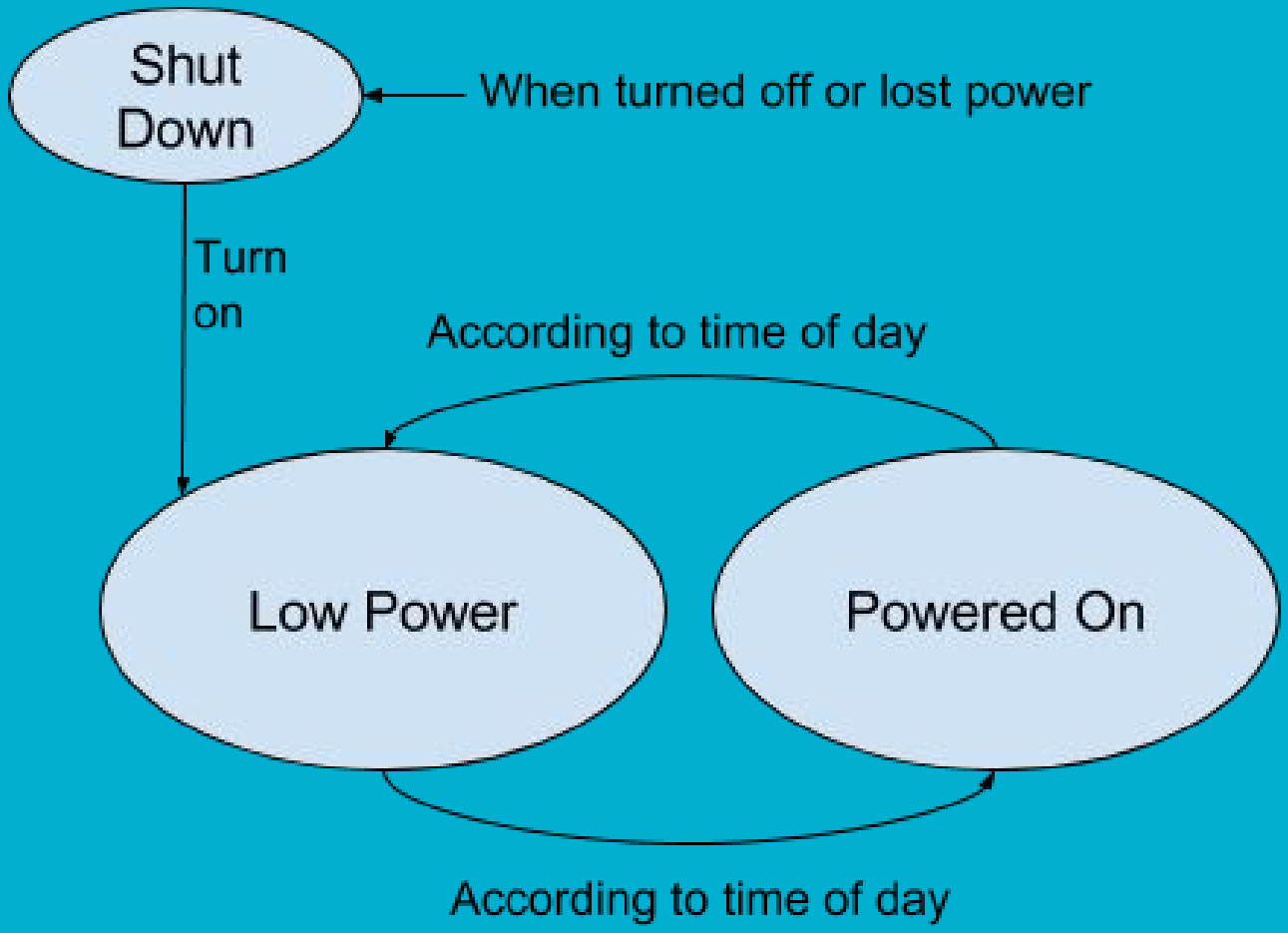
- Customizable wall display
- Can showcase photos, calendar events, and weather
- Allows user to customize information to be displayed
- Everything is done through web interface

Functional Requirements

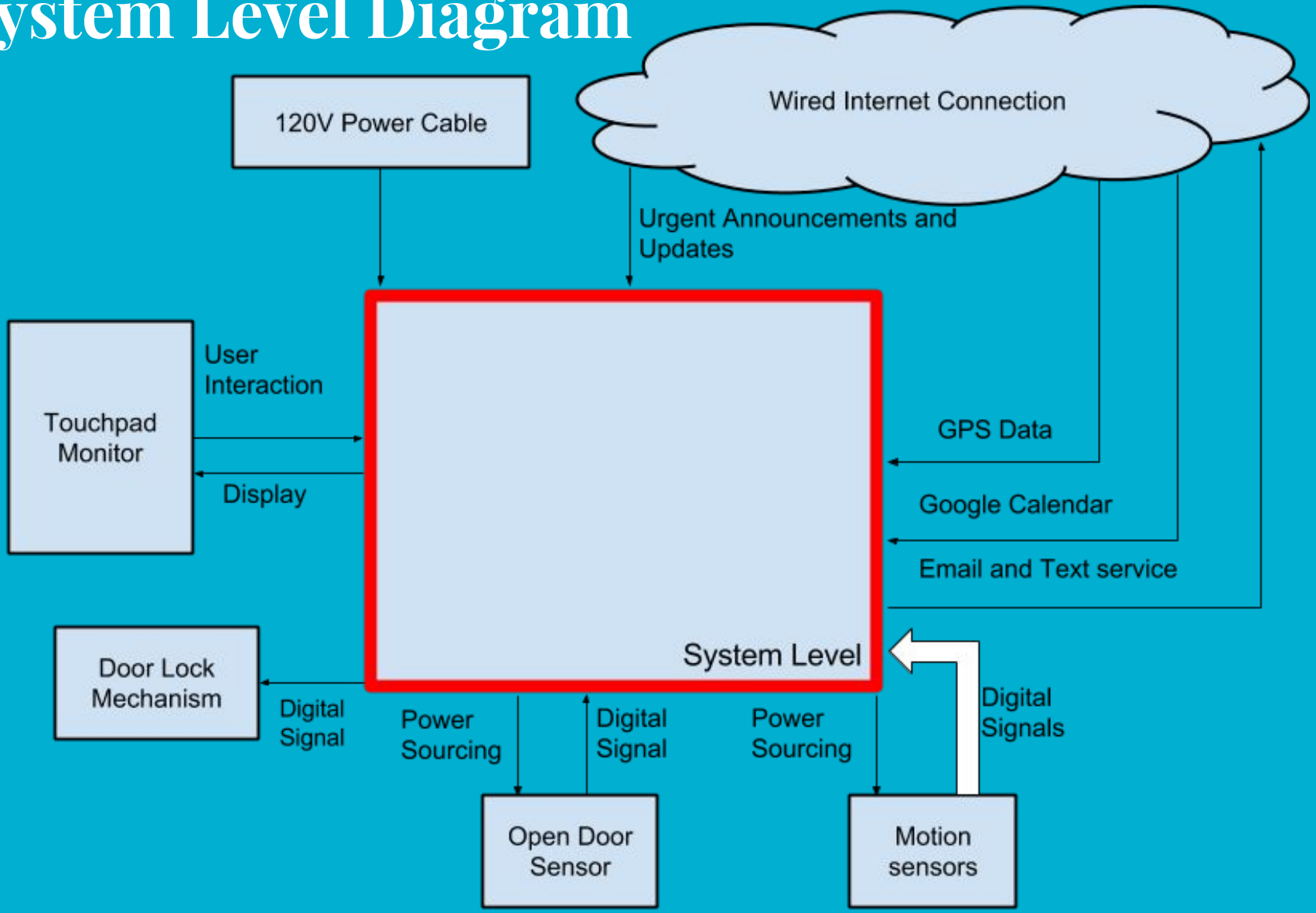
Modes of Operation:

- Low Power
  - Smart Calendar is saving power, display is off, sensors are off
- Shut Down
  - Everything is completely powered off
- Powered On
  - Display is on and capable of showing showing calendar, advertisements, messenger service, or announcements
  - Users can manually switch what the display shows
  - Switches to advertisements automatically when idle
  - When idle, switches to calendar if user stops at calendar
    - Automatically alert professor of the person stopped at the calendar if office hours are concurrent, the office door is closed, and the professor is on campus
  - Perform periodic updates of the calendar, advertisements and announcements

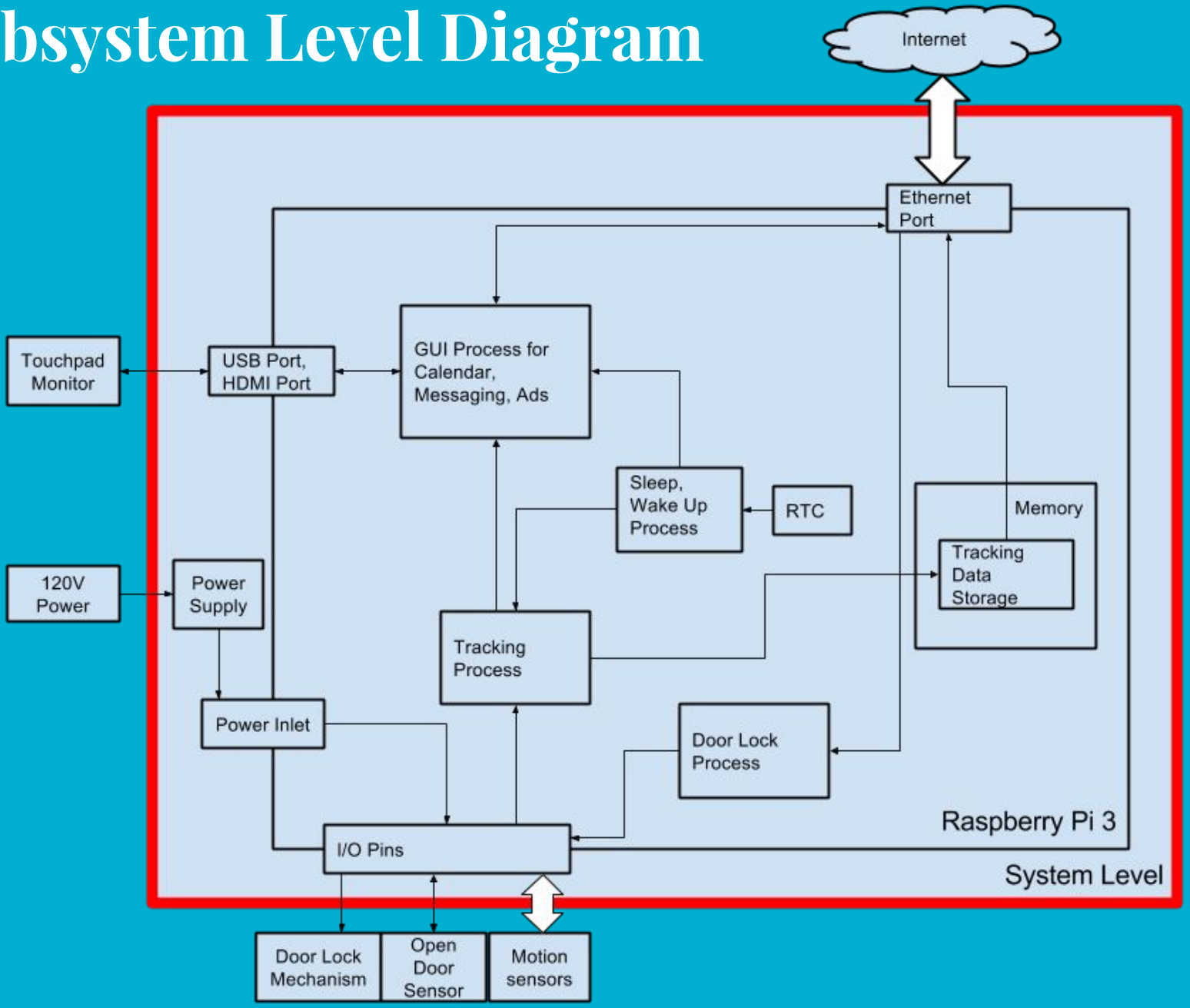
Functional Requirements



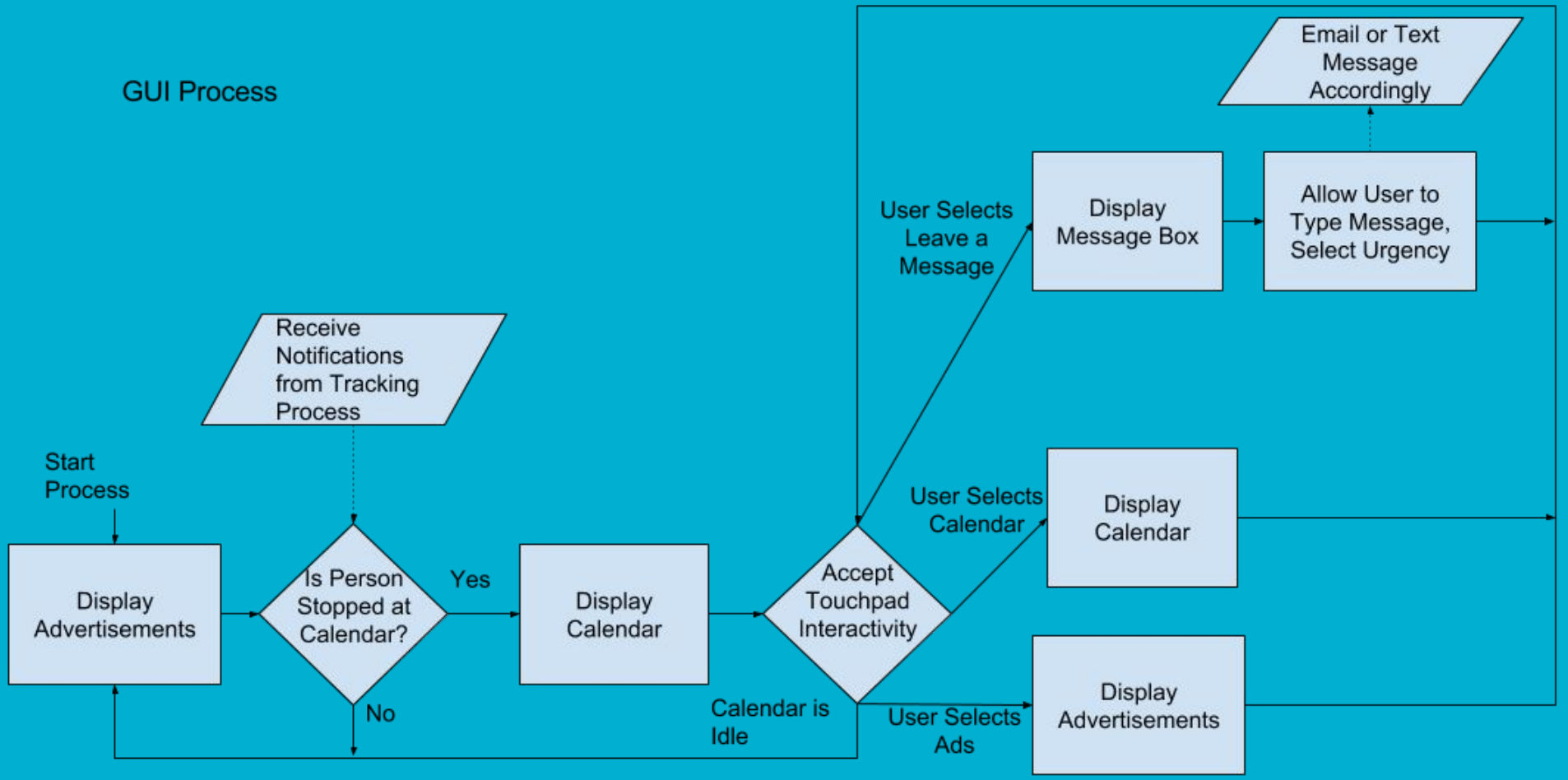
System Level Diagram



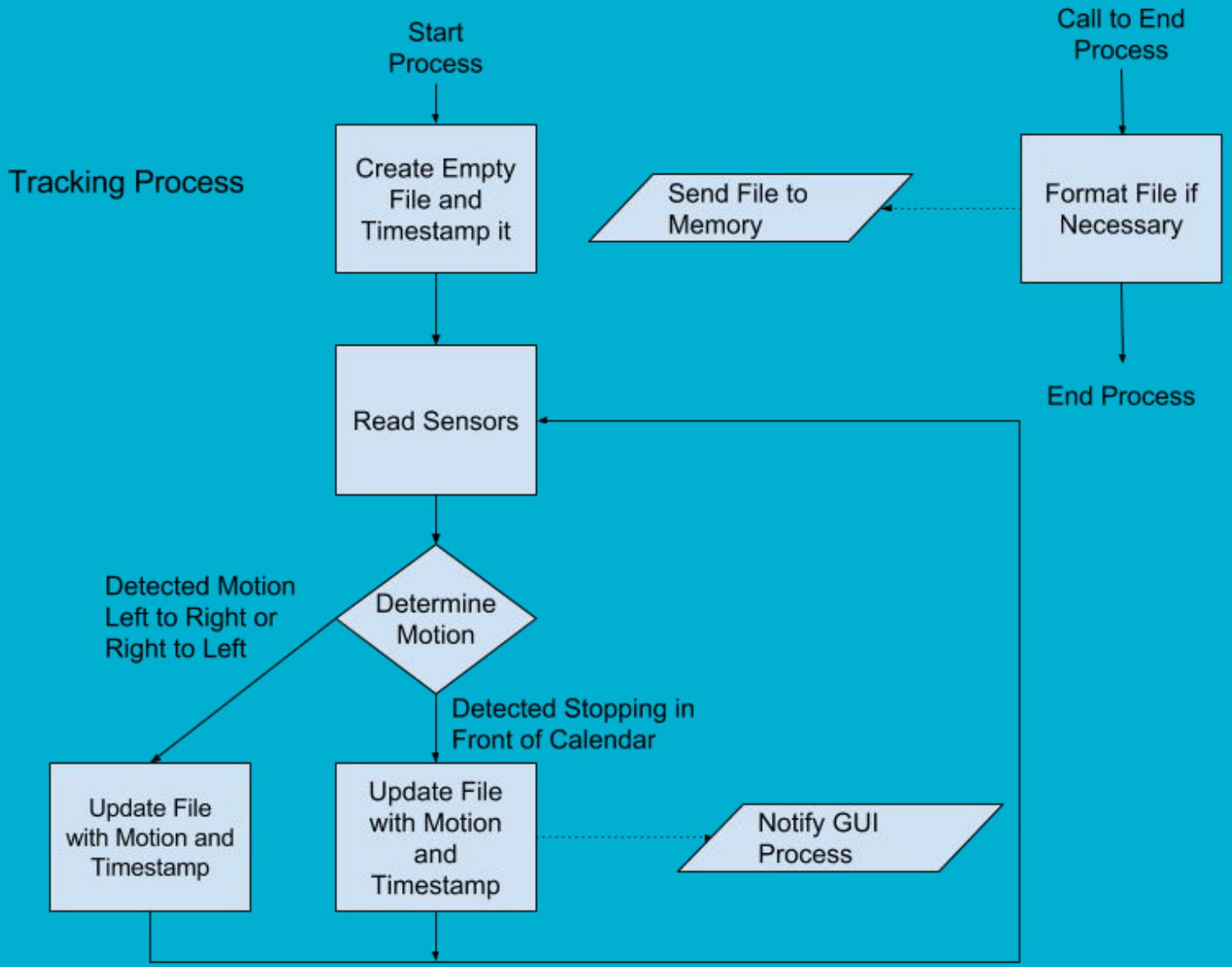
Subsystem Level Diagram



GUI Process Subsystem



Tracking Process Subsystem



Efforts Completed

Basic designs already completed

Studied language platforms:

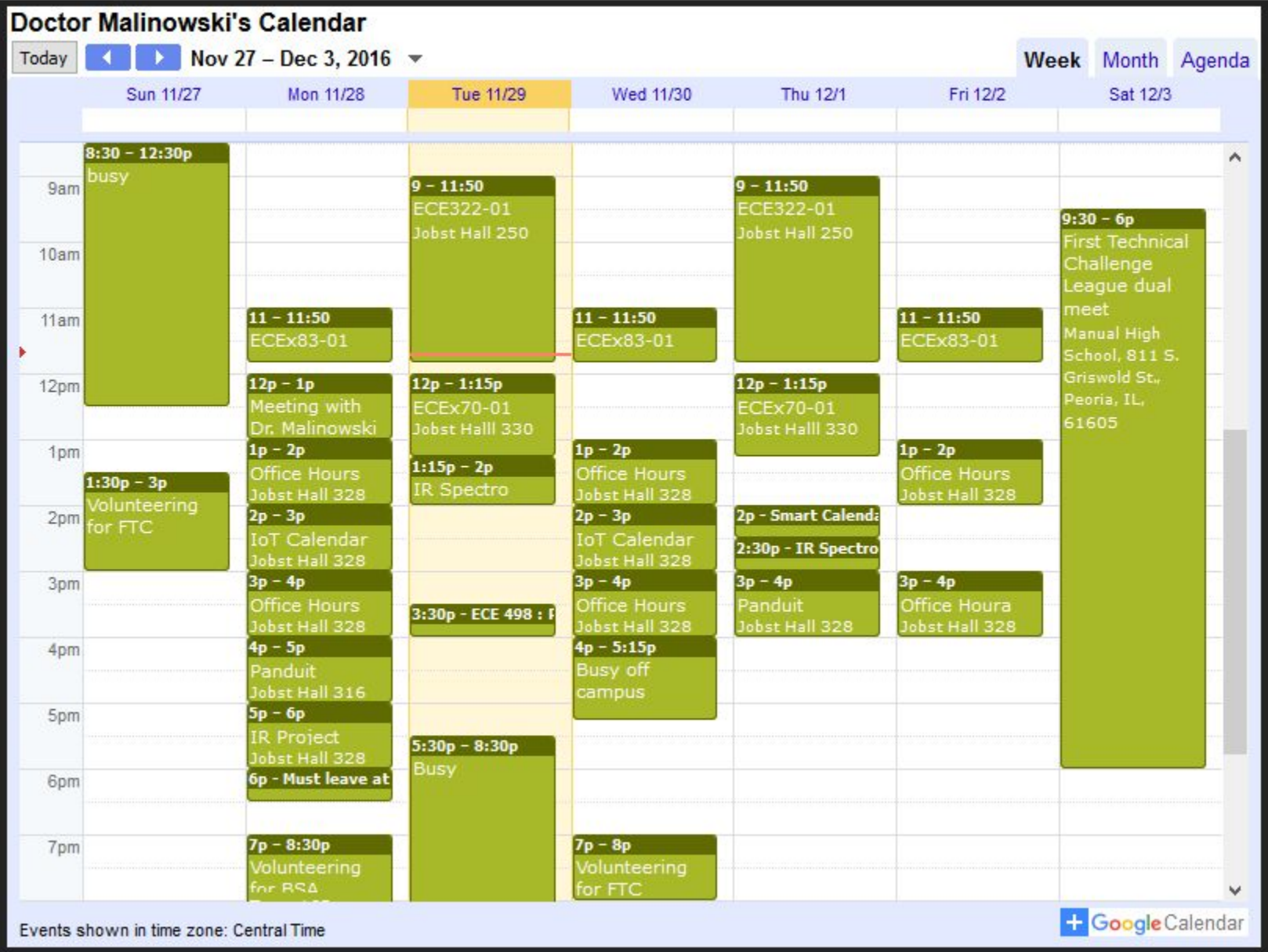
- Python
- HTML with Javascript, CSS, and Ajax
- PHP
- Bash

Some functionality is already implemented

Tested in similar environments:

- Raspberry Pi 2, Ubuntu MATE
- Virtual Machine, Xubuntu

Calendar | Advertisements | Mali Messenger



Parts

- Waveshare 10.1 inch 1280x800 IPS LCD Capacitive Touchscreen with case
  - $118.99
- Raspberry Pi 3 with power supply, case and heatsinks
  - $45.99
- Sandisk 32GB microSDHC card with normal SD card adapter
  - $10.59
- Aleko magnetic reed switches
  - $9.99
- Emy passive infrared motion sensor detector modules
  - $5.49
- Ethernet, HDMI, USB and digital I/O cables
  - $14.89

Total:$211.89

Schedule

Week		Jason's work	Cole's work
1/15/17	1/21/17	**Spring Semester begins** Write Python code to host HTTP web server for Ajax to communicate with	
1/22/17	1/28/17	Write XML code using Ajax to direct browser	Continue writing Python code to communicate with Ajax
1/29/17	2/4/17	Write HTML code to direct browser back to ads when idle for long enough	Setup Raspberry Pi Setup monitor for Pi
2/5/17	2/11/17	Write javascript for ads that "follow" passersby	Figure out reading, writing, and permissions for I/O pins Connect sensors to Pi
2/12/17	2/18/17		Write Python script to poll I/O pins Write Python script to enable and disable I/O pins
2/19/17	2/25/17	Find method to upload text files Write script to use method to upload tracking text file	Write Python script to track movement with IR sensors
2/26/17	3/4/17		Write Python script to compile movement information into a text file

Schedule Continued

Week		Jason's work	Cole's work
3/5/17	3/11/17	Write Python script to send commands to Ajax using movement information	Write Python script to communicate with door lock
3/12/17	3/18/17	**Spring Break**	
3/19/17	3/25/17	Test Internet communication	Write script for sleep/wakeup process
3/26/17	4/1/17	Test mount setup for project	
4/2/17	4/8/17	Mount project	
4/9/17	4/15/17	Spare time in case of changes	
4/16/17	4/22/17	Spare time in case of changes	
4/23/17	4/29/17	Spare time in case of changes	

Discussion

Near Completion:

- Create a Python local web server
- Communicate using Ajax to control to interface with local sensors

Still Remaining:

- GitHub automatic updates
  - Or other form of convenient updating the notifications and advertisements
- Smart Calendar functionality
  - Checking for movement
  - Checking if the office door is open
  - Some form of simplified geofencing
- Optional companion phone application for students

References

Archambault, Michael. "DAKboard Is a Customizable Wall Display for Photos, Calendar Events, and Weather." PetaPixel. N.p., 19 Aug. 2015. Web. <http://petapixel.com/2015/08/19/dakboard-is-a-Customizable-wall-display-for-photos-calendar-events-and-weather/>.

Barrett, Daniel J. *Linux Pocket Guide*. Sebastopol, CA: O'Reilly, 2004. Print.

"jQuery.ajax()." *Ajax jQuery API Documentation*. JQuery Foundation, n.d. Web. <http://api.jquery.com/jquery.ajax/>.

Kmccb. "Raspberry Pi Framed Informational Display - Google Calendar, Weather, and More.." Imgur. N.p., 07 Apr. 2016. Web. <http://imgur.com/gallery/z94Vr>.

"Linux Documentation." *Linux Documentation*. N.p., n.d. Web. <https://linux.die.net/>.

“PHP 5 Tutorial” PHP 5 Tutorial. W3 Schools, n.d. Web. 2016. <http://www.w3schools.com/php/default.asp>

"Python 2.7.12 Documentation." *Python 2.7.12 Documentation*. Python Software Foundation, n.d. Web. 2016. <https://docs.python.org/2.7/>.

Questions?


Calendar
Sun Mon Tue Wed Thu Fri Sat
1 2 3 4
5 6 7 8 9 10 11
12 13 14 15 16 17 18
19 20 21 22 23 24 25
26 27 28 29 30 31