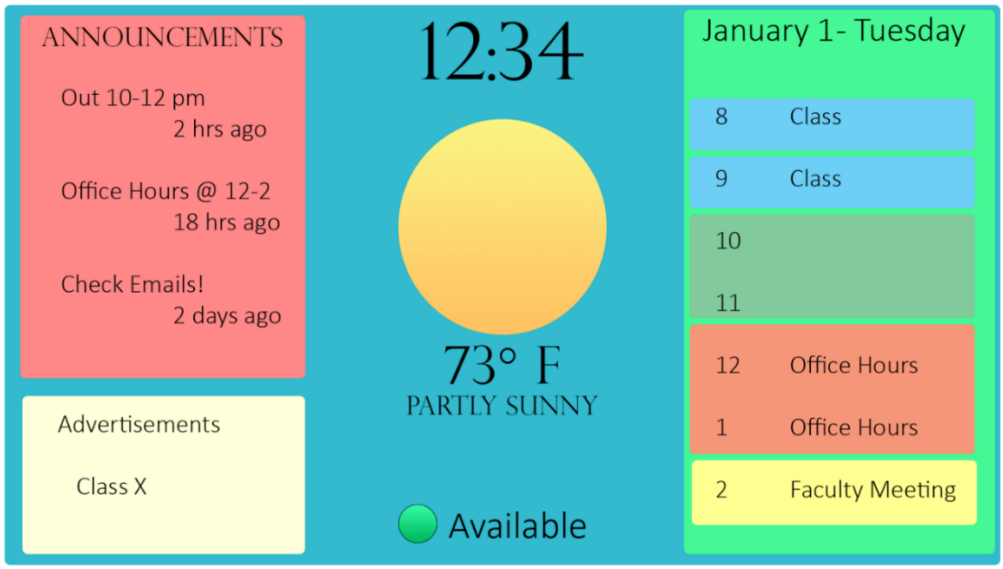
IoT Information Management Display (with Geofencing)

Fall 2017 - Spring 2018



Concept Idea Image

Benjamin Daszkiewicz

bdaszkiewicz@mail.bradley.edu

Jacob Nading

jnading@mail.bradley.edu

Advised by:

Dr. Aleksander Malinowski

olekmali@bradley.edu

Abstract

This document provides a problem statement and functional requirements for the IoT Information Management Display (formerly IoT Smart Calendar) project advised by Dr. Aleksander Malinowski of the Electrical and Computer Engineering department at the Caterpillar College of Engineering and Technology at Bradley University. It also offers precursory solutions to some of the project's core requirements. The project is now in its second year of development. Benjamin Daszkiewicz and Jacob Nading are continuing the work started by Jason Morris and Cole Lindeman in the Fall of 2016.

Problem Statement

The problem this project looks to address is lack of communication between faculty and students. Too often in higher learning, busy schedules and an inability to communicate face-to-face result in underachievement by students. While a professor may have office hours, busy students may opt to skip the office hours instead of take the time to seek out the professor or send an email from their personal device to schedule an appointment.

Dr. Malinowski, and later another faculty member, expressed interest in having an interactive board with which to communicate with students. The IoT Information Management Display is a central location where a professor or other professionals can display information regarding their schedule, announcements, and information that may otherwise be cumbersome for others to find. Where an office visitor may otherwise have to navigate clumsy websites on their device to find out when the faculty will return, they could instead use the display to instantly see his or her general location and information on their office hours. The basic functionality of the calendar is an ongoing project; however, as will be described in Proposed Solution, the goal is to implement more features, but also to create a more approachable display that encourages user touch interaction. New display software will be developed using both recycled technologies from the existing prototype and new technologies. The new software will be compatible with the prototype display and may be installed on that hardware as well as on as many new instances of hardware as is desired. Currently one new display is desired.

Minimum Functionality Requirements

The list of features for the calendar project will remain flexible aside from a few foundational ideas that must be included for minimum functionality. Once the minimum functionality is integrated, project development will continue to implement extra useful functionality and improve the visual characteristics of what is already installed until the project deadline is reached. Project management software will help to ensure that there is always a stable version of the software available to ensure a deliverable product at any time one is due during the development process. Additional details of the requirements may be found in the form of a solution description under their respective Solutions sub-heading.

The minimum functionality demanded by Dr. Malinowski and other project backers includes the following:

- Daily calendar data
    - Display must show professor's schedule, synchronized automatically from their Google calendar account
- Short memos/announcements including an urgent announcement feature
    - Professor must be able to easily publish an announcement to the Display remotely from their personal device
- Advertisements for courses instructed by the calendar owner
    - Display must contain automatic timeout feature that displays relevant course advertisements when other functionality is not in use
- Paging functionality based on geofencing
    - Server must receive GPS data from professor's personal device to detect location relative to geofence and cross-reference calendar. Paging button must send message to professor's cell phone via SMS
- Current and forecast weather data
    - Display users should be able to access current and forecast local weather data and maps

Additional Functionality

The proposed timeline is to begin implementing these features halfway through the design process after the minimal functionality is completed. Not all of the listed functionality is expected to be completed by the completion date. Instead, we will implement as much as we can, and remaining features may be implemented by future teams. Due to the uncertain nature of when they will be implemented and by whom, description of additional functionality is left general.

- Attractive and user-friendly app-like display
- E-mail and/or text faculty via Display
- Appointment scheduler
- Twitter account link for announcements

Proposed Solutions

Our preliminary solutions to minimal functionality will first include reviewing Jason Morris and Cole Lindeman's prior work on the Display. Dr. Malinowski expressed a desire to alter some of the methods used to meet the project requirements. The review of their work will be in search of code or implementation strategies that help facilitate what will be closer to a ground-up redesign of the current Display. Comparison is drawn here for how functionality will differ between the new and prototype versions when discussing functionality that is already included in the prototype.

Features and Functionality

Calendar

The calendar will display the current day's schedule. We will first attempt to embed the calendar directly from Google Calendar using the Google API, as that should be less technically difficult and will provide a starting point for functionality. Once it is up and working, the calendar section may be revisited to customize the display, while pulling only the data using the Google Calendar API. Clicking on the daily view of the calendar from the main display screen will bring up a larger calendar display that can be navigated to show past and future events.

The current version of the Display includes calendar functionality. Our goal is for the new version to maintain the calendar functionality. If there is enough time, we would like to improve the default look of the calendar that is used be the Google API. This would be accomplished by pulling data from the calendar and embedding it into a custom user interface.

Memos/Announcements

The current Display does not have an easy interface for the professor to post announcements that are last minute or not included on the calendar. We plan to implement such a system through SMS, a Twitter feed, or some other such easy messaging platform. The most recent couple of announcements will display on the home screen and less recent announcements will be reachable by tapping the announcements section.

Current and Forecast Weather Data

The prototype display developed last year had weather displays added after the end of the project deadline. The data is displayed in HTML format from a link that gets current weather data. The new Display may use a more dynamic API that allows custom UI development to improve its visual appeal. The main portion of the display homepage will be a real-time image representing current weather conditions. It may also include a visual for a condensed forecast. The data will be accessed via weather.gov. Like the calendar portion of the display, a more detailed version will appear when it is touched from the homepage.

Paging Functionality Based on Geofencing

Geofencing is a method of GPS location where instead of the user-end utilizing the precise coordinates of the GPS data, it instead cross-references the exact coordinates with a geographical border or “fence,” and returns whether or not the GPS transmitter is inside the fence.

In this case, the fence will be drawn to the specifications of the professor. For instance, if the professor wants to be page-able while inside the building, the GPS coordinates of the building boundaries will be used. GPS data will then be taken periodically from the professor’s cellphone via an app and checked against the fence. Apps to achieve this purpose already exist and would be a preferable solution over building our own app from scratch. If the phone is within the paging boundaries, and the professor has available hours (cross-referenced with the calendar), a paging “button” will be touchable on the display and a message will be sent to the professor’s phone indicating that they are needed at their office. This allows the professor to have a degree of flexibility during office hours to get coffee, visit a colleague’s office, or use the restroom.

A camera at the Display may be included to discourage extraneous paging, and door sensors may be used to complement the geofencing data to help locate the faculty member. The current Display does not have any of this functionality.

Advertisements

The advertisements section of the software will appear under two circumstances. In one circumstance, the Display will time out and switch over to a slideshow of static images that the professor can prepare in advance. The images will advertise courses that the professor teaches. The advertisements will also be accessible on demand through the touch functionality.

The prototype Display has advertisements, but we look to improve how they are integrated. Presently the entire prototype software operates on a sort of “slideshow” basis. The improved version will look to integrate the advertisements in a less invasive, yet equally attention-garnering way.

Hardware

The required hardware will be minimal. A Raspberry Pi 3B will be used as the application host for the Display and can host any servers needed by the application. The Raspberry Pi will require a hardwired network connection to avoid connection issues over wireless communication and will require a 5V power supply. It will connect via HDMI and USB to a touch display. The hardware for the touch display will recycle the current Waveshare 10.1 inch touch display. Additional touch interfaces will be discussed when creating Displays for another faculty.

Software

The software for the Display will be a combination of several systems. The heart of the app’s processing will be written in Python. This includes API calls for web information, calls to display data, and interfacing with internal and external servers. The user interface will be either written entirely in Python or will call a series of HTML documents which would display in a browser. The system will be hosted on Raspbian, installed on the Raspberry Pi 3B.

Conclusion

There are too many instances in today’s world of fast-paced convenience where conventional methods of communication breakdown. In an academic or professional setting this can be detrimental to success and progress. The IoT Information Management Display will allow for a method of communication that is in-line with that modern lifestyle. A user-friendly and interactive message board outside faculty offices will provide increased ease of access for students.