Senior Project Proposal

WayStream



Submitted By: Jacob Chesley

Submitted To: Calvin Caldwell

Email: [Jacob.Chesley@oit.edu](mailto:Jacob.Chesley@oit.edu)

Date: 5/19/18

Version: 0.1

# Legal Notice

WayStream shall be the sole owner of all the products and proceeds of Jacob Chesley’s services hereunder, including, but not limited to, all materials, ideas, concepts, formats, suggestions, developments, arrangements, packages, programs and other intellectual properties that Jacob Chesley may acquire, obtain, develop or create in connection with and during the Term, free and clear of any claims by Jacob Chesley (or anyone claiming under Jacob Chesley) of any kind or character whatsoever (other than Jacob Chesley’s right to receive payments hereunder). Jacob Chesley shall, at the request of WayStream, execute such assignments, certificates or other instruments as WayStream may from time to time deem necessary or desirable to evidence, establish, maintain, perfect, protect, enforce or defend its right, title or interest in or to any such properties.

# Copyright Notice

Copyrights shall mean any and all rights in, arising out of, or associated with works of authorship, including without limitation rights in copyrights and copyrightable works and registrations, applications and renewals for registration thereof, mask works and registrations and applications for registration or renewals thereof, computer software, data, databases and documentation including copies and tangible embodiments (in whatever form or medium) thereof whether protected, created or arising under the laws of the United States or any other jurisdiction, whether published or unpublished, including rights to prepare, reproduce, perform, display and distribute copyrighted works and copies, compilations and derivative works thereof.

# Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Version | Date | Author | Company | Comments |
| 0.1 | 5/19/18 | Jacob Chesley | “Name Pending” | Beginning of rough draft |
| 0.2 | 5/30/18 | Jacob Chesley | “Name Pending” | Updated signature and data description |
| 1.0 | 6/08/18 | Jacob Chesley | WayStream | Added hardware functional specs |
| 1.1 | 10/23/18 | Jacob Chesley | WayStream | Added additional windows to functional spec |
| 1.2 | 11/30/18 | Jacob Chesley | WayStream | Converted from MAC to GUID addressing system |

# Signatory Page

Document accepted by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Receiver (Calvin Caldwell) Date

Document submitted by:

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Signature of Author (Jacob Chesley) Date

Table of Contents

[Legal Notice 2](#_gjdgxs)

[Copyright Notice 2](#_30j0zll)

[Revision History 2](#_1fob9te)

[Signatory Page 3](#_3znysh7)

[Introduction 6](#_2et92p0)

[Purpose 6](#_tyjcwt)

[Scope 6](#_3dy6vkm)

[Intended Audience 6](#_1t3h5sf)

[Project Management 7](#_4d34og8)

[Change Management Procedure 7](#_2s8eyo1)

[CAT Team 7](#_17dp8vu)

[Medium 7](#_3rdcrjn)

[Protocol 7](#_26in1rg)

[Impact Analysis 7](#_lnxbz9)

[Archive 7](#_35nkun2)

[Software Delivery, Installation and Acceptance Criteria 7](#_1ksv4uv)

[Documentation and On-Line Help 8](#_44sinio)

[Project Risks 8](#_2jxsxqh)

[Customer Responsibilities 8](#_z337ya)

[Status Reporting 8](#_3j2qqm3)

[System General Description 9](#_1y810tw)

[Project Summary 9](#_4i7ojhp)

[Major Subsystems 11](#_2xcytpi)

[Relation of System to Existing System(s) 11](#_1ci93xb)

[Hardware Platform Description 11](#_3whwml4)

[Software Platform Description 11](#_2bn6wsx)

[Third Party Libraries 11](#_qsh70q)

[Product Requirements 12](#_3as4poj)

[Functional 12](#_1pxezwc)

[Performance 14](#_49x2ik5)

[Reliability 14](#_2p2csry)

[Data Description 14](#_147n2zr)

[Security/Safety 14](#_3o7alnk)

[Constraints 14](#_23ckvvd)

[User Profiles 14](#_ihv636)

[Glossary 15](#_32hioqz)

[Appendix A – Status Report Form 16](#_41mghml)

# Introduction

## Purpose

The purpose of this document is to practice writing a software proposal as required by CST 334. This document will be a constantly evolving reference guide to the production and distribution of the project described herein.

## Scope

The scope of this document follows the requirements described in the Senior Project Proposal Outline and include the general description, product requirements, user profiles, and project management procedure.

## Intended Audience

The intended audience of this project is Calvin Caldwell, the instructor for CST 334 and Jesse Kinder, the physics professor who will be working closely with the author to ensure the authors understanding of the fundamental properties required from this project as well as a contact and supplier of required hardware.

# Project Management

## Change Management Procedure

### CAT Team

The CAT team will consist of Jacob Chesley, Calvin Caldwell, and Jesse Kinder.

### Medium

Any changes after the initial rough draft submission must be submitted in a hard copy version to Calvin Caldwell, either in person or to his mail box. Any hardware changes or future analysis changes require a digital copy to be submitted to Jesse Kinder for review before submission to Calvin Caldwell may take place.

### Protocol

From June 15th 2018 to December 30th 2018, all submissions will take place electronically to both Calvin Caldwell and Jesse Kinder and a response given by the next Thursday evening after submission if the submission takes place before Monday morning. If submitted after Monday morning, a response will be given by the Thursday of the week after. Submissions made after December 30th 2018 will be in hard copy form and hand delivered to Calvin Caldwell while remaining in digital form for any submissions to Jesse Kinder.

### Impact Analysis

Any changes made to a submission must be accepted by Calvin Caldwell. Any changes made to the hardware portion of the project must also be accepted by Jesse Kinder as well. Any proposed changes that are financially unfeasible or extend the time table past the projects final submission deadline will be denied.

### Archive

All changes made will be stored virtually in Jacob Chesley’s school OneDrive account with File History as well as a hard copy stored at the authors house at all times.

## Software Delivery, Installation and Acceptance Criteria

The project will be delivered through submission and subsequent download from the Google Play Store. Any updates will be automatically downloaded once submitted to the Google Play Store. Connection to the test hardware will be through a hard-coded GUID in the Android application that will be changed upon production release. Acceptance of the project will depend upon the functionality of the application and hardware as specified in the Product Requirements page of the proposal.

## Documentation and On-Line Help

The project’s source code will be stored in a GitHub repository that both Calvin Caldwell and Jesse Kinder will have direct access to. All code will be commented before submission to the repository and the initial installation of the Android application for Calvin Caldwell will be in person to ensure compatibility issues that may arise are quickly remedied.

## Project Risks

The hardware required for such precise measurements required by this project may be cost prohibitive as the amount of equipment and specialized hardware may increase. The author is familiar with the physics relating to the project, but not the possible interactions multiple measurements may have on each other. The author also has little knowledge of Wi-Fi communication between hardware devices.

## Customer Responsibilities

The customer is responsible in knowing their hardware’s GUID to connect with the application and the ownership of an Android device that can connect to the internet without trouble.

## Status Reporting

Unless otherwise stated, status reports will be submitted by 11:59 PM Thursday night. To focus on a simplified reporting system, the format will be the same as was used in Junior Project created by Todd Breedlove. Each report will include a list of last weeks tasks, next weeks tasks, and priority of completion. An example of this status report is included in Appendix A.

## System General Description

### Project Summary

One problem my family has always struggled with was the control system for their lawn irrigation system. It only allows static settings and hardware input. It enables timing and length of irrigation. Even so, the system is hard to understand and often works unexpectedly. This project be the solution to this problem. Many underground irrigation systems (as opposed to one connected to a hose) are now controlled electronically with a microcontroller and some sort of hardware input. My senior project will Raspberry Pi v.3 with a Wi-Fi connection. The Wi-Fi connection works two-fold; the microcontroller may ping googles weather API for its location to measure the cloud cover, temperature, humidity, expected precipitation, and daylight hours, it will then decide the best time to run the irrigation system and how often to run it.

The second reason for the Wi-Fi enabled microcontroller, is to communicate with the application. Each time the micro-controller connects with the internet, it will ping the server to check whether any settings have been updated from the application. If not, then it will continue to ping the Google weather API. If so, it will either update the times it runs according to the settings or enter manual mode and downloads the irrigation schedule that the user has saved to the server. As the microcontroller contains enough processing power and memory to run windows 10, the microcontroller will run a looping script every fifteen minutes updating its irrigation schedule. Upon a power loss, the microcontroller will reboot and automatically restart the script to maintain continuity.

Autonomous mode pulls data from the National Oceanic and Atmospheric Administration (NOAA) free API and (through an initial bit of personal guess and check), calculate the best times to run the irrigation system to minimize evaporation, maintain preferred soil moisture, and obey local water usage laws. Each time it pings the Google API, it also checks the server for any updates to its settings. These settings include a switch to manual mode, specified “off hours” where the system should avoid running on, password update, and user designated water usage scaling which is described in the next section. This functionality only requires a location to be set and a password to be saved by the user in the android application. After which, the microcontroller may run without a connection to the application.

For reviewing statistics, the microcontroller will periodically signal an off-site database on its water usage, times ran, and Wi-Fi usage. The application will then call the server using the RESTful API and receive a JSON object. Parsing this information, the statistics page will display standard statistics including run time, total water usage, average daily water usage, and next time to run. Below, graphs showing water usage over time, Wi-Fi usage over time, times run, and currently calculated future run times (if in autonomous mode). If the user believes that the timing of irrigation is correct, but the amount of irrigation is off, then there is a slider to change the run time of each irrigation cycle +- 100%.

Manual control switches the microcontroller off-of autonomous control and enables the user to manually set when to irrigate, how much to irrigate, and how often the microcontroller pings the server for an update. Saving the settings will save them to the server and bind the settings to the GUID. The next time the microcontroller connects to the server, the settings will be uploaded and stored locally on the microcontroller, so a constant connection is not required. The microcontroller then periodically pings the server for any updates. The user may also run a test to run the irrigation system immediately for a set of time to ensure that the irrigation system is functioning properly.

## Major Subsystems

The project will have four major subsystems: The android application, the database, the laser hardware, the sensor hardware. The application will include the user interface, the database is comprised of user information and resource data, and the hardware collects the sensor data.

## Relation of System to Existing System(s)

None

## Hardware Platform Description

A custom build device consisting of three lasers at 375 nm, 850 nm, and 400nm, and three light sensors to each corresponding laser as well as a Wi-Fi communication terminal for sending test specifications and receiving test results.

## Software Platform Description

The Android application will be targeted for Android devices from versions 6.0 to version 8.1.

## Third Party Libraries

The android application will utilize “numpy” from the python library.

# Product Requirements

## Hardware

Description: The Raspberry Pi 3 Model B+ is a Wi-Fi enabled microcontroller that enables real time diagnostics through remote desktop and high-end specifications packaged in a small form factor.

Processor: Broadcom BCM2837B0, Cortex-A53 (ARMv8) 64-bit SoC @ 1.4GHz

Memory: 1 GB DDR2 SDRAM

Wireless: 2.4 and 5 GHz 802.11 b/g/n/ac LAN, Bluetooth Low Energy, and Bluetooth 4.2

Hardline: Gigabit ethernet up to 300 Mbps

Input: 40-pin GPIO header, HDMI, USB 2.0 x 4 ports, DSI touchscreen input support

Power: 5V/2.5A DC input

External Storage: Micro SD enabled

## Functional

1. Android application will be programmed using Android Studio and the Java language
   1. All application-server communication will be performed using the REST API and JSON objects
   2. Enumerated time span list will include hour, day, week, month, and year
2. Upon launching, application loads a splash screen
   1. Splash screen will contain the applications icon and a white background
      1. In another thread, the login screen will load in the background
      2. Single API test call consisting of an empty JSON object will be sent to the server
      3. A port will be opened from this call and remain open until the application closes, this is to decrease lag time in subsequent communication with the server
   2. Splash screen closes, and login screen appears
3. Login screen contains:
   1. Username/Password fields
      1. Password field will have the password condition set in the XML file
   2. Forgot password:
      1. Clicking creates a popup with a username and email field
      2. Clicking either accept of cancel
         1. Clicking accept checks both fields are filled
         2. Application will request an account with email and username to be sent a password recovery email
            1. Email consists of a link to send a confirmation JSON call
   3. Create account:
      1. Fields consist of username, email, password, password confirmation, and username
         1. Email will check for formatting and throw a popup error to fix the email
         2. Passwords will check they are the same
         3. Username field checks to see if email and username fields are filled
            1. If filled, ping server to check that both are available

Display warning if one or both are unavailable

* + - 1. Accepting will add a new user to the database and a confirmation email will be sent
    1. Confirmation email must be confirmed for a boolean field on the database to be set to active. The user may not log in without an active account.
  1. Log in:
     1. Request user account from server
     2. Application will check boolean if account is active
        1. If so, the main page will load

1. Main page
   1. Bottom menu bar with four clickable icons
   2. Clicking each icon switches between fragments
   3. Icons are:
      1. Status page (default view upon login):
         1. Current connectivity to irrigation system
         2. Display will be blue light for connected, red for disconnection, green for running
            1. Clicking will attempt to reconnect

Successful reconnection changes light to blue

Unsuccessful reconnection will display respective error

* + - 1. Next expected watering displayed as a countdown timer
         1. Clicking opens popup checks if in autonomous mode or manual mode

If manual mode, opens Manual Control fragment

If in autonomous mode, display results of National Oceanic and Atmospheric Administration (NOAA) weather API call

* + - 1. Total run time by time span
         1. Clicking will cycle through the time span list
      2. GUID/Nickname of current irrigation system
         1. Clicking will open the system list fragment (described in 4.3.3)
    1. Statistics page (4 graphs):
       1. Clicking any icon will change the graph time span cycling through the list of time spans
       2. When displaying day, 24 data points to create a 2-D line graph, otherwise, each data point will be per day
       3. Holding click will full screen the selected statistic
          1. Graph 1 (water usage) cumulative water usage in gallons over time
          2. Graph 2 (connectivity) Number of successful periodic checks the micro-controller makes to the server to check for updates
          3. Graph 3 (Soil moisture) Soil moisture over time measured one hour after each watering
          4. Graph 4 (Scheduled run time) Past and future runtime where half way mark on horizontal plane is present (time = 0).
       4. A slider at the bottom default set to center (0)
          1. Sliding from +- 100 saves to the server a percent change to scale length or each run time. Can used in manual or autonomous mode.
    2. System page:
       1. Application calls server requesting list of irrigation systems linked to account
       2. List displayed in alphabetically descending order of nicknames
       3. Lines include:
          1. Radio button to select which system to control
          2. Editable field of nickname for system
          3. Non-editable field containing GUID
          4. Last time system communicated with server
          5. Time added to account
       4. Create button at top
          1. Clicking creates popup for GUID field and Nickname field
          2. Save button checks with server if GUID exists and is unclaimed

If successful, popup closes, and list reloads

Failure displays error popup with warning but keeps first popup open

* + - 1. Delete button to right of Create button
         1. Clicking creates popup for nickname field
         2. Save button checks if nickname is valid nickname in list

If valid, contact server to remove GUID from user’s owned irrigation systems

* + - 1. Account settings button at bottom of page
         1. Upgrade User button

Clicking brings user to PayPal widget

Upon filling out information PayPal returns to app

Server is notified and user account is updated to Enthusiast tier

* + - * 1. Enter Manager Mode button

Clicking checks if account is Enterprise tier and if user is the manager

Load manager window (described in 5)

* + - * 1. Delete Account button

Password verification required

Disabled if account is Enterprise Edition (Requires contact to support to delete)

* + - * 1. Contact support button at bottom

Default email app loads with support email and user email preloaded into required fields

* + 1. Manual Control page:
       1. Toggle at top changes server preferences between manual and autonomous irrigation control
       2. When in manual control, calendar become editable
       3. Week long calendar with each day selectable
          1. Selecting day opens a sliding schedule from 12:00AM - 11:45PM with fields incrementing in 15-minute intervals
          2. Selecting field pops up a required end time

Check box requesting if all seven days should follow this schedule

Pressing back or save will save preferences to server

* + - * 1. Calendar will update to display scheduled run times in green
      1. Quick run button at bottom
         1. System will run for 30 minutes
         2. Clicking again will stop system

1. Manager page
   1. Add technician
      1. Clicking brings new window with contents:
         1. Email field
         2. List of permissions including
            1. Add systems
            2. Remove systems
            3. Update system schedule (if in manual)
            4. Change system mode (to/from manual)
            5. Quick Run system
            6. View usage statistics
         3. Email sent that requires user to create account with email field locked to select email
   2. Remove technician
      1. Field requires email
         1. Manager password confirmation window appears
   3. Update technician
      1. Manage roles of technician
   4. Promote technician
      1. Email field filled out of active technician to become permanent manager
         1. Manager password confirmation window appears
         2. Manager will be demoted to technician will all roles active

## Performance

The application is targeted to utilize less then 250 MB of heap memory, so it may work on lower end phones. Any computational renderings are targeted to happen in a separate thread and complete to update the UI layer within half a second of being created. Any images that are used in the application will be dynamically scaled to the users’ personal phone aspect ratio before loading into the heap.

## Reliability

For the hardware to remain useful, it must remain connected to the server 100% of the time and the server must always have a stable connection to the internet.

## Data Description

Data communication will utilize JSON object text with a maximum data rate of 128Kb per hour. A total of four requests per hour with a single request every quarter hour will be made with a maximum response of 15Kb per response. The need for minimal network latency will be minimal. The APK will remain below 50 MB and allocate a maximum of 256 Mb of ram from the users’ phone.

## Security/Safety

The application requires a stable internet connection and the hardware’s GUID to connect. Without either, the application will be unable to connect. The user must also log on to a profile where the users password will be saved server side in the form of a salted and hashed binary object. The password will be hashed client side and the server will never receive a plain text password.

## Constraints

The hardware must conform to the FCC Part 15 rule set where it must accept any interference produced by other equipment and not produce any interference against other equipment.

## User Profiles

This application is targeted towards aquarium owners and researchers who require constant monitoring of particulate or impurity levels within a specific body of water.

# Glossary

**numpy:** A python data visualization library that can create multiple different graphs from vectored data.

**Federal Communication Commission (FCC):** The United States governmental body in charge of wireless broadcast limitations.

## Appendix A – Status Report Form

