Automated Gardening Project: System Administration Guide

Stephan Moncavage

CST-452 Capstone Project Proposal

Grand Canyon University

Instructor: Professor Michael Landreth

Revision: 0.1.0

Date: 05 June 2022

**TABLE OF CONTENTS**

Project Overview and Project Objectives 3

Project Scope 5

Agile Scope Example 6

Project Success Measures 7

Project High-Level Solution 8

Project Controls 12

Software Services 14

Appendix A – References 20

Appendix B – Copyright Compliance 23

**TABLE OF FIGURES**

[Figure 1 Agile SCRUM/Sprint Example 6](#_Toc105175405)

[Figure 2 Annual Climate Data for Zip code 30252 8](#_Toc105175406)

[Figure 3 UML Class Diagram 10](#_Toc105175407)

[Figure 4 Component Diagram 10](#_Toc105175408)

[Figure 5 Use Case Diagram 10](#_Toc105175409)

[Figure 6 Database Enhance Entity Relationship Design 11](#_Toc105175410)

[Figure 7 github.com/smoncavage/CST451\_Capstone.git 15](#_Toc105175411)

[Figure 8 github.com/smoncavage/CST451\_Capstone/actions 16](#_Toc105175412)

[Figure 9 gitlab.com/smoncavage/CST-Capstone-Project/ 16](#_Toc105175413)

[Figure 10 dashboard.gitguardian.com/ 17](#_Toc105175414)

[Figure 11 dashboard.heroku.com/pipelines 17](#_Toc105175415)

[Figure 12 smoncavage.loggly.com/ 17](#_Toc105175416)

[Figure 13 Dashboard.heroku.com/ 18](#_Toc105175417)

[Figure 14 moncavage.atlassian.net/jira/ 19](#_Toc105175418)

Project Overview and Project Objectives

**State the Problem and Background**

A previous project from a prior class for an Internet of Things (IoT) weather station will be used as a base code that will be converted into a different development language. It is a project which was not fully completed at the time and some additional features should be possible to add in along the way as well. The original project was developed in a small group and consists of taking Global Position System (GPS) coordinates as well as atmospheric data from a Raspberry Pi 4 and adding this data to a database through a Python script and then read by a Java web program for display to a user. The plan for this project is to take this concept and convert it into PHP while simultaneously enhancing the user experience. This project will be expanded upon to conceivably control water solenoids to determine whether a garden should be watered for proper fruit and vegetable growth while accounting for current and forecasted weather conditions.

**Christian Worldview**

Guided by a Christian worldview this project is being developed to support this concept through being a good steward of the land and resources provided by God. “As every man hath received the gift, even so minister the same one to another, as good stewards of the manifold grace of God.” 1 Peter 4:10 KJV. In this world we are and hold nothing of our own accord and owe all to God’s grace. By taking the gifts set upon us by God we can better utilize the resources on hand to help ensure a fruitful venture in a literal sense through this project.

**Benefits and Opportunities**

The benefits for this project include but are not limited to decreased water usage and subsequent cost, easier maintenance of a garden at home allowing for more quality time with family and being able to grow food which would otherwise purchase or bartered for. Through a viable design effort allowing for a continuation of development and cost reduction a hardware kit can be made available to other people. This would allow more people the opportunity to grow their own garden with a reduced effort and costs necessary to do so. Through enough sales, growth, and support better software and hardware support and technologies could be taken advantage of to better refine the product and expand upon up-to-date features and technologies.

Project Scope

1. Give a clear, concise statement that states the scope of the project. This should also include items that are to be out of scope.
   1. A PHP based website consisting of a modern design with a functional user interface for extracting reliable localized weather data obtained from a Raspberry Pi and compared to area forecasting to determine the best possible times for watering a garden and how long to water them for. Full scale hardware design and functionality is not to be included in this process due to current time and budget constraints, as many options are available for a fully functional hardware kit. Focus is to be given to the IoT phase of the overall expected outcome.
2. Use the template to list all known stakeholders and contacts, if applicable, including self (for some projects self may be the only name listed)

|  |  |  |
| --- | --- | --- |
| Stakeholder Name | Role(s) | Responsibilities |
|  |  |  |
|  |  |  |

1. List the work breakdown required to satisfy the project objectives. Identify teams and other resources that may be required to successfully complete the project.

Agile Scope Example

Table

Description automatically generated

Figure 1 Agile SCRUM/Sprint Example

Project Success Measures

1. Use the template to list the project completion criteria.

|  |
| --- |
| Project Completion Criteria |
| 1 – Planning Documentation |
| 2 – Implementation Plan |
| 3 – Architecture Plan |
| 4 – Test Plan |
| 5 – Deployment Plan |
| 6 – Publish |

1. Use the template to list the project assumptions and constraints, if applicable. An assumption is an educated guess that a likely condition or circumstance is presumed to be true. A constraint is a limiting condition or circumstance that defines the project boundaries. Assumptions allow the project to succeed. Constraints restrict or limit the project execution.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Assumptions and Constraints | | | | | |
| ID | Description | Comments | Type | Status | Date Entered |
| 1 |  |  |  |  |  |
| 2 |  |  |  |  |  |

Project High-Level Solution

**Introduction**

To ensure a fully watered garden consisting of various fruits, vegetables, and trees, an automated solution is the ideal scenario which allows users to focus on other things without the necessity to constantly check weather forecasts and turn current available program controllers off manually or allowing them to constantly run-on schedule resulting in over watering.

Graphical user interface, application

Description automatically generated

Figure 2 Annual Climate Data for Zip code 30252 from <https://www.plantmaps.com/30252?msclkid=66e99ca1c32d11ecbcd252689170e23c>

According to perfect-vegetable-garden.com a rough calculation for proper water in garden soil is approximately 0.62 gallons per 1 square foot of garden which is also approximately equal to 1 inch of water per week. Given that the average annual rainfall for the 30252-zip code is 49 inches per year, this roughly means that a garden should only require approximately 3 inches of external watering per year. This however can be unpredictable in that weather does not cooperate on a schedule. Current commercially available water timer controllers only allow for manual rain overrides.

**Solution**

UML, Component Design, Use Case, and Entity Relationship diagrams are available below in figures 2-5. This project will expand upon these concepts and include additional data tables, software functionality and further development of the current User Interface (UI) in figure 6. This project takes inputs from a GPS module, an environmental module, and a weather API for localized weather forecasting based upon the GPS coordinates to determine whether a garden should be watered or not on any given day automatically without requiring a user to intervene in the process. The updated website will list their available device controller to the end user and show data for monitoring the last known date and time the system ran as well as when the next scheduled watering cycle will take place. Graphical user interface, application, Word

Description automatically generated

Figure 3 UML Class Diagram

Timeline

Description automatically generated

Figure 4 Component Diagram

Diagram

Description automatically generated

Figure 5 Use Case Diagram

A picture containing diagram

Description automatically generated

Figure 6 Database Enhance Entity Relationship Design

Project Controls

1. Use the template to define the risk and list the steps to prevent the risk from occurring or the steps to minimize the chances of it happening. The contingency plan describes alternative solutions to reduce the impact of the risk. An example of a contingency plan is to provide the customer a temporary web server if there are delays in delivery/completion. If the risk has already happened, then provide an entry in the issue log.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Risk Management | | | | |
|  | **Risk Probability** | **Risk Impact** |  |  |
| **Event Risk** | **(High, medium, low)** | **Risk Mitigation** | **Contingency Plan** |
| What is the risk? | What is the probability? | What is the impact if the risk occurs? | What can be done to minimize the risk? | What can be done to minimize the impact of the risk? |
| Timeline | High | Unfinished Product Publication | Strict adherence to schedule | Provide unfinished milestones for turn in. |
| Electronic Concept Design | Low | Electronic Design must be error free for full functionality | Verify Schematics via internal tools to test circuits before finalization | Use an external source for schematic data. |
| Additional External Tool | Low | It may be necessary for more external tools to be added to project | Fully verify that an additional external tool is necessary prior to adding in. | Develop a personal solution for software issues, re-design hardware issues |

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Issues Log | | | | | | | | |
| **ID** | **Description** | **Project Impact** | **Action Plan/Resolution** | **Owner** | **Importance** | **Date Entered** | **Date to Review** | **Date Resolved** |
| 1 | What is the issue? | How will this impact scope, schedule & cost? | How do you intend to deal with this issue? | Who manages this issue? |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |
| 3 |  |  |  |  |  |  |  |  |

1. All projects have either anticipated and planned or unexpected changes. Describe any issues in management or change management due to the anticipated and planned or unexpected changes. Use the template to list anticipated and planned or unexpected changes.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Change Control Log | | | | | | | | | |
| **ID** | **Change Description** | **Priority** | **Originator** | **Date Entered** | **Date Assigned** | **Evaluator** | **Status** | **Date of Decision** | **Included in Rev. #** |
| 1 |  |  |  |  |  |  |  |  |  |
| 2 |  |  |  |  |  |  |  |  |  |

1. Use the template to describe how the end user is involved in the software development, if applicable. Include relevant information about meetings, reviews, presentations, etc.

|  |  |  |  |
| --- | --- | --- | --- |
| Roles and Responsibilities | | | |
| Name | Team | Project Role | Responsibility |
|  |  |  |  |
|  |  |  |  |

Software Services

Below is a table of the necessary software services and tools necessary in order to maintain the Automated Garden Watering project.



The main repository is hosted on GitHub @ github.com/smoncavage/CST451\_Capstone.git

Graphical user interface, text

Description automatically generated

Figure 7 github.com/smoncavage/CST451\_Capstone.git

GitHub’s built in “Actions” will run the PHP Composer workflow to verify that the repository has the latest versions of the necessary PHP packages for the application.

A screenshot of a computer screen

Description automatically generated with medium confidence

Figure 8 github.com/smoncavage/CST451\_Capstone/actions

Simultaneously, the repository is mirrored on GitLab which runs more CI/CD implementations. This can be found at gitlab.com/smoncavage/CST-Capstone-Project

A picture containing application

Description automatically generated

Figure 9 gitlab.com/smoncavage/CST-Capstone-Project/-/pipelines

GitLab will run a “build”, “unit-test”, “lint-test”, and “deploy” job within the pipeline.

Test scripts can be added for unit and system testing in each of the testing jobs respectively.

At the same time GitGuardian will also test for any “secret” data in the repository such as database login information and API keys.

Graphical user interface, application, Teams

Description automatically generated

Figure 10 dashboard.gitguardian.com/workspace/172387/incidents/secrets

Once these tests have been completed with a passing result the project is passed over to Heroku, where the application is made public through their servers.

Graphical user interface, text, application

Description automatically generated

Figure 11 dashboard.heroku.com/pipelines

Heroku hold the main database in “JawsDB MySQL”, the security certificate for the website through “BrandSSL”, and part of the logging infrastructure with “Papertrail”. The other portion of the logging infrastructure loads into “Loggly” and not through Papertrail which is an extension of the "Loggly” service.

Graphical user interface, text, application

Description automatically generated

Figure 12 smoncavage.loggly.com/search#terms=syslog.host:host

Graphical user interface, text, application, email

Description automatically generated

Figure 13 ashboard.heroku.com/apps/cst451-capstone

All of this is completely automated and does not require daily oversight or maintenance unless new functionality or testing is added for the web-application.

When this is needed the system administrator will need to update the necessary portions of the CI/CD pipeline to implement the new tests or functionality.

If at any point in the pipeline, there is a failure and/or errors, each phase of the pipeline is also connected to Jira which will automatically implement a new “Issue” or “Bug Report” to the main project’s Agile SCRUM Roadmap and Backlog. These issues can then be seen by the system administrator or designated person to troubleshoot the issues and errors.

**Table

Description automatically generated**

Figure 14 moncavage.atlassian.net/jira/software/projects/MC/issues/

**Project and Implementation Schedule**

* 2 Week Sprint in Agile SCRUM environment hosted by Jira are recommended.

Appendix A – References

Atlassian. (2022, March 15). *For-jira/readme.md at main · Atlassian/github-for-jira*. GitHub. Retrieved April 23, 2022, from https://github.com/atlassian/github-for-jira/blob/main/README.md

Atlassian. (n.d.). *Products*. Atlassian. Retrieved April 23, 2022, from https://www.atlassian.com/software

*Bible gateway passage: 1 peter 4:10 - king James Version*. Bible Gateway. (n.d.). Retrieved April 23, 2022, from https://www.biblegateway.com/passage/?search=1+Peter+4%3A10&version=KJV

*Dashboard*. Heroku. (n.d.). Retrieved April 23, 2022, from https://dashboard.heroku.com/apps

*Git security scanning & secrets detection*. GitGuardian. (n.d.). Retrieved April 23, 2022, from https://www.gitguardian.com/?utm\_feeditemid=&utm\_device=c&utm\_term=gitguardian&utm\_source=bing&utm\_medium=cpc&utm\_campaign=US+%7C+GitGuardian+Brand&hsa\_cam=10399074694&hsa\_grp=1343603812663068&hsa\_mt=e&hsa\_src=o&hsa\_ad=&hsa\_acc=5867098142&hsa\_net=adwords&hsa\_kw=gitguardian&hsa\_tgt=kwd-83975619750113%3Aloc-190&hsa\_ver=3&utm\_adgroupid=1343603812663068&msclkid=fedc9efc9695194b76c0fc98a6c7bbda&utm\_content=GG+Brand+Exact

Goettsche Partners. (2011). *GP*. Amazon. Retrieved April 23, 2022, from https://smile.amazon.com/gp/charity/homepage.html/ref=smi\_ge\_raas\_btngs\_rl?ie=UTF8&newts=1

Jim, A. (2019, January 26). *How much water does a garden need?* Here's an easy way to find out! Retrieved April 23, 2022, from https://perfect-vegetable-garden.com/water-garden-inch-per-week#:~:text=Hence%2C%20each%20square%20foot%20of%20garden%2C%20which%20is,is%20especially%20beneficial%20to%20all%20container%20gardening%20enthusiasts.?msclkid=f6aabe87c32d11eca43be49b9a78564e

*KiCad Eda*. Schematic Capture & PCB Design Software. (n.d.). Retrieved April 23, 2022, from https://www.kicad.org/?msclkid=4cc11ae9c32711ecaba200f89bd7b725

Moncavage, S. (2022, April 17). *Roadmap*. Agile Board - Jira. Retrieved April 23, 2022, from https://moncavage.atlassian.net/jira/software/projects/MC/boards/1/roadmap?timeline=WEEKS

Moncavage, S. (2022, April 23). *Smoncavage/cst451\_capstone: Senior capstone project - weather based garden watering system*. GitHub. Retrieved April 23, 2022, from https://github.com/smoncavage/CST451\_Capstone.git

*MySQL Workbench*. MySQL. (n.d.). Retrieved April 23, 2022, from https://www.mysql.com/products/workbench/?msclkid=1a63eb01c32711ec856610783e71b303

OpenWeatherMap.org. (n.d.). *Weather API*. OpenWeatherMap. Retrieved April 23, 2022, from https://openweathermap.org/api?msclkid=8ad82b8ac32511ecb5b47e95530f22f0

*PhpStorm: The lightning-smart ide for PHP programming by jetbrains*. JetBrains. (n.d.). Retrieved April 23, 2022, from https://www.jetbrains.com/phpstorm/

*Zipcode 30252 - McDonough Georgia is in hardiness zones 8a*. Zipcode 30252 - McDonough, Georgia Hardiness Zones. (n.d.). Retrieved April 23, 2022, from https://www.plantmaps.com/30252?msclkid=66e99ca1c32d11ecbcd252689170e23c

Appendix B – Copyright Compliance

Below is a list of tools that will be used which are all free for personal development as well as student usage. OpenWeather is the only external data platform incorporation using their API calls to bring in forecasted weather data into the application. Without this functionality it would not be possible to accurately predict whether water should flow to the garden on a particular day. All other tools are development tools to write, test, track issues and schedules, and publish the application.

Atlassian. (n.d.). *Jira pricing - monthly and annual subscription cost per user*. Atlassian. Retrieved April 23, 2022, from https://www.atlassian.com/software/jira/pricing

*Buy PhpStorm: Pricing and licensing, Discounts - JetBrains Toolbox Subscription*. JetBrains. (n.d.). Retrieved April 23, 2022, from https://www.jetbrains.com/phpstorm/buy/#discounts?billing=monthly

*Libraries license*. KiCad EDA. (n.d.). Retrieved April 23, 2022, from https://www.kicad.org/libraries/license/

*MySQL Workbench* ". MySQL. (n.d.). Retrieved April 23, 2022, from https://www.mysql.com/products/workbench/

OpenWeatherMap.org. (n.d.). *Pricing*. OpenWeatherMap. Retrieved April 23, 2022, from https://openweathermap.org/price

*Personal & Team Billing and Usage*. Heroku help. (n.d.). Retrieved April 23, 2022, from https://help.heroku.com/n/320/billing-verification-payments/personal-team-billing-usage

*Pricing · plans for every developer*. GitHub. (n.d.). Retrieved April 23, 2022, from https://github.com/pricing

*Pricing*. GitGuardian. (n.d.). Retrieved April 23, 2022, from https://www.gitguardian.com/pricing