SDP 2020: Team 19
Project: Water Wizard

Andrew Grote, EE, Aneurin Sutton, EE, Zach Toscanini, EE, and Parth Nanavati, CSE

Abstract:

We are an agrarian people. For centuries, we have thrived because we mastered the art of agriculture. We should take pride in the fact that we are one of the only species in the whole world who grows their food. In a fast-paced society, most people are too busy to manage their plants themselves. On top of that, most plants have very specific watering needs; needs most people don’t understand. Poor watering is the number one issue consumers face in growing plants.

Water Wizard is our solution. Water Wizard is a small device equipped with a pump and moisture sensor that automatically waters a plant, according to its specific needs. It can monitor multiple plants at once, and can be programmed and controlled by the user with their smartphone. The product will meet the specific needs of plants, and remain cheap for the sake of scalability.

Introduction:

A. Significance

Most consumers don't realize that plants have very specific needs when it comes to their water consumption. People often assume they even need water their plants daily, but there are very few common houseplant plants that actually need that much water. Such frequent watering will even kill some common houseplants. According to Chris Spilly, a local greenhouse expert on UMass Amherst’s Campus, overwatering and underwatering are the primary issues consumers face when growing plants.

B. Context and Existing Products

Automatic watering is a problem we have addressed since our early Mesopotamian start. We like to grow our own food, but watering is extremely tedious. Thus, we have tried to devise a means of doing so automatically.

One of these methods- the first method ever invented - is irrigation. Irrigation at first conception, just consisted of canals used for controlled flooding and has evolved into huge networks of pipes which distribute water either through with a drip architecture or timed release. Traditional Irrigation is used for industrial farming. They were a very effective solution for watering hundreds (or even thousands) of plants in an automated fashion. However, these systems tend to be huge. They are difficult to scale down for a consumer, and would be nearly impossible to set up in an apartment. They are simply not fit for a consumer. On top of that, they are very simple systems. Traditional irrigation methods include canals, drip release, and timer release. Most canals systems just flood the plants in a controlled manner that prevents overwatering, but this method wastes tons of water. Drip releases and timer release irrigation are more sophisticated, and waste less water, but still do not meet the individual needs of plants. Irrigation simply can not meet the needs of individual plants, and can not be scaled for consumer use easily.

Another automated watering product that has shown popularity over the years are aqua bulbs. Aqua bulbs are pretty bulb shaped glass ornaments which use the limited water supply in their bulb to water plants automatically. The user fills the bulb and pokes it into the soil, after which the bulb waters the soil until it's “too wet”, at which water stops leaving the bulb. This is a very unsophisticated mechanical feedback systems which waters when the soil gets dry. In that way it resembles Water Wizard. The major difference, is that the Aqua Bulb’s feedback is not programmable or controllable. The threshold at which the water stops leaving the bulb is determined by the width of the bulb’s neck and the compactness of the soil. On top of all that, the products mechanical components simply don't work well. The neck of the aqua bulbs tend to clog, and prevent water maintenance. Though it's convenient and scaled to a consumer's needs, it is too unsophisticated for optimal plant growth, and needs more robust mechanical components.

Another product out there is arduino based sensor systems. This is perhaps a far more primitive version of Water Wizard. These "systems" are often kits to be used with arduino. They are composed of moisture sensors and some simple controls for a pump. When properly assembled and programmed, these devices are capable of meeting low level programmable feedback for plants. These devices are extremely primitive and meet very simple needs--they water to one simple threshold, and can only handle one plant at a time. Our devices will meet sophisticated needs of multiple plants. On top of that, these devices are not complete products. They are just the sensor, and some pump controls. You have to buy the other mechanical components yourself, and assemble yourself. This greatly limits their accessibility to consumers, making them unfeasible as a solution.

An automatic watering systems has been attempted before, but the best solutions out there are designed for industrial use or simply don't work. There is a clear and demonstrated need for this product. It has been attempted by many, but it is poorly executed. Our major selling points are accessibility and sophistication.

C. Societal Impacts

Our Water Wizard is designed for the average household consumer. It can be sold ready-made in a box on a garden store shelf, and it is easy to use. Our casual conversations with the general public have revealed a high demand for such a product, and a large, untapped market. If successful, Water Wizard could become an integral component to any household arrangement of live plants. Further, we expect that more people will become interested in cultivating plants if our product makes it easier.

This invention serves to enhance and optimize existing household water usage by adding an electronic management system. It is environmentally benign and low impact. We anticipate that this invention will have significant benefits for consumers and plants, without any associated drawbacks.

D. Requirements Analysis and Specifications

The project's success is based on several criteria.

First, it is essential that Water Wizard tends multiple plants simultaneously. This enables the user to consolidate their efforts into one system, which distributes resources automatically.

Second, Water Wizard features a complete mechanical system for dispensing water. This includes a central reservoir for water storage, a collection of pumps and valves which are operated by the electronic controller, and flexible hoses for transmitting water to different plants.

Third, the heart of the project is the microcontroller. It is programmed to operate the mechanical systems according to sensor inputs and feedback algorithms, with user-adjustable performance and variable parameters pertaining to the individual needs of different plants.

Fourth, the user interacts with the controller using a convenient app on their mobile phone. This app receives instructions for plant care, and returns statistics on system performance.

All plants are uniquely adapted to their native environments, and there are as many variations between different plant species as there are between the different regions of the world. In a domestic situation, the environment is very homogeneous, and there is no natural precipitation.

Conversely, there is an incredible variety of plant species which are cultivated in households, with plants originating from all corners of the world. In houses, plants which would never be found together in nature are cultivated side by side, often with exactly the same watering patterns.

The most important goal of our project is to virtually replicate the rainfall patterns of a plant's natural environment as closely as possible. To this end, we introduce the idea of 'watering profiles.' A watering profile is a two-dimensional curve which plots the expected moisture content of a plant's soil against time. This curve is different for every plant.

The variations depend primarily on the plant's size and species, but they also depend on planting conditions, such as soil and pot size. The ideal watering profile perfectly reflects the precipitation patterns of a plant's natural environment.

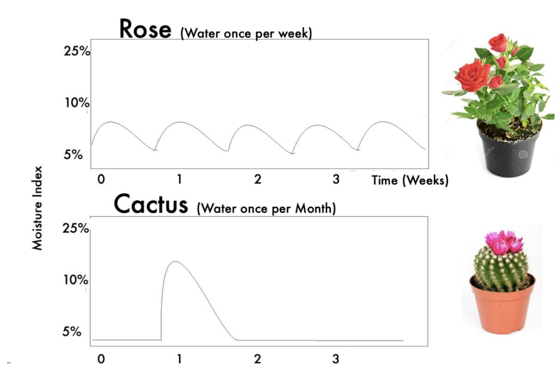


Figure A: Ideal watering profiles for two different plants

We propose that a successful project will automatically operate mechanical watering systems so that the soil's actual moisture content will be within 10% of the ideal watering profile, at least 80% of the time.

The primary considerations of plant size and species can be pre-programmed and scaled, but the non-trivial variations due to planting conditions must be corrected using real-time feedback control. Over time, our system learns more about these conditions and improves performance by adjusting watering frequency and duration.

Our system is designed for the average consumer, so it must be affordable and easy to use. The finished design transmits information via bluetooth communications, which are compatible with most mobile devices. Using it requires only a domestic power source and periodic replacement of the water supply. The finished design should cost under $50.00.

Design:

A. Overview

The Water Wizard is a collection of watering sub-units which are simultaneously controlled by a main control unit.

The Water Wizard will be composed of four connected subsystems: Water Supply Control, UI/Database, Watering Unit, and the Control unit, all of which can be found below in Figure B.

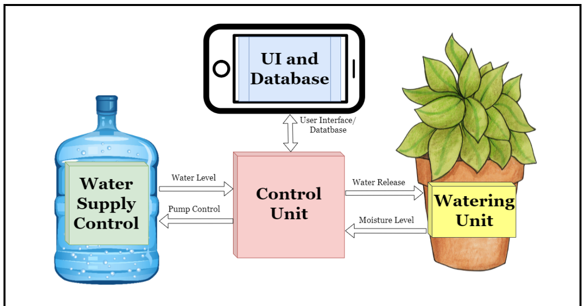
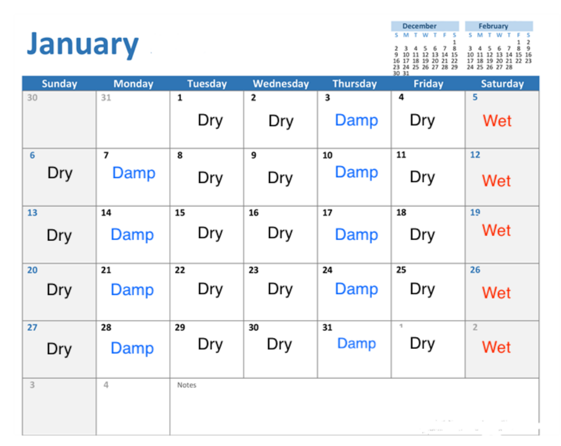


Figure B: Overview Block Diagram

Each plant the user has set up will be associated with an individual watering unit, which contains the mechanical components to test the soil as well as release water into the soil. The control unit monitors the plants moisture levels and releases water accordingly. The watering units release water which is sent by the Water Supply Control UNit of which there is only one for the whole system. This water supply control unit maintains pressure in the lines as needed, so that when the Watering Unit gets the cue to water from the control unit, the water is properly released. The control unit manages all plants simultaneously, making sure each is watered in accordance with their associated water profile. The control unit is programmed through an app on a smartphone, which enables the user to easily interface with the device. The control unit communicates with the phone through robust and reliable bluetooth communications.

The most significant aspects of this project lie in our use of Water Profiles. The Water profiles are a set of watering conditions that need to be met for optimal plant growth. These watering conditions most importantly include, moisture levels in the soil, as well as frequency. More specifically water profiles will look like a month long array, where each day is associated with a soil moisture level. It ends up looking a lot like a calendar. Below in figure C you can see a simple abstraction of these water profiles. The water wizard essentially follow this calendar like the user would with more precise watering than a human can achieve.



January

Sunday	Monday	Tuesday	Wednesday	Thursday	Friday	Saturday
30	31	1 Dry	2 Dry	3 Damp	4 Dry	5 Wet
6 Dry	7 Damp	8 Dry	9 Dry	10 Damp	11 Dry	12 Wet
13 Dry	14 Damp	15 Dry	16 Dry	17 Damp	18 Dry	19 Wet
20 Dry	21 Damp	22 Dry	23 Dry	24 Damp	25 Dry	26 Wet
27 Dry	28 Damp	29 Dry	30 Dry	31 Damp	1 Dry	2 Wet
3	4	Notes				

Figure C: Water profile Abstraction

B. Watering Unit

The watering unit is a simple pump, which operates on a binary on/off basis as directed by the controller. Our final project will feature valves which direct waterflow, and drip-irrigation caps which ensure an optimal water dispersion rate.

Our reservoir includes a water level sensor, which detects and displays the amount of water it contains, and notifies the user if the water supply needs replenishing

C. Control Unit

The control unit is the main unit which controls all peripherals.It Tracks all plants water cycles internally, providing feedback. It can be seen in Figure D below.

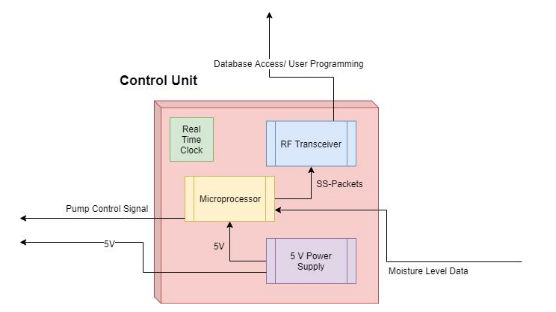


Figure D: Control Unit

(Note: the abstract block diagram provides additional context as to where the stray signals are going)

The controller consists of a microprocessor, an RF Transceiver, a 9V Power Supply, and a real time clock module. Right now an Arduino Uno is taking the place of our Microprocessor. The Arduino contains all of the software. It is in this software that the plants are individually managed. Each plant is paired with a water profile, whose moisture indexes are met with Proportional Feedback Controller. The pumps and valves are all managed from this arduino using control signals based on the Feedback parameters.

Additionally, all communications with the user are managed and executed by the arduino using an HC-05 Bluetooth Chip. The bluetooth chip communicates using simple serial communication. Even in professional systems, bluetooth systems are error prone and fail often. In order to combat this error risk I designed a simple transmission protocol for sending packets of data: I called it Semaphore Secured Packet Communications (SSC for short). The advantage to SSC is that the software will only receive packets who end and start with my Semaphore Character. When the Arduino receives bytes, it checks that the first character is ‘#’. If the first character is not ‘#’, the program ignores any incoming bytes until said character is received. If the first character is ‘#’, the process of reading bytes can not be interrupted by any other processes, besides an enabled timeout interrupt. During this process, the program expects a 4 byte opcode and 32 byte payload, making the packet total 38 bytes. This protocol ensures the communication are reliable and trustworthy.

(Note: The proportional feedback is not fully implemented yet, so there is no experimental data acquired regarding it.)

D. User Interface

We created an app which interacts with the Bluetooth device within the circuit. It's a very simple interactive app that starts and stops the watering system based on the feedback it receives from the moisture level. The app contains 3 different modes. The first button starts programming the first water profile. The second button puts the systems in watering mode, this allows the system to check whether the "plant" needs water or not. The third button is a buffer or clear button for the system.

E. Database

The database is integrated directly into the app, because we only directly need these when we are programming the control unit. The Database will be a collection of common houseplants, and their associated water profiles. These profiles will be generated based in Appendix F.

Figure F1: Plant Care Specifications from Farmtek on data collected by expert Agricultural Engineers from Farmtek[1]. Some of the raw data that will be used to generate these profiles can be found in Figure F1.

Management:

Our MDR prototype was a complete success.
At the demonstration, we transmitted directives to the controller using the mobile interface. The controller activated multiple water pumps in response to sensor readings and water thresholds, and dispensed actual water to a virtual plant. We evaluated performance using our basic profile test, and the controller transmitted information back to the mobile device.

Our remaining work is to refine our measurements and specifications, and develop a program which is sophisticated enough to meet the proposed ideal water profile with error correction and feedback control. We also need a more informative, robust and user-friendly mobile app.

A timeline of our expectations and an associated Gantt Chart are included in Appendix E-F.

Conclusion:

The project is still in a prototypical state. We have a working prototype. Right now the device is capable of watering up to two plants individually based on simple threshold water profiles. The device is programmable, with an app, though the app is in its early stages. Additionally, though it is its own discrete system, it is monitoring water levels, and presenting them to the user. We spent most of the semester demonstrating our control over the individual components, with a simple prototype composing most of the subsystems coming together closer to the end. Right now the Water Supply Monitor is seperate from the pumps. In its current state, the device does not rely on any sophisticated feedback for managing plants; it just uses a simple threshold. For MDR, we met our MDR requirements, which is essentially the prototype described earlier.

One of the most important remaining tasks is to derive more scientifically rigorous performance specifications. Our moisture level sensors are not accurate enough to be considered ground truth. Further experiments are required to align our measurement data with actual moisture content levels, and to develop more satisfying water profile models accordingly.

For CDR, we need to be meeting our specs which includes watering our plants to within +/- 10% of their desired moisture content at a given time. This may be quite difficult because the sensor we are using to check the desired moisture content, does not return a value associated with any observable metric we can use directly to measure our performance. We will have to spend plenty of time thinking about how to tackle this issue over break.

Acknowledgments:

We'd like to thank Cristopher Phillips, of the UMass College of Natural Sciences, for welcoming us at the Morrill Greenhouse and providing us with valuable horticultural insight and expertise.

Thanks to Professor Kris Hollot, for managing the Senior Design program and helping us approach our feedback control problem with his professional understanding.

Thanks to Professors Dennis Goeckel and Beatriz Lorenzo, for reviewing our Preliminary Design and Midterm Designs with patience, fairness, and constructive feedback.

Lastly, but most especially, thanks to Professor Do-Hoon Kwon for serving as our team's faculty advisor throughout the semester.

References:

1. *[1] Farmtek, "Rapitest Moisture Meter: Watering Guide and Table" . 2010, pp. 2. https://www.farmtek.com/wcsstore/EngineeringServices/allbizunits/techdocs/103399.pdf*

Appendix:

A. Design Alternatives

Settling on this idea for our Senior Design Project required several weeks of discussion. We considered many ideas, including a floor navigation system for wheeled robots, a drone-based mosquito exterminator, and an electronically amplified hurdy-gurdy. However, many of our ideas were extravagant, and dependent on unrealized theoretical insights. We decided on Water Wizard because it had obvious consumer appeal, and it seemed straightforward enough that we could accomplish the task with the skills we possessed already.

In the early stages of the project, we imagined that Water Wizard should be completely wireless. It would manage all of the plants in a house with individual reservoirs, and without cumbersome hoses and wires. Our wireless design could be applied to outdoor fixtures, such as the hanging plants around the UMass campus. However, we decided that the wireless implementation would introduce unnecessary complications, and that a basic proof of concept was required first. The fully wireless Water Wizard may be realized at a later stage of development, beyond our SDP.

Our original conception of Water Wizard included stricter water profiles which specified the moisture levels and watering frequency, and also the rate of change in the moisture level. We planned to implement this using PID, (Proportion Integral Derivative,) feedback control. As the project progressed and our ideas became more refined, we realized that this was too complicated. It implied that we would control the rate at which water evaporates from the pot, which is difficult, and beyond the scope of this project. Thus, we decided on a simpler watering model with proportional feedback only.

We also encountered some complications while selecting mechanical components. Ideally, Water Wizard features a single water pump with valves for flow control. At first, we found that using valves was bulkier and more expensive than simply adding additional pumps. However, we have recently found a source of cheap, efficient valves, which will enable us to complete the project as planned.

B. Testing Methods

Our first goal was to establish a functional bluetooth connection with the controller. To test this, we transmitted simple char data packets to the controller, and programmed the controller to retain and resend this data repeatedly until we sent something new. After some development, we were able to transmit more complex strings. With a successful bluetooth connection, we based all of our further experiments on interactions using a mobile phone terminal.

Our MDR prototype was intended to respond to a specific moisture threshold by activating mechanical systems. To test this, it was sufficient to immerse our sensor in a cup of water for a 'wet' reading, and withdraw it for a 'dry' reading. At first, we used an LED in place of a water pump. The LED would activate whenever the sensor was dry, and de-activate when it was wet. This was a simulation of the most basic possible watering scenario. Eventually, we incorporated actual pumps, and developed a program which could manage three different sensors simultaneously.

As a form of rudimentary performance evaluation, we designed a game based on the wet/dry threshold. As with an actual watering profile, the controller would expect the sensor to read 'dry' for a fixed amount of time, then read 'wet' for a fixed amount of time. If the sensor was dry when it was expecting to be wet, it would inform us that the sensor was too dry, and vice-versa. It was our task to simulate the action of the watering system, and produce the expected reading at the expected time. If we took too long, or neglected to do this, the controller would inform us later that we'd failed to maintain the watering profile.

C. Team Organization

Each member of the team has found a place in this project according to their individual interests and experience.

Aneurin Sutton, an EE major, has devoted his efforts to the mechanical systems, particularly the water level sensor.

Zach Toscanini, also an EE major, has focused on overall project management, specifications, and performance evaluation.

Parth Nanavati, a CSE major, has focused on developing the mobile user interface.

Andrew Grote, also a CSE major, has developed the communications, the threshold program, and the integration of mechanical components with software.

Our team meets regularly on Mondays and Fridays, to coordinate efforts and discuss progress and ideas. We communicate all significant project developments with group text messages.

D. Beyond the Classroom

Andrew Grote - *"From a technical perspective, I have become much more comfortable with programming on low power devices. I have plenty of experience programming, but it is a totally different ball park, programming on small embedded devices. You simply have way less resources directly at your disposal than when programming on a complete computer system. Though I failed at it during this semester, I have learned the importance of readability in code. Readability is crucial, for effective teamwork.*

More important than the technical skills I have earned, is my growing understanding of group dynamics. Though I do not completely understand, I am learning the techniques that will enable me to work effectively on a team."

Zach Toscanini - *"What I like best about Water Wizard is that it seamlessly integrates technology with nature. It is concerningly rare to find inventions without adverse environmental impacts.*

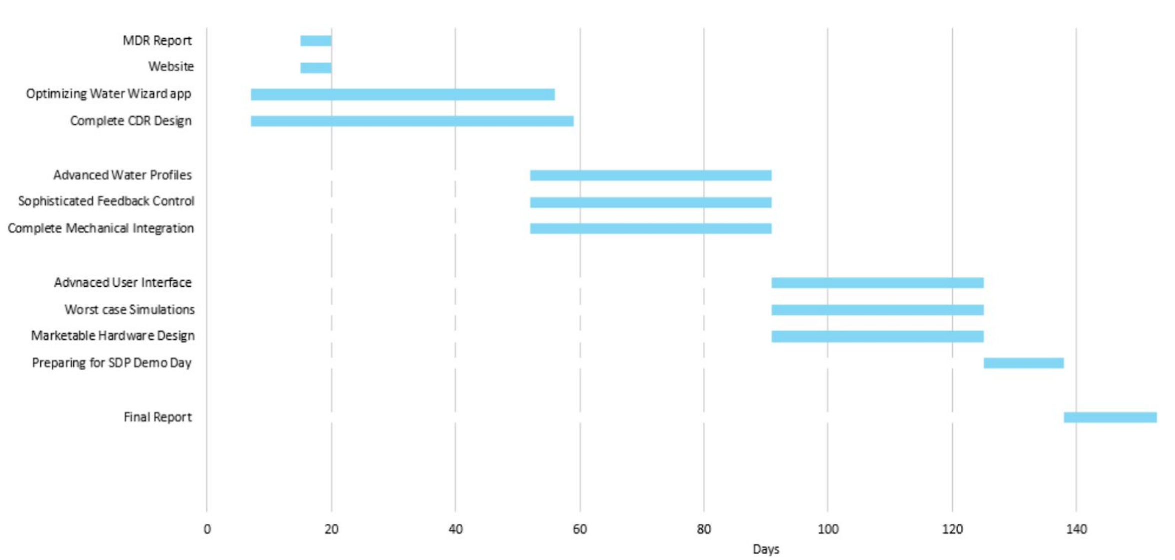
The Bicycle is my idea of the highest standard of what engineers can achieve. It is an invention which harnesses our understanding of natural laws to make our lives more efficient, but not at the expense of the fragile natural ecosystems which make our lives possible. Water Wizard is an invention which serves us by serving plants, and it does so harmlessly.

In this respect, this project has been a departure from the schoolwork I've been accustomed to. Instead of abstract virtual results, we're trying to meet specifications with rigorous natural criteria. In my discussions with the various professors involved in SDP, I've learned that aligning these ideal aspirations with actual realities is more complicated than it first appears. Developing more satisfying virtual models for complex realities, which facilitate our understanding and make engineering possible, is the ongoing work of science."

Appendix E: Anticipated development timeline

TASK NAME	START DATE	DAY OF MONTH*	END DATE	DURATION* (WORK DAYS)	DAYS COMPLETE*	DAYS REMAINING*	TEAM MEMBER
Winter Break							
MDR Report	12/15	15	12/19	5	0	5	All
Website	12/15	15	12/19	5	0	5	All
Optimizing Water Wizard app	12/7	7	1/24	49	0	49	Parth
Complete CDR Design	12/7	7	1/27	52	0	52	All
Complete Design Review							
Advanced Water Profiles	1/27	52	3/5	39	0	39	Parth
Sophisticated Feedback Control	1/28	52	3/6	39	0	39	Andrew/Zach
Complete Mechanical Integration	1/29	52	3/7	39	0	39	Ni
Professionalizing & Debugging							
Advnaced User Interface	3/10	91	4/12	34	0	34	Parth
Worst case Simulations	3/11	91	4/13	34	0	34	Andrew
Marketable Hardware Design	3/12	91	4/14	34	0	34	Zach/Ni
Preparing for SDP Demo Day	4/12	125	4/24	13	0	13	All
Post FDR							
Final Report	4/25	138	5/9	15	0	15	All

Appendix F: Gantt Chart for further progress, beginning on 12/12



Appendix G: Raw Data on water profiles[1]

PLANT	MOISTURE	FREQUENCY	SPECIAL NEEDS	PLANT	MOISTURE	FREQUENCY	SPECIAL NEEDS
African Violet	3	**		Heliotrope	7	**	
Aloe	1	*		Hen-and-Chicken	1	*	
Amaryllis	3	**		Hibiscus	4	*	
Anthurium	8	***	I	Hyacinth	8	***	III, VI
Aralia	4	**		Impatiens Wallerania	7	**	VI
Arrowhead Vine	4	**		Inch Plant	1	**	VI
Asparagus Fern	6	**	III	Jacaranda	6	**	I
Azalea	8	***	II	Jade Plant	1	*	VI
Bamboo Plant	4	**		Jacobinia	7	***	
Baby's Tears	6	**		Japanese Sedge	7	***	III
Bay Tree	6	**	I	Jasmine	7	***	
Begonia Rex	6	**	VI	Jasmine Plant	8	***	I
Begonia	7	***	III	Jelly Bean Plant	1	*	
Bird of Paradise	3	*		Jerusalem Cherry	3	**	
Billbergia	6	***	I	Jessamine	7	***	
Black-eyed Susan	4	**	III	Kalanchoe	1	*	VI
Bloodleaf	7	***	III	Ka r Lily	1	*	VI
Blue African Lily	7	**		Kangaroo Thorn	6	**	
Boston Fern	6	**		Kentia Palm	6	**	
Bottlebrush Plant	6	**		Lady Palm	8	**	

Plant Care Specifications from Farmtek,
on data collected by expert Agricultural Engineers
from Farmtek[1].