HW #3 Requirements Specification, Analysis, Prototyping and Review

Com S/SE 409 & Com S 509, Fall, 2020

Due by 8 p.m. Thursday, Oct. 15; turn in on Gradescope as a pdf.

*NEW: **509** students individually turn in answer to last problem as a separate assignment on Gradescope*

Textbook reading assignment: Chaps. 12, 13 and 16

Team assignment: one assignment is turned in for this part with the names of all the team members who participated on it. **Remember that all of you must work together on all the problems**. If you prefer, you may instead do the entire assignment individually. Re-read the Homework Policy at the top of HW#1.

1. *Fit Criteria and Rationale.* Read Chaps. 12 and 13. Create a new document. This will evolve into the Software Requirements Specification (SRS) that you deliver for the final project. It will be more detailed, complete, and accurate than the specifications that you created for HW#2 and will be **updated** in response to feedback from HW#2 and from your increased understanding of what the software product needs to do.

   (a) For each correct requirement in HW#2, and each new/revised requirement that you add before turning in HW#3: (30 pts.)
   Add a **fit criterion or criteria for it** , as appropriate, and record these fit criteria (each of which is a detailed requirement) in the new document. **This is when you want to "get the requirements right", meaning that if you handed it to a developer not on your team, you will be very likely to get back the software product that is needed.**

   Functional Requirements:

   - The system shall operate in one of the two modes: manual and automatic.
   - The system shall provide a user with a way of switching modes via the control panel.
   - The system shall display its current status - operating mode, scheduled watering time, malfunctioning valves or sensors - and related environmental parameters - moisture level, water consumption level.
   - If the user enters a value greater than 100 for moisture level, the system shall reject the value and display an error message.
   - When the water consumption reaches the maximum water level, the system shall close all the valves.
   - The system shall support division of landscape into 1 to 32 zones.
   - The system shall support only one moisture sensor per watering zone.
   - The system shall support 1 to 32 valves per watering zone.
   - The system shall display sensor data for each watering zone.
   - The system shall support simultaneous watering of each zone.
   - When the user switches modes, then the system will operate according to the values used previously for that mode.
   - When the user wakes up the control panel, then the system will display its current status - operating mode, scheduled watering time, malfunctioning valves or sensors; and related environmental parameters - moisture level, water consumption level.
   - While in manual mode, the system shall accept and record the user's input about the status of the valves.

- While in automatic mode, the system shall accept and record the set user parameter for the ideal moisture level.
- While in automatic mode, the system shall accept and record the user parameter for maximum water usage.
- If the user enters a negative value or non-numerical value for desired moisture level or maximum water usage, the system shall reject the values and display an error message.
- While in automatic mode, the system shall accept and record user time settings for watering intervals.
- The system shall provide the user with watering time values to choose from (e. g. Monday - Sunday, 12:00 am - 11:59 pm).
- While in automatic mode, the system shall accept and record parameters for individual watering zones.
- While in automatic mode, when the user provides only a total maximum water usage value, then the system shall compute maximum water usage for each zone individually.
- While in automatic mode, when the user provides new parameters, then the system shall recompute the watering parameters.
- When in automatic mode, when the current time reaches the watering time, the system shall start watering the landscape.
- While in manual mode, if one of the sensors/valves becomes inoperable, then the system shall inform the user by displaying a warning message.
- If one of the sensors/valves becomes inoperable, then the system shall close all valves in the corresponding watering zone and treat the zone as if it had reached the critical moisture level.
- While in automatic mode, the system shall not let the user control the valves.
- While in automatic mode, when the desired moisture level for a zone is reached, the system shall close all valves in the zone.

(b) Identify (with a "*") the 5 fit criteria for your nonfunctional requirements where it is least self-evident why it is needed in this product and specify a clear, accurate **rationale** for each of those 5. (If a 3rd party developer/maintainer could understand why the requirement is there without further explanation, we consider it to be self-evident.) (15 pts.)

Non-Functional Requirements:

- The system shall record entered data fast.
    - Rationale: Users become annoyed when the system takes a long time to respond, and may forget what they wanted to do with the system next.
    - Fit criterion: The response time shall be no greater than a second 90% of the time, and no greater than 2 seconds for the remainder.
- The system shall be accurate.
    - Fit criterion: The system shall maintain desired parameters with an error margin no greater than 2%.
- The product shall support an easy addition of new sensors and valves.
    - Fit criterion: The user can register new equipment with the system in less than 5 minutes without any errors.
- The product shall support an easy addition of new watering zones.
    - Fit criterion: The user can modify the watering zones in the system in less than 5 minutes without any errors.
- The product shall retain past watering schedules and parameters for a month.
    - Rationale: The user may want to switch to past settings quickly.
    - Fit criterion: Past schedules are accessible for only the last 30 days, and older data is deleted.
- The product shall run on different control panels with different operating systems.
    - Fit criterion: The system shall run on control panels with the Linux operating system starting with version 17 and later.
    - Fit criterion: The system shall run on control panels with the Chrome operating system starting with version 75 and later.
    - Fit criterion: The system shall run on control panels with the Raspberry Pi operating system starting with version 2019-06-20 and later.
- The product shall be intuitive.
    - Rationale: New users must find the system easy-to-use, otherwise they will not use it.
    - Fit criterion: New users shall be able to create new watering schedules in the automatic mode or control valves in the manual mode within 5 minutes of their first encounter with the product without any outside help.
- The product's user interface shall make use of graphics to make it more appealing to the user.
    - Rationale: The users must like the system to start using it instead of the competitors' systems.
    - Fit criterion: 60% of the users will recognize the system as more appealing than the competitors' systems.
- The product shall be accessible to users with disabilities.
    - Rationale: The system shall be used by wider demographic groups.
    - Fit criterion: 95% of users with disabilities will be able to interact with the system after initial training.
- The product shall prevent access to the user's confidential information by unauthorized parties.

  - Fit criterion: The system shall only display data to users who provide a valid passcode.
- The product shall correctly compute maximum water usage for each zone.
  - Fit criteria: The system shall compute the maximum water usage by dividing the maximum water usage value by the number of valves in the system and then multiplying the result by the number of valves in each zone.

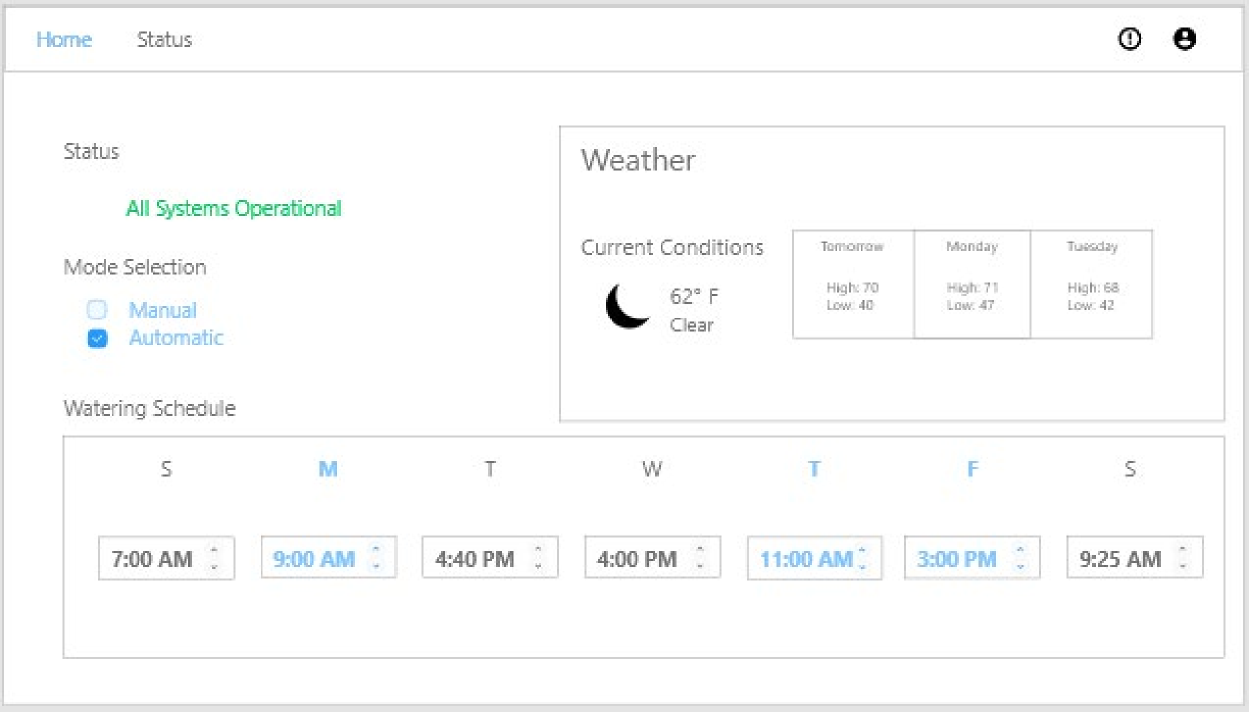
2. *Rapid Prototype of an EverGreen Control Panel.* The client has requested that you develop a prototype of the control panel to be used by the homeowner to set, monitor and oversee the landscape watering process. The control panel displays real-time data from the sensors regarding the state of the system (software as well as hardware) and allows the operator to set up and adjust the settings. (Note that most real-world prototyping involves creating more than one alternative prototype; we are doing just one here.)

   (a) Description (10 pts.) Describe the prototype for the control panel that you're proposing. This description needs to have enough details that a reader/user can understand what it does and to convince us that your control panel prototype has merit. Describe, as well, what prototype technique &/or software tool you're using to create the prototype, as well as the level of prototype fidelity that you've chosen and why.

   We have created two screens for the EverGreen control panel prototype: A home page and a status page. This prototype was created using Adobe Xd, meaning it is to be considered a high-fidelity concept of the EverGreen UI. At the top of each screen in the prototype, we see a navigation bar that is consistent throughout the control panel. The only two pages we have included are "Home" and "Status," but these are subject to some additions. On the right side of the navigation bar, there are icons meant to resemble buttons which will link to a notifications page and user profile page respectively. The homepage shown by figure 1 consists of a quick reference for the system status, as well as a weather widget that displays the weather information for the next three days, as well as a way for the user to toggle manual or automatic mode for the watering system. We have also included a section that enables the user to set the designated watering schedule when in automatic mode. For the sake of simplicity, we show that automatic mode is enabled by default. Note that weather forecasts are not currently a part of EverGreen, but it is something we think will be of use to the clients. If we take a look at figure 2, we see some mock data representing the states of the various valves and sensors that are linked to our system. Here, the user is able to read sensor values, toggle which sensors and valves to activate during manual mode, water usage, moisture level, and the status for each individual valve and sensor.

   (b) Control Panel Prototype (20 pts.) Very briefly describe two scenarios that your prototype supports. (This can be brief text; you don't need to use the scenario template). One of the scenarios must involve something changing in the control panel display as a result of the EverGreen software receiving input data from a sensor(s) indicating some failure. The other scenario must show some interaction of the homeowner with the control panel settings.

Submit a single pdf with the two labeled screenshots of the control panel prototype for EverGreen, one for each scenario.



Status

All Systems Operational

Mode Selection

□ Manual
☑ Automatic

Watering Schedule

S	M	T	W	T	F	S
7:00 AM	9:00 AM	4:40 PM	4:00 PM	11:00 AM	3:00 PM	9:25 AM

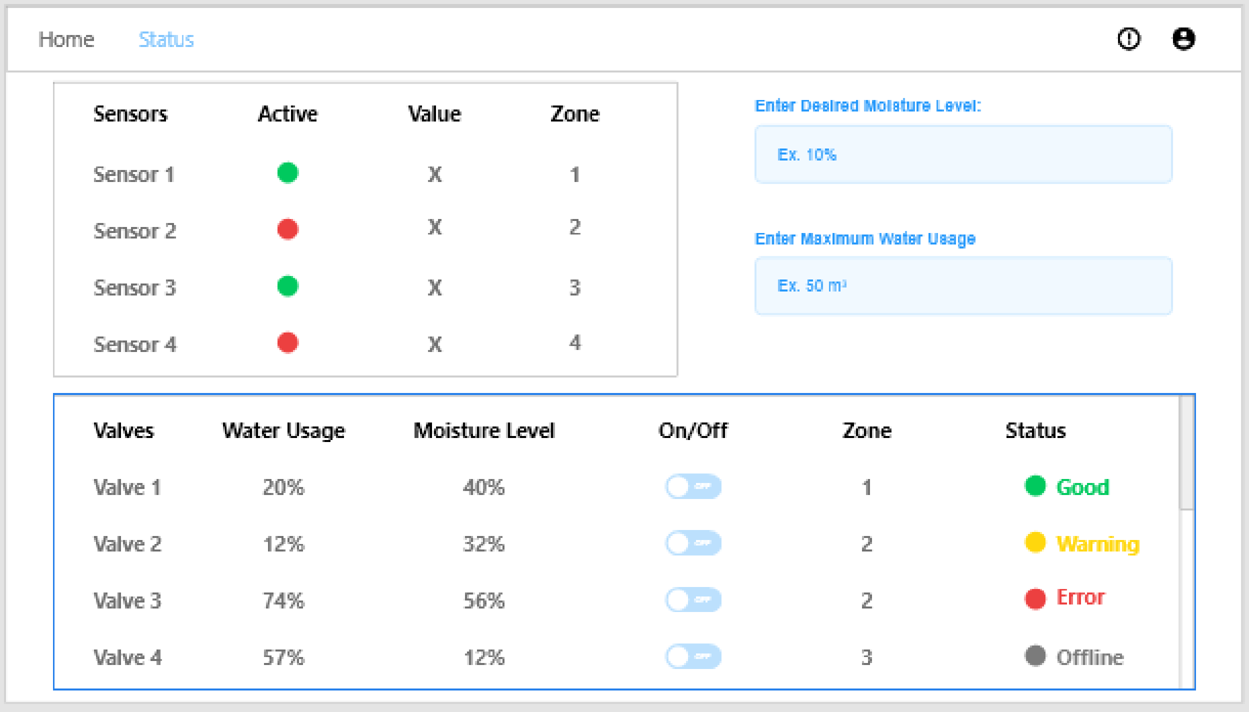
Weather

Current Conditions

62° F
Clear

Tomorrow	Monday	Tuesday
High: 70 Low: 40	High: 71 Low: 47	High: 68 Low: 42

Figure 1: The homepage of the EverGreen control panel



Sensors	Active	Value	Zone
Sensor 1	●	X	1
Sensor 2	●	X	2
Sensor 3	●	X	3
Sensor 4	●	X	4

Enter Desired Moisture Level:

Ex. 10%

Enter Maximum Water Usage

Ex. 50 m³

Valves	Water Usage	Moisture Level	On/Off	Zone	Status
Valve 1	20%	40%	OFF	1	● Good
Valve 2	12%	32%	OFF	2	● Warning
Valve 3	74%	56%	OFF	2	● Error
Valve 4	57%	12%	OFF	3	● Offline

Figure 2: The status page of the EverGreen control panel

3. Using the prototype. (15 pts.)

(a) Design 4 questions that are likely to elicit useful feedback on the prototype of the EverGreen control panel from a representative user of the system.

- Using the prototype, are all the features you were hoping to see implemented in the prototype?
- Viewing the prototype, what more features (if any) would you like to see in the final product?
- Would you make any changes to the already existing functions implemented in the prototype and please explain each change you would make.
- Was the prototype a good implementation of the requirements you were looking for?

(b) Each member of your team should individually show the prototype to, and get feedback on it from, a "representative" user of the prototype: show them the prototype, ask the 4 questions and write down their answers. Turn in the answers, mapped to the name of the team member who tested the prototype and the name of their representative user.

1. Representative User: Meghan Shouse
    - Team member: Aashray Mehta
    - Answer:
        - Yes, all the features in the prototype are adequate.
        - SMS feature to alert the user of errors.
        - I wouldn't change anything about the existing product.
        - It covers all requirements.
2. Representative User: Kyle K
    - Team member: Rithvik Menon
    - Answer:
        - Yes. All features shown in the prototype seemed useful and needed for a watering system.
        - A notification feature that would notify the owner if and when a valve or sensor status changes from good to warning or error would be a quality of life feature that would be great if implemented.
        - The features currently available in the prototype are well implemented. No changes need to be made for the existing features.
        - The prototype covers all the basic and necessary requirements.

3. Representative User: Patrick S
    - Team member: Maksym Nakonechnyy
    - Answer:
        - Yes, the current prototype shows all environmental readings requested. It shows the status of each valve and sensor. It allows one to switch between modes and prevents from controlling valves in automatic mode. It also lets set the desired schedule and moisture levels.
        - The weather forecasting feature seems useful. It helps make watering decisions and change the schedule based on the information. It would be a great addition to the system.
        - It would be nice to allow assigning valves different names, not just numbers.

- This prototype is a good draft of the system. It looks appealing and includes all components required.

4. Representative User: Tyler T
    - Team member: Stamati Morellas
    - Answer:
        - Yes, they are formatted in a logical and easy-to-follow order
        - Show a unit measurement of the water used instead of a percentage
        - No, everything looks as it should.
        - Yes, I think it gets the job done and covers all the requirements.

(c) What changes to your EverGreen requirements will you consider as a result of the feedback you received?

- Show a unit of measurement for water usage in valve status instead of a percentage so that it is more clear to the user.
- Add SMS notifications about failures in the system.
- Add a weather forecasting feature to the system
- Add valve aliasing.

4. Anticipating change. (10 pts.)
Read the 1-page article, "When bits bite," posted with the homework assignment. Schumpeter says that upgrading a software system is too often "like rebuilding an aircraft in flight." How does including a requirement's *rationale* in its description help reduce bugs later on in testing and maintenance?

Having a requirements rationale helps developers to understand why the requirement is needed, and why the system needs to do it. With a better understanding of the system,developers can collaborate better with one another. It also helps new developers who come to work on an existing project to understand it better and thus make fewer mistakes. Having rationales also helps to make connections with other requirements, which helps eliminate unnecessary requirements and decreases the ambiguity of each one. Having a rationale also exposes bad assumptions early in the process and helps derive the correct fit criteria. Since software is tested against the fit criteria, developers know in advance how the software will be tested and naturally write code that will pass the tests. This drastically reduces the number of failing tests and thus bugs.

Q & A

Com S/SE 409 & Com S 509, Fall, 2020

from Homework 1, problem 3

for use in Homework 2 & project assignments

Good job trawling for requirements! Overall, the Homework #1 papers showed that insightful & useful consideration was given to the wide variety of stakeholders that may affect the software requirements for the EverGreen system.

Our client interface (TA Wandi Xiong) collected the questions from problem 3 on Homework 1, and provided the answers below from our client.

- **What exact parameters do we need to have available for the user?**

  Watering time, critical moisture level, maximum water usage

- **Does the user choose the zones? How should zones be computed if not set by users?**

  How do we compute the max water consumption? Can the user set it?

  The user can set the maximum value in total and others will be computed by the system automatically.

- **Can you override automatic mode mid-watering?**

  Yes. The system will recompute the watering parameters.

- **"Users can enter time settings to schedule the watering" Does the user have to do this even in automatic mode?**

  Yes. The watering time setting.

- **How do we determine the number of sensors and valves for a given yard?**

  Are zones determined by the user or computer?

  How many sensors do we have?

  How are zones established?

  The landscape site must be divided into 1 to 32 watering zones.

  Each watering zone must have exactly one moisture sensor.

  Each watering zone must have 1 to 32 watering valves

- **Do the sensors constantly update or does the control panel request that info on an as needed basis?**

  Constantly update

- **When in manual mode can the user see each zone individually?**

  Yes.

- **Does it water each zone consecutively or rotate between zones?**

  Zones water simultaneously

- **What controls the valves and calculations? The Control Panel? A separate computer? Does the user set the maximum water, or is that entirely determined by the computer? If the user sets the maximum water, can they make it waste lots of water?**

  The software controls the computation. The user set the maximum water. They can.

- **Does the system notify the users if the sensors or valves are broken?**

  How will the program continue with any sensor faults?

  If a sensor can't be read after a few tries, then EverGreen should stop using it in watering.
  If a sensor can't be read after a few tries in manual-mode operation, then EverGreen let the user know that the sensor has failed.
  If a sensor can't be read after a few tries when EverGreen is watering that sensor's zone, then EverGreen has to close all valves in that zone and treat that watering cycle as if that sensor had showed that the zone had reached its critical moisture level.

- **What features differentiate the manual and automatic modes?**

  In manual mode, EverGreen must allow users to select sets of valves and command that they be opened or closed.
  In auto mode operation, EverGreen must begin a watering cycle when the current time reaches the watering time

- **Can the user specify between manual and automatic for each zone independently?**

  No

  Should manually turning the valves on/off override the auto mode, or should auto mode stay on?

  In the auto mode, valves can't be turned on/off manually.

- **How is the maximum water usage for each zone computed?**

  The maximum water usage divided by number of valves, then multiply the number of valves in the zone.

- **Are certain features more critical than others?**

  Yes; see info above.

- **What does the current equipment do?**
  How does the current equipment send and receive information?

  There isn't a current system in place.

- **Is this intended to be a final, static product, or a modular one to be built upon later?**

  Client envisions a future EverGreen product line.

- **What customizations options are needed other than the manual/automatic option?**

  You can add customization options and features as long as they don't restrict the capabilities of EverGreen that the clients wants. (For example, you can't delete the automatic mode.)

Context/Environmental/Hardware/Societal:
Good questions & suggestions here! No binding answers from client, so you can make—and document—any reasonable assumptions you need, and add options/features as you see fit.

- How is damaged equipment handled within the project constraints?
- What form should data be sent/received?
- What size is the landscape in order to figure out if splitting watering into sections is viable?
- What types of plants are being watered to determine moisture thresholds?
- What is the quality of water in the locale and how would that affect the plants the users want to water?
- Are there any local laws against watering plants on certain days?
- What sort of sensors measure the size of zones and total area?
- Will the control panel connect to the home network?
- If the control panel does connect to the internet, should we stream analytics?
- Will software updates be done through the internet or a possible USB port?
- Would using a standard android tablet as the control panel be the best solution?
- How can we take village water consumption rules into consideration?
- When do you expect the project to be completed?
- What is our budget?
- Should users be notified when they make ineffective choices in manual mode, e.g., setting moisture level to low in dry weather?
- Is there an alert to change maximum water usage when water is running low?