

---

# Software Requirements Specification

for

*Remote Environment Monitoring System  
(REMS)*

Version 1.2 approved

Prepared by Paul Walter

August 9th 2010

# Table of Contents

<b>1. Introduction.....</b>	<b>1</b>
1.1 Purpose.....	1
1.2 Description.....	1
1.3 Document Conventions.....	2
1.4 Intended Audience and Reading Suggestions.....	2
1.5 Project Scope.....	2
1.6 References.....	2
<b>2. Overall Description.....</b>	<b>3</b>
2.1 Product Perspective.....	3
2.2 Product Features.....	3
2.3 User Personas.....	3
2.4 User Classes and Characteristics.....	3
2.5 Operating Environment.....	3
2.6 Design and Implementation Constraints.....	3
2.7 User Documentation.....	3
2.8 Assumptions and Dependencies.....	3
<b>3. System Features.....</b>	<b>4</b>
3.1.1 Technician's Use Cases.....	5
3.1.2 Steward's Use Cases.....	28
3.1.3 System's Use Cases.....	35
<b>4. External Interface Requirements.....</b>	<b>36</b>
4.1 User Interfaces.....	36
4.2 Hardware Interfaces.....	42
4.3 Software Interfaces.....	42
4.4 Communications Interfaces.....	42
<b>5. Other Nonfunctional Requirements.....</b>	<b>42</b>
5.1 Performance Requirements.....	42
5.2 Safety Requirements.....	43
5.3 Security Requirements.....	43
5.4 Software Quality Attributes.....	43
<b>6. Other Requirements.....</b>	<b>44</b>
6.1 Appendix A: Data Dictionary.....	44
6.2 Appendix B: Entity-Relationship Diagram.....	46
6.3 Appendix C: Sequence Diagrams.....	47
6.3.1 The Custom Data Layer.....	49
6.4 Appendix E: Issues List.....	53
6.4.1 Registering new devices into the system.....	53
6.4.2 Scalability.....	53
6.4.3 Cross Platform Capabilities for the Client GUI.....	53
6.4.4 Separate Technician GUI for the Sensor Client.....	53
6.4.5 Authentication.....	53
6.4.6 Java Media Framework (JMF) is Dead.....	54
6.4.7 “1 Wire” Framework is Dead.....	54
6.4.8 This project needs to be Geospatial.....	54

## Revision History

Name	Date	Reason For Changes	Version
Paul Walter	June 1 <sup>st</sup> , 2008	Initial document. Solidified use cases.	1
Paul Walter	August 20 <sup>th</sup> , 2008	Scope change: the client needed more than just a command line prototype, they needed a Graphical User Interface to accompany it so the semantics of the design could be validated in a proper context.	1.1
Paul Walter	August 4 <sup>th</sup> , 2009	Changed because the Java Swing Graphical User Interface, based on low-fidelity usability testing, wasn't semantically rich enough. Needed to go through some design iterations before a proper design was accepted.	1.2

# **1. Introduction**

## **1.1 Purpose**

The purpose of REMS is twofold: provide ubiquitous visibility of the data and images of indoor organisms (regardless of where its observers are physically located), and provide a decision support tool that can help the observer make good decisions about managing their remote environment by providing trends in the data that ultimately would be invisible if the system wasn't there.

This is a decision support tool for individuals that are involved with the care and maintenance of indoor organisms. Cultivating these indoor organisms can be a complex and arcane, especially since these organisms 1.) are outside the self-regulating cycle of life provided by nature and 2.) live in small environments, and 3.) rarely give an indication that there is an issue. Subtle changes in their environment can go unchecked and quickly cause organisms to die.

The symptoms of these catastrophic changes can be difficult to observe, especially slowly forming trends, and often go unnoticed until the window of opportunity to make lifesaving corrections has closed. Examples of such variables in tropical aquariums are pH, temperature, and salinity. Another example is the amount of sun that potted plants receives, or the pH of its soil.

Variables such as pH, and solar energy (lux) can be actively measured in order to ensure optimal growing conditions. The need to monitor these environments is even more critical because of the inability of the most indoor organisms to report any degradation in their environment. Unlike cats, dogs, or birds that can indicate when they are experiencing such degradation, fish, plants and reptiles are generally not able to communicate that they are in danger. These organisms will quietly expire if their environments are not properly maintained.

The ability to monitor contained indoor environments is a necessary, but not sufficient, tool to ensure the survival of the organisms living therein. Environmental changes can occur at any time, and so these environments must be monitored constantly. Accordingly, this tool must have the ability not only to sense these changes, but also to notify the end user, who may be in close proximity to the environment, or thousands of miles away. Such ubiquitous notification capability is the cornerstone of this project.

Even though difficulties like those outlined above are common in maintaining organisms in indoor environments, there are currently few tools that can 1) simultaneously give a voice to these organisms, and 2) aid in the decision making process of the people who maintain their habitat, and 3) do so at a reasonable cost.

## **1.2 Description**

### **1.3 Document Conventions**

Words that have a formalized meaning will be *italicized*.

### **1.4 Intended Audience and Reading Suggestions**

The intended audience of the main part of this document are Clients, Project Managers, and Marketing staff. This document will be written in plain language (no technical terms) for the benefit of everyone on the team. It will define the business rules of this application.

The Appendix will contain information for the Developers (UML and DFD blue prints for the software design), and may contain technical jargon.

### **1.5 Project Scope**

### **1.6 References**

## **2. Overall Description**

### **2.1 Product Perspective**

This is a new self-contained product.

### **2.2 Product Features**

### **2.3 User Personas**

### **2.4 User Classes and Characteristics**

### **2.5 Operating Environment**

### **2.6 Design and Implementation Constraints**

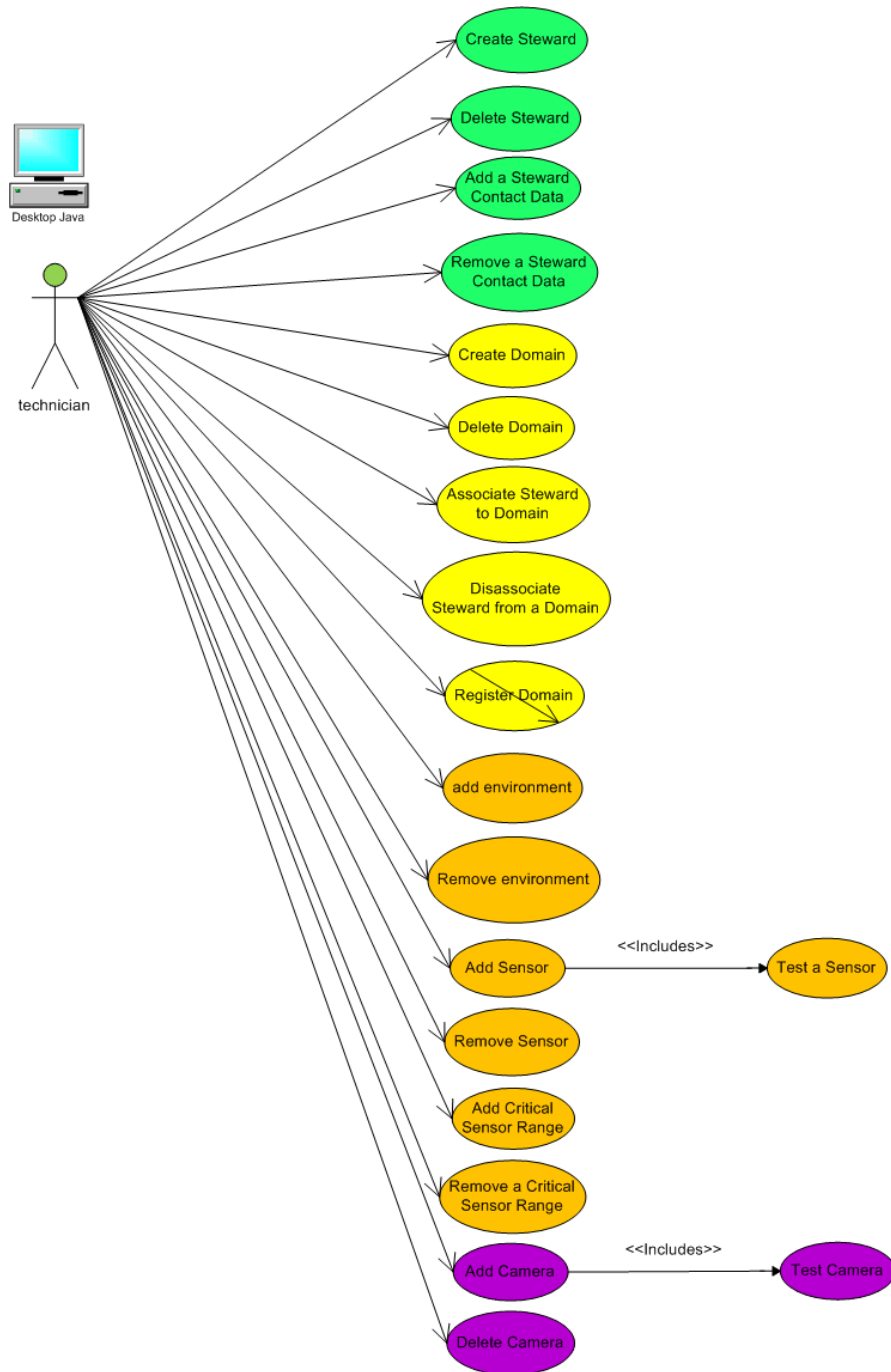
### **2.7 User Documentation**

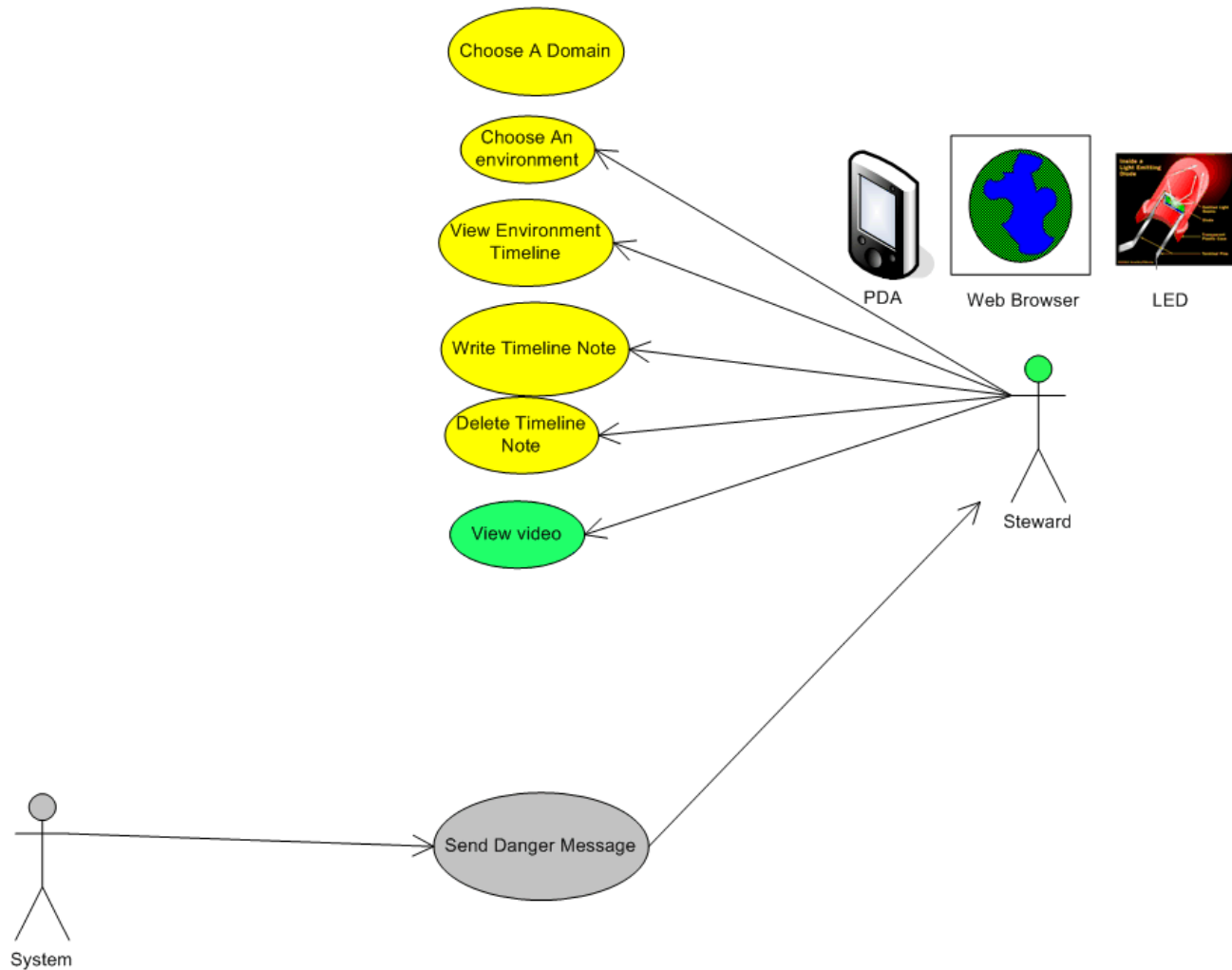
### **2.8 Assumptions and Dependencies**

### 3. System Features

These use cases are arranged according to user class (see section above with the same title). Here is a use case diagram to give a visual representation.

Note that the use cases are color coded to identify sets of functionality.





### 3.1.1 Technician's Use Cases

--	--

Use Case ID: 1

Use Case Name: Create Steward

Created By Paul Walter

Last Updated By:

Paul Walter

Date Created 07/01/2008

Date Last Updated:

07/01/2008

Actor: Technician

Description: Technician adds a steward to the system



Preconditions:           That the database is up and running

Post conditions:        That there will be a steward to associate with any number of Domains on the system

Priority:                low

Frequency of Use:      low

Normal Course of  
Events:

1. Technician desires to add a Steward to the catalog of stewards
2. System asks for the Stewards title
3. Technician inputs Stewards user name *[user\_username]*
4. System asks for the Stewards password
5. Technician inputs password *[user\_password]* (alpha-numeric)
6. System creates a unique identifier *[user\_id]* and set their status to active *[user\_status]* (i.e. active or inactive) inserts the completed steward *[user\_username , user\_password, user\_id, user\_status]* into the database
7. System returns feedback of a successful steward creation

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID:	2		
Use Case Name:	Delete Steward		
Created By	Paul Walter	Last Updated By:	Paul Walter
Date Created	07/01/2008	Date Last Updated:	07/01/2008
Actor:	Technician		
Description:	Technician removes the Steward <i>[user_username, user_password, user_id, user_status]</i> and associated Contact Data <i>[user_id, contact_data_id, contact_data_name, contact_data_address, contact_data_description]</i> from the catalog of available Stewards.		
Preconditions:	That the database is up and running, and there are Stewards to remove from the steward catalog.		
Post conditions:	That the chosen Steward will no longer exist in the Steward catalog, and that some Domain's will no longer have a registered Steward associated with them		
Priority:	low		
Frequency of Use:	low		
Normal Course of Events:	<div>1. Technician desires to remove a Steward from the catalog of available Stewards</div> <div>2. System presents a list of available stewards.<i>[user_id, steward_username]</i> and the Domains that the are associated with <i>[domain_name]</i></div> <div>3. Technician selects the Steward they wish to remove <i>[user_id]</i> System informs the Technician that some domains may be without a registered Steward, and if they still wish to proceed.</div> <div>4. Technician confirms that they still wish to proceed.</div> <div>5. System removes the Steward from the Domains they are currently associated with.</div> <div>6. System removes the Contact info <i>[user_id, contact_data_id, contact_data_name, contact_data_address, contact_data_description]</i> that is associated with that Steward</div> <div>7. System removes the Steward from the catalog of available stewards. <i>[user_username, user_password, user_id, user_status]</i></div> <div>8. System informs Technician that the Steward has successfully been removed from the catalog.</div>		
Alternative Courses:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

--	--

Use Case ID:	3		
Use Case Name:	Add a Steward Contact Data		
Created By	Paul Walter	Last Updated By:	Paul Walter
Date Created	07/01/2008	Date Last Updated:	07/01/2008
Actor:	Technician		
Description:	Technician adds an email address for email or for text messaging phones		
Preconditions:	That the database is up and running, and there are Stewards to add contact data too		
Post conditions:	The chosen Steward will have a email address that can be contacted if the need arises.		
Priority:	low		
Frequency of Use:	low		
Normal Course of Events:	<ol style="list-style-type: none"> <li>1. Technician desires to add a contact address for a Steward</li> <li>2. System presents a list of available Stewards</li> <li>3. Technician chooses a Steward to add a contact address too.</li> <li>4. System asks what the address will be</li> <li>5. Technician inputs contact address <i>[contact_data_address]</i></li> <li>6. System asks what the type of the contact_address will be. <i>[contact_data_type]</i></li> <li>7. Technician inputs the contact_type (i.e. email, or phone)</li> <li>8. System inputs the completed contact_data <i>[user_id (fk), contact_data_id, contact_data_name, contact_data_address, contact_data_description, contact_data_type_id ]</i> for that Steward into the database</li> </ol>		
Alternative Courses:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

--	--

<b>Use Case ID:</b>	4		
<b>Use Case Name:</b>	Remove a Steward Contact Data		
<b>Created By</b>	Paul Walter	<b>Last Updated By:</b>	Paul Walter
<b>Date Created</b>	07/01/2008	<b>Date Last Updated:</b>	07/01/2008
<b>Actor:</b>	Technician		
<b>Description:</b>	Technician removes an email address for email or for text messaging phones		
<b>Preconditions:</b>	That the database is up and running, and there are contact info for the Stewards to remove		
<b>Post conditions:</b>	The chosen Steward will have an email address that can be contacted if the need arises.		
<b>Priority:</b>	low		
<b>Frequency of Use:</b>	low		
<b>Normal Course of Events:</b>	<ol style="list-style-type: none"> <li>1. Technician desires to remove a contact address for a Steward</li> <li>2. System presents a list of available Stewards</li> <li>3. Technician chooses a Steward to remove a contact address from.</li> <li>4. System presents a list of available contact addresses to choose from <i>[contact_data_id, contact_address, contact_type]</i></li> <li>5. Technician selects the address they wish to remove from the system</li> <li>6. System asks them if they are sure that they wish to remove this address</li> <li>7. Technician confirms</li> <li>8. System removes this Contact Data from the database</li> <li>9. System returns feedback to the Technician that the contact info has been removed.</li> </ol>		
<b>Alternative Courses:</b>			
<b>Exceptions:</b>			
<b>Includes:</b>			
<b>Special Requirements:</b>			
<b>Assumptions:</b>			
<b>Notes and Issues:</b>			

--	--

Use Case ID: 5

Use Case Name: Create Domain

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/01/2008 Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician adds a Domain, a grouping of environment being monitored

Preconditions: That the database is up and running

Post conditions: That there will be a new Domain to monitor

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to add a new domain
2. System requests *[domain\_name]*
3. Technician inputs it
4. System requests *[domain\_description]*
5. Technician inputs it
6. System writes Domain *[domain\_id, domain\_name, domain\_description]* to the database
7. System informs Technician that the addition is successful

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 6

Use Case Name: Delete Domain

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/01/2008 Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician deletes a Domain, and all associated Environments, Sensors, and Cameras.

Preconditions: That the database is up and running, and that there are Domains to remove

Post conditions: That the selected Domain will no longer exist

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires destroy a Domain
2. System presents a list of Domains
3. Technician selects one to destroy
4. System informs user that this will be permanent
5. Technician accepts
6. System deletes all associated Cameras
7. System deletes all associated Sensors
8. System deletes all associated Environments
9. System deletes Domain
10. System informs Technician that the Domain has been removed

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 7

Use Case Name: Associate Steward to a Domain

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/01/2008

Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician associates a Steward with a particular domain.

Preconditions: That the domain exists

Post conditions: That the domain has a Steward associated with it.

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to associate a Steward to a Domain
2. System presents a list of Domains
3. Technician selects a Domain from the list
4. System presents a list of available stewards not currently associated with that Domain (with active status)
5. Technician selects a Steward to associate
6. System informs Technician that the association has been made

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 8

Use Case Name: Disassociate a Steward from a Domain

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/01/2008 Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician disassociates a Steward from a particular domain.

Preconditions: That the domain exists

Post conditions: That the domain has a Steward associated with it.

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to disassociate a Steward from a Domain
2. System presents a list of Domains
3. Technician selects a Domain from the list
4. System presents a list of Stewards associated with that Domain
5. Technician selects a Steward to disassociate
6. System asks Technician if they are sure they wish to do this
7. Technician confirms their wish
8. System removes the association between that Steward from that Domain
9. System informs Technician that the disassociation has been made

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:



--	--

Use Case ID: 9

Use Case Name: ~~Register Domain~~ (Not to be Implemented yet)

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/01/2008

Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician chooses to register this domain in a list of other systems that are out there, so each domain could be accessed from other observers.

Preconditions: That the domain exists

Post conditions: That the domain has a Steward associated with it.

Priority: low

Frequency of Use: low

Normal Course of Events: Technician desires register this domain to a UDDI like list  
TBA....

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 10

Use Case Name: Add an environment

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/01/2008

Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician adds another domain to the monitoring system

Preconditions: That the database is up and running

Post conditions: The new environment will be ready to configure

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to add an environment to be monitored
2. System asks what will the environment's title be *[environment\_title]*
3. Technician inputs that data
4. System asks what the environment's description will be.
5. *[environment\_description]*
6. Technician inputs this data
7. System confirms that the environment is set up and ready to go.
8. System returns Technician to a catalog of available Environments *[environment\_id, domain\_id (fk), environment\_title, environment\_description, environment\_creation\_date]*

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 11

Use Case Name: Remove an environment

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/01/2008

Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician deletes an environment and all associated data with it from the system.

Preconditions: That the database is up and running, and that there are environments available.

Post conditions: The new environment will be ready to configure

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to delete an environment from the system
2. System presents a list of Environments available for deletion on the system
3. Technician selects which Environment they wish to delete *[Environment ID]*
4. System confirms that technician really wishes to delete this environment
5. Technician confirms that this is true
6. System deletes all the Environment's components:
7. Sensors *[sensor name, sensor description, sensor id, critical min, critical max, metric name]*
8. Cameras *[camera name, camera description, camera id, camera media locator,*
9. Environment *[environment name, environment description, environment id]*
10. System supplies feedback that the environment has successfully been removed
11. System returns Technician to a catalog of available Environments *[Environment Id, Environment Name, Environment Description, Environment creation date]*

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID:	12		
Use Case Name:	Add a Sensor		
Created By	Paul Walter	Last Updated By:	Paul Walter
Date Created	07/01/2008	Date Last Updated:	07/01/2008
Actor:	Technician		
Description:	Technician adds a new sensor to an Environment		
Preconditions:	That the database is up and running, and that the Environment they wish to add sensors to exists		
Post conditions:	The Environment will have a new sensor associated with it		
Priority:	low		
Frequency of Use:	low		
1. Normal Course of Events:	2. Technician desires to add a new sensor to an existing Environment 3. System presents a list of available environments 4. Technician selects desired Environment 5. System presents the Environment (details, sensors, and cameras associated with it) 6. Technician selects “add sensor” 7. System prompts for <i>[sensor_title]</i> 8. Technician inputs title 9. System prompts for <i>[sensor_description]</i> to allow for a small blurb explaining why it is there. (can be null) 10. Technician inputs description 11. System prompts for <i>[sensor_type]</i> (Analog or digital) 12. Technician inputs it 13. System prompts for <i>[sensor_address]</i> where does it live on the network? 14. Technician inputs it 15. System prompts for <i>[sensor_metric_name]</i> (if it is degrees, grams, pounds, etc...) 16. Technician inputs the sensor metric name 17. System prompts for <i>[sensor_update_speed]</i> to see how quickly it is going to sample data and write it to the database. (30 times a second, 30 times a minute, 30 times an hour?) 18. Technician inputs the sensor update speed 19. System takes a test sensor reading to see if it is working correctly 20. System presents sensor reading to Technician and asks if this is acceptable, or if they wish to change the <i>[sensor_address]</i> 21. Technician accepts 22. System writes Sensor <i>[sensor_id, environment_id, sensor_title, sensor_description, sensor_type, sensor_address, sensor_metric_name]</i> information to the database 23. System informs Technician that the database write was successful.		

## Alternative Courses:

- 18b. Technician doesn't accept.
- 19b. System returns technician to Sensor details

## Exceptions:

## Includes:

Special  
Requirements:

Assumptions: That the sensor being used is already calibrated

## Notes and Issues:

--	--

Use Case ID: 12

Use Case Name: Add a Sensor

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/01/2008 Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician adds a new sensor to an Environment

Preconditions: That the database is up and running, and that the Environment they wish to add sensors to exists

Post conditions: The Environment will have a new sensor associated with it

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to add a new sensor to an existing Environment
2. System presents a list of available environments
3. Technician selects desired Environment
4. System presents the Environment (details, sensors, and cameras associated with it)
5. Technician selects “add sensor”
6. System prompts for *[sensor\_title]*
7. Technician inputs title
8. System prompts for *[sensor\_description]* to allow for a small blurb explaining why it is there.  
(can be null)
9. Technician inputs description
10. System prompts for *[sensor\_type]* (Analog or digital)

11. Technician inputs it
12. System prompts for *[sensor\_address]*
13. Technician inputs it
14. System prompts for *[sensor\_metric\_name]* (if it is degrees, grams, pounds, etc...)
15. System takes a test sensor reading to see if it is working correctly
16. System presents sensor reading to Technician and asks if this is acceptable, or if they wish to change the *[sensor\_address]*
17. Technician accepts
18. System writes Sensor *[sensor\_id, environment\_id, sensor\_title, sensor\_description, sensor\_type, sensor\_address, sensor\_metric\_name]* information to the database
19. System informs Technician that the database write was successful.

## Alternative Courses:

- 18b. Technician doesn't accept.
- 19b. System returns technician to Sensor details

## Exceptions:

## Includes:

Special  
Requirements:

Assumptions: That the sensor being used is already calibrated

## Notes and Issues:

--	--

Use Case ID: 13

Use Case Name: Remove Sensor

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/01/2008

Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician adds a new sensor to an Environment

Preconditions: That the database is up and running, and that the sensor they wish to delete exists

Post conditions: The Environment will have one less sensor associated with it

Priority: low

Frequency of Use: low

- Normal Course of Events:
1. Technician desires to remove a sensor to an existing Environment
  2. System presents a list of available Environments
  3. Technician selects desired Environment
  4. System presents the Environment (details, sensors, and cameras associated with it)
  5. Technician selects “remove sensor” next to the sensor they wish to remove
  6. System asks if they are sure
  7. Technician confirms
  8. System deletes this sensor from the database
  9. Paul, where is the data from the sensors being written to the database? Will it still exist after its database reference gets deleted?
  10. System informs Technician that the database removal of the Sensor was successful.

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions: That the sensor being used is already calibrated

Notes and Issues:



Use Case ID: 14

Use Case Name: Add Critical Sensor Range

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/01/2008

Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician adds a new Critical Sensor Range

Preconditions: There is a sensor for which you want to associate a Range for.

Post conditions: That the Sensor readings will have a trigger to put the system into action

Priority: low

Frequency of Use: low

Normal Course of  
Events:

1. Technician desires to add a *Sensor Range Trigger*. (The technician can input messages to be sent for nominal ranges, and critical ranges alike)
2. System prompts for which sensor you wish to add this range too.
3. Technician inputs the sensor\_id to be monitored [*sensor\_id*]
4. System prompts for the *sensor\_range\_title* (which will be used to list this range)
5. Technician inputs the *sensor\_range\_title*
6. System prompts for the *sensor\_range\_description* (used to describe what it is we are testing for at this range, why it exists)
7. System prompts for the *sensor\_range\_min* (The minimum threshold that will cause this trigger to fire)
8. Technician inputs the *sensor\_range\_description*
9. Technician inputs the *sensor\_range\_description*
10. System displays the names and descriptions of possible *Sensor Action Scripts* that are available to run once this level has been reached.
11. While (user has more *sensor\_action\_scripts* to input)
12. {
13. System prompts the user for the *XML script* to run once this range is met (*maybe this is in the database, with this I can utilize Reflection of Java in order to call the right methods*) which will outline what actions should be performed once this range is met. [*sensor\_action\_script*]
14. Technician enters the *sensor\_action\_script* name they wish to utilize
15. }
16. Technician indicates that they are finished inputting sensor\_action\_scripts
17. System writes Sensor Range Trigger [*sensor\_range\_id, sensor\_id (fk), sensor\_range\_description, sensor\_range\_title, sensor\_range\_min, sensor\_range\_max*] to the Database
18. System writes associative table between the *Sensor Range Trigger*, and the possibly many *Sensor Action Scripts* that will be called in the event that this range is met
19. System delivers feedback that the Sensor Range has been successfully written.

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions: That there is Sensor Action Scripts already in existence.

Notes and Issues:

--	--

Use Case ID: 15

Use Case Name: Remove a Critical Sensor Range



Created By	Paul Walter	Last Updated By:	Paul Walter
Date Created	07/01/2008	Date Last Updated:	07/01/2008
Actor:	Technician		
Description:	Technician adds a new Critical Sensor Range		
Preconditions:	There is a sensor for which you want to disassociate a Sensor Range with.		
Post conditions:	That the Sensor readings will have a trigger to put the system into action		
Priority:	low		
Frequency of Use:	low		
Normal Course of Events:	<ol style="list-style-type: none"> <li>1. Technician desires to remove a Sensor Range from a particular Sensor</li> <li>2. System provides a list of available <i>Sensor Ranges</i> for a given <i>Sensor</i></li> <li>3. Technician selects the sensor they wish to remove</li> <li>4. System prompts for confirmation (E.g. “are you sure you want to do this?”)</li> <li>5. Technician confirms</li> <li>6. System removes all references between this Sensor Range and the Sensor Action Script table in the Database</li> <li>7. System removes Sensor Range from Database</li> <li>8. System returns a success message</li> </ol>		
Alternative Courses:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:	That there are <i>Sensor Ranges</i> to remove from a given <i>Sensor</i>		
Notes and Issues:			

--	--

Use Case ID:	15		
Use Case Name:	Remove a Critical Sensor Range		
Created By	Paul Walter	Last Updated By:	Paul Walter

Date Created	07/01/2008	Date Last Updated:	07/01/2008
Actor:	Technician		
Description:	Technician adds a new Critical Sensor Range		
Preconditions:	There is a sensor for which you want to disassociate a Sensor Range with.		
Post conditions:	That the Sensor readings will have a trigger to put the system into action		
Priority:	low		
Frequency of Use:	low		
Normal Course of Events:	<ol style="list-style-type: none"> <li>1. Technician desires to remove a Sensor Range from a particular Sensor</li> <li>2. System provides a list of available <i>Sensor Ranges</i> for a given <i>Sensor</i></li> <li>3. Technician selects the sensor they wish to remove</li> <li>4. System prompts for confirmation (E.g. “are you sure you want to do this?”)</li> <li>5. Technician confirms</li> <li>6. System removes all references between this Sensor Range and the Sensor Action Script table in the Database</li> <li>7. System removes Sensor Range from Database</li> <li>8. System returns a success message</li> </ol>		
Alternative Courses:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:	That there are <i>Sensor Ranges</i> to remove from a given <i>Sensor</i>		
Notes and Issues:			

--	--

Use Case ID:	16		
Use Case Name:	Test a Sensor		
Created By	Paul Walter	Last Updated By:	Paul Walter
Date Created	07/01/2008	Date Last Updated:	07/01/2008

Actor: Technician

Description: Technician tests a Sensor that was just added, to make sure that it works correctly

Preconditions: There is a sensor for which you want to test

Post conditions: That the Technician will get feedback as to the Sensor's status, and a reading if it is properly configured

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to take a sensor reading from a particular sensor
2. System displays the sensors of a given Environment registered to a given Domain.
3. Technician selects that Sensor to be tested.
4. System returns a success message and a reading that the sensor returned

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions: That there is a Sensor to return

Notes and Issues:

--	--

Use Case ID: 17

Use Case Name: Add a Camera

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/01/2008 Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician configures a camera to work with a particular Environment

Preconditions: There is an Environment set up to receive this Camera

Post conditions: That the Environment will have a Camera associated with it.

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to associate a Camera on the Network with a particular Environment
2. System prompts user for the *camera\_name*.
3. Technician supplies this name
4. System prompts user for the *camera\_description*
5. Technician supplies this description
6. System prompts user for the *camera\_media\_location*
7. Technician supplies this information
8. System prompts user for the *camera\_snapshot\_interval* (that is the interval between the when the software fires off a picture)
9. Technician supplies this information
10. System attempts to contact this device, and if successful returns a live image coming from this camera
11. System writes the Camera [*camera\_id, environment\_id (fk), camera\_name, camera\_title, camera\_media\_location, camera\_snapshot\_interval*] to the Database
12. System informs user of the success of this operation.

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions: That there is a Sensor to return

Notes and Issues:

--	--

Use Case ID: 17

Use Case Name: Remove a Camera

Created By Paul Walter

Last Updated By:

Paul Walter

Date Created 07/01/2008

Date Last Updated:

07/01/2008

Actor: Technician

Description: Technician wishes to disassociate a Camera from a given Environment

Preconditions: There is an Environment set up to remove this Camera from

Post conditions: That the Environment will have a Camera associated with it.

Priority: low

Frequency of Use: low

Normal Course of Events:

1. Technician desires to associate a Camera on the Network with a particular Environment
2. System supplies a list of Cameras available for that particular Environment
3. Technician selects one to remove
4. System removes the Camera and returns feedback to the user.

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions: That there is a Sensor to return

Notes and Issues:



Use Case ID: 18

Use Case Name: Test a Camera

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/01/2008 Date Last Updated: 07/01/2008

Actor: Technician

Description: Technician wishes to Test a Camera

Preconditions: There is an Environment set up to remove this Camera from

Post conditions: That the Camera will be tested to see if it properly configured.

Priority: low

Frequency of Use: low

- Normal Course of Events:
1. Technician desires to test a given Camera associated with a given Environment
  2. System supplies list of available cameras in the Environment.
  3. System prompts user for *camera\_name*
  4. Technician chooses *camera\_name* from list
  5. System pulls a current picture from that camera

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions:        That there is a Sensor to return

Notes and Issues:

### 3.1.2 Steward's Use Cases

Use Case ID:	19.5
Use Case Name:	Login
Created By	Paul Walter
Last Updated By:	Paul Walter
Date Created	11/22/2008
Date Last Updated:	11/22/2008
Actor:	Steward
Description:	The steward will be able to log into the system.
Preconditions:	That the steward be registered in the database.
Post conditions:	That the Steward will have access to the system afterwards.
Priority:	High
Frequency of Use:	High
Normal Course of Events:	<ol style="list-style-type: none"> <li>1. Steward wishes to login, and provides a username and password [<i>username, password</i>]</li> <li>2. The system checks to see if they are a steward or technician, and if so returns a success message</li> <li>3. The system displays a list of domains available to the user to select.</li> </ol>
Alternative Courses:	2a. The system tells them their username and password combo are invalid, and to try again.
Exceptions:	
Includes:	
Special Requirements:	
Assumptions:	
Notes and Issues:	

Use Case ID: 20

Use Case Name: Choose a Domain

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/08/2008

Date Last Updated: 07/08/2008

Actor: Steward

Description: The Steward will be able to choose a specific Domain from a list of Domains available on the server

Preconditions: That the Domains exist to be chosen on the server

Post conditions: That a domain has been chosen to be viewed

Priority: High

Frequency of Use: High

Normal Course of Events:

1. Steward wishes to select an Domain available on the System
2. System presents the Steward with a list of available Environments associated with a specific Domain *[domain\_id, domain\_name, domain\_description]*
3. Steward selects a domain
4. System displays details associated with that Environments contained within

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 20

Use Case Name: Choose an Environment

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/08/2008

Date Last Updated: 07/08/2008

Actor: Steward



Description: The Steward will be able to choose a specific Environments from a specified Domain available on the server

Preconditions: That the Domain and the Environments exist to be chosen on the server

Post conditions: That a domain has been chosen to be viewed

Priority: High

Frequency of Use: High

Normal Course of Events:

1. Steward wishes to select an Environment attached to a specific Domain
2. System presents the Steward with a list of available Environments associated with a specific Domain *[environment\_id, Environment\_name, Environment\_description]*
3. Steward selects a domain
4. System displays details associated with that Environment like Sensors, and Video Cameras registered to that Environment

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 21

Use Case Name: View Environment Time line

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/08/2008

Date Last Updated: 07/08/2008

Actor: Steward

Description: The Steward will be able to view the Sensor data collected on that particular environment

Preconditions: That the specified Environment has recorded sensor and video data along with Time line Notes to display.

Post conditions:

Priority: High

Frequency of Use: High

Normal Course of Events:

1. Steward wishes to view the recorded history of a particular Environment
2. System displays a time line
3. Recorded Sensor Data
4. Recorded Video Data? (Still picture thumbnails?)
5. Time line Notes left by the Steward

Alternative Courses:

Exceptions:

Includes:

Special Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 22

Use Case Name: Write Time line Note

Created By Paul Walter Last Updated By: Paul Walter

Date Created 07/08/2008 Date Last Updated: 07/08/2008

Actor: Steward

Description: The steward will be able to make notes in the Time line

Preconditions:

Post conditions:

Priority: High

Frequency of Use: High

Normal Course of Events:

1. Steward wishes to write a note to be visible in the Timeline
2. System prompts for *note\_title*
3. Steward inputs it

4. System prompts for *note\_date* which will identify what day it is to be displayed on
5. Steward inputs it
6. System prompts for *note\_body*
7. Steward inputs the note they wish to leave
8. System confirms that the note has been created

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 23

Use Case Name: Delete Time line Note

Created By Paul Walter

Last Updated By:

Paul Walter

Date Created 07/08/2008

Date Last Updated:

07/08/2008

Actor: Steward

Description: The steward will be able to make notes in the Time line

Preconditions:

Post conditions:

Priority: High

Frequency of Use: High

Normal Course of  
Events:

1. Steward wishes to delete a note from an Environment
2. System displays a list of notes available on the Environment
3. Steward selects one to delete
4. System asks if they are sure they want to do this
5. Steward confirms yes this is what they want to do
6. System deletes the note and tells the Steward that this has been done correctly.

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions:

Notes and Issues:

--	--

Use Case ID: 24

Use Case Name: View Video

Created By Paul Walter

Last Updated By: Paul Walter

Date Created 07/08/2008

Date Last Updated: 07/08/2008

Actor: Steward

Description: The steward will be able to pick a video camera and view the information from it.

Preconditions:

Post conditions:

Priority: High

Frequency of Use: High

Normal Course of Events:

1. Steward wishes to view video from a specific camera
2. System displays a list of cameras available on that Environment
3. Steward selects the camera that they wish to see video from
4. System displays that video

Alternative Courses:

Exceptions:

Includes:

Special  
Requirements:

Assumptions:

Notes and Issues:

### 3.1.3 System's Use Cases

Use Case ID:	24		
Use Case Name:	Send Danger Message		
Created By	Paul Walter	Last Updated By:	Paul Walter
Date Created	07/08/2008	Date Last Updated:	07/08/2008
Actor:	System		
Description:	The system will notify a steward of any sensor that has reached a critical measurement range and needs to be looked at		
Preconditions:			
Post conditions:			
Priority:	High		
Frequency of Use:	High		
Normal Course of Events:	<ol style="list-style-type: none"> <li>1. The system gets notification from a sensor that its current measurement warrants Steward notification</li> <li>2. The system sends a text message or email to the cell phone or email address of that user.</li> </ol>		
Alternative Courses:			
Exceptions:			
Includes:			
Special Requirements:			
Assumptions:			
Notes and Issues:			

## 4. External Interface Requirements

### 4.1 User Interfaces




During our busy lives it is easy to overlook the green spaces in our homes. We sometimes forget to water our rosemary, to limit the light for our lizards and turtles, or to keep our desert terrariums from becoming too moist.

Throw them a lifeline. Login and set up devices that will alert you when your cherished green spaces require more attention.

How it works: upload a map or drawing of the area you would like to raise awareness for such as the first floor of an apartment or house, set up areas you wish to be alerted about, like a terrarium in the kitchen, or potted plants in the living room, then choose what devices are going to be keeping tabs on those areas.

Username  
Password

Login



+

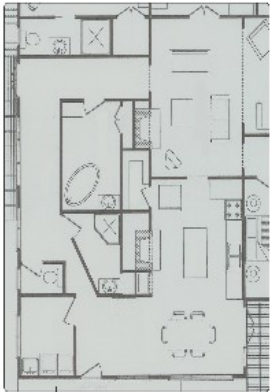
-

Maps

Users

●

First floor, Paul's House




Status:

- The turtle tank is getting too hot.
- The kitchen terrarium is getting too much sun.

●

Second floor

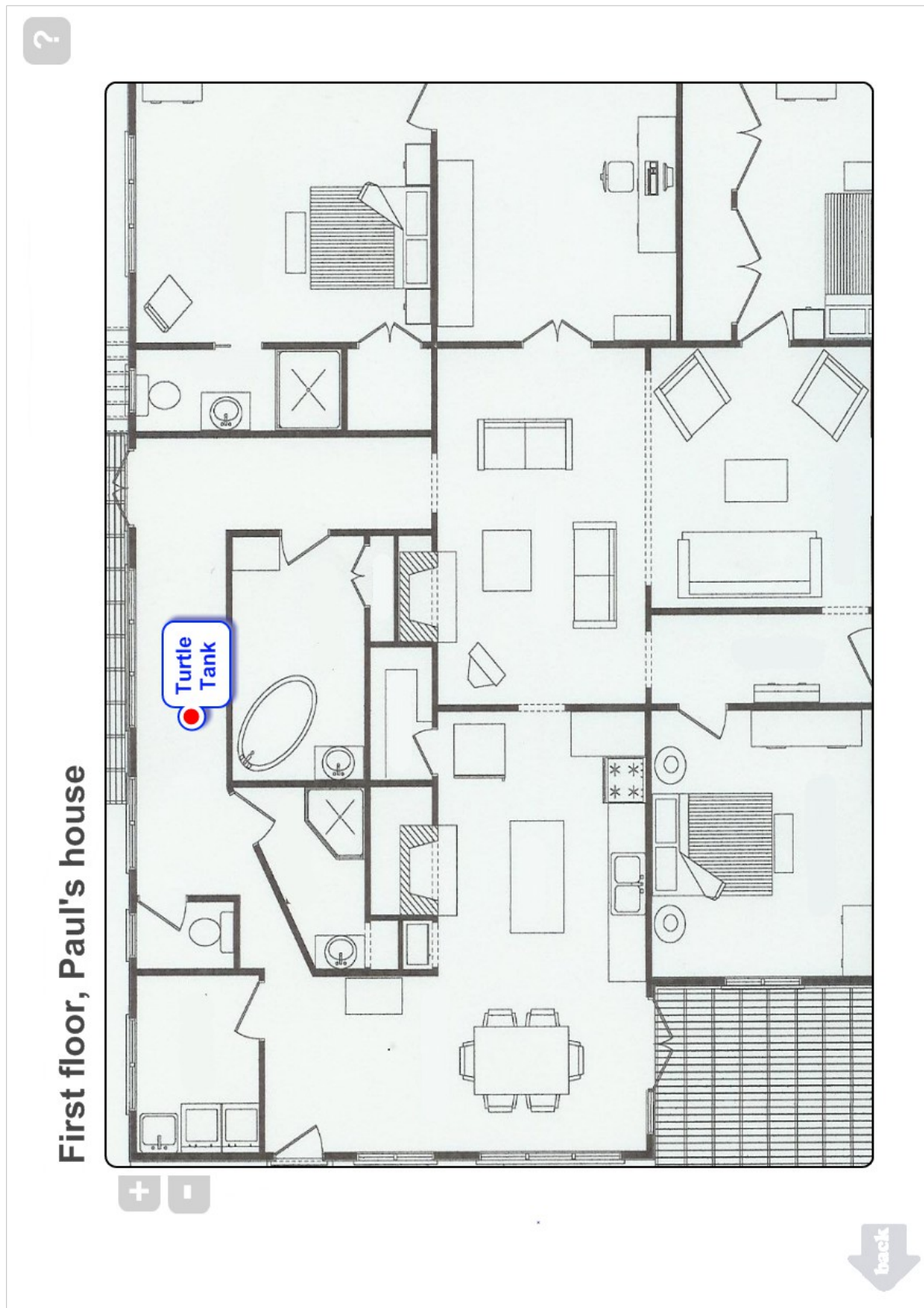


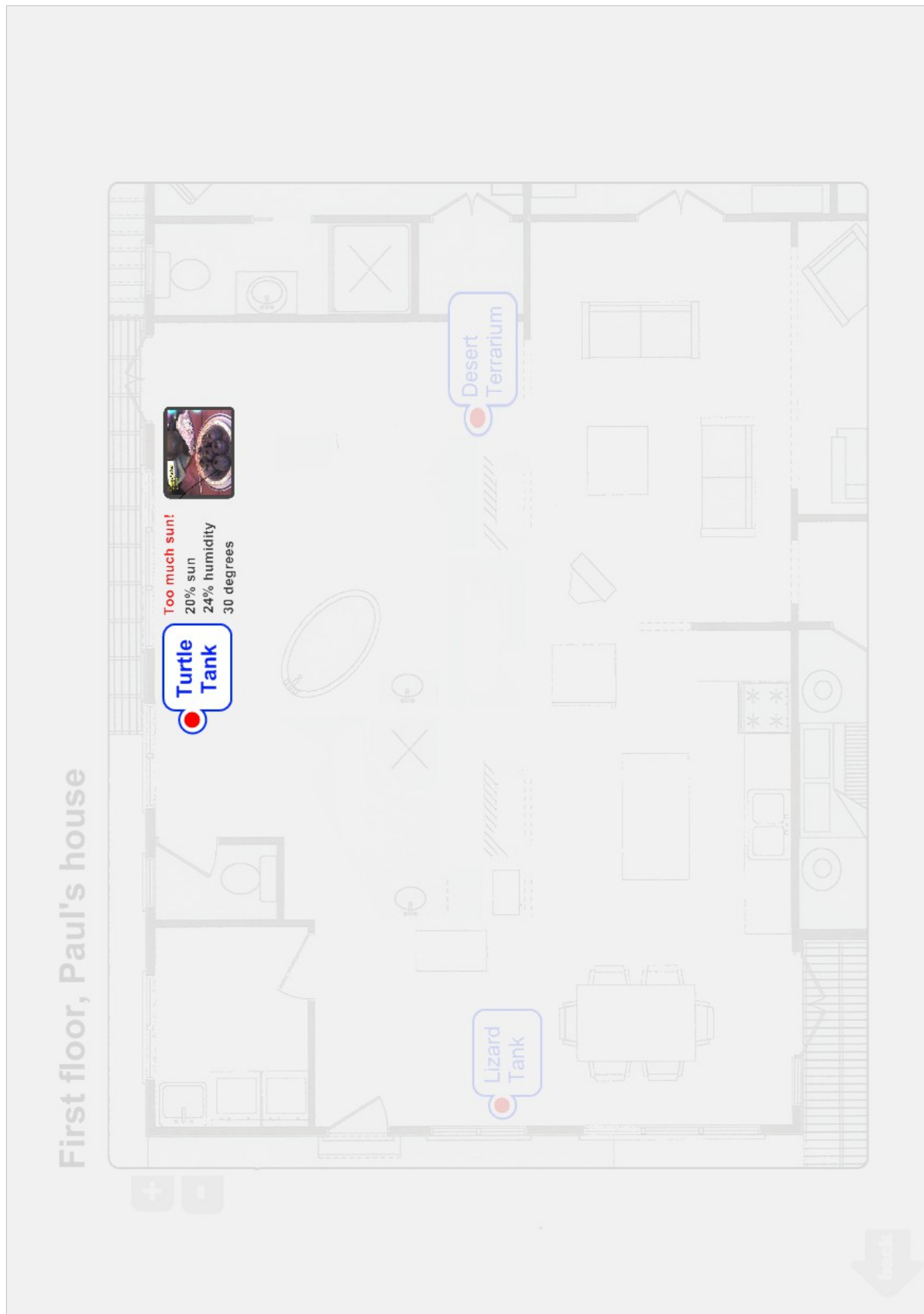
Status :

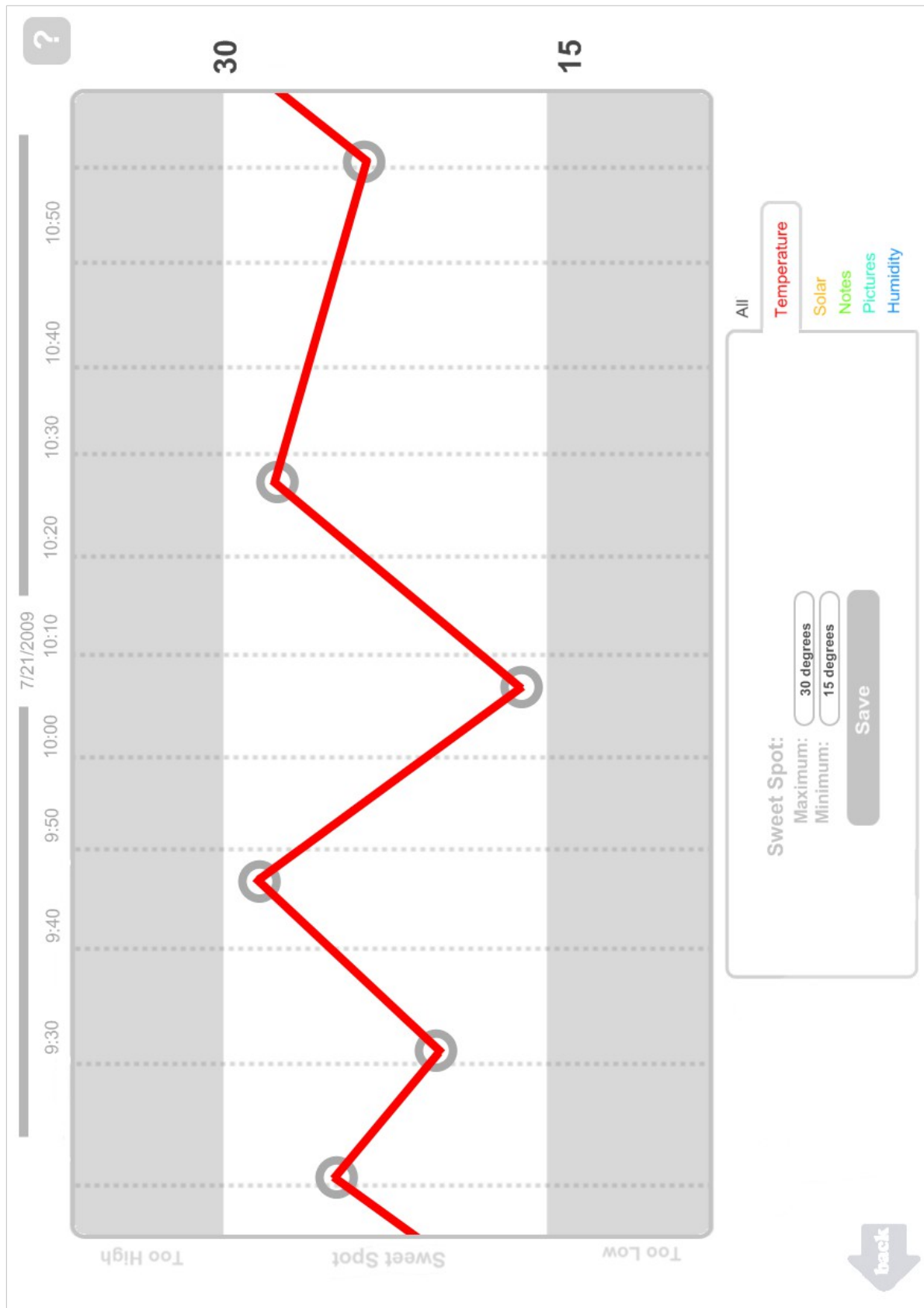
- The iguana cage is doing ok.
- The rosemary plants are doing fine as well.

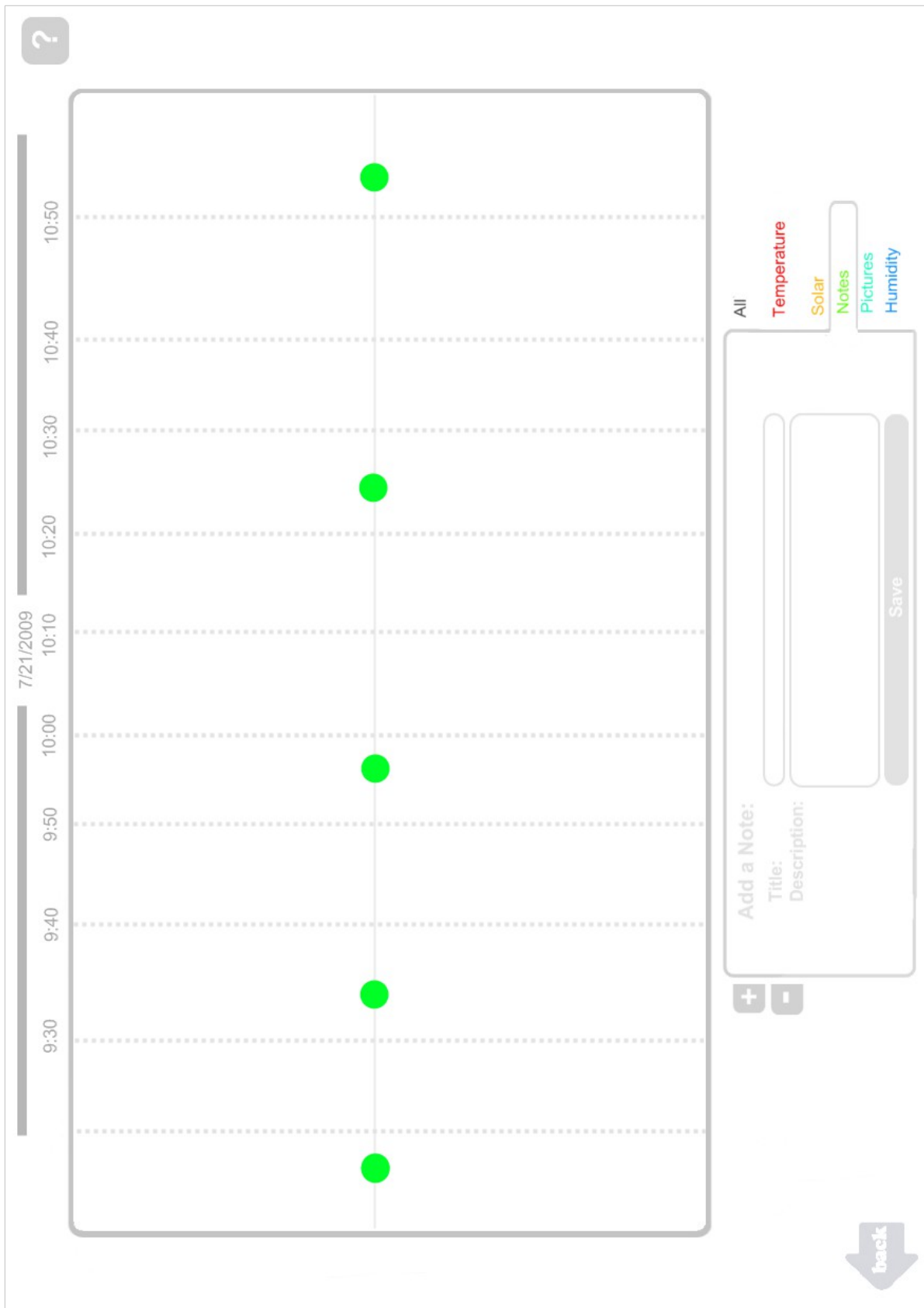
Logout











*<Describe the logical characteristics of each interface between the software product and the users. This may include sample screen images, any GUI standards or product family style guides that are to be followed, screen layout constraints, standard buttons and functions (e.g., help) that will appear on every screen, keyboard shortcuts, error message display standards, and so on. Define the software components for which a user interface is needed. Details of the user interface design should be documented in a separate user interface specification.>*

## 4.2 Hardware Interfaces

The hardware interfaces are two fold in this system: Web Cameras and “1 Wire” sensors. The web camera is just a USB model that plugs into the computer, and the one wire sensors plug into a serial port.

*<Describe the logical and physical characteristics of each interface between the software product and the hardware components of the system. This may include the supported device types, the nature of the data and control interactions between the software and the hardware, and communication protocols to be used.>*

## 4.3 Software Interfaces

- Java version 6.
- Java Mail
- Java Media Framework
- Glassfish, Java Application Server
- WPF
- Maxim's “1 Wire” drivers available from their website.
- VMWare Workstation / VMWare Player
- Oracle Database.
- Gmail

*<Describe the connections between this product and other specific software components (name and version), including databases, operating systems, tools, libraries, and integrated commercial components. Identify the data items or messages coming into the system and going out and describe the purpose of each. Describe the services needed and the nature of communications. Refer to documents that describe detailed application programming interface protocols. Identify data that will be shared across software components. If the data sharing mechanism must be implemented in a specific way (for example, use of a global data area in a multitasking operating system), specify this as an implementation constraint.>*

## 4.4 Communications Interfaces

*<Describe the requirements associated with any communications functions required by this product, including e-mail, web browser, network server communications protocols, electronic forms, and so on. Define any pertinent message formatting. Identify any communication standards that will be used, such as FTP or HTTP. Specify any communication security or encryption issues, data transfer rates, and synchronization mechanisms.>*

# 5. Other Nonfunctional Requirements

## 5.1 Performance Requirements

*<If there are performance requirements for the product under various circumstances, state them here and explain their rationale, to help the developers understand the intent and make suitable design choices. Specify the timing relationships for real time systems. Make such requirements as specific as possible. You may need to state performance requirements for individual functional requirements or features.>*

## 5.2 Safety Requirements

*<Specify those requirements that are concerned with possible loss, damage, or harm that could result from the use of the product. Define any safeguards or actions that must be taken, as well as actions that must be prevented. Refer to any external policies or regulations that state safety issues that affect the product’s design or use. Define any safety certifications that must be satisfied.>*

## 5.3 Security Requirements

*<Specify any requirements regarding security or privacy issues surrounding use of the product or protection of the data used or created by the product. Define any user identity authentication requirements. Refer to any external policies or regulations containing security issues that affect the product. Define any security or privacy certifications that must be satisfied.>*

## 5.4 Software Quality Attributes

*<Specify any additional quality characteristics for the product that will be important to either the customers or the developers. Some to consider are: adaptability, availability, correctness, flexibility, interoperability, maintainability, portability, reliability, reusability, robustness, testability, and usability. Write these to be specific, quantitative, and verifiable when possible. At the least, clarify the relative preferences for various attributes, such as ease of use over ease of learning.>*

## 6. Other Requirements

*<Define any other requirements not covered elsewhere in the SRS. This might include database requirements, internationalization requirements, legal requirements, reuse objectives for the project, and so on. Add any new sections that are pertinent to the project.>*

### 6.1 Appendix A: Data Dictionary

Descriptions use Genus Differentia format (“*X is a Y that Z*”, where “X” is a thing, “Y” is the family that “X” belongs to, and “Z” is how this thing is different from all other things in that family.)

It didn't seem necessary to define the Entity attributes in the same format.

**User:** Is either a Steward or Technician that has a vested interest in maintaining the Domains and their associated Environments

user\_id

user\_username

user\_password

user\_status

**Contact Data:** Is an address (for text message or email) that allows the system to contact the user

user\_id (fk)

contact\_data\_id

contact\_data\_name

contact\_data\_address (could be a phone number or email addy)

contact\_data\_description

contact\_data\_type\_id (whether it is a phone or an email)

**Domain:** Is a collection of environments that are grouped together

domain\_id

domain\_name

domain\_description

**Domain Steward** (associative): Is an association that links Steward\_Contact\_Data to User

domain\_steward\_id

domain\_id (fk)

user\_id (fk)

**Environment:** Is a collection of sensors and cameras that the steward is looking after

environment\_id

domain\_id (fk)

environment\_title

environment\_description

environment\_creation\_date

**Environment\_Note:** Is a note written by a user regarding a particular Environment

environment\_note\_id text summarizes the note.

environment\_note\_message the body of the note

environment\_note\_datetime when the note was written  
 environment\_id (FK) is the environment this note is associated with  
 user\_id (FK) is the user who wrote this note

**Sensor:** Is an input device that can take measurements.

sensor\_id  
 environment\_id (fk)  
 sensor\_title  
 sensor\_description  
 sensor\_type\_id  
 sensor\_address  
 sensor\_metric\_id (fk)  
 sensor\_update\_speed

**Sensor Reading:** Is a recorded measurement of a sensor that the Steward uses to track data trends over time

sensor\_reading\_id  
 sensor\_id (fk)  
 sensor\_reading\_datetime  
 sensor\_reading\_measurement

**Sensor Range:** Is a measurement range of sensor, that once reached will trigger a Sensor\_Range\_Script to be called.

sensor\_range\_id  
 sensor\_id (fk)  
 sensor\_range\_description  
 sensor\_range\_title  
 sensor\_range\_min  
 sensor\_range\_max

**Sensor Range Script:** Is a piece of xml that defines what actions to take once the Sensor has reached a particular range. Here we are going to use Java reflection to call the right methods defined in the document.

sensor\_range\_script\_id  
 sensor\_range\_id (fk)  
 sensor\_range\_script\_xml  
 sensor\_range\_script\_title  
 sensor\_range\_script\_description

**Sensor\_Range\_And\_Its\_Script (associative table):** is a link that associates a Sensor\_Range and any number of Sensor\_Range\_Scripts

sensor\_range\_and\_its\_script\_id  
 sensor\_range\_id  
 sensor\_range\_script\_id

**Camera:** Is an input device that can record still-pictures in a binary format.

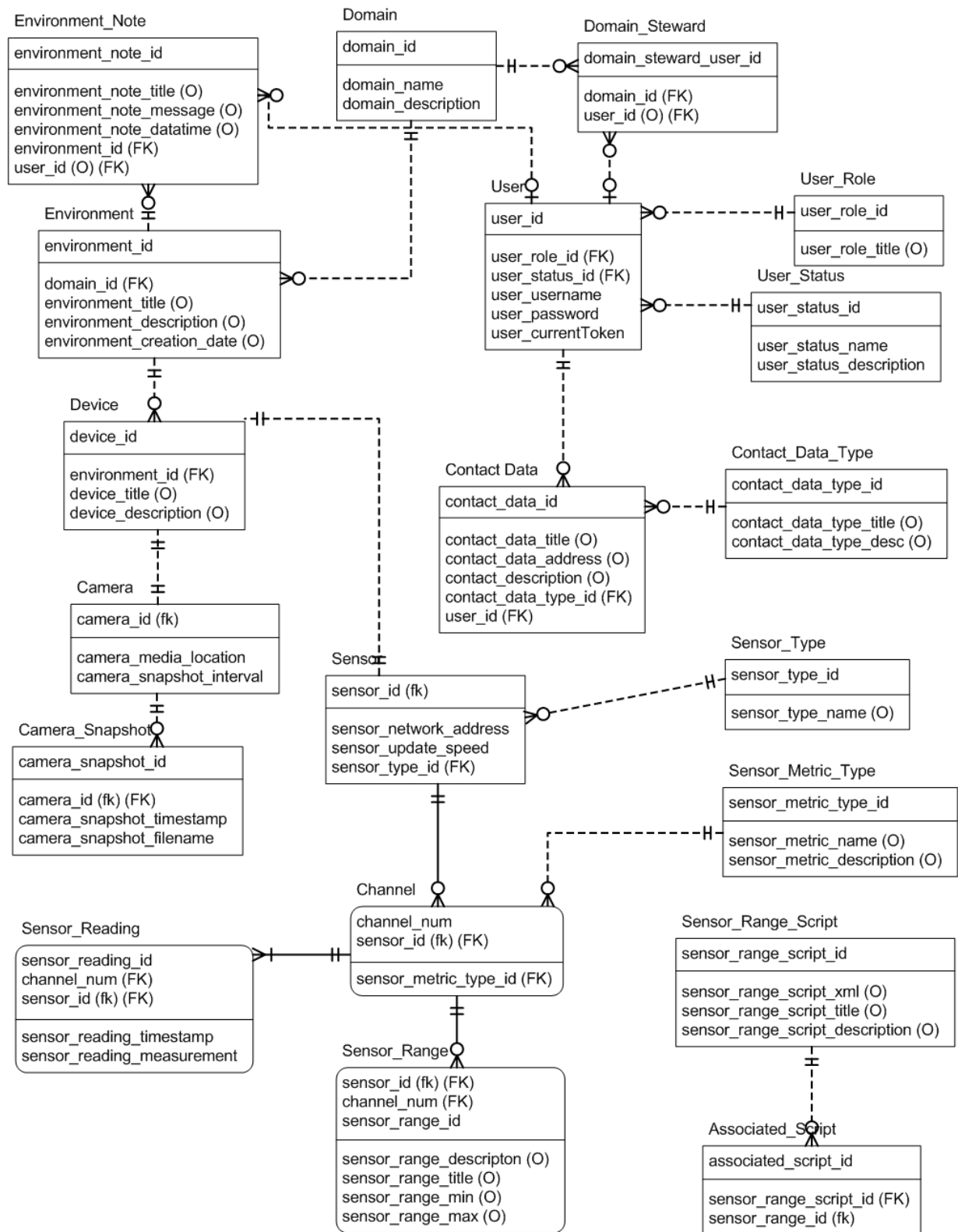
camera\_id  
 environment\_id (fk)  
 camera\_title  
 camera\_media\_location  
 camera\_snapshot\_interval

**Camera Snapshot:** is a still frame picture that gets recorded by a Camera.

camera\_snapshot\_id  
 camera\_snapshot\_datetime  
 camera\_snapshot



## **6.2 Appendix B: Entity-Relationship Diagram**

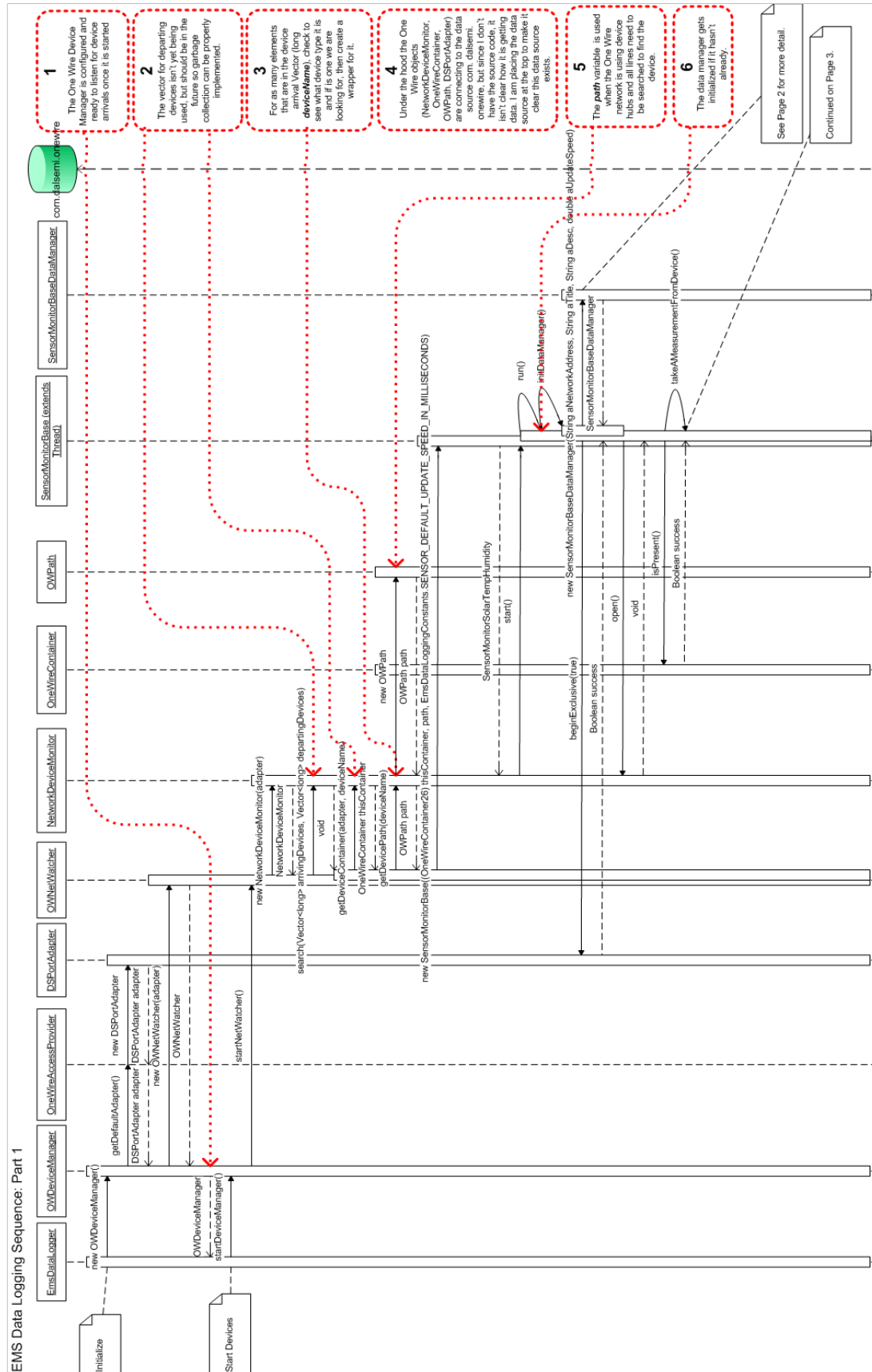


## 6.3 Appendix C: Sequence Diagrams

The following are sequence diagrams that illustrate the structure and behavior of the resulting application. See the sequence below.

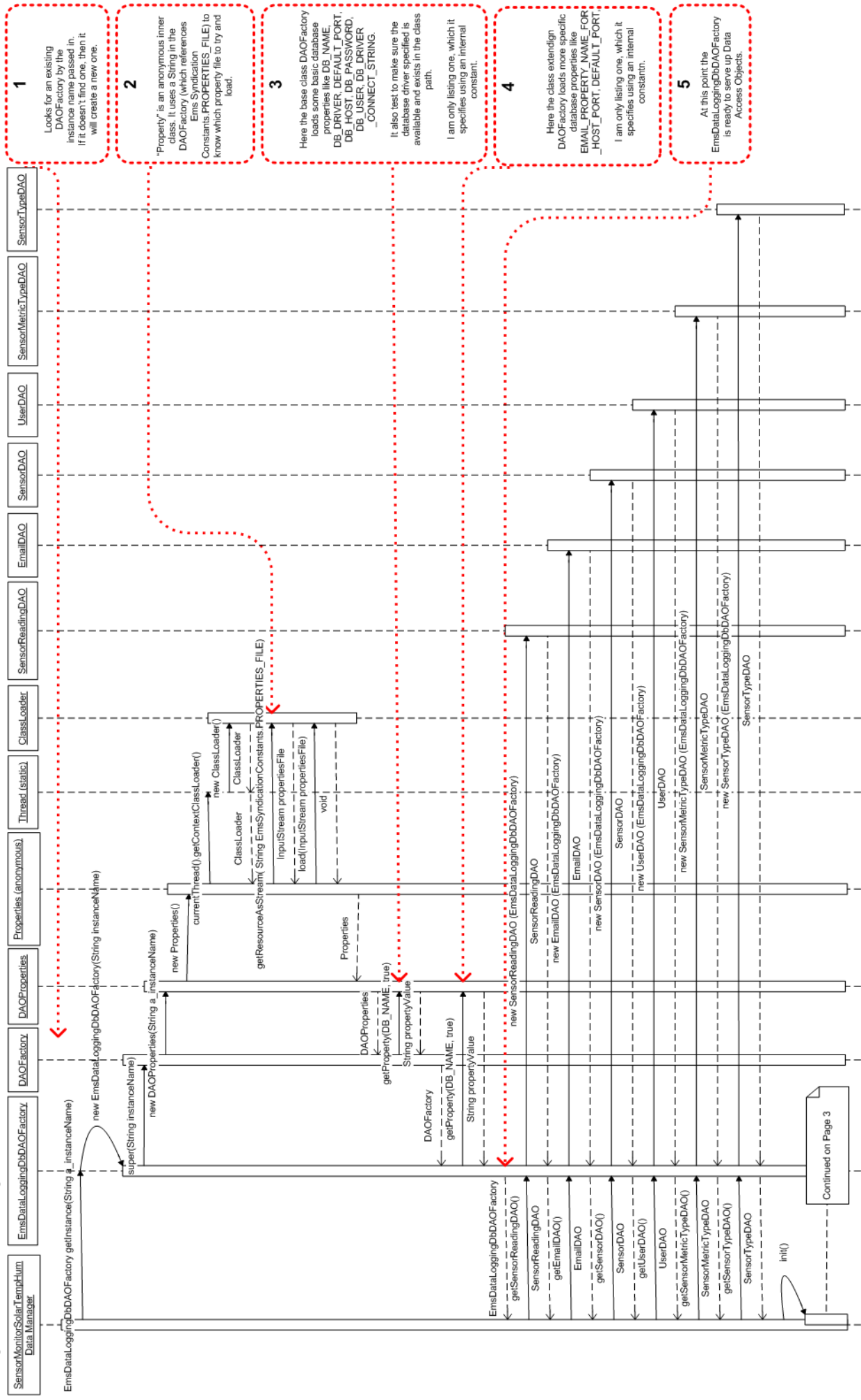
The first one is illustrates the use of a custom data layer that marshaled data objects from the database.

## 6.3.1 The Custom Data Layer

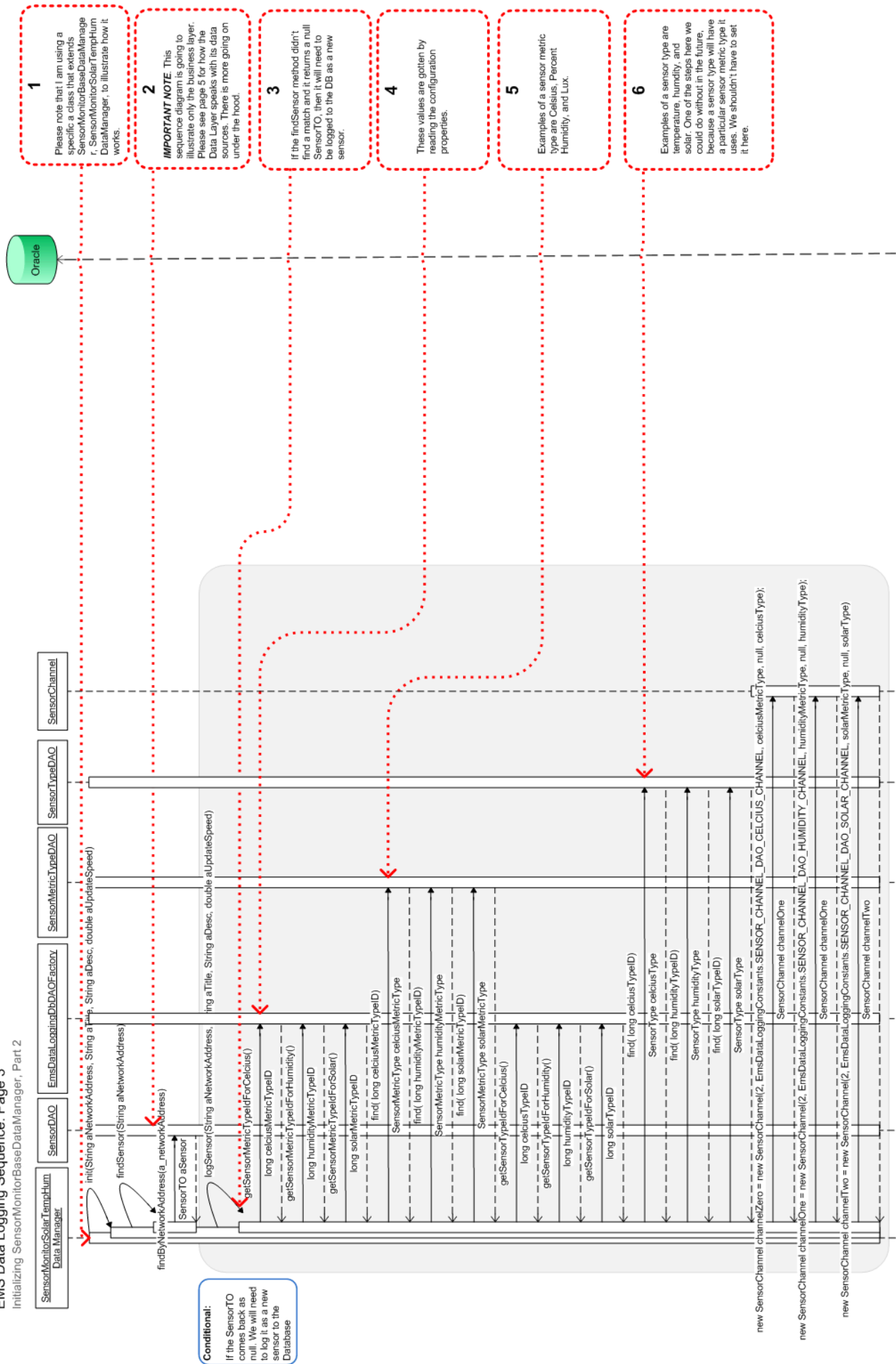


EMS Data Logging Sequence: Page 2

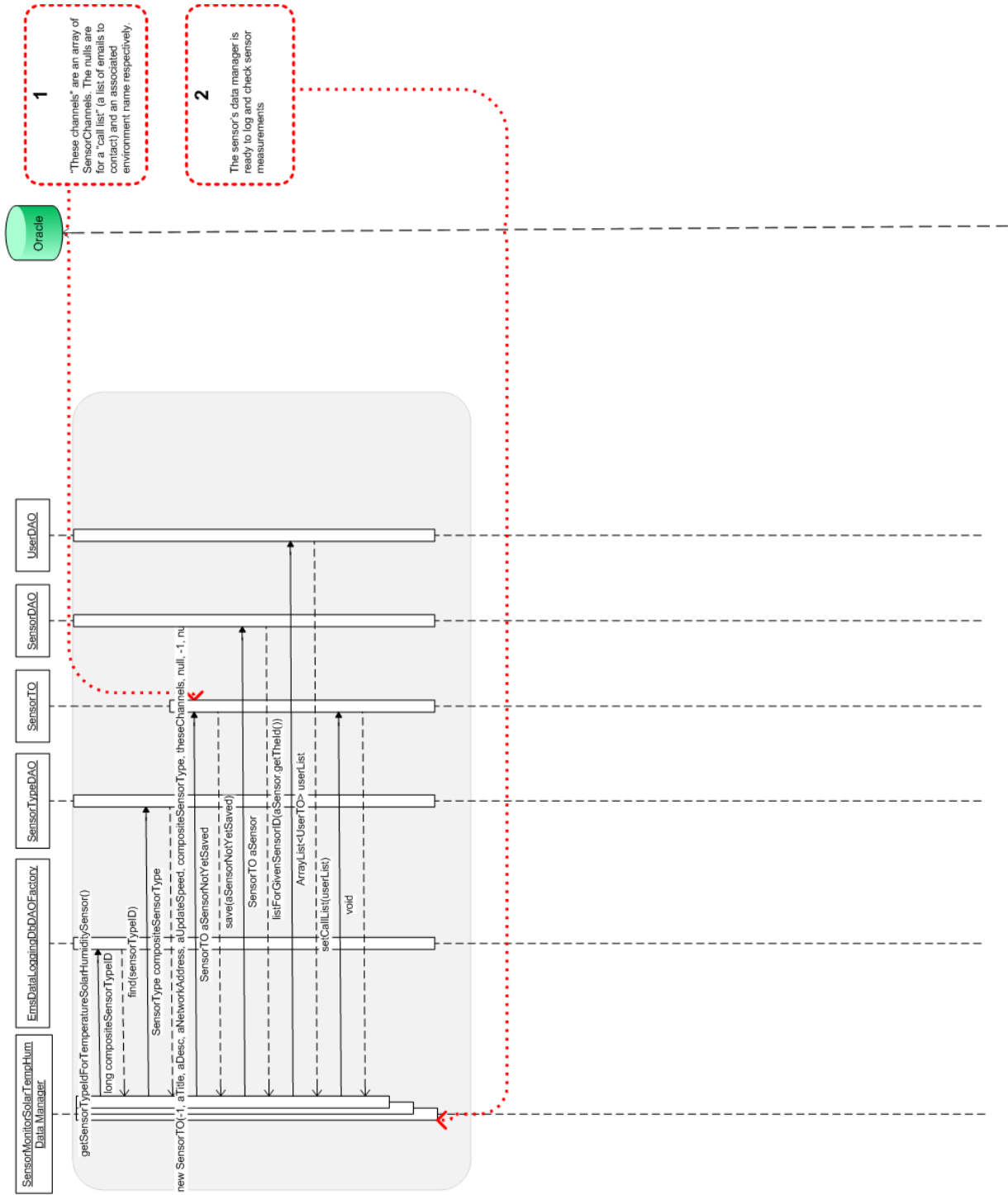
## Initializing SensorMonitorBaseDataManager, Part 1



EMS Data Logging Sequence: Page 3  
Initializing SensorMonitorBaseDataManager, Part 2



EMS Data Logging Sequence: Page 4  
 Initializing SensorMonitorBaseDataManager, Part 3



## 6.4 Appendix E: Issues List

### 6.4.1 Registering new devices into the system

The first issue is that I didn't work into the use case diagram the need for the technician to register sensors. This is addressed regardless in the resulting software: the sensor client scans its network, finds all the network addresses on the network, and sends that list to the server. The server saves any addresses not already in the database into the database. The problem is that it doesn't know what kind of sensor is being reported. The technician needs to go in later and manually update the database.

In addition to the use case diagram, there needs to be another use case added that illustrates how this interaction will occur. This isn't in the Mock up, and it is a mystery as to what is the best course of action.

Ideally, when a new device gets registered, it gets listed in the Technician's view of the system as a notification, and then the Technician can make the semantic decision as to what type of sensor this is so it's measurements get assigned the correct metrics (temp, lux, percent humidity).

### 6.4.2 Scalability

Another issue is that this design doesn't scale. It is limited in two ways: the first is that it can only hold as many sensors and web cameras as can be connected to one computer through it's USB and serial ports. Not a very good design, but as this was developed as a proof of concept, it is adequate for now. The second limitation is that it uses an Object Oriented approach which means that multithreading needs to occur and lock shared objects. This would make it very slow at scale.

In a future design, the developers should look at distributing the network AND either use functional programming (C#, Scala, or Erlang) or use an Actor Model (in Java, specifically the Play Framework).

### 6.4.3 Cross Platform Capabilities for the Client GUI

Since the client is written in WPF it can only be run on windows systems. In the future this should/could be written in something more compatible like Javascript and HTML.

### 6.4.4 Separate Technician GUI for the Sensor Client

In addition to the GUIs that already exist, there should be another one, perhaps command line that walks the Technician through setting up a new Sensor Client node. Something that will list all the sensors and cameras and perhaps let the technician choose from the point of entry, which user this will be registered to, what *Environment*, and what *Domain*.

### 6.4.5 Authentication



This is something that needs to be addressed. At the moment, the database doesn't use MD5 hashing for passwords, nor is there a “I forgot my password” functionality.

#### **6.4.6 Java Media Framework (JMF) is Dead**

The JMF is the library I used to interact with a web camera. One of the things discovered during this process is that the JMF is only partially implemented and doesn't have any plans to complete. The incredibly frustrating thing is that the project doesn't say this anywhere. It's only digging through user group posts on the “Stack Overflow” site, that I found this.

This project may want to look at, in the future, writing the Sensor Client in C++ or Python . Both of these languages have far richer support for interacting with hardware. Java's strength seems to be with business computing (i.e. running a server). It doesn't want to get it's hands dirty with the low-level of hardware.

C++ also has a library called “Open Layers” which might be worth a look.

Java has another library which is slated to be complete in the future, but after the way that Java Media Framework went, I don't have much faith. It's name is Freedom for Media in Java (FMJ). It simplifies the

#### **6.4.7 “1 Wire” Framework is Dead**

This framework was only ever meant as a hobbyist tool. In the future you need to consider using the “I2C” for sensor networks. It is like “1 Wire” in that it only needs one wire to communicate with sensor nodes, BUT it is an industry standard and is used for mission critical devices, yet retains a reasonable level of complexity.

#### **6.4.8 This project needs to be Geospatial.**

This prototype needs to be Geospatial in the future. At the moment the Technician needs to manually specify (in the database) the location of the sensors in the database so they show up in the correct location when rendered to the Technician GUI. In the future the Data Model needs to be changed to incorporate GPS locations as well as indoor GPS capabilities so that maps can be rendered to the screen that are accurate, and not hand-drawn.

The Sensor Client in the future might need to be GPS aware to help with the data logging aspect of the application.

*< This is a dynamic list of the open requirements issues that remain to be resolved, including TBDs, pending decisions, information that is needed, conflicts awaiting resolution, and the like.>*