Team 7 Sprint Three Planning Document

Anna Benjamin, Kathryn Frankewich, Austin Klasa, Bridgette Kuehn, Matt Molo

Sprint Overview:

This planning document will describe how our project will be divided into each member's tasks for the course of sprint three.

Goals:

- Allow web application users to customize data viewing.
- Allow weather station owners to publish data.
- Share the source code information so developers can write and publish their own code.
- Make sure the webpage has quick performance and only shares information that is okay to be shared.
- Make sure the webpage is scalable and usable.
- Complete the physical weather station.
- Inspect all code and functioning.

Scrum Meetings:

We will meet every Tuesday, Thursday, and Saturday from 3:00 PM to 4:15 PM for our SCRUM stand up meetings with all group members that are available at the time. We will discuss the progress of each task and any issues that have arisen.

Scrum Master:

Kathryn Frankewich

Risks and Challenges:

We have a few challenges for this sprint. Making sure data is sent and received in a reliable manner and that use info is secure will be worked on. Also, everything is now hooked together, but it's a bit rough around the edges, so making sure that things don't break for the end user and it's responsive for them.

Current Sprint Detail:

1. As an amateur meteorologist (web application user), I would like to:
    a. Customize what weather data is shown.
2. As a weather station owner, I would like to:
    a. Publish well-formatted data for analysis of the weather.
    b. Choose to publish my personal weather data.

Task Description	Owner	Estimated Time (hours)
Enable custom selection of fields to view in the filtered data	Bridgette	5
Enhance the previous location filtering	Bridgette	2
Add in a check to make sure no data is made public that has been marked not sharable	Bridgette	1
Incorporate filtered data into JSON file in order to create graphs	Bridgette	3
Send JSON filtered data to web page to display	Bridgette	2
Accept old and new weather station info requests into database	Bridgette	2
Modify the code so that when there are SQL errors, the web pages are still user friendly	Bridgette	5
Narrow down current data to all data only updated in the last hour	Bridgette	2
Incorporate ability to sort data in the filtered data table by columns	Bridgette	2
Create a page where users can view their data without using filters and bookmark their favorite weather station.	Austin	2
Create more advanced web application filter where a user can choose which fields to filter. - Field selection - Adaptive filter based on selection - Save previous input	Austin	7
Create filter form validation.	Austin	5
Create organized charts from filtered data.	Austin, Anna	10
Load current weather if no WS is close enough	Anna	5
Load forecast for current location from 3rd party	Anna	4
Add icons and visualizations to display current weather	Anna	5

1. As a Raspberry Pi developer, I would like to:
    a. Modify the source code to build upon the framework.
    b. Write the code for the weather sensors in the Raspberry Pis.
    c. Publish the aforementioned code.

Task Description	Owner	Estimated Time (hours)
Clean up sensor code base and add comments	Matt	2
Add instructions and source from Github to webpage	Kathryn	2

Non-Functional Tasks:

1. Performance:
    a. Make sure pages load in 3 seconds or fewer.
    b. Make sure the server, database, and Raspberry Pi clients send/receive information to and from one another in less than one second.
2. Security:
    a. Make sure people can successfully keep their data private.
3. Scalability:
    a. Make sure that the system can handle as many or as few clients as necessary.
4. Usability:
    a. Make sure the users can easily and quickly figure out how to find their weather information on our product.
    b. Make sure users know how to customize the interface to their liking.

Task Description	Owner	Estimated Time (hours)
Stress test server	Matt	2
Add option to save favorite weather station on webpage	Kathryn	2
Get all weather station data sensors into a thread and save it to a local variable in weatherStation so the Raspberry Pi web response is quicker.	Kathryn	3

Non-Functional Weather Station Tasks:

Task Description	Owner	Estimated Time (hours)
Write code and have working wind vane	Matt	8
Package and solder weather station together	Matt	10

Complete installation instructions	Matt	2
Save previous sensor data while running	Kathryn	5
Respond to web requests for "historical data"	Kathryn	5
Implement saving an instance of a weather station in the settings text file instead of having to save multiple variable values.	Kathryn	5
Add reconfiguration settings for config of the Raspberry Pi	Kathryn	2

Software Test:

Task Description	Owner	Estimated Time (hours)
Website code inspection and test	Everyone	2
Server backend code inspection and test	Everyone	2
Weather station code inspection and test	Everyone	2

Team Member	**Hours**
Anna Benjamin	30
Austin Klasa	30
Bridgette Kuehn	30
Kathryn Frankewich	30
Matt Molo	30
TOTAL:	**150**

Backlog:

Functional:

3. As an amateur meteorologist (web application user), I would like to:
    a. View weather data on a mobile app (if time allows)
4. As a weather station owner, I would like to:
    a. Integrate my weather station data with a mobile app (if time allows)