

Team 7 Sprint One 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 one.

Goals:

- Set up database and phpMyAdmin
- Configure server and get team members access
- Accomplish the basic website design using HTML, CSS, and JavaScript
- Create "fake" data to be used in developing the webapp
- Spec out weather sensors for Raspberry Pi
- Investigate anemometer (wind sensor)
- Work on Java REST api
- Setup php page to post weather data from Raspberry Pi's to database
- Write Classes
- Write the Java code to run the server

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. The first is getting all team members acquainted with the different software and tools that we will be using over the course of our entire project. Another challenge will be keeping the interaction between the database, server, and web application consistent since multiple team members will be working on these different components. The last challenge for this sprint will be building or finding an anemometer that will work well for our purposes.

Current Sprint Detail:

User Case: As an amateur meteorologist, I would like to view current and historical weather data on the web application.

Task Description	Owner	Estimated Time (hours)
Use JavaScript to optimize the functionality of the webpage	Anna	15
Use CSS to optimize the front end of the webpage	Anna	15
Basic web application structure	Austin	15
Setup interaction between webapp and server	Kathryn	5
Pull basic data from database to display to user	Kathryn	1
Write text to go on webpages	Kathryn	1.5

User Case: As an amateur meteorologist, I would like to navigate the webpage.

Task Description	Owner	Estimated Time (hours)
Create menu bar to navigate between pages.	Austin	1
Build a basic homepage to display welcome information, display current weather data, and a menu bar button to allow navigation.	Austin	2
Build a basic about page to display information about our product and a menu bar button.	Austin	2
Build a basic contact page to display information to contact our team and a menu bar button to allow navigation.	Austin	2
Build a basic how to page to display information about how to create a weather station, view source code and a menu bar button.	Austin	4
Build a basic view data page to display historical weather data, a form to collect input from the user, and a menu bar button.	Austin	4

Non-Functional Code Setup:

Task Description	Owner	Estimated Time (hours)
Setup Raspberry Pi ability to send data to the server	Bridgette	2
Create Raspberry Pi classes to parse all weather sensor data	Bridgette	5
Setup server to receive data from the Raspberry Pi	Kathryn	3
Create and format "fake" weather sensor data for all sensors	Bridgette	8
Write Weather Station Class	Kathryn	5
Setup model(s) to send data to/from the database	Kathryn	4
Setup controller to handle web application requests	Kathryn	8
Setup view to display information to users on webapp (coordinate with Austin)	Kathryn	3
Test all classes for functionality	Bridgette	10

Non-Functional Architecture Server Side Development:

Task Description	Owner	Estimated Time (hours)
Set up MySQL database and tables	Matt	2
Get phpMyAdmin setup for management	Matt	1.5
Get access for everyone to run tools	Matt	2
Setup php/web server and give access	Matt	4
Write page to post weather data to database	Bridgette	3

Non-Functional Weather Station Tasks:

Task Description	Owner	Estimated Time (hours)
Spec out sensors	Matt	5
Investigate anemometer (wind sensor)	Matt	5.5
Work on Java REST api to be able to poll Pi for data	Matt	10
Write How-To Documentation for setting up a Pi and Creating a weather station	Bridgette	2

Team Member	Hours
Anna Benjamin	30
Austin Klasa	30
Bridgette Kuehn	30
Kathryn Frankewich	30.5
Matt Molo	30
TOTAL:	150.5

Backlog:

Functional:

1. As an amateur meteorologist (web application user), I would like to:
 - a. View weather data from all weather stations.
 - b. View weather data from a specific station.
 - c. View weather data on a mobile app (if time allows)
 - d. Customize what weather data is shown.
2. As a weather station owner, I would like to:
 - a. View my personal weather station data.
 - b. View aggregated weather data.
 - c. Publish well-formatted data for analysis of the weather.
 - d. Choose to publish my personal weather data.
 - e. Integrate my weather station data with a mobile app (if time allows)
3. 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.

Non-functional:

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.