Sprint 3 Planning Document
TEAM 5
WPEAR

Stephen Harrell, Lala Vaishno De, Mengxue Luo, Dhairya Doshi

1. Sprint Overview

This sprint will be focused on polish and usability. A key functionality this sprint will be to expand the functionality and usability of the web. Beyond that we will be cleaning up some loose ends and adding some new features such as finding the best and worst forecasts within a time period and finishing some user stories from the last sprint. By the end of the sprint we expect to have a command line tool that creates a usable website for view weather data and forecast evaluation data.

Scrum Master: Stephen Harrell
Scrum Meeting Time: Monday at 3:00pm and Thursday at 4:00pm
Risks/Challenges:

- Getting a usable web design.
- Finding good visualizations
- Making the whole application user friendly

2. Current Sprint Details

Clean-up user stories from sprint 2:

1) **User Story:** As a user, I would like to have the data interpolated before evaluation for the visualizations.

Task Description	Estimated Hours	Owner
Be able to interpolate a forecast data file to a grid size of 1 sq. km.	1	Lala
Be able to interpolate an observation data file to a grid size of 1 sq. km.	1	Lala
Test to ensure the interpolated objects for forecasts and observations can be compared. (compatible with each other)	1	Lala

Acceptance Criteria:

1. Interpolate the data values of a given message object belonging to a forecast file to the required grid size.
2. Interpolate the data values of a given message object belonging to an observation file to the required grid size.
3. The interpolated objects from a forecast and an observation file can be compared. Passes test cases to prove their compatibility.

2) **User Story:** As a user, I would like to be able to see the trend in temperature for a given region (5 x 5 mile around a given point).

Task Description	Estimated Hours	Owner
Create a method that takes in multiple files from the WPEARController along with co-ordinates for a given point	3	Dhairya
Create a trend line graph for this that can be displayed on the website.	6	Dhairya

Acceptance Criteria:

1. Given that we have the GRIB files and the co-ordinates for the point, we should be able to find the 4 points in the GRIB file that enclose the region
2. Given that we can find the 4 points, we should be able to calculate the average temperature over the 4 points for every file individually.
3. Given that we can calculate the average for each point, we should be able to create a trend graph to display the temperature trend.

New user stories:

3) **User Story:** As a user, I would like to view a moving heatmap of the observed weather versus forecasted weather by difference.

Task Description	Estimated Hours	Owner
Compare 24 hours worth of observations with their corresponding 6 hour forecasts	4	Mengxue
Visualize each comparison and create a “frame”	4	Mengxue
Combine all frames into an animated gif	4	Mengxue

Acceptance Criteria:

1. Given that we have the data, we should have an animated gif for the difference between the 6 hour forecast and the observations for the day available in the corresponding date directory.

4) **User Story:** As a user, I would like to view a X-Y scatter graph showing the standard deviation between a selected observation and its forecasts.

Task Description	Estimated Hours	Owner
Calculate the individual standard deviations between a single forecast and an observation.	3	Lala
Join the individual standard deviations into a single data entity.	4	Lala
Plot the graph showing the changing standard deviation between the forecasts against a single observation	10	Lala
Store the graph in its required destination	5	Lala

Acceptance Criteria:

1. Given the observation and its corresponding forecasts, be able to generate a new file with the X-Y scatter graph showing the changing standard deviation of the forecasts and the observation.

5) **User Story:** As a user, I would like to use date/hour information to navigate the website and pick the visualization I want to see.

Task Description	Estimated Hours	Owner
Implement a homepage that allows user to select the date-hour for which they want to see information using a calendar-like interface	10	Lala
Create multiple HTML static files with the necessary information and visualizations for each date-hour-atmospheric variable	8	Mengxue

Acceptance Criteria:

1. Given that we have a homepage, the user should be able to select a date-hour. The homepage should redirect to a secondary page.

2. Given that we have all the information, there should be one static HTML page for each date-hour-atmospheric variable.

6) **User Story:** As a user, I would like to view the results via a website.

Task Description	Estimated Hours	Owner
For each date-hour-atmospheric page, sectionalize and allow navigation to different sections using buttons linked to tags.	8	Dhairya
Be able to incorporate additional features that are/will be implemented and provide appropriate labels for each	5	Dhairya
Implement a cleaner layout using some form of CSS/Style management	10	Dhairya

Acceptance Criteria:

1. Given that we are able to display the page, the information and visualizations should be displayed in a sectional format with clear labels and descriptions.
2. Clicking on individual buttons scrolls/snaps to the corresponding section on the page.
3. Given that all the necessary data is available, we should be able to integrate and display results from various forms of analysis as well as the appropriate visualizations for the same.

7) **User Story:** As a user, I would like to be able to download the data files used to generate the visualizations

Task Description	Estimated Hours	Owner
Add links to the grib2 files that were used to generate visualizations	4	Dhairya
Allow the user to download the file(s) when the link is clicked	4	Dhairya

Acceptance Criteria:

1. Given that we are able to display the visualizations, the user should also have links to the data files used to generate the visualizations.
2. Given that we have the links, the user should be able to download the files.

8) **User Story:** As a developer, I would like to standardize and cleanup the code from the last two sprints.

Task Description	Estimated Hours	Owner
Remove hardcoded atmospheric variables and make variables that can be programmatically changed	5	Stephen
Standardize temp file placement	5	Stephen
Create command line arguments for many hard coded variables	10	Stephen

Acceptance Criteria:

1. Given that the code is working, atmospheric variables can be programmatically changed
2. Given that the whole system is working, all temporary files should be placed under one directory and removed appropriately.

9) **User Story:** As a developer, I would like to integrate all of the new visualizations into the controller workflow.

Task Description	Estimated Hours	Owner
Integrate observation vs forecast difference	5	Stephen
Integrate website changes	10	Stephen
Integrate standard deviation	5	Stephen

Acceptance Criteria:

1. Given that the visualization and website code is working, it is run from the WPEARController across all relevant data.

10) **User Story:** As a developer, I would like to create a wiki that would explain the software we have created and give detailed instructions on how to use it.

Task Description	Estimated Hours	Owner
Explanation of what the product does, its use case, system requirements	5	Lala
Detailed instructions on how to run the program and what each different flag/setting does	10	Mengxue
Detailed notes on how the product works and where the files are stored	10	Mengxue
Troubleshooting tips and possible next steps	5	Lala

Acceptance Criteria:

1. Wiki is updated to Github and clearly explains what the product does, what some of the ideal use cases are and the min system requirements to run the product.
2. Wiki gives clear instruction on how to run the program with different configurations and settings.
3. Wiki explains the design of the product and each component works.

Summary of allocated time:

Team Member	Estimated Total Hours
Stephen	40
Lala	40
Mengxue	40

Dhairya	40
Total Hours Assigned	160 hours

3. Remaining Backlog

(a) Include all the other user stories from your Product Backlog document.

Functional Requirements:

1	As a user, I would like to view a moving visualization of the comparison done using root mean square difference. (if time permits)
2	As a user, I would like to view a graph showing the accuracy of a specific variable based on how far out have they been forecasted (if time permits).
3	As a developer, I would like to perform evaluation calculations on the observation and forecast and compare the models. (if time permits)
4	As a user, I would like to see the difference between the results from top accurate weather forecast and the worst accurate weather forecast. (if time permits)

Non-Functional Requirements:

1. **Web Enabled:** Visualizations must be available over the web. Webpages must be autogenerated based on the visualizations that are to be displayed using type of visualization, date/time, location and variable as parameters.

2. **Intermediate Data Archival:** Interpolated intermediate data must be available historically for reanalysis of visualizations. Archived data must be clearly marked with location, time/date and variable parameters. Archived data must be available via the web and linked to the visualizations that are created from it.

3. **Operational Weather:** Every hour this tool should retrieve the observations and forecasts, convert them to a common grid spacing and format and create visualizations based on an evaluation of the observation and forecast for specific points in time.

4. **Web access must be fast:** Webpages should be static HTML/CSS and not include any server side programing.

5. **Modular:** Code must be modularized in a way that adding new types of observations and forecasts is trivial.