# Rocket Weather App Progress Report

Wayne Johnson <joh09024@umn.edu>, Jacob Long <longx329@umn.edu>,
William Beksi <beksi001@umn.edu>, Andrew Schmid <schm1498@umn.edu>

## Project Accomplishments

The Rocket Weather App project can be broken up into two main components: the back end server infrastructure, and the client front end mobile Android application.

### Back end accomplishments

Feature implementation of the back end server as specified by the design document is 100% complete. During development, we made several modifications to the design which helped reduce complexity and allowed us to reuse existing modules and frameworks.

A significant burden was lifted by using a METAR Python module for decoding METAR codes into parsable object code (https://pypi.python.org/pypi/metar/). We still had to implement a parser for GFS MAV data decoding which took significant effort, but we were able to leverage the design of the METAR module to bootstrap the development.

Another design improvement we made was to use an ORM library as a front end to the SQL cache. Using the ORM SQLAlchemy gives us the ability to switch out the back end data-store without modifying the interface code. With this in mind, we opted to use SQLite as the SQL backend Instead of using MySQL due to its simplicity for this project.

Aside from these deviations the server was implemented to specification. The only thing left to do is additional validation and testing of METAR and MAV decoding.

### Front end accomplishments

Created an Android mobile application that queries the back end server using a HTTP Post request with JSON parameters, parsed the returning JSON data for either METAR or GFS MAV requests, and displayed the information on the screen in a current conditions or forecast activity. Application also has a menu entry and activity that maintains persistent data containing the weather station ID and the server's URL.

## Future Work

New features and enhancements can be done to the product in future development.

- Find the nearest weather station based on geolocation (GPS and/or network locator).
- Add winds aloft data
- User Interface is mostly test. We could make better use of the limited screen space by using icons for items like sky conditions (overcast, broken, scattered or sunny).

# Group Member Contributions

## Wayne Johnson

Wrote project proposal and major parts of Design Document. Created server VM running Centos 6.3 with Apache, MySQL, and Python to be used for the project. Built basic Android App environment using the Eclipse/Android SDK. Created JSON interface for Android side. GFS MAV forecast. Wrote basic settings screen that permits user to change weather station and server URL settings and maintains them as persistent preferences. Coached Jacob on Eclipse/Android SDK. Also wrote several unit tests for the METAR and GFS data beans and JSON demarshaller. Wrote Button in UI go from GFS page to METAR page.

## Jacob Long

Daily updates of progress calendar if needed. Wrote User interface for the METAR page and the GFS page. Tested codes written by teammates (minimal). Learned a lot about Python and Android programming.

## William Beksi

Setup source code repository and configured Apache on the back end server. Leveraged existing Python METAR package for parsing METAR weather reports and added functionality to the package for returning the report in JSON format. Implemented MAV module for parsing MAV weather reports.

## Andrew Schmid

Designed server architecture, Implemented client-server JSON interface, server request processing (request path), server request caching, JSON response code in MAV module, test code, and contributed to report and design doc preparation.