Plans and Specifications

This is where you should include an outline of the key milestones for your project and who will achieve them.  
  
Get my comments.

Requirements Specification:

* What is this project supposed to do?
* Be specific.
* The specification should include scenarios (HINT, HINT)!

System Specification:

* Hardware and software base for your project.
* A PC with JDK 6.0 is reasonable, but you might add more details, or make other selections, as appropriate.

**Sortie Logging – Zachary Hager, Charles Kimmel, and Matthew White**

-GUI  
 \*Language – Java  
 \*GUI – Java application, with goal of integrating into database  
 \*Drop down with various aircraft  
 \*Some sort of selection for the scheduled time AND actual time  
 \*Warning/Error for a flight time +- 30 minutes from scheduled time  
 \*Logging – Early goal to write a file and store it within the program, with the goal to store information in a database.  
 \*\*either way I think “it” will need to be in a database so that the information is retrievable and displayable. Might want to put into a database to be able to possible utilize data visualization tools.

**What is this project supposed to do?**

-Basic Functionality-  
1. Some sort of GUI for user to input:  
 a. Aircraft (dropdown)  
 b. Scheduled Time (text box)  
 c. Actual Takeoff time (same as scheduled time)  
2. Warning (pop up) for a time that is outside of 30 minutes from scheduled time  
3. Storage solution  
 a. Write and save a file  
 b. Write to a database

-Upgrades (in no particular order)-  
1. Aesthetics  
2. Database security  
3. Cloud options

**Minimum System Specifications**

RAM: 256 MB

Disk space: 256 MB Total: (124 MB for JRE; 2 MB for Java Update, *tentative* 128 MB for program)

Processor: Minimum Pentium 2 266 MHz processor

Browsers: Internet Explorer 9 and above, Microsoft Edge, Firefox, Chrome

I was floating around ideas for an aircraft sortie logger. The final product would be a GUI that would have drop downs of your aircraft, then enter scheduled and actual take-off times. Exceptions would be implemented as well, such as requesting the user to input reasons why an aircraft took off more than 30 minutes early or late if the “actual time” is greater or less than 30 minutes from the “scheduled time” as well as cancelations. After the data is collected and submitted it will make a separate file that is stored in the program in a separate directory that can be retrieved.

That’s the nuts and bolts of my idea, but I was mainly interest in working on a project where the user can interact with a GUI, input information, then archive records that can later be retrievable. I think a program like this would be good for this class too as the project will be moving through phases. Maybe early phases we would strive to get the crucial parts of the program working in the console, then move to exception handling and building a GUI. In later phases we can have a Beta, and add components we may not have initially thought off. I am not married to my sortie tracker by any means. That being said I think working on a project that follows this trajectory would be easy to set defined goals for each week, and end with a product that encapsulates a lot of what we have learned up to this point in a useful product.