**Danielle F**

**Survey github:** <https://github.com/koa2019/survey>

1. **Time spent?**
   1. Friday 7-8 hours
   2. Saturday 8 hours
   3. Sunday 1 hour
2. **How was your time spent?**
   1. **Number of lines of code written/modified**: 500
   2. **Documentation**:
      1. Created 2 versions of survey
      2. In the top section of the main.cpp in every version of a program, I write a list of what I did to that and what still needs to be done as comments.
3. **Progress for Project - Gantt Chart**
   1. **Show schedule and progress.** 
      1. The Gantt chart is in the same folder as this document.
      2. Friday: restructure hierarchy for Admin and User classes.
      3. Saturday:
         1. Reviewed Twitter Bootstrap for my GUI. I’ve used it before, but needed to check if I wanted to use it again.
         2. Rewrote 1 record in text file
         3. Researched:
            1. How to edit a record in a binary file?
            2. How to delete a record in a binary file?
            3. How to return a pointer array of class objects within a classes cpp file?
      4. Sunday:
         1. Write-up progress report.
         2. Update gantt chart.
         3. Display questions with menu choices working in survey\_v2 main.cpp.
   2. **Discuss specific tasks accomplished during the week.**
      1. I reviewed all four versions of your Multiple\_Structure\_ReadWrite files in github and searched for the syntax on how to define a function that returns a pointer in the the classes cpp file to avail.
      2. Friday I took my working User and Admin classes from yahtzee\_\_v13, and started restructuring its hierarchy. Yahtzee has Admin as base class and User class inherits the Admin, but I had to write a lot of extra code to accommodate the Admin’s private member functions who read binary files and the flow didn’t feel right.
      3. Now my Survey project has the User as the base class and Admin inherits the User class.
         1. User & Admin can write to text and binary files
         2. Admin writes to its personal text & binary files.
         3. Created read1Bin( ) because findByIndx( ) and findByEmail( ) had redundancy. read1Bin accepts a char, ‘a’ for admin.dat or ‘u’ for usr.dat, and the location to set the cursor to begin reading.
      4. Fixed bug: Admin is crashing. Added a cout inside ~Admin().
      5. Added recSize & begnFile variables to Admin because it made it easier to handle throughout the program.
      6. reWrtBin( ) rewrites the value of hiScore in userData.dat after it calls findByEmail() to set this record’s size & the beginning bit location of this file in binary. Later I was able to rewrite the entire record.
      7. reWrtTxt( ) rewrites an entire record in userData.txt.
      8. BUG! Admin findByIndx() isn't working correctly and crashes after 1 run. I changed the while loop because it wouldn’t calculate index 0’s record size. I need the record size because at the time I was trying to find a record by its index and then edit it. I changed direct and used findMyEmail( ) because it was working properly.
      9. I copied v1 to my yahtzee project as v15 and v2 as v16