**Software Development Life Cycle**

Name: Jack Gilliam

1. Requirements
   1. What does the application do?
      1. The program is meant to perform operations such as creating, updating, deleting, and reading from a file
   2. Is it for-profit or non-profit
      1. The program is meant to be non-profit
   3. Is it open source or proprietary?
      1. The program is proprietary
   4. Requirements by Priority
      1. Level one: High Priority
         1. It is high priority that the program performs file operations such as:
            1. Being able to create, read, write, and delete from a file.
         2. Data that is written includes:
            1. Firstname
            2. Lastname
            3. Email
      2. Level two: Medium Priority
      3. Level Three: Low Priority
      4. Level four: Extras
         1. A Graphical User Interface
         2. Validation of the email
   5. USE cases(all of them)
      1. User scenarios
2. Specifications (Technical Lead)
   1. Design Structure
      1. What does it look like?
         1. It looks like a console window
      2. What are its boundaries?
         1. The keyboard is used as input
   2. Coding Standards
   3. Infrastructure
   4. Integration tests
3. Architecture (Foundation)
   1. The basic flow of the application
      1. A command from the keyboard goes to the console
      2. The program uses the inputted answer
      3. The program performs the chosen operation
      4. The program asks the user what else they want to do
4. Design (Appearance)
   1. User Interface (UI) Design
      1. The UI is a console window
   2. Model Design
   3. UML Diagrams
5. Implementation (Code it)
   1. Code according to the steps above
6. Testing (Stress it)
   1. Take the ‘tree’ approach, how will the user progress through the UI? What snags will they hit?
      1. The program asks for the name of a file to open
         1. Menu is displayed (The Menu)
            1. DoCreate

Three values are entered: Firstname, Lastname, and Email

Program creates the entry and returns to (The Menu)

* + - * 1. DoUpdate

A list of entries in the file is displayed, the user chooses an entry

The user repeats the previous step but with the properties of the entries

The user changes the entry chosen, and goes to (The Menu)

* + - * 1. DoDelete

The user goes through the same process as DoUpdate, but deletes the entry chosen instead

* + - * 1. Exit

The application quits

* 1. Will bugs be introduced when adding, modifying, or removing code?
     1. No, unless necessary code is being removed, such as the Main method.

1. Debugging
   1. Fix errors revealed through testing (Squish dem bugs)
   2. If errors were fixed go to step 6
      1. No errors were found
2. Deployment
   1. Making executable files or plug-ins from code
   2. Put your application on a server where the public can reach it (or private is so desired)
   3. Exporting your application to desired platform/other platforms
3. Maintenance
   1. Fix any bugs after deployment, and make sure the users are satisfied with their service. Repeat Step 1 if necessary
   2. Make changes based on customer feedback
   3. Refactor code to make it run faster, more effectively, etc.