Software Qualities

* User friendliness

It is essential the web application is user friendly since it is intended to be used by individuals with all skill levels. The website should be intuitive, easy to navigate, and aesthetically pleasing. To achieve user-friendliness, we will use a conventional design to reduce the learning curve with starting the application and a colour scheme that is minimal and visually pleasing.

* Correctness

Correctness is another essential software quality. Users will be trusting our system to manage their groups through uploading files, and creating events and messages. For the application to be successful it will need to be correct. To achieve this software quality our implementation will check user input and do error handling so the system performs correctly. We will also reuse code where available to take advantage of it’s correctness qualities.

* Robustness

Robustness is also required for our web application for the same reason as correctness. Since they are relying on our web application it should be able to handle unexpected scenarios without crashing. In order achieve robustness our application will be implemented to handle unexpected user input and actions through error checking, error handling and setting constraint on user inputs so they do not affect our system. We will also reuse code to take advantage of their robustness qualities.

* Performance – Time Efficiency

Time efficiency is another important software quality for our web application. Users will be using our site to improve their productivity during group projects. To ensure this the application will need to have sufficient performance. We will achieve this through code modularly and limiting redundancy. We will also incorporate googles APIs to improve performance. Lastly, we will limit size of files to upload to ensure performance on the site.

* Problem Definition
  + Brandon Tessier, Tasha Baller, Mykola Kyba, Ryan Kopp, Brenna Curran, Duane Classen
* Use Case Diagrams
  + Website Access
    - Brandon Tessier, Tasha Baller
  + Group/Personal Calendar
    - Brenna Curran, Tasha Baller
  + Group Chat
    - Duane Classen, Tasha Baller
  + Group Features
    - Tasha Baller
  + Admin Leader Features
    - Tasha Baller, Brandon Tessier
  + Personal Profile
    - Mykola Kyba, Tasha Baller
  + File System
    - Ryan Kopp, Tasha Baller
* Use Case Descriptions
  + Upload File
    - Duane Classen, Brandon Tessier, Ryan Kopp
  + Leave Group
    - Brandon Tessier, Tasha Baller, Mykola Kyba, Ryan Kopp, Brenna Curran, Duane Classen
  + Edit Calendar Entry
    - Duane Classen, Brandon Tressier, Ryan Kopp
* Software Qualities
  + Tasha Baller
* Design Specification Document
  + Software Architecture
    - Brandon Tessier, Tasha Baller, Mykola Kyba, Ryan Kopp, Brenna Curran, Duane Classen
* Sequence Diagrams
  + Upload File
    - Duane Classen, Brandon Tressier, Ryan Kopp, Brenna Curran
  + Leave Group
    - Brenna Curran, Tasha Baller
  + Edit Calendar Entry
    - Brenna Curran, Mykola Kyba
* Class Diagram
  + Brandon Tessier, Tasha Baller, Mykola Kyba, Ryan Kopp, Brenna Curran, Duane Classen
* Object Diagrams
  + Mykola Kyba, Ryan Kopp, Brandon Tessier
* Component Diagrams
  + Ryan Kopp, Brandon Tessier
* Deployment Diagram
  + Brandon Tessier
* Code Construction
  + User Interface
    - Tasha Baller
  + Calendar Implementation
    - Ryan Kopp
  + File Implementation
    - Duane Classen
  + Database
    - Brenna Curran
  + Dynamic Website Generation
    - Brandon Tessier, Tasha Baller, Ryan Kopp, Brenna Curran, Duane Classen
  + Website Testing
    - Brandon Tessier, Tasha Baller, Ryan Kopp, Brenna Curran, Duane Classen
* Technical Documentation
  + Brandon Tessier, Ryan Kopp
* User Documentation
  + Graphical User Interface Screenshots
    - Ryan Kopp, Duane Classen, Brandon Tessier
  + Correctness Test Cases
    - Ryan Kopp, Duane Classen, Brandon Tessier
  + Robustness Test Cases
    - Ryan Kopp, Duane Classen, Brandon Tessier
  + Performance Test Cases
    - Ryan Kopp, Duane Classen, Brandon Tessier