**Requirements Analysis**

Based on the results of your requirements elicitation, goals for your project, and course materials, please complete the following tasks:

1. Provide an example of five hypothetical non-functional requirements for your system. Be sure to include the specific type of requirement discussed in class, with each requirement coming from a unique category.

    1. **Usability (human factors, help documentation, etc.)**

        - Choice of colors in application must contrast well enough to ensure all text is readable while maintaining aesthetics.

    2. **Reliability (failure frequency and recoverability):**

        - The system should not fail or have a very low failure frequency. If there is a failure, the app should attempt to create an error log to explain the issue. It should also try to save any of the user's data to be acquired in the next boot up, such as any message drafts they were writing that were not sent.

        - A message in the app should not fail to send, and there should not be a failure in receiving messages.

    3. **Performance (response times, throughput, availability, resource usage, etc.)**

        - The performance of this application should handle a large number of users (from different companies) which would mean that the application should not have a centralized server but a distributed system which would greatly increase throughput. Another solution is to implement multithreading or concurrency (only 1 because both can dramatically increase code complexity and potential bugs)

    4. **Supportability (adaptability, maintainability, internationalization, configurability)**

        - The application should be accessible across a wide range of devices to support collaboration among team members using different platforms, such as Android, iOS, Windows, and macOS.

    5. **Implementation/Constraints (resource limitations, languages, tools, hardware etc.)**

        - Messages sent should not be more than 1000 words. Messages should be limited to text messages, images, and files.

        - The app should support displaying Unicode characters.

        - The application should be cross platform, supporting computers and mobile devices.

2. Provide an example of five hypothetical functional requirements for your system.

    1. Ability to assign priorities to tasks on TODO lists.
    2. Ability to filter by task type, priority, name, date, completion date, etc.
    3. Managers should be able to synchronize by interacting with a list with another employee user. A manager should be able to leave comments on the list.
    4. Messages must have the option to attach images or other files.
    5. The ability to send messages to groups, or individually.

3. Write five formal use cases for your system and provide use case or sequence diagrams to represent each use case.

    1. Use case for assigning priorities to tasks on TODO lists.
        - Precondition: TODO list is populated with at least one task
        - Main flow:
            - User selects a task and selects edit
            - Select the “Prioritize” button
            - Select which level of priority the selected task is
        - Sub flow:
            - The edit button among other options will show when a task is highlighted by clicking on it, this will bring up a window with more details about the task
            - The “Prioritize” button is located somewhere near the bottom of the window
            - A dropdown menu will appear with the different priority levels that the user can select
        - Postconditions:
            - The tasks will be ordered hierarchically within the TODO list. Higher priority tasks will show at the top of the list with lower priority tasks following after it
    2. Use case for Ability to filter by task type, priority, name, date, completion date
        - Precondition: the task list contains at least 1 task, and that task has the required fields (listed above) filled out.

    - Main Flow:
        - User selects a group and clicks the filter button
        - The user selects the field(s) they want to filter by
        - The user applies their filter (using the apply button)
    - Sub flow:
        - The filter button will show somewhere on the utility bar located near the top of the screen
        - The filter button will be greyed out if the task list is empty
            - If the task list contains at least 1 task the filter button will be colored in blue
        - A dropdown menu will appear when the user selects the filter button where they can select the fields to filter by.
    - Postconditions:
        - The tasks shown on the screen will be the ones that the filter applies to. Any task filtered out will not show in this view and will not be deleted (i.e. each user applies a filter only for their view not everyone else's view)

3. Message including an attachment.
    - Precondition: Must have a connection with the recipient or have their ID
    - Main flow: Messaging tab, select recipient or group chat, click add attachment or add image. Open system browser and select image/file to attach to upload. Hitting send, sends the image/file to the recipient
    - Sub Condition: Various buttons must be pressed like attach file or clicking send
    - Post Condition: file is viewable and identifiable by both sender and recipient.
4. Manager must be able to leave a comment on the list
    - Precondition: A list must exist already
    - Main flow: The manager opens the list clicks on an item. Clicks to add a comment. Types up a comment. It is saved and is reflected to the employee
    - Sub Condition: With the list open, an icon will appear next to an item on the list to add a comment. If clicked on, it will create a window to type a comment on the item.

5. Use case for sending messages to groups, or individually.

    - Precondition: The recipients must have a registered account in TaskFlow.

    - Main flow:

        - Navigate to the Messaging tab

        - Select the option to create a new messaging group

        - Choose individuals to include in the group

        - Send a message to the group

    - Sub flow:

  - Locate the "Create New Group" button, positioned near the top of the window
  - Use the search bar to find and add individuals by username to the new messaging group, display a notification if a username is not found
  - If the group already exists, prompt the user to choose whether to add to the existing group or create a new one
- Postconditions:
  - The new messaging group will appear in the Messaging tab under each member's list of groups.

**Requirements Specification**

Based on the results of your requirements elicitation, goals for your project, and course materials, please complete the following tasks.

Write four user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories.

1. As a user of the TODO list messaging app, I want the ability to prioritize tasks, so that I complete the most mission critical tasks first.
   - Acceptance Criteria:
     - **Given –** User has a populated TODO list or is creating a new task
     - **When –** User goes to edit a task or creates a new one
     - **Then –** Tasks are presented in a visual hierarchy on a TODO list
2. As a manager, I want the ability to filter assigned tasks easily to see what is on track and what isn't
   - Acceptance Criteria:
     - **Given**- User has at least 1 task that has been assigned to a team member
     - **When-** User loads up the tasks
     - **And-** the user selects as many fields to filter by as they like (can filter by required and optional fields)
     - **Then-** Tasks that are filtered out are not displayed and the view displays are showing the tasks that are filtered in.
3. As a user, I want to be able to send images or files to others on the app
   - Acceptance criteria:

        - **Given:** user has an image or file to send
        - **When:** the user wants to send a file or image to someone
        - **AND:** they can send a message to that user
        - Then: they attach the image/file to a message and sends it

    4. As a manager, I want to be able to leave comments on a user's list mentioning tasks they should prioritize or specify
        - Acceptance criteria:
            - **Given:** a list that a manager wants to comment on
            - **When:** the list has an item in it to comment on.
            - **Then:** The manager accesses the list, clicks on an item, leaves a comment, and it saves.

2. For each user story mentioned above, estimate the amount of effort needed to complete relevant subtasks using function points. Explain your answer.
    1. Task priority
        - 10 points for creating the "Prioritize" button
        - 30 points for creating the drop-down menu with different levels
        - 10 points for creating specific levels
        - 20 points for sorting the tasks by priority level
        - 30 points for displaying the tasks in hierarchal order in the TODO list
    2. Task prioritization
        - 40 points for a tag system for list items
        - 30 points to create a dynamic sorting system for each and multiple tags
        - 20 points to hide items
    3. Image/file attachment
        - 40 points to add functionality for embedding images
        - 30 points to create an FTP for uploaded files
        - 10 points to delete or remove image/file after being sent
    4. Manager Comments
        - 25 points for a sub UI when clicking on an item on the list

- 30 points to create comment functionality
- 20 points to make sure it works on both manager and client side