0.a Come up with a team name for your group.
Tomorrow's Problem

0.b Please list the names and PIDs of the team members who are present today (or knowingly absent)

- Mollie Mero (molliem16)
- Hannah Solis (hannahs2)
- Shreya Thothathri (shreyathoth)
- Lucas Kopp (lucaszacharykopp)

0.c Provide your preliminary project idea (or set of ideas). This is not a commitment to a project.
Using the approved idea for your group's course project, complete the following activities related to requirements analysis.

- Better project management tool (better Jira)
    - Limit work in progress (number of stories open at once)
    - Simplify the software to limit any wasted time
- **Better sprints**
    - **Continuous learning (break sprints up into work weeks and learning weeks)**
- AI to help make teams more efficient (trained on software processes)
- AI to help train new members on projects (ex. Software processes, project specific details)
- **Customizable Team Planning**

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

- **Usability**: should have a user friendly interface that's easy to navigate and provides clear error messages when needed
- **Reliability**: cannot fail often (more than once a month) and data must be recoverable to help keep teams on track
- **Performance**: must have quick response times to save teams time rather than lose it
- **Supportability**: must be maintainable for quickly changing systems and adaptable for different team types and structures
- **Implementation/Constraint**s: must indicate what software limitations and tools/languages that are available

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

- **User authentication**: Users will have a login and password unique to them
- **Task management**: Users will have the ability to create, assign, track, and delete different tasks in the system
- **Project management**: Users can create different teams and projects and organize them
- **Backups**: backs up data often (every few hours or every day) to make sure teams can keep their data safe from failure
- **Communication/Collaboration**: users must be able to communicate with one another

3. Think of a specific task required to complete each of the functional requirements and non-functional requirements mentioned above (10 total). Estimate the amount of effort needed to complete this task using function points (i.e., using the values here). Briefly explain your answer.

- User Authentication: 3 story points
    - Database to store different users
    - Frontend to allow users to enter their information, access website, and create an account
- Task Management: 2 story points
    - Frontend representation of tasks
    - Database storing tasks, categories for tasks, and handling access
- Project Management: 1 story point
    - Frontend to allow users to edit or create teams and projects
    - Regular updates across user boards to make sure all team members have access
- Backups: 4 story points
    - Frequent backups to save user data
    - Restoration on event of failure of lost data
- Communication/Collaboration: 2 story points
    - Store user chats between each other with set deletion
    - Frontend allowing users to easily switch between chats
- Usability: 4 story points
    - Create a frontend that follows HTML, CSS, and other standards
- Reliability: 4 story points (integrated with backups)
    - Use of backups supports reliability
- Performance: 5 story points
    - Make sure system runs successfully and in ideal amount of time
- Supportability: 3 story points
    - Must work and be maintainable across a variety of systems without significant loss of function
- Implementation/Constraints: 2 story points
    - Highlights project constraints and communicates that to user
    - System will function within the constraints

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

Actor 1: John Doey
John wants to set up his account for his teams management system

Acceptance Criteria:

- John is allowed in the website and his authentication data is saved to database

- John signs up for the application with his google account, is allowed access, and starts to manage his project.

Actor 2: Benny Gonzolas
Benny wants to create a new task in a team management system so he can assign it to team members

Acceptance Criteria:
- The project manager needs to be able to access tasks
- There should be a button to create a new task with task specifications (title, assignee, due date)
- Project manager can save task and assign it to a member

Actor 3: Tina Chompers
Tina wants to mark her tasks in the system as complete so her team knows it's done

Acceptance Criteria:
- Tina needs to be able to access her assigned tasks
- Each task should have an option to track work and mark it as complete
- When completing a task, Tina should be able to leave any comments related to her task

5. Provide two examples of risk that could potentially impact this project. Explain how you would mitigate these risks if you were implementing your project as a software system.
Security: If the system has any security problems, team data may leak, so it is important to minimize any threats and document any security gaps. It is also important to regularly conduct tests that will ensure that data does not leak, any new threats don't arise, and that there is enough process documentation.
Loss of Data: If teams lose their boards, their project would largely be derailed as they wouldn't know which tasks have been completed and which tasks still must be done. To protect against this, we will have frequent backups of their projects and keep them ready to restore on the event of failure. It would also allow for merge conflicts when users update tasks at the same time as they can choose which parts of the backups they wish to keep.
6. Describe which process your team would use for requirements elicitation from clients or customers, and explain why.

We would follow an iterative process (agile) that would allow us to frequently meet with our client to get feedback on our development and allow them to give us further requirements. This will tell us which features to prioritize for our final design and which feature might not be as useful. It will also allow us to personalize features for the client's future use that would benefit their specific needs.