Requirements Workshop (3%)
We will have a requirements workshop in class on February 12. This workshop should be completed and turned in with the remaining materials for PM1.

Come up with a team name for your group.
The Task Titans

Please list the names and PIDs of the team members who are present today (or knowingly absent)
Emily Knoll (emilyk77)
Katelyn Crumpacker (katelync22)
Christina Casey (christinacasey)
Brooke Ritter (britter2)
Matt Lorenzo (mlorenzo7)

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.
Task App: an app that will help keep track of assignments, tasks, and more. Will give users reminders prior to deadlines.

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.
   1.Must use Python to develop app.
   2. The app response time must be less than 1.5 seconds for over 90% of users.
   3. The app needs to work with or without network connection while maintaining all input data.
   4. Text display of the app should automatically configure and adjust to the size of the user's screen.
   5. Navigation within the app should follow design patterns that are easy for the users to follow and ensure usability.
2. Provide an example of five hypothetical functional requirements for this system.
   1. The app must be able to handle multiple tasks with similar deadlines
   2. The app must notify the user at the time the user specified before the deadline
   3. The app must allow the user to create sub tasks that go under bigger tasks
   4. When the user completes a task they can check it off and it will disappear
   5. The app must allow the user to create separate “lists” of different tasks
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. (We are using a 1-5 scale where. 1 is the least difficult, and 5 is the most)
Our function points system goes from 1-5

Task Creation - 2 (this would just involve creating a task in a database and displaying it in the UI)
Task Deletion - 2 (this would just involve deleting a task from the database and reflecting that in the UI)
Task Prioritization - 1 (this requires us to set priorities of task (high, medium, low) in the database and updating this in the UI)
Task Filtering/Grouping - 4 (this is more difficult as this would require us to search the database to sort and filter tasks based on the what the user desires)
UI Design - 2 (we can use other applications that are similar to create the best UI design)
Drag and Drop -2 (drag and drop is common across applications so there are most likely many resources available)
Task Editing - 2 (this would involve ensuring the user can edit the task description, deadline, etc. This is not difficult because it involves making a small editing function)
Ability to add/edit deadlines / due dates -2 (this involves allowing the user to edit the deadline, requiring us to update it in the database)
Reminders -3 (this would involve sending notifications through the system running the app and we would need to do research on that)
Offline mode -5 (this would involve doing research into offline mode and how a database connection would work when offline)

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

   I am a developer for meta using this app to keep track of my tasks. I am very concerned with security because the tasks I have are revealing of the project we are working on, and if that gets leaked, I will lose my job and the project will fail.

   I am a student that would like to divide my tasks into different sub tasks. I would also like to be able to have different folders for my different classes so all the tasks can be viewed separately, but also be able to see them all as a whole.

   I am a developer that is working on a project of which the requirements and tasks are not fully known yet. I need a way to easily delete, edit, and rename my tasks as the project progresses and we find out more about what exactly we need to do.

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.
   1. Data getting wiped - keep copies of the database

2. Data getting accessed by someone who is not the user - implement a password and hashing

6. Describe which process your team would use for requirements elicitation from clients or customers, and explain why.

We would use agile for our software process because we like the idea of breaking tasks into different sprints. We have little features that need to be done, that build on each other and we would like to hold each other accountable with weekly sprint planning meetings. If one person is stuck we can help them or change the task assignees.

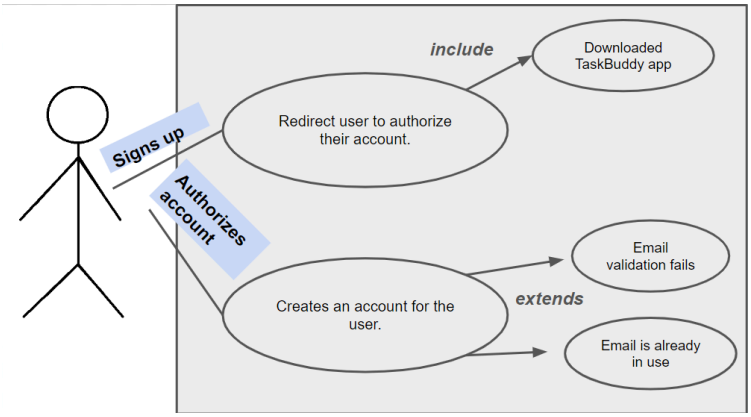
Requirements Analysis (5%)
Requirements analysis is the process of understanding the requirements for a software application. This deliverable will include 5:

Fully dressed use cases
Model (use case or sequence diagrams) for representing each use case

Use Case 1: User Registration

- Precondition
  - User has Task App downloaded on their device
- Main/Subflows
  - User registers an account with Task App by either connecting an account such as Gmail or Outlook
  - User is redirected to authorize account registration
  - Task App successfully creates an account for the user
- Alternative Flows
  - Email validation fails
  - Email already in use

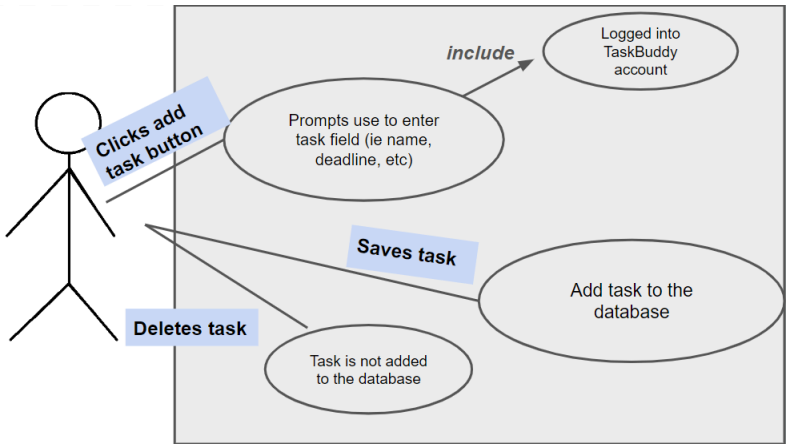
Model 1:



Use Case 2: New Task Creation

- Precondition
    - User is logged into their account
- Main/Subflows
    - User clicks button to add a new task
    - App prompts the user to enter the name of the task and desired deadline
    - Once the user confirms the task, app will add the task to the database which will be reflected in the UI
- Alternative Flows
    - User cancels adding the task prior to confirmation

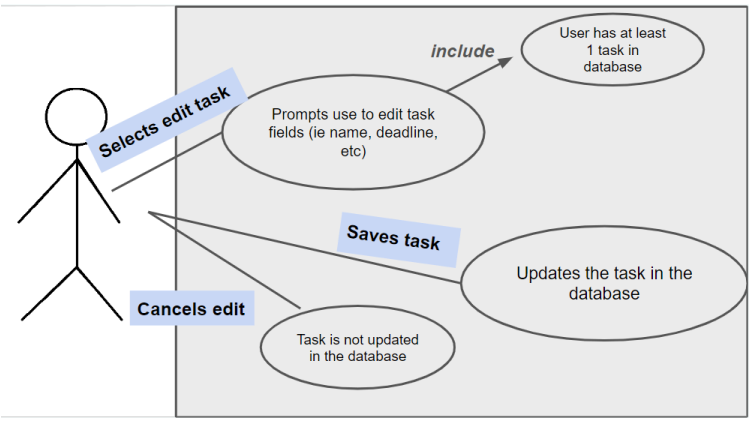
Model 2:



Use Case 3: Task Editing

- Precondition
    - User has at least one task to edit
- Main/Subflows
    - User selects the edit option on a task
    - The user edits the details of the task
    - Once the user confirms the task, the app will update the task in the database and the UI
- Alternative Flows
    - User cancels editing the task before confirming

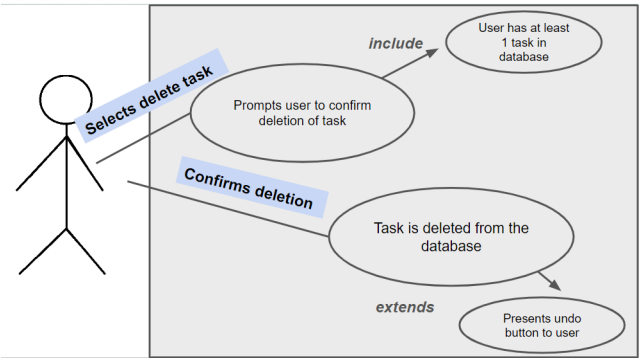
Model 3:



Use Case 4: Task Deletion

- User has at least one task to delete
- Main/Subflows
  - User selects to delete the task
  - App brings a popup to the user asking if they are sure that they wish to delete the task
  - User confirms to delete the task
  - Task is deleted from the database and removed from the UI
- Alternative Flows
  - User deletes a task and wishes to undo the deletion causing the task to be readded to the database and added back to the UI

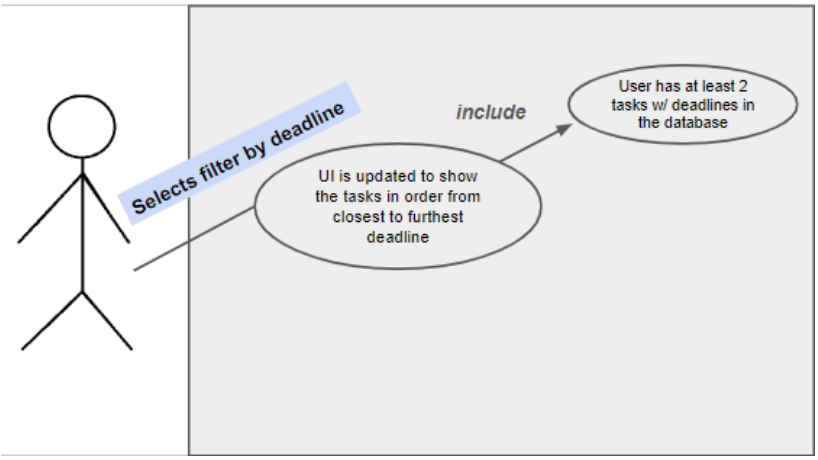
Model 4:



Use Case 5: Filter by Deadline

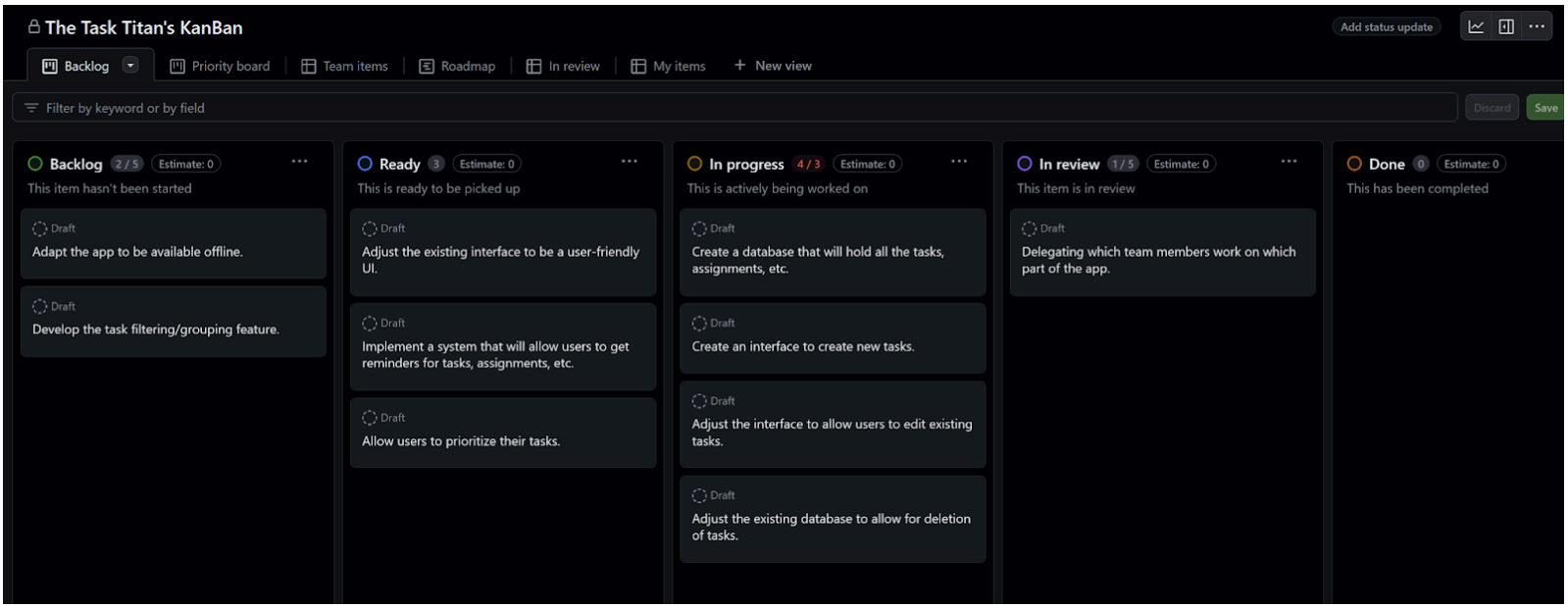
- Precondition
  - User has 2 or more tasks (with specified deadlines) to filter by
- Main/Subflows
  - User clicks the drop down bringing up filtering options
  - User selects the box to filter by deadline
  - Only the UI is updated to show the tasks in order from closest to furthest deadline
- Alternative Flows
  - User decides to filter by task group as well which maintains the deadline filter but also adds the task group filter.

Model 5:



Process Deliverable (2%)
The submission for this deliverable will depend on the specific SE process model your team plans to use to complete the group project (as described in your project proposal). Example submissions for different processes include:

Prototyping: submit a prototype of your system (can be as formal/informal as needed)
Scrum: submit the notes (including each teammate) from your most recent scrum meeting
Kanban: submit a list of prioritized tasks from your task management system (and why they are prioritized)
Waterfall: submit supplementary planning documentation
Extreme programming: submit acceptance test criteria
Spiral: submit risk analysis
Code-and-fix: essay on why you used code-and-fix or up-to-date source code in your GitHub repository For other processes not listed above, the instructor will contact you with the exact submission requirements for this task.



Kanban

Prioritized Tasks

1. Delegate which team members work on which part of the app.
    a. By determining who works on which aspect of the app, we are able to efficiently begin work on the project. Without beginning the project by delegating tasks, everyone would not have a clear understanding of their responsibilities in the project, leading to an inefficient team.
2. Create a database that will hold all the tasks, assignments, etc.
    a. This is the second high priority task because the database is the foundation of the app. This database is where all the tasks and their corresponding information will be held. Without this database, we will not be able to properly test the "add task" feature, and more. Thus, we must develop the database before we can effectively integrate the interface to edit the database.
3. Create an interface to create new tasks.
    a. This is the next feature that we will prioritize developing. One of the most important features of this project is the ability to create tasks that hold all necessary information. Then, you can edit it, delete it, etc. However, none of that can be done if we can not create the task. Thus, we must develop the program to create the task with its information including description, deadline, etc., before we can develop and test the other features for the task.
4. Adjust the interface to allow users to edit existing tasks.
    a. Now that we are able to create tasks in the UI and database, we can move on to developing the program that will allow us to edit information in existing tasks. This includes adjusting the task name, description, deadline, etc. This is a high priority task because in our app, it is essential that users can edit their existing tasks and we want to ensure that this feature is fully implemented and functional prior to launching the app.
5. Adjust the existing database to allow for the deletion of tasks.
    a. After ensuring our UI is user-friendly, we will adjust the database to enable the deletion of tasks. In our app, our users will be able to delete any tasks that they have completed. Otherwise, the tasks would remain on the UI and in the database forever. Therefore, we want to adjust the database to allow users to delete any completed tasks. This is also extremely important for testing purposes since this is an important feature in our app that we want to ensure is fully functional before launching the app.
6. Adjust the existing interface to be a user-friendly interface.
    a. Next, we will create the user-friendly interface for the users to utilize in the app. Creating a user-friendly, accessible interface is extremely important to our app.

Thus, we want to ensure that this remains a high priority task since we want our UI to be accessible by all.

7. Implement a system that will allow users to get reminders for tasks, assignments, etc.
    a. Another feature we would like to implement in our app is a feature that will allow users to get reminders for upcoming deadlines. Although this feature is important, our app will still be fully functional without it upon launching the app. If we are not able to complete this feature upon launching the app, we are able to complete this feature and release a new version of this app that includes this feature. Thus, it is an important task to do but not essential to the app's overall success.
8. Allow users to prioritize tasks.
    a. Next, we would like to prioritize the feature that enables users to prioritize their tasks. Similar to the reminder feature, this is a feature we would like to implement, but not essential for our app to work upon launching.
9. Develop the task filtering/grouping feature.
    a. Another feature we would like to implement is the task filtering/grouping feature. Although this is a feature we would like to implement since it is for user convenience, it is not as important as allowing users to set reminders or prioritize their tasks. Thus, we would prioritize this after completing the above tasks.
10. Adapt the app to be available offline.
    a. This task is the lowest priority of our app. The reason for this is the offline feature is a feature we would like to implement, but it is not essential to the app. This feature would provide convenience for the users, however we would like to implement all the features that assist in the task grouping, prioritization, and reminders first since they are more integral to the goal of the app. Thus, the offline feature is the least prioritized feature since it is for convenience and is not integral to the purpose of the task app.

Due: March 15 at 11:59pm
Requirements Workshop
Requirements Analysis
Process I deliverable