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.**
    a. Usability
        i. Text on screen must be legible with a font size of at least 12 to ensure readability for all users.
    b. Reliability
        i. The TODO Bot should have an average of at least 30 days between failures, indicating that the Bot can run without critical issues for extended periods.
    c. Performance
        i. TODO Bot must be able to handle a minimum of 5000 concurrent users with an average response time of 2 seconds for 90% of user interactions.
    d. Supportability
        i. Must allow easy configuration to changes in task management processes, so that administrators can modify workflow without extensive support from the development team.
    e. Implementation/Constraints
        i. The system shall be compatible with Linux-based servers and should not be tied to specific hardware configurations to ensure deployment flexibility so that every user has access.

2. **Provide an example of five hypothetical functional requirements for this system.**
    a. User can create new tasks on TODO Bot
    b. Project Managers and Team Leads can assign tasks to specific team members, indicating responsibility and accountability.
    c. The system provides real-time updates and notifications to users when there are changes in task assignments or status.
    d. Project Managers can generate detailed reports, such as progress reports and task completion reports, to monitor the status of tasks and overall project progress.
    e. Customer Support representatives can create and manage support tickets for user inquiries and issues.

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.**
    a. Text on screen must be legible with a font size of at least 12 to ensure readability for all users.
        i. 8 function points
        ii. Task involves modifying user interface to ensure text remains legible
    b. The TODO Bot should have an average of at least 30 days between failures, indicating that the Bot can run without critical issues for extended periods.
        i. 12 function points
        ii. Task involves developing error handling mechanisms, and implementing failure detection
    c. Optimize server architecture for handling concurrent user connections
        i. 14 function points
        ii. Task includes load testing, server optimizing, and improving system scalability
    d. Develop intuitive panel for admin to modify workflow and settings
        i. 10 function points
        ii. Task involves designing user-friendly admin panel for configuration and ensuring changes can be made without needing developers
    e. Ensure system is compatible with Linux-based servers
        i. 8 function points
        ii. Task involves platform testing
    f. Design and implement user interface for creating new tasks
        i. 12 function points
        ii. Task involves designing user-friendly interface for task creation
    g. Develop functionality to assign tasks to specific users within the system
        i. 10 function points
        ii. Task includes designing the assignment workflow, and implementing user roles
    h. Implement real-time notification features for users when tasks change status or are reassigned
        i. 8 function points

        ii. Task involves integrating real-time communication technologies
    i. Develop a report generation feature that allows project managers to create progress reports
        i. 10 function points
        ii. Task includes designing report templates, implementing report generation.
    j. Create support ticketing system for customer support reps
        i. 14 function points
        ii. Task involves designing ticket creation form, implementing support ticket management system, ensuring automated confirmations are sent to users.

4. **Write three user stories from the perspective of at least two different actors. Provide the acceptance criteria for these stories.**
    a. As a Developer, I want to create a new task on the TODO Bot so that I can keep track of my coding assignments.
        i. **Given** the user is logged into the TODO Bot,
        ii. **When** the user clicks the "Create Task" button and fills in the task details,
        iii. **And** the user clicks the "Submit" button,
        iv. **Then** a new task is added to the user's task list with the provided information, and the user receives a real-time notification confirming task creation. The task should persist on the task list even after a page refresh.
    b. As a Project Manager, I want to generate progress reports using the TODO Bot to monitor the status of tasks and overall project progress.
        i. **Given** the Project Manager is using the TODO Bot,
        ii. **When** the Project Manager selects the "Generate Progress Report" option and specifies a date range for the report,
        iii. **And** the report generation is completed,
        iv. **Then** the generated report should include a list of tasks with their current statuses (e.g., in progress, completed) and progress percentages. The Project Manager can download the report in PDF format or export it.

c. As a Developer, I want to receive task assignments on the TODO Bot so that I know my responsibilities and can collaborate with my team.
    i. **Given** the user is logged into the TODO Bot,
    ii. **When** a Project Manager assigns a task to the user,
    iii. **And** the user receives a notification with task details,
    iv. **Then** the assigned task appears in the user's task list, indicating the user as the assignee. The user can view all task details. Even after logging out and back in, the assigned task remains associated with the user's account.

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.

Considering the fact that our software system plans to use AI to help users with their planning and strategizing for their own software development, it may be difficult to plan the exact requirements of using and implementing AI into the system. Conversational AI is a relatively new technology, especially one that can be trusted to complete and understand complex tasks and requirements from the users, which means there may be unexpected delays while our development team can fully understand and grasp how to implement a functional and accurate conversational AI.

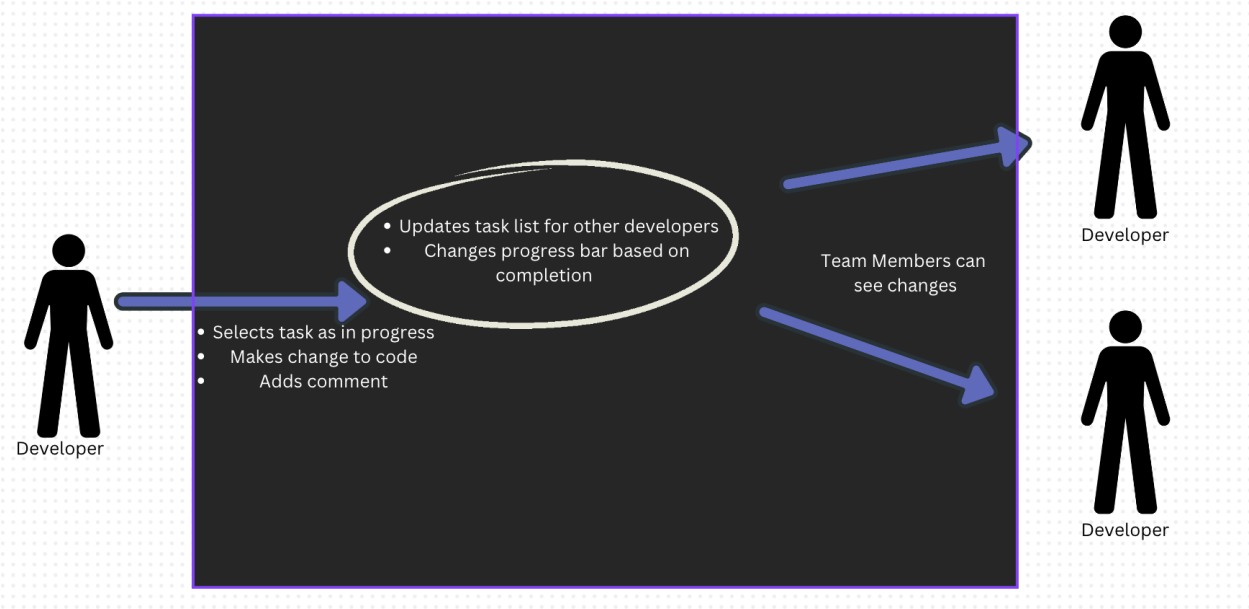
Another risk is the potential for our development process to go over budget. This can be the result of many things, such as the risk previously mentioned, but also developer unavailability, unexpected errors and bugs, and redesigning certain features that are not up to the standard determined by our use cases, user stories, and focus group. Going over budget is one of the biggest challenges any software development team faces, so factoring many of these causes of going over budget will be crucial to ensuring that our system is developed accurately and efficiently.

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

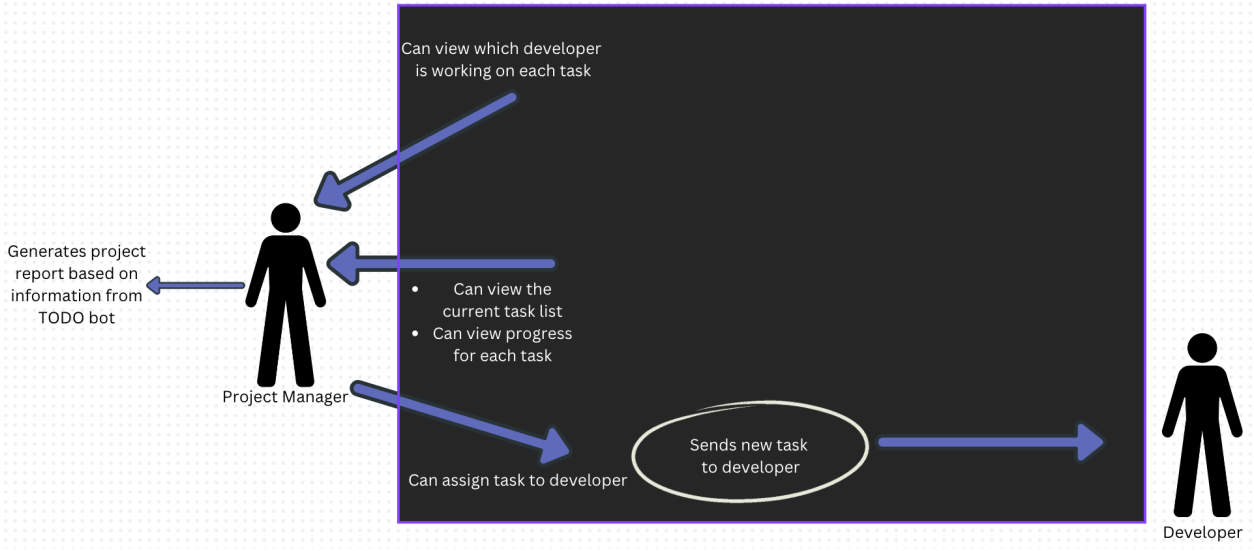
Our team would develop a focus group of other developers, product managers, scrum masters, and other roles that could use our application. With this focus group, we

can implement features that are the most desired based on our focus group. In addition to this, our team would also have feedback surveys in the actual application so that users can tell the developers what they like, what they don't like, and what can be improved. These surveys will be small pop-ups in the bottom right corner of the user's screen and gives the option of providing useful feedback for the development team. Finally, we would also have use cases and stories for the main functions of our application to ensure that the process of using these features is as intuitive and user friendly as possible.

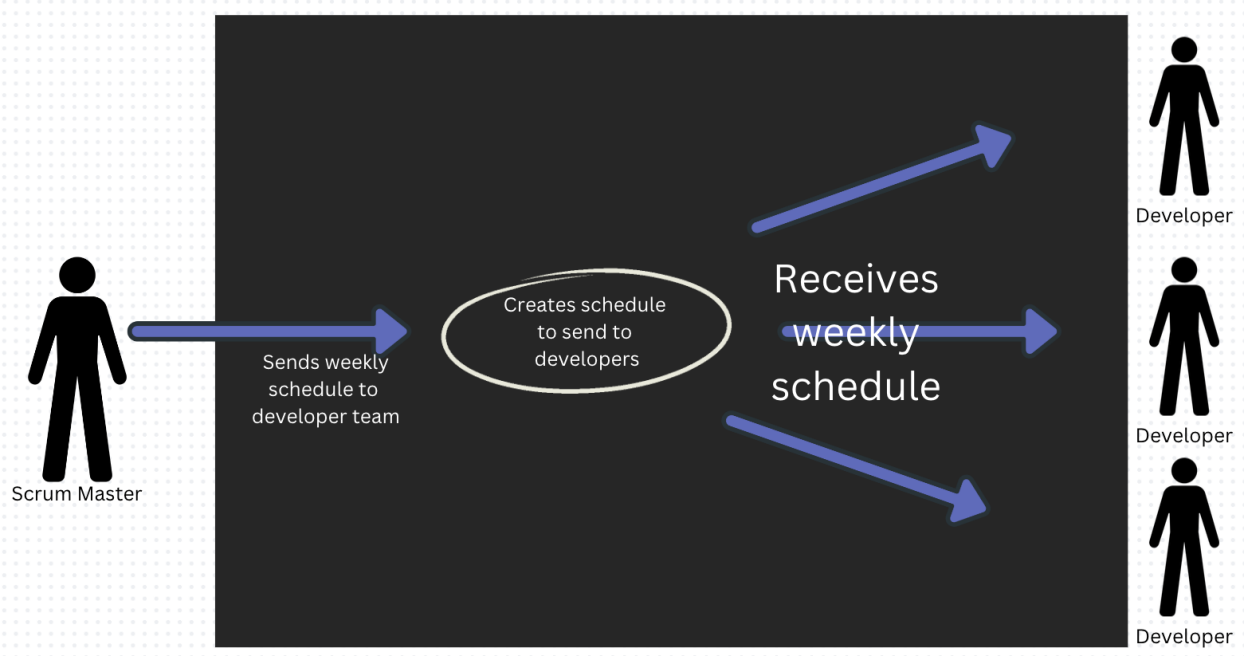
a. Goal - Efficiently manage coding tasks and maintain workflow in TODO Bot
    i. Actors
        1. Developer
        2. Other developers(team members)
    ii. Preconditions
        1. Developer has the TODO Bot application
        2. Developer has coding tasks assigned to them from another user
    iii. Steps
        1. Developer logs into TODO Bot application
        2. Developer views their task list
        3. Developer selects a specific task to complete
        4. Developer updates the status of the task, “In progress”
        5. Developer adds comment, “Almost completed”
    iv. Alternatives
        1. If developer completes project, they update status of task to, “Completed” and then requests for Project Manager approval.
        2. If developer encounters difficulties with task they can click “Request Clarification” to set up a meeting or more details
    v. Postconditions
        1. Developer’s tasks are up-to-date and organized within TODO Bot



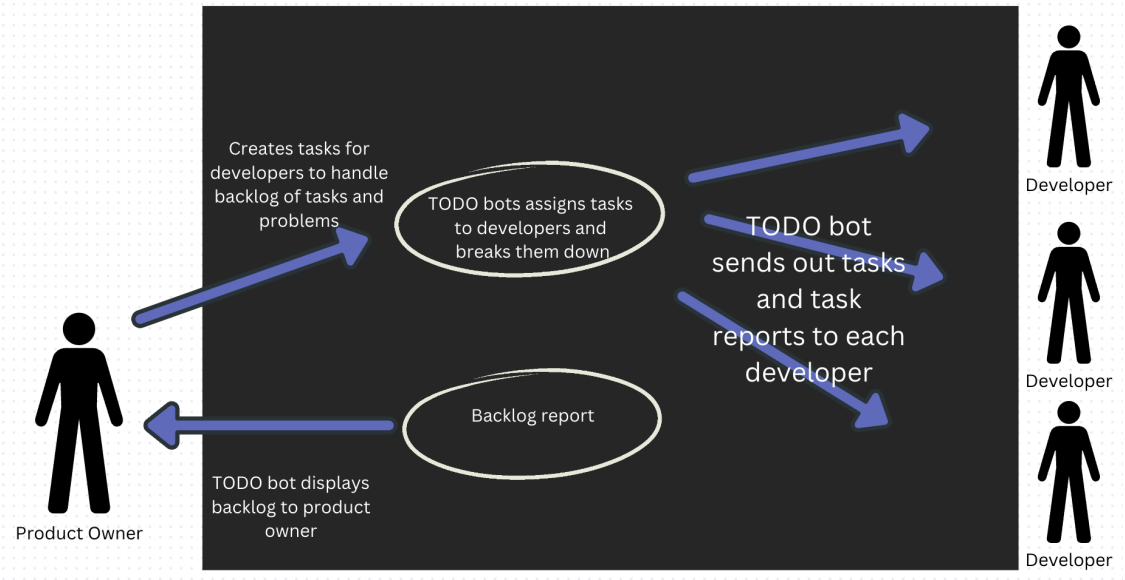
b. Goal - Efficiently oversee and manage project tasks, monitor progress, allocate tasks within TODO Bot
    i. Actors
        1. Project Manager
    ii. Preconditions
        1. User has the TODO Bot application
        2. Project manager has admin controls on TODO Bot application
    iii. Steps
        1. User logs into their account
        2. User views the project task list and the developers' assignments
        3. User creates a new project and adds it to the task list
        4. User assigns new project to a specific developer
        5. User monitors progress of of each developer
        6. User generates report based on project status
    iv. Alternatives
        1. If user encounters issues with progress tracking, they can communicate with customer support
    v. Postconditions
        1. Project tasks are monitored and manages, user has clear overview of each task



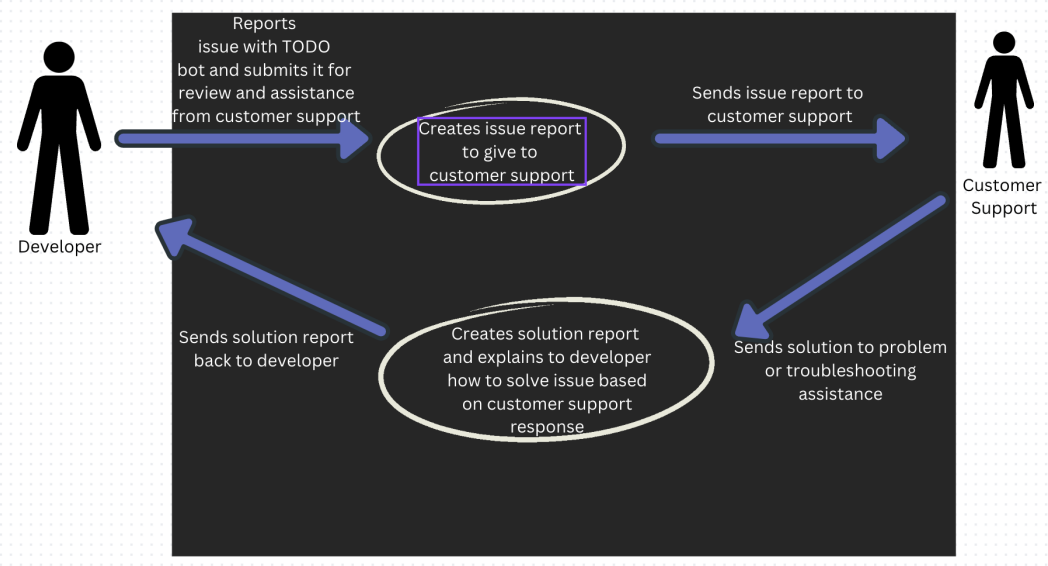
c. Goal - Facilitate Agile development processes, daily stand-ups using TODO Bot. Uses Bot to maintain transparency and communication within teams
    i. Actors
        1. Scrum Master
    ii. Preconditions
        1. User has active account in TODO Bot application
        2. Development team and tasks are defined within TODO Bot
    iii. Steps
        1. User logs into TODO Bot application
        2. User create schedule for the week
        3. User conducts daily stand-up meetings with different teams
        4. User reassigns tasks if developer has too much work, or can provide clarification
    iv. Alternatives
        1. If user has issues creating schedules, they can contact customer support for assistance
    v. Postconditions
        1. Agile processes are facilitated
        2. User maintains communication with every team



d. Goal
    i. Actors
        1. Product Owner
    ii. Preconditions
        1. User has an active account in the TODO Bot.
        2. The product backlog and development team are defined within the TODO Bot.
    iii. Steps
        1. User logs into TODO Bot
        2. User reviews existing items in backlog and new feature requests
        3. User sends messages to stakeholders and team members to gather and document requirements
        4. User adds/updates items in product backlog
        5. User sets up meeting with development team to clarify vision of new items in backlog
    iv. Alternatives
        1. If user encounters challenges, they can communicate with development team
    v. Postconditions
        1. Product backlog is updated and defined. Development team has understanding of the vision of the product.



e. Goal - Efficiently handles user inquiries and issues using TODO Bot
    i. Actors
        1. Customer Support
        2. Developer
    ii. Preconditions
        1. User has active account in TODO Bot
        2. Development team and tasks are defined within TODO Bot
        3. User inquiries and issues are entered in TODO Bot
    iii. Steps
        1. User logs into their TODO Bot account
        2. User reviews list of issues and inquiries
        3. User chooses one of the issues to deal with
        4. User completes task and notifies the individual who put in the issue
        5. User completes the rest of their tasks
    iv. Alternatives
        1. If user encounters challenges, user can contact software developers
    v. Postconditions
        1. Inquiries and issues are all managed and resolved.



SCRUM Meeting Notes

Date: 10/06/2023
Project: TODO Bot Development
Attendees: Mathew Padath, Pranav Poodari, Raghav Pajjur, Prat Chopra

1. Review of Requirements Document:
   - Reviewed the set of functional and non-functional requirements provided.
   - Discussed the estimated effort required for each, using function points.

2. Non-functional Requirements Discussion:
   a) Usability
      i) Text visible from 1m.
      ii) Consider varying screen sizes.

   b) Reliability
      i) Search: 28 hours MTBF.
      ii) Need comprehensive testing.

   c) Performance
      i) Response <1 sec for 90% accesses.
      ii) Consider CDN.

   d) Supportability
      i) Allow network configuration changes.
      ii) Need a user-friendly interface.

   e) Implementation/Constraints
      i) Use Linux and Java.
      ii) Check team proficiency.

3. Functional Requirements Discussion:
   - Walked through each hypothetical functional requirement provided, including task creation, real-time updates, messaging, reporting, and notifications.
   - Shared user stories and their respective acceptance criteria.

- Prioritize user stories for the next sprint.

4. Risks Discussion:
    - Highlighted risks regarding server response time and potential data breaches.
    - Reviewed mitigation strategies for each risk.

5. Requirements Elicitation Process:
    - Discussed the benefits of combining interviews and surveys for gathering client feedback.
    - Set a tentative schedule for the next client interaction.

6. Open Discussion:
    - Mathew: Emphasized the importance of keeping user experience at the forefront.
    - Pranav: Suggested setting up a beta-testing phase for immediate feedback.
    - Raghav: Raised potential integration with other popular developer tools.
    - Prat: Proposed looking into AI-driven suggestions for developers based on their activity.

7. Action Items:
    - Define and prioritize tasks for the next sprint based on the functional requirements discussed.
    - Assign team members to research potential tools for server optimization.
    - Schedule client interviews and draft survey questions for feedback.

Meeting Adjourned: 45 minutes