Sprint Planning Checklist

Preparation	Meeting	Follow Up
☒ Have ideas about high-level design, low-level design, and design sketch	☒ Talk about the different ideas formulated for PM3 ☒ Delegate tasks between team members	☒ Review of the different design deliverables

Sprint Team Members 11/15/2024

Name	Role
Andrew Tang	Project Manager
Daniel Abaye	Developer
Dylan Lau	Developer
Mazin Abdelrahman	Developer

Agenda

Review and close previous sprint.

1. Review the requirements elicitation, analysis, and specification deliverables.

2. Review scope of the project.

Previous Sprint Summary

Sprint theme	Requirements
Story points	40
Summary	Successfully completed the requirements elicitation, analysis, and specification. However, lost some points for failing to follow rubric closely.

Details

Start date	11/16/2024
End date	11/22/2024
Sprint theme	Design deliverables

Velocity Tracking

Completing each task should take a shorter amount of time compared to the previous assignment concerning requirements. This assignment is less intensive on writing and more diagraming. The following tasks: High-level design, low-level design, and design sketch require more diagraming, which could be more tedious. We estimate each should take between 1 to 2 hours to complete. The design sketch task might be closer to 2 hours because it requires a diagram and a short paragraph to complete. With this said either spreading out the tasks across multiple days or completing it in one day should suffice as long as they are completed before the deadline.

Capacity Planning

	Current Sprint	Previous Sprint
Total days	7	5 days
Team Capacity	100%	50%
Projected Capacity	70 points (10 for sprint meeting / 60 for design deliverables – 20 points each)	100 points (40 for analysis / 60 for requirements)
Individual Capacity	Andrew – 20 Daniel – 26 Dylan – 26 Mazin – 26	Andrew – 30 Daniel – 30 Mazin – 20 Dylan - 20

Potential Risks

Risk	Mitigation
Thanksgiving break is coming up	Getting the assignment done early will help avoid getting lazy before break.
Projects and midterms	During this week there are typically projects due and midterms. Plan completing tasks mentioned previously around those commitments.

High-level Design (4%)

The system will be structured using a **Client-Server Architecture**, where the client-side will consist of a web application built with technologies like React and JavaScript, responsible for rendering the user interface and handling user interactions, such as creating tickets or sending messages. The client will communicate with the server through RESTful APIs to perform operations like fetching ticket data, updating statuses, and sending or receiving messages. On the server side, the application will be built using key components such as a **Ticket Management Service** (handling ticket creation, status updates, and assignment), a **Messaging Service** (managing real-time communication between developers and project managers, using WebSockets), and a **User Management Service** (responsible for authentication, authorization, and user roles). The backend will connect to a relational database (e.g., PostgreSQL) for ticket data storage and a NoSQL database (e.g., MongoDB) for storing chat messages, ensuring scalability and efficient data access. The server will also handle session management and provide APIs for handling notifications, using services like **Push Notification Service** for real-time alerts. Each component of the server will interact with the database and other services in a modular fashion, ensuring clear separation of concerns and easy maintenance.

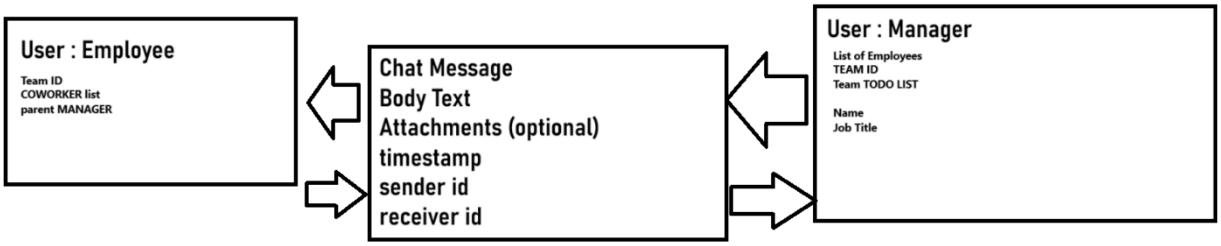
Low-level Design (4%)

The Low Level Design for the Messaging Service of our project will utilize Creation/Builder Design. This is because the communication between the user and a manager can best be visualized as a singleton design, as it is two devices/users communicating with each other. In messaging, there will be a reused/commonly built class of a Message that can contain different types of data.

Psuedocode:

```
Class User{
List Friends;
Chat userchat;
}
Class Message {
        String text;
        File attachedFile;
        ID fromUser
        ID sendUser
}
```

Informal diagram:



User : Employee

Team ID
COWORKER list
parent MANAGER

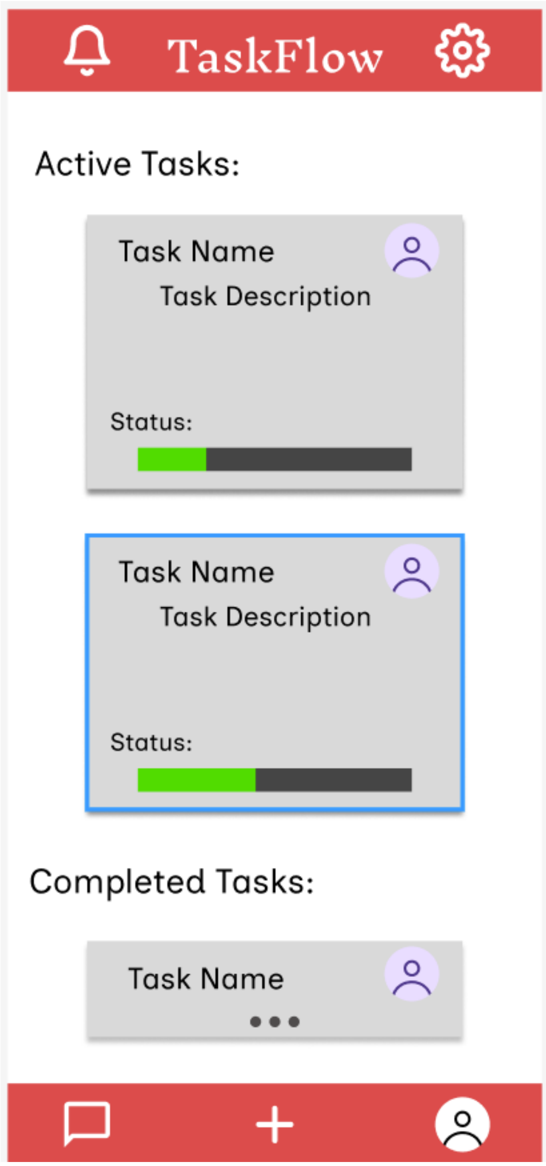
Chat Message
Body Text
Attachments (optional)
timestamp
sender id
receiver id

User : Manager

List of Employees
TEAM ID
Team TODO LIST

Name
Job Title

Design Sketch (4%)



For our TaskFlow project, we created a wireframe mockup of the app’s home screen to visualize the app’s user interface, focusing on its functionality and user experience. The design organizes the interface into clearly defined sections, including "Active Tasks" and "Completed Tasks," to support task prioritization and reduce cognitive load. Interactive task cards were included to display essential details like task names, descriptions, progress bars, and assigned users, ensuring clarity and ease of use. The top navigation bar includes recognizable icons, such as a bell for notifications and a gear for settings, while the bottom navigation bar provides consistent access to key features like chat, task creation, and profile settings. These design decisions follow the principles of visual hierarchy, consistency, and user-centered design, ensuring the app is intuitive, organized, and functional for its intended audience.