

Product Design Document: Discord Accountability Bot

1. Product Overview

A Discord bot designed for a small, private server of friends to enforce mutual accountability. The bot tracks daily, weekly, and one-off tasks, gamifies completion with a point system, and features both automated and user-triggered "nagging" mechanics for overdue items.

To minimize friction, task management is handled via one-click interactive buttons in private user channels, while accountability and celebrations are broadcasted in public channels.




2. Channel Architecture

The bot operates across three distinct types of channels:

- `#accountability-board` **(Public, Read-Only)**: A locked channel containing a single, persistent Discord Embed. The bot continuously edits this message to reflect the real-time status of all users, their active tasks, and their current point scores.
- `#the-meat-grinder` **(Public, Interactive)**: The main social hub (can be any general chat). The bot posts celebratory messages here when tasks are completed, drops the daily "Wall of Shame" for overdue tasks, and listens for the `/prod` command.
- `#username-tasks` **(Private, 1-on-1)**: A dedicated, private channel created by the bot for each individual user. This acts as the user's personal inbox. The bot posts new tasks here with interactive UI buttons for management.

3. Core Features & Mechanics

A. Task Lifecycle & Management (Low Friction)

- **Creation:** Users create tasks using a slash command (e.g., `/task add [description] [recurrence]`).
- **Delivery:** The bot posts an embed in the user's private `#username-tasks` channel containing the task details and interactive Discord UI buttons: [ Mark Done] and [ Snooze (1 Day)] .
- **Completion Flow:** When a user clicks [ Mark Done] :
 1. The bot edits the message in the private channel to show it crossed out/completed, removing the interactive buttons.
 2. If the task is recurring, the bot automatically generates a *new* task message for the next cycle (e.g., next day/week) and drops it in the private channel.

3. The bot fires a celebration message into `#the-meat-grinder` (e.g., "🎉 Amiel just crushed: *Spin up Docker container*").
4. The bot updates the `#accountability-board`.

B. Gamification System

- **Earning Points:** Completing a task grants points (e.g., +10 points).
- **Losing Points:** Overdue tasks incur a penalty (e.g., -5 points for every day past the deadline). Snoozing a task incurs a minor penalty (e.g., -2 points) to discourage procrastination.
- **Leaderboard:** The `#accountability-board` ranks users based on their current score.

C. The Nagging Engine (Accountability Enforcement)

- **Manual Shaming:** A user can type `/prod @user` in `#the-meat-grinder`. The bot checks if the tagged user has overdue tasks. If yes, the bot tags them with a randomized, slightly judgmental call-out message listing their overdue items.
- **Automated Wall of Shame:** A scheduled cron job (e.g., 9:00 PM local time daily) scans the database. If any users have tasks due that day that are still pending, the bot tags them in `#the-meat-grinder` with a warning before the midnight point deduction.

4. Expected Bot Commands (Slash Commands)

- `/opt-in` or `/init` – Registers the user in the database, sets their score to 0, and spins up their private `#username-tasks` channel.
- `/task add [description] [due_date] [recurrence (none/daily/weekly)]` – Adds a new task.
- `/prod [user]` – Triggers a public reminder for a user's overdue tasks.
- `/board refresh` (Admin) – Forces a manual redraw of the `#accountability-board` in case the message is deleted or corrupted.

5. Styling and Tone Guidelines

The bot should utilize Discord Embeds and markdown formatting extensively to keep information clean and scannable.

- **The Tone of Voice:** Playfully judgmental, slightly snarky, but encouraging. Think "drill sergeant who is secretly your best friend."
- **The Accountability Board Embed:** * Color: Neutral/Dark Blue (`#2B2D31` to blend with Discord dark mode).
 - Format: Grouped by User. Use bolding for Usernames and Scores (🏆 **Score: X**).

- Status Emojis: 🟢 [DONE] (clears at midnight), 🟡 [PENDING] , 🔴 [OVERDUE] .
- **Private Inbox Task Embeds:**
 - Default Color: Blue/Gray (#5865F2).
 - On Complete: Edit embed color to Green (#57F287), strike through the text (~~task name~~), remove buttons.
 - On Overdue: Edit embed color to Red (#ED4245), add an "Overdue" prefix.
- **Public Notifications:**
 - Use Blockquotes (>) for the bot's dialogue to make it stand out in the chat feed.
 - Pings should explicitly mention the user (e.g., <@userid>) to trigger notifications, especially for the Wall of Shame.

6. Target Environment Constraints

- **Hosting:** The bot will be containerized using Docker and hosted on an Intel Synology NAS.
- **State Management:** The bot must use a lightweight, local database (SQLite) mapped to a Docker volume so data persists across container rebuilds and restarts.