

TrackMe PRD

Новгородский Андрей

Цель

создание удобного, масштабируемого трекера с AI ассистентом как для локальных команд, так и для личного использования. AI ассистент должен помогать структурировать и фильтровать текст задачи с точки зрения грамматики, обценной лексики и общей структуры.

Для кого?

1. Маленькие коллективы которым удобен менеджмент в режиме телеграмма
2. Менеджеры команд побольше которые хотят без усилий получить грамотный текст с расставленными знаками препинания в выбранном стиле или в каком либо шаблоне
3. Люди которые хотят выстроить менеджмент себя самих, чтобы не держать все в голове

Почему делаю?

Область менеджмента времени мне интересна, всегда испытывал трудности с этим, которые учился преодолевать самыми различными способами.

Кажется интересным использовать ИИ ассистента в оптимизации составления задач, это удобно и ново

Структура

- Данные о пользователях и задачах хранятся в нескольких БД.
- Телеграм являет собой средство взаимодействия конечного пользователя с API, которое будет делать CRUD в БД.
- Для ИИ ассистента будет использоваться либо GPT-like API либо самописный сервис с хорошей моделью с huggingface

Глоссарий

1. Задача - объект состоящий из текстового описания задачи и ее различных свойств
2. BetterTask - название для ИИ ассистента, который помогает составлять задачи
3. Приоритет - поле задачи связанное с текущим приоритетом
4. Исполнитель - тот кому назначена задача. Пользователь получит ее когда запросит список своих задач с фильтром TODO
5. Заказчик - автор задачи
6. Комментарий - объект который можно закрепить за задачей - содержит в себе текст
7. Статус - выполнена/отложена/удалена/назначена
8. Тип задачи - произвольное строковое представление класса задачи
9. Параметры задачи - срок выполнения, приоритет, тип задачи, заказчик, исполнитель, частота уведомления
10. id задачи - идентификатор задачи
11. Id пользователя - идентификатор пользователя
12. Better flags - bool флаги для апи ИИ ассистента (grammar, punctuation, cens, и т.д.)

Виды пользователей

По задумке все пользователи обладают одинаковыми правами - могут создавать задачи и назначать их. Младший сотрудник не должен быть лишен возможности назначить задачу старшему. При возникновении каких либо вопросов задачу можно всегда удалить

Интерфейс работы в телеграме

Пользователю будет доступен пул команд

1. create_task {optional parameters} создает черновую задачу с возможными изначальными параметрами и возвращает ее id
2. post_task {id} - публикует задачу id
3. drop_task {id} - удаляет задачу

4. comment {text, id, notify: bool} - оставляет комментарий к задаче id с временной меткой и в зависимости от notify уведомляет заказчика или исполнителя
5. set {parameter_name:value, id} - приравнивает значение parameter_name задачи id к value
6. better_task {better_flags: [flag1...], id} - выполняет better_task над задачей id с флагами better_flags
7. resolve {id} - переводит задачу id в статус решенной
8. set_solver {id, uid} - делает пользователя uid исполнителем задачи id
9. get_tasks {filters} - возвращает пользователю список задач, зависимый от filters (возможно только задачи пользователя как исполнителя, возможно только как заказчика и тд)

Информация в БД

В базе данных будут схемы подобного рода:

Задачи

1. task_id : ui64
2. creator_uid : ui64
3. worker_uid : ui64
4. text : string
5. parameters : json-like object
6. timestamp : timestamp
7. task_status : value from enum of statuses (string likely)

Комментарии

1. task_id : ui64
2. text : string
3. parametest : json-like object
4. creator_id : ui64

Пользователи

1. user_id : ui64
2. role : text

Планы на будущее

Интеграция с Jira

Развитие ИИ ассистента - улучшение качества

Внедрение различного распространенного в других трекерах функционала, активная обратная связь с пользователями