Journaling with GTasks

Introduction

Overview: This technical design document describes the design and implementation of a tool that uses Google Tasks and Google Spreadsheet to facilitate reflective journaling and metacognition. The tool allows users to easily create journal entries using any Google Tasks app, including the native Google Tasks panel in Google Workspace. The entries are then imported into a Google Spreadsheet, where they are automatically labeled and summarized using OpenAI's GPT-3.5 Turbo. The tool also provides the ability to generate insightful reports about the user's journaling habits and the content of their entries.

Purpose: The main purpose of this tool is to help users grow and learn through reflective writing. By making it easy to create journal entries and providing insights into their content, the tool encourages users to engage in regular self-reflection and metacognition.

Intended Audience: The intended audience for this tool is anyone who is interested in using reflective writing as a tool for personal growth and learning.

Requirements

Functional Requirements

Frictionless and Intuitive Journal Entries: The tool should provide an easy and intuitive way for users to create journal entries using any Google Tasks app, including the native Google Tasks panel in Google Workspace. This will encourage users to engage in regular journaling about anything that comes to mind.

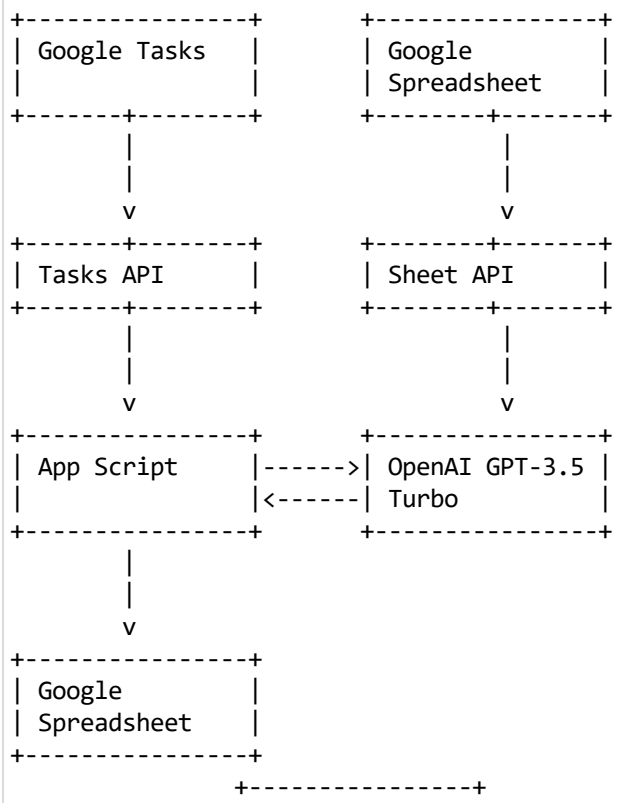
Automatic Labeling and Summarizing of Entries: The tool should automatically label and summarize journal entries as they are imported into a Google Spreadsheet. This will make it easier for users to review and analyze their entries at a later time.

Non-Functional Requirements

Usability of Data for Insights: The data collected by the tool should be easily usable for generating insights into the user's journaling habits and the content of their entries. This will help users make the most of the tool and achieve their personal growth goals.

Architecture

High-Level Architecture: The tool is built on top of Google Tasks and Google Spreadsheet, using App Script to interact with these services. The tool fetches tasks from a specified Google Tasks list using the Tasks API, and imports them into a Google Spreadsheet using the Sheet API. The imported tasks are then sent to OpenAI's GPT-3.5 Turbo for analysis, and the resulting labels and summaries are added to the spreadsheet.



Component Interaction: The main components of the tool interact with each other as follows:
The App Script fetches tasks from a specified Google Tasks list using the Tasks API.
The App Script imports the fetched tasks into a Google Spreadsheet using the Sheet API.
The App Script sends the imported tasks to OpenAI's GPT-3.5 Turbo for analysis.
OpenAI's GPT-3.5 Turbo returns labels and summaries for the imported tasks.
The App Script adds the returned labels and summaries to the Google Spreadsheet.

Design

Data Structures: The tool uses a Google Spreadsheet to store the imported tasks and their associated labels and summaries. Each row in the spreadsheet represents a single task, with columns for the task ID, creation date, modified date, parent task name, parent task description, title, and description. Additional columns are used to store the labels and summaries generated by OpenAI's GPT-3.5 Turbo.

Algorithms: The tool uses several algorithms to fetch tasks from a specified Google Tasks list, import them into a Google Spreadsheet, send them to OpenAI's GPT-3.5 Turbo for analysis, and add the resulting labels and summaries to the spreadsheet. These algorithms are implemented using App Script and make use of the Tasks API and Sheet API to interact with Google Tasks and Google Spreadsheet.

Interfaces: The tool provides a user interface in the form of a Google Spreadsheet, where users can view the imported tasks and their associated labels and summaries.

Implementation

Technologies Used: The tool is implemented using a combination of technologies, including Google Tasks, Google Spreadsheet, App Script, the Sheet API, the Tasks API, and OpenAI's GPT-3.5 Turbo. These technologies are used to fetch tasks from a specified Google Tasks list, import them into a Google Spreadsheet, send them to OpenAI's GPT-3.5 Turbo for analysis, and add the resulting labels and summaries to the spreadsheet.

Development Environment: The tool is developed using the App Script editor in Google Drive and included as part of the Extensions in Google Sheet. This editor provides a web-based development environment for creating and editing App Script projects. The editor includes features such as syntax highlighting, code completion, and debugging tools to help developers write and test their code.

Testing

Test Plan: The tool is tested to ensure that it meets its functional and non-functional requirements. The test plan includes a set of test cases that cover the main features and capabilities of the tool, as well as its performance and reliability.

Test Cases: The test cases for the tool include scenarios such as creating journal entries using a Google Tasks app, importing tasks into a Google Spreadsheet, automatically labeling and summarizing entries.

Test Results: The test results for the tool are documented to provide evidence that the tool meets its requirements. The results include information about the test cases that were executed, their outcomes, and any issues or defects that were identified during testing.

Deployment

Deployment Environment: The tool is deployed in a Google Workspace environment, where it can be accessed by users with the appropriate permissions. The tool requires a Google Tasks list and a Google Spreadsheet to store its data, as well as an OpenAI API key to access OpenAI's GPT-3.5 Turbo.

Deployment Process: The deployment process for the tool involves making a copy of a read-only Google Spreadsheet that contains the App Script code for the tool. Users can then run the tool from the script editor of their copy of the spreadsheet or trigger it automatically using time-based triggers.

Maintenance and Support: The tool is provided as-is, without any ongoing maintenance or support from its creator. Users are responsible for fixing any issues or defects that they encounter, as well as adding new features and capabilities to the tool as needed.

Conclusion

Summary: This technical design document provides a detailed description of a tool that uses Google Tasks and Google Spreadsheet to facilitate reflective journaling and metacognition. The tool allows users to easily create journal entries using any Google Tasks app, and provides insights into their content through automatic labeling and summarizing, as well as the generation of insightful reports. The tool is implemented using a combination of technologies, including App Script, the Sheet API, the Tasks API, and OpenAI's GPT-3.5 Turbo, and is deployed in a Google Workspace environment.

Future Work: Possible future work on the tool could include adding new features and capabilities to improve its usability and effectiveness. For example, the tool could be extended to support reports or to provide more advanced analysis of the user's journaling habits and the content of their entries. The tool could also be integrated with other services or tools to provide additional functionality or to improve its ease of use.