

Software Development

Konzept iOS App

Die App ist eine Scanner App, um Dokumente zu scannen. Die App bietet eine direkte Anbindung an den persönlichen Google Drive der User. Nach dem Scannen des Dokuments wird ein KI Agenten Netzwerk getriggert. Die Agenten erhalten das gescannte Dokument in PDF Form und erledigen die Sortierung. Der Inhalt der PDF Datei wird von einem Agenten gelesen, um das Dokument eigenständig zu benennen. Mit der Information, wie die Datei heißt, erstellt ein weiterer Agent einen Ordner, der zu der PDF Datei passt, oder sortiert die Datei in einen entsprechenden vorhandenen Ordner.

Die Idee der App soll sein, dass die User von diesem Vorgang nichts mitbekommen. Die Agenten sollen den Usern die Arbeit erleichtern, sich um die Organisation und Verwaltung von Dokumenten zu kümmern. Um diese Erleichterung so anwendbar wie möglich zu machen, soll die Scanner App hauptsächlich über Shortcuts, wie z.B. als Icon im Kontrollzentrum oder Homescreen Widget, genutzt werden. Die eigentliche App braucht nach der Einrichtung nicht aktiv benutzt werden. Damit bietet die App eine nahtlose Schnittstelle an das Google Ökosystem und wird von Usern nicht als zusätzliche App wahrgenommen.

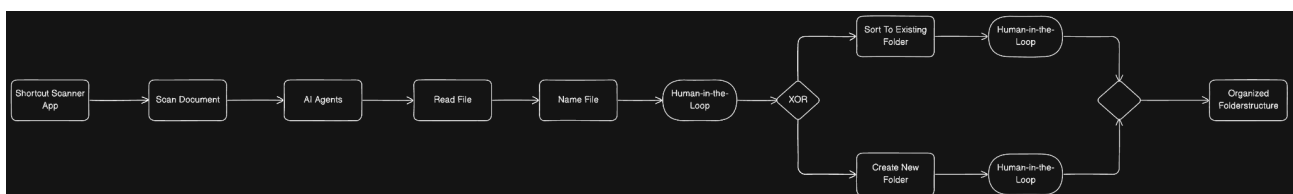
Um das Vertrauen in die KI Agenten zu stärken und die Organisation zu validieren, wird ein Human-in-the-Loop Prozess verwendet, welcher optional aktiviert oder deaktiviert werden kann. Die Scanner App benötigt kein eigenes User Management. Es wird kein Profil für User angelegt, da für die App nur eine Verknüpfung zum Google Konto benötigt wird. Dies erhöht die Nutzerfreundlichkeit, da keine zusätzliche Registrierung notwendig ist.

User müssen einen eigenen API Key für die Nutzung der KI Agenten anlegen.

Die Skalierbarkeit dieser App ist im Hinzufügen von weiteren Funktionen gegeben. Ideen für weitere Funktionen neben einer reinen Scanner App sind:

- Chatten mit Dokumenten
- Google Workspace Erweiterungen z.B. Gmail
- Zusätzliche Funktionen der Scans und Verarbeitung deren Inhalt
- Sprachfunktionen
- Weitere File Storages z.B. OneDrive, Dropbox, Lokal
- Weitere KI Modell Schnittstellen z.B. Anthropic, Gemini
- Lokale KI Modelle nutzen

UML



Agiles Vorgehen

Die iOS Scanner App wird iterativ unter Verwendung des Scrum Frameworks entwickelt. Da das Projekt alleine entwickelt wird, wird auf eine aufwändige Software für das Anwenden von Scrum verzichtet. Stattdessen wird die Software Todoist verwendet.

Mit der Board Ansicht werden die Spalten "Backlog", "Selected For Development", "In Progress", "Waiting For Feedback" und "Done" erstellt. Im Backlog werden User Stories gesammelt, um die Kernanforderungen darzustellen. Für jeden Sprint werden User Stories aus dem Backlog in Selected For Development verschoben, um den Start des Sprints zu demonstrieren. In diesem Sprint werden lediglich die Aufgaben aus Selected For Development bearbeitet. Wird eine Aufgabe bearbeitet, wird sie in In Progress verschoben. In den Kommentaren der Aufgaben wird der Fortschritt dokumentiert. Ist eine Aufgabe abgearbeitet, wird sie in Waiting For Feedback verschoben. Da hier nicht auf das Feedback eines Projektleiters gewartet werden kann, wird das Feedback mit Tests simuliert, die die Fertigstellung der jeweiligen Aufgabe belegen. Werden alle Tests erfolgreich bestanden, wird die Aufgabe in Done verschoben. Sind alle Aufgaben des Sprint in Done verschoben, findet ein Backlog Grooming statt, um die dort liegenden User Stories an den aktuellen Entwicklungsstand anzupassen und deren Wichtigkeit zu überprüfen. Des Weiteren wird überprüft, ob während des Sprints weitere Aufgaben entdeckt wurden, die dem Backlog hinzugefügt werden. Anschließend werden die nächsten Aufgaben in Selected For Development für den nächsten Sprint verschoben.

Entwicklung

Für die Entwicklung der Scanner App wird zunächst das Agent Netzwerk entwickelt. Hierfür wird das Agent SDK von OpenAI in Python verwendet. Die Agenten Struktur für das Lesen und Verstehen einer PDF Datei wird der Ausgangspunkt für die weitere Entwicklung. Anschließend wird ein Agent für das Umbenennen der Datei und einer für das Verwalten von Ordnern entwickelt. Weiterhin müssen Funktionen geschrieben werden, die die Agenten für die Bearbeitung ihrer Aufgaben nutzen können. Hierfür wird die Google Drive API benötigt, um alle Anpassungen vornehmen zu können. Eine weitere Funktion für das Benennen der PDF Datei muss ebenfalls implementiert werden, damit der entsprechende Agent die Datei korrekt benennen kann. Sobald das Backend für die Verarbeitung der PDF Datei entwickelt und getestet wurde, wird das Frontend der iOS Anwendung in Swift entwickelt. Dabei soll ein minimalistisches Erscheinungsbild gewahrt werden, da die Anwendung nicht primär genutzt werden soll. Wichtig sind hier die Screens: Google Anmeldung, Einstellungen (vor allem API Key) und Scan. Wenn möglich, sollte beim erstmaligem Öffnen der Einstellungsscreen geöffnet werden, damit die User den API Key hinzufügen können. Beim nächsten Öffnen der App, sollte sich immer direkt der Scan öffnen, damit User, die keine Shortcuts nutzen, ebenfalls direkt zum Kern der Anwendung gelangen. Die Versionierung der iOS Scanner App erfolgt über GitHub.

Anforderungen

- Als User möchte ich meine Dokumente scannen.
- Als User möchte ich, dass meine gescannten Dokumente als PDF gespeichert werden.
- Als User möchte ich, dass meine gescannten Dokumente, dem Inhalt entsprechend, automatisch umbenannt werden.
- Als User möchte ich, dass meine gescannten Dokumente automatisch organisiert werden.
- Als User möchte ich, dass automatisch Ordner erstellt werden, wenn sie benötigt werden.
- Als User möchte ich, dass Scout die PDF Dateien automatisch in die richtigen Ordner verschiebt.
- Als User möchte ich die Möglichkeit haben einzuschreiten, wenn ich mit der Benennung oder Verwaltung von Scout nicht zufrieden bin.