

Projektantrag

ReceiptManager

Projekt Name	ReceiptManager
Projekt Leiter	J. Richtsfeld
Dokumenten Status	In Arbeit
Version	V. 1.0

Revisions

Date	Author	Change
16.09.2019	J.R./G.R./M.H./M.K.	Erste version
20.09.2019	J.R./G.R./M.H./M.K.	Projektziele überarbeitet
23.09.2019	J.R./G.R./M.H./M.K.	Rechtliches und Rahmenbedingungen
27.09.2019	J.R./G.R./M.H./M.K.	Projektantrag Korrektur und Finalisierung

Inhaltsverzeichnis

1	Einleitung	3
2	Ausgangssituation	4
3	Konkurrenz	5
4	Allgemeine Bedingungen und Einschränkung	6
5	Projektziele und Systemkonzepte	7
6	Chancen und Risiken	8
7	Planung	9

1 Einleitung

Unser Projekt ist eine elektronische Belegsverwaltung, genannt ReceiptManager. Die Idee fuer dieses Projekt kommt von Gregor Rechberger. Diese Software sollte faehig sein aus einem Foto von einem Beleg die wichtigsten Daten zu erfassen, um sie schließlich elektronisch zu speichern.

2 Ausgangssituation

Aktuell müssen Klienten einer Steuerberatungskanzlei die Belege noch wöchentlich bzw. monatlich zur Kanzlei bringen, damit die Buchhaltung erledigt werden kann. Danach müssen die Belege in Ordner abgelegt werden, diese Ordner werden dann in Wandschränke gebracht und alphabetisch und nach Jahr sortiert. Aktuell werden die Belegordner nach 5 Jahren aus den Schränken entfernt und in einen Lagerraum gebracht, dort sind sie wie in den Wandschränken sortiert und müssen noch weitere 2 Jahre aufgehoben werden. Durch die Buchhalter werden die essentiellen Werte (Rechnungsbetrag, Rechnungstyp, Steuersatz, usw.) der aktuellen Belege in die Buchhaltungssoftware übertragen, damit dann die Buchungen durchgeführt werden können und die Buchhaltung erledigt werden kann.

3 Konkurrenz

Scopevisio und Parashift bieten Komplettlösungen von der Buchhaltung bis zur automatischen Rechnungsverwaltung an. Die Aktien Gesellschaft Parashift wirbt stark mit deren Rechnungserkennungsalgorithmus, der sehr Treffsicher sein sollte. Sie bieten diesen Service als eine API oder verbunden mit ihrer Software an. Diese Software liest komplett ohne manuelle Eingaben alle wichtigen Elemente aus einer Rechnung oder einem Beleg, doch die Informationen können natürlich darauffolgend bearbeitet werden, wenn es nötig ist. Die Scopevisio AG bietet eine Komplettlösung an und setzt auch auf eine KI, doch verlässt sich nach der Erkennung auf den Steuerberater, der diese Daten überprüfen muss. Dies hat den Nachteil, dass es zu Lasten von dem Steuerberater fällt, jedoch sorgt dies für eine fast 100 Prozentige Trefferquote. Wir wollen für Buchhaltungssysteme, welche diese Rechnungsverwaltung noch nicht anbieten als Erweiterung dienen.

4 Allgemeine Bedingungen und Einschränkung

Unser System muss mit folgenden Problemen umgehen können:

- Die Rechnungen müssen mindestens 7 Jahre aufgehoben werden, wobei es keine Beschränkung auf die Form der Speicherung gibt.
- Jeder Nutzer muss sich authentifizieren, damit die sensiblen Daten der Rechnungen geschützt werden können.
- Der Beleg muss vom Kunden lediglich fotografiert oder gescannt werden, welcher anschließend in der kundenspezifischen Inbox des Steuerberaters erscheint. Bereits beim Hochladen wird eine OCR eingesetzt um die Rechnungsdaten herauszufinden. Die Resultate der OCR in Kombination mit dem Text-Parser werden in der Inbox als Standardwerte angezeigt und müssen bei Bestätigung vom Buchhalter korrigiert werden, um die Belege in das System aufzunehmen.
- Es wird sowohl ein Web- als auch Mobileclient benötigt.
- Es soll dem Steuerberater möglich sein, schnell Belege und die dazu relevanten Informationen zu finden.

5 Projektziele und Systemkonzepte

Die Projektziele werden wie folgt zusammen gefasst:

- Die Klienten laden ihre Belege hoch, anschließend erfolgt die Datenextraktion durch die OCR. Schließlich muss der Buchhalter die Daten nach Korrektheit kontrollieren, damit die Rechnungen abgespeichert werden können.
- Die Rechnungen werden nach Klienten gruppiert, innerhalb der Klienten werden die Rechnungen nach Jahr gruppiert. Innerhalb eines Jahres können die Rechnungen nach Datum und Betrag sortiert werden. Dadurch können die Belege vom Buchhalter/Steuerberater schnell gefunden werden.
- Die Datenbank muss täglich gesichert werden, damit im Falle eines Systemausfalls oder -schadens keine Daten verloren gehen.
- Das Backend soll auf zwei Teile aufgeteilt werden. Zum einen das OCR- System als Microservice und zum anderen das Hauptbackend zum Speichern und Verwalten der Belege.
- Beim Frontend wird für die Klienten eine Wepapp bzw Mobileapp angeboten. Der Steuerberater sieht eine Admin Oberfläche ebenfalls realisiert als Webapp.

6 Chancen und Risiken

Unsere Software hat die folgenden Vorteile:

- Der Klient/Steuerberater kann sich Zeit und daher Geld sparen.
- Die Verwaltung der Belege wird uebersichtlicher.
- Der Steuerberater spart sich Platz, da die Belege nur noch digital gelagert werden.

Folgende Risiken muessen beachtet werden:

- Verlust von Belegen, die sensible Informationen enthalten.
- Durch falsch eingegebene Informationen koennten inkonsistente Datenbestaende entstehen.

7 Planung

Teammitglieder:

- Teamleader: Julian Richtsfeld
- Frontenddeveloper: Maximilian Kaindl
- Mobiledeveloper: Mathias Hofmarcher
- Backenddeveloper: Gregor Rechberger, Julian Richtsfeld

Meilensteine:

- Klienten müssen sich Autorisieren um Belege hochladen zu können. Das selbe gilt für den Steuerberater um die Belege zu verwalten. - Fälligkeitsdatum 24.12
- Verwaltung der Rechnungen durch den Steuerberater - Fälligkeitsdatum Semesterende
- Design des Web -und Mobileclients - Fälligkeitsdatum April
- Einbindung einer OCR Software zur automatischen Texterkennung - Fälligkeitsdatum Juli

Benoetigte Ressourcen:

- Server
- OCR-Software Lizenz

Das Projekt beginnt mit 20.09.2019 und endet mit dem Ende des Schuljahres 2019/20. Der erste Prototyp wird Mitte November zur Verfuegung stehen. Die Implementierung beginnt mit Abschluss des Prototyps. Die größeren Arbeitspakete sind die Entwicklung des Front -und Backends und die Einbindung einer OCR-Software. Unsere Ziele schätzen wir durchaus realistisch ein und gehen von einer fristgerechten Umsetzung aus.