

Projektgruppe P10



FakturaAutomata

Inhaltsverzeichnis

Nutzen und Kurzbeschreibung	3
Arbeitsteilung und Teamrollen	3
Aktueller Projektstand/Prototyp	4
Geplante Tools, Frameworks etc.	12
Grobe Architektur und Zeitplan	13
Mögliche Herausforderungen und Risiken	14
Links zu Online-Ressourcen	14

Nutzen und Kurzbeschreibung

FakturaAutomata ist eine Anwendung zur automatisierten Zusammenstellung von Rechnungen im ZUGFeRD-Format in einer simplen Übersicht, die es dem Nutzer ermöglicht, trotz der mitunter im Verlauf eines Projektes auftretenden großen Anzahl dieser, stets einen detaillierten und dennoch leicht zu überblickenden Bericht über dessen Kosten, sowie eventuell auftretende Fehler bei der Preiskalkulation einzelner Posten innerhalb dieses Projektes zu erhalten. Hierzu wird dem Nutzer ein Webinterface bereitgestellt, welches ihm einen Überblick über seine laufenden Projekte, sowie alle bereits eingelesenen Rechnungen dieser ermöglicht. Weiterhin lassen sich hier neue Rechnungen im ZUGFeRD-Format einlesen, deren Daten daraufhin persistent im Datenbestand der Anwendung gespeichert werden. Dieser kann vom Nutzer bei Bedarf nachträglich bearbeitet und in das Tabellenkalkulationsprogramm Microsoft Excel überführt werden, wobei eine tabellarische Übersicht der zum Projekt gehörigen Rechnungen erzeugt wird, welche eine detaillierte Auflistung der Kosten des ausgewählten Projektes beinhaltet, sowie den Nutzer auf eventuell vorliegende Ungereimtheiten bzw. Differenzen zwischen vereinbarten und tatsächlichen Preisen aufmerksam macht und es ihm so ermöglicht, diese aufzuklären und unnötige Kosten einzusparen. Neben der Möglichkeit zur Einsparung, die aufgrund von fehlerhaften Preisangaben entstehen, bietet FakturaAutomata durch die Zuteilung von Rechnungen zu Projekten auch eine simple Möglichkeit zur zentralen Organisation des unternehmensinternen Rechnungswesens. Weiterhin spart der Nutzer wertvolle Arbeitsstunden, die sonst zur manuellen Eingabe der einzelnen Rechnungen und Rechnungsposten in eine etwaige Übersicht notwendig wären und minimiert das Risiko von menschlichen Übertragungsfehlern.

Arbeitsteilung und Teamrollen

*Tabelle 1: Ansicht der Aufteilung von Arbeitsbereichen und Teamrollen für jedes Teammitglied.
Legende: Hauptverantwortlich - dunkelblau; Mitverantwortlich - hellblau*

	Orga	Backend	Frontend	UI	Algorithmus (Texterkennung)	Product Owner
Adele			X	X		
Isabelle			X	X		
Tim	X	X			X	X
Leon		X			X	
Sebastian	X	X			X	

Aktueller Projektstand/Prototyp

Zum aktuellen Stand des Projektes lässt sich sagen, dass die Analyse- und Planungsphase erfolgreich und planmäßig abgeschlossen werden konnte. Es fanden bereits erste Gespräche mit dem Kunden statt, in denen die Anforderungen an die Anwendung, sowie deren Details genauer festgelegt werden konnten. Nachdem nun auch der Prototyp erfolgreich umgesetzt werden konnte und wir uns somit auch eine erste Vision vom Design und Aufbau der Anwendung schaffen konnten, kann nun mit der Arbeit an der programmatischen Umsetzung von FakturaAutomata begonnen werden.

Link zum clickbaren Figma-Prototypen:

https://www.figma.com/proto/kjCXLtKgiHnKmXK11WirGo/Projekt_Prototyp_FakturaAutomata?page-id=0%3A1&node-id=1%3A61&viewport=314%2C-169%2C0.7962720990180969&scaling=min-zoom

Login Screen:

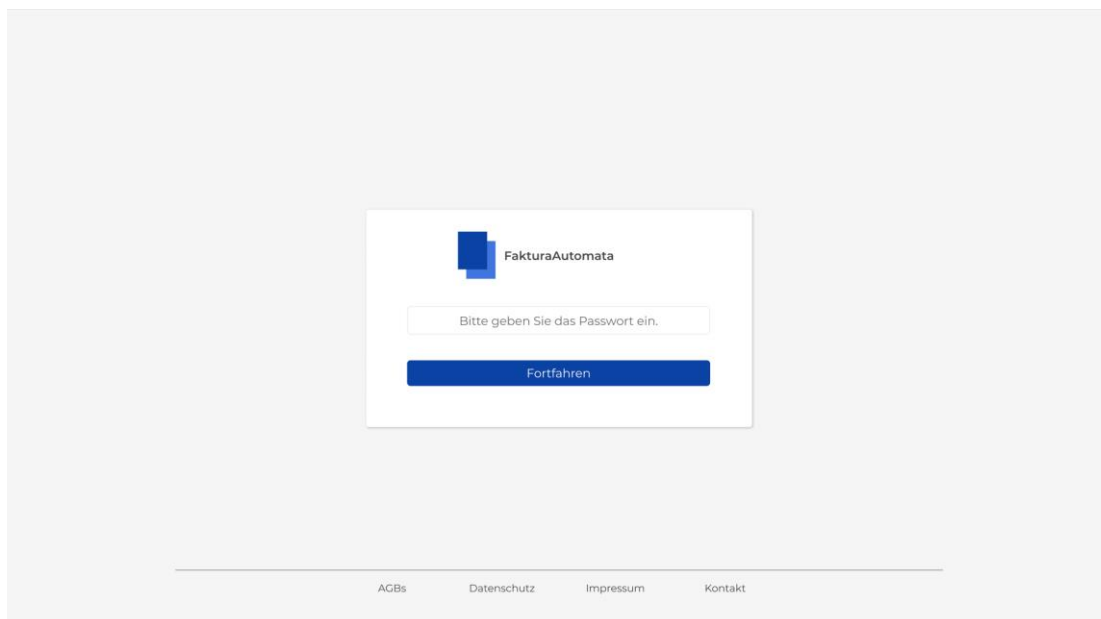


Abbildung 1: Login Dialog

Bevor der Nutzer Zugriff auf die Anwendung und ihre Funktionen erhält, wird er mithilfe eines Login Dialogs dazu aufgefordert, sich durch die Eingabe eines Passwortes zu authentifizieren, wodurch die in der Anwendung hinterlegten sensiblen Rechnungsdaten vor unbefugten Zugriffen geschützt werden sollen.

Startseite/Projektübersicht:

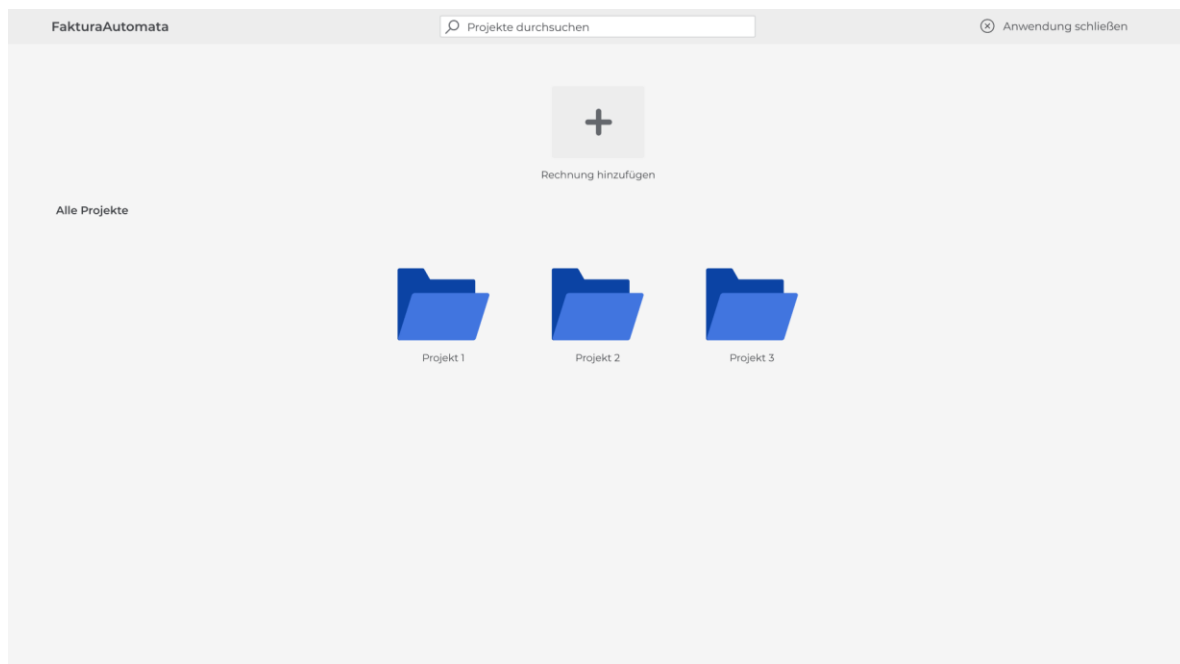


Abbildung 2: Startseite der Anwendung/Projektübersicht

Nach erfolgreicher Authentifizierung wird der Nutzer auf die eigentliche Startseite von FakturaAutomata weitergeleitet und kann die Anwendung in vollem Umfang nutzen. Auf dieser angekommen, findet er eine Übersicht aller bisher angelegten Projekte, kann diese einsehen und hat die Möglichkeit, neue Rechnungen in den Datenbestand der Anwendung einzulesen. Hierbei sei anzumerken, dass die Projektordner nicht vom Nutzer erstellt werden, sondern automatisch von der Anwendung generiert werden, falls eine Rechnung mit einer noch nicht vorhandenen Projekt-ID eingelesen wird. Zusätzlich kann auf diesem, wie auch auf jedem anderen Screen nach Projekten/Rechnungen gesucht und die Anwendung beendet werden.

Detailansicht eines Projektes:

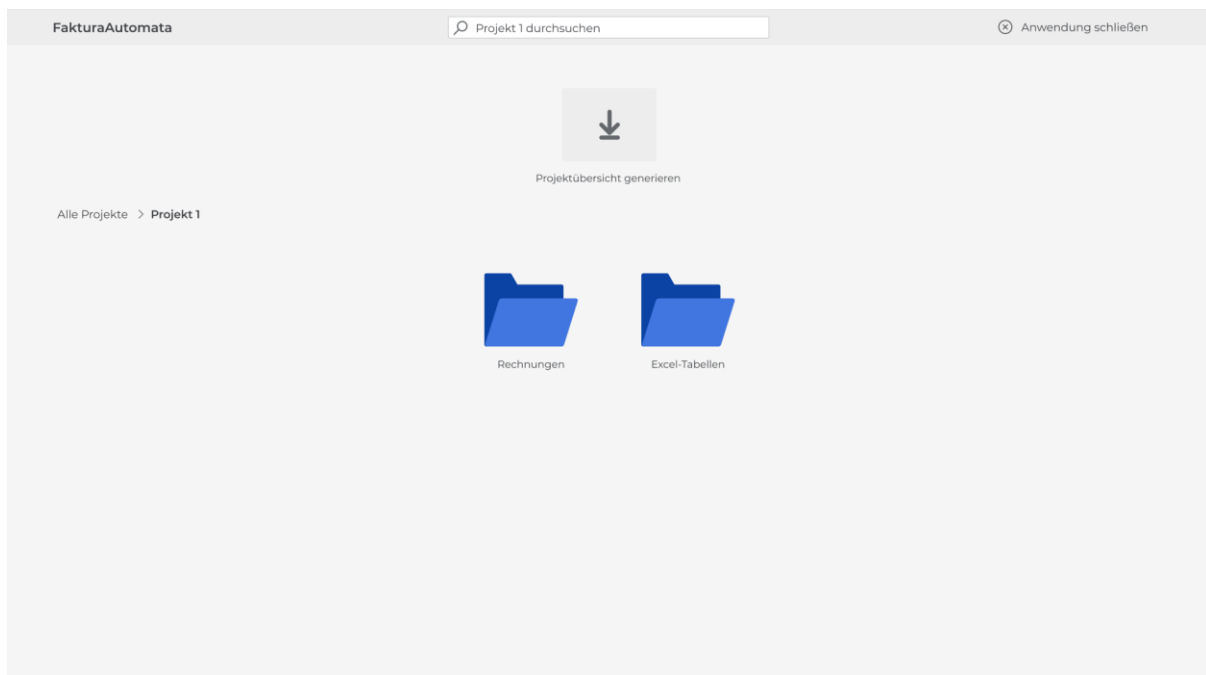


Abbildung 3: Detaillierte Ansicht eines Projektes

Klickt der Nutzer auf einen der auf der Startseite sichtbaren Projektordner, wird die Detailansicht dieses Projektes geöffnet, in welcher er die Möglichkeit hat, alle bereits eingeleseenen und zum Projekt zugehörigen Rechnungen, sowie alle bisherigen Versionen der aus ihnen generierten Excel-Übersicht, einzusehen. Weiterhin lässt sich in dieser Ansicht auch der aktuelle Datenbestand in das Tabellenkalkulationsprogramm Microsoft Excel überführen und somit eine neue Projektübersicht erzeugen.

Excelordneransicht eines Projektes:

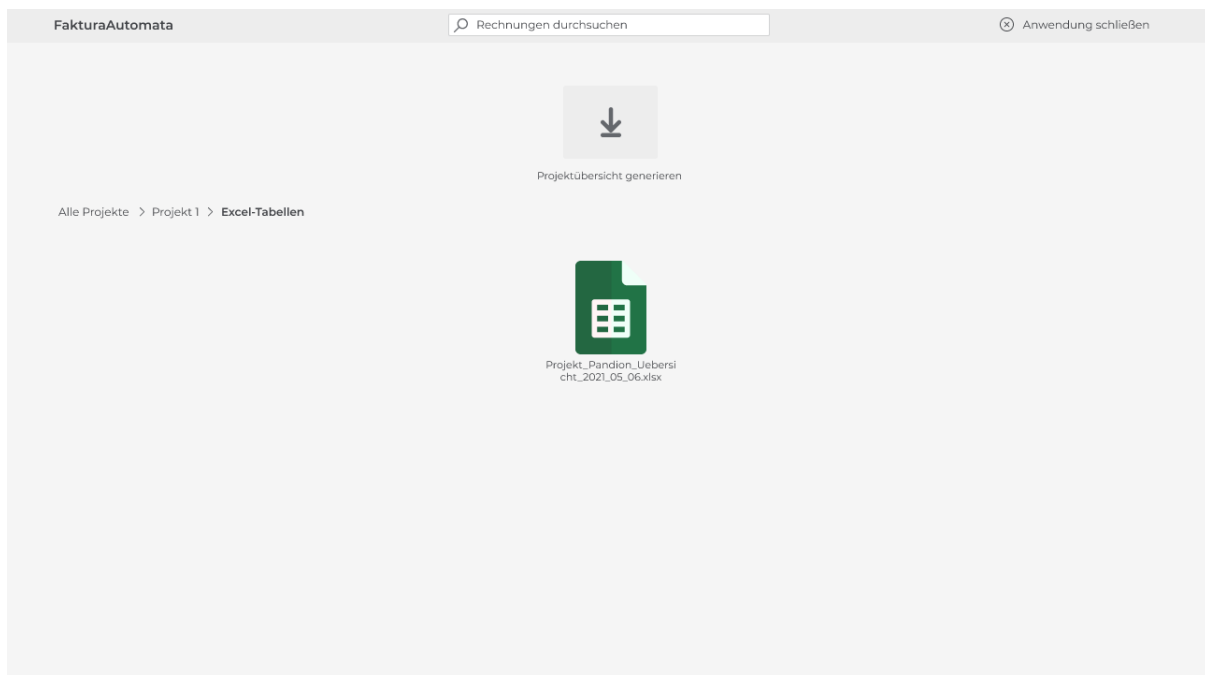


Abbildung 4: Listenansicht der generierten Excel-Tabellen eines Projektes

Das Öffnen des Excel-Tabellen-Ordners führt den Nutzer in eine Ansicht innerhalb des Projektes, in welcher er die Möglichkeit hat, alle bereits generierten Excel-Dateien zu sehen und zu downloaden. Auch in dieser Ansicht lässt sich aus dem Projekt mit dem aktuellen Datenbestand eine Übersicht.

Ansicht einer aus einem Projekt generierten Excel-Tabelle:

Projekt-ID	Projekt	Rechnungssteller	Rechnungs Nr.	Datum	Artikel Nr.	Artikel	Menge	Stückpreis	Rabatt	Stückpreis mit Rabatt	Bestpreis	Angebot	Stückpreis lt. Zwischenrechnung	Summe lt. Rechnung	MwSt Rechnung	Summe lt. Angebot	Differenz
1	Pandion	Bär & Ollenbroth	5209278	10.01.2020	648015	Verschussstopfen	14	3,6	50,00%	1,8	1,80	1,50	25,20	0,19	29,99	24,99	-5,00
3	1	Pandion	5209278	10.01.2020	648015	Verschussstopfen	14	3,8	46,80%	2,0	1,80		53,20	0,19	63,31	0,00	
4	1	Pandion	5209278	10.01.2020	648015	Verschussstopfen	14	3,7	46,80%	2,0	1,80		51,80	0,19	61,64	0,00	
5	1	Pandion	5209279	11.01.2020	648016	Verschussstopfen	14	3,7	46,80%	3,7	3,70	3,70	51,80	0,19	61,64	61,64	0,00

Abbildung 5: Generierte Excel-Tabelle mit Übersicht aller Rechnungsdaten des Projektes und farblich gekennzeichneten Werten

Die Excel-Tabelle beinhaltet eine Auflistung der Artikel aus allen Rechnungen eines Projektes mit den Daten, die zur Überprüfung von möglichen Abweichungen zwischen Rechnungspreisen und Angebots- bzw. bisherigen Bestpreisen notwendig sind.

Rechnungsordneransicht eines Projektes:

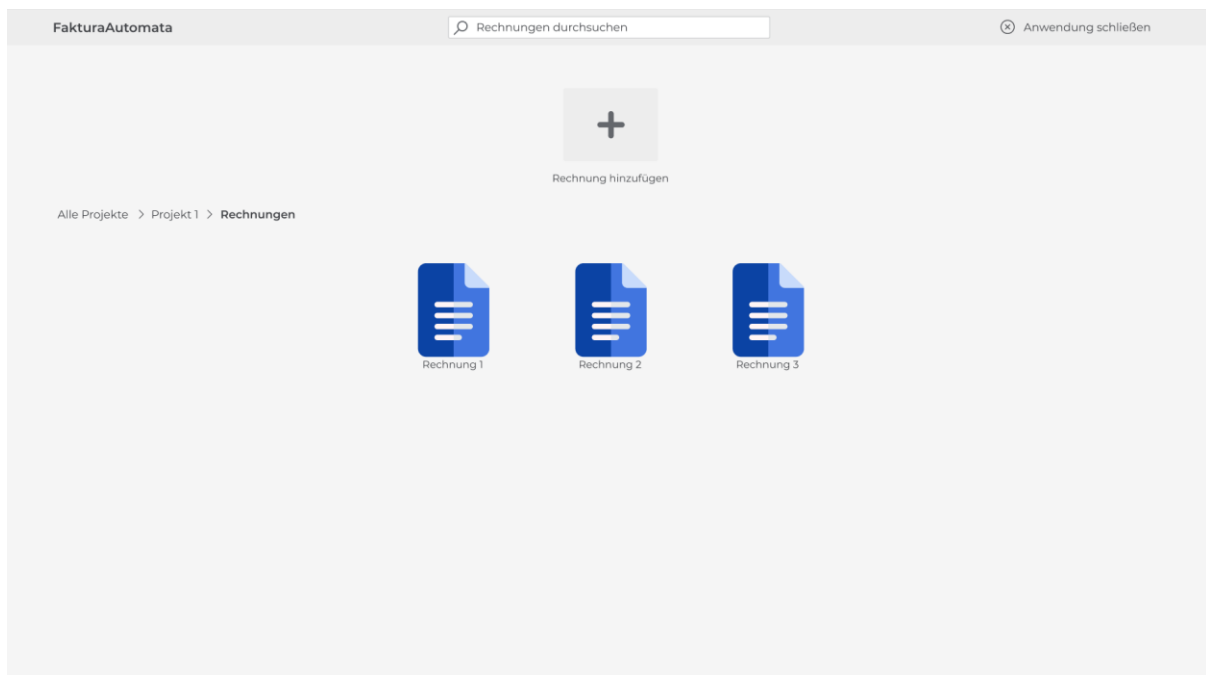


Abbildung 6: Listenansicht der Rechnungen eines Projektes

Wie beim Ordner der Excel-Tabellen, führt der Rechnungsordner in eine Detailansicht innerhalb des Projektes, in welcher er, wie bereits angesprochen, die Möglichkeit hat, alle bereits eingelesenen Rechnungen des Projektes zu sehen. Auch in dieser Ansicht lassen sich neue Rechnungen einlesen.

Dialog zum Einlesen einer neuen Rechnung:

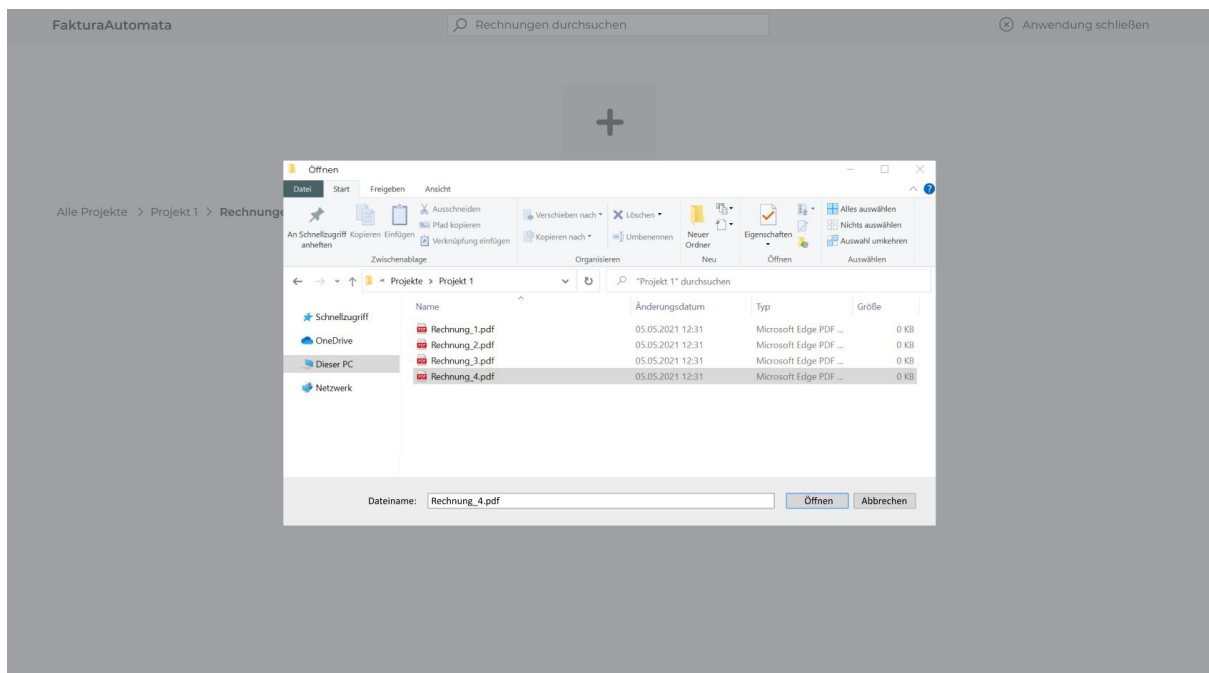


Abbildung 7: Dialogfenster zur Auswahl einer einzulesenden Rechnung

Soll nun eine neue Rechnung eingelesen werden und der dafür vorgesehene Button betätigt, so wird in der Anwendung ein Dialogfenster des Dateisystems (hier gezeigt am Beispiel des Windows File Explorers) geöffnet, welcher den Nutzer dazu auffordert, die einzulesende Rechnung auszuwählen, sowie die Auswahl zu bestätigen. Der Prozess des Einlesens wird anschließend automatisch gestartet.

Bearbeitungsansicht einer eingelesenen Rechnung:

FakturaAutomata

Anwendung schließen

Rechnung bestätigen

Alle Projekte > Projekt 1 > Rechnungen > Rechnung 4

Projekt ID	Rechnungssteller	Rechnungsnummer	Rechnungsdatum	Einlesedatum
319521	Bär & Ollenbroth	105	10.01.2021	10.01.2021

Artikelnummer	Stückzahl	Stückkosten in €	Kosten gesamt in €	Angebotspreis Stück in €	Angebotspreis gesamt in €
51359	3	3	9		
1251	2	6	12		
13241	52	10	520		

Abbildung 8: Ansicht zum Bearbeiten einer erfolgreich eingelesenen Rechnung

Nach erfolgreichem Abschluss des Einlesevorgangs wird der Nutzer auf diese Ansicht weitergeleitet. Hier findet er eine Übersicht aller für die Excel-Übersicht wichtigen Daten, wobei diejenigen, die bereits aus der Rechnung eingelesen werden konnten, automatisch in die entsprechenden Feldern eingefügt werden. Alle Felder in dieser Ansicht sind vollständig editierbar, was dem Nutzer die Möglichkeit bietet, die Eingaben vor der Speicherung im Datenbestand nochmals zu überprüfen und ggf. zu ergänzen. Sind alle Eingaben in Ordnung, kann die Rechnung durch Betätigen des dafür vorgesehenen Buttons in den Datenbestand eingespeist werden.

Speicheraufforderung:

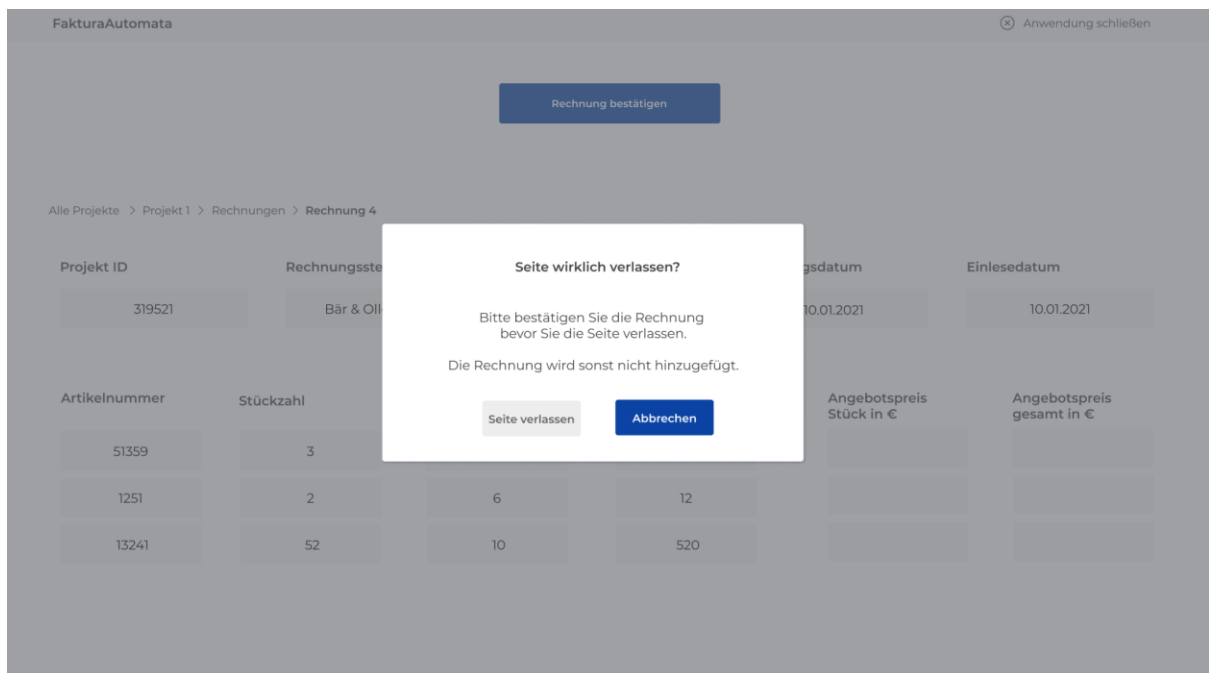


Abbildung 9: Aufforderung zum Speichern der Rechnung beim Verlassen der Bearbeitungsansicht

Sollte der Nutzer versuchen, die Bearbeitungsansicht zu verlassen, ohne die dort angezeigte Rechnung vorher im Datenbestand gespeichert zu haben, so wird ihm der obige Dialog angezeigt, welcher vor einem entsprechenden Datenverlust warnt und ihn so auf ungespeicherte Änderungen bzw. Rechnungen aufmerksam macht.

Rechnungsansicht nach erfolgreichem Speichern:

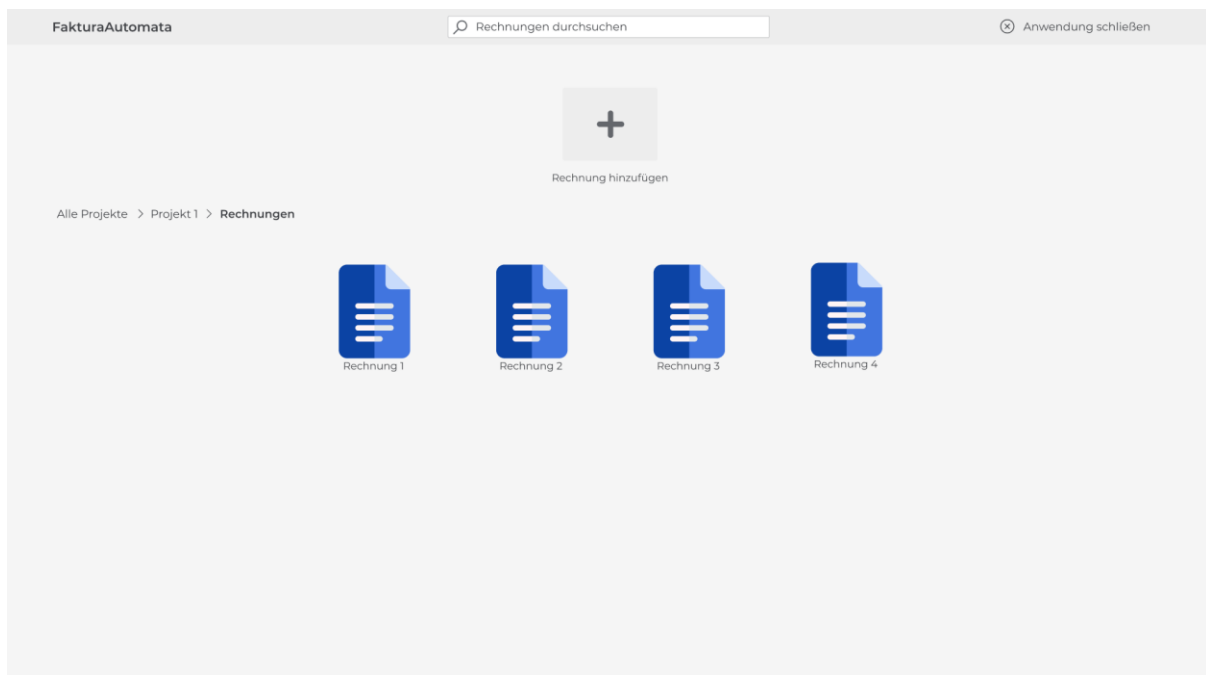


Abbildung 10: Übersicht aller Rechnungen des aktuellen Projektes nach erfolgreichem Speichern einer Rechnung im Datenbestand

Nachdem der Nutzer die eingeleseene Rechnung erfolgreich im Datenbestand gespeichert hat, wird er automatisch zur aktualisierten Ansicht aller Rechnungen des aktuellen Projektes weitergeleitet, in welcher die neue Rechnung nun auch auftaucht.

Geplante Tools, Frameworks etc.

Organisation: Trello, Slack, Discord, Google Docs

Frameworks: Express (Node.js), React, Bootstrap, MongoDB Compass

Datenbank: MongoDB

Entwicklungsumgebung: Visual Studio Code

Programmiersprachen: JavaScript, HTML, CSS, VBA

Tools zur Nutzung der Anwendung: beliebiger Browser, Microsoft Excel

Grobe Architektur und Zeitplan

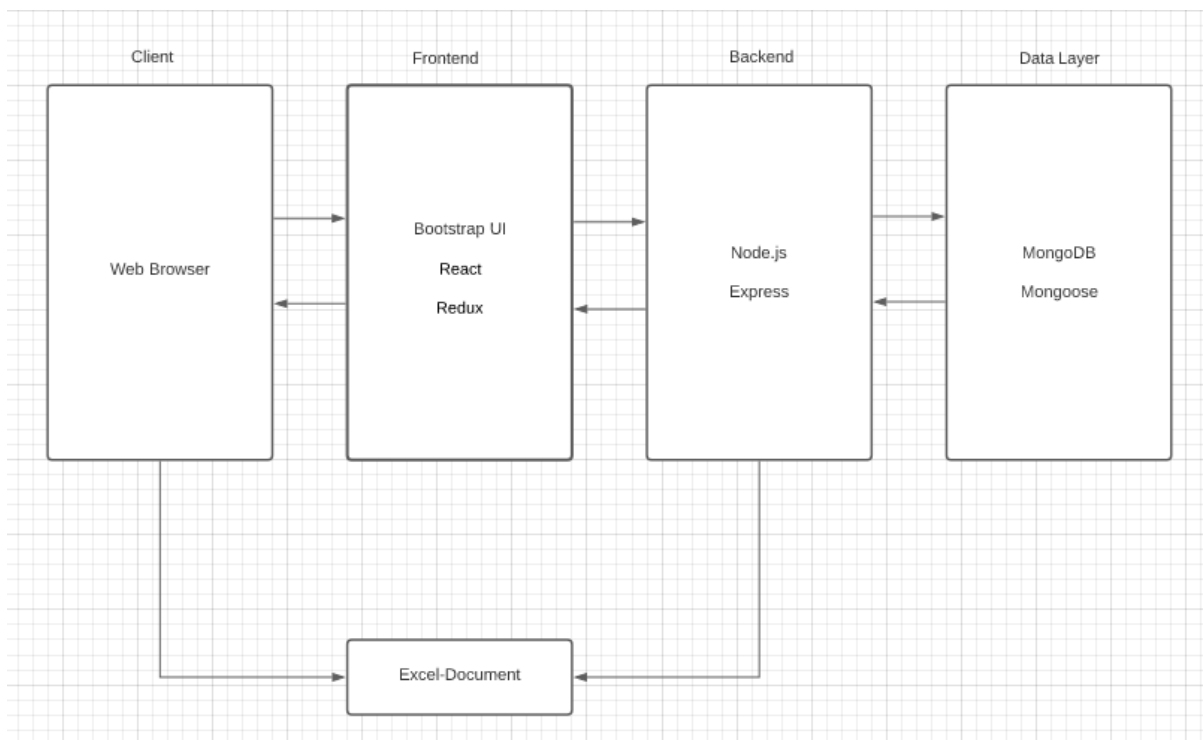


Abbildung 11: Grobe Skizze der geplanten Architektur für FakturaAutomata

Mindestziel: Die Software erlaubt das Einlesen von PDF-Rechnungen und gibt eine übersichtliche editierbare Excel-Datei zurück.

Dafür müssen die folgenden User Stories umgesetzt werden:

- PDF-Rechnung einlesen [8 Points]
- Rechnungsdaten abspeichern [5 Points]
- Export aus DB zu .xlsx [8 Points]
- Excel-Struktur und Formeln [3 Points]
- Angebotspreise ergänzen [5 Points]
- Neue Rechnungen hinzufügen [1 Point]
- Rechnungsdaten verwalten [8 Points]
- Farbliche Differenzerkennung [8 Points]

Sprint 1 (09.04 - 23.04): Ideenfindung und Konkretisierung der Projektvorstellung in Absprache mit dem Kunden

Sprint 2 (23.04 - 07.05): Analyse und Planungsphase, Design-Prototyp, Projektbeschreibung

Sprint 3 (07.05 - 21.05): Backend und Frontend funktionstüchtig bekommen, Models, Routen und Server einrichten, UI Bootstrap Design beginnen, Einlesen der PDFs und Rechnungsdaten in Datenbank überführen

Sprint 4 (21.05 - 04.06): Backend/Frontend/UI um Funktionen zum Export des Datenbestands in strukturierte Excel-Datei erweitern, Angebotspreise eintragen ermöglichen, mehr Rechnungen hinzufügen gewährleisten

Sprint 5 (04.06 - 18.06): Rechnungsdaten verwalten/korrigieren, automatische Einordnung von Rechnungen in zugehörige Projekte, Farbliche Hervorhebung der Differenzen

Sprint 6 (18.06 - 02.07): Login, Remote Access zur gesamten Anwendung (Server/Datenbestand), ggf. letzte Extras implementieren

Sprint 7 (02.07 - 16.07): Feinschliff an Back- und Frontend, Kontrolle, Präsentation vorbereiten

Mögliche Herausforderungen und Risiken

1. Serveranbindung (nicht lokal)
 - Mangel an Erfahrung eine Serveranbindung einzurichten für eine Applikation auf die mehrere Nutzer eines lokalen Netzwerkes zugreifen können
 - eventuell Änderungen in Firewall-Einstellungen der jeweiligen Nutzer nötig
 - eventuell Remote-Server nötig, auf dem die Web-Applikation läuft. Up-Times nur während festgelegten Betriebszeiten? Kosten?
2. Datenhaltung (nicht lokal)
 - Entscheidung ob gleichzeitige Nutzung der Applikation von 2 oder mehr Nutzern zugelassen werden sollte, da Risiko für gegenseitiges Überschreiben von Daten
 - Spezielle Berechtigungen nötig, damit die Applikation den Datenbestand und Verzeichnisse auf dem Netzwerklaufwerk ändern kann?
3. Zuverlässige Konversion
 - Unterschiedliche Zeichensätze (z.B. ASCII zu Unicode) für Quell-/Zielformat könnte zu falscher Zeichenkonversion oder auch Applikations-Absturz führen
 - mögliche Unterschiede im ZUGFeRD-Format könnten die automatische Texterkennung erschweren, z.B. falls XML-Tags für gleiche Felder in verschiedenen Rechnungen nicht gleiche Bezeichnung haben
4. Generierung der Excel-Tabelle (Schnittstelle zu Javascript)
 - Formeln und Formatierungen, wie automatische farbliche Kennzeichnung für negative/positive Werte, sollen vor der Generierung per Code festgelegt sein
 - Schnittstelle zum Erstellen von .xlsx in Javascript mit beliebigen VBA-Formatierungen könnte schwer zu finden sein

Links zu Online-Ressourcen

Gitlab:

<https://gitlab.beuth-hochschule.de/s78206/projekt-rechnungspruefung>

Trello Main Board:

<https://trello.com/invite/b/XQJPP1W6/1313e5fb4bad4051614101a5e07987f5/main>

Link zum clickbaren Figma-Prototypen:

https://www.figma.com/proto/kjCXLtKgiHnKmXK11WirGo/Projekt_Prototyp_FakturaAutomat?page-id=0%3A1&node-id=1%3A61&viewport=314%2C-169%2C0.7962720990180969&scaling=min-zoom