



IHK Kassel-Marburg

Abschlussprüfung Winter 2022/23

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

Rechnungserstellung

Der Prozess für die Erstellung von Rechnungen für die bereits existierende greenhats® Plattform, welche in Ruby on Rails entwickelt ist, soll neu entwickelt werden.

Azubinummer: 0000947538

Abgabedatum: 23.11.2022

Prüfungsbewerber:

Mirco Mettbach
17.02.1994
Schwanallee 34
35037, Marburg

Ausbildungsbetrieb:

IAD GmbH Marburg
Neue Kasseler Str. 62E
35039, Marburg

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	III
Tabellenverzeichnis	III
1. Einleitung	1
1.1 Projektumfeld	1
1.1.1 IAD GmbH	1
1.1.2 evait security GmbH.....	2
1.1.3 greenhats® GmbH.....	2
1.2 Projektziel.....	2
1.3 Projektbegründung.....	3
1.4 Projektschnittstellen.....	3
1.5 Projektabgrenzung	3
2. Projektplanung.....	4
2.1 Projektphasen	4
2.2 Abweichung vom Zeitplan	4
2.3 Ressourcenplanung	5
2.4 Entwicklungsprozess.....	5
3. Analysephase	6
3.1 Ist-Analyse	6
3.2 Projektkosten	6
3.2.1 Amortisationsdauer	7
3.2.2 Make or buy Entscheidung.....	7
3.3 Nutzwertanalyse.....	7
3.5 Qualitätsanforderungen	8
3.6 Lastenheft	8
4. Entwurfsphase	9
4.1 Zielplattform	9
4.2 Architekturdesign.....	9
4.4 Datenmodell	10
4.5 Geschäftslogik.....	11
4.6 Maßnahmen zur Qualitätssicherung	12
4.7 Pflichtenheft	12
5. Realisierungsphase	13
5.1 Implementierung der Geschäftslogik	13
5.2 Implementierung der Datenstruktur	15

5.3 Test der Anwendung auf der greenhats® Plattform	15
6. Abschlussgespräch.....	15
7. Einführung	16
8. Dokumentation.....	16
9. Fazit.....	17
9.1 Soll/Ist-Vergleich	17
9.2 Lessons learned.....	17
9.3 Ausblick.....	17
10. Anhang	IV
A1. Flowchart bisheriger Prozess	IV
A2. Flowchart neuer Prozess	IV
A3. Rechnungsvorlage	V
A4. Nachbau der Rechnungsvorlage	VI
A4. Datenbankmodell	VII
A5. HTML-Template	VIII
10.1 Quellenverzeichnis	X
10.3 Quellcode	XI
10.3.1 Vollständiger Source-Code der Funktion.....	XI

Abbildungsverzeichnis

Abbildung 1: Screenshot Cursor positionieren	13
Abbildung 2: Screenshot Iterieren von Rechnungspositionen	13
Abbildung 3: Screenshot Kunden Referenz prüfen	14
Abbildung 4: Screenshot Input/Output	14
Abbildung 5: Screenshot SVG einfügen.....	14
Abbildung 6: Flowchart bisheriger Prozess	IV
Abbildung 7: Flowchart neuer Prozess.....	IV
Abbildung 8: Rechnungsvorlage	V
Abbildung 9: Nachbau der Rechnungsvorlage	VI
Abbildung 10: Auszug relevanter Entitäten	VII
Abbildung 11: Screenshot HTML-Template	IX
Abbildung 12: Screenshot des Funktion Quellcodes	XII

Tabellenverzeichnis

Tabelle 1: Zeitplanung	4
Tabelle 2: Ressourcenplanung	5
Tabelle 3: Kosten-Kalkulation	6
Tabelle 4: Nutzwertanalyse.....	7
Tabelle 5: Soll-Ist Vergleich	17

1. Einleitung

Die Dokumentation beschreibt den Ablauf des IHK-Abschlussprojektes, welches im Rahmen einer sechsmonatigen Praxisausbildung bei der Firma evait Security vom Autor umgesetzt wurde. Eine Vertiefung und Veranschaulichung, der während der zweijährigen Umschulung erworbenen Fertigkeiten können auf diesem Wege gewährleistet werden. Die Erarbeitung des Projektes stellt eine praxisnahe Aufgabe dar, die dem Auszubildenden im späteren Berufsalltag oft begegnen wird und somit eine realistische Übung darstellt. Das Projekt wurde im Auftrag der greenhats® GmbH, von der Firma evait security GmbH im Verlauf der schulischen Praxisqualifikation umgesetzt. Die Betreuung der Umsetzung wurde an den Autor weitergegeben. Die Firma greenhats® GmbH stellte eine Test-Instanz der greenhats® Plattform für dieses Projekt bereit.

1.1 Projektumfeld

In dieser Dokumentation ist das Projektumfeld auf drei Bereiche aufgegliedert. Der Autor hat seine Ausbildung schulisch absolviert und für den Erwerb von Arbeits- und Praxiserfahrung ein Praktikum bei der Firma evait Security durchlaufen.

1.1.1 IAD GmbH

Die IAD GmbH ist ein Schulungsunternehmen im Schwerpunkt der Informatik. Das Unternehmen betreibt mehrere Standorte in Deutschland und bietet Umschulungen, Weiterbildungen und Zertifikate an.

Über die IAD GmbH wurde die schulische Qualifikation des Autors im Bereich Fachinformatik Anwendungsentwicklung übermittelt.

1.1.2 evait security GmbH

evait security GmbH ist ein pentesting Unternehmen in Roth. Die Arbeit der evait security GmbH besteht daraus, IT-Infrastrukturen und Software aus der Sicht eines Hackers zu überprüfen, um Unternehmen ein hohes Niveau an Sicherheit bieten zu können. Hierbei werden realistische Angriffsszenarien simuliert, um proaktiv Schwachstellen aufzudecken. Mit diesen Informationen können die Unternehmen ihre Sicherheitslücken schließen und für potenzielle Angreifer unschädlich machen. Der Autor hat hier seine praxisrelevanten Fachkenntnisse sammeln können und die Möglichkeit erhalten, diese für ein reales Projekt anzuwenden.

1.1.3 greenhats® GmbH

Auftraggeber des hier durchgeführten und dokumentierten Projekts, ist die greenhats® GmbH. greenhats® ist die erste in Deutschland gegründete Plattform für Security Checks, Phishing, Penetrationstests, Bug Bounty, IT-Sicherheit und Compliance. In diesem einzigartigen Konzept werden Unternehmen und Spezialisten der IT-Sicherheit zusammengeführt, um gemeinsam Schwachstellen zu beheben und Risiken proaktiv entgegenzuwirken.

1.2 Projektziel

Ziel des Projektes ist es mittels der Programmiersprache Ruby und der Bibliothek (in Ruby auch "Gem" genannt) **Prawn**, den Prozess zur Erstellung von Rechnungen auf der greenhats® Plattform neu zu gestalten, vereinfachen und zu optimieren.

1.3 Projektbegründung

Aktuell wird dieser Prozess über einen externen Docker Container mit der Software Selenium (Headless Chrome) und ein entsprechendes HTML-Template durchgeführt. Die dafür genutzte Lösung: Athenapdf ist nicht nur umständlich, sondern stellt als microservice einen zusätzlichen Point of failure dar. Darüber hinaus wird das Projekt Athenapdf seit 2018 nicht mehr aktualisiert und weitergeführt. Als IT-Sicherheitsfirma, die mit hoch sensiblen Daten arbeitet, könnte eine in Zukunft vorkommende Schwachstelle im Athenapdf-Code gravierende Folgen haben. Somit ist die Entscheidung den Prozess zu erneuern alternativlos.

Die Ruby-Gem Prawn ist schnell und leichtgewichtig, und der Funktions-Code kann direkt in die Plattform integriert werden, sodass eine externe Fehlerquelle bei der Erzeugung von PDFs eliminiert wird.

1.4 Projektschnittstellen

Der Prawn Source-Code wird direkt im bestehenden Controller auf der greenhats® Plattform implementiert, und ersetzt damit den alten Code. Dabei interagiert der Controller mit der dahinterliegenden Datenbank und verwertet bereitgestellte Daten.

Benutzer der Plattform sind Mitarbeiter und alle Kunden der greenhats® GmbH.

Der geschlossene Ruby on Rails Source-Code der greenhats® Plattform, sowie eine lokale Test-Instanz inklusive Test-Daten wurden dem Autor bereitgestellt.

1.5 Projektabgrenzung

Das Projekt umfasst ausschließlich die Erzeugung von PDF-Dateien mithilfe von Daten, welche durch die Plattform dynamisch bereitgestellt werden. Speicherung der PDF-Datei in der Datenbank, sowie Abrufen von Daten, Versand von E-Mails mit Rechnungsanhang und Dateien oder Löschen der entsprechenden Datei-Objekte sind bereits implementiert und nicht Bestandteil des Projektes.

2. Projektplanung

2.1 Projektphasen

Tabelle 1: Zeitplanung

Zeitplanung	
1. Analysephase	5h
Ist- Zustand	1h
Soll- Zustand	1h
Recherche	2h
Planung des Projektes	1h
2. Entwurf Phase	8h
Konzepterstellung	3h
Entwurf erstellen	5h
3. Realisierungsphase	57h
Entwickeln der PDF mit Ruby und Prawn	30h
Implementation in den Plattform Quellcode	15h
Testphase und Debugging	12h
4. Abschlussphase	10h
Dokumentation	9h
Abschlussgespräch mit dem Auftraggeber	1h
Gesamt	80h

2.2 Abweichung vom Zeitplan

Der zuvor erstellte Projektplan konnte eingehalten werden. Der Autor hat jedoch eine geringfügig längere Recherche der Prawn Bibliothek betrieben als geplant. Dies konnte allerdings in der Entwicklungsphase kompensiert werden, sodass das Projekt im festgelegten Zeitrahmen umgesetzt werden konnte.

2.3 Ressourcenplanung

Tabelle 2: Ressourcenplanung

<p>Analyse und Entwurf Phase:</p> <ul style="list-style-type: none"> • Meeting-Raum • Arbeitsplatz (Schreibtisch, Stuhl, Strom, Laptop (Linux, Manjaro)) • Software (Office-Software, Whimsical.com (Flowchart), Prawn Dokumentation (https://prawnpdf.org))
<p>Realisierungsphase</p> <ul style="list-style-type: none"> • Arbeitsplatz • Software (Visual Studio Code, Ruby on Rails server) • Sprachen (Ruby) • Bibliotheken/Gems (Prawn, Prawn-svg, Prawn/measurement_extensions) • Datenbank, Ruby on Rails Source-Code der greenhats® Plattform, Test- Instanz
<p>Abschlussphase</p> <ul style="list-style-type: none"> • Meeting-Raum • Arbeitsplatz • Software (Office-Software, Prezi) • Diagramme (Flowchart, Datenbankdiagramm bereitgestellt von greenhats® GmbH) • Hardware (Beamer)

2.4 Entwicklungsprozess

Bevor mit der Realisierung des Projektes begonnen werden konnte, musste sich der Autor für einen geeigneten Entwicklungsprozess entscheiden. Dieser bestimmt das Vorgehen, nach der die Durchführung des Projektes realisiert werden soll. Es wurde sich für einen agilen Entwicklungsprozess entschieden. Dieser erlaubt das zyklische Durchlaufen der Projektphasen und ermöglicht beständige Rücksprache mit dem Auftraggeber.

3. Analysephase

3.1 Ist-Analyse

Funktionsanalyse:

Aktuell wird dieser Prozess über einen externen Docker Container mit der Software Selenium (Headless Chrome), ein entsprechendes HTML-Template und Athenapdf durchgeführt.

In regelmäßigen Abständen stößt der „create_invoice_files“ Controller einen Job an und übergibt die entsprechenden Datenbankobjekte an ein Render HTML-Template.

Das fertige HTML-Template „Athena.html.erb“ wird anschließend per HTTP Request an einen Docker-Container mit Selenium (Headless Chrome) und dem Dienst Athenapdf geschickt. Athenapdf (Ein einfaches Tool zum Konvertieren von HTML zu PDF) konvertiert und rendert das HTML-Template anschließend zu einer PDF und schickt es zurück zur greenhats® Plattform, wo die PDF-Datei als Blob¹ in der Datenbank gespeichert wird.

Eine entsprechende Veranschaulichung finden sie im [Anhang A1](#).

3.2 Projektkosten

Tabelle 3: Kosten-Kalkulation

Kosten-Kalkulation

Vorgang	Mitarbeiter	Zeit	Kostenpauschale (Stunde)	Gesamtkosten
Entwicklung	Auszubildender	80h	12 €	960 €
Fachgespräch	Mitarbeiter	2h	20 €	40 €
Fachgespräch	Ausbilder / IT-Leiter	3h	25 €	75 €
Code-Review	Mitarbeiter	3h	20 €	60 €
Abnahme	Geschäftsführer	2h	50 €	100 €
Projektkosten Gesamt:				1.235 €

¹ Blob: Binary Large Objects sind große binäre Datenobjekte wie z. B. Bild- oder Audiodateien.

3.2.1 Amortisationsdauer

Eine Amortisationsdauer kann im Falle des Autors nicht berechnet werden, da es sich in erster Linie um eine Prozessoptimierung handelt.

Wie im Absatz 1.3 beschrieben, ist ein mögliches Sicherheitsrisiko beim Verwenden nicht aktualisierter Technik für den Auftraggeber keine Option. Um die Rechnungserstellung ausfallsicher und performant zu gestalten, wurde Geld investiert, ohne eine monetäre Amortisierung zu erwarten.

3.2.2 Make or buy Entscheidung

Die greenhats® Plattform wurde hausintern entwickelt und wird stetig erweitert und gewartet.

Da es sich bei diesem Projekt um eine Prozessoptimierung handelt und gleichzeitig mit sensiblen Kundendaten gearbeitet wird, hat sich die Frage des Fremdkaufs für greenhats® und dem Autor nicht gestellt.

3.3 Nutzwertanalyse

Tabelle 4: Nutzwertanalyse

Kriterien	Gewicht	Prozess mit Prawn programmieren		Prozess nicht erneuern	
		Bewertung 0 - 5	Gesamt	Bewertung 0 - 5	Gesamt
Zeitaufwand bei Anpassungen	10%	5	0,5	2	0,2
Datenhoheit	30%	5	1,5	0	-
Kosten	10%	0	-	5	0,5
Sicherheitsniveau	50%	4	2,0	2	1,0
Gesamt	100%		4,0		1,7

In der Nutzwertanalyse wurde eine Erneuerung des Prozesses, einem beibehalten des alten gegenübergestellt.

Das Ergebnis spricht eindeutig für eine Erneuerung des Prozesses.

3.5 Qualitätsanforderungen

Zur Absicherung der Qualität und Funktionalität des Prozesses wurden folgende Anforderungen festgelegt:

- Die Rechnungsvorlage soll mit allen Design-Elementen nahezu pixelgenau nachgebaut werden.
- Dynamische Erzeugung von verschiedenen langen Produktbeschreibungen und das Iterieren mehrerer Angebote muss gegeben sein.

3.6 Lastenheft

Anforderungen

Von der Funktion müssen folgende Anforderungen erfüllt werden:

Frontend

- Die Rechnung soll mit allen Design-Elementen (Logo, Tabellen, etc.) fast pixelgenau dem Muster nachempfunden sein.
- Eine Rechnungsvorlage wurde vom Auftraggeber übergeben und befindet sich im [Anhang A3](#).

Backend

- Die entwickelte Funktion soll in die bestehende Plattform an der Stelle des spezifischen Rechnungs-Controllers implementiert werden.
- Der Controller-Code soll so verändert werden, dass die neu entwickelte Rechnungserzeugung die alte Funktion vollständig ersetzt.
- Der Source-Code soll simpel und wartbar sein, damit dieser in Zukunft leicht weiterentwickelt werden kann.
- Der Proof of Concept Code soll mit Plattformspezifischen Funktionen und Variablen erweitert werden.

4. Entwurfsphase

4.1 Zielplattform



Die Zielplattform ist die hausinterne greenhats® Plattform, welche in Ruby on Rails programmiert wurde.

Ruby on Rails nutzt das MVC-Modell, bei der View, Controller und Model miteinander interagieren, um Ausgaben zu erzeugen und gleichzeitig den Source- Code übersichtlich strukturieren und modular gestalten.

Die Aufgabe des Autors bezieht sich auf einen spezifischen Controller-Code, der bisher mit der Rechnungserstellung beauftragt war.

4.2 Architekturdesign

Da die greenhats® Plattform mit Ruby on Rails geschrieben wurde, musste der Autor ebenfalls Ruby als Programmiersprache wählen.

Ruby



Die Programmiersprache Ruby ist objektorientiert, unterstützt zudem aber weitere Programmierparadigmen wie z.B. funktionales- sowie prozedurales Programmieren.

Prawn



Die Ruby- Gem Prawn ermöglicht es PDFs in reinem Ruby Code zu erstellen. Dabei ist es schnell und leicht gewichtig.

Prawn ist ein Open-Source Projekt, welches 2014 ins Leben gerufen und seither von verschiedenen Entwicklern gewartet und erweitert wird.

Der Autor konnte sich im Rahmen des betrieblichen Praktikums ein Grundverständnis von Ruby und Ruby on Rails aneignen.

Im Laufe des Projektes konnte sich der Autor mit der Ruby- Gem Prawn befassen, welche eine weitere Einarbeitung benötigte.

4.4 Datenmodell

Die Datenbank der greenhats® Plattform war nicht Teil des Projektes. Während dem Debugging musste sich der Autor allerdings in geringem Maße damit auseinandersetzen. Eine Darstellung der für den Autor relevanten Entitäten ist im [Anhang A4](#) zu finden und wurde dem Autor von greenhats® zur Verfügung gestellt.

4.5 Geschäftslogik



Visual Studio Code

Der Autor hat sich zum Realisieren der Geschäftslogik für Visual Studio Code als Entwicklerumgebung entschieden. Die ressourcenarme und leichte Bedienbarkeit, sowie die Option eine Vielzahl von Add-ons zu installieren, haben den Autor überzeugt.

Im ersten Schritt war es nötig ein Proof of Concept Code zu erstellen und sich soweit es möglich war an die Mustervorlage der Rechnungsdatei zu orientieren. Dazu war es nötig, anhand der Prawn Dokumentation, Schritt für Schritt Besonderheiten zu verstehen und Lösungen zu finden.

Nachdem der Proof of Concept Code mit eigenen Test-Variablen getestet und erstellt war, konnte der Autor mit der nächsten Phase des Projektes beginnen.

Die greenhats® Plattform wurde dem Autor als Test-Instanz lokal zur Verfügung gestellt und konnte zum Testen der Implementierung genutzt werden.

Im Source- Code der greenhats® Plattform wurde der spezifische Controller ausfindig gemacht und durch den Proof of Concept Code weitestgehend ersetzt.

Nun war es nötig, die zuvor erstellten Test-Variablen durch Plattformspezifische Variablen zu ersetzen.

Nachdem die Implementierung der Variablen beendet war, konnte der Autor an der Anbindung und Ausgabe der PDF-Datei arbeiten.

Der Controller musste durch eine Rails spezifische Funktion erweitert werden, um eine Ausgabe der PDF-Datei, beim Betätigen des Download-Buttons auf der greenhats® Plattform, zu erzeugen.

Zum Schluss wurde die Funktion ausgiebig geprüft und auf Edge-Cases getestet.

Das agile Vorgehensmodell hat es dem Autor ermöglicht, umfassende und kleinere Test während der Entwicklung durchzuführen.

4.6 Maßnahmen zur Qualitätssicherung

Um die Qualität und Funktionsweise sicherzustellen wurden im Laufe der Entwicklung eine Vielzahl von Testausgaben vorgenommen und mit Hilfe der Test-Instanz der greenhats® Plattform verschiedene Szenarien abgebildet.

Unter anderem war es dem Autor möglich eigene Aufträge zu erstellen und fiktive Rechnungen zu erzeugen und zu testen.

4.7 Pflichtenheft

Der folgende Auszug aus dem Pflichtenheft beschreibt die geplante Umsetzung der im Lastenheft definierten Anforderungen.

Frontend

- Die Mustervorlage mit der Ruby Gem Prawn nachbauen und alle Design Elemente implementieren.

Backend

- Mit Prawn Proof of Concept Code Programmieren.
- Proof of Concept Code durch zuvor ermittelten Plattformspezifischen Variablen Mithilfe des HTML-Templates „Athena.html.erb“ erweitern.
- Ausgabe Funktion Implementieren.
- Neue Funktion im Controller „create_invoice_files.rb“ implementieren und alte ersetzen.

5. Realisierungsphase

Die Dokumentation der Realisierungsphase enthält einzelne Code Ausschnitte die zur Veranschaulichung dienen.

5.1 Implementierung der Geschäftslogik

Die Implementierung der vorgegebenen Design Elemente ist ein Hauptbestandteil des Projektes und erforderte ein tiefes Auseinandersetzen mit der Ruby Bibliothek Prawn.

```
98 | # Cursor Position speichern
99 | cursor_position = pdf.cursor
100 |
101 | # Nettobetrag/Umsatzsteuer ect einfügen und positionieren
102 | pdf.text_box "Nettobetrag" , at: [350, cursor_position ], size: 9
103 | pdf.transparent(0.1) {pdf.stroke_horizontal_line 350, 515, at: cursor_position -15 }
104 | pdf.draw_text "Umsatzsteuer #{invoice.vat_display} %" , at: [350, cursor_position -27 ], size: 9
105 | pdf.transparent(0.1) {pdf.stroke_horizontal_line 350, 515, at: cursor_position -32}
106 | pdf.draw_text "Rechnungsbetrag" , at: [350, cursor_position -45 ], size: 9, style: :bold
107 |
108 | # Preise einfügen und positionieren
109 | pdf.text_box "#{invoice.price.format} €" , at: [455, cursor_position], size: 9, align: :right, width: 60
110 | pdf.text_box "#{invoice.vat.format} €", at: [455, cursor_position -20 ], size: 9, align: :right, width: 60
111 | pdf.text_box "#{invoice.with_vat.format} €", at: [455, cursor_position -39 ], size: 9, align: :right, width: 60, style: :bold
112 |
```

Abbildung 1: Screenshot Cursor positionieren

Prawn arbeitet mit X- und Y Koordinaten, sowie einem dynamischen Cursor, um Text zu positionieren. Eine wiederkehrende Herausforderung war die korrekte Darstellung und Positionierung der gewünschten Daten.

```
88 | # rechnungspositionen einfügen und iterieren falls mehrere Positionen vorhanden sind
89 | invoice.invoice_positions.each_with_index do |ip, i|
90 |   pdf.text_box "#{ip.amount}", at: [0, 332 -20 * i ], size: 9
91 |   pdf.text_box "#{ip.description}", at: [65, 332 -20 * i], width: 300, size: 9, height: 10, :overflow => :shrink_to_fit
92 |   pdf.text_box "#{ip.price.format} €", at: [360, 332 -20 * i], size: 9, align: :right, width: 60
93 |   pdf.text_box "#{(ip.price * ip.amount).format} €", at: [455, 332 -20 * i], size: 9, align: :right, width: 60
94 |   pdf.transparent(0.1) {pdf.stroke_horizontal_line 0, 515, at: 320 -20 * i }
95 |   pdf.move_cursor_to (332 -20 * (i + 1))
96 | end
```

Abbildung 2: Screenshot Iterieren von Rechnungspositionen

Das Iterieren von mehreren Rechnungspositionen musste sensibel positioniert werden.

```

52 | # Kundendatenbezeichnung einfügen und positionieren
53 | pdf.transparent(0.6) {pdf.text_box ["Datum", "Kundennummer", "Auftrag", "Referenz", "Rechnungsnummer"].reject{|d| d == "Referenz" && !invoice.
customer_reference}.join("\
54 | " ), size: 10, at: [290,580], width: 100}
55 |
56 | # Kundendaten einfügen und positionieren
57 | pdf.transparent(0.6) {pdf.text_box ["#{invoice.invoice_date}", "#{invoice.customer_number}", "#{invoice.job_number}", "#{invoice.
customer_reference}", "<b>#{invoice.number}</b>"].reject{|d| d == "#{invoice.customer_reference}" && !invoice.customer_reference}.join("\
58 | " ), size: 10, at: [430,580], width: 80, inline_format: true}
59 |

```

Abbildung 3: Screenshot Kunden Referenz prüfen

Eine Besonderheit war die Prüfung, ob eine Kunden Referenz vorhanden ist, und die entsprechenden Daten einzufügen.

```

124 | begin
125 |   io = StringIO.new pdf.render
126 |   invoice.invoice_file.attach(io: io, filename: "invoice_#{invoice.number}.pdf")
127 | end

```

Abbildung 4: Screenshot Input/Output

Die Klasse „StringIO“ wird verwendet um den Standard Input/Output zu simulieren und „pdf.render“ zu übergeben. Die Anbindung zum Download Button auf der greenhats® Plattform ist bereits implementiert.

```

33 | pdf.svg '<?xml version="1.0" encoding="UTF-8" standalone="no"?> <!-- Created with Inkscape (http://www.inkscape.org/) --><svg
34 | xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/
22-rdf-syntax-ns#" xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg" xmlns:sodipodi="http://sodipodi.sourceforge.
net/DTD/sodipodi-0.dtd" xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape" version="1.1" id="svg2" xml:space="preserve"
width="809.38269" height="197.81467" viewBox="0 0 809.38269 197.81467"
sodipodi:docname="logo_horizontal-1b2efec7b1e0d88f6d374259322426a695ca2fafb909635169b364f7dabb62a4.svg" inkscape:version="0.92.4
5da689c313, 2019-01-14" inkscape:export-xdpi="474.43564" inkscape:export-ydpi="474.43564"><metadata id="metadata8"><rdf:RDF><cc:Work
rdf:about=""><dc:format>image/svg+xml</dc:format><dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage" /><dc:title /></cc:Work></
rdf:RDF></metadata><defs id="defs6"><clipPath clipPathUnits="userSpaceOnUse" id="clipPath18"><path d="M 0,148.361 H 607.037 V 0 H 0 Z"
id="path16" inkscape:connector-curvature="0" /></clipPath></defs><sodipodi:namedview pagecolor="#ffffff" bordercolor="#666666"
borderopacity="1" objecttolerance="10" gridtolerance="10" guidetolerance="10" inkscape:pageopacity="0" inkscape:pageshadow="2"
inkscape:window-width="1920" inkscape:window-height="1080" id="namedview4" showgrid="false" inkscape:zoom="1.2146778" inkscape:cx="407.
18617" inkscape:cy="99.09517" inkscape:window-x="0" inkscape:window-y="0" inkscape:window-maximized="1"
inkscape:current-layer="g4931" /><g id="g10" inkscape:groupmode="layer" inkscape:label="horizontal" transform="matrix(1.3333333,0,0,-1.

```

Abbildung 5: Screenshot SVG einfügen

Die Gem „prawn-svg“ hat es ermöglicht, das Logo der greenhats® GmbH direkt in den Code einzubetten, um damit externe Abhängigkeiten zu minimieren.

5.2 Implementierung der Datenstruktur

Nachdem der Proof of Concept Code erstellt wurde, konnten die Plattform spezifischen Variablen, anschließend mithilfe des bereits existierenden Athena HTML-Templates (siehe [Anhang A5](#)) analysiert, extrahiert, und implementiert werden.

5.3 Test der Anwendung auf der greenhats® Plattform

Nachdem der alte Controller Code durch den neuen Code ersetzt wurde, war es nötig in der Test-Instanz alte Rechnungsdateien zu löschen und neue zu erzeugen. Eine zu Test Zwecken bereits implementierte Methode „Invoice.invoice_file.purge“ ermöglicht die Löschung der PDF Dateien im internen Speicher der Plattform.

„Invoice.last(50).each.do |i| i.invoice_file.purge end“ ermöglicht es beispielweise, durch iterieren der Methode, die letzten fünfzig Rechnungs-PDF Dateien zu löschen.

Mit der ebenfalls bereits implementierten Methode „CreateInvoiceFiles.run“ wird der Controller „create_invoice_files“ angestoßen und neue Rechnungsdateien werden erzeugt.

Beide Methoden konnten in der Rails Konsole der Plattform direkt ausgeführt werden.

Mit diesen Werkzeugen wurde die Funktion ausgiebig getestet und verschiedene Test-Fälle konnten abgedeckt werden.

Ein fertiges Muster der Rechnungserstellung ist im [Anhang A4](#) zu finden.

6. Abschlussgespräch

Abschließend wurde ein Termin mit dem Auftraggeber wahrgenommen. Dabei wurde das Projekt abgenommen und eine kurze Einweisung in die Funktion gegeben. Insgesamt fiel das Fazit des Auftraggebers positiv und zufriedenstellend aus.

7. Einführung

Zur weiteren Nutzung wurde der entsprechende Code in ein eigens vorgesehenes Git-Repository hochgeladen, auf das die Entwickler der greenhats® Plattform Zugriff haben.

Während der Entwicklungsphase gab es mehrere Rücksprachen mit den entsprechenden Mitarbeitern der Plattform, was eine zusätzliche Einführung in den Code nicht erforderlich machte.

8. Dokumentation

Die Erstellung der Dokumentation erfolgte, nachdem das Projekt beendet wurde und soll Aufschluss über dessen Aufbau und Ablauf geben. Ein Benutzerhandbuch war im Falle des Autors nicht nötig, da es sich beim durchgeführten Projekt um einen bestimmten Controller Code Abschnitt handelt. Der Code wurde vom Autor mit prägnanten Kommentaren versehen, welche beim zukünftigen Weiterentwickeln des Controllers Aufschluss über dessen Funktionsweise geben soll.

Viele der Screenshots und Abbildungen sind bereits im Verlauf der Entwicklung erstellt worden.

9. Fazit

9.1 Soll/Ist-Vergleich

Tabelle 5: Soll-Ist Vergleich

Kundenwünsche	
Kriterium	Status
Rechnungsdesign beibehalten	✓
Controller-Code ersetzen	✓
Simpler und performanter Code	✓

Bei einer rückblickenden Betrachtung des IHK-Abschlussprojektes, kann festgehalten werden, dass alle zuvor festgelegten Anforderungen gemäß dem Pflichtenheft erfüllt werden konnten.

9.2 Lessons learned

Während der Realisierung des Projektes, konnte der Autor Fertigkeiten sammeln, welche ihm in seiner weiteren Laufbahn als Fachinformatiker helfen werden. Im Verlauf der Entwicklung wurde dem Autor bewusst, wie unabdingbar eine gute Struktur und Vorbereitung bei der Durchführung eines Projektes ist. Der Autor konnte hierbei auf sein erlangtes Vorwissen in Bezug auf Ruby und Ruby on Rails zugreifen. Zusätzlich musste er sich jedoch aufgrund der Ruby-Gem Prawn, neues Wissen aneignen.

Abschließend kann man sagen, dass die Realisierung des Projektes sowohl für den Autor eine große Bereicherung war, als auch einen Mehrwert für die Firma darstellt.

9.3 Ausblick

Obwohl alle im Lastenheft definierten Anforderungen realisiert werden konnten, können in Zukunft dennoch neue Anforderungen definiert und Erweiterungen entwickelt werden. So gibt es bereits Überlegungen, die von Prawn erzeugten PDFs, nicht im internen Speicher der Plattform, sondern direkt dynamisch beim Download zu erzeugen.

10. Anhang

A1. Flowchart bisheriger Prozess

Flowchart des bisherigen Prozesses zum rendern von Rechnungs-PDF Dateien mit Athena.

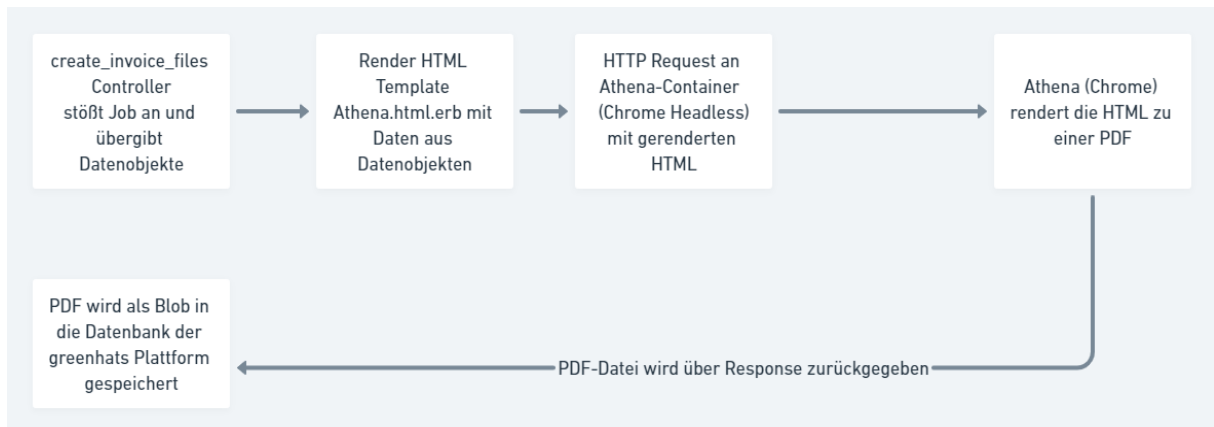


Abbildung 6: Flowchart bisheriger Prozess

A2. Flowchart neuer Prozess

Flowchart des neuen Prozesses mit Prawn.

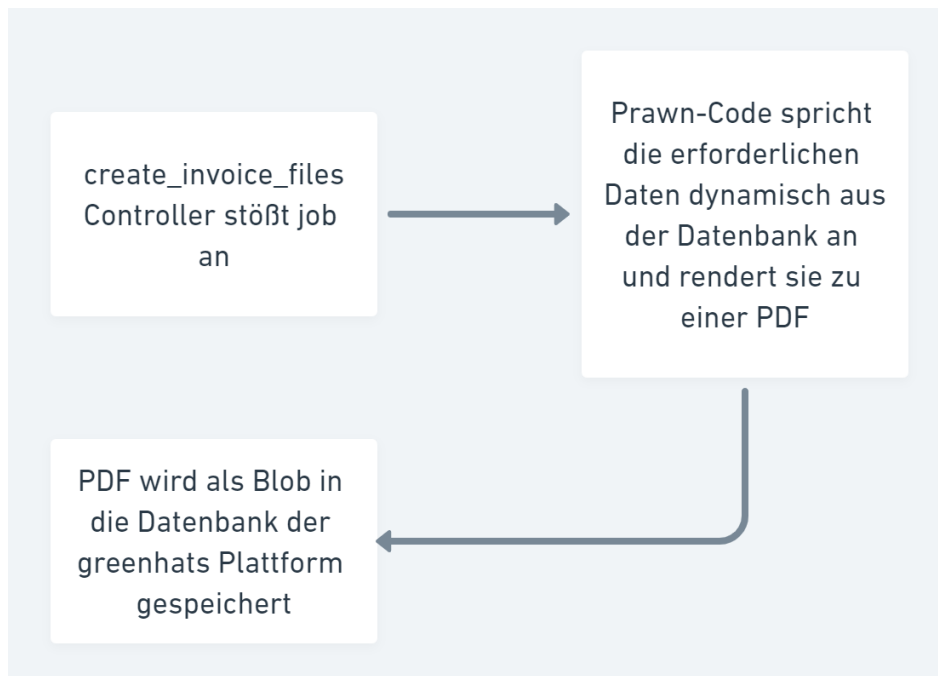


Abbildung 7: Flowchart neuer Prozess

A3. Rechnungsvorlage

Vom Auftraggeber übergeben, Kundendaten wurden geschwärzt.



greenhats® GmbH - Buchenweg 22 - 35096 Weimar (Lahn)

Wichtige Informationen für Sie:
Keine Rückgabe
Keine Umtausch
Keine Rückerstattung

greenhats® GmbH

Buchenweg 22
35096 Weimar (Lahn)
Deutschland

+49 6426 8989-022
support@greenhats.com
www.greenhats.com

Datum

Kundennummer

Auftrag

Rechnungsnummer

12.05.2024

12345678

12345678

1234567890

Rechnung

Abrechnungszeitraum: 01.01.2024 - 31.03.2024

Sehr geehrte Damen und Herren,
wir danken für Ihr Vertrauen und erlauben uns wie folgt Rechnung zu stellen.

Anzahl	Artikel/Beschreibung	Einzelpreis	Gesamtpreis
1	Greenhats® Security Software Lizenz	100,00 €	100,00 €
		Nettobetrag	100,00 €
		Umsatzsteuer 19 %	19,00 €
		Rechnungsbetrag	119,00 €

Zahlungsbedingung: 14 Tage Netto bis zum 15.06.2024

greenhats® GmbH
Buchenweg 22
35096 Weimar (Lahn)

Bankverbindung
IBAN: DE26 5335 0000 0000 1057 16
BIC: HELADEF1MAR

Handelsregister Marburg
Ust.-Nr.: DE310205797
St.-Nr.: 2023480903

Abbildung 8: Rechnungsvorlage

A4. Nachbau der Rechnungsvorlage

Abschließender Nachbau der Rechnungsvorlage.



greenhats® GmbH - Buchenweg 22 - 35096 Weimar (Lahn)

Muster GmbH & Co. KG
Max Mustermann
Muster Str. 1
12345 Musterstadt

greenhats® GmbH

Buchenweg 22
35096 Weimar (Lahn)
Deutschland

+49 6426 8989-022
support@greenhats.com
www.greenhats.com

Datum 1.1.2023
Kundennummer 123456
Auftrag aBC1234x
Rechnungsnummer **1234567891011**

Rechnung

Abrechnungszeitraum: 1.01.2023 - 30.01.2023

Sehr geehrte Damen und Herren,
wir danken für ihr Vertrauen und erlauben uns wie folgt Rechnung zu stellen.

Anzahl	Artikel/Beschreibung	Einzelpreis	Gesamtpreis
1	Muster Kampagne	139.0 €	139.0 €
2	Muster Dienstleistung	250.0 €	500.0 €
Nettobetrag			639.0 €
Umsatzsteuer 19			121.41 €
Rechnungsbetrag			760.41 €

Zahlungsbedingung: 14 Tage Netto bis zum 15.01.2023

greenhats® GmbH
Buchenweg 22
35096 Weimar (Lahn)

Bankverbindung
IBAN: DE26 5335 0000 0000 1057 16
BIC: HELADEF1MAR

Handelsregister Marburg
Ust-Nr.: DE310205797
St-Nr.: 2023480903

Abbildung 9: Nachbau der Rechnungsvorlage

A4. Datenbankmodell

Auszug relevanter Entitäten bei der Erstellung von Rechnungsdateien, die dem Autor von der greenhats® GmbH zur Veranschaulichung bereitgestellt wurde.

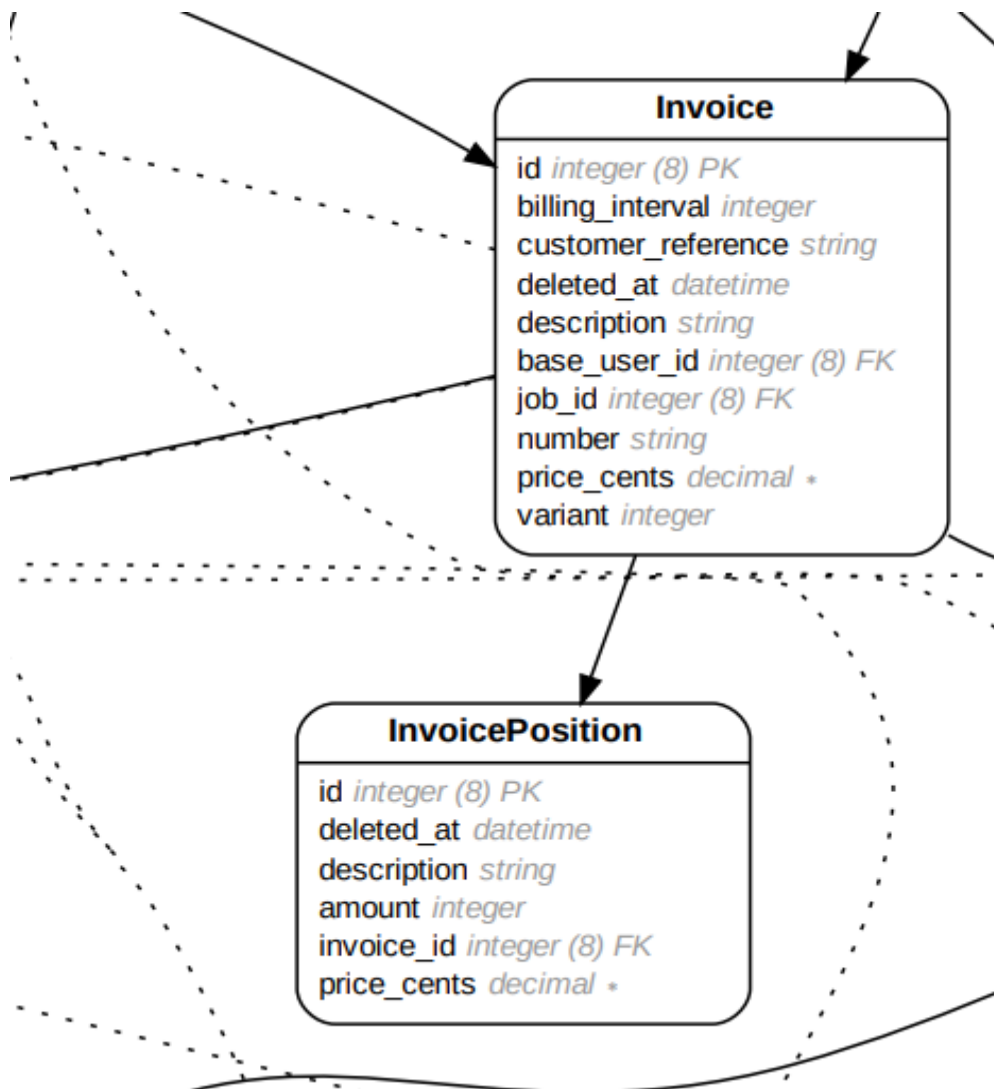


Abbildung 10: Auszug relevanter Entitäten

A5. HTML-Template

Ausschnitt des HTML-Templates, dass über Athena (Chrome) gerendert wird.

```

1  <div class="container-invoice">
2    <div class="row">
3      <div class="col s8 left-align">
4        <%= image_tag "#{MAIN_LOGO_NAME}.svg", class: 'invoice-logo' %><br />
5        <span class="grey-text text-darken-2">greenhats<sup class="smallsuper">*</sup> GmbH - Buchenweg 22 - 35096 Weimar (Lahn)</span><br />
6        <p>
7          <% @invoice.recipient.each do |line| -%>
8            <span><%= line %></span><br />
9          <% end %>
10       </p>
11     </div>
12     <div class="col s4 right-align">
13       <p class="primary-text right-align">greenhats<sup class="smallsuper">*</sup> GmbH</p>
14       <span class="grey-text text-darken-2">Buchenweg 22</span><br />
15       <span class="grey-text text-darken-2">35096 Weimar (Lahn)</span><br />
16       <span class="grey-text text-darken-2">Deutschland</span><br />
17       <span class="grey-text text-darken-2">+49 6426 8989-022</span><br />
18       <span class="grey-text text-darken-2">support@greenhats.com</span><br />
19       <span class="grey-text text-darken-2">www.greenhats.com</span><br />
20     </div>
21   </div>
22   <div class="row">
23     <div class="col s3 offset-s7">
24       <p>
25         <span class="grey-text text-darken-2">Datum</span><br />
26         <span class="grey-text text-darken-2">Kundennummer</span><br />
27         <span class="grey-text text-darken-2">Auftrag</span><br />
28         <% if @invoice.customer_reference -%>
29           <span class="grey-text text-darken-2">Referenz</span><br />
30         <% end -%>
31         <span class="grey-text text-darken-2">Rechnungsnummer</span><br />
32       </p>
33     </div>
34     <div class="col s2 right-align">
35       <p>
36         <span><%= @invoice.invoice_date %></span><br />
37         <span><%= @invoice.customer_number %></span><br />
38         <span><%= @invoice.job_number %></span><br />
39         <% if @invoice.customer_reference -%>
40           <span><%= @invoice.customer_reference %></span><br />
41         <% end -%>
42         <span class="strong"><%= @invoice.number %></span>
43       </p>
44     </div>
45   </div>
46   <div class="row">
47     <div class="col s12">
48       <h3>Rechnung</h3>
49       <% if @invoice.onetime? %>
50         <p>Einmalige Rechnung</p>
51       <% else %>
52         <p>Abrechnungszeitraum: <%= "#{@invoice.period_strings.first} - #{@invoice.period_strings.last}" %></p>
53       <% end %>
54       <p>Sehr geehrte Damen und Herren,<br />wir danken für Ihr Vertrauen und erlauben uns wie folgt Rechnung zu stellen.</p>

```

```

55     <table>
56     <thead>
57     <tr>
58         <th>Anzahl</th>
59         <th>Artikel/Beschreibung</th>
60         <th class="right-align">Einzelpreis</th>
61         <th class="right-align">Gesamtpreis</th>
62     </tr>
63 </thead>
64 <tbody>
65     <% @invoice.invoice_positions.each do |ip| -%>
66     <tr>
67         <td><%= ip.amount %></td>
68         <td><%= ip.description %></td>
69         <td class="right-align"><%= ip.price.format %> €</td>
70         <td class="right-align"><%= (ip.price * ip.amount).format %> €</td>
71     </tr>
72     <% end -%>
73 </tbody>
74 </table>
75 </div>
76 <div class="col s4 offset-s8">
77     <table class="table-no-last-line">
78     <tr>
79         <td>Nettobetrag</td>
80         <td class="right-align"><%= @invoice.price.format %> €</td>
81     </tr>
82     <tr>
83         <td>Umsatzsteuer <%= @invoice.vat_display -%> %</td>
84         <td class="right-align"><%= @invoice.vat.format %> €</td>
85     </tr>
86     <tr>
87         <td class="strong">Rechnungsbetrag</td>
88         <td class="right-align strong"><%= @invoice.with_vat.format %> €</td>
89     </tr>
90 </table>
91 </div>
92
93 </div>
94 <div class="row">
95     <p>Zahlungsbedingung: <%= @invoice.due_days %> Tage Netto bis zum <%= @invoice.due %></p>
96 </div>
97 <div class="row footer">
98     <div class="col s4">
99         <span>greenhats<sup class="smallsuper">®</sup> GmbH</span><br />
100        <span>Buchenweg 22</span><br />
101        <span>35096 Weimar (Lahn)</span><br />
102    </div>
103    <div class="col s4">
104        <span>Bankverbindung</span><br />
105        <span>IBAN: DE26 5335 0000 0000 1057 16</span><br />
106        <span>BIC: HELADEF1MAR</span><br />
107    </div>
108    <div class="col s4">
109        <span>Handelsregister Marburg</span><br />
110        <span>Ust.-Nr.: DE310205797</span><br />
111        <span>St.-Nr.: 2023480903</span><br />
112    </div>

```

Abbildung 11: Screenshot HTML-Template

10.1 Quellenverzeichnis

IAD GmbH:

<https://www.iad.de/>

evait security GmbH:

<https://www.evait.de/>

greenhats® GmbH:

<https://www.greenhats.com/>

Prawn:

<https://github.com/prawnpdf/prawn>

<https://prawnpdf.org/docs/0.11.1/index.html>

Ruby:

<https://www.ruby-lang.org/de/>

Ruby on Rails:

<https://rubyonrails.org/>

Athenapdf:

<https://github.com/arachnys/athenapdf>

10.3 Quellcode

10.3.1 Vollständiger Source-Code der Funktion

```

1 invoices_without_pdf = Invoice.ft_joins(:invoice_file_attachment)
2   .group(:id)
3   .having('COUNT(active_storage_attachments) = 0') # .with_attached_invoice_file
4 invoices_without_pdf.each do |invoice|
5
6   # Neues PDF Dokument erstellen und Abstände festlegen
7   pdf = Prawn::Document.new(page_size: "A4", left_margin: 1.4.send('cm'), right_margin: 1.5.send('cm'), top_margin: 1.6.send('cm'),
8     bottom_margin: 1.5.send('cm')) do |pdf|
9
10    # Open Sans Font, Größe und Zeilenabstand festlegen
11    pdf.font_families.update('Open Sans' => {
12      :normal => "#{Rails.root}/app/assets/fonts/OpenSans/mem5YaGs126MiZpBA-UN-r8OUuhp.ttf",
13      :bold => "#{Rails.root}/app/assets/fonts/OpenSans/mem5YaGs126MiZpBA-UNr7g0Uuhp.ttf"
14    })
15    pdf.font "Open Sans", size: 10
16    pdf.default_leading 3.4
17
18    pdf.svg '<?xml version="1.0" encoding="UTF-8" standalone="no"?> <!-- Created with Inkscape (http://www.inkscape.org/) --><svg
19      xmlns:dc="http://purl.org/dc/elements/1.1/" xmlns:cc="http://creativecommons.org/ns#" xmlns:rdf="http://www.w3.org/1999/02/
20      22-rdf-syntax-ns#" xmlns:svg="http://www.w3.org/2000/svg" xmlns="http://www.w3.org/2000/svg" xmlns:sodipodi="http://sodipodi.sourceforge.
21      net/DTD/sodipodi-0.dtd" xmlns:inkscape="http://www.inkscape.org/namespaces/inkscape" version="1.1" id="svg2" xml:space="preserve"
22      width="809.38269" height="197.81467" viewBox="0 0 809.38269 197.81467"
23      sodipodi:docname="logo_horizontal-1b2efec7b1e0d88f6d374259322426a695ca2fafb909635169b364f7dabb62a4.svg" inkscape:version="0.92.4
24      5da689c313, 2019-01-14" inkscape:export-xdpi="474.43564" inkscape:export-ydpi="474.43564"><metadata id="metadata8"><rdf:RDF><cc:Work
25      rdf:about=""><dc:format>image/svg+xml</dc:format><dc:type rdf:resource="http://purl.org/dc/dcmitype/StillImage" /><dc:title /></cc:Work></
26      rdf:RDF></metadata><defs id="defs6"><clipPath clipPathUnits="userSpaceOnUse" id="clipPath18"><path d="M 0,148.361 H 607.037 V 0 H 0 Z"
27      id="path16" inkscape:connector-curvature="0" /></clipPath></defs><sodipodi:namedview pagecolor="#ffffff" bordercolor="#666666"
28      borderopacity="1" objecttolerance="10" gridtolerance="10" guidetolerance="10" inkscape:pageopacity="0" inkscape:pageshadow="2"
29      inkscape:window-width="1920" inkscape:window-height="1080" id="namedview4" showgrid="false" inkscape:zoom="1.2146778" inkscape:cx="407.
30      18617" inkscape:cy="99.09517" inkscape:window-x="0" inkscape:window-y="0" inkscape:window-maximized="1"
31      inkscape:current-layer="g4931" /><g id="g10" inkscape:groupmode="layer" inkscape:label="horizontal" transform="matrix(1.3333333,0,0,-1.
32
33
34
35
36   # Absendeadresse festlegen und positionieren
37   pdf.move_down 0.7.send('cm')
38   pdf.transparent(0.6) {pdf.text "greenhats<sup>®</sup> GmbH - Buchenweg 22 - 35096 Weimar (Lahn)", size:8, width: 100, inline_format: true}
39
40   # Kundendaten einfügen und positionieren
41   pdf.move_down 0.3.send('cm')
42   invoice.recipient.each do |recipient_row|
43     pdf.transparent(0.6) {pdf.text recipient_row}
44   end
45
46   # greenhats Info einfügen und positionieren
47   pdf.transparent(0.9) {pdf.text_box "<color rgb='3fb079'>greenhats<sup>®</sup> GmbH</color>", inline_format: true, size: 9, at: [414,740]}
48   pdf.transparent(0.6) {pdf.text_box "Buchenweg 22 \n 35096 Weimar (Lahn) \n Deutschland", size: 8, at:[410,715], align: :right, width: 80}
49   pdf.transparent(0.6) {pdf.text_box "+49 6426 8989-022\n support@greenhats.com \n www.greenhats.com", size: 8, at:[390,673], align: :right,
50     width: 100}
51
52   # Kundendatenbezeichnung einfügen und positionieren
53   pdf.transparent(0.6) {pdf.text_box ["Datum", "Kundennummer", "Auftrag", "Referenz", "Rechnungsnummer"].reject{|d|d == "Referenz" && !invoice.
54     customer_reference}.join("\n
55     "), size: 10, at: [290,580], width: 100}
56
57   # Kundendaten einfügen und positionieren
58   pdf.transparent(0.6) {pdf.text_box ["#{invoice.invoice_date}", "#{invoice.customer_number}", "#{invoice.job_number}", "#{invoice.
59     customer_reference}", "<b>#{invoice.number}</b>"].reject{|d|d == "#{invoice.customer_reference}" && !invoice.customer_reference}.join("\n

```

```

61 # RechnungsText einfügen und positionieren
62 pdf.move_down 5.send('cm')
63 pdf.transparent(0.8) {pdf.text "Rechnung", size: 16}
64
65 # Test ob einmalige Rechnung
66 pdf.move_down 0.5.send('cm')
67 if invoice.onetime? then
68 | pdf.transparent(0.8) {pdf.text "Einmalige Rechnung", size: 10}
69 else
70 | pdf.transparent(0.6) {pdf.text "Abrechnungszeitraum: #{invoice.period_strings.first} - #{invoice.period_strings.last}"}
71 end
72
73 # Rechnung Text einfügen und positionieren
74 pdf.move_down 0.5.send('cm')
75 pdf.transparent(0.6) {pdf.text "Sehr geehrte Damen und Herren,
76 \ wir danken für ihr Vertrauen und erlauben uns wie folgt Rechnung zu stellen."}
77 pdf.move_down 1.send('cm')
78 pdf.text "Anzahl" , size: 9, style: :bold
79 pdf.draw_text "Artikel/Beschreibung", at: [65, 348], size: 9, style: :bold
80 pdf.draw_text "Einzelpreis", at: [370, 348], size: 9, style: :bold
81 pdf.draw_text "Gesamtpreis", at: [460, 348], size: 9, style: :bold
82
83 # Horizontal Line
84 pdf.move_down 0.3.send('cm')
85 pdf.transparent(0.1) {pdf.stroke_horizontal_line 0, 515, at: 338}
86
87
88 # rechnungspositionen einfügen und iterieren falls mehrere Positionen vorhanden sind
89 invoice.invoice_positions.each_with_index do |ip, i|
90 | pdf.text_box "#{ip.amount}", at: [0, 332 -20 * i ], size: 9
91 | pdf.text_box "#{ip.description}", at: [65, 332 -20 * i ], width: 300, size: 9, height: 10, :overflow => :shrink_to_fit
92 | pdf.text_box "#{ip.price.format} €", at: [360, 332 -20 * i ], size: 9, align: :right, width: 60
93 | pdf.text_box "#{(ip.price * ip.amount).format} €", at: [455, 332 -20 * i ], size: 9, align: :right, width: 60
94 | pdf.transparent(0.1) {pdf.stroke_horizontal_line 0, 515, at: 320 -20 * i }
95 | pdf.move_cursor_to (332 -20 * (i + 1))
96 end
97
98 # Cursor Position speichern
99 cursor_position = pdf.cursor
100
101 # Nettobetrag/Umsatzsteuer ect einfügen und positionieren
102 pdf.text_box "Nettobetrag" , at: [350, cursor_position ], size: 9
103 pdf.transparent(0.1) {pdf.stroke_horizontal_line 350, 515, at: cursor_position -15 }
104 pdf.draw_text "Umsatzsteuer #{invoice.vat_display} %", at: [350, cursor_position -27 ], size: 9
105 pdf.transparent(0.1) {pdf.stroke_horizontal_line 350, 515, at: cursor_position -32}
106 pdf.draw_text "Rechnungsbetrag" , at: [350, cursor_position -45 ], size: 9, style: :bold
107
108 # Preise einfügen und positionieren
109 pdf.text_box "#{invoice.price.format} €" , at: [455, cursor_position], size: 9, align: :right, width: 60
110 pdf.text_box "#{invoice.vat.format} €", at: [455, cursor_position -20 ], size: 9, align: :right, width: 60
111 pdf.text_box "#{invoice.with_vat.format} €", at: [455, cursor_position -39 ], size: 9, align: :right, width: 60, style: :bold
112
113 # Zahlungsbedingung einfügen und positionieren
114 pdf.move_down 2.5.send('cm')
115 pdf.transparent(0.6) {pdf.text "Zahlungsbedingung: #{invoice.due_days} Tage Netto bis zum #{invoice.due}"}
116
117 # Fußzeile Greenhats_info einfügen
118 pdf.text_box "greenhats<sup>®</sup> GmbH\n Buchenweg 22\n 35096 Weimar (Lahn)" , size: 8, at:[5,34],inline_format: true
119 pdf.text_box "Bankverbindung\n IBAN: DE26 5335 0000 0000 1057 16\n BIC: HELADEF1MAR" , size: 8, at:[180,34] , width: 200, height: 2.send
('cm')
120 pdf.text_box "Handelsregister Marburg\n Ust.-Nr.: DE310205797\n St.-Nr.: 2023480903" , size: 8, at:[390,34] , width: 200, height: 2.send
('cm')
121
122 end
123
124 begin
125 | io = StringIO.new pdf.render
126 | invoice.invoice_file.attach(io: io, filename: "invoice_#{invoice.number}.pdf")
127 end
128 end
129 end
130 end

```

Abbildung 12: Screenshot des Funktion Quellcodes