

Bachelorthesis

Dennis Oestermann

Entwicklung eines Softwaretools zur (teil-)automatischen Erstellung von Produktionsberichten für Photovoltaikanlagen unter Nutzung heterogener Datenschnittstellen

Fakultät Technik und Informatik
Department Informations- und Elektrotechnik

Faculty of Computer Science and Engineering
Department of Information and Electrical Engineering

Dennis Oestermann

Entwicklung eines Softwaretools zur
(teil-)automatischen Erstellung von
Produktionsberichten für Photovoltaikanlagen unter
Nutzung heterogener Datenschnittstellen

Bachelorarbeit eingereicht im Rahmen der Bachelorprüfung
im Studiengang *Bachelor of Science -
Regenerative Energiesysteme und Energiemanagement -
Elektro- und Informationstechnik*
am Department Informations- und Elektrotechnik
der Fakultät Technik und Informatik
der Hochschule für Angewandte Wissenschaften Hamburg

Betreuender Prüfer: Prof. Dr. Sven Wanser - HAW Hamburg
Zweitgutachter: Prof. Dr. Sebastian Rohjans - Jade Hochschule Oldenburg

Eingereicht am: 16. April 2020

Dennis Oestermann

Thema der Arbeit

Entwicklung eines Softwaretools zur (teil-)automatischen Erstellung von Produktionsberichten für Photovoltaikanlagen unter Nutzung heterogener Datenschnittstellen

Stichworte

Automatisierung, Monitoring, Python, Excel, Reporting, Solarenergie, Solaranlagen, Betriebsführung

Kurzzusammenfassung

Um die Produktion von erneuerbaren Energieanlagen effizient überwachen zu können, ist das regelmäßige Erstellen von Produktionsberichten unerlässlich. Die Nutzung von Softwaresystemen zur (teil-)automatischen Erstellung von Produktionsberichten ist dabei sinnvoll, um den Arbeitsaufwand bei der Erstellung von Produktionsberichten möglichst gering zu halten. Ziel dieser Thesis soll die Entwicklung eines Softwaretools sein, welches eine effiziente und herstellerunabhängige Überwachung der Produktionsdaten von Solaranlagen ermöglicht.

Title of Thesis

Development of a software tool for the (semi-)automatic creation of production reports of photovoltaic plants, using heterogeneous data interfaces

Keywords

automation, monitoring, Python, Excel, reporting, solar energy, solar plants, technical management

Abstract

In order to efficiently track the production of renewable energy plants, the continuous creation of production reports is essential. The use of software systems for the (semi-) automatic generation of production reports is an effective way to keep the workload for the creation of production reports as low as possible. The aim of this thesis is the development of a software tool, which allows efficient and manufacturer-independent monitoring of the production data of solar plants.

Inhaltsverzeichnis

Abbildungsverzeichnis	v
Tabellenverzeichnis	vii
Abkürzungen	viii
1 Einleitung	1
1.1 Motivation und Problemstellung	1
1.2 Zielsetzung	2
1.3 Aufbau der Arbeit	3
2 Grundlagen und Status quo	5
2.1 Datenverarbeitung in Photovoltaikanlagen	5
2.2 Schnittstellen	6
2.2.1 Application Programming Interface (API)	6
2.2.2 File Transfer Protocol (FTP)	7
2.3 Programmiersprache Python	8
2.4 Produktionsauswertung von Photovoltaikanlagen	9
2.5 Status quo	9
3 Anforderungsanalyse	13
3.1 Funktionale Anforderungen	14
3.2 Nicht funktionale Anforderungen	14
4 Design und Entwurf	16
4.1 Datenbank für Betriebsdaten	16
4.2 Berichtsstruktur	17
4.3 Softwarearchitektur	18

5	Umsetzung und Entwicklung	21
5.1	Vorbereitung und Softwareablauf	21
5.1.1	Vorbereitung	21
5.1.2	Starten des Tools	21
5.1.3	Abruf der Daten vom FTP-Server	22
5.1.4	Abruf der Daten unter Nutzung der API	24
5.2	Verarbeiten der Daten im XLSX-Format	26
5.3	Ausgabe des Produktionsberichtes im PDF-Format	27
6	Evaluation	29
7	Schlussbetrachtung	40
7.1	Zusammenfassung	40
7.2	Fazit	41
7.2.1	Mockup	42
	Literaturverzeichnis	47
A	Digitaler Anhang	49
A.1	Formelles	49
A.2	Softwarekomponenten	49
	Selbstständigkeitserklärung	50

Abbildungsverzeichnis

1.1	Schematische Abbildung des zukünftigen Ablaufs der softwaregestützten Berichtserstellung	3
2.1	Ablauf der Datenverarbeitung innerhalb von Photovoltaikanlagen	6
2.2	Ablauf des Bezugs einer JSON Datei von einer Datenbank unter Verwendung einer REST API	7
2.3	Schematische Abbildung einer Verbindung zwischen einem Client und einem FTP Server	8
2.4	Ablauf der Erstellung eines Produktionsberichtes für mehrere Anlagen mit unterschiedlichen Monitoringsystemen und ohne unterstützende Software .	10
2.5	Die Prognosedaten aus den vor Anlagenbau erstellten Ertragsprognosen sind in einer zentralen Datenbank hinterlegt	11
2.6	Die Produktionsdaten der Anlage werden laufend in der Tabelle festgehalten, was dem Tracking der Anlagenperformance dient	12
4.1	Ermittlung des aktuellen Monats innerhalb der Datenbank	17
4.2	Ermittlung der Produktion des aktuellen Monats innerhalb der Datenbank	18
4.3	Architektur der entwickelten Python Software	20
5.1	Der Softwareablauf wird mit Hilfe eines Sequenzdiagramms beschrieben .	22
5.2	Auszug aus den vom FTP-Server bezogenen Daten	24
5.3	Auszug aus den durch Nutzung der API bezogenen Daten	25
6.1	Feedback zur bereits durchgeführten oder noch notwendigen Installation von Python	35
6.2	Feedback zur Frage ob Python mit Hilfe der zur Verfügung gestellten Anleitung installiert werden konnte	35
6.3	Feedback zu Schwierigkeiten bei der Installation von Python	36

6.4	Feedback zur Frage ob die Installation der Bibliotheken mit Hilfe der Batch-Datei gelungen ist	36
6.5	Feedback zu der Installation der Bibliotheken	36
6.6	Feedback zur Berichtserstellung mit Hilfe des zur Verfügung gestellten Batch-Files	37
6.7	Feedback zur Frage ob die Nutzer auch zukünftig mit dem Tool arbeiten werden	38
6.8	Darstellung der Verbesserungsvorschläge der Tester	38
7.1	Mockup einer grafischen Nutzeroberfläche welche nach dem Start der Software angezeigt wird	43
7.2	Mockup einer grafischen Nutzeroberfläche welche nach Eingabe der Zugangsdaten angezeigt wird	44
7.3	Mockup einer grafischen Nutzeroberfläche mit Auswahlmöglichkeiten für den Nutzer	45
7.4	Mockup einer grafischen Nutzeroberfläche welche nach Fertigstellung des Berichtes angezeigt wird	46

Tabellenverzeichnis

6.1	Evaluation der funktionalen Anforderungen	31
6.2	Evaluation der nicht funktionalen Anforderungen	33
6.3	Evaluation des Berichtes	34

Abkürzungen

Abb Abbildung.

API Application Programming Interface.

BRX Anforderung (Requirement) an den Bericht.

CFG Configuration.

CPU Central Processing Unit.

CSV Comma Separated Value.

DIN Deutsches Institut für Normung.

EEA Erneuerbare-Energien-Anlagen.

EEG Erneuerbare Energien Gesetz.

FTP File Transfer Protocol.

GB Gigabyte.

GUI Graphical User Interface.

HAW Hochschule für Angewandte Wissenschaften.

IDE Integrated Development Environment.

IKT Informations- und Kommunikationstechnik.

IP Internet Protocol.

JSON JavaScript Object Notation.

kWh Kilowattstunden.

Mrd Milliarden.

MS Microsoft.

PDF Portable Document Format.

PV Photovoltaik.

RAM random-access memory.

REST Representational State Transfer.

RFX Funktionale Anforderung mit Nummerncode.

RNFX Nicht funktionale Anforderung mit Nummerncode.

URL Universal Resource Locator.

XLSX Excel Microsoft Office Open XML Format Spreadsheet file.

1 Einleitung

Im ersten Schritt wird erläutert, welche Problemstellung vorliegt und welche Zielsetzung sich daraus ergibt. Anschließend wird das konkrete Ziel der Arbeit beschrieben und verdeutlicht wie die Arbeit und damit auch der Lösungsweg strukturiert ist.

1.1 Motivation und Problemstellung

Die globale Erwärmung stellt eine der größten globalen Herausforderungen unseres Zeitalters dar [13]. Die Verminderung von Treibhausgasemissionen zur Erreichung langfristiger Klimaziele wird umfangreiche gesellschaftliche Transformationen erfordern [6]. Die erneuerbaren Energien leisten dabei den mit Abstand wichtigsten Beitrag [12]. Während im Jahr 2000 etwa 9,7 Mrd. Kilowattstunden (kWh) aus Wind- und eine zu vernachlässigende Menge Energie aus Photovoltaik gewonnen wurde, so waren es im Jahr 2019 125,9 Mrd. kWh aus Wind- beziehungsweise 46,4 Mrd. kWh aus Photovoltaikanlagen [12]. Als Instrument zur Förderung des Stroms aus erneuerbaren Energien wurde deshalb das Erneuerbare-Energien-Gesetz (EEG) [1] erlassen, welches im Jahr 2000 erstmals in Kraft getreten ist und die Entwicklung der erneuerbaren Energien regeln soll.

Entsprechend der Entwicklung der erneuerbaren Energien auf dem deutschen Strommarkt ändern sich auch die Marktbedingungen für Erneuerbare-Energien-Anlagen (EEA), was wiederum zu einer zwingend notwendigen Weiterentwicklung des EEG und damit zum Einsatz von Regulierungsmechanismen führt.

Die letzten Revisionen des EEG und die darin enthaltenen Vorgaben und Mechanismen mindern die Vergütung von Photovoltaikanlagen stetig [2], was zu einem enormen Leistungsdruck auf Seiten der Anlagenbetreiber führt. Um den effektiven Betrieb der Systeme sicher zu stellen, müssen die Anlagen ständig überwacht, mögliche Störungen

behoben und die Produktion und Verfügbarkeiten der Anlagen ausgewertet werden. Um den Betreibern das Monitoring zu ermöglichen, stellen Anlagenhersteller häufig eigens entwickelte Monitoringsysteme zur Verfügung, mit deren Hilfe die Produktion dokumentiert und Störungen angezeigt werden können. Zwar reichen diese Monitoringsysteme für Betreiber von Einzelanlagen meist aus, stellen aber für größere Investoren und Betreiber mehrerer Anlagen von unterschiedlichen Herstellern keine effektive Lösung dar. Die Auswertung der Betriebsdaten ist für Anlagenbetreiber unerlässlich, um Vergütungsverluste möglichst gering zu halten. Zudem besteht häufig sowohl für private als auch für gewerbliche Betreiber eine Berichtspflicht gegenüber Kreditinstituten, welche ebenfalls ein Interesse an dem effektiven Betrieb der Anlagen und somit an der gesicherten Rückzahlung eventueller Kreditraten durch die Vergütung der erzeugten Energie haben. Diese Raten basieren häufig auf Ertragsprognosen, welche vor dem Bau der Anlagen erstellt wurden und die Produktionsziele der jeweiligen Anlage vorgeben. Da Kreditraten dieser Art häufig mit Hilfe der erzielten Einspeisevergütung zurückgezahlt werden, ist das Einhalten dieser Zielvorgaben für den Anlagenbetreiber unerlässlich. Von unterschiedlichen Entwicklern werden deshalb Softwaresysteme angeboten, welche in der Lage sind Erzeugungsanlagen unterschiedlicher Hersteller einzubinden und zu überwachen. Diese Systeme sind allerdings häufig sowohl mit einmaligen Kosten für die Einbindung der Anlagen als auch mit laufenden Kosten verbunden, welche gerade von Betreibern weniger kleinerer Anlagen - welche ohnehin unter finanziellem Druck stehen - vermieden werden sollen.

1.2 Zielsetzung

Ziel dieser Arbeit ist die Entwicklung eines Softwaretools, welches den Prozess der Produktionsauswertung von Photovoltaikanlagen vereinfacht und repetitive Aufgaben bei der Erstellung von Produktionsberichten automatisiert. Abbildung 1.1 zeigt den schematischen Ablauf der Berichtserstellung mit Hilfe des zu entwickelnden Softwaresystems. Der User interagiert dabei nur mit dem Softwaresystem, statt mit den zur Verfügung gestellten Systemen des Herstellers beziehungsweise deren Schnittstellen. Hierbei ist zu beachten, dass Photovoltaikanlagen unterschiedlicher Hersteller oft über unterschiedliche Schnittstellen und unterschiedliche Datenstrukturen verfügen. Dies hat zur Folge, dass die Software in der Lage sein muss, Daten aus heterogenen Schnittstellen und in unterschiedlichen Dateiformaten zu beziehen, um diese anschließend zu verarbeiten und dem Nutzer im gewünschten Format zur Verfügung zu stellen. Mit Hilfe des Softwaretools soll

dem Nutzer die Möglichkeit gegeben werden, die monatliche Produktion seiner Anlage zu überwachen, ohne Gebrauch von den diversen Monitoringsystemen der Hersteller zu machen, sich in die unterschiedlichen Systeme einzuloggen und Daten in unterschiedlichen Formatierungen zu beziehen.

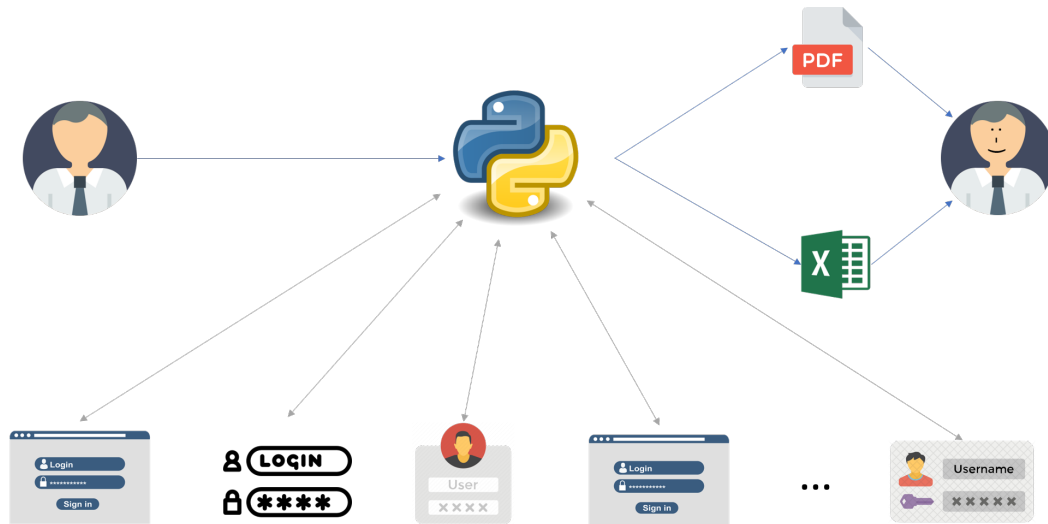


Abbildung 1.1: Schematische Abbildung des zukünftigen Ablaufs der softwaregestützten Berichtserstellung

1.3 Aufbau der Arbeit

Die Arbeit beginnt mit der Erläuterung der für die Thematik relevanten Grundlagen. Der grundsätzliche Aufbau von Informations- und Kommunikationstechnik (IKT) von Photovoltaikanlagen wird beschrieben, um verständlich zu machen auf welche Art die Produktionsdaten bezogen werden. Anschließend wird beleuchtet, welche Datenschnittstellen mit Hilfe der Software angebunden werden sollen, wie diese Schnittstellen grundsätzlich aufgebaut sind und wie der entsprechende Datenfluss aussieht. Es wird außerdem erläutert, welche Parameter für eine Produktionsauswertung von Photovoltaikanlagen relevant sind und welche Referenzwerte gegeben sein sollten, um die erhaltenen Daten auswerten zu können. Daraufhin wird dem Leser beschrieben, wie die Erstellung eines einheitlichen Produktionsberichtes mit Hilfe der von den Herstellern zur Verfügung gestellten Tools aussehen kann und es wird deutlich gemacht, welche Vorteile eine softwareunterstützte Berichtserstellung dem Nutzer bietet. Zur Erstellung des Softwaredesigns müssen die

grundsätzlichen Anforderungen an die Software ermittelt und kategorisiert werden. Den Hauptteil der Arbeit bilden die darauf folgenden Kapitel 'Design und Entwurf' und 'Umsetzung und Entwicklung'. Unter Berücksichtigung der ermittelten Anforderungen wird die Softwarearchitektur und das Sequenzdiagramm entwickelt. Die Softwarearchitektur beschreibt grundlegend, welche Komponenten innerhalb des Softwaresystems miteinander interagieren und auf welche Art Informationen ausgetauscht werden. Mit Hilfe eines Ablaufdiagramms wird dabei anschaulich gezeigt, welche Vorgänge beim Ausführen der Software stattfinden und wie das gewünschte Ziel erreicht werden soll. In den daran anschließenden Kapiteln wird die Umsetzung des entwickelten Konzeptes erläutert und anschließend mit Hilfe eines Anwenderfeedbacks geprüft, inwiefern die zuvor ermittelten Anforderungen umgesetzt werden konnten. Im Fazit wird eine kritische Bewertung des Konzeptes und der entwickelten Software vorgenommen. Es wird geprüft, ob die an das Produkt gestellten Anforderungen erfüllt werden konnten und es werden Möglichkeiten aufgezeigt, wie das Produkt in Zukunft weiterentwickelt werden kann.

2 Grundlagen und Status quo

Dieser Teil der Arbeit beschreibt den grundlegenden Ablauf der Datenverarbeitung in Photovoltaikanlagen und Funktion und Aufbau der genutzten Datenschnittstellen. Anschließend wird erläutert, welche Betriebsdaten von den genannten Schnittstellen bezogen werden und wie diese Daten verarbeitet werden, um den Produktionsbericht zu erstellen. Es wird angenommen, dass Leser dieser Arbeit über grundlegendes technisches Wissen verfügen.

2.1 Datenverarbeitung in Photovoltaikanlagen

Um sowohl die Produktion als auch den Status der Anlage aus der Ferne überwachen zu können, muss innerhalb der Anlage eine Weitergabe und Verarbeitung der Daten stattfinden und dem Nutzer muss eine Schnittstelle geboten werden, um auf die jeweiligen Daten zuzugreifen. Abbildung 2.1 zeigt exemplarisch, welche Komponenten am Datenfluss von Photovoltaikanlagen beteiligt sind. Die Module von Photovoltaikanlagen werden in Strings zusammengeschaltet und entsprechend ihrer und der Leistung des jeweiligen Wechselrichters verbunden. Die Hauptfunktion der Wechselrichter besteht in der Umwandlung der von den Modulen erzeugten Gleichspannung in Wechselspannung. Moderne Wechselrichter sind allerdings auch mit digitalen Zählern ausgestattet, welche die produzierte Energie jedes Strings ermitteln und die Werte an den zentralen Datenlogger der Anlage übermitteln können. Der Datenfluss beginnt also beim Wechselrichter und endet innerhalb der Anlage beim Datenlogger. Sowohl die Statusmeldungen der Wechselrichter als auch die übermittelten Produktionsmengen jedes Modulstrings werden vom Datenlogger protokolliert. Der Anlagenbetreiber ist gemäß §10b EEG dazu verpflichtet, die Fernsteuerbarkeit seiner Anlage zu gewährleisten. Sowohl der Netzbetreiber als auch der jeweilige Direktvermarkter müssen Zugriff auf die Anlage haben, die Produktionsmengen auslesen und die Anlage gegebenenfalls abregeln können. Sollte vom Hersteller

selbst kein Monitoringtool angeboten werden, so ist damit dennoch sichergestellt, dass es eine Schnittstelle zum Auslesen der benötigten Daten geben muss.

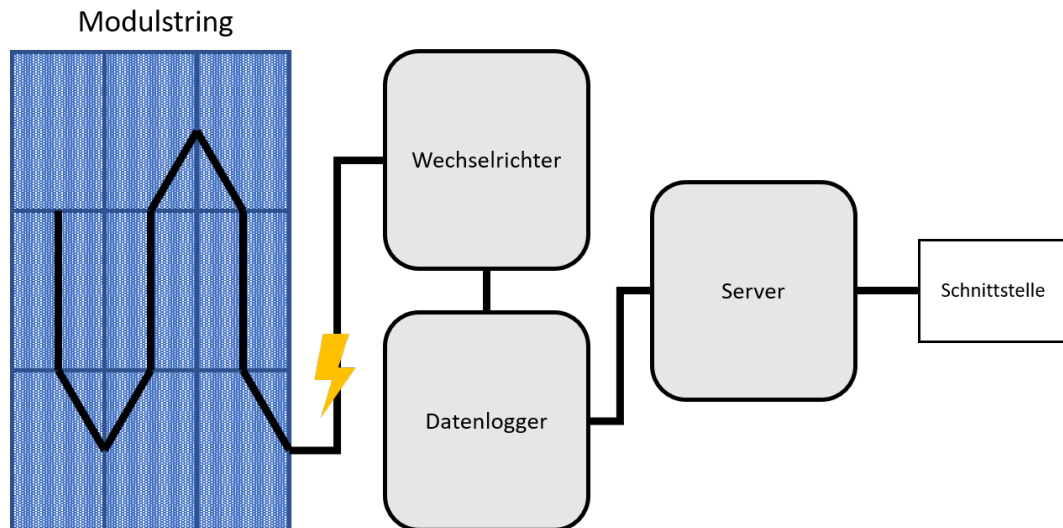


Abbildung 2.1: Ablauf der Datenverarbeitung innerhalb von Photovoltaikanlagen

2.2 Schnittstellen

Um sicher zu stellen, dass Anlagen mit heterogenen Schnittstellen angebunden werden können, werden zur Entwicklung der Software exemplarisch zwei Anlagen mit unterschiedlichen Schnittstellen eingebunden. Im Folgenden werden die verwendeten Schnittstellentypen und deren Vor- und Nachteile erläutert.

2.2.1 Application Programming Interface (API)

Bei dem Application Programming Interface (im Folgenden 'API') handelt es sich um eine Programmierschnittstelle, welche Programmen von einem Softwaresystem auf Quelltextebene zur Verfügung gestellt wird. Im Falle dieser Arbeit wird die vom Anlagenhersteller bereitgestellte REST API genutzt um die Anlagendaten abzurufen, d.h. es findet ausschließlich ein lesender Zugriff statt. REST steht für Representational State Transfer [14] und stellt einen Architekturstil dar, welcher zum Entwerfen von Netzwerkanwendungen entwickelt wurde [15]. Um auf die REST API zugreifen zu können,

wird ein API-Key benötigt, welcher im Fall der zu betrachtenden Anlage vom Hersteller zur Verfügung gestellt wird. Abbildung 2.2 zeigt den generellen Ablauf einer solchen Anfrage, welche mit einem 'Request' vom Client beginnt. Die Anfrage wird von der Programmierschnittstelle an die angebundene Datenbank weitergeleitet und die angefragten Daten werden von der API aus der Datenbank bezogen und an den Client zurückgegeben. Diese Schnittstelle ist speziell für die Anwendung in Softwaresystemen konzipiert, in der Regel sehr gut dokumentiert und entsprechend einfach einzubinden. Allerdings wird die Schnittstelle und die dazugehörigen Anmeldedaten in diesem Fall als kostenlose Dienstleistung vom Hersteller zur Verfügung gestellt. Sollte der Hersteller diese Dienstleistung einstellen, so wäre die API für das zu entwickelnde Softwaretool nicht mehr nutzbar und die Daten müssten auf anderem Wege akquiriert werden.

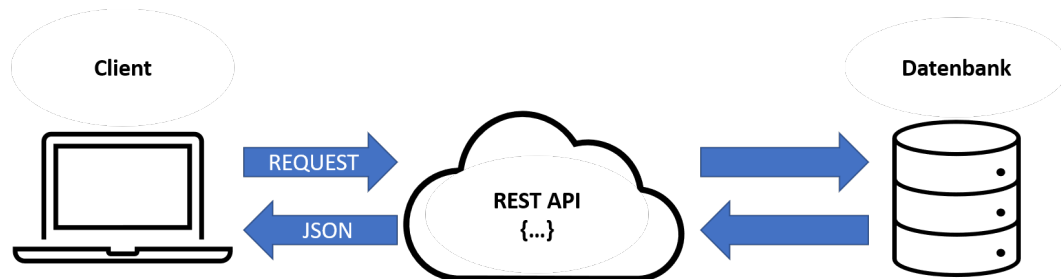


Abbildung 2.2: Ablauf des Bezugs einer JSON Datei von einer Datenbank unter Verwendung einer REST API

2.2.2 File Transfer Protocol (FTP)

Bei dem File Transfer Protocol (im Folgenden 'FTP') handelt es sich um ein Netzwerkprotokoll zur Übertragung von Daten über IP-Netzwerke [11]. In dieser Arbeit wird es genutzt, um Dateien vom Server zum Client zu übertragen, d.h. es findet ausschließlich ein lesender Zugriff statt. Da der Client mit diesem Verfahren einen direkten Zugriff auf den Server hat und ihm weitreichende Rechte eingeräumt werden, muss sich der User mit einem speziellen Benutzernamen und einem Passwort anmelden. Abbildung 2.3 zeigt, dass die Verbindung zwischen Client und Server in diesem Fall über zwei Ports ausgeführt wird. Port 21 dient der Kommunikation von Befehlen und Antworten zwischen Client und Server, während Port 20 dem Austausch von Daten dient.

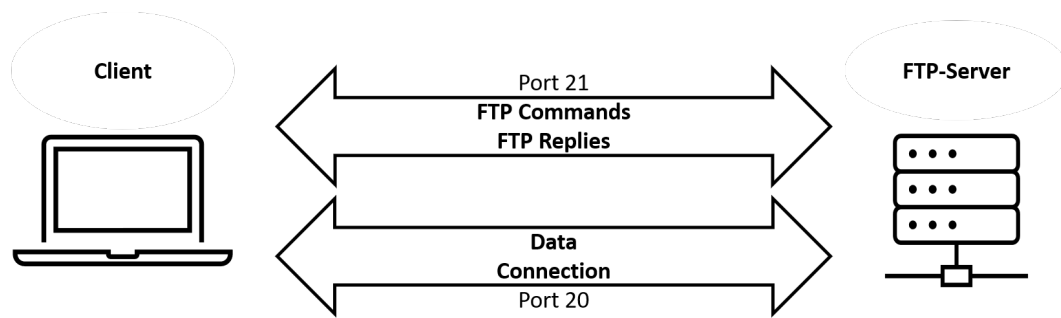


Abbildung 2.3: Schematische Abbildung einer Verbindung zwischen einem Client und einem FTP Server

2.3 Programmiersprache Python

Das angestrebte Produkt dieser Arbeit ist ein Softwaretool, welches dem User repetitive Aufgaben im Prozess der Berichterstellung abnimmt. Dazu gehören das Akquirieren von Daten unterschiedlicher Anlagen, das Verarbeiten dieser mit Hilfe gängiger Office-Anwendungen und das Aufbereiten der Daten zur Ausgabe eines Produktionsberichtes im PDF-Format. Die zeitliche Begrenzung der Arbeit und die diversen Aufgaben, welche von der zu erstellenden Software übernommen werden sollen, fordern daher eine Programmiersprache, welche möglichst multifunktional und gleichzeitig schnell erlernbar ist. Mit zahlreichen zur Verfügung gestellten Modulen, wie 'NumPy' [8] zur numerischen Programmierung oder 'pandas' [7] zur Verwaltung von Daten und deren Analyse, sowie 'Matplotlib' [5] zur Erstellung mathematischer Darstellungen aller Art, stellt Python eine universell anwendbare Programmiersprache dar. Neben den unterschiedlichen Modulen bietet Python eine im Vergleich zu vielen anderen Programmiersprachen intuitive Syntax. Die Bibliothek 'ftplib' dient der Steuerung des Datenaustausches zwischen FTP-Client und FTP-Server. Die Bibliothek 'json' ermöglicht den Bezug von Daten und das Handling des JSON-Datenformates, welches das am häufigsten verwendete Format zum Senden von API Requests und Responses ist [10]. Mit diesen und weiteren zusätzlichen Werkzeugen stellt Python eine hoch entwickelte Programmiersprache dar, welche wissenschaftliche und ingenieurstechnische Arbeiten ermöglicht und damit zum Erreichen des Ziels dieser Arbeit geeignet ist [9].

2.4 Produktionsauswertung von Photovoltaikanlagen

Vor dem Bau einer Photovoltaikanlage werden Ertragsprognosen erstellt, die als Grundlage für die Wirtschaftlichkeitsberechnung der Anlage dienen. Zur Erstellung dieser Prognosen werden die geplante installierte Gesamtleistung der Anlage, der spezifische Jahresertrag, die vertikale und horizontale Ausrichtung der Module und die Wetterdaten für den geplanten Standort der Anlage benötigt. Anhand dieser Daten kann die monatliche Produktion der Anlagen prognostiziert werden, welche in Verbindung mit der Vergütung die Grundlage für die Berechnung der Wirtschaftlichkeit der Anlage bildet. In der operativen Phase der Anlage liegt die Verantwortung für das Erreichen der Ertragsprognose beim Anlagenbetreiber, weshalb eine effiziente Überwachung der Anlage von großer Wichtigkeit ist. Gerade in der frühen Betriebsphase von Anlagen spielt der Vergleich der tatsächlichen Produktion mit den zuvor erstellten Prognosen eine große Rolle, da zu diesem Zeitpunkt keine realen Werte zum Vergleich vorliegen. Um die Produktion detaillierter bewerten zu können, müssen Daten über die tatsächlich eingespeiste Energie der Anlage und Daten über die Sonneneinstrahlung vorliegen, die von einem oder mehreren Einstrahlungssensoren ermittelt werden. Anhand dieser Daten kann nun eine erste Aussage darüber getroffen werden, ob eine eventuelle Abweichung von der Ertragsprognose auf eine verringerte Sonneneinstrahlung zurückzuführen ist. Liegt die Produktion der Anlage trotz Einhaltung der prognostizierten Sonneneinstrahlung unterhalb der Vorhersage, kann dies auf technische Probleme an der Anlage hindeuten, welche vom Betreiber behoben werden müssen um weitere Ertragsverluste zu vermeiden. Mit zunehmender Betriebsdauer ist ein Vergleich mit der Vorjahresproduktion sinnvoll, da periodisch auftretende Abweichungen von der Ertragsprognose auf Umweltfaktoren wie Verschattungen der Anlage hinweisen können, welche bei der Erstellung der Ertragsprognose möglicherweise nicht berücksichtigt wurden.

2.5 Status quo

Für Betreiber mehrerer Photovoltaikanlagen von unterschiedlichen Herstellern ist der Prozess der manuellen Produktionsüberwachung, je nach Anzahl der zu überwachenden Anlagen, im Vergleich zu einer softwareunterstützten Überwachung sehr aufwendig.

Sofern der Anlagenhersteller ein Monitoringssystem anbietet, muss der Betreiber sich, wie in Abb. 2.4 dargestellt, mit den jeweiligen Login-Daten in die unterschiedlichen Systeme

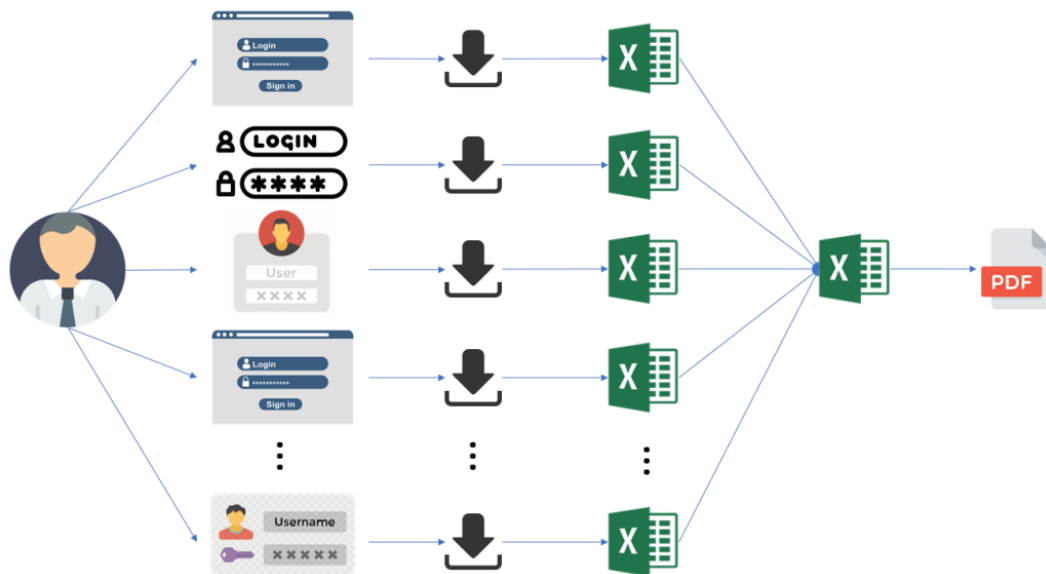


Abbildung 2.4: Ablauf der Erstellung eines Produktionsberichtes für mehrere Anlagen mit unterschiedlichen Monitoringsystemen und ohne unterstützende Software

einloggen, um Zugriff auf die Produktionsdaten der Anlagen zu erhalten. Zwar bieten die meisten Systeme die Möglichkeit Daten in einem Dateiformat herunterzuladen, welches sich mit gängigen Office Anwendungen verarbeiten lässt, allerdings sind die Daten nicht einheitlich formatiert und somit untereinander nicht kompatibel. Um die Daten zu zentralisieren, muss der Nutzer die benötigten Werte extrahieren und in eine separate Datenbank einpflegen. Da für die Auswertung der Produktionsdaten Prognosedaten notwendig sind, mit welchen die tatsächliche Produktion verglichen werden kann, werden diese zuvor aus den vorhandenen Ertragsprognosen der Anlagen extrahiert und ebenfalls in die Datenbank eingepflegt. Wie in Abbildung 2.5 dargestellt, werden im Falle der in dieser Arbeit betrachteten Anlagen, die Daten aus zwei unterschiedlichen Ertragsprognosen in der Datenbank hinterlegt. Die tatsächliche Produktion wird dann mit dem Mittelwert aus den beiden Prognosen verglichen.

Neben der prognostizierten Produktion wird auch die tatsächliche Energieproduktion der Anlagen laufend in der Datenbank festgehalten, wie in Abb. 2.6 zu sehen ist. Dies dient zum einen dazu, die Entwicklung der Produktion im laufenden Betriebsjahr zu protokollieren und ist zum anderen sinnvoll, um die monatliche Produktion mit der Produktion des selben Monats im Vorjahr bzw. den Vorjahren zu vergleichen. Mit steigendem Al-

	A	B	C	D	E
1					
2					
3	Monat	Prognose aus Ertragsgutachten 1 [kWh]	Prognose aus Ertragsgutachten 2 [kWh]	Ertragsmittelwert [kWh]	kumulierte Energie ans Netz [kWh]
4	Januar	13.300,00	14.600,00	13.950,00	13.950,00
5	Februar	24.100,00	25.764,00	24.932,00	38.882,00
6	März	57.900,00	54.285,00	56.092,50	94.974,50
7	April	87.400,00	87.124,00	87.262,00	182.236,50
8	Mai	102.800,00	100.459,00	101.629,50	283.866,00
9	Juni	99.800,00	102.193,00	100.996,50	384.862,50
10	Juli	100.200,00	101.091,00	100.645,50	485.508,00
11	August	87.700,00	85.103,00	86.401,50	571.909,50
12	September	60.900,00	66.443,00	63.671,50	635.581,00
13	Oktober	38.000,00	41.729,00	39.864,50	675.445,50
14	November	16.300,00	18.444,00	17.372,00	692.817,50
15	Dezember	9.300,00	11.617,00	10.458,50	703.276,00
16	Gesamt	697.700,00	708.852,00	703.276,00	

Abbildung 2.5: Die Prognosedaten aus den vor Anlagenbau erstellten Ertragsprognosen sind in einer zentralen Datenbank hinterlegt

ter der Anlagen gibt die Produktion aus den Vorjahren einen realistischen Rahmen vor, sodass die Anlagen laufend 'mit sich selbst verglichen' werden können. Mit diesem Vorgehen können auch Ertragsminderungen durch Verschmutzung oder später Degradation der Module festgestellt werden, um geeignete Maßnahmen einzuleiten.

Die Daten jeder Anlage werden in separaten Worksheets eingepflegt und innerhalb der Tabelle weiter verarbeitet. Ein finales Worksheet übernimmt die Daten und stellt die Ergebnisse übersichtlich und grafisch dar. Auf den konkreten Aufbau des auszugebenden Berichtes wird im Kapitel 'Design' eingegangen.

Monatliche Produktionswerte [kWh]							
Produktion 2021 [kWh]	Produktion kummuliert [kWh]	Monatsertrag laut Gutachten erreicht zu [%]	Jahresertrag laut Gutachten erreicht zu [%]	Produktion 2020 [kWh]	Produktion kummuliert [kWh]	Monatsertrag laut Gutachten erreicht zu [%]	Jahresertrag laut Gutachten erreicht zu [%]
8.038,80	8.038,80	57,63%	1,14%	10.550,30	10.550,30	79,33%	1,50%
23.128,00	31.166,80	92,76%	4,43%	21.004,10	31.554,40	87,15%	4,49%
47.923,00	79.089,80	85,44%	11,25%	63.427,10	94.981,50	109,55%	13,51%
				108.883,80	203.865,30	124,58%	28,99%
				109.221,80	313.087,10	106,25%	44,52%
				112.443,10	425.530,20	112,67%	60,51%
				93.434,10	518.964,30	93,25%	73,79%
				99.960,90	618.925,20	113,98%	88,01%
				78.010,60	696.935,80	128,10%	99,10%
				29.731,80	726.667,60	78,24%	103,33%
				15.077,40	741.745,00	92,50%	105,47%
				6.712,60	748.457,60	72,18%	106,42%
79.089,80				748.457,60			

Abbildung 2.6: Die Produktionsdaten der Anlage werden laufend in der Tabelle festgehalten, was dem Tracking der Anlagenperformance dient

3 Anforderungsanalyse

Zur Ermittlung der Anforderungen an die Software, wird die Aufgabenstellung analysiert und die darin enthaltenen Anforderungen interpretiert und extrahiert. Um die daraus entstandene Anforderungsliste zu vervollständigen wurde ein Brainstorming durchgeführt und die zusätzlichen Anforderungen wurden dem bisherigen Profil hinzugefügt. Anschließend wurden die Anforderungen in funktionale (Code RFX) und nicht funktionale Anforderungen (RNFX) unterteilt. Funktionale Anforderungen beschreiben die Funktionalitäten und das Verhalten einer Software. Nicht funktionale Anforderungen beschreiben Anforderungen an die Qualität der Software und Anforderungen welche durch gegebene Randbedingungen entstehen. [4]

Zur Übersichtlichkeit und zur Vereinfachung der Evaluation des Ergebnisses wurden die Anforderungen hinsichtlich ihrer Relevanz wie folgt gewichtet:

- **Muss-Anforderungen** beschreiben die Mindestanforderungen und damit die Kernfunktionen der Software. Diese Anforderungen sind zwingend zu erfüllen und werden daher dreifach gewichtet.
- **Soll-Anforderungen** beschreiben Merkmale und Funktionen der Software welche zwar wünschenswert, zum Erfüllen der Mindestanforderungen aber nicht zwingend notwendig sind. Sie beschreiben Erweiterungen der Kernsoftware und bieten bei erfolgreicher Umsetzung einen relevanten Mehrwert für den Anwender, weshalb sie zweifach gewertet werden.
- **Wird-Anforderungen** beschreiben optionale Funktionen der Software, welche zwar wünschenswert für die generelle Lösung der Problemstellung aber von niedriger Bedeutung sind. Diese Anforderungen werden aufgrund der geringen Relevanz einfach gewertet.

3.1 Funktionale Anforderungen

- RF1: Das Softwaretool wird ohne die zusätzliche Nutzung einer Python IDE unter Windows geöffnet [MUSS]
- RF2: Rohdaten werden in .csv oder .xlsx Format in einem festgelegten Ordner gespeichert [MUSS]
- RF3: Der User wird über die Fertigstellung des Berichtes optisch benachrichtigt [SOLL]
- RF4: Bericht wird im PDF-Format gespeichert [SOLL]
- RF5: Zum Erstellen des aktuellen Berichtes wird der vorangegangene Bericht geöffnet, aktualisiert und unter einem neuen Dateinamen gespeichert [MUSS]
- RF6 : Der vorangegangene Bericht wird in ein Archiv verschoben [SOLL]
- RF7: Der User hat die Möglichkeit einzelne Berichte pro Anlage oder einen Gesamtbericht zu erstellen [WIRD]
- RF8: Der Berichtszeitraum ist frei wählbar [WIRD]

3.2 Nicht funktionale Anforderungen

- RNF1: Bericht des aktuellen Monats soll sich mit maximal fünf Klicks erstellen lassen [MUSS]
- RNF2: Für die Integration neuer Anlagen in die Software ist ein Template innerhalb der Software hinterlegt, welches kopiert und um die entsprechenden Anlagendaten ergänzt werden kann, um neue Anlagen hinzuzufügen [WIRD]
- RNF3: Die Software ist mit gängiger Office-Hardware und dazugehörigen Betriebssystemen ausführbar [MUSS]
- RNF4: Die Software wird über eine grafische Benutzeroberfläche bedient [WIRD]
- RNF5: Dem User wird ein Kontakt zur Fehlermeldung zur Verfügung gestellt [SOLL]

- RNF6: Dem User wird eine Anleitung zur Verfügung gestellt, welche die Erstnutzung der Software erleichtert [MUSS]

Getestet wird die Software auf einem System mit folgenden Spezifikationen (auszugsweise aufgeführt):

- Betriebssystem: Microsoft Windows 10 Pro 10.0.19041 Build 19041
- Prozessor: Intel(R) Core(TM) i5-8265U CPU @ 1.60 GHz
- Installierter RAM: 8.00 GB (7,77 GB verwendbar)
- Systemtyp: 64-Bit-Betriebssystem
- Verbindung zum Internet
- Installiertes Microsoft Office 365 Paket mit MS Excel

Die Anforderungen an den Inhalt des fertigen Berichtes werden der Übersicht halber separat aufgeführt (Code BR1 bis BR9). Sie gehören ebenfalls zu den nicht funktionalen Anforderungen an die zu entwickelnde Software und sind im Einzelnen nochmals in 'MUSS', 'SOLL', und 'WIRD' kategorisiert worden.

Inhalt des Berichtes:

- BR1: Berichtszeitraum [MUSS]
- BR2: Projektname [MUSS]
- BR3: Produktion des aktuellen Monats in kWh [MUSS]
- BR4: Prognose erreicht zu X Prozent [MUSS]
- BR5: Verfügbarkeit in Prozent [WIRD]
- BR6: Einstrahlungsmenge in kWh/m² [WIRD]
- BR7: Produktionsdaten des Vorjahres [MUSS]
- BR8: Grafik monatliche Produktion im Vergleich zu Prognose [SOLL]
- BR9: Grafik kumulierte Produktion im Vergleich zu Prognose [SOLL]

4 Design und Entwurf

In diesem Kapitel wird erläutert, wie die Lösung aussehen soll und welche Entscheidungen bezüglich der Anforderungen getroffen wurden. Im ersten Schritt wird erläutert, wie der auszugebende Bericht strukturiert wird. Anschließend wird die Softwarearchitektur erläutert, welche auf Basis der Anforderungen entworfen wurde.

4.1 Datenbank für Betriebsdaten

Wie schon im Kapitel 'Grundlagen' dargelegt, werden die von den Anlagenschnittstellen bezogenen Daten in einer Datenbank im xlsx-Format weiterverarbeitet und für die Ausgabe eines Produktionsberichtes vorbereitet. Dies soll Nutzern ohne Programmierkenntnisse die Möglichkeit geben, die Daten auch außerhalb des Softwaretools zu nutzen, um gegebenenfalls Berichte mit abweichendem Design zu erstellen oder den Ausgabebericht anzupassen. Vor allem dient dies aber dem Zweck, die Anlagenproduktion auch außerhalb des aktuellen Berichtszeitraums analysieren zu können und die Anlagenperformance zu überwachen. Für jede zu betrachtende Anlage wird ein Arbeitsblatt in einer xlsx-Datei angelegt, in welchem die Daten aus den Ertragsprognosen der jeweiligen Anlage dokumentiert werden. Sofern mehr als eine Prognose vorliegt, wird der Mittelwert aus den prognostizierten Produktionsmengen pro Monat errechnet. Diese Daten dienen als Referenz für die Produktionsdaten, welche im laufenden Betrieb der Anlage ermittelt werden. Um die Produktionsdaten der jeweiligen Anlage vergleichen zu können, werden die ermittelten Produktionswerte in der Zeile des jeweiligen Monats eingetragen. Innerhalb der Datenbank wird die Prognose nun mit der tatsächlichen Produktionsmenge verglichen und das Ergebnis prozentual dargestellt. Dieses Verfahren wird sowohl für die monatlichen Produktionswerte als auch für das gesamte Kalenderjahr angewendet.

4.2 Berichtsstruktur

Um die Ergebnisse pro Anlage möglichst übersichtlich darzustellen, sollen die relevanten Daten sowohl numerisch als auch grafisch dargestellt werden und der Umfang eines Berichtes pro Anlage soll eine DIN A4 Seite nicht überschreiten. Auf die in den jeweiligen Arbeitsblättern ermittelten Werte und Ergebnisse wird von einem weiteren Arbeitsblatt mit dem Titel 'OUTPUT_BERICHT' zugegriffen, das die Ergebnisse darstellt und in dem das Design und Format des Ausgabeberichtes festgelegt werden. Das erste Element des Berichtes ist der Zeitraum, welcher von dem Bericht abgedeckt wird und welcher sich immer vom ersten Tag eines Monats bis zum vorangegangenen Tag erstreckt, da für die Berichtserstellung nur ganze Tage berücksichtigt werden. Der aktuelle Monat wird nicht durch den User eingetragen, sondern wie in Abb. 4.1 dargestellt von der Datenbank selbst ermittelt.

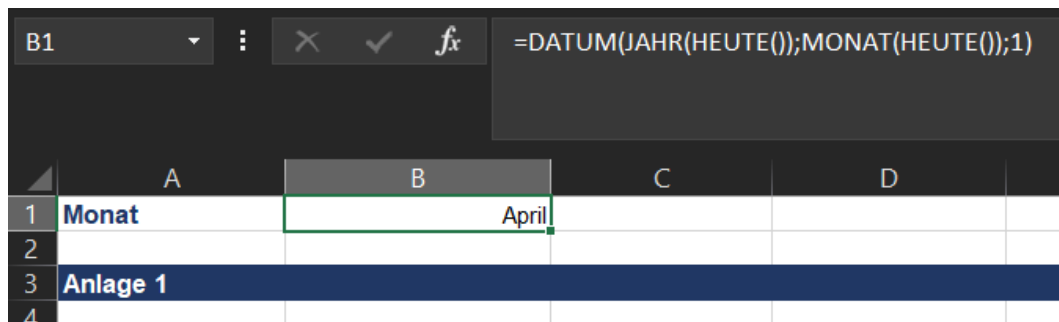


Abbildung 4.1: Ermittlung des aktuellen Monats innerhalb der Datenbank

Dies hat den Vorteil, dass der Nutzer beim Monatswechsel nicht manuell in die Datenbank eingreifen muss, um das aktuelle Datum einzutragen. Da die Berichte fortlaufend auf Basis des jeweils vorangegangenen Berichtes erstellt werden, stellt der letzte Bericht des Jahres gleichzeitig auch den Jahresbericht dar. Auf der linken Seite des Berichtes wird dem Nutzer die Produktion im aktuellen Monat, sowie der Vergleich zum selben Monat des Vorjahres dargestellt. Beide Werte werden wie in Abb. 4.2 gezeigt ebenfalls mit Hilfe des von der Datenbank ermittelten Datums und Verweisen auf die entsprechenden Arbeitsblätter von der Datenbank ermittelt.

Im nächsten Schritt wird die Produktion des aktuellen Jahres mit der Produktion des vergangenen Jahres verglichen, was vor allem am Ende des Betriebsjahres relevant ist und dem Nutzer langfristig aufzeigt ob die Anlagenperformance konstant ist oder auf-

The screenshot shows an Excel spreadsheet with the following data:

	A	B	C	D
1	Monat	April		
2				
3	Anlage 1			
4				
5				
6				
7		Produktion aktueller Monat [kWh]	Produktion aktueller Monat im Vorjahr [kWh]	Produktion des selben Monats im Vorjahr erreicht zu
8		3.293,00	108.883,80	3,02%
9				

Abbildung 4.2: Ermittlung der Produktion des aktuellen Monats innerhalb der Datenbank

grund von Moduldegradation oder anderen Gründen abnimmt. Im letzten Schritt wird ermittelt, zu wie viel Prozent die prognostizierte Produktionsmenge für den entsprechenden Monat und für das gesamte Jahr erreicht wurde. Um dem Nutzer mit jedem Bericht einen Überblick über die ganzjährige Anlagenperformance zu geben, werden sowohl die monatlichen Produktionswerte als auch die kumulierte Produktion der Anlage in zwei Grafiken mit den jeweiligen Prognosen und den Produktionsmengen der vergangenen Jahre verglichen.

4.3 Softwarearchitektur

Die Softwarearchitektur soll ausgehend von den Anforderungen festlegen, aus welchen wesentlichen Bestandteilen die Software besteht, ohne auf deren interne Details einzugehen [3]. Wie in Abbildung 4.3 gezeigt, wird bei der Software eine Unterteilung in Module vorgenommen, welche jeweils eine bestimmte Funktion der Software übernehmen. Die Methoden innerhalb der Module werden jeweils von dem Objekt 'reporting.py' aufgerufen. Der erste Vorgang des Objektes 'reporting.py', welcher in Abb. 5.1 zu sehen ist, dient der Deklaration der benötigten Variablen und dem Aufruf der benötigten Metho-

den, welche sich innerhalb der darunter angeordneten Objekte befinden. Da die Software in der Lage sein soll, Daten aus heterogenen Schnittstellen zu beziehen, wird für jede Schnittstelle und deren Behandlung ein separates Modul zur Verfügung gestellt, um die zukünftige Einbindung weiterer Anlagen mit gleichen Schnittstellen zu vereinfachen. Der Datenbezug vom Server der jeweiligen Anlagen wird durch den Aufruf einer Methode gestartet. Um auf den Server zugreifen zu können, werden anlagenspezifische Daten benötigt, welche sich in einer separaten JSON-Datei befinden. Dies hat den Vorteil, dass Anmeldedaten geändert werden können, ohne auf die Kernsoftware zuzugreifen. Weiterhin können die Daten vom Systemadministrator abgelegt und verwaltet werden, was die Sicherheit erhöht. Sollten zukünftig Anlagen mit weiteren Schnittstellentypen hinzugefügt werden, kann ein separates Python-Package angelegt werden ohne die bereits existierenden Objekte zu modifizieren. Die Verarbeitung der Daten hängt nicht von den unterschiedlichen Schnittstellen der Anlagen ab, weshalb neben dem Objekt `'datahandler.py'` keine weiteren Objekte angelegt werden müssen. Die Module `'xlsx_handler.py'` und `'pdf_handler.py'` verarbeiten die bezogenen Daten, geben dem Nutzer den aktuellen Produktionsbericht zurück und verschieben die jeweils vorangegangenen Daten in ein Archiv.

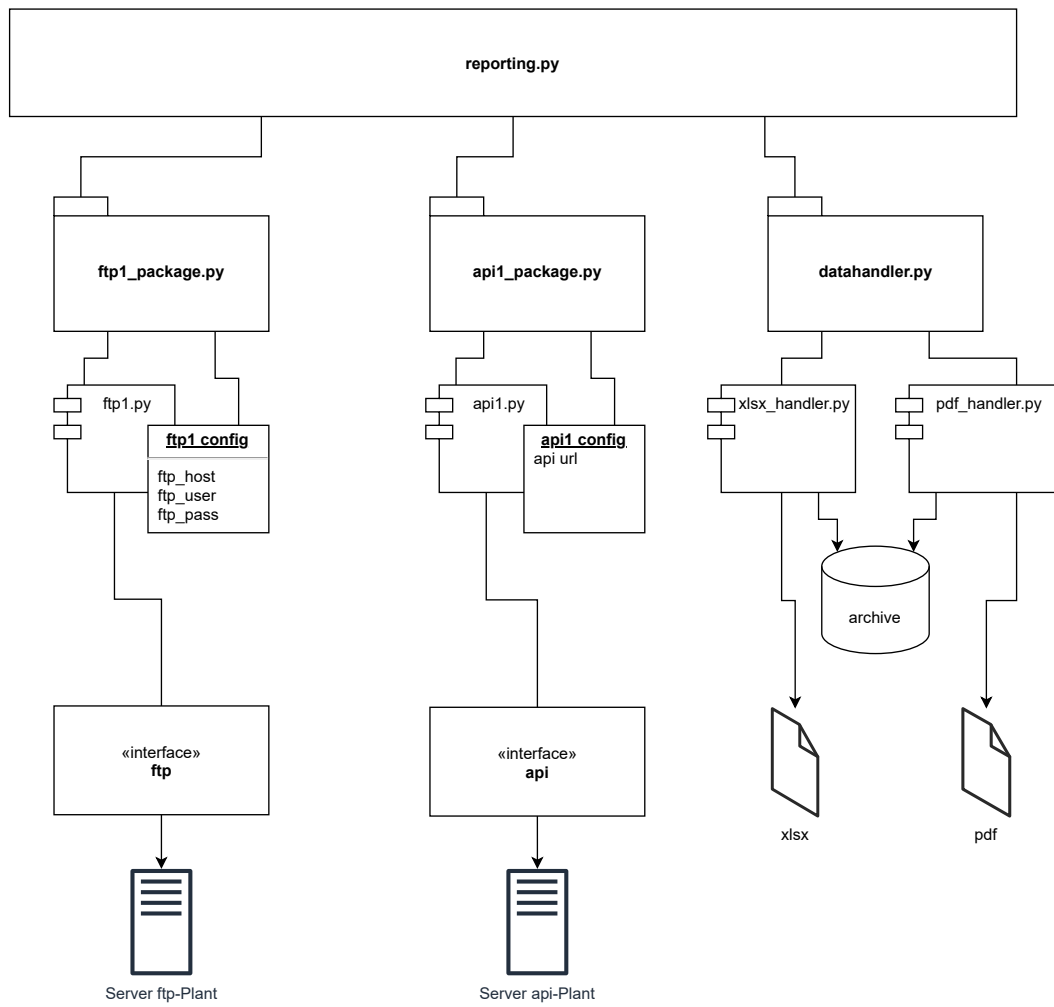


Abbildung 4.3: Architektur der entwickelten Python Software

5 Umsetzung und Entwicklung

5.1 Vorbereitung und Softwareablauf

In diesem Kapitel wird beschrieben, wie die Lösung auf Basis des Designs umgesetzt wurde. Es wird erklärt, welche Vorbereitungen vor dem ersten Einsatz der Software getroffen werden müssen und wie das Tool gestartet wird. Anschließend wird der Ablauf der Software und die einzelnen von der Software durchgeführten Schritte erläutert.

5.1.1 Vorbereitung

Das Produkt wird dem Nutzer in Form einer ZIP-Datei zur Verfügung gestellt, welche sowohl Instruktionen zur ersten Ausführung des Tools, als auch alle zur Ausführung benötigten Dateien enthält. Nachdem der Nutzer die in der ZIP-Datei enthaltenen Daten extrahiert hat, findet er zwei README-Dateien, welche jeweils eine kurze Anleitung zur Installation der notwendigen Drittsoftware und der Nutzung des Tools in deutscher und englischer Sprache bieten. Die Installation der zur Ausführung der Software benötigten Python-Version wird beschrieben und der Nutzer wird gebeten die benötigten Bibliotheken zu installieren. Um den Aufwand für den Nutzer so gering wie möglich zu halten, wird eine BATCH-Datei mit dem Namen 'install_libs' zur Verfügung gestellt, welche die Installation der benötigten Python-Bibliotheken übernimmt. Nachdem die benötigten Bibliotheken installiert wurden, sind die Randbedingungen für die erste Ausführung der Software geschaffen.

5.1.2 Starten des Tools

Um die Software zu starten, wird dem Nutzer ein weiterer BATCH-File mit dem Titel 'get_report' zur Verfügung gestellt. Durch einen Doppelklick auf diese Datei wird die Kommandozeile geöffnet und die Ausführung des Softwaretools wird gestartet. Python

bietet die Möglichkeit Softwarepakete zu einer sogenannten 'Standalone'-Datei zu kompilieren, worauf in diesem Fall bewusst verzichtet wurde. Im Gegensatz zu der Ausführung als Standalone-Version bietet die Ausführung der Software unter Nutzung eines BATCH-Files die Möglichkeit den Quellcode zu bearbeiten, ohne dem Nutzer eine komplett neue Version der Software zur Verfügung stellen zu müssen. Sollten weitere Anlagen in die Software eingebunden werden oder Softwarefehler behoben werden müssen, entfällt somit der Prozess des erneuten Kompilierens. Daraus ergibt sich die Möglichkeit dem Nutzer kleine Updates zur bereits bestehenden Software zur Verfügung zu stellen.

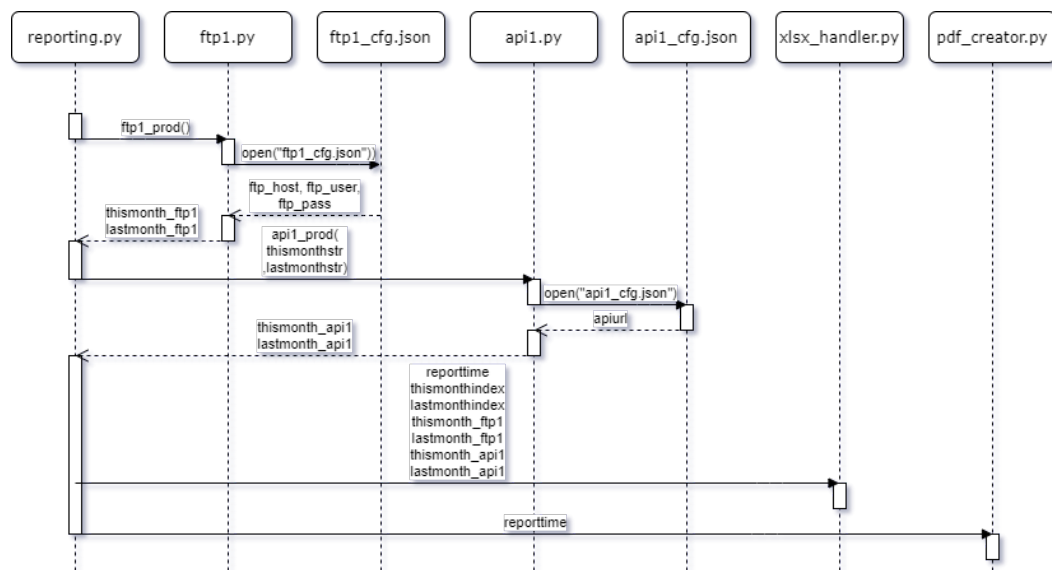


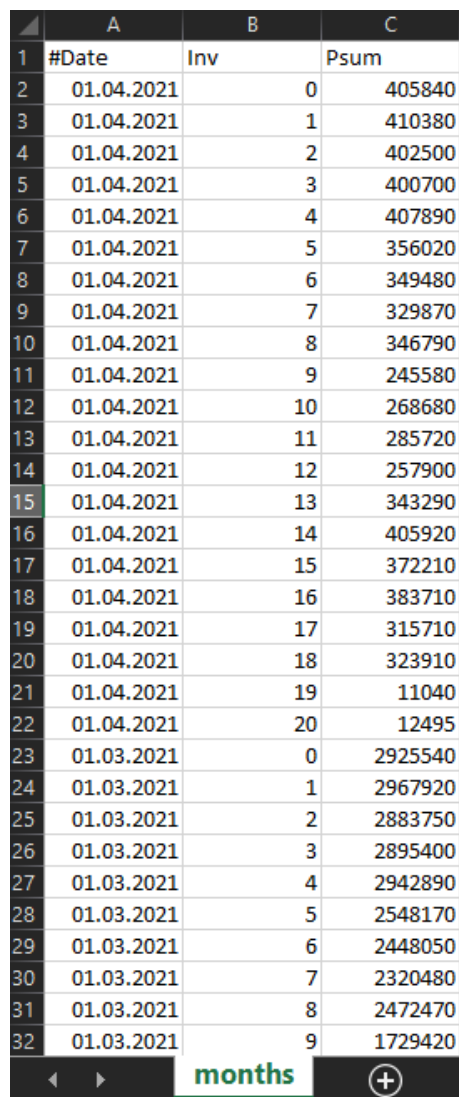
Abbildung 5.1: Der Softwareablauf wird mit Hilfe eines Sequenzdiagramms beschrieben

Der Ablauf der Software wird anhand des in Abb. 5.1 gezeigten Sequenzdiagramms beschrieben. Im ersten Schritt werden die benötigten Bibliotheken und Packages importiert. Es werden außerdem die Python-Packages importiert, in denen sich die Funktionen zum Bezug und zur Verarbeitung der benötigten Daten befinden. Mit Hilfe der Bibliothek 'datetime' werden im Objekt 'reporting.py' die Zeitvariablen deklariert und formatiert, mit denen die Daten für den entsprechenden Berichtszeitraum abgerufen werden.

5.1.3 Abruf der Daten vom FTP-Server

Nachdem alle benötigten Variablen deklariert sind, wird die Funktion 'ftp1_prod()', welche sich im Objekt 'ftp1_pack.py' befindet, ohne die Übergabe von Parametern aufgerufen.

fen. Diese Funktion dient dem Abrufen der Produktionsdaten von dem FTP-Server der ersten Anlage. Zum Abruf der Daten benötigt die Methode Anmeldedaten für den Server, welche in eine separate Konfigurationsdatei ausgelagert wurden. Die Daten werden aus der Datei 'ftp1_cfg.json' abgerufen und die benötigten Daten werden mit Hilfe der zuvor eingebunden Bibliothek 'ftplib' vom FTP-Server bezogen. Die Daten werden dann in einer zuvor erstellten CSV-Datei zwischengespeichert. Für den Abruf der Daten wurde eine Ausnahmebehandlung vorgenommen, welche dem Nutzer eine Fehlermeldung ausgibt, sollte der Abruf der Daten nicht möglich sein. Vom FTP-Server wird eine Datei mit dem Namen 'months.csv' bezogen, deren Datenstruktur in Abb. 5.2 zu sehen ist. Die Datei beinhaltet die Produktionsdaten der Anlage in Wattstunden [Wh] bis zum Inbetriebnahmedatum. Die Gesamtproduktion jedes Monats setzt sich aus der Produktion der 21 in der Anlage verbauten Wechselrichter zusammen. Um die Produktion des aktuellen und des vorangegangenen Monats zu erhalten, werden die in den 21 Zeilen unter der Überschrift 'Psum' befindlichen Zahlen summiert und durch 1000 dividiert um die Produktion des aktuellen Monats in Kilowattstunden zu erhalten. Zur Ermittlung der Produktionsmenge des Vormonats werden die 21 darauf folgenden Zeilen summiert und ebenfalls durch 1000 dividiert. Die ermittelten Produktionsmengen werden in den Variablen 'thismonth_ftp1' und 'lastmonth_ftp1' gespeichert und an die aufrufende Methode 'reporting.py' zurückgegeben. Vor dem Beenden der Funktion wird die Datei 'ftp_plant.csv', welche dem Zwischenspeichern der Daten dient, entfernt.



	A	B	C
1	#Date	Inv	Psum
2	01.04.2021	0	405840
3	01.04.2021	1	410380
4	01.04.2021	2	402500
5	01.04.2021	3	400700
6	01.04.2021	4	407890
7	01.04.2021	5	356020
8	01.04.2021	6	349480
9	01.04.2021	7	329870
10	01.04.2021	8	346790
11	01.04.2021	9	245580
12	01.04.2021	10	268680
13	01.04.2021	11	285720
14	01.04.2021	12	257900
15	01.04.2021	13	343290
16	01.04.2021	14	405920
17	01.04.2021	15	372210
18	01.04.2021	16	383710
19	01.04.2021	17	315710
20	01.04.2021	18	323910
21	01.04.2021	19	11040
22	01.04.2021	20	12495
23	01.03.2021	0	2925540
24	01.03.2021	1	2967920
25	01.03.2021	2	2883750
26	01.03.2021	3	2895400
27	01.03.2021	4	2942890
28	01.03.2021	5	2548170
29	01.03.2021	6	2448050
30	01.03.2021	7	2320480
31	01.03.2021	8	2472470
32	01.03.2021	9	1729420

Abbildung 5.2: Auszug aus den vom FTP-Server bezogenen Daten

5.1.4 Abruf der Daten unter Nutzung der API

Um die Daten für den aktuellen Berichtszeitraum von der zweiten Anlage zu beziehen, wird die Funktion 'api1_prod' mit den Parametern 'thismonthstr' und 'lastmonthstr' aufgerufen. Die Programmierschnittstelle wird mit einer URL bedient, welche einen API-Key und einen speziellen Anlagenkey enthält. Die API-URL wurde in eine separate Konfigurationsdatei mit dem Namen 'api1_cfg.json' ausgelagert. Mit dem Befehl 'with

`open('api1_cfg.json')` wird die Konfigurationsdatei aufgerufen und die API-URL kann nun von der Methode `api1.py` weiter angepasst werden. Um die Produktionsdaten von dem aktuellen Monat und dem letzten Monat zu erhalten, muss die URL durch einsetzen der zuvor übergebenen Zeitvariablen entsprechend des aktuellen Monats modifiziert werden. Es werden anschließend zwei Variablen deklariert, in denen die entsprechende URL für den aktuellen und für den vorangegangenen Monat hinterlegt ist. Für den Abruf der Daten wurde eine Ausnahmebehandlung vorgenommen, um dem Nutzer eine Fehlermeldung auszugeben, sollte der Abruf der Daten nicht möglich sein. Durch Nutzung der Bibliotheken `requests` und `json` wird unter Nutzung der zuvor modifizierten URL ein Request-Befehl an die API gestellt. Die von der API zurückgegebenen Daten werden in der Variable `response` zwischengespeichert und im nächsten Schritt für die Variable `jsdata` als JSON-Daten deklariert. Dem Nutzer wird die Rückmeldung gegeben, dass die Daten erfolgreich abgerufen wurden. Abb. 5.3 zeigt einen Auszug aus den Daten, welche von der API zurückgegeben wurden und welches dem Python Datentyp `'Dictionary'` entspricht. Es ist zu sehen, dass entsprechend des Zieles die monatlichen Produktionsdaten zu ermitteln, der Typ `'month'` gewählt wurde. Dies hat zur Folge, dass die Anzahl der Tage des ausgewählten Monats äquivalent zu der Anzahl der Datenpunkte in dem bezogenen Datensatz ist.

```
{
  "chartData": {
    "prev": "2020-03-01",
    "next": "2020-05-01",
    "date": "2020-04-01",
    "type": "month",
    "data": [
      [
        1585699200000,
        1700.6
      ],
      [
        1585785600000,
        1772.8
      ],
      [
        1585872000000,
        3379.7
      ],
      [
        1585958400000,
        3811.1
      ],
    ]
  }
}
```

Abbildung 5.3: Auszug aus den durch Nutzung der API bezogenen Daten

Um die Produktionsdaten aus den bezogenen Daten zu extrahieren, werden die benötigten `'Keys'` festgelegt und die Variable `'data'` wird zur Speicherung der Produktionsdaten

vorbereitet. In Abb. 5.3 ist zu sehen, dass unter dem Key 'data' eine Liste von Listen zu finden ist. Die Listen innerhalb der Liste enthalten jeweils zwei Einträge. Bei dem ersten Listeneintrag handelt es sich um eine Zeit-ID, während es sich bei dem zweiten Eintrag um die Produktion der fortlaufenden Tage des Monats handelt. Die benötigten Daten sollen nun mit Hilfe einer 'for-Schleife' extrahiert werden. Zu diesem Zweck wird eine Variable 'indexval' mit dem Wert '0' initialisiert, welcher den Startwert der 'for-Schleife' angibt. mit dem Befehl `for i in range(len(data)):` wird festgelegt, dass die Anzahl der Schleifendurchgänge der Anzahl der Listeneinträge des Keys 'data' entsprechen soll. Innerhalb der Schleife wird nun der Befehl `prod.append(data[indexval][1])` ausgeführt. Mit Hilfe der Schleifenvariable werden die Listeneinträge nacheinander durchlaufen und es wird immer der zweite Listeneintrag betrachtet. Mit dem Befehl 'append' wird der jeweilige Listeneintrag an das Ende der zuvor als Liste initialisierten Variable 'prod' geschrieben. Nachdem der Vorgang abgeschlossen ist, enthält die Variable 'prod' alle benötigten Produktionsdaten des aktuellen Monats, welche nun summiert werden müssen, was unter Verwendung der Python Bibliothek 'pandas' durchgeführt wird. Zu diesem Zweck wird mit 'dfprod' ein Dataframe angelegt und der Spalte wird der Key 'production' zugewiesen. Im nächsten Schritt wird die Gesamtproduktion des Monats berechnet, indem alle in der Spalte 'production' befindlichen Einträge mit dem Befehl `sum()` summiert werden und anschließend mit dem Befehl `round(decimals=1)` auf eine Nachkommastelle gerundet werden. Die Summe der Werte wird in der Variable 'thismonth_apil' gespeichert und am Ende der Funktion an die aufrufende Methode `reporting.py` zurückgegeben. Dieser Vorgang wird anschließend für die Produktionsdaten des vorangegangenen Monats wiederholt.

5.2 Verarbeiten der Daten im XLSX-Format

Nachdem die Produktionsdaten der eingebunden Anlagen für den aktuellen und den vorangegangenen Monat berechnet wurden, müssen die Daten nun in die zuvor angelegte Tabelle eingetragen werden, aus der im weiteren Verlauf der Produktionsbericht extrahiert wird. Mit dem Befehl `xlsx.xlsx_handler` wird nun die Funktion zur Verarbeitung der Daten innerhalb einer Excel-Tabelle aufgerufen. Um die Operation korrekt durchführen zu können, werden die Parameter 'reporttime', 'thismonthindex', 'lastmonthindex', 'thismonth_apil', 'lastmonth_apil', 'thismonth_ftpl' und 'lastmonth_ftpl' an die Funktion 'xlsx_handler' zur späteren Verwendung übergeben. Innerhalb der Funktion werden

zuerst die benötigten Bibliotheken importiert, deren Funktionen im weiteren Verlauf beschrieben werden. Da der aktuelle Bericht auf dem vorangegangenen Bericht basieren soll, wird mit Hilfe der Bibliothek 'glob' nach dem vorangegangenen Bericht gesucht. Da die Dateinamen der Berichte beim Abspeichern einen Zeitstempel erhalten, beschränkt sich die Suche nur auf den feststehenden Teil des Dateinamens, welcher in der Variable 'i' als String gespeichert ist. Außerdem wird der Bibliothek der Pfad übergeben, in dem nach der Datei gesucht werden soll. Für die Suche nach dem vorangegangenen Bericht wurde eine Ausnahmebehandlung vorgenommen, um dem Nutzer eine Fehlermeldung auszugeben, sollte der vorangegangene Bericht nicht gefunden werden. Sofern der Bericht gefunden wurde, wird der zugehörige Dateipfad in der Variable 'lastfile' gespeichert. Im nächsten Schritt werden die zuvor ermittelten Produktionswerte unter Verwendung der Bibliothek 'openpyxl' in die Tabelle eingetragen. Um diesen Schritt durchzuführen, wird die zuvor ermittelte Datei mit dem Befehl `load_workbook` geladen und das entsprechende Arbeitsblatt wird ausgewählt. Um die Zeile zu ermitteln, in der die Daten für den aktuellen und den vorangegangenen Monat eingetragen werden, werden die Variablen 'thismonth-index' und 'lastmonthindex' verwendet und die Produktionswerte werden eingetragen. Anschließend wird die Arbeitsmappe mit dem vorgegebenen Dateinamen und dem aktuellen Zeitstempel gespeichert. Der letzte Arbeitsschritt der Methode besteht darin, den vorangegangenen Bericht in das Archiv zu verschieben, sodass nur der aktuelle Produktionsbericht auf der obersten Ebene des Verzeichnisses zu finden ist. Dazu wird, der Dateipfad des alten Berichtes angepasst.

5.3 Ausgabe des Produktionsberichtes im PDF-Format

Um die Produktion der Anlagen übersichtlich darzustellen und dem Nutzer die Möglichkeit zu bieten beteiligten Stakeholdern die Ergebnisse zu präsentieren, soll ein formatierter Produktionsbericht im PDF-Format ausgegeben werden. Hierzu wird die Methode 'pdf_handler' von der 'reporting.py'-Methode aufgerufen und es wird der Parameter 'reporttime' übergeben, welcher zur Speicherung des Berichtes mit dem aktuellen Zeitstempel dient. Das Format des Berichtes wird in dem letzten Arbeitsblatt der zuvor bearbeiteten Arbeitsmappe festgelegt, welche die aktualisierten Produktionsdaten der Anlagen beinhaltet. In diesem Fall besteht keine Notwendigkeit, den aktuellen Bericht auf dem vorangegangenen Bericht aufzubauen, weshalb der erste Schritt der aufgerufenen Funktion 'pdf_handler' darin besteht, den zuletzt erstellten Bericht in das Archiv zu verschieben. Wie schon in der Funktion 'xlsx_handler' wird auch in diesem Fall die Bibliothek 'glob'

verwendet um den vorangegangenen Bericht zu lokalisieren. Da das Auffinden der Datei zur Erstellung des neuen Berichtes nicht zwingend notwendig ist, wurde in diesem Fall auf eine Ausnahmebehandlung verzichtet. Durch die Verwendung eines 'if...else'-Statements wird der User dennoch darüber informiert, ob der vorangegangene Bericht aufgefunden und in das Archiv verschoben wurde. Sowohl der Dateipfad der zur Erstellung des PDF-Berichtes verwendeten Excel-Datei, als auch der Pfad der PDF-Datei werden unter Verwendung der 'reporttime'-Variablen festgelegt und in den Variablen 'excel_file' und 'pdf_file' gespeichert. Die Bibliothek 'pathlib' nutzt diese Variable, um die entsprechende Datei im `cwd()` also dem aktuellen Arbeitsverzeichnis zu öffnen. Dieser Prozess beginnt mit dem Starten der Excel-Anwendung, was durch den Befehl `excel.Visible = 0` für den Nutzer nicht sichtbar ist. Die zuvor ausgewählte Datei wird dann in Excel geöffnet und es wird das Arbeitsblatt ausgewählt, welches das Berichtslayout beinhaltet. Für die Speicherung des Arbeitsblattes im PDF-Format wurde eine Ausnahmebehandlung vorgenommen, so dass der Nutzer im Falle eines Fehlers eine entsprechende Fehlermeldung erhält. Nachdem das Arbeitsblatt mit dem Befehl `ws.SaveAs(pdf_path, FileFormat=57)` unter dem zuvor festgelegten Dateinamen gespeichert wurde, wird die noch geöffnete Arbeitsmappe geschlossen und die Excel-Anwendung wird beendet. Der Nutzer erhält nun mittels eines Dialogfensters die Nachricht, dass der Bericht erstellt wurde. Sobald der User mit einem Klick auf den 'Okay'-Button seine Kenntnisnahme bestätigt, ist die Berichtserstellung abgeschlossen und das Programm wird beendet.

6 Evaluation

Die Evaluation des Produktes soll zeigen, ob die an die Software gestellten Anforderungen umgesetzt wurden. Die Software soll zuvor vom User ausgeführte repetitive Aufgaben übernehmen und einen Produktionsbericht ausgeben. Messbare Anforderungen an die technische Performance der Software wurden dabei nicht gestellt. Um die Erfüllung der Anforderungen zu evaluieren wurde ein Scoreboard entwickelt, mit Hilfe dessen die Erfüllung der einzelnen Anforderungen geprüft wird. Wie bei den Anforderungen wurde auch bei der Evaluation zwischen funktionalen und nicht funktionalen Anforderungen unterschieden und die Wertung der Anforderungen wurde berücksichtigt, indem für Anforderungen der Kategorie 'MUSS' drei Punkte, 'SOLL' zwei Punkte und 'WIRD' je ein Punkt für die Erfüllung vergeben wurde.

Tabelle 6.1 zeigt die Evaluationsergebnisse der funktionalen Anforderungen, angeordnet nach den Anforderungscodes, die im Folgenden auch als Bezeichnung für die Anforderungen dienen. Durch die Entwicklung einer Batch-Datei zum Starten der Software ist die zusätzliche Installation einer Python IDE seitens des Nutzers nicht nötig. Zwar ist eine Python-Installation auf dem System des Users notwendig, dabei handelt es sich jedoch um einen einmaligen Prozess, daher gilt die Anforderung RF1 als erfüllt. Da die Erstellung des Produktionsberichtes auf dem vorangegangenen Bericht aufbaut, werden die Produktionsdaten der Anlage fortgeschrieben und es entsteht eine Datenbank mit Rohdaten, die unter einem für den Nutzer erreichbaren Verzeichnis und mit einem festgelegten Dateinamen abgelegt wird, womit RF2 als erfüllt betrachtet wird. Nachdem der User die Berichtserstellung startet, öffnet sich die Windows Kommandozeile und informiert den User über den aktuellen Stand der Berichtserstellung. Die Fertigstellung des Berichtes wird dem User sowohl in der Kommandozeile, als auch in einem Dialogfenster angezeigt, womit RF3 erfüllt ist. Der erstellte Bericht inklusive der Rohdaten wird softwareseitig im .xlsx-Format abgelegt. Die abgelegte Tabelle enthält ein Arbeitsblatt, welches das Layout des Ausgabeberichtes enthält und die ermittelten Daten aus den Arbeitsblättern bezieht, welche den einzelnen Produktionsanlagen zugeordnet sind. Bei der

Berichtserstellung mit Hilfe des Reportingtools, wird dieses Arbeitsblatt automatisiert im PDF-Format abgelegt, womit RF4 als erfüllt betrachtet wird. Sollte seitens des Nutzers eine Anpassung des Layouts nötig sein, so ist eine manuelle Anpassung mit Hilfe der *xlsx*-Datei möglich. Zum Zweck der Nachverfolgung sollen die Produktionsdaten der Anlagen laufend protokolliert werden, weshalb der vorangegangene Bericht als Basis für den zu erstellenden Bericht dienen soll. Um dies zu gewährleisten, werden die Daten des vorangegangenen Berichtes vom Softwaretool eingelesen und aktualisiert. Der neue Bericht wird mit dem aktuellen Zeitstempel abgelegt und der vorangegangene Bericht wird in ein Archiv verschoben, womit RF5 und RF6 erfüllt wurden. Die Erstellung einzelner Berichte pro Produktionsanlage ist in der Software bisher nicht berücksichtigt worden, weshalb RF7 als nicht erfüllt betrachtet wird. Zur anwenderfreundlichen Umsetzung dieser Funktion ist der Einsatz einer grafischen Nutzeroberfläche sinnvoll, auf welche bislang verzichtet wurde. Eine Implementierung dieser Funktion ist allerdings technisch realisierbar und wird bei der zukünftigen Weiterentwicklung berücksichtigt. Bei den durch das Reportingtool erstellten Produktionsberichten wird der aktuelle Monat bis zum vorangegangenen Tag betrachtet. Bei der Entwicklung der Software wurde davon ausgegangen, dass die Erstellung von Produktionsberichten mindestens wöchentlich durchgeführt wird, weshalb für jeden vorangegangenen Monat ein Bericht vorliegen sollte. Die Erstellung von Berichten über einen frei wählbaren Zeitraum ist bislang nicht möglich, daher wird RF8 als nicht erfüllt betrachtet. Durch die Erfüllung aller funktionalen Anforderungen konnte unter Berücksichtigung der Wertung eine Gesamtpunktzahl von 17 erreicht werden. Da zwei Anforderungen der Kategorie 'WIRD' nicht erfüllt sind, erreicht die Software bezüglich der Umsetzung der funktionalen Anforderungen eine Punktzahl von 15 und damit einen Score von 88,2 %.

Die Evaluation der nicht funktionalen Anforderungen an die Software wurde analog zu dem vorangegangenen Vorgehen umgesetzt und wird in Tabelle 6.2 gezeigt. Da die Erstellung der Berichte bislang mit hohem Aufwand für den Nutzer verbunden war, soll durch die Software eine Berichtserstellung mit maximal fünf Mausklicks ermöglicht werden. Diese Anforderung bezieht sich auf die regelmäßige Erstellung der Berichte, nicht aber auf die erstmalige Nutzung der Software zu der möglicherweise die Installation von Python und den benötigten Bibliotheken notwendig ist. Um mit Hilfe des Reportingtools einen Bericht für den aktuellen Monat zu erstellen, wird dem Nutzer eine Batch-Datei zur Verfügung gestellt, womit das Reportingtool durch einen Doppelklick gestartet wird. Ein weiterer Klick durch den User bestätigt lediglich die Kenntnisnahme der Berichtserstellung, weshalb RNF1 als erfüllt betrachtet wird. Die Möglichkeit weitere Anlagen

Tabelle 6.1: Evaluation der funktionalen Anforderungen

Anforderung	Code	Wertung	Erfüllt	Nicht erfüllt	Ergebnis
Funktionale Anforderungen					
Das Softwaretool wird ohne die zusätzliche Nutzung einer Python IDE unter Windows geöffnet [MUSS]	RF1	3	x		3
Rohdaten werden in .csv oder .xlsx Format mit Dateinamen "\\Datum[DD.MM.YYYY]_Produktion_Projektname]" in festgelegtem Ordner gespeichert [MUSS]	RF2	3	x		3
Der User wird über die Fertigstellung des Berichtes optisch benachrichtigt [SOLL]	RF3	2	x		2
Bericht wird im Format PDF gespeichert [SOLL]	RF4	2	x		2
Zum Erstellen des aktuellen Berichtes wird der vorangegangene Bericht geöffnet, aktualisiert und unter einem neuen Dateinamen gespeichert [MUSS]	RF5	3	x		3
Der vorangegangene Bericht wird in eine Archiv verschoben [SOLL]	RF6	2	x		2
Der User hat die Möglichkeit einzelne Berichte pro Anlage oder einen Gesamtbericht zu erstellen [WIRD]	RF7	1		x	0
Der Berichtszeitraum ist frei wählbar [WIRD]	RF8	1		x	0
	Total	17		Gesamtpunktzahl	15
				Score	88,2%

in die Software und damit in die Berichtserstellung einzubinden, wird durch den modularen Aufbau der Software grundsätzlich geboten. Das Einbinden weiterer Anlagen mit Hilfe eines hinterlegten Templates lässt sich technisch zwar umsetzen, ist aber mit einem Aufwand verbunden, der den Nutzen dieser Funktion übersteigt. Das Einbinden weiterer Anlagen in die Software ist nur in seltenen Fällen notwendig und stellt keinen Prozess dar, welcher einer Automatisierung bedarf. Perspektivisch kann ein solcher Prozess implementiert werden, allerdings wurde vorerst darauf verzichtet, weshalb RNF2 als nicht erfüllt betrachtet wird. Innerhalb der Anforderungen wurden Spezifikationen eines gängigen Office-Systems beschrieben, welche als Hardwareanforderungen zur Ausführung der Software dienen. Bei diesem System handelt es sich um das aktuell zur Berichtserstellung verwendete System, welches daher als Mindestanforderung für die Software angenommen

wird. Der Test der Software mit der festgelegten Konfiguration wurde erfolgreich durchgeführt, weshalb RNF3 als erfüllt betrachtet wird. In der vorliegenden Version des Reportingtools wird nach dem Starten der Software keine weitere Eingabe durch den Nutzer zum Erstellen des Berichtes benötigt, weshalb eine grafische Benutzeroberfläche bislang nicht implementiert wurde. Für die perspektivische Weiterentwicklung der Software mit Auswahlmöglichkeiten bezüglich der berücksichtigten Anlagen und des Berichtszeitraumes, erscheint die Entwicklung einer grafischen Benutzeroberfläche sinnvoll, allerdings wurde vorerst darauf verzichtet, weshalb RNF4 als nicht erfüllt betrachtet wird. Sollte es bei der Erstellung von Berichten zu Fehlern kommen, hat der User über eine eigens dafür angelegte E-Mail-Adresse die Möglichkeit den Softwareentwickler zu kontaktieren. Die Kontaktdaten sind in den beigegeführten README-Files zu finden, RNF5 wird damit als erfüllt betrachtet. Zum Ausführen der Software wird eine Python 3.X Installation auf dem System des Nutzers benötigt. Sollte eine solche Installation nicht bereits vorliegen, muss dies nachgeholt werden. Neben der Python-Version sind auch Bibliotheken zum Nutzen der Software nötig, die ebenfalls installiert werden müssen. Zu diesem Zweck werden dem User zwei README-Dateien zur Verfügung gestellt, welche sich inhaltlich gleichen, jedoch in deutscher und in englischer Version verfügbar sind. Um sicher zu stellen, dass die Installation der benötigten Software korrekt durchgeführt wird, wird dem User hier eine Anleitung zur Verfügung gestellt. Zur Installation der benötigten Bibliotheken wurde ein Batch-File entwickelt, dessen Handhabung und Funktion ebenfalls in den README-Files beschrieben wird und welcher die benötigten Bibliotheken mit einem Doppelklick automatisch installiert. Durch das zur Verfügung stellen dieser Dokumente und der Batch-Dateien zur vereinfachten Installation, wird RNF6 als erfüllt betrachtet. Durch die Erfüllung aller nicht funktionalen Anforderungen konnte unter Berücksichtigung der Wertung eine Gesamtpunktzahl von 13 erreicht werden. Da zwei Anforderungen der Kategorie 'WIRD' nicht erfüllt sind, erreicht die Software bezüglich der Umsetzung der nicht funktionalen Anforderungen eine Punktzahl von 11 und damit einen Score von 84,6 %. Beide Auswertungen betrachtend wird die erstellte Lösung mit einem Score von 86,4 % bewertet.

Auch die Bewertung des Ausgabeberichtes erfolgt unter Verwendung der entwickelten Bewertungsmatrix. Dadurch soll sichergestellt werden, dass die benötigten Daten mit dem erstellten Produktionsbericht abgebildet werden. Bei der Bewertung des Berichtes ist zu beachten, dass das Layout und die Darstellung der benötigten Daten in einer Excel-Datei umgesetzt werden, welche unabhängig von dem entwickelten Softwaretool funktioniert. Mit Hilfe des Reportingtools werden die Daten in der Excel-Tabelle ledig-

Tabelle 6.2: Evaluation der nicht funktionalen Anforderungen

Anforderung	Code	Wertung	Erfüllt	Nicht erfüllt	Ergebnis
Nicht funktionale Anforderungen					
Bericht des aktuellen Monats soll sich mit maximal 5 Klicks erstellen lassen [MUSS]	RNF1	3	x		3
Für die Integration neuer Anlagen in die Software ist ein Template innerhalb der Software hinterlegt, welches kopiert und um die entsprechend Anlagendaten ergänzt werden kann, um neue Anlagen hinzuzufügen [WIRD]	RNF2	1		x	0
Die Software ist mit gängiger Office-Hardware und dazugehörigen Betriebssystemen ausführbar [MUSS]	RNF3	3	x		3
Die Software wird über eine grafische Benutzeroberfläche bedient [WIRD]	RNF4	1		x	0
Dem User wird ein Kontakt zur Fehlerbehebung zur Verfügung gestellt [SOLL]	RNF5	2	x		2
Dem User wird Anleitung zur Verfügung gestellt, welche die Erstnutzung der Software erleichtert [MUSS]	RNF6	3	x		3
	Total	13		Gesamtpunktzahl	11
				Score	84,6%

lich aktualisiert. Um die Daten zuordnen zu können, müssen sowohl der Berichtszeitraum als auch die Bezeichnung des jeweiligen Projektes auf dem Bericht ersichtlich sein, was erfolgreich umgesetzt wurde. Um eine Bewertung der Produktion vornehmen zu können, wird die Menge der produzierten Energie des aktuellen Monats abgebildet und mit der zuvor prognostizierten Produktionsmenge verglichen. Die technische Verfügbarkeit der Anlage, sowie die Einstrahlungsmenge an der Anlage konnten nicht umgesetzt werden, da zu diesen Punkten keine Daten von den Schnittstellen bezogen werden konnten. Beide Anforderungen lassen sich technisch zwar umsetzen, allerdings erfordert die Umsetzung einen erhöhten Programmieraufwand, welcher aufgrund der geringen Relevanz der beiden Anforderungen nicht umgesetzt wurde. Um die Produktion des aktuellen Jahres bewerten zu können, wird neben dem Vergleich mit der Prognose auch ein Vergleich mit der Produktion des Vorjahres vorgenommen. Da die historischen Daten der Anlage laufend in der Datenbank protokolliert werden, konnte diese Anforderung erfolgreich umgesetzt

werden. Grafische Darstellungen ermöglichen es Betreibern und weiteren Stakeholdern der Anlagen, mit nur einem Blick eine Übersicht über die Produktion der Anlage zu erhalten. Sowohl die monatlichen Daten als auch kumulierte Daten, welche den Produktionsverlauf im laufenden Betriebsjahr darstellen, konnten erfolgreich umgesetzt werden. Durch die Erfüllung aller Anforderungen an den Bericht konnte unter Berücksichtigung der Wertung eine Gesamtpunktzahl von 21 erreicht werden. Da zwei Anforderungen der Kategorie 'WIRD' nicht erfüllt sind, erreicht die Software bezüglich der Umsetzung eine Punktzahl von 19 und damit einen Score von 90,5 %. Insgesamt erreicht die umgesetzte Lösung somit einen Score von 88,45 %, was einem guten Ergebnis entspricht.

Tabelle 6.3: Evaluation des Berichtes

Anforderung	Code	Wertung	Erfüllt	Nicht erfüllt	Ergebnis
Nicht funktionale Anforderungen					
Berichtszeitraum [MUSS]	BR1	3	x		3
Projektname [MUSS]	BR2	3	x		3
Produktion des aktuellen Monats in kWh [MUSS]	BR3	3	x		3
Prognose erreicht zu X Prozent [MUSS]	BR4	3	x		3
Verfügbarkeit in Prozent [WIRD]	BR5	1		x	0
Einstrahlungsmenge in kWh [WIRD]	BR6	1		x	0
Produktionsdaten des Vorjahres [MUSS]	BR7	3	x		3
Grafik monatliche Produktion im Vergleich zu Prognose [SOLL]	BR8	2	x		2
Grafik kumulierte Produktion im Vergleich zu Prognose [SOLL]	BR9	2	x		2
	Total	21		Gesamtpunktzahl	19
				Score	90,5%

Um die Anwenderfreundlichkeit evaluieren zu können, wurde die Software fünf Experten zur Verfügung gestellt, welche sie zukünftig zur Erstellung von Berichten nutzen sollen. Die Tester wurden darum gebeten nach Nutzung der Software einen zuvor erstellten Feedbackbogen auszufüllen, welcher im Folgenden näher betrachtet wird. Neben der Anwenderfreundlichkeit wurde den Testern anhand des Bogens die Möglichkeit geboten, Verbesserungsvorschläge zu äußern und die Weiterentwicklung der Software somit aktiv mit zu gestalten.

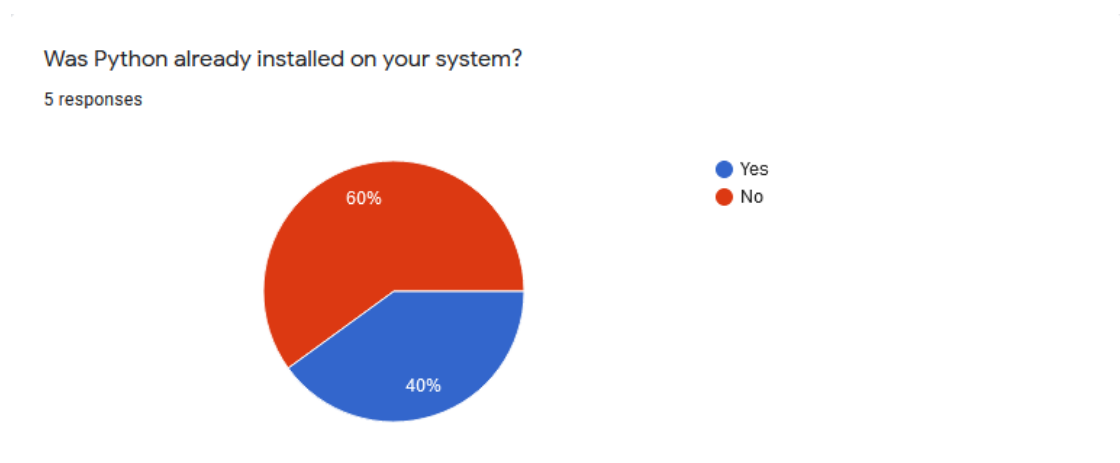


Abbildung 6.1: Feedback zur bereits durchgeführten oder noch notwendigen Installation von Python

Abbildung 6.1 zeigt, dass drei der fünf Tester keine Python Version auf Ihrem System installiert hatten und die Installation vor der ersten Nutzung der Software vornehmen mussten. Für diese Personen wurde in den README-Dateien eine Installationsanleitung hinterlegt, welche dem Nutzer einen Link zum Download der notwendigen Software, sowie eine Anleitung zur korrekten Installation bietet. Wie in Abb. 6.2 zu sehen, waren alle drei Nutzer mit Hilfe der bereitgestellten Anleitung in der Lage, Python korrekt zu installieren, weshalb die in Abb. 6.3 gestellte Frage nach Schwierigkeiten bei der Installation unbeantwortet blieb.

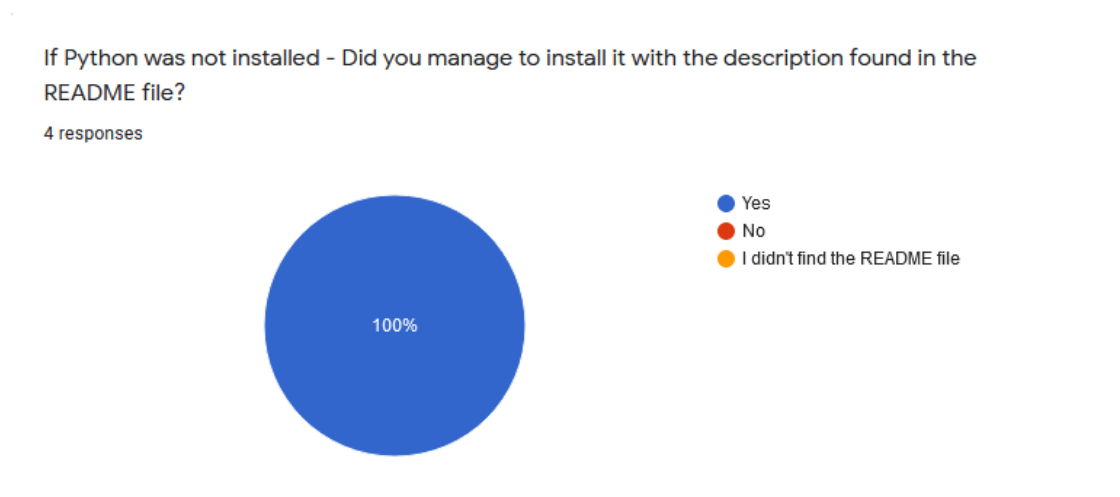


Abbildung 6.2: Feedback zur Frage ob Python mit Hilfe der zur Verfügung gestellten Anleitung installiert werden konnte

If you had problems installing python, please describe what kinds of problems you had.

0 responses

No responses yet for this question.

Abbildung 6.3: Feedback zu Schwierigkeiten bei der Installation von Python

Für die Ausführung des Reportingtools ist die Installation der benötigten Python-Bibliotheken unerlässlich. Dem Nutzer wird zu diesem Zweck eine Batch-Datei zur Verfügung gestellt, welche die Bibliotheken für den Nutzer installiert. Abb. 6.4 zeigt, dass die Installation bei allen Testern funktioniert hat, weshalb die in Abb. 6.5 gestellte Frage nach Schwierigkeiten oder Fehlermeldungen während der Installation unbeantwortet blieb.

Have you been able to install the required libraries with the 'install_libs.bat'?

5 responses

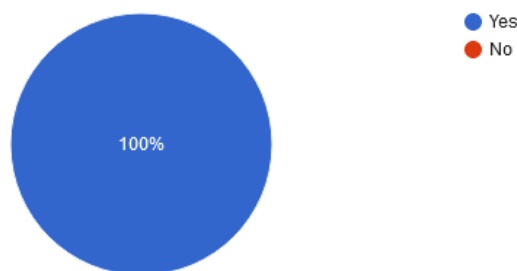


Abbildung 6.4: Feedback zur Frage ob die Installation der Bibliotheken mit Hilfe der Batch-Datei gelungen ist

If you have not been able to install the libraries, please copy the error message you got from the command prompt in the field below or describe your problem briefly.

0 responses

No responses yet for this question.

Abbildung 6.5: Feedback zu der Installation der Bibliotheken

Abbildung 6.6 zeigt, wie hoch der Anteil der Experten war, bei denen die Berichtserstellung ohne Probleme ausgeführt werden konnte. Bei einem der vier Tester brach die Software mit einer bis dahin unbekanntem Fehlermeldung ab, was sich nach Rücksprache auf das Fehlen einer MS Office Installation zurückführen ließ. Da ein installiertes Office Paket inklusive MS Excel teil der Mindestanforderungen an das Benutzersystem ist, ist dieser Fehler nicht auf die Software zurückzuführen. Um Nutzer zukünftig dennoch über diesen Fehler zu informieren, wurde eine entsprechende Fehlermeldung implementiert. Da MS Excel ausschließlich zur Ausgabe des Berichtes im PDF-Format verwendet wird, hatte der Nutzer trotz der Fehlermeldung einen aktuellen Produktionsbericht, welcher nur noch in eine PDF-Datei umgewandelt werden musste.

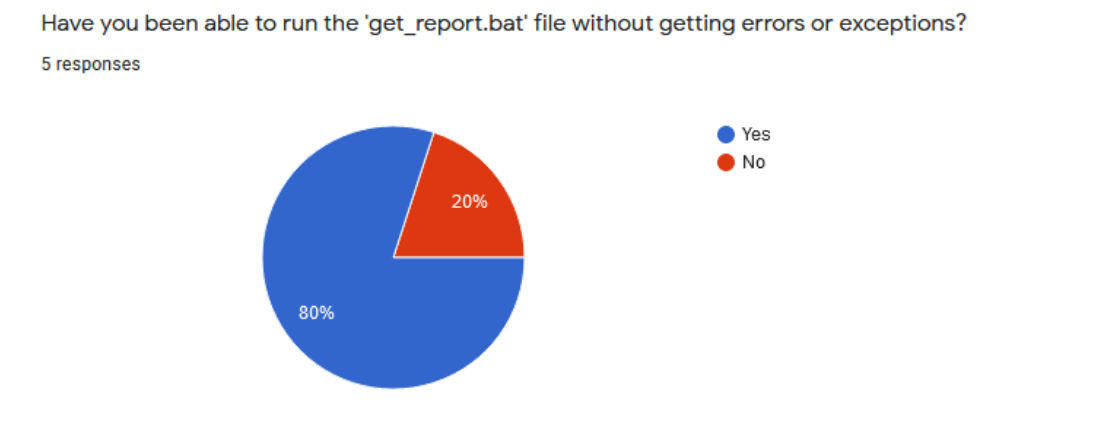


Abbildung 6.6: Feedback zur Berichtserstellung mit Hilfe des zur Verfügung gestellten Batch-Files

Trotz des unvollständigen Ablaufes der Software bei einem der fünf Nutzer, gaben wie in Abb. 6.7 zu sehen alle Nutzer an, fortan auf die softwareunterstützte Berichtserstellung zurückzugreifen, was für eine hohe Nutzerfreundlichkeit und das Erfüllen der Nutzeranforderungen spricht.

In der letzten Frage wurde den Nutzern die Möglichkeit geboten, eigene Ideen und Wünsche einzubringen, welche in der perspektivischen Weiterentwicklung der Software berücksichtigt werden sollen. Trotz bisher kaum notwendiger Nutzereingaben und der Ausgaben der Windows-Kommandozeile, wurde mehrfach der Wunsch nach einer grafischen Benutzeroberfläche geäußert. Ferner wünschen sich die Tester, den Monat oder sogar den gewünschten Berichtszeitraum frei wählen zu können. Diese Anregungen wurden aufgenommen und im Ausblick näher beschrieben. Ebenso gab es den Wunsch nach einer

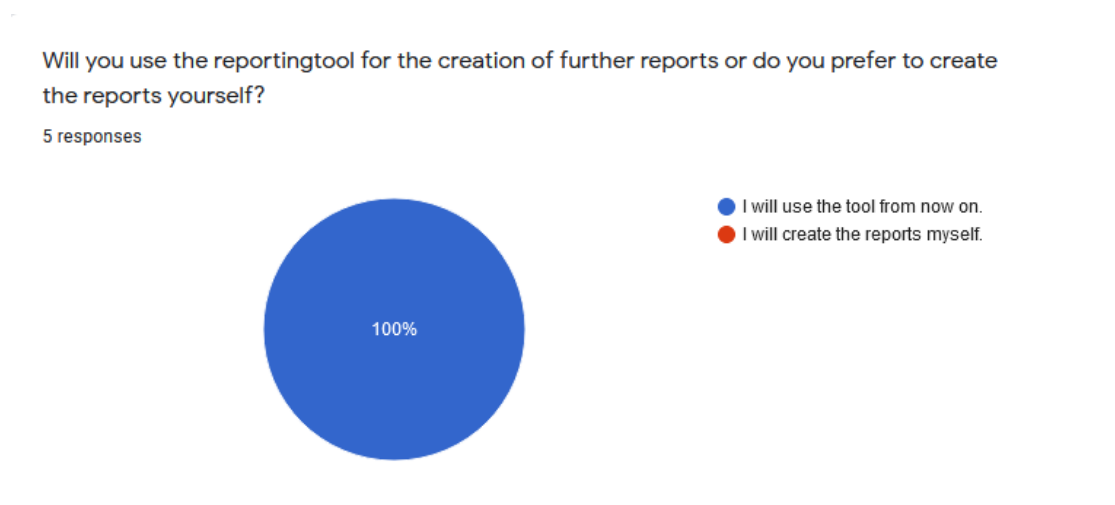


Abbildung 6.7: Feedback zur Frage ob die Nutzer auch zukünftig mit dem Tool arbeiten werden

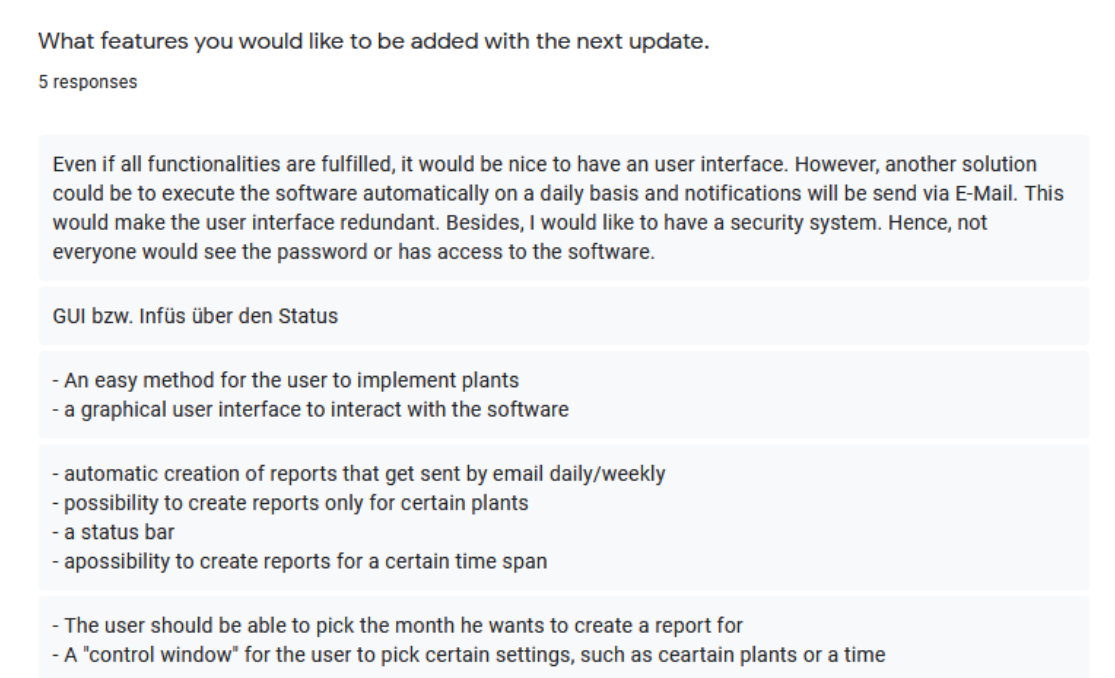


Abbildung 6.8: Darstellung der Verbesserungsvorschläge der Tester

vollautomatischen Berichtserstellung mit automatischem Versendung der Berichte via E-Mail. Diese Wünsche werden zukünftig nicht umgesetzt, da Produktionsberichte vor dem Versenden geprüft und freigegeben werden sollten. Gerade das Versenden von Produk-

tionsberichten an Banken oder sonstige Stakeholder ohne weitere Kommentierung, kann bei nicht befriedigenden Produktionszahlen zu Problemen führen. Wie bereits zuvor beschrieben, wird auch dem Wunsch nach einer teilautomatisierten Einbindung weiterer Anlagen anhand eines Templates oder anderen Methoden vorerst nicht nachgegangen.

7 Schlussbetrachtung

7.1 Zusammenfassung

Die Erstellung von Produktionsberichten von erneuerbaren Energienanlagen ist für Betreiber und Betriebsführer essentiell, um die Performance der Anlagen überwachen und steuern zu können. Gerade für Betreiber oder Betreuer mehrerer Anlagen unterschiedlicher Hersteller kann dieser Prozess aufgrund unterschiedlicher Schnittstellen und herstellereigener Systeme, zu großem Aufwand und damit verbunden zu hohen Kosten führen. Dies resultiert vor Allem aus der repetitiven Ausführung simpler Prozesse, wie beispielsweise Anmeldungen auf unterschiedlichen Nutzeroberflächen, dem Formatieren und der Analyse heruntergeladener Daten und letztlich der Erstellung eines einheitlichen, aussagekräftigen Berichtes. Um die dadurch entstehenden Kosten seitens der Betreiber von erneuerbaren Energieanlagen zu verringern, werden Softwaresysteme eingesetzt, die den mit der Berichtserstellung verbundenen Aufwand verringern sollen. Ziel dieser Arbeit war es, ein Softwaretool zu entwickeln, das den Prozess der Berichtserstellung für zwei Photovoltaikanlagen mit heterogenen Schnittstellen weitestgehend automatisiert. Zu diesem Zweck wurde das aktuelle Vorgehen bei der Erstellung von Produktionsberichten beider Photovoltaikanlagen analysiert. Es wurde anschließend festgelegt, wie der Ablauf der Berichtserstellung zukünftig aussehen soll und welche konkreten Aufgaben von dem Softwaresystem übernommen werden sollen. Anschließend wurden die Anforderungen an das Produkt formuliert und entsprechend ihrer Relevanz bewertet. Basierend auf den Anforderungen wurde eine Softwarearchitektur entwickelt, um die Komponenten des Softwaresystems und ihre Beziehungen zueinander festzulegen. Auf Basis der entwickelten Softwarearchitektur, konnte anschließend ein Sequenzdiagramm entwickelt werden, welches den Ablauf der Software zeigt. Auf Basis der Softwarearchitektur wurden die benötigten Module der Software zusammengestellt und mit Hilfe des Sequenzdiagramms wurden die jeweiligen Aufrufe und Abläufe umgesetzt.

7.2 Fazit

Um die Lösung bewerten zu können, wurde sowohl eine Bewertungsmatrix entwickelt, als auch ein Expertentest durchgeführt. Mit Hilfe der Bewertungsmatrix wurde die Umsetzung der zuvor erstellten funktionalen und nicht funktionalen Anforderungen geprüft und bewertet. Entsprechend der entwickelten Matrix, wurde die erstellte Softwarelösung mit einem Scoring von 90,5 % bewertet, was einem guten Ergebnis entspricht. Alle als relevant kategorisierten Anforderungen konnten umgesetzt werden und die daraus resultierenden Funktionen stehen dem Nutzer zur Verfügung. Die bisher nicht umgesetzten und als weniger relevant eingestuften Anforderungen, wurden in ihrer technischen Machbarkeit geprüft und werden bei der Weiterentwicklung der Software berücksichtigt. Die aktuelle Lösung bietet Verbesserungspotential bezüglich der Nutzerinteraktion, sowie auf den Gebieten Sicherheit und Berechtigungen. In der jetzigen Version der Lösung wird nach dem Start der Software keine weitere Nutzereingabe benötigt um die Erstellung des Produktionsberichtes vollständig durchzuführen. Dazu kommt, dass es sich bei der Erstellung von grafischen Nutzeroberflächen um eine Aufgabe handelt, die dem Bereich der Frontend-Entwicklung zugeschrieben wird, während die Entwicklung der Kernsoftware eine Aufgabe der Backend-Entwicklung ist. Aus diesen Gründen wurde bislang auf die Realisation eines Userinterface verzichtet. Da das Feedback aus dem Expertentest allerdings zeigt, dass ein solches Interface gewünscht wird, wurde ein Mockup für zukünftige Softwareversionen erstellt, welches im Kapitel 'Ausblick' näher betrachtet wird. Da die erstellte Software bisher ausschließlich autorisierten Personen zur Verfügung gestellt wird, wurde kein Anmeldeprozess zum Schutz der Software vor unbefugtem Zugriff implementiert. Dies sollte in zukünftigen Versionen ergänzt werden, da mit wachsender Funktionalität auch sensible Anlagendaten wie EEG-Schlüssel oder Angaben über den Betreiber verarbeitet werden könnten. Zu diesem Zweck muss eine Datenbank der Nutzerkonten hinterlegt werden, die von einem autorisierten Administrator verwaltet wird, welcher auch neu angelegte Nutzerkonten freischaltet. Eine weitere Verbesserungsmöglichkeit liegt darin, die Daten aus dem zur Erstellung des aktuellen Berichtes benötigten Vorgängerberichtes vor dem Verschieben zu sichern. Um Fehler bei der Berichtserstellung zu vermeiden, muss der Vorgängerbericht schreibgeschützt abgelegt und ein Verschieben des Berichtes verhindert werden. Um dies zu vermeiden, wäre die Ablage des Berichtes an einem für den Nutzer unzugänglichen Speicherort sinnvoll, was allerdings in Rücksprache mit dem jeweiligen Systemadministrator geschehen muss. Eine Alternativmöglichkeit besteht darin, die Daten des aktuellen Berichtes innerhalb der Software zwischenzuspei-

chern, sodass kein Zugriff auf eine externe Datei mehr nötig wäre. Zu diesem Zweck müsste eine programminterne Datenbank entworfen werden.

Die bisher durchgeführten Tests wurden von einer sehr kleinen Nutzergruppe (fünf Personen) durchgeführt und bilden nur eine Momentaufnahme der jetzigen Version ab. Um sicher zu stellen, dass das entwickelte Reportingtool dauerhaft von Betreibern und Anlagenbetreuern eingesetzt werden kann und um weitere Verbesserungen und mögliche Fehler zu lokalisieren, ist ein Langzeittest sinnvoll. Zu diesem Zweck sollte eine automatisierte Fehlerübermittlung implementiert werden, welche dem Entwickler die Möglichkeit bietet einen kontinuierlichen Verbesserungsprozess zu etablieren. Ferner sollten die Grenzen des Erweiterungspotentials der Software erkundet werden. Zu diesem Zweck sollte geprüft werden wie die Einbindung von Produktionsanlagen unter der Verwendung weiterer, bisher nicht betrachteter Schnittstellen aussehen kann. In diesem Zusammenhang kann auch die Erweiterung der Software auf die Erstellung von Produktionsberichten für Windkraftanlagen geprüft werden.

7.2.1 Mockup

Das bisher erhaltene Feedback zeigt, dass eine grafische Nutzeroberfläche ausdrücklich gewünscht wird. Da die Umsetzung einer solchen Nutzeroberfläche bislang als weniger relevant eingestuft wurde, wurde in der derzeitigen Version der Software darauf verzichtet. Um dennoch einen Ausblick darauf zu bieten, wie eine solche Nutzeroberfläche aussehen könnte und wie zukünftig geplante Funktionen in der Oberfläche implementiert werden könnten, wurde ein sogenanntes Mockup entworfen, in welchem sowohl die aus dem Feedback erhaltenen Wünsche, als auch selbst entwickelte Verbesserungen für zukünftige Versionen berücksichtigt werden. Abb. 7.2 zeigt, das Anmeldefenster der Software. Um die Software bedienen zu können, muss der User sich mit seinen Anmeldendaten einloggen. Sofern nicht bereits ein Userkonto vorhanden ist, kann der Button 'SIGN UP' genutzt werden, um ein neues Konto anzulegen.

Nachdem sich der Nutzer mit Hilfe seiner Zugangsdaten eingeloggt hat, wird das Startfenster der Software angezeigt, welches exemplarisch in Abb. 7.2 gezeigt wird. Innerhalb dieses Fensters hat der User die Möglichkeit, einen Berichtsmonat oder sogar einen frei wählbaren Berichtszeitraum zu bestimmen. Es kann außerdem mit Hilfe von Checkboxes ausgewählt werden, für welche Anlagen ein Produktionsbericht erstellt werden soll. Dem User wird außerdem die Möglichkeit geboten, zwischen der Erstellung einzelner Berichte

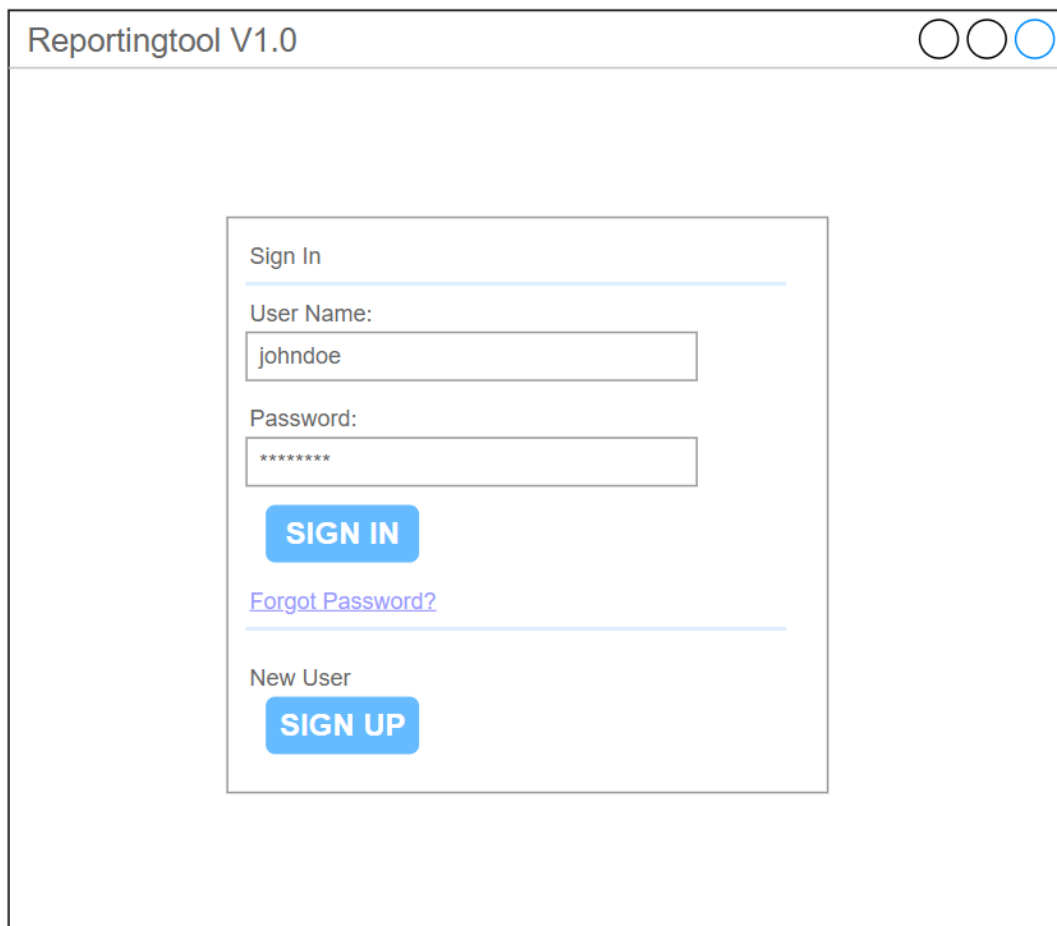


Abbildung 7.1: Mockup einer grafischen Benutzeroberfläche welche nach dem Start der Software angezeigt wird

für die jeweiligen Anlagen oder der Erstellung eines Gesamtberichtes zu wählen. Anschließend wird die Berichtserstellung mit einem Klick auf den 'Bericht erstellen' Button gestartet.

Nachdem alle Auswahlmöglichkeiten getroffen wurden und der Nutzer die Berichtserstellung startet, erscheint das in Abb. 7.4 dargestellte Fenster. Mittels einer Textausgabe wird der Nutzer darauf hingewiesen, dass die Erstellung des Berichtes begonnen hat. Zusätzlich wird dem Nutzer ein Ladebalken angezeigt, welcher den Fortschritt der Berichtserstellung anzeigt. Am unteren Rand des Fensters befinden sich zwei bislang ausgegraute Buttons mit den Beschriftungen 'PDF-Bericht(e) anzeigen' bzw. 'Excel-Bericht(e) anzeigen', welche zu diesem Zeitpunkt ohne Funktion sind. Erst wenn der Vorgang zu



Abbildung 7.2: Mockup einer grafischen Benutzeroberfläche welche nach Eingabe der Zugangsdaten angezeigt wird

100 % abgeschlossen ist, werden die Buttons wie in Abb. 7.4 zu sehen deutlich farblich hinterlegt. Der Nutzer hat nun die Möglichkeit, den erstellten Bericht entweder in Excel zu öffnen um eventuelle Modifikationen vorzunehmen oder den Bericht im PDF-Format mit einer geeigneten Software zu betrachten.

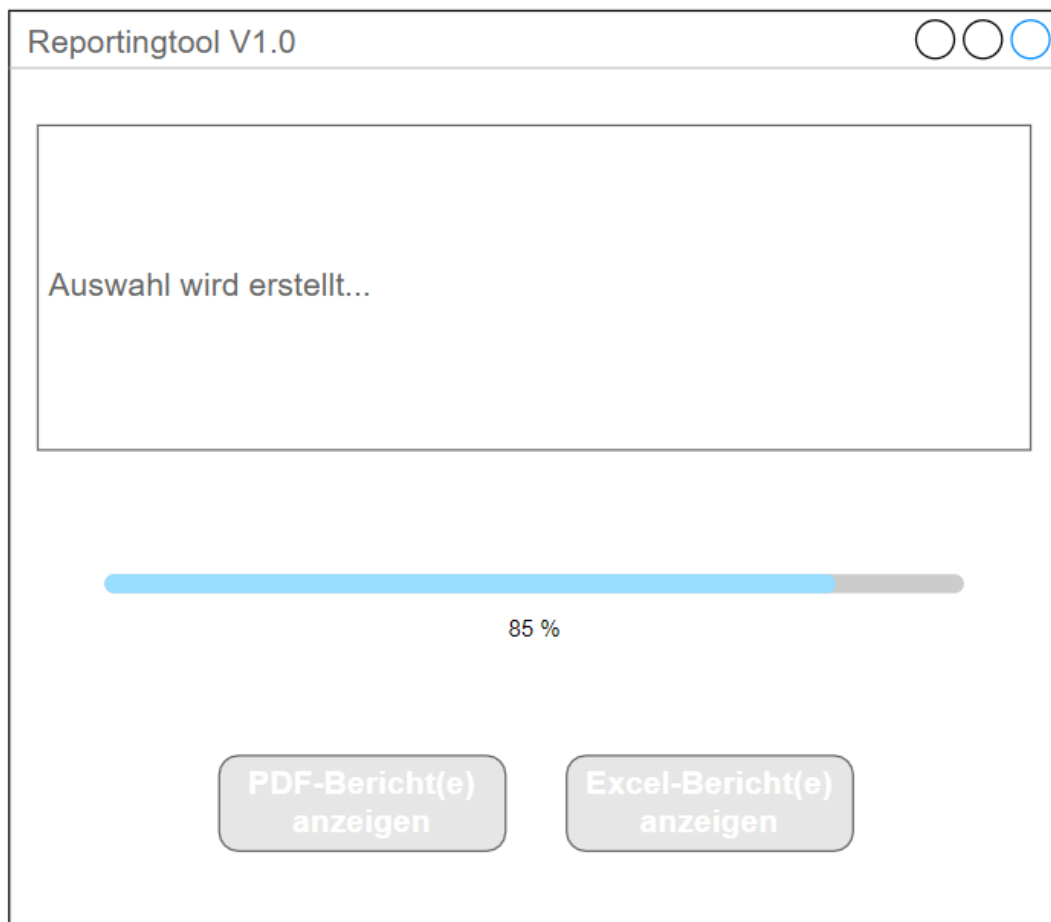


Abbildung 7.3: Mockup einer grafischen Nutzeroberfläche mit Auswahlmöglichkeiten für den Nutzer

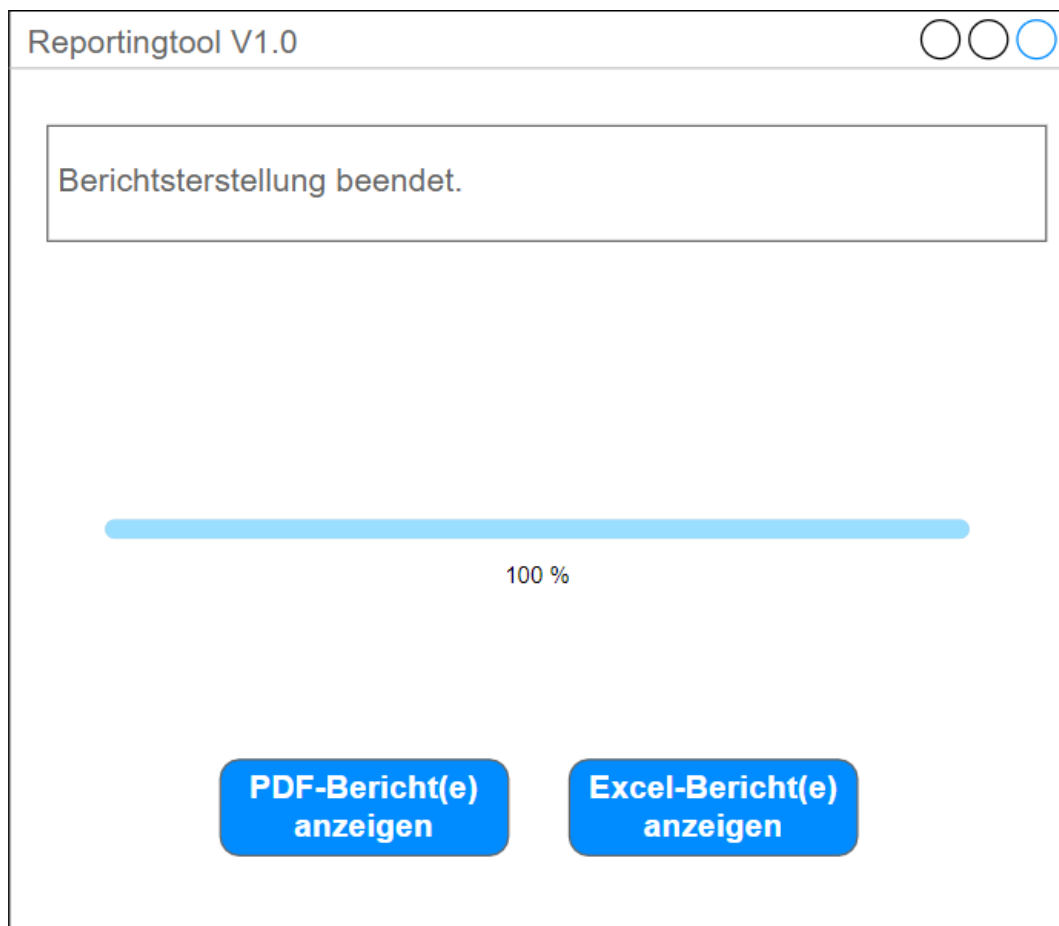


Abbildung 7.4: Mockup einer grafischen Nutzeroberfläche welche nach Fertigstellung des Berichtes angezeigt wird

Literaturverzeichnis

- [1] BUNDESMINISTERIUM FÜR WIRTSCHAFT UND ENERGIE: *Gesetz für den Ausbau erneuerbarer Energien (Erneuerbare-Energien-Gesetz - EEG 2017)*. 2019. – URL http://www.gesetze-im-internet.de/eeg_2014/EEG_2017.pdf. – Zugriffsdatum: 2021-04-15
- [2] BUNDESNETZAGENTUR: *EEG in Zahlen: Vergütungen, Differenzkosten und EEG-Umlage 2000 bis 2021*. 2019. – URL https://www.erneuerbare-energien.de/EE/Redaktion/DE/Downloads/eeg-in-zahlen-xls.xlsx?__blob=publicationFile&v=8. – Zugriffsdatum: 2021-04-15
- [3] CHUGHTAI, O Vogel I Arnold A. ; MEHLIG, E Ihler T Kehrer U. ; ZDUN, U: *Software-Architektur*. (2009)
- [4] GRANDE, Marcus: *100 Minuten für Anforderungsmanagement - Kompaktes Wissen nicht nur für Projektleiter und Entwickler*. Vieweg+Teubner Verlag |Springer Fachmedien Wiesbaden GmbH, 2011. – URL <https://link.springer.com/book/10.1007%2F978-3-8348-8135-9#toc>. – ISBN 978-3-8348-1431-9
- [5] HUNTER, John D.: Matplotlib: A 2D graphics environment. In: *IEEE Annals of the History of Computing* 9 (2007), Nr. 03, S. 90–95
- [6] IPCC: *Klimaänderung 2014 - Minderung des Klimawandels*. 2014. – URL https://www.ipcc.ch/site/assets/uploads/2019/03/IPCC_AR5_WGIII_FAQ_deutsch-1.pdf. – Zugriffsdatum: 2021-04-15
- [7] MCKINNEY, Wes u. a.: pandas: a foundational Python library for data analysis and statistics. In: *Python for High Performance and Scientific Computing* 14 (2011), Nr. 9, S. 1–9
- [8] OLIPHANT, Travis E.: *A guide to NumPy*. Bd. 1. Trelgol Publishing USA, 2006
- [9] OLIPHANT, Travis E.: Python for scientific computing. In: *Computing in Science & Engineering* 9 (2007), Nr. 3, S. 10–20

- [10] PEZOA, Felipe ; REUTTER, Juan L. ; SUAREZ, Fernando ; UGARTE, Martín ; VRGOČ, Domagoj: Foundations of JSON schema. In: *Proceedings of the 25th International Conference on World Wide Web*, 2016, S. 263–273
- [11] POSTEL, Jon ; REYNOLDS, JK: *RFC0959: File Transfer Protocol*. 1985
- [12] UMWELTBUNDESAMT: *Erneuerbare Energien in Deutschland Daten zur Entwicklung im Jahr 2019*. 2020. – URL https://www.umweltbundesamt.de/sites/default/files/medien/1410/publikationen/2020-04-03_hgp-ee-in-zahlen_bf.pdf. – Zugriffsdatum: 2021-04-15
- [13] VEREINTE NATIONEN: *Climate Change*. 2019. – URL <https://www.un.org/press/en/2021/sc14445.doc.htm>. – Zugriffsdatum: 2021-04-15
- [14] YATES, Andrew ; BEAL, Kathryn ; KEENAN, Stephen ; MCLAREN, William ; PIGNATELLI, Miguel ; RITCHIE, Graham R. ; RUFFIER, Magali ; TAYLOR, Kieron ; VULLO, Alessandro ; FLICEK, Paul: The Ensembl REST API: Ensembl data for any language. In: *Bioinformatics* 31 (2015), Nr. 1, S. 143–145
- [15] ZHOU, W. ; LI, L. ; LUO, M. ; CHOU, W.: REST API Design Patterns for SDN Northbound API. In: *2014 28th International Conference on Advanced Information Networking and Applications Workshops*, 2014, S. 358–365

A Digitaler Anhang

A.1 Formelles

- Bachelorthesis im PDF-Format

A.2 Softwarekomponenten

- Quellcode des Softwaretools
- Datei zur Installation der Software
- Datei zum Ausführen der Software
- README-Datei mit Installations- und Gebrauchsanweisung auf deutsch
- README-Datei mit Installations- und Gebrauchsanweisung auf englisch
- Excel-Bericht als Grundlage für weitere Berichte
- Ausgabebericht im PDF-Format

Erklärung zur selbstständigen Bearbeitung einer Abschlussarbeit

Hiermit versichere ich, dass ich die vorliegende Arbeit ohne fremde Hilfe selbständig verfasst und nur die angegebenen Hilfsmittel benutzt habe. Wörtlich oder dem Sinn nach aus anderen Werken entnommene Stellen sind unter Angabe der Quellen kenntlich gemacht.

Ort

Datum

Unterschrift im Original