
Projektbericht

Energieoptimierte Terminplanung

Philipp Arbogast (2351987), Christina Cser (2212074), Nils Heilemann
(2278387), Louis Mauser (2268421), Hannes Metz (2340041)



Projektseminar

Lehrstuhl für BWL und Wirtschaftsinformatik
Julius-Maximilians-Universität Würzburg

Betreuer: Prof. Dr. Axel Winkelmann
Bearbeitungszeitraum: 01.11.2022 bis 31.03.2023

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungsverzeichnis	IV
Tabellenverzeichnis	VII
1 Einleitung	1
1.1 Themenstellung und Motivation	1
1.2 Ziele und Vorgehen.....	2
1.3 Aufbau der Arbeit	3
2 Theoretische Grundlagen	4
2.1 Frameworks.....	4
2.2 Visualisierungstools	5
2.3 Schnittstellen	7
2.4 OPC UA und Streaming Data	8
2.5 Mathematische Optimierung.....	9
3 Projektvorbereitung und Projektmanagement	11
3.1 Project Canvas.....	11
3.1.1 <i>Teilbereich Zweck</i>	12
3.1.2 <i>Teilbereich Inputfaktoren</i>	12
3.1.3 <i>Teilbereich Projektdurchführung</i>	12
3.1.4 <i>Teilbereich Outputfaktoren</i>	15
3.1.5 <i>Teilbereich Zeit</i>	16
3.2 Lean Startup	16
3.2.1 <i>Personas und User Stories</i>	17
3.2.2 <i>MVP und Anforderungskatalog nach dem MoSCoW-Prinzip</i>	18
3.3 Agiles Projektmanagement	19
3.3.1 <i>Agile Vorgehensmodelle</i>	19
3.3.2 <i>Scrum Framework</i>	20
3.3.3 <i>Sprintablauf</i>	21
3.4 Tooleinsatz	22
3.4.1 <i>Kommunikation</i>	23
3.4.2 <i>Organisation</i>	23
3.4.3 <i>Entwicklungsumgebung</i>	24
3.4.4 <i>Programmierung Backend</i>	24

3.4.5	<i>Programmierung Frontend</i>	24
3.4.6	<i>Datenverarbeitung</i>	25
3.5	Reflexion des Projektmanagements	26
4	Frontend	27
4.1	Inhaltskonzept	27
4.1.1	<i>Landing Page</i>	28
4.1.2	<i>Dashboard</i>	28
4.1.3	<i>Terminplanung</i>	29
4.1.4	<i>Wetterbericht</i>	31
4.1.5	<i>Kalender</i>	32
4.1.6	<i>Photovoltaikanlage</i>	33
4.1.7	<i>Einstellungen und Profil</i>	33
4.2	Prozessmodellierung zur Terminerstellung	35
4.3	Designkonzept	37
4.3.1	<i>Branding</i>	37
4.3.2	<i>Globales Design</i>	38
4.4	Reflexion Frontend	40
5	Backend	41
5.1	Aufbau der Flask-Applikation	42
5.1.1	<i>Home-Route</i>	44
5.1.2	<i>PV-Anlage-Route</i>	46
5.2	Datenhaltung	47
5.2.1	<i>Relationale Datenbankspeicherung</i>	47
5.2.2	<i>Konfigurationsdatei im Applikationsverzeichnis</i>	49
5.3	Externe Komponenten	50
5.3.1	<i>Wetter-API</i>	50
5.3.2	<i>Streaming-Data-Platform</i>	52
5.3.3	<i>Solar-Vorhersage</i>	53
5.3.4	<i>Kalender-API (Microsoft Graph)</i>	57
5.3.5	<i>OPC UA</i>	60
5.4	Terminoptimierung	61
5.4.1	<i>Verarbeitung der Input-Daten</i>	63
5.4.2	<i>Optimierungsmodell</i>	66
5.4.3	<i>Umsetzung des Optimierungsmodells in Gurobi</i>	68

5.4.4	<i>Verarbeitung des Outputs</i>	69
5.4.5	<i>Ausgabe und Speicherung von Terminen</i>	70
5.5	API-Schnittstelle	72
5.5.1	<i>Authentifizierung</i>	72
5.5.2	<i>Nutzerverwaltung</i>	73
5.5.3	<i>Error Handling</i>	74
5.5.4	<i>OPC UA</i>	74
5.5.5	<i>Terminoptimierung</i>	75
6	Diskussion	78
6.1	Projektmanagement.....	78
6.2	SolarSync-System	79
7	Schlussfolgerung	82
7.1	Projektzusammenfassung.....	82
7.2	Ausblick	84
Literaturverzeichnis		86
Anhang		90
	Anhang 1 – Risikotabelle mit Risikoindex.....	90
	Anhang 2 – Protokolle.....	91
	Anhang 3 – Personas	99
	Anhang 4 – User Stories.....	102
	Anhang 5 – Detaillierte Dokumentation Sprintablauf.....	104
	Anhang 6 – Design Mockups	122
	Anhang 7 – Simulation der Photovoltaikanlage	126

Abbildungsverzeichnis

Abbildung 1: Project Canvas "Energiebasierte Terminplanung"	11
Abbildung 2: erwartete Einflüsse durch Projektstakeholder	14
Abbildung 3: Risikomatrix	15
Abbildung 4: nicht-maßstabsgetreuer Phasen- und Meilensteinplan	16
Abbildung 5: Prinzip der kleinen Pyramide (in Anlehnung Meyerbröker 2011)	19
Abbildung 6: Sprintablauf und Sprintziele	22
Abbildung 7: Technologie-Stack des Projekts „Energiebasierte Terminplanung“	23
Abbildung 8: Wireframe-Visualisierung Inhaltskonzept	27
Abbildung 9: Finales Release (Landing Page)	28
Abbildung 10: Release 6 (Dashboard).....	28
Abbildung 11: Release 6 (Terminabfrage)	30
Abbildung 12: Release 6 (Terminausgabe)	30
Abbildung 13: Release 6 (Terminbestätigung).....	31
Abbildung 14: Release 6 (Terminversand).....	31
Abbildung 15: Release 6 (Wetterausblick).....	32
Abbildung 16: Release 6 (Terminübersicht)	32
Abbildung 17: Finales Release (Auswertung Energiedaten PV-Anlage).....	33
Abbildung 18: Finales Release (Profilseite)	34
Abbildung 19: Finales Release (Administrationsbereich für Einstellungen)	34
Abbildung 20: Visualisierung des Ist-Prozesses	35
Abbildung 21: Visualisierung des Soll-Prozesses	36
Abbildung 22: Produkt Branding	38
Abbildung 23: Iterative Erstellung eines Layouts für SolarSync	39
Abbildung 24: Iterative Erstellung eines Farbkonzepts für SolarSync	40
Abbildung 25: Backend-Architektur	41
Abbildung 26: Grafische Darstellung der Optimierungsfunktion	63
Abbildung 27: MVP (Terminanforderung)	105
Abbildung 28: MVP (Terminausgabe)	105

Abbildung 29: Release 2 (Mehrfachauswahl im Formularfeld Maschinen)	106
Abbildung 30: Release 2 (Erweiterung um Infografiken zu vorgeschlagenen Terminen)	107
Abbildung 31: Release 2 (Download des erzeugten Terminvorschlags)	107
Abbildung 32: Release 2 (User-Logins)	108
Abbildung 33: Release 2 (responsives Design)	109
Abbildung 34: Release 3 (OPC UA Anbindung der PV-Anlage)	110
Abbildung 35: Release 3 (Integration von Wetterdaten).....	110
Abbildung 36: Release 3 (Anbindung Kafka-Streaming-Server).....	111
Abbildung 37: Erweiterung des Terminformulars um die Auswahl von Maschinen und Mitarbeitern	112
Abbildung 38: Release 3 (Übersicht über alle in der Anwendung generierten Termine)	112
Abbildung 39: Release 3 (Integration Emailversand)	113
Abbildung 40: Release 3 (Versand des erzeugten Termins)	113
Abbildung 41: Release 3 (Versandbestätigung)	114
Abbildung 42: Release 3 (Fehlermeldung bei falschen oder nicht existenten Nutzerdaten)	114
Abbildung 43: Release 3 (Erweiterung um Administrationsbereich)	115
Abbildung 44: Release 4 (Date Picker im Formular)	116
Abbildung 45: Release 4 (Dropdown-Auswahl für Mitarbeiter und Maschinen)	117
Abbildung 46: Release 5 (Erweiterung der Terminplanung um das Formularfeld Lötprodukt)	118
Abbildung 47: Release 5 (modularer Aufbau von SolarSync)	119
Abbildung 48: Release 5 (OPC UA Anbindung der Maschinen).....	119
Abbildung 49: Release 5 (erstes Dashboard Layout)	120
Abbildung 50: Release 5 (Custom Error Handling)	120
Abbildung 51: Finales Release (SolarSync Icon im Tab des Browsers)	121
Abbildung 52: Mockup Dashboard	122
Abbildung 53: Mockup Kalender	122
Abbildung 54: Mockup Wettervorhersage	123
Abbildung 55: Mockup PV-Anlage.....	123

Abbildung 56: Mockup Terminplanung (Formular ohne Dialog- und Auswahlfenster)	124
Abbildung 57: Mockup Terminplanung (Formular mit Dialog- und Auswahlfenster).....	124
Abbildung 58: Mockup Terminplanung (Terminausgabe mit Auswertungen)	125

Tabellenverzeichnis

Tabelle 1: Projektumfeld-Klassifizierung	13
Tabelle 2: Skizze der Applikationsstruktur	42
Tabelle 3: Grundlegende Komponenten der Flask-Applikation.....	43
Tabelle 4: Übersicht aller Routen in app/routes.py	44
Tabelle 5: Anlegen eines neuen Nutzers in der Datenbank über die Register-Route.....	48
Tabelle 6: Einloggen eines Nutzers in die Applikation	49
Tabelle 7: Abfrage der OpenWeatherMap-API.....	52
Tabelle 8: Parameter zum Erstellen eines Graph-Bearer Tokens	59
Tabelle 9: Übersicht der speziellen Attribute eines Maschinenbenutzers in M365	59
Tabelle 10: JSON-Format zum Anlegen eines Kalenderevents über Microsoft Graph	60
Tabelle 11: Variablen des Optimierungsproblems	66
Tabelle 12: Erstellen eines API-Tokens	72
Tabelle 13: Löschen eines API-Tokens.....	72
Tabelle 14: Liste aller Nutzer abfragen mittels API.....	73
Tabelle 15: Daten eines Nutzers abfragen mittels API.....	73
Tabelle 16: Anlegen eines neuen Nutzers mittels API	74
Tabelle 17: Maschinen mit OPC UA via API starten.....	74
Tabelle 18: Abfragen von MaschinenIDs mittels API	75
Tabelle 19: Abfrage von MitarbeiterIDs mittels API	75
Tabelle 20: Terminoptimierung mittels API durchführen	76
Tabelle 21: Speichern eines Terms durch die API	77
Tabelle 22: Fragebogen zur Evaluation des SolarSync-Systems	80
Tabelle 23: Risikotabelle mit Risikoindex	90

1 Einleitung

Das Projektseminar ist ein elementarer Bestandteil des Masterstudiengangs Information Systems. Entsprechend des Studienverlaufsplans dient das Projektseminar als letztes Modul vor der Masterarbeit dazu, die erlernten theoretischen Kenntnisse des Studiums in einem realen praktischen Projekt anzuwenden und zu vertiefen. Ziel ist es, für eine gegebene Aufgabenstellung ein Lösungskonzept zu erarbeiten und in Form einer Anwendung umzusetzen. Neben praktischer Erfahrung ermöglicht es aufgrund der regelmäßigen Interaktion mit Teammitgliedern sowie internen und externen Projektpartnern außerdem die Erweiterung von Soft Skills im Bereich Teamfähigkeit, Kommunikation und Präsentation. Die rund sechsmonatige Projektarbeit bietet daher eine ausgezeichnete Grundlage für den Berufseinstieg nach dem Studium. Im Rahmen dieses Projektseminars entstand die vorliegende Arbeit, die im Folgenden die Problemstellung beschreibt und vertieft auf die Umsetzung im Einzelnen eingeht.

1.1 Themenstellung und Motivation

Klimaschutz und Nachhaltigkeit ist bereits seit vielen Jahren ein allgegenwärtiges Thema. Im Klimaabkommen von Paris im Jahr 2015 verpflichteten sich über 195 Länder weltweit dazu, den Klimawandel einzudämmen und eine klimafreundlichere Weltwirtschaft aufzubauen. Das Ziel ist, bis 2050 klimaneutral zu werden. Konkret bedeutet das, den weltweiten Temperaturanstieg auf 1,5 Grad Celsius zu beschränken. Dies soll durch die Reduktion von klimaschädlichen Gasen und einer Umlenkung der globalen Finanzströme in nachhaltige Investitionen erreicht werden. Die EU hat zur Umsetzung des Abkommens den European Green Deal eingeführt (Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung 2023). Darüber hinaus gibt es einige Anreize in Form von Förderungen und Normen wie die ISO 50005, die insbesondere Unternehmen mit besonders hohem Energieverbrauch bei entsprechenden Maßnahmen unterstützen (Umweltbundesamt o. J.). Zusätzlich zu den Bemühungen im Rahmen des Klimaschutzes stellen die steigenden Energiepreise der letzten Monate eine erhebliche Herausforderung dar und wirkt auf Maßnahmen wie Investitionen in die Eigenstromproduktion wie ein Katalysator (Wirth o. J.).

Dieser Herausforderung hat sich auch die SEHO Systems GmbH (im Folgenden SEHO genannt) gegenübergesehen. Das Unternehmen mit Hauptsitz in Kreuzwertheim konzentriert sich auf Komplettlösungen für Lötprozesse und automatische Fertigungsprozesse (SEHO Systems GmbH o. J.). Zu Demonstrationszwecken und Schulungen für seine Kunden stellt das Unternehmen seine Maschinen in einem Ausstellungsraum aus. Die Nutzung des Ausstellungsraums erfordert einen signifikanten Energiebedarf, der im Zuge der Nachhaltigkeit und zur Reduzierung der gestiegenen Kosten primär durch die neu installierte

Photovoltaikanlage gedeckt und sekundär von zugekauftem Strom aus Wasserkraft ergänzt wird (SEHO Systems GmbH 2022).

Die Ergänzung des Systems um einen entsprechenden Energiespeicher ist mittelfristig angedacht, jedoch zum Zeitpunkt der Projektdurchführung nicht vorgesehen.

Die SEHO wird bei ihrem Projekt unterstützt von der Axsol GmbH (im Folgenden Axsol genannt), einem regionalen Unternehmen für autarke Energiespeichersysteme für mobilen, semi- und vollstationären Einsatz mit Standort in Würzburg (AXSOL GmbH o. J.). Darüber hinaus steht der SEHO ein weiterer regionaler Partner mit einem Energieaudit beratend zur Seite. Die WUQM Consulting GmbH (im Folgenden WUQM genannt), ebenfalls mit Sitz in Würzburg, bietet Qualitätsmanagement Consulting für Unternehmen an, um diese bei der schrittweisen Umsetzung und Optimierung prozessorientierter Managementsysteme mit den Schwerpunkten Umwelt- und Qualitätsmanagement zu unterstützen (WUQM Consulting GmbH o. J.).

Um den Energieverbrauch möglichst durch die eigene erzeugte Solarenergie zu decken und die extern bezogene Energie zu minimieren, möchte die SEHO die Terminvergabe für Kundenbesuche im Ausstellungsraum entlang der Leistungsspitzen der Photovoltaikanlage ausrichten. Aus diesem Bedarf heraus entstand das Projekt „Energiebasierte Terminplanung“.

1.2 Ziele und Vorgehen

Im Rahmen des Projektseminars zum Thema „Energiebasierte Terminplanung“ war die Aufgabe des fünfköpfigen Projektteams die Konzeption und Implementierung einer Web-Anwendung, mit deren Hilfe die SEHO die Planung ihrer Kundenbesuche basierend auf Stromerzeugung und -verbrauch automatisiert optimieren kann. Die Energieerzeugung ist abhängig von exogenen Parametern wie tageszeit-, saison- und wetterabhängigen Faktoren sowie endogenen Einflüssen wie der Kapazitätsgrenze der Photovoltaikanlage. Der Energieverbrauch ist neben der Grundlast, die das Unternehmen unabhängig vom Ausstellungsraum trägt, abhängig von Einsatzdauer und Anzahl der verwendeten Maschinen. Dementsprechend erhielt die Anwendung geeignete Schnittstellen an die jeweiligen Systeme wie Photovoltaikanlage, Kalender und Wettervorhersagen, um notwendige Daten auf einem Server im Backend zu hinterlegen. Auf dieser Basis wird ebenfalls im Backend mit Hilfe einer Optimierungsfunktion unter Berücksichtigung verschiedener Nebenbedingungen wie Arbeitszeiten, Feiertage oder bereits bestehender Termine im Kalender der jeweiligen Maschine oder des Mitarbeiters eine Reihe Terminvorschläge generiert, welche die Leistung der Photovoltaikanlage optimal ausnutzt. Eingabe, Aufbereitung und Visualisierung der Daten erfolgt im Frontend anwenderorientiert über ein Dashboard, zu dem der Anwender durch einen personalisierten Login Zugriff erhält.

1.3 Aufbau der Arbeit

Die Umsetzung der Projektaufgaben wird in der vorliegenden Arbeit detailliert dargelegt. Die Einleitung umfasst die Ausgangssituation und grundsätzliche Motivation für die vorliegende Projektarbeit sowie einen groben Abriss von Zielsetzung und Vorgehen. Der Hauptteil der Arbeit umfasst vier Teilbereiche. Das Kapitel Projektvorbereitung und Projektmanagement geht auf die Projektorganisation, Anwendung von Vorgehensmodellen sowie die verwendeten Werkzeuge ein. Im Kapitel Frontend werden Inhalts- und Designkonzept erläutert und der Terminerstellungsprozess modelliert. Das Kapitel Backend beschäftigt sich mit verwendeten Schnittstellen, technische Anforderungen und der Programmierung der Anwendung. Das Kapitel Diskussion rundet den Hauptteil in einer kritischen Auseinandersetzung ab. Den Schluss der vorliegenden Arbeit bilden die Projektzusammenfassung und der Ausblick.

2 Theoretische Grundlagen

Dieses Kapitel widmet sich den theoretischen Grundlagen, auf denen die Projektarbeit aus diesem Bericht aufbaut. Sie wurden bereits im Vorfeld in Form von Hausarbeiten umfangreich recherchiert, um ein solides Basiswissen zu schaffen. Im Folgenden werden die erarbeiteten Inhalte kurz aufgegriffen und erläutert, wie sie in die Projektarbeit eingeflossen sind.

2.1 Frameworks

Die Anforderungen an geschäftliche Software sind mit der stetigen Weiterentwicklung des Internets gestiegen. So muss eine Anwendung einen hohen Grad an Effizienz, Verlässlichkeit, Wartbarkeit und Skalierbarkeit aufweisen, um einem Unternehmen auf Dauer einen Mehrwert bieten zu können (Cui et al. 2009, 947).

Um diesen Anforderungen gerecht zu werden, hat sich die Verwendung von Frameworks in der Softwareentwicklung bewehrt. Ein Framework bildet einen vordefinierten Strukturrahmen, der dem Programmierer ermöglicht, verschiedene Softwareapplikationen zu entwickeln und zu organisieren (Shan und Hua 2006, 1 und 2). Dafür wird dem Entwickler eine Sammlung aus verschiedenen Bausteinen wie Bibliotheken, unterschiedliche Services, APIs (siehe Kapitel 2.3), Interfaces und weiteren Funktionen bereitgestellt (Shan und Hua 2006, 1 und 2). Durch die Verwendung eines Softwareframeworks können die vorgefertigten Komponenten, die für die Softwareanwendung benötigt werden, aus bestehenden Komponenten zusammengeführt werden und müssen nicht von Anfang an neu programmiert werden (Shan und Hua 2006, 1 und 2). Durch das Zurückgreifen auf bereits getestete und funktionierende Bausteine reduziert sich der Zeitbedarf. Der Entwickler kann somit bei einer niedrigen Anstrengung und geringem Zeit- und Ressourceneinsatz eine funktionsreiche und leistungsstarke Software programmieren (Shan und Hua 2006, 1 und 2). Zusätzlich wird die Fehleranfälligkeit des Codes reduziert und die Konzentration kann verstärkt auf dem betrieblichen Nutzen der Anwendung liegen (Shan und Hua 2006, 1 und 2). Die Vielfältigkeit von Frameworks umfasst Backend-, Frontend- und Fullstack-Frameworks. Diese unterscheiden sich in ihrem Funktionsumfang und in dem jeweiligen Einsatzbereich.

Zur Umsetzung des Projektseminars wird auf das Framework *Flask* zurückgegriffen. Es baut auf der *WSGI-Spezifikation* (Web Server Gateway Interface) auf, was die Kompatibilität mit Serversystemen wie Nginx und Apache sicherstellt (Flask 2010; GitHub Inc. 2023a). *Flask* ist ein Python-Webframework, dass sich den „*micro-frameworks*“ zuordnen lässt (Flask 2010; GitHub Inc. 2023a; IONOS SE 2023). Darunter definiert sich in diesem Zusammenhang ein Framework, dass nur die grundlegenden Funktionalitäten zur Verfügung stellt, sich aber leicht durch weitere Bibliotheken und Funktionen erweitern lässt oder bereits erweitert wird (Flask 2010). So kommt als Template-Engine bei *Flask Jinja2* standardmäßig zum Einsatz. Zur

Bereitstellung von Entwicklungstools und Debugging-Diensten wird *Werkzeug* eingesetzt (Flask 2010; GitHub Inc. 2023a). Im Frontend ist ebenso die Erweiterung von *Flask* durch die Implementierung weiterer Bibliotheken und Frameworks möglich. Da im Rahmen des Projektseminars die Erstellung eines strukturierten und übersichtlichen User Interfaces erforderlich ist, wird das Framework durch *Bootstrap* und *SCSS* erweitert. *Bootstrap* ist ein Open-Source Framework, das vorgefertigte HTML-, CSS- sowie JavaScript-Komponenten enthält und die Erstellung von Websitelayouts und Funktionen im User Interface vereinfacht (Bootstrap 2023). Um die Ausdrucksfähigkeit und Lesbarkeit der CSS zu verbessern, wird im Projektseminar die CSS-Präprozessor-Sprache *SCSS* (Sass Cascading Style Sheet) verwendet. Durch die Verwendung von *SCSS* kann sich wiederholender Code reduziert werden und eine übersichtlichere Darstellung bei reduzierter Codelänge erzielt werden (Sass 2023).

Der kombinierte Einsatz des beschriebenen Technologie-Stacks schafft eine auf mehreren Ebenen vorteilhafte Basis der Implementierung: Der Stack ermöglicht es, dass die verschiedenen Komponenten im Backend und Frontend problemlos miteinander kommunizieren können. Zusätzlich wird durch den Einsatz von vorgefertigten Komponenten der Zeit- und Ressourcenaufwand in der Entwicklungsphase reduziert. Der schlanke Aufbau und die Erweiterbarkeit ermöglichen es in Zukunft, dass weitere Frameworks und Bibliotheken eingesetzt werden können, wenn es zusätzliche Anforderungen notwendig machen. Die Bestandteile des verwendeten Technologie-Stacks erfreuen sich einer großen Community. Das stellt eine nicht zu vernachlässigende Rolle bei der Implementierung von Projekten dar (GitHub Inc. 2023a): Entwickler tauschen sich regelmäßig über Plattformen wie GitHub aus und können Lösungen für auftretende Probleme im Projekt liefern.

2.2 Visualisierungstools

Im Rahmen des vorliegenden Projekts wurde die Anwendung SolarSync entwickelt, die es dem Nutzer ermöglicht, Kundentermine basierend auf der von der unternehmenseigenen PV-Anlage erzeugten Energie zu planen. Die Anwendung integriert verschiedene interne und externe Datenquellen, die dem Nutzer neben den funktionalen Elementen ergänzend zur Verfügung gestellt werden, um die generierten Termine logisch nachvollziehbar und transparent zu gestalten. Diese Inhalte wurden als Dashboard konzipiert.

Ein Dashboard ist eine Geschäftsanwendung mit einer umfangreichen Benutzeroberfläche, die Benutzern ermöglicht, mit Daten zu interagieren. Zu den wichtigsten Elementen der Benutzeroberfläche eines Dashboards gehören das Layout, allgemeine Formatierungs- bzw. Stilkomponenten sowie Steuerelemente für die Benutzerinteraktion. Dashboards sind dabei stark visualisierungsgtrieben (Sprotten 2022). Oftmals genügt es nicht, die Daten mit Hilfe nur einer einzelnen Grafik zu visualisieren. Die Zusammenfassung verschiedener Diagramme in

einem Dashboard erlaubt es dem Entscheider, verschiedene Sichten auf entscheidungsrelevante Daten zu erlangen (Kahl und Zimmer 2017).

Bei der Aufbereitung von Daten steht die Frage im Mittelpunkt, welche Technik am besten geeignet ist, um die gewünschte Information zu vermitteln (Sprotten 2022). Das Ziel ist eine verständliche Darstellung der Daten. Um eine optimale Kommunikation durch grafische Darstellungen zu erreichen, widmen sich spezialisierte wissenschaftliche Disziplinen wie das Informationsdesign oder die Kommunikationswissenschaft der Erstellung von Richtlinien (Sprotten 2022).

Im Vorfeld der Projektarbeit wurden verschiedene Quellen recherchiert, um Richtlinien für die Erstellung von qualitativ hochwertigen Grafiken zu finden. Drei relevante Quellen wurden identifiziert. Die International Business Communication Standards (IBCS) bieten Hilfestellungen für die inhaltliche Konzeption, die visuelle Wahrnehmbarkeit und die semantische Notation von Berichten, Präsentationen und Dashboards (IBCS Association 2022). Der CLEAR(I) Leitfaden von Jacobs und Hensel-Börner (2020) verfolgt einen ähnlichen Ansatz und legt seinen Fokus auf Visualisierungen für Dashboards (Jacobs und Hensel-Börner 2020). Ben Shneidermans acht goldene Regeln des Interface Designs hingegen bieten allgemeine Prinzipien für die Gestaltung von Benutzeroberflächen und Interaktionsdesigns, um die Benutzerfreundlichkeit und Effektivität zu verbessern (Shneiderman et al. 2016).

Das Inhalts- und Designkonzept wurde unter Berücksichtigung der zuvor identifizierten Richtlinien erarbeitet und sämtliche Grafiken wurden entsprechend gestaltet. Bei der Umsetzung wurde besonders auf Klarheit, Einfachheit, Lesbarkeit und Verständlichkeit geachtet, um dem Betrachter eine schnelle und effektive Informationsaufnahme zu ermöglichen. Durch die Einhaltung der Richtlinien soll sichergestellt werden, dass die Grafiken nicht nur ästhetisch ansprechend, sondern auch sachlich korrekt und aussagekräftig sind.

Um die theoretischen Grundsätze der Datenvisualisierung in der Praxis anzuwenden, sind geeignete Visualisierungswerkzeuge erforderlich. In der Regel werden Visualisierungen und Dashboards mithilfe von JavaScript im Frontend entwickelt. JavaScript bietet verschiedene Frameworks und Bibliotheken, die eine effiziente Entwicklung von Anwendungen ermöglichen. Die Auswahl der richtigen Bibliothek ist von großer Bedeutung, da der gesamte Code der Grafiken darauf aufbaut. Aus 80 identifizierten Bibliotheken wurden drei nach zuvor festgelegten Auswahlkriterien in die engere Auswahl genommen. Im Anwendungstest mit einer prototypischen Datenvisualisierung fiel die Wahl auf die JavaScript-Bibliothek ChartJS, die eine ausreichende Auswahl an Diagrammtypen bietet und auch von Programmierern mit weniger umfangreichen Kenntnissen leicht erstellt werden kann. Die einfache Syntax und die unkomplizierte Anbindung der Bibliothek an die Webanwendung machten ChartJS besonders attraktiv für die Projektarbeit.

2.3 Schnittstellen

Geschäftliche Software ist ein essenzieller Bestandteil von Unternehmen geworden, unabhängig von deren Größe (Ofoeda et al. 2019, 76). Die zunehmende Bereitstellung von Geschäftssoftware bedeutet für Entwickler einen enormen Arbeitsaufwand, da viele Aktivitäten, Funktionen und Integrationen zu anderen Softwarekomponenten konzeptioniert und umgesetzt werden müssen (Ofoeda et al. 2019, 76). Diese Entwicklung hat zu einer zunehmenden Einführung und Nutzung von Application Programming Interfaces (API) geführt (Lamothe et al. 2021, 1). Eine API ist eine Technologie, die den Austausch von Nachrichten oder Daten zwischen verschiedenen Softwareanwendungen ermöglicht (de Souza et al. 2004, 64). Heutzutage hat die Omnipräsenz von APIs einen großen Einfluss auf die Softwareentwicklung und führt zu Zeit- und Aufwandseinsparungen für Endnutzer und Programmierer, da eine vorgefertigte Funktionalität beliebig oft wiederverwendet werden kann (Lamothe et al. 2021, 1f.).

Die für das Projektseminar entwickelte Softwarelösung befasst sich mit der energiebasierten Planung von Terminen. Mittels eines Optimierungsmodells sollen Termine in freien Zeitslots mit möglichst hohem Anteil an eigener Energieproduktion realisiert werden. Aus dieser Anforderung geht hervor, dass zu einer möglichst präzisen Energieplanung zwei API-Integrationen notwendig sind. Eine API kommt im Projektseminar bei der Abfrage von Wetter- und Solardaten zum Einsatz. Zudem wird eine Kalender-API zum Abfragen freier Zeiten sowie zur Buchung von Terminen mit Mitarbeitern und Maschinen eingesetzt. Als Wetter-API wird der Anbieter OpenWeatherMap (<https://openweathermap.org/>) mit deren „Climatic Forecast 30 days“ verwendet. Die Kombination aus dem Grad der Bewölkung dieses Berichts mit historischen Wetterdaten wird vom Optimierungsmodell zur Wettervorhersage der nächsten 30 Tage genutzt. Als Kalender-API ist ursprünglich der Google Calendar geplant gewesen, da dieser einfach zu implementieren ist und mit einem eigenen Python-Package (google-api-python-client) geliefert wird. Aus der Zusammenarbeit mit der SEHO Systems GmbH hat sich allerdings ergeben, dass das Unternehmen plant, Microsoft 365 und den damit verbundenen Microsoft Kalender einzuführen. Daher wird im Projektseminar die Microsoft-Graph-API genutzt, womit ein gesamter Mandant eines Unternehmens auf einmal angebunden werden kann. Die Microsoft Graph-API wird im Projektseminar dazu genutzt, alle Nutzer eines Mandanten abzufragen. Zudem wird sie genutzt, um bei M365-Accounts zwischen Maschinen und Mitarbeiteraccounts zu unterscheiden, Termine aller oder ausgewählter Mitarbeiter in einem Zeitraum anzuzeigen und neue Termine nach erfolgreicher Planung einzutragen.

Neben der einfachen Änderbarkeit bringt die API-Nutzung mit Portabilität, Komplexitätsreduktion, Modularisierung und Stabilität durch lose Kopplung mehrere Vorteile mit sich (Spichale 2019). Diese Vorteile von APIs sollen nicht nur dem Entwicklerteam der Softwarelösung des Projektseminars die Arbeit erleichtern, sondern auch Endanwendern des

Portals zur energiebasierten Terminplanung einen Mehrwert liefern. Daher besitzt die entwickelte Softwarelösung einen eigenen API-Endpunkt (<https://www.demotenant.com/api/>) mit Token-Authentifizierungsverfahren. Dieser ermöglicht es, Nutzer des Webportals anzuzeigen sowie neu zu erstellen und IDs von Maschinen und Mitarbeitern zur Terminplanung abzufragen. Als abschließenden Punkt können über den API-Endpunkt auch optimierte Terminvorschläge generiert und auch weiterführend in der Datenbank sowie in den angebundenen Office-Kalendern gespeichert werden.

2.4 OPC UA und Streaming Data

Der Begriff Streaming Data beschreibt einen kontinuierlichen Strom an Daten, die in Echtzeit übertragen werden (Tatbul 2010, 156). Diese Daten können aus verschiedenen Quellen stammen, z.B. Sensoren, Maschinen oder anderen IoT-Geräten, die permanent Informationen bereitstellen (Muthukrishnan 2005, 128-130). Im Zuge der Digitalisierung der Industrie hat Streaming Data an Bedeutung gewonnen, da es Unternehmen und Organisationen ermöglicht, in Echtzeit Daten zu sammeln, zu verarbeiten und zu analysieren (Tatbul 2010, 157). Durch die Echtzeitverarbeitung können Unternehmen schneller auf Veränderungen reagieren, Trends erkennen und Entscheidungen treffen, die auf aktuellen Daten basieren (Gaber et al. 2005, 18-20). OPC UA (Open Platform Communications Unified Architecture) ist ein plattformunabhängiges Framework für die industrielle Automatisierung (Hannelius et al. 2008, 756 und 757). Es bietet eine sichere, zuverlässige und interoperable Möglichkeit für die Kommunikation zwischen verschiedenen Geräten und Anwendungen (Mahnke et al. 2009, 8ff). Im Bereich von Streaming Data kann OPC UA eine wichtige Rolle spielen, da es eine Möglichkeit bietet, Daten in Echtzeit zu übertragen (Schwarz und Borcsok 2013, 1 und 2). Zudem kann es das Potential, die Effizienz und Leistung von IoT-Anwendungen zu verbessern, indem es eine nahtlose Integration von Daten aus verschiedenen Quellen ermöglicht (Imtiaz und Jasperneite 2013, 500-505).

Im Rahmen des Projektseminars spielt Streaming Data an zwei Stellen eine entscheidende Rolle: Bei dem Input an relevanten Daten für das Optimierungsmodell und bei dem Output von Steuersignalen. Für den Input an entscheidungsrelevanten Informationen wurde eine Streaming-Plattform mit *Apache Kafka* erschaffen. In dieser senden sogenannte *Producer* Daten in einer festgelegten Frequenz an die Streaming Plattform, die diese Daten sowohl in Echtzeit an *Consumer* weitergeben als auch zwischenspeichern kann. Als *Producer* dienen dabei die Schnittstellen (in unserem Fall die OpenWeatherMap-API). Diese Daten werden dann von den *Consumern* (in unserem Fall dem Optimierungsmodell) abgerufen und verarbeitet. Beim Output von Steuersignalen kommt im Projektseminar OPC UA zum Einsatz. Unsere Anwendung dient als *OPC UA-Client*. Wenn ein Termin bevorsteht und eine Maschine

vorheizen soll, sendet der Client die Steuersignale (An/Aus) in Echtzeit an die Maschinen, die jeweils einen *OPC UA-Server* repräsentieren. (Apache Software Foundation 2022; Garg 2013)

Dieser Aufbau liefert eine Reihe an Vorteilen. Durch den Einsatz der Streaming-Plattform wird sichergestellt, dass auch bei einem temporären Ausfall der Schnittstellen die zuletzt abgerufenen Daten zur Verfügung stehen und so das Optimierungsmodell trotzdem ausgeführt werden kann. Zudem muss nicht bei jeder Terminplanung ein neuer API-Call ausgeführt werden, der mit Kosten verbunden ist. Stattdessen bekommt der Streaming-Server die Daten in einer festen Taktung und gibt diese zwischengespeichert beliebig oft an das Optimierungsmodell weiter. Der Einsatz von OPC UA ermöglicht einen einheitlichen Austauschstandard. Dadurch können alle denkbaren Maschinen, unabhängig des Herstellers und des eingesetzten Technologie-Stacks, in die Planungssoftware eingebunden werden. Das ist gerade in der Zukunft, wenn noch weitere Unternehmen das Ergebnis des Projektseminars einsetzen möchten, ein großer Vorteil. (Huang und Garcia-Molina 2001; Tarkoma 2012; Tatbul 2010)

2.5 Mathematische Optimierung

Das Konzept der mathematischen Optimierung teilt sich in dem folgenden Anwendungsfall in zwei Forschungsbereiche auf. Zum einen dient die Termin- und Kapazitätsplanung der Einplanung verschiedener Produktionsaufträge auf eine kapazitätsbeschränkte Fertigungsanlage (Steinrücke 2007, 1). Die Kapazität wird dabei durch verschiedene Faktoren begrenzt, wie beispielsweise die Anzahl der Maschinen. Das grundlegende Ziel besteht dabei darin, die produktionsbezogenen Kosten zu minimieren. Um die Modelle der Termin- und Kapazitätsplanung zu lösen, werden häufig Methoden der mathematischen Optimierung eingesetzt (Domschke et al. 2005, 4). Hierbei wird das jeweilige Problem in ein mathematisches, meist lineares, Optimierungsmodell überführt. Dieses Modell besteht aus einer Zielfunktion, die maximiert oder minimiert wird, und aus Nebenbedingungen, welche den Lösungsraum des Problems beschränken. Mit verschiedenen Algorithmen werden diese mathematischen Optimierungsmodelle, unter der Zuhilfenahme von Softwarelösungen, gelöst und eine oder mehrere optimale Lösungen für das Problem gefunden (Conforti et al. 2014, 5).

Die Techniken der mathematischen Optimierung bilden die Basis für die Anwendung der energiebasierten Terminplanung, welche in diesem Projektseminar entwickelt wurde. Die grundlegende Anforderung ist die automatisierte Planung von Terminen zur Belegung von energieintensiven Lötanlagen unter dem Ziel der Minimierung der Strommenge, welche hierfür aus dem Netz bezogen wird. Die Zielfunktion sowie die Nebenbedingungen dieses Optimierungsproblems werden in Python mithilfe des Gurobi-Solvers formuliert. Die Lösung des Problems erfolgt ebenfalls durch die Lösungsalgorithmen von Gurobi, welche über die Python-Bibliothek GurobiPy aufgerufen werden. Die Grundlage der Optimierung bilden zum

einen Daten einer Vorhersage der zukünftig erzeugten Solarenergie und den, in der Anwendung anpassbaren, Werten der Stromverbräuche der Maschinen sowie der Grundlast. Zum anderen werden die Nutzereingaben bezüglich des Planungshorizont, der Termindauer, der benötigten Maschinen bzw. Mitarbeitern sowie der Komplexität des Lötprodukts berücksichtigt. Die Eigenschaften des zu lötenden Produkts beeinflussen den Energieverbrauch der Maschinen. Basierend auf diesen unterschiedlichen Daten wird der theoretische Netzbezug von jedem möglichen Termin im Planungshorizont berechnet und minimiert. Als Ausgabe der Optimierung werden dem Nutzer drei energieoptimale Termine zur Maschinennutzung vorgeschlagen.

Die mathematische Modellierung des Optimierungsproblems bietet mehrere Vorteile. Durch die klar strukturierte Formulierung der Zielfunktion sowie der Nebenbedingungen sind diese leicht anpassbar und können somit gut an die Anforderungen der SEHO angepasst werden. Zudem ist das Optimierungsmodell durch diese hohe Flexibilität auf unterschiedliche Anwendungsfälle in weiteren Unternehmen mit anderen Daten und Anforderungen übertragbar. Ein weiterer Vorteil ist die Verwendung der effizienten Lösungsverfahren, die von Gurobi bereitgestellt werden. Diese sorgen für eine schnelle Berechnung des Optimierungsproblems, sodass der Benutzer nicht lange auf die optimalen Terminvorschläge warten muss.

3 Projektvorbereitung und Projektmanagement

In diesem Kapitel werden die Projektvorbereitung und das Projektmanagement ausführlicher dargestellt. Im Rahmen der Projektvorbereitung erfolgte eine Auseinandersetzung mit dem Project Canvas sowie die Anwendung der Lean Startup Methode des Projektmanagements. Die Projektdurchführung und Organisation richteten sich in erster Linie nach den Regelungen des Scrum Frameworks und betrachtet den projektrelevanten Tooleinsatz. Eine Reflexion des Projektmanagement der angewendeten Methoden schließen das Kapitel ab.

3.1 Project Canvas

Der Project Canvas ist ein im Projektmanagement etabliertes Modell, das die komplexen Sachverhalte eines Projekts in thematisch gruppierte Bereiche zerlegt. Ziel ist es für Struktur und Übersichtlichkeit des Projekts zu sorgen. Abbildung 1 zeigt den Project Canvas des Projekts "Energiebasierte Terminplanung". Auf die Teilbereiche wird im Folgenden eingegangen.

PROJECT CANVAS					
Zweck	Unterstützung von Unternehmen bei der Umgestaltung ihres Energiemanagements im Rahmen der Nachhaltigkeit und Kosteneffizienz Planungsoptimierung von Kundenbesuchen basierend auf der Energieerzeugung einer Photovoltaikanlage				
Budget alle notwendigen Hard- und Software steht bereits zur Verfügung keine finanziellen Investitionen im Rahmen des Projekts notwendig	Team Projektteam - Hannes Metz - Philipp Arboagast - Nils Heilemann - Louis Mauser - Christina Cser Betreuerteam - Nicolas Neis - Fabian Gwinner - Lukas-Valentin Herm - Christoph Tomitsa	Umfeld soziales internes Umfeld Projektteam, Betreuerteam, Auftraggeber soziales externes Umfeld externe Projektteilnehmer, Mitarbeiter & Kunden der SEHO Systems GmbH sachliches internes Umfeld Projektmanagement & Scrum Handbuch, - Produktions- & Testumgebung, Hard- & Softwareverfügbarkeit sachliches externes Umfeld Gesetze, Normen, Standards	Etappenziele Einarbeitung in Unterthemen während Vorbereitungsphase Vorbereitung des Projekts mit Methoden des modernen Projektmanagements Erstellung eines Anforderungskatalogs Konzeptentwicklung als Grundlage für das MVP Entwicklung eines Prototyps (MVP) Erweiterung des MVP um weitere Anforderungen im Anforderungskatalog Entwicklung Desigkonzept visuelle Optimierung der Web-Anwendung auf Basis entsprechender Designkonzepte Projektabchluss durch Projektbericht	Qualität Funktionen des finalen Produkts wird die Terminplanung des Auftraggebers unterstützen und zu energieeffizienterer Nutzung der Maschinen sorgen Der Kunde wird zu einem nachhaltigeren Energiemanagement beitragen und finanzielle Erleichterung durch optimierte Energiekosten erfahren	Ergebnis Web-Anwendung, mit deren Hilfe energieeffiziente Terminplanung durchgeführt werden kann (Kategorie Produkt) Kunde Projektleiter: Betreuerteam Projektempfänger: SEHO Systems GmbH Projektkontakt: Axsol GmbH WUQM Consulting GmbH
Ressourcen Kompetenzen des Projektteams - Wissen und Daten von Betreuerteam, Auftraggeber und externen Projektteilnehmern - technische Infrastruktur des Projektteams - Arbeitsräume privat - Arbeitsräume universitär	Risiken & Chancen Risiken: - Umsetzung entspricht nicht den Anforderungen - unfertiges Produkt am Ende des Projektzeitraums Chancen: - Potenzial auch für andere Unternehmen als Produkt in Frage zu kommen				
Zeit 08.09.2022 - 23.10.2022 Pre-Kick-Off Vorbereitungsphase	24.10.2022 - 09.11.2022 Einarbeitung Orientierungsphase	10.11.2022- 10.02.2023 Kick-Off Entwicklung Backend	11.02.2023 - 05.03.2023 Entwicklung Frontend	06.03.2023 - 31.03.2023 Abschlussphase Projektbericht	Start- und Endtermin fest vorgegeben zeitliche Aufteilung innerhalb des Rahmens frei

Abbildung 1: Project Canvas "Energiebasierte Terminplanung"

3.1.1 Teilbereich Zweck

Zu Beginn eines Projekts ist es wichtig, die Ziele des Auftraggebers zu identifizieren, um die Zielgruppe bestimmen zu können. Zweck des Projekts "Energiebasierte Terminplanung" ist, Unternehmen bei der Umgestaltung ihres Energiemanagements im Rahmen der Nachhaltigkeit und Kosteneffizienz zu unterstützen. Dies erfolgt durch die Planungsoptimierung von Kundenbesuchen basierend auf der Energieerzeugung einer Photovoltaikanlage, um den Energieverbrauch durch eigene Solarenergie zu decken. Die Zielgruppe sind Unternehmen, die eine Photovoltaikanlage nutzen und den Einsatz ihrer Energieträger daran ausrichten möchten.

3.1.2 Teilbereich Inputfaktoren

Nach der Festlegung der Ziele ist es wichtig, die verfügbaren Inputfaktoren zu ermitteln. Hierbei bezieht sich der Project Canvas auf drei Schlüsselfaktoren: Budget, Team und Ressourcen.

Das Budget bezieht sich auf die finanziellen Mittel, die für das Projekt benötigt werden und zur Verfügung stehen. Das Team bezieht sich auf die Rolle und Zugehörigkeit jedes Mitglieds, während die Ressourcen alle Arbeitsmittel, Materialien, Methoden, Modelle, Arbeits- und Besprechungsräume umfassen, die für den erfolgreichen Abschluss des Projekts benötigt werden.

Im vorliegenden Projekt sind keine finanziellen Investitionen notwendig, da dem Team alle notwendigen Hard- und Softwareanforderungen bereits zur Verfügung stehen. Das Kernteam setzt sich aus fünf Mitgliedern zusammen und wird durch das Betreuerteam des Lehrstuhls für BWL und Wirtschaftsinformatik der Julius-Maximilians-Universität Würzburg unterstützt. Die Ressourcen umfassen das Projektteam, das Wissen und die Daten, die durch Auftraggeber und externe Projektteilnehmer zur Verfügung stellen, sowie die technische Infrastruktur, die durch die Projektmitglieder bereitgestellt wird.

3.1.3 Teilbereich Projektdurchführung

Nachdem die Inputfaktoren identifiziert wurden, kann die Planung des Projekts angegangen werden. In diesem Zusammenhang werden im Project Canvas drei Bereiche besprochen: das Umfeld, die Risiken und Chancen sowie die Etappenziele.

Das Umfeld bezieht sich auf die Ergebnisse einer Umfeldanalyse. Hier werden Faktoren identifiziert und klassifiziert, die das Projekt beeinflussen oder von ihm beeinflusst werden. Dabei unterscheidet man nach sozialen Faktoren wie Personen oder Personengruppen und nach sachlichen Faktoren wie Standards, Gesetze oder Marktbedingungen. Diese werden jeweils

nach internen und externen Faktoren unterschieden. Die Klassifizierung für das Projekt „energiebasierte Terminplanung“ ist in Tabelle 1 erfasst.

Tabelle 1: Projektumfeld-Klassifizierung

	Soziale Faktoren (Personen, Gruppen)	Sachliche Faktoren (Standards, Gesetze, Marktbedingungen)
Interne Faktoren	<ul style="list-style-type: none"> • Projektteam • Betreuerteam • Lehrstuhlinhaber • Auftraggeber 	<ul style="list-style-type: none"> • Kompetenzen des Projektteams • Projektmanagement Handbuch • Scrum Handbuch • Produktions-/Testumgebung • Hard- und Softwareverfügbarkeit
Externe Faktoren	<ul style="list-style-type: none"> • Externe Projektteilnehmer (Axsol, WUQM) • Mitarbeiter der SEHO • Kunden der SEHO 	<ul style="list-style-type: none"> • Gesetzgebung • IT-Sicherheit • Datenschutz • Normen und Standards • Mitbewerbersituation • Marktentwicklung

Die Überlegungen zu sozialen internen, wie externen Faktoren basieren auf den Konzepten des Stakeholdermanagements. Dabei werden relevante Stakeholder identifiziert und bewertet, um deren Einfluss und Einstellung zum Projekt zu ermitteln. Basierend auf dieser Analyse können Maßnahmen eingeleitet werden, um negative Auswirkungen zu vermeiden oder zu mildern und um positive Auswirkungen zu verstärken. Wir nehmen an, dass Mitarbeiter und Kunden des zukünftigen Anwenderunternehmens keinen Einfluss auf das Projekt nehmen können und dass deren Einstellung von neutraler Natur ist. Externe Projektteilnehmer wie die Axsol und die WUQM stufen wir als positiv wohlwollend hinsichtlich des Projekts ein. Jedoch unterstellen wir einen mittelstarken Einfluss auf das Projekt und dessen Umsetzung, da sie sich in keiner entscheidungsrelevanten Position befinden, die das Projekt endgültig terminieren. Allerdings können sie positiv auf die weitere Entwicklung einwirken. Ähnlich schätzen wir den Auftraggeber ein. Dies erscheint zunächst kontraintuitiv, da in der Regel der Auftraggeber alle Anforderungen an das Projekt definiert und ultimativ über Inhalte, Umsetzung und Durchführung entscheidet. Im Rahmen des Projektseminars unterliegt das Projekt jedoch einer Sonderstellung, da der Auftraggeber mit seinem Bedarf zwar den Anstoß und die Richtung vorgibt, nach Beginn des Projekts nur noch richtungsweisend auf das Projekt einwirken kann. Dem gegenüber stehen das Betreuerteam und übergeordnet der Lehrstuhlinhaber als Initiator des Projekts im Rahmen des Projektseminars in einer deutlich mächtigeren Position, da sie neben dem Projektteam selbst maßgeblich auf das Projekt einwirken und dieses sogar

terminieren können. Wir unterstellen diesen Parteien ein grundsätzlich positives Interesse am Projekterfolg. Nach Evaluation der Projektstakeholder hinsichtlich Einstellung und Einfluss erwartet das Projektteam durchweg positives bis neutrales Feedback, sodass das Projekt ohne Widerstände durchgeführt werden kann (vgl. Abbildung 2). Darüber hinaus impliziert das Ergebnis, dass ein enger Kontakt zwischen dem Projektteam und dem Betreuerteam bestehen und ein regelmäßiger Informationsaustausch zwischen dem Projektteam und den Parteien Auftraggeber und externe Projektteilnehmer stattfinden sollte.

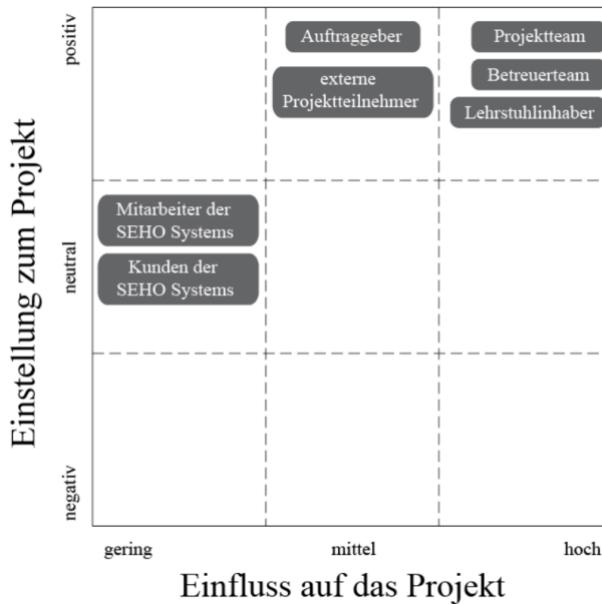


Abbildung 2: erwartete Einflüsse durch Projektstakeholder

Auf Basis der Umfeldanalyse werden in der zweiten Komponente Risiken und Chancen betrachtet, die den Projekterfolg beeinflussen können. Risiken werden als mögliche negative Abweichungen von der Projektplanung durch ungeplante Ereignisse oder Umstände definiert (DIN Deutsches Institut für Normung e. V. 2009).

Im Rahmen des Risikomanagements werden sie nach ihrer Ursache kategorisiert und auf qualitative und quantitative Merkmale hin analysiert und bewertet. Eine subjektive Reflexion führt zu einer Einteilung in fünf Stufen, wobei eins die niedrigste Eintrittswahrscheinlichkeit bzw. Schadenshöhe darstellt. Der Grad des Risikos wird berechnet, indem man die Eintrittswahrscheinlichkeit mit der Schadenshöhe multipliziert. Die für das Projekt „energiebasierte Terminplanung“ identifizierten Risiken sind in Tabelle 23 im Anhang 1 – Risikotabelle mit Risikoindex aufgeführt.

Durch die Visualisierung der Risiken in der Risikomatrix lässt sich ablesen, dass sich alle identifizierten Risiken in einem unkritischen Bereich befinden und durch präventive und korrektive Strategien gemindert oder vermieden werden können (vgl. Abbildung 3). Risiken R1 bis R3 könnten den Projekterfolg einschränken, aber aufgrund der hohen Expertise des Projektteams ist die Eintrittswahrscheinlichkeit äußerst gering. Risiken R5 bis R7 können durch

besondere Sorgfalt präventiv vermieden werden, während Risiko R4 durch korrektive Maßnahmen im schlimmsten Fall zu kurzfristigen Verzögerungen im Projektablauf führen kann.

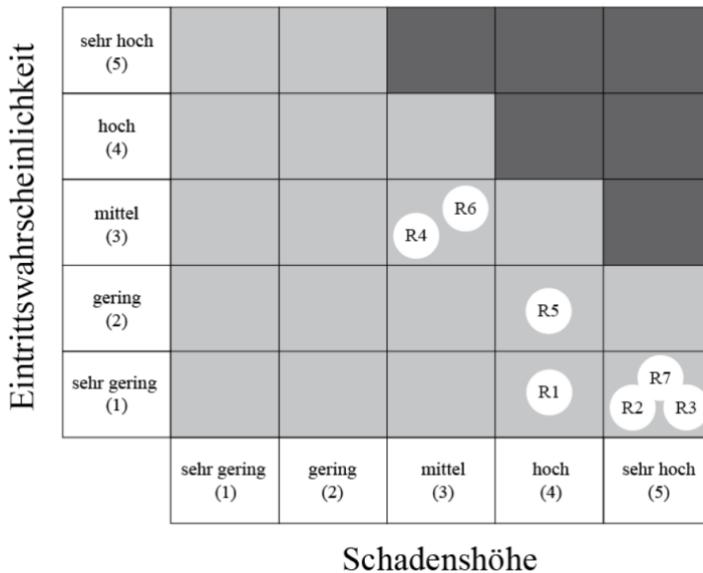


Abbildung 3: Risikomatrix

Die Umfeldanalyse des Projekts "Energiebasierte Terminplanung" ergab, dass keine nennenswerten negativen Faktoren dem Projekt im Weg stehen. Stattdessen besteht eine hohe Wahrscheinlichkeit, dass das Projekt von verschiedenen Bereichen positiv aufgenommen und möglicherweise sogar unterstützt wird.

Die dritte Komponente der Projektplanung befasst sich mit der Festlegung von Etappenzielen, um sicherzustellen, dass die Ambitionen des Projektteams und des Auftraggebers im zeitlichen Rahmen bleiben. Die Etappenziele sollten messbare Erfolge darstellen und zeigen, dass das Projekt zu vorab festgelegten Zeitpunkten fortschreitet. Im Rahmen der Lean Startup Methode werden Epics und User Stories formuliert, um den Anforderungskatalog zu erstellen und dessen Umsetzung nach zeitlicher Aufwandsschätzung zu planen. Weitere Informationen hierzu finden sich im Unterkapitel Lean Startup.

3.1.4 Teilbereich Outputfaktoren

Der Projektdurchführung folgen üblicherweise Output Faktoren, die im Project Canvas in den Bausteinen Kunde, Ergebnis und Qualität dargestellt werden.

Der Kunde ist in diesem Projekt die SEHO als Auftraggeber. Allerdings werden auch andere Unternehmen und sogar Privathaushalte, die ähnliche Ziele verfolgen, berücksichtigt. Diese Anwender bezeichnet man im Rahmen des Project Canvas als Empfänger des Projektergebnisses. Das Ergebnis des Projekts definiert, was am Ende an den Kunden geliefert

werden soll. Es wird üblicherweise in eine der drei Kategorien Produkt, Service oder neues Wissen eingeordnet. Ziel dieser Projektarbeit war die Erstellung einer Web-Applikation, mit deren Hilfe energieeffiziente Terminplanung durchgeführt werden kann – eine klare Einordnung in die Kategorie Produkt.

Die Qualität des Ergebnisses muss den Kundenwünschen entsprechen, weshalb regelmäßige Treffen und Besprechungen mit dem Betreuungsteam und dem Auftraggeber stattfinden. Damit soll erreicht werden, dass der Projektfortschritt abgestimmt wird und sichergestellt wird, dass das angestrebte Ergebnis erreicht wird. Die Protokolle aller Besprechungen der Projektteilnehmer sind im Anhang 2 – Protokolle zu finden.

3.1.5 Teilbereich Zeit

Der letzte Teil des Project Canvas beschäftigt sich mit der Organisation des Projekts. In diesem Teil werden der Start- und Endtermin, Vorbereitungsphasen sowie Toleranzgrenzen für mögliche Abweichungen von den geplanten Zeiten berücksichtigt. Das Projektteam setzte sich das Ziel, die funktionalen Anforderungen bis Ende Januar 2023 zu erfüllen. Der Februar 2023 wurde für die Ausgestaltung der Anwendung im Frontend eingeplant. Im März 2023 steht die Erstellung des Projektberichts im Vordergrund und bietet Toleranzzeit bei Abweichungen des Zeitplans. Das Projekt dauert damit insgesamt sechs Monate, begann am 08.09.2022 mit einem Pre-Kick-Off und endete am 31.03.2023 mit der Abgabe der Projektarbeit und des Projektberichts (vgl. Abbildung 4).



Abbildung 4: nicht-maßstabsgetreuer Phasen- und Meilensteinplan

3.2 Lean Startup

Lean Startup ist ein wissenschaftlich fundierter Ansatz, der von Gründer und Softwareentwickler Eric Ries entwickelt wurde, um schnell und kosteneffizient von einer Idee zur Unternehmensgründung zu gelangen. Dieser Ansatz basiert auf dem Build-Measure-Learn-Prinzip, welches sehr gut auf agile Projektarbeit übertragbar ist. Ziel ist es, schnell und effektiv

zu lernen, was Kunden wollen, um das Risiko eines Scheiterns des Projekts zu minimieren. Die Methode beinhaltet die Anwendung von Personas, User Stories und die Erstellung eines Minimal Viable Products (MVP), welche auch im vorliegenden Projekt Anwendung finden (Ries 2011).

3.2.1 Personas und User Stories

Personas sind ein Nutzermodell, bei welchem detaillierte Beschreibungen von hypothetischen typischen Endnutzern, die demographische Details, Erfahrungen, Interessen sowie Wünsche und Bedürfnisse in Bezug auf das Endprodukt zugeschrieben werden (Agile Business Consortium o. J.-b). Die Erstellung von Personas in Form von kurzen individuellen Geschichten eines potenziellen Nutzers in Verbindung mit dem zu erarbeitenden Projektziel hilft dem Projektteam, ein Verständnis für die Anforderungen aufzubauen und ein Gefühl für die Bedürfnisse der Nutzer zu entwickeln.

Im Projekt wurden fünf Nutzergruppen identifiziert – Geschäftsführung mit Leitung der Forschungs- und Entwicklungsabteilung, Service- und Vertriebsmitarbeiter, Kunden, externe Projektpartner der Axsol sowie externe Projektpartner der WUQM. Als Grundlage diente die im Vorfeld durchgeführte Stakeholder Analyse. Entsprechend der Empfehlungen der Lean Startup Methode wurden so viele bekannte reale Informationen verwendet, wie verfügbar waren. Es wurden ergänzend reflektierte Annahmen getroffen, um Informationslücken zu schließen, die die Vollständigkeit der Personas als Grundlage für die User Stories sicherstellen sollen. Die erstellten Personas sind im Anhang 3 – Personas zu finden.

Aus den erstellten Personas konnten User Stories erarbeitet werden. User Stories sind informelle Beschreibungen von Anforderungen an ein Produkt, die aus der Perspektive eines Nutzers formuliert sind. Sie beschreiben den Nutzen, den der Anwender durch das Produkt erhalten soll und dienen als Grundlage, um die tatsächlichen Anforderungen an das finale Produkt abzuleiten (Agile Business Consortium o. J.-b). Nach dem klassischen Schema werden Rolle, Ziel und Nutzen in einer User Story beschrieben:

Als <Rolle> möchte ich <Ziel/Wunsch> um <Nutzen>.

In Gesprächen mit dem Auftraggeber konnte der endgültige Nutzerkreis der Anwendung bestehend aus insgesamt vier Mitarbeitern und zwei Mitgliedern der Geschäftsleitung identifiziert werden. Darüber hinaus konnten drei konkrete Anwendungsfälle für den Einsatz des Tools herausgearbeitet werden – erstens, die Kundenanfrage für die Demonstration der Maschinen im Ausstellungsraum; zweitens, die Kundenanfrage für das Testen eines neuen Produkts; drittens, die Mitarbeiteranfrage zur Qualitätssicherung produzierter Maschinen vor der Auslieferung. So konnten 21 User Stories für unterschiedliche Rollen und Nutzergruppen erstellt werden. Alle erstellten User Stories können dem Anhang 4 – User Stories dieser Arbeit

entnommen werden. Welche Funktionalitäten im Detail abgeleitet wurden, ist im Abschnitt zum Inhaltskonzept nachzulesen.

3.2.2 MVP und Anforderungskatalog nach dem MoSCoW-Prinzip

Eines der wichtigsten Charakteristika eines Projekts ist der zeitlich fest vorgegebene Rahmen. Aufgrund dessen ist die Priorisierung der identifizierten Anforderungen entscheidend. Das MoSCoW-Prinzip ist ein Ansatz zur Priorisierung von Anforderungen in der Softwareentwicklung. Es wurde ursprünglich 1994 von Dai Clegg für das agile Projektmanagement-Framework Dynamic Systems Development Method (DSDM) erstellt (Agile Business Consortium o. J.-a). Die Abkürzung MoSCoW steht für die vier Kategorien von Anforderungen „Must Have“, „Should Have“, „Could Have“ oder „Won’t Have“, deren Verwendung Klarheit hinsichtlich der Erwartungen an das Projekt schafft:

- **Must Have** Die Mindestanforderungen, die zwingend zu erfüllen sind.
- **Should Have** Anforderungen, die erfüllt werden sollten, wenn es möglich ist, aber nicht zwingend erforderlich für den Projekterfolg sind.
- **Could Have** Anforderungen, die wünschenswert aber nicht zwingend erforderlich sind und deren Entfall geringere Auswirkungen auf den Projekterfolg hat als der Entfall von „Should Have“ Anforderungen. Sie werden nur im besten Fall vollständig ausgeliefert.
- **Won’t Have** Anforderungen, die im vorgegebenen Zeitrahmen nicht erfüllt werden, deren Nennung jedoch dazu beiträgt, die Erwartungshaltung externer Partner und den Fokus des Projektteams zu steuern.

Das Projektteam verwendete das MoSCoW-Prinzip zur Priorisierung der User Stories, um den Anforderungskatalog und das Minimal Viable Product (MVP) abzuleiten. Das MVP ist eine Basisversion des finalen Produkts, die als Prototyp oder Betaversion mit minimalem Aufwand entwickelt wird und nur die notwendigsten Funktionen enthält, um dem Auftraggeber einen Eindruck des Mehrwerts des finalen Produkts zu vermitteln. Das Ziel ist es, schnell Feedback zu erhalten und bei Bedarf Anpassungen vorzunehmen (Ries 2011). Die Definition des MVP im Projekt „Energiebasierte Terminplanung“ lautet wie folgt:

Eine lokal funktionierende Web-Anwendung, die Daten von einem Server abruft, diese mit Hilfe eines Optimierungsmodells unter Berücksichtigung der Nebenbedingungen Energiebedarf und voraussichtlich erzeugte PV-Energie weiterverarbeitet und für die optimale Terminvergabe Vorschläge generiert, um Kunden zu Demonstrationen in den Ausstellungsraum einzuladen.

Alle übrigen priorisierten Anforderungen wurden vom Projektteam in kleine Arbeitspakete zerlegt, im Jira Board eingepflegt und entsprechend ihrer Priorität abgearbeitet. Eine Aufstellung der priorisierten User Stories ist im Anhang 4 – User Stories nachzulesen.

3.3 Agiles Projektmanagement

Agiles Projektmanagement umfasst die neuesten Methoden, Prinzipien und Vorgehensmodelle des modernen Projektmanagements. Die agilen Modelle wurden entwickelt, um Projekte mit unsicheren Anforderungen strukturiert und flexibel angehen zu können.

3.3.1 Agile Vorgehensmodelle

Der Erfolg agiler Vorgehensmodelle beruht auf den Werten und Prinzipien des Agilen Manifests, welches auf der zwischenmenschlichen Zusammenarbeit basiert und ergebnisorientierte Zusammenarbeit höher gewichtet als formelle Prozesse. Die zwölf Prinzipien beschreiben eine effektive Teamarbeit, um dem Kunden in einem frühen Stadium des Projekts funktionale Software zur Verfügung zu stellen, welche dann schrittweise erweitert wird. Das Ziel ist es, ein Ergebnis zu erreichen, das möglichst den Bedürfnissen des Kunden entspricht während eine lösungsorientierte Herangehensweise im Vordergrund steht (Beck et al. 2001).

Das "Prinzip der kleinen Pyramide" von Philipp Meyerbröker (vgl. Abbildung 5) verbildlicht das agile Manifest, demzufolge nach jeder Iteration ein funktionsfähiges Ergebnis vorliegen sollte, und stellt es dem traditionellen Vorgehen gegenüber, bei dem zunächst das Fundament gelegt und erst am Ende ein funktionstüchtiges Ergebnis ausgeliefert werden kann (Meyerbröker 2011).

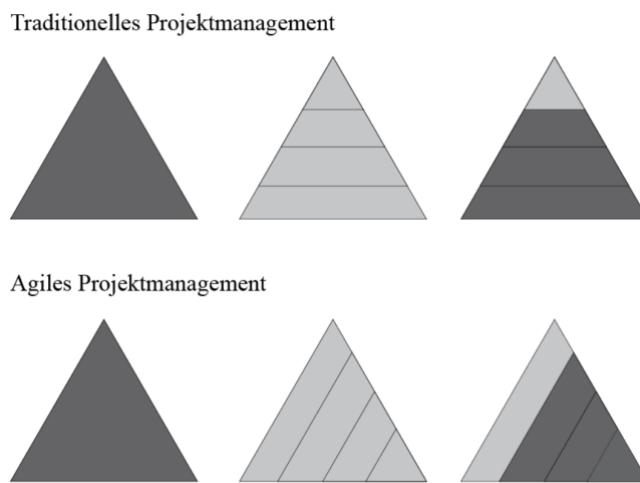


Abbildung 5: Prinzip der kleinen Pyramide (in Anlehnung Meyerbröker 2011)

Die Entscheidung für ein agiles Vorgehensmodell fiel aufgrund der Tatsache, dass das Ziel des Projekts nicht festgelegt, sondern nur eine Idee des endgültigen Produkts vorhanden war und somit viel Freiraum für Erweiterungen bestand. Agile Methoden eignen sich in diesem Fall, da sie nicht das Ziel, sondern die verfügbaren Ressourcen und einen festgelegten Termin in den Fokus rücken.

3.3.2 Scrum Framework

Das Projektteam hat sich für das Scrum Framework als agiles Vorgehensmodell entschieden, da es weit verbreitet ist und alle Teammitglieder damit vertraut sind. Der Scrum Guide dient als Referenz für die Anwendung von Scrum und wird als Grundlage für die Begriffsdefinitionen verwendet (Schwaber und Sutherland 2022). Scrum ist ein Rahmenwerk für die Entwicklung, Lieferung und Wartung komplexer Produkte, welches einen iterativen und inkrementellen Ansatz mit kurzen Feedback-Schleifen basiert. Es definiert Rollen innerhalb des Teams, Planungs- und Arbeitsergebnisse (Artefakte), Ereignisse (Events) und die Interaktion zwischen diesen Elementen. Es legt keine strikten Prozesse fest, sondern bietet lediglich einen Rahmen mit Regeln. Die Anwender von Scrum können die Arbeitsweise innerhalb dieses Rahmens selbst bestimmen, wobei Transparenz von entscheidender Bedeutung ist, da alle Entscheidungen auf dem wahrgenommenen Zustand der drei Artefakte basieren.

Scrum zeichnet sich durch seine regelmäßigen Arbeitszyklen, sogenannte Sprints, aus. Jeder Sprint kann als ein kleines, zeitlich begrenztes Projekt betrachtet werden, dessen Ziel es ist, ein funktionsfähiges, potenziell auslieferbares Zwischenprodukt zu entwickeln. Zu Beginn jedes Sprints wird ein definiertes Ziel festgelegt, das Teil des großen Projekts ist und innerhalb der vorgegebenen Zeit erreicht werden soll. Dazu wählt das Scrum-Team aus dem Product Backlog, dem Anforderungskatalog an das gesamte Projekt, die Aufgaben für den kommenden Sprint aus und erstellt daraus das Sprint Backlog. Der Scrum Master ist für die Einhaltung von Zielen und Überwachung des Teams verantwortlich, während der Product Owner die Interessen des Kunden gegenüber dem Entwicklungsteam vertritt. Die drei Scrum-Artefakte Product Backlog, Sprint Backlog und Increment basieren alle auf Transparenz, Überprüfung und Anpassung. Das Sprint Backlog, dient als Plan für das Team und gibt ein aktuelles Bild ihrer Arbeit am Increment, einer Erweiterung des funktionalen Produkts um weitere Funktionen.

Zu Überprüfung und Anpassung gibt es vier Events innerhalb eines Sprints: Sprint Planning, Daily Scrum, Sprint Review und Sprint Retrospective. Die Sprint Planung basiert auf dem Sprint Backlog und findet zu Beginn eines Sprints statt. Hier wird das Sprintziel definiert und die Sprint Backlog Items für den kommenden Sprint geplant. Die Sprintplanung erfolgte in der Regel nach den Jour Fixes mit dem Betreuerteam, sodass Feedback zu Zwischenständen berücksichtigt werden konnte. Der Daily Scrum ist eine tägliche Besprechung von 15 Minuten, bei der das Entwicklerteam über seine Fortschritte und Herausforderungen spricht und Pläne

für die nächsten Schritte entwirft. Da das Team auch anderen Verpflichtungen nachgeht, wurde das Daily Scrum in ein Weekly Scrum umgewandelt, mit einer durchschnittlichen Dauer von einer Stunde, anstatt 15 Minuten. Nach Abschluss eines Sprints findet das Sprint Review statt, bei dem die Ergebnisse des Sprints präsentiert und mit dem Product Backlog verglichen werden, um künftige Anpassungen zu bestimmen. Das Sprint Review wurde in den regelmäßigen Jour Fixes mit dem Betreuerteam abgehalten, um Feedback zum aktuellen Stand und weiteren Schritten zu erhalten. Nach dem Sprint Review folgt die Sprint Retrospektive, in der das Entwicklungsteam den Sprintzyklus abschließt und einen neuen beginnt. Hier werden Probleme und Möglichkeiten für Verbesserungen besprochen, um den nächsten Sprint zu verbessern. Die Besprechungen fanden normalerweise unmittelbar nach den regulären Jour Fixes mit dem Betreuerteam statt.

Der kurze Zeitraum eines Sprints hat den Vorteil, dass die Komplexität des Sprints gering gehalten wird und somit dem Risiko von Fehlentwicklungen entgegengewirkt wird. Ähnlich wie beim Daily Scrum wurde auch die Sprintdauer vom Projektteam an die eigenen Bedürfnisse angepasst, sodass externe universitäre, berufliche und private Verpflichtungen weiterhin eingehalten werden konnten. Das Projektteam entschied sich dementsprechend für eine Sprintdauer von zwei Wochen.

3.3.3 Sprintablauf

Während der Entwicklungsphasen des Projekts, also dem Zeitraum, in dem aktiv an der Erstellung des Endprodukts gearbeitet wurde, fanden insgesamt sieben operative Sprints von jeweils 14 Tagen sowie ein organisatorischer Sprint von sieben Tagen statt. Entsprechend den Regeln des Scrum Guide startete jeder Sprint mit der Sprint Planung im Anschluss an die zweiwöchentlichen Jour Fixes mit dem Betreuerteam, und endete mit der Präsentation und Demonstration des aktuellen Stands in der Sprint Review im Jour Fixe mit dem Betreuerteam. Abbildung 6 gibt einen Überblick über den zeitlichen Verlauf der jeweiligen Sprints und der darin erarbeiteten Sprintziele. Eine detaillierte Dokumentation jedes Sprints unterstützt durch Screenshots zu den jeweiligen Zwischenständen ist im Anhang 5 – Detaillierte Dokumentation Sprintablauf zu finden.



Abbildung 6: Sprintablauf und Sprintziele

3.4 Tooleinsatz

Das Projektteam hat vor Beginn des Projekts eine Auswahl an Softwareprodukten getroffen und diese Auswahl später um weitere Werkzeuge ergänzt. Dabei wurden insbesondere Tools für Projektmanagement, Organisation und Kommunikation sowie Programmiersprachen für die Anwendungsentwicklung ausgewählt. Während des Projekts wurden nach Bedarf weitere Tools hinzugefügt. Bei der Auswahl wurde darauf geachtet, dass alle Teammitglieder die Software einsetzen können. Im Folgenden werden alle verwendeten Werkzeuge kurz vorgestellt.

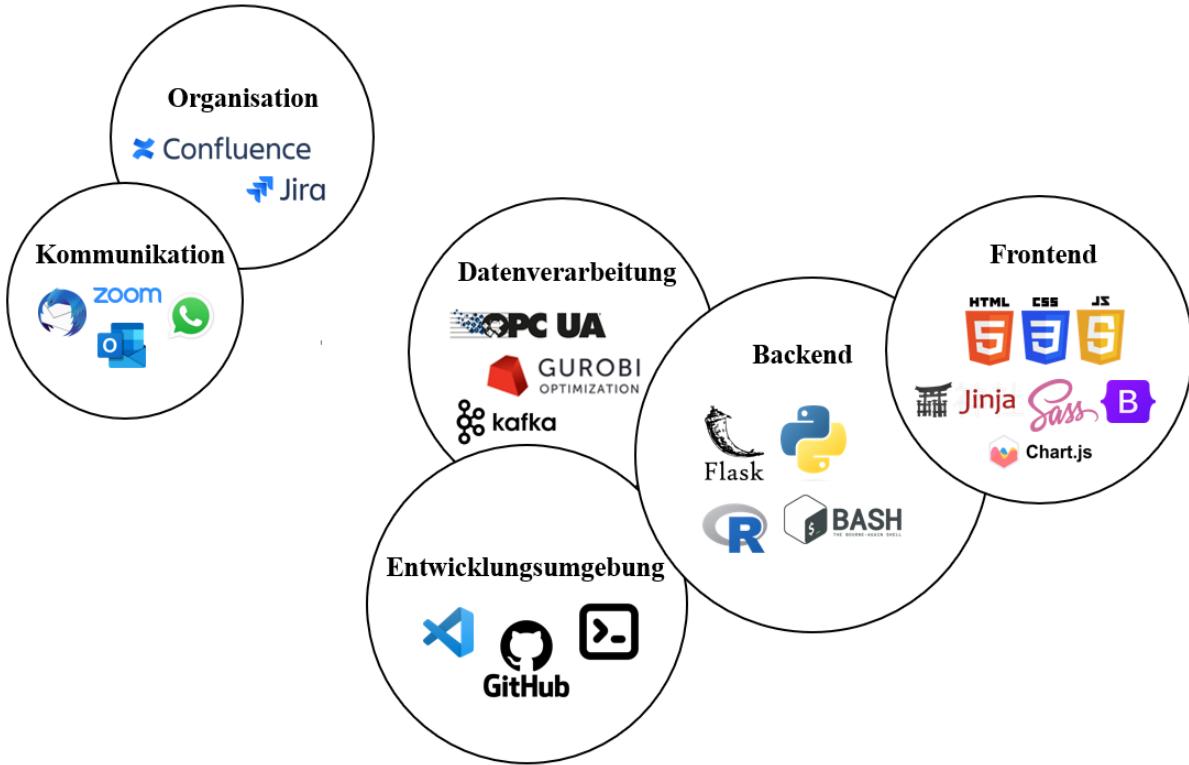


Abbildung 7: Technologie-Stack des Projekts „Energiebasierte Terminplanung“

3.4.1 Kommunikation

Das Projektteam nutzte verschiedene Tools wie Zoom, WhatsApp und E-Mail-Programme, um eine erfolgreiche Kommunikation zwischen den Teammitgliedern zu ermöglichen. Diese Tools ermöglichen eine schnelle und effektive Kommunikation unabhängig vom Standort der Teammitglieder. Zoom wurde für Meetings mit Projektpartnern und Fortschrittsbesprechungen mit dem Betreuerteam genutzt. WhatsApp wurde für schnelle Abstimmungen zwischen den Teammitgliedern eingesetzt, während E-Mail-Programme zum Austausch von Dokumenten und wichtigen Informationen dienten. Diese Tools führte zu höherer Effizienz und Produktivität.

3.4.2 Organisation

Für Organisation und Projektmanagement wurden Atlassian-Produkte verwendet, insbesondere Confluence und Jira. Confluence diente als zentrale Plattform für die Verwaltung und den Austausch von Informationen, während Jira für das agile Projektmanagement genutzt wurde, um Aufgaben und Arbeitspakete im Product bzw. Sprint Backlog zu verwalten. Obwohl Jira für das Team zunächst unbekannt war, wurde es schnell zu einem wichtigen Werkzeug im Projektmanagement-Prozess (Atlassian. Pty Ltd 2023). Allerdings entdeckten die Teammitglieder die Möglichkeit zur Priorisierung von Arbeitspaketen innerhalb des Tools erst

spät, was die Effizienz des Projektmanagements einschränkte, jedoch als wichtige Erkenntnis auch für spätere Projekte aufgenommen wurde.

3.4.3 Entwicklungsumgebung

Das Projektteam nutzte verschiedene Plattformen und Interfaces zur Erstellung und Verwaltung des Programmcodes. GitHub, eine webbasierte Hosting-Plattform für Softwareentwicklung, diente als zentraler Ablageort für den Code und ermöglichte die gemeinsame Arbeit mehrerer Entwickler an einem Projekt. Das Team hat volle Kontrolle über die Repositories durch Versionierung, Versionskontrolle und Issue-Tracking, was die Qualität der Software verbessert, und die Zusammenarbeit erleichtert (GitHub Inc. 2023b).

Visual Studio Code (VSCode) wurde als plattformübergreifende Code-Editor-Software zur Erstellung der Webanwendung genutzt. Dank des großen Funktionsumfangs konnte die Webanwendung mit allen notwendigen Programmiersprachen innerhalb eines Tools erstellt werden (Microsoft Corp. 2023).

Das Command Line Interface (CLI), eine textbasierte Schnittstelle, über die Benutzer mit einem Computer oder einem Programm interagieren können, diente als Secure Shell (SSH) und Secure File Transfer Protocol (SFTP) Client für die Sicherstellung der Netzwerksicherheit und Kontrolle bei der Verwaltung der Server und des Dateisystems.

3.4.4 Programmierung Backend

Das Backend der Anwendung wurde unter Verwendung verschiedener Programmiersprachen, Frameworks und Bibliotheken erstellt. Python war die am häufigsten verwendete Sprache und wurde aufgrund des vorhandenen Wissens des Teams eingesetzt. Python bietet eine Vielzahl von Bibliotheken und Frameworks, die bei der Programmierung von Anwendungen unterstützen, darunter Flask, ein leichtgewichtiges Webframework, das die Integration von Backend- und Frontend-Komponenten ermöglicht (Pallets 2010).

R, eine Sprache zur statistischen Berechnung und Datenanalyse, wurde zur Erstellung synthetischer Daten verwendet, die insbesondere für die Erstellung des MVPs essenziell waren. Bash, eine Shell-Skriptsprache, die in Unix-ähnlichen Betriebssystemen wie Linux verwendet wird, wurde für die Systemadministration und die Konfiguration von Servern verwendet (Loshin 2022).

3.4.5 Programmierung Frontend

Das Projektteam hat für die Frontendentwicklung einen Technologie-Stack bestehend aus HTML, SCSS, JavaScript, Bootstrap, ChartJS und Jinja2 verwendet. HTML dient als

Grundstruktur, welche die Inhalte und Semantik festlegt, SCSS definiert das Erscheinungsbild und JavaScript sorgt für Interaktivität und Funktionalität. Die meisten Funktionen der Anwendung wurden mit Flask ins Backend überführt. Die Datenvisualisierung wurde im Frontend mit der JavaScript Bibliothek ChartJS umgesetzt. Bootstrap wurde verwendet, um vorgefertigte HTML-, CSS- und JavaScript-Komponenten und -Layouts für responsive Webanwendungen zu nutzen. Den Abschluss des Frontend-Technologie-Stacks bildet Jinja2, eine Template-Engine, die bei der Erstellung von dynamischen Frontends hilft, indem sie eine Vorlage (Template) mit Daten aus dem Backend befüllt und daraus eine HTML-Seite als Frontend generiert. Da es Python-basiert ist, ist ein einfacher und flexibler Einsatz möglich. Es ist eine der am häufigsten verwendeten Template-Engines für Python-Webanwendungen und wird von Frameworks wie Flask unterstützt. Jinja unterstützt auch weitreichende Funktionen wie Vererbung, Filterung und Blöcke, die bei der Erstellung komplexer Templates notwendig sind (Pallets 2007).

3.4.6 Datenverarbeitung

Im Bereich des Datenaustauschs, der Datenhaltung und der Datenverarbeitung fanden drei verschiedene Technologien Anwendung: Der IoT-Kommunikationsstandard OPC UA, der Gurobi Optimizer und ein Kafka-Streaming-Server.

Der OPC UA-Kommunikationsstandard für Industrie 4.0 und IoT ermöglicht den herstellerunabhängigen Datenaustausch zwischen verschiedenen Maschinen, Geräten und Systemen. Für das Projekt bedeutet das, dass die Daten der Photovoltaikanlage (z.B. Leistung, aktuelle Energieerzeugung, etc.) in Echtzeit abgerufen und an den Server übertragen werden, von wo aus sie von der Optimierungsfunktion aufgerufen und weiterverarbeitet werden können. Zudem ist die automatisierte Steuerung der Lötmaschinen möglich (Inray Industriesoftware GmbH 2023).

Der Gurobi Optimizer ist eine Software, die mathematische Modelle und Algorithmen verwendet, um optimale Lösungen für ein gegebenes Problem zu berechnen. Im Projekt wird Gurobi verwendet, um optimale Zeiten für Terminpläne basierend auf Solarenergiedaten zu bestimmen (Gurobi Optimization LLC o. J.).

Der Kafka-Streaming-Server fungiert als Datenpipeline, welche Daten aus verschiedenen Datenquellen empfängt, gruppiert und bei Bedarf anderen Anwendungen zur Verfügung stellt. Das Team nutzt den Kafka-Server, um die Wetterdaten von OpenWeatherMap einmal täglich abzurufen und über den Server an die Anwendung zu übergeben (Apache Software Foundation 2023).

3.5 Reflexion des Projektmanagements

Rückblickend lässt sich sagen, dass die intensive Vorbereitung des Projekts von entscheidender Bedeutung für seinen erfolgreichen Abschluss war. Durch ein effektives Zeitmanagement auf Basis des Phasen-Meilenstein-Plans und den danach strukturierten Sprints sowie die Verwendung von Lean-Startup-Methoden wie User Stories und Personas konnte das Team die Anforderungen potenzieller Anwender präzise definieren und die Anwendung entsprechend konzipieren. Die Wahl des agilen Vorgehensmodells SCRUM ermöglichte dem Projektteam, schnell auf Feedback des Auftraggebers zu reagieren und Anpassungen vorzunehmen. Eine offene Kommunikation mit dem Auftraggeber, regelmäßige Meetings mit Projektbetreuern und Feedback-Loops mit allen Projektpartnern halfen dem Team, auf Kurs zu bleiben und wertvolle Erkenntnisse aus den Erfahrungen der Projektteilnehmer zu ziehen. Die intensive vorherige Auseinandersetzung mit Tools, Frameworks und Richtlinien war ein wichtiger Faktor für den Erfolg des Projekts, da dies bei der Auswahl geeigneter Werkzeuge half und den direkten Einstieg in die Umsetzung der Anforderungen erleichterte.

4 Frontend

In diesem Kapitel wird die Erstellung des Inhaltskonzepts und des Designkonzepts für SolarSync erläutert. Zudem wird auf die Modellierung der bisherigen Ist-Prozesse der Terminplanung bei der SEHO sowie der künftigen Soll-Prozesse mit SolarSync eingegangen. Dabei werden die wichtigsten Aspekte der Planung und Umsetzung dieser Schritte im Entwicklungsprozess einer effektiven und benutzerfreundlichen Anwendung erläutert.

4.1 Inhaltskonzept

Das Inhaltskonzept einer Webanwendung definiert, welche Informationen und Daten dem Nutzer zur Verfügung gestellt werden und wie diese visuell aufbereitet werden sollen. Es stellt den Zweck und die Beziehung der Inhalte untereinander dar und unterstützt die thematische Strukturierung. Für die Anwendung SolarSync orientierte sich das Projektteam an Empfehlungen des Content Marketing Institute (CMI) und der Nielsen Norman Group (NNG) (Harris 2016; Nielsen und Caya o. J.). Dabei wurde zunächst die Zielgruppe und der Zweck aus den erstellten Personas aus der Projektvorbereitung abgeleitet. Inhalte und Funktionen wurden auf Basis des Anforderungskatalogs festgelegt und die Struktur und Navigation des Datenraums definiert. Das Ergebnis wurde in einem Wireframe festgehalten (vgl. Abbildung 8) und wird im Anschluss näher erläutert.

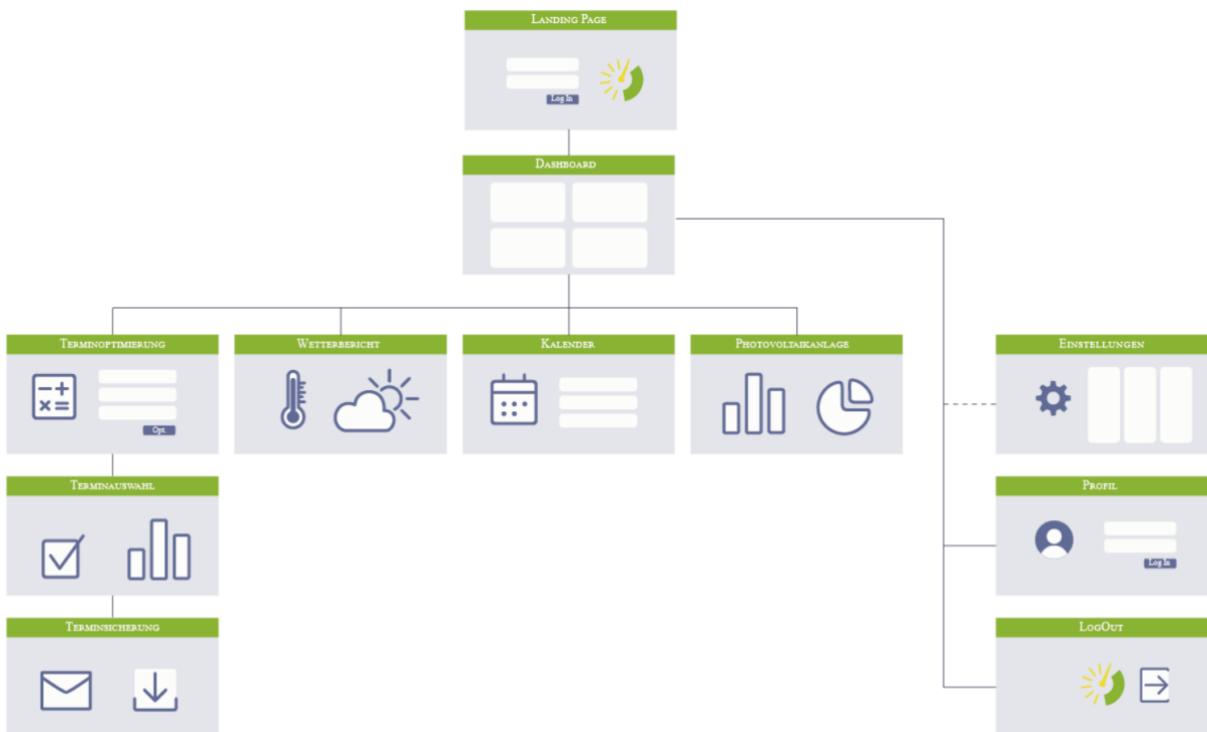


Abbildung 8: Wireframe-Visualisierung Inhaltskonzept

4.1.1 Landing Page

Die Landing Page dient als übergeordnete Instanz zur Authentifizierung zugriffsberechtigter Nutzer. Bei nicht autorisiertem Zugriff auf das Dashboard oder auf eine Unterseite leitet die Software hier hin zurück und fordert zum Login mittels Nutzernamen und Passwort auf.

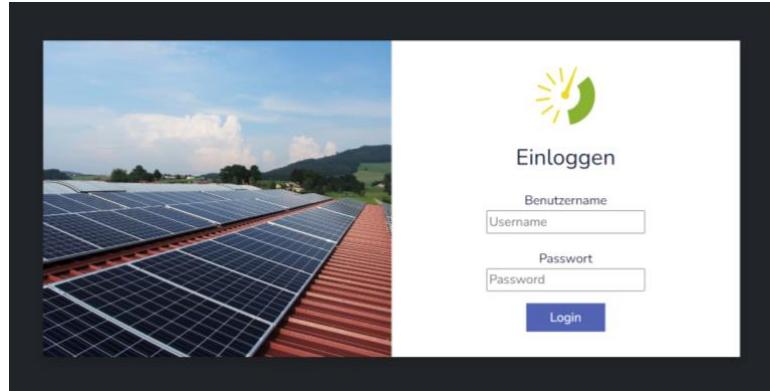


Abbildung 9: Finales Release (Landing Page)

4.1.2 Dashboard

Der Anwender wird nach erfolgreicher Authentifizierung auf das Dashboard geleitet, welches gemäß Ben Shneidermans Regeln für Interface Design gestaltet ist. Damit folgt es dem Mantra „Overview first, zoom and filter, then details on demand“ (Shneiderman et al. 2016). Das Dashboard bietet dem Nutzer eine Übersicht über alle relevanten Informationen und KPIs der Anwendung (vgl. Abbildung 10).

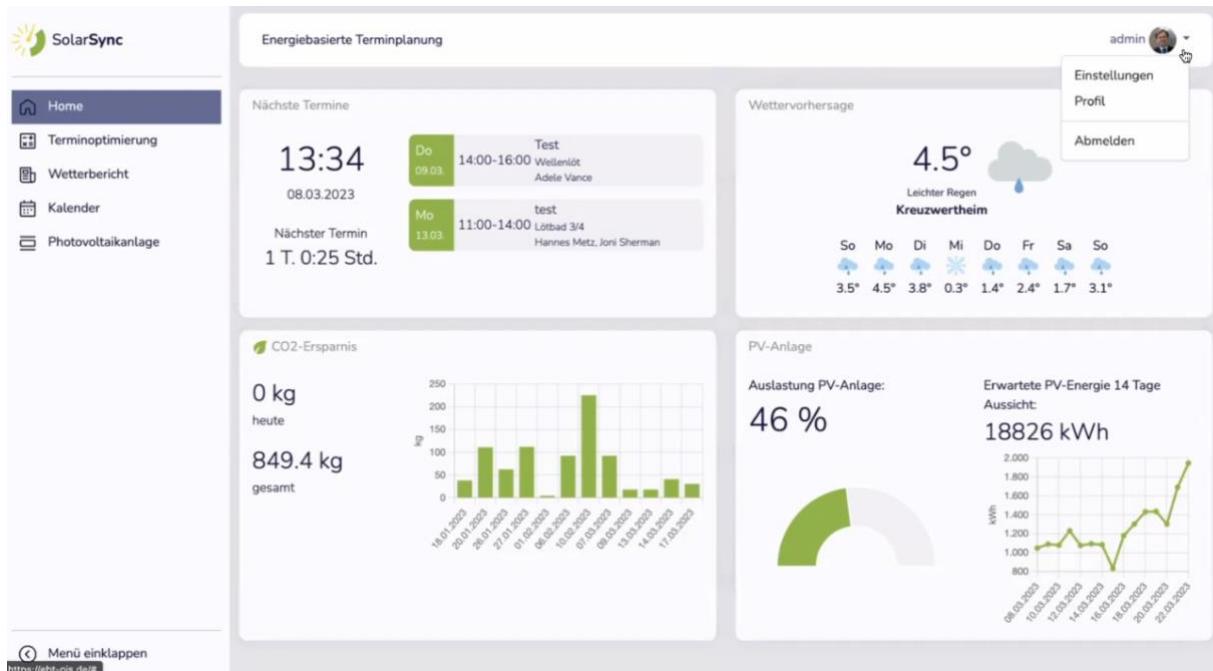


Abbildung 10: Release 6 (Dashboard)

Die Benutzeroberfläche der SolarSync Anwendung besteht aus vier Elementen. Das Panel über geplante Termine zeigt die aktuelle Uhrzeit, das Datum, die nächsten zwei Termine und einen Countdown bis zum nächsten Termin mit SolarSync. Es ist oben links platziert, um seine Bedeutung zu betonen und in Leserichtung an erster Stelle zu stehen. Das Wettervorhersage-Panel zeigt die Wettervorhersage, die für die Terminplanung wichtig ist. Es steht an zweiter Stelle in Leserichtung. Ein weiteres Panel informiert den Nutzer über die CO₂-Ersparnis, die durch den Einsatz von SolarSync erzielt werden kann. Dies unterstreicht den hohen Stellenwert der Nachhaltigkeit, der durch die Anwendung erreicht werden kann. Das letzte Panel zeigt Informationen zur Leistung der PV-Anlage und zur erwarteten Energieerzeugung der nächsten 14 Tage. Es gibt dem Nutzer einen Überblick über die Analysefunktionen der Anwendung und ermöglicht es ihm, die Effizienz von SolarSync jederzeit nachzuvollziehen.

Die enthaltenen Daten wurden bewusst ausgewählt, um eine hohe Transparenz für den Nutzer zu gewährleisten und ihm die Möglichkeit zu geben, die Optimierungsfunktion nachzuvollziehen. Vom zentralen Dashboard aus kann der Benutzer detaillierte Informationen über die einzelnen Termine abrufen, indem er sogenannte "Drill-down"-Aktivitäten ausführt.

Ein Dashboard kann auch interaktiv sein, indem es dem Nutzer erlaubt, Daten zu filtern, zu sortieren oder zu manipulieren, um verschiedene Perspektiven und Einblicke in die Daten zu erhalten. Interaktivität birgt jedoch auch die Gefahr der Überforderung des Nutzers. Insbesondere, wenn dieser nur über geringe Kenntnisse in Bezug auf BI-Anwendungen verfügt (O'Donnell und Zimmer 2020). Das Projektteam hat sich bewusst gegen umfangreiche Interaktionsmöglichkeiten entschieden, da aus Gesprächen mit dem Auftraggeber klar wurde, dass das Unternehmen als Ganzes tendenziell ein Late Mover im Bereich Digitalisierung ist. Die Anwendung kann jedoch jederzeit in einer späteren Iteration um derartige Funktionen erweitert und komplexer gestaltet werden.

4.1.3 Terminplanung

Die Unterseite zur Terminplanung ist der funktionale Teil von SolarSync und das Herzstück der Anwendung. Auf der ersten Ebene erstellt der Nutzer Terminvorschläge, indem er ein Formular ausfüllt. Dieses enthält alle notwendigen Informationen zur Generierung der Vorschläge (vgl. Abbildung 11). Der Nutzer wird darüber informiert, dass die Terminplanung aufgrund verfügbarer Wetterdaten nur für die nächsten 30 Tage möglich ist. Bei der Erstellung wurden besonders die Richtlinien der NNG zur sprachlichen Klarheit und Eindeutigkeit berücksichtigt. Ein deutlich gekennzeichneter Button startet die Optimierung und leitet den Nutzer auf die zweite Ebene der Unterseite.

Energiebasierte Terminoptimierung

Abbildung 11: Release 6 (Terminabfrage)

Die zweite Ebene zeigt die generierten Termine und ermöglicht es, einen Termin auszuwählen, zu bestätigen, zu speichern und auf Wunsch per E-Mail zu versenden (vgl. Abbildung 12). Entsprechende Buttons und Dialogfelder leiten Schritt für Schritt durch den Prozess (vgl. Abbildung 13 und Abbildung 14). Das System soll dem Nutzer Klarheit und Transparenz bieten und die Nachvollziehbarkeit der Optimierung erleichtern. Daher wurden Übersichtsgrafiken ergänzt, welche die Energieeinsparung durch die Terminvorschläge visualisieren und bei der Auswahl unterstützen.

Abbildung 12: Release 6 (Terminausbgabe)

Energiebasierte Terminoptimierung

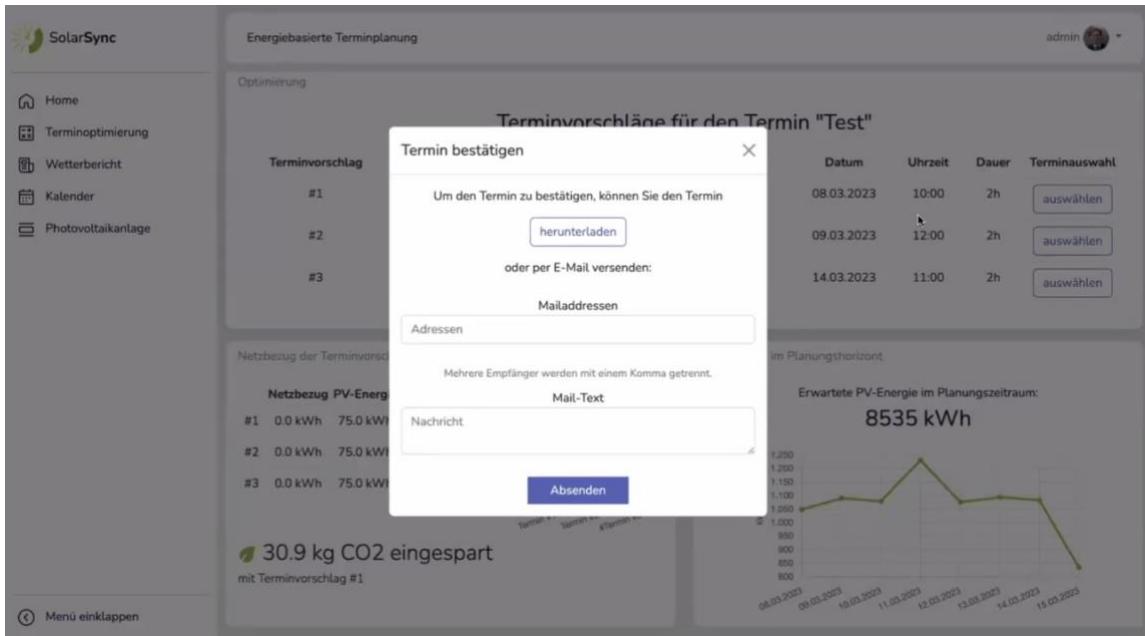


Abbildung 13: Release 6 (Terminbestätigung)

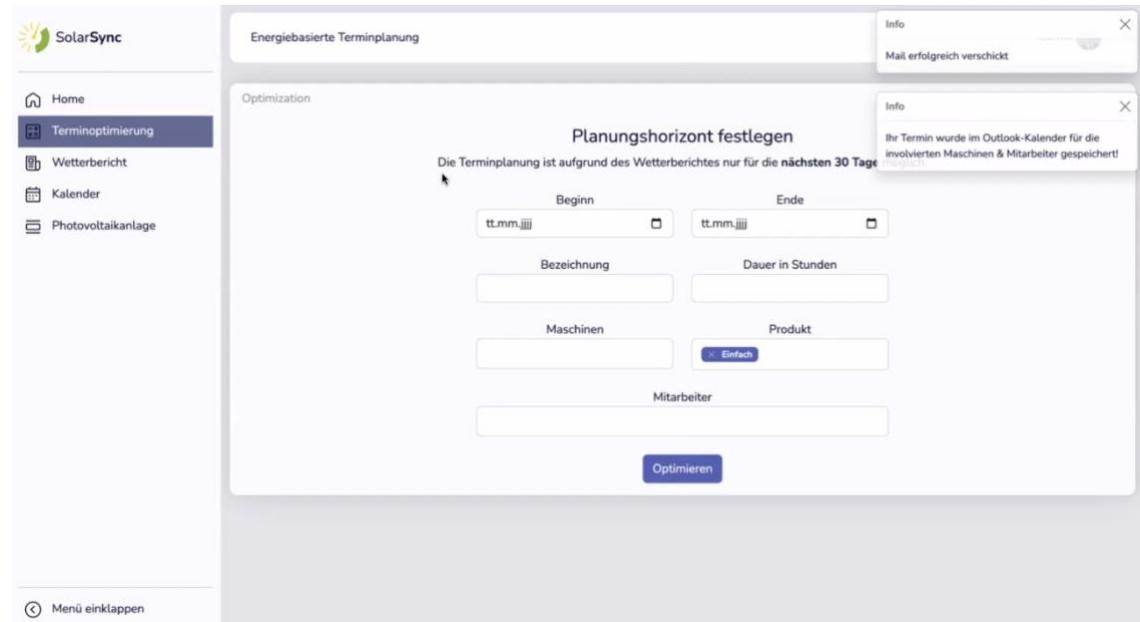


Abbildung 14: Release 6 (Terminversand)

4.1.4 Wetterbericht

Die Wetterbericht-Unterseite bietet eine Übersicht über aktuelle und zukünftige Wetterdaten (vgl. Abbildung 15). Die Informationen werden in einer Grafik und Tabelle präsentiert, die es dem Nutzer ermöglichen, die potenzielle PV-Energie der nächsten 30 Tage abzuleiten. Durch diese Funktion wird die Transparenz der Anwendung erhöht und die Informationen auf dem Dashboard erweitert.

Energiebasierte Terminoptimierung

The screenshot shows the SolarSync software interface. On the left, a sidebar menu includes Home, Terminoptimierung, Wetterbericht (selected), Kalender, and Photovoltaikanlage. The main area displays a weather forecast for Kreuzwertheim on 04.03.2023, showing a temperature of 4.5°C, a cloud icon with a raindrop, and the text "Leichter Regen". Below this, a weekly forecast for Sunday to Saturday shows temperatures ranging from 0.3°C to 3.8°C. The "Wetterbericht" section provides a detailed table of weather conditions for four days in March 2023.

Tag	Temperatur	Wetter	Sonne	Wolken	Druck	Airfeuchte
04.03.2023	0.1 - 4.5°C	Leichter Regen	05:59 - 17:07	77 %	1024 hPa	82 %
05.03.2023	0.7 - 3.5°C	Regen und Schnee	05:57 - 17:09	95 %	1020 hPa	77 %
06.03.2023	0.4 - 4.5°C	Regen und Schnee	05:55 - 17:11	84 %	1011 hPa	83 %
07.03.2023	0.3 - 3.8°C	Regen und Schnee	05:53 - 17:12	92 %	1001 hPa	91 %

Abbildung 15: Release 6 (Wetterausblick)

4.1.5 Kalender

Die Unterseite zum Kalender ist ähnlich zum Wetterbericht von informativem Charakter. Sie bietet eine Übersicht über alle Kundentermine, die mit SolarSync geplant wurden (vgl. Abbildung 16). Dies erweitert die Informationen auf dem Dashboard über die nächsten zwei Termine hinaus.

The screenshot shows the SolarSync software interface. The sidebar menu includes Home, Terminoptimierung, Wetterbericht, Kalender (selected), and Photovoltaikanlage. The main area displays a "Terminübersicht" (Schedule Overview) for the week from March 8 to 17, 2023. The schedule lists five entries: a test session on Wednesday, March 8, from 10:00 to 12:00; another test session on Thursday, March 9, from 14:00 to 16:00; a test session on Monday, March 13, from 11:00 to 14:00; a test session on Tuesday, March 14, from 13:00 to 15:00; and a test session on Friday, March 17, from 10:00 to 12:00. Each entry includes details such as machine names and staff members involved.

Abbildung 16: Release 6 (Terminübersicht)

4.1.6 Photovoltaikanlage

Auf der Unterseite zur Photovoltaikanlage wird dem Nutzer Einblick in die Leistung der Anlage gewährt (vgl. Abbildung 17). Die Seite hat informativen Charakter, jedoch werden hier keine Hintergründe zur Optimierungsfunktion gegeben. Stattdessen sind Leistungsdaten der Anlage integriert, sodass der Nutzer die Anwendung nicht verlassen muss, um sich darüber zu informieren. Daten über die erwartete PV-Energie und die Auslastung der Anlage dienen der generellen Abschätzung, wie effizient das System gegenwärtig läuft. Die 14-Tage-Aussicht der Erzeuger und Verbraucher erlaubt erste Informationen zum Autarkiegrad der PV-Anlage durch die Nutzung von SolarSync. Der Autarkiegrad wird zudem kumuliert angezeigt, um auch rückblickende Werte miteinzuschließen. In Kombination dieser Daten können Ableitungen getroffen werden, wie sich durch den Einsatz von SolarSync die Nutzung der PV-Anlage verbessert. Abschließend wird ein Wochenüberblick über die eingesparten CO₂-Werte gegeben, welcher die Informationen auf dem Dashboard ergänzt, da dort nur die CO₂-Einsparungen je Termin angezeigt werden.



Abbildung 17: Finales Release (Auswertung Energiedaten PV-Anlage)

4.1.7 Einstellungen und Profil

Die Unterseiten Einstellungen und Profil sind über das Dropdown Element in der oberen rechten Ecke zu erreichen. Auf diese Inhalte kann von jedem Punkt der Anwendung aus zugegriffen werden. Die Profilseite dient als Self-Service Bereich, in welchem der Nutzer sein Profilbild, Nutzernamen und Passwort selbstständig anpassen kann (vgl. Abbildung 18).

Energiebasierte Terminoptimierung

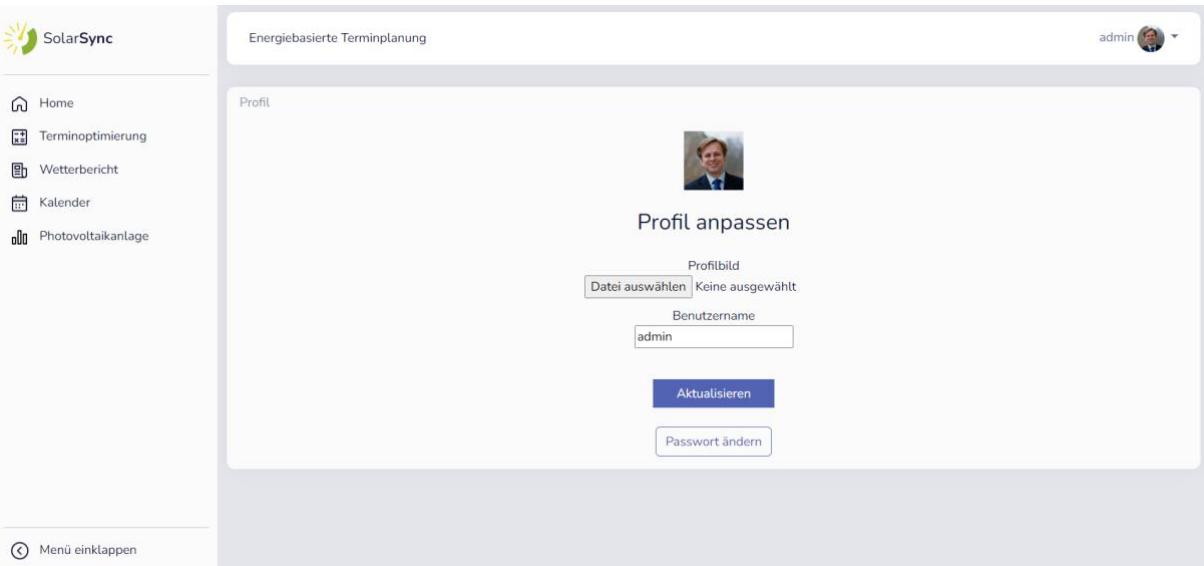


Abbildung 18: Finales Release (Profilseite)

Die Seite zu den Systemeinstellungen ist nur für berechtigte Nutzer zugänglich. Alle Elemente zur Administration der Anwendung wurden in diesem Bereich untergebracht (vgl. Abbildung 19). Er umfasst die Benutzerverwaltung sowie die modulare Anbindung aller Datenströme, welche die Anwendung und insbesondere das Optimierungsmodell benötigen. Alle Paneele wurden den Richtlinien der NNG mit sprechenden Titeln und Feldern konzipiert, sodass der Nutzer jederzeit darüber im Klaren ist, welche Informationen im jeweiligen Panel abgerufen bzw. hinterlegt werden (Nielsen und Caya o. J.).

The screenshot shows the SolarSync application's administration settings page. The left sidebar includes "Home", "Terminoptimierung", "Wetterbericht", "Kalender", and "Photovoltaikanlage", with "Menü einklappen" at the bottom. The main area has several panels: "Benutzerverwaltung" (User Management) showing a list of users with roles like Admin and User; "Wetter und Standort" (Weather and Location) with fields for API-Key, Latitude (49.7672), Longitude (9.5183), and an "Aktualisieren" button; "Verbräuche der Anlagen & Grundlast" (Consumption of installations & base load) with two input fields both set to 45; and "Mail-Server" (Mail Server) with fields for Mail-Server (smtp.ionos.de) and Mail-Port (587).

Abbildung 19: Finales Release (Administrationsbereich für Einstellungen)

4.2 Prozessmodellierung zur Terminerstellung

Die Prozessmodellierung der Ist- und Soll-Prozesse ist ein wichtiger Schritt bei der Einführung einer neuen Software in einem Unternehmen. Die Modellierung der Ist-Prozesse unterstützt das Projektteam dabei die bestehenden Geschäftsprozesse des Unternehmens nachzuvollziehen und zu dokumentieren. Eine übersichtliche Darstellung in einer allgemein anerkannten Notation erlaubt die erfassten Prozesse zu analysieren, Schwachstellen zu identifizieren und daraus Prozessoptimierungen abzuleiten. Auf der Basis der Ist-Prozesse werden dementsprechend die Soll-Prozesse definiert. Ihre Modellierung kommuniziert, inwiefern die bestehenden Prozesse mit Hilfe der neuen Software standardisiert, automatisiert oder optimiert werden und dadurch zu einer insgesamt gesteigerten Prozessqualität führen. Dieses Vorgehen wurde auch im Projekt „Energiebasierte Terminplanung“ angewendet und dessen Ergebnisse im Folgenden dargelegt.

Die geplante Software wird für die Terminplanung in Abhängigkeit der Energieproduktion der unternehmenseigenen PV-Anlage erstellt und beeinflusst keine weiteren Prozesse des Unternehmens. Folglich wurde auch nur gezielt dieser Prozess näher betrachtet. Der Ist-Prozess der SEHO umfasst die telefonische Terminvereinbarung mit einem Kunden zur Demonstration der Maschinen im Ausstellungsraum und die manuelle Pflege des Terminkalenders mit Vermerken zu beteiligten Mitarbeitern und Maschinen (vgl. Abbildung 20).

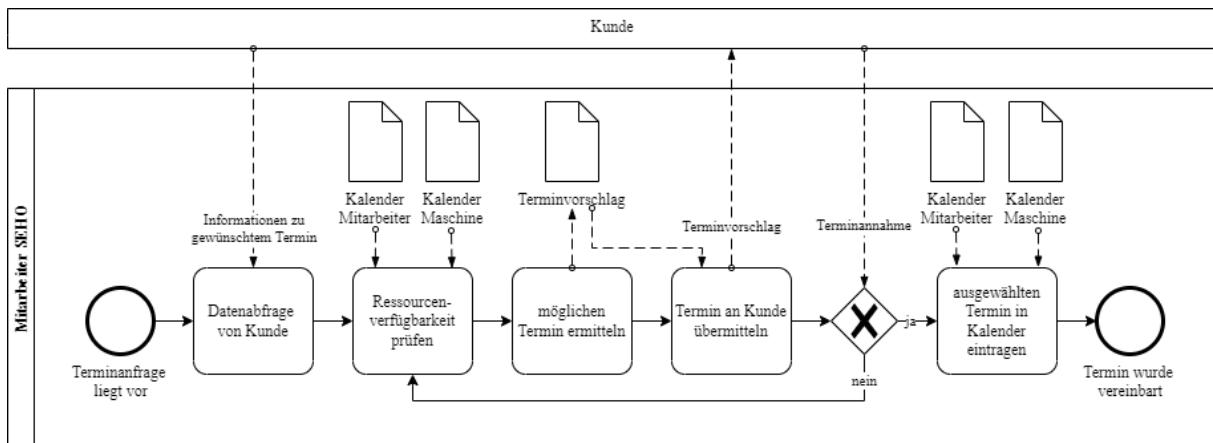


Abbildung 20: Visualisierung des Ist-Prozesses

Das auslösende Event für den Prozess ist der Terminwunsch eines Kunden zur Begutachtung der angebotenen Produkte der SEHO. Der zuständige Ansprechpartner des Kunden fragt die Daten des Kunden ab und nimmt dabei dessen Anforderungen an den Termin auf. Anhand dieser Anforderung kann der Mitarbeiter die benötigten Ressourcen Mitarbeiter, Maschinen und Zeit abschätzen. Auf Basis dieser Rahmenbedingungen prüft der Mitarbeiter manuell die benötigten Ressourcen auf ihre Verfügbarkeit und ermittelt damit einen möglichen Termin. Bei der Ermittlung eines Termins können Kundenpräferenzen für einen bestimmten Zeitraum berücksichtigt werden. Darüber hinaus ist der Mitarbeiter verantwortlich, dass es zu keinen Terminüberschneidungen kommt. Der Termin wird an den Kunden übermittelt, den dieser in

Abstimmung mit seinen eigenen Kapazitäten annehmen oder ablehnen kann. Wird der vorgeschlagene Termin durch den Kunden abgelehnt, entsteht eine Prozessschleife, die so lange durchlaufen wird, bis ein geeigneter Termin identifiziert und vom Kunden angenommen wurde. Der vereinbarte Termin wird anschließend manuell in die Kalender der betroffenen Ressourcen Mitarbeiter und Maschinen eingetragen. Die eingeplanten Maschinen unterliegen einer Farbcodierung, die allen Mitarbeitern die Übersicht über den Einsatz der Maschinen erleichtern soll. Die Pflege der Kalender beendet den Prozess der Terminplanung. Der gesamte Prozess findet in der Regel telefonisch oder schriftlich via E-Mail statt.

Die Analyse des Ist-Prozesses ergab, dass die bisherige Terminplanung der SEHO sehr zeitaufwendig und aufgrund einer Vielzahl an Medienbrüchen fehleranfällig ist. Wir sind überzeugt, dass die Mitarbeiter der SEHO sehr sorgfältig bei der Terminplanung mit ihren Kunden vorgeht. Dennoch sehen wir großes Potenzial der Prozessoptimierung durch die Digitalisierung und Teilautomatisierung des gegenwärtigen Prozesses. Darüber hinaus ist die Berücksichtigung eines energieeffizienten Einsatzes der PV-Anlage durch den manuellen Prozess nur sehr schwer bis gar nicht umsetzbar, weshalb sich die Integration dieser Anforderungen in einer Anwendung anbietet. In der Modellierung der Soll-Prozesse (vgl. Abbildung 21) schlagen wir dementsprechend die Optimierung des bisherigen Ist-Prozesses durch Einsatz der Anwendung SolarSync vor.

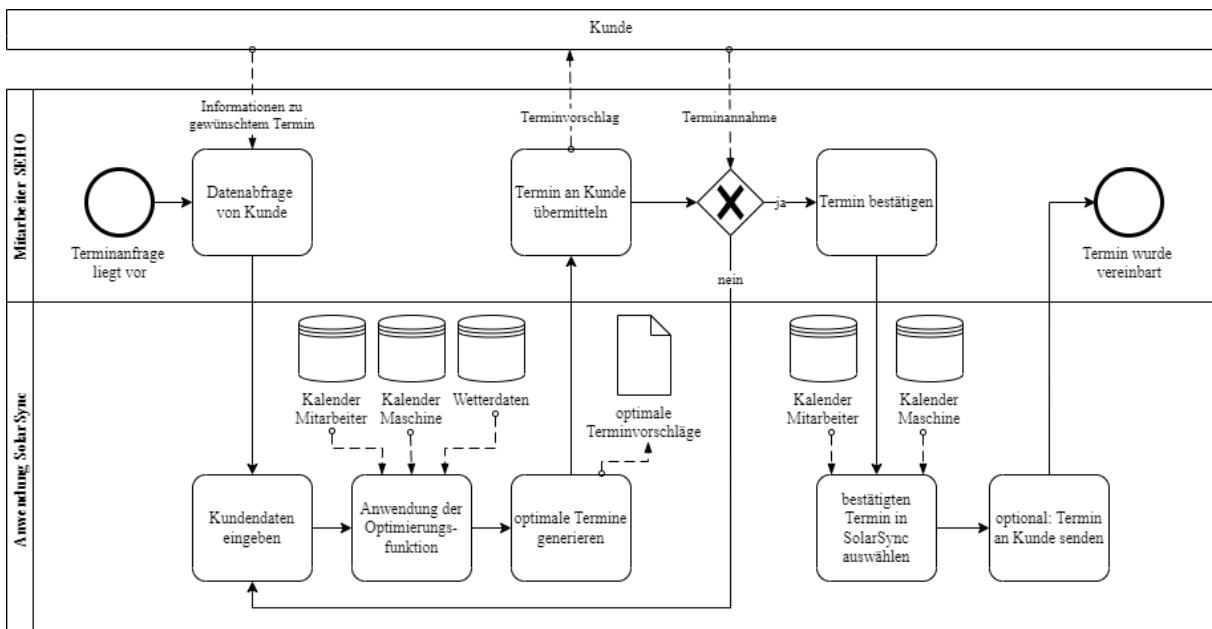


Abbildung 21: Visualisierung des Soll-Prozesses

Das auslösende Event für den Prozess ist weiterhin der eingehende Terminwunsch eines Kunden, welchen der zuständige Ansprechpartner telefonisch oder per E-Mail entgegennimmt. Aus dem Dialog mit dem Kunden leitet der Mitarbeiter den Ressourcenbedarf hinsichtlich Mitarbeiter, Maschinen, Komplexitätsgrad, Zeitraum und Dauer ab. Im nächsten Schritt greift der Mitarbeiter auf SolarSync zu und trägt die Anforderungen des Kunden in die Maske zur

Terminoptimierung ein. Durch das Auslösen der Optimierungsfunktion werden im Hintergrund eine bestimmte Anzahl geeigneter Termine generiert und ausgegeben. Die Berücksichtigung von Kundenanforderungen, bereits belegter Termine von Mitarbeitern und Maschinen, sowie eine möglichst hohe Auslastung der PV-Anlage erfolgt dabei automatisiert durch Integration der Daten über Schnittstellen. Die erzeugten Termine werden an den Kunden übermittelt, welcher dieser in Abstimmung mit seinen eigenen Kapazitäten annehmen oder ablehnen kann. Wird keiner der vorgeschlagenen Termine durch den Kunden angenommen, entsteht eine Prozessschleife, die so lange durchlaufen wird, bis ein geeigneter Termin vereinbart werden kann. Der angenommene Termin wird in SolarSync ausgewählt, was ihn automatisch in den integrierten Kalendern aller beteiligten Mitarbeiter und Maschinen speichert. Zusätzlich kann der Termin als ICS-Datei heruntergeladen oder direkt aus der Anwendung heraus an den Kunden versendet werden. Durch die Bestätigung des Terms in SolarSync ist der Prozess der Terminplanung beendet.

Die Modellierung des Soll-Prozesses verdeutlicht den Mehrwert, der durch SolarSync entsteht. Durch Integration interner Daten wie die Terminkalender von Mitarbeitern und Maschinen kann eine Teilautomatisierung des Prozesses stattfinden. Dies eliminiert Fehlerquellen wie Medienbrüche und beschleunigt den Abgleich von verfügbaren Zeiträumen der jeweiligen Ressource. Durch die Integration externer Daten wie Wetterbedingungen kann zusätzlich die unternehmenseigene PV-Anlage zielgerichtet ausgelastet werden. Durch die so reduzierten Energiekosten entsteht der SEHO ein gewinnbringender Vorteil, während sie gleichzeitig zur Nachhaltigkeit des Unternehmens beitragen. Nicht zuletzt wird kundenseitig ein Mehrwert durch die Integration des vereinbarten Terms in den Kundenkalender geschaffen.

4.3 Designkonzept

Nach dem Designprinzip „Form Follows Function“ müssen zuerst alle Inhalte und Funktionen identifiziert sein, bevor die Ausgestaltung vorgenommen werden kann (RedDot Design Museum 2023). Das Designkonzept baut dementsprechend auf dem Inhaltskonzept auf. In diesem Abschnitt werden das Branding sowie die Erarbeitung des globalen Designs und die Gestaltung der Inhalte besprochen.

4.3.1 Branding

Das Branding umfasst den Entwurf eines markanten Logos mit hohem Wiedererkennungswert sowie die Auswahl eines passenden Namens für die Anwendung. Beim Entwurf spielten unterschiedliche Überlegungen zur Aussagekraft und zur Markenbotschaft eine Rolle. Die Darstellung soll klar und unverwechselbar sein und die Funktion des Produkts symbolisieren. Das Farbschema und die Symbolik sind wichtige Elemente, mit denen die Qualität und der hohe Stellenwert der Nachhaltigkeit durch den Einsatz der Anwendung kommunizieren. Ebenso

sollte der Name den Zweck des Produkts verdeutlichen und zur Marke passen, zudem einprägsam und leicht zu merken sein, um bei den Kunden einen bleibenden Eindruck zu hinterlassen.

Nach Erstellung verschiedener Entwürfe entschied sich das Projektteam für das Design, wie es in Abbildung 22 dargestellt ist. Das Logo greift symbolisch die Funktion eines Dashboards auf und integriert die Nutzung nachhaltiger Sonnenstrahlung. Dies wird vom Farbkonzept betont, nach welchem Gelb für die Sonne, aber auch Kraft und Energie steht, während Grün traditionell Natur und somit Nachhaltigkeit repräsentiert. Der eingängige Name SolarSync kommuniziert die Funktion des Systems – die Synchronisation des Energieverbrauchs von Solarenergie.



Abbildung 22: Produkt Branding

4.3.2 Globales Design

Ein gut gestaltetes Dashboard soll dem Benutzer ermöglichen, Daten schnell zu verstehen und entsprechend zu reagieren (Sprotten 2022). Die Anforderungen an ein gutes Design lassen sich in zwei Kriterien zusammenfassen – die Effektivität eines Dashboards und seine Effizienz. Ein Dashboard ist effektiv, wenn die relevanten Informationen abgedeckt, übermittelt und vom Empfänger verstanden werden. Ein Dashboard ist effizient, wenn zusätzlich die Informationsaufnahme schnell und ohne Verzerrungen der Wahrnehmung erfolgt (Jacobs und Hensel-Börner 2020). Nach Sprotten gibt es drei Gestaltungselemente für Benutzeroberflächen: (1) Das Layout, das für die Anordnung von Daten und Visualisierungen verantwortlich ist und eine klare Struktur haben sollte, um die Orientierung im Datenraum zu erleichtern. (2) Die Systembedienung sollte leicht erlernbar sein, um dem Nutzer eine optimale Unterstützung zu bieten. (3) Bei allgemeinen Formatierungskomponenten sollte darauf geachtet werden, dass Begriffe, Abkürzungen und Meldungen der Benutzeroberfläche auch für Nicht-Experten verständlich sind (Sprotten 2022).

Die globale Gestaltung der Anwendung erfolgte nach Sprotten unter Verwendung des IBCS und des CLEAR(I) Leitfadens nach Jacobs und Hensel-Börner sowie allgemein anerkannten Designprinzipien. Im ersten Schritt wurde ein adaptiv anwendbares Raster erstellt. Je nach Umfang und Struktur der zu transportierenden Information auf der jeweiligen Seite kann eine Variation davon eingesetzt werden (vgl. Abbildung 23). Dabei steht die Informationsaufnahme durch den Nutzer auf einen Blick im Zentrum. Bei der Erstellung des Rasters wurde außerdem die Möglichkeit der Adaption auf Handheld Geräte berücksichtigt, sodass sich Umbrüche nicht negativ auf die Inhalte und dadurch auch auf die User Experience auswirken.

Energiebasierte Terminoptimierung

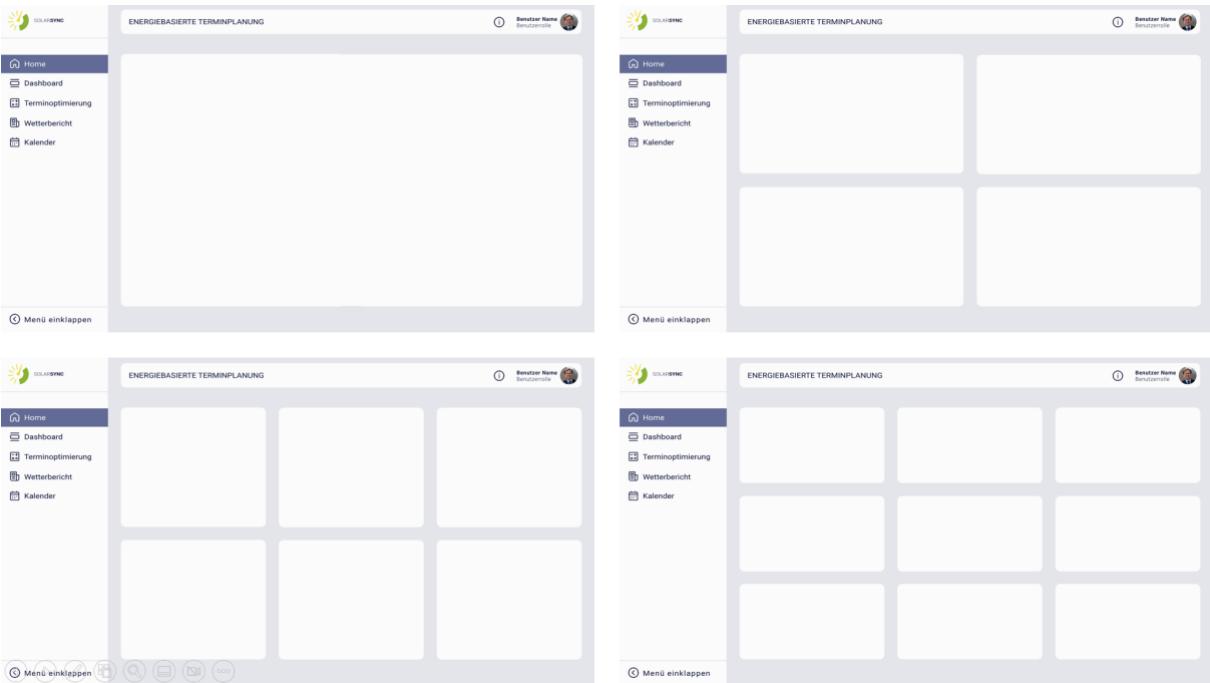


Abbildung 23: Iterative Erstellung eines Layouts für SolarSync

Zudem wurde iterativ ein Farbkonzept auf Basis des 60-30-10-Designprinzips entwickelt, nach welchem 60 % des Platzes für die Primärfarbe vorgesehen ist, 30 % für die sekundäre/unterstützende Farbe und 10 % für einen Akzentfarbton. Diese Verteilung sorgt dafür, dass Farben sparsam verwendet werden und sinnvoll eingesetzt werden. Auf diese Weise wird der Nutzer bei der Eigenlokation und der Navigation durch den Datenraum visuell unterstützt und die User Experience weiter gesteigert. Für die Anwendung SolarSync wurden verschiedene Farbräume in hellen und dunklen Varianten erstellt (vgl. Abbildung 24). Nach einem iterativen Prozess einigte sich das Projektteam auf die Verwendung des hellen Farbschemas bestehend aus einer Kombination von zwei dezent getönten Weißnuancen als Primärfarbe, dem bereits im Logo verwendeten Grünton als Sekundärfarbe zur Betonung wichtiger Informationen und der Konsistenz im Design sowie einem reduzierten Blauton als Akzentfarbe, welche besonders mit den übrigen Farben harmoniert und dennoch auffällig genug ist, um Aufmerksamkeit zu gewinnen. Dabei wurde besonders darauf geachtet, dass bei der Verwendung von Text der Kontrast zwischen Textfarbe und Hintergrundfarbe möglichst hoch ist, da er über die Lesbarkeit entscheidet.



Abbildung 24: Iterative Erstellung eines Farbkonzepts für SolarSync

Zur Umsetzung der Designs und der für alle Seiten der Anwendung erstellten Mockups (vgl. Anhang 6 – Design Mockups) wurde Bootstrap in Kombination mit dem HTML-Code der Anwendung und SCSS zur Ausgestaltung verwendet. Dank dem modularen Charakter von Bootstrap konnten Formulare, Menüs, Icons, Buttons, u.a. konsistent über alle Seiten einheitlich gestaltet werden.

4.4 Reflexion Frontend

Rückblickend lässt sich sagen, dass die Vorabrecherche von Designprinzipien und Gestaltungsrichtlinien sowie eine Auswahl geeigneter Tools und die intensive Einarbeitung in diese bei der Erstellung und Anwendung von Inhalts- und Designkonzept immens hilfreich war. Die Modellierung von Ist- und Sollprozessen half dabei, die bestehenden Prozesse und Bedingungen im Unternehmen besser zu verstehen und die Anwendungskomplexität der geplanten Anwendung entsprechend an die Bedürfnisse der Nutzer zu gestalten. Zudem war die Modellierung der Soll-Prozesse in Verbindung mit den User Stories und daraus entstandenen Projektanforderungen eine solide Grundlage für die Erstellung des Inhaltskonzepts. Dadurch konnte eine klare Strukturierung und hierarchische Ausrichtung aller Funktionen erarbeitet werden. Designprinzipien wie "Form Follows Function" und die 60-30-10-Regel halfen das Design der Anwendung auf ihre Funktionen auszurichten und den Nutzer bei der Navigation und Anwendung von SolarSync optimal zu unterstützen. Die sorgfältige Planung und Umsetzung führten zur Erstellung einer verständlichen und benutzerfreundlichen Anwendung, die den Anforderungen und Bedürfnissen der Nutzer entspricht.

5 Backend

Den Kern des SolarSync-Systems bildet das Backend, welches auf dem Flask-Framework basiert. Der größte Teil hiervon wurde mit Python entwickelt. In diesem Kapitel erfolgt eine ausführliche technische Beschreibung der Umsetzung des Backend sowie eine Vorstellung aller implementierten Funktionen. In der nachfolgenden Abbildung 25 ist der Aufbau des Backend skizziert.

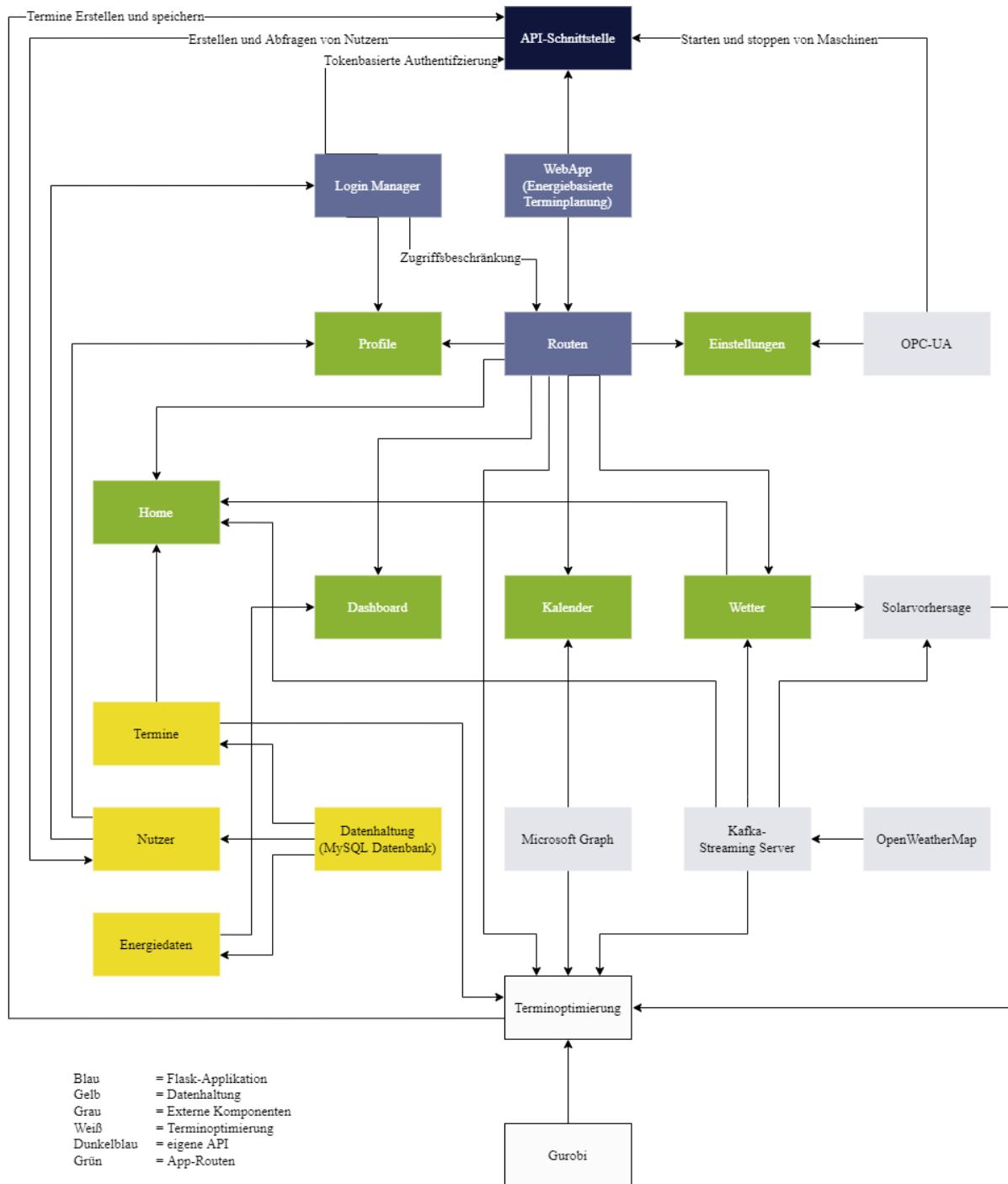


Abbildung 25: Backend-Architektur

Den Kern des Backend bilden die Flask-Module, die dunkelblau gekennzeichnet sind. Eine genaue Erklärung der Flask-Applikation ist dem Kapitel 5.1 zu entnehmen. Die Datenhaltung der Applikation ist in der Abbildung gelb eingefärbt und wird in Kapitel 5.2 beschrieben. Die extern eingebundenen Komponenten wie die Wetter-API, der Streaming-Server, OPC UA, die Solar-Vorhersage und die Microsoftdienste werden in Kapitel 5.3 respektive den jeweiligen Unterkapiteln aufgeführt und deren Implementierung erklärt. In dem darauffolgenden Kapitel 5.4 wird die Terminoptimierung mittels Gurobi aufgeführt. Das letzte Kapitel beschäftigt sich mit der eigenen API der Applikation. Den in Abbildung 25 grün eingefärbten App-Routen wird kein eigenes Kapitel gewidmet, weil sie in den vorhergehenden Kapiteln mit erläutert werden.

5.1 Aufbau der Flask-Applikation

Da die Flask-Applikation eine Vielzahl an Routen und implementierter Logik enthält, ist das Programm auf mehrere Dateien aufgeteilt. Statische Elemente, HTML-Templates und Geschäftslogik ist dadurch räumlich voneinander getrennt und es wird eine einfachere Zusammenarbeit bei der Programmierung gewährleistet. In der Tabelle 2 wird ein grober Überblick der Applikationsstruktur skizziert, wobei der Ordner *app/static/* alle statischen Elemente, wie Bilder, CSS-Dateien und JavaScript-Dateien beinhaltet und der Ordner *app/templates/* alle HTML-Seiten beinhaltet, welche vom Backend aufgerufen werden.

Tabelle 2: Skizze der Applikationsstruktur

Pfad	Beschreibung
app/	
api/	API-Funktionen
routing/	Logik für einzelne Seiten
static/	CSS, SCSS, Schriften, Bilder, JavaScript
templates/	HTML-Seiten
_init.py	App-Initialisierung
errors.py	Fehlermeldungen
forms.py	Definition aller Formulare
graph_settings.json	M365-API Einstellungen
helper.py	Diverse Hilfsfunktionen
models.py	Datenbank-Integration
routes.py	Umsetzung des gesamten Routings

settings.cfg	Konfigurationsdatei
updates_files.sh	Skript zu Entwicklungszwecken
energy_prediction/	Machine Learning Modul
migrations/	Datenbankmigrationen
streaming_data_platform/	Kafka Streaming Server
synthetic_data_generation/	Modellierung synthetischer Daten
config.py	Datenbank-Konfiguration
terminplanung.py	Einstiegspunkt

Die Datei *terminplanung.py* dient als Einstiegspunkt in die Flask-Applikation. Die Datei selbst enthält allerdings keine Logik, sondern importiert lediglich eine Instanz der Flask-Applikation aus einer Variable *app* und startet diese mit dem Befehl *app.run()*. Die Variable *app* sowie alle anderen benötigten Flask-Komponenten werden in der Datei *app/__init__.py* deklariert:

```
app = Flask(__name__)
app.config.from_object(Config)
db = SQLAlchemy(app)
migrate = Migrate(app, db)
login = LoginManager(app)
login.init_app(app)
login.login_view = 'login'
bcrypt = Bcrypt(app)
```

Tabelle 3: Grundlegende Komponenten der Flask-Applikation

Die Config (*config.py*), SQLAlchemy und Migrate stellen die Grundlagen für die Datenbankverbindung und die relationale Datenhaltung dar. Der LoginManager ist die Grundlage des Berechtigungskonzepts und prüft, ob ein Nutzer berechtigt ist, auf bestimmte Inhalte zuzugreifen. Bcrypt ist für die verschlüsselte Speicherung von den Login-Daten in der Datenbank verantwortlich. Diese Komponenten werden im Kapitel Datenhaltung näher erläutert. Nach der Deklarierung der benötigten Variablen werden abschließend der API-Layer als Blaupause sowie Routen, Models und Errors importiert. Details zum API-Layer werden im Kapitel API-Schnittstelle erläutert.

Der Import *app/errors.py* ist ein Errorhandler und fängt verschiedene Fehlerquellen ab:

- Error 404 (Not Found): Tritt auf, wenn versucht wird, eine App-Route anzusprechen, welche nicht existiert. Hier wird auf eine Fehlerseite weitergeleitet.
- Error 500 (Bad Request): Tritt auf, wenn ein Fehler im Python Code auftritt (z. B. eine Variable ist *None*, welche nicht *None* sein darf). Hier wird auf eine Fehlerseite weitergeleitet und potenzielle Datenbankänderungen der Funktion werden rückgängig gemacht.

Das Errorhandling ist hier zentral sowohl für die Nutzung der Webanwendung als auch für die Nutzung von API-Aufrufen implementiert. Der Import `app/models.py` wird im Kapitel Datenhaltung näher spezifiziert. Der Import `app/routes.py` enthält alle App-Routen. In der Datei selbst sind die Routen `/home`, `/dashboard`, `/pv_anlage` und `/return-files` verankert, die restlichen Routen werden aus dem Verzeichnis `app/routing/` importiert. Tabelle 4 enthält eine vollständige Übersicht aller Routen, welche von der Datei `app/routes.py` bereitgestellt werden. Der Aufbau der nachfolgenden Kapitel orientiert sich an dieser Routing-Struktur.

Route	In Datei	Behandelt in Kapitel
<code>/home</code>	<code>/app/routes.py</code>	5.1 Aufbau der Flask-Applikation
<code>/pv_anlage</code>	<code>/app/routes.py</code>	5.1 Aufbau der Flask-Applikation
<code>/return-files</code>	<code>/app/routes.py</code>	5.3.4 Kalender-API
<code>/profile</code>	<code>/app/routing/auth.py</code>	5.2 Datenhaltung
<code>/login</code>	<code>/app/routing/auth.py</code>	5.2 Datenhaltung
<code>/logout</code>	<code>/app/routing/auth.py</code>	5.2 Datenhaltung
<code>/calendar</code>	<code>/app/routing/calendar.py</code>	5.3.4 Kalender-API
<code>/calendar/<id></code>	<code>/app/routing/calendar.py</code>	5.3.4 Kalender-API
<code>/optimization</code>	<code>/app/routing/optimization.py</code>	5.4 Terminoptimierung
<code>/appointments</code>	<code>/app/routing/optimization.py</code>	5.4 Terminoptimierung
<code>/save-optimization</code>	<code>/app/routing/optimization.py</code>	5.4 Terminoptimierung
<code>/settings</code>	<code>/app/routing/settings.py</code>	5.2 Datenhaltung
<code>/weather</code>	<code>/app/routing/weather.py</code>	5.3.1 Wetter-API

Tabelle 4: Übersicht aller Routen in `app/routes.py`

Der Ordner `migrations/` enthält Informationen zu Datenbankupdates. `energy_prediction/`, `streaming_data_platform/` und `synthetic_data_generation/` enthalten Inhalte, welche von der Flask-Applikation genutzt werden, allerdings kein direkter Teil davon sind. Diese werden in den folgenden Kapiteln behandelt. Abschließend bündelt das Verzeichnis `/app/api/` die gesamte Funktionalität der Flask-Anwendung und stellt einen REST-Layer für die Kernfunktionen zur Verfügung. Die API wird im Kapitel API-Schnittstelle im Detail behandelt.

5.1.1 Home-Route

Die Route `/home` ist der Hauptendpunkt der Webanwendung. Diese Route dient der Darstellung der Startseite des Systems, welche als Dashboard mit wichtigen Daten umgesetzt ist. Die

gleichnamige Funktion `home()` dient der Bereitstellung der benötigten Daten sowie dem Aufruf der HTML-Seite. Das Dashboard unterteilt sich in vier Elemente, welche unterschiedliche Daten visualisieren. Im Folgenden wird beschrieben, wie die Daten dieser vier Bausteine bereitgestellt werden.

Zur Berechnung der erwarteten PV-Energie in den nächsten 14 Tagen werden zunächst die Solardaten aus einer CSV-Datei mithilfe der Pandas-Methode `read_csv()` eingelesen. Diese Datei enthält für das Jahr 2023 die stündlich maximale und minimale Strommenge in Kilowattstunden, welche in dieser Stunde durch die PV-Anlage der SEHO erzeugt werden kann. Eine genauere Beschreibung der Erstellung dieses Datensatzes findet sich in Kapitel 5.4. Anschließend werden die benötigten Wetterdaten von dem Streaming Server in JSON-Format abgerufen. Aus diesen Daten wird nur der Grad der Bewölkung extrahiert. Da die Wettervorhersage auf täglichen Werten basiert, wird ein Resampling der Wolkendaten von täglich auf stündlich durchgeführt, wobei die Werte interpoliert werden. Schließlich werden die Wolkendaten mit den Energiedaten zusammengeführt und der gewünschte Zeitraum der nächsten zwei Wochen gewählt. Anschließend wird die zu erwartende Strommenge, welche durch die PV-Anlage erzeugt wird, anhand einer linearen Interpolation zwischen den minimalen und maximalen Energiewerten und dem Bewölkungsanteil berechnet. Je stärker die Bewölkung, desto geringer ist die erzeugte Strommenge der PV-Anlage. Die erwartete PV-Energie wird auf Tageswerte aggregiert und als Liste dem Dashboard übergeben.

Die Kennzahl zur Auslastung der PV-Anlage bildet das Verhältnis zwischen der tatsächlichen PV-Strommenge und der maximal möglichen Strommenge am aktuellen Tag. Die tatsächliche Strommenge wird basierend auf dem tagesaktuellen Bewölkungsgrad berechnet. Dieser Wert wird mit dem Maximalwert aus den Solardaten zu einer prozentualen Kennzahl verrechnet.

In der Kachel „CO₂-Ersparnis“ werden Informationen rund um die CO₂-Einsparung angezeigt. Hierzu sind die Daten aller jemals geplanten Termine erforderlich. Hierzu werden alle gespeicherten Termine aus der Datenbanktabelle „Termin“ mit Hilfe von SQLAlchemy abgerufen und formatiert. Für jeden Termin werden der prozentuale Anteil der PV-Energie und die eingesparten CO₂-Emissionen berechnet. Zur Berechnung der CO₂-Emissionen wird der Faktor 0,412 kg CO₂ pro kWh zugrunde gelegt. Des Weiteren wird die Summe der am aktuellen Tag eingesparten Menge an CO₂ sowie der gesamten CO₂-Einsparung gebildet.

Für die Kachel „nächste Termine“ wird die aktuelle Uhrzeit sowie das Datum übergeben. Zudem wird aus der Liste aller geplanten Termine in der Datenbank die nächsten zwei zukünftigen Termine gefiltert und ausgegeben. Außerdem wird ein „Countdown“ zu dem nächsten Termin berechnet.

Für die „Wetterbericht“-Kachel werden erneut die Wetterdaten der kommenden 30 Tage von dem Streaming Server im JSON-Format abgerufen. Hierbei werden alle verfügbaren Daten in

mehreren Listen formatiert und abgelegt. Eine Liste mit allen relevanten Wetterinformationen wird dem Dashboard übergeben.

Die `home()`-Funktion gibt am Ende die Flask-Funktion `render_template()` zurück. Diese Funktion ruft die `home.html` Seite auf und übergibt alle Daten, die zuvor abgerufen wurden.

5.1.2 PV-Anlage-Route

Die Route der PV-Anlage dient der Darstellung einer Analyseseite, welche Daten rund um die PV-Anlage visualisiert. Diese Seite ist wie die Home-Seite in vier Kachel aufgeteilt. Eine erste Übersichtskachel stellt die Auslastung der PV-Anlage sowie die erwartete PV-Energie in den nächsten 14 Tagen dar. Die Vorverarbeitung der hierfür benötigten Daten wurde bereits in dem vorigen Unterkapitel erläutert, da die gleiche Kachel auf der Home-Seite dargestellt wird.

In einer zweiten Kachel wird ein Liniendiagramm mit der erwarteten PV-Energie und den Stromverbräuchen der Termine in den kommenden 14 Tagen dargestellt. Hierfür wird erneut auf die Daten der PV-Vorhersage zurückgegriffen. Die Beschreibung hierfür befindet sich ebenfalls im Kapitel 5.1.1. Um die Verbrauchsdaten der Termine zu erhalten, werden alle geplanten Termine aus der Termindatenbank abgefragt und nach zukünftigen Terminen gefiltert. Die Termindaten werden in ein Dataframe überführt und ein Resampling nach Tagen durchgeführt. Durch das Resampling entsteht für jeden Tag ein Eintrag, auch wenn an diesem kein Termin geplant ist und es somit keinen Stromverbrauch gibt. Der Energieverbrauch von Terminen an dem gleichen Tag wird summiert. Zudem wird das resultierende Dataframe auf die kommenden 14 Tage gefiltert. Eine Liste dieser Stromverbrauchswerte wird zusammen mit der Liste der erwarteten PV-Energie an das Frontend gegeben.

Die dritte Kachel dient der Darstellung der CO₂-Ersparnis im Verlauf der letzten Woche als kumulierte Darstellung. Hierzu wird erneut auf die Termindaten, welche bereits aus der Datenbank geladen wurden, zugegriffen. Es wird ebenfalls ein Resampling dieser Daten auf Tage durchgeführt. Zudem wird für jeden Tag die kumulierte Summe der CO₂-Ersparnis in Kilogramm historisch bis zum aktuellen Tag berechnet. Eine Liste der Werte für die letzten 7 Tage wird dem Frontend übergeben.

In der vierten Kachel wird der Autarkiegrad der Stromversorgung der Termine sowohl insgesamt als auch für die letzte Woche separat dargestellt. Hierzu werden die Termindaten basierend auf Tagen gruppiert. Zur Berechnung der Autarkie über den Gesamtzeitraum wird zunächst die Summe der Stromverbräuche aller Termine sowie die Summe der genutzten PV-Energie aller Termine gebildet. Im Anschluss wird die Summe der PV-Energie durch die Summe der Verbräuche geteilt und mit 100 multipliziert. Das Ergebnis ist die globale Autarkie in Prozent. Die Berechnung für die 7-Tages-Autarkie erfolgt analog auf den Termindaten der letzten Woche.

Zum Abschluss dieser Route werden alle erzeugten Daten dem Frontend mit der `render_template`-Methode übergeben. Dabei wird die HTML-Datei `pv_anlage.html` aufgerufen.

5.2 Datenhaltung

Die Datenhaltung der Flask-Applikation ist zweiteilig. Einerseits wird zur Speicherung von Nutzerdaten und Termindaten eine relationale MySQL-Datenbank genutzt. Andererseits existiert im Applikationsverzeichnis die Datei `/app/settings.cfg`, welche durch einen Configparser durch die `/settings`-Route der Applikation verwaltet wird. Diese Konfigurationsdatei enthält andere Variablen, welche für eine korrekte Funktionsweise der Software verantwortlich sind. Diese Komponenten der Datenhaltung werden in den folgenden beiden Unterkapiteln näher beleuchtet.

5.2.1 Relationale Datenbankspeicherung

Die Applikation nutzt eine relationale MySQL-Datenbank, um erstellte Termine und Nutzerdaten zu speichern. Zur Verwaltung der Datenbank werden die Python-Importe `flask_sqlalchemy` und `flask_migrate` genutzt. `flask_sqlalchemy` ermöglicht klassische Datenbankabfragen. Zum Herstellen einer Verbindung werden in der Flask-Config die Parameter „`SECRET_KEY`“ und „`SQLALCHEMY_DATABASE_URI`“ gesetzt. Der Connectionstring ist nach dem Schema „`mysql+pymysql://Nutzer:Passwort@localhost:3306/Datenbankname`“ aufgebaut, insofern eine MySQL-Datenbank mit Standardeinstellungen auf derselben Maschine aufgesetzt ist. Der Import `flask_migrate` ermöglicht, Änderungen an Datenbankmodellen durchzuführen und diese versioniert zu speichern. Dazu wird im Applikationsverzeichnis der Ordner `migrations/` angelegt.

In der Datei `models.py` werden die Tabellen mit Nutzern und Terminen für die Datenbank spezifiziert. Dazu wird jeweils eine Pythonklasse mit dem Parameter `db.model` angelegt. Für einen Termin wird die TerminID als Primärschlüssel festgelegt. Zudem werden Datum und Uhrzeit, Terminbeschreibung, Dauer, Energieverbrauch, Solarstrommenge, involvierte Maschinen und Mitarbeiter sowie Erstellungszeit in der Datenbank gespeichert. Für jeden Nutzer wird eine ID, Nutzername, Nutzerrolle, Profilbild-URL, API-Token mit Ablaufdatum und ein durch Bcrypt gehashtes Passwort gespeichert. Nach einer Anpassung der Datenbankmodelle kann die relationale Datenbank über das Flask-CLI angepasst werden. Dazu werden die Befehle `flask db init` (nur beim erstmaligen Aufsetzen der Datenbank), `flask db migrate -m „Updatenachricht“` und `flask db upgrade` genutzt.

Innerhalb der Flask-Applikation kann auf Datensätze zugegriffen werden, wenn die Objekte der Klasse, welche für die Erstellung einer Datenbanktabelle genutzt wird, gefiltert wird.

Beispielsweise kann der aktuell eingeloggte Nutzer über den Befehl `User.query.filter_by(id = flask_login.current_user.id).first()` identifiziert werden.

Neue Termine können im Laufe der Terminoptimierung in der Datenbank gespeichert werden. Genauer ist dieses Vorgehen im Kapitel dieser Funktion beschrieben. Nutzerdaten können einerseits durch den Login-Manager, als auch durch Nutzer mit der Admin-Rolle erstellt, bearbeitet und gelöscht werden. Beim erstmaligen Aufsetzen der Applikation wird immer die Route `login` mit dem Registrierungsformular aufgerufen, da sich zu diesem Zeitpunkt noch kein Nutzer in der Datenbank befindet. Auf der Registrierungsseite muss der Anwender ein Formular mit gewünschtem Nutzernamen, Passwort und Rollenzuweisung ausfüllen.

```
form = RegisterForm()
if form.validate_on_submit() and 'registerForm' in request.form:
    hashed_password = bcrypt.generate_password_hash
        (form.password.data)
    new_user = User(username=form.username.data,
                    password=hashed_password,
                    role=form.role.data)
    db.session.add(new_user)
    db.session.commit()
    return redirect('/')
```

Tabelle 5: Anlegen eines neuen Nutzers in der Datenbank über die Register-Route

Tabelle 5 zeigt, dass innerhalb der Route `login`, wenn das Registrierungsformular erfolgreich an den Server gesendet wird, ein neues Objekt vom Typ User mit den Daten des Formulars und einem verschlüsselten Passwort angelegt und der Datenbanktabelle User hinzugefügt wird. Nach erfolgreicher Registrierung wird der Nutzer auf die Startseite umgeleitet. Wenn bereits ein Nutzer existiert, wird anstatt dem Registrierungsformular ein Login-Formular erzeugt. Bei der Auswertung des Login-Formulars wird überprüft, ob der Anwender eine korrekte Kombination aus Nutzernamen und Passwort eingegeben hat. Falls dies der Fall ist, wird der Nutzer eingeloggt und auf die Startseite weitergeleitet, ansonsten wird die `login`-Route mit einer passenden Fehlermeldung erneut aufgerufen. Existiert kein Nutzeraccount mit dem eingegebenen Namen wird die Fehlermeldung „Der Benutzer existiert nicht“ ausgegeben, falls das eingegebene Passwort nicht mit dem entschlüsselten Passwort der Datenbank übereinstimmt, wird die Fehlermeldung „Das Passwort ist falsch“ zurückgegeben (vgl. Tabelle 6).

```
def validate(self):
    if not FlaskForm.validate(self):
        return False
    user = User.query.filter_by(username=self.username.data).first()
    if user:
        if not bcrypt.check_password_hash(user.password,
self.password.data):
            self.password.errors.append('Das Passwort ist falsch')
            return False
    else:
```

```
    self.username.errors.append('Der Benutzer existiert nicht')
    return False
login_user(user)
return True
```

Tabelle 6: Einloggen eines Nutzers in die Applikation

Beim Ausloggen eines Nutzers (die Route `/logout`) werden lediglich Daten des eingeloggten Nutzers aus der aktuellen Sitzung gelöscht und die Funktion `logout_user()` des Login-Managers aufgerufen. Der Anwender wird daraufhin auf die Login-Seite weitergeleitet, da ansonsten keine Route ohne einen eingeloggten Nutzer aufrufbar ist.

Der eingeloggte Nutzer selbst hat über die Route `/profile` die Möglichkeit, sein Passwort zu ändern oder ein Profilbild hochzuladen. Falls ein Profilbild hochgeladen wird, wird es in der Ordnerstruktur `/static/img/profile/` abgelegt und in der Datenbank wird lediglich der Link zu dem Bild hinterlegt. Beim Ändern des Passworts muss der Nutzer das bisherige Passwort korrekt eingeben und darauffolgend zwei identische Passwörter in den Feldern „Neues Passwort“ und „Neues Passwort wiederholen“. Sollte eine der beiden Anforderungen nicht erfüllt sein, wird das Passwort nicht geändert und der Anwender erhält eine passende Fehlermeldung. Ein Administrator hat über die Route `/settings` die Möglichkeit, neue Nutzer über das Registrierungsformular anzulegen sowie Nutzer zu löschen. Weiterhin können hier die Rollen der einzelnen Nutzeraccounts zwischen „Standard“ und „Administrator“ gewechselt werden.

5.2.2 Konfigurationsdatei im Applikationsverzeichnis

Bei der Implementierung der Applikation werden global gültige Einstellungswerte in der `settings.cfg` gespeichert. Die Auslagerung in eine eigene Datei schafft mehrere Vorteile für die Anwendung:

Flexibilität: Die Verwendung der Konfigurationsdatei ermöglicht es Benutzern, Einstellungen wie beispielsweise Pfade, Ports, Verbrauchswerte und Kennwörter flexibel anzupassen, ohne dass Änderungen am eigentlichen Code erforderlich sind. Das ist notwendig, da die Anwendung nicht für ein spezifisches Unternehmen entwickelt wird, sondern universell einsetzbar sein soll. Zudem können sich die Werte bei einer langfristigen Verwendung der Software bei den Unternehmen ändern.

Sicherheit: Durch die Verwendung der Konfigurationsdatei können sensible Informationen wie API-Secrets und Kennwörter von dem öffentlich zugänglichen Code abgetrennt werden. Dadurch wird die Sicherheit des Projekts verbessert, da sensible Informationen nicht versehentlich eingesehen oder abgerufen werden können.

Wiederverwendbarkeit: Durch die Benutzung der Konfigurationsdatei in dem Projekt können die Einstellungen in verschiedenen Teilen des Codes wiederverwendet werden. Dies vermeidet

die Notwendigkeit, die gleichen Werte mehrfach im Code anzugeben. Dadurch wird die Lesbarkeit und Struktur des Codes verbessert.

Technisch wird die Unterstützung der Konfigurationsdatei mit der Python Standardbibliothek *configparser* umgesetzt. In der *helper.py* befinden sich die beiden Hilfsfunktionen *get_config()* bzw. *write_config()*, die es ermöglichen, Werte aus der *settings.cfg* auszulesen oder reinzuschreiben. Über diese Hilfsfunktionen sind alle verwendeten Routen in der Lage, Lese- sowie Schreibzugriff auf die Konfigurationsdatei zu erhalten. In dieser sind Informationen für die OpenWeather-API sowie für die Maschinendaten hinterlegt. Zudem befinden sich dort die Verbindungsinformationen für den Mail-, Kafka- und OPC UA-Server.

Der Anwender der Applikation kann über die Einstellungsseite im Frontend bequem alle aufgeführten Werte einsehen und anpassen, wenn er in der Nutzerverwaltung der Applikation die entsprechenden Berechtigungen besitzt.

5.3 Externe Komponenten

In diesem Kapitel wird beschrieben, wie die externen Komponenten der Applikation implementiert werden. Das erste Unterkapitel 5.3.1 erläutert den Abruf der Wetter-API. Die dadurch erhaltenen Daten werden an die im Unterkapitel 5.3.2 erläuterte Streaming-Data-Platform übergeben. Diese verwaltet die Datenströme innerhalb der Applikation und bietet eine Schnittstelle für die externen Komponenten. Das Unterkapitel 5.3.3 ruft Daten der Streaming-Data-Platform ab und beschreibt die Umsetzung der Solar-Vorhersage. Die Einbindung der Kalender-API wird im Unterkapitel 5.3.4 aufgeführt. Im letzten Unterkapitel 5.3.5 wird die Maschinensteuerung mittels OPC UA beschrieben. Dieser Vorgang stellt den letzten Prozess in der Terminkalenderoptimierung dar: Zuvor wurde der optimale Termin aus der Solar-Vorhersage bestimmt und mittels API im Kalender hinterlegt. Ist nun der Zeitpunkt gekommen, an dem der Termin bevorsteht, werden mittels OPC UA die im Termin hinterlegten Maschinen gestartet.

5.3.1 Wetter-API

Die Applikation benötigt für den Optimierungsalgorithmus der Terminkalenderplanung präzise Informationen über die Bewölkung am Standort der PV-Anlage. Daraus wird die Sonneneinstrahlung berechnet. Genaue Erläuterungen der Optimierung sind in Kapitel 5.3.3 zu finden. Um die benötigten Informationen bereitzustellen, wird auf Daten der OpenWeatherMap-Schnittstelle zurückgegriffen. Da die Applikation eine Terminplanung über einen Zeitraum von 30 Tagen zulässt, werden Wetterinformationen über diesen Zeitraum benötigt. Dafür wird der Climatic Forecast angebunden, der tägliche Wetterdaten der nächsten 30 Tage bereitstellt (OpenWeather 2023).

Die technische Umsetzung der Anbindung erfolgt als Teil der Streaming-Data-Platform in der Datei *producer_weather.py*. Dort werden zuerst die Informationen über den Standort der PV-Anlage sowie der OpenWeatherMap-API-Key aus der Konfigurationsdatei abgerufen. Der API-Request wird mit der Python Library *Requests* umgesetzt. Diese Library ermöglicht es, in Python HTTP/1.1-Anfragen durchzuführen und die Antworten als JSON-Format weiterzuverarbeiten (Requests 2023). Der Aufbau des POST-Befehl und der Antwort von OpenWeatherMap sind in der nachfolgenden Tabelle ersichtlich.

POST	https://pro.openweathermap.org/data/2.5/forecast/climate	
	Parameter	units „metric“ lat <config: lat> lon <config: lon> appid <config: key> lang „DE“
Antwort (Auszug)	Body	<pre>{ "city": { "id": Integer, "name": String, "coord": { "lon": Float, "lat": Float }, "country": String, "population": Integer, "timezone": Integer }, "code": String, "message": Float, "cnt": Integer, "list": [{ "dt": Integer, "sunrise": Integer, "sunset": Integer, "temp": { "day": Float, "min": Float, "max": Float, "night": Float, "eve": Float, "morn": Float }, "feels_like": { "day": Float, "night": Float, "eve": Float, "morn": Float } }] }</pre>

```

        },
        "pressure":Integer,
        "humidity":Integer,
        "weather": [
            {
                "id":Integer,
                "main":String,
                "description":String,
                "icon":String
            }
        ],
        "speed": Float,
        "deg":Integer,
        "clouds":Integer,
        "rain": Float,
        "snow": Float
    }
}

```

Tabelle 7: Abfrage der OpenWeatherMap-API

Die Antwort enthält im ersten Teil generelle Informationen über die Anfrage und die Lokalisierung des Wetterberichtes. Darauf folgt für jeden der nächsten 30 Tage ein detaillierter Wetterbericht. Die oben abgebildete Tabelle enthält die Daten für den ersten Tag. Der JSON-String der Antwort wird mittels eines *Kafka Producers* an die Streaming-Data-Platform der Applikation übergeben. Der genaue Ablauf wird in dem folgenden Kapitel beschrieben.

5.3.2 Streaming-Data-Platform

Die Streaming-Data-Plattform dient als zentrale Verwaltung der Datenströme innerhalb der Applikation. Sie ist bewusst so konzipiert, dass sie offen angelegt ist, um in zukünftigen Ausbaustufen des Projekts weitere Module miteinander zu verbinden. In der aktuellen Ausbaustufe ist die Wetter-API aus Kapitel 5.3.1 sowie die Solar-Vorhersage aus Kapitel 5.3.3 angebunden. Das technische Grundgerüst der Streaming-Data-Plattform bildet Apache Kafka. Mit der Streaming-Data-Plattform kommunizieren zwei Arten von Anwendungen: Die Producer, welche Daten schreiben und die Consumer, welche Daten abrufen.

In Python erfolgt die Implementierung mit der Bibliothek *Kafka*, welche sowohl *Producer* als auch *Consumer* bereitstellt. In der Applikation liegt innerhalb des Verzeichnisses der Streaming-Data-Plattform die *producer_weather.py*, welche Wetterdaten per API abruft und diese an die Streaming-Data-Plattform sendet. Dies erfolgt in einem festen Zeitintervall, um die Anfragen an die API zu minimieren, da dafür individuelle Kosten anfallen können. Benötigt die Applikation die Wetterinformationen, ruft die *consumer_weather.py* den aktuellen Wetterbericht von der Streaming-Data-Plattform ab. Das kommt jedes Mal vor, wenn im Frontend ein neuer Termin geplant wird. An dieser Stelle wird der Vorteil der Lösung deutlich: Wenn hintereinander mehrere Termine geplant werden, wird nur eine API-Anfrage an die OpenWeatherMaps-API gestellt. Ohne die Streaming-Data-Plattform würde jedes Mal ein

Request an die OpenWeatherMap-API durchgeführt werden und gegebenenfalls deutlich höhere Kosten entstehen. Das Ausfallrisiko dieser Lösung wird zusätzlich erhöht, indem die *consumer_weather.py* die Daten nicht direkt an die Solar-Vorhersage übergibt, sondern sie als JSON-Datei zwischenspeichert. Sollte der Server, auf dem die Streaming-Data-Platform gehostet wird, technische Probleme haben, steht so zumindest der zuletzt abgerufene Wetterbericht zur Verfügung und es kommt zu keinem Totalausfall der Applikation.

In weiteren Entwicklungsstufen ist angedacht, dass die realen Echtzeitdaten der PV-Anlage an die Streaming-Data-Plattform übergeben werden, die dann von dem Optimierungsmodell abgerufen und verarbeitet werden. Zum aktuellen Zeitpunkt liegen die Daten nicht vor, da die PV-Anlage der SEHO noch nicht installiert ist. Deshalb werden in der Solar-Vorhersage in Kapitel 5.3.3 Annahmen für die Daten getroffen und dem Optimierungsmodell übergeben.

5.3.3 Solar-Vorhersage

Die energiebasierte Terminoptimierung hat das Ziel der Minimierung des Strom-Netzbezuges, welcher für die Durchführung der Termine erforderlich ist. Hierfür werden zum einen Daten der Verbraucher, in diesem Fall also die Grundlast und Maschinen der SEHO, und zum anderen die Daten der Erzeuger benötigt. Da die SEHO die Installation einer Photovoltaikanlage (PV) plant, stellt diese Anlage den Erzeuger dar. Um den möglichen Netzbezug, welcher durch die Termine verursacht wird, zu berechnen, ist zunächst eine Vorhersage der erzeugten PV-Energie in der Zukunft erforderlich. In diesem Kapitel wird zunächst die Technik dieser Vorhersage detailliert beschrieben und im Anschluss ein Ausblick auf eine zukünftig mögliche Methode zur Vorhersage gegeben.

5.3.3.1 Ist-Zustand

Das Optimierungsmodell in dem Produktivsystem basiert auf Solardaten, welche für das Jahr 2023 die stündlich minimale und maximale PV-Energie in Kilowattstunden enthalten. Nachfolgend wird die Erstellung dieses Datensatzes ausführlich beschrieben. Die Grundlage der Solardaten ist ein Datensatz der National Solar Radiation Database, welcher Daten zur stündlichen Globalstrahlung für das Jahr 2019 an dem Standort der SEHO enthält (NSRD 2019).

In dem besagten Datensatz gibt das Attribut "Clearsky GHI" (Global Horizontal Irradiance) die Globalstrahlung an, welche bei einer Bewölkung von 0 % auf eine horizontale Ebene an dem Standort der SEHO im Jahr 2019 eingetroffen ist. Dieser Wert stellt das Maximum an möglicher Sonneneinstrahlung für jede Stunde im Jahr dar. Es wird angenommen, dass dieser Wert zwischen verschiedenen Jahren nur geringfügig schwankt. Auf Grundlage dieser Werte soll mithilfe der Bewölkungsdaten, die über die OpenWeatherMap-API für die nächsten 30 Tage

abgerufen werden, die Globalstrahlung berechnet werden, die auf den Modulen der Photovoltaikanlage ankommt. Dadurch wird es möglich, die potenzielle Strommenge zu prognostizieren, die in den nächsten 30 Tagen durch die PV-Anlage erzeugt werden kann.

Nachfolgend wird das Vorgehen zur Berechnung der maximalen und minimalen Strommenge, welche durch die PV-Anlage erzeugt werden kann, beschrieben. Bei der Berechnung der maximal möglichen Strommenge, die von einer Photovoltaikanlage bei klarem Himmel produziert werden kann, müssen verschiedene Faktoren berücksichtigt werden. Dazu gehören die Sommer- und Winterzeit, da sich die Sonnenstunden und damit die Globalstrahlung im Jahresverlauf ändern. Des Weiteren muss die Globalstrahlung in kWh/m² umgerechnet werden, um sie mit anderen Kennwerten vergleichbar zu machen. Zudem ist es wichtig, die Neigung und Ausrichtung der Solarmodule zu berücksichtigen, wie es in der Simulation von ENVIRIA (siehe Anhang 7) dargestellt wird. Die Simulation geht von einer Energiebilanz aus..., bei der 2 % der Globalstrahlung aufgrund von Ausrichtung, Neigung, Bodenreflexion, Abweichung vom Standardspektrum und modulunabhängiger Abschattung verloren gehen. Um die mögliche Strommenge zu berechnen, wird die Globalstrahlung mit der Modulfläche von 2.921,31 m² multipliziert. Dabei werden auch weitere Aspekte berücksichtigt, wie der Modul-Nennwirkungsgrad (21,77 %), die modulspezifische Teilabschattung (-0,57 %), das Schwachlichtverhalten (+ 0,26 %), die Abweichung von der Nenn-Modultemperatur (-1,2 %), der Mismatch (Herstellerangaben, Verschaltung, Abschattung) (-2,25 %) sowie Verluste durch DC/AC-Wandlung, Kabelverluste, Wechselrichter und Abregelung (- 3,6 %). All diese Werte stammen aus der Simulation von ENVIRIA. Nach Abzug dieser Faktoren ergibt sich die mögliche Strommenge, die von der PV-Anlage unter "Clearsky"-Bedingungen (0 % Bewölkung) produziert werden kann.

Im Folgenden wird das Vorgehen zur Berechnung der minimalen Strommenge durch die PV-Anlage bei 100 % Bewölkung erläutert. Hierbei wird zunächst das Verhältnis zwischen der tatsächlichen Globalstrahlung (GHI) und der Globalstrahlung ohne Bewölkung (Clearsky GHI) berechnet. Dieses Verhältnis gibt an, wie viel Sonnenstrahlung auf die Solarmodule bei Bewölkung trifft. Da es verschiedene Arten von Wolken in dem Datensatz gibt, wird im nächsten Schritt ein Durchschnittswert der relativen Reststrahlung bei Bewölkung über alle Wolkenarten berechnet. An dem Standort der SEHO ergibt sich eine durchschnittliche relative Reststrahlung von 33,15 % des Clearsky GHI Wertes. Dieser Wert stellt somit die minimale relative Strommenge dar, welche bei 100 % Bewölkung erzeugt werden kann. Um die absolute minimale Strommenge zu ermitteln, wird der Maximalwert für jeden Zeitpunkt mit 0,3315 multipliziert. Dieses Ergebnis gibt Auskunft darüber, wie viel Energie auch unter den schlechtesten Bedingungen produziert werden kann.

Nachdem der Maximal- und Minimalwert berechnet wurde, ist die Vorhersage der zu erwartenden PV-Energie in den nächsten 30 Tagen möglich. Hierfür dient der maximale Wert als Referenzwert bei vollständiger Sonneneinstrahlung und der minimale Wert bei vollständiger

Bewölkung. Anschließend werden über die OpenWeather-API Bewölkungsdaten am Standort der SEHO für die nächsten 30 Tage abgerufen. Diese Bewölkungsdaten sind in Prozent angegeben, wobei 100 % eine vollständige Bewölkung darstellt.

Mit Hilfe der oben genannten Referenzwerte sowie der Bewölkungsdaten wird nun die zu erwartende Strommenge mithilfe der folgenden Formel berechnet. Diese Formel berücksichtigt die Bewölkung in Prozent sowie den maximalen und minimalen Wert der Stromerzeugung. Dabei wird für jede Stunde im Planungshorizont die erwartete Strommenge an PV-Energie berechnet.

$$PV - Output = ((Max_Energy - Min_Energy) * (1 - Clouds)) + Min_Energy$$

Die Variablen Max_Energy und Min_Energy enthalten dabei die zuvor ermittelten Werte für die jeweilige Stunde. Der Wert Clouds gibt die Bewölkung in Prozent an und wird von 1 abgezogen, um einen "Sonnenindex" zu erhalten. Liegt dieser bei 100 %, bedeutet dies eine Bewölkung von 0 %, und umgekehrt.

Ein Beispiel zur besseren Interpretation der Formel: Angenommen, der maximale Wert der Stromerzeugung beträgt 100 kWh und der minimale Wert beträgt 33,15 kWh. Bei einer Bewölkung von 50 % ergibt sich mit der oben genannten Formel ein PV-Output von 66,575 kWh. Dies bedeutet, dass in diesem Beispiel bei einer Bewölkung von 50 % eine Strommenge von 66,575 kWh durch die PV-Anlage erzeugt werden kann.

5.3.3.2 Potenzieller Machine Learning Use Case

Da sich die SEHO aktuell noch in der Planungsphase der PV-Anlage befindet, basiert die vorherig beschriebene Vorhersage der PV-Energie auf allgemeinen Globalstrahlungsdaten sowie der Simulation des Energiedienstleisters ENVIRIA, welche individuell für die SEHO durchgeführt wurde. Mit der Installation der Solaranlage wird die Datenerfassung hinsichtlich der erzeugten PV-Energie relevant. Hierbei können über die PV-Anlage die Daten der erzeugten Strommengen zeitbasiert abgefragt und auf dem Streaming Server gespeichert werden. Des Weiteren wäre eine Investition in die kostenpflichtige Solar Radiation-API von OpenWeatherMap ein wichtiger Schritt hin zu einer noch genaueren Vorhersage der PV-Energie und somit einem präziseren Optimierungsmodell. Auch diese Wetterdaten könnten auf dem Streaming Server abgelegt werden. Der aus diesen beiden Datenquellen entstehende Datensatz wäre eine gute Grundlage für den Einsatz eines Machine Learning Algorithmus zur Vorhersage der PV-Energie in der Zukunft. Da zum einen die PV-Anlage noch nicht installiert ist und folglich die Solardaten noch nicht existieren und in diesem Projektseminar mit der kostenfreien OpenWeatherMap 30 Tage Vorhersage gearbeitet wird, wurde die Umsetzung eines Machine Learning Anwendungsfalls auf einem beispielhaften Datensatz (Kaggle 2022) entwickelt. Der verwendete Datensatz umfasst Zeitreihendaten einer PV-Anlage und gibt die erzeugte Energie in kWh in der jeweiligen Stunde an. Ein weiterer Datensatz enthält die dazu

passenden Zeitreihendaten mit der Sonneneinstrahlung (Solar Irradiation) an dem Standort der PV-Anlage. Dieser Datensatz wurde bewusst ausgewählt, um möglichst nah an dem potenziellen Anwendungsfall bei der SEHO zu bleiben. Eine Übertragung der Pipeline zum Training und zur Vorhersage auf die „neuen“ Daten der SEHO ist daher mit wenig Aufwand möglich.

Im Folgenden wird die Entwicklung einer Pipeline zur Vorhersage der PV-Energie basierend auf Machine Learning beschrieben. Die Pipeline besteht aus drei Funktionen, welche der Datenvorverarbeitung, dem Modeltraining und der Vorhersage dienen. Der dazugehörige Python-Code findet sich in dem Ordner *energy_prediction/* in der Datei *model_training.py*.

1) Datenvorverarbeitung

In der Python-Funktion *preprocessing()* werden mehrere Schritte zur Vorverarbeitung von zwei Datensätzen im Zusammenhang mit der Solarenergieerzeugung durchführt. Zunächst werden zwei CSV-Dateien geladen, welche historische Zeitreihen-Daten über die von der PV-Anlage erzeugte Energie und die dabei gemessene Sonneneinstrahlung enthalten.

Im Anschluss werden die für die Vorhersage benötigten Attribute der Datensätze ausgewählt bzw. nicht benötigte Attribute entfernt. Der Datensatz besteht im Anschluss aus den Attributen AC POWER, was die erzeugte PV-Energie angibt, und der SOLAR IRRADIATION, welche die Sonneneinstrahlung angibt. Zudem gibt das Attribut DATETIME den Zeitstempel der Daten an. In einem nächsten Schritt werden die beiden Datensätze bereinigt und zu einem Datensatz zusammengeführt. Nach den beschriebenen Vorverarbeitungsschritten gibt die Funktion den bereinigten und zusammengeführten Datensatz für die weitere Analyse zurück.

2) Training des Machine Learning Models

Die Funktion *model_training()* dient dem Training des Machine Learning Modells und könnte z. B. einmal pro Tag auf dem Server mit den tagesaktuellen Daten ausgeführt werden. Somit kann täglich ein aktualisiertes Modell erstellt werden, welches später für die Vorhersage der PV-Energie genutzt wird. Durch den iterativen Aufbau des Trainingsprozesses wird das Modell über die Zeit immer genauer, da es täglich auf den neuen Solar- und Wetterdaten trainiert wird.

In der Funktion wird zunächst die oben beschriebene *preprocessing()*-Funktion aufgerufen, um die Daten für das Training vorzubereiten. Nach der Vorbereitung der Daten werden die relevanten Merkmale (AC_POWER und SOLAR IRRADIATION) für das Modell ausgewählt und eine sklearn-Pipeline zur Transformation und Vorverarbeitung der Daten eingerichtet.

Die Pipeline umfasst einen numerischen Transformator, der auf das Merkmal "IRRADIATION" angewendet wird. Dieser Transformator ist für die Skalierung der Merkmalswerte zwischen 0 und 1 mithilfe des *MinMaxScalers* verantwortlich. Zudem wird für das Training des Modells der beliebte Regressions-Algorithmus XGBoost ausgewählt und der Pipeline als Regressor hinzugefügt. Die Hyperparameter des Algorithmus werden basierend auf

mehreren Versuchen eingestellt. Diese sind in diesem Anwendungsfall allerdings weniger bedeutend, da der XGBoost mit unterschiedlichen Einstellungen auf dem beispielhaften Datensatz zu Genauigkeitswerten von bis zu 98 % kommt.

Sobald die Pipeline und der Algorithmus eingerichtet sind, trainiert die Funktion das Modell mithilfe der Pipeline und speichert das trainierte Modell *solar_model.sav* auf dem Server und wird in Zukunft für die Solarenergievorhersagen verwendet. Schließlich gibt die Funktion das trainierte Modell zurück.

3) Vorhersage der PV-Energie

In der Funktion *model_prediction()* erfolgt die Vorhersage der PV-Energie, welche in Zukunft erzeugt wird. Diese Funktion kann in der Optimierungsfunktion in *optimization.py* aufgerufen werden, um die Vorhersage basierend auf der Vorhersage der Sonneneinstrahlung durchzuführen. Hierzu müsste die kostenpflichtige Solar Radiation-API von OpenWeatherMap angesprochen werden, welche diese Wetterdaten enthält. Die Vorhersage erfolgt auf Grundlage des tagesaktuellen Modells, welches in der obigen Funktion *model_training()* täglich erstellt und auf dem Server gespeichert wird. Die Funktion *model_prediction()* wird mit den Wetterdaten für die nächsten 30 Tage aufgerufen. Zunächst wird das ML-Modell vom Server geladen und darauf eine Vorhersage basierend auf den übergebenen Wetterdaten durchgeführt. Die vorhergesagten Werte der PV-Energie werden als neue Spalte dem übergebenen Datensatz hinzugefügt. Dieser Datensatz wird von der Funktion zurückgegeben und kann im Anschluss in der Optimierungsfunktion weiterverwendet werden.

Allgemein lässt sich festhalten, dass die Vorhersage der Solarenergie in Abhängigkeit von der Sonneneinstrahlung in dem beispielhaft verwendeten Datensatz auf eine Genauigkeit von 98 % kommt. Dies begründet sich vermutlich in der starken Korrelation zwischen der Sonneneinstrahlung mit der erzeugten PV-Energie. In den Werten der Solar Radation sind bereits die Sonnenentwicklung über den Jahresverlauf sowie die tagesaktuelle Bewölkung enthalten. Daher ist der Zusammenhang zwischen diesen beiden Variablen für Machine Learning Algorithmen gut zu erlernen. Sobald die PV-Anlage der SEHO installiert ist und Daten erzeugt werden, bietet der Einsatz von Machine Learning Verfahren eine noch präzisere Möglichkeit zur Vorhersage der zu erwartenden PV-Energie.

5.3.4 Kalender-API (Microsoft Graph)

Wie bereits im Grundlagenkapitel zum Thema Schnittstellen aufgegriffen, werden für die optimierte, energiebasierte Terminplanung zwei externe Komponenten benötigt. Mit der Wetter-API von OpenWeatherMap ist bereits eine der beiden externen APIs in einem vorigen Kapitel vorgestellt worden. Der zweite Teil ist die Anbindung einer Kalender-API zum Lesen und Schreiben von Terminen. Die Herausforderung besteht hierbei, dass auf mehrere Kalender

gleichzeitig zugegriffen werden muss und von jedem Kalender Daten in einem strukturierten Format wie JSON angefordert werden müssen. Während einer Anforderungsaufnahme mit der Firma SEHO Systems GmbH, einem Stakeholder des Projekts, am 16.12.2022 hat sich herausgestellt, dass das Unternehmen plant, in naher Zukunft auf eine Microsoft 365-Umgebung zu setzen und diese unternehmensweit zu etablieren. Aufgrund des in Microsoft 365 enthaltenen Outlook-Kalenders hat sich eine weitere Toolevaluation als obsolet herausgestellt.

Microsoft 365 und auch der darin enthaltene Kalender nutzen Microsoft Graph als Gateway, um Daten in einem strukturierten Format in die Microsoft 365 Plattform zu importieren und exportieren (<https://learn.microsoft.com/de-de/graph/overview>). Voraussetzung für die Nutzung der Graph-API ist ein M365-Mandant. Im Zuge der Entwicklung wird ein kostenloser M365-Mandant, bereitgestellt durch das Microsoft 365 Developer Program (<https://developer.microsoft.com/de-de/microsoft-365/dev-program>) genutzt. Nach Aufsetzen des Mandanten erfolgt im Azure-Portal (<https://portal.azure.com/>) eine neue App-Registrierung für den Kalenderzugriff. Nach dem Erstellen des geheimen Client-Schlüssels müssen für die Applikation noch anwendungsübergreifende Berechtigungen für die Aktionen „User.Read.All“ und „Calendars.ReadWrite“ gesetzt und durch einen Administrator freigegeben werden. Mithilfe des zuvor erzeugten ClientSecret können nun kalender- und nutzerbezogene Anfragen an den M365-Mandanten gestellt werden.

Die Grundlage für die Nutzung der Graph-API innerhalb der Flask-Applikation werden in den Dateien `/app/graph_settings.json` und in der Funktion `get_graph_params(root_path)` innerhalb der Datei `/app/helper.py` bereitgestellt. In der `graph_settings.json` sind die ClientID, die TenantID und das ClientSecret aus dem Azure-Portal hinterlegt. Weiterhin ist ein Bearer-Token mitsamt Ablaufdatum zur Authentifizierung hinterlegt. Das Ablaufdatum beträgt eine Stunde ausgehend vom Zeitpunkt der Erstellung des Tokens. Die Funktion `get_graph_params(root_path)` hat den Zweck, diesen Bearer-Token auf Gültigkeit zu prüfen und gegebenenfalls einen neuen Token zu generieren. Falls kein gültiger Token vorhanden ist, wird eine POST-Request an den Endpunkt <https://login.microsoftonline.com/TenantID/oauth2/v2.0/token> gesendet. Mit der Anfrage werden außerdem die Header und Body Elemente aus Tabelle 8 gesendet.

Header	
Name	Wert
Host	login.microsoftonline.com
Content-Type	application/x-www-form-urlencoded
Body	
Name	Wert

client_id	<i>ClientID</i> (Aus Azure-Portal)
scope	https://graph.microsoft.com/.default
client_secret	<i>ClientSecret</i> (Aus Azure-Portal)
grant_type	client_credentials

Tabelle 8: Parameter zum Erstellen eines Graph-Bearer Tokens

Die Antwort im JSON-Format enthält einen Parameter Namens „token“, welcher in der *graph_settings.json* hinterlegt wird. Weiterhin wird das Ablaufdatum hinterlegt. Dieses berechnet sich aus der aktuellen Uhrzeit addiert mit dem Wert „expires_in“ in Sekunden, welcher ebenfalls in der Antwort enthalten ist. Dabei handelt es sich im Regelfall um exakt eine Stunde (3600 Sekunden).

Die Graph-API trägt an mehreren Stellen ihren Beitrag zur energiebasierten Terminplanung bei. Den Anfang macht das Optimierungsformular in der Route */optimization*. Die auswählbaren Maschinen und Mitarbeiter werden direkt aus dem M365-Mandanten gezogen. Dazu wird der GET-Endpunkt <https://graph.microsoft.com/v1.0/users/> mit dem Bearer-Token der Funktion *get_graph_params(root_path)* angesprochen. Dieser Endpunkt liefert eine Liste aller Nutzer des Mandanten im JSON-Format. Tabelle 9 enthält die Attribute jedes Nutzers, die, falls es sich um einen Maschinenaccount handelt, anders belegt sind als bei Standardnutzern.

Attribut	Ausprägung, falls Maschine
Vorname	ID der Maschine, z.B. 3x4
Displayname	Name der Maschine
Position	pjs_machine

Tabelle 9: Übersicht der speziellen Attribute eines Maschinenbenutzers in M365

Wie der Tabelle 9 zu entnehmen ist, erkennt die Flask-Applikation anhand des Feldes „Position“, ob es sich um einen Maschinenaccount oder einen regulären Mitarbeiteraccount handelt. Zur Vorbereitung der Optimierung wird erneut der oben genannte GET-Endpunkt angesprochen, um die IDs der M365-Accounts abzuspeichern, die im Optimierungsformular ausgewählt worden sind. Für jede(n) in der Optimierungsanfrage enthaltene(n) Maschine oder Mitarbeiter wird weiterhin dessen Kalender im Zeitraum der Optimierungsanfrage ausgelesen.

Dazu wird der GET-Endpunkt [https://graph.microsoft.com/v1.0/users/MaschinenIDoderMitarbeiterID/calendarView?startDateTime=FrühesterTermin&endDateTime=SpätesterTermin&\\$select=start,end](https://graph.microsoft.com/v1.0/users/MaschinenIDoderMitarbeiterID/calendarView?startDateTime=FrühesterTermin&endDateTime=SpätesterTermin&$select=start,end) mit dem Bearer-Token der Funktion *get_graph_params(root_path)* angesprochen. Nähere Informationen, wie diese Daten genutzt werden, enthält das Kapitel zur Terminoptimierung.

Nach der Terminoptimierung kann einer der drei Terminvorschläge ausgewählt werden. Hier wird die Graph-API genutzt, um neue Termine in den Kalendern der beteiligten Accounts anzulegen. Für jeden der beteiligten M365-Accounts wird der POST-Endpunkt <https://graph.microsoft.com/v1.0/users/KalenderID/calendar/events> mit dem Bearer-Token der Funktion *get_graph_params(root_path)* als Header und dem in Tabelle 10 enthaltenen JSON-Format als Body übergeben.

```
payload = {
    "subject": termin_db['bezeichnung'],
    "body": {
        "contentType": "HTML",
        "content": "Dieser Termin wurde von https://ebt-pjs.de generiert und blockiert die Maschine / den Mitarbeiter für einen Kundentermin"
    },
    "start": {
        "dateTime": graph_start_dt,
        "timeZone": "Europe/Berlin"
    },
    "end": {
        "dateTime": graph_end_dt,
        "timeZone": "Europe/Berlin"
    },
    "location": {
        "displayName": f"Seho Systems GmbH Showroom, Maschinen: {termin_db['maschinen_string']}"
    }
}
```

Tabelle 10: JSON-Format zum Anlegen eines Kalenderevents über Microsoft Graph

Neben dem Speichern im M365-Kalender der einzelnen Nutzeraccounts ist es auch möglich, die Termineinladung mittels eines konfigurierbaren Emailpostfachs an externe Empfänger zu senden oder direkt als ICS-Datei herunterzuladen. Dazu wird die Funktion *create_file_object(start, end, summary)* der Datei */app/helper.py*, welche auch durch die Route */return-files* erreichbar ist, verwendet. Mittels Nutzung der Python-Bibliothek *iCalendar* wird ein Kalenderobjekt erstellt, als ICS-Datei formatiert und in einem Byte-Array der E-Mail übergeben oder direkt zum Download angeboten.

5.3.5 OPC UA

OPC UA wird in der Applikation genutzt, um Maschinen hochzufahren, wenn ein geplanter Besuchstermin bevorsteht und sie im Anschluss wieder herunterzufahren. Dabei wird ein Masschinen-Offset berücksichtigt. Das ist die Vorlaufzeit, die eine Maschine braucht, um hochzufahren. Der Aufruf muss über die API der Applikation erfolgen, welche im Kapitel 5.5 beschrieben wird.

Die technische Implementierung der OPC UA Schnittstelle erfolgt in der Funktion *opc_ua_sender()*, die in der Python-Datei *helper.py* zu finden ist. Die Funktion sendet per OPC

UA abhängig von dem Zustand und dem Zeitpunkt Befehle zum Starten/Stoppen von Maschinen.

Die Funktion empfängt die folgenden Parameter:

- „*machineIDs*“: Eine Liste von Maschinen-IDs, zu denen eine Verbindung aufgebaut und Befehle versendet werden sollen.
- „*state*“: Der gewünschte Zustand, in denen die Maschinen gesetzt werden sollen. Zulässig sind die Werte „on“ und „off“.
- „*root_path*“: Der Pfad der Konfigurationsdatei, aus der die Einstellungs-Werte der OPC UA-Verbindung gelesen werden sollen.
- „*terminDateTime*“: Der gewünschte Zeitpunkt, an dem die Maschinen gestartet/gestoppt werden sollen.

Die Funktion liest zuerst die Konfigurationseinstellungen aus der durch *root_path* angegebenen Datei, einschließlich der OPC UA Server-URLs, Variablennamen und Offsets für jede Maschine. Sie bestimmt dann den Typ jeder Maschine anhand der Maschinen-ID und ruft die entsprechenden Konfigurationseinstellungen ab.

Als Nächstes berechnet die Funktion die Zeitdifferenz zwischen der aktuellen Zeit und der angegebenen Startdatum des Terms *terminDateTime* für jede Maschine. Dabei wird der Maschinen-Offset berücksichtigt. Wenn die Zeitdifferenz kleiner oder gleich null ist, stellt die Funktion eine Verbindung zu dem in der Konfiguration angegebenen OPC UA Server her. Dort durchsucht die Funktion den Adressraum, um den zur gewünschten Maschine zugehörigen Knoten zu finden. Ist das erfolgt, sendet die Funktion den gewünschten Zustand der Maschine an den Knoten, um die Maschine zu starten oder anzuhalten. Wenn die Verbindung und der Schreibvorgang erfolgreich war, fügt die Funktion eine Erfolgsnachricht an die *return_notification*-Liste an, die den Status der Maschine angibt. Schlägt die Verbindung fehl, fügt die Funktion eine Fehlermeldung an die Liste *return_notification* an. Ist die Zeitdifferenz größer als null, fügt die Funktion der Liste *return_notification* einen Benachrichtigungsstring hinzu, der angibt, dass es zu früh ist, die Maschine zu starten. Schließlich gibt die Funktion die Liste *return_notification* zurück, die die Status- und Fehlermeldungen für jede Maschine enthält.

5.4 Terminoptimierung

Die Hauptfunktionalität des SolarSync Systems besteht in der energiebasierten Planung von Terminen. Das Ziel der Terminoptimierung ist, den Stromverbrauch der Termine möglichst durch die PV-Anlage der SEHO zu decken und somit den Netzbezug zu minimieren. Dieses Kapitel dient der ausführlichen Erläuterung der Funktionsweise der Terminoptimierung, einschließlich des mathematischen Optimierungsmodells.

Die Funktionalität der Optimierung wird in der Datei *optimization.py* umgesetzt. Der Einstieg in die Optimierung läuft über die Flask-Route */optimization*. Die dazugehörige Funktion nimmt die Nutzereingaben, welche über zwei Flask-Forms bereitgestellt werden, entgegen und ruft die Validierung der Eingaben auf. Diese Validierung ist in der *forms.py* implementiert. Das erste Formular *TerminOptimizationForm* dient der Eingabe der Terminbeschreibung, der benötigten Maschinen und Mitarbeiter, der Lötkomplexität sowie der Dauer des Termins. Dabei sind alle Eingabefelder als Pflichtfelder charakterisiert. Mit einer Validierungsfunktion werden folgende Anforderungen überprüft:

- Es muss mindestens eine Maschine sowie ein Mitarbeiter ausgewählt werden.
- Die Dauer des Termins muss mindestens eine Stunde sein, darf nur ganze Stunden betragen und muss im INT-Format angegeben werden.

Falls eine dieser Anforderungen nicht erfüllt ist, wird eine passende Fehlermeldung als Pop-Up ausgegeben. Das zweite Formular *OptimizationForm* nimmt das Start- und Enddatum des Planungshorizonts entgegen. Zudem werden die Termindaten aus dem *TerminOptimizationForm* übergeben. Das Formular enthält ein Submit-Field, welches dem Aufruf der Optimierung mit den Start- und Enddaten sowie den Termindaten dient. Für dieses zweite Formular ist ebenfalls eine Validierungsfunktion implementiert. Diese prüft folgende Anforderungen:

- Es darf nur ein Termin geplant werden.
- Das Enddatum liegt hinter dem Startdatum.
- Das Startdatum darf nicht in der Vergangenheit liegen.
- Der Planungshorizont umfasst maximal 30 Tage vom aktuellen Tag aus.

Falls eine dieser Anforderungen nicht erfüllt ist, wird ebenfalls eine passende Fehlermeldung als Pop-Up ausgegeben. Nach erfolgreicher Validierung der Eingabedaten wird die eigentliche Optimierungsfunktion *optimization_table()* aufgerufen. Dabei werden das Start- und Enddatum des Planungshorizontes sowie die weiteren Nutzereingaben von dem zu planenden Termin übergeben. Die Funktionsvariable *api* muss im Fall eines Aufrufes über die eigens erstellte API, welche in Kapitel 5.5 vorgestellt wird, auf True gesetzt werden. Zudem müsste in diesem Fall ein Session-Token zur Authentifizierung übergeben werden.

In der Funktion *optimization_table()* erfolgt die Terminoptimierung. Hierzu werden zunächst die benötigten Inputdaten aus den unterschiedlichen Quellen abgerufen und verarbeitet. Basierend auf diesen Daten wird das Optimierungsmodell in Gurobi formuliert und anschließend gelöst. Die Outputdaten der Optimierung werden so verarbeitet, dass diese zum einen dem Nutzer angezeigt und zum anderen in einer Datenbank für spätere Analysen gespeichert werden können. In Abbildung 26 ist eine Übersicht der Datenströme des Optimierungsmodells grafisch dargestellt.

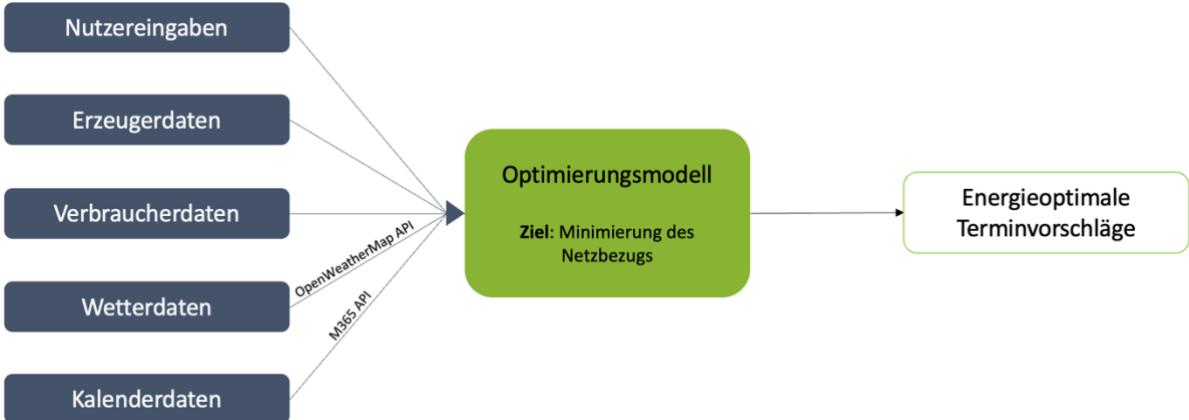


Abbildung 26: Grafische Darstellung der Optimierungsfunktion

Wie bereits beschrieben, teilt sich die Optimierungsfunktion in die drei Teilebereiche der Datenvorverarbeitung, der Optimierung sowie der Weiterverarbeitung des Outputs. In den folgenden Kapiteln wird der Ablauf in diesen Bereichen genauer beschrieben, wobei die obige Abbildung hierbei als Orientierung dient.

5.4.1 Verarbeitung der Input-Daten

Der erste Schritt der Optimierungsfunktion `optimization_table()` ist der Abruf und die Bereinigung aller für die Optimierung benötigten Daten. Diese Daten unterteilen sich in die fünf Gruppen Nutzereingaben, Erzeuger-/Wetterdaten, Verbraucher- und Kalenderdaten. Die Verarbeitung dieser Daten wird im Folgenden beschrieben.

1) Erzeuger- und Wetterdaten

Zu Beginn werden die Solardaten aus einer CSV-Datei eingelesen. Die Erzeugung dieses Datensatzes wurde bereits in Kapitel 5.3.3 erläutert. Außerdem erfolgt der Abruf der Wetterdaten der nächsten 30 Tage von dem Streaming Server im JSON-Format. Aus diesen Daten wird der tägliche Bewölkungsgrad „clouds“ sowie der Zeitstempel „dt“ in einem Dictionary gespeichert. Dieses wird in ein *pandas*-Dataframe umgewandelt. Da die Bewölkungsdaten nur tageweise vorliegen, werden diese auf stündliche Werte interpoliert. Die Bewölkungsdaten werden mit den Solardaten in einem Dataframe *df* verknüpft und auf den angegebenen Planungshorizont gefiltert. Im Anschluss wird die Vorhersage der PV-Energie durchgeführt. Das Vorgehen hierzu wurde ebenfalls in Kapitel 5.3.3 beschrieben. Die vorhergesagte PV-Energie wird in einer neuen Spalte des Dataframe *df* gespeichert.

2) Verbraucherdaten

Nun wird der Stromverbrauch von bereits geplanten Terminen im Planungshorizont abgerufen. Hierzu werden alle Termine, welche im Planungshorizont geplant sind, aus der Tabelle *Termin* abgefragt. Die stündlichen Stromverbrauchsdaten der Termine werden in einem neuen

Dataframe gespeichert. Falls keine zukünftigen Termine geplant sind, werden Dummy-Daten mit einem Energieverbrauch von null in das Dataframe eingefügt. Dadurch wird sichergestellt, dass Termindaten vorhanden sind. Diese werden mit anderen Energiedaten zusammengeführt. Im nächsten Schritt werden die Termindaten nach der „dateTime“ gruppiert und die resultierenden Energieverbrauchswerte summiert. Danach werden die Termindaten mit dem Dataframe der Vorhersage der PV-Energie zusammengeführt. Zudem wird die eingestellte Grundlast aus der Konfigurationsdatei abgerufen. Durch Subtraktion der vorhergesagten PV-Energie von der Summe der Grundlast und der Terminverbräuche wird der Netzbezug im Planungshorizont auf stündlicher Basis berechnet.

3) Nutzereingaben / Termindaten

Zur Verarbeitung der Nutzereingaben wird ein neues Dictionary initialisiert und mit den Daten des Termin-Formulars gefüllt. Im Anschluss wird das Dictionary in ein Dataframe mit den Attributen „Bezeichnung“, „Dauer“, „Maschinen“, „Komplexität“, „Mitarbeiter“, „Energieverbrauch“ und „Komplexität_Prozent“ umgewandelt. Dieses Dataframe enthält alle Eingaben, die der Nutzer zu den Eigenschaften des zu planenden Terms angegeben hat.

Da in dem Attribut „Maschinen“ nur die ID's der benötigten Maschinen enthalten sind, erfolgt ein API-Call, welcher die Outlook-Profile aller Maschinen abfragt. Über das Outlook-Profil wird der Name der Maschine abgerufen und in einem String gespeichert. Dieser dient der Darstellung der vollständigen Maschinenbezeichnung und nicht nur der ID. Das gleiche Vorgehen wird analog für die Mitarbeiter durchgeführt, da in dem Attribut „Mitarbeiter“ ebenfalls nur die ID's enthalten sind. Des Weiteren wird über den API-Call der Maschinentyp der benötigten Maschinen abgefragt, welcher für die Berechnung des Energieverbrauchs relevant ist.

Die Basis des Stromverbrauchs, welcher durch die Termine verursacht wird, ist der Stromverbrauch der Maschinen des Terms. Dieser unterteilt sich in die Phase des eigentlichen Betriebs sowie in die Aufheizphase. Die anpassbaren Verbrauchsdaten werden für alle Maschinentypen für beide Phasen aus der Konfigurationsdatei ausgelesen und in einem Dictionary zwischengespeichert. Zudem wird, basierend auf der durch den Nutzer eingestellten Lötkomplexität sowie der Energieverbrauch des zu planenden Lötorganges angepasst. Hierzu werden die benötigten Verbrauchsdaten ebenfalls aus der Konfigurationsdatei ausgelesen. Anschließend wird der Energieverbrauch für den Termin auf Grundlage der Lötkomplexität berechnet. Hierzu wird die Liste der benötigten Maschinen inklusive der Maschinentypen durchlaufen. Für jede Maschine wird der Energieverbrauch mit der Lötkomplexität und der Dauer des Terms multipliziert sowie das Ergebnis zu dem Gesamtenergieverbrauch des Terms addiert. Abschließend wird der errechnete Energieverbrauch dem Dataframe mit allen weiteren Termindaten hinzugefügt.

4) Kalenderdaten

Ein wichtiger Teil für das Optimierungsproblem sind die Verfügbarkeiten der Maschinen sowie der Mitarbeiter. Die Kalenderdaten der benötigten Maschinen und Mitarbeiter werden jeweils über die Microsoft Graph-API abgerufen. Hierbei erfolgt die Abfrage nur für den eingestellten Planungshorizont. Aus der Antwort, welche im JSON-Format übergeben wird, werden die Start- und Enddaten der bereits geplanten Termine der Maschinen bzw. Mitarbeiter extrahiert und in einer Liste gespeichert. Schließlich wird ein Dictionary erstellt, in welchem alle geplanten Termine im Planungshorizont der benötigten Maschinen und Mitarbeiter enthalten sind. Hierbei wird die ID der Maschinen bzw. Mitarbeiter als Schlüssel verwendet.

5.4.2 Optimierungsmodell

Nachdem im vorigen Kapitel die Verarbeitung der Inputdaten beschrieben wurde, wird in diesem Kapitel die Formulierung des Optimierungsproblems erläutert. Hierzu werden zunächst die benötigten Variablen erläutert und das Optimierungsproblem mathematisch aufgestellt. Im Anschluss folgt eine genaue Erklärung der Zielfunktion sowie der Nebenbedingungen des Problems. Nachfolgend findet sich eine Übersicht der Variablen des Optimierungsproblems.

Tabelle 11: Variablen des Optimierungsproblems

Name der Variable	Beschreibung
$consumption_{dt,t}$	Netzbezug des Termins t an dem Datum dt
$consumptionWithoutHeating_{dt,t}$	Netzbezug des Termins t an dem Datum dt ohne Berücksichtigung der Aufheizenergie
$start_{dt,t}$	Binäre Variable, die angibt, ob der Termin t an dem Datum dt beginnt.
$startHour_t$	Numerische Variable, die die Startstunde des Termins t speichert.
$startDay_t$	Numerische Variable, die den Starttag des Termins t speichert.
$endHour_t$	Numerische Variable, die die Endstunde des Termins t speichert.

Die Variable $consumption$ wird genutzt, um den resultierenden Netzbezug für jeden möglichen Startzeitpunkt des Termins zu berechnen. Dieser ist für die spätere Minimierung des Netzbezuges relevant. Eine genaue Beschreibung dieser Berechnung folgt in dem nachfolgenden Kapitel 5.4.3. Die Variable $consumptionWithoutHeating$ berechnet ebenfalls den möglichen Netzbezug des Termins, berücksichtigt allerdings nicht die etwaige Aufheizphase der Maschinen und dem daraus resultierenden Energieverbrauch. Diese Variable wird für die spätere Analyse und Darstellung des Ergebnisses des Optimierungsmodells genutzt. Die binäre $start$ -Variable ist die Entscheidungsvariable des Optimierungsproblems, welche in der Zielfunktion zum Einsatz kommt. Die weiteren numerischen Variablen dienen als Grundlage für die Nebenbedingungen sowie für die Darstellung des Ergebnisses. Generell dient der Index t der Erstellung mehrerer Terminvorschläge, in diesem konkreten Anwendungsfall werden drei Vorschläge erstellt, für den zu planenden Termin.

Aus den obigen Variablen ergibt sich folgende Zielfunktion:

$$\min \sum_{dt \in D} \sum_{t \in T} consumption_{dt,t} * start_{dt,t}$$

Die Zielfunktion minimiert den Netzbezug des Terms, wodurch der Anteil der PV-Energie des Terms maximiert wird. Hierbei wird mit den beiden Summenfunktion für jeden möglichen Startzeitpunkt des Terms sowie für alle möglichen Terminvorschläge der resultierende Netzbezug berücksichtigt. Mit der Binärvariable $start$ wird mit einem Wert von 0 oder 1 angegeben, ob der Terminvorschlag zu diesem Startzeitpunkt geplant wird.

Die Zielfunktion gilt unter folgenden Nebenbedingungen:

- 1) Planung nur an Wochentagen:

$$\forall dt \in D, t \in T, \text{ wenn ein Wochenende ist, dann } start_{dt,t} = 0$$

- 2) Keine Überschneidung:

$$start_{dt,t} = 1 \rightarrow \sum_{dt}^{dt+length(t)} \sum_{t2=1}^T start_{d+length(t), t2}$$

- 3) Start- und Endzeiten:

$$\forall t \in T: startHour_t \geq 8$$

$$\forall t \in T: endHour_t \leq 18$$

- 4) Verfügbarkeiten der Maschinen:

Wenn die Maschine m für Termin t benötigt wird, dann gilt:

$$(dt > startTime_m) \wedge (dt < endTime_m - duration) \\ \rightarrow \left(\forall i \in [0, duration], \sum_{t2 \in T} start_{dt+i, t2} = 0 \right)$$

Wobei $startTime_m$ und $endTime_m$ die Start- und Endzeiten der bereits geplanten Termine auf der Maschine m und die $duration$ die Dauer des Terms ist.

- 5) Verfügbarkeiten der Mitarbeiter:

Wenn der Mitarbeiter e für Termin t benötigt wird, dann gilt:

$$(dt > startTime_e) \wedge (dt < endTime_e - duration) \\ \rightarrow \left(\forall i \in [0, duration], \sum_{t2 \in T} start_{dt+i, t2} = 0 \right)$$

Wobei $startTime_e$ und $endTime_e$ die Start- und Endzeiten der bereits geplanten Termine des Mitarbeiters e und die $duration$ die Dauer des Terms ist.

Nachfolgend werden die mathematisch formulierten Nebenbedingungen genauer erläutert. Mit der ersten Nebenbedingung wird sichergestellt, dass Termine nur an Wochentagen geplant werden. Die zweite Nebenbedingung dient der Vermeidung von Überschneidungen der verschiedenen Terminvorschläge. In diesem Anwendungsfall erhält der Nutzer drei Terminvorschläge, wobei diese logischerweise zu unterschiedlichen Zeiten geplant sein sollen. Zudem wird das Zeitfenster der Termine an Wochentagen zwischen 8 und 18 Uhr gesetzt. In der vierten und fünften Nebenbedingung wird sichergestellt, dass keine Termine geplant werden, wenn die benötigten Maschinen bzw. Mitarbeiter durch andere Termine zu dieser Zeit nicht verfügbar sind.

5.4.3 Umsetzung des Optimierungsmodells in Gurobi

Das im vorigen Kapitel mathematisch formulierte Optimierungsproblem wird mithilfe der *GurobiPy*-Bibliothek in dem Programmcode integriert. *GurobiPy* ermöglicht die Modellierung mathematischer Optimierungsmodelle mit Gurobi in Python. Dieses Kapitel dient der Beschreibung der Umsetzung des Optimierungsproblems in *GurobiPy*.

Der gesamte Code zur Umsetzung des Optimierungsmodells wird über eine Schleife gesteuert, welche dazu dient, die Anzahl der Terminvorschläge anzupassen. Sollte es nicht möglich sein, die gewünschten drei Terminvorschläge in dem möglichen Zeitraum zu finden, wird die Anzahl der Terminvorschläge um 1 reduziert. Falls das Problem unlösbar ist und es keinen Terminvorschlag findet, wird der Optimierungsprozess beendet und eine Fehlermeldung ausgegeben. Des Weiteren sorgen *break*-Befehle dafür, dass die Schleife beendet wird, sobald eine mögliche Lösung gefunden wurde.

Innerhalb der Schleife wird das Gurobi-Modell initialisiert und die benötigten Variablen (siehe Tabelle 11) definiert. Im Anschluss wird der Netzbezug der Terminvorschläge berechnet und in der Variable *consumption* gespeichert. Hierzu wird über alle Terminvorschläge und möglichen Startzeiten iteriert. Der Netzbezug wird anhand des stündlichen Energieverbrauches des Terms und dem bereits vorhandenen Netzbezug in der jeweiligen Stunde berechnet. Zudem wird der mögliche Stromverbrauch der Aufheizphase der Maschinen in den Netzbezug miteinberechnet. Dieser entsteht, wenn die Maschine am gleichen Tag oder am Vortag des Terms nicht genutzt wurde, da die Maschine in diesem Fall erst hochgefahren werden muss. Die Überprüfung, ob die Maschine in naher Vergangenheit genutzt wurde, wird in der Funktion *machine_used_before()* implementiert.

Nachdem die *consumption*-Variable mit den Daten gefüllt wurde, folgt die Initialisierung der Zielfunktion des Modells. Zudem werden die Nebenbedingungen formuliert. Nach der Definition von Zielfunktion und Nebenbedingungen wird der Gurobi-Solver mit der Funktion *model.optimize()* aufgerufen und die Optimierung durchgeführt. Mit diesem Schritt ist die eigentliche Optimierung abgeschlossen. Im folgenden Kapitel wird erklärt, wie der Output von

Gurobi verarbeitet wird, um dem Nutzer die Terminvorschläge anzuzeigen bzw. eine Analyse der unterschiedlichen Vorschläge zu ermöglichen.

5.4.4 Verarbeitung des Outputs

Nachdem die Optimierung der Termine abgeschlossen ist, sollen die Ergebnisse dem Nutzer präsentiert werden. Hierbei werden die Terminvorschläge mit Datum und Uhrzeit ausgegeben, welche der Nutzer auswählen und per E-Mail verschicken kann. Zudem wird eine Analyse bereitgestellt, welche die verschiedenen Terminvorschläge hinsichtlich des Energieverbrauchs, Netzbezugs und der CO₂-Ersparnis vergleicht. Nachfolgend wird die Verarbeitung des Outputs von der Optimierung beschrieben.

Die gesamten Termindaten der Optimierung werden in einem neuen Dataframe gespeichert. Mit der Iteration über die Variablen des gelösten Optimierungsmodells werden zunächst die Startdaten (Datum und Uhrzeit) mithilfe des Gurobi-Befehls `getVars()` abgerufen und in dem Dataframe gespeichert. Da die Ausgabe der Gurobi-Variable in einem abweichenden Format funktioniert, erfolgen einige Schritte, um diese Daten zu transformieren. Das Datum und die Uhrzeit werden in separaten Spalten des Dataframes gespeichert. Zudem erhält jeder Terminvorschlag eine ID.

Im Anschluss wird für jeden Terminvorschlag im Dataframe der resultierende Netzbezug berechnet. Hierzu wird erneut auf eine Gurobi-Variable zugegriffen, welche die Daten des Netzbezugs enthält und zu Beginn der Optimierung mit diesen Daten befüllt wird. Falls ein negativer Netzbezug aus der Berechnung resultieren sollte, wird dieser auf 0 gesetzt. Ein negativer Netzbezug bedeutet eine Einspeisung von Strom in das Netz. Daraus lässt sich ableiten, dass in einer solchen Situation 100 % der benötigten Energie aus der PV-Anlage kommt. Abschließend wird das Dataframe basierend auf dem Netzbezug sortiert.

Das neu generierte Dataframe wird mit dem bestehenden Dataframe, welches die Nutzereingaben des Termsins enthält, zusammengeführt. Die Datumsangaben werden in das deutsche Format transformiert. Das Dataframe wird in ein Dictionary konvertiert und später dem Frontend übergeben. Im Anschluss wird für alle Terminvorschläge der Anteil der PV-Energie ermittelt. Hierfür wird zunächst der Energieverbrauch des Termsins aus dem Termindaten-Dataframe abgerufen. Basierend auf dem Netzbezug und dem Energieverbrauch der Terminvorschläge wird der PV-Anteil in Prozent berechnet. Dieser wird als neue Spalte in dem Dataframe gespeichert. Zudem wird die durch die Nutzung von Solarenergie eingesparte CO₂-Menge berechnet, wobei erneut der Faktor von 0,412 kg CO₂ pro kWh zugrunde gelegt wird.

Zuletzt erfolgt noch das bereits beschriebene Resampling der Daten der vorhergesagten PV-Energie auf Tage. Des Weiteren wird die Summe der zu erwartenden PV-Energie in dem Planungshorizont berechnet. Der Abschnitt des Codes zur Optimierung endet mit einem `break`.

Befehl, welcher die Schleife, in der die Optimierung sowie die Verarbeitung des Outputs abläuft, abbricht, sobald eine Lösung gefunden wurde.

Am Ende der Funktion `optimization_table()` wird die Möglichkeit der API-Nutzung integriert. Hierbei wird geprüft, ob die Optimierung über die Benutzeroberfläche oder über die eigene API aufgerufen wurde. Ist das Letztere der Fall, werden die relevanten Termindaten der Optimierung als Dictionary zurückgegeben. Falls die Optimierung über die Benutzeroberfläche initiiert wurde, werden die Termindaten in `session`-Variablen gespeichert und mit einem `redirect`-Befehl die Funktion `appointment_list()` aufgerufen, welche die Funktionalität der Ergebnisseite der Optimierung implementiert. Im nachfolgenden Kapitel werden diese und weitere Hilfsfunktionen, die für die Speicherung der Termine erforderlich sind, beschrieben.

5.4.5 Ausgabe und Speicherung von Terminen

Die App-Route `/appointments` dient einerseits der Ausgabe der Termindaten, welche im vorigen Kapitel erläutert wurden, an das Frontend. Andererseits erfolgt die Implementierung der Anforderung zum Versenden der Termindaten per E-Mail. Beide Anforderungen werden in der Funktion `appointment_list()` umgesetzt. Diese Funktion ruft zunächst die Termindaten ab, welche zuvor in Session-Variablen gespeichert wurden. Diese Daten werden am Ende der Funktion mit der `render_template()`-Funktion an die HTML-Seite `optimization_table.html` übergeben. Hier erfolgt die Darstellung der möglichen Terminvorschläge sowie die Visualisierung dieser. Der Nutzer hat dabei die Möglichkeit, den gewünschten Terminvorschlag auszuwählen und verbindlich zu „buchen“. Eine Buchung eines Terms erfolgt durch die Speicherung der Termindaten in der eigenen Datenbank des SolarSync-Systems und dem Eintragen des Terms in die Outlook-Kalender der beteiligten Mitarbeiter und Maschinen.

Um einen Termin zu buchen, stehen dem Nutzer zwei Möglichkeiten zur Verfügung: Zum einen kann die Speicherung des Terms durch den Download einer ICS-Datei mit den Termindaten angestoßen werden. Die Implementierung dieser Funktionalität erfolgt in der `return_files`-Route, welche bereits in Kapitel 5.3.4 vorgestellt wurde.

Zum anderen ist eine Mail-Funktion implementiert, die es dem Nutzer ermöglicht, den gewählten Termint per Mail an Kunden und Mitarbeiter zu versenden. Hierzu wird eine `SendMailForm()` integriert, welches eine E-Mail an die eingegebenen Empfänger mit dem vom Nutzer festgelegten Text erstellt. Die Absenderadresse ist diesem Fall termine@pjs-mail.de. Als Betreff wird die Terminbezeichnung übernommen. Zudem wird eine ICS-Datei mit den Termindaten (Datum, Uhrzeit, Ort, Bezeichnung, Absender) als Anhang der E-Mail hinzugefügt. Die Daten des Mailservers werden aus der Konfigurationsdatei geladen. Mithilfe der `smtplib`-Bibliothek wird die Verbindung zum SMTP-Server hergestellt. Sollte es bei dem

Versand der E-Mail zu Fehlern kommen, wird die *optimization_table.html* Seite aufgerufen und eine Fehlermeldung mithilfe der *flash()*-Funktion angezeigt.

Sobald die E-Mail erfolgreich versendet wurde, wird mithilfe der *flash()*-Funktion von Flask eine Erfolgsmeldung angezeigt. Zudem wird die Termin-ID abgerufen und die Funktion *save_to_calendar()* aufgerufen, um den Termin sowohl in der eigenen Datenbank zu speichern, als auch in die Outlook-Kalender der beteiligten Mitarbeiter und Maschinen einzutragen.

Die *save_to_calendar()*-Funktion wird in der Route *save-optimization* implementiert und mit der Termin-ID aufgerufen. Zudem wird festgelegt, ob der Aufruf über die API kommt oder über die Benutzeroberfläche des SolarSync-Systems. Zunächst wird überprüft, ob die Termindaten in den Session-Variablen korrekt hinterlegt sind. Ist dies nicht der Fall, wird entweder über die API eine *bad_request*-Meldung zurückgegeben oder dem Nutzer eine Fehlermeldung auf der Benutzeroberfläche angezeigt. Sind die Termindaten korrekt, werden diese in einem Dictionary gespeichert. In dem Dictionary befinden sich alle Terminvorschläge. Es wird mit der übergebenen ID der passende Termin gefiltert. Falls die Termin-ID nicht in dem Dictionary gefunden wird, erfolgt ebenfalls eine Fehlermeldung.

Ist der Termin vorhanden, wird auf der Grundlage des Datums sowie der Uhrzeit des Termins ein Datetime-Objekt erstellt. Anschließend wird ein neues Termin-Objekt mit den Informationen des Termins erstellt und der Datenbank mit den Befehlen *db.session.add()* und *db.session.commit()* hinzugefügt.

Der nächste Abschnitt der Funktion ermöglicht die der Speicherung des Termins in die Outlook-Kalender der Maschinen und Mitarbeiter. Hierfür wird für jeden Mitarbeiter und jede Maschine, welche an dem Termin beteiligt sind, ein Kalenderereignis erstellt. Über die Microsoft Graph-API wird auf die Kalender der jeweiligen Maschinen bzw. Mitarbeiter zugegriffen und das Kalenderereignis mit dem Betreff, den Start- und Endzeiten sowie Standortinformationen gespeichert. Schließlich wird der Nutzer auf die Startseite der Optimierung geleitet und eine Meldung über die erfolgreiche Speicherung des Termins ausgegeben. Wird die Funktion über die API aufgerufen, wird die Erfolgsmeldung als JSON-Objekt zurückgegeben.

5.5 API-Schnittstelle

Bisher ist das Backend nur darauf ausgelegt, über eine grafische Benutzeroberfläche angesteuert zu werden. Dadurch ist es nicht möglich, den Prozess der Terminoptimierung zu automatisieren und jede Planung muss händisch von einem Mitarbeiter mit Zugang zum Tool ausgelöst werden. Um an dieser Stelle Zukunftssicherheit, Automatisierbarkeit und Integrierbarkeit zu gewährleisten, wird in der Applikation eine eigene REST-API implementiert. Diese liegt in dem Verzeichnis `/app/api/` und wird als Blueprint mit dem Präfix „`/api`“ in die eigentliche Applikation integriert. In diesem Kapitel wird die gesamte Funktionalität der API und allen vorhandenen API-Calls beschrieben. Dazu zählen die Bereiche Authentifizierung, Nutzerverwaltung, Errorhandling, OPC UA und die eigentliche Terminoptimierung.

5.5.1 Authentifizierung

Die API unterstützt zwei Authentifizierungsmethoden. Neben der klassischen Authentifizierung mit Nutzernamen und Passwort, wie es auch über die Benutzeroberfläche der Fall ist, kommt eine tokenbasierte Methode hinzu. Die klassische Methode wird nur genutzt, um einen Bearer-Token zu generieren, welcher für eine Stunde gültig ist. Der Token wird gemeinsam mit dem Ablaufzeitpunkt in der Datenbank hinterlegt. Durch das Ansprechen des folgenden Endpunkts kann ein Token generiert werden (vgl. Tabelle 12).

POST	http(s)://www.tenant.domain/api/tokens
	Authentifizierung Username <username>
	Password <password>
Antwort	Body { "token": String }

Tabelle 12: Erstellen eines API-Tokens

Sollte die Nutzungsdauer unter einer Stunde betragen, kann der Token durch das Ansprechen eines DELETE-Endpunkts auch gelöscht werden. Die Formatierung des API-Aufrufs ist Tabelle 13 zu entnehmen.

DELETE	http(s)://www.tenant.domain/api/tokens
	Authentifizierung Token <token>
Antwort	204 NO CONTENT

Tabelle 13: Löschen eines API-Tokens

5.5.2 Nutzerverwaltung

Weiterhin können über die API Nutzeraccounts für die Software abgerufen und angelegt werden, damit dazu nicht die grafische Benutzeroberfläche verwendet werden muss. Um sich eine Übersicht aller Nutzeraccounts der Plattform anzuzeigen, wird der in Tabelle 14 gezeigte GET-Endpunkt genutzt.

GET	http(s)://www.tenant.domain/api/users	
	Authentifizierung	Token <token>
Antwort	Body	<pre>{ "users": [{ "id": Integer, "role": String, "username": String }] }</pre>

Tabelle 14: Liste aller Nutzer abfragen mittels API

Ist die ID eines Nutzers bereits bekannt, kann auch direkt ein entsprechender GET-Endpunkt angesprochen werden, um nur die Informationen eines Nutzers abzufragen (vgl. Tabelle 15).

GET	http(s)://www.tenant.domain/api/users/<id:integer>	
	Authentifizierung	Token <token>
Antwort	Body	<pre>{ "id": Integer, "role": String, "username": String }</pre>

Tabelle 15: Daten eines Nutzers abfragen mittels API

Abschließend kann mittels eines POST-Endpunkts auch ein neuer Nutzeraccount angelegt werden. In der zugehörigen Funktion werden Fehlermeldungen geworfen, wenn bereits ein Nutzer mit dem Namen existiert und wenn die vom Anwender zu übergebenden Parameter „Username“, „Role“ oder „Password“ nicht enthalten sind. Als Antwort wird der neu generierte Nutzer mitsamt ID ausgegeben. Tabelle 16 zeigt die Struktur des Aufrufs.

POST	http(s)://www.tenant.domain/api/users	
	Authentifizierung	Token <token>
	Body (JSON)	<pre>{ "username": String, "role": String, "password": String }</pre>

	Content-Type	application/json
Antwort	Body	{ "id": Integer, "role": String, "username": String }

Tabelle 16: Anlegen eines neuen Nutzers mittels API

5.5.3 Error Handling

Da nicht mit Sicherheit gewährleistet werden kann, dass immer das richtige Format ohne Fehler übergeben wird, muss auch für die API-Komponente ein Error-Handling implementiert werden. Dazu dient die Datei `/app/api/errors.py`. Die darin enthaltene Funktion `bad_request(message)` generiert einen Rückgabewert mit dem Fehlercode 400 und der übergebenen Nachricht im JSON-Format. Jede API-Route kann diese Funktion referenzieren, um individuelle Fehlernachrichten auszugeben.

5.5.4 OPC UA

Unternehmen, die die Flask-Applikation und OPC UA nutzen, sollen auch die Möglichkeit bekommen, automatisierte Maschinenstarts mithilfe der integrierten OPC UA Schnittstelle durchzuführen. Dazu wird der API-Endpunkt `opcua_start()` bereitgestellt. Im ersten Schritt werden alle von der Applikation generierten Termine durchsucht. Anschließend werden die Termine, die innerhalb der nächsten sieben Stunden starten, an eine Hilfsfunktion übergeben. Innerhalb dieser Funktion wird zuerst eine Verbindung zu den in den Einstellungen hinterlegten OPC UA-Servern aufgebaut. Wenn die Verbindung hergestellt werden kann, erhalten die Maschinen den Startbefehl. Die Struktur des API-Aufrufs kann Tabelle 17 entnommen werden.

POST	http(s)://www.tenant.domain/api/opcua_start	
	Authentifizierung	Token <token>
Antwort	Body	{ "Termine": [{ "Beschreibung": String, >Status": String }] }

Tabelle 17: Maschinen mit OPC UA via API starten

5.5.5 Terminoptimierung

Abschließend wird der wichtigste Teil der API, die Optimierung von Terminen, behandelt. Vor der eigentlichen Optimierung werden zwei GET-Aufrufe benötigt, um die IDs für Maschinen und Mitarbeiter offen zu legen. Diese werden darauffolgend der Optimierungsfunktion übergeben. Tabelle 18 dokumentiert, wie die IDs für Maschinen angefragt werden können.

GET	http(s)://www.tenant.domain/api/maschinen	
	Authentifizierung	Token <token>
Antwort	<pre>Body { "Maschinen": [{ "Maschinename": String, "MaschinenID": String }] }</pre>	

Tabelle 18: Abfragen von MaschinenIDs mittels API

Die Abfrage von MitarbeiterIDs hat dasselbe Schema und ist aufgrund der Vollständigkeit in Tabelle 19 abgebildet.

GET	http(s)://www.tenant.domain/api/mitarbeiter	
	Authentifizierung	Token <token>
Antwort	<pre>Body { "Mitarbeiter": [{ "Mitarbeitername": String, "MitarbeiterID": String }] }</pre>	

Tabelle 19: Abfrage von MitarbeiterIDs mittels API

Mit den aus diesen Abfragen gewonnenen Informationen kann nun eine Terminoptimierung durchgeführt werden. Hierzu wird der API-Endpunkt */optimization* angesprochen. Der Terminoptimierung wird im Body ein JSON mit Start- und Enddatum sowie einer Terminliste mit Maschinen, Mitarbeitern, Beschreibung und Termindauer übergeben. Die zugehörige Funktion der Flask-Applikation verfügt über eine komplexe Validierung, um die Präsenz von Feldern, die Schreibweise von Datumsfeldern und die Korrektheit der übergebenen Mitarbeiter- und MaschinenIDs zu überprüfen. Sind alle Daten vollständig und korrekt, wird wie über die grafische Benutzeroberfläche die Optimierung gestartet und die Ergebnisse im JSON-Format zurückgegeben. Der Unterschied besteht nur darin, dass die Termine nicht in der Flask-Session unter dem Keyword UserID, sondern unter dem Authentifizierungstoken zwischengespeichert werden. Tabelle 20 enthält die vollständige Dokumentation der POST-Anfrage.

POST	http(s)://www.tenant.domain/api/optimization
Authentifizierung	Token <token>
Body (JSON)	<pre>{ "startdate": String (YYYY-MM-DD), "enddate": String (YYYY-MM-DD), "termine": [{ "description": String, "duration": Integer, "machines": [String (MaschinenID)], "employees": [String (MitarbeiterID)] }] }</pre>
Content-Type	application/json
Antwort	<p>Body</p> <pre>{ "Optimierungszeitpunkt": String, "Termine": [{ "Date": String, "TerminID": Integer, "Termin_ID": Integer, "Time": String, "bezeichnung": String, "complexity": String, "complexity_percent": String, "dauer": Integer, "energieverbrauch": Float, "index": Integer, "machine_types": [String], "maschinen": [String], "maschinen_string": String, "mitarbeiter": [String], "mitarbeiter_string": String, "netzbezug": Float, "termin_id": Integer }] }</pre>

Tabelle 20: Terminoptimierung mittels API durchführen

Die Terminliste der Antwort enthält drei Terminvorschläge. Von diesen enthält jeder der Termine wiederum eine TerminID. Ebenso wie in der Benutzeroberfläche können auch mittels der API die Termine in der Datenbank und in den M365-Kalendern gespeichert werden. Dazu wird der GET-Endpunkt */save-appointment* angesprochen. Dieser speichert Termine nach dem Schema, welches auch in der Terminoptimierung verwendet wird. Tabelle 21 enthält die vollständige Dokumentation dieses Aufrufs.

GET	http(s)://www.tenant.domain/api/save-termin	
	Authentifizierung	Token <token>
	Parameter	id <TerminID>
Antwort	Body <pre>{ "Information": "Der Termin wurde gespeichert" }</pre>	

Tabelle 21: Speichern eines Termins durch die API

Damit ist die Dokumentation der API vollständig und alle grundlegenden Funktionen der Flask-Applikation sind auch außerhalb der Benutzeroberfläche zugänglich.

6 Diskussion

Das folgende Kapitel dient der Reflexion des Projektseminars aus Sicht des Projektteams. Hierbei wird auf die Herausforderungen eingegangen, die das Team bewältigen mussten. Zusätzlich werden Aspekte hervorgehoben, die wichtig für die erfolgreiche Projektdurchführung waren. Des Weiteren erfolgt eine Evaluation des SolarSync-Systems basierend auf einer Umfrage der Projektstakeholder.

6.1 Projektmanagement

Die Umsetzung des Projektmanagements bildet die Grundlage für die erfolgreiche Durchführung von Projekten. Nachfolgend werden zunächst die Herausforderungen beschrieben, die das Projektseminar mit sich brachte.

Die wohl größte Herausforderung bei der Entwicklung des SolarSync-Systems war der Trade-Off zwischen der Individualisierung der Software auf die Anforderungen der SEHO und der Entwicklung einer universellen Lösung, die auch in anderen Unternehmen integriert werden kann. Im Projektverlauf wurde mehrmals über die Wünsche der SEHO und deren Umsetzung diskutiert. Hierbei konnten die Betreuer des Lehrstuhls Hilfestellungen leisten und dem Projektteam eine Leitlinie aufzeigen.

Da die Erstellung einer Software zur energiebasierten Terminoptimierung nur auf der Grundlage von validen Daten möglich ist, wurde zu Beginn des Projekts die Qualität der zur Verfügung gestellten Daten ermittelt. Da diese für die Umsetzung der Anforderungen nicht ausreichend war, wurden zunächst synthetische Daten für die Entwicklung modelliert. Hierdurch entstand das Risiko, aufwändige Änderungen der Software vornehmen zu müssen, falls die realen Daten sich erheblich von den modellierten Daten unterscheiden sollten. Daher wurde bei der Datenmodellierung viel Zeit und Aufwand in die Simulation bzw. Analyse der Solaranlage und den Verbrauchern der SEHO investiert.

Trotz der genannten Herausforderungen konnte das Projektteam das Projekt erfolgreich durchführen. Ausschlaggebend hierfür war die gute Kommunikation des Teams. Durch den regelmäßigen Austausch über WhatsApp konnten organisatorische Aufgaben oder zeitliche Abstimmung schnell kommuniziert werden. Daneben waren die wöchentlichen Meetings via Zoom wichtig für strategische Entscheidungen über die Softwareentwicklung sowie die Sprintplanung. Durch die Verwendung verschiedener Kommunikationstools konnten Arbeitspakete verständlich zugeteilt werden. Zudem hatte das Team immer eine Übersicht über den aktuellen Stand der Aufgaben.

Neben der internen Kommunikation gab es auch einen regelmäßigen Austausch mit den Projektstakeholdern. Sowohl das Betreuerteam als auch Dr. Reinhardt von der SEHO konnten

eine beratende Rolle einnehmen, die zum erfolgreichen Umsetzen des Projekts beigetragen hat. Besonders der Besuch vor Ort bei der SEHO wird hier hervorgehoben. Die Führung durch den Showroom und die Produktionshallen sowie die Möglichkeit des direkten Austauschs mit Dr. Reinhardt haben Klarheit gebracht. Zudem hat der Besuch das zuvor noch teilweise abstrakte Projekt greifbarer gemacht.

Ein weiterer wichtiger Aspekt der erfolgreichen Projektdurchführung war die Umsetzung des Projektmanagements. Zu Beginn des Projekts einigte sich das Team auf den Ratschlag der Betreuer hin, für die anstehende Aufgabe eine Projektmanagementmethode umzusetzen. Hierbei fiel die Wahl auf das agile Projektmanagement-Framework „Scrum“. Dadurch wurden intern verschiedene Rollen festgelegt, die nach den Stärken der Mitglieder verteilt wurden. Durch klare Verantwortlichkeiten konnte die Effizienz und Effektivität innerhalb des Teams gesteigert werden. Gleichzeitig wurden die unterschiedlichen Fähigkeiten gewinnbringend eingesetzt. Durch „Scrum“ wurde das Projekt in Sprints unterteilt. Diese halfen dem Projektteam bei der zeitlichen Koordination. Zu Beginn eines jeden Sprints wurden Aufgabenpakete verteilt, die über den Zeitraum des Sprints zu bearbeiten waren. Dies ermöglichte den einzelnen Teammitgliedern eine flexible Arbeitsweise und die Abhängigkeit untereinander wurde reduziert.

6.2 SolarSync-System

Zur Evaluation des entwickelten SolarSync-Systems wurde ein Fragebogen entwickelt, welcher an die Projektstakeholder versendet wurde. Die Umfrage dient der Ermittlung von Feedback zu der Softwarelösung sowie deren Funktionen. Auf Basis dieser Angaben können Anpassungen an der Software vorgenommen und ein Ausblick für weiteres Potenzial ausgearbeitet werden. Die Stakeholder sind Dr. Reinhardt (SEHO), Herr Fiedler (WUQM) und Herr Beyer (Axsol). Die Teilnehmer haben eine Präsentation der Anwendung erhalten, in der auch auf die Funktionalität sowie auf die zugrundeliegenden Mechanismen eingegangen wurde. Zusätzlich dazu wurden Screenshots der Anwendung bereitgestellt, welche die Beantwortung der Fragen unterstützen. Insgesamt haben zwei von drei der Befragten an der Umfrage teilgenommen. Nachfolgend werden die gestellten Fragen genannt, der Hintergrund der Frage erläutert und die Ergebnisse eingeordnet. Der Fragebogen enthält Fragen zur Funktion, den Integrationen, der Benutzeroberfläche sowie deren Nutzerfreundlichkeit, der Notwendigkeit einer Mobilversion und dem möglichen Potenzial für eine zukünftige Weiterentwicklungen der Software.

Auf Grund der Teilnehmerzahl und der gewählten Fragen wird von einer quantitativen Auswertung abgesehen und es wird nur eine rein qualitative Auswertung vorgenommen. Die Fragen sind in Tabelle 22 dargestellt.

1	Wie zufrieden sind Sie mit dem Umfang der Funktionen? Sind alle Anforderungen abgedeckt?
---	--

2	Gibt es weitere Funktionen, die Sie sich wünschen würden?
3	Benötigt es weitere Integrationen in andere Anwendungen neben Outlook, die fehlen?
4	Als wie intuitiv erachten Sie die Benutzeroberfläche?
5	Wie empfinden Sie die Nutzerfreundlichkeit?
6	Soll die Anwendung auch als Mobile-App angeboten werden?
7	Welchen Mehrwert sehen Sie in der Software im Unternehmenskontext?

Tabelle 22: Fragebogen zur Evaluation des SolarSync-Systems

Frage 1:

Es handelt sich um eine geschlossene Frage mit einer Likert-Skala, deren Bewertungsmöglichkeiten von „sehr unzufrieden“ bis „sehr zufrieden“ angegeben sind. Die Frage zielt darauf ab, zu evaluieren, ob alle Funktionen zur erfolgreichen Bewältigung der gestellten Anforderungen vorhanden sind. Die Befragten antworteten alle mit „sehr zufrieden“, woraus sich schließen lässt, dass alle benötigten Funktionen zum Erreichen des Geschäftszwecks vorhanden sind.

Frage 2:

Es handelt sich um eine halb offene dichotomische Frage. Die Antwortmöglichkeit „Ja“ wird durch ein Textfeld ergänzt, indem der Befragte Funktionen, die er zusätzlich implementiert haben möchte, nennen kann. Einer der Befragten nennt hier weitere Möglichkeiten, genauere Daten und eine bessere Übersicht der berücksichtigten Maschinen umzusetzen. Es wird gewünscht, dass man für die Maschinen weitere Parameter wie zum Beispiel Aufheizdauer oder den exakten Stromverbrauch festlegen kann. Im Projekt wurde davon abgesehen, da die verfügbare Datengrundlage keine Auskunft über solche Parameter zur Verfügung stellt. Die Anregung wird allerdings im Ausblick berücksichtigt.

Frage 3:

Es handelt sich um eine halb offene dichotomische Frage. Die Antwortmöglichkeit „Ja“ wird durch ein Textfeld ergänzt, indem die Befragten weitere Anwendungen nennen können, die in SolarSync implementiert werden könnten. Einer der Befragten forderte die Integration von qualitativ hochwertigen Wetterdaten. Auf Grund des finanziellen Aspekts war es dem Team allerdings nicht möglich, noch bessere Wetterdaten zu integrieren. Die Anregungen wurden in den Ausblick übernommen.

Frage 4:

Es handelt sich um eine geschlossene Frage mit einer Likert-Skala, deren Bewertungsmöglichkeiten von „sehr gut“ bis „sehr schlecht“ angegeben sind. Die Frage zielt darauf ab, zu evaluieren, wie intuitiv und gut die Benutzeroberfläche angesehen wird. Die Befragten antworteten hier mit „eher gut“ und „sehr gut“. Das Team schließt daraus, dass der Aufbau der Benutzeroberfläche verständlich ist und keine Anpassungen vorgenommen werden müssen.

Frage 5:

Es handelt sich um eine geschlossene Frage mit einer Likert-Skala, deren Bewertungsmöglichkeiten von „sehr gut“ bis „sehr schlecht“ angegeben sind. Die Frage zielt darauf ab, zu evaluieren, wie nutzerfreundlich SolarSync wahrgenommen wird. Die Befragten antworteten hier mit „eher gut“ und „sehr gut“. Das Team schließt daraus, dass die Nutzerfreundlichkeit hoch ist und keine Anpassungen vorgenommen werden müssen.

Frage 6:

Es handelt sich um eine geschlossene dichotomische Frage. Die Befragten können nur mit „Ja“ oder „Nein“ antworten. Die Ausprägungen sind hier gleichmäßig verteilt. Das Team nimmt zur Kenntnis, dass eine mobile Version des SolarSync-Systems gewünscht ist. Eine detailliertere Aufarbeitung folgt im Ausblick.

Frage 7:

Es handelt sich um eine offene Frage. Es gibt keine vorgegebenen Antwortmöglichkeiten. Die Befragten können hier freie Angaben zum möglichen Potenzial der Anwendung treffen. Diese Frage wird intensiv im Ausblick behandelt.

Es lässt sich zusammenfassen, dass die Befragten mit dem aktuellen Stand der Software zufrieden sind. In der weiteren Entwicklung bedarf es somit nur kleinerer Anpassungen und Erweiterungen, die im Ausblick diskutiert werden.

7 Schlussfolgerung

Das Ziel des Projektseminars zum Thema „Energiebasierte Terminplanung“ war die Konzeption und Implementierung einer Web-Applikation zur automatisierten Optimierung von Kundenterminen der SEHO Systems GmbH. Die SEHO Systems produziert energieintensive Lötmassen für die Industrie. In einem Showroom werden Schulungen und Versuche mit den Kunden der SEHO durchgeführt. Zudem werden im Zuge der Qualitätssicherung vor der Auslieferung der Maschinen testweise Lötprozesse auf diesen durchgeführt. Durch den hohen Stromverbrauch der Lötmassen ist das Unternehmen daran interessiert, eine Photovoltaikanlage zu installieren. Dies bringt zum einen Kostenvorteile mit sich, da weniger Energie aus dem Stromnetz bezogen werden muss. Zum anderen wird durch die Reduktion des Netzbzuges weniger CO₂ ausgestoßen. Neben der SEHO Systems GmbH war die Axsol GmbH als Anbieter für Energiespeichersysteme ein Projektpartner. Zudem war die WUQM Consulting GmbH an dem Projekt beteiligt, welches die SEHO in Themen rund um Umwelt- und Qualitätsmanagement unterstützt. Von Seiten der Universität Würzburg wurde das Projekt von Mitarbeitern des Lehrstuhls BWL und Wirtschaftsinformatik begleitet. Bei der Entwicklung der Webapplikation sollten folgenden Anforderungen umgesetzt werden:

- Entwicklung einer Web-Anwendung mit passendem Framework
- Integration von Schnittstellen zur Einbindung externer Daten (z. B. Wetter)
- Nutzung von mathematischen Optimierungstechniken zur Terminplanung
- Einsatz eines Streaming-Servers
- Integration von OPC UA
- Datavisualisierung mit geeignetem Framework
- Integration von Outlook-Kalendern zur Terminplanung

7.1 Projektzusammenfassung

Die Umsetzung erfolgte durch ein fünfköpfiges Projektteam, Master-Studierende der Universität Würzburg. Das Team nutzte dabei die agile Arbeitsweise Scrum. Meilensteine und die dazugehörigen Aufgaben wurden in zweiwöchigen Sprints geplant. Weekly Scrums dienten dem Austausch innerhalb des Projektteams zur Abstimmung der Wochenaufgaben und -ziele. Die Ergebnisse der abgeschlossenen Sprints wurden in zweiwöchigen Meetings den Projektbetreuern der Universität Würzburg vorgestellt. Durch den regelmäßigen internen und externen Austausch des Projektteams wurde eine hohe Produktivität der Sprints und somit eine schnelle Zielerreichung erreicht. In den Weekly Scrums konnten zudem strategischen Fragen und Entscheidungen über den weiteren Projektverlauf getroffen werden, was zu einem schnellen Entscheidungsprozess führte. Neben den wöchentlichen Meetings wurden Messengerdienste zur kurzfristigen und spontanen Abstimmung genutzt. Diese war

insbesondere in dem Entwicklungsprozess der Webapplikation nützlich, da so kurzfristige Probleme und Herausforderungen dezentral gelöst werden konnten.

Die gesamte Kommunikation in dem Projekt basierte auf verschiedenen Tools. Für die Weekly Scrums und den Meetings mit den Projektbetreuern wurde Zoom genutzt. Der kurzfristige Austausch von Textnachrichten erfolgte via WhatsApp. Für die gesamte Projektplanung kam Jira zum Einsatz. Zudem wurde Confluence zu Dokumentationszwecken sowie der Protokollierung aller externen Meetings genutzt. Zur Entwicklung der Webanwendung wurde hauptsächlich Visual Studio Code in Kombination mit GitHub genutzt.

Die technische Umsetzung des Projekts erfolgt durch die Entwicklung einer webbasierten Flask-Anwendung, welche auf den Namen SolarSync getauft wurde. Das Python-Framework Flask wurde aufgrund der schlanken Architektur sowie der Möglichkeit, das gesamte Backend und Frontend abzubilden, gewählt. Zudem verfügten mehrere Teammitglieder über Fähigkeiten in der Programmiersprache Python, was die Einarbeitungszeit deutlich verkürzte.

Den Kern des Backends bildet eine Flask-Applikation, welche auf einem Linux-Server gehostet wird. Der Zugriff auf die Webanwendung erfolgt über eine SSL-gesicherte Verbindung auf die Domain ebt-pjs.de. Das SolarSync-System ist mit einem Nutzermanagement ausgestattet. Vom Administrator erstellte Nutzer können sich über eine Login-Seite mit Nutzernamen und Passwort im System anmelden. Zur Integration einer SQL-Datenbank wird SQLAlchemy und Flask-Migrate genutzt. Zudem ist ein Kafka Streaming-Server integriert. Über eine API werden einmal täglich Wetterdaten von OpenWeatherMap für die nächsten 30 Tage abgefragt und auf dem Streaming-Server gespeichert. Zur Integration des Microsoft Outlook-Kalenders wird ein Entwickler-Tenant von M365 genutzt. Über die Microsoft Graph-API werden Kalenderdaten abgefragt und übergeben. Für die Terminoptimierung wurde ein mathematisches Optimierungsmodell in Gurobi entwickelt, welches das Ziel der Minimierung des Netzbezugs für den Stromverbrauch der Termine umsetzt. Das Modell basiert auf einem umfassenden Datenfundament aus einer Kombination von externen Daten (Wetter und Kalender), Nutzereingaben (Planungshorizont, Dauer, Maschinen, Mitarbeiter, Lötkomplexität) und Systemeinstellungen (Grundlast, Maschinenverbräuche im Betrieb und Aufheizprozess). Die Speicherung der gewählten Termine erfolgt in der SQL-Datenbank und in den Outlook-Kalendern der beteiligten Mitarbeiter und Maschinen. Über den Industriestandard OPC UA können die Maschinen gesteuert werden. Zudem ist ein Offset implementiert, der die unterschiedlichen Aufheizzeiten der Maschinen berücksichtigt. Über die gesamte Anwendung ist eine eigene API gespannt. Hiermit wird die Integration der Funktionalität des SolarSync-Systems in jede beliebige Software möglich. So ist beispielsweise die Einbettung in ein ERP-System denkbar. Des Weiteren ermöglicht ein Prototyp die Vorhersage der PV-Energie auf der Basis von Machine Learning. Dieser kann, sobald die PV-Anlage installiert wurde, in das Produktivsystem überführt werden.

Das Frontend des SolarSync-Systems wird mit HTML (Bootstrap), JavaScript, CSS und SCSS umgesetzt. Die Anwendung ist in die folgenden Seiten unterteilt. Auf der Startseite befindet sich ein Dashboard, welches einen Überblick über die nächsten Termine, die Wettervorhersage, die PV-Anlage sowie der CO₂-Einsparung gibt. Auf einer Detailseite zur PV-Anlage finden sich weitere Visualisierungen zu den Erzeugern und Verbrauchern, der gesamten CO₂-Einsparung sowie der Autarkie der Terminplanung. Die Datenvisualisierungen werden mit ChartJS implementiert. Auf einer Kalenderseite werden alle geplanten Termine und auf der Wetterseite die Wetterdaten der nächsten 30 Tage abgebildet. Unter dem Menüpunkt „Terminoptimierung“ ist die optimierte Planung von Terminen zur Nutzung der Lötmaschinen möglich. Nachdem die relevanten Termindaten in einer Maske eingegeben wurden, öffnet sich eine Ergebnisseite. Hier werden dem Nutzer drei Terminvorschläge präsentiert. Zudem sind Visualisierungen zum Vergleich der Vorschläge hinsichtlich des Netzbezugs und der CO₂-Einsparung implementiert. Der Nutzer hat die Möglichkeit, einen Termin auszuwählen und diesen automatisiert in der Datenbank sowie in die Outlook-Kalender der beteiligten Maschinen und Mitarbeiter einzutragen. Zudem kann entweder eine ICS-Datei des Termins heruntergeladen oder die Termindaten über einen eigenen Mailserver direkt an Terminbeteiligte via E-Mail versendet werden. Das SolarSync-System verfügt über weitere Seiten zur Einstellung der Nutzerverwaltung, den Maschinendaten, dem Mailserver, der Wetter-API, dem Kafka Streaming-Server sowie den OPC UA Verbindungen. Über eine Profilseite kann das Profilbild und Passwort des angemeldeten Nutzers geändert werden.

7.2 Ausblick

Zusammenfassend lässt sich festhalten, dass das entwickelte SolarSync-System die Anforderungen der Projektstakeholder, maßgeblich die SEHO Systems GmbH und das Betreuerteam der Universität, erfolgreich umsetzt. Der ausführlich kommentierte und dokumentierte Programmcode befindet sich in einem GitHub-Repository und kann von dort aus auf einem Linux-Server aufgesetzt werden. Eine ausführliche Anleitung hierzu liegt dem Projektbericht bei. Das Projekt wurde somit zum 31.03.2023 erfolgreich abgeschlossen. In den folgenden Zeilen wird ein Ausblick gegeben, an welchen Bausteinen eine Weiterentwicklung des SolarSync-Systems als sinnvoll eingestuft wird. Der Ausblick teilt sich in die eigenen Ideen des Projektteams und die Ergebnisse der Evaluation, welche in Kapitel 6 präsentiert wurden, auf.

Aus Sicht des Projektteams könnten folgende Erweiterungen zu einer Verbesserung des SolarSync-Systems beitragen. Die Präzision der Optimierungsfunktion ist stark abhängig von den Input-Daten, insbesondere bei den Wetterdaten besteht hier Potenzial. Aktuell wird die kostenlose OpenWeatherMap-API genutzt, welche Bewölkungsdaten auf Tagesbasis bereitstellt. Die Integration der kostenpflichtigen Solar Radiation-API, welche den

stundengenauen Solarindex ausgibt, würde zu einer Verbesserung der Vorhersage der PV-Energie führen. Zudem könnten nach Installation der PV-Anlage die Daten dieser gesammelt werden. Die verbesserten Wetterdaten und die neuen PV-Daten könnten ebenfalls auf dem Streaming-Server gespeichert werden. Der Prototyp des Machine Learning Modells zur Vorhersage der PV-Energie könnte somit mit qualitativen Daten täglich trainiert werden. Die Integration der ML-Algorithmen würde die PV-Vorhersage und damit das Datenfundament der Optimierung weiter verbessern.

Langfristig könnte eine höhere Abstraktion des SolarSync-Systems erreicht werden, wenn der Aufbau der Optimierungsfunktion noch weiter modularisiert wird. Durch die Integration mehrerer Einstellmöglichkeiten der Optimierung könnte diese besser an weitere Unternehmen angepasst werden. Zudem ist die Berücksichtigung eines möglichen Energiespeichers sowohl bei der Visualisierung der Energiedaten als auch im Optimierungsmodell sinnvoll. Das System könnte zu einem Energiemanagementtool wachsen, welches auch den Netzbezug, die Ladung sowie die Entladung des Speichers und die PV-Anlage automatisiert steuert.

Gemäß der durchgeführten Evaluation des SolarSync-Systems konnten folgende Entwicklungspotenziale identifiziert werden. Bei der Entwicklung des Systems wurde die mobile Nutzung der Webanwendung nur eingeschränkt berücksichtigt. Zwar ist die Anwendung responsiv, allerdings besteht im Bereich der Nutzerfreundlichkeit noch Verbesserungspotenzial. Hier wünschen sich die Projektbeteiligten die Umsetzung einer mobilen Version des SolarSync-Systems. Des Weiteren wurde Potenzial für die Abstraktion der Softwarelösung erkannt. Mit unternehmensspezifischen Einstellungen und Anpassungsmöglichkeiten der Erzeuger und Verbraucher kann das SolarSync-System in einer größeren Zielgruppe an Unternehmen zum Einsatz kommen. Hierdurch können etwaige Speicherlösungen oder weitere Stromerzeuger, beispielsweise unternehmenseigene Windkraftanlagen, dem System hinzugefügt werden. Die Einführung einer Übersichtsseite, welche die Erzeuger und Verbraucher inklusive der relevanten Energiedaten abbildet, würde diesen Ansatz zudem unterstützen.

Neben diesen weiterführenden Entwicklungsschritten wäre der erste Schritt in Zukunft der Roll-Out des SolarSync-Systems bei der SEHO Systems GmbH. Durch die Integration des Systems in die Geschäftsprozesse der SEHO könnte eine umfassende Evaluation durchgeführt werden, die für die Weiterentwicklung wertvoll wäre. In Kombination mit der Softwareverbesserung könnte in einem darauffolgenden Schritt das SolarSync-System bei weiteren Unternehmen eingeführt werden. Abschließend lässt sich festhalten, dass durch die Projektbeteiligten großes Potenzial in SolarSync erkannt wurde. Durch die Weiterentwicklung der Softwarelösung kann das System zu einem umfassenden digitalen Energiemanager reifen, welcher Unternehmen bei der Reduktion ihrer Energiekosten sowie der Erreichung ihrer Klimaziele unterstützt.

Literaturverzeichnis

- Agile Business Consortium. (o. J.-a): *Moscow Priorisation*. In: <https://www.agilebusiness.org/dsdm-project-framework/moscow-priorisation.html>, zugegriffen am 21.03.2023.
- Agile Business Consortium. (o. J.-b): *User Stories*. In: <https://www.agilebusiness.org/resource/user-stories.html>, zugegriffen am 21.03.2023.
- Apache Software Foundation. (2022): *Apache Kafka*. In: <https://kafka.apache.org>, zugegriffen am 29.09.2022.
- Apache Software Foundation. (2023): *Apache Kafka*. In: <https://kafka.apache.org>, zugegriffen am 21.03.2023.
- Atlassian. Pty Ltd. (2023): *Tutorial: Using Confluence Cloud and Jira Software Cloud Together*. In: <https://www.atlassian.com/software/confluence/guides/expanding-confluence/confluence-and-jira>, zugegriffen am 21.03.2023.
- AXSOL GmbH. (o. J.): *Über Uns*. In: <https://www. axsol.de/ueber-uns-2/>, zugegriffen am 21.03.2023.
- Beck, K.; Beedle, M.; van Bennekum, A.; Cockburn, A.; Cunningham, W.; Fowler, M.; Grenning, J.; Highsmith, J.; Hunt, A.; Jeffries, R.; Kern, J.; Marick, B.; Martin, R. C.; Mellor, S.; Schwaber, K.; Sutherland, J.; Thomas, D. (2001): *Manifest Für Agile Softwareentwicklung*. In: <https://agilemanifesto.org/iso/de/manifesto.html>, zugegriffen am 21.03.2023.
- Bootstrap. (2022): *Bootstrap · the Most Popular Html, Css, and Js Library in the World*. In: <https://getbootstrap.com>, zugegriffen am 30.09.2022.
- Bootstrap. (2023): *Bootstrap*. In: <https://getbootstrap.com>, zugegriffen am 19.03.2023.
- Bundesministerium für wirtschaftliche Zusammenarbeit und Entwicklung. (2023): *Klimaabkommen Von Paris*. In: <https://www.bmz.de/de/service/lexikon/klimaabkommen-von-paris-14602>, zugegriffen am 21.03.2023.
- Conforti, M.; Cornuéjols, G.; Zambelli, G. (2014): *Integer Programming*, Springer.
- Cui, W.; Huang, L.; Liang, L.; Li, J. 2009. "The Research of Php Development Framework Based on Mvc Pattern," *2009 Fourth International Conference on Computer Sciences and Convergence Information Technology*: IEEE, pp. 947-949.
- de Souza, C. R.; Redmiles, D.; Cheng, L.-T.; Millen, D.; Patterson, J. 2004. "Sometimes You Need to See through Walls: A Field Study of Application Programming Interfaces," *Proceedings of the 2004 ACM conference on Computer supported cooperative work*, pp. 63-71.
- DIN Deutsches Institut für Normung e. V. (2009): *Projektmanagement - Projektmanagementsysteme*. In: <https://www.din.de/de/meta/suche/62730!search?query=69901>, zugegriffen am 21.03.2023.
- Domschke, W.; Drexl, A.; Klein, R.; Scholl, A. (2005): *Einführung in Operations Research*, Springer.
- Flask. (2010): *Design Decisions in Flask*. In: <https://flask.palletsprojects.com/en/2.2.x/design/#what-does-micro-mean>, zugegriffen am 19.03.2023.
- Gaber, M. M.; Zaslavsky, A.; Krishnaswamy, S. (2005): *Mining Data Streams*. In: ACM SIGMOD Record, 34 (2), 18-26.
- Garg, N. (2013): *Apache Kafka*, Packt Publishing, Birmingham.

- GitHub Inc. (2023a): *Flask*. In: <https://github.com/pallets/flask>, zugegriffen am 19.03.2023.
- GitHub Inc. (2023b): *Let's Build from Here*. In: <https://github.com/>, zugegriffen am 21.03.2023.
- Gurobi Optimization LLC. (o. J.): *Gurobi Optimizer. The World's Fastest Solver*. In: <https://www.gurobi.com/solutions/gurobi-optimizer/>, zugegriffen am 21.03.2023.
- Hannelius, T.; Salmenpera, M.; Kuikka, S. 2008. "Roadmap to Adopting Opc Ua," 2008 6th IEEE International Conference on Industrial Informatics: IEEE, pp. 756-761.
- Harris, J. (2016): *2016 Content Marketing Toolkit: 23 Checklists, Templates, and Guides*. In: <https://contentmarketinginstitute.com/articles/checklists-templates-guides>, zugegriffen am 21.03.2023.
- Huang, Y.; Garcia-Molina, H. 2001. "Publish/Subscribe in a Mobile Environment," Proceedings of the 2nd ACM international workshop on Data engineering for wireless and mobile access, pp. 27-34.
- IBCS Association. (2022): *International Business Communication Standards*, IBCS Media, Hilden.
- Imtiaz, J.; Jasperneite, J. 2013. "Scalability of Opc-Ua Down to the Chip Level Enables "Internet of Things"," 2013 11th IEEE International Conference on Industrial Informatics (INDIN): IEEE, pp. 500-505.
- Inray Industriesoftware GmbH. (2023): *Was Ist Opc Ua? Die Wichtigsten Begriffe Im Überblick*. In: <https://www.opc-router.de/was-ist-opc-ua/>, zugegriffen am 21.03.2023.
- IONOS SE. (2023): *Flask: Alles Wichtige Zum Micro-Framework*. In: <https://www.ionos.de/digitalguide/websites/web-entwicklung/flask-framework-im-ueberblick/>, zugegriffen am 19.03.2023.
- Jacobs, L.; Hensel-Börner, S. (2020): *Die Kraft Effektiver Daten-Visualisierung—Clear (I): Ein Leitfaden Zur Wirkungsvollen Dashboard-Gestaltung*. In: Data-driven Marketing: Insights aus Wissenschaft und Praxis, 43-75.
- Kaggle. (2022): *Solar Power Generation Data*. In: <https://www.kaggle.com/datasets/anikannal/solar-power-generation-data>, zugegriffen am 01.02.2023.
- Kahl, T.; Zimmer, F. (2017): *Kontextspezifische Visualisierung Von Prozesskennzahlen*. In: Geschäftsprozesse: Von der Modellierung zur Implementierung, 75-97.
- Lamothe, M.; Guéhéneuc, Y.-G.; Shang, W. (2021): *A Systematic Review of API Evolution Literature*. In: ACM Comput. Surv., 54 (8), Article 171.
- Loshin, P. (2022): *Definition Bash (Bourne Again Shell)*. In: <https://www.computerweekly.com/de/definition/Bash-Bourne-again-Shell>, zugegriffen am 31.03.2023.
- Mahnke, W.; Leitner, S.-H.; Damm, M. (2009): *Opc Unified Architecture*, Springer Verlag, Berlin, Heidelberg.
- Meyerbröker, P. (2011): *Agiles Projektmanagement – Eine Einführung*. In: Projekt Magazin, 12-21.
- Microsoft Corp. (2023): *Code Editing. Redefined*. In: <https://code.visualstudio.com/>, zugegriffen am 21.03.2023.
- Muthukrishnan, S. (2005): *Data Streams: Algorithms and Applications*. In: Foundations and Trends® in Theoretical Computer Science, 1 (2), 117-236.
- Nielsen, J.; Caya, P. (o. J.): *Intranet Portals: Ux Design Experience from Real-Life Projects*, (5 ed.), Nielsen Norman Group.
- NSRD. (2019): *Global Horizontal Solar Irradiance*. In: <https://nsrdb.nrel.gov/data-viewer>, zugegriffen am 01.02.2023.

- O'Donnell, D.; Zimmer, F. (2020): *Interaktive Datenvisualisierung Statistischer Daten*. In: Interaktive Datenvisualisierung in Wissenschaft und Unternehmenspraxis, 67-93.
- Ofoeda, J.; Boateng, R.; Effah, J. (2019): *Application Programming Interface (Api) Research: A Review of the Past to Inform the Future*. In: International Journal of Enterprise Information Systems (IJEIS), 15 (3), 76-95.
- OpenWeather. (2023): Climate Forecast 30 Days. In: <https://openweathermap.org/api/forecast30>, zugegriffen am 11.03.
- Pallets. (2007): *Introduction to Jinja*. In: <https://jinja.palletsprojects.com/en/3.1.x/intro/>, zugegriffen am 21.03.2023.
- Pallets. (2010): *Flask. Web Development, One Drop at a Time*. In: <https://flask.palletsprojects.com/en/2.2.x/>, zugegriffen am 21.03.2023.
- RedDot Design Museum. (2023): *Form Follows Function*. In: <https://www.red-dot-design-museum.de/essen/ausstellungen/design-grundlagen/designprinzipien/form-follows-function>, zugegriffen am 21.03.2023.
- Requests. (2023): *Requests: Http for Humans*. In: <https://requests.readthedocs.io/en/latest/>, zugegriffen am 11.03.
- Ries, E. (2011): *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Currency.
- Sass. (2023): *Sass*. In: <https://sass-lang.com>, zugegriffen am 19.03.2023.
- Schwaber, K.; Sutherland, J. 2022. "The Scrum Guide, the Definitive Guide to Scrum: The Rules of the Game, November 2020."
- Schwarz, M. H.; Borcsok, J. 2013. "A Survey on Opc and Opc-Ua: About the Standard, Developments and Investigations," 2013 XXIV International Conference on Information, Communication and Automation Technologies (ICAT): IEEE, pp. 1-6.
- SEHO Systems GmbH. (2022): Facebook Beitrag: Green Technology. In: <https://www.facebook.com/SEHOSystems/photos/a.137461986272592/5317044234980982/> zugegriffen am 21.02.2023.
- SEHO Systems GmbH. (o. J.): Willkommen Bei Seho. In: <https://www.seho.de>, zugegriffen am 21.03.2023.
- Shan, T. C.; Hua, W. W. 2006. "Taxonomy of Java Web Application Frameworks," 2006 IEEE International Conference on e-Business Engineering (ICEBE'06): IEEE, pp. 378-385.
- Shneiderman, B.; Plaisant, C.; Cohen, M. S.; Jacobs, S.; Elmquist, N.; Diakopoulos, N. (2016): *Designing the User Interface: Strategies for Effective Human-Computer Interaction*, Pearson.
- Spichale, K. (2019): *Api-Design: Praxishandbuch Für Java-Und Webservice-Entwickler*, dpunkt. verlag.
- Sprotten, A. N. (2022): *Datenvisualisierung Und Business Intelligence–Methoden Zur Unterstützung Der Entscheidungsfindung*. In: Marketing Analytics: Perspektiven–Technologien–Anwendungsfelder, 139-159.
- Steinrücke, M. (2007): *Grundlagen Der Termin-Und Kapazitätsplanung*. In: Termin-, Kapazitäts-und Materialflussplanung bei auftragsorientierter Werkstattfertigung, 7-16.
- Tarkoma, S. (2012): *Publish/Subscribe Systems: Design and Principles*, John Wiley & Sons, Hoboken.
- Tatbul, N. 2010. "Streaming Data Integration: Challenges and Opportunities," in: 2010 IEEE 26th International Conference on Data Engineering Workshops (ICDEW 2010). pp. 155-158.
- Umweltbundesamt. (o. J.): *Energiemanagementsysteme*. In: <https://www.umweltbundesamt.de/themen/wirtschaft-konsum/wirtschaft->

umwelt.umwelt-energiemanagement/energiemanagementsysteme, zugegriffen am 21.03.2023.

Wirth, H. (o. J.): *Aktuelle Fakten Zur Photovoltaik in Deutschland*. In: <https://www.ise.fraunhofer.de/content/dam/ise/de/documents/publications/studies/aktuelle-fakten-zur-photovoltaik-in-deutschland.pdf>, zugegriffen am 21.03.2023.

WUQM Consulting GmbH. (o. J.): *Als Team Zum Nachhaltigen Erfolg*. In: <https://www.wuqm.de/startseite/>, zugegriffen am 21.03.2023.

Anhang

Anhang 1 – Risikotabelle mit Risikoindex

Tabelle 23: Risikotabelle mit Risikoindex

Nr	Beschreibung	Art	Ursache	Auswirkung	EW	SH	RI
R 1	Fehlende Erfahrung mit Entwicklungsumgebung	Personell	Mangelnde Fachkenntnisse	Geringere Qualität des Produkts	1	4	4
R 2	Fehlendes Knowhow in Softwarereprogrammierung	Personell	Mangelnde Fachkenntnisse	Geringere Qualität des Produkts	1	5	5
R 3	Fehlendes Knowhow im Umgang mit Hardware	Personell	Mangelnde Fachkenntnisse	Geringere Qualität des Produkts	1	5	5
R 4	Ausfall von Hardwarekomponenten (z.B. Server)	Technisch	Fehlende technische Ressourcen	Verzögerung in der Systementwicklung	3	3	9
R 5	Datenverlust	Technisch / Prozess	Unzureichende Datensicherung	Verzögerung in der Systementwicklung	2	4	8
R 6	Verstoß gegen Datenschutzgesetze	Personell	Mangelnde Fachkenntnisse	Geringere Qualität des Produkts	3	3	9
R 7	Nichteinhalten der zeitlichen Vorgaben	Prozess	Schlechte Zeitplanung	Nichtbestehen des Projektseminars	1	5	5

Anhang 2 – Protokolle

In diesem Abschnitt sind alle Protokolle, die während Besprechungen geführt wurden in chronologischer Reihenfolge und stichpunktartig aufgeführt.

23.08.2022 – Pre-Kick-Off

- Vorstellungsrunde
 - Who-is-who übersprungen, man kennt sich
- Projektthema
 - Überblick über die Problemstellung und die (für uns) externen Projektteilnehmer
 - primäres Projektziel: Webanwendung die Rahmen- und Plandaten (PV-Anlage, Akkustand, Kalender, Wetter) als Input verarbeitet und für Anwender verständlich visualisiert
 - zusätzlich vorstellbar: Machine Learning, Optimierungsthemen
- Organisation
 - Vorschläge zu Projektrollenverteilung (Frontend Engineer, Backend Engineer, Product Owner, Allgemeine Aufgaben)
 - Vorschläge zu Workflow (alles dokumentieren erleichtert Nachvollziehbarkeit und Berichterstellung)
 - Projektinfrastruktur (Code-Verwaltung > Git, Arbeitspaketverwaltung > Trello)
 - Coding Tools (Anaconda mit Python Package-Verwaltung, VSCode, IntelliJ, PyCharm, etc.)
 - Information über Workload des Projektseminars (Hausarbeit, Projektbericht, Präsentation 1:2:1)
- weitere Organisation
 - Jour Fixes (feste Termine mit Betreuern, z.B. jede zweite Woche)
 - Seminarräume für Präsenzmeetings oder gemeinsames Arbeiten können gebucht werden
 - Datenaustausch über Teamdrive
 - Projektvorgehen: basic Prototyp entwickeln und darauf anschließend aufbauen
- Themenvorstellung Hausarbeiten
 - Überblick über Themen
 - jedes Thema soll einen kleinen Prototyp enthalten (praktische Erfahrung sammeln)

10.11.2022 – Präsentation Hausarbeiten

- Präsentation
 - 10 Minuten pro Person/Hausarbeit
 - gemeinsamer Ausblick auf das Projekt
- Feedback Runde

23.11.2022 – Jour Fixe #1

- Sprint Review - Sprint 0
 - Präsentation der Projektplanung
 - Information über zu verwendende Tools
 - Präsentation Phasen-/Meilensteinplan
 - Einladung des Betreuerteams in Confluence und Jira
- Bedarf / To Do
 - Verknüpfung von Jira und GitHub
 - SEHO
 - Zugang zu OPC UA
 - Maschinengenaue Messung der Verbrauchsdaten
 - Aufteilung der Verbrauchsdaten in Seminartermine & andere Unternehmensbereiche (z.B. Produktion)
 - Termin zur Besichtigung und Besprechung
 - Herstellung Kontakt zu Ansprechpartner
- Teambuilding
 - Weihnachtsmarkt

30.11.2022 – Jour Fixe #2

- Sprint Review - Sprint 1
 - Präsentation und Demonstration des MVP
 - Rückblick auf Sprint 1
- **Bedarf / To Do**
 - SEHO
 - Zugang zu OPC UA
 - Maschinengenaue Messung der Verbrauchsdaten
 - Aufteilung der Verbrauchsdaten in Seminartermine & andere Unternehmensbereiche (z.B. Produktion)
 - Termin zur Besichtigung und Besprechung
 - Herstellung Kontakt zu Ansprechpartner
- Teambuilding
 - Weihnachtsmarkt

16.12.2022 – Meeting mit SEHO

- Kennenlernen
 - übersprungen, da verspätet angefangen
- Präsentation
 - Demonstration der Anwendung zum aktuellen Stand

- Ausblick auf zukünftige Funktionen
- Fragenkatalog
 - Wie werden aktuell Termine geplant? (macht Calendly o.ä. Sinn, oder schlagen euere Mitarbeiter den Kunden einfach freie Terminslots vor?)
 - “Normale Terminvergabe” → Kunden schickt Anfrage, Mitarbeiter schauen, wann eine Maschine frei ist und senden dann 3-4 Terminvorschläge.
 - Wer nutzt das Tool? (ITler, Business User, Management?)
 - Begrenzter Mitarbeitertool, die wissen sollten, was sie tun
 - Fragen zur Optimierung:
 - Wie sehen Termine aus?
 - Dauer, Uhrzeiten (frühester Start, spätestes Ende)
 - Arbeitstage → entweder halbtags oder ganztags Slots
 - noch kein Outlook Kalender, kommt nächstes Jahr
 - Einschränkung: Maschine und Mitarbeiter muss frei sein
 - Belegung der Maschinen soll in Outlook hinterlegt sein
 - 2 Mitarbeiter pro Maschinengruppe benötigt (eine Maschinengruppe umfasst bis zu 4 Maschinen)
 - Terminplanung auch in Abhängigkeit anderer Termine, wenn die Maschine eh schon warm ist
 - Parallele Termine
 - Anzahl an Maschinen im Showroom & Backup-Maschinen
 - Interessant: Auch die Fertigungsanlagen in der Produktion könnten durch das Tool optimiert werden
 - Energiespeicher geplant?
 - Aktuell nicht
 - Gesamtabschaltung der Maschinen möglich?
 - Starke Belastung durch Abschaltung, daher soll die Anzahl des Herunter- und Hochfahrens minimiert werden
 - Entlohnung der Netzeinspeisung
 - OPC UA Zugang
 - Detaillierte Verbrauchsdaten anfragen (Maschinen, Verwaltung, Produktion, etc.)
 - Stand jetzt ein Stromzähler für das gesamte UN, er lässt uns Verbrauchsdaten zukommen
 - Interessant, dass Energieverbräuche getrennt werden können → implementieren?
 - Verbrauchsdaten der Maschinen basierend auf Leistungsdaten, die wir bereits haben
 - Termin zur Besichtigung vor Ort im Januar
 - 24, 25, 26 und 27. Januar möglich → besprechen

21.12.2022 – Jour Fixe #3

- Sprint Review - Sprint 2
 - Präsentation / Demonstration aktueller Stand
 - Erklärung des erweiterten Funktionsumfangs

- Bedarf / To Do
 - Überarbeitung der Optimierungsfunktion entsprechend den Energiedaten der SEHO
 - Erzeugung von Dummy Kalendern für Maschinen für Optimierungsfunktion
 - Anbindung von Kalendern der Mitarbeiter für Optimierungsfunktion
 - Wetter-/Energieerzeugungsvorhersage mit Solar API von OpenWeather.net als zusätzliche Funktion
- Teambuilding
 - Weihnachtsmarkt

11.01.2023 – Jour Fixe #4

- Präsentation aktueller Stand
 - Rollenkonzept
 - Fehlermeldungen bei Falscheingaben
 - Settings Page
 - Profilseite
 - Einbindung API OpenWeather
 - Kalenderintegration
 - Versenden von Terminen als E-Mail direkt aus dem Tool möglich
- Ausblick / nächste Schritte
 - Anlage Kalender für jeweilige Maschinen
 - Vorhersagemodell für die Energiegewinnung erstellen
 - PV-Anlage gibts noch nicht, daher auch keine historischen Daten, mit denen wir arbeiten können
 - Verwendung von Daten der Globalstrahlung am Standort unter 0% Bewölkung Bedingungen (Durchschnittswerte über ein Jahr)
 - zusätzlich Verwendung von jetzt verfügbaren Wettervorhersagedaten (stündliches Intervall)
 - Maschinen- und Mitarbeiterverfügbarkeit über Kalender API
 - Anpassung des Terminplanungsformulars im Frontend
 - Ergänzung um Auswahlfeld für Mitarbeiter
- Grober Ausblick - 2 Themenblöcke
 - Vorhersage der Energieversorgung durch Erweiterung des Optimierungsmodells
 - Inhalts-/Designkonzept
- Feedback der Betreuer:
 - statt Mobile App lieber Prediction/Forecasting angehen (wenigstens prototypisch)
 - → eigentlich kein ML notwendig, da nichts “gelernt” wird, außerdem: keine historischen Daten und auch keine standortspezifischen Daten extern verfügbar
 - → außerdem: keine unbekannte Variable in der Gleichung, die ML rechtfertigt

- → Datenqualität würde sich nicht ändern, Wetterdatenbank gibts bereits kostenfrei als Library, wozu dann ML-Wettervorhersagedaten generieren (redundant)
- Deadline für Abgabe: Mitte/Ende März voraussichtlich, kein festes Datum bekannt, muss mit anderen Lehrstühlen geklärt werden
 - Bitte um Anforderungen an Bericht

24.01.2023 – Meeting bei SEHO vor Ort

- Offene Fragen
 - Welchen Anteil macht der Showroom in den Stromverbrauchsdaten? Laufen die Maschinen momentan durchgehend an den Wochentagen (Wochenende = Standby)?
 - Anforderung von Daten der Stromverbräuche vom gesamten Jahr 2022 (bisher haben wir Daten bis 12.12.2022) und Daten aus Jahren 2020 und älter
 - Welche Mitarbeiter greifen auf die Software zu?
 - Wie lief die Terminplanung bis jetzt ab?
 - Wie funktioniert das Pachtsystem der PV-Anlage? Wann und was wird gezahlt?
Vergleiche zu herkömmlichen Netzbezug
- Ausblick / To Do
 - Verbrauch ist davon abhängig, was im Betrieb gemacht wird → bei Optimierung Datenfeld mit Option zu niedrigerer Auslastung (Vollast, 70 %, 50 %, 30 %)
 - Dummy Maschine hinzufügen zur optimalen Planung, damit auch Termine zur internen Testung geplant werden können
- Erhaltene Informationen
 - OPC UA
 - Alle Maschinen im Showroom sind OPC UA fähig
 - Es könnten aktuelle Leistungsdaten (z. B. Temperatur) ausgelesen werden → Diese werden allerdings nicht geloggt.
 - Stromverbräuche der Produktion
 - Verbräuche sind unerheblich, da diese sich nicht über verschiedene Wochentage unterscheiden → Unterstellung einer einheitlichen Grundlast genügt
 - Maschinen in Produktion werden von der Qualitätssicherung vor der Auslieferung testweise angeschaltet → Diese Zeitpunkte dieser Tests sollen auch über das Tool optimiert werden
 - Maschinen
 - Erweiterung der Einstellmöglichkeiten bei der Terminoptimierung: Der Energieverbrauch der Maschine während des Betriebs hängt von dem zu lögenden Produkt ab → 3 Kategorien sollen wählbar sein (einfach, normal, schwer), wobei “schwer” = 100 % der Betriebsleistung der Maschine ist. “Einfach” und “normal” sind dementsprechend Abstufungen davon.
 - Die Verbrauchsdaten in den Phasen Aufheizen und Betrieb sollen anpassbar sein.
 - Maschinenübersicht:
 - StartSelective, LeanSelect: Lötfeld 5

- SelectLine C: 2x Lötbad 5
- LeanSelect+: 2x Lötbad 5
- Power: bis zu 2 Lötbäder (3/4 und/oder 5), im Showroom immer 2 Bäder, Standard-Config: 2x Lötbad 4
- MaxiSelective: für Betrachtung egal - existiert nicht im Showroom (Lötbad 4)
- Reflow: Keine Aufheizzeiten, kein Stand-By berücksichtigen (reine Betriebsverbräuche)
- Generelle Infos zu Lötbädern:
 - Lötbad 4 hat 20 % mehr Energie als Lötbad 3, aber ist laut Reinhardt irrelevant und soll nicht unterschieden werden → Deshalb Lötbad 3/4 mit einem Verbrauchswert
 - Generell ist meistens die Anzahl der Lötbäder anpassbar
- Daten
 - Beispielhafte Log-Daten einer Wellenlötmachine
 - Gesamte Stromverbräuche der SEHO ab 2020
 - Unternehmenspräsentation → Übersicht Maschinen
- PV-Anlage
 - Kauf der PV-Anlage
 - Evt. Batteriespeicher → für Terminplanung irrelevant
 - Speicher soll auch über Netz geladen werden
 - Planung mit 6,2 Cent Einspeisevergütung
 - Im Juni/Juli soll die PV aufs Dach
- User Stories
 - 2 Fälle zur Verwendung des Planungstools:
 - Kunde fragt Anlage an → Versuch auf dieser Anlage soll in bestimmtem Zeitraum geplant werden
 - Kunde möchte neues Produkt auf den Maschinen testen → deutlich kurzfristiger als Fall 1
 - Nutzung des Tools durch 4 Applikationsmitarbeiter sowie deren 2 Chefs und Qualitätssicherung

25.01.2023 – Jour Fixe #5

- Bericht über Treffen bei SEHO
- Anforderungen an Projektbericht
 - Abgabe 31.03.2023
 - 15-20 Seiten pro Person (insges. max. 100 Seiten, lieber weniger und kompakt!)
 - Einleitung, technische/generelle Grundlagen (Vorberichte gerne nutzen aber nicht copy paste), weiterer Aufbau unsere Lösung
 - Produkt soll im Mittelpunkt stehen (nicht jede Tages- und Nachtzeit das Vorgehen dokumentieren) Projektmanagement ist wichtig, aber Produkt ist wichtiger
 - Ausblick, weitere Möglichkeiten, Lessons Learned (studentisches Projekt, was haben wir mitgenommen allgemein und einzelne Bereiche)
 - alles ohne großen Mehrwert in den Anhang

- Code nicht direkt in den Bericht, lieber subsummieren oder konzeptuelle Diagramme verwenden
- zusätzlich technische Doku abgeben, in dem Code beschrieben wird (Dokument das erklärt, welche Hard- und Software notwendig sind, um die Anwendung laufen zu lassen und wie setzt man die Software auf)
 - technische Doku für Administrator (Inbetriebnahme der Software)
 - technische Doku für User (Anwendung der Software)
- lokales Repository als .zip Datei auf Teamdrive hochladen (Code entsprechend auskommentieren, um nachzuvollziehen, was gemacht wurde)
- Softwareevaluation
- Bericht aufbauen mit
 - Was waren die Grundannahmen/Ausgangspunkt?
 - Wie wurde es umgesetzt?
 - Verweis auf Code
 - Reflexion
- Ausblick / To Do
 - Umsetzung der neuen Anforderungen aus dem Meeting vor Ort bei SEHO
 - Speicherung aller relevanten Maschinendaten im Outlook (änderbar über Tool)
 - Anpassbarkeit der Leistungsdaten der Maschinen im Tool
 - Umstellung der Optimierungsfunktion auf Grundlast + PV-Energie
 - Erweiterung um Kategorien des zu lötenden Produkts
 - Log-Daten analysieren
 - Anpassung User Stories und Anforderungskatalog
 - Finalisierung und Umsetzung Inhalts- und Designkonzept
 - Feedback auch von Nico: Farbkonzept, Inhaltskonzept, optisch angenehme Aufbereitung nach Designprinzipien

01.02.2023 – Meeting mit allen Projektpartnern

- Vorstellungsrunde
- Präsentation
 - kurze Vorstellung der implementierten Funktionen
 - Live Demo der Anwendung
 - Feedbackrunde
 - Ausblick

15.02.2023 – Jour Fixe #6

- Vorstellung der finalen Funktionen
- Vorstellung der Gliederung für den Projektbericht
 - Passt so
 - Theoretische Grundlagen benötigt, aber relativ kurz halten → Theoretische Einführung und Anwendung auf Projekt (Welche Tools etc. wurden ausgewählt?)

- Zusatzinformationen in den Anhang
- Ausblick / To Do
 - Finalisierung Design
 - Frontend-Arbeiten
 - Fragebogen
 - Erstellung
 - Versenden
 - Auswertung
 - Beginn mit Projektbericht

08.03.2023 – Jour Fixe #7

- Präsentation der abschließenden Funktionen
- Feedback zu Projektbericht Gliederung

Anhang 3 – Personas

Kunden

Britta Schneider ist als Head of Production im Bereich Navigationstechnik bei einem führenden deutschen Automobilhersteller tätig. Auf Grund ihres Fachbereichs ist Frau Schneider gut im Umgang mit digitaler Kommunikation. In ihrem Unternehmen wird zur internen Kommunikation Office 365 genutzt.

Es kommt ihr sehr entgegen, wenn sie vereinbarte Termine direkt als Termineinladung in ihren Kalender integrieren kann, da sie dadurch stets einen Überblick über wichtige Ereignisse behalten und automatisch daran erinnert werden kann.

Termine zu verpassen oder parallel zu anderen Ereignissen zu erhalten wäre problematisch für Frau Schneider, daher sind wir der Meinung, dass eine Versendung der erzeugten und ausgewählten Termine direkt aus der Anwendung heraus eine ideale Ergänzung der Software ist, um Kunden eine angenehme User Experience zu bieten.

Management

Dr. Andreas Reinhardt ist Mitglied der Geschäftsführung bei SEHO. Sein Fokus liegt auf der effizienten Nutzung der PV-Anlage des Unternehmens, um durch die optimale Auslastung der Maschinen Energie zu sparen und Kosten zu senken. Daneben ist ihm wichtig, dass die Kunden zufrieden sind und die Termine reibungslos ablaufen. Er wünscht sich zudem eine übersichtliche Darstellung der Solardaten.

Dr. Reinhardt ist technikaffin, tendenziell ein Late Mover, jedoch offen und motiviert für technische Fortschritte im Unternehmen und plant als Nächstes den Roll-Out von Office 365. Um ihm zu helfen, können wir ihm eine übersichtliche Darstellung der Stromerzeugnisse durch die PV-Anlage bieten und Live-Updates über Stromkosten und Stromverbrauch der Showroom-Maschinen bereitstellen. Eine automatisierte Terminerstellung und Möglichkeit der Integration in gängige Kalenderanwendungen sowie eine Optimierung der Terminplanung können helfen, Kundetermine wie gewünscht mit eigens generiertem Strom zu betreiben. Durch die optimierte Nutzung der PV-Anlage kann SEHO Systems Energie einsparen und somit Kosten senken.

Vertrieb

Herr Lausecker ist Vertriebsbereichsleiter bei SEHO. Als staatlich geprüfter Maschinenbauer ist er technikaffin und dank seines jungen Alters vertraut im Umgang mit neuesten Technologien. Der Ausblick auf eine moderne Ausrichtung des Unternehmens mit integrierten Systemen und automatisierten Abläufen motiviert ihn.

Als Vertriebsmitarbeiter ist er im ständigen Austausch mit Kunden, präsentiert die Produkte des Unternehmens und vereinbart die zugehörigen Termine. Die manuelle Auswahl eines geeigneten Termins stellt für ihn zusätzlichen Zeitaufwand dar, der vermieden werden kann. Auch die Medienbrüche durch die Bearbeitung verschiedener Kalender bei der Terminplanung können in seinen Augen reduziert werden.

Wir können Herrn Lausecker helfen, indem wir verschiedene ideale Termine vorschlagen, die er sofort mit den Kunden besprechen kann. Eine automatische Integration der gebuchten Termine in alle betroffenen Kalender optimiert den Prozess der Terminplanung zusätzlich, da die kritisierten Medienbrüche reduziert werden.

externe Projektpartner – WUQM

Dr. Sebastian Fiedler ist ein Experte für Energieeffizienz und Nachhaltigkeit. Als Berater und Projektmanager bei WUQM hilft er Unternehmen, ihre Energiekosten zu senken und gleichzeitig die Umweltbilanz zu verbessern. Dabei liegt ihm besonders die sinnvolle Nutzung und Einsparung von Energie am Herzen.

Dr. Fiedler ist frustriert darüber, dass viele Unternehmen sich nicht mit Energiespitzen beschäftigen und dadurch unnötige Kosten verursachen. Er setzt sich dafür ein, dass Energieerzeugung und -nutzung intelligent gesteuert werden, um Kosten zu senken und die Umwelt zu schonen.

Als Early Mover und technikaffiner Experte ist Dr. Fiedler offen für innovative Lösungen, die eine nachhaltige Energienutzung unterstützen. Er wünscht sich, dass PV-Anlagen nicht nur aufgrund einer allgemeinen Kosten-Nutzen-Analyse installiert werden, sondern auch intelligente Nutzungsmöglichkeiten berücksichtigt werden.

Um Dr. Fiedlers Vorstellungen umzusetzen, können wir eine modular aufgebaute Anwendung erstellen, die es ermöglicht, Erweiterungen auf andere Verbraucher und Versorger zu einem späteren Zeitpunkt zu integrieren. Zudem möchte er die Daten der PV-Anlage auswerten können, um das Nutzungsverhalten zu optimieren und weitere Einsparpotenziale zu identifizieren.

externe Projektpartner – Axsol

Felix Beyer ist Key Account Manager und IT-Officer bei Axsol, einem Unternehmen, das Batteriespeichersysteme herstellt und integriert.

Seine Motivation liegt darin, die verfügbare Technik im Bereich der Photovoltaikanlagen und Batteriespeichernutzung optimal zu nutzen. Seine Frustration resultiert aus der unreflektierten Installation von PV-Anlagen ohne Überlegungen zur Energiespeichernutzung und Netzeinspeisung. Als Absolvent eines Wirtschaftsinformatikstudiums ist er sehr technikaffin.

Als Key Account Manager betreut er die wichtigsten Kunden von Axsol und gestaltet intelligente Lösungen für sie.

Wir können Felix helfen, indem wir die Anwender für den Einsatz von Energiespeichern sensibilisieren und ihnen den Autarkie- und Versorgungsgrad aufzeigen. Auf diese Weise können wir sicherstellen, dass die PV-Anlagen und Batteriespeichersysteme optimal genutzt werden und die Kunden von Axsol das volle Potenzial ihrer Systeme ausschöpfen können.

Anhang 4 – User Stories

Must Have

- Als Geschäftsführer möchte ich die Belegung der Maschinen möglichst nah beieinander, um unnötige Aufheiz- und Abkühlphasen zu vermeiden und so lange Wartezeiten meiner Mitarbeiter zu vermeiden und Energiekosten zu sparen.
- Als Geschäftsführer möchte ich ein fehlerresistentes System, um meinen Kunden die Terminplanung und Durchführung so angenehm wie möglich zu gestalten.
- Als Geschäftsführer möchte ich die optimale Ausnutzung der PV-Anlage, um einen hohen Return of Investment zu erreichen und die anfallenden Energiekosten deutlich zu senken.
- Als Geschäftsführer möchte ich einen gesonderten Bereich, in dem Einstellungen für die Anwendung hinterlegt werden, um die Kontrolle über das System zu behalten und potenziellen Änderungen durch Unbefugte zu vermeiden.
- Als Vertriebsmitarbeiter möchte ich eine Kalenderübersicht mit allen Terminen meiner Mitarbeiter, um sehen zu können, wann meine Kollegen für weitere Aufgaben verfügbar sind.
- Als Vertriebsmitarbeiter möchte ich eine Kalenderübersicht mit allen Terminen der Maschinen, um mich stets darüber informieren zu können, wann welche Maschinen bereits gebucht sind.
- Als Vertriebsmitarbeiter möchte ich eine Kalenderübersicht mit meinen eigenen Terminen, um stets einen Überblick zu haben, wann Termine mit mir geplant wurden.
- Als Mitarbeiter im Kundenservice möchte ich eine möglichst intuitive Anwendung, die keine langen Schulungen voraussetzt, um ohne steile Lernkurve die neue Terminplanung mit den Kunden ausführen kann.
- Als Mitarbeiter im Kundenservice möchte ich eine übersichtliche Software, um bei der Terminvergabe Zeit zu sparen.
- Als Mitarbeiter im Kundenservice möchte ich eine Kalenderübersicht, um sehen zu können, wann Mitarbeiter und/oder Maschinen bereits gebucht sind.
- Als Kunde möchte ich, dass meine Kontaktarten den Datenschutzrichtlinien entsprechend verwaltet werden, um meine persönlichen Daten zu schützen.

Should Have

- Als Geschäftsführer möchte ich ein transparent gestaltetes System, um die Terminerstellung jederzeit nachvollziehen zu können.

- Als Vertriebsmitarbeiter möchte ich einen persönlichen Account, um jederzeit auf meine eigenen Informationen zugreifen und volle Kontrolle darüber haben können, wer von meinem Account aus Termine plant.
- Als Kunde möchte ich mehrere Terminvorschläge erhalten, um einen geeigneten Termin flexibel in meine persönliche Terminplanung zu integrieren.
- Als Kunde möchte ich den vereinbarten Termin direkt an mein persönliches E-Mail-Postfach gesendet bekommen, um automatisiert und rechtzeitig an den bevorstehenden Termin erinnert zu werden.

Could Have

- Als externer Projektpartner und potenzieller Promoter der Anwendung bei anderen Unternehmen möchte ich eine Anzeige des aktuellen Autarkie- und Versorgungsgrads, um Anwender für den Einsatz von Energiespeichersystemen zu sensibilisieren.
- Als externer Projektpartner und potenzieller Promoter der Anwendung bei anderen Unternehmen möchte ich einen modularen Aufbau der Anwendung, um Erweiterungen auf andere Verbraucher und Versorger zu ermöglichen.
- Als Geschäftsführer möchte ich eine Anzeige, auf der Energiedaten des Unternehmens ausgewertet werden können, um die Energienutzung nachhaltig zu gestalten.
- Als externer Projektpartner und potenzieller Promoter der Anwendung bei anderen Unternehmen möchte ich eine Auswertung der Energiedaten des Unternehmens zur nachhaltigen Energienutzung.
- Als Mitarbeiter möchte ich Zugriff auf meine Benutzerdaten haben, um mein Profilbild oder mein Passwort ändern zu können.

Won't Have

- Als Kunde möchte ich direkten Zugriff auf die Anwendung haben und den Terminkalender und verfügbare Termine selbst einsehen können, um meine eine Terminplanung flexibler zu gestalten und mir den Termin auszusuchen, der für mich am passendsten ist.

Anhang 5 – Detaillierte Dokumentation Sprintablauf

SPRINT 0: ORGANISATIONSPHASE: 16.11.2022 – 22.11.2022

Um die aktive Projektzeit vorzubereiten, wurde ein verkürzter siebentägiger Sprint durchgeführt. In diesem wurden die zukünftigen Treffen mit dem Betreuerteam sowie interne Besprechungen geplant und Werkzeuge für die Projektdurchführung festgelegt. Hierbei wurden unter anderem Zoom für Besprechungen, WhatsApp für die interne Kommunikation im Team, GitHub für die gemeinsame Arbeit am Quellcode und Atlassian-Produkte wie Jira und Confluence zur Koordination und Dokumentation gewählt. Zusätzlich wurde der Technologie-Stack an Tools und Frameworks für das Projekt bestimmt.

Sprint 1: Implementierung MVP: 23.11.2022 – 07.12.2022

Im ersten Sprint des Projekts konzentrierte sich das Team darauf, das MVP zu erstellen, indem es sich auf die funktionalen Aspekte konzentrierte. Um sicherzustellen, dass die Anwendung nicht nur lokal auf den Rechnern des Projektteams läuft, sondern auch über einen Browser abgerufen werden kann, wurde eine virtuelle Maschine mit einem Linux-Server im ERP-Labor der Julius-Maximilians-Universität aufgesetzt und an das gemeinsame GitHub Repository angebunden. Eine MySQL-Datenbank wurde auf dem Server erstellt, um die benötigten Daten für die Optimierungsfunktion zu speichern. Da zu diesem Zeitpunkt nur reale Daten zum allgemeinen Stromverbrauch des Auftraggebers vorlagen, mussten synthetische Daten zu Stromverbräuchen der Maschinen sowie zu Stromerzeugung durch die Photovoltaikanlage modelliert werden. Der HTML-Code wurde parallel dazu strukturiert, um die Anwendung übersichtlich und pflegeleicht zu gestalten. Das Ergebnis des ersten Sprints war eine Anwendung zur Terminoptimierung, basierend auf den synthetisch generierten Energiedaten (vgl. Abbildung 27 und Abbildung 28).

Energiebasierte Terminoptimierung

The screenshot shows the 'Optimization' section of the application. On the left, a sidebar menu lists 'Home', 'Dashboard', and 'Optimization'. The 'Optimization' item is highlighted with a blue background. The main area has a header 'Optimization' and a sub-header 'Planungshorizont festlegen'. It contains two date input fields: 'Beginn *' (01.11.2022) and 'Ende *' (15.11.2022). Below this is a section titled 'Termine definieren' with a table. The table has columns: 'Termin ID', 'Bezeichnung', 'Dauer (in Stunden)', and 'Aktionen'. A row is shown with values: 1, Test Termin, 4, and edit/refresh icons. A button '+ Neuer Termin' is at the top right of the table. At the bottom of the main area is a 'Optimieren' button. The bottom left corner of the sidebar shows a user icon with 'User'.

Abbildung 27: MVP (Terminanforderung)

This screenshot shows the 'Optimization' section after term creation. The sidebar and header are identical to the previous screenshot. The main area now displays a table with two rows of data. The columns are 'Termin ID', 'Datum', and 'Uhrzeit'. The first row has values 1, 01.11.2022, and 14:00. The second row has values 2, 03.11.2022, and 12:00. A 'Submit' button is visible below the table. The bottom left corner of the sidebar shows a user icon with 'User'.

Abbildung 28: MVP (Terminausgabe)

SPRINT 2: IMPLEMENTIERUNG RELEASE 2: 07.12.2022 – 21.12.2022

Im zweiten Sprint sollte der Funktionsumfang der Anwendung erweitert werden, jedoch wurden zunächst bestehende Herausforderungen aus dem vorherigen Sprint behoben. Probleme mit der Kommunikation zwischen dem Server und Gurobi aufgrund der verwendeten Gurobi-Lizenz wurden durch Proxy-Einstellungen und Anpassung der Uniserver IP gelöst.

Das Formular für die Terminoptimierung wurde fehlerresistent gemacht, indem die Maschinenauswahl als Pflichtfeld hinterlegt wurde. Wurden keine Daten oder ein falsches Datum (z. B. Vergangenheitswerte) angegeben, fordert nun eine Fehlermeldung den Nutzer auf, die Angaben zu korrigieren oder zu ergänzen. Zudem wurden Terminen automatisch eine ID hinzugefügt und der Zwischenspeicher der Termine wurde überarbeitet, da die Löschung von Terminen bis zu diesem Zeitpunkt zwar visuell im Frontend möglich war, die Daten jedoch nicht aus den Tabellen im Backend entfernt wurden.

Nachdem alle Schwierigkeiten behoben waren, konnten weitere Funktionen auf Basis des Sprint Backlogs umgesetzt werden. Die Maschinenauswahl im Formular für die Terminoptimierung wurde erweitert, sodass nun mehrere Maschinen für einen Kundentermin ausgewählt und in der Optimierungsfunktion berücksichtigt werden können (vgl. Abbildung 29).

ID	Bezeichnung	Maschinen	Dauer (h)	Aktionen
#1		Bitte wählen ... Wellenlöt Lötbad 3/4 Lötbad 5		±

Abbildung 29: Release 2 (Mehrfachauswahl im Formularfeld Maschinen)

Nach der Optimierungsanfrage wurde die Ausgabe der Terminvorschläge um Infografiken ergänzt, die Informationen zu Stromverbräuchen der Termine einschließlich der Verteilung anteilig nach Netzbezug und Eigenstromnutzung anzeigen (vgl. Abbildung 30).

Energiebasierte Terminoptimierung

The screenshot shows the 'Optimization' section of the application. It displays a table of scheduled tasks:

ID	Bezeichnung	Maschinen	Datum	Uhrzeit	Dauer (h)	Kalendereinladung
#1	Termin 1	Wellenlöt	09.05.2022	13:00	3	Kalendereintrag
#2	Termin 2	Lötbad3/4, Lötbad5	10.05.2022	13:00	2	Kalendereintrag

Below the table are three infographics:

- Stromverbrauch der Termine:** A donut chart showing energy consumption. The segments are labeled 100 and 150.
- Netzbezug der Termine:** A table showing power generation from PV:

Termin ID	Strom durch PV
#1	42.0 %
#2	45.0 %

A note states: "Die Planung aller Termine führt zu einem Stromverbrauch in Höhe von 250 kWh." A second note states: "Es werden 142.0 kWh aus dem Stromnetz bezogen."
- Anteil PV-Strom:** A donut chart showing the percentage of PV power. The segments are labeled 55, 87, 142, and 105.

Abbildung 30: Release 2 (Erweiterung um Infografiken zu vorgeschlagenen Terminen)

Zusätzlich wurde eine Option zum Download der erzeugten Termine als ICS-Datei erarbeitet, um diese in verschiedene Kalenderanwendungen einzufügen. Die heruntergeladene Datei kann wie eine normale Termineinladung über Outlook als Mail-Anhang versendet werden (vgl. Abbildung 31).

The screenshot shows the 'Optimization' section of the application. It displays a table of scheduled tasks:

ID	Bezeichnung	Maschinen	Datum	Uhrzeit	Dauer (h)	Kalendereinladung
#1	Test	Lötbad3/4	19.12.2022	08:00	4	Kalendereintrag

Below the table is a note: "Die optimierte Planung der Termine führt zu einem Netzbezug in Höhe von 120.0 kWh. Somit kann 0.0 % der benötigten Strommenge durch die Photovoltaikanlage gedeckt werden." A download icon in the top right corner indicates a file named 'Termineinladung 1.ics' has been downloaded.

Abbildung 31: Release 2 (Download des erzeugten Terminvorschlags)

Als weitere Sicherheitsmaßnahme wurde die Anwendung über eine eigene URL adressierbar gemacht und durch SSL-Zertifikate, DNS-Konfiguration und User-Logins geschützt (vgl. Abbildung 32). Mitarbeiter können jetzt über ein Rollenkonzept mit unterschiedlichen Zugriffsrechten angelegt und verwaltet werden.



Abbildung 32: Release 2 (User-Logins)

Maßnahmen zur Verbesserung der Code Qualität umfassten die Einführung von SCSS zur besseren Strukturierung und Wartbarkeit sowie die Optimierung der Template-Struktur für mehr Übersicht. Zudem wurden die JavaScript Bibliotheken für die Visualisierungen lokal bereitgestellt, um externe Zugriffe zu reduzieren und die Anwendung schneller und sicherer zu machen.

Neben funktionalen Erweiterungen wurden auch erste Schritte in der Gestaltung der Anwendung unternommen. Das Design der Anwendung wurde responsiv gemacht, sodass die Anwendung auf verschiedenen Geräten verwendet werden kann (vgl. Abbildung 33). Auch wurden Farben und Formen angepasst, um eine modernere Optik zu schaffen, sowie Icons hinzugefügt, um die Navigation intuitiver zu gestalten. Zu diesem Zeitpunkt stand noch kein konkretes Design- und Inhaltskonzept fest. Das Designkonzept baut entsprechend dem Designprinzip „Form Follows Function“ auf dem Inhaltskonzept auf, für welches das Team erst gegen Ende des Sprints finale Informationen erhalten hat.

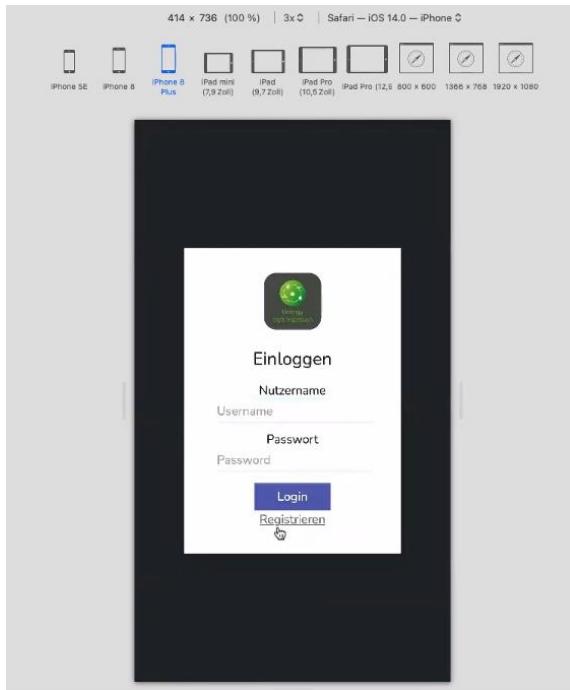


Abbildung 33: Release 2 (responsives Design)

Im zweiten Sprint hatte das Projektteam die Gelegenheit, direkt mit dem Auftraggeber der SEHO in Kontakt zu treten. In einem Onlinemeeting wurden dem Leiter der Forschungs- und Entwicklungsabteilung die Funktionen des Prototyps gezeigt und offene Fragen wurden geklärt. Dadurch erhielt das Team wichtige Informationen zum aktuellen Terminplanungsprozess zwischen Kunde und Mitarbeiter sowie zur Anzahl der Anwender und Kernbedingungen für die Optimierungsfunktion, z. B. die terminliche Verfügbarkeit von Mitarbeitern und Maschinen. Außerdem erhielt das Team detaillierte Daten zur geplanten PV-Anlage und realen Verbrauchsdaten von 2021 bis 2022. Die wichtigste Erkenntnis war, dass das Unternehmen keine Trennung der Energieverbraucher vornimmt, was die Ermittlung des Energiebedarfs einer Maschine nur aus ihrer angegebenen Leistung heraus erschwert. Insgesamt war das Meeting sehr informativ und ein Termin für weitere Gespräche und eine Besichtigung des Hauptsitzes wurde vereinbart.

Sprint 3: Implementierung Release 3: 22.12.2022 – 08.01.2023

Im dritten Sprint wurde das Ziel verfolgt, synthetische Daten durch reale Daten zu ersetzen, um die Anwendung für den aktiven Betrieb vorzubereiten. Hierfür mussten die benötigten Datenquellen identifiziert und technisch integriert werden. Dazu wurden drei Datenquellen benötigt: Energiedaten der PV-Anlage, aktuelle und prognostizierte Wetterdaten sowie Informationen zur Verfügbarkeit von Maschinen und Mitarbeitern. Durch die Integration der Daten über Schnittstellen konnte die Optimierungsfunktion als Kern der Anwendung realisiert werden.

Die Anbindung der PV-Anlage erfolgt über den IoT-Kommunikationsstandard OPC UA und ermöglicht den Abruf und die Übertragung von Energieinformationen wie erzeugter, genutzter oder ins Stromnetz eingespeister Energie (vgl. Abbildung 34). Obwohl die PV-Anlage der SEHO zu diesem Zeitpunkt noch nicht installiert war, wurde die Anwendung für die Integration der realen Daten vorbereitet und muss nur noch angebunden werden.



Abbildung 34: Release 3 (OPC UA Anbindung der PV-Anlage)

Die Wetterprognosedaten über einen Zeitraum von 30 Tagen erhält die Anwendung kostenfrei über die Integration der OpenWeatherMap-API. Sie liefert Informationen zu Sonneneinstrahlung, Sonnenstunden u. a. für die Optimierungsfunktion relevante Daten. Zur Nachvollziehbarkeit wurde die Anwendung um eine Unterseite erweitert, welche die abgerufenen Daten visualisiert (vgl. Abbildung 35).

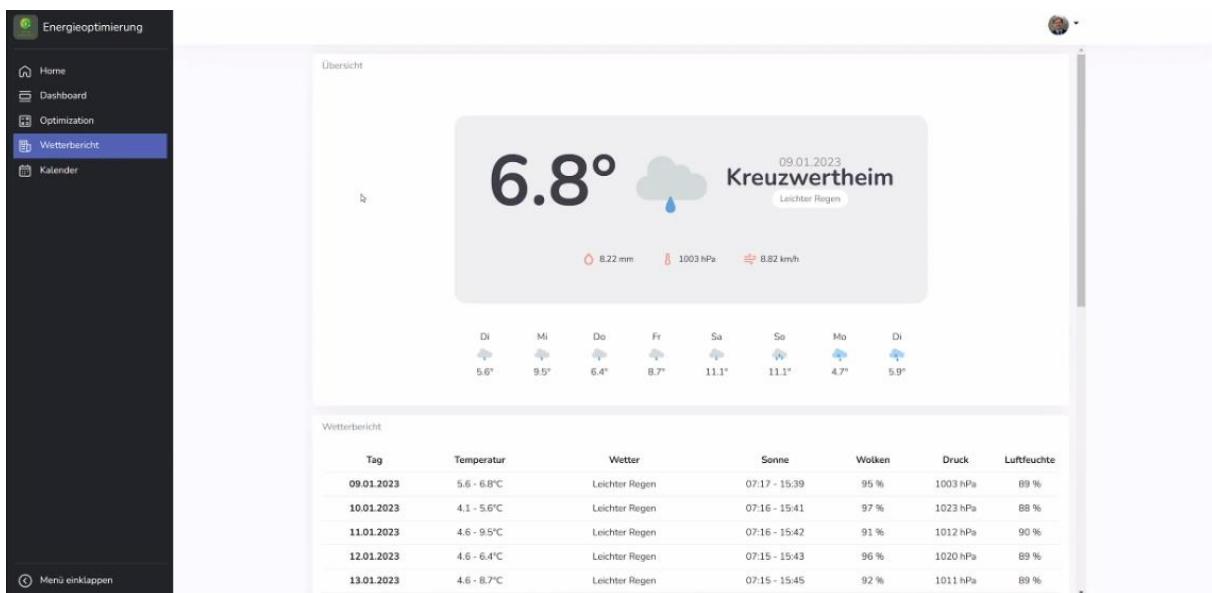


Abbildung 35: Release 3 (Integration von Wetterdaten)

Die OpenWeatherMap-API wird dafür jedoch nicht direkt von der Anwendung angesprochen. Stattdessen wurde ein Apache Kafka Streaming Server aufgesetzt, um die Wetterdaten einmal täglich abzurufen und über den Server an die Anwendung zu übergeben, sodass API-Aufrufe reduziert werden (vgl. Abbildung 36).



Abbildung 36: Release 3 (Anbindung Kafka-Streaming-Server)

Die Anbindung von Kalenderdaten für die Verfügbarkeit von Mitarbeitern und Maschinen ist für die Terminplanung besonders relevant. Dafür plant die SEHO für das Geschäftsjahr 2023 den Einsatz von Microsoft Office 365 in der Version E5. Das Projektteam hat einen Microsoft Office 365 E5 Developer Tenant angefragt, der Zugriff auf 17 Beispielaccounts mit eigenen Kalendern gibt. Die Anbindung der Kalenderinformationen erfolgt über die Microsoft Graph-API, sodass die Termine innerhalb der Anwendung angezeigt werden können. Durch die Integration von Mitarbeiter- und Maschinenkalendern können generierte Termine in die Outlook Kalender der beteiligten Mitarbeiter und Maschinen sowie in der Datenbank für Analysezwecke gespeichert werden. In der Anwendung können Mitarbeiter und Maschinen aus einem Auswahlmenü im jeweiligen Formularfeld ausgewählt und generierte Termine können innerhalb der Anwendung eingesehen (vgl. Abbildung 37 und Abbildung 38).

The screenshot shows a user interface for booking a machine. At the top, there are input fields for 'Beginn' (Start) and 'Ende' (End) with date pickers. Below these are two dropdown menus. The first dropdown, titled 'Maschinen', is currently empty. The second dropdown, titled 'Mitarbeiter' (Employees), is open and displays a list of names: Adele Vance (selected), Henrietta Mueller, Hannes Metz, Isaiah Langer, and Johanna Lorenz. A scroll bar is visible on the right side of the dropdown menu.

Abbildung 37: Erweiterung des Terminformulars um die Auswahl von Maschinen und Mitarbeitern

Events					
Datum	Startzeit	Endzeit	Titel	Beschreibung	Ort
20.01.2023	11:00	15:00	Test	Dieser Termin wurde von https://ebt-pjs.de generiert und blockiert die Maschine / den Mitarbeiter für einen Kundentermin	Seho Systems GmbH Showroom, Maschinen: Lötbad 3/4
23.01.2023	10:00	12:00	Testtermin	Dieser Termin wurde von https://ebt-pjs.de generiert und blockiert die Maschine / den Mitarbeiter für einen Kundentermin	Seho Systems GmbH Showroom, Maschinen: Wellenlöt, Lötbad 5
23.01.2023	15:00	17:00	Test	Dieser Termin wurde von https://ebt-pjs.de generiert und blockiert die Maschine / den Mitarbeiter für einen Kundentermin	Seho Systems GmbH Showroom, Maschinen: Wellenlöt, Lötbad 5

Abbildung 38: Release 3 (Übersicht über alle in der Anwendung generierten Termine)

Auf Basis der neuen Daten wurde ein Vorhersagemodell erstellt und das Optimierungsmodell unter Berücksichtigung der zusätzlichen Nebenbedingungen sowie der Terminbelegung von Mitarbeitern und Maschinen weiter verbessert.

Um Medienbrüche zu eliminieren und einen effizienteren Prozessablauf zu gewährleisten, wurde eine Funktion ergänzt, die den Versand der Termine direkt aus der Anwendung heraus ermöglicht. Zu diesem Zweck wurde ein eigener Mail Server eingerichtet (vgl. Abbildung 39 ff).

Energiebasierte Terminoptimierung

The screenshot shows the software's main interface with a sidebar menu on the left. The menu items include Home, Dashboard, Optimization (which is highlighted in blue), Wetterbericht, and Kalender. The main content area is titled "Optimization" and displays the following information:

Hier sind die optimalen Zeitpunkte für Ihre Termine:

ID	Bezeichnung	Maschinen	Datum	Uhrzeit	Dauer (h)	Einladung
#1	Test	Wetlenöt	01.09.2022	13:00	2	herunterladen senden

Stromverbrauch der Termine
Die Planung aller Termine führt zu einem Stromverbrauch in Höhe von 100 kWh.

Netzbezug der Termine
Termin ID: #1, Strom durch PV: 17.0 %

Anteil PV-Strom
Es kann 17.0 % der benötigten Strommenge durch die Photovoltaikanlage gedeckt werden.

Abbildung 39: Release 3 (Integration Emailversand)

The screenshot shows the software's main interface with a sidebar menu on the left. The menu items include Home, Dashboard, Optimization (highlighted in blue), Wetterbericht, and Kalender. A modal dialog box titled "Termineinladung per Mail" is open in the center. It contains the following fields:

Mailadressen: hannes.metz@gmx.de
Mehrere Empfänger werden mit einem Komma getrennt.

Mail-Text:
Bitte nehmen Sie den Termin an

Absenden

The background of the interface shows the same optimization results as in Abbildung 39.

Abbildung 40: Release 3 (Versand des erzeugten Terms)

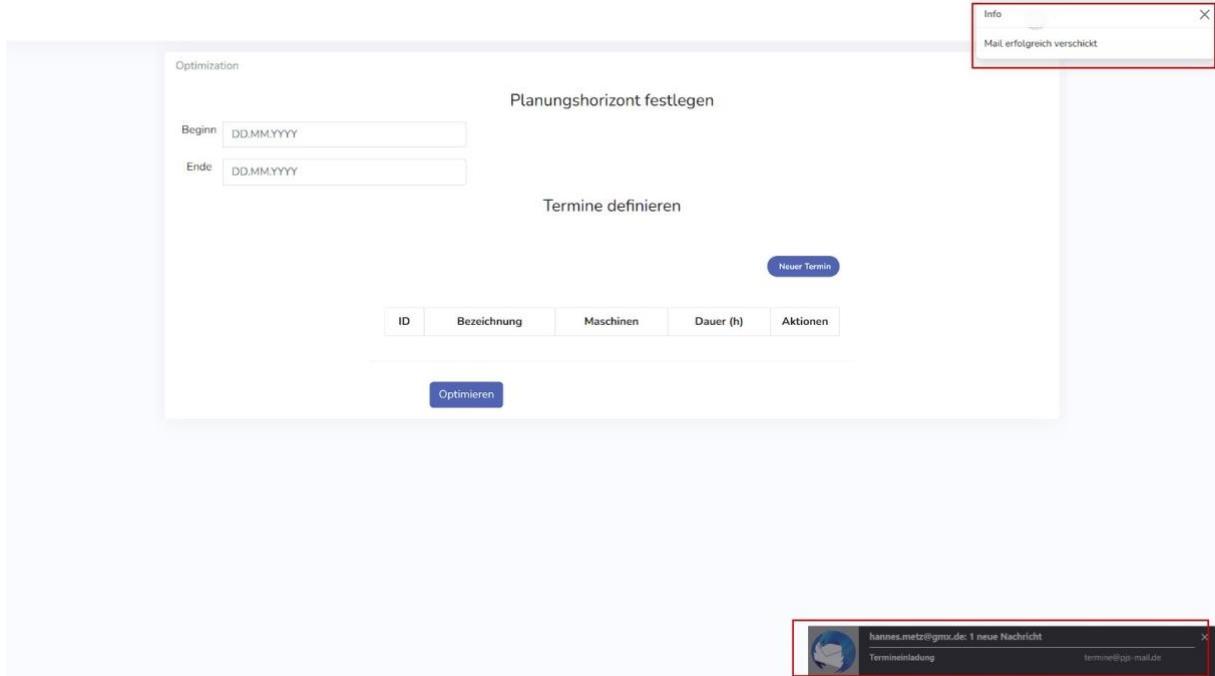


Abbildung 41: Release 3 (Versandbestätigung)

Die Erweiterung um die genannten Schnittstellen erforderte ein Refactoring im Backend. Für eine verbesserte Struktur des Codes wurden Python-Files in einzelne Routen aufgeteilt.

Außerdem erweiterte das Team das Sicherheitskonzept. Die Login Seite zeigt nun Fehlermeldungen, wenn ein Benutzername oder Passwort falsch eingegeben wurde (vgl. Abbildung 42).

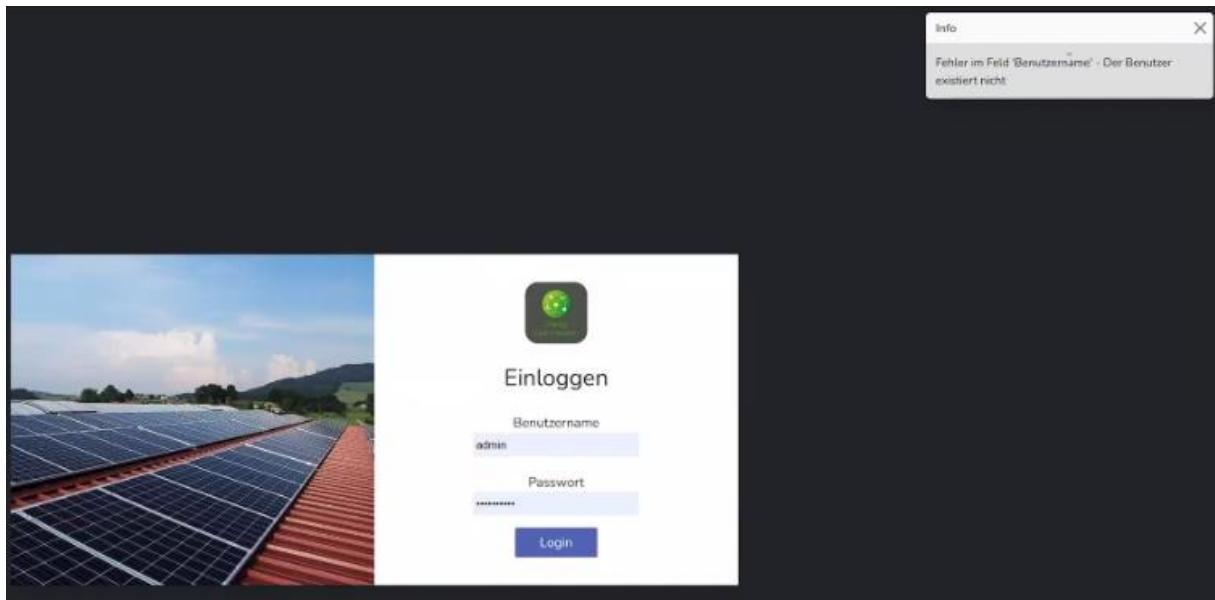


Abbildung 42: Release 3 (Fehlermeldung bei falschen oder nicht existenten Nutzerdaten)

Zudem wurden neue Seiten für die Verwaltung von Nutzern, Rollen und Rechten erstellt (vgl. Abbildung 43). Ein Rollenkonzept unterscheidet zwischen Standard- und

Administrationszugriff. Administratoren haben erweiterten Zugriff auf die Einstellungsseite, wo sie einen Überblick über alle Benutzer der Anwendung erhalten und diese verwalten oder löschen können. Dadurch wird die Qualität und Sicherheit des Systems verbessert, da Benutzer nicht mehr eigenständig einen Account erstellen und sofort Zugriff auf die Anwendung erhalten können.

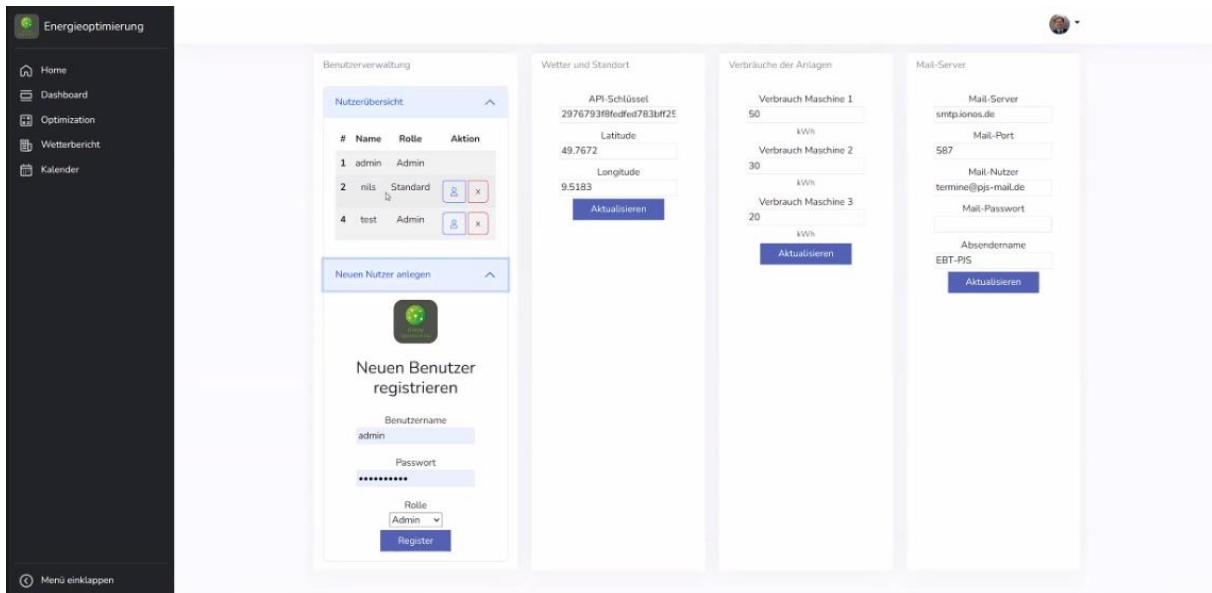


Abbildung 43: Release 3 (Erweiterung um Administrationsbereich)

Die Settings Seite ermöglicht es, unternehmensbezogene Daten wie Wetter, Standort, Mail-Server, PV-Anlage und andere Daten für die Integration mit anderen Systemen bereitzustellen. Diese Funktion ist wichtig, da sie eine Anpassung an die Anforderungen des jeweiligen Anwenderunternehmens ermöglicht. Das System ist auch hier mit einer Fehlermeldung bei Falscheingaben ausgestattet. Die Profil-Seite wurde als Selfservice-Oberfläche gestaltet und erlaubt die Anpassung von Profilbild, Benutzername und Passwort durch den jeweiligen Benutzer.

Abschließend wurden die Visualisierungen in der Anwendung verbessert, um die Auswirkungen der Terminplanungsoptimierung besser darzustellen. Es gibt jetzt eine Anzeige des gesamten Energieverbrauchs für jeden Terminvorschlag, einschließlich der Anteile von Eigenstrom und Netzstrom. Dies hilft Mitarbeitern bei der Auswahl eines Terms, indem es ihnen ermöglicht, den Energieverbrauch der verschiedenen Optionen direkt zu vergleichen. Durch diese zusätzliche Information wird die Transparenz der Anwendung verbessert.

Sprint 4: Implementierung Release 4: 09.01.2023 – 24.01.2023

Im Sprint 4 des Projekts wurden drei Ziele verfolgt: die Erstellung eines Increments auf Basis des Backlogs, die Planung und Vorbereitung eines Besuchs beim Auftraggeber SEHO und die Vorbereitung einer Berichtsstruktur für das Projektende.

In der operativen Entwicklung überführte das Team das Formular für die Terminplanung mit Hilfe von Flask Forms von einem reinen Frontend-gesteuerten HTML-Formular vollständig ins Backend, um den Code schlanker und übersichtlicher zu gestalten. Bis zu diesem Zeitpunkt war lediglich die Aufforderung „input required“ im Frontend hinterlegt. Durch die Implementierung von WTForms und der Validierung im Backend können Termine nun korrekt generiert werden und fehlerhafte Eingaben vermieden werden. Die Umsetzung erfolgte mit Jinja2 Markup Language, einem Templatesystem, das von Flask eingebunden wird.

Das Team hat auch an der Gestaltung der Anwendung gearbeitet, indem es ein Date Picker als Designelement (vgl. Abbildung 44) und eine Dropdown-Auswahl für Mitarbeiter und Maschinen (vgl. Abbildung 45) hinzugefügt hat, bei der auch das Entfernen von individuellen Elementen möglich ist.

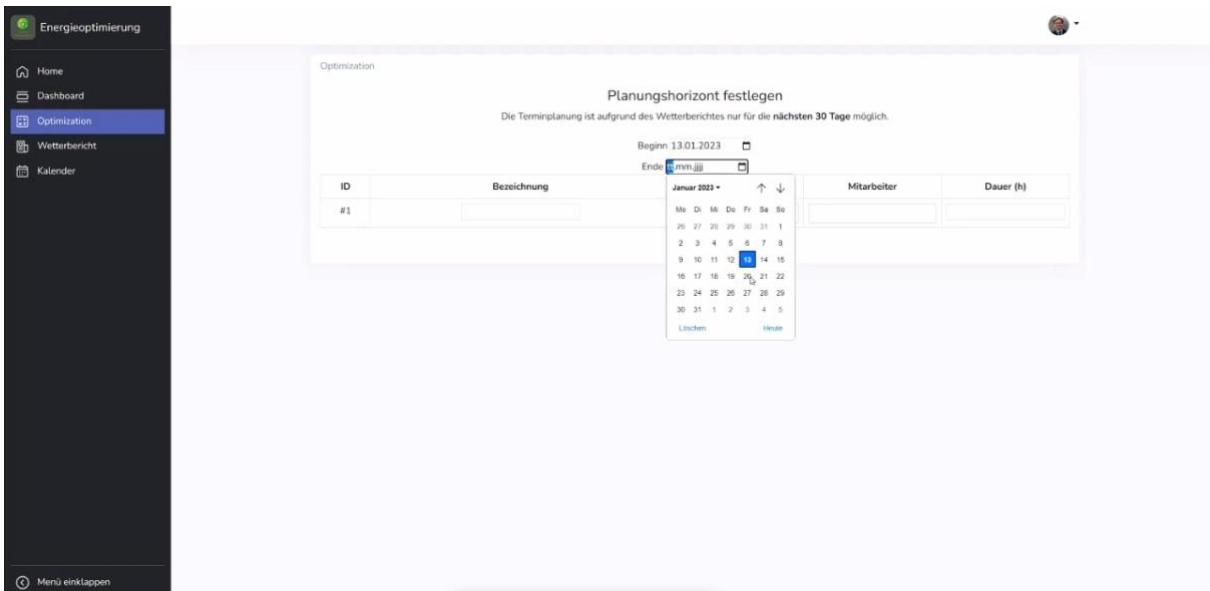


Abbildung 44: Release 4 (Date Picker im Formular)

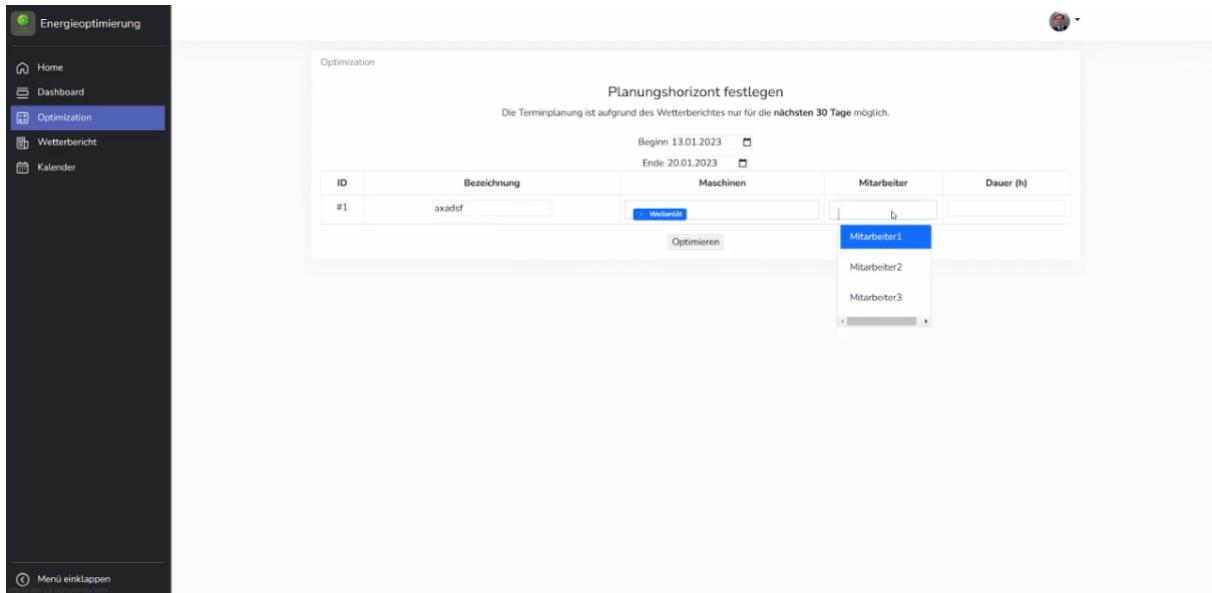


Abbildung 45: Release 4 (Dropdown-Auswahl für Mitarbeiter und Maschinen)

Die Umsetzung weiterer Backlog Items wurde zurückgehalten, da das Team das Treffen mit dem Auftraggeber vor Ort und die Beantwortung offener Fragen abwarten wollte.

Sprint 5: Implementierung Release 5: 25.01.2023 – 06.02.2023

Das Ziel des fünften Sprints bestand darin, die letzten funktionalen Anforderungen der Entwicklungsphase 1 zu implementieren, bevor in Entwicklungsphase 2 mit der Ausgestaltung des Frontends begonnen wird. Das Projektteam hatte im vorherigen Sprint die Gelegenheit, den Auftraggeber in seinem Hauptsitz zu besuchen und somit wichtige Einblicke und Informationen zu gewinnen, um den Anforderungskatalog für das Product Backlog zu finalisieren und ein endgültiges Inhaltskonzept zu entwickeln. Außerdem wurde ein Feedback Loop nach der Hälfte des laufenden Sprints durchgeführt, bei dem externe Projektpartner letztes Feedback zur funktionalen Entwicklung einfließen ließen.

Insbesondere wurde das Feedback zur Anpassung der Optimierungsfunktion verwendet, um die Berechnung des Energieverbrauchs zu vereinfachen und die tatsächlich verwendeten Maschinen zu identifizieren. Bei der Berechnung kann die Grundlast vernachlässigt werden, da diese die Grundlast in erster Linie durch die nahezu gleichbleibende Beleuchtung des Ausstellungsraums verursacht wird und somit keine Energiespikes zu berücksichtigen sind. Dies vereinfacht das Optimierungsmodell, da nun keine Regression der Stromverbräuche notwendig ist. Die Maschinenleistung hingegen ist abhängig von der Komplexität des zu lögenden Produkts, weshalb die Optimierungsfunktion um weitere Einschränkungen ergänzt wurde (vgl. Abbildung 46). Auch konnten die tatsächlich verwendeten Maschinen identifiziert und über individuelle Outlook Kalender abgebildet werden. Dabei wird nun zwischen

Maschinen im Ausstellungsraum und den produzierten verkaufsfertigen Maschinen in der Qualitätssicherung unterschieden.

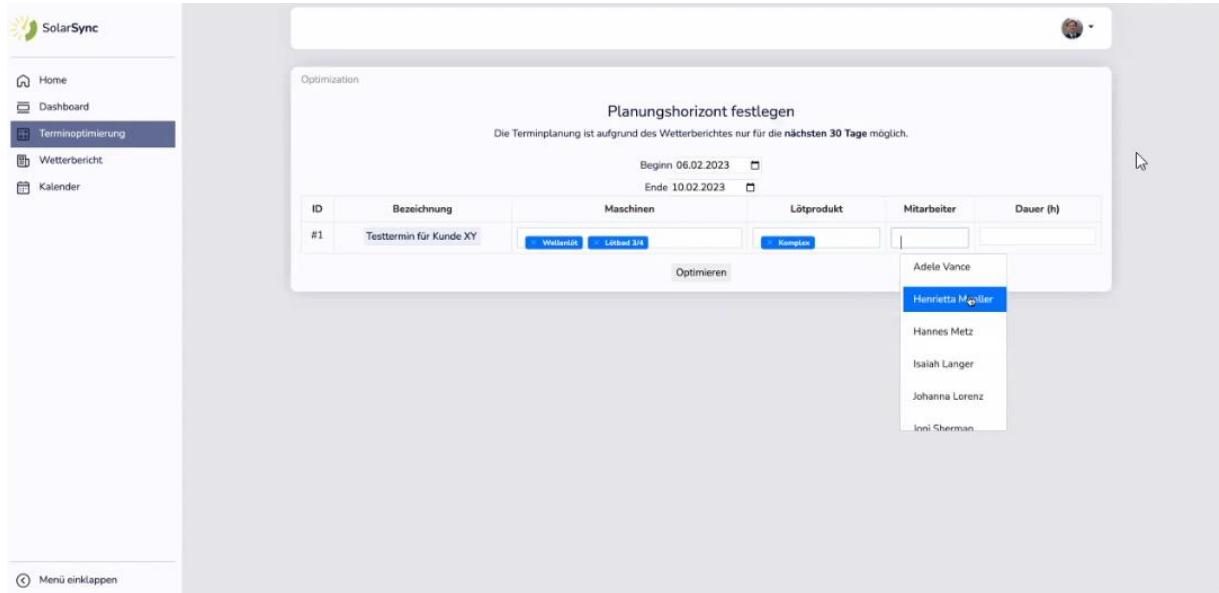


Abbildung 46: Release 5 (Erweiterung der Terminplanung um das Formularfeld Lötprodukt)

Darüber hinaus findet eine Unterscheidung zwischen Aufheizenergieverbräuchen und normalen Energieverbräuchen statt. Das Optimierungsmodell plant Termine nun so, dass Maschinen direkt eingeschaltet bleiben können, wodurch eine erhebliche Energieeinsparung erfolgt.

Damit die Anwendung nicht zu einem Nischenprodukt wird, wurden alle Funktionen modular aufgebaut. Die Einstellungen sind im Administrationsbereich individuell anpassbar (vgl. Abbildung 47). Auch die Integration der Anwendung in externe Systeme wurde in diesem Zug vom Projektteam berücksichtigt. Die Webanwendung wurde mit einem API-Layer ausgestattet, sodass alle Funktionen der Anwendung wie das Starten von Maschinen, die Terminoptimierung oder das Speichern von Terminen in den jeweiligen Kalendern mit GET und POST Requests implementiert werden können. Die REST API wurde aus Sicherheitsgründen mit einer Token-Authentifizierung ausgestattet.

Energiebasierte Terminoptimierung

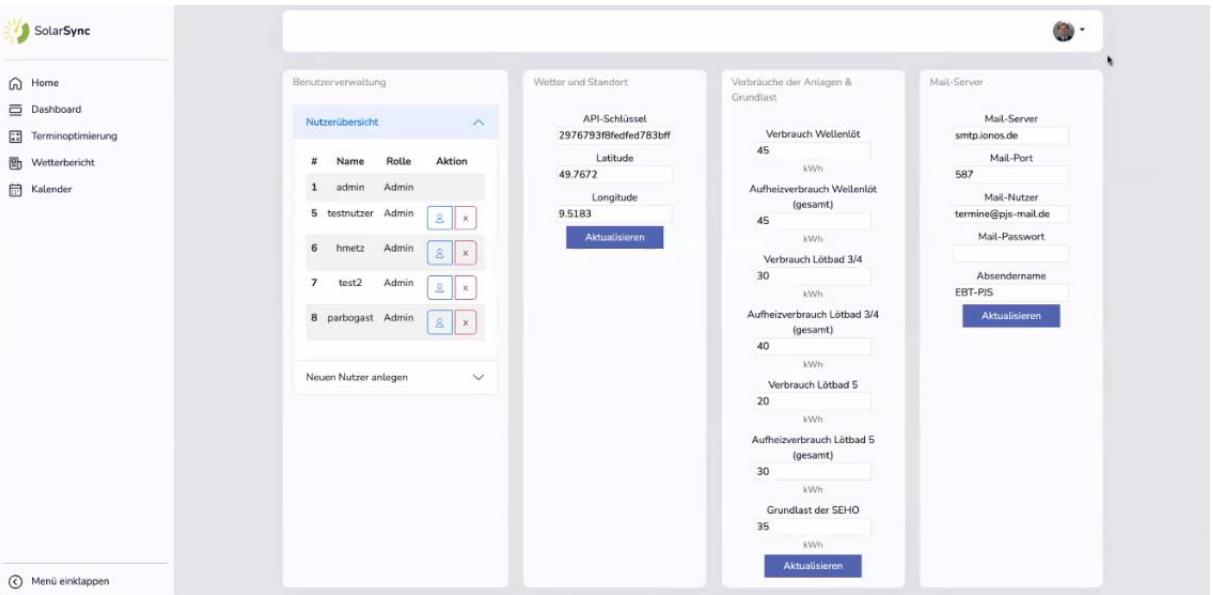


Abbildung 47: Release 5 (modularer Aufbau von SolarSync)

Weiterhin hat sich aus den Gesprächen mit den Vertretern der SEHO ergeben, dass alle Maschinen im Ausstellungsraum OPC UA-fähig sind. Somit können alle aktuellen Leistungsdaten wie Temperatur u. a. ausgelesen werden. Bislang ist jedoch kein Data Warehouse angeschlossen, in dem Daten geloggt werden können. Dennoch hat das Projektteam die OPC UA Schnittstelle technologisch hinterlegt, sodass in späteren Iterationen ein Zugang vorhanden ist (vgl. Abbildung 48).

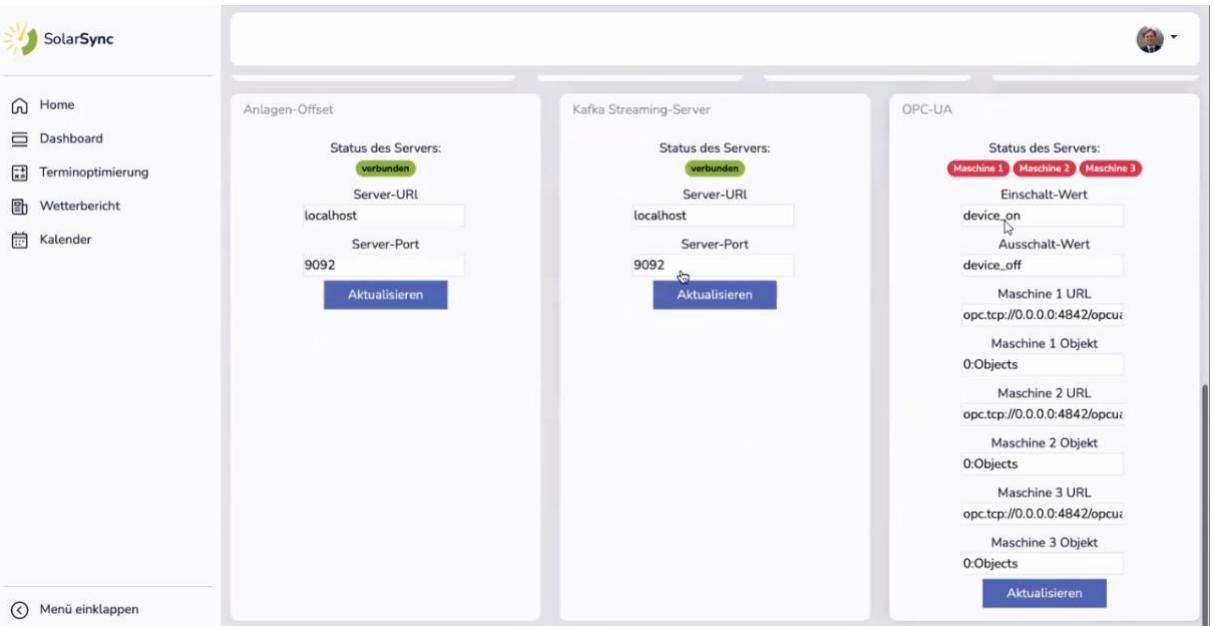


Abbildung 48: Release 5 (OPC UA Anbindung der Maschinen)

Im Frontend wurden strukturelle Anpassungen vorgenommen. So ist die ursprüngliche Home-Seite zu einem klassischen Dashboard umgewandelt worden, wobei die Menüführung zu diesem Zeitpunkt noch nicht angeglichen war (vgl. Abbildung 49).

Energiebasierte Terminoptimierung

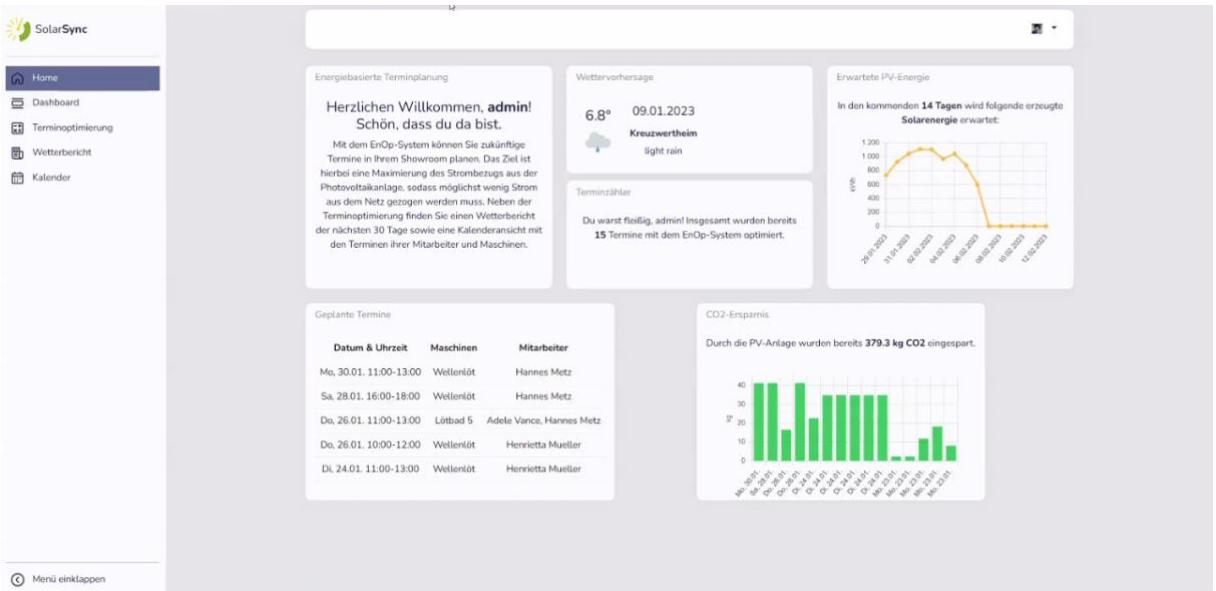


Abbildung 49: Release 5 (erstes Dashboard Layout)

Außerdem wurden Custom Error Handling Seiten für verschiedene Fehlermeldungen hinterlegt, um die Arbeit mit der Anwendung für den User angenehm ohne viel Fließtext zu gestalten (vgl. Abbildung 50).



Abbildung 50: Release 5 (Custom Error Handling)

Auf Basis des erarbeiteten Inhaltskonzepts wurde ein Designkonzept geschaffen, welches in Vorbereitung für die Umsetzung in Entwicklungsphase 2 Mockups für alle vorgesehenen Seiten beinhaltet. Im Zuge der Layoutgestaltung entwarf das Projektteam ein Branding für die Anwendung mit hohem Wiedererkennungswert (vgl. Abbildung 22).

Sprint 6: Implementierung Release 6: 07.02.2023 – 21.02.2023

Mit Sprint 6 begann die im Phasen-Meilenstein-Plan geplante Entwicklungsphase 2, in der aufgrund der anstehenden Klausurenphase keine weiteren funktionalen Anforderungen umgesetzt werden. Lediglich diverse kleinere Bug Fixes wurden im funktionalen Bereich ausgeführt. Hauptziel des Sprints war die Umsetzung der erarbeiteten Inhalts- und Designkonzepte in Anlehnung an die in Sprint 5 erstellten Mockups. Dementsprechend arbeitete das Projektteam im Frontend der Anwendung, wo das Framework Bootstrap eingesetzt und mit Hilfe von SCSS ein einheitliches und konsistentes Design für die Benutzeroberfläche erarbeitet wurde. Der Fokus lag dabei in erster Linie auf der Überarbeitung des Dashboards (vgl. Abbildung 10) als Startseite der Anwendung sowie auf der Anpassung der Unterseite zur Terminoptimierung (vgl. Abbildung 11 ff). Darüber hinaus wurden die Unterseiten zu Kalender (vgl. Abbildung 16) und Wetterbericht (vgl. Abbildung 15) an das globale Design der Anwendung angeglichen.

Sprint 7: Implementierung finales Release: 22.02.2023 – 07.03.2023

Im letzten Sprint hat das Projektteam das Frontend finalisiert und eine Unterseite zur Auswertung der Daten der PV-Anlage hinzugefügt (vgl. Abbildung 17). Hierzu wurden gezielte Abfragen von Daten der PV-Anlage hinterlegt und verschiedene Grafiken mit JavaScript und ChartJS erstellt, um diese Informationen leicht verständlich und auf einen Blick visualisieren zu können. Die Seiten zur Profileinstellung (vgl. Abbildung 18) und Systemadministration (vgl. Abbildung 19) wurden an das globale Design der Anwendung angepasst und das Icon der Anwendung in den Tab des Browsers integriert (vgl. Abbildung 51), um den Wiedererkennungswert des Tools zu erhöhen und das Erscheinungsbild konsistent zu gestalten.

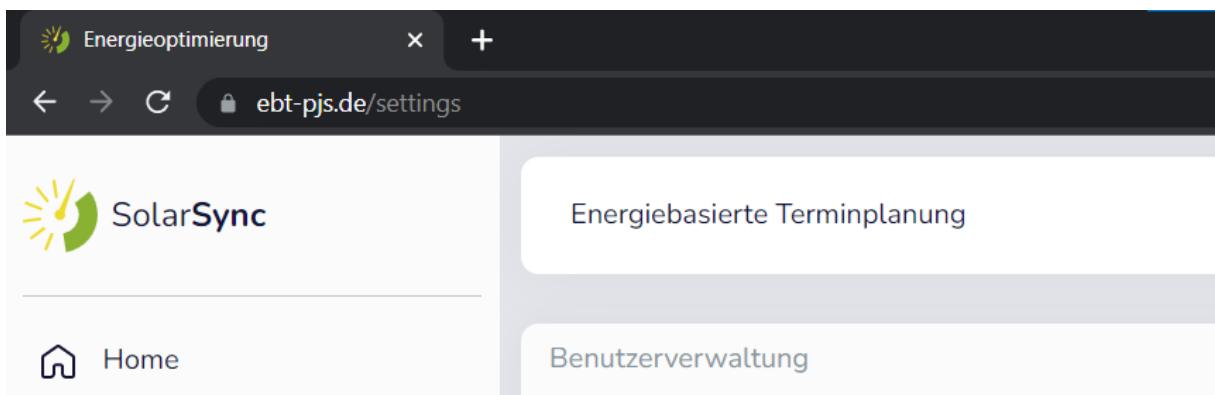


Abbildung 51: Finales Release (SolarSync Icon im Tab des Browsers)

Anhang 6 – Design Mockups

In diesem Abschnitt werden die Mockups zur Verfügung gestellt, die zur Vorbereitung und Orientierung der Gestaltung der einzelnen Seiten der Anwendung SolarSync erstellt wurden (vgl. Abbildung 52 ff). Die Entwürfe wurden mit Adobe Photoshop gestaltet.

The dashboard features a sidebar with navigation links: Home, Dashboard, Terminoptimierung, Wetterbericht, and Kalender. The main area displays:

- Real-time clock:** 9:24 FR. Jan. 13
- Upcoming appointment:** 70:36:00
- Weather forecast for Kreuzwertheim:** 1.5°C, 0.3 mm, 1033 hPa, 6.51 km/h
- PV system status:** Auslastung PV-Anlage: 63%, Erwartete PV-Energie: 14 Tage Aussicht: 11.578 kWh
- CO2 savings:** CO2 Ersparnis: 29.3 kg heute, 329.3 kg gesamt

Abbildung 52: Mockup Dashboard

The sidebar shows the same navigation as in Abbildung 52. The main area displays:

- Real-time clock:** 9:24 FR. Jan. 13
- Upcoming appointment:** 70:36:00
- Calendars:**
 - Maschinen:** Terminübersicht Maschinen, nächster Termin: 20.01.2023 10:00 - 13:00, Wellenlöt
 - Mitarbeiter:** Terminübersicht Mitarbeiter, nächster Termin: 20.01.2023 10:00 - 13:00, Hannes Metz, Nils Heilemann
- Scheduled appointments table:**

KW	Datum	Uhrzeit	Dauer	Mitarbeiter	Maschinen	Titel
4	20.01.2023	10:00 - 13:00	3h	Hannes Metz Philipp Arbogast	Wellenlöt	Kundendemo Max Mustermann, Muster GmbH
4	20.01.2023	13:00 - 15:00	2h	Hannes Metz	Lötbad 3/4	Kundendemo Lisa Müller, BSP AG
5	26.01.2023	13:00 - 17:00	4h	Nils Heilemann	Lötbad 3/4 Lötbad 5	Kundendemo Robin Freitag, zB Gbr.
5	27.01.2023	11:00 - 15:00	4h	Louis Mauser Christina Cser	Wellenlöt Lötbad 5	Kundendemo Henry Jones, Indiana Inc.
6	01.02.2023	12:00 - 15:00	3h	Louis Mauser	Wellenlöt	Kundendemo Bruce Wayne

Abbildung 53: Mockup Kalender

Energiebasierte Terminoptimierung

The screenshot shows the 'Wetterbericht' (Weather Report) section of the application. At the top, there's a header with the SOLAR SYNC logo, the title 'ENERGIEBASIERTE TERMINPLANUNG', and a user profile icon. Below the header, there's a weather summary for 'Kreuzwertheim' showing current conditions: 1.5°C, Regen und Schnee, 0.3 mm precipitation, 1033 hPa pressure, and 6.51 km/h wind. A 10-day forecast table follows, showing temperatures, weather, and other metrics for each day from Friday (FR) to Thursday (DO). Below the table is a detailed 24-hour forecast for Friday, including temperature, weather icons, and specific times for sunrise, sunset, and cloud cover.

Tag	Datum	Temperatur	Wetter	Tageslicht	Wolken	Luftdruck	Luftfeuchte	Niederschlag
FR	20.01.2023	-0.5 - 1.5°C	Regen und Schnee	07:06 - 16:00	92%	1033 hPa	92%	0.3 mm
SA	20.01.2023	0.8 - 2.1°C	Regen	07:05 - 16:01	97%	1037 hPa	94%	0.5 mm
SO	26.01.2023	0.4 - 2.4°C	Regen	07:04 - 16:03	74%	1032 hPa	90%	0.4 mm
MO	27.01.2023	-0.9 - 2.1°C	Regen und Schnee	07:03 - 16:05	88%	1020 hPa	86%	0.3 mm
DI	01.02.2023	-1.9 - 1.6°C	Regen und Schnee	07:02 - 16:06	78%	1022 hPa	77%	0.2 mm
MI	02.02.2023	-1.5 - 1.2°C	Schnee	07:01 - 16:07	80%	1030 hPa	80%	0.3 mm

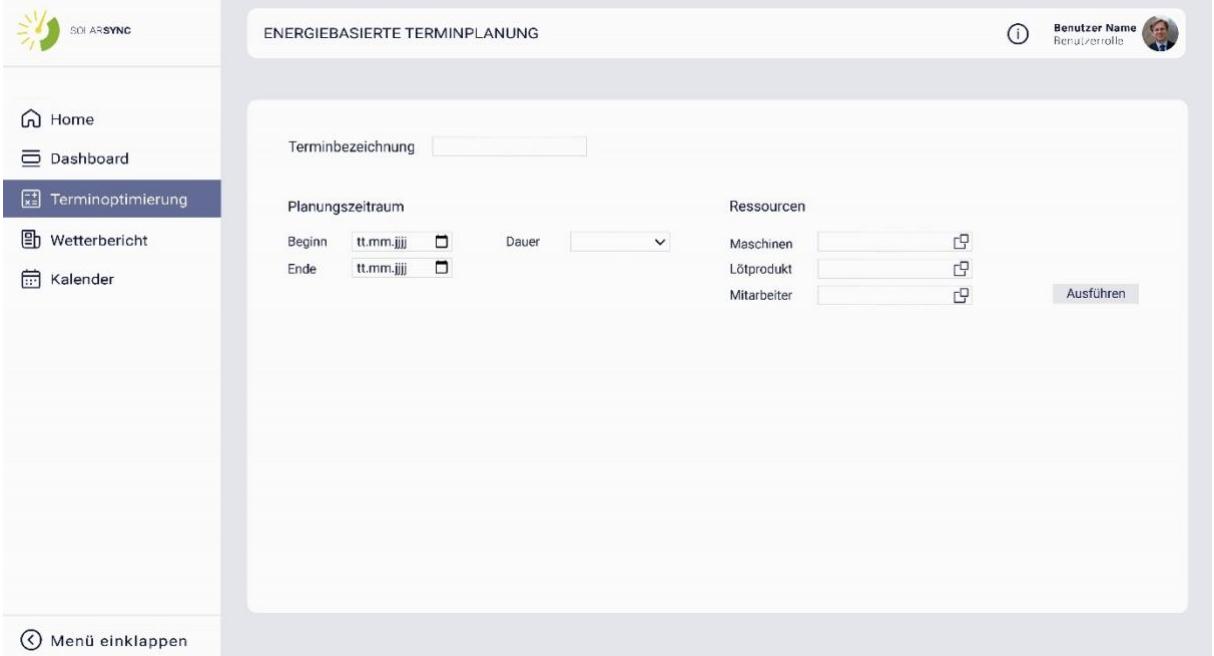
Below the table is a 24-hour forecast for Friday, showing temperature, weather icons, and specific times for sunrise, sunset, and cloud cover.

Abbildung 54: Mockup Wettervorhersage

The screenshot shows the 'PV-Anlage' (PV System) section of the application. At the top, there's a header with the SOLAR SYNC logo, the title 'ENERGIEBASIERTE TERMINPLANUNG', and a user profile icon. Below the header, there are two main sections: 'Auslastung PV-Anlage' (PV System Load) and 'Erwartete PV-Energie 14 Tage Aussicht' (Expected PV Energy 14-day Outlook). The load section features a large green gauge showing 63% load. The outlook section shows a graph of expected energy production over 14 days, with a peak of 11.578 kW. To the right, there are two more gauges: 'Autarkiegrad' (Autarky Degree) at 72% and 'Versorgungsgrad' (Supply Degree) at 49%. Below these gauges are two line graphs showing energy production, consumption, and basic consumption over time. The bottom of the screen has a 'Menü einklappen' (Collapse menu) button.

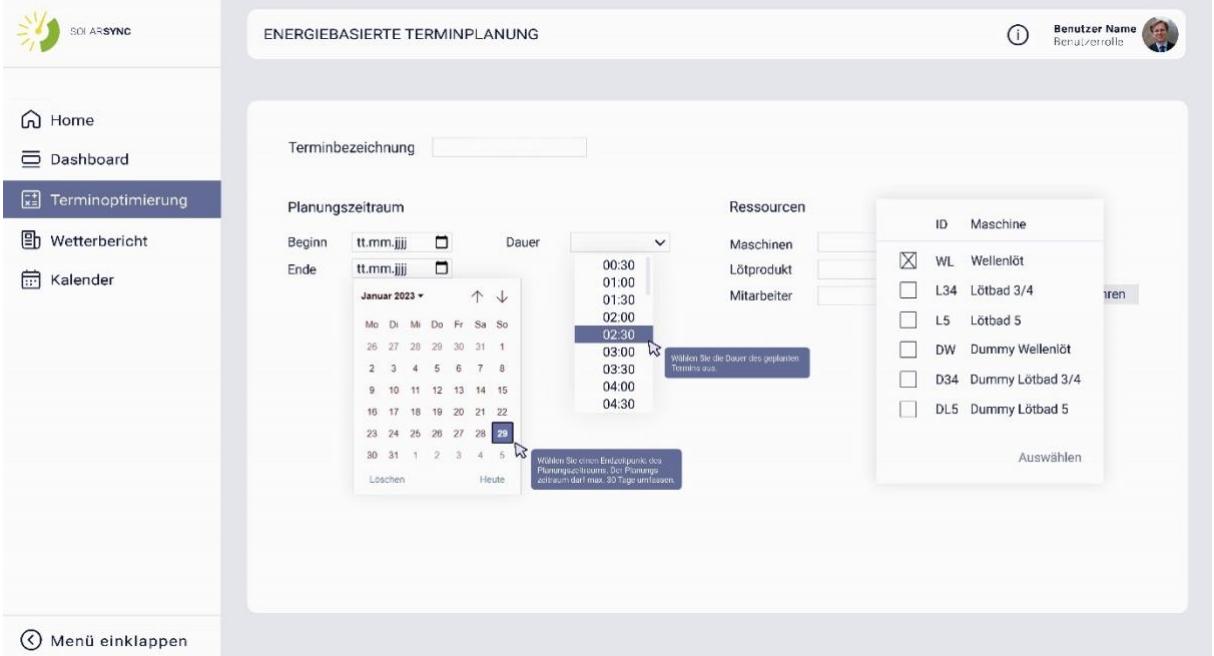
Abbildung 55: Mockup PV-Anlage

Energiebasierte Terminoptimierung



The screenshot shows a user interface for 'ENERGIEBASIERTE TERMINPLANUNG'. On the left is a sidebar with a logo and navigation links: Home, Dashboard, Terminoptimierung (which is selected), Wetterbericht, and Kalender. A 'Menü einklappen' button is at the bottom. The main area has a title 'ENERGIEBASIERTE TERMINPLANUNG' and sections for 'Terminbezeichnung' (Task name), 'Planungszeitraum' (Planning period) with fields for 'Beginn' (Start) and 'Ende' (End), and 'Dauer' (Duration), and 'Ressourcen' (Resources) for 'Maschinen' (Machines), 'Lötprodukt' (Solder product), and 'Mitarbeiter' (Employees). An 'Ausführen' (Execute) button is at the bottom right.

Abbildung 56: Mockup Terminplanung (Formular ohne Dialog- und Auswahlfenster)



This screenshot shows the same application interface as Abbildung 56, but with more detailed views of dialog boxes. In the 'Planungszeitraum' section, a date picker for January 2023 is open, showing days from 26 to 31. The number '29' is highlighted with a blue selection box. A tooltip below the calendar says: 'Wählen Sie einen Endzeitpunkt des Planungszeitraums; der Planungszeitraum darf max. 30 Tage umfassen.' To the right, a resource selection dialog lists machines: WL (Wellenlöten), L34 (Lötbad 3/4), L5 (Lötbad 5), DW (Dummy Wellenlöten), D34 (Dummy Lötbad 3/4), and DL5 (Dummy Lötbad 5). The 'WL' checkbox is checked. A 'Wählen Sie die Dauer des geplanten Termins aus.' (Select the duration of the planned term) button is visible. At the bottom right of the dialog, there's an 'Auswählen' (Select) button.

Abbildung 57: Mockup Terminplanung (Formular mit Dialog- und Auswahlfenster)

Energiebasierte Terminoptimierung

SOLAR SYNC

ENERGIEBASIERTE TERMINPLANUNG

Benutzer Name
Benutzerrollen

Home

Dashboard

Terminoptimierung

Wetterbericht

Kalender

Menü einklappen

Terminvorschläge für Termin [Terminbezeichnung]

Terminvorschlag	Maschinen	Mitarbeiter	Datum	Uhrzeit	Dauer	Einladung	
#1	WL	HM	30.01.2023	13:00	4h		
#1	WL	HM	30.01.2023	13:00	4h		
#1	WL	HM	30.01.2023	13:00	4h		
#1	WL	HM	30.01.2023	13:00	4h		
#1	WL	HM	30.01.2023	13:00	4h		

Abbrechen

Erwartete PV-Energie im Planungszeitraum
11.578 kW/h



Erwarteter Energieverbrauch

Termin	Netzbezug	PV-Energie	PV-Anteil
#1	30,0 kWh	60,0 kWh	66,67 %
#2	36,0 kWh	54,0 kWh	60,00 %
#3	38,7 kWh	51,3 kWh	57,00 %
#4	50,8 kWh	39,2 kWh	43,50 %
#5	75,4 kWh	24,6 kWh	27,30 %

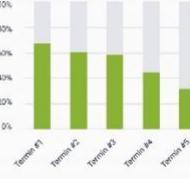


Abbildung 58: Mockup Terminplanung (Terminausgabe mit Auswertungen)

Anhang 7 – Simulation der Photovoltaikanlage

ENV 2022_Simulation_ENVIRIA & SEHO Systems GmbH_20221118

Enviria Energy



Projektübersicht



Abbildung: Übersichtsbild, 3D-Planung

PV-Anlage

3D, Netzgekoppelte PV-Anlage mit elektrischen Verbrauchern

Klimadaten	Kreuzwertheim, DEU (1996 - 2015)
Quelle der Werte	Meteonorm 8.1(i)
PV-Generatorleistung	635,8 kWp
PV-Generatorfläche	2.921,3 m ²
Anzahl PV-Module	1496
Anzahl Wechselrichter	8

ENV 2022_Simulation_ENVIRIA & SEHO Systems GmbH_20221118

Enviria Energy



Simulationsergebnisse

Ergebnisse Gesamtanlage

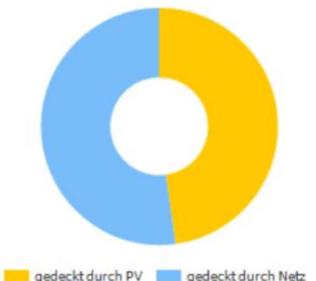
PV-Anlage

PV-Generatorleistung		PV-Generatorenergie (AC-Netz)
PV-Generatorleistung	635,80 kWp	
Spez. Jahresertrag	1.029,90 kWh/kWp	
Anlagennutzungsgrad (PR)	92,86 %	
Ertragsminderung durch Abschattung	0,9 %	
PV-Generatorenergie (AC-Netz)	654.971 kWh/Jahr	
Eigenverbrauch	331.335 kWh/Jahr	
Abregelung am Einspeisepunkt	0 kWh/Jahr	
Netzeinspeisung	323.636 kWh/Jahr	
Eigenverbrauchsanteil	50,6 %	
Vermiedene CO ₂ -Emissionen	307.761 kg/Jahr	



Verbraucher

Gesamtverbrauch	
Verbraucher	693.120 kWh/Jahr
Standby-Verbrauch (Wechselrichter)	161 kWh/Jahr
Gesamtverbrauch	693.281 kWh/Jahr
gedeckt durch PV	331.335 kWh/Jahr
gedeckt durch Netz	361.946 kWh/Jahr
Solarer Deckungsanteil	47,8 %



Autarkiegrad

Gesamtverbrauch	693.281 kWh/Jahr
gedeckt durch Netz	361.946 kWh/Jahr
Autarkiegrad	47,8 %

ENV 2022_Simulation_ENVIRIA & SEHO Systems GmbH_20221118

Enviria Energy



Screenshots, 3D-Planung

Umgebung



Abbildung: Screenshot03



Abbildung: Screenshot02

Energiebilanz PV-Anlage

Energiebilanz PV-Anlage

Globalstrahlung horizontal	1.131,50 kWh/m²	
Abweichung vom Standardspektrum	-11,31 kWh/m ²	-1,00 %
Bodenreflexion (Albedo)	1,72 kWh/m ²	0,15 %
Ausrichtung und Neigung der Modulebene	-12,47 kWh/m ²	-1,11 %
Modulunabhängige Abschattung	-0,58 kWh/m ²	-0,05 %
Reflexion an Moduloberfläche	0,00 kWh/m ²	0,00 %
Globalstrahlung auf Modul	1.108,85 kWh/m²	
	1.108,85 kWh/m ²	
	x 2921,311 m ²	
	= 3.239.299,14 kWh	
PV Globalstrahlung	3.239.299,14 kWh	
Verschmutzung	0,00 kWh	0,00 %
STC Konversion (Modul-Nennwirkungsgrad 21,77 %)	-2.534.130,65 kWh	-78,23 %
PV Nennenergie	705.168,48 kWh	
Modulspezifische Teilabschattung	-4.014,09 kWh	-0,57 %
Schwachlichtverhalten	1.812,90 kWh	0,26 %
Abweichung von der Nenn-Modultemperatur	-8.419,51 kWh	-1,20 %
Dioden	-246,24 kWh	-0,04 %
Mismatch (Herstellerangaben)	-13.886,03 kWh	-2,00 %
Mismatch (Verschaltung/Abschattung)	-1.454,37 kWh	-0,21 %
PV-Energie (DC) ohne Wechselrichter-Abregelung	678.961,15 kWh	
Unterschreitung der DC-Startleistung	-4,90 kWh	0,00 %
Abregelung wegen MPP-Spannungsbereich	-3,76 kWh	0,00 %
Abregelung wegen max. DC-Strom	-0,31 kWh	0,00 %
Abregelung wegen max. DC-Leistung	0,00 kWh	0,00 %
Abregelung wegen max. AC-Leistung/cos phi	-510,98 kWh	-0,08 %
MPP Anpassung	-63,90 kWh	-0,01 %
PV-Energie (DC)	678.377,29 kWh	
Energie am WR-Eingang	678.377,29 kWh	
Abweichung der Eingangs- von der Nennspannung	-130,08 kWh	-0,02 %
DC/AC-Wandlung	-13.302,06 kWh	-1,96 %
Standby-Verbrauch (Wechselrichter)	-160,97 kWh	-0,02 %
Kabelverluste Gesamt	-9.974,18 kWh	-1,50 %
PV-Energie (AC) abzgl. Standby-Verbrauch	654.810,01 kWh	
PV-Generatorennergie (AC-Netz)	654.970,98 kWh	

Erklärung

Wir versichern, dass wir die Arbeit ohne fremde Hilfe und ohne Benutzung anderer als der angegebenen Quellen angefertigt haben und dass die Arbeit in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegen hat und von dieser als Teil einer Prüfungsleistung angenommen wurde. Alle Ausführungen, die wörtlich oder sinngemäß übernommen wurden, sind als solche gekennzeichnet.

P. Abegart Christina Gasser L. Mauser Miriam H. Möller

Würzburg, den 29. März 2023