




BACHELORARBEIT

Herr
Ben Noack

**Entwicklung und Implementierung
eines normgerechten
Managementsystems zur Verwaltung
von Geräten eines
Diagnostikunternehmens**

Mittweida, Juni 2023



Fakultät **Angewandte Computer- und Biowissenschaften**

BACHELORARBEIT

Entwicklung und Implementierung eines normgerechten Managementsystems zur Verwaltung von Geräten eines Diagnostikunternehmens

Autor:

Ben Noack

Studiengang:

Medieninformatik und Interaktives Entertainment

Seminargruppe:

MI18w1-B

Erstprüfer:

Prof. Dr.-Ing. Christian Roschke

Zweitprüfer:

Dr. Rico Hiemann

Einreichung:

Mittweida, 02.06.2023

Verteidigung/Bewertung:

Mittweida, 2023

Bibliografische Beschreibung:

Noack, Ben:

Entwicklung und Implementierung eines normgerechten Managementsystems zur Verwaltung von Geräten eines Diagnostikunternehmens. – 2023. – 61 S.

Mittweida, Hochschule Mittweida – University of Applied Sciences, Fakultät Angewandte Computer- und Biowissenschaften, Bachelorarbeit, 2023.

Abstract:

Immer komplexer werdende Prozesse in der Geschäftswelt erfordern eine fortschreitende Digitalisierung und Automatisierung. Zusätzlich stellen Gesetze weitere Dokumentationsvorschriften, die zu noch mehr Aufwand führen. Diesen wachsenden Anforderungen kann häufig nur mit wachsender Digitalisierung und somit Automatisierung begegnet werden. In dieser Bachelor-Thesis werden die Möglichkeiten zur Automatisierung von Verwaltungsprozessen analysiert und ein Programm zur digitalen Geräteverwaltung erstellt. Es wird auf Normen zur Software-Entwicklung und -Validierung eingegangen, die dann in die Entwicklung einbezogen werden. Anhand der Verwaltungssoftware werden die einzelnen Entwicklungsphasen beschrieben. Die erstellte Software vereinfacht also nicht nur den Verwaltungsaufwand, sondern folgt auch noch einem gesetzlich vorgegebenen Entwicklungsprozess.

Inhaltsverzeichnis

Inhaltsverzeichnis	I
Abbildungs- und Tabellenverzeichnis	V
1 Einleitung	3
1.1 Automatisierung	3
1.2 Normen	3
1.2.1 Validierung von Software anhand der International Organization for Standardization (ISO) 80002-2	4
1.2.2 International Electrotechnical Commission (IEC) 62304 Medizin-Software-Lebenszyklen-Prozesse	5
1.2.2.1 Software-Entwicklungsprozess	6
1.2.2.2 Software-Wartungs-Prozess	7
1.2.2.3 Software-Risikomanagement-Prozess	7
1.2.3 ISO 25010 Software-Qualitätssicherung	8
2 Ausgangspunkt	9
2.1 Unternehmen GA Generic Assays GmbH	9
2.2 Problemstellung	9
2.3 Aufgabenstellung	9
3 Anforderungsanalyse	11
3.1 Anwenderprofil	11
3.2 Use Case	11
3.3 Anforderungen an das Tool	11
3.4 Arbeitsumgebung	12
3.4.1 Entwicklungsplattform	12
3.4.2 UI Technologien	12
3.4.3 Datenbank	14
4 Designphase	17
4.1 Design	17
4.2 Code Architektur	20
4.2.1 Wartungen und Kalibrierungen aktualisieren	20
4.2.2 neue Geräte erstellen	22
4.2.3 Erinnerung per E-Mail	22
5 Implementierungsphase	25
5.1 Design	25
5.1.1 Mainwindow	25
5.1.2 Detailinfo	26
5.2 Code	27
5.2.1 Datenbank-Zugriff	27
5.2.2 MainWindow	28
5.2.2.1 Methode MainWindowLoad	28

5.2.2.2	Sortierung	29
5.2.2.3	Suchfunktion	30
5.2.2.4	Löschen	30
5.2.2.5	Erstellen	30
5.2.2.6	Bearbeiten	31
5.2.3	DetailInfo	31
5.2.3.1	Get	31
5.2.3.2	Save	32
5.2.4	SaveWindow	32
5.2.5	SendMail	33
6	Testphase	37
6.1	Testen des Designs	37
6.2	Integration des Codes	38
6.2.1	Speichern neuer Daten	38
6.2.2	Automatischer Termin	39
6.2.3	Standard Operating Procedure	39
7	Validierung	41
7.1	Kundenbedürfnisse	41
7.1.1	Ersetzung der Excel Tabelle	41
7.1.2	Automatisierte Aktualisierungen	42
7.1.3	Wöchentliche Erinnerungen	42
7.2	Designanforderungen	42
7.3	Anforderungen nach IEC 62304	43
7.4	Neue Anforderungen	43
7.4.1	Bestätigung per Mail	44
7.4.2	Login	45
7.4.3	Logbuch	46
8	Freigabe	49
9	Wartungsplan	51
9.1	Allgemein	51
9.2	Wartungsziele	51
9.3	Wartungsprozesse	51
10	Risikomanagement	53
10.1	Identifizierung	53
10.2	Analyse und Bewertung	53
10.2.1	Risikoakzeptanz Matrix	53
10.2.2	Datenverlust	54
10.2.3	Fehlerhafte Daten	54
10.2.4	Kompatibilitätsprobleme	55
10.2.5	Probleme mit der Firewall	55
10.2.6	Probleme mit der Nutzerfreundlichkeit	55
10.2.7	Cybersecurity	56
10.3	Risikominimierung	56

Inhaltsverzeichnis	III
11 Ergebnisse	59
12 Ausblick	61
Anhang	63
A SOP Aktualisierung von Wartungs -und Kalibrierungstermin	63
Literaturverzeichnis	71
Eidesstattliche Erklärung	75

Abbildungs- und Tabellenverzeichnis

Abbildungsverzeichnis

1.1	Überblick über Software-Entwicklungs-Prozesse und Aktivitäten[9]	5
4.1	Konzept Tabelle	17
4.2	Konzept Main Window	18
4.3	Konzept DetailInfo	19
4.4	Sequenzdiagramm zur Aktualisierung von Wartungen und Kaliberberungen	21
4.5	Sequenzdiagramm zur Erstellung neuer Geräte	22
4.6	Sequenzdiagramm zur Erinnerung per E-Mail	23
5.1	Benutzeroberfläche MainWindow	26
5.2	Benutzeroberfläche DetailInfo	27
6.1	Benutzeroberfläche mit Hervorhebungen	38
10.1	Risikoakzeptanz Matrix[42]	53

Abkürzungsverzeichnis

CLR	Common Language Runtime
DBMS	Datenbank-Managementsystem
GUI	grafische Benutzeroberfläche
IEC	International Electrotechnical Commission
ISO	International Organization for Standardization
MVCC	Multiversion Concurrency Control
SOP	Standard operating procedures
SQL	Structured Query Language
WPF	Windows Presentation Foundation
XAML	eXtensible Application Markup Language

Quelltextverzeichnis

5.1	Abfrage einer Tabelle	28
5.2	Empfang von Geräte-Objekten aus der Datenbank	28
5.3	Sortierung von Geräten nach Wartungsdatum	29
5.4	Filterung der Datenbank	30
5.5	Löschen von Geräten aus der Datenbank	30
5.6	Aufruf des SaveWindow Dialogs zur Erstellung neuer Geräte	31
5.7	Abfrage der Geräte ID	31
5.8	Speichern der Geräte ID	32
5.9	Speichern der Geräteinformationen	32
5.10	Timer	33
5.11	Abfrage der Daten in der aktuellen Woche	33
5.12	Informationen in Listen ablegen	33
5.13	Zuordnung im dictionary	34
5.14	Inhalt der E-Mail Adresse bereitstellen	35
5.15	E-Mail senden	35
6.1	Null-Coalescing-Operator	38
7.1	Verweis auf ausführbare Datei	44
7.2	Abfrage des Verantwortlichen und deren Termine	44
7.3	Schleife zum Aktualisieren der Termine	45
7.4	Abfrage des Passworts	46
7.5	Speichern des aktuellen Datums in der Historie von Kalibrierungen	46
7.6	Schreiben eines Dokuments	47

1 Einleitung

In der aktuellen Welt, in der künstliche Intelligenz eine immer wichtigere Rolle einnimmt, laufen Prozesse immer schneller ab, ohne dass ein Mensch sich um diese Aufgaben kümmern muss. Die Automatisierung ist ein größeres Thema als jemals zuvor.

Schon Aristoteles sagte: "Wenn jedes Werkzeug auf Geheiß, oder auch vorausahnend, das ihm zukommende Werk verrichten könnte, wie des Dädalus Kunstwerke sich von selbst bewegten oder die Dreifüße des Hephästos aus eigenem Antrieb an die heilige Arbeit gingen, wenn so die Weberschiffe von selbst webten, so bedürfte es weder für den Werkmeister der Gehilfen noch für die Herren der Sklaven."^[1] In der heutigen Zeit sind automatisierte Prozesse allgegenwärtig. Insbesondere in der industriellen Fertigung werden die meisten Produkte mithilfe von Maschinen hergestellt. Durch den Einsatz von Maschinen und automatisierten Systemen wird der Mensch zunehmend entlastet und kann sich vermehrt auf administrative Tätigkeiten konzentrieren.^[2]

Automatisierte Prozesse sind im heutigen Zeitalter zu einem Standard geworden und bieten eine Vielzahl von Vorteilen für Unternehmen.

1.1 Automatisierung

Die Automatisierung bezeichnet den Einsatz von Technologien, um Prozesse oder Systeme selbstständig und mit minimalen Eingriffen auszuführen. Das Ziel besteht darin, wiederkehrende Routinehandlungen in selbstständige Aktionen zu transformieren. Eine solche Transformation soll zu einer Steigerung der Effizienz und Qualität von Arbeitsprozessen führen, indem komplexe Aufgaben durch den Einsatz von Maschinen und anderen Technologien übernommen werden.^[3] Dadurch können Unternehmen ihre Produktionskapazität steigern und ihre Wettbewerbsfähigkeit erhöhen. Die Übertragung der Aufgaben auf Maschinen ermöglicht es, die menschliche Arbeitskraft zu reduzieren. Durch die Freisetzung menschlicher Arbeitskraft können Unternehmen ihre Arbeitsprozesse optimieren und ihre Mitarbeiter auf höherwertige Aufgaben konzentrieren. Diese Aufgaben erfordern in der Regel eine höhere kognitive Komplexität und Flexibilität, die von Maschinen nicht vollständig abgedeckt werden können.^[4] Durch den Einsatz automatisierter Prozesse und Maschinen wird die Fehlerquote deutlich reduziert, da menschliches Versagen minimiert wird. Die Abhängigkeit von manuellen Eingriffen und damit verbundene potenzielle Fehlerquellen werden stark reduziert. Dies führt zu einer erhöhten Zuverlässigkeit und Genauigkeit der Prozesse. Benutzer schätzen es, wenn ihre Anforderungen erfüllt werden und die Software zuverlässig funktioniert. Eine hohe Nutzerzufriedenheit kann sich positiv auf das Image des Unternehmens auswirken und die Kundenbindung stärken.^[5]

1.2 Normen

Normung, auch unter dem Begriff geplante Standardisierung bekannt, ist ein systematischer Prozess, der darauf abzielt Regelungen zu schaffen und in Kraft zu setzen, die materielle und immaterielle Gegenstände vereinheitlichen. Die Normung trägt dazu bei die Kompatibilität,

Interoperabilität, Sicherheit und Qualität von Produkten, Dienstleistungen und Verfahren sicherzustellen.[6]

Die Normung kann sowohl von staatlichen Institutionen als auch von privaten Organisationen durchgeführt werden und umfasst verschiedene Aktivitäten, wie z.B. die Entwicklung von Normen, Standards und Spezifikationen, die Überwachung und Implementierung von Normen, die Schulung von Mitarbeitern sowie die Evaluierung von Normen und deren Aktualisierung.

Ein wichtiger Aspekt der Normung ist die Schaffung von Normen und Standards, die von einer breiten Gruppe von Interessengruppen akzeptiert werden. Dazu gehört die Berücksichtigung der Bedürfnisse von Herstellern, Verbrauchern, Regulierungsbehörden und anderen relevanten Stakeholdern. Die Entwicklung von Normen und Standards erfolgt oft in einem kollaborativen Prozess, der auf einer breiten Basis von Expertenwissen und Meinungen basiert.[7]

Die Normung hat auch wichtige Auswirkungen auf den internationalen Handel und die globalen Märkte. Durch die Harmonisierung von Normen und Standards wird der Handel erleichtert und die Wettbewerbsfähigkeit von Unternehmen gestärkt. Zudem trägt die Normung dazu bei, Verbraucher und die Umwelt zu schützen, indem sie die Einhaltung von Sicherheits- und Umweltstandards sicherstellt.[7]

Insgesamt ist die Normung ein wichtiger Aspekt der modernen Gesellschaft und spielt eine zentrale Rolle bei der Förderung von Innovation, Effizienz, Nachhaltigkeit und Wettbewerbsfähigkeit.

1.2.1 Validierung von Software anhand der ISO 80002-2

Zur Validierung der Software muss die ISO TR 80002-2 [8] eingehalten werden. Die Validierung einer Software erfordert eine gründliche Überprüfung, ob sie den vorgesehenen Verwendungszwecken entspricht und den Anforderungen an Vertrauenswürdigkeit und Zuverlässigkeit gerecht wird. Dabei spielt das sogenannte "kritische Denken" eine entscheidende Rolle. Kritisches Denken beinhaltet die Analyse und Bewertung verschiedener Aspekte der Software und ihrer Umgebung, um mögliche Schwachstellen und Risiken zu identifizieren.

Die Entscheidung, ob eine Validierung gemäß den ISO-Standards erforderlich ist, erfolgt durch eine sorgfältige Prüfung und Identifizierung regulatorischer Anforderungen. Dabei wird untersucht, ob mögliche Mängel oder Ausfälle der Software zu Qualitätsbeeinträchtigungen führen können. Ebenso wird analysiert, ob die Software automatisierte Aktivitäten durchführt, die von regulatorischer Bedeutung sind. Ein Beispiel hierfür wäre die Rückverfolgbarkeit, bei der die Software in der Lage sein muss, den Ursprung von Produkten oder Informationen nachzuvollziehen. Ebenso könnten die Durchführung und Erfassung von Testergebnissen, das Führen von Datenprotokollen, die Verarbeitung von Beschwerden sowie die Unterstützung von Kalibrierungs- und Wartungsaktivitäten von regulatorischem Interesse sein. Falls eine oder mehrere dieser Aspekte auf die Software zutreffen, ist eine Validierung erforderlich. Es ist jedoch möglich, dass nur bestimmte Teile der Software in den Geltungsbereich der Validierung fallen, während andere Aspekte davon ausgenommen sind.

Nachdem festgestellt wurde, dass eine Validierung gemäß den ISO-Standards erforderlich ist, erfolgt die Dokumentation der Verwendung der Software. Dies umfasst die detaillierte Aufzeichnung des vollständigen Verwendungszwecks der Software. Die Dokumentation beschreibt, wer die Software verwendet, welche spezifischen Aufgaben oder Funktionen von

Interesse sind und zu welchem Zeitpunkt und unter welchen Umständen die Software verwendet wird. Es wird erläutert, warum die Software benötigt wird und wie sie den Gesamtprozess automatisiert oder unterstützt. Es werden die Erwartungen an die Software festgehalten, d.h. welche Leistungen von ihr erwartet werden. Ein weiterer Aspekt der Dokumentation betrifft das Vertrauen in die Software. Durch die detaillierte Dokumentation der Software können potenzielle Risiken und Schwachstellen schneller erkannt werden.

In der ISO werden drei Arten von Risiken behandelt, die sorgfältig analysiert werden müssen. Erstens müssen Risiken, die potenzielle Schäden für Menschen verursachen können, erkannt werden. Diese Schäden können direkte oder indirekte Auswirkungen haben. Zweitens besteht das Risiko, dass regulatorische Anforderungen nicht erfüllt werden. Dies kann zu schwerwiegenden Konsequenzen führen, wie dem Verlust wichtiger Aufzeichnungen oder der Nichteinhaltung von Vorschriften und Richtlinien. Drittens müssen Umweltrisiken berücksichtigt werden, die auftreten können, wenn die Software in ihrer spezifischen Umgebung genutzt wird. Diese Risiken beziehen sich auf mögliche Probleme, die sich auf die Umwelt oder andere Systeme auswirken können.

Anhand der identifizierten Risiken kann der Aufwand bestimmt werden, der für die Dokumentation und Validierung der Software erforderlich ist. Es ist von großer Bedeutung, eine kontrollierte Methodik während des gesamten Entwicklungsprozesses anzuwenden, um die Validierung der Software für ihre beabsichtigte Verwendung zu etablieren. Zunächst müssen die Anforderungen für die Software definiert werden. Dies umfasst sowohl die Kundenbedürfnisse als auch die regulatorischen Anforderungen, die erfüllt werden müssen. Nach der Definition der Anforderungen erfolgt die Implementierung und Entwicklung der Software. Danach erfolgt eine erneute Überprüfung, ob die Anforderungen erfüllt sind. Es werden Validierungs- und Verifikationsprozesse durchgeführt, um sicherzustellen, dass die Software den vorgegebenen Spezifikationen entspricht und die gewünschte Funktionalität aufweist. Sobald die Software erfolgreich validiert und verifiziert wurde, kann sie bereitgestellt werden.

1.2.2 IEC 62304 Medizin-Software-Lebenszyklen-Prozesse

Die Norm IEC 62304 [9] beschreibt Prozesse im Software-Lebenszyklus, die darauf abzielen, einen sicheren Entwurf bis hin zur sicheren Wartung zu gewährleisten. Die Norm spezifiziert Anforderungen an diese Prozesse, um eine angemessene Umsetzung zu garantieren.

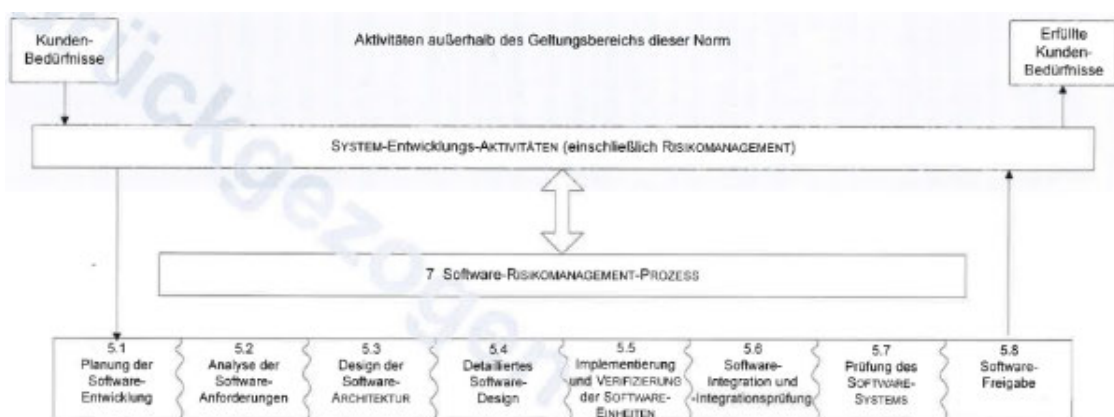


Abbildung 1.1: Überblick über Software-Entwicklungs-Prozesse und Aktivitäten[9]

1.2.2.1 Software-Entwicklungsprozess

Der Software-Entwicklungsprozess ist eine Zusammenstellung von Aktivitäten, die bei der Entwicklung von Software zu beachten sind.

Auf Basis der Kundenbedürfnisse wird ein Software-Entwicklungsplan erstellt(s. Abbildung 1.1), welcher die zu liefernden Ergebnisse, die Prozesse der Entwicklung, das Update Management sowie die Lösung von Problemen, welche während des Lebenszyklus entdeckt werden, spezifiziert.

Anschließend werden die Anforderungen an die zu entwickelnde Software herausgearbeitet(s. Abbildung 1.1). Hierbei wird der Zweck der Software bestimmt und die verwendete Programmiersprache, Plattform sowie das Betriebssystem definiert. Es werden Anforderungen an die benötigte Hardware gestellt und Schnittstellen zu anderen Systemen festgehalten. Die Sicherheit der verwendeten Daten spielt eine wichtige Rolle, wobei Aspekte wie Authentifizierung, Rückverfolgung und Autorisierung zu berücksichtigen sind. Anforderungen an eine benutzerfreundliche Mensch-Maschine-Interaktion werden gestellt. Dabei wird beleuchtet, wie das menschliche Versagen minimiert werden kann. Zudem müssen Form, Funktion und Eignung der Datenbank geprüft werden. Abschließend ist eine Anwenderdokumentation zu erstellen.

Im Anschluss wird die Software-Architektur entworfen (s. Abbildung 1.1) und die zuvor definierten Anforderungen werden umgesetzt. Hierbei werden auch die Schnittstellen zwischen den unterschiedlichen Komponenten entwickelt und gegebenenfalls Leistungsanforderungen für nicht selbst entwickelte Software definiert. Zum Abschluss müssen sämtliche genannte Punkte verifiziert werden, um sicherzustellen, dass die Software den gestellten Anforderungen entspricht und die gewünschten Funktionalitäten bereitstellt.

Im detaillierten Software-Design wird eine Aufteilung der zuvor entworfenen Architektur in einzelne Komponenten durchgeführt (s. Abbildung 1.1). Für jede dieser Komponenten wird ein detailliertes Design erstellt, welches die genauen Funktionalitäten und Schnittstellen definiert. Anschließend müssen die einzelnen Komponenten und deren Designs verifiziert werden, um sicherzustellen, dass sie den gestellten Anforderungen entsprechen und korrekt umgesetzt wurden.

Im nächsten Schritt des Software-Entwicklungsprozesses wird jede zuvor im detaillierten Design entworfene Komponente implementiert (s. Abbildung 1.1). Hierbei müssen bestimmte Kriterien erfüllt werden, wie beispielsweise die Umsetzung von Risikomaßnahmen, die Vermeidung von Widersprüchen und die korrekte Abfolge der Ereignisse. Zum Abschluss dieser Phase müssen erneut sämtliche Punkte verifiziert werden, um sicherzustellen, dass die implementierten Komponenten den gestellten Anforderungen entsprechen und die gewünschten Funktionalitäten bereitstellen.

Alle zuvor implementierten Komponenten werden integriert (s. Abbildung 1.1). Dabei wird überprüft, ob die Funktionalität der Software wie erwartet gegeben ist und ob die implementierten Risikomaßnahmen korrekt umgesetzt wurden. Es werden auch mögliche Missbräuche der Software betrachtet und geprüft, ob die Software in der Lage ist, solchen Missbräuchen entgegenzuwirken. Diese Integration und Überprüfung bildet eine wichtige Phase im Entwicklungsprozess, um sicherzustellen, dass alle Komponenten der Software ordnungsgemäß zusammenarbeiten und die erwartete Funktionalität bereitstellen.

Nach der Integration der einzelnen Komponenten und Überprüfung der Funktionalität des gesamten Systems, muss das System nun auf die zuvor festgelegten Anforderungen geprüft

werden (s. Abbildung 1.1). Dabei werden alle Funktionen und Eigenschaften der Software getestet und überprüft, um sicherzustellen, dass sie den Spezifikationen entsprechen. Falls während dieser Prüfung Änderungen am System vorgenommen werden, muss die Prüfung erneut durchgeführt werden, um sicherzustellen, dass die Änderungen den Anforderungen entsprechen und die Funktionalität nicht beeinträchtigen. Nach Abschluss der Prüfung wird die Verifikation durchgeführt.

Abschließend wird die Software freigegeben (s. Abbildung 1.1), wobei eine erneute Überprüfung der Verifizierung durchgeführt wird, um sicherzustellen, dass diese vollständig ist. Die Version, die freigegeben wird, muss sorgfältig dokumentiert sein, um die Nachvollziehbarkeit und Reproduzierbarkeit zu gewährleisten. Darüber hinaus ist sicherzustellen, dass alle Aufgaben vollständig abgeschlossen wurden, um einen reibungslosen Einsatz der Software zu gewährleisten.

1.2.2.2 Software-Wartungs-Prozess

Die Software erfordert einen Wartungsplan, der die Regelmäßigkeit und Vorgehensweise der Wartungsarbeiten festlegt. Dabei ist zu berücksichtigen, dass auch Feedback, welches seit der Freigabe der Software eingegangen ist, in die Wartungsplanung integriert werden muss. Es gilt, auftretende Probleme im Betrieb der Software zu diskutieren und Lösungsansätze festzulegen, um eine kontinuierliche Verbesserung der Software zu gewährleisten.

Zu Beginn des Prozesses erfolgt eine umfassende Analyse der auftretenden Probleme. Das eingegangene Feedback wird sorgfältig dokumentiert und anschließend geprüft, ob es sich um ein Softwareproblem handelt. Für jedes identifizierte Problem ist ein Bericht anzufertigen, der alle relevanten Informationen zu den Ursachen, Auswirkungen und möglichen Lösungsansätzen enthält. Auf Basis der Analyseergebnisse und der Identifizierung neuer Anforderungen werden neue Anforderungsdokumente erstellt. In einigen Fällen kann es notwendig sein, die zuständige Behörde über die identifizierten Probleme und Lösungsansätze zu informieren, insbesondere wenn es sich um sicherheitsrelevante Probleme handelt oder wenn regulatorische Anforderungen dies erfordern. Die Implementierung der neuen Anforderungen erfolgt gemäß dem definierten Software-Entwicklungsprozess. Nach Abschluss der Implementierungsarbeiten und der Durchführung von ausführlichen Tests erfolgt eine erneute Freigabe der Software.

1.2.2.3 Software-Risikomanagement-Prozess

Zu Beginn des Prozesses ist die Erstellung eines Plans erforderlich, der zunächst definiert, was als Risiko betrachtet wird und wie es entsprechend eingestuft wird. In der Analyse werden potenzielle Gefährdungssituationen identifiziert und bewertet. Dazu werden mögliche Szenarien betrachtet, in denen es zu einer Gefährdung kommen könnte, z.B. durch Fehlfunktionen oder falsche Anwendungen der Software. Es müssen Softwarekomponenten identifiziert werden, die Ursache einer Gefährdung sein könnten, um gezielt Maßnahmen zur Risikoreduzierung treffen zu können. Darüber hinaus müssen auch Ereignisse berücksichtigt werden, die zur Gefährdung der Software beitragen können, wie beispielsweise der Ausfall der gesamten Software oder menschlicher Missbrauch.

Für jede identifizierte Ursache von Gefährdungssituationen müssen geeignete Maßnahmen

zur Risikoreduzierung definiert und dokumentiert werden. Die Maßnahmen können dabei unterschiedlicher Natur sein, z.B. technische oder organisatorische Maßnahmen.

Die festgelegten Maßnahmen zur Risikoreduzierung bedürfen einer Verifizierung, um sicherzustellen, dass sie angemessen und wirksam sind. Darüber hinaus müssen die neuen Maßnahmen evaluiert werden, um mögliche neue Folgen oder Risiken zu identifizieren.

Der beschriebene Prozess zur Risikoreduzierung und Evaluierung betrifft nicht nur die erstmalige Entwicklung der Software, sondern auch alle Änderungen oder Erweiterungen, die an der Software vorgenommen werden.

1.2.3 ISO 25010 Software-Qualitätssicherung

Die ISO 25010 spezifiziert eine Reihe von Methoden zur Evaluation der Qualität von Software. Die Norm enthält eine Checkliste von Aspekten, die für die Qualität von Software von Bedeutung sind.[10] Eine offizielle Bewertung durch einen Prüfer kann genutzt werden, um eine Note für die Software zu erhalten.

Die Bewertung der Software-Funktionalität stellt den ersten Aspekt bei der Qualitätsbewertung von Software dar. Hierbei ist es von großer Bedeutung, inwiefern die gewünschte Funktionalität der Software zur Verfügung gestellt wird. Im Rahmen der Bewertung sind sowohl die funktionale Korrektheit als auch die Vollständigkeit der Softwarefunktionen zu berücksichtigen.

Eine umfassende Bewertung der Effizienz und Performance der Software erfordert die Analyse verschiedener Faktoren, wie z.B. der Rechenzeit, der Speichernutzung und der Netzwerkkommunikation. Es ist wichtig zu prüfen, ob das System die verfügbaren Ressourcen optimal nutzt und ob es in der Lage ist, seine Aufgaben schnell und effizient zu erledigen.

Eine umfassende Bewertung der Kompatibilität der Software erfordert die Analyse verschiedener Faktoren, wie z.B. der Kompatibilität mit verschiedenen Betriebssystemen, Datenformaten und Protokollen. Es ist wichtig sicherzustellen, dass das System mit anderen Systemen und Anwendungen problemlos interagieren kann und dass die Koexistenz mit anderer Software optimal ist.

Eine umfassende Bewertung der Benutzbarkeit der Software erfordert die Analyse verschiedener Faktoren, wie z.B. der Usability, der User-Experience, der Sicherheit und des Designs des User-Interfaces. Es ist wichtig sicherzustellen, dass das System leicht zu erlernen und zu bedienen ist und dass es vor Nutzerfehlern geschützt ist.

Zuverlässige Software trägt dazu bei, dass der Benutzer seine Aufgaben effektiv und effizient erledigen kann und die Gesamtqualität der Software verbessert wird. Um die Zuverlässigkeit einer Software zu bewerten, können verschiedene Methoden wie Stresstests, Fehleranalysen und Überwachung der Systemleistung eingesetzt werden.

Sichere Software trägt dazu bei, dass die Daten des Benutzers geschützt sind und dass die Gesamtqualität der Software verbessert wird. Es ist von Bedeutung, wie die Systemdaten geschützt werden. Daten sollten nur durch autorisierte Benutzer und über entsprechende Mechanismen geändert werden können.[11]

In den nachfolgenden Kapiteln werden exemplarisch die von der Norm IEC 62304 geforderten Entwicklungsphasen anhand der Entwicklung der Geräte-Verwaltungssoftware vorgestellt.

2 Ausgangspunkt

2.1 Unternehmen GA Generic Assays GmbH

Die GA Generic Assays GmbH ist ein führendes Unternehmen im Bereich der Entwicklung, Produktion und Vermarktung von Diagnostik-Kits. Das Unternehmen hat seinen Hauptsitz in Dahlewitz, Deutschland, und ist seit seiner Gründung im Jahr 2002 kontinuierlich gewachsen. Zusätzlich wurde ein weiterer Standort in Senftenberg etabliert, an dem derzeit 16 Mitarbeiter beschäftigt sind.

Das Unternehmen hat sich auf die Herstellung von Diagnostik-Kits spezialisiert, die zur Identifizierung von Autoimmunerkrankungen eingesetzt werden. Durch diese Tätigkeit trägt GA Generic Assays GmbH dazu bei, die Diagnose von Erkrankungen zu verbessern und somit eine optimale Behandlung zu ermöglichen. Zudem ist die Forschung und Entwicklung von innovativen diagnostischen Verfahren und Technologien ein wichtiger Fokus der Firma. [12]

2.2 Problemstellung

Am Standort Senftenberg sind derzeit etwa 500 Geräte verschiedenster Art in Betrieb, was eine große Herausforderung bei der Verwaltung und Überwachung der Geräte darstellt. Durch die Vielzahl unterschiedlicher Geräte ist es schwierig, den Überblick über den Zustand jedes Geräts zu behalten und sicherzustellen, dass es ordnungsgemäß gewartet, geprüft und kalibriert wird.

Aktuell wird jedes Gerät von einem zugeordneten Labormitarbeiter betreut und in einer Excel-Tabelle werden die dazugehörigen Informationen wie der nächste Wartungstermin, die Verantwortlichen und das Wartungsintervall dokumentiert. Die Suche nach den zugeordneten Geräten und deren Informationen erfolgt manuell durch den jeweiligen Verantwortlichen anhand dieser Excel-Tabelle.

Im Bereich der Laborüberwachung und -kontrolle spielt das effiziente Management von Geräten eine entscheidende Rolle. Es ist von großer Bedeutung, dass alle Geräte regelmäßig gewartet, geprüft und kalibriert werden, um eine einwandfreie Funktionsweise sicherzustellen. Hierbei können manuelle Verfahren zeitaufwendig und fehleranfällig sein, insbesondere bei einer großen Anzahl von Geräten.

2.3 Aufgabenstellung

Eine umfassende Verwaltung ermöglicht es den Verantwortlichen, den Wartungsstatus jedes Geräts genau zu verfolgen und sicherzustellen, dass es gemäß den Vorgaben gewartet und geprüft wird. Dies ist insbesondere wichtig, um eine hohe Verfügbarkeit und Zuverlässigkeit der Geräte zu gewährleisten und Ausfallzeiten und Reparaturkosten zu minimieren.

Um diesen Prozess zu optimieren, ist die Entwicklung eines Geräte-Management Tools erforderlich. Das Tool soll es ermöglichen, alle Geräte auf einfache und effektive Weise zu verwalten. Hierzu gehört auch die Möglichkeit, Wartungs- und Servicepläne für jedes Gerät individuell zu konfigurieren. Auf diese Weise können Wartungsarbeiten automatisch geplant

werden, ohne dass es zu Unterbrechungen im Laborbetrieb kommt.

Die Verwaltung der Geräte und Wartungspläne soll mittels einer Datenbank realisiert werden. Hierbei werden alle relevanten Daten wie beispielsweise Gerätetyp, Seriennummer, Datum der letzten Wartung, nächster Wartungstermin, Wartungsplan und weitere Informationen erfasst und gespeichert. Diese Datenbank kann auch für die Erstellung von Reports und Statistiken genutzt werden.

Um eine benutzerfreundliche Handhabung zu gewährleisten, soll das Geräte-Management Tool über ein intuitives User Interface verfügen. Hierbei werden alle Funktionen und Informationen übersichtlich dargestellt und können einfach ausgewählt und bearbeitet werden. Durch eine systematische Darstellung der Daten kann eine schnelle Fehlererkennung und Behebung ermöglicht werden.

Außerdem sollen wöchentliche Erinnerungen per E-Mail an die zugeordneten Mitarbeiter gesendet werden. Diese Erinnerungen enthalten eine Liste der Geräte, die in der kommenden Woche kalibriert, geprüft oder gewartet werden müssen. Jeder Mitarbeiter erhält somit eine klare Übersicht über die zu bearbeitenden Geräte in dieser Woche. Eine solche regelmäßige Erinnerung vermindert den manuellen Aufwand der Suche nach den zu bearbeitenden Geräten.

Darüber hinaus kann eine gut organisierte Dokumentation dabei helfen, die Einhaltung von Vorschriften und Normen zu gewährleisten, die für die jeweiligen Geräte und Anwendungen relevant sind. Die Verwaltung und Überwachung von so vielen Geräten erfordert eine systematische und effiziente Dokumentation, um eine effektive Wartung und Überwachung zu garantieren.

Insgesamt soll das Geräte-Management Tool dazu beitragen, den Arbeitsaufwand für die Verwaltung der Geräte zu reduzieren und somit Zeit und Kosten zu sparen. Durch eine regelmäßige und normgerechte Wartung und Überwachung der Geräte kann dann eine hohe Qualität der Laborergebnisse sichergestellt werden.

3 Anforderungsanalyse

3.1 Anwenderprofil

Die Arbeitsumgebung der Labormitarbeiter umfasst eine Reihe von Software-Anwendungen, darunter hauptsächlich Excel, Word, Microsoft Edge und Outlook. Die Arbeitsumgebung ist durch ein durchgängiges helles Design geprägt, das aus den Farben Weiß, Hellgrau, Grün und Blau besteht.

Bei der Arbeit mit der Excel-Tabelle, die Informationen über die Geräte enthält, nutzen die Labormitarbeiter häufig zwei bestimmte Funktionen - das Suchen nach den eigenen Verantwortlichkeiten und das Filtern nach kommenden Terminen.

Im Rahmen der Wartungs -und Kalibrierungsprozesse im Labor sind bestimmte Informationen der Gerätetabelle von größerer Bedeutung, während andere Informationen weniger relevant sind. Zum Beispiel sind Informationen über den Hersteller, die Risikobewertung und die Seriennummer der Geräte nebensächlich. Die kommenden Termine für Wartungen und Kalibrierungen sind die wichtigsten Informationen.

Um die Geräte zu identifizieren, werden weitere Informationen benötigt, wie beispielsweise die Gerätenummer, die Art des Geräts und der Standort im Labor. Diese Informationen helfen den Mitarbeitern, das Gerät schnell zu identifizieren und gezielte Wartungsarbeiten durchzuführen.

3.2 Use Case

1. Der Labormitarbeiter öffnet die Excel-Tabelle, um einen Überblick über die Geräte und deren Kalibrierungs -und Wartungsbedarf zu erhalten.
2. Es wird die Filterfunktion in der Spalte der Geräte Verantwortlichen angewendet, um lediglich die eigenen Geräte anzuzeigen. Geräte, die weitere Verantwortliche haben, werden jedoch nicht angezeigt. Für jede Kombination an Mitarbeitern gibt es einen eigenen Filter.
3. Anschließend ist es möglich, die Wartungs -und Kalibrierungstermine in der jeweiligen Spalte anzusehen. Mittels eines Filters werden die Termine für nur einen Monat ungeordnet angezeigt.
4. Die nötigen Kalibrierungen und Wartungen werden vom Mitarbeiter durchgeführt.
5. Die bearbeiteten Geräte müssen aktualisiert werden. Das Intervall, in welchem die jeweiligen Geräte zu bearbeiten sind, ist in der Tabelle festgehalten. Dieses Intervall muss vom User auf das Datum gerechnet werden und als neuer Termin eingetragen werden.
6. Die Excel-Tabelle wird geschlossen und der Prozess ist abgeschlossen.

3.3 Anforderungen an das Tool

Um sicherzustellen, dass der User weiterhin in der Lage ist, eigenständige Aktualisierungen durchzuführen, ist es wichtig, dass es eine grafische Benutzeroberfläche für die Datenbank gibt. Es sollte also sichergestellt werden, dass der User weiterhin die Möglichkeit hat, die Gerätedaten manuell in der Tabelle zu aktualisieren, wenn er dies wünscht. Dies kann durch

die Bereitstellung von Schaltflächen oder Funktionen innerhalb der grafische Benutzeroberfläche (GUI) ermöglicht werden, die dem User die Möglichkeit geben, die entsprechende Funktion manuell auszuführen.

Die Automatisierung des Prozesses ist das Hauptziel, um den Arbeitsaufwand für den Nutzer zu minimieren. In diesem Zusammenhang soll der Nutzer lediglich an anstehende Termine erinnert werden und nur die Aktualisierungen bestätigen müssen. Hierzu ist es notwendig, dass ein entsprechendes System implementiert wird, das automatisch Termine erfasst und den Nutzer durch entsprechende Benachrichtigungen auf anstehende Wartungen und Kalibrierungen hinweist. Der Nutzer soll dann in der Lage sein, die Aktualisierungen durch eine einfache Bestätigung zu genehmigen.

3.4 Arbeitsumgebung

3.4.1 Entwicklungsplattform

Das .NET Framework war die vom Unternehmen vorgegebene Entwicklungsplattform. Es ist eine von dem Unternehmen Microsoft bereitgestellte Entwicklungsumgebung, die als Grundlage für die Erstellung von Anwendungen in der Windows-Plattform dient. Es besteht aus zwei Hauptkomponenten: dem Common Language Runtime (CLR) und den Klassenbibliotheken. Die CLR bildet das Fundament des .NET Frameworks und bietet eine standardisierte Laufzeitumgebung, die von allen unterstützten Sprachen genutzt werden kann. Zu den unterstützten Sprachen gehören C++, C# und Visual Basic .NET. Die CLR ist dafür verantwortlich, den ausführbaren Code in maschinenlesbare Anweisungen umzuwandeln. Durch die Nutzung der CLR als gemeinsame Laufzeitumgebung können Entwickler in verschiedenen Sprachen arbeiten und dennoch auf die gleichen Funktionen und Dienste zugreifen. Die Klassenbibliotheken sind eine Sammlung von wiederverwendbaren Codekomponenten, die unabhängig von der Programmiersprache genutzt werden können.[13] In der vorliegenden Arbeit wurde die Version .NET Core 6.0 verwendet. Diese Version wurde von Microsoft im Jahr 2021 veröffentlicht.[14]

Es wurde die Entwicklungsumgebung Visual Studio Community 2022 in der Version 17.5.2 genutzt. Visual Studio ist eine integrierte Entwicklungsumgebung, die eine Vielzahl von Funktionen und Werkzeugen zur Optimierung des Entwicklungsprozesses bietet, wie die Funktionen automatisch die Syntax der Programmiersprache zu überprüfen oder automatisch Methoden oder Funktionen zu ergänzen.[15]

Im Zuge der vorliegenden Arbeit wurde die Programmiersprache C# Sprachversion 10.0 verwendet. .NET Anwendungen sind auf die Existenz des .NET Frameworks beim Ausführen angewiesen. Die eigenständige Übersetzung in Maschinensprache findet nicht statt, jedoch wird die Plattform durch Updates des Windows Betriebssystems gepflegt.

3.4.2 UI Technologien

Die .NET-Plattform stellt verschiedene Optionen zur Entwicklung von Benutzeroberflächen zur Verfügung. Unter anderem gibt es die "Windows Forms Plattform, die "Windows Presentation Foundation und die Uno Platform als Möglichkeiten zur Entwicklung von Desktop-Anwendungen in C#.[16]

Windows Forms ist eine Programmierplattform zur Erstellung von Benutzeroberflächen für Desktop-Anwendungen unter dem .NET Framework. Es bietet eine umfangreiche Bibliothek von vorgefertigten Steuerelementen und Komponenten, die verwendet werden können, um eine benutzerfreundliche Oberfläche für Windows-Anwendungen zu erstellen.[17]

Vorteile:

1. einfache Bedienbarkeit: Die Verwendung von umfangreichen Bibliotheken ermöglicht eine einfache und intuitive Bedienung der Software. Die Entwickler können auf eine Vielzahl von vorgefertigten Komponenten und Steuerelementen zugreifen
2. Hohe Produktivität: Entwickler können auf eine breite Palette von Funktionen und Werkzeugen zurückgreifen. Zudem vereinfacht WindowsForms die Grafikprogrammierung.

Nachteile:

1. Designbeschränkungen: WindowsForms bietet eine begrenzte Flexibilität bei der Gestaltung der Benutzeroberfläche.
2. Veraltet: WindowsForms wurde seit 2002 kaum weiterentwickelt.
3. Mangelnde Skalierbarkeit: Die Darstellung von Grafiken können möglicherweise unscharf oder verschwommen wirken, insbesondere auf hochauflösenden Bildschirmen.

Windows Presentation Foundation ist eine Plattform zur Erstellung von grafischen Benutzeroberflächen für Windows-Anwendungen unter dem .NET Framework. Windows Presentation Foundation (WPF) basiert auf einer Vektorgrafik-Engine. Es gibt eine umfangreiche Sammlung von Klassen, die speziell für die Erstellung moderner Benutzeroberflächen entwickelt wurden. Neben der grundlegenden Funktionalität zur Erstellung von Benutzeroberflächen enthält WPF auch viele Klassen für fortgeschrittenere Funktionen wie Animationen, Stildefinitionen und 2D-/3D-Grafiken.[18] Um eine Benutzeroberfläche zu erstellen, wird die Markup Sprache eXtensible Application Markup Language (XAML) verwendet.[19]

Vorteile:

1. Grafische Leistung: Durch die hardwarebeschleunigte Grafikpipeline können anspruchsvolle grafische Effekte und Animationen reibungslos und effizient dargestellt werden.
2. Skalierbarkeit: WPF verfügt über eine Vektor-Engine, die es ermöglicht, grafische Elemente skalierbar zu gestalten.
3. XAML: WPF verwendet die XAML-Sprache, die eine klare Trennung zwischen der Oberflächengestaltung und dem dahinterliegenden Code ermöglicht.

Nachteile:

1. Höherer Ressourcenbedarf: Aufgrund der hardwarebeschleunigten Grafikpipeline kann WPF einen höheren Ressourcenbedarf haben als andere Frameworks.
2. Steile Lernkurve: Die Konzepte von XAML, Styling und Templates in WPF können anfangs komplex und schwer zu verstehen sein.
3. Langsamer Start: WPF-Anwendungen haben oft einen langsameren Start im Vergleich zu anderen Frameworks.[20]

Uno Platform ist eine Open-Source-Plattform, die es Entwicklern ermöglicht, plattformübergreifende Anwendungen, wie Windows, iOS, Android oder Web zu erstellen. Die Plattform basiert auf der Programmiersprache C und der .NET-Plattform und bietet eine gemeinsame Codebasis, die auf allen Plattformen ausgeführt werden kann.

Uno Platform nutzt die nativen Frameworks von iOS, Android und WebAssembly, um Funktionen der Plattformen vollständig zu nutzen. Uno Platform verwendet auch Webtechnologien wie HTML, CSS und JavaScript, um eine gemeinsame Benutzeroberfläche zu erstellen, die auf allen Plattformen funktioniert.[21]

Vorteile:[22]

1. Plattformübergreifend: Die Uno-Plattform ermöglicht die Entwicklung von Anwendungen.
2. XAML: Die Uno-Plattform verwendet XAML, um die Benutzeroberfläche von der Logik des Codes zu trennen.
3. Vielzahl an UI-Steuer-elementen und Bibliotheken: Die Uno-Plattform bietet eine umfangreiche Sammlung von vorgefertigten UI-Steuer-elementen und Bibliotheken.

Nachteile:

1. Steile Lernkurve: Es kann einige Zeit dauern, um sich mit den Konzepten und Best Practices vertraut zu machen.
2. Inkonsistenz: Aufgrund der Unterschiede in den nativen Plattformen kann es zu inkonsistentem Aussehen und Verhalten der Anwendung kommen.

Nach sorgfältiger Abwägung der verschiedenen Optionen wurde letztendlich die Entscheidung getroffen, WPF für das Projekt zu verwenden. Diese Wahl wurde getroffen, da WinForms als veraltet gilt und kaum noch verwendet wird.

Im Gegensatz dazu bietet WPF eine umfassende grafische Leistung, einfache Skalierbarkeit und die Möglichkeit, mit XAML zu arbeiten, was eine klare Trennung zwischen Code und Benutzeroberfläche ermöglicht. Der höhere Ressourcenbedarf von WPF spielt bei einem Projekt dieser Größe keine wesentliche Rolle und kann vernachlässigt werden. Die steile Lernkurve, ist in diesem Fall nicht relevant, da bereits Vorkenntnisse in der Entwicklungsumgebung vorhanden sind. Der einzige Nachteil von WPF, der als bedeutsam erachtet wurde, ist der langsame Start der Anwendung.

Die Uno-Plattform wurde nicht als optimale Lösung für das Projekt angesehen, da die Möglichkeit der plattformübergreifenden Entwicklung nicht benötigt wurde.

3.4.3 Datenbank

PostgreSQL ist das vom Unternehmen vorgegebene Datenbankmanagementsystem.

Es ist ein objektrelationales Datenbankverwaltungssystem, das als freie und Open-Source-Software entwickelt wurde, die die Standard SQL-Abfragesprache unterstützt.[23] Structured Query Language (SQL) ist eine standardisierte Programmiersprache, die für die Interaktion mit Datenbank-Managementssystem (DBMS) verwendet wird. Sie ermöglicht es Entwicklern, verschiedene Arten von Anweisungen zur Datenverwaltung auszuführen.[24] Die Entwicklung startete 1986 und wird heute von einer großen Community von Entwicklern auf der ganzen Welt unterstützt. Daten werden in strukturierten Tabellen gespeichert und organisiert. PostgreSQL unterstützt auch erweiterte Funktionen.[25]

Unterabfragen, als auch geschachtelte Unterabfragen sind möglich, was komplexe Abfragen erlaubt. Eine Unterabfrage ist eine SQL-Abfrage, die innerhalb einer anderen Abfrage eingebettet ist und dazu dient, die Ergebnismenge der äußeren Abfrage einzuschränken oder

zu erweitern.[26] Dies ermöglicht eine noch feinere Steuerung der Ergebnismenge und eine genauere Filterung von Daten. Die Verwendung von geschachtelten Unterabfragen kann jedoch zu einer höheren Komplexität der Abfrage führen.

Constraints sind Regeln, die auf Tabellen oder Spalten angewendet werden, um sicherzustellen, dass die in der Datenbank gespeicherten Daten korrekt sind. Beispiele für Constraints sind die "Unique" Constraint, die sicherstellt, dass ein bestimmtes Attribut in einer Tabelle nur einmal vorkommt, oder die "Not Null" Constraint, die sicherstellt, dass eine Spalte nicht mit einem Nullwert befüllt wird.[27]

Fremdschlüssel sind eine Art von Constraint, die sicherstellen, dass Beziehungen zwischen Tabellen in der Datenbank korrekt sind. Ein Fremdschlüssel ist ein Attribut in einer Tabelle, das auf einen Primärschlüssel in einer anderen Tabelle verweist. Dadurch wird sichergestellt, dass Daten nur dann in einer Tabelle gespeichert werden können, wenn die entsprechenden Beziehungen zu anderen Tabellen korrekt sind.[28]

Multiversion Concurrency Control (MVCC) ist ein Konzept der Transaktionsverwaltung, das es ermöglicht, dass mehrere Transaktionen gleichzeitig auf eine Datenbank zugreifen können, ohne dass es zu Konflikten oder Inkonsistenzen kommt. Dies wird erreicht, indem eine Transaktion eine Version eines Datensatzes erhält, anstatt direkt auf den aktuellen Datensatz zuzugreifen. Auf diese Weise können mehrere Transaktionen gleichzeitig auf verschiedene Versionen eines Datensatzes zugreifen und Änderungen durchführen, ohne dass es zu Konflikten oder Inkonsistenzen kommt.[29]

Views ermöglichen es, komplexe Abfragen auf mehreren Tabellen zu kombinieren und als eine virtuelle Tabelle darzustellen, auf die zugegriffen werden kann, als ob es sich um eine tatsächliche Tabelle handelt.[24]

Trigger sind eine spezielle Art von Datenbankobjekt, die automatisch ausgeführt wird, wenn eine bestimmte Aktion auf einer Tabelle oder einem bestimmten Datensatz ausgeführt wird. Trigger können beispielsweise verwendet werden, um vor der Einfügung, Aktualisierung oder Löschung von Daten Geschäftsregeln zu überprüfen oder um eine Benachrichtigung an einen Benutzer zu senden, wenn eine bestimmte Aktion auftritt.[30]

Funktionen sind in der Regel spezielle Prozeduren, die auf Daten angewendet werden können, um spezifische Ergebnisse zu erzielen. Sie können beispielsweise verwendet werden, um Berechnungen durchzuführen, Daten zu transformieren oder Abfragen zu optimieren. Dadurch wird eine Datenbank erweiterbar.[31]

Die Speicherkapazität der Tabellen ist lediglich durch den physischen Speicher der Maschine begrenzt. Der Server ist verantwortlich für die Verwaltung und Speicherung der Datenbank, während der Client eine Verbindung zum Server herstellt, um auf die Daten zuzugreifen und Anfragen zu stellen. Durch die klare Trennung der Verantwortlichkeit des Servers und des Client können beide Teile auf unterschiedlichen Maschinen laufen. Auf diese Weise ist ein entfernter Zugriff möglich.[32]

Die Software pgAdmin bietet eine grafische Benutzeroberfläche zur einfachen Handhabung der Datenbank.

4 Designphase

4.1 Design

In Anbetracht der Arbeitsumgebung der Mitarbeiter, die in einem hellen Design stattfindet, wurde beschlossen, dass auch das Tool in hellen Farben gehalten wird. Dabei wurde berücksichtigt, dass die Arbeitsumgebung nur begrenzt Farben aufweist und hauptsächlich auf Weiß, Schwarz, Hellgrau und Blau beschränkt ist. Um die Ähnlichkeit zu einer Excel-Tabelle im Hauptfenster zu wahren, wurde das Farbschema insgesamt nicht gebrochen.

Eine geeignete Benutzeroberfläche für die Software sollte in Form einer Tabelle gestaltet werden, um den Nutzern ein vertrautes Arbeitsumfeld zu bieten. Um eine effiziente Verwaltung der Geräte in der Tabelle zu ermöglichen, sind bestimmte Spalten von besonderer Bedeutung. Insbesondere handelt es sich dabei um die Termine für die nächsten Prüfungen, Kalibrierungen und Wartungen, da sie darüber informieren, wann eine Aktualisierung notwendig ist. Die Spalte des Verantwortlichen in der Tabelle gibt den Namen des Mitarbeiters wieder, welcher das jeweilige Gerät bearbeitet und aktualisiert. Darüber hinaus ist die Angabe der Gerätenummer unverzichtbar, um zu wissen, um welches Gerät es sich genau handelt. Die Geräteart gibt Aufschluss darüber, wie das Gerät behandelt werden muss und welche Besonderheiten es aufweist. Ebenso ist die Information zum Standort notwendig, um zu erfahren, wo sich das Gerät befindet. Durch die Einbeziehung dieser Spalten in die Tabelle wird eine effiziente Verwaltung der Geräte gewährleistet. Statt 24 Spalten umfasst die neue Version nur noch 7 Spalten(s. Abbildung 4.1). Dieser Schritt wurde unternommen, um den Nutzern einen leichteren Überblick über die relevanten Informationen zu verschaffen und das Handling der Tabelle zu vereinfachen.

Gerätenummer	Gerätegruppe	Typ/Name	Standort	Verantwortlicher	Kalibrierung	Wartung
SFB_GA_001	Kühlschrank	Gefrierschrnk	Raum 1.13	JN	2023-03-24	
SFB_GA_002	Divers	Cryoster NX 70HOMP	Raum 1.07	SLi	2023-10-01	2023-10-01

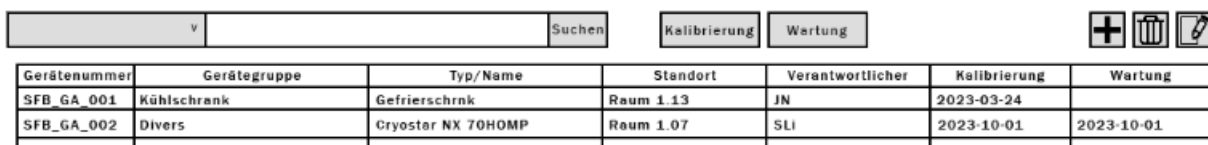
Abbildung 4.1: Konzept Tabelle

Um die Suche nach bestimmten Verantwortlichen in der Tabelle zu erleichtern, wurde in der Benutzeroberfläche ein Suchfeld implementiert. Da eine Dropdown-Liste aufgrund der großen Anzahl von Mitarbeitern unübersichtlich und unpraktisch wäre, ermöglicht das Suchfeld eine effiziente Filterung der angezeigten Daten. Der Benutzer kann den Namen des gesuchten Mitarbeiters in das Suchfeld eingeben, und die Tabelle zeigt automatisch nur die relevanten Einträge an. Auf diese Weise kann der Benutzer weiterhin schnell und einfach seine eigenen Verantwortlichkeiten einsehen.

Die Positionierung der Suchleiste ist ein wichtiger Aspekt der Benutzeroberfläche, der Einfluss auf die Usability hat. Für eine effektive Suche, auch mit längeren Suchbegriffen, sollte die Suchleiste ausreichend Platz bieten. Eine Möglichkeit, um die Breite der Oberfläche nicht unnötig zu vergrößern, bestand darin, die Suchleiste entweder über oder unter der Tabelle zu platzieren. In vielen gängigen Benutzeroberflächen ist die Suchleiste im oberen Bereich zu finden, weshalb auch in diesem Fall eine Positionierung oberhalb der Tabelle gewählt wurde (s. Abbildung 4.2)

Die Möglichkeit, die Termine in der Software nach dem kommenden Datum zu sortieren, ähnlich wie in Excel, ermöglicht den Nutzern einen schnellen Überblick über die nächsten anstehenden Wartungen und Kalibrierungen. Zur Umsetzung dieser Funktion wurden Schaltflächen in die Benutzeroberfläche integriert, um dem Nutzer die Möglichkeit zu geben, die gewünschte Sortierung mit einem einzigen Klick durchzuführen.

Für eine klare visuelle Hierarchie in der Benutzeroberfläche und eine erhöhte Bedienbarkeit, wurden die Buttons zur Sortierung von Wartungs- und Kalibrierungsterminen von den anderen Steuerelementen der Oberfläche räumlich abgetrennt (s. Abbildung 4.2). Dies ist aufgrund ihrer spezifischen Funktionen notwendig und ermöglicht eine bessere Orientierung für den Benutzer. Die Buttons wurden neben der Suchleiste platziert, da sich alle anderen interaktiven Steuerelemente ebenfalls in diesem Bereich befinden und somit eine konsistente Platzierung gewährleistet wird.



Gerätenummer	Gerätegruppe	Typ/Name	Standort	Verantwortlicher	Kalibrierung	Wartung
SFB_GA_001	Kühlschrank	Gefrierschrank	Raum 1.13	JN	2023-03-24	
SFB_GA_002	Divers	Cryostar NX 70HOMP	Raum 1.07	SLI	2023-10-01	2023-10-01

Abbildung 4.2: Konzept Main Window

Um den schnellen Zugriff auf bestimmte Geräte zu ermöglichen, wurde in die Benutzeroberfläche eine Filteroption für das Suchfeld integriert (s. Abbildung 4.2). Auf diese Weise kann der Nutzer gezielt nach Gerätenummern suchen und somit das gewünschte Gerät schnell finden. Diese Funktionalität ist besonders nützlich, wenn der Nutzer eine bestimmte Gerätenummer sucht oder er schnell auf ein bestimmtes Gerät zugreifen will, ohne alle Geräte durchsuchen zu müssen. Die Filteroption im Suchfeld ermöglicht somit eine effektive und zeitsparende Suche nach Geräten in der Tabelle.

In Situationen, in denen eine angekündigte Überprüfung durchgeführt wird (beispielsweise durch Behörden), erfolgt eine Überprüfung sämtlicher Geräte in bestimmten Räumen. Zur effizienten Identifikation aller Geräte in einem Raum wurde eine zusätzliche Filteroption für die Suche entwickelt, nämlich der Standort (s. Abbildung 4.2). Diese Funktion erweist sich auch als nützlich, um beispielsweise eine Checkliste für den Umbau eines Labors zu erstellen oder eine übersichtliche Bestandsaufnahme beim Umzug von Laboren zu führen.

Nicht mehr in Betrieb befindliche Geräte, die jedoch noch vorhanden sind, werden in einer separaten Tabelle innerhalb der Excel-Datei aufgeführt. Da diese Geräte nicht mehr genutzt werden, haben sie keine bevorstehenden Wartungs- oder Kalibrierungstermine und sollten daher nicht in der Haupttabelle des zu entwickelnden Programms erscheinen. Zur Verwaltung dieser außer Betrieb genommenen Geräten wurde eine zusätzliche Filteroption in die Suchleiste implementiert. Durch diese Maßnahme ist es nun möglich, die außer Betrieb genommenen Geräte aufzufinden und zu verwalten, ohne dass dafür eine separate Tabelle angelegt werden muss. Die betreffenden Geräte interferieren auf diese Weise nicht mit den anderen Geräten. Die nicht verwendeten Geräte sind klar vom Rest abgetrennt.

Es war von großer Bedeutung sicherzustellen, dass sämtliche Informationen zu den Geräten weiterhin einsehbar sind. Beispielsweise werden zusätzliche Informationen zur Kalibrierung und Wartung in der Kommentarspalte festgehalten. Dabei handelt es sich um wichtige Details, die zur Durchführung von Wartungs- und Kalibrierungsarbeiten an den Geräten notwendig sind. Eine detaillierte Ansicht der Geräteinformationen war hierbei unverzichtbar.

Es galt zu beachten, dass das Tool nicht ausschließlich zur Verwaltung von Wartungs- und Kalibrierungsarbeiten genutzt werden, sondern als allgemeines Verwaltungstool für die Geräte dienen soll. Es ist geplant, dass das Tool die Excel-Tabelle ablöst, weshalb es notwendig war, sämtliche Informationen aus der Excel-Tabelle in das neue System zu übertragen.

Aus diesen Gründen wurde eine Detailansicht für jedes einzelne Gerät implementiert(s. Abbildung 4.3). In dieser Detailansicht sind sämtliche Informationen zum betreffenden Gerät aufgeführt. Die Möglichkeit, die Informationen in dieser Ansicht zu bearbeiten und zu speichern, ermöglicht eine einfache und schnelle Aktualisierung der Geräteinformationen. Darüber hinaus stellt die Detailansicht sicher, dass sämtliche Informationen zu den Geräten leicht zugänglich und übersichtlich dargestellt werden.

Zur Erlangung einer detaillierten Ansicht eines Geräts wird zunächst das betreffende Gerät selektiert und im Anschluss darauf ein Button gedrückt. Bei jedem Gerät müssen bis zu 25 einzelne Informationen erfasst werden. Aufgrund der Vielzahl an Informationen ist es oft schwierig, den Überblick über bestimmte Themenbereiche wie Kalibrierung und Wartung zu behalten. Aus diesem Grund und für eine übersichtlichere Darstellung wurden die Informationen visuell in verschiedene Absätze für die jeweiligen Themengebiete unterteilt. Durch die Unterteilung in Absätze können die Benutzer schnell und einfach zu den relevanten Informationen navigieren.

interne Gerätenummer	<input type="text" value="SFB_GA_001"/>
Gerätegruppe	<input type="text" value="Gerätegruppe"/>
Geräteart	<input type="text" value="Gefrierschrank"/>
Hersteller	<input type="text" value="Liebherr"/>
weitere	<input type="text" value="..."/>
<hr/>	
Kalibrierungspflichtig	<input checked="" type="checkbox"/>
Intervall	<input type="text" value="30"/> Tage
nächster Termin	<input type="text" value="2023-03-24"/> <input type="button" value="13"/>

Abbildung 4.3: Konzept DetailInfo

Um eine effektive und benutzerfreundliche Darstellung der Geräteinformationen zu gewährleisten, wurde je nach gegebener Information eine passende Darstellungsweise gewählt. Da die meisten Informationen in Textform vorliegen, wurde das TextBox-Format als geeignetste Option ausgewählt, um eine einfache Bearbeitung der Informationen zu ermöglichen. Die Verwendung von TextBoxen erleichtert die Eingabe und Bearbeitung von Textdaten.

Für Informationen wie Kalibrierungs- und Wartungspflichten, die lediglich zwei Zustände annehmen können, nämlich wahr oder falsch, wurde in der Datenbank das Boolean-Datentyp verwendet. Eine geeignete Möglichkeit, diese beiden Zustände visuell darzustellen, ist die Verwendung einer Checkbox. Wenn die Information zutrifft, wird ein Häkchen in der Checkbox angezeigt, andernfalls bleibt die Checkbox leer. Diese Methode bietet eine klare und unmissverständliche Darstellung der Informationen, die schnell durch lediglich einen Klick bearbeitbar ist.

Die Verwendung von Datepickern ermöglicht es dem Benutzer, ein Datum mithilfe einer Ka-

lenderansicht auszuwählen. Durch die Kalenderansicht kann der Benutzer das gewünschte Datum leicht identifizieren und auswählen. Ein weiterer Vorteil der Verwendung von Datepickern besteht darin, dass der Benutzer den Wochentag des ausgewählten Datums einfacher bestimmen kann.[33]

Die Möglichkeit zur Ergänzung von neuen Geräten sowie zur Löschung aussortierter Geräte ist unabdingbar, um die Aktualität der Geräteliste durch den Nutzer zu gewährleisten. Zur Abhilfe ist die Implementierung entsprechender Buttons erforderlich, welche analog zum Bearbeitungs-Button das selektierte Gerät als Zielobjekt adressieren.

Für die Buttons zum Löschen und Hinzufügen werden oftmals etablierte Symbole wie ein Papierkorb oder ein Pluszeichen verwendet, da diese eine klare Assoziation zur entsprechenden Funktion haben. Der Stift gilt in der Benutzeroberfläche als etabliertes Symbol, um eine Bearbeitungsfunktion zu kennzeichnen. Somit ist eine textliche Beschriftung nicht zwingend notwendig, da die Symbole eine eindeutige Bedeutung vermitteln können.

In der Benutzeroberfläche gehören diese Steuerelemente räumlich gesehen zur Toolbar, welche aus allen interaktiven Steuerelementen besteht. Allerdings wurden sie aufgrund ihrer spezifischen Funktionen von den anderen Elementen der Toolbar abgegrenzt. Dies ist eine gängige Praxis in verschiedenen Softwareanwendungen, um eine klare Trennung und Visualisierung der Funktionen zu gewährleisten.

Sämtliche Funktionen der Excel-Tabelle wurden somit realisiert. Eine Übersichtstabelle enthält sämtliche relevanten Spalten der Geräte. Es ist dem Nutzer möglich, seine individuellen Verantwortlichkeiten schnell ausfindig zu machen und nach Datum zu sortieren. Aussortierte Geräte sind von den noch im Betrieb befindlichen Geräten abgegrenzt. Eine vollständige Einsicht und Bearbeitung aller Geräteinformationen ist möglich. Es besteht die Möglichkeit, neue Geräte hinzuzufügen, während defekte oder abgegebene Geräte aus der Liste gelöscht werden können.

4.2 Code Architektur

Zur visuellen Darstellung und Veranschaulichung von Interaktionen zwischen Objekten und deren zeitlichen Ablauf wurden in diesem Kontext Sequenzdiagramme verwendet. Ein Sequenzdiagramm besteht aus einer vertikalen Zeitdimension und einer horizontalen Beschreibung der Interaktionen.[34]

4.2.1 Wartungen und Kalibrierungen aktualisieren

Der primäre Prozess konzentriert sich auf die Aktualisierung von Wartungs- und Kalibrierungsterminen. Dabei werden verschiedene Schritte durchgeführt, die Suche nach relevanten Geräten, die Sortierung der gefundenen Geräte und die Aktualisierung der Termine.

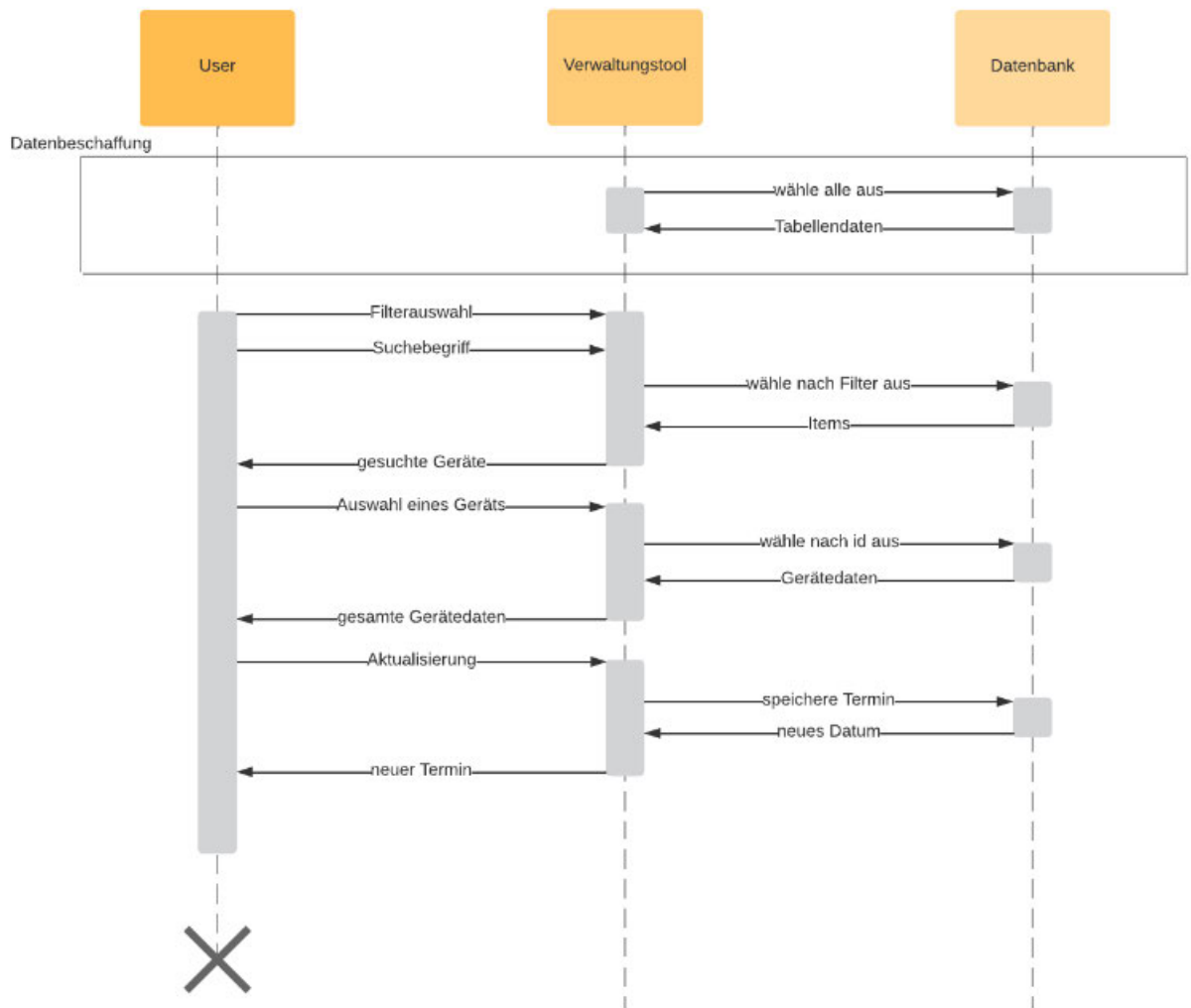


Abbildung 4.4: Sequenzdiagramm zur Aktualisierung von Wartungen und Kalibrierungen

Zu Beginn des Prozesses wird eine Beschaffung der Daten durchgeführt. Die Software greift auf die Datenbank zu und fragt Informationen zu allen Geräten ab. Die Datenbank gibt die angeforderten Informationen zurück, und daraufhin werden die entsprechenden Daten in die Tabelle eingefügt (s. Abbildung 4.4).

Der Benutzer verwendet eine Auswahlbox, um einen bestimmten Filter auszuwählen. Anschließend gibt der Benutzer einen Suchbegriff ein, den er unter Verwendung des Filters abfragen möchte. Das Verwaltungstool fordert daraufhin die entsprechend gefilterten Daten an. Die Datenbank gibt daraufhin nur die relevanten Gerätedaten zurück. Diese Informationen werden vom Tool in Form einer Tabelle dem Benutzer angezeigt (s. Abbildung 4.4).

Der Benutzer wählt ein Gerät aus, für das er die erforderlichen Wartungs- oder Kalibrierungsschritte durchgeführt hat. Die Software bestimmt die ID des ausgewählten Geräts und fordert alle Informationen zu diesem Gerät an. Die Datenbank gibt daraufhin alle Informationen über das Gerät zurück. Das Tool zeigt anschließend alle Gerätedaten des ausgewählten Geräts in einer Bearbeitungsansicht an.

Der Benutzer aktualisiert einen Termin mithilfe des Tools. Das Tool übermittelt den aktuali-

sierten Termin zur Speicherung an die Datenbank. Die Datenbank bestätigt die erfolgreiche Speicherung und gibt den neuen Wert zurück, der dem Benutzer durch das Verwaltungstool angezeigt wird (s. Abbildung 4.4).

4.2.2 neue Geräte erstellen

Die Daten werden beschafft, indem das Verwaltungstool Informationen zu allen Geräten anfordert. Die Datenbank stellt daraufhin die benötigten Informationen zur Verfügung, und das Verwaltungstool füllt die Tabelle mit den Daten zu allen Geräten.

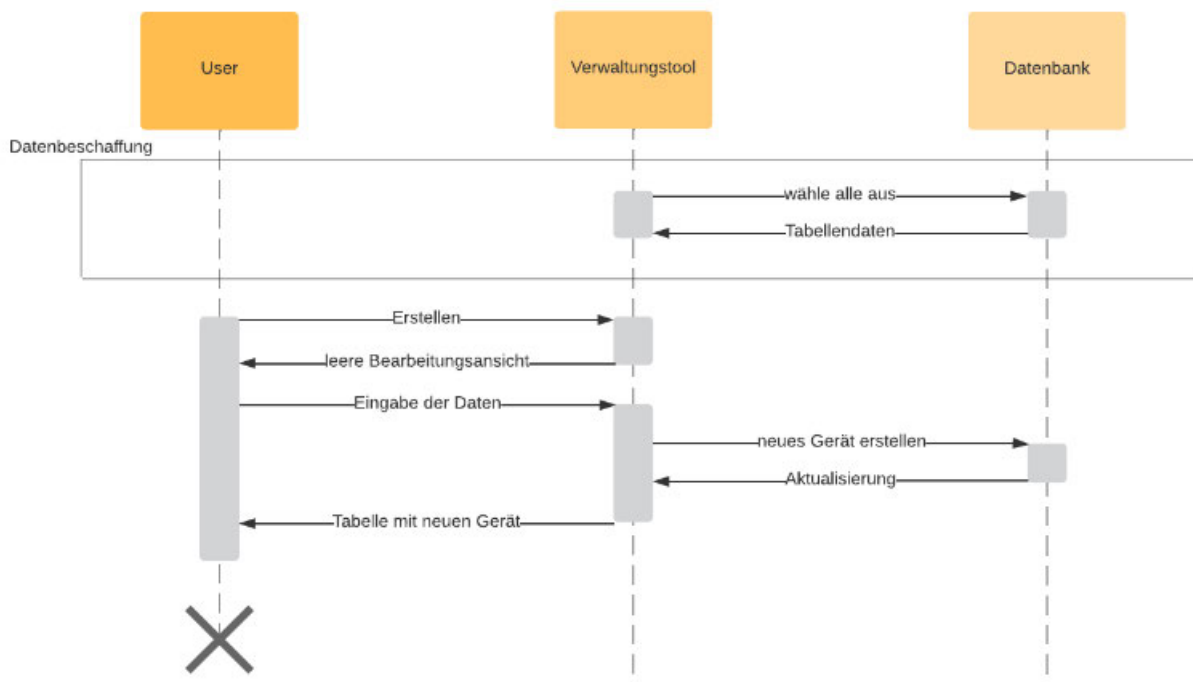


Abbildung 4.5: Sequenzdiagramm zur Erstellung neuer Geräte

Der Nutzer beabsichtigt, ein neues Gerät hinzuzufügen. Daraufhin öffnet das Tool eine leere Bearbeitungsansicht, in der Daten für das neue Gerät eingetragen werden können.

Der Benutzer gibt alle geforderten Daten zum neuen Gerät ein. Die Software überträgt diese Daten an die Datenbank, wo sie als neue Zeile gespeichert werden. Anschließend werden die aktualisierten Daten aus der Datenbank abgerufen und vom Tool dem Benutzer in der Tabelle angezeigt (s. Abbildung 4.5).

4.2.3 Erinnerung per E-Mail

Die Mitarbeiter sollen zu Beginn der Woche auf die anstehenden Termine in derselben Woche aufmerksam gemacht werden. Dafür wird eine E-Mail an die jeweiligen Verantwortlichen gesendet, die ihn an die Geräte erinnern, die eine Wartung oder Kalibrierung benötigen.

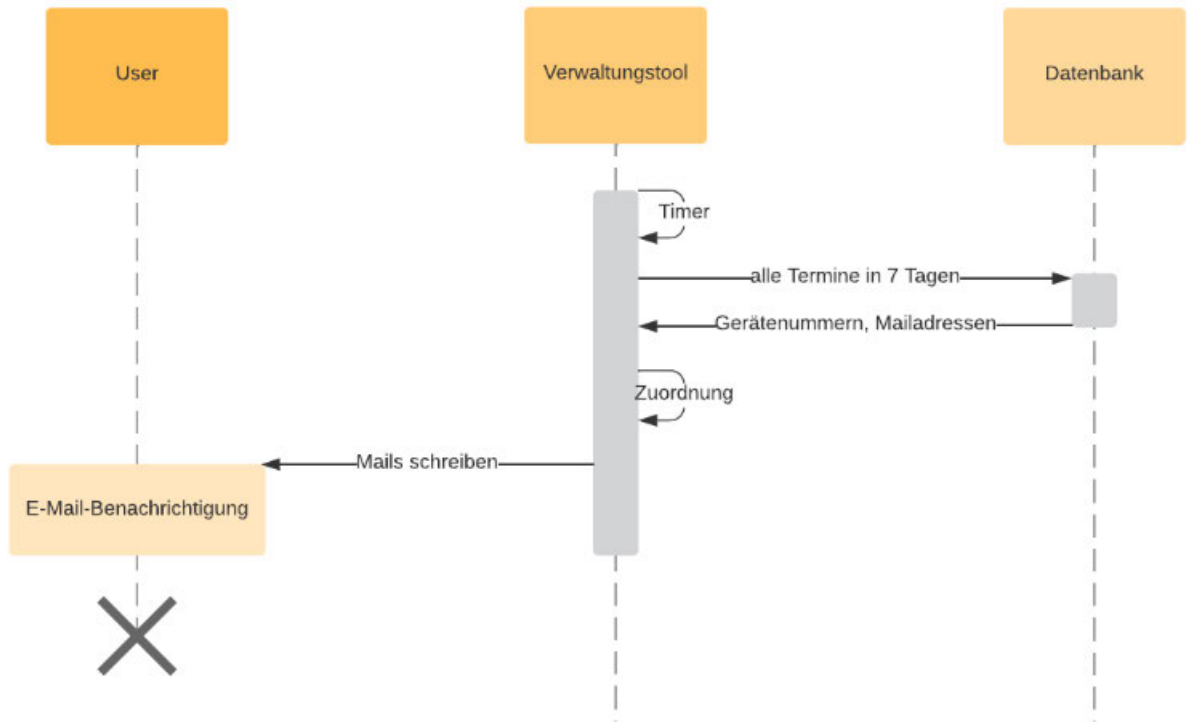


Abbildung 4.6: Sequenzdiagramm zur Erinnerung per E-Mail

Im Tool wird kontinuierlich ein Timer ausgeführt, der überprüft, ob es Montag 0 Uhr ist. Alle Geräte in der Datenbank werden nach ihren Terminen durchsucht. Die Datenbank gibt die Gerätenummern und die dazugehörigen E-Mail-Adressen aller Geräte zurück, die in den nächsten 7 Tagen einen Termin haben. Die Gerätenummern werden den E-Mail-Adressen zugeordnet, so dass jede E-Mail-Adresse nur einmal vorkommt und alle zugehörigen Gerätenummern eingetragen sind. Nach der Zuordnung der Gerätenummern zu den E-Mail-Adressen wird eine Nachricht verfasst, in der die Gerätenummern aufgezählt werden. Diese Nachricht wird dann an jede einzelne E-Mail-Adresse gesendet, und der Benutzer erhält die entsprechende E-Mail (s. Abbildung 4.6).

5 Implementierungsphase

5.1 Design

Die Umsetzung des Designs erfolgte mittels des Frameworks WPF. Hierbei wird die Benutzeroberfläche mittels XAML, einer Auszeichnungssprache, beschrieben.

5.1.1 Mainwindow

Es existieren diverse Methoden, um eine Tabelle mittels XAML zu implementieren. Dabei ist es unabdingbar, dass die Tabelle auf eine einfache und flexible Art und Weise befüllbar ist, da die Anzahl der Zeilen variabel ist.

Eine elegante Lösung bietet die Verwendung einer ListView. Diese ermöglicht die Darstellung mehrerer Elemente in einer horizontalen Ausrichtung. Dabei handelt es sich um ListViewItems, welche lediglich ein einzelnes Datenelement aus einer Sammlung repräsentieren. Durch die Verwendung der ListView können die Geräte in einer Liste dargestellt werden.[35] Im Unterschied zum ItemsControl, welches lediglich eine einfache Auflistung von Elementen darstellt, ermöglicht die ListView dem Nutzer die Auswahl von einzelnen Elementen.

In der ListView ist eine GridView verschachtelt implementiert. Durch die Verwendung der GridView werden Daten an eine bestimmte Spalte gebunden. Jede Spalte verfügt wie in einer Tabelle über eine Überschrift, welche optional als Schaltfläche genutzt werden kann. Die GridViewColumns definieren die Eigenschaften der einzelnen Spalten innerhalb der GridView.[36] Durch das Anwenden des Data Binding-Verfahrens können Daten in der Benutzeroberfläche dargestellt werden. Eine Aktualisierung der Daten im Hintergrund wird ausgelöst, sobald eine Änderung der Daten in der Oberfläche erfolgt. Dies erleichtert das Einbinden von Daten vom Server in die Oberfläche und ermöglicht eine einfache Aktualisierung der Daten.[37]

Die TextBox ist ein Steuerelement, welches die Anzeige und Bearbeitung von Text ermöglicht. Sie eignet sich für die Sucheiste, da der Nutzer einfach Text eingeben und bearbeiten kann.[38]

Eine ComboBox ist ein Steuerelement in einer grafischen Benutzeroberfläche, das dem Benutzer eine Dropdown-Liste von Optionen zur Auswahl bereitstellt. Beim Klick auf die ComboBox öffnet sich eine Liste von Auswahlmöglichkeiten, aus der der Benutzer eine Option auswählen kann. Nach der Auswahl einer Option wird der ausgewählte Wert in der ComboBox angezeigt. Beim Klick auf eine Auswahl in einer Combobox wird eine Methode aufgerufen, welche die ausgewählte Option verarbeitet. Dabei kann die Methode die Auswahl speichern, eine Aktion ausführen oder weitere Elemente der Oberfläche aktualisieren.[39]

Die ComboBox stellt aufgrund ihrer Funktionalität, bei der durch das Klicken verschiedene Auswahlmöglichkeiten angezeigt werden, eine geeignete Lösung für die Auswahl der Suchkriterien dar. Durch die räumliche Nähe zur Sucheiste wird die Verbindung zur Suchfunktion hervorgehoben (s. Abbildung 5.1).

Ein Button ist ein interaktives Steuerelement, das dem Nutzer eine Möglichkeit bietet, eine Aktion auszuführen. Bei einem Klick auf den Button wird ein Click Event ausgelöst, welches dazu führt, dass eine bestimmte Methode aufgerufen wird. Eine klare und eindeutige Beschriftung des Buttons ist entscheidend, um die Funktion des Buttons verständlich zu machen.

Der Such-Button wird in der Benutzeroberfläche mit dem Label "Suchen" beschriftet, um die Funktion der Schaltfläche für den Benutzer klar und unmissverständlich zu kommunizieren. Durch die sinnvolle Positionierung des Buttons in unmittelbarer Nähe zur Suchleiste wird die Verbindung zwischen beiden Elementen betont und die Bedienung der Suche intuitiv gestaltet (s. Abbildung 5.1).



The screenshot shows a software window titled 'MainWindow'. At the top, there is a search bar with a dropdown arrow and three buttons: 'Suchen', 'Kalibrierung', and 'Wartung'. To the right of these buttons are standard window controls (minimize, maximize, close) and three icons: a plus sign, a trash can, and a pencil. Below the search bar is a table with the following data:

Gerätenummer	Gerätegruppe	Typ/Name	Standort	Verantwortlicher	Qualifizierung	Kalibrierung	Wartung
SFB 03	Sterilwerkbank Heraeus HS 12	Sicherheitswerkbank Heraeus HS 12	Raum 1.11	FK, SMi, TS		2024-05-21	2024-05-21
SFB 09	Trockenschrank	SN450	Raum 1.10	MS		2023-05-25	2023-05-25
SFB 08	CO2 Inkubator	CB53	Raum1.11	FK,TS		2023-05-29	2024-05-21
SFB 06	Kühl- und Gefrierkombination	KGE39AW42	Raum1.11	JM		2023-05-25	
SFB 02	Sterilwerkbank Heraeus HS 18	Sicherheitswerkbank Heraeus HS 18	Raum 1.08	FK, SMi, TS		2024-05-21	2024-05-21
SFB 11	N2 Tank	HC34	Raum 1.12	SMi			
SFB 12	N2 Vorratstank	Classic 25	Raum 1.12	SMi			

Abbildung 5.1: Benutzeroberfläche MainWindow

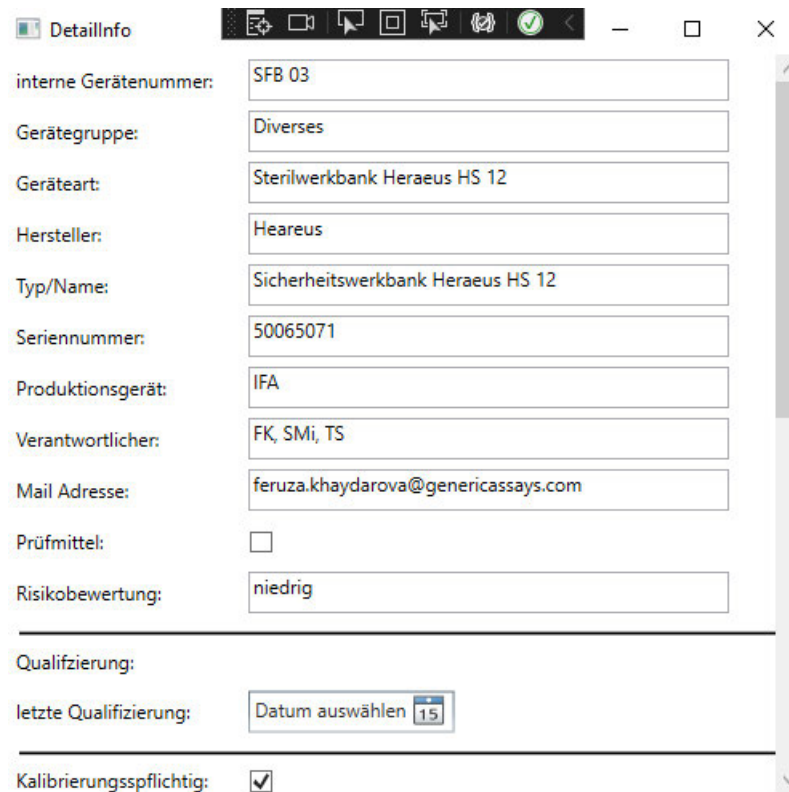
Es ist sinnvoll, die Funktion zur Sortierung der Tabelle nach Wartungs- und Kalibrierungstermin durch Buttons zu ermöglichen, da dies eine einfache und benutzerfreundliche Interaktionsmöglichkeit bietet. Um eine klare Verbindung zur dazugehörigen Spalte herzustellen, werden die Buttons mit den Beschriftungen "Wartung" und "Kalibrierung" versehen, die jeweils den Spaltenköpfen entsprechen (s. Abbildung 8).

5.1.2 Detailinfo

In der Detailansicht werden ausschließlich Beschreibungen und ihre zugehörigen Inhalte angezeigt. Eine Label-Klasse dient als Container zur Anzeige von Textinhalten und wird häufig genutzt, um Bezeichnungen und Beschriftungen in einer Benutzeroberfläche darzustellen. Im Gegensatz zu anderen Steuerelementen ist das Label nicht interaktiv, sondern lediglich informativ. Im Tool wird für jede Spalte der Excel-Tabelle ein Label verwendet, um die jeweiligen Informationen anzuzeigen.

Die zu den Labels gehörenden Geräteinformationen werden in den TextBoxen angezeigt, da TextBoxen dem Benutzer es erlauben, einfach Informationen zu aktualisieren oder hinzuzufügen (s. Abbildung 5.2).

Es wurde entschieden, auf eine farbliche Unterteilung der Bereiche zu verzichten, da die Farbgebung der gesamten Oberfläche bislang eher dezent gehalten ist und die Einführung weiterer Farben das Gesamtdesign beeinträchtigen könnte. Stattdessen wurden die Bereiche durch eine Linie getrennt (s. Abbildung 5.2).



The screenshot shows a window titled 'DetailInfo' with a standard Windows-style title bar. The window contains a form with the following fields and values:

interne Gerätenummer:	SFB 03
Gerätegruppe:	Diverses
Geräteart:	Sterilwerkbank Heraeus HS 12
Hersteller:	Heareus
Typ/Name:	Sicherheitswerkbank Heraeus HS 12
Seriennummer:	50065071
Produktionsgerät:	IFA
Verantwortlicher:	FK, SMi, TS
Mail Adresse:	feruza.khaydarova@genericassays.com
Prüfmittel:	<input type="checkbox"/>
Risikobewertung:	niedrig
Qualifizierung:	
letzte Qualifizierung:	Datum auswählen 15
Kalibrierungspflichtig:	<input checked="" type="checkbox"/>

Abbildung 5.2: Benutzeroberfläche DetailInfo

5.2 Code

Eine gute Software-Architektur zeichnet sich durch die Trennung von Oberfläche (Frontend) und Ausführungslogik (Backend) aus. Das Backend bezieht sich auf den Teil der Anwendung, der für die Verarbeitung von Daten und die Durchführung von Berechnungen verantwortlich ist, während das Frontend die Benutzeroberfläche bereitstellt, über die Benutzer mit der Anwendung interagieren können. In diesem Kapitel liegt der Fokus auf dem Backend, das in C# geschrieben ist. WPF unterstützt diese Architektur durch den sogenannten "Code behind". Da das Frontend Design in der Beschreibungssprache XAML realisiert wird, ist das Backend schon grundsätzlich vom Frontend getrennt.

5.2.1 Datenbank-Zugriff

In jedem Fenster ist ein Zugriff auf die Datenbank erforderlich, um Daten abzurufen oder zu speichern. Um diese Verbindung zur Datenbank herzustellen, wird ein Builder verwendet, der die erforderlichen Informationen für den Verbindungsaufbau bereitstellt.

Die `DbConnectionStringBuilder`-Klasse stellt eine Funktionalität zur Verwaltung von Verbindungszeichenfolgen bereit. Mit Hilfe dieser Klasse kann ein Entwickler Verbindungsparameter setzen, um eine Verbindung zur Datenbank herzustellen. Dazu gehören der Host (Server), der Port, der Benutzername, das Passwort und der Datenbankname. Durch die Verwendung der `DbConnectionStringBuilder`-Klasse wird sichergestellt, dass die erzeugte Verbindungszeichenfolge die korrekte Syntax aufweist.[40]

5.2.2 MainWindow

Im MainWindow wird am Frontend die Tabelle angezeigt, die alle Geräte enthält. Das Backend ist daher zuständig, diese Daten zu laden und anzuzeigen.

5.2.2.1 Methode MainWindowLoad

Die Methode MainWindowLoad wird aufgerufen, wenn das Hauptfenster geladen wurde. In dieser Methode wird die Tabelle mit allen Gerätedaten befüllt. Hierfür wird ein NpgsqlCommand-Objekt erstellt, das die SQL-Abfrage definiert.

Quelltext 5.1: Abfrage einer Tabelle

```
conn.Open();
NpgsqlCommand DBQuery = new NpgsqlCommand("SELECT * FROM
device_list WHERE out_of_service = false", conn);
NpgsqlDataAdapter iAdapter = new NpgsqlDataAdapter(DBQuery);
DataSet DBDataSet = new DataSet();
iAdapter.Fill(DBDataSet, "LIST");
```

Alle Spalten (dargestellt durch *) werden aus der Tabelle *device_list* ausgewählt. Die Bedingung *out_of_service == false* sortiert alle Geräte aus, die nicht in Betrieb sind. Es wird ein NpgsqlDataAdapter-Objekt *iAdapter* erstellt und mit dem NpgsqlCommand-Objekt initialisiert. Der Adapter wird verwendet, um die Daten aus der Datenbank abzurufen. Diese Daten werden in einem DataSet-Objekt gespeichert.

Quelltext 5.2: Empfang von Geräte-Objekten aus der Datenbank

```
while (lstCount > i;)
{
    items.Add(new Device()
    {
        list_id = DBDataSet.Tables["LIST"].Rows[i]["id"].ToString(),
        list_device_number = DBDataSet.Tables["LIST"].Rows[i]
["device_number"].ToString(),
        list_device_type = DBDataSet.Tables["LIST"].Rows[i]
["device_type"].ToString(),
        list_type_name = DBDataSet.Tables["LIST"].Rows[i]
["type_name"].ToString(),
        list_device_locality = DBDataSet.Tables["LIST"].Rows[i]
["device_locality"].ToString(),
        list_responsible = DBDataSet.Tables["LIST"].Rows[i]
["responsible"].ToString(),
        list_qualification = qualificationDate.HasValue ?
qualificationDate.Value.ToString("yyyy-MM-dd") : null,
        list_calibration = calibrationDate.HasValue ?
calibrationDate.Value.ToString("yyyy-MM-dd") : null,
        list_maintenance = maintenanceDate.HasValue ?
```



```
        maintenanceDate.Value.ToString("yyyy-MM-dd") : null
    });
    i++;
}
dgv_list.ItemsSource = items;
```

Die gesamte Liste an Geräten wird mittels einer while Schleife durchlaufen. In jeder Schleifeniteration wird ein neues Geräteobjekt erstellt und der Liste "items" hinzugefügt. Das Geräteobjekt wird mit verschiedenen Eigenschaften initialisiert, die aus dem DataSet abgerufen werden. Nach der Schleife wird die Liste "items" als Datenquelle dem Datagrid "dgv_list" zugewiesen. Dadurch werden die Geräteobjekte in Form einer Tabelle in der Oberfläche angezeigt.

5.2.2.2 Sortierung

Ein Eventhandler für den Klick auf den "maintenance" Button sorgt für die Sortierung der Geräte nach Fälligkeitsdatum der nächsten Wartung.

Quelltext 5.3: Sortierung von Geräten nach Wartungsdatum

```
private void maintenance_Click(object sender, RoutedEventArgs e)
{
    int columnIndex = 9;
    RefreshWindow();
    CollectionViewSource viewSource = new CollectionViewSource();
    viewSource.Source = dgv_list.ItemsSource;

    viewSource.SortDescriptions.Clear();
    viewSource.SortDescriptions.Add(new SortDescription
    ("list_maintenance", ListSortDirection.Ascending));

    // Filter out items with null dates
    viewSource.Filter += ViewSource_Filter_maintenance;

    dgv_list.ItemsSource = viewSource.View;
}
```

Die "ItemsSource" der "dgv_list" wird als Quelle für die "CollectionViewSource" festgelegt. Dadurch können Sortierung, Filterung und andere Optionen auf die Daten angewendet werden. Die vorhandenen Sortierbeschreibungen werden gelöscht. Die Sortierung erfolgt anhand der Eigenschaft "list_maintenance" in aufsteigender Richtung. Der Event-Handler "ViewSource_filter_maintenance" wird dem Filter-Event zugewiesen. Dadurch werden Elemente mit null-Daten aus der angezeigten Sammlung herausgefiltert.

Eine ähnliche Methode bietet die gleiche Funktion für die kommenden Kalibrierungstermine.

5.2.2.3 Suchfunktion

Die Suchfunktion ermöglicht die Filterung der Gerätedaten nach Werten von bestimmten Tabellenspalten.

Quelltext 5.4: Filterung der Datenbank

```
private void Btn_search_Click(object sender, RoutedEventArgs e)
{
    string searchValue = Txt_search.Text;
    string searchColumnName = search_selection_option;
    string oos = "out_of_service";
    string query = $"SELECT *_FROM_device_listWHERE{searchColumnName}
LIKE_{searchValue}%";
    NpgsqlCommand cmd = new NpgsqlCommand(query, conn);
```

Es wird eine Variable *searchValue* erstellt und mit dem Wert des Textes im Textfeld *Txt_search* initialisiert. Danach wird eine Variable *searchColumnName* erstellt und mit dem Wert der Auswahlbox initialisiert. *searchColumnName* enthält somit den Namen der ausgewählten Spalte für die Suche. Es wird eine Variable *query* deklariert und mit der SQL-Abfrage initialisiert. Die SQL-Abfrage extrahiert alle Geräte, in denen die Spalte *searchColumnName* den gesuchten Wert *searchValue* enthält. Wenn der Wert von *searchColumnName* gleich *out_of_service* ist, werden alle Geräte gesucht, die außer Betrieb genommen wurden. Diese Funktion ermöglicht es dem Benutzer, die ausgesonderten Geräte anzuzeigen, die sonst nicht angezeigt würden. Anschließend erfolgt die Befüllung der Tabelle entsprechend der Methode *MainWindowLoad*.

5.2.2.4 Löschen

Quelltext 5.5: Löschen von Geräten aus der Datenbank

```
private void delete_Click(object sender, RoutedEventArgs e)
{
    if (dgv_list.SelectedItems.Count > 0)
    {
        int deleteID = Int32.Parse(Global.DB_id);
        string sql = "DELETE_FROM_device_list_WHERE_id=@id";
```

Es wird überprüft, ob mindestens ein Element im ListView *dgv_list* ausgewählt ist. Die globale Variable *Global.DB_id* ist die ID des ausgewählten Geräts. Nachfolgend wird eine SQL-Abfrage erstellt, um das Gerät mit der entsprechenden ID, die der ausgewählten ID entspricht, zu löschen. Anschließend wird die Tabelle aktualisiert, um sicherzustellen, dass das gelöschte Gerät nicht mehr angezeigt wird.

5.2.2.5 Erstellen

Um neue Geräte anzulegen, wird ein eigener Dialog *SaveWindow* aufgerufen.

Quelltext 5.6: Aufruf des SaveWindow Dialogs zur Erstellung neuer Geräte

```
SaveWindow savewindow = new SaveWindow ();
savewindow.Closing += SaveWindow_Closing;
savewindow.Show ();
```

Dazu wird eine neue Instanz des SaveWindow-Fensters erstellt und in der Variable "savewindow" gespeichert. Eine Ereignisbehandlungsroutine "SaveWindow_Closing" wird dem Closing-Ereignis des SaveWindow-Fensters zugeordnet. Dies bedeutet, dass der Code in der Ereignisbehandlungsroutine ausgeführt wird, wenn das Fenster geschlossen wird. Auf diese Weise wird die Tabelle aktualisiert, sobald "SaveWindow" geschlossen wird. Nun kann das SaveWindow-Fenster angezeigt werden.

5.2.2.6 Bearbeiten

Um Gerätedaten zu bearbeiten, wird das Fenster "DetailInfo" aufgerufen. Diese Methode funktioniert ähnlich wie das Erstellen eines Geräts. Der einzige Unterschied besteht darin, dass ein anderes Fenster geöffnet wird.

5.2.3 DetailInfo

Im DetailInfo-Fenster werden alle Informationen zu einem Gerät angezeigt, und es wird die Möglichkeit geboten, diese Informationen zu ändern.

5.2.3.1 Get

Um die Geräteinformationen in den Textfeldern anzuzeigen, wird die "Get" Methode aufgerufen.

Quelltext 5.7: Abfrage der Geräte ID

```
string query = $"SELECT_device_number_FROM_device_list_WHERE_id_{DB_id}";
using (NpgsqlConnection conn = new NpgsqlConnection(builder.ToString ()))
{
    conn.Open ();
    using (NpgsqlCommand cmd = new NpgsqlCommand (query, conn))
    {
        //filter null and empty object from database
        object? nuller = cmd.ExecuteScalar ();
        if (nuller != null && nuller != DBNull.Value)
        {
            string device_number_value = (string)
            cmd.ExecuteScalar ();
            Txt_device_number.Text = device_number_value;
        }
    }
}
```

Es wird eine Datenbankabfrage ausgeführt, um die *“device_number”* aus der Tabelle *“device_list”* abzufragen, basierend auf *“DB_id”*. Die Variable *“DB_id”* ist eine globale Variable, die die eindeutige ID des ausgewählten Geräts enthält. Nach der SQL-Abfrage wird geprüft, ob der zurückgegebene Spaltenwert den Werten *“null”* oder *“DBNull.Value”* entspricht. Diese Überprüfung stellt sicher, dass der abgerufene Wert gültig und nicht leer ist. Falls der Wert nicht *“null”* oder *“DBNull.Value”* entspricht, wird der abgerufene Wert der Gerätenummer in das Textfeld mit der Bezeichnung *“Txt_device_number”* geschrieben. Auf diese Weise wird die korrespondierende Gerätenummer zu der ausgewählten ID angezeigt und ermöglicht so die Anzeige der gewünschten Informationen.

5.2.3.2 Save

Um die bearbeitete Gerätenummer in der Datenbank zu speichern wird die Methode *“Save”* benutzt.

Quelltext 5.8: Speichern der Geräte ID

```
string sql_command = "UPDATE_ device_ list_ SET_ device_number_ =
@myValue_ WHERE_ id_ =_ @myID" ;
using (NpgsqlCommand command = new NpgsqlCommand(sql_command, connection))
{
    command.Parameters.AddWithValue ("@myValue" ,
    Txt_device_number.Text ?? null );
    command.Parameters.AddWithValue ("@myID" , DB_id );
    command.ExecuteNonQuery ();
}
```

Es wird ein SQL-Befehl definiert, der die Aktualisierung der Spalte *“device_number”* in der Tabelle *device_list* basierend auf *“DB_id”* durchführt. Mit *“command.Parameters.AddWithValue()”* werden die Parameterwerte gesetzt. Dabei wird der Wert des Textfeldes *“Txt_device_number”* als Wert für den Platzhalter *“@myValue”* verwendet. Der eingetragene Wert im Textfeld wird in der entsprechenden Spalte der Datenbank Tabelle gespeichert.

5.2.4 SaveWindow

Das vorliegende Fenster weist eine optische Identität mit dem DetailInfo-Fenster auf, jedoch sind alle Einträge in diesem Fenster leer, da es dazu dient, ein neues Gerät zu erstellen.

Quelltext 5.9: Speichern der Geräteinformationen

```
// create a new NpgsqlCommand object with an INSERT statement
var cmd = new NpgsqlCommand("INSERT INTO_ device_ list_ (device_number, ...
....."VALUES (@device_number_value, ...

// add parameter values to the command
cmd.Parameters.AddWithValue ("@device_number_value" ,
Txt_device_number.Text);
cmd.Parameters.AddWithValue ("@device_group_value" ,
```

```

Txt_device_group.Text);
cmd.Parameters.AddWithValue("@device_type_value",
Txt_device_type.Text ?? null);
...

```

Es wird ein `NpgsqlCommand`-Objekt erstellt, das den INSERT-Befehl enthält. Der INSERT-Befehl fügt Datensätze in die Tabelle *“device_list”* ein. Dabei werden Platzhalter verwendet, um die tatsächlichen Werte der Spalten zu ersetzen. Anschließend werden die Parameterwerte für den INSERT-Befehl festgelegt. Die Methode *“AddWithValue()”* wird verwendet, um die Platzhalter mit den entsprechenden Werten zu verbinden. Die Werte werden aus verschiedenen Textfeldern und Steuerelementen entnommen.

5.2.5 SendMail

Die wöchentliche Erinnerung an die zu bearbeitenden Geräte wird per E-Mail verschickt. Die enthält die Gerätenummern, die in dieser Woche gewartet oder kalibriert werden müssen.

Quelltext 5.10: Timer

```

// Check if it's currently Monday at midnight.
TimeSpan midnight_seconds = TimeSpan.FromHours(00) +
TimeSpan.FromMinutes(00);
TimeSpan currentTime = DateTime.Now.TimeOfDay;
if (DateTime.Now.DayOfWeek == DayOfWeek.Monday && currentTime.Hours ==
midnight_seconds.Hours && currentTime.Minutes ==
midnight_seconds.Minutes)

```

Die Variable *“midnight_seconds”* wird mit einem Zeitintervall von null Stunden und null Minuten initialisiert. Die Variable repräsentiert die Zeit 00:00 Uhr. Wenn der aktuelle Tag Montag ist und die Stunden und Minuten mit Mitternacht übereinstimmen, wird die Mail versendet.

Quelltext 5.11: Abfrage der Daten in der aktuellen Woche

```

string sql_calibration = "SELECT * FROM device_list WHERE
calibration_date >= NOW() AND calibration_date < NOW() + INTERVAL '7 days'";

```

Die SQL-Abfrage wählt alle Spalten (`SELECT *`) aus der Tabelle *“device_list”* aus, bei denen das Feld *“calibration_date”* größer oder gleich der aktuellen Zeit (*“(NOW())”*) und kleiner als die aktuelle Zeit plus sieben Tage (*“(NOW() + INTERVAL '7 days’)”*) ist. Die Abfrage stellt sicher, dass nur Datensätze ausgewählt werden, bei denen die Kalibrierungsdaten innerhalb der nächsten 7 Tage liegen.

Quelltext 5.12: Informationen in Listen ablegen

```

using (NpgsqlDataReader reader = cmd.ExecuteReader())
{
    if (reader.HasRows)
    {
        while (reader.Read())

```

```

    {
        calibration_device_number.Add(reader.GetString
            (reader.GetOrdinal("device_number")));
        calibration_dates.Add(reader.GetDateTime
            (reader.GetOrdinal("calibration_date")));
        calibration_mail.Add(reader.GetString
            (reader.GetOrdinal("mail")));
        calibration_id.Add(reader.GetInt32
            (reader.GetOrdinal("id")));
    }
}

```

Die Methode *ExecuteReader()* gibt ein *NpgsqlDataReader*-Objekt *reader* zurück, das zum Lesen der zurückgegebenen Datensätze verwendet wird. Es wird überprüft, ob der *reader* Datensätze enthält, indem *HasRows* überprüft wird. Wenn dies der Fall ist, wird eine Schleife gestartet *while (reader.Read())*, um durch die einzelnen Datensätze zu iterieren. Innerhalb der Schleife werden bestimmte Spaltenwerte aus jedem Datensatz extrahiert und in entsprechende Listen hinzugefügt. Auf diese Weise werden die relevanten Gerätedaten für den Mailversand strukturiert erfasst und können nun effizient weiterverarbeitet werden.

Quelltext 5.13: Zuordnung im dictionary

```

for (int i = 0; i < calibration_mail.Count; i++)
{
    string calibration_recipient = calibration_mail[i];
    string calibration_content = calibration_device_number[i];

    if (!email_contents.ContainsKey(calibration_recipient))
    {
        email_contents[calibration_recipient] = new Tuple<List<string>,
            List<string>>(new List<string>(){ calibration_content },
            new List<string>());
    }
    else
    {
        email_contents[calibration_recipient].Item1.Add(calibration_content);
    }
}

```

Es wird eine Schleife implementiert, die basierend auf der Anzahl der Einträge in der Liste *calibration_mail* wiederholt durchlaufen wird. Diese Liste enthält die E-Mail-Adressen der Verantwortlichen der Geräte, für die in den kommenden 7 Tagen ein Kalibrierungstermin ansteht. Die wöchentliche Erinnerung an die zu bearbeitenden Geräte wird per E-Mail verschickt. Die enthält die Gerätenummern, die in dieser Woche gewartet oder kalibriert werden müssen. In jedem Iterationsschritt der Schleife wird ein Eintrag aus der Liste der E-Mail-Adressen überprüft, ob er bereits im dictionary *email_contents* vorhanden ist. Durch die Verwendung

des dictionaries wird sichergestellt, dass jedem Verantwortlichen nur eine E-Mail gesendet wird und nicht eine E-Mail pro Gerät. Wenn die E-Mail Adresse noch nicht im dictionary enthalten ist, wird ein neuer Eintrag mit dem Empfänger und seiner Gerätenummer hinzugefügt. Wenn der Empfänger bereits im Wörterbuch enthalten ist, wird seine Gerätenummer zu dem vorhandenen Gerätenummern des dictionaries unter diesen Eintrag erweitert. Jede E-Mail Adresse befindet sich nur einmal im dictionary und den jeweiligen E-Mail-Adressen werden die entsprechenden Gerätenummern zugeordnet.

Quelltext 5.14: Inhalt der E-Mail Adresse bereitstellen

```
foreach (KeyValuePair<string, Tuple<List<string>, List<string>>>
entry in email_contents)
{
    string recipient = entry.Key;

    List<string> calibration_content = entry.Value.Item1;
    List<string> maintenance_content = entry.Value.Item2;

    string calibration_numbers = String.Join(",", calibration_content);
    string maintenance_numbers = String.Join(",", maintenance_content);
```

Während jeder Iteration der Schleife wird die aktuelle E-Mail-Adresse als string gespeichert, und die beiden zugehörigen Listen werden ebenfalls extrahiert. Eine Liste enthält die Informationen zu den Gerätenummern der zu kalibrierenden Geräte und die andere Liste enthält die Gerätenummern der zu wartenden Geräte. Anschließend werden die Elemente der Listen mit Hilfe der Methode *String.Join* zu Zeichenketten zusammengefügt. Dies geschieht sowohl für die Kalibrationsinhalte *calibration_content* als auch für die Wartungsinhalte *maintenance_content*.

Quelltext 5.15: E-Mail senden

```
MailMessage message = new MailMessage();
message.From = new MailAddress("example@mail.de");
message.To.Add(new MailAddress(recipient));
message.Subject = "Geraeteverwaltung_";
message.Body =
"Guten_Morgen,<br><br>" +
"durchzufuehrende_Kalibrationen_in_dieser_Woche:<br>" +
calibration_numbers + "<br>" +
"durchzufuehrende_Wartungen_in_dieser_Woche:<br>" +
maintenance_numbers + "<br><br>" +
"Mit_freundlichen_Gruessen,<br>" +
"Geraeteverwaltungstool";
```

Danach wird eine *MailMessage* erstellt, in der die Betreffzeile, der Textkörper und die Empfängeradresse festgelegt werden. Der Textkörper enthält Informationen über die durchzuführenden Kalibrationen und Wartungen. Schließlich wird der E-Mail-Versand über den *SmtplibClient* mit den erforderlichen Einstellungen wie Hostname, Port, Anmeldeinformationen und SSL-Konfiguration durchgeführt.

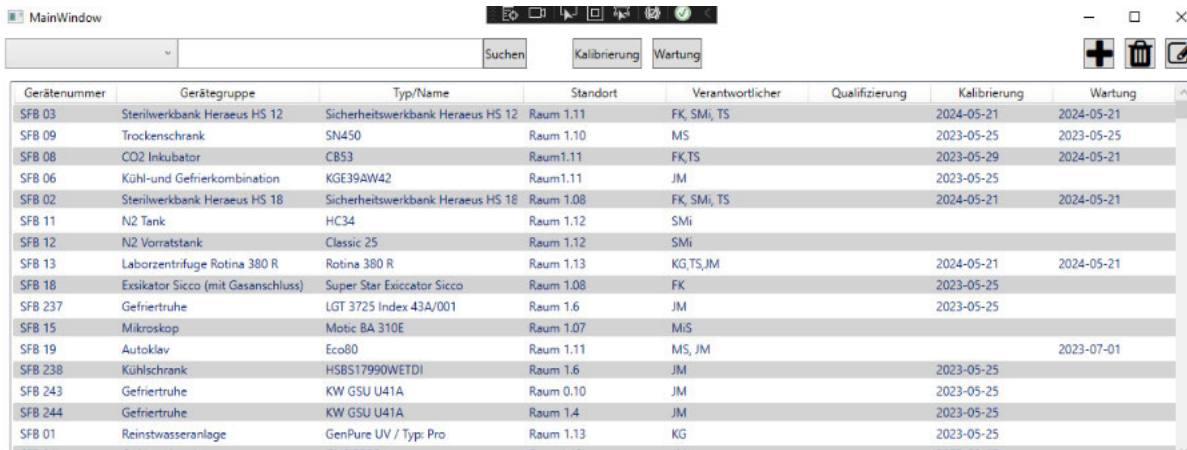
6 Testphase

Zur Evaluierung und Fehlererkennung wurde die Software mit Demodaten befüllt und den Mitarbeitern zur Verfügung gestellt. Diese Testphase ermöglichte es den Mitarbeitern, die Funktionalität der Software zu überprüfen und Feedback zu geben. Anschließend wurden die Rückmeldungen gesammelt und ausgewertet, um die Qualität und Leistung der Software zu bewerten. Um die korrekte Funktionsweise des Mailtools mit dem aktuellen Sicherheitssystem zu testen, müssen zunächst vier neue Geräte in der GUI erstellt werden. Jedes Gerät benötigt eine eindeutige Gerätenummer sowie einen verantwortlichen Benutzer mit einer zugehörigen E-Mail-Adresse. In diesem Fall sollte das wartende Personal als verantwortliche Personen eingetragen werden. Für das erste Gerät wurden Wartungs- und Kalibrierungstermine vom Vortag zugewiesen. Das zweite Gerät erhielt Termine vom aktuellen Tag, während dem dritten Gerät Termine in sieben Tagen und dem letzten Gerät Termine in acht Tagen zugeteilt wurden. Anschließend wurde die Software zur Mailversendung mithilfe einer Entwicklungsumgebung geöffnet. Dabei sollte der Zeitpunkt zur Überprüfung der Termine auf den aktuellen Tag und die aktuelle Zeit plus drei Minuten eingestellt werden. Das Projekt wurde in der Entwicklungsumgebung gestartet und für einen Zeitraum von 5 Minuten ausgeführt. Nach Ablauf dieser Zeit wurden die Inhalte des angegebenen Mailpostfachs überprüft. Wenn in der E-Mail nur die Informationen zu den Geräten des zweiten und dritten Geräts enthalten waren und durch einen Klick auf den Link in der E-Mail die Gerätetermine in der Datenbank aktualisiert wurden, galt der Test als erfolgreich. Nach Abschluss des Tests wurde das Datum zur Überprüfung der Termine wieder auf Montag, 1 Uhr, zurückgestellt. Falls keine E-Mails versendet wurden oder die falschen Gerätenummern in der E-Mail enthalten waren oder der Link die Gerätetermine nicht aktualisiert hat, musste der Code überprüft werden. Es war wichtig, den genauen Ort im Code zu identifizieren, an dem Probleme auftraten, um gezielte Anpassungen vornehmen zu können. Aus den Ergebnissen wurden anschließend neue Anforderungen abgeleitet. Um die Funktionstüchtigkeit der GUI zu testen, galt es sämtliche Funktionen auszuführen. Ein neues Gerät wurde ohne Informationen erstellt und gespeichert. Das Gerät erhielt vollständige Testinformationen und wurde erneut gespeichert. Es musste überprüft werden, ob das Gerät in der Tabelle im Hauptfenster angezeigt wird. Zudem wurde die Gerätenummer gesucht. Manuelle Aktualisierungen der Kalibrierungs- und Wartungstermine wurden durchgeführt, wobei darauf geachtet wurde, dass die neuen Termine korrekt berechnet werden. Das Gerät wurde gelöscht, und es wurde sichergestellt, dass es nicht mehr in der Tabelle vorhanden war. Die Sortierfunktionen der Tabelle wurden getestet, wobei die frühesten Termine oben angezeigt werden sollten. Während des Tests war es wichtig, jeden Schritt genau zu dokumentieren. Falls es zu Fehlern oder unerwarteten Ergebnissen kam, mussten diese ebenfalls dokumentiert werden. Diese Fehler und ihre Ursachen wurden anschließend identifiziert und daraus neue Anforderungen abgeleitet.

6.1 Testen des Designs

Im Allgemeinen waren die Mitarbeiter mit dem Design zufrieden, da es sich an bestehenden Systemen orientierte. Die Ähnlichkeit zum vorhandenen Design führte zu einer schnellen Akzeptanz der neuen Software.

Es wurde festgestellt, dass die tabellarische Anzeige der Geräte als zu unübersichtlich empfunden wurde, da keine klare Abgrenzung der einzelnen Zeilen erkennbar war. Um diese Problematik zu beheben, wurden die Zeilen der Tabelle abwechselnd in hellgrauer und weißer Farbe hervorgehoben (s. Abbildung 6.1). Durch diese farbliche Differenzierung wurde es den Nutzern erleichtert, die Informationen zu unterscheiden und die Tabelle zu lesen.



Gerätenummer	Gerätegruppe	Typ/Name	Standort	Verantwortlicher	Qualifizierung	Kalibrierung	Wartung
SFB 03	Sterilwerkbank Heraeus HS 12	Sicherheitswerkbank Heraeus HS 12	Raum 1.11	FK, SMi, TS		2024-05-21	2024-05-21
SFB 09	Trockenschrank	SN450	Raum 1.10	MS		2023-05-25	2023-05-25
SFB 08	CO2 Inkubator	CB53	Raum 1.11	FK, TS		2023-05-29	2024-05-21
SFB 06	Kühl- und Gefrierkombination	KGE39AW42	Raum 1.11	JM		2023-05-25	
SFB 02	Sterilwerkbank Heraeus HS 18	Sicherheitswerkbank Heraeus HS 18	Raum 1.08	FK, SMi, TS		2024-05-21	2024-05-21
SFB 11	N2 Tank	HC34	Raum 1.12	SMi			
SFB 12	N2 Vorratsstank	Classic 25	Raum 1.12	SMi			
SFB 13	Laborzentrifuge Rotina 380 R	Rotina 380 R	Raum 1.13	KG, TS, JM		2024-05-21	2024-05-21
SFB 18	Exsikator Sicco (mit Gesanschluss)	Super Star Exiccator Sicco	Raum 1.08	FK		2023-05-25	
SFB 237	Gefriertruhe	LGT 3725 Index 43A/001	Raum 1.6	JM		2023-05-25	
SFB 15	Mikroskop	Motic BA 310E	Raum 1.07	MS			
SFB 19	Autoklav	Eco80	Raum 1.11	MS, JM			2023-07-01
SFB 238	Kühlschrank	H5B517990WETDI	Raum 1.6	JM		2023-05-25	
SFB 243	Gefriertruhe	KW GSU U41A	Raum 0.10	JM		2023-05-25	
SFB 244	Gefriertruhe	KW GSU U41A	Raum 1.4	JM		2023-05-25	
SFB 01	Reinstwasseranlage	GenPure UV / Typ: Pro	Raum 1.13	KG		2023-05-25	

Abbildung 6.1: Benutzeroberfläche mit Hervorhebungen

6.2 Integration des Codes

Die Software funktionierte größtenteils einwandfrei. Es traten jedoch Probleme beim Speichern von Daten in den Fenstern "DetailInfo" und "SaveWindow" auf. Durch diese Schwierigkeiten konnten die eingegebenen Daten nicht korrekt gespeichert werden. Des Weiteren stellte sich das manuelle Berechnen des nächsten Termins als umständlich und fehleranfällig heraus.

6.2.1 Speichern neuer Daten

Im Zusammenhang mit der Speicherung von Daten trat ein Fehler auf, wenn bestimmte Felder nicht ausgefüllt waren. Dies führte dazu, dass das Programm abbrach und nicht erfolgreich abgeschlossen wurde.

Quelltext 6.1: Null-Coalescing-Operator

```
cmd.Parameters.AddWithValue("@device_number_value",
    Txt_device_number.Text ?? null)
```

Die ??-Operator-Syntax wird verwendet, um den Null-Coalescing-Operator darzustellen. Dieser Operator überprüft den linken Operanden "Txt_device_number.Text" auf null. Wenn der linke Operand nicht null ist, wird er zurückgegeben. Andernfalls wird der rechte Operand "null" zurückgegeben. Auf diese Weise konnte das Problem erfolgreich gelöst werden.

6.2.2 Automatischer Termin

Im aktuellen Stand des Tools ist es erforderlich, dass der Nutzer das Intervall zum aktuellen Datum addiert und anschließend in das entsprechende Feld einträgt. Um diesen Prozess zu automatisieren und dem Nutzer die manuelle Berechnung zu ersparen, wurde ein zusätzlicher Button neben dem Datepicker implementiert. Beim Klick auf diesen Button wird das Datum der nächsten Wartung automatisch berechnet und in der Benutzeroberfläche aktualisiert. Dadurch entfällt die Notwendigkeit für den Nutzer, das Datum manuell einzugeben oder zu berechnen.

6.2.3 Standard Operating Procedure

Eine Standard operating procedures (SOP) ist ein detailliertes Dokument, das präzise Anweisungen und Richtlinien enthält, um spezifische Arbeitsabläufe oder Aufgaben innerhalb eines Unternehmens zu steuern. Der Zweck einer SOP besteht darin, eine konsistente und korrekte Umsetzung dieser Aufgaben zu gewährleisten. Durch die Bereitstellung eines einheitlichen Handlungsrahmens ermöglicht die SOP die Standardisierung von Prozessen und trägt dazu bei, menschliche Fehler auf ein Minimum zu reduzieren. Das Dokument wurde gemäß den unternehmensinternen Vorgaben verfasst und dient als Schulungsmaterial(siehe Anhang SOP).[41]

7 Validierung

Es ist von wesentlicher Bedeutung, dass eine gründliche Überprüfung durchgeführt wird, um sicherzustellen, dass alle vordefinierten Anforderungen vollständig erfüllt sind und in der Zielumgebung funktionieren. Durch die Evaluierung der Ergebnisse der Implementierung können potenzielle Mängel oder Abweichungen identifiziert und entsprechende Maßnahmen ergriffen werden, um die gewünschten Standards zu erreichen. Die Validierung ist auch ein wichtiger Aspekt zur Einhaltung der IEC 62304.

7.1 Kundenbedürfnisse

7.1.1 Ersetzung der Excel Tabelle

Die vollständige Umsetzung aller Funktionen, die in der Excel-Tabelle vorgesehen sind, soll erfüllt sein. Diese Funktionen umfassen die Suche nach verantwortlichen Personen, die Sortierung der Daten nach bevorstehenden Terminen, die übersichtliche Darstellung der Gerätedaten sowie die Möglichkeit zur Bearbeitung, Erstellung und Löschung von Geräten. Zusätzlich ist es wichtig, dass die Verwaltung außer Betrieb genommener Geräte ebenfalls in das Tool integriert wird.

Die Darstellung der Geräteliste erfolgt durch eine implementierte grafische Benutzeroberfläche. Die GUI bietet eine übersichtliche und benutzerfreundliche Darstellung der Gerätedaten in Form einer Tabelle. Um eine bessere Übersichtlichkeit zu gewährleisten, werden nur eine reduzierte Anzahl von Spalten angezeigt, die die relevanten Informationen zu den Geräten enthalten. Die umfassende Einsicht in die vollständigen Geräteinformationen bleibt gewährleistet, indem diese in einem separaten Fenster dargestellt werden.

Die Suchfunktion innerhalb ausgewählter Spalten wurde durch die Implementierung einer Suchleiste mit einer Auswahlbox realisiert. Dadurch können Nutzer gezielt nach bestimmten Geräten suchen. Es besteht nun die Möglichkeit, Geräte schnell anhand ihrer Nummer oder ihres Raums zu finden.

Die Sortierung der Geräte nach bevorstehenden Kalibrierungen oder Wartungen wurde durch die Implementierung von zwei Buttons realisiert. Durch Betätigung der jeweiligen Buttons werden die Geräte entsprechend ihres nächsten Kalibrierungs- oder Wartungstermins automatisch sortiert.

Die Funktionen zum Löschen und Erstellen von Geräten werden durch dedizierte Buttons ermöglicht. Beim Erstellen eines neuen Geräts öffnet sich ein neues Fenster mit einer leeren Vorlage, in der Benutzer die erforderlichen Informationen für das neue Gerät eingeben können.

Der Button zur Bearbeitung von Geräteinformationen öffnet eine Vorlage, die bereits mit den entsprechenden Daten des ausgewählten Geräts befüllt ist. Diese Vorlage ermöglicht es Benutzern, die vorhandenen Informationen zu überprüfen und bei Bedarf Aktualisierungen oder Änderungen vorzunehmen.

Die Einsicht in außer Betrieb genommene Geräte wird durch die Verwendung einer Auswahlbox in der Suchleiste ermöglicht. Benutzer haben die Möglichkeit, gezielt nach Geräten zu

suchen, die nicht mehr in Betrieb sind. Sobald die entsprechenden Suchkriterien eingegeben werden, werden die Geräteinformationen zu diesen außer Betrieb genommenen Geräten in der Tabelle angezeigt.

7.1.2 Automatisierte Aktualisierungen

Der Prozess der Aktualisierung von durchgeführten Wartungen und Kalibrierungen soll automatisiert werden. In der Excel-Tabelle erfordert dies eine manuelle Berechnung des Intervalls auf das Datum und eine anschließende manuelle Bearbeitung der Daten.

In der grafischen Benutzeroberfläche wurden zwei zusätzliche Button implementiert, die sich in unmittelbarer Nähe des angezeigten Termins für Wartungen und Kalibrierungen befinden. Dieser ermöglicht es, die durchgeführte Wartung oder Kalibrierung zu bestätigen. Durch das Bestätigen der Wartung oder Kalibrierung wird automatisch das Intervall bis zum nächsten Termin berechnet, basierend auf dem aktuellen Datum. Das System aktualisiert dann das angezeigte Datum und zeigt dem Nutzer den neuen Termin für die nächste Wartung oder Kalibrierung an.

7.1.3 Wöchentliche Erinnerungen

Aufgrund der Tatsache, dass viele Termine in der Liste nur einmal im Jahr auftreten und leicht von Mitarbeitern vergessen werden können, wurde die Notwendigkeit erkannt, ein automatisches Benachrichtigungssystem einzuführen.

Zu Beginn jeder Woche wird eine systematische Überprüfung durchgeführt, um festzustellen, welche Geräte in der aktuellen Woche gewartet oder kalibriert werden müssen. Während dieses Prozesses werden die verantwortlichen Mitarbeiter identifiziert, die für die Durchführung dieser Aufgaben zuständig sind. Um sicherzustellen, dass keine Termine übersehen werden, wird automatisch eine E-Mail an jeden dieser Verantwortlichen gesendet, in der sie auf die anstehenden Wartungs- oder Kalibrierungsaufgaben in dieser Woche hingewiesen werden. Die E-Mail-Benachrichtigung enthält eine vollständige Auflistung der betreffenden Geräte, die in dieser Woche gewartet oder kalibriert werden müssen.

7.2 Designanforderungen

Die Gestaltung des Systems sollte einen minimalistischen Ansatz verfolgen, der sich eng an der Struktur und dem Erscheinungsbild der Excel-Tabelle orientiert. Das Ziel besteht darin, eine konsistente Benutzererfahrung zu bieten und sicherzustellen, dass sich die Benutzeroberfläche nahtlos in die Arbeitsumgebung einfügt, ohne herauszustechen.

Das Design des Systems ist eng an die Struktur und das Erscheinungsbild der Software Excel angelehnt. Es wurde eine ähnliche Aufteilung der Benutzeroberfläche gewählt, um eine vertraute Umgebung für die Benutzer zu schaffen. Im Zentrum der Oberfläche befindet sich die große Tabelle, die den Hauptbereich der Anwendung darstellt. Diese Tabelle dient zur Darstellung von Gerätedaten und bietet eine übersichtliche Ansicht der Informationen. Das

Eingabefeld, das zur Eingabe von Suchbegriffen oder Filterkriterien dient, befindet sich oberhalb der Tabelle. Dadurch ist es leicht zugänglich. Rechts neben dem Eingabefeld befinden sich Einstellungsmöglichkeiten, die das Suchfeld ergänzen. Diese Einstellungen ermöglichen es den Benutzern, die Suche nach ihren individuellen Anforderungen anzupassen.

7.3 Anforderungen nach IEC 62304

Gemäß den Anforderungen der IEC werden zusätzliche Kriterien und Anforderungen definiert, die über die reinen Kundenbedürfnisse hinausgehen. Diese Anforderungen dienen dazu, den Zweck der Software klar zu beschreiben und sicherzustellen, dass sie den spezifischen Anforderungen der Arbeitsumgebung gerecht wird. Darüber hinaus ist es von entscheidender Bedeutung, die Sicherheit der Daten zu gewährleisten und das Risiko menschlichen Versagens zu minimieren.

Der Zweck der Software ist die Erfüllung der Kundenbedürfnisse. Die Software wird unter Verwendung der .NET-Plattform entwickelt und in der Programmiersprache C# programmiert, um eine nahtlose Kompatibilität mit dem Windows-Betriebssystem zu gewährleisten. Um die Software auszuführen, ist es erforderlich, dass der Nutzer über eine Arbeitsumgebung verfügt, die den spezifischen Anforderungen der Software entspricht. Dazu muss das .NET Framework auf dem Computer des Nutzers installiert sein. Darüber hinaus wird Windows 10 oder Windows 11 als Betriebssystem vorausgesetzt. Die Software stellt auch bestimmte Ressourcenanforderungen an die Hardware des Systems. Sie benötigt einen Arbeitsspeicher von mindestens 100 Megabyte, um ordnungsgemäß ausgeführt zu werden. Für eine erfolgreiche Verbindung zur Datenbank ist es erforderlich, dass der PC sich im lokalen Netzwerk befindet.

Es gibt noch zahlreiche Möglichkeiten, wie menschliche Fehler zu großen Problemen führen können. Diese Faktoren gilt es zu minimieren.

Die Tatsache, dass in der Software keine Sicherheitsvorkehrungen implementiert wurden und der Nutzer sich nicht autorisieren muss, stellt ein erhebliches Problem dar. Gemäß den Anforderungen der IEC 62304 sind Sicherheitsvorkehrungen und eine klare Rückverfolgbarkeit der Datenänderungen essenziell. Basierend auf dem identifizierten Mangel an Sicherheitsvorkehrungen und Rückverfolgbarkeit müssen neue Anforderungen für die Software definiert werden. Diese neuen Anforderungen sollen sicherstellen, dass die Software den Richtlinien und Standards der IEC 62304 entspricht und die erforderliche Sicherheit, Integrität und Rückverfolgbarkeit gewährleistet wird.

Basierend auf dem identifizierten Mangel an Sicherheitsvorkehrungen und Rückverfolgbarkeit müssen neue Anforderungen für die Software definiert werden. Diese neuen Anforderungen sollen sicherstellen, dass die Software den Richtlinien und Standards der IEC 62304 entspricht und die erforderliche Sicherheit, Integrität und Rückverfolgbarkeit gewährleistet wird.

7.4 Neue Anforderungen

Um menschliche Fehler weiter zu minimieren, wird vorgeschlagen, die Hauptfunktion der Oberfläche auf die Verwaltung der Geräte zu fokussieren. Ein besonderer Ansatz zur Aktualisierung der Termine besteht darin, die E-Mail-Funktion zu nutzen. Der Nutzer kann einen

Link anklicken, um zu bestätigen, dass alle aufgeführten Geräte bearbeitet wurden.

Um den Zugang nur autorisierter Nutzer zu ermöglichen, soll ein Login-System implementiert werden. Das Login-System ermöglicht es, Benutzerkonten zu erstellen und spezifische Zugriffsrechte für jeden Nutzer festzulegen.

Um die Aktualisierungen der Wartungen und Kalibrierungen rückverfolgen zu können, wird vorgeschlagen, die entsprechenden Einträge in einem Logbuch oder Protokoll festzuhalten. Es soll für jedes Gerät ein eigenes Dokument erstellt werden, auf dem die relevanten Informationen zur Wartung und Kalibrierung festgehalten werden.

7.4.1 Bestätigung per Mail

Der Link in der E-Mail soll auf eine ausführbare Datei (.exe) auf einem Netzwerklaufwerk verweisen, um den Mitarbeitern den Zugriff auf das neue Programm zu ermöglichen, ohne dass eine separate Installation erforderlich ist. Der Link in der E-Mail soll zusätzlich Parameter in Form von Geräte-IDs übernehmen. Diese IDs werden bereits in einer Liste gespeichert und müssen lediglich als Zeichenkette abgerufen und als Parameter dem Link angehängt werden.

Quelltext 7.1: Verweis auf ausführbare Datei

```
string calibration_id_string = string.Join(", ", calibration_id);
string encoded_calibration_string =
    HttpUtility.UrlEncode(calibration_id_string);
string fileLink = "file:/path/program.exe?param1="
    + encoded_calibration_string;
```

Im *"calibration_id_string"* werden die einzelnen IDs zu einer Zeichenkette verbunden, wobei die IDs durch Komma getrennt werden. Danach wird die Zeichenkette mit der Methode *"UrlEncode"* kodiert. Dadurch werden Sonderzeichen in der Zeichenkette entsprechend kodiert, um sie für die Verwendung in einer URL sicher zu machen. Es wird ein endgültiger Dateilink erstellt und enthält den Pfad zur ausführbaren Datei. An den Link wird der Parameter *"param1"* angehängt, der die zuvor kodierten Kalibrierungs-IDs enthält.

Es stellte sich heraus, dass die Weitergabe der Parameter in der angegebenen Form nicht möglich war. Dies könnte auf Sicherheitsvorkehrungen seitens des Unternehmens zurückzuführen sein.

Durch die Verwendung des Ausdrucks *"System.Security.Principal.WindowsIdentity.GetCurrent().Name;"* in C# ist es möglich, Informationen über die Identität des aktuellen Windows-Benutzers abzurufen. Dieser Code ermittelt den Namen des angemeldeten Benutzers, der am Windows PC angemeldet ist. Der erhaltene Benutzername kann dann zur Überprüfung in einer Datenbank verwendet werden. Der Name wird als Kürzel im string *"abbreviation"* gespeichert.

Quelltext 7.2: Abfrage des Verantwortlichen und deren Termine

```
string sql_calibration = "SELECT *_FROM_device_list WHERE
.....responsible_LIKE_'%" + abbreviation + "%'" + "AND_calibration_date
.....>=_date_trunc('week', _NOW()) - _interval_'1_week') + _interval_'1_day' "
    + "AND_calibration_date <_date_trunc('week', _NOW())
.....+_interval_'1_day' + _interval_'7_days'";
```


Es wird eine SQL-Abfrage gestellt, bei der mehrere Kriterien zu erfüllen sind. *“responsible LIKE %”+ abbreviation + %”* sucht nach Datensätzen, bei denen das Feld *“responsible”* eine Teilübereinstimmung mit der angegebenen Abkürzung *“abbreviation”* aufweist. *“DATE_TRUNC(‘week’, CURRENT_DATE)”* gibt das Startdatum der aktuellen Woche zurück. Das *“INTERVAL ‘1 day’”* fügt einen Tag hinzu, um sicherzustellen, dass das Datum in der aktuellen Woche liegt oder danach. In der nächsten Bedingung wird das Datum mit dem Startdatum der aktuellen Woche plus acht Tagen verglichen, um sicherzustellen, dass das Datum innerhalb der aktuellen Woche liegt und nicht in der folgenden Woche. Die IDs der betreffenden Geräte werden in einer Liste gespeichert. Gleichzeitig werden auch die Termine der Wartungen durchsucht.

Quelltext 7.3: Schleife zum Aktualisieren der Termine

```
foreach (int item in calibration_id)
{
    DateTime calibration_date = new DateTime();
    current_calibration_id = item;

    string cmd_interval = $"SELECT _calibration_interval _FROM
device_list _WHERE _id _=_{current_calibration_id}";
    using (NpgsqlConnection conn = new NpgsqlConnection
(builder.ToString()))
    {
        conn.Open();
        using (NpgsqlCommand cmd = new NpgsqlCommand(cmd_interval, conn))
        {
            int calibration_interval_value =(int)cmd.ExecuteScalar();
            calibration_date = DateTime.Today.AddDays
(calibration_interval_value);
        }
    }
}
```

Eine Schleife wird für jedes Element in der Liste durchlaufen. Eine SQL Abfrage wird gestellt, um das Kalibrierungsintervall für das aktuelle Gerät anhand seiner ID abzurufen. Das Ergebnis wird in der Variablen *“calibration_interval_value”* gespeichert. Das Kalibrierungsdatum wird berechnet, indem zum heutigen Datum *“DateTime.Today”* das Kalibrierungsintervall in Tagen *“calibration_interval_value”* addiert wird. Innerhalb der Schleife wird der neue Kalibrierungstermin in der Datenbank aktualisiert. Ein ähnlicher Prozess wird für die Aktualisierung der Wartungstermine durchgeführt.

7.4.2 Login

Es wird eine weitere Tabelle in der Datenbank erstellt, die die Benutzernamen und Passwörter der Mitarbeiter enthält. Diese Tabelle dient als Grundlage für den Login-Vorgang, um den Zugriff auf das System zu steuern. Für den Login-Vorgang wird ein neues Fenster entwickelt, das bei Programmstart automatisch geöffnet wird. Das Login-Fenster enthält zwei beschriftete Textfelder, in die der Benutzer seinen Benutzernamen und sein Passwort eingeben kann, sowie einen Button, um den Login-Vorgang auszuführen.

Quelltext 7.4: Abfrage des Passworts

```

using (NpgsqlCommand cmd = new NpgsqlCommand("SELECT _password_FROM_ user_list
WHERE_username_=_@username", conn))
{
    cmd.Parameters.AddWithValue("@username", username);
    string hashedPassword = (string)cmd.ExecuteScalar();

    if (hashedPassword != null && hashedPassword == password)
    {
        MainWindow mainWindow = new MainWindow();
        mainWindow.Show();
        Global.logged_user = username;
        Close();
    }
    else
    {
        MessageBox.Show("Falsches_Passwort_oder_Nutzername.");
    }
}

```

Zunächst wird eine NpgsqlCommand-Instanz erstellt, die eine SELECT-Abfrage für das Passwort des Benutzers aus der Datenbank ausführt, wobei der Benutzername als Parameter verwendet wird. Das Passwort wird dann in der Variable *hashedPassword* gespeichert. Als nächstes wird überprüft, ob das gehashte Passwort in der Variable *hashedPassword* nicht null ist und ob es dem eingegebenen Passwort entspricht. Wenn dies der Fall ist, wird ein neues MainWindow erstellt und angezeigt, *Global.logged_user* wird auf den Benutzernamen gesetzt.

7.4.3 Logbuch

Zur Gewährleistung der Rückverfolgbarkeit und Dokumentation der durchgeführten Wartungen und Kalibrierungen wird ein Logbuch eingeführt.

Quelltext 7.5: Speichern des aktuellen Datums in der Historie von Kalibrierungen

```

string add_newest = DateTime.Today.ToString("yyyy-MM-dd");
using (NpgsqlConnection connection = new NpgsqlConnection(builder.ToString()))
{
    connection.Open();
    string sql_command = "UPDATE_device_list_SET_qualification_history_
COALESCE(qualification_history, '') || _@myValue_WHERE_id_=_@myID";
    using (NpgsqlCommand command = new NpgsqlCommand(sql_command, connection))
    {
        command.Parameters.AddWithValue("@myValue", "|_am_" +
add_newest + "_von_" + logged_user);
        command.Parameters.AddWithValue("@myID", DB_id);
        command.ExecuteNonQuery();
    }
}

```

```

    }
}

```

Bei einer Aktualisierung der Termine wird das aktuelle Datum als String in der Variable *“add_newest”* gespeichert. Anschließend wird das UPDATE-Statement definiert, das die Spalte *“calibration_history”* mit dem neuen Wert aktualisiert. Der neue Wert wird mit dem vorhandenen Wert der Spalte verknüpft, wobei der COALESCE-Ausdruck verwendet wird, um sicherzustellen, dass auch leere Werte ordnungsgemäß behandelt werden. Der neue Wert enthält das heutige Datum, den Benutzernamen *“logged_user”* und wird mit *“|”* als Trennzeichen formatiert. Die Parameter *“@myValue”* und *“@myID”* werden mit den entsprechenden Werten aus den Variablen *“add_newest”* und *“DB_id”* versehen.

Im Fenster *“DetailInfos”* eines Geräts steht den Benutzern ein Button zur Verfügung, um das Logbuch des jeweiligen Geräts aufzurufen. Dieser Button befindet sich neben der Checkbox, die anzeigt, ob ein Logbuch für das Gerät vorhanden ist. Der Inhalt der Variable *“calibration_history_list”* wird anhand des Trennzeichens *“|”* aufgespalten und einzelne Teile werden dann dem Array *“calibration_history_array”* zugewiesen. Das Logbuch eines Geräts soll mithilfe der iTextSharp-Bibliothek als PDF exportiert werden.

Quelltext 7.6: Schreiben eines Dokuments

```

// Create a new PDF document
Document document = new Document();
// Create a new PdfWriter object and associate it with the document
PdfWriter.GetInstance(document, new FileStream
(@"C:\path_" + name + ".pdf", FileMode.Create));

// Open the document for writing
document.Open();

// Add the content to the document
iTextSharp.text.Paragraph title = new iTextSharp.text.Paragraph
("Logbuch_des_Ger tes_" + name, new Font
(Font.FontFamily.HELVETICA, 18, Font.BOLD));
document.Add(title);

// Add an empty line
document.Add(new iTextSharp.text.Paragraph("\n"));

// Add the list of qualifications
iTextSharp.text.Paragraph qualificationsTitle =
new iTextSharp.text.Paragraph("durchgef hrte_Qualifizierungen:",
new Font(Font.FontFamily.HELVETICA, 14, Font.BOLD));
document.Add(qualificationsTitle);

iTextSharp.text.List qualificationsList =
new iTextSharp.text.List(iTextSharp.text.List.UNORDERED);
qualificationsList.SetListSymbol("");

```

```
qualificationsList.IndentationLeft = 12f;  
foreach (string entry in qualification)  
{  
    qualificationsList.Add(new iTextSharp.text.ListItem($"{entry}_"));  
}
```

Es wird ein Dokumentobjekt erstellt und ein PDF-Writer wird damit verbunden, der das Dokument in eine Datei schreibt. Der Name der Datei wird basierend auf dem Gerätenamen festgelegt. Das Dokument wird geöffnet und ein Titelabsatz mit dem Namen des Geräts wird hinzugefügt. Ein weiterer Absatz mit dem Titel "durchgeführte Kalibrierungen" wird erstellt und dem Dokument hinzugefügt. Es wird eine ungeordnete Liste für die Kalibrierungen erstellt. Jeder Eintrag in der Kalibrierungs Liste wird zu einem Listenelement hinzugefügt.

8 Freigabe

Nachdem alle Anforderungen auf ihre Umsetzung geprüft und offene Anforderungen erfolgreich umgesetzt wurden, wurde das Programm einer gründlichen Testphase unterzogen, um mögliche Fehler zu identifizieren. Es wurden Funktionalitätstests durchgeführt, bei denen die einzelnen Funktionen des Tools auf ihre korrekte Ausführung überprüft werden. Nachdem das Tool erfolgreich getestet wurde, ist die Verifizierung abgeschlossen und das Tool wurde zur Inbetriebnahme freigegeben.

9 Wartungsplan

9.1 Allgemein

Während der Durchführung des Wartungsplans muss eine umfassende Dokumentation jedes Schrittes erfolgen, um die Effizienz und Nachvollziehbarkeit für die Zukunft zu sichern. Die Wartung des Tools obliegt dem Systemadministrator des Standorts Senftenberg, da dieser über die erforderlichen Administratorrechte für den direkten Zugriff auf die Datenbank verfügt. Es ist zu empfehlen, die Wartung jährlich durchzuführen, um sicherzustellen, dass alle Funktionen reibungslos ablaufen. Bei Änderungen der Verbindungsdaten zum Mailtool oder zur Datenbank sind unverzüglich entsprechende Anpassungen vorzunehmen. Bei Änderungen jeglicher Art, die Auswirkungen auf die Software oder das System haben könnten, wie beispielsweise die Einführung einer neuen Version von .NET oder Veränderungen im Sicherheitssystem, unterliegt ein Medizinproduktehersteller spezifischen Vorgaben und Richtlinien. Gemäß den regulatorischen Anforderungen ist der Hersteller dazu verpflichtet, solche Änderungen umgehend zu validieren.

Zur Durchführung der Wartung ist eine Entwicklungsumgebung für Postgresql, wie zum Beispiel pgAdmin, erforderlich. Für die Überprüfung des Codes wird eine Installation von .NET sowie eine Entwicklungsplattform für .NET benötigt. Zur Überprüfung der Funktionalität des Links in der E-Mail, ist die Outlook-App notwendig. Alle durchgeführten Wartungsaktivitäten werden ausführlich im Logbuch der Software dokumentiert. Dieses Logbuch dient als Aufzeichnung aller Wartungsarbeiten und enthält Informationen über den Zeitpunkt der Wartung, die ausgeführten Aufgaben, mögliche Änderungen oder Anpassungen, durchgeführte Tests und deren Ergebnisse sowie etwaige auftretende Probleme und deren Lösungen. Die Dokumentation im Logbuch ermöglicht eine nachvollziehbare Nachverfolgung der durchgeführten Wartungsarbeiten und dient als wichtige Informationsquelle für zukünftige Referenzen oder etwaige Weiterentwicklungen des Tools.

Um im Falle eines Datenverlustes die Wiederherstellung der Daten zu ermöglichen, wird regelmäßig ein wöchentliches Backup der Datenbank erstellt.

9.2 Wartungsziele

Nachdem das Feedback von den Nutzern des Tools gesammelt wurde, müssen diese Daten systematisch erfasst und analysiert werden. Basierend auf der Analyse des Feedbacks können neue Anforderungen formuliert und bei Bedarf Verbindungsdaten und Nutzerdaten aktualisiert werden. Die Kompatibilität des Tools mit den Sicherheitssystemen muss ebenso überprüft werden.

9.3 Wartungsprozesse

Die Implementierung der neuen Anforderungen erfordert eine sorgfältige Planung und Umsetzung, um Komplikationen mit den bestehenden Funktionen zu vermeiden. Es ist wichtig, dass die neuen Anforderungen nahtlos in das bestehende System integriert werden und die

Funktionalität des Tools nicht beeinträchtigt wird.

Die Überprüfung der Variablen für die Verbindungsdaten des Mailtools und der Datenbank erfordert den Zugriff auf das Projekt in einer Entwicklungsumgebung. Dort können die entsprechenden Konfigurationsdateien oder -einstellungen geöffnet und überprüft werden. Gegebenenfalls müssen die Verbindungsdaten aktualisiert werden. Für die Überprüfung der Nutzerdaten ist der direkte Zugriff auf die Datenbankumgebung erforderlich. Dort können die Nutzerdaten abgerufen und überprüft werden, um sicherzustellen, dass sie korrekt und aktuell sind.

Um die Funktionalität der Software zu gewährleisten, sind gründliche Tests erforderlich, in denen alle Funktionen und Komponenten auf ihre ordnungsgemäße Arbeitsweise überprüft werden. Das Vorgehen ist in Kapitel 7 "Testphase" festgehalten. Während des Testprozesses können neue Anforderungen und Verbesserungsvorschläge identifiziert werden. Die neu formulierten Anforderungen müssen gesammelt und umgesetzt werden.

Nach der Implementierung müssen die neuen Anforderungen einer gründlichen Überprüfung unterzogen werden. Diese beinhaltet eine umfangreiche Testphase, in der sowohl bestehende als auch neue Funktionen auf ihre Funktionalität überprüft werden. Dabei werden auch Extremfälle und Randbedingungen berücksichtigt, um sicherzustellen, dass das Tool in verschiedenen Szenarien zuverlässig und fehlerfrei arbeitet. Erst wenn die Implementierungen der neuen Anforderungen erfolgreich getestet wurden und keine Fehler mehr auftreten, gilt die Wartung als abgeschlossen.

10 Risikomanagement

10.1 Identifizierung

Es besteht das Risiko eines potenziellen Datenverlusts. Datenverluste können menschlichem Versagen geschuldet sein, beispielsweise wenn unbeabsichtigt Geräte gelöscht werden. Zudem besteht die Möglichkeit einer Zugriffsverweigerung auf die Datenbank, falls das firmeninterne Sicherheitssystem den Zugriff blockiert.

Die Entstehung fehlerhafter Daten kann ebenfalls auf menschliches Versagen zurückzuführen sein, beispielsweise wenn Mitarbeiter falsche Daten eingeben oder Termine für Geräte aktualisieren, die noch nicht bearbeitet wurden.

Es können Kompatibilitätsprobleme auftreten, insbesondere da der Link in einer E-Mail nur über die Office-App geöffnet und nicht über die Weboberfläche ausgeführt werden kann. Zusätzlich funktionieren weder die Aktualisierung des Gerätestatus über den Link in der E-Mail noch das Öffnen der Benutzeroberfläche, falls das .NET-Framework nicht installiert ist.

Ein weiteres Problem besteht darin, dass bei ausbleibenden E-Mails sämtliche Erinnerungen entfallen, auf die sich die Mitarbeiter verlassen.

Zudem fehlt eine angemessene Cybersecurity, wie der Schutz gegen SQL-Injektionen.

Weiterhin kann die Aktualisierung von Logindaten ausschließlich in der Datenbank vorgenommen werden. Diese Möglichkeit ist nicht für Labormitarbeiter zugänglich. Die Daten für den Mailserver sowie die Verbindungsdaten für die Datenbank sind nur im Quellcode der Anwendung änderbar.

10.2 Analyse und Bewertung

10.2.1 Risikoakzeptanz Matrix

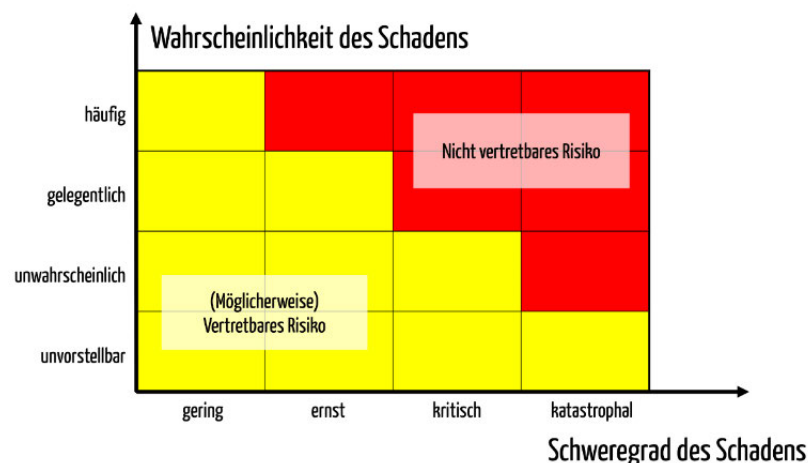


Abbildung 10.1: Risikoakzeptanz Matrix[42]

Die Risikoakzeptanz Matrix dient zur Darstellung von Risiken basierend auf ihrer Wahrscheinlichkeit und den damit verbundenen Effekten. Durch eine zweidimensionale Darstellung können Risiken grafisch in einem Diagramm visualisiert werden (s. Abbildung 10.1). Es gibt

keine festgelegte Standardform für die Matrix. Die Wahrscheinlichkeit und die Effekte werden je auf einer Achse eingetragen. Um den Graphen übersichtlich zu gestalten, wird er in drei bis sieben Stufen für jede Achse unterteilt. Eine höhere Wahrscheinlichkeit und stärkere Auswirkungen weisen auf ein ernstzunehmendes Risiko hin. Durch die Einteilung und Kategorisierung von Risiken mittels der Risikoakzeptanz Matrix können sie besser eingeordnet und bewertet werden. Dies ermöglicht es, Prioritäten zu setzen und angemessene Maßnahmen zur Risikobewältigung zu ergreifen.[42]

10.2.2 Datenverlust

Im Falle eines Datenverlusts besteht die Möglichkeit, dass die betroffenen Geräte nicht mehr in der Datenbank erfasst sind und somit nicht ordnungsgemäß überprüft werden können. Dies kann zu fehlerhaften Ergebnissen bei der Geräteprüfung führen. Es ist jedoch unwahrscheinlich, dass Geräte versehentlich vom Benutzer gelöscht werden, wodurch die Wahrscheinlichkeit für dieses Risiko als sehr gering einzustufen ist. Allerdings wäre der Effekt eines solchen Vorfalls katastrophal. Durch die großen Auswirkungen ist das Risiko nicht vertretbar (s. Abbildung 10.1).

Die Wartungen und Kalibrierungen sind sehr wichtig. Es existieren vorgeschriebene Toleranzbereiche, innerhalb derer ein Gerät funktionieren darf. Wenn die Abweichung dieser Geräte zu groß ist, können fehlerhafte Ergebnisse auftreten. Ein Beispiel dafür ist ein Gefrierschrank, der auf eine Temperatur von -80°C kühlen soll. Wenn dieser Wert nicht mehr erreicht werden kann, könnte der Inhalt des Gefrierschranks unbrauchbar werden, was zu einem finanziellen Verlust führen und die Produktion gefährden würde. Das Risiko ist wegen des großen Ausmaßes demnach nicht vertretbar (s. Abbildung 10.1).

Ein verweigerter Zugriff auf die Datenbank führt ebenfalls zu Datenverlust, der wiederum leichte bis schwerwiegende Folgen haben kann. Der Zugriff auf die Datenbank kann verweigert werden, wenn beispielsweise die Internetverbindung zum Server nicht funktioniert. In einigen Büros kann dies gelegentlich auftreten. Daher ist die Wahrscheinlichkeit nicht unvorstellbar, jedoch auch nicht unwahrscheinlich. Dieses Risiko ist nicht vertretbar (s. Abbildung 10.1).

10.2.3 Fehlerhafte Daten

Fehlerhafte Eingaben seitens des Nutzers sind durchaus möglich und können zu Problemen führen. Wenn wichtige Daten nicht korrekt eingegeben werden, können daraus Probleme resultieren. Beispielsweise kann es zu Fehlern kommen, wenn Termine ohne entsprechende Wartung oder Kalibrierung aktualisiert werden, und somit eine unzureichende Überprüfung des Geräts erfolgt. Dies kann zu fehlerhaften Ergebnissen führen. Die Wahrscheinlichkeit für eine fehlerhafte Eingabe seitens eines Labormitarbeiters ist durchaus gegeben. Die Folgen und die Wahrscheinlichkeit sind zu groß, um das Risiko als vertretbar zu erklären (s. Abbildung 10.1).

10.2.4 Kompatibilitätsprobleme

Wenn der Link in einer E-Mail nicht funktioniert, sei es aufgrund der Verwendung der Outlook-Weboberfläche oder aufgrund fehlender NET-Installation, werden die Termine nicht aktualisiert. Es werden auch in der Zukunft keine automatischen Erinnerungen generiert, und das Problem würde erst beim manuellen Sortieren der Termine in der Oberfläche bemerkt werden. Es besteht somit die Möglichkeit, dass Geräte den vorgeschriebenen Toleranzbereich verlassen, ohne dass dies von einem Mitarbeiter bemerkt wird, was erhebliche Auswirkungen auf die Laborarbeit haben kann. Die Wahrscheinlichkeit für einen solchen Vorfall ist jedoch äußerst gering. Jeder Mitarbeiter arbeitet mit einem firmeninternen Rechner, auf dem alle erforderlichen Installationen bereitgestellt werden. Zudem nutzen alle Mitarbeiter die Outlook-App, da die Anmeldung automatisch funktioniert. Dadurch werden potenzielle Kompatibilitätsprobleme aufgrund der verwendeten Office-Anwendungen und fehlender Softwarekomponenten minimiert. Das Risiko ist daher vertretbar (s. Abbildung 10.1).

10.2.5 Probleme mit der Firewall

Wenn das Mail-Tool abstürzt oder die Mails durch die Firewall blockiert werden, besteht die Gefahr, dass keine Erinnerungen angezeigt werden. Der Benutzer verlässt sich in solchen Fällen auf die ordnungsgemäße Funktion der Software. Dies könnte zu ausbleibenden Kalibrierungen und Wartungen führen, was erhebliche Auswirkungen haben könnte. Eine Änderung der Umgebung, insbesondere in Form von Softwareupdates, kann dazu führen, dass die Firewall den E-Mail-Verkehr blockiert. Es ist tatsächlich recht unwahrscheinlich, dass eine Firewall den E-Mail-Verkehr direkt blockiert. Das Risiko ist wegen des Ausmaßes trotzdem nicht vertretbar (s. Abbildung 10.1)

10.2.6 Probleme mit der Nutzerfreundlichkeit

Wenn Benutzer ihre Zugangsdaten vergessen, verlieren sie die Möglichkeit, auf die Oberfläche des Tools zuzugreifen, und können die Geräte nicht mehr verwalten. Dadurch ist es ihnen nicht mehr möglich, außerplanmäßige Wartungen und Kalibrierungen festzuhalten. Dieses Risiko ist nicht unwahrscheinlich. Es ist jedoch zu beachten, dass dieses Problem nicht unbemerkt bleibt und der Benutzer angemessen darauf reagieren kann. Das Passwort kann schnell von einem IT-Mitarbeiter aktualisiert werden. Das Risiko ist somit vertretbar, da planmäßige Wartungen und Kalibrierungen durch den Link in der Mail weiterhin möglich sind. Im Falle von Änderungen an den Verbindungsdaten des Mail-Servers oder der Datenbank können diese nur im Code aktualisiert werden. Eine solche Aktualisierung sollte nur von einem IT-Spezialisten durchgeführt werden, der auf eine umfassende Dokumentation angewiesen ist, um die erforderlichen Änderungen schnell und korrekt vorzunehmen. Es ist zu beachten, dass solche Änderungen nur äußerst selten vorgenommen werden und dass detaillierte Aufzeichnungen darüber existieren, an welchen Stellen im Code Aktualisierungen erforderlich sind. Aber es gilt zu beachten, dass bei ausbleibenden Aktualisierungen der Verbindungsdaten kein Zugriff mehr auf die Datenbank möglich ist und das gesamte System nutzlos wird. Durch die unvorstellbare Wahrscheinlichkeit ist dieses Risiko jedoch vertretbar (s. Abbildung 10.1).

10.2.7 Cybersecurity

Im Falle eines Angriffs sind das firmeninterne Sicherheitssystem und die damit verbundenen Schutzmaßnahmen vorhanden, um das System zu verteidigen. Ein erfolgreicher Angriff könnte zum Ausfall des gesamten Systems führen, was wiederum zur Folge hätte, dass die Geräte nicht mehr ordnungsgemäß kalibriert oder gewartet werden können. Die Wahrscheinlichkeit eines solchen Angriffs ist in der Regel relativ gering. Außerdem ist die Ausführung nur innerhalb des lokalen Netzwerks möglich. Somit ist das Risiko wegen der unvorstellbaren Wahrscheinlichkeit vertretbar (s. Abbildung 10.1).

10.3 Risikominimierung

Damit ein problemloser Umgang mit dem Geräteverwaltungstool ermöglicht wird, ist eine Minimierung der im vorherigen Kapitel aufgeführten Risiken angebracht.

Um versehentliches Löschen von Geräten zu verhindern, kann eine Sicherheitsabfrage implementiert werden, die den Nutzer auffordert, zu bestätigen, dass das Gerät tatsächlich gelöscht werden soll. Durch diese Sicherheitsabfrage wird eine zusätzliche Hürde geschaffen, um unbeabsichtigtes Löschen zu vermeiden. Wenn trotz der Sicherheitsabfrage ein Gerät fälschlicherweise gelöscht wird, obwohl es noch vorhanden sein sollte, können die Daten aus dem Backup geladen werden.

Ferner trägt die Implementierung von Nutzerrollen in einem System dazu bei, den potenziellen Missbrauch durch menschliche Benutzer weiter zu minimieren. Durch die Zuweisung spezifischer Rollen und Berechtigungen können die Zugriffsrechte auf bestimmte Funktionen und Daten eingeschränkt werden. Dadurch wird gewährleistet, dass nur autorisierte Benutzer die Möglichkeit haben, bestimmte Aktionen wie das Löschen von Geräten durchzuführen. Um Nutzern die Möglichkeit zu geben, vergessene Nutzerdaten selbst zu aktualisieren, könnte eine Funktion implementiert werden, die es ihnen ermöglicht, ihr Passwort nach Authentifizierung zu ändern.

Beim Speichern von Daten, die manuell in der GUI geändert wurden, kann eine Überprüfungsabfrage implementiert werden, um den Benutzer aufzufordern, die eingegebenen Daten zu überprüfen und sicherzustellen, dass sie korrekt sind. Hierbei könnte die Implementierung von Eingabevorschlägen in Betracht gezogen werden. Diese Funktion würde dem Nutzer bei der Eingabe von Daten unterstützen und ihm anzeigen, wie bestimmte Eingaben aussehen sollten. Ein solcher Ansatz könnte beispielsweise bei der Erstellung neuer Geräteinformationen angewendet werden und dem Nutzer unter anderem die nächste verfügbare Geräte-nummer vorschlagen.

Damit eine problemlose Verbindung zur Datenbank und zum Server sichergestellt werden kann, wäre es angebracht, das WLAN-Netzwerk in allen Büros auszubauen, um eine zuverlässige und funktionstüchtige Internetverbindung bereitzustellen.

Zur frühzeitigen Erkennung und Behebung potenzieller Probleme ist es ratsam, regelmäßige Überprüfungen der Gerätedaten in der GUI vorzunehmen. Dabei sollten die Geräte nach anstehenden Terminen sortiert werden, um möglicherweise versäumte Kalibrierungen und Wartungen zu identifizieren.

Ein weiteres Problem stellt der Link in der E-Mail dar. Wenn der Link beim Anklicken keine Aufforderung zur Bestätigung der Programmausführung anzeigt, wird das Programm nicht

gestartet. Das Fehlen dieses Feedbacks sollte dem Nutzer auffallen. In der SOP ist die korrekte Vorgehensweise zur Aktualisierung per Link in der E-Mail beschrieben, damit die Nutzer es in solchen Fällen nachlesen können.

Um die Verbindungsdaten zum Mailtool und zur Datenbank stets auf dem neuesten Stand zu halten, ist eine umfassende Dokumentation erforderlich. Diese Dokumentation sollte im Falle einer Änderung der Verbindungsdaten auf die entsprechende Aktualisierung im Tool hinweisen. Falls es trotz einer Aktualisierung der Verbindungsdaten keine entsprechende Umsetzung im Tool gibt, erfolgen regelmäßige Wartungen, um mögliche Probleme zu erkennen. Während dieser Wartungen sollte auch die Funktionalität der Schnittstellen, einschließlich der Verbindung zur Datenbank und zum Mailtool, überprüft werden.

Ebenfalls wird im Rahmen einer Wartung die Kompatibilität des Geräteverwaltungstools mit dem aktuellen Sicherheitssystem überprüft, um mögliche Störungen zu identifizieren und zu beheben.

11 Ergebnisse

Das entwickelte Tool erfüllt vollumfänglich die Kundenbedürfnisse und Anforderungen gemäß den betreffenden Normen, wie in Kapitel 8 beschrieben. Es bietet eine benutzerfreundliche grafische Benutzeroberfläche, die alle Funktionen der vorherigen Excel-Tabelle bereitstellt. Dabei wurden die Funktionen teilweise verbessert, um eine einfachere Handhabung und bessere Übersichtlichkeit zu gewährleisten. Durch die Reduzierung der Spaltenanzahl ist die Tabelle deutlich übersichtlicher gestaltet. Die Sortierfunktionen wurden optimiert, sodass vergangene Termine im Falle einer vergessenen Wartung oder Kalibrierung an oberster Stelle angezeigt werden. In der Excel-Tabelle war es bisher nur möglich, alle Wartungen und Kalibrierungen der zukünftigen Monate anzuzeigen. Durch die Einführung einer Datenbank werden die Datentypen in einer Spalte einheitlich gehandhabt, anstatt dass pro Spalte verschiedene Formate verwendet werden. Dadurch wird die Verwaltung vereinfacht und einheitlich gestaltet.

Die Implementierung der automatischen Erinnerungsfunktion per E-Mail für bevorstehende Wartungs- und Kalibrierungstermine wurde erfolgreich umgesetzt. Zudem besteht die Möglichkeit, die angegebenen Geräte mit einem Klick nach ihrer Wartung oder Kalibrierung zu aktualisieren, ohne die Software separat starten zu müssen. Des Weiteren wurde ein Logbuch eingeführt, um eine Historie der durchgeführten Wartungen und Kalibrierungen zu erstellen. In diesem Logbuch werden Informationen darüber festgehalten, welcher Mitarbeiter das jeweilige Gerät zu welchem Zeitpunkt kalibriert oder gewartet hat. Dadurch wird eine umfassende Nachverfolgbarkeit und Dokumentation der durchgeführten Arbeiten gewährleistet. Insgesamt hat das entwickelte Tool somit sämtliche Anforderungen erfüllt und bietet eine verbesserte und effiziente Lösung für das Geräte-Management. Die Nutzer profitieren von einer benutzerfreundlichen Oberfläche, einer übersichtlichen Tabelle, automatischen Erinnerungen, einfachen Aktualisierungsmöglichkeiten und einer umfassenden Historie der durchgeführten Wartungen und Kalibrierungen.

12 Ausblick

Es besteht das Potenzial, das Geräte-Management-Tool durch das Qualitätsmanagement zu validieren und offiziell in die bestehenden Handlungsabläufe zu integrieren. Die Validierung durch das Qualitätsmanagement ist ein umfangreicher Prozess, der sicherstellt, dass das Tool den geltenden Qualitätsstandards und -anforderungen entspricht.

In zukünftigen Weiterentwicklungen des Geräte-Management-Tools eröffnen sich verschiedene Möglichkeiten zur Erweiterung und Optimierung der Funktionalitäten. Ein vielversprechender Aspekt ist dabei die Integration der im Logbuch des Tools erfassten Daten in das Deckblatt des offiziellen Logbuchs, das in der Qualitätsmanagement-Software "Roxtra" verwaltet wird. Roxtra ist eine umfassende Softwarelösung für das Qualitätsmanagement, die unter anderem die Möglichkeit bietet, Dokumente normgerecht zu verwalten und zu lenken.[43] Eine automatische Aktualisierung der Roxtra-Dokumente hätte zweifellos erhebliche Vorteile in Bezug auf den Arbeitsaufwand und das Risiko menschlicher Fehler beim manuellen Übertragen von Daten.

Um die Effizienz bei der Standortsuche von Geräten zu verbessern, kann die Integration von Gerätebildern in die Detailinformationen Abhilfe schaffen. Durch die visuelle Darstellung der Geräte innerhalb des Tools können Benutzer schnell und einfach die gewünschten Geräte lokalisieren. Insbesondere neue Mitarbeiter, die sich noch nicht gut genug mit dem Equipment der Labore auskennen, würde diese Funktion unterstützen.

Außerdem wäre die Implementierung einer Exportfunktion der Datenbank in das Excel-Format eine Unterstützung bei der Aufrechterhaltung der Aktualität und Genauigkeit der Excel-Tabelle. Durch die automatische Übertragung der Datenbankinhalte in das Excel-Format wird die Notwendigkeit manueller Aktualisierungen und damit verbundener repetitiver Aufgaben eliminiert. Dies reduziert das Risiko von Fehlern, die bei manuellen Eingriffen auftreten können.

Anhang A: SOP Aktualisierung von Wartungs -und Kalibrierungstermin



Aktualisierung von Wartungs -und Kalibrierungstermin

Code	PR-PA-SOP
D	29481
Revision	001/02.2023
Status	freigegeben
Freigegeben am	23.02.2023



Inhaltsverzeichnis

1	Metainformation	2
1.1	Zweck	2
1.2	Anwendungsbereich	2
1.3	Verantwortlichkeiten	2
1.4	Prozessbeteiligte	2
1.5	Empfänger	2
1.6	Schulung	2
1.7	Prozesseigner	2
1.8	Begriffe	2
2	Manuelle Aktualisierung	3
2.1	Übersicht	3
2.1.1	Phase 1- Wartung und Kalibrierung	3
2.1.2	Phase 2 - Suche	3
2.1.3	Phase 3 - Aktualisierung	4
3	Suchen nach eigenen Verantwortungen	4
3.1	Übersicht	4
3.1.1	Phase 1- Suche	5
3.1.2	Phase 2 – Sortierung	5
3.1.3	Phase 3 – Kalibrierung und Wartung	5
4	Aktualisierung per E-Mail	6
4.1	Übersicht	6
4.1.1	Phase 1- Kalibrierung und Wartung	6
4.1.2	Phase 2 - Bestätigung	6
5	Anhang	7
5.1	Audit- Trail	7



Aktualisierung von Wartungs -und Kalibrierungstermin

Code	PR-PA-SOP
D	29481
Revision	001/02.2023
Status	freigegeben
Freigegeben am	23.02.2023



1 Metainformation

1.1 Zweck

Die vorliegende SOP dient als umfassende Anleitung für den Umgang mit dem Geräte-Management-Tool. Sie umfasst sowohl den Prozess der wöchentlichen Erinnerungen per E-Mail als auch die Handhabung der Benutzeroberfläche zur Verwaltung der Geräte.

1.2 Anwendungsbereich

Die SOP ist speziell für den Bereich der Geräteverwaltung am Standort Senftenberg entwickelt worden.

1.3 Verantwortlichkeiten

Der Produktionsleiter ist für die effektive Durchführung dieses Prozesses und für den Inhalt und die Aktualisierung dieser SOP verantwortlich. Jeder Mitarbeitende in der Produktion von Proteinen muss sich an die Vorgaben dieser SOP beim Produktionsprozess halten.

Diese SOP muss vom QM auf Vollständigkeit, Korrektheit und Angemessenheit überprüft und freigegeben werden.

1.4 Prozessbeteiligte

Jeder Mitarbeiter, der für ein bestimmtes Gerät verantwortlich ist, hat sich strikt an die festgelegten Vorgaben zu halten, wenn es um die Aktualisierung von Wartungs- und Kalibrierungsterminen geht. Diese SOP muss vom QM auf Vollständigkeit, Korrektheit und Angemessenheit überprüft und freigegeben werden.

1.5 Empfänger

- Mitarbeiter des Standorts Senftenberg

1.6 Schulung

Der Produktionsleiter ist für die Schulung der Mitarbeitenden verantwortlich:

Jeder neue Mitarbeiter ist initial zu schulen. Der Produktionsleiter entscheidet, ob, wenn ja wann, wie oft und in welcher Form Schulungen bzw. Nachschulungen notwendig sind.

1.7 Prozesseigner

- Produktionsleiter

1.8 Begriffe

Wir definieren jegliche Begriffe und Abkürzungen im Dokument Glossar [ID 805](#). Zusätzlich dürfen spezielle Begriffe hier definiert werden.



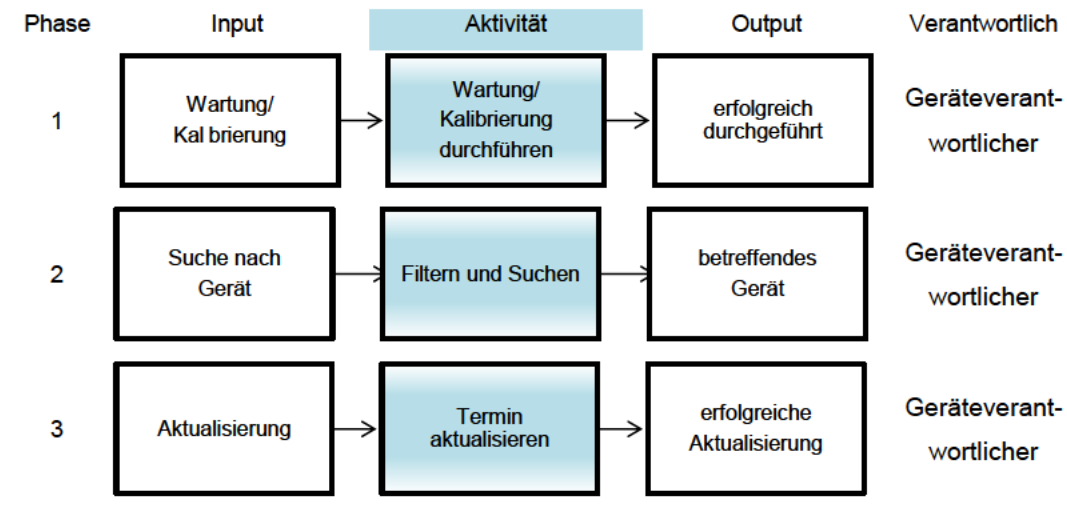
Aktualisierung von Wartungs -und Kalibrierungstermin

Code	PR-PA-SOP
D	29481
Revision	001/02.2023
Status	freigegeben
Freigegeben am	23.02.2023



2 Manuelle Aktualisierung

2.1 Übersicht





2.1.1 Phase 1- Wartung und Kalibrierung

Input:	<ul style="list-style-type: none"> Wartung/Kalibrierung
Aktivität:	Gemäß den entsprechenden Vorschriften und Richtlinien des Geräts ist die Wartung bzw. Kalibrierung gemäß den festgelegten Verfahren durchzuführen. Es ist wichtig, dass jeder Schritt des Wartungs- oder Kalibrierungsprozesses sorgfältig dokumentiert wird.
Output:	<ul style="list-style-type: none"> Erfolgreich durchgeführte Wartung/Kalibrierung



2.1.2 Phase 2 - Suche

Input:	<ul style="list-style-type: none"> Suche nach Gerät
Aktivität:	<p>Zunächst muss das Geräte-Management-Tool gestartet werden. Nach dem Start wird der Nutzer aufgefordert, sich mit seinen Zugangsdaten einzuloggen, um Zugriff auf die Funktionen des Tools zu erhalten.</p> <p>Die Option "Gerätenummer" soll in der Auswahlbox ausgewählt werden. Die Gerätenummer des zu wartenden bzw. kalibrierten Geräts wird in das entsprechende Eingabefeld eingegeben. Die Eingaben sind mit den "Suchen" Button zu bestätigen, um die Suche nach dem Gerät zu starten.</p>
	<div style="border: 1px solid gray; padding: 5px; display: inline-block;"> Gerätenummer Suchen </div>

	Aktualisierung von Wartungs -und Kalibrierungstermin	Code	PR-PA-SOP	
		D	29481	
		Revision	001/02.2023	
		Status	freigegeben	
		Freigegeben am	23.02.2023	

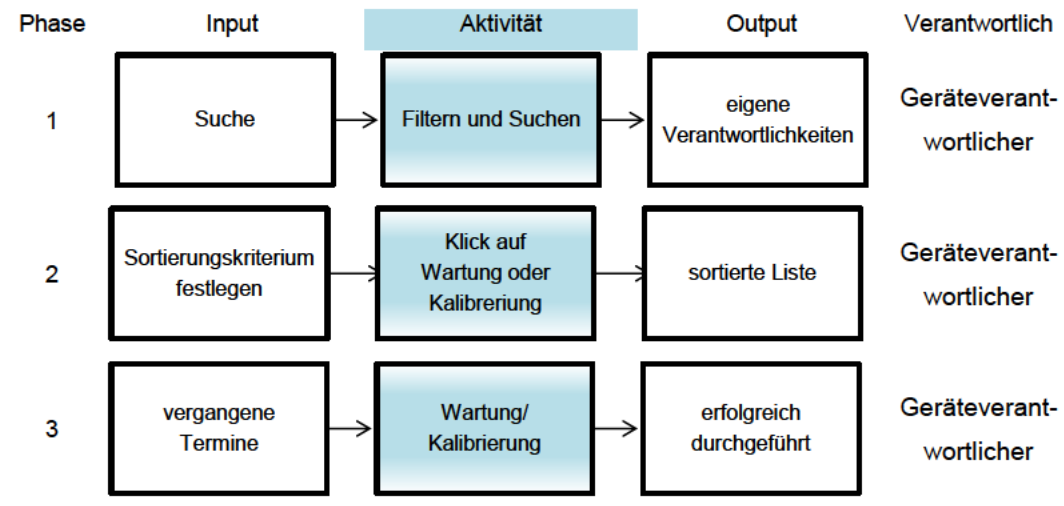
Output:	<ul style="list-style-type: none"> Betreffendes Gerät
----------------	--

2.1.3 Phase 3 - Aktualisierung

Input:	<ul style="list-style-type: none"> Aktualisierung
Aktivität:	<p>Die Auswahl des Geräts erforderlich. Dies erfolgt durch das Anklicken des entsprechenden Geräts in der Benutzeroberfläche des Geräte-Management-Tools. Das Bearbeitungsfenster für das Gerät ist aufzurufen.</p> <div style="text-align: center;">  </div> <p>Abhängig davon, ob eine Wartung oder Kalibrierung abgeschlossen wurde, ist der "Refresh" Button, der sich hinter dem entsprechenden Wartungs- oder Kalibrierungstermin befindet, zu betätigen.</p> <div style="border: 1px solid gray; padding: 5px; margin: 5px 0;"> Kalibrierungspflichtig: <input checked="" type="checkbox"/> </div> <div style="margin: 5px 0;"> Intervall: <input type="text" value="30"/> Tage </div> <div style="margin: 5px 0;"> nächster Termin: <input type="text" value="22.05.2023"/> <input type="button" value="15"/>  </div> <p>Die Aktualisierung des Termin muss gespeichert werden.</p>
Output:	<ul style="list-style-type: none"> Erfolgreiche Aktualisierung

3 Suchen nach eigenen Verantwortungen

3.1 Übersicht






Aktualisierung von Wartungs -und Kalibrierungstermin

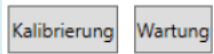
Code	PR-PA-SOP
D	29481
Revision	001/02.2023
Status	freigegeben
Freigegeben am	23.02.2023



3.1.1 Phase 1- Suche

Input:	<ul style="list-style-type: none">Suche
Aktivität:	<p>Zunächst muss das Geräte-Management-Tool gestartet werden. Nach dem Start wird der Nutzer aufgefordert, sich mit seinen Zugangsdaten einzuloggen, um Zugriff auf die Funktionen des Tools zu erhalten.</p> <p>Die Option "Verantwortlicher" soll in der Auswahlbox ausgewählt werden. Das eigene Kürzel muss in das entsprechende Eingabefeld eingegeben. Die Eingaben sind mit den "Suchen" Button zu bestätigen, um die Suche zu starten.</p> 
Output:	<ul style="list-style-type: none">Eigene Verantwortlichkeiten

3.1.2 Phase 2 – Sortierung

Input:	<ul style="list-style-type: none">Sortierungskriterium festlegen
Aktivität:	<p>Durch einen Klick auf die Schaltfläche "Wartung" erfolgt eine Sortierung der Geräte basierend auf dem frühesten Wartungstermin. Ein Klick auf die Schaltfläche "Kalibrierung" sorgt für Sortierung der Geräte nach dem frühesten Kalibrierungstermin. Beide Optionen sind durchzuführen.</p> 
Output:	<ul style="list-style-type: none">Sortierte Liste

3.1.3 Phase 3 – Kalibrierung und Wartung

Input:	<ul style="list-style-type: none">Vergangene Termine
Aktivität:	<p>Im Falle, dass Wartungs- oder Kalibrierungstermine angezeigt werden, die bereits vergangen sind, ist es von entscheidender Bedeutung, dass die Wartungs- oder Kalibrierungsaktivitäten schnellstmöglich durchgeführt werden. Die Schritte gemäß dem Abschnitt "2 Manuelle Aktualisierung" sind durchzuführen.</p>
Output:	<ul style="list-style-type: none">Erfolgreich durchgeführte Kalibrierung/Wartung



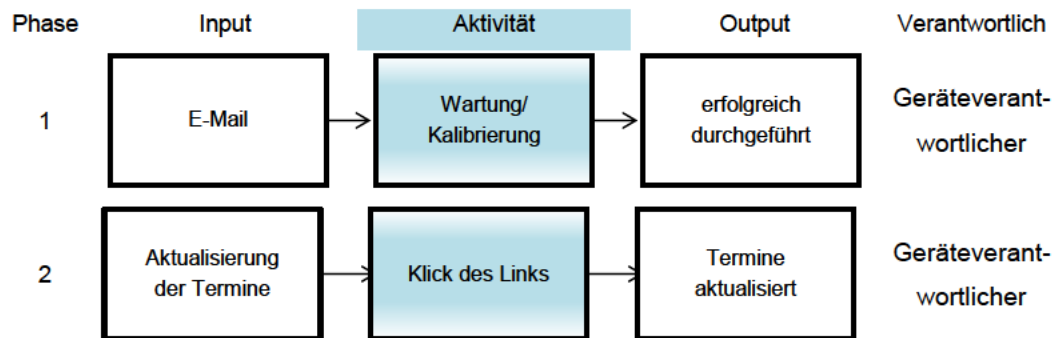
Aktualisierung von Wartungs -und Kalibrierungstermin

Code	PR-PA-SOP
D	29481
Revision	001/02.2023
Status	freigegeben
Freigegeben am	23.02.2023



4 Aktualisierung per E-Mail

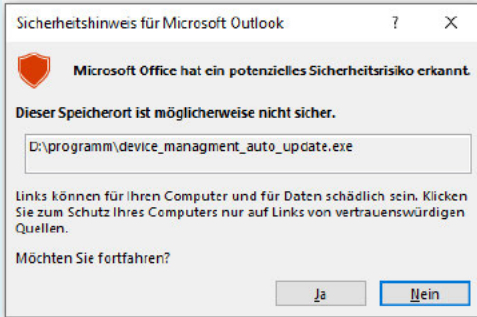
4.1 Übersicht



4.1.1 Phase 1- Kalibrierung und Wartung

Input:	<ul style="list-style-type: none"> E-Mail
Aktivität:	Wenn ein Benutzer eine E-Mail vom Geräte-Management-Tool erhält, die darauf hinweist, dass in dieser Woche Wartungs- oder Kalibrierungsaktivitäten anstehen, ist es verpflichtend, die in der E-Mail aufgeführten Geräte gemäß den vorgeschriebenen Vorschriften und Richtlinien zu bearbeiten. Dabei müssen alle erforderlichen Schritte sorgfältig dokumentiert werden.
Output:	<ul style="list-style-type: none"> Erfolgreich durchgeführte Wartung/Kalibrierung

4.1.2 Phase 2 - Bestätigung

Input:	<ul style="list-style-type: none"> Aktualisierung der Termine
Aktivität:	<p>Es ist erforderlich durch einen Klick auf den bereitgestellten Link in der E-Mail zu klicken, um die durchgeführten Wartungen und Kalibrierungen zu bestätigen.</p>  <p>Daraufhin öffnet sich ein entsprechendes Fenster, das den Nutzer fragt, ob er den Vorgang fortsetzen möchte. Um die Fortsetzung zu bestätigen, wählt der Nutzer die Option "Ja" aus und klickt darauf.</p>

Literaturverzeichnis

- [1] Aristoteles, *Die Philosophie des Aristoteles. Zweiter Band*. Felix Meiner Verlag, 2019.
- [2] „Automatisierung“. (), Adresse: <https://de.wikipedia.org/wiki/Automatisierung> (besucht am 16. 05. 2023).
- [3] „Was ist Automatisierung“. (), Adresse: <https://www.brickmakers.de/glossar/was-ist-automatisierung> (besucht am 16. 05. 2023).
- [4] „Was ist Automatisierungssoftware?“ (), Adresse: <https://www.automationanywhere.com/de/rpa/automation-software#:~:text=Automatisierungssoftware%5C%20ist%5C%20in%5C%20erster%5C%20Linie,n%5C%3%5C%A4chste%5C%20Schritt%5C%20ist%5C%20die%5C%20Prozessautomatisierung> (besucht am 16. 05. 2023).
- [5] „Automatisierung“. (), Adresse: <https://agile-unternehmen.de/automatisierung-vorteile-unternehmen/> (besucht am 16. 05. 2023).
- [6] „Software Standards, Normen und Modelle“. (), Adresse: <https://www.qz-online.de/a/grundlagenartikel/software-standards-normen-und-modelle-320738> (besucht am 17. 05. 2023).
- [7] „Grundlagen der Normung“. (), Adresse: <https://www.dke.de/de/normen-standards/grundlagen-der-normung> (besucht am 16. 05. 2023).
- [8] I. O. for Standardization, *ISO TR 80002-2 Software Validierung fuer medizinische Geraete*. International Organization for Standardization, 2017.
- [9] I. O. for Standardization, *IEC 62304 Medizingeräte-Software - Software-Lebenszyklus-Prozesse*. International Organization for Standardization, 2007.
- [10] „ISO 25010 und ISO 9126“. (), Adresse: <https://www.johner-institut.de/blog/iec-62304-%5C%20medizinische-software/iso-9126-und-iso-25010/> (besucht am 19. 05. 2023).
- [11] „Intuitive ISO 25010 Software-Qualitätssicherung mit digitalen Checklisten“. (), Adresse: <https://lumiformapp.com/de/ressourcen-checklisten/iso-25010-zertifizierung%5C#wichtig> (besucht am 19. 05. 2023).
- [12] „ISO 25010 und ISO 9126“. (), Adresse: <https://www.johner-institut.de/blog/iec-62304-%5C%20medizinische-software/iso-9126-und-iso-25010/> (besucht am 19. 05. 2023).
- [13] D. Chappell, *.NET verstehen . Einführung und Analyse*. Addison-Wesley, 2002.
- [14] „.Net-Framework“. (), Adresse: <https://de.wikipedia.org/wiki/.Net-Framework> (besucht am 10. 05. 2023).
- [15] „Was ist Visual Studio? Verständlich erklärt“. (), Adresse: <https://praxistipps.chip.de/was-ist-visual-studio-verstaendlich-erklart> 41914 (besucht am 10. 05. 2023).
- [16] „Die .NET UI Technologieüberishct 2022“. (), Adresse: <https://schwabencode.com/blog/2022/01/30/net-ui-overview-2022#:~:text=WPF%5C%20hat%5C%20noch%5C%20viel%5C%20Potential,WPF%5C%20am%5C%20Leben%5C%20zu%5C%20erhalten> (besucht am 10. 05. 2023).

- [17] „Desktopteitfaden (Windows Forms .NET)“. (), Adresse: <https://learn.microsoft.com/de-de/dotnet/desktop/winforms/overview/?view=netdesktop-7.0> (besucht am 11.05.2023).
- [18] D. Frischalowski, *Windows Presentation Foundation. WPF - Grafische Oberflächen entwickeln mit .NET 3.0*. Addison-Wesley, 2007.
- [19] „Was ist XAML?“ (), Adresse: <https://wpf-tutorial.com/de/5/xaml/was-ist-xaml/> (besucht am 12.05.2023).
- [20] „WPF vs. WinForms“. (), Adresse: <https://wpf-tutorial.com/about-wpf/wpf-vs-winforms/> (besucht am 11.05.2023).
- [21] „Uno Platform documentation“. (), Adresse: <https://platform.uno/docs/articles/intro.html> (besucht am 11.05.2023).
- [22] „Cross-Platform-Apps mit der Uno Platform“. (), Adresse: <https://entwickler.de/dotnet/cross-platform-apps-mit-der-uno-platform#:~:text=Die%5C%20Uno%5C%20Platform&text=Das%5C%20bedeutet%5C%20unter%5C%20anderem%5C%2C%5C%20dass%2C,auf%5C%20WebAssembly%5C%20und%5C%20Xamarin%5C%20auf> (besucht am 12.05.2023).
- [23] „PostgreSQL“. (), Adresse: <https://de.wikipedia.org/wiki/PostgreSQL> (besucht am 12.05.2023).
- [24] S. B. N. Ramez Elmasri, *Grundlagen von Datenbanksystemen: Ausgabe Grundstudium*. Addison-Wesley, 2005.
- [25] „Was ist PostgreSQL?“ (), Adresse: <http://postgresql.de/was-ist-postgresql> (besucht am 12.05.2023).
- [26] „PostgreSQL Subqueries“. (), Adresse: <https://www.w3resource.com/PostgreSQL/postgresql-subqueries.php> (besucht am 12.05.2023).
- [27] „Constraints“. (), Adresse: <https://www.postgresql.org/docs/current/ddl-constraints.html> (besucht am 12.05.2023).
- [28] „Foreign Keys“. (), Adresse: <https://www.postgresql.org/docs/current/tutorial-fk.html> (besucht am 12.05.2023).
- [29] „Multi-Version Concurrency Control“. (), Adresse: <https://www.postgresql.org/docs/7.1/mvcc.html> (besucht am 12.05.2023).
- [30] „PostgreSQL Triggers“. (), Adresse: <https://www.postgresqltutorial.com/postgresql-triggers/> (besucht am 12.05.2023).
- [31] „Functions and Operators“. (), Adresse: <https://www.postgresql.org/docs/current/functions.html> (besucht am 12.05.2023).
- [32] „Was ist PostgreSQL?“ (), Adresse: <https://kinsta.com/de/wissensdatenbank/was-ist-postgresql/> (besucht am 12.05.2023).
- [33] „The DatePicker control“. (), Adresse: <https://wpf-tutorial.com/misc-controls/the-datepicker-control/> (besucht am 14.05.2023).
- [34] M. R. Thomas Erler, *UML. Das bhv Taschenbuch*. vmi Buch, 2002.
- [35] „ListView Overview“. (), Adresse: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/controls/listview-overview?view=netframeworkdesktop-4.8> (besucht am 14.05.2023).

- [36] „GridView Overview“. (), Adresse: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/controls/gridview-overview?view=netframeworkdesktop-4.8> (besucht am 14.05.2023).
- [37] „Data binding overview (WPF .NET)“. (), Adresse: <https://learn.microsoft.com/en-us/dotnet/desktop/wpf/data/?view=netdesktop-7.0> (besucht am 14.05.2023).
- [38] „Übersicht über TextBox“. (), Adresse: <https://learn.microsoft.com/de-de/dotnet/desktop/wpf/controls/textbox-overview?view=netframeworkdesktop-4.8> (besucht am 14.05.2023).
- [39] „ComboBox Klasse“. (), Adresse: <https://learn.microsoft.com/de-de/dotnet/api/system.windows.controls.combobox?view=windowsdesktop-7.0> (besucht am 14.05.2023).
- [40] „DbConnectionStringBuilder Class“. (), Adresse: <https://learn.microsoft.com/en-us/dotnet/api/system.data.common.dbconnectionstringbuilder?view=net-7.0> (besucht am 29.05.2023).
- [41] „What is a Standard Operating Procedure (SOP)?“ (), Adresse: [https://www.brampton.ca/EN/Business/BEC/resources/Documents/What%5C%20is%5C%20a%5C%20Standard%5C%20operating%5C%20Procedure\(SOP\).pdf](https://www.brampton.ca/EN/Business/BEC/resources/Documents/What%5C%20is%5C%20a%5C%20Standard%5C%20operating%5C%20Procedure(SOP).pdf) (besucht am 23.05.2023).
- [42] „Risikoakzeptanzmatrix – Risikobewertungsmatrix“. (), Adresse: <https://www.johner-institut.de/blog/iso-14971-risikomanagement/risikoakzeptanzmatrix-risikobewertungsmatrix/> (besucht am 25.05.2023).
- [43] „Die Software für Ihr digitales Qualitätsmanagement“. (), Adresse: <https://www.roxtara.com/> (besucht am 28.05.2023).

Eidesstattliche Erklärung

Hiermit versichere ich – Ben Noack – an Eides statt, dass ich die vorliegende Arbeit selbstständig und nur unter Verwendung der angegebenen Quellen und Hilfsmittel angefertigt habe.

Sämtliche Stellen der Arbeit, die im Wortlaut oder dem Sinn nach Publikationen oder Vorträgen anderer Autoren entnommen sind, habe ich als solche kenntlich gemacht.

Diese Arbeit wurde in gleicher oder ähnlicher Form noch keiner anderen Prüfungsbehörde vorgelegt oder anderweitig veröffentlicht.

Mittweida, 02. Juni 2023

Ort, Datum



Ben Noack