

ABSCHLUSSPRÜFUNG WINTER 2021

FACHINFORMATIKER FÜR ANWENDUNGSENTWICKLUNG

DOKUMENTATION ZUR BETRIEBLICHEN PROJEKTARBEIT

Entwicklung einer allgemeinen Ressourcenverwaltung

FÜR DIAGNOSTIKGERÄTE VIA GRPC

PRÜFUNGSBEWERBER:

Lukas Klettke

Am Mühlenteich 17

23611 Bad Schartau

10. November 2021

Inhaltsverzeichnis

Abbildungsverzeichnis	II
Tabellenverzeichnis	III
Listings	IV
Abkürzungsverzeichnis	V
1 Einleitung	1
1.1 Vorstellung der eigenen Person	1
1.2 Vorstellung des Ausbildungsbetriebs	1
1.3 Projektauslöser	1
1.4 Projektumfeld	2
1.5 Projektziel	2
1.6 Projektschnittstellen	3
2 Projektplanung	3
2.1 Projektphasen	3
2.2 Ist-Analyse	3
2.3 Soll-Konzept	4
2.4 „Make or Buy“	4
2.5 Kosten- und Ablaufplanung	4
2.5.1 Wirtschaftlichkeitsprüfung	4
2.5.2 Projektkosten	4
2.5.3 Amortisationsdauer	5
2.6 Qualitätsanforderungen	5
3 Entwurfsphase	6
3.1 Zielplattform	6
3.2 Qualitätssicherung	6
4 Realisierung	6
4.1 Eingesetzte Technologien	6
4.2 Entwicklungsumgebung	7
4.3 Erstellung einer Benutzeroberfläche	7
Eidesstattliche Erklärung	8
A Anhang	i

Abbildungsverzeichnis

Tabellenverzeichnis

1	Zeitplanung	3
2	Kostenaufstellung	5

Listings

Abkürzungsverzeichnis

RPC	Remote Procedure Call
gRPC	Open-Source „Remote Procedure Call“ System, entwickelt von Google
SoC	System-on-a-Chip
WDDM	Windows Display Driver Model
NuGet	Paketmanager basierend auf dem .NET Framework
Prism	Application Framework für WPF
WPF	Windows Presentation Foundation
IoC	Inversion of Control - Programmierprinzip
MVVM	Model View ViewModel

1 Einleitung

1.1 Vorstellung der eigenen Person

Mein Name ist Lukas Klettke. Ich bin am 14.01.2001 in Lübeck geboren und in Bad Schwartau aufgewachsen. Dort habe ich die Grundschule und das Leibniz Gymnasium besucht. Nach zwölf Jahren Schulzeit habe ich meinen Schulweg im Jahre 2019 mit dem Abitur abgeschlossen.

Direkt nach Abschluss der Schule habe ich im August 2019 eine Ausbildung zum Fachinformatiker für Anwendungsentwicklung begonnen und bin in der Softwareentwicklung für Diagnostikgeräte tätig.

In meiner Freizeit bin ich als ehrenamtlicher Schwimmtrainer tätig, segle und fahre Rennrad.

1.2 Vorstellung des Ausbildungsbetriebs

Mein Ausbildungsbetrieb ist die EUROIMMUN Medizinische Labordiagnostika AG mit Sitz in 23560 Lübeck und Zweigstellen in Groß Grönau, Selmsdorf und Dassow im Norden und Rennersdorf, Pegnitz und Bernstadt im Süden Deutschlands. Durch den Verkauf der Firma im Dezember 2017 befindet sich EUROIMMUN in Besitz von PERKINELMER Inc., einem US-amerikanischen Technologieunternehmen im Bereich der Chemie- und Medizintechnik.

EUROIMMUN ist ein Hersteller für diverse medizinische Diagnostika von Autoimmun-, Infektionskrankheiten und Allergien, aber auch im Bereich der Automatisierung. Die Ausbildung findet in Dassow in der Forschung und Entwicklung von Software zur Steuerung von Diagnostikautomaten statt.

Insgesamt hat EUROIMMUN mehr als 3.200 Mitarbeiter in 17 Ländern.

1.3 Projektauslöser

Neben der Herstellung von medizinischen Diagnostika zur manuellen Durchführung, werden Geräte zur automatisierten Durchführung hergestellt und vertrieben. Diese Diagnostikautomaten arbeiten mit diversen unterschiedlichen Betriebsmitteln (z.B. Reinigungsflüssigkeit zur Reinigung der Schläuche, Probenträger, etc.), welche ebenfalls von EUROIMMUN an die Kunden verkauft werden.

Anhand der verbrauchten Betriebsmittel der Geräte, wird die Menge, der in Zukunft benötigten, berechnet und die Preise dementsprechend auf den Kunden angepasst. Außerdem wird je nach Verbrauch der Labore die Produktions- und Lagermenge optimiert.

Derzeit werden jedoch keine Daten der Verbräuche von den Gerätesoftwarens erhoben, was eine manuelle Berechnung derer zur Folge hat. Diese Berechnung wird durch Außendienstmitarbeiter durchgeführt, welche die Labore besuchen und die durchgeführten Testmengen als Maßstab nutzen. Je nach Art des Gerätebetriebs ist der Verbrauch jedoch unterschiedlich: Werden z.B. 500 Tests am Stück durchgeführt, ist das Verhalten des Geräts ein Anderes, als wenn über eine Zeit von zwei Wochen 500 Tests durchgeführt werden. Somit ist bei der Berechnung eine gewisse Ungenauigkeit vorhanden, die in Zusammenhang mit einer großen Anzahl an Kunden Differenzen zwischen berechneten und realen Verbräuchen verursacht.

Durch eine automatisierte und genauere Berechnung mithilfe von protokollierten Ressourcenverbräuchen könnten Punkte wie Preisgestaltung, Produktionsmenge oder Lagerhaltung weiter optimiert werden und somit Geld einsparen bzw. Gewinn maximieren.

1.4 Projektumfeld

Das Projektumfeld ist der EUROIMMUN Standort in Dassow. Dort befindet sich ein Teil der Entwicklung der Diagnostikgeräte und der zugehörigen Software.

Bei diesem Projekt handelt es sich um eine Software, die ausschließlich intern eingesetzt werden soll.

1.5 Projektziel

Ziel des Projekts ist es eine Schnittstelle zu definieren, die unabhängig von Diagnostikgerät und der entsprechenden Software implementiert werden kann. Durch diese Schnittstelle wird definiert, in welcher Form die Verbrauchsdaten abgefragt werden.

Anhand dessen wird eine Software geschrieben, die die Ressourcenverbräuche verarbeitet, eine Gesamtberechnung durchführt und den Export einer „.xlsx“ ermöglicht, um die nachstehende Kalkulation mittels Microsoft Excel zu gewährleisten.

Die Planung eines solchen Projekts existiert bereits mehrere Jahre und wurde von der Geschäftsführung in Auftrag gegeben. Das Ziel dessen ist es Daten über die Nutzung der EUROIMMUN Diagnostikgeräte zu erheben, welche zur Analyse von weiteren Optimierungsmöglichkeiten dienen.

1.6 Projektschnittstellen

Das Projekt stellt eine **gRPC** Schnittstelle bereit. Diese wird mithilfe einer „.proto“ Datei definiert. Die Schnittstelle wird auf Seite des Clients implementiert und auf Seite des Servers offen gelassen, sodass die unterschiedlichen Softwares der Diagnostikgeräte diese implementieren können und die Freiheit haben, je nach Architektur und Speicherung, die Daten bereitzustellen.

Des Weiteren wird der Export einer „.xlsx“ Datei angeboten.

2 Projektplanung

2.1 Projektphasen

Tabelle 1 zeigt die vorgesehenen Phase des Projektes.

Projektphase	Geplante Zeit
Projektplanung	7 h
User-Interface Grundfunktionen	5 h
Implementierung der Anwendungslogik	40 h
Testen und Nacharbeiten	10 h
Dokumentation	8 h
Gesamt	70 h

Tabelle 1: Zeitplanung

2.2 Ist-Analyse

Zur Zeit besuchen Außendienstmitarbeiter Labore, die Diagnostikgeräte von EUROIMMUN verwenden, um die Ressourcenverbräuche zu berechnen. Um weiterhin genügend Betriebsmittel vorzuhalten, werden retrospektiv die Verbräuche entsprechend zur Zeit berechnet und die weitere Versorgung sicher gestellt. Diese Berechnung geschieht anhand der Anzahl durchschnittlich durchgeführter Tests und Aussagen der Labormitarbeiter, falls absehbar ist, dass zukünftig vom Durchschnitt abgewichen wird.

Diese Arbeitsweise erfordert gut geschulte Mitarbeiter.

2.3 Soll-Konzept

Durch die neu entwickelte Software soll es den Außendienstmitarbeitern möglich sein die Ressourcenverbräuche genauer zu berechnen, den zukünftigen Bedarf präziser zu planen, somit die Preisgestaltung anzupassen und die Produktion und Lagerhaltung zu optimieren. Der Mitarbeiter kann mit einem Windows PC sich in das lokale Netzwerk des Labors einwählen und über die gegebene Schnittstelle eine Verbindung mit den Computern der Diagnostikautomaten aufbauen. Mithilfe dieser Verbindung werden die Verbräuche der Betriebsmittel abgefragt.

Das Programm soll für die Möglichkeit auf internationale Anwendung in Englisch geschrieben werden und auf einem Windows 7/8/10/11 PC laufen. Die Software wird generisch entwickelt, sodass der zukünftige Ausbau mithilfe anderer Technologien möglich ist. Sowohl die Seite des Diagnostikgeräts, als auch die des Außendienstmitarbeiters werden in C# entwickelt.

2.4 „Make or Buy“

Da der Anwendungsfall spezifisch für von EUROIMMUN entwickelte Diagnostikgeräte gilt, gab es keine käufliche Software, die den Ansprüchen gerecht wird.

2.5 Kosten- und Ablaufplanung

2.5.1 Wirtschaftlichkeitsprüfung

Die Wirtschaftlichkeit wird in den beiden folgenden Punkten genau beschrieben.

2.5.2 Projektkosten

Die Projektkosten werden mit einigen variablen Parametern betrachtet, da echte Daten seitens der Geschäftsleitung nicht herausgegeben werden. Aus diesem Grund wird mit fiktiven Stundensätzen gearbeitet. Der Stundensatz eines Auszubildenden wird mit 75 EUR und der eines Mitarbeiters mit 100 EUR angesetzt. In diesen Stundensätzen sind neben Gehaltszahlungen Kostenbeiträge wie Lohnnebenkosten und Sozialbeiträge enthalten. Die im Unternehmen typischen Gemeinkosten, wie Miete, Reinigungskosten der Räumlichkeiten oder Abschreibungen auf das technische Equipment werden zusätzlich mit 15 EUR/Stunde angesetzt. Das Projekt wurde mit 70 Stunden angesetzt, woraus sich ein Gesamtbudget von 6.300 EUR ergibt (s. Tabelle 2).

Vorgang	Zeit	Kosten pro Stunde	Kosten
Entwicklungskosten	70 h	75 €	5250 €
Gemeinkosten	70 h	15 €	1050 €
Gesamt			6300 €

Tabelle 2: Kostenaufstellung

Kosten für die genutzte Hardware werden nicht angesetzt, da davon ausgegangen wird, dass diese schon vorhanden ist.

2.5.3 Amortisationsdauer

Die Amortisation des Projekts ist nicht das primäre Ziel. Wie oben genannt dient es zur Analyse von Daten in Bezug auf das Nutzungsverhalten der Kunden und zukünftige Optimierungen. Aus diesem Grund ist keine Amortisationsdauer zu berechnen.

Durch die hohe Anzahl an genutzten Diagnostikgeräten von EUROIMMUN in Laboren auf der ganzen Welt, ist der Gewinnverlust hochgerechnet auf ein Gerät sehr gering. Bei einer Anzahl von beispielsweise 5.000 verkauften Geräten beträgt der Gewinnverlust ca. 1,25 EUR pro Gerät.

2.6 Qualitätsanforderungen

Da die Anwendung unter Anderem die Kosten für Mitarbeiterschulungen senken soll, muss sie einfach und intuitiv zu bedienen sein. Bei geringfügigen Fehlern soll die Software ausfallsicher sein und durch die Abfrage von unbestimmten Größen an Datenmengen soll die Kommunikation performant und schlank sein.

Die Bereitstellung einer frei implementierbaren Schnittstelle verlangt eine gute Dokumentation und Kommentierung des Quellcodes, sodass Entwickler diese Schnittstelle einfach nutzen können. Da die Softwares zur Steuerung der Diagnostikgeräte für Windows entwickelt werden, ist es nötig die Schnittstelle ebenfalls für Windows zu entwickeln und daraus resultierend auch die Software zur Abfrage der Betriebsmittelverbräuche.

3 Entwurfsphase

3.1 Zielplattform

Für die 64 Bit Version von Windows 10 wird ein Prozessor mit mindestens 1 GHz Arbeitsleistung oder ein SoC benötigt. Die Mindestkapazität des RAM liegt bei 2 GB, der Festplattenspeicher muss mindestens 32 GB groß sein. Die Grafikkarte muss über DirectX 9 oder höher mit einem WDDM 1.0 Treiber verfügen.

3.2 Qualitätssicherung

Kommt noch...

4 Realisierung

4.1 Eingesetzte Technologien

Für die Entwicklung der Schnittstelle, mit der die Verbräuche der Betriebsmittel der Diagnostikgeräte abgefragt werden, wird gRPC verwendet. gRPC ist ein Open-Source RPC System, welches von Google entwickelt wird.

RPC ist eine Technologie, die es möglich macht, Prozeduren auf anderen Geräte auszuführen, als auf dem, wo es aufgerufen wird (meistens innerhalb eines Netzwerks). Durch diese Auslagerung kann Datenverarbeitung auf andere Geräte ausgelagert werden, ohne dass ein Unterschied in der Entwicklung entsteht. Daraus ergibt sich eine Form der Client-Server-Architektur, bei der der aufrufende Part den Client und der ausführende Part den Server darstellt. Die Kommunikation basiert auf dem „request-response“ Protokoll, welche synchron via http abläuft. Schickt der Client eine Abfrage, ist er während der Bearbeitung durch den Server blockiert¹.

Die Wahl der Technologie fiel auf gRPC, da dieses bereits in der Firma genutzt wurde und somit eine Vorgabe darstellte.

Die Anwendung wird mit C# entwickelt. Dafür wird die Version 4.8 des .NET Frameworks verwendet.

¹ Vgl. Wikipedia: https://en.wikipedia.org/wiki/Remote_procedure_call (Stand 08.11.2021 10:15 Uhr)

Zum exportieren von „.xlsx“ Dateien wird das [NuGet](#)-Paket „Microsoft.Office.Interop.Excel“ verwendet. Die Dokumentation befindet sich auf der Microsoft Docs Webseite² des Frameworks.

Als Hilfe zur Implementierung wird [Prism](#) eingesetzt. Durch die Nutzung ergibt sich eine einfach Einbindung von [IoC](#). Mithilfe von [IoC](#) können Konstruktoren automatisch initialisiert werden, woraus sich eine deutlich erhöhte Übersichtlichkeit des Programmcodes ergibt.

4.2 Entwicklungsumgebung

Zur Entwicklung wird Visual Studio Professional 2019 verwendet. Die IDE wurde durch die ReSharper Extension von JetBrains erweitert. ReSharper fügt Tools für Refactoring und das Erkennen von Code Smells hinzu. So wird gewährleistet, dass sauberer und qualitativ hochwertiger Quellcode produziert wird.

4.3 Erstellung einer Benutzeroberfläche

Die Oberfläche wurde mit [WPF](#) entworfen. Als Architekturmodell wird ein Hybrid aus zwei [MVVM](#) Architekturen angewendet. Die einzelnen Bestandteile (im Folgenden „Views“ genannt) verwenden je eine [MVVM](#) Architektur, womit ist die Benutzeroberfläche von der Logik abgekoppelt ist. Die Werte werden mittels Data Binding in die Screens integriert. Um die einzelnen Views in der Oberfläche zu verwenden, wird über ein einzelnes ViewModel kontrolliert, welche View gerade zu sehen ist.

Für die Erstellung eines einheitlichen Designs wird die EUROIMMUN interne Design Bibliothek genutzt, die nach der Material Design Sprache von Google geschrieben ist. Implementiert ist diese ebenfalls in [WPF](#) und lässt sich durch Auslagerung in sogenannte „Resource Dictionaries“ einbinden.

² Microsoft.Office.Interop.Excel: <https://docs.microsoft.com/en-us/dotnet/api/microsoft.office.interop.excel?view=excel-pia>

Eidesstattliche Erklärung

Ich, Lukas Klettke, versichere hiermit, dass ich meine **Dokumentation zur betrieblichen Projektarbeit** mit dem Thema

Entwicklung einer allgemeinen Ressourcenverwaltung für Diagnostikgeräte via gRPC

selbständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, wobei ich alle wörtlichen und sinngemäßen Zitate als solche gekennzeichnet habe. Die Arbeit wurde bisher keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Lübeck, den 01.12.2021

LUKAS KLETTKE

A Anhang