****

Abschlussprüfung Sommer 2024

Fachinformatiker für Anwendungsentwicklung

Dokumentation zur betrieblichen Projektarbeit

**Lagerverwaltungssoftware**

**Prüfungsbewerber:**

Angeline Loeding

Alexander Kalinka

Björn Loges

**Projektbetreuer:**

Tom Dallmeyer

**EPSolutions GmbH**

Georg-Kerschensteiner-Straße 27

23554 Lübeck

Ein Bild, das Text, Screenshot, Design, Schrift enthält.

Automatisch generierte Beschreibung

Inhalt

[1. Einleitung 3](#_Toc168312197)

[1.1. Projektumfeld 3](#_Toc168312198)

[1.2. Projektbeschreibung 3](#_Toc168312199)

[1.3. Projektziel 4](#_Toc168312200)

[1.4. Projektbegründung 4](#_Toc168312201)

[1.5 Projektbegrenzung 4](#_Toc168312202)

[1.6 Projektschnittstellen 4](#_Toc168312203)

[2. Planungsphase 5](#_Toc168312204)

[2.1. Projektphasen 5](#_Toc168312205)

[2.2. Ressourcenplanung 5](#_Toc168312206)

[2.3. Entwicklungsprozess 6](#_Toc168312207)

[3. Analysephase 6](#_Toc168312208)

[3.1. Soll-/Ist-Analyse 6](#_Toc168312209)

[3.2 Qualitätsanforderungen 7](#_Toc168312210)

[3.3 Risikoanalyse 7](#_Toc168312211)

[3.3.1 “Make-or-Buy“ – Entscheidung 8](#_Toc168312212)

[3.3.2 Projektkosten 8](#_Toc168312213)

[3.4 Business Case 9](#_Toc168312214)

[3.4.1 Projektübersicht: 10](#_Toc168312215)

[3.4.2 Investitionskosten 10](#_Toc168312216)

[3.4.3 Einnahmen 10](#_Toc168312217)

[3.4.4 Gewinn 10](#_Toc168312218)

[3.4.5 Return on Investment (ROI) 10](#_Toc168312219)

[3.4.6 Fazit 11](#_Toc168312220)

[4 Entwurfsphase 11](#_Toc168312221)

[4.1 Zielplattform 11](#_Toc168312222)

[4.2 Entwurf der Benutzeroberfläche 11](#_Toc168312223)

[4.3 Datenmodell 12](#_Toc168312224)

[4.4 Pflichtenheft 12](#_Toc168312225)

[4.5 Architekturdesign 12](#_Toc168312226)

[5. Implementation 13](#_Toc168312227)

[5.1. Implementation von Datenstrukturen 13](#_Toc168312228)

[5.2. Implementierung der Anwendung 13](#_Toc168312229)

[6. Fazit 14](#_Toc168312230)

[6.1 Lessons Learned 14](#_Toc168312231)

[6.2 Ausblick 14](#_Toc168312232)

[7. Anhang 15](#_Toc168312233)

[7.1 A1R\_ Ressourcen 15](#_Toc168312234)

[7.1.1 Hardware 15](#_Toc168312235)

[7.1.2 Software 15](#_Toc168312236)

[7.2 A2ERM\_Entity-Relationship-Model 15](#_Toc168312237)

[7.3 A3PH\_Pflichtenheft 15](#_Toc168312238)

[7.3.1 Zielgruppe 15](#_Toc168312239)

[7.3.2 Grobablauf 16](#_Toc168312240)

[7.3.3 Designs 17](#_Toc168312241)

# Einleitung

Im Rahmen der Ausbildung zum Fachinformatiker für Anwendungsentwicklung haben die Autoren das IHK-Abschlussprojekt durchgeführt, dessen Ablauf in der folgenden Projektdokumentation detailliert erläutert wird.

## Projektumfeld

Die EPSolutions GmbH wurde 2020 als GmbH gegründet und hat sich seitdem in der Branche für Softwarelösungen im Lernbereich etabliert. EPSolutions ist ein Softwareunternehmen und ein führender Anbieter von Lehrsoftware im Großraum Lübeck. Mit über 50 Mitarbeitern betreuen wir mehr als 3700 Kunden.

Bei EPSolutions wird unter anderem das System „Frontalunterricht“ entwickelt und vertrieben, das speziell auf die Schüler und deren Bedürfnisse zugeschnitten ist. Darüber hinaus wird Software rund um Homeschooling und Verwaltung erstellt.

Das Projekt „Inventar-App“, auch „Lagerverwaltungssoftware“ genannt, wird in den Büros der Emil-Possehl-Schule in Lübeck, Deutschland, im Auftrag der IHK Lübeck durchgeführt. Das Projekt wird von Tom Dallmeyer begleitet und kontrolliert. Eine abschließende Qualitätskontrolle erfolgt durch die Qualitätssicherung.

## Projektbeschreibung

In der Applikation wird zunächst eine Startseite angezeigt, in der der Benutzer die Möglichkeit hat, Optionen auszuwählen, ob gescannt werden soll oder ob nur das Inventar angezeigt werden soll. Wenn der Nutzer z.B. Was scannen möchte und dementsprechend die Schaltfläche Scannen ausführt, wird die Seite zum Scannen angezeigt, in der das Scannen des Schrankes durchgeführt werden kann. Nach dem Scannen werden die Objekte angezeigt, die sich im Schrank befinden. Möchte der Benutzer das Inventar ansehen oder nach den Gegenständen suchen? Er kann die Optionen auf der Startseite auswählen.

## Projektziel

Ziel des Projektes ist es, eine flexible Plattform zu schaffen, die es dem Kunden ermöglicht, einen individuellen Überblick über seine Lagerräume und das darin gelagerte Inventar zu erhalten. Dabei soll der Datenaustausch so schnell wie möglich erfolgen und bei Ausfällen Alternativen geschaffen werden. Des Weiteren sollen die Daten zugriffssicher abgerufen oder bearbeitet werden können, um eine möglichst sichere Datenverfügbarkeit und -integrität zu schaffen. Des Weiteren soll die Benutzeroberfläche auf der Nutzerseite barrierefrei und benutzerfreundlich gestaltet werden.

## Projektbegründung

Die IHK erhofft sich von diesem Projekt eine bessere Übersicht über Ihr Inventar und damit verbunden eine Optimierung des Bestellungs- und Bevorratungsmanagements.

Für die EPSolutions GmbH besteht der Anreiz in einer dauerhaften Kundenbindung der IHK Lübeck und damit verbunden langfristige Einnahmen durch einen Service Vertrag.

## Projektbegrenzung

Aufgrund des begrenzten Projektumfangs wird die Funktion der PDF-Anzeige derzeit nicht berücksichtigt. Diese Funktion wäre nur eine Erweiterung der Bestandsanzeige, die bereits als Tabelle im Abschnitt Design im Anhang zu sehen ist. Außerdem stehen den Teammitgliedern nur 80 Stunden für die Durchführung des Projekts zur Verfügung.

## Projektschnittstellen

Die Inventar-App verfügt über eine technische Schnittstelle an eine separate Datenbank. Darüber hinaus weist die Applikation eine API-Schnittstelle zur Verwendung des inhärenten QR-Code Scanners auf.

# Planungsphase

Im nächsten Abschnitt wird beschrieben, wie das weitere Vorgehen für das Projekt bestimmt wurde. Dazu wurden zunächst die verschiedenen Projektphasen geplant, die benötigten Ressourcen ermittelt und der Entwicklungsprozess festgelegt.

## Projektphasen

Für das Projekt wurde ein Zeitrahmen von 80 Stunden zur Verfügung gestellt. Der Zeitraum des Projektes erstreckte sich über das gesamte Schulhalbjahr im Winter 2023. Dabei wird die Hauptsächliche Bearbeitungszeit zwischen dem 16.10.2023 und 27.10.2023 liegen. In dieser Zeit sind die zur Verfügung gestellten 80 Stunden frei belegbar.

|  |  |
| --- | --- |
| Zeitplanung | |
| Projektphase | **Geplante Zeit** |
| Anforderungsanalyse | 3 Stunden |
| Modellierung | 5 Stunden |
| Entwicklung | 30 Stunden |
| Implementierung | 20 Stunden |
| Feedback | 5 Stunden |
| Tests | 5 Stunden |
| Dokumentation | 12 Stunden |

Abbildung 1 Zeitplanung

## Ressourcenplanung

Die Planung berücksichtigt sowohl Hard- als auch Software-Ressourcen sowie das beteiligte Personal. Bei der Auswahl der verwendeten Software wurde darauf geachtet, dass keine Lizenzgebühren anfallen, erforderliches Fachwissen vorhanden ist und die Architekturrichtlinien der MVC eingehalten werden. Die Architekturrichtlinien der MCV legen unter anderem die Nutzung von VSCode als Entwicklungsumgebung und für die Datenbank Microsoft MySQL-Server fest. Die Einhaltung dieser Richtlinien stellt sicher, dass das Projekt den Anforderungen der MVC entspricht und nahtlos in die bestehende Infrastruktur integriert wird.

## Entwicklungsprozess

Bei der Entwicklung der „Inventar-App“ wurden die Standardphasen des Wasserfallmodells durchlaufen, aber durch die Validierung der Komponenten und Zwischenschritte ist der Prozess des V-Modells besser geeignet.

Das V-Modell zeichnet sich durch seinen linearen Ablauf und die damit verbundenen Schritte der Validierung aus. Für kleine Projekte mit geringer bis mittlerer Komplexität ist es trotz seines Alters geeignet.

Da es sich bei der App um eine rudimentäre Lagerhaltungssoftware handelt, ergibt sich eine reduzierte Komplexität.

# Analysephase

## Soll-/Ist-Analyse

Die Ist-Analyse soll aufzeigen, wie die Daten derzeit ermittelt und verarbeitet werden. Bei der Lagerhaltung gibt es derzeit keine spezielle Software zur Optimierung der Lagerhaltung und Materialverwaltung. Die Inventur und Materialverwaltung werden derzeit analog mittels Papierformularen und Excel-Listen durchgeführt.  
Dadurch ergeben sich folgende Probleme:

- Aktualisierungsprobleme können auftreten, wenn die Formulare nur lokal verfügbar sind  
- Schwierigkeiten bei der Suche und Filterung  
- Der Austausch von Inventurdaten ist begrenzt, wenn keine zentrale Plattform zur Verfügung steht, auf der alle benötigten Formulare verfügbar sind  
-Suchen von Gegenständen wären ohne Hilfe Zeitaufwendig

Zum Abschluss wird die Projektplanung retrospektiv überprüft. Dabei wird insbesondere darauf geachtet, ob die festgelegten 80 Stunden, die den Auszubildenden zur Verfügung standen, strikt eingehalten wurden. Die Funktionen des Systems wurden erfolgreich umgesetzt, wie in der ursprünglichen Anfrage und den Vorgaben des erstellten Pflichtenhefts beschrieben. Besonderes Augenmerk wurde auf die Entwicklung einer App gelegt, die das effiziente Auffinden von Orten und Gegenständen ermöglicht.

Die Anwendung bietet eine übersichtliche Darstellung von Inventur, Lagerorten und Gegenständen. Zusätzlich ermöglicht sie das Scannen von Schränken, um eine präzise Lokalisierung der Gegenstände innerhalb des gescannten Schranks in der App zu ermöglichen. Die erfolgreiche Integration dieser Funktionen gewährleistet, dass die definierten Anforderungen erfüllt wurden und das System einen wirksamen Beitrag zur Optimierung der Lagerverwaltung leistet.

## Qualitätsanforderungen

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

- Bei falschem Scannen des Schrankes soll dem Nutzer eine Fehlermeldung angezeigt werden  
-Wird keine korrekte Eingabe in der Suchleiste gemacht, wird ebenfalls eine Fehlermeldung angezeigt

- Zur Vermeidung von Bearbeitungsfehlern sollten Funktionen sauber in der Businesslogik verwendet werden

## Risikoanalyse

Ein neues Modul mit Kundenauftrag ist mit einem geringeren Risiko behaftet als ein Modul ohne Kundenauftrag, da es bereits einen Erstkäufer gibt. Somit ist das finanzielle Risiko für die EPSolutions GmbH äußerst begrenzt.  
Das Risiko, das der derzeit geplante Kostenrahmen nicht eingehalten wird, ist äußerst gering, da die durchführenden Mitarbeiter keine Vergütung für dieses Projekt erhalten.

### “Make-or-Buy“ – Entscheidung

Zwar gibt es Lösungen, welche die Anforderungen des Projektes teilweise erfüllen, doch es gibt einige Punkte, die zu einer Make-Entscheidung geführt haben.

Dazu zählt die kostengünstige Entwicklung für die IHK-Lübeck und die mit dem Auftrag einhergehende Individualisierungsmöglichkeit für den Kunden.   
  
Für die EPSoloutions GmbH ist einer der wichtigen Gründe das die beteiligten Mitarbeiter durch die Durchführung wichtige Kenntnisse erlangen und diese später im Unternehmen gewinnbringend einbringen können. Diese Kenntniszugewinne negieren die Nachteile der Eigenentwicklung.

Darüber hinaus bindet dieses neue Produkt den Kunden an das Unternehmen und es kann zu Folgeaufträgen führen.

### Projektkosten

Aufgrund von Betriebs- und Geschäftsgeheimnissen sowie aus Gründen des Datenschutzes können keine exakten Zahlen veröffentlicht werden. Alle folgenden Werte sind realistische Beispielwerte für die Kostenrechnung.

Bei Projektkosten spielen nicht nur Personal und Hardware eine Rolle, sondern auch anteilige Kosten von Lizenzen, Räumen und andere Nebenkosten spielen mit rein. Im Folgenden werden daher alle Ressourcen aus Anlage [A1R](#_A*__Ressourcen) berücksichtigt.

Nun folgt die Beispielrechnung der Kosten, bei der Projektumsetzung. Angefangen mit den Personalkosten, hier haben wir den Umsetzer des Projektes mit einem Nettolohn von 1300 Euro und den Begleitenden Entwickler, sowie den Mitarbeiter der Qualitätssicherung (QA) mit einem Nettolohn von 4000 Euro.

Um die Betriebswirtschaftlichen Personalkosten zu bekommen, gehen wir von Lohnnebenkosten, wie zum Beispiel Versicherungen, von 30% des Bruttolohns aus. Dann erhalten wir mit folgender Formel:

Einen kaufmännisch gerundeten Stundenlohn von 10€/Std. für den Auszubildenen (Umsetzer) und 35€/Std. für den Begleitenden Entwickler und QA-Mitarbeiter. Es sind keine weiteren Betriebskosten zu erwarten.

Nehmen wir nun eine Pauschale von 15€ pro Stunde für Beschaffungskosten und andere Nebenkosten, wie Lizenzen oder Hardware, welche in Anlage [A1R](#_A*__Ressourcen) aufgelistet sind, erhalten wir folgende Gesamtkosten.

|  |  |  |  |
| --- | --- | --- | --- |
| Projektkosten | | | |
| Mitarbeiter | **Geplanter Aufwand** | **Stundensatz** | **Kosten** |
| Auszubildender (Umsetzer) | 80 Stunden | 25€/Std. | 2.000€ |
| Entwickler | 5 Stunden | 50€/Std. | 250€ |
| QA-Mitarbeiter | 10 Stunden | 50€/Std. | 500€ |
| Gesamt |  |  | 2.750€ |

Abbildung 2 Kosten Rechung TT-Dashboard

## Business Case

In diesem Business Case schauen wir uns die Entwicklung und den Vertrieb der Inventar-App an.

Die Entwicklungskosten belaufen sich, wie in 3.2.2 erläutert, auf 2750€. Für diese Analyse gehen wir davon aus das EPSoloutions über 5 Kunden verfügt, die als Potenzielle Käufer angesehen werden können. Ähnliche Module haben Verkaufspreis von 15000€ pro Lizenz, daher gehen wir vom selben Preis für dieses Produkt aus.

Dieser Business Case analysiert die potenziellen Einnahmen, Gewinne und den Return on Investment (ROI) für dieses Projekt.

### Projektübersicht:

Ziel des Projektes ist es eine Lagerverwaltungssoftware und Inventar-App zu schaffen, auf dessen Basis eine benutzerfreundliche Lagerverwaltung erfolgen kann.

### Investitionskosten

Die Entwicklungskosten belaufen sich, wie weiter oben aufgezeigt, auf einmalig 2750€.

### Einnahmen

Wir nehmen an das die Software zu einem Lizenzpreis von 15.000€ pro Einheit an unsere 5 Kunden verkauft werden kann.

Gesamteinnahmen = Lizenzpreis pro Einheit \* Anzahl der Kunden

Gesamteinnahmen = 15.000 € \* 5 Kunden

Gesamteinnahmen = 75.000 €

### Gewinn

Der Gewinn wird berechnet, indem man die Gesamteinnahmen von den Investitionskosten abzieht.

Gewinn = Gesamteinnahmen - Investitionskosten

Gewinn = 75.000 € - 2750 €

Gewinn = 72.250 €

### Return on Investment (ROI)

Der ROI gibt an, wie viel Gewinn im Verhältnis zu den Investitionskosten erwirtschaftet wird.

ROI = (Gewinn / Investitionskosten) \* 100

ROI = (72.250 € / 2750 €) \* 100

ROI =2.627,27%

### Fazit

Dieser Business Case zeigt, das sich die Entwicklung der App lohnen wird. Bei möglichen Gesamteinnahmen von 75.000€ und Investitionskosten von 2.750€ kann ein Gewinn von 72.250€ erzielt werden.

Das bedeutet dieses Projekt hat das Potenzial ein ROI von 2.627.27% zu erzielen. Wobei beachtet werden muss das zusätzliche Kosten, wie z.B. Marketing, weitere Lizenzen oder Einnahmefaktoren dazukommen können. Diese können das tatsächliche Ergebnis verändern. Deutlich macht diese Analyse aber, dass es sich lohnen wird die Ressourcen in das Projekt zu investieren.

# Entwurfsphase

## Zielplattform

Da das Projekt eine Stand-Alone Lösung darstellt, ist die Zielplattform klar definiert. Es handelt sich um eine Client Anwendung, welche für den Gebrauch auf einem Mobilegerät wie Smartphones vorgesehen ist.

Das Projekt wird in C# umgesetzt. Als Datenspeicher wird entweder ein Microsoft-SQL Server, oder ein lokal abgelegtes JSON-File benutzt.

Der SQL-Server nutzt die Data Manipulation Language Transact-SQL.

## Entwurf der Benutzeroberfläche

Bei der Benutzeroberfläche wurde auf eine benutzerfreundliche, intuitive und ergonomische Oberflächengestaltung geachtet. Diese wurde mittels der kollaborativen Software FIGMA erstellt und die Designs in das Hauptprogramm implementiert.

## Datenmodell

Durch das Fachkonzept wurde ein Entity-Relationship-Model (ERM) erstellt, welches das Datenmodell abbildet. Eine Detaillierte Beschreibung und Abbildung befindet sich im Anhang [A2ERM](#_A2ERM_Entity-Relationship-Model).

## Pflichtenheft

Basierend auf dem Konzept wurde im Vorfeld des Projektes ein Pflichtenheft erstellt, in dem die geplante Vorgehensweise und die Strukturen festgehalten wurden. Im Laufe des Projektes wurde das Pflichtenheft an die sich ergebenden Änderungen angepasst. Dieses Pflichtenheft ist im Anhang unter [A3PH](#_A3PH_Pflichtenheft) angehängt.

## Architekturdesign

Wie bereits im Abschnitt Projektschnittstelle erwähnt, wird in diesem Abschnitt die Struktur der Schnittstelle erläutert. Dabei wird das MVC-Modell verwendet. Jede Komponente einer Software wird einem der drei Teile - Model, View und Controller - dieses Modells zugeordnet. Jeder dieser Teile hat eine spezifische Aufgabe.

Die View stellt die Benutzerschnittstelle dar, über die der Benutzer mit der Anwendung interagiert. Der Benutzer greift über ein Client-System (Anwendung auf PC/Handy) auf Informationen zu und gibt Eingaben ein, wie z. B. das Scannen von Objekten oder das Suchen von Objekten in der Suchleiste. Der Controller aktualisiert, koordiniert und leitet diese Eingaben an das Schichtenmodell weiter, das alle aktualisierten Daten übernimmt und speichert.

Die klare Trennung der Komponenten ermöglicht eine unabhängige Entwicklung, flexible Erweiterung und das Testen der Komponenten. So ist es beispielsweise möglich, die View-Schicht zu ändern, ohne das Modell anpassen zu müssen. Aufgrund dieser Vorteile wird dieses Architekturmodell für die Umsetzung des Projektes verwendet.

Zusätzlich wurde eine API-Schnittstelle verwendet, um den Datenaustausch zu ermöglichen. Diese Schnittstelle ermöglicht eine synchronisierte Interaktion zwischen den Komponenten Controller und View sowie dem Model, dem ORM-Modell. Das ORM ermöglicht die Generierung von Klassen basierend auf unserem Datenbankschema, die in weiteren Schritten verwendet werden können.

Möchte der Benutzer den Schrank scannen, wird die Anfrage über die API an den Controller weitergeleitet. Dieser liest den QR-Code und fragt die Daten im Modell ab. Anschließend sendet die API die Daten an die View.

# 5. **Implementation**

## 5.1. Implementation von Datenstrukturen

Derzeit ist der Software noch keine Datenbank hinterlegt, jedoch kann deren theoretischen Grundstruktur dem ERM entnommen werden. Diese ist im Anhang [A2ERM](#_A2ERM_Entity-Relationship-Model) dargestellt. Dort sind alle erforderlichen Attribute, Schlüsselspalten und Referenzen aufgeführt.

Alle Elemente der Datenstruktur würden mit T-SQL-Befehlen des Microsoft Management Systems erstellt werden.

## 5.2. Implementierung der Anwendung

Da dies eine komplette Neuentwicklung ist, wird dies in kein bestehendes System integriert.   
Die Applikation basiert auf einem Interface, um sicher zu stellen, dass alle wichtigen Grundfunktionen von zukünftigen Entwicklern zentral entwickelt und implementiert werden können.

Der Zugriff auf die Daten erfolgt direkt über einen Import und Export einer JSON-Datei.  
Der später geplante Zugriff auf die Datenbank erfolgt nicht direkt über SQL-Abfragen auf die Basistabellen, sondern ist gekapselt in Views/ Stored Procedures. Dies ist ‚Best Practise‘ in der Softwareentwicklung und dient zum einen der Wartbarkeit der Datenzugriffe, zum anderen der Wahrung von ‚Geschäftsgeheimnissen‘ und dem Schutz vor unerlaubten Datenzugriffen.

# Fazit

## Lessons Learned

Während der Umsetzung sammelten die Teammitglieder wertvolle Erfahrungen, insbesondere in Bezug auf die Zusammenarbeit an einem Projekt und die damit verbundenen Arbeitsabläufe. Die praktische Umsetzung machte die theoretischen Konzepte greifbar, die zuvor in der Prüfung behandelt wurden. Diese Phase ermöglichte nicht nur die Anwendung theoretischen Wissens, sondern förderte auch ein tieferes Verständnis der Konzepte durch die direkte Umsetzung in die Praxis.

Jedes Teammitglied brachte individuelle Kenntnisse in verschiedenen Programmiersprachen und Frameworks ein, wodurch das Wissen erheblich erweitert wurde. Das Team konnte durch die Vielfalt an Kompetenzen ein umfassenderes Verständnis für die verschiedenen Aspekte der Projektarbeit entwickeln.

## Ausblick

Aufgrund des Projektumfangs und der geplanten Zeiten wird die geplante Funktion zur Anzeige von PDFs vorerst verschoben. Wenn versucht wird, die Inventuranzeige innerhalb dieses Zeitrahmens zu verbessern, können die vorgegebenen Zeiten überschritten werden.

Da in dem Auftrag keine weiteren Entwicklungen erwähnt wurden, wird die Software vorerst nur gewartet und unterstützt, wenn Kunden auf Fehler stoßen oder eine Anleitung benötigen.

# Anhang

## 7.1 A1R\_ Ressourcen

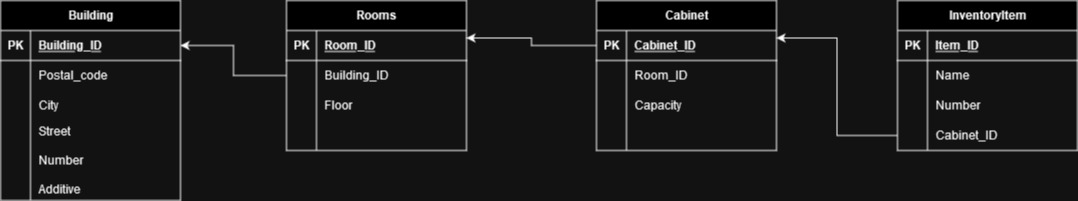
### 7.1.1 Hardware

* Arbeitsplatz mit voller Ausstattung
* Arbeitsrechner Terra

### Software

* Windows 11
* Visual Studio Code
* Structured Query Language (SQL) Server 2022
* Microsoft SQL Server Management Studio 19.0
* Cryptpad.fr

## A2ERM\_Entity-Relationship-Model



## A3PH\_Pflichtenheft

### Zielgruppe

Das Projekt ist zielgerichtet auf die Bedürfnisse unseres Erstkunden IHK-Lübeck. Das Programm ist jedoch so gestaltet, dass auch weitere Kunden diese Applikation nutzen können. Der dabei entstehende Mehraufwand zur Personalisierung der verwendeten Software wird sich auf das Austauschen von Logos und das Einpflegen der Daten beschränken. Somit sind alle kleine mittelständischen Unternehmen, die derzeit keine Lagerverwaltungssoftware innerhalb eines Büros verwenden potenzielle Kunden.

### Grobablauf

Erste Phase: Planung:

Erstellung von einer Zielbestimmung und Besprechung des Vorgehens und der Möglichkeiten die uns offenstehen. Im Anschluss Ausführung in Dokumenten oder Aktivitäten niederschreiben.

Zweite Phase: Das Design:

Hier folgt die Erstellung des Grund Designs und die verschiedenen Oberflächen der verschiedenen Seiten der Applikation.

Dritte Phase: Die Programmlogik:

Als erstes werden die Klassen und Objekte definiert sowie eine Prozedur erstellen, die die Daten aus einer angefertigten JSON-Datei auslesen kann.   
Erstellen weiterer Logik zur Verarbeitung und Darstellung der Daten.

Fünfte Phase: Testen:

Testen der Oberfläche und Logik. Entwicklung automatischer Tests.

### Designs

###### **Cover Design**

Cover:

Nachdem der Nutzer unsere App startet, wird zunächst das Cover angezeigt. Dieses beinhaltet zentrisch das Logo des Unternehmens. Die genauen Strukturen zur Bestimmung der Maße der einzelnen Komponenten werden noch im Programm festgelegt.

###### **Startseite**

Die Homepage zeigt folgende Navigationsmenüs: Inventar, Karte, Scannen, Edit und Lager. Darunter wird angezeigt auf welcher Webseite wir sind. Das highlighten des Haus Icons zeigt folglich, dass wir und auf der Homepage befinden. Zusätzlich findet der Nutzer ein Einstellungs-Tool, dass die Möglichkeit bietet die App zu konfigurieren, beispielsweise in Dark- oder Light- Mode

###### **Scanner**

Bei der Scann Seite kann der Nutzer den QR-Code des Schranks scannen.

###### **Inhalt**

Nach dem Scannen wird eine Seite geöffnet, die den Inhalt des Schrankes anzeigt. Zudem gibt es die Möglichkeit den dazu gehörigen Lageort anzeigen zu lassen oder zurück zu der Scann Seite zu kehren.

###### **Karte**

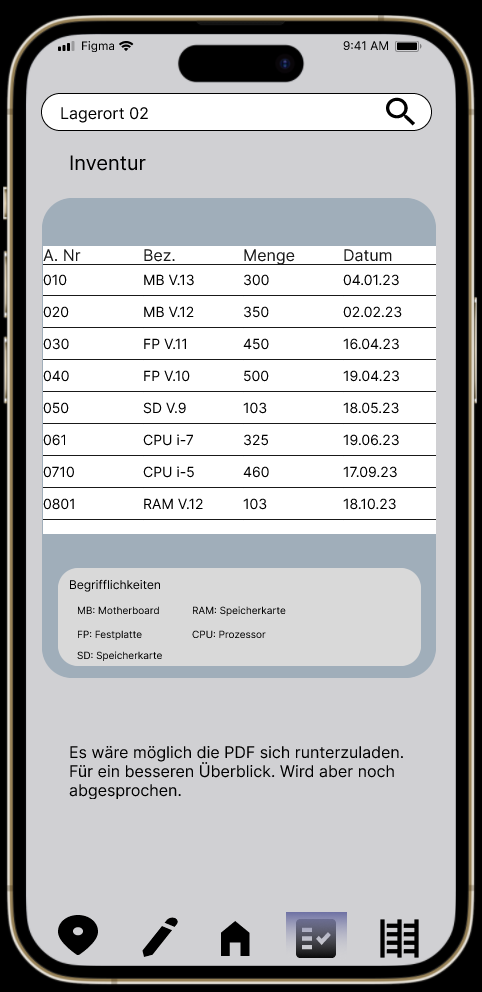
Mit Hilfe der Karte, kann der Nutzer sich orientieren an welcher Stelle die jeweiligen Gegenstände liegen. Hierfür gibt es ein Suchfeld, mit Hilfe dessen der zu suchende Gegenstand ausgewählt werden kann. Durch Hochziehen der Leiste am unteren Ende der Karte kann die nächste Seite, die des Lager-Edits, aufgerufen werden.

###### **Lager-Edit**



Nachdem der Nutzer weitergeleitet wurde, kann ein Gegenstand gewählt werden, um dieses zu bearbeiten. Zudem kann direkt ein neuer Gegenstand hinzugefügt werden, oder der bearbeitete Stand gespeichert werden. Wurde man vom Scanner aus weitergeleitet wird die Seite direkt mit dem Typ bzw. Name des Gegenstands gefüllt. Kommt der Nutzer von der Startseite wird zunächst, wenn in der Suchleiste nichts eingegeben wurde, eine Liste aller Gegenstände angezeigt.

###### **Inventur**



In der Inventur werden die Artikel Nummer, Bezeichnung, Menge und das (Bearbeitungs-)Datum angezeigt. Außerdem kann hier noch die Möglichkeit entstehen die angezeigten Daten lokal als PDF herunterzuladen.