

# **Erstellung einer Anwendung zur effizienten Verwaltung von mehreren Maschinen und Mitarbeitern**

## **Informatik 2 Projektbericht**

des Studiengangs Mechatronik

an der Dualen Hochschule Baden-Württemberg Heidenheim

von

**Marvin Bullinger, Julian Hagmann,  
Carla Vogel, Nikolas Munz**

11. März 2024

**Bearbeitungszeitraum**  
**Matrikelnummer, Kurs**  
**Bearbeitungsort**  
**Gutachter**

19.01.2024 – 11.03.2024  
5508863, 1109211, 1817945, 3507936, TMT2022  
DHBW Heidenheim, 89158 Heidenheim  
Dr. Lucas Bonifacius

# Erklärung

Wir versichern hiermit, dass wir unsere Projektarbeit mit dem Thema: *Erstellung einer Anwendung zur effizienten Verwaltung von mehreren Maschinen und Mitarbeitern* selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel benutzt haben. Wir versichern zudem, dass die eingereichte elektronische Fassung mit der gedruckten Fassung übereinstimmt.

Heidenheim an der Brenz, 11. März 2024

---

Marvin Bullinger

---

Julian Hagmann

---

Carla Vogel

---

Nikolas Munz

## **Projektbeschreibung**

Das Projekt **”MaschinenManagementSystem”** konzentriert sich auf die Entwicklung eines User Interfaces (**UIs**), die speziell für die Bedürfnisse in der Personal- und Maschinenplanung konzipiert ist. Ziel ist es, den Ausbildern und Führungskräften in diesen Bereichen die Verwaltungs- und Planungsaufgaben zu erleichtern und die allgemeine Produktivität innerhalb des Unternehmens zu steigern.

Die Anwendung bietet einen umfassenden Überblick über Schlüsselemente wie

- **Maschinenauslastung**
- **Mitarbeiterauslastung**
- **Kosten pro Auftrag**

Ein wesentlicher Aspekt des Projekts besteht in der pädagogischen Weiterentwicklung der Mitarbeiter, besonders in der Ausbildung. Es soll Auszubildenden ermöglicht werden, durch die Nutzung der Anwendung frühzeitig selbstständiges Arbeiten und verantwortungsvolles Handeln zu erlernen. Dies wird durch eine intuitive und benutzerfreundliche Oberfläche gefördert, die das Ändern und Speichern von Daten vereinfacht und den gesamten Planungsprozess effizienter gestaltet.

Dadurch wird in jenen Bereichen, in denen eine präzise Planung von Personal- und Maschinenressourcen benötigt wird, ein großer Mehrwert geleistet.

# Inhaltsverzeichnis

<b>Abkürzungsverzeichnis</b>	<b>IV</b>
<b>Abbildungsverzeichnis</b>	<b>V</b>
<b>1 Einführung</b>	<b>1</b>
1.1 Motivation des Projektes . . . . .	1
1.2 Ziele des Projektes . . . . .	2
<b>2 Stand der Technik</b>	<b>3</b>
<b>3 Planung</b>	<b>4</b>
3.1 Nicht triviale Frage . . . . .	4
3.2 GANTT-Diagramm . . . . .	5
3.3 Klassendiagramm . . . . .	7
3.4 ER-Diagramm . . . . .	8
3.5 MVVM-Pattern . . . . .	10
<b>4 Erstellung des Maschinen Management Systems</b>	<b>11</b>
4.1 Recherche der NuGet-Pakete . . . . .	11
4.2 Erstellung der Datenbanken . . . . .	11
4.3 Erstellung der Benutzeroberfläche . . . . .	13
4.4 ViewModelBase . . . . .	14
4.5 ViewModelCommand . . . . .	14
4.6 Probleme im Projektverlauf . . . . .	15
<b>5 Zusammenfassung und Ausblick</b>	<b>16</b>
<b>Literatur</b>	<b>X</b>
<b>Anhang</b>	<b>XI</b>

# Abkürzungsverzeichnis

<b>EF</b>	Entity Framework
<b>ER-Diagramm</b>	Entity-Relationship-Diagramm
<b>HTML</b>	Hypertext Markup Language
<b>LINQ</b>	Language Integrated Query
<b>MMS</b>	Maschinen Management System
<b>MVVM-Pattern</b>	Model-View-ViewModel-Pattern
<b>SQL</b>	Structured Query Language
<b>UI</b>	User Interface
<b>WPF</b>	Windows Presentation Foundation

# Abbildungsverzeichnis

3.1	Projekt Gantt-Diagramm . . . . .	5
3.2	Klassendiagramm . . . . .	7
3.3	ER-Diagramm . . . . .	9
3.4	MMVM-Pattern . . . . .	10
4.1	Benutzeroberfläche MMS . . . . .	13

# 1 Einführung

Im folgenden Kapitel wird der Nutzen des Projektes Maschinen Management System ([MMS](#)) beleuchtet. Zunächst wird die Motivation dargelegt, um ein Verständnis für die Ziele und Vorgehensweisen zu schaffen. Anschließend erläutern wir die spezifischen Methoden und Techniken, die zur Erreichung dieser Ziele eingesetzt werden. Abschließend werden die wesentlichen Meilensteine des Projekts hervorgehoben, um einen Überblick über den geplanten Projektverlauf zu geben.

## 1.1 Motivation des Projektes

Die Motivation hinter dem Projekt [MMS](#) liegt in einer zentralen Herausforderung, die wir während unserer Ausbildungszeit beobachtet haben: Ausbilder waren oft durch kleinere Organisationsaufgaben so sehr beansprucht, dass ihnen kaum Zeit blieb, sich intensiv um die Auszubildenden kümmern zu können. Diese Problematik führt zu einer Reihe von nachteiligen Folgen, sowohl für die Auszubildenden als auch für die Ausbilder selbst.

Erstens wird die Qualität der Ausbildung beeinträchtigt. Wenn Ausbilder mit organisatorischen Aufgaben überladen sind, können sie sich nicht ausreichend auf die Vermittlung von Fachwissen und praktischen Fähigkeiten konzentrieren. Dies kann zu einer suboptimalen Wissensvermittlung führen und die Entwicklung der Auszubildenden behindern.[\[Per23\]](#)

Zweitens entsteht ein Mangel an individueller Betreuung und Unterstützung. Auszubildende benötigen oft individuelle Anleitung und Feedback, um ihre Fähigkeiten effektiv zu entwickeln. Wenn Ausbilder durch Verwaltungsaufgaben abgelenkt sind, bleibt wenig Zeit für diese wichtige persönliche Interaktion.[\[Per23\]](#)

Drittens führt die Überlastung der Ausbilder zu einer erhöhten Arbeitsbelastung und Stress. Dies kann nicht nur die Arbeitszufriedenheit der Ausbilder beeinträchtigen, sondern auch zu Burnout führen und das Ausfallrisiko erhöhen, was wiederum die Stabilität und Kontinuität des Ausbildungsprogramms gefährdet.[\[Per23\]](#)

Um diese Probleme zu verbessern, entstand die Idee, eine spezialisierte Applikation zu entwickeln. Das Ziel dieser Anwendung ist es, die Ausbilder von zeitraubenden organisatorischen Aufgaben zu entlasten, damit sie sich auf ihre Kernkompetenzen konzentrieren können: die Ausbildung und Entwicklung der Auszubildenden.

## 1.2 Ziele des Projektes

Nachfolgend werden alle Ziele des Projektes dargelegt, die das Projekt in der Kernstruktur danach lösen wird bzw. vereinfachen und effizienter gestalten wird.

- **Automatisierte Planung:** Die Applikation ermöglicht es den Ausbildern, Arbeitsaufgaben effizient zu planen und zu verteilen. Durch die Automatisierung dieses Prozesses wird wertvolle Zeit eingespart, die sonst für manuelle Planung aufgewendet werden müsste.
- **Überblick über Maschinen:** Ein zentrales Feature der Applikation ist die Übersicht über alle Maschinen. Dies erlaubt den Ausbildern, den Status und die Verfügbarkeit der Ressourcen in Echtzeit zu überwachen und entsprechend die Aufträge zu koordinieren.
- **Direkte Kommunikation und Feedback:** Die Applikation ermöglicht eine direkte Kommunikationslinie zwischen Ausbildern und Auszubildenden. Dies fördert den schnellen Austausch von Feedback und Anweisungen, was für eine effektive Ausbildung unerlässlich ist.
- **Anpassbare Aufgabenverwaltung:** Jeder Auszubildende hat unterschiedliche Bedürfnisse und Fortschritte. Die Applikation ermöglicht es den Ausbildern, Aufgaben individuell anzupassen und so eine personalisierte Ausbildungserfahrung zu schaffen.
- **Benutzerfreundlichkeit und Zugänglichkeit:** Um eine hohe Akzeptanz zu gewährleisten, wird bei der Entwicklung der Applikation besonderer Wert auf Benutzerfreundlichkeit und intuitive Bedienung gelegt.

Zusätzliche Features die einfach implementiert werden könnten:

- **Analyse und Reporting:** Durch die Sammlung und Analyse von Daten über die Nutzung der Maschinen und die Leistung der Auszubildenden kann die Applikation wertvolle Einblicke für die stetige Verbesserung des Ausbildungsprogramms bieten.



## 2 Stand der Technik

Die fortschreitende Automatisierung und Umstellung auf Industrie 4.0 spielt eine entscheidende Rolle in vielen Produktionsprozessen moderner Firmen. Die Einrichtung von [MMS](#) in Werkstätten bildet dabei einen bedeutenden Schritt in Richtung intelligente Produktion zur Verwaltung der Ressourcen ab. Der Stand der Technik soll zunächst aufzeigen welcher Stand in einem Großteil der Firmen noch vorherrscht und warum dieser veraltet ist.[\[Wir23\]](#) Nun zur aktuellen Problemsituation. Viele Unternehmen stehen derzeit vor der Herausforderung, dass die Ausbilder zu viel mit organisatorischen Aufwand konfrontiert sind. Sowohl die Verwaltung der Maschinen als auch die Koordination vieler Auszubildenden machen einen Großteil des Arbeitstages aus. Diese Doppelbelastung hat die Folge, dass die Betreuer zu stark in administrative Aufgaben eingebunden sind, was zu längeren Wartezeiten eines einzelnen Auszubildenden führt. Anstatt die vielen organisatorischen Angelegenheiten zu klären, sollten die Ausbilder mehr Zeit für die Betreuung der Auszubildenden zur Verfügung haben.

Ein weiteres zentrales Problem, welches mit der manuellen Planung einhergeht, sind hohe Standzeiten von Maschinen. Die ineffiziente Nutzung von Maschinenkapazitäten hat direkte Auswirkungen auf die Wirtschaftlichkeit einer Firma. Durch die langen Stillstandszeiten wird das Anstreben einer maximalen Produktivität nicht ausgenutzt, was letztendlich zu finanziellen Einbußen führt. Da diese Problematik den realen Alltag vieler Ausbildungsbetriebe darstellt, soll in Zukunft ein [MMS](#) zum Einsatz kommen. Dieses System wird dann die Verwaltungsprozesse erleichtern. Das heißt, dass die Auszubildenden zukünftig in der Lage sein werden, schnell und direkt auf erforderliche Informationen zugreifen zu können. Außerdem werden Ausbilder mehr Zeit in die Betreuung der Jungfacharbeiter stecken können, was die Qualität der Ausbildung nachhaltig verbessern und gleichzeitig die allgemeine Produktivität des Betriebs erhöhen wird.

# 3 Planung

Die Planungsphase ist ein wichtiger Schritt zur Entwicklung des [MMS](#). Um die Ziele rechtzeitig zu erreichen wird zunächst ein Gantt-Diagramm angefertigt, um den groben Projektverlauf inklusive Pufferzeiten zu planen. Anschließend folgt die Erstellung des [ER-Diagramms](#) und Klassendiagramms, um die Struktur und Beziehungen der Datenbanktabellen zu visualisieren. Als letztes folgt das [MVVM-Pattern](#) welches die Trennung von Benutzeroberfläche und Anwendungslogik ermöglicht.

## 3.1 Nicht triviale Frage

Die grundlegende Idee unseres Projektes und damit auch unsere nicht triviale Frage ist es, ein Programm zu entwickeln, welches es ermöglicht, einem Ausbilder organisatorische Arbeit abzunehmen und mehr Zeit und Qualität in das Ausbilden der Auszubildenden zur Verfügung steht. Genauer beschrieben sollen in einer Ausbildungswerkstatt die Maschinen nicht mehr manuell verteilt werden, um dem Ausbilder organisatorischen Aufwand wegzunehmen. Die Maschinen- und Auftragsverteilung soll automatisiert über ein [MMS](#) funktionieren, sodass Standzeiten für Maschinen kleiner ausfallen und die mündliche oder per Papier protokollierte Einteilung digitalisiert wird. Dadurch wird der ganze Prozess effizienter und flexibler gestaltet, da die Aufträge leicht vom Ausbilder gelöscht werden können und die Auszubildenden ihre Aufgabe digital und somit einfach zu erreichen einsehen können. Es soll eine dazugehörige Benutzeroberfläche mittels Windows Presentation Foundation ([WPF](#)) programmiert werden, welche eine intuitive Bedienung ohne Vorkenntnisse bereitstellt. Im [UI](#) werden dann alle für den Benutzer relevanten Funktionen sichtbar sein und die Programmierung soll möglichst optimal gestaltet werden.[\[Hub19\]](#)

### 3.2 GANTT-Diagramm

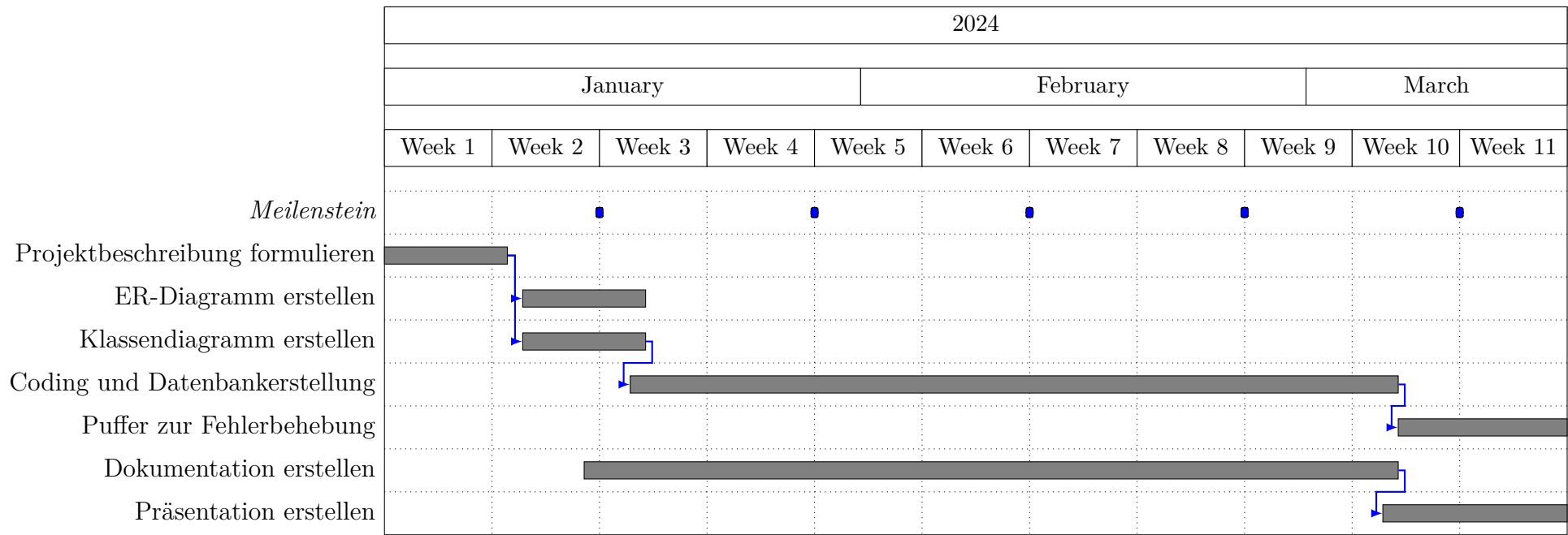


Abbildung 3.1: Projekt Gantt-Diagramm

Das Gantt-Diagramm stellt den Projektfortschritt auf grafische Weise dar und dient dazu, einen konkreten Überblick über die Aufgaben und Aktivitäten des Projekts zu erhalten. Es zeigt sowohl den Start- als auch den Endzeitpunkt jedes Projektabschnitts und nach jeder geraden Kalenderwoche erfolgen Meilensteine, welche zusätzlich zur Vorlesung stattfinden. In den Meilensteinen aktualisieren sich alle Gruppenmitglieder und der bisherige und zukünftige Projektverlauf wird besprochen. Außerdem werden neue Aufgaben zugeteilt, besprochene Aufgaben nachbearbeitet und Fragen geklärt.

Zu Beginn des Projektes wird zunächst die Projektbeschreibung ausformuliert und damit auch die Ziele der Projektarbeit definiert. Anschließend erfolgt die Ausarbeitung des [ER-Diagramms](#) und Klassendiagramms, was als Grundlage zur Übersicht über Beziehungen und Struktur des Projektes dient. Nach bzw. während der Fertigstellung der Diagramme wird mit der Dokumentation und dem Schreiben des Programmcodes begonnen. Als Plattform für den Austausch untereinander wird GitHub verwendet und als Entwicklungsumgebung die neuste Version von Microsoft Visual Studio Community. Bei der Erstellung wird besonders darauf geachtet, dass genügend Zeit für die Programmierung eingeplant wird sowie auch ein ausreichender Puffer zur Fehlerbehebung und Korrektur des Codes. Die Erstellung der Präsentation folgt nach Abschluss der Dokumentation.



richtigen Schlüssel für die Abfragen einer Datenbank verwendet und die Aktualisierungen der Datenbank gemacht werden können.

Einige weitere Dateien, welche Teil des namespaces `MMSLib.Model` sind, werden nachfolgend erwähnt. Die Auflistung gibt eine kurze Übersicht über diese Dateien und deren Eigenschaften, welche dazu dienen die Entitäten im `MMS` zu speichern und zu modellieren.[Kun18]

- **AufgabenZuweisen.cs:** Die Fremdschlüssel `AuftragsID`, `FacharbeiterID` und `VorgesetzterID` verweisen auf die jeweilig benötigte ID und die Eigenschaft `Eintrag` dient als Primärschlüssel und wird automatisch inkrementiert.
- **Aufträge.cs:** Beinhaltet den Fremdschlüssel `MaschinenID` und die Spalten `AuftragsID`, `Beschreibung`, `Material`, `Abgabe` und `Dauer`
- **Facharbeiter.cs:** Enthält die Spalten `FacharbeiterID`, `FacharbeiterName`
- **Lehrgang.cs:** Inkludiert das Auto-Inkrement `LehrgangID` und die Spalte `LehrgangName`
- **LehrgangZuweisung.cs:** Umfasst das Auto-Inkrement und zugleich Primary-Key `EintragLehrgang`, den Fremdschlüssel `FacharbeiterID` und die Spalte `LehrgangID`
- **Maschine.cs:** Beinhaltet den Fremdschlüssel `Auftragsnummer` und die Spalten `MaschinenID`, `MaschinenName`, `Material`, `MaschinenStatus` und `MaschinenStunde`
- **Vorgesetzter.cs:** Enthält die Spalten `VorgesetzterID`, `VorgesetzterName` und `VorgesetzterVorname`

## 3.4 ER-Diagramm

Das Entity-Relationship-Diagramm (**ER-Diagramm**) ist ein wichtiger Bestandteil des `MMS`. Es visualisiert alle Beziehungen zwischen den verschiedenen Entitäten in der Datenbank des Systems. Durch das Diagramm ist es uns möglich die Struktur und Beziehungen zwischen den Daten einfach darzustellen, was eine effiziente Verwaltung und Planung für das Gesamtprojekt ermöglicht.[Bon23]

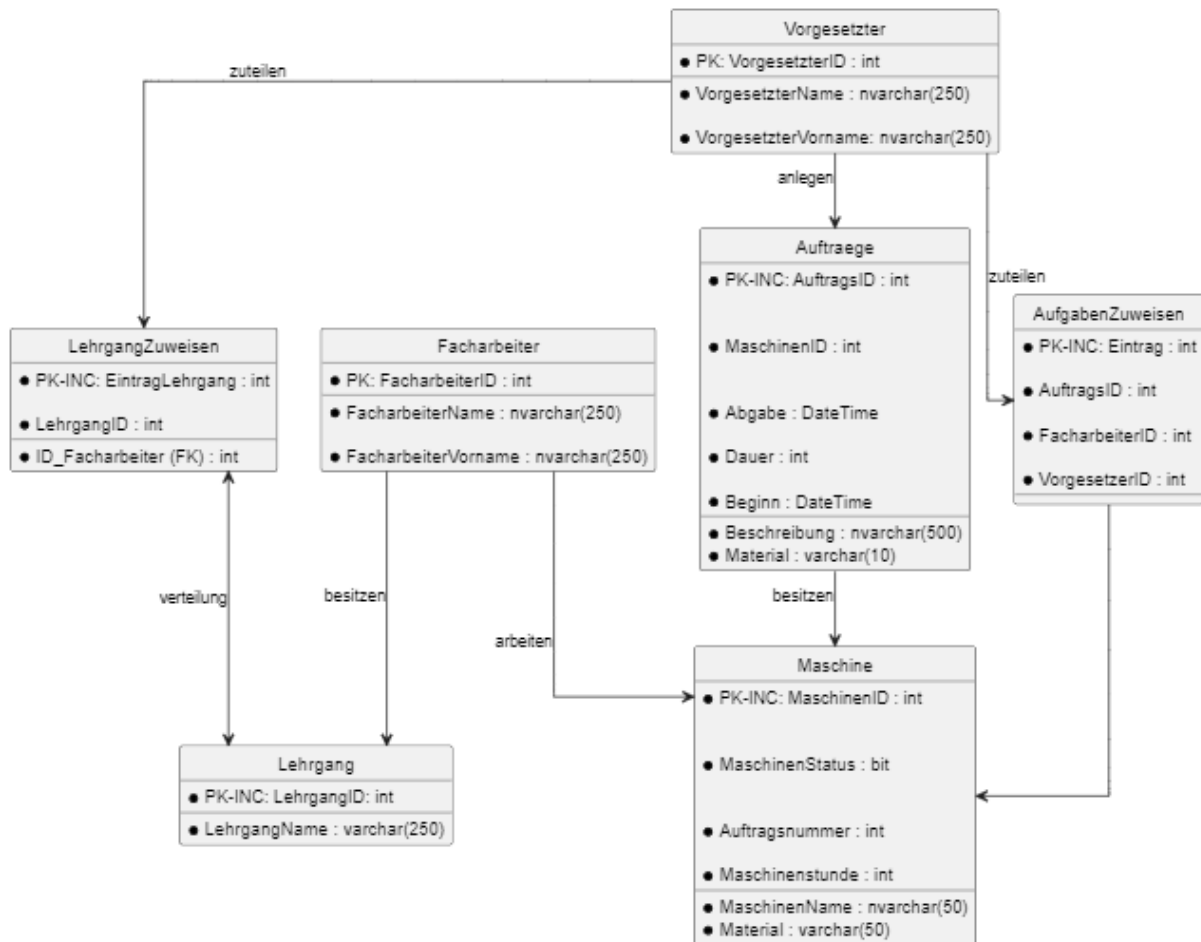


Abbildung 3.3: ER-Diagramm

Das [ER-Diagramm](#) zeigt die folgenden Entitäten:

- **Facharbeiter:** Enthält Vorname, Nachname sowie FacharbeiterID
- **Vorgesetzter:** Speichert Vorname, Nachname sowie VorgesetzterID
- **LehrgangZuweisung:** Verknüpft die FacharbeiterID mit der LehrgangID, welche sie besuchen oder schon besucht haben. EintragLehrgang ist der Primary-Key
- **Lehrgang:** Besteht aus Informationen über die verschiedenen möglichen Lehrgänge, welche durch LehrgangID und LehrgangName verknüpft sind.
- **Auftrage:** Enthält die folgenden Details zu den folgenden Aufträgen: AuftragsID, Beschreibung, MaschinenID, Material, Abgabe (Datum) und Dauer
- **AufgabenZuweisung:** Enthält AuftragsID, Facharbeiter, VorgesetzterID und Eintrag als Primary-Key

- **Maschine:** Dort werden alle notwendigen Informationen zu den Maschinen gespeichert wie: die `MaschinenID`, `MaschinenName`, `Material`, `MaschinenStatus` und `MaschinenStunden`

Die Verbindungen zwischen den Entitäten sind die Interaktionen, welche durch die Beziehungen 'anlegen', 'besitzen' und 'zuteilen' dargestellt werden.

## 3.5 MVVM-Pattern

Das [MVVM-Pattern](#) soll dem Projekt eine Struktur geben und als Architektur für Entwurfsmuster dienen. Dieses ermöglicht dann eine klare Trennung zwischen Datenmodell (Model), Benutzeroberfläche (View) und Anwendungslogik (ViewModel). Bei dem Vorgehen, die Anwendung mit diesem Muster zu gestalten, werden zahlreiche Entwicklungsprobleme verhindert. Außerdem erleichtert es das Testen und das Verwalten der Anwendung. Die folgende Abbildung veranschaulicht die Beziehung zwischen den drei oben genannten Komponenten.[\[Kun18\]](#)

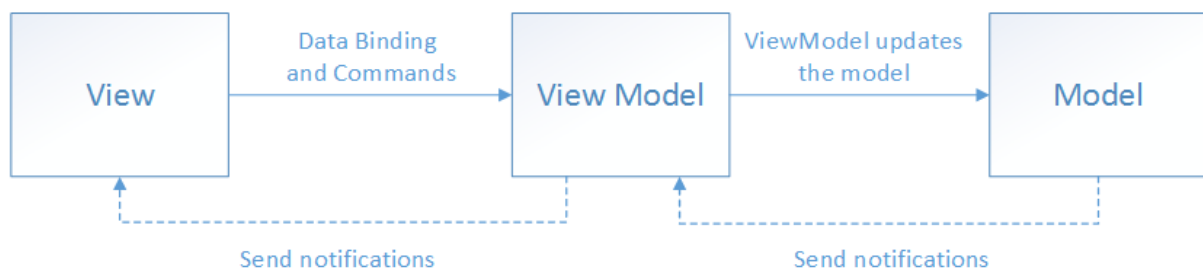


Abbildung 3.4: MMVM-Pattern

In unserem Projekt haben wir das Model-View-ViewModel-Pattern ([MVVM-Pattern](#)) folgendermaßen angewendet:

- **View:** Dort liegen die Dateien für die einzelnen Benutzeroberflächenansichten, welche mit XAML entwickelt wurden. Jedes Fenster wurde individuell gestaltet.
- **ViewModel:** Die komplette Logik für die Darstellung der Daten unserer Anwendung sowie alle zugehörigen Childviews sind hinterlegt.
- **Model:** In diesen Dateien sind sämtliche Informationen zu unseren Datenbanken hinterlegt, einschließlich der Erstellung neuer Datenbanken, Verknüpfungen und Suchfunktionen.



# 4 Erstellung des Maschinen Management Systems

## 4.1 Recherche der NuGet-Pakete

NuGet-Pakete sind Codebibliotheken für .NET-Anwendungen, die Wiederverwendbarkeit und einfaches Management ermöglichen. Sie bieten Zugriff auf ein zentrales Repository, unterstützen Versionskontrolle und erleichtern die Entwicklung durch Verringerung der Notwendigkeit, Code neu zu schreiben.[[Mic22b](#)]

In diesem Projekt wurde besonderer Wert auf die Gestaltung einer modernen, attraktiven und intuitiven Benutzeroberfläche gelegt. Zur Umsetzung dieser Anforderungen wurde das **FontAwesome.Sharp**-Paket ausgewählt. Die Entscheidung basierte auf der bereits vorhandenen Erfahrung des Teams mit diesem Paket sowie der Attraktivität der enthaltenen Icons, Komponenten und Anwendungen. In unserem Projekt wurde überwiegend die Implementation der Icons verwendet, um ein besseres User-Experience-Design zu gewährleisten.[[Mic23c](#)]

Das NuGet-Paket **EF** wurde eingebunden, um die Datenbankkompatibilität sicherzustellen. Das Paket bietet eine abstrakte Datenbankzugriffsebene, objektorientierte Interaktion, flexible Modellierung und unterstützt verschiedene Datenbankanbieter, was die Entwicklung vereinfacht.

## 4.2 Erstellung der Datenbanken

Zur Realisierung des **MVVM-Pattern** werden zwei verschiedene Projekte verwendet, um eine Trennung zwischen Datenvisualisierung und den Datenverarbeitung umsetzen zu können. Hierbei findet die gesamte Datenverarbeitung im Projekt **MMSLib** statt.[[Kun18](#)]  
Das Bearbeiten der Datenbank geschieht mithilfe der im **EF** enthaltenen Klasse **DbContext**. Diese kann zu bereits bestehenden Tabellen Werte hinzufügen, bearbeiten oder gar löschen. In der Klasse **DbConnect**, welche von **DbContext** erbt, findet über die **DbSet**-Eigenschaft die Berechtigung statt, dass **DbConnect** auf die lokale Datenbank **MMS-Database** zugreifen kann. In dieser Klasse werden **get** und **set** Eigenschaften für alle Tabellen definiert, was den Zugriff auf diese Tabellen erlaubt.

Datenbankabfragen erfolgen über Language Integrated Query (**LINQ**), welches eine integrierte Abfragesprache ist und im Gegensatz zu Structured Query Language (**SQL**) eine sehr intuitive und kompakte Möglichkeit zur Datenbankabfrage bietet. Dank seiner Integration in **NET**-Programmiersprachen wird der Entwicklungsprozess vereinfacht[Mic22a], da Abfragen nahtlos in derselben Sprache wie der restliche Code verfasst werden können. Dies reduziert den Programmieraufwand für Entwickler im Vergleich zu herkömmlichen **SQL**-Abfragen erheblich. Im Ordner **MMS/MMSLib/Klassen** sind zwei Klassen mit Datenbank-Interaktion definiert.

In der Klasse **AuftragAnzeigen.cs** wird anhand des Eingabeparameter **FacharbeiterID** eine Datenbankabfrage gestartet, welche alle aktuell verfügbaren Aufträge des aufgerufenen Facharbeiters anzeigt.

Die Klasse **AuftragLoeschen.cs** ermöglicht das Löschen von versehentlich angelegten Aufträgen. Dadurch wird die Arbeitseffizienz verbessert und verhindert, dass eine Maschine für einen bestimmten Zeitraum blockiert wird. Dies ermöglicht wiederum eine indirekte Bearbeitung oder Neuanlage der Aufträge.

## 4.3 Erstellung der Benutzeroberfläche

Zunächst die Frage im Raum, ob es [WPF](#) oder Hypertext Markup Language ([HTML](#)) für das [UI](#) des [MMS](#) wird. Die Entscheidung, [WPF](#) zu verwenden, beruht auf verschiedenen Faktoren und Überlegungen. Zum einen ist das Projekt darauf ausgerichtet, ein Problem für große Industriefirmen zu lösen. Daher erscheint eine professionelle und lokal installierbare Desktopanwendung als die passendere Wahl im Vergleich zu einer [HTML](#)-Seite.[[Hub19](#)] Darüber hinaus hat [WPF](#) eine langjährige Verwendung in verschiedenen Bereichen erfahren und wird auch weiterhin aktiv eingesetzt.

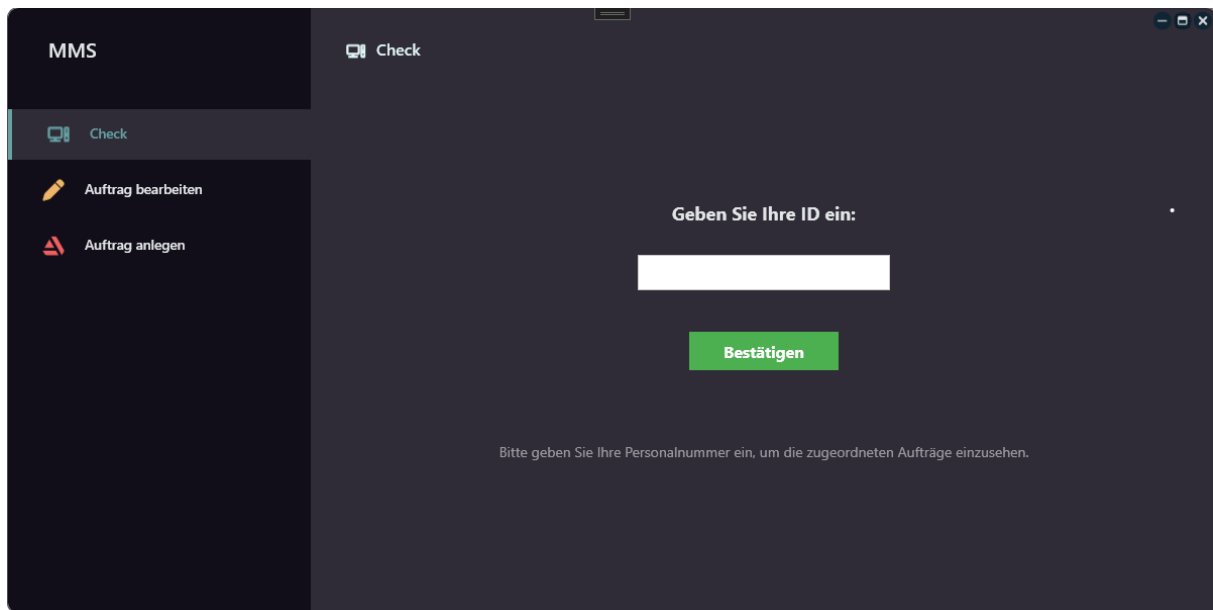


Abbildung 4.1: Benutzeroberfläche MMS

Ebenso vorteilhaft von [WPF](#) ist, dass dies Teil des [.NET](#)-Frameworks ist und es deshalb von der Sicherheit dieses Frameworks profitiert.[[Mic23a](#)] Darüber hinaus können [WPF](#)-Anwendungen leicht angepasst und erweitert werden, um spezifischen Anforderungen zu entsprechen.

Zusätzlich gibt es für [WPF](#) eine Vielzahl von Design-Paketen, wie beispielsweise [FontAwesome.Sharp](#), die Entwicklern umfangreiche Möglichkeiten bieten, die Benutzeroberfläche ihrer Anwendungen mit hochwertigen Icons und Grafiken mit geringen Programmieraufwand zu bereichern.

## 4.4 ViewModelBase

Die Klasse `ViewModelBase` im [MVVM-Pattern](#) ist essentiell für die Realisierung der Datenbindung in [WPF](#)-Anwendungen. Sie implementiert das `INotifyPropertyChanged-Interface`, wodurch Änderungen an Eigenschaften automatisch an die [UI](#) kommuniziert werden, was eine reaktive Aktualisierung der [UI](#) ermöglicht.[\[Mic23b\]](#) Durch die Bereitstellung einer standardisierten Methode für Benachrichtigungen unterstützt `ViewModelBase` die Trennung zwischen Anwendungslogik und Darstellungsschicht, fördert die Code-Wiederverwendung und erleichtert die Erstellung wartbarer und testbarer Anwendungen. Diese Klasse sorgt für eine bessere Modularität und somit die schnelle Wiederverwendung einzelner Bausteine.

## 4.5 ViewModelCommand

Die Klasse `ViewModelCommand` ist in [WPF](#)-Anwendungen innerhalb des [MVVM-Pattern](#) essentiell, da sie die Verbindung zwischen [UI](#)-Elementen und Aktionen oder Kommandos in der Anwendungslogik herstellt. Sie ermöglicht es Entwicklern, [UI](#)-Ereignisse, wie Klicks oder Auswahländerungen, direkt an Methoden oder Befehle im `ViewModel` zu binden, wodurch die Trennung von [UI](#) und Geschäftslogik gefördert wird.[\[Mic23b\]](#) Durch die Implementierung des `ICommand-Interfaces` unterstützt `ViewModelCommand` die Ausführungslogik und die Verfügbarkeitsprüfung von Aktionen, was eine bedingte Aktivierung oder Deaktivierung von [UI](#)-Elementen basierend auf dem Anwendungsstatus ermöglicht. Diese Klasse trägt somit zur Steigerung der Modularität, Wartbarkeit und Testbarkeit von [WPF](#)-Anwendungen bei.

## 4.6 Probleme im Projektverlauf

Im Verlauf des Projekts wurde das [MVVM-Pattern](#) eingeführt. Dies bedingte erhebliche Umstrukturierungen und zusätzlichen Arbeitsaufwand, da ein beträchtlicher Teil des Codes bereits implementiert war. Anschließend wurden die Datenbanken aufgrund sich ändernder Anforderungen der Codeentwicklung angepasst, was zusätzliche Komplexität mit sich brachte. Dies war dem geschuldet, dass keine einheitliche cloudbasierte Datenbank zur Verfügung stand und jeder Entwickler dazu gezwungen war eine lokale Datenbank zu verwenden.

In der Mitte des Projektes war eine umfassende Umstellung der Variablennamen nötig, da die alten nicht den Richtlinien entsprach. Dieser Schritt wurde vorgenommen, nachdem bereits ein Teil des Programms entwickelt worden war. Leider wurden dabei einige Variablennamen, insbesondere in den lokalen Datenbanken der Entwickler, nicht aktualisiert, was zu vorübergehenden Fehlern beim Ausführen des Programms führte.

Schließlich erschwerte die fehlerhafte Rechtschreibung in den Tabellen einer lokalen Datenbanken die Verbindung vom Programm zur Datenbank. Es kostete erhebliche Zeit, den Fehler in der Datenbank statt im Code zu finden.

## 5 Zusammenfassung und Ausblick

Das [MMS](#) bietet eine effiziente Lösung in Sachen Personal- Maschinenplanung und -verwaltung. Durch eine zu Beginn erstellte klare Strukturierung mittels [ER-Diagramm](#), Klassendiagramm und Projektplanung war der Projektablauf klar strukturiert. So konnten die Aufgaben direkt verteilt und die Datenbanken erstellt werden. Durch die Integration des [EF-Core](#) ist es gelungen eine robuste Datenbankverwaltung aufzubauen und die Aufgabenstellung zu erfüllen. Trotz einiger programmtechnischer Probleme konnte der Zeitplan eingehalten und die Hürden zusammen überwunden werden. Besonders herausfordernd war dabei die Programmierung der Maschinenbelegung. Die Aufgaben, dass Aufträge automatisch gelöscht werden oder Maschinen nicht doppelt belegt werden können gehörten definitiv zu den schwersten im Projekt und haben den Zeitpuffer in Anspruch genommen. Letztendlich ist es jedoch gelungen diese beiden Ideen programmtechnisch umzusetzen.

Die Verwendung dieses Programms soll in Zukunft maßgeblich zur Verbesserung der Abläufe in Werkstätten von Unternehmen beitragen. Um Unternehmen eine attraktive Software vorzustellen soll das [UI](#) nochmals überarbeitet werden und der Reiter Auftrag bearbeiten soll erweiterte Funktionen erhalten. Darunter fallen dann: spezifischere Eingabe der Aufträge, genauere Zeitangabe der Aufträge und das speichern von abgelaufenen oder beendeten Aufträgen zur besseren Nachvollziehbarkeit.

# Literatur

- [Bon23] Dr. Lucas Bonfacius. Objekt-orientierte Programmierung Vorlesung. 2023.
- [Hub19] Thomas Huber. Windows Presentation Foundation. Rheinwerk Computing, 2019.
- [Kun18] Stefan Kunik. Software entwickeln mit C, WPF und dem MVVM-Konzept. Ratgeber Verlag, 2018.
- [Mic22a] Microsoft. Language Integrated Query (LINQ) provider for C. 2022. URL: <https://learn.microsoft.com/en-us/dynamics365/fin-ops-core/dev-itpro/dev-tools/linq-provider-c>.
- [Mic22b] Microsoft. What is a Nugget? 2022. URL: <https://learn.microsoft.com/en-us/nuget/what-is-nuget>.
- [Mic23a] Microsoft. Erste Schritte mit .NET Framework. 2023. URL: <https://learn.microsoft.com/de-de/dotnet/framework/get-started/>.
- [Mic23b] Microsoft. Erste Schritte mit ViewModelBase. 2023. URL: [https://learn.microsoft.com/en-us/previous-versions/hh721872\(v=vs.120\)](https://learn.microsoft.com/en-us/previous-versions/hh721872(v=vs.120)).
- [Mic23c] Microsoft. Icons von Font Awesome. 2023. URL: <https://www.victoriaweber.de/blog/font-awesome>.
- [Per23] Personalpraxis24. Vor diesen Irrtümern sollten sich Ausbilder schützen. 2023. URL: <https://www.personalpraxis24.de/top-thema/berufsausbildung-6-vor-diesen-irrtuemern-sollten-sich-ausbilder-schuetzen-252/>.
- [Rus10] Marco Russo. Programming Microsoft LINQ in Microsoft .NET Framework 4. Ratgeber Verlag, 2010.
- [Wir23] Ministerium für Wirtschaft. Neue innovative Ausbildungsprojekte prämiert. 2023. URL: <https://wm.baden-wuerttemberg.de/de/service/presse-und-oeffentlichkeitsarbeit/pressemitteilung/pid/industrie-40-talente-neue-innovative-ausbildungsprojekte-praemiert>.

# Anhang

- Klassendiagramm
- ER-Diagramm
- gesamter Code von VisualStudio