Projektdokumentation des IHK-Abschlussprojektes Sommer 2021

**vFleet Management Dashboard**

Entwicklung eines Dashboard-Moduls für die Flottenmanagement-Software der vertical GmbH

Fachinformatiker Fachrichtung Anwendungsentwicklung

Abgabedatum: 03.05.2021

**Autor:**

Jaime Gamero Maya

Hainer Weg 148

60599 Frankfurt am Main

**Ausbildungsbetrieb:**

vertical GmbH

Otto-Volger-Str. 3 – 5a/b

65843 Sulzbach Ts.



**Inhaltsverzeichnis**

[1. Thematisierung des Projektauftrags 1](#_Toc70940210)

[1.1 Einleitung 1](#_Toc70940211)

[1.2 Projektbeschreibung 1](#_Toc70940212)

[1.3 Projektziel 2](#_Toc70940213)

[1.4 Projektbegründung 2](#_Toc70940214)

[1.5 Projektabgrenzung 3](#_Toc70940215)

[2. Ressourcen- und Ablaufplanung 4](#_Toc70940216)

[2.1 Ablaufplan 4](#_Toc70940217)

[2.2 Entwicklungsprozess 4](#_Toc70940218)

[3. Projektvorbereitung 5](#_Toc70940219)

[3.1 Ist-Analyse 5](#_Toc70940220)

[3.2 Anwendungsfälle 6](#_Toc70940221)

[3.3 Wirtschaftliche Betrachtung 6](#_Toc70940222)

[3.3.1 Projektkosten 6](#_Toc70940223)

[3.3.2 Amortisationsdauer 7](#_Toc70940224)

[3.3.3 Nicht-monetäre Vorteile 8](#_Toc70940225)

[3.4 Entwicklungsplattform 8](#_Toc70940226)

[3.5 Architekturdesign 9](#_Toc70940227)

[3.6 Entwurf der Datenstruktur 10](#_Toc70940228)

[3.7 Entwurf der Benutzeroberfläche 10](#_Toc70940229)

[4. Implementierung 11](#_Toc70940230)

[4.1 Implementierung der Datenstruktur 11](#_Toc70940231)

[4.2 Implementierung der Benutzeroberfläche 11](#_Toc70940232)

[4.3 Implementierung Geschäftslogik 11](#_Toc70940233)

[5. Testphase 12](#_Toc70940234)

[6. Projektabschluss 12](#_Toc70940235)

[6.1 Deployment 12](#_Toc70940236)

[6.2 Abnahme 13](#_Toc70940237)

[7. Projektergebnis 13](#_Toc70940238)

[7.1 Soll-Ist-Vergleich 13](#_Toc70940239)

[7.2 Fazit 14](#_Toc70940240)

[7.3 Ausblick 14](#_Toc70940241)

[Anlage a](#_Toc70940242)

[a. Glossar a](#_Toc70940243)

[b. Quellenverzeichnis a](#_Toc70940244)

[c. Verwendete Software b](#_Toc70940245)

[d. Eidesstattliche Erklärung b](#_Toc70940246)

[Anhang i](#_Toc70940247)

[A.1 Screenshot einer Buchung im Altsystem i](#_Toc70940248)

[A.2 Use-Case Diagramm i](#_Toc70940249)

[A.3 GUI Prototyp ii](#_Toc70940250)

[A.4 Screenshots der Anwendung iv](#_Toc70940251)

# Thematisierung des Projektauftrags

## Einleitung

In der folgenden Dokumentation wird das Abschlussprojekt des Autors beschrieben, welches im Rahmen der Ausbildung zum Fachinformatiker Fachrichtung Anwendungsentwicklung durchgeführt wurde.

Das Ausbildungsunternehmen ist die vertical GmbH mit Sitz in Sulzbach (Taunus), die etwa 90 Mitarbeiter hat. Die Haupttätigkeit des Unternehmens ist die Bereitstellung des digitalen Arbeitsplatzes und dessen Verwaltung in einem „IT as a service“ Modell. Dieses Modell entlastet die IT-Abteilungen der Kunden und hilft bei der Digitalisierung von Geschäftsprozessen.

## Projektbeschreibung

Die Mitarbeiter der vertical GmbH nutzen im Rahmen ihrer Tätigkeit Firmenfahrzeuge für den Support der Kunden. Dafür stellt die Firma eine Fahrzeugflotte zur Verfügung. Derzeit gibt es keinen Dienst, der die einfache Reservierung für die Mitarbeiter ermöglicht und dem Backoffice Informationen zu Buchungen und Fahrzeugen bereitstellt. Aktuell werden die Fahrzeuge über einen Outlook-Kalender gebucht und das Backoffice verwaltet die Fahrzeuge und Buchungen in Papierform.

Zur Vereinfachung der Reservierung und verbesserten Verwaltung wird eine Webkomponente erstellt, welche diese Anforderungen erfüllt. Die Anwendung ist aus verschiedenen Modulen zusammengesetzt. Jedes Modul ist für eine bestimmte Funktionalität zuständig. Dieses Projekt umfasst das Dashboard-Modul, welches dem Benutzer einen Einstieg in die Anwendung gibt und eine erste Übersicht gewährt.

Es gibt zwei unterschiedliche Benutzeroberflächen, je nachdem mit welcher Benutzerrolle auf die Anwendung zugegriffen wird. Die Benutzer werden unterschieden in Backoffice (Mitarbeiterrolle) und User (Benutzerrolle). Die Applikation ist dafür verantwortlich die Rolle jedes Mitarbeiters zu erkennen und die entsprechende Oberfläche anzuzeigen.

Die Oberfläche des Backoffice zeigt an erster Stelle den aktuellen Tag mit allen Reservierungen an. Zu jeder Buchung gehören detaillierte Informationen (Kennzeichen, Buchungszeitraum, Mitarbeiter, Fahrtziel). Zusätzlich wird die Historie der Reparaturen des Fuhrparks mit allen relevanten Informationen angezeigt (Kennzeichen, Status, Beschreibung, gemeldet von, gemeldet am).

Wenn hingegen der Zugriff auf die Anwendung durch den User erfolgt, zeigt das Dashboard-Modul die wichtigsten Informationen für die an diesem Tag geplanten Fahrten an (Fahrtbeginn, Fahrzeugrückgabe, Fahrtziel, Kennzeichen, Fahrzeugtyp). Zusätzlich wird eine Liste der für die nächsten sieben Tage geplanten Fahrten angezeigt, die die wichtigsten Buchungsinformationen enthält (Datum, Fahrbeginn, Fahrtziel).

Diesen beiden Versionen des Dashboard-Moduls erfüllen die zu Beginn des Projekts festgelegten Anforderungen und entsprechen somit den Bedürfnissen der beiden Rollen, die die Anwendung nutzen.

## Projektziel

Ziel des Projekts ist die Entwicklung eines Dashboard-Modul, das in die globale Anwendung vFleet Management integriert wird. Dadurch wird die Verwaltung des Fuhrparks sowie die Buchung der Fahrzeuge vereinfacht. Dieses entwickelte Dashboard-Modul zielt darauf ab, die Buchungen von Benutzern anzuzeigen sowie Informationen zu Reparaturen an Fahrzeugen im Fuhrpark bereitzustellen. Durch die digitale Bereitstellung der Informationen wird die Verwaltung und Nutzung für alle Mitarbeiter vereinfacht und besser zugänglich. Die Anwendung wird mit allen aktuellen gängigen Browsern kompatibel sein.

## Projektbegründung

Das derzeitige Verfahren zur Buchung eines Fahrzeugs bietet keine Informationen über die Reservierungen für den Mitarbeiter. Darüber hinaus fehlt eine Übersicht zu Reservierungen und Reparaturstatus des Fahrzeugs für die Backoffice-Abteilung (s. Kapitel 3.1 Ist-Analyse).

Mithilfe des Dashboard-Moduls erhalten die unterschiedlichen Benutzer alle relevanten Informationen über ein personalisiertes Dashboard. Der Mitarbeiter, der die Reservierung vornimmt, erhält somit alle notwendigen Informationen auf seinem Dashboard. Beim Zugriff durch das Backoffice bietet das Dashboard Daten zur Analyse sowie einen Überblick über den aktuellen Status der Flotte und der Reservierungen.

## Projektabgrenzung

Das Dashboard-Modul ist Teil einer größeren Anwendung. Die Applikation wird von mehreren Mitarbeitern der vertical GmbH entwickelt. Die Anwendung ist getrennt in folgende Bereiche:

* Administration & Management (Fahrzeugverwaltung)
* Service & Booking (Fahrzeuge buchen)
* **Controlling & Monitoring (Dashboard)**

Jedes Modul ist ein in sich geschlossenes Projekt, welches über dedizierte Schnittstellen mit den anderen Modulen kommuniziert. Jedes Modul besteht aus den folgenden technischen Komponenten:

* **UI (Angular Komponente(n))**
* API (REST Web-API)
* Repository (Datenbank)

Der Umfang dieser Projektarbeit ist die Entwicklung des Dashboard-Moduls. Die Implementierung der Module für Administration und Booking, der Backend-Logik (API) sowie das Datenbankdesign ist nicht Teil dieser Projektarbeit.

Graphical user interface

Description automatically generated

*Diagramm 1: Projektabgrenzung*

# Ressourcen- und Ablaufplanung

## Ablaufplan

In der folgenden Tabelle sind die verschiedenen Phasen der Projektausführung aufgeführt. Jede Phase ist mit einer Stundenzahl für die Bearbeitung hinterlegt. Insgesamt wird das Ziel von 70 Stunden für die Fertigstellung des Projekts erreicht.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Projektvorbereitung** | | | | **25 h** | | | |
| 1. | Ist-Analyse (inkl. Fachgespräche) |  | 4 h |  |
| 2. | Design Anwendungsfälle |  | 2 h |  |
| 3. | Kalkulation der Wirtschaftlichkeit |  | 2 h |  |
| 4. | Auswahl der Entwicklungsplattform |  | 3 h |  |
| 5. | Entwerfen des Architekturdesigns |  | 4 h |  |
| 6. | Modellierung der Datenstruktur |  | 5 h |  |
| 7. | Entwerfen der Benutzeroberfläche |  | 5 h |  |
| **Implementierung** | | | | **24 h** | | |
| 7. | Implementieren der Datenstruktur |  | 6 h |  | |
| 8. | Implementieren der Benutzeroberfläche |  | 9 h |  | |
| 9. | Implementierung der Geschäftslogik |  | 9 h |  | |
| **Testphase** | | | | **7 h** | | |
| 10. | Durchführen von Tests |  | 7 h |  | | |
| **Projektabschluss** | | |  | **14 h** | | |
| 11. | Deployment der Anwendung |  | 1 h |  | | |
| 12. | Abnahme der Anwendung |  | 1 h |  | | |
| 13. | Erstellen der Projektdokumentation |  | 12 h |  | | |
| Gesamt  *Tabelle 1: Zeitplanung* | | | | 70 h | | |

## Entwicklungsprozess

Für die Entwicklung des Abschlussprojekts wird eine agile Methodik gewählt. Ein Vorteil des agilen Ansatzes sind die regelmäßigen Rücksprachen mit den Stakeholdern, in diesem Fall dem Backoffice sowie den Mitarbeiter, die Fahrzeuge buchen. Dadurch kann zeitnah auf weitere Anforderungen reagiert werden. Wenn zusätzliche Anforderungen gestellt werden, wird geprüft, ob diese in das aktuelle Projekt miteinbezogen oder als zukünftige Funktion vorgemerkt werden.

Ein weiterer Vorteil ist die direkte Rückmeldung der Nutzer im Entwicklungsprozess. Dadurch können mögliche Anwendungsprobleme frühzeitig erkannt und behoben werden. Durch die Bereitstellung des Prototyps können sich die Nutzer bereits mit der Applikation vertraut machen. Dies führt dazu, dass sowohl die Schulung als auch die Abnahme beschleunigt werden kann. Ein erster Testlauf wird in dieser Phase bereits durchgeführt.

Diese Methodik bietet demnach eine dynamische Sicht auf den Entwicklungsprozess. Durch die aktive Einbindung des Backoffice und der Mitarbeiter ist eine anwendungsfreundliche Entwicklung der Applikation möglich und Fehlentwicklungen werden verhindert

# Projektvorbereitung

## Ist-Analyse

Das aktuelle Buchungssystem wird über einen Outlook Kalender abgebildet. Zur Reservierung eines Fahrzeugs wählt der Mitarbeiter das gewünschte Fahrzeug aus einer Liste von Fahrzeugen aus und sendet eine Buchungsanfrage an das Backoffice. Dem Mitarbeiter fehlt jedoch die Information, ob das gewünschte Fahrzeug verfügbar ist. Zudem gibt es keine Übersicht der aktuellen Reservierung und kommenden Fahrten. Das Backoffice überprüft die Verfügbarkeit des Fahrzeugs und gibt eine entsprechende Rückmeldung an den Mitarbeiter. Insbesondere dem Backoffice fehlt mit dem derzeitigen System eine Übersicht aller Reservierungen zur besseren Verwaltung des Fuhrparks.

Von besonderer Bedeutung für die Sicherheit der Mitarbeiter sind die erforderlichen Reparaturen an Fahrzeugen. Derzeit teilen die Mitarbeiter Defekte mündlich oder schriftlich mit. Die Mitarbeiter des Backoffice müssen dann eigenverantwortlich eine Liste der Reparaturen führen und den aktuellen Status im Blick haben. Eine Historie zu bereits erfolgten Reparaturen gibt es nicht.

Um einen Eindruck vom aktuellen System zu bekommen, zeigt Anhang A.1 einen Screenshot einer Buchung, die über den Outlook-Kalender vorgenommen wurde.

## Anwendungsfälle

Nach einem kontinuierlichen Feedback-Prozess zwischen der Backoffice-Abteilung und dem Autor des Projekts, wurden die verschiedenen Anwendungsfälle der Applikation definiert, die im Folgenden näher erläutert werden:

Das Backoffice möchte die heutigen Buchungen einsehen, da sie so sicherstellen können, dass die Schlüsselausgabe zur benötigten Zeit besetzt ist. Beim Anklicken einer Reservierung wird die Anwendung zu dem Buchen-Modul umgeleitet. Dieses Modul ist für die Verwaltung der Reservierungen verantwortlich.

Außerdem sind für das Backoffice die aktuellen Reparaturen relevant, um zum einen die Fahrzeuge rechtzeitig von der Werkstatt abzuholen, aber auch einen möglichen Engpass in der Flotte zu erkennen. Wenn in diesem Fall eine bestimmte Reparatur in der Liste angeklickt wird, wird die Anwendung an das Modul Verwalten weitergeleitet, das für die Verwaltung zuständig ist.

Ein Benutzer möchte auf einen Blick erkennen können, dass seine benötigten Fahrten korrekt im System gebucht sind. Dafür wird ihm eine Liste der nächsten Fahrten angezeigt. Auch in diesem Fall wird die Anwendung beim Anklicken der Reservierung auf das Buchen-Modul geleitet.

Alle möglichen Anwendungsfälle sind in Anhang A.2 dargestellt.

## Wirtschaftliche Betrachtung

## Projektkosten

Als Grundlage für die wirtschaftliche Betrachtung wurden die Kosten für die Durchführung des Projekts berücksichtigt.

Die folgende Tabelle zeigt alle Kosten des Projekts, sowohl auf Seiten des Personals als auch auf Seiten der eingesetzten Ressourcen. Bei den Personalkosten wird berücksichtigt, dass ein Auszubildender, der in der Tabelle als Azubi bezeichnet wird, das Unternehmen 25 Euro kostet. Die Kosten für einen Mitarbeiter betragen dagegen 100 Euro. Für technische Ressourcen wird eine Pauschale von 15 Euro pro Stunde angesetzt.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *Vorgang* | *Mitarbeiter* | *Zeit* | *Personal* | *Ressourcen* | *Gesamt* |
| Entwicklung | 1x Azubi | 70 h | 1.750,00 € | 1.050,00 € | 2.800,00 € |
| Fachgespräch | 1x Mitarbeiter | 5 h | 500,00 € | 75,00 € | 575,00 € |
| Endanwendertests | 2x Mitarbeiter | 1 h | 200,00 € | 15,00 € | 215,00 € |
| Abnahme | 1x Mitarbeiter | 1 h | 100,00 € | 15,00 € | 115,00 € |
|  |  |  | *Projektkosten gesamt* | | 3.705,00 € |

*Tabelle 2: Kalkulation der Projektkosten*

## Amortisationsdauer

Im Rahmen der Wirtschaftlichkeitsbetrachtung wurde eine Berechnung der Amortisation des Projektes durchgeführt. Wenn ein Benutzer über Outlook auf die Reservierungsinformationen zugreifen möchte, stößt er im aktuellen Prozess auf ein Hauptproblem, das gelöst werden sollte:

Um vollständige Informationen über die Reservierung zu erhalten, muss der Benutzer die Backoffice-Abteilung anrufen oder eine E-Mail schreiben. Sobald die Anfrage registriert ist, wird sie per E-Mail mit den vollständigen Informationen beantwortet. Es wurde geschätzt, dass der gesamte Vorgang im Durchschnitt sieben Minuten dauert.

Die Anzahl der monatlichen Anfragen an das Backoffice und damit die voraussichtliche Anzahl an Zugriffen auf die Dashboard-Komponente beläuft sich auf 50.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Vorgang |  | Zeit (alt) | Zeit (neu) | Einsparung |
| Buchungsinformationen anfordern |  | 2 min | 1 min | 1 min |
| Antwort auf die Reservierungsanfrage |  | 4 min | 1 min | 3 min |
| Bestätigung der Antwort |  | 1 min | 0 min | 1 min |
| *Zeiteinsparung gesamt pro Kalkulation* | | | | *5 min* |
| *Zugriffe pro Monat* | | | | *50* |
| Zeiteinsparung gesamt pro Monat  *Tabelle 3: Kalkulation der Zeiteinsparung* | | | | 250 min |

Berechnung der Amortisationsdauer:

*250 min/Monat x 12 Monate = 3000 min/Jahr = 50 h/Jahr  
50 h/Jahr x (100 € + 15€) = 5.750 €/Jahr  
 = 0,644 Jahre ≈ 235 Tage ≈ 8 Monate*

Die Amortisationszeit des Projekts beträgt knapp acht Monate. Die Umsetzung des Projektes verringert den Arbeitsaufwand und damit die laufenden Kosten stark. Wenn das Projekt einmal umgesetzt und gestartet ist, ist es wirtschaftlich tragfähig. Damit ist das Projekt wirtschaftlich gerechtfertigt.

## Nicht-monetäre Vorteile

In diesem Kapitel werden die Vorteile der Entwicklung und Implementierung dieser Anwendung aus nicht-monetärer Sicht aufgezeigt.

Eine der grundlegenden Ideen ist es dieses Produkt dem Portfolio des Unternehmens hinzufügen zu können. Dies kann zu weiteren Umsätzen führen oder neue Kunden auf das Unternehmen aufmerksam machen. Andererseits gewinnt die Backoffice-Abteilung an Komfort beim Zugriff auf die Informationen der Reservierungen und Fahrzeuge und hat die Möglichkeit, Statistiken zur Optimierung des Services zu erstellen.

Zusätzlich besteht das Potenzial die Anwendung den Anforderungen des Unternehmens laufend anzupassen und zu erweitern. Die Möglichkeit einer Analyse aller Buchungen hilft zudem dabei den Fuhrpark des Unternehmens zu optimieren, indem die Auslastung der unterschiedlichen Fahrzeugtypen ausgewertet wird.

## Entwicklungsplattform

Aufgrund von Erfahrungswerten und vorangegangenen Projekten wurde Angular als Frontend-Framework gewählt. Einer der Gründe für die Wahl dieses Frameworks ist, dass es die Programmierung von einseitigen Webanwendungen (Single Page Application) erleichtert. Darüber hinaus ergibt sich durch die Verwendung von ausgewählten Komponenten, die durch das "Material-Design" bereitgestellt werden, eine saubere und intuitive Oberfläche. Ein weiterer Grund, sich für dieses Framework zu entscheiden, ist die ausführliche Dokumentation, die zur Verfügung steht.

## Architekturdesign

Aufgrund der Wahl des Angular Frameworks wurde eine MVVM-Architektur gewählt. Dieses Entwurfsmuster ist eine Verfeinerung des MVC-Modells.

Die MVVM-Architektur hilft, die Geschäftslogik von der View oder Benutzeroberfläche zu trennen. Es fügt eine Zwischenkomponente für die Kommunikation zwischen der View (V) und dem Modell (M) hinzu, das ViewModel (VM).

Das Modell (M) beinhaltet die Implementierung der Anwendungsdomäne, sowie die Logik und den Zustand der Anwendung.

Die Ansicht (V) ist für die Anzeige der Informationen für den Benutzer zuständig. Sie bezieht die Daten über Bindings oder ruft auch die Methoden der ViewModel (VM)-Komponente auf.

Die Komponente ViewModel (VM) fungiert als Vermittler zwischen der Ansicht (V) und dem Modell (M). Das bedeutet, dass sie die Logik die Ansicht steuert. Dieses Verhalten impliziert, dass es die Aufrufe an das Modell (M) verarbeitet, um die Daten zu erhalten und sie so umzuwandeln, dass sie zum Zeitpunkt der Anzeige so strukturiert wie möglich sind. Auch diese Komponente implementiert Methoden oder Funktionen, die von der Ansicht aus zugänglich sind.

Dies führt zu einer völligen Unabhängigkeit bei der Entwicklung der Ansicht und der Logik der Komponenten.

Diagram

Description automatically generated

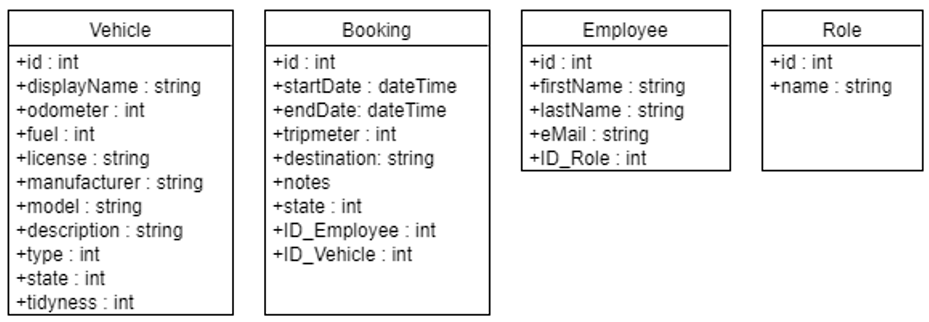
*Diagramm 2: MVVM-Modell*

## Entwurf der Datenstruktur

In Abstimmung mit allen Beteiligten ist eine Datenstruktur entstanden, mit der das Projekt des neuen Fahrzeugreservierungssystems des Unternehmens entwickelt wird. Das Projekt besteht aus mehreren Modulen, sodass alle erforderlichen Datenstrukturen für das Gesamtprojekt definiert wurden. Auf diese Weise werden alle Anforderungen, die die Backoffice-Abteilung stellt, erfüllt. Im Ergebnis gibt es vier verschiedene Klassen, die sich die unterschiedlichen Module der Anwendung teilen.

Für das Dashboard-Modul wird es notwendig sein, jede der Klassen zu verwalten. Dadurch können die geplanten Reservierungen sowie die damit verbundenen Informationen wie Zielort, Mitarbeiterdaten usw. angezeigt werden.

Diagramm der Klassen, die die Dashboard-Komponente verwendet:



*Diagramm 3: Klassendiagramm*

## Entwurf der Benutzeroberfläche

Bei der Entwicklung einer Benutzeroberfläche für die Dashboard-Komponente werden die Anforderungen der unterschiedlichen Nutzer berücksichtigt. Die Entwurfsmuster eines Dashboards sind immer ähnlich aufgebaut. Je nach Ausrichtung der Anwendung werden unterschiedliche Anforderungen gestellt. In diesem Fall werden die Anforderungen von der Backoffice-Abteilung gestellt.

Die wesentlichen Informationen sind sofort zugänglich. In der Backoffice-orientierten Oberfläche können die wichtigsten Informationen zu jeder geplanten Reservierung eingesehen werden. Diese Funktion ähnelt der Benutzeroberfläche für Mitarbeiter, wo sie auf einen Blick die wichtigsten Informationen der aktuellen Tagesbuchung sehen. Zudem ist die Reparaturhistorie der Fahrzeugflotte einsehbar.

Ein weiteres Merkmal jeder Dashboard-Komponente ist die Reduktion der Rohdaten, um Übersichtlichkeit zu schaffen. Dies wurde durch die Vereinfachung aller Informationen erreicht, die in der im vorherigen Kapitel erläuterten Datenstruktur enthalten sind. Daraus entsteht eine sortierte Darstellung der Informationen.

Die Prototypen der beiden Benutzeroberflächen sind in Anhang A.3 dargestellt.

# Implementierung

## Implementierung der Datenstruktur

Die Datenstruktur besteht aus vier Klassen (Fahrzeug, Buchung, Mitarbeiter und Rolle). Diese Klassen werden von der Backend-Logik bereitgestellt und im Dashboard-Client durch die entsprechenden Interfaces-Klassen übersetzt.

## Implementierung der Benutzeroberfläche

Die Implementierung der Benutzeroberfläche basiert auf den in der frühen Phase des Projekts entworfenen Mockups. Bei der Entwicklung der Dashboard-Komponente wurde eine einfache Logik gewählt, bei der es keine Interaktionsbuttons gibt. Dies gilt sowohl für die Benutzeroberfläche als auch für die Backoffice-Oberfläche. Beide Oberflächen, obwohl sie minimale Unterschiede aufweisen, haben eine ähnliche Struktur; eine Liste von informativen Tafeln mit den wichtigsten Daten. Alle Informationen werden automatisch aktualisiert. Die Informationen werden mithilfe des Observables-Patterns aktualisiert. Dieses Pattern erlaubt es, eine Datenbindung zwischen zwei oder mehreren Objekten zu definieren, um Daten zu übertragen. Das bedeutet, dass bei Änderungen an einem Objekt (Subject) das Objekt oder die Objekte, die dieses Objekt abonniert haben (Observer), die Änderungen wiedergeben. Die in den Mockups zu sehenden weiteren Menüpunkte wurden in anderen Projekten umgesetzt. Screenshots sind im Anhang A.4 beigefügt.

## Implementierung Geschäftslogik

Die Geschäftslogik besteht im Wesentlichen darin, zu entscheiden, welches Dashboard angezeigt werden soll. Sie wird in Abhängigkeit von der Rolle des Benutzers, der auf die Anwendung zugreift, festgelegt. In diesem Projekt ist es relevant, welche Informationen auf dem Dashboard angezeigt werden. Um die Informationen zu erhalten und dann in der Oberfläche darzustellen, wurde ein Service-Layer erstellt.

Die vorhandenen Klassen im Backend werden so gemappt, dass sie als Interfaces-Klassen verwendet werden können. Genauer gesagt, ist die Serviceschicht für die Serialisierung der verschiedenen JSON-Objekte durch bestimmte Anweisungen zuständig. Nach der Serialisierung verwenden die Funktionen des Service-Layer die verschiedenen Interfaces-Klassen. Diese Funktionen werden aufgerufen, um die erforderlichen Informationen zu rendern, die in den Panels angezeigt werden sollen.

Diese Panels werden ständig aktualisiert. Wird in der Buchungskomponente ein Fahrzeug gebucht, werden beim Zurückkehren zum Dashboard-Panel die aktualisierten Informationen mit der neuen Buchung angezeigt. Dies wird durch die Funktionalitäten der Service-Layer erreicht. Wenn die Dashboard-Komponente geladen wird, führen die Funktionen dieses Layers Aufrufe an die API durch, die wiederum Abfragen an die Datenbank stellt. Wenn die Datenbank mit der neuen Reservierung aktualisiert wird, gibt sie die aktualisierten Informationen an die Dienstebene zurück.

# Testphase

Es wurden eine Reihe von abschließenden Tests durchgeführt. Einerseits wurden falsche Buchungsdaten in die Datenbank eingegeben, um zu sehen, wie sich das Dashboard verhält. Die zuständige Logik hat korrekt reagiert und den Fehler in den Buchungsinformationen gemeldet. Es wurde auch der Fall getestet, in dem es keine Informationen zum Fahrzeugmodell gab. Die Logik reagierte wie erwartet und zeigte das dem Fahrzeugmodell gewidmete Feld leer an, anstatt unempfindlich zu bleiben und die Oberfläche nicht zu rendern.

# Projektabschluss

## Deployment

Nachdem das Projekt die Qualitätskontrolltests erfolgreich bestanden hat, wurde es freigegeben und den Anwendern zur Verfügung gestellt. Die Bereitstellung erfolgt auf einem Windows Server 2012 R2 über IIS (Internet Information Service) im Intranet des Unternehmens. Da es sich um eine Single Page Application (SPA) handelt, musste die Datei "web.config" korrekt konfiguriert werden.

## Abnahme

In der Abnahmephase der globalen Anwendung, zu der die Dashboard-Komponente gehört, wurde diese sowohl den Mitarbeitern als auch der Backoffice-Abteilung vorgestellt. Jeder der Benutzer testete die Funktionalitäten der Anwendung. Die Dashboard-Komponente wurde zunächst von einem Mitarbeiter mit einer normalen Rolle getestet. Er konnte sehen, wie die Anforderungen erfüllt wurden. Das Gleiche geschah mit einem Backoffice-Mitarbeiter. Am Ende wurde vereinbart, die Mitarbeiter in der Nutzung der globalen Anwendung zu schulen, zu der auch die Dashboard-Komponente gehört.

# Projektergebnis

## Soll-Ist-Vergleich

Es kann festgehalten werden, dass die Anforderungen erfüllt sind. Was die Projektvorbereitung betrifft, so konnte Zeit eingespart werden. Das ist darauf zurückzuführen, dass die Dashboard-Komponente zusammen mit anderen Komponenten in die globale Anwendung integriert ist. Was die Implementierung betrifft, so hat sich der Prozess aufgrund bestimmter Komplikationen in die Länge gezogen. Die Testphase, wie auch die Projektabnahme dauerte aufgrund der leichten Adaption der Anwender etwas weniger lang. Aufgrund der sprachlichen Herausforderung erforderte die Projektdokumentation etwas mehr Zeit. Insgesamt ist damit die von der IHK gesetzte Frist eingehalten worden.

|  |  |  |  |
| --- | --- | --- | --- |
| Vorgang | Soll | Ist | Differenz |
| Projektvorbereitung | 25 h | 22 h | -3 h |
| Implementierung | 24 h | 26 h | 2 h |
| Testphase | 7 h | 6 h | -1 h |
| Projektabschluss | 14 h | 16 h | 2 h |
| Gesamt | 70 h | 70 h | 0 h |

*Tabelle 4: Soll-Ist-Vergleich*

## Fazit

Während der Entwicklung des Projekts wurden die entsprechenden Kenntnisse für die korrekte Planung und Entwicklung desselben erworben. Ein wichtiges Thema war das Verständnis und die korrekte Ausführung von agilen Methodiken. Ein weiterer Punkt, der geholfen hat, war die vorherige Zusammenarbeit in anderen Projekten des Unternehmens, sowohl bei der Verwaltung von Zeit und Aufwand als auch bei der Gesamtvision dessen, was entwickelt wurde.

Dies zeigt sich z. B. im flüssigen Umgang mit dem für die Entwicklung der Anwendung gewählten Framework. Weiterhin wurden die von der Backoffice-Abteilung vorgeschlagenen Anforderungen erfüllt.

Abschließend ist festzuhalten, dass das Projekt einerseits die Kompetenzen des Autors als Anwendungsentwickler erhöht hat und andererseits die digitale Transformation des Unternehmens zum Leitmotiv des Geschäftsmodells gemacht hat.

## Ausblick

In Anbetracht der Tatsache, dass es eine Abgrenzung im Projekt gegeben hat, konnten einige der ursprünglich aufgeworfenen Anforderungen nicht entwickelt werden.

Eine dieser zukünftigen Funktionen ist, dass die Anwendung im Hintergrund laufen kann. Dies würde eine weitere damit verbundene Funktionalität implizieren: Das Anzeigen eines Dialogs eine halbe Stunde vor dem Beginn der Reservierung. In diesem Dialog werden die wichtigsten Informationen der Buchung angezeigt.

Eine weitere zukünftige Funktion wird die Anzeige einer Karte sein, die mit der aktuellen Buchung verbunden ist. Auf dieser Karte wird die Route zwischen dem Ausgangspunkt und dem Ziel angezeigt. Auf der Route werden Echtzeitinformationen wie Staus, Wetterbedingungen usw. angezeigt.

# Anlage

### Glossar

|  |  |  |
| --- | --- | --- |
| *Begriff* | *Abkürzung* | *Beschreibung* |
| Angular | - | Framework für Webanwendungen |
| TypeScript | Ts | Eine auf JavaScript basierende Programmiersprache |
| Cascading Style Sheets | CSS | Stylesheetsprache um das Gestalten von Webseiten und -anwendungen zu zentralisieren und zu vereinfachen |
| Hypertext Markup Language | HTML | Markuplanguage für die Darstellung von Text und Bildern auf Webseiten und -anwendungen |
| Material Design | - | Ein von Google bereitgestelltes System, das die Erstellung von qualitativ hochwertigen Android, iOS und Webapplikationen. |
| Model-View-Viewmodel | MVVM | Entwurfsmuster, das Darstellung und Logik einer Benutzeroberfläche trennt. |
| Graphical User Interface | GUI | Grafische Benutzeroberfläche |
| JavaScript Object Notation | JSON | Es ist ein leichtgewichtiges Textformat, das die Übertragung von Daten zwischen Anwendungen ermöglicht. |
| Application Programming Interface | API | Es handelt sich um eine Reihe von Definitionen und Protokollen, die zur Entwicklung und Integration von Anwendungssoftware verwendet werden. Weiterhin ermöglichen eine Serie von Regeln die Kommunikation zwischen zwei Softwareanwendungen. |

### Quellenverzeichnis

* Agile Methoden:

<https://agilemanifesto.org/>

<https://en.wikipedia.org/wiki/Agile_software_development>

* Angular Dokumentation:

<https://angular.io/docs>

* MVVM:

[https://naveenpete.wordpress.com/2016/09/07/mvc-mvvm-and- angular/](https://naveenpete.wordpress.com/2016/09/07/mvc-mvvm-and-%20%20angular/)

<https://medium.com/somos-codeicus/arquitectura-mvc-parte-2-daca367575a1>,

* Datenstruktur:

Dive into Design Patterns - Alexander Shvets

### Verwendete Software

Entwicklung:

* Microsoft Visual Studio Code 2019 (IDE)

Dokumentation:

* Microsoft Office 365 Word (Dokumentation und Tabellen)
* Microsoft Visio (Diagramme)

### Eidesstattliche Erklärung

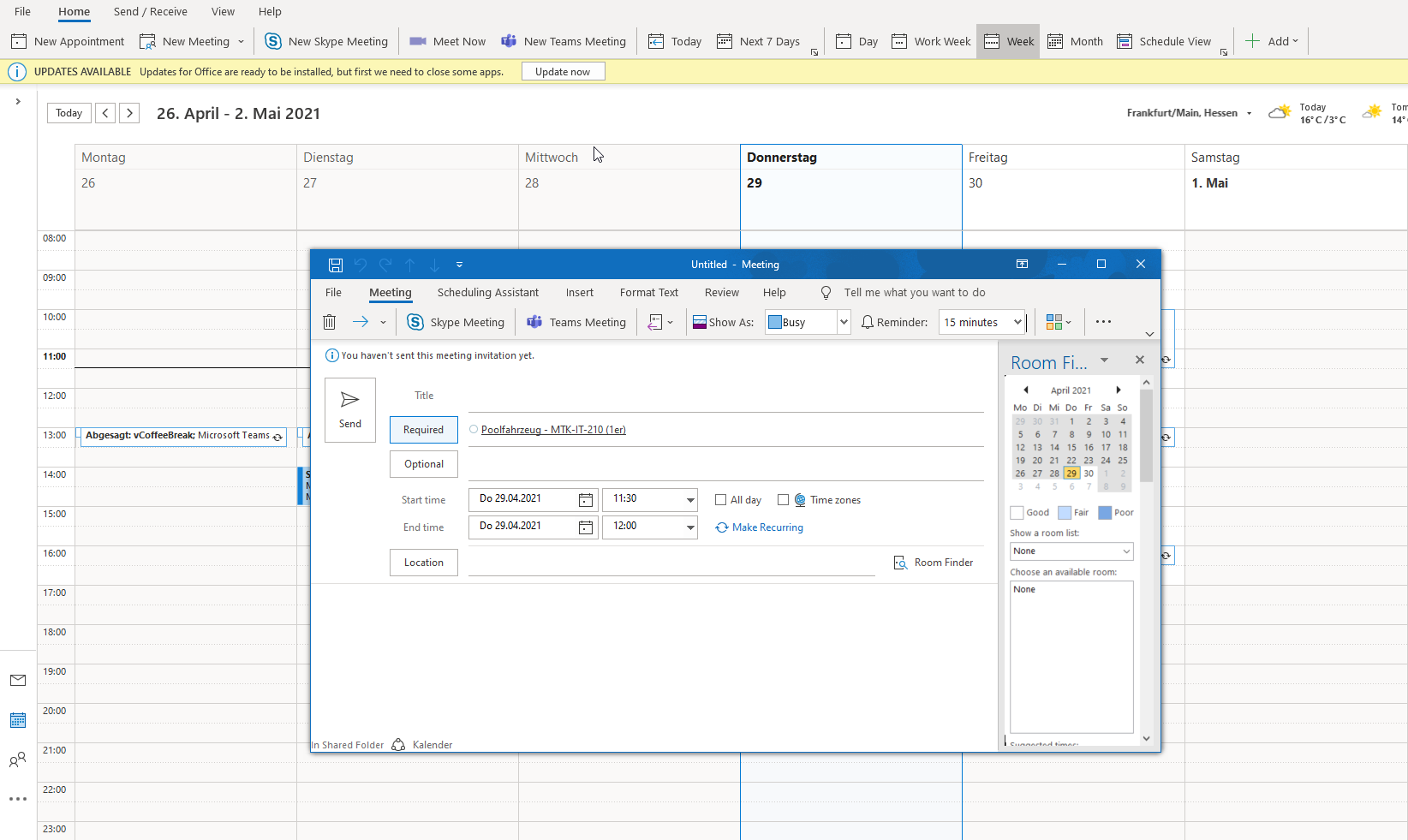
Hiermit versichere ich, dass ich die vorliegende Ausarbeitung selbständig verfasst und keine anderen, als die angegebenen Hilfsmittel benutzt habe.



Sulzbach, den 03.05.2021

# Anhang

### A.1 Screenshot einer Buchung im Altsystem



### A.2 Use-Case Diagramm

Diagram

Description automatically generated

### A.3 GUI Prototyp

Graphical user interface, application

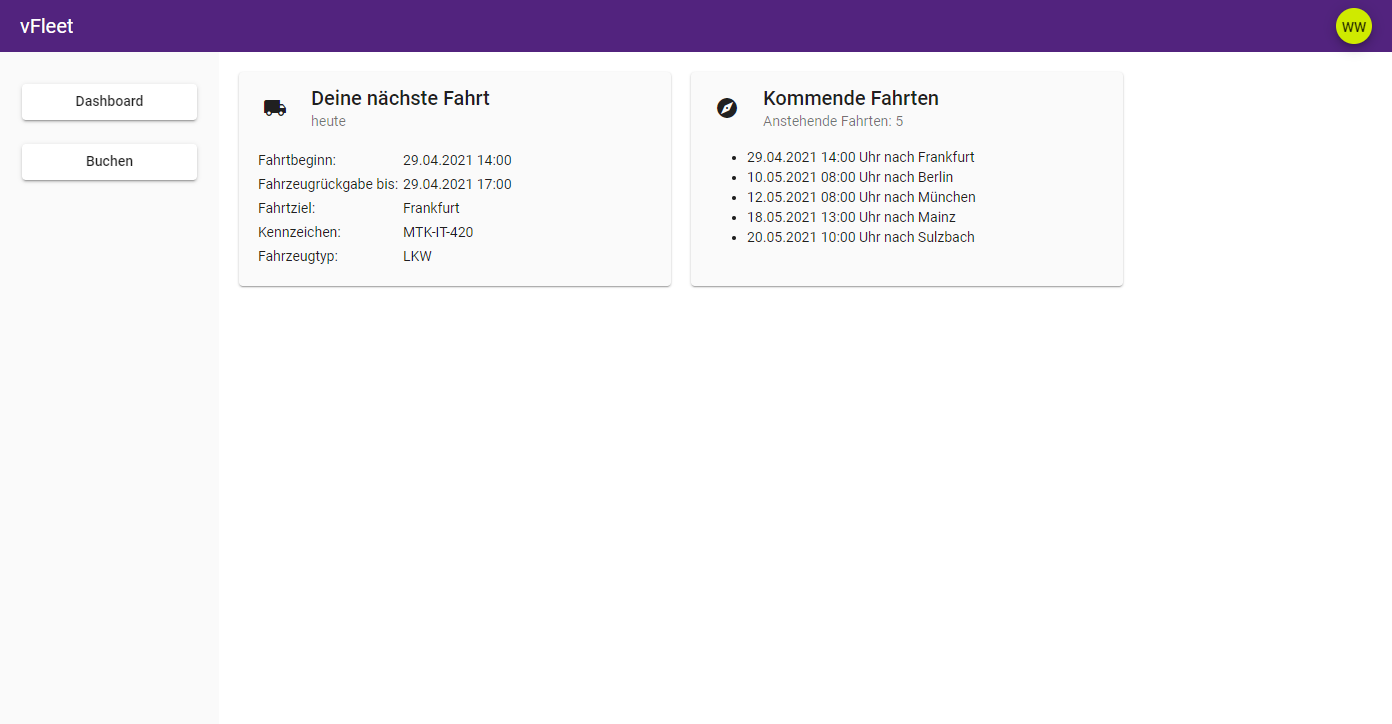
Description automatically generated

A picture containing graphical user interface

Description automatically generated

### A.4 Screenshots der Anwendung

Mitarbeiter Dashboard



Backoffice Dashboard

