

Projektaufgabe INF2018: C# und .NET - „Fleet Management“

Das aufstrebende Jungunternehmen „moomax UG (haftungsbeschränkt)“ ist dabei, eine standesgemäße Dienstwagenflotte aufzubauen, um richtig durchstarten zu können (mit Porsche startet es sich ja bekanntlich am besten durch).

Um sich nicht unnötig mit Nebensächlichkeiten aufzuhalten, beauftragt das Start-Up den IT-Dienstleister „Give me Power GmbH“ mit der Erstellung eines Systems für das Flottenmanagement.

Leider investiert die „Give me Power GmbH“ das Budget vor allem in Latte Macchiato und Röhrenjeans, so dass zum Ende des Projekts lediglich eine Datenbank (SQLite-Format) vorliegt.

Geschockt von diesem unseriösen Geschäftsgebaren wendet sich der Geschäftsführer der „moomax UG (haftungsbeschränkt) Christian Wolfgang L. an Sie und beauftragt Sie mit der Entwicklung eines neuen „Fleet Management“-Systems. Da aus dem Vorprojekt gelernt wurde, wie teuer die Erstellung einer Datenbank sein kann, soll Ihr Flottenmanagementsystem kompatibel zur vorhandenen Datenbank sein.

Lesen Sie zuerst die allgemeinen Tipps zur Lösung der Projektaufgabe im Anhang!

Bearbeitungszeitraum und Abgabe

Der späteste Termin für die **Abgabe** des Projekts ist der **21.06.2020 um 23:50 Uhr**. Spätere Abgaben können nicht entgegen genommen werden.

Bitte beachten Sie:

Aus Gründen der Gleichbehandlung können inhaltliche Fragen zur Klausurersatzleistung nicht mehr beantwortet werden, sobald die erste Abgabe erfolgt ist.

Form der Abgabe:

Maßgeblich ist der Quelltext und das Visual Studio Projekt inkl. aller referenzierter Dateien. Nicht abgegeben werden sollen die erstellten Objekt- und Binärdateien sowie referenzierte Nuget-Pakete. Bitte entfernen Sie diese vor der Abgabe aus diesem Projekt (Referenzen zu Nuget-Paketen müssen erhalten bleiben).

Die Abgabe muss per E-Mail an **nils.schmid@me.com** erfolgen.

Andere Wege der Abgabe (z. B. durch das Hochladen der Dateien auf einen Cloud-Service) werden nicht akzeptiert.

Aufgabenstellung (70 Punkte)

Hinweis: Für diese Aufgabenstellung können Sie **70 Punkte** erhalten. Zusammen mit den maximal **30 erreichbaren Punkten** aus der Bearbeitung der Aufgabenblätter ergibt sich eine Gesamtpunktzahl von **100 maximal möglichen Punkten**.

Entwickeln Sie für das Unternehmen eine moderne Anwendung, welche kompatibel zur bestehenden Datenbank ist.

Die Realisierung muss mit C#, WPF und WCF auf Basis von .NET 4.8.x erfolgen. Die Verwendung von MVVMC, dem Nuget-Paketmanager, MSTest, Autofac und Fluent NHibernate mit Repository-Pattern wird vorgeschrieben.

Die Software soll als Client-Server-Software mittels Verwendung von WCF (wsHttpBinding) realisiert werden.

Es werden nachfolgend nur die zu realisierenden Funktionalitäten beschrieben. Bei der eigentlichen Umsetzung werden den Studierenden große Freiheiten eingeräumt. Dies soll kreative Eigenleistungen fördern und deren Bewertung ermöglichen.

Dies bezieht sich sowohl auf die Architektur der Anwendung als auch auf das Design der Oberflächen und das Bedienkonzept der Anwendung.

Die bestehende Datenbank soll zur Entwicklung der neuen Software analysiert und genutzt werden. Hierfür stehen im Internet verschiedene Open-Source-Programme zur Verwaltung von SQLite-Datenbanken zur Verfügung (z. B. SQLite Studio).

Das vorliegende Datenbankschema darf nicht verändert werden.

Bitte beachten Sie:

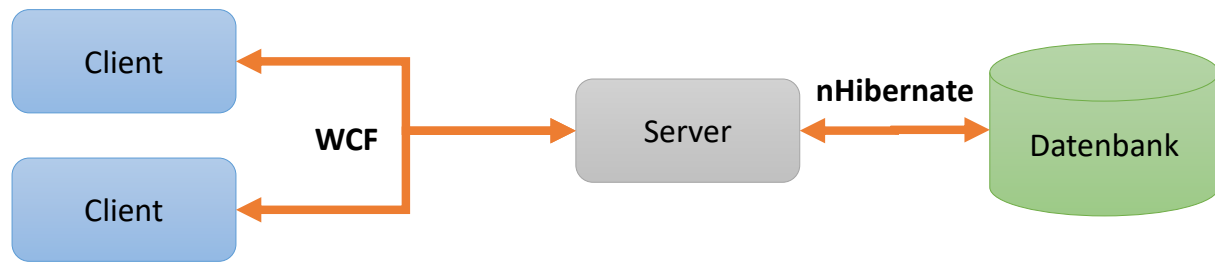
- Auftretende Programmabstürze führen zu Punktabzug!
- Fehlende Eingabevalidierungen führen ebenfalls zu Punktabzug, auch wenn das Programm nicht abstürzt.
- Lässt sich das abgegebene Projekt nicht fehlerfrei kompilieren und starten, führt dies ebenfalls zu Punktabzug!
- Fehlermeldungen müssen jederzeit klar formuliert, für den Anwender verständlich und situationsbezogen sein. Ansonsten führt dies zu Punktabzug!

Von den Studierenden wird die Fähigkeit zur Recherche im Internet mittels geeigneter Suchmaschinen erwartet.

Hinweis: Das Punktesystem ist analog zur Bewertung von Studienarbeiten

Client-Server-System (10 Punkte):

Nachfolgend wird der grundlegende Aufbau beschrieben, den Sie umsetzen sollen.



Server:

Realisieren Sie den Server auf Basis einer Konsolenanwendung, die einen WCF-Dienst zur Kommunikation bereitstellt.

Verwenden Sie Autofac als Dependency Injection Container.

Implementieren Sie die Kommunikation mit der Datenbank mittels Fluent-nHibernate.

Setzen Sie hierzu das Konzept des „Optimistic Locking“ um.

Ermöglichen Sie den gleichzeitigen Zugriff von mehr als einer Client-Anwendung zur selben Zeit.

Client:

Realisieren Sie den Client auf Basis einer WPF-Anwendung, die Sie mit dem MVVMC-Entwurfsmuster aufbauen.

Der Client darf nicht direkt mit der Datenbank kommunizieren. Der Client darf ausschließlich mittels WCF mit dem Server kommunizieren.

Verwenden Sie Autofac als Dependency Injection Container.

Der Client dient sowohl zum Management der Aufträge als auch zur Administration des Systems.

Authentifizierung:

Anwender und Administratoren, die das Softwaresystem verwenden möchten, müssen sich zunächst am System anmelden. Nur berechtigte Benutzer dürfen dies tun.

Navigationsmöglichkeit:

Das Softwaresystem muss über eine Navigationsmöglichkeit zum Öffnen der nachgehend beschriebenen Module verfügen. Es wird dabei zwischen Modulen unterschieden, die für alle Benutzer sichtbar sind und solchen, die lediglich Administratoren zur Verfügung stehen.

Realisieren Sie hierzu eine Menüführung, die die Auswahl der Module ermöglicht.

Allgemeine Module:

- **Startmodul**
Begrüßung des Benutzers
- **Dashboard: Kosten pro Monat**
Ansicht über die laufenden Kosten pro Monat
- **Dashboard: Kosten nach Geschäftsbereich**
Ansicht über die lfd. Kosten pro Monat und Geschäftsbereich
- **Fahrzeuge**
Verwaltung der Dienstwagen
- **Mitarbeiter**
Verwaltung der Mitarbeiter
- **Geschäftsbereiche**
Verwaltung der Geschäftsbereiche

Administrative Module:

- **Benutzer**
Benutzerverwaltung

Erscheinungsbild der Oberflächen:

Das Design soll sich an modernen Windows-Anwendungen orientieren und für Anwender ohne Expertenwissen intuitiv bedienbar sein.

Die Anwendung soll optisch ansprechend gestaltet werden. Hierbei ist auf die integrierten Funktionalitäten für Theming und Styling von WPF zurückzugreifen.

Um die ansprechende Gestaltung der Oberfläche zu unterstützen, finden Sie im Internet auch frei verfügbare Icon-Bibliotheken zur Gestaltung der Oberfläche (z. B. Schaltflächen, etc.).

The screenshot shows a Windows-style application window titled "Fleet Management". On the left is a sidebar menu with options: Start, Kosten/Monat, Kosten/Geschäftsbereich, **Fahrzeuge** (highlighted), Mitarbeiter, Geschäftsbereiche, Administration, and Benutzer. The main area is titled "Fahrzeuge verwalten" and contains a table with three columns: Kennzeichen, Marke, and Modell. The table lists three vehicles: K-O 2000 (Porsche 911 Turbo), K-MM 1300 (BMW M5), and K-MM 1400 (Audi S8). Below the table are tabs for "Allgemein" and "Mitarbeiter". The "Allgemein" tab is active, showing a form for vehicle details. The form includes fields for Kennzeichen (K-MM1300), Marke (BMW), Modell (M5), Versicherung/Jahr (EUR) (1900,00), and a Leasing section with fields for Beginn (01.01.2019), Ende (01.01.2024), and Leasingrate (EUR) (2300,00). At the top of the main area, there are buttons for "Neu", "Speichern", and "Löschen".

Kennzeichen	Marke	Modell
K-O 2000	Porsche	911 Turbo
K-MM 1300	BMW	M5
K-MM 1400	Audi	S8

Allgemein | Mitarbeiter

Kennzeichen: K-MM1300

Marke: BMW | Modell: M5

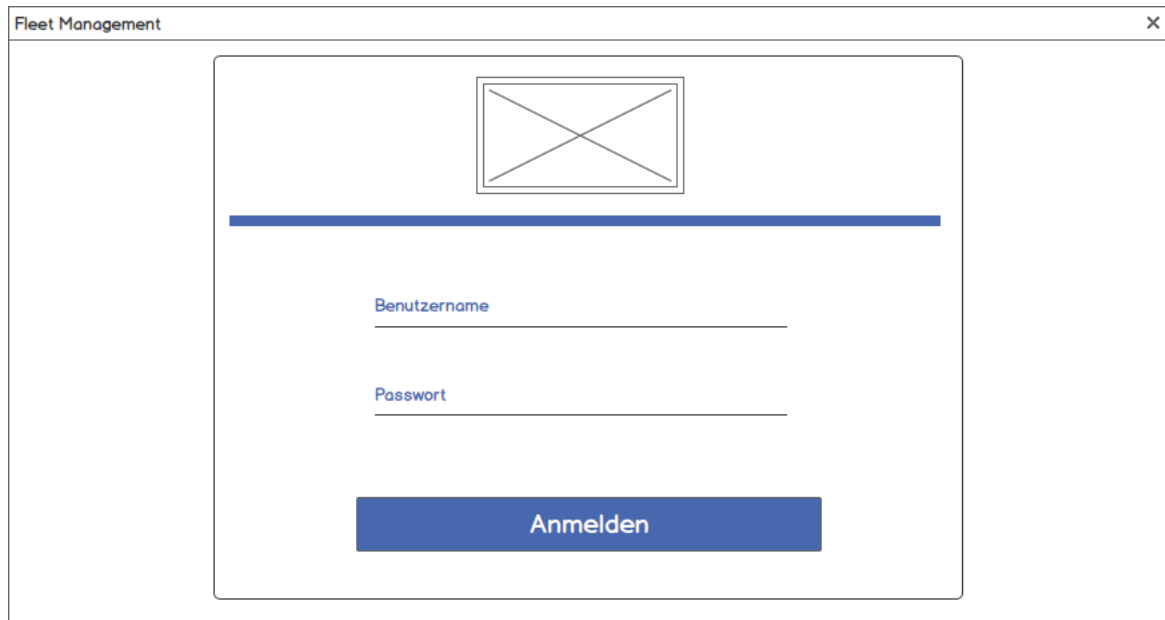
Versicherung/Jahr (EUR): 1900,00

Leasing: Beginn: 01.01.2019 | Ende: 01.01.2024 | Leasingrate (EUR): 2300,00

Authentifizierung (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite sind gemäß den Anforderungen umzusetzen.

Zeigen Sie nach dem Start des Clients zunächst ein Anmeldefenster an. Hier können sich die Anwender mit Benutzername und Passwort anmelden.



Bei erfolgreicher Anmeldung schließen Sie das Anmeldefenster und starten Sie das Hauptfenster (mit dem Startmodul).

Bei misslungener Anmeldung zeigen Sie eine entsprechende (benutzerfreundliche) Fehlermeldung an.

Benutzer zeichnen sich durch folgende (hier relevante) Eigenschaften aus:

- Benutzername (Username | Benutzername (Anmeldename) des Benutzers)
- Passwort (Password | Passwort des Benutzers)

Tabelle: Users

Der Benutzername wird ohne Beachtung von Groß- und Kleinschreibung überprüft (also wäre folgendes **äquivalent**: „Rolf“, „rolf“).

Wichtig:

In einer Datenbank speichert man niemals Passwörter! **Niemals!** Deshalb liegen die Passwörter in der vorliegenden Datenbank auch nicht im Klartext, sondern nur als „gesalzene“ Hashwerte (BCrypt) vor.

Zur Überprüfung des eingegebenen Passworts müssen Sie also zunächst dessen BCrypt-Hash errechnen und diesen „salzen“. Da die Hashwerte von Passwörtern niemals (**Niemals!**) den Server verlassen dürfen, führen Sie diese Berechnung auf dem Server durch!

Laden Sie hierzu die Bibliothek „**BCrypt.Net-Next**“ mit dem *Nuget-Paketmanager* herunter und verwenden Sie folgende Methode der Klasse „**BCrypt**“ zur Überprüfung des Passworts:

Methode: Verify

Beispiel: `var equals = BCrypt.Net.BCrypt.Verify("geheim", hash);`

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

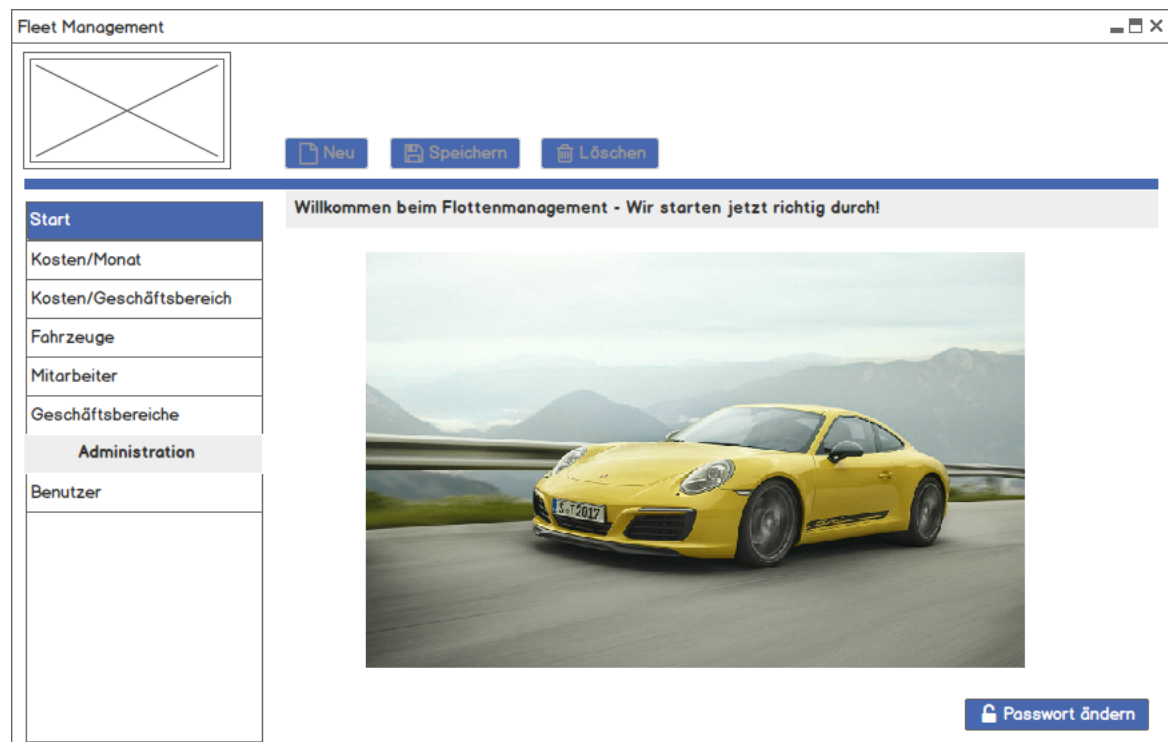
Tipp: In der mitgelieferten Datenbank sind bereits 3 Benutzer angelegt:

Benutzername	Passwort	Administrator?
admin	admin	ja
horst	geheim	nein
frank	geheim	nein

Startmodul (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.

Zeigen Sie nach der erfolgreichen Anmeldung zunächst das Startmodul im Hauptfenster an.



Das Startmodul begrüßt den Anwender und ermöglicht ihm, sein Passwort zu ändern (Schaltfläche).

Passwort ändern (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.

Das Startmodul enthält eine Schaltfläche, deren Betätigung ein Fenster zur Änderung des Passworts des angemeldeten Benutzers anzeigt.



Der Benutzer muss einmal sein Passwort und zweimal das neue Passwort eingeben. Prüfen Sie, dass die neuen Passwörter übereinstimmen und verwenden Sie danach folgende Methode der Klasse „**BCrypt**“ aus der Bibliothek „**BCrypt.Net-Next**“, um einen „gesalzenen“ Hash-Wert für das neue Passwort zu generieren:

Methode: ValidateAndReplacePassword

Beispiel: BCrypt.Net.BCrypt.ValidateAndReplacePassword("alt", hash, "neu");

Speichern Sie den neuen Hash-Wert in der Datenbank!

Benutzer zeichnen sich durch folgende (hier relevante) Eigenschaften aus:

- Passwort (Password | Hash-Wert des Passworts des Benutzers)

Tabelle: Users

Stimmen die neuen Passwörter nicht überein oder hat der Anwender ein falsches altes Passwort eingegeben, zeigen Sie dies dem Anwender mithilfe einer adäquaten, benutzerfreundlichen Fehlermeldung an.

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

Administration: Benutzer verwalten (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.

Dieses Modul ist nur für angemeldete Administratoren sichtbar.

Realisieren Sie folgende Funktionalitäten zur Abbildung der Verwaltung:
Möglichkeiten zur Anzeige, Neuanlage, zum Speichern/Bearbeiten und Löschen.

Benutzer müssen sich durch einen eindeutigen Benutzernamen (ohne Beachtung von Groß- und Kleinschreibung, also wäre folgendes **nicht** möglich: „Rolf“, „rolf“) unterscheiden.

Ein Benutzer darf nicht gelöscht werden, so lange es noch Daten gibt, die mit ihm verknüpft sind.

Dem Anwender soll dieser Sachverhalt in einer verständlichen Meldung (oder einem anderen benutzerfreundlichen Vorgehen) verdeutlicht werden.

Benutzer zeichnen sich durch folgende Eigenschaften aus:

- Benutzername (Username | Benutzername (Anmeldename) des Benutzers)
- Vorname (Firstname | Vorname des Benutzers)
- Nachname (Lastname | Nachname des Benutzers)
- Passwort (Password | Hash-Wert des Passworts des Benutzers)
- Ist Administrator? (IsAdmin | Ist der Benutzer ein Administrator?)

Tabelle: Users

Die Benutzer sollen in einer Listendarstellung angezeigt werden. Benutzer sollen auswählbar sein. Für den ausgewählten Benutzer soll eine Detailansicht eingeblendet werden, auf der die weiteren Eigenschaften verwaltet werden können.

The screenshot shows a web application titled 'Fleet Management'. On the left is a sidebar menu with options: Start, Kosten/Monat, Kosten/Geschäftsbereich, Fahrzeuge, Mitarbeiter, Geschäftsbereiche, Administration, and Benutzer (highlighted). The main content area is titled 'Benutzer verwalten' and contains a table of users. Above the table are buttons for 'Neu', 'Speichern', and 'Löschen'. Below the table, the details for the selected user 'horst (Horst Müller)' are shown, including input fields for 'Benutzername', 'Vorname', and 'Nachname', and a checkbox for 'Administrator'.

Benutzername	Vorname	Nachname	Admin
admin		Administrator	<input checked="" type="checkbox"/>
horst	Horst	Müller	<input type="checkbox"/>
frank	Frank	Meyer	<input type="checkbox"/>

horst (Horst Müller)

Benutzername

Vorname

Nachname

☐ Administrator

Das Passwort eines Benutzers wird nicht angezeigt und kann nicht geändert werden. Bei der Neuanlage wird ein Initialpasswort „geheim“ (kleingeschrieben) erzeugt und als Hashwert gespeichert. Verwenden Sie hierzu die folgende Methode der Klasse „**BCrypt**“ aus der Bibliothek „**BCrypt.Net-Next**“:

Methode: HashPassword

Beispiel: `var hash = BCrypt.Net.BCrypt.HashPassword("geheim");`

Vergessen Sie nicht die Abbildung des Optimistic Lockings!

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

Geschäftsbereiche verwalten (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.

Dieses Modul ist für alle angemeldeten Benutzer sichtbar.

Realisieren Sie folgende Funktionalitäten zur Abbildung der Verwaltung:

Möglichkeiten zur Anzeige, Neuanlage, zum Speichern/Bearbeiten und Löschen.

Geschäftsbereiche müssen sich durch einen eindeutigen Namen unterscheiden (ohne Beachtung von Groß- und Kleinschreibung, also wäre folgendes **nicht** möglich: „human resources“, „Human resources“).

Ein Geschäftsbereich darf nicht gelöscht werden, so lange es noch Daten gibt, die ihm zugeordnet sind.

Dem Anwender soll dieser Sachverhalt in einer verständlichen Meldung (oder einem anderen benutzerfreundlichen Vorgehen) verdeutlicht werden.

Geschäftsbereiche zeichnen sich durch folgende Eigenschaften aus:

- Name (Name | Name des Geschäftsbereichs)
- Beschreibung (Description | Optionale Beschreibung des Geschäftsbereichs)

Tabelle: BusinessUnits

Die Geschäftsbereiche sollen in einer Listendarstellung angezeigt werden. Geschäftsbereiche sollen auswählbar sein. Für den ausgewählte Geschäftsbereich soll eine Detailansicht eingeblendet werden, auf der die weiteren Eigenschaften verwaltet werden können.

The screenshot shows a web application window titled "Fleet Management - Geschäftsbereiche verwalten". On the left is a sidebar menu with options: Start, Kosten/Monat, Kosten/Geschäftsbereich, Fahrzeuge, Mitarbeiter, **Geschäftsbereiche** (selected), Administration, and Benutzer. The main content area has a header with a placeholder image and three buttons: "Neu", "Speichern", and "Löschen". Below this is a table titled "Geschäftsbereiche verwalten" with two columns: "Name" and "Beschreibung". The table contains three rows: "Business Development" (Durchstarten!), "Geschäftsführung" (der Chef), and "Facility Management" (der Schwager vom Chef). The "Geschäftsführung" row is highlighted in blue. Below the table is a dashed line and a section titled "Geschäftsführung" which contains two form fields: "Name" with the value "Geschäftsführung" and "Beschreibung" with the value "der Chef".

Geschäftsbereiche verwalten	
Name	Beschreibung
Business Development	Durchstarten!
Geschäftsführung	der Chef
Facility Management	der Schwager vom Chef

Geschäftsführung	
Name	Geschäftsführung
Beschreibung	der Chef

Vergessen Sie nicht die Abbildung des Optimistic Lockings!

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

Mitarbeiter verwalten (10 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen. Dieses Modul ist für alle angemeldeten Benutzer sichtbar.

Realisieren Sie folgende Funktionalitäten zur Abbildung der Verwaltung:
Möglichkeiten zur Anzeige, Neuanlage, zum Speichern/Bearbeiten und Löschen.

Mitarbeiter müssen sich durch eine eindeutige Personalnummer unterscheiden.

Ein Mitarbeiter darf nicht gelöscht werden, so lange es noch Daten gibt, die ihm zugeordnet sind. Dem Anwender soll dieser Sachverhalt in einer verständlichen Meldung (oder einem anderen benutzerfreundlichen Vorgehen) verdeutlicht werden.

Mitarbeiter zeichnen sich durch folgende Eigenschaften aus:

- Vorname (FirstName | Vorname des Mitarbeiters)
- Nachname (LastName | Nachname des Mitarbeiters)
- Personalnummer (EmployeeNumber | Personalnummer des Mitarbeiters)
- Anrede (Salutation | Anrede des Mitarbeiters - optional)
- Titel (Title | Akademischer Titel des Mitarbeiters - optional)
- Geschäftsbereich (BusinessUnitId | Verknüpfung zum Geschäftsbereich)

Tabelle: Employees

Die Mitarbeiter sollen in einer Listendarstellung angezeigt werden. Mitarbeiter sollen auswählbar sein. Für den ausgewählten Mitarbeiter soll eine Detailansicht eingeblendet werden, auf der die weiteren Eigenschaften verwaltet werden können.

Jeder Mitarbeiter muss einem Geschäftsbereich zugeordnet werden (Combobox, Fremdschlüssel zur Tabelle „BusinessUnits“).

The screenshot shows a web application titled "Fleet Management". On the left is a sidebar menu with items: Start, Kosten/Monat, Kosten/Geschäftsbereich, Fahrzeuge, Mitarbeiter (highlighted), Geschäftsbereiche, Administration, and Benutzer. The main content area is titled "Mitarbeiter verwalten". It features a table with three columns: "Nr.", "Vorname", and "Nachname". The table contains three rows: (4711, Jens, Mander), (2342, Christian Wolfgang, L.), and (110, Ralf, Raser). The row with ID 2342 is selected. Below the table, the details for "Christian Wolfgang L. (2342)" are shown. These details include input fields for "Personalnr." (2342), "Anrede" (Herr), "Titel" (Dr.), "Vorname" (Christian Wolfgang), "Nachname" (L.), and a dropdown menu for "Geschäftsbereich" (Geschäftsführung). Above the table, there are buttons for "Neu", "Speichern", and "Löschen".

Nr.	Vorname	Nachname
4711	Jens	Mander
2342	Christian Wolfgang	L.
110	Ralf	Raser

Christian Wolfgang L. (2342)

Personalnr. 2342 Anrede Herr Titel Dr.

Vorname Christian Wolfgang Nachname L.

Geschäftsbereich Geschäftsführung

Vergessen Sie nicht die Abbildung des Optimistic Lockings!

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

Fahrzeuge verwalten (15 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.
Dieses Modul ist für alle angemeldeten Benutzer sichtbar.

Realisieren Sie folgende Funktionalitäten zur Abbildung der Verwaltung:
Möglichkeiten zur Anzeige, Neuanlage, zum Speichern/Bearbeiten und Löschen.

Fahrzeuge müssen sich durch ein eindeutiges Kennzeichen unterscheiden (ohne Beachtung von Groß- und Kleinschreibung, also wäre folgendes **nicht** möglich: „FDS-E 1337“, „FDS-e 1337“).

Fahrzeuge werden mit Mitarbeitern verknüpft. Wird ein Fahrzeug gelöscht, werden automatisch alle Verknüpfungen zu Mitarbeitern mitgelöscht (nicht jedoch die Mitarbeiter).

Fahrzeuge zeichnen sich durch folgende Eigenschaften aus:

- Kennzeichen (LicensePlate | KFZ-Kennzeichen)
- Marke (Brand | Automobilmarke)
- Modell (Model | Fahrzeugmodell)
- Versicherung/Jahr (Insurance | Jährliche Kosten für die KFZ-Versicherung)
- Leasing – Beginn (LeasingFrom | Startdatum des Leasingvertrags)
- Leasing – Ende (LeasingTo | Enddatum des Leasingvertrags)
- Leasing – Leasingrate (LeasingRate | Monatliche Leasingrate)

Tabelle: Vehicles

Verknüpfungen zu Mitarbeitern zeichnen sich durch folgende Eigenschaften aus:

- Beginn (StartDate | Startdatum für die Verknüpfung)
- Ende (EndDate | Optionales Enddatum für die Verknüpfung)
- Fahrzeug (VehicleId | Verknüpfung zum Fahrzeug)
- Mitarbeiter (EmployeeId | Verknüpfung zum Mitarbeiter)

Tabelle: VehicleToEmployeeRelation

Die Fahrzeuge sollen in einer Listendarstellung angezeigt werden. Fahrzeuge sollen auswählbar sein. Für das ausgewählte Fahrzeug soll eine Detailansicht eingeblendet werden, diese enthält ein TabControl mit 2 Tabs („Allgemein“ und „Mitarbeiter“), auf denen die weiteren Eigenschaften verwaltet werden können.

Im Tab „Allgemein“ können die grundlegenden Fahrzeugdaten verwaltet werden.

Fleet Management

Neu

Speichern

Löschen

Start

Kosten/Monat

Kosten/Geschäftsbereich

Fahrzeuge

Mitarbeiter

Geschäftsbereiche

Administration

Benutzer

Fahrzeuge verwalten

Kennzeichen	Marke	Modell
K-O 2000	Porsche	911 Turbo
K-MM 1300	BMW	M5
K-MM 1400	Audi	S8

Allgemein

Mitarbeiter

Kennzeichen

K-MM1300

Marke

BMW

Modell

M5

Versicherung/Jahr (EUR)

1900,00

Leasing

Beginn

01.01.2019

Ende

01.01.2024

Leasingrate (EUR)

2300,00

Im Tab „Mitarbeiter“ werden die Mitarbeiter mit den Fahrzeugen verknüpft. Über 2 Schaltflächen ist es möglich

- Mitarbeiter hinzuzufügen
- Mitarbeiter zu entfernen

Die verknüpften Mitarbeiter werden in einer Listenansicht angezeigt (siehe Mockup). Die Reihenfolge erfolgt gemäß der Beginn- und End-Daten der Verknüpfung (aktuellste unten).

Fleet Management

Neu

Speichern

Löschen

Start

Kosten/Monat

Kosten/Geschäftsbereich

Fahrzeuge

Mitarbeiter

Geschäftsbereiche

Administration

Benutzer

Fahrzeuge verwalten

Kennzeichen	Marke	Modell
K-O 2000	Porsche	911 Turbo
K-MM 1300	BMW	M5
K-MM 1400	Audi	S8

Allgemein

Mitarbeiter

+

-

Mitarbeiter	Beginn	Ende
Dr. Petra Maier	01.01.2019	31.01.2020
Anita Kupfer	01.02.2020	

Nach Betätigung der Schaltfläche für das Hinzufügen einer Verknüpfung zu einem Mitarbeiter wird ein Dialog geöffnet, der die Auswahl eines Mitarbeiters und die Angabe des Beginn-Datums (vorbelegt mit dem aktuellen Datum) erzwingt sowie die Angabe des optionalen End-Datums (nicht vorbelegt, initial leer) ermöglicht. Es werden nur die Mitarbeiter zur Auswahl angeboten, die noch nicht mit dem aktuellen Fahrzeug verknüpft sind.

Nr.	Vorname	Nachname	Bereich
4711	Jens	Mander	
2342	Christian Wolfgang	L.	
110	Ralf	Raser	

von 01.01.2019 bis 01.01.2024

Hinzufügen Abbrechen

Nach Betätigung der Schaltfläche für das Entfernen der Verknüpfung zu einem Mitarbeiter wird eine MessageBox mit der Frage „Sind Sie sicher, dass Sie die Verknüpfung zum ausgewählten Mitarbeiter entfernen wollen?“ (Ja / Nein Schaltflächen) angezeigt, sofern ein Element der Liste ausgewählt ist. Wird die Frage mit „Ja“ bestätigt, wird die Verknüpfung gelöscht.

Vergessen Sie nicht die Abbildung des Optimistic Lockings!

Schreiben Sie 6 relevante Unit-Tests für diese Funktionalität.

Kosten pro Monat (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.

Dieses Modul ist für alle angemeldeten Benutzer sichtbar.

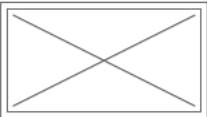
In einer Listenansicht sollen die monatlichen Kosten für die Fahrzeuge ausgegeben werden.

Folgende Spalten sollen dargestellt werden:

- Monat (Monat und Jahr, z. B. „Januar 2019“)
- Anzahl (Anzahl der Fahrzeuge, für die es einen aktiven Leasingvertrag gibt)
- Kosten (Summe der Kosten, die in diesem Monat entstehen (Leasingrate + anteilige Versicherungssumme))

Tabellen: Vehicles

Die Liste ist nach Monaten sortiert (neuester steht unten). Die Zeilen der Liste werden über die bestehenden Leasingverträge ermittelt (vom Monat, in dem der „älteste“ Leasingvertrag beginnt, bis zum Monat, in dem der „neueste“ Leasingvertrag endet).

Fleet Management			
			
<div>Neu Speichern Löschen</div>			
Start	Kosten/Monat		
Kosten/Monat	Monat	Anzahl	Kosten
Kosten/Geschäftsbereich	Januar 2019	25	€ 5555,00
Fahrzeuge	Februar 2019	25	€ 5555,00
Mitarbeiter	März 2019	25	€ 5555,00
Geschäftsbereiche	April 2019	25	€ 5555,00
Administration	Mai 2019	25	€ 5555,00
Benutzer	Juni 2019	25	€ 5555,00
	Juli 2019	25	€ 5555,00
	August 2019	25	€ 5555,00
	September 2019	25	€ 5555,00
	Oktober 2019	25	€ 5555,00
	November 2019	25	€ 5555,00
	Dezember 2019	25	€ 5555,00

Verwenden Sie zusätzlich zur nHibernate-Funktionalität LINQ, um die notwendige Abfrage zu implementieren.

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

Kosten je Geschäftsbereich pro Monat (5 Punkte)

Nachfolgend werden die Anforderungen an diese Funktionalität beschrieben. Sowohl die Client- als auch die Serverseite ist gemäß den Anforderungen umzusetzen.

Dieses Modul ist für alle angemeldeten Benutzer sichtbar.

In einer Listenansicht sollen die monatlichen Kosten für die Fahrzeuge je Geschäftsbereich ausgegeben werden.

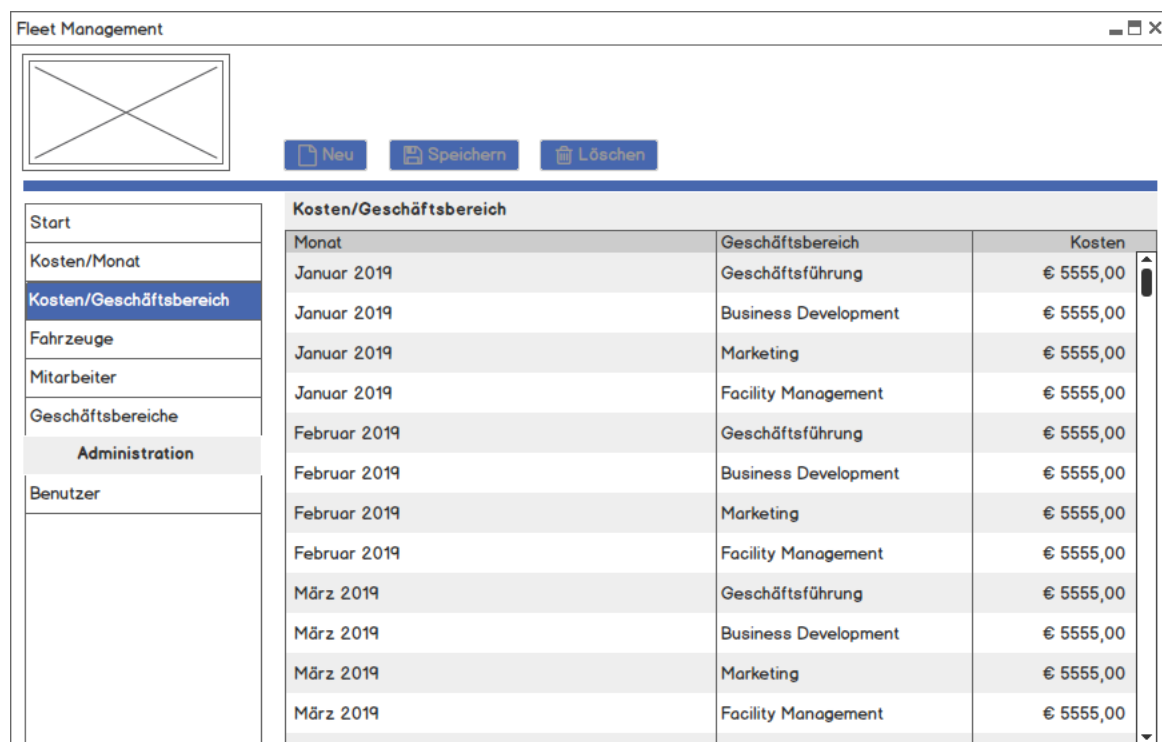
Folgende Spalten sollen dargestellt werden:

- Monat (Monat und Jahr, z. B. „Januar 2019“)
- Geschäftsbereich (Name des Geschäftsbereichs)
- Kosten (Summe der Kosten, die für alle Fahrzeuge des Geschäftsbereichs in diesem Monat entstehen (Leasingrate + anteilige Versicherungssumme))

Tabellen: Vehicles, VehicleToEmployeeRelation, Employees, BusinessUnits

Die Liste ist nach Monaten und Geschäftsbereichen sortiert (neuester steht unten, Geschäftsbereich alphabetisch). Die Zeilen der Liste werden über die bestehenden Leasingverträge ermittelt (vom Monat, in dem der „älteste“ Leasingvertrag beginnt, bis zum Monat, in dem der „neueste“ Leasingvertrag endet).

Für jeden Monat und Geschäftsbereich gibt es eine Zeile (z. B. für den Monat „Januar 2019“ gäbe es so viele Zeilen, wie es „betroffene“ Geschäftsbereiche gibt).



Kosten/Geschäftsbereich		
Monat	Geschäftsbereich	Kosten
Januar 2019	Geschäftsführung	€ 5555,00
Januar 2019	Business Development	€ 5555,00
Januar 2019	Marketing	€ 5555,00
Januar 2019	Facility Management	€ 5555,00
Februar 2019	Geschäftsführung	€ 5555,00
Februar 2019	Business Development	€ 5555,00
Februar 2019	Marketing	€ 5555,00
Februar 2019	Facility Management	€ 5555,00
März 2019	Geschäftsführung	€ 5555,00
März 2019	Business Development	€ 5555,00
März 2019	Marketing	€ 5555,00
März 2019	Facility Management	€ 5555,00

Verwenden Sie zusätzlich zur nHibernate-Funktionalität LINQ, um die notwendige Abfrage zu implementieren.

Schreiben Sie 3 relevante Unit-Tests für diese Funktionalität.

Allgemeine Tipps zur Lösung der Projektaufgabe:

Lesen Sie zuerst alle Aufgaben durch und auch diese Liste mit Tipps. Sie sparen Zeit, wenn Sie zuerst alle Anforderungen sichten und bei der Implementierung der Aufgaben berücksichtigen. Dies ist besonders bei grundsätzlichen Entscheidungen wichtig!

Gehen Sie „datengetrieben“ vor.

Legen Sie für die notwendigen Daten das Datenmodell in Klassenform fest.

Stellen Sie danach die notwendigen Grundfunktionalitäten Richtung Datenbank sicher, bevor Sie die Anwendung als solche implementieren (z.B. Laden, Speichern, Aktualisieren und Löschen der entsprechenden Daten).

Verschönerungen an der Oberfläche sollten Sie erst zum Schluss umsetzen.

Hängen Sie bei der Implementierung einer Teilfunktionalität, sollten Sie sich frühzeitig für die Implementierung der anderen Teilaufgaben entscheiden.

Haben Sie später noch genug Zeit, können Sie sich um die „Problemaufgabe“ kümmern.

Programmabstürze führen zu Punktabzug!

Rechnen Sie immer damit, dass Sie Fehleingaben erhalten. Prüfen Sie diese ab.

In der Realität kann es durchaus vorkommen, dass Sie fehlerhafte Daten für einen Datenimport erhalten.

Wenn Sie bei einigen geforderten Funktionalitäten nicht genau wissen, wie Sie diese implementieren können, steht es Ihnen frei, eine Suchmaschine Ihrer Wahl zu Rate zu ziehen. Hierbei hat sich die englische Sprache als besonders geeignet erwiesen.

Versuchen Sie die einzelnen Aufgaben vollständig zu lösen. Lösen Sie Aufgaben nicht nur zum Teil.

Vermeiden Sie Rechenübungen bzgl. ausreichender Punktzahl und der Anzahl bearbeiteter Aufgaben. Lösen Sie stattdessen die Aufgaben.