Kurzzeitparktikum Zühlke

Elia Kuster & Fabio Dauru

08.04. - 19.04.2024

# Inhaltsverzeichnis

[Inhaltsverzeichnis 1](#_Toc663109204)

[Projektauftrag 2](#_Toc13497105)

[Entwurf 4](#_Toc457554759)

[Organisatorisches 4](#_Toc612501440)

[Kontextdiagramm 4](#_Toc647509140)

[Use-Case-Diagramm 4](#_Toc1356479119)

[Design 5](#_Toc1412061755)

[Logisches Datenmodell 6](#_Toc649110248)

[Klassendiagramm 6](#_Toc1936659739)

[Arbeitsjournal 9](#_Toc360433082)

[Elia Kuster, 08.04.2024 9](#_Toc1065393715)

[Elia Kuster, 09.04.2024 9](#_Toc1539145555)

[Elia Kuster, 10.04.2024 9](#_Toc1681562884)

[Elia Kuster, 11.04.2024 9](#_Toc1505383256)

[Elia Kuster, 12.04.2024 10](#_Toc804157391)

[Elia Kuster, 15.04.2024 10](#_Toc143709360)

[Elia Kuster, 16.04.2024 10](#_Toc805397762)

[Elia Kuster, 17.04.2024 10](#_Toc401351380)

[Elia Kuster, 18.04.2024 11](#_Toc1390135837)

[Reflexion 11](#_Toc470613004)

[Was habe ich gelernt? 12](#_Toc224042329)

[Was habe ich über Zühlke erfahren? 12](#_Toc720207200)

[Was habe ich schwierig gefunden? 12](#_Toc1524611450)

# Projektauftrag

In den zwei Wochen bei Zühlke der Auftrag in Zweierteams ein Projekt umzusetzen. Fabio und ich schlossen uns zusammen und wir kriegten eine Aufgabenstellung. Das Ziel ist ein Car Tuning Configurator umzusetzen. Im Configurator sollte ein Benutzer ein Auto wählen können und dieses dann mit Tuningparts ausstatten können. Je nach Tuningteil und Auto sollen die Eigenschaften z.B. PS des Autos variieren. Mindestens die fertig modifizierten Autos sollen in eine Datenbank gespeichert werden sollen. Von der Technologieauswahl sind wir frei, es wurde uns jedoch empfohlen WPF und C# mit der MongoDB zu verwenden.

Unten ist ein schon vorhandenes Beispiel aufgezeigt.



# Entwurf

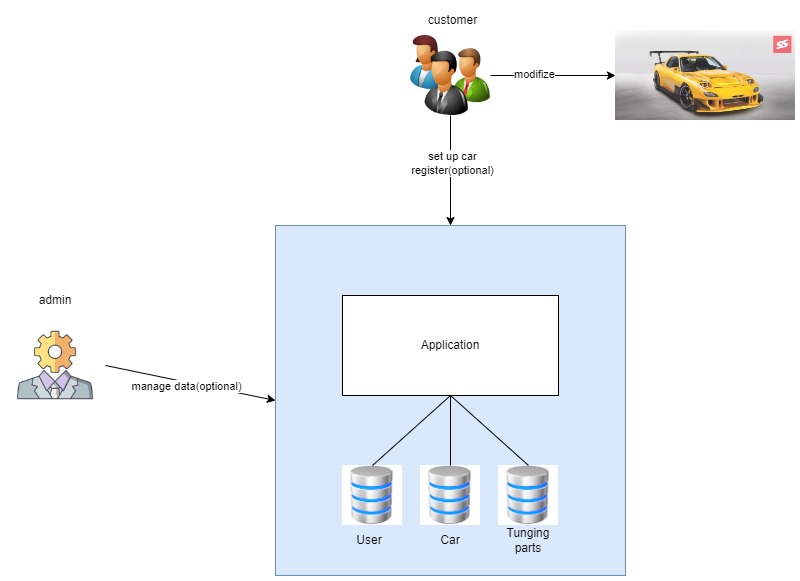
Als erstes müssen wir einen groben Überblick über die Aufgabenstellung bekommen und uns überlegen mit welchen Technologien, mit welcher Arbeitsaufteilung, mit welchem Programmumfangs und mit welchem Speicherumfangs wir das Projekt durchführen wollen.

## Organisatorisches

Wir haben beschlossen die vorgeschlagenen Technologien WPF, C# und MongoDB zu verwenden. Da wir entschieden haben auch ein Login-Fenster und ein Administrator-Fenster umzusetzen, brauchen wir eine effiziente Arbeitsaufteilung. Wir haben uns entschieden, dass Fabio die Datenbankconnection macht und ich das User Interface. Am Mittwoch haben wir unsere Aufteilung geändert und ich werde an der Datenbankconnection und an den Controller arbeiten und Fabio wird das User Interface der eigentlichen Seite erstellen.

## Kontextdiagramm

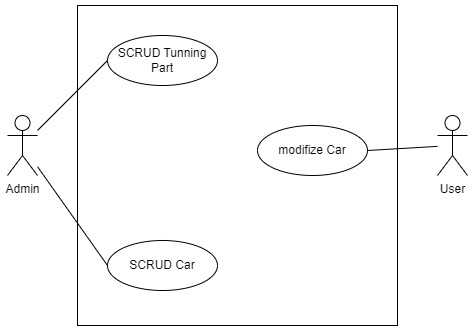
Als erstens arbeiten wir das Konzeptdiagramm miteinander aus, um die groben Zusammenhänge zwischen den User, der Datenbank und der Applikation zu verstehen. Zusätzlich leisten können wir dieses an einen Kunden fürs Verständnis zeigen. In unserem Fall haben wir jedoch keinen.



## Use-Case-Diagramm

Um das User Interface design zu können, müssen wir herausfinden, welche Entscheidungen der User treffen kann/soll/muss, damit das Programm mit seinen Anforderungen umgesetzt werden kann. Zusätzlich können wir mit Hilfe des Use-Case-Diagramm etwa einordnen, welche Funktionen wir in den Controllern und im DB-Access brauchen werden.

Beim unterem Use-Case-Diagramm ist die Autorisierung vorausgesetzt.

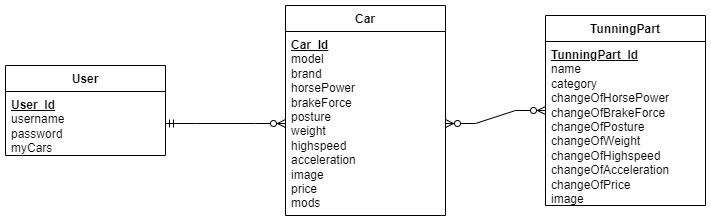


# Design

Als nächster Schritt müssen wir ein konkretes Design ausarbeiten, auf welches wir uns während dem Implementieren referenzieren können. Es geht jetzt spezifisch um die verschiedenen zu speichernden Daten und um die Ausarbeitung der Projektarchitektur. Mit der Projektarchitektur können wir das Projekt Schritt für Schritt aufbauen, sodass eine gewisse Übersicht besteht.

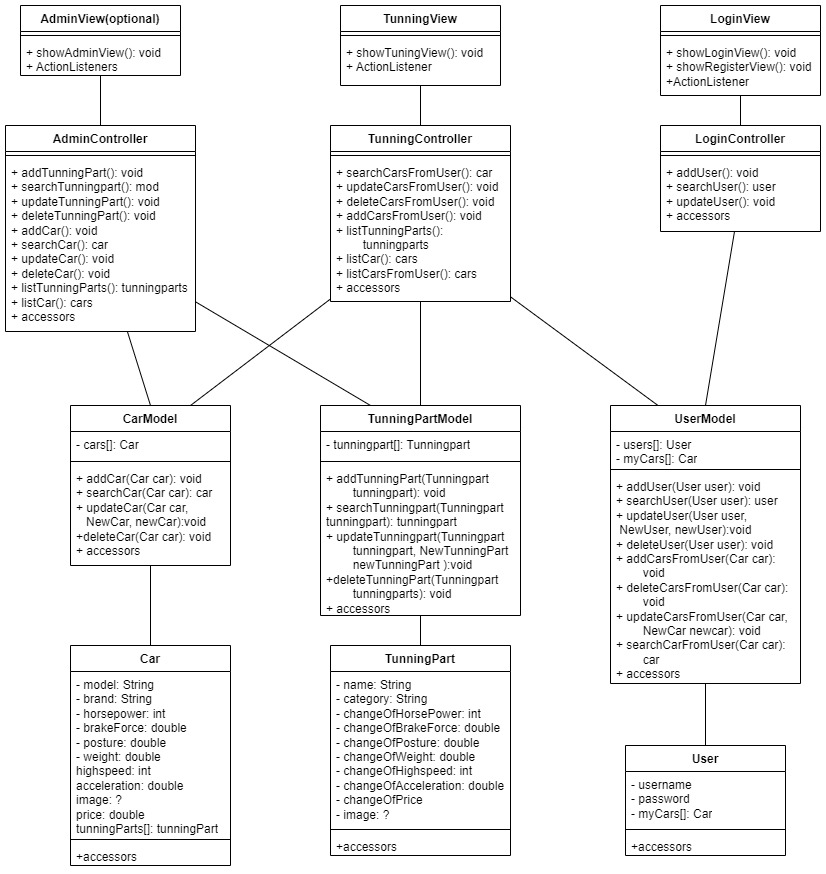
## Logisches Datenmodell

Um die Frage zu klären, welche Daten in der Datenbank gespeichert werden müssen und wie die Tabellen und Zeilen aussehen haben wir ein logisches Datenmodell erstellt. Dieses wird uns später bei der Erstellung des DB-Access behilflich sein und auch vorgeben, welche Attribute die verschiedenen Rohklassen haben.



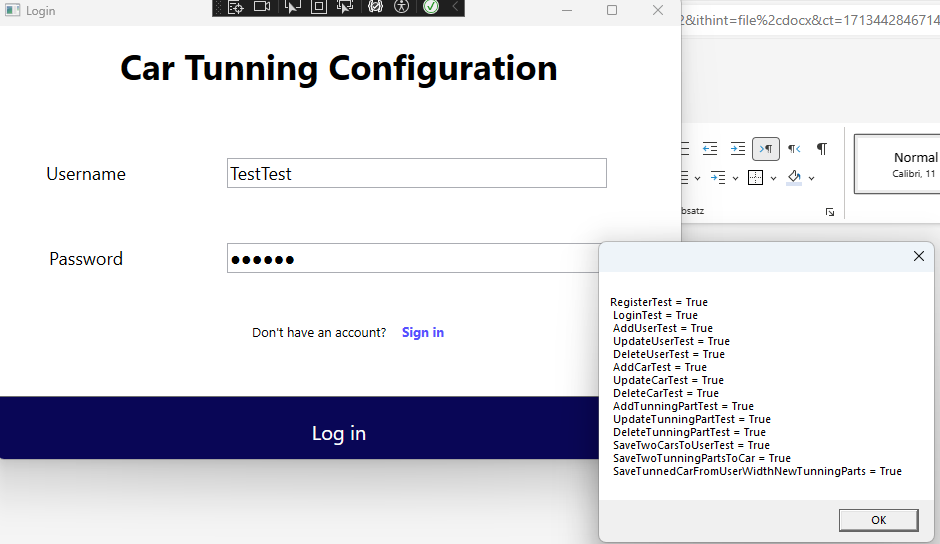
## Klassendiagramm

Das Klassendiagramm ist für mich das wichtigste Diagramm. Es zeigt auf welche Klassen es gibt, welche Funktionen und Attribute diese haben und wie sie miteinander verbunden sind. Mit diesem Diagramm können wir uns später orientieren und so entstehen keine logischen Fehler auf Klassenebene. Das Ausarbeiten des Klassendiagramm braucht viel Zeit und meist kann man im Vorhinein nicht alle einzelnen Funktionen kennen, da verschiedenste Probleme auftreten können. So ist auch unser Klassendiagramm nicht mehr aktuell und hat einzelne Funktionen und Attribute nicht enthalten.



Test

Die Tests der Applikation wurden automatisiert und können im Verzeichnis TestFile gefunden werden. Das Test.cs enthält Funktionen für jede Funktion aus den drei Controller, dem HomePageController, dem AdminController und dem LoginController. Der Test kann im Login im Feld für den Username aktivierte werden, indem man den Text “TestTest” eingibt. Nach der Eingabe muss man normal auf den Loginbutton klicken. Danach wird eine Messagebox gezeigt, auf welcher die verschiedenen Tests, welche durchgeführt wurden, aufgelistet und das Resultat des Tests wird mit True oder False angezeigt. Wenn das Resultat True ist, wurde der Test erfolgreich durchgelaufen und es müssen keine Anpassungen gemacht werden. Falls der Test False ist, müssen dringend Anpassungen gemacht werden, da die Funktionen nicht mehr so funktionieren, wie sie sollen. Für die Behandlung der Fehler muss man dann nur auf die jeweilig betroffenen Funktionen eingegangen werden.



# Arbeitsjournal

## Elia Kuster, 08.04.2024

Heute sind wir in Zühlke angekommen und Mario hat uns erst einmal die Firma gezeigt, uns auf wichtige Dinge aufmerksam gemacht, uns abgeschlossene Projekte von Zühlke gezeigt und uns die 4. Jahres Praktikanten vorgestellt, welche unsere Ansprechpersonen im Praktikum sind. Danach sind wir ins Daily gegangen, wo die Praktikanten besprechen, was sie am letzten Tag gemacht haben und was sie heute machen werden. Nach einer kurzen Pause teilte uns Mario die Aufgabe für die nächsten zwei Wochen mit und wir haben mit der Teamaufteilung und dem Organisieren in den Teams begonnen. Um 11.45 Uhr sind wir im Personalrestaurant Mittagessen gegangen. Am Nachmittag konnten wir mit dem Projekt starten und Fabio und ich haben erstmals wichtige Entscheide getroffen und uns organisiert. Um 16.30 Uhr schauten wir mit Mario an, was die verschiedensten Teams an diesem Tag gemacht haben. Danach sind wir nachhause gegangen.

## Elia Kuster, 09.04.2024

Als wir heute am Morgen gekommen sind, konnten wir gerade mit dem Arbeiten anfangen. Ich habe zuerst die Diagramme vom Vortag überarbeitet und gestartet die Rohklassen zu erstellen. Das Klassendiagramm war zu unübersichtlich und ich habe nur einen Controller eingeplant, was bei drei Views sehr unübersichtlich wird. Zusätzlich habe ich das Use-Case-Diagramm viel simpler gestaltet. Danach erstellte ich die Klassen User.cs, Car.cs TunningPart.cs, UserModel.cs, CarModel.cs und TunningPartModel.cs. Zwischendurch gab es noch Mittagessen, trotzdem hatte ich am Nachmittag noch Zeit das Login- und Registrierfenster zu gestalten und vorübergehend mit Hardcode auszuprobieren. Um 16.00 Uhr hatten wir wieder die Besprechung und um 16.40 Uhr bin ich nach Hause gefahren.

## Elia Kuster, 10.04.2024

Am Morgen früh sind Fabio und ich zusammengehockt, um über die neue Arbeitsaufteilung zu diskutieren. Fabio hat sich Gedanken über die Home-Seite gemacht und möchte diese umsetzen. So haben wir uns geeinigt, da ich neu den Codebehind übernehme und er das User Interface gestaltet. An diesem ganzen Tag habe ich im DB-Access Funktionen geschrieben. Vor und nach dem Mittag war ich daran, die CRUD-Methoden der Cars, Users und TunningParts zu schreiben. Ich hatte mit vielen Fehlern zu kämpfen und kam nur schleppend voran. Vor allem mit dem Speichern von der TunningPart Liste im Car hatte ich Probleme. Um 16.00 Uhr haben wir alles wieder mit Mario besprochen und es stellte sich heraus, dass ich fürs Speichern in die MongoDB einfach einen Automapper gebrauchen kann.

## Elia Kuster, 11.04.2024

Mario hat uns heute morgen das Ganze um den Automapper gezeigt und wie man ihn anwendet. Ich habe dann den ganzen DB-Access umgeschrieben, damit das mit dem Automapper funktioniert. Der Automapper ist recht simpel und ich bin schnell fertig geworden. Um 09.00 Uhr sind wir zum Helpdesk gegangen und konnten dann zwei Stunden lang in den Arbeitsalltag eines IT-Spezialisten hineinschauen. Sie zeigten uns die Tools, die sie verwenden und auch die verschiedensten Räume, in denen sie arbeiten, oder bei IT-Probleme auch helfen müssen. Ein bisschen vor dem Mittag und nach dem Mittag habe ich das Login- und Registrierfenster mit dem LoginController.cs und dem DB-Access verknüpft. Hier hat es nur kleine Probleme gegeben und ich bin so schnell vorwärtsgekommen, dass ich es noch geschafft habe, die Übersichtsseite für den Admin zu gestalten, wo er Cars, Users und TunningParts adden, updaten und deleten kann. Heute haben wir keine Sitzung gemacht, sondern sind am 16.45 Uhr gerade nach Hause gegangen.

## Elia Kuster, 12.04.2024

Heute habe ich wahrscheinlich am meisten in dieser Woche geschafft. Vom Morgen bis zum späten Nachmittag habe ich den Controller für den Administrator erstellt und mit dem DB-Access verbunden. Die ganzen CRUD-Operationen funktionierten am Ende des Tages, auch wenn das Suchen und vor allem das Updaten mit der Filter-Funktion sehr schwierig gewesen ist und auch viel Zeit in Anspruch genommen hat. Ich habe auch mein Car.cs nochmals updaten müssen und mit Fabio das neue Design von der Main-Seite angeschaut. Am Nachmittag vor dem Tagesabschluss und der Verabschiedung von Mario, da er nächste Woche in den Ferien ist, habe ich es noch geschafft, dass das Password gehasht gespeichert wird und das Login weiterhin funktioniert. Um 16.50 Uhr sind wir dann gegangen.

## Elia Kuster, 15.04.2024

An jenem Morgen habe ich daran gearbeitet, dass die CRUD-Operatoren im AdminConroller.cs auf der DetailAdminSites aufgerufen werden können und auch funktionieren. Dafür musste ich neun neue Windows erstellen und diese zum Controller verbinden. Es hat sich herausgestellt, dass einige doch nicht ganz funktionieren, und ich fühle mich gezwungen, diese zu beheben. Das habe ich dann auch gemacht und es funktioniert alles. Ich habe noch ein paar kleine Fehler behoben und die eine oder andere Kleinigkeit verbessert. Am Abend bin ich etwas früher gegangen, da ich noch eine Sitzung habe.

## Elia Kuster, 16.04.2024

An diesem Tag, als ich angekommen bin, habe ich nicht gewusst, was ich machen soll. Nach einer kurzen Besprechung mit Fabio, habe ich mich entschieden, dass ich die Dokumentation zusammenstelle. Ich habe die Arbeitsjournale gesammelt und jeweils kurze Texte zu den verschiedenen Etappen geschrieben. Nach dem Mittag habe ich noch von Fabio mitgeteilt bekommen, dass noch einige Bucks zu fixen sind. So machte ich Funktionen für die Save-Buttons und des Panels nach dem Updaten der TunningParts. Nach diesem Task habe ich Fabio geholfen, die Radio-Buttons zum Funktionieren zu bringen, sodass Objekte den Radio-Buttons zugewiesen werden und auch wieder ausgelesen werden können.

## Elia Kuster, 17.04.2024

Heute morgen habe ich mir zum Ziel gesetzt, dass wenn man die TunningParts nacheinander anwählt, die Werte sich richtig zusammenzählen lassen und sich nicht addieren, ohne den nicht mehr ausgewählten TunningPart zurückzusetzen. Ich bin recht lange daran gewesen und habe vor allem die folgenden zwei Probleme gehabt. Ich musste schauen, dass sich von jedem TunningPart-Kategorie nur ein Teil zuteilen lässt und sich mit den TunningParts von den anderen Kategorien addieren lässt. Das etwas schwerere Problem ist, dass wenn man zwischen den TunningParts in einer Kategorie wechselt, dass die Werte immer wieder addiert wurden. Mit ein paar If-Else-Schleifen konnte ich die zwei Probleme am späten Nachmittag lösen. Danach ist noch ein Problem bei der Übergabe des aktuellen Cars im TunnedCarPanel nicht bzw. nur mit nullübergeben. Das Problem habe ich schnell behoben und ich bin nach Hause gegangen.

## Elia Kuster, 18.04.2024

Als ich gekommen bin, habe ich erstmals das ganze Programm zweimal durchgespielt und Fehler bzw. Verbesserungen aufgeschrieben. Diese sind vor allem im User Interface entstanden und ich habe Fabio gebeten diese zu lösen. Danach habe ich mich dem Testen gewidmet. Ich habe lange überlegt und mich gefragt was ich wie testen soll. Ich habe mich entschieden, ein TestFile zu schreiben, auf welchem man alle wichtigen Funktionen in den jeweiligen Controller testen kann. Das File kann durch die Eingabe “TestTest” im Username Feld aufgerufen werden. Es ist auch für die spätere Entwicklung des Projektes nützlich, wenn man die Tests ausführen kann, um dann zu merken, ob alles im Controller funktioniert und dann der Fehler im User Interface entstanden ist, oder ob eine bestimmte Funktion in einem Controller nicht funktioniert. Man kann so seine Fehlerbehebung spezifizieren. Nachdem ich das TestFile geschrieben habe, erweiterte ich meine Dokumentation und bin dann gegangen.

# Reflexion

Heute, am letzten Tag, reflektiere ich nochmals ganz kurz über das Kurzzeitpraktikum bei Zühlke vom 08.04. Bis am 19.04.2024. Ich werden spezifisch auf die folgenden Punkte eingehen:  
Was habe ich gelernt?  
Was habe ich über Zühlke erfahren?  
Was habe ich schwierig gefunden?

## Was habe ich gelernt?

Da wir das Projekt in C# machen und ich etwas ausser Übung bin im C# habe ich nicht nur C# repetiert, ich habe es auch vertiefen können und beherrsche jetzt schliesslich C# fast so gut wie Java. Auch WPF habe ich etwas vertiefen können. Jedoch ist das, was ich am meisten gelernt habe, das Arbeiten im Team. Von der Ausarbeitung und der Umfangsbestimmung, bis zur Umsetzung bin ich immer im direkten Austausch mit Fabio gewesen und bei Schwierigkeiten und Besprechungen sind wir koordinativ auf einem sehr hohen Level gewesen.

## Was habe ich über Zühlke erfahren?

Zühlke ist eine sehr grosses und in vielen Bereichen tätiges Unternehmen. Von der Ausarbeitung über die mechanische Anfertigung zur schliesslichen Softwareapplikation fertigt Zühlke ihre von Kunden beauftragten Projekte. Zühlke hat in der Vergangenheit schon sehr viele sehr komplexe Projekte umgesetzt. Wie zum Beispiel eine Maschine, welche künstlich Haut herstellen kann, oder einen Sensor, welche den Strassenzustand messen und anschliessend auswerten kann. Vom Gebäude her ist Zühlke sehr speziell eingerichtet. Sie haben einen sehr offenen und ansprechenden Eingangsbereich, ein Personalrestaurant mit sehr gutem Essen und sehr schöne, offene und mit Pflanzen dekorierte Arbeitsplätze. Die Menschen, welche bei Zühlke arbeiten, sind sehr höflich und geben auch gerne Auskunft über ihre Arbeit. Sie helfen gerne und kommen, so wie ich es aufgefasst habe, auch sehr gerne zur Arbeit. Da die Firma international tätig ist, ist die Umgangssprache mit den einten Leuten Englisch. Auch beim Helpdesk ist Englisch gesprochen worden, was mich doch auch etwas überrascht hat.

## Was habe ich schwierig gefunden?

Das Übergeben von Objekten ins User Interface habe ich sehr anstrengend gefunden. Weil die Objekte meistens in Lists von anderen Objekten verschachtelt sind, muss man diese immer ganz aufwendig holen, um dann dessen Attribute im User Interface aufzuzeigen. Es sind immer wieder null-Objekte übergeben worden, was ich immer wieder gefixt habe, aber es doch wieder nach einer kleinen Anpassung nicht mehr funktioniert hat.