Pflichtenheft

**Aufgabenstellung:** Fahrzeuge und ihre Besitzer verwalten

**Gruppennummer:** 1

**Gruppenmitglieder:** Melvin Böke,

Thanh Binh Nguyen,

David Leinweber

Inhalt

[Aufgabenstellung 3](#_Toc17894514)

[Meilensteine 3](#_Toc17894515)

[Team- und Arbeitsaufteilung 3](#_Toc17894516)

[Mockup 4](#_Toc17894517)

[GUI 4](#_Toc17894518)

[TUI 5](#_Toc17894519)

[Testszenarien 6](#_Toc17894520)

[Datenverwaltung 6](#_Toc17894521)

# Aufgabenstellung

Es soll eine Fahrzeug- und Besitzerverwaltung mit Hilfe einer 3-Schichten Architektur erstellt werden.

# Meilensteine

Zur Erledigung der gesamten Komplexität des Projekts wurden simplere Zwischenziele erstellt. Diese Meilensteine umfassen den Projektzeitraum vom 27.08.2019 bis 08.01.2020.

|  |  |  |  |
| --- | --- | --- | --- |
| Meilenstein | Beschreibung | Zwischenprodukt | Datum |
| MS-1 | Anforderungsanalyse,  Anforderungsdefinition | Erste Dokumentation („Pflichtenheft“) | 28.08.2019 |
| MS-2 | Design, Planung der Schnittstellen,  Implementierung | 1. Definierte Interfaces vom Fachkonzept und Datenhaltung 2. Implementierung der TUI und GUI | 20.09.2019 |
| MS-3 | Implementierung | Implementierung von Datenhaltung mit SQLite und JSON Textdatei | 25.10.2019 |
| MS-4 | Recherche, Informieren, Implementation | Implementierung der Fachkonzepte | 15.11.2019 |
| MS-5 | Testen, Fehlerkorrektion | Fehlerfreie Software | 06.12.2019 |
| MS-6 | Dokumentation | Vollständige Dokumentation | 08.01.2020 |

# Team- und Arbeitsaufteilung

Um das Projekt effizient anzugehen wird die Arbeit auf die drei Mitglieder verteilt.

Beim ersten Meilenstein werden die Interfaces in Zusammenarbeit aller drei Mitglieder ausdefiniert, sodass alle wichtigen Methoden besprochen werden, die von den Interfaces implementiert werden müssen.  
Für die UI Implementierung wird Herr Böke an der TUI Implementierung arbeiten während Herr Nguyen sich mit der GUI Implementierung befasst. Herr Leinweber wird zwischen TUI und GUI abwechselnd mit Herrn Nguyen und Herrn Böke im Pair-Programming arbeiten.

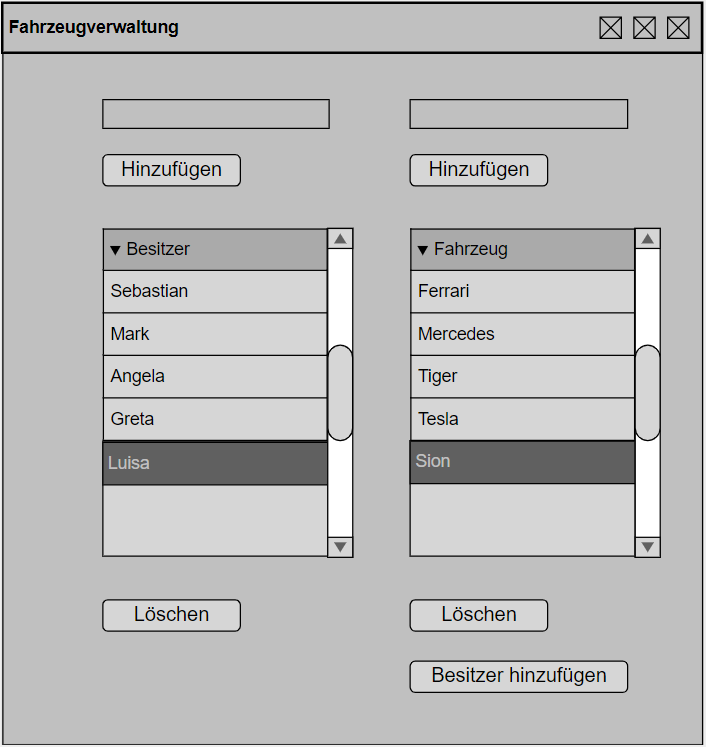
Daraufhin wird die Datenhaltung implementiert mit Fokus auf die SQLite Implementation, da diese wegen der selbstzuschreibenden SQL Befehle eine höhere Komplexität besitzt werden die Mitglieder Nguyen und Leinweber primär an diesem arbeiten. Herr Böke arbeitet dann an der Datenhaltung mit JSON.

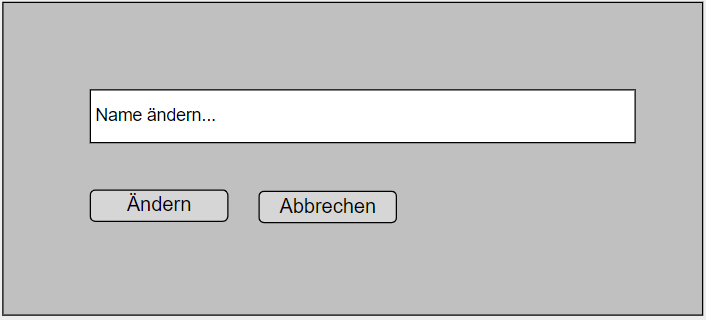
Für den Meilenstein vier muss Recherche zum Strategy-Pattern gemacht werden, um die Fachkonzepte zu entwickeln. Die Recherche wird von allen drei Mitgliedern einzeln gemacht und zusammengetragen.  
Mit dem neu erlangten wissen werden die Fachkonzepte nach der Strategie von der UI implementiert. Herr Nguyen und Herr Böke arbeiten an Fachkonzept eins und Fachkonzept zwei respektive und Herr Leinweber wechselt zwischen den Fachkonzepten.

Das letzte Meilenstein besteht aus Testen der Software und der Fehlerbehebung. Hier werden alle Mitglieder parallel testen und alle Fehler beheben.

# Mockup

## GUI



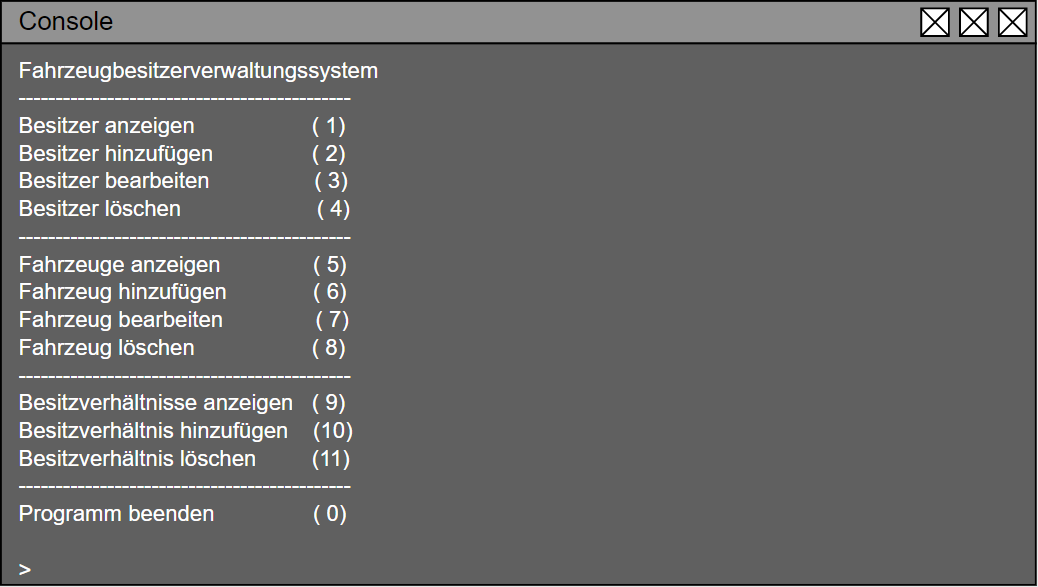


Wenn der Besitzer angeklickt wird, dann werden die ihm zugehörigen Fahrzeuge angezeigt. Zusätzlich werden alle Fahrzeuge ohne Besitzer angezeigt, welche eine etwas unauffälligere Farbe haben, sodass sich die Fahrzeuge, die dem Besitzer gehören und die Fahrzeuge ohne Besitzer unterscheiden.  
Sollte ein Fahrzeug ausgewählt werden, welches dem Besitzer gehört, so ändert sich der Text „Besitzer hinzufügen“ unter der Fahrzeug Selectbox auf „Besitzer entfernen“.

Beim Doppelklick auf einen Besitzer soll ein Popup erscheinen, in dem der Name des Besitzers geändert werden kann.

Bei einem Doppelklick das Fahrzeug soll ein ähnlicher Popup erscheinen, in dem die Fahrzeugbezeichnung geändert werden kann.

## TUI



# Testszenarien

|  |  |
| --- | --- |
| Test | Erwartung |
| Besitzer mit dem Namen „Max“ erstellen. | Es soll ein neuer Besitzer vorhanden sein, der den Namen „Max“ hat. Er soll noch kein Fahrzeug besitzen |
| Besitzer löschen. | Der ausgewählte Besitzer soll gelöscht werden. Alle Fahrzeuge, die der Besitzer besaß, sollen nun keinen Besitzer mehr haben.  Der Besitzer soll nicht mehr angezeigt werden. |
| Namen des Besitzers ändern. | Der ausgewählte Besitzer soll einen neuen Namen bekommen. Alle Stellen, an denen der Name referenziert wird, sollen den neuen Namen anzeigen. |
| Alle Besitzer zeigen. | Es sollen alle zuvor erstellten und nicht bereits gelöschten Besitzer angezeigt werden. |
| Fahrzeug mit dem Typen „VW Golf“ erstellen. | Es soll ein neues Fahrzeug mit einer neuen ID vorhanden sein, der den Typen „VW Golf“ hat.  Das Fahrzeug soll noch keinen Besitzer haben. |
| Fahrzeug löschen. | Das ausgewählte Fahrzeug soll gelöscht werden. Der Besitzer des Fahrzeugs soll dieses Fahrzeug nicht mehr besitzen.  Das Fahrzeug soll nicht mehr angezeigt werden. |
| Typ des Fahrzeugs ändern. | Das ausgewählte Fahrzeug soll einen neuen Typen bekommen. Alle stellen, an denen der Typ referenziert wird, sollen den neuen Typen anzeigen. |
| Alle Fahrzeuge anzeigen. | Es sollen alle zuvor erstellten und nicht bereits gelöschten Fahrzeuge angezeigt werden. |
| Alle Fahrzeuge pro Benutzer anzeigen. | Alle Fahrzeuge anzeigen, die zum ausgewählten Besitzer gehören. |
| Den Besitzer des Fahrzeugs anzeigen. | Den Besitzer des ausgewählten Fahrzeugs anzeigen. |
| Dem Fahrzeug einen Besitzer geben. | Das ausgewählte Fahrzeug soll dem ausgewählten Benutzer gehören. Wenn das Fahrzeug bereits einen Besitzer hat soll das Besitztum automatisch gelöscht werden. |
| Dem Besitzer des Fahrzeugs entfernen. | Das ausgewählte Fahrzeug soll keinen Besitzer mehr haben. Der Besitzer soll dennoch seine übrigen Fahrzeuge behalten. |

# Datenverwaltung

Die Pflichtdatenbank implementieren wir mit SQLite, weil es keinen separaten Datenbank Server benötigt. Für die Wahldatenhaltung haben wir uns für JSON entschieden.

Unser ER-Diagramm sieht wie folgt aus:

