



PROJEKTDOKUMENTATION

Samira Bingesser, Gani Nurceski, Juan Gutierrez

Inhaltsverzeichnis

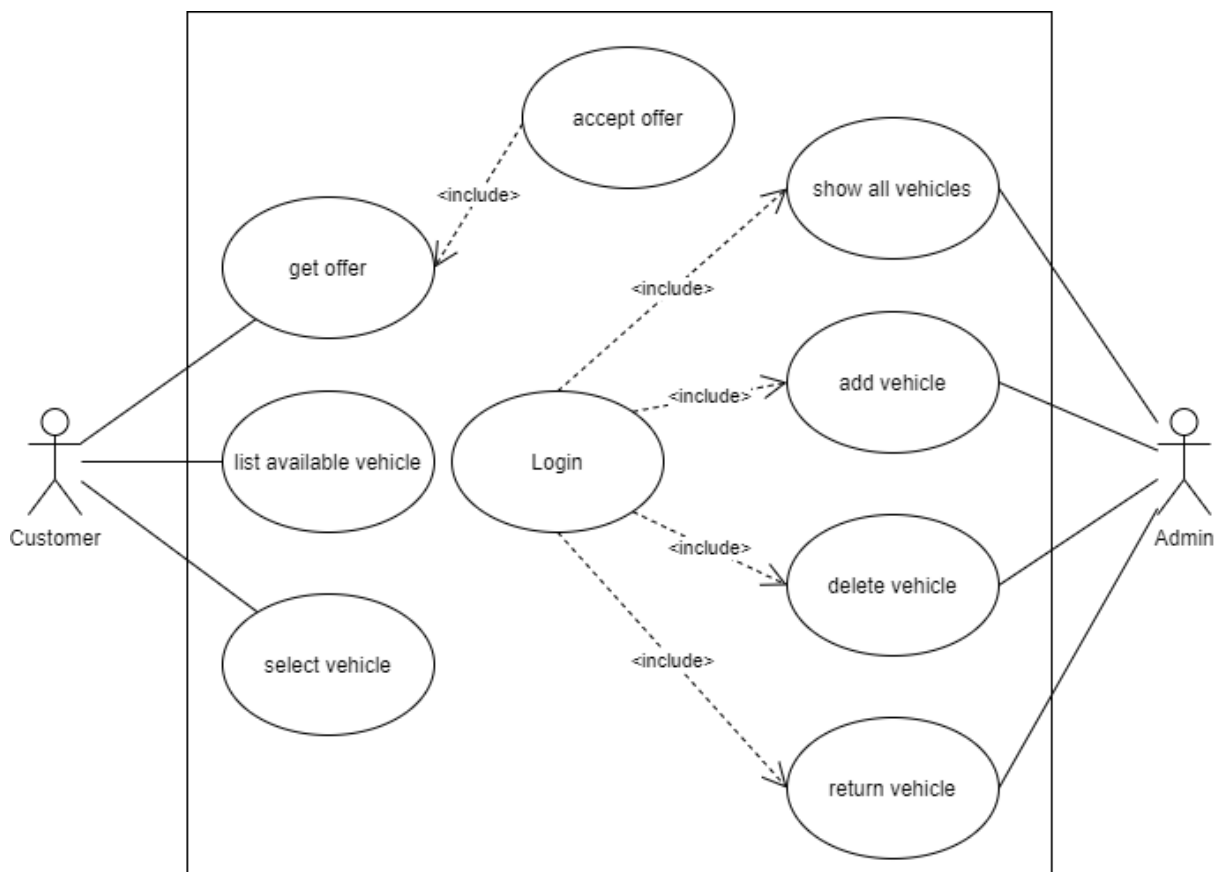
Projektbeschreibung	2
Use-Case Diagramm	2
Klassendiagramm	3
Aktivitätsdiagramm	4
Objektorientierte Konzepte	5
Vererbung	5
Abstraktion.....	5
Git Repository	5
Reflexion	6

Projektbeschreibung

In diesem Projekt wird eine C#-Anwendung für die Verwaltung von Fahrzeugvermietungen erstellt. Der Administrator kann Fahrzeuge hinzufügen, zurücknehmen und löschen. Benutzer können ohne Anmeldung auf der Startseite ein Fahrzeug auswählen und eine Offerte erhalten und diese akzeptieren oder ablehnen. Mithilfe einer Datenbank im Hintergrund wird der Status und die Informationen aller Fahrzeuge erfasst.

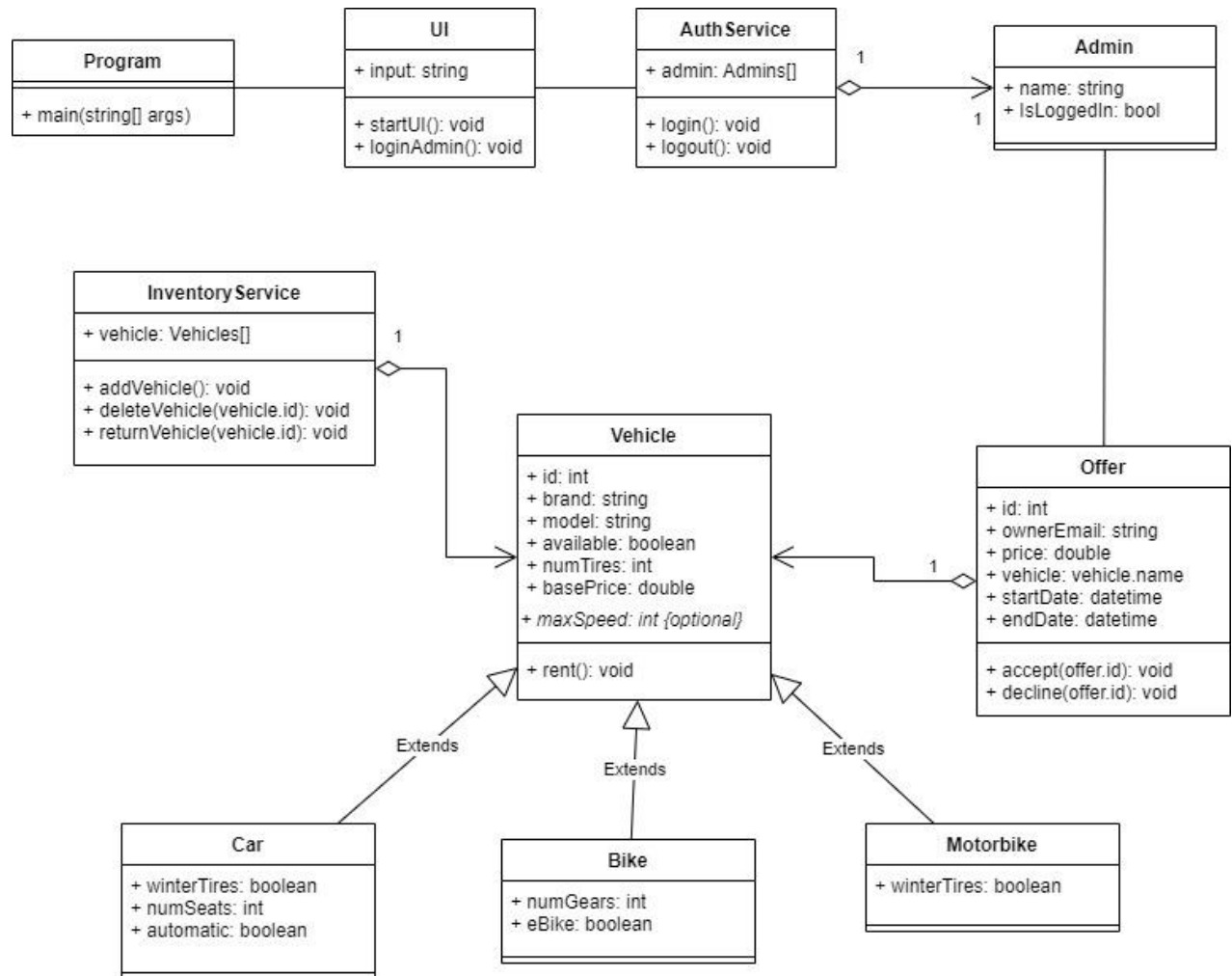
Use-Case Diagramm

Der Kunde kann ohne Registrierung ein verfügbares Fahrzeug reservieren. Dazu erhält er eine Offerte, welche der Admin akzeptieren oder ablehnen kann. Der Admin kann alle Fahrzeuge angezeigt bekommen, weitere Fahrzeuge hinzufügen und entfernen und auch Fahrzeuge, welche nicht verfügbar sind, erneut zur Verfügung stellen. Der Admin muss sich jedoch vorher anmelden.

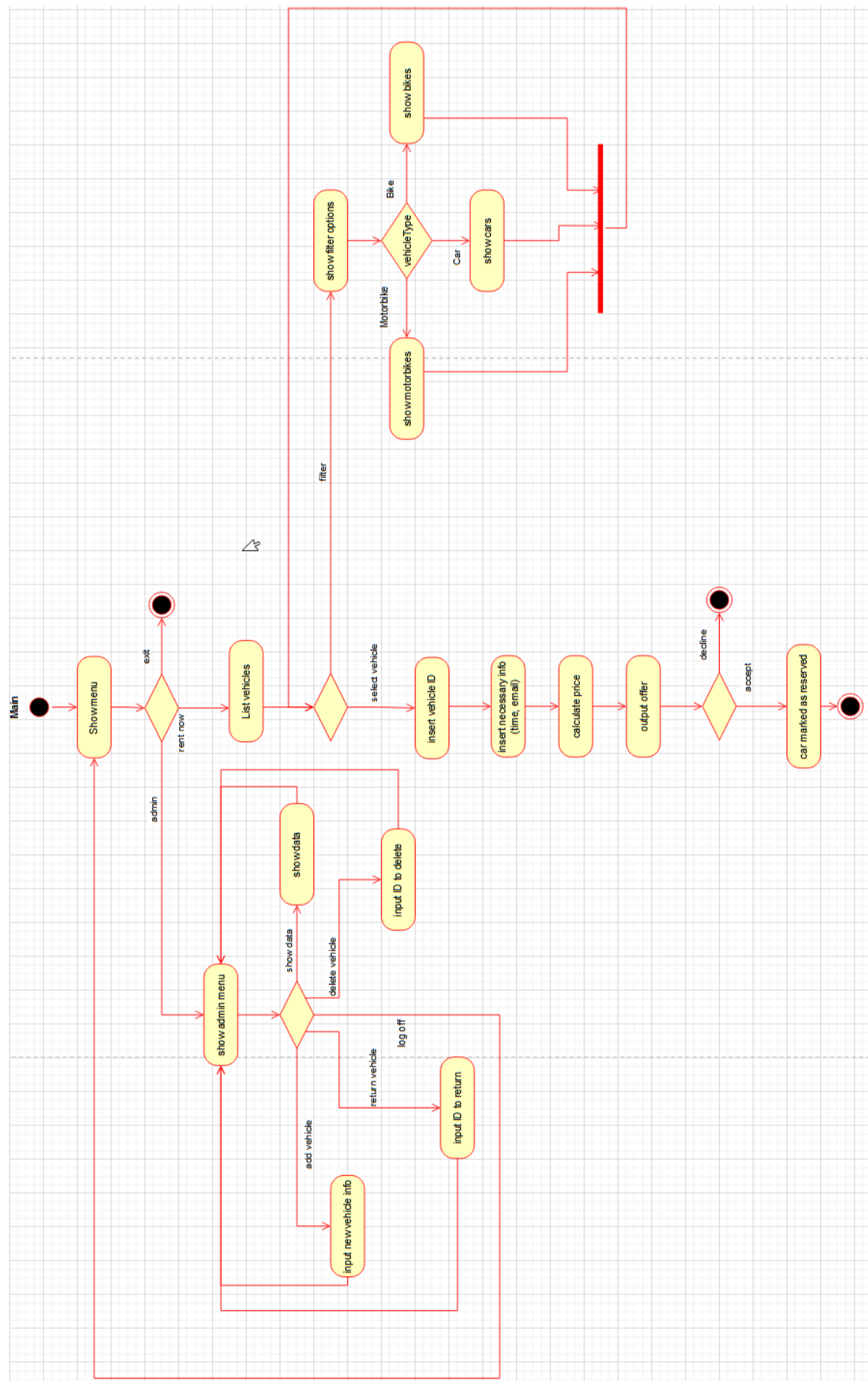


Klassendiagramm

Hier ist unser Klassendiagramm vor der Implementation zu sehen:



Aktivitätsdiagramm



Objektorientierte Konzepte

Vererbung

Wie in unserem Klassendiagramm gut zu erkennen ist, sind die Klassen Car, Bike und Motorbike Unterklassen der Oberklasse Vehicle. Sie erben also alle Attribute eines Fahrzeuges.

```
public class Car : Vehicle
{
    public Boolean winterTires;
    public int numSeats;
    public Boolean automatic;
}

public class Bike : Vehicle
{
    public int numGears;
    public Boolean eBike;
}

public class Motorbike : Vehicle
{
    public Boolean winterTires;
```

Abstraktion

Da die maximale Geschwindigkeit bei jedem der Fahrzeuge unterschiedlich ist, ist das Attribut maxSpeed abstrakt.

```
public class Vehicle
{
    public int vehicleId;
    public string brand;
    public string model;
    public Boolean available;
    public int numTires;
    public double basePrice;
    public abstract int maxSpeed;

    public void rent()
    {
    }
}
```

Git Repository

Um gemeinsam am Projekt arbeiten zu können, haben wir ein Repository auf GitHub erstellt und unsere Fortschritte regelmässig hochgeladen.

Hier ist der Link zu unserem Repository: <https://github.com/JuanGut01/RentACar>

Reflexion

In dieser Projektarbeit haben wir uns erneut mit C# beschäftigt. Im Vergleich zum letzten Projekt haben wir uns nicht gut geschlagen. Am Schluss geben wir ein Programm mit einigen Bugs ab. So funktioniert zum Schluss die Menüführung nicht. Trotz mühsamer Arbeit, vor allem zu Beginn bei der Erstellung der Diagramme und der Ideenfindung, konnten wir unseren Anforderungen nicht gerecht werden. Nach der Projektplanung haben wir unsere eigenen Anforderungen in Frage gestellt und fanden schwierig diese umzusetzen. Zu Beginn wollten wir uns noch die Mühe machen eine Datenbank anzubinden, doch wir haben es nicht hinbekommen und haben uns erneut Gedanken darüber gemacht. Die Zusammenarbeit zwischen uns Teammitglieder funktionierte gut. Auf Git kamen wir besser zurecht als im letzten Projekt, da wir in der Zwischenzeit Git besser kennengelernt haben. Für die nächste Arbeit nehmen wir uns vor unsere Idee praktisch und simpel zu halten.