

# **DIPLOMARBEIT**

Gesamtprojekt

## **LPCS**

Entwicklung eines Carsharing Systems

Noah Laireiter

5CHEL

Betreuer:

Mag. Dr. Anton Hofmann

Thomas Ploder

5CHEL

Kooperationspartner: Freiraum Gneis mobil

ausgeführt im Schuljahr 2021/22

---

Abgabevermerk:

Datum: 04.04.2022

übernommen von:

## Eidesstattliche Erklärung

Ich erkläre an Eides statt, dass ich die vorliegende Diplomarbeit selbständig und ohne fremde Hilfe verfasst, keine anderen als die angegebenen Quellen und Hilfsmittel benutzt und die den benutzten Quellen wörtlich und inhaltlich entnommenen Stellen als solche kenntlich gemacht habe.

Salzburg, am 04.04.2022

Verfasserinnen / Verfasser:



Noah Laireiter



Thomas Ploder

## DIPLOMARBEIT DOKUMENTATION

Namen der Verfasserinnen / Verfasser	Noah Laireiter Thomas Ploder
Jahrgang Schuljahr	5CHEL 2021/22
Thema der Diplomarbeit	LPCS – Entwicklung eines Carsharing Systems
Kooperationspartner	Freiraum Gneis mobil

Aufgabenstellung	Das bestehende System erfüllt nicht die Ansprüche für einen stetig wachsenden Carsharing-Verein, da grundlegende Funktionen wie z.B. eine Automatisierung des Fahrtenbuchs nicht vorhanden sind. Ältere Menschen haben aktuell Probleme das System aufgrund seiner Komplexität zu bedienen. Ein sicheres Aufbewahren des Autoschlüssels ist aufgrund fehlender Sicherheitsinfrastrukturen nicht möglich.
------------------	--

Realisierung	Dieses Projekt verfolgt das Ziel, den Betriebsablauf durch Automatisierung, Digitalisierung und eine bessere Systemverwaltung zu optimieren. Dadurch wird die Effizienz gesteigert und die durch den Menschen verursachten Fehler können auf ein Minimum reduziert werden.
--------------	--

Ergebnisse	Das Ergebnis ist ein KFZ-Tracking-Modul, eine Keybox und die entsprechende Software für einen reibungslosen Buchungsablauf und dessen Verwaltung. Das Übertragungsmodul im Auto sendet regelmäßig fahrtrelevante Daten (Standort, gefahrene Kilometer) an das Server-Backend. Das Buchungssystem ist über eine Website und App erreichbar.
------------	--

Typische Grafik, Foto etc.  
(mit Erläuterung)



**LPCS  
CARSHARING**

Abbildung 1.1.1: LPCS Carsharing System Logo

Teilnahme an Wettbewerben,  
Auszeichnungen

—

Möglichkeiten der  
Einsichtnahme in die Arbeit

Bibliothek – HTBLuVA Salzburg

Approbation  
(Datum / Unterschrift)

Prüferin / Prüfer

Direktorin / Direktor  
Abteilungsvorständin / Abteilungsvorstand

	<b>COLLEGE OF ENGINEERING Salzburg</b>
	<b>Electronics and Computer Engineering</b>

## **DIPLOMA THESIS Documentation**

Author(s)	Noah Laireiter Thomas Ploder
Form Academic year	5CHEL 2021/22
Topic	LPCS – development of a carsharing system
Co-operation Partners	Freiraum Gneis mobil

Assignment of Tasks	The existing system does not meet the requirements for a steadily growing car sharing club, as basic functions such as an automated driver's logbook are not available. Elderly people currently have problems using the system due to its complexity. Furthermore, the secure storage of the car key is not guaranteed due to a lack of safety infrastructure.
---------------------	---

Realisation	This project aims to optimise the operational process using automation, digitalisation, and an enhanced system management. As a result, efficiency is increased and errors caused by humans can be significantly reduced to a minimum.
-------------	--

Results	The result is a vehicle tracking module, a key-box and the corresponding software for a smooth booking process and its administrative management. The in-car transmission module periodically sends trip-related data (location, kilometres driven) to the backend of the server. The booking system can be reached in the form of a website as well as an app.
---------	---

Illustrative Graph, Photo  
(incl. explanation)



**LPCS**  
**CARSHARING**

*Abbildung 1.1.2: LPCS Carsharing System logo*

Participation in Competitions  
Awards

—

Accessibility of  
Diploma Thesis

Library – Secondary Technical College Salzburg

Approval  
(Date / Sign)

Examiner

Head of College  
Head of Department

## Vorwort

In der vorliegenden Diplomarbeit beschäftigen wir uns mit der Entwicklung eines Carsharing Systems. Da es im Moment nur wenige Angebote für solche Systeme gibt und diese meist sehr teuer sind, wird eine kostengünstigere Alternative entwickelt. Diese soll es kleinen Vereinen ermöglichen, ihre Fahrzeuge zu verwalten und zu vermieten, ohne hohe Summen für die Software zu bezahlen. Gegliedert ist dieses Projekt in drei Teile: die Software, das Tracking-Modul und die Schlüsselbox. Diese haben alle voneinander unterschiedliche Aufgaben und erfüllen gemeinsam unser Gesamtziel, einen reibungslosen Buchungs- und Verwaltungsablauf in einem Carsharing Verein.

Während der Diplomarbeit haben wir sehr viel der in den letzten Jahren erlernten Kenntnisse anwenden können, aber auch unzählige neue Erfahrungen gesammelt und haben damit unser Wissen noch erweitern können. Zusätzlich haben wir im Bereich der Strukturierung und Vorgehensweise bei der Umsetzung von Projekten wertvolle Erfahrung für Zukunft sammeln können.

Anmerkung zu gendergerechter Formulierung:

Aus Gründen der besseren Lesbarkeit wird in dieser Diplomarbeit die Sprachform des generischen Maskulinums angewandt. Diese Art von Formulierung impliziert jedoch keine Benachteiligung des weiblichen Geschlechtes, sondern soll als geschlechtsunabhängig verstanden werden.

## Danksagung

Wir möchten uns an dieser Stelle ganz herzlich bei allen Unterstützern und Mitwirkenden bedanken, die uns bei der Entwicklung des Projekts geholfen, unterstützt und motiviert haben.

Zunächst ein besonders großer Dank geht an unsere beiden Projektbetreuer *Prof. Mag. Dr. Anton Hofmann* und *Prof. DI Martin Aichriedler*, die uns stets mit ihrer Expertise zu Seite gestanden haben. Durch die zahlreichen Tipps und Verbesserungsvorschläge, sowie der motivierenden Worte hat dieses Projekt erst richtig erfolgreich werden können.

Wir bedanken uns auch bei unserem Sponsor und Projektpartner, dem gemeinnützigen Carsharing Verein Freiraum Gneis Mobil und Obmann Johann Lehenuer, ohne den dieses Projekt nicht zustande gekommen wäre. Bei Ing. Gerhard Huber möchten wir uns ebenfalls bedanken, welcher gemeinsam mit Herrn Lehenuer regelmäßig für Gespräche in die HTBLuVA Salzburg gekommen ist und mit seinem Wissen zur Verbesserung und Erweiterung des Projekts beigetragen hat.

Weiters bedanken wir uns herzlich bei Prof. Mag. Paul Schwaiger für das gemeinsame Vorbereiten unserer Präsentation.

Zusätzlich bedanken wir uns bei AV Ing. DI Karl-Heinz Steiner, Prof. DI Anton Jerey, Prof. DI Siegbert Schrempf, DI Michael Offenhuber und DI Dr. Andreas Magauer, die uns mit ihren Fachkenntnissen viele Tipps gegeben haben und ebenfalls immer für Fragen zur Verfügung gestanden haben.

Außerdem möchten wir uns bei den Fachlehrkräften, darunter FL Ing. Dipl.-Päd. Helmut Strasser, FL Ing. Dipl.-Päd. Wolfgang Straßl, FL Ing. BEd Robert Pöttinger, FL Ing. Dipl.-Päd. BEd Herbert Pölzer und FL Ing. Dipl.-Päd. Norbert Wen bedanken, die uns bei der Fertigung unserer Hardware- und Gehäusekomponenten tatkräftig zur Seite gestanden haben.

Vielen Dank an alle Mitwirkenden!

# 1 Inhalt

---

2 Überblick – „The big picture“ .....	13
2.1 Einleitung UML.....	13
2.2 Big Picture .....	13
3 Systemspezifikation.....	14
3.1 Zielbestimmungen .....	14
3.1.1 Musskriterien.....	14
3.1.2 Wunschkriterien .....	15
3.1.3 Abgrenzungskriterien .....	15
3.2 Produkteinsatz .....	16
3.2.1 Anwendungsbereiche .....	16
3.2.2 Zielgruppen.....	16
3.2.3 Betriebsbedingungen .....	16
3.3 Produktumgebung .....	16
3.3.1 Software .....	16
3.3.2 Hardware .....	16
3.4 Produktfunktionen.....	17
3.5 Produktdaten .....	19
3.6 Produktleistungen.....	20
3.7 Qualitätszielbestimmungen .....	21
3.8 Globale Testszenarien und Testfälle .....	22
3.9 Entwicklungsumgebung .....	23
3.9.1 Software .....	23
3.9.2 Hardware .....	23
3.9.3 Orgware .....	23
4 Organisation - Projektmanagement .....	24
4.1 Projektteam .....	24
4.2 Individuelle Aufgabenstellungen inkl. Arbeits- und Terminplan (GANTT-Diagramme) .....	25
4.2.1 Noah Laireiter .....	25
4.2.2 Thomas Ploder.....	26
4.3 Scrum – agile Entwicklungsmethode .....	27
5 Grundlagen und Methoden – Thomas Ploder.....	28
5.1 Aufgabenbereiche.....	28
5.1.1 Projektverwaltung .....	28
5.1.2 Backend .....	28
5.1.3 Frontend .....	28

5.2	Environment .....	28
5.2.1	Betriebssystem .....	28
5.2.2	Server und Domain .....	28
5.2.3	VestaCP .....	29
5.3	Datenbank & Datensicherheit.....	30
5.3.1	Entity-Relationship-Modell.....	30
5.3.2	Ansprüche an eine Datenbank.....	31
5.3.3	Maßnahmen zur Passwortsicherheit.....	31
5.3.4	Passwort zurücksetzen .....	32
5.3.5	SQL-Injection .....	32
5.3.6	XSS (Cross-Site-Scripting).....	35
5.3.7	uXSS (Universal-Cross-Site-Scripting) .....	38
5.3.8	Sicherheitsmaßnahmen.....	39
5.4	Webentwicklung .....	39
5.4.1	Prinzip .....	39
5.4.2	Adaptierbarkeit.....	39
5.4.3	Konfigurierbarkeit.....	40
5.4.4	Besonderheiten .....	41
5.5	Appentwicklung .....	44
5.5.1	Aspekte der Appentwicklung.....	44
5.5.2	Qualitätsmerkmale .....	48
5.5.3	Push-Notifications .....	52
5.5.4	AppStoreConnect .....	62
5.5.5	Play Console.....	63
6	Grundlagen und Methoden – Noah Laireiter .....	64
6.1	Aufgabenbereiche.....	64
6.1.1	KFZ-Tracking-Modul.....	64
6.1.2	Keybox .....	64
6.1.3	Unit-Tests und App-Security.....	64
6.1.4	Dokumentation.....	64
6.2	Hardwareentwicklung .....	64
6.2.1	Autodesk EAGLE .....	64
6.2.2	Autodesk Fusion 360 .....	66
6.3	Hardwareprogrammierung .....	67
6.3.1	Microchip Studio .....	67
6.3.2	chip45boot2 GUI.....	67

6.3.3	HTerm .....	67
6.3.4	Arduino IDE.....	68
6.3.5	IntelliJ IDEA und Visual Studio Code .....	68
6.4	Graphische Visualisierung der Hardware.....	68
6.4.1	Draw.io .....	68
6.5	Hardware .....	69
6.5.1	Wahl des Mikrokontrollers .....	69
6.6	Bussysteme .....	70
6.6.1	UART .....	70
6.6.2	SPI .....	70
6.6.3	I2C.....	70
6.7	Tracking-Modul.....	71
6.7.1	Steuerelement.....	71
6.7.2	Energieversorgung.....	71
6.7.3	GPS-Funktionsweise .....	72
6.7.4	GPS-Modul.....	73
6.7.5	Zwischenspeicher und Logging .....	74
6.7.6	GSM-Funktionsweise .....	75
6.7.7	GSM-Modul .....	76
6.7.8	Gehäuse .....	77
6.8	Schlüsselbox.....	77
6.8.1	Steuerelemente .....	77
6.8.2	Energieversorgung.....	78
6.8.3	Öffnungscode-Generierung.....	78
6.8.4	Codeeingabe .....	79
6.8.5	Schlüsselabholung und Rückgabe.....	79
6.8.6	Gehäuse .....	79
6.9	UNIT-Test .....	80
6.10	App-Security.....	80
7	Ergebnisse – Abnahme .....	81
8	Quellen- und Literaturverzeichnis.....	83
9	Verzeichnis der Abbildungen und Tabellen.....	86
9.1	Glossar .....	89
10	Begleitprotokoll gemäß § 9 Abs. 2 PrO .....	91
10.1	Begleitprotokoll Noah Laireiter.....	91
10.2	Begleitprotokoll Thomas Ploder .....	92

11	Besprechungsprotokolle.....	93
11.1	Besprechungsprotokoll 27.09.2021 .....	93
11.2	Besprechungsprotokoll 15.10.2021 .....	94
11.3	Besprechungsprotokoll 05.11.2021 .....	95
11.4	Besprechungsprotokoll 19.11.2021 .....	96
11.5	Besprechungsprotokoll 10.12.2021 .....	97
11.6	Besprechungsprotokoll 21.01.2022 .....	98
11.7	Besprechungsprotokoll 21.02.2022 .....	99
11.8	Besprechungsprotokoll 18.03.2022 .....	100

## 2 Überblick – „The big picture“

### 2.1 Einleitung UML

Die UML (Unified Modeling Language) stellt Bausteine zu Verfügung, mit welchen es ermöglicht wird, Struktur, Verhalten und Aufbau von Software visuell darzustellen. Mit einem UML-Diagramm kann anderen Personen ein funktionaler, überschaubarer Einblick in ein komplexes System gewährt werden. Aber auch in anderen Bereichen, weit entfernt von Software, bei denen Abläufe und Zusammenhänge beschrieben werden, können diese Diagramme abhelfen. [1]

### 2.2 Big Picture

In Zusammenarbeit mit dem gemeinnützigen Carsharing-Verein „Freiraum Gneis mobil“ in der Stadt Salzburg wird ein digitales Buchungssystem entwickelt. Dabei werden auch die Fahrtenkosten jeder Fahrt automatisch für den Benutzer kalkuliert.

Die Messung der zurückgelegten Distanz erfolgt durch eine Hardware im Auto, die universell auf jedes Kraftfahrzeug adaptierbar ist. Sie setzt sich aus mehreren Sensoren und einem Übertragungsmodul zusammen. Dieses übermittelt in regelmäßigen Abständen aktuelle Fahrtinformationen an den Server.

Weiters wird eine Schlüsselbox entworfen, in welcher der Autoschlüssel aufbewahrt wird. Mittels eines Zahlencodes, der dem Benutzer bei der Buchung des Autos zugewiesen wird, kann die Box später geöffnet werden.

Das gesamte Buchungssystem ist modular auf eine beliebige Anzahl an Fahrzeugen erweiterbar, kann über eine Website oder eine Applikation am Smartphone erreicht werden und ist für alle Altersgruppen einfach zu bedienen.

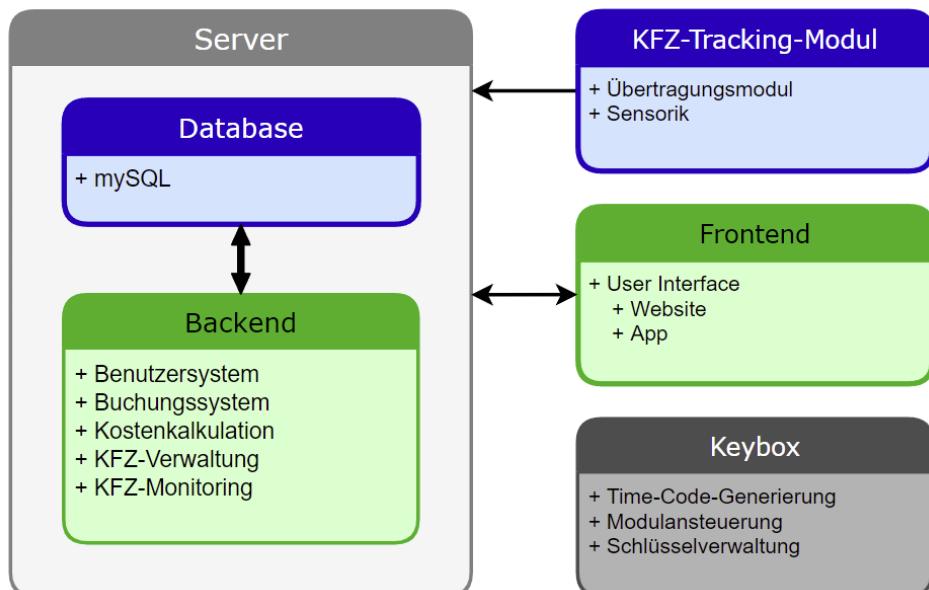


Abbildung 2.2.1: Aufbau des LPCS Carsharing Systems

## 3 Systemspezifikation

---

### 3.1 Zielbestimmungen

#### 3.1.1 Musskriterien

Diese Punkte müssen von den einzelnen Teilbereichen des Projekts erfüllt werden, um als vollständige Anwendung zu funktionieren und als fertiges Projekt zu gelten:

##### **Benutzersystem:**

- Registrierung und Login im System
- Benutzerdaten editieren
- Account löschen

##### **Buchungssystem:**

- Fahrzeug buchen
- Buchung stornieren mit Angabe eines Grundes
- Gesamte Kosten der Buchungen einsehen
- Ladedauer für E-Autos über Fahrdistanz berechnen

##### **Administration:**

- Benutzer editieren, einschränken und löschen
- Fahrzeuge anlegen, editieren und löschen
- Buchungsübersicht über alle Fahrzeuge
- Fahrttarife anlegen, zuweisen und löschen

##### **Tracking-Modul:**

- Mit USB im Fahrzeug zu versorgen
- Standort der Fahrzeuge empfangen
- Standort speichern und an den Server senden

##### **Zugriff auf das System:**

- Website
- Responsive Ansicht
- Einfache Bedienung für alle Altersgruppen

### 3.1.2 Wunschkriterien

Wenn Zeit übrigbleibt, sollte die Wichtigkeit folgender Punkte evaluiert und Teilaaspekte davon im Projekt eingebunden werden.

**Benutzersystem:**

- Daten bei Registrierung überprüfen (Adresse mit PLZ, IBAN)

**Buchungssystem:**

- Fahrzeug mit Chauffeurdienst buchen
- Zeitcodes zum Öffnen der Schlüsselbox generieren
- Monatliche Kostenabrechnung als PDF herunterladen

**Administration:**

- Monatliche Kostenabrechnung als PDF herunterladen

**Schlüsselbox:**

- Öffnen, wenn korrekter Zeitcode eingegeben wird
- Gleicher Code für Abholung und Rückgabe (diese sind zu unterschiedlichen Zeitpunkten)
- Codeeingabe auf Display anzeigen

**Zugriff auf das System:**

- Applikation am Smartphone

### 3.1.3 Abgrenzungskriterien

Die folgenden Aspekte sind nie in Planung gewesen und werden nicht vom Endprodukt unterstützt:

**Administration:**

- Vollständige Automatisierung der Einnahmen-Ausgaben-Rechnung im Verein
- Automatische Kostenabbuchung von Bankkonten der Nutzer

**Schlüsselbox:**

- wassererdicht
- autark mit Solarpanel und Akku versorgen

**Zugriff auf das System:**

- Visuelle Ansicht individuell anpassbar (Schriftgröße, Farben, ...)

## 3.2 Produkteinsatz

### 3.2.1 Anwendungsbereiche

Das LPCS Carsharing System kann bei jedem Carsharing Anbieter implementiert werden. Es bietet eine schlichte, übersichtliche Lösung seine Fahrzeuge zu verwalten und diese für registrierte Nutzer zur Verfügung zu stellen.

### 3.2.2 Zielgruppen

Aufgrund des simplen Designs mit großen Buttons und großer Schrift, ist dieses System für alle erwachsenen Altersgruppen geeignet. Nach dem Login kann der Nutzer mit einem Klick direkt mit der Buchung starten und diese unkompliziert in 3 kurzen Schritten abschließen. Mit einem weiteren Klick erhält der Nutzer eine Übersicht über alle Buchungen und Kosten.

Das Admin-Control-Panel ermöglicht es dem Inhaber, die Benutzer, Fahrzeug, Buchungen, Tarife und Aktivitäten im System im Überblick zu behalten.

### 3.2.3 Betriebsbedingungen

Die Software läuft auf einem Server und ist einfach von jedem internetfähigen Gerät überall auf der Welt erreichbar. Es ist zu Beginn lediglich die Schlüsselbox aufzubauen, die später für neue Fahrzeuge modular erweitert werden kann. Anschließend muss in jedem Fahrzeug nur mehr das Tracking-Modul angesteckt werden und das System ist betriebsbereit.

## 3.3 Produktumgebung

### 3.3.1 Software

- Website auf einem Server
- Buchungs- und Verwaltungssoftware mit angebundener Datenbank

### 3.3.2 Hardware

- Tracking-Modul für jedes Fahrzeug
- Eine für beliebig viele Fahrzeuge erweiterbare Schlüsselbox

### 3.4 Produktfunktionen

#### **/F0010/ Anmeldung bei dem Benutzersystem (erreicht)**

Nutzer können sich einfach und unkompliziert auf der Website mit E-Mail und Passwort anmelden. Bei Erstanmeldung müssen zusätzlich personenbezogene Daten (Name, Telefon, Adresse, ...) angegeben werden. Und sollte es dazu kommen, dass jemand sein Passwort vergisst, kann es über die E-Mail-Adresse rückgesetzt werden.

#### **/F0020/ Konto bearbeiten (erreicht)**

In der Benutzerübersicht können die Daten bearbeitet werden und das eigene Konto gelöscht werden. Administratoren können das Konto ebenfalls editieren oder entfernen. Zusätzlich dürfen diese Konten freischalten, sperren und ihnen Rollen oder Sonderrechte zuteilen.

#### **/F0030/ Chauffeur-Dienst (nicht erreicht)**

Jene mit der Rolle des Chauffeurs haben die Möglichkeit anzugeben, zu welchen Zeitpunkten sie zur Verfügung stehen. Während ihrer angegebenen Zeiten, haben sie trotzdem die Möglichkeit ein Auto zu reservieren. Für dieses Zeitfenster werden sie als Chauffeur gesperrt.

#### **/F1010/ Reservierung eines Fahrzeuges (erreicht)**

Freigeschaltene Nutzer mit einer Rolle, die es erlaubt ein Auto zu reservieren, haben die Möglichkeit dazu. Dabei gibt es eine Auswahl des Fahrzeugs, des Fahrzeitraumes, wie weit in etwa gefahren werden soll und ob ein Chauffeur benötigt wird. Vor Abschluss der Buchung werden die Fahrtkosten grob aus den Eingaben ermittelt und ein Zeitschlüssel zum Öffnen der Schlüsselbox wird angezeigt. Mit Bestätigung wird eine Reservierungsbestätigung per E-Mail versendet. Diese und alle anderen Reservierungen werden des Weiteren in der Kontoübersicht einsehbar sein.

#### **/F1020/ Reservierung bearbeiten (erreicht)**

Außerhalb einer angegebenen Frist ist es dem Nutzer möglich, seine Reservierungen kostenfrei zu stornieren. Ansonsten werden die vollen Kosten verrechnet (Kosten der Buchungsdauer und gefahrene Kilometer in dieser Zeit). Den Administratoren ist es vorbehalten Reservierungen jederzeit abzuändern oder zu stornieren. Bei Änderungen wird eine E-Mail an den Nutzer gesendet.

#### **/F1030/ generelle Buchungseinstellungen (erreicht)**

Diese sind ausschließlich durch den Administrator änderbar. Hierunter fallen: die maximale Reservierungsdauer, Zeitfenster in denen Buchungen möglich sind, wie weit im Voraus Fahrzeuge reserviert werden können, das Sperren einzelner KFZ (zum Beispiel: Wartung) und die Kosten, die pro Kilometer und Zeiteinheit verrechnet werden.

**/F2010/ KFZ-Daten Verwaltung (erreicht)**

Dem Administrator ist es möglich ein neues Auto mit allen fahrzeugrelevanten Daten anzulegen. Diese können im Nachhinein jederzeit editiert oder gelöscht werden.

**/F2020/ Visualisierung des Zustandes (größtenteils erreicht)**

In einer Art Tabelle wird für einzelne Zeiteinheiten angezeigt, ob Autos gebucht sind, gesperrt sind oder aufgeladen werden (für E-Autos). Zusätzlich werden Daten wie zum Beispiel der Kilometerstand angezeigt. Es gibt weiters die Möglichkeit die aktuelle Fahrzeugposition abzurufen.

**/F3010/ Fahrtenbuch (nicht erreicht)**

Während der Fahrt und am Ende der Fahrt wird das Fahrtenbuch regelmäßig aktualisiert, damit die Abrechnung von den Nutzern automatisiert erfolgt. Dabei können die Rechnungen als Datei heruntergeladen werden.

**/F3020/ Statistiken (erreicht)**

Es wird erfasst wie viele Mitglieder aktiv sind, seit wann diese registriert sind und welche Rolle sie haben. Daraus wird eine Mitgliederliste generiert, welche als Datei heruntergeladen werden kann. Weiters wird aufgezeichnet welche Personen am häufigsten ein Fahrzeug buchen und wie weit sie fahren.

**/F3030/ Logging (erreicht)**

In der Datenbank werden alle Aktionen des Systems mitprotokolliert und können jederzeit abgerufen werden.

**/F3110/ Gruppennachrichten senden (erreicht)**

Um nicht jeder Person einzeln eine E-Mail schicken zu müssen, gibt es die Möglichkeit Gruppennachrichten an ausgewählte Personen, an eine Rolle oder alle zu versenden. Das System sendet Benachrichtigungen automatisch aus, sollten sich zum Beispiel Tarife oder Nutzungsbedingungen ändern.

**/F4010/ Responsive Ansicht (erreicht)**

Die Benutzeroberfläche ist für Alle einfach zu bedienen und gut leserlich. Auf kleineren Geräten (zum Beispiel: Smartphones) passt sich das Layout an, um diesen Anforderungen nachzukommen.

## 3.5 Produktdaten

### /D110/ Datenbank

Unter anderem die Benutzerdaten (Name, Wohnort, IBAN, ...) und Fahrzeugdaten werden in einer Datenbank gespeichert.

### /D111/ Systemlogs

Die Software protokolliert alle Aktionen (jeden Tastendruck, jeden Knopfdruck, jede Änderung) der Nutzer mit, speichert diese mit dem Zeitstempel in einer Datenbank und sind für die Administratoren im Control Panel einsehbar.

The screenshot shows a web-based administration interface for a vehicle tracking system. The URL is lpcs.at/demo/controlpanel/users/userdetails/?uid=aaaaaaaaaaaaaab. The page title is 'Nutzerdetails - Greis mobil'. The main content area is titled 'Laireiter Noah' and displays two tabs: 'Übersicht' (selected), 'Buchungen & Kosten', and 'Aktivität'. The 'Aktivität' tab is currently active and shows a list of activities under the heading 'Aktivitäten (158)'. The activities are categorized into sections: 'TARIF GEÄNDERT', 'NEUE BUCHUNG ABGESCHLOSSEN', 'FREMDEN ACCOUNT BEARBEITET', and 'NUTZER FREIGESCHALTEN'. Each section contains detailed log entries with timestamps (e.g., 14.03.2022 um 11:21:53 Uhr, 14.03.2022 um 11:19:12 Uhr, 14.03.2022 um 11:15:23 Uhr, 14.03.2022 um 11:15:06 Uhr). Below each section, there are links to 'Sitzungsdetails' and 'Sitzung beenden'.

Abbildung 3.5.1: Anzeige der Nutzeraktivitäten im Admin-Control-Panel

### /D201/ Standort

Für jede Fahrt wird eine lokale Sicherungsdatei mit Datum und Nummer der Fahrt am Tracking-Modul erstellt, um im Falle eines Verbindungsauftauchs die Fahrtdata nicht zu verlieren.

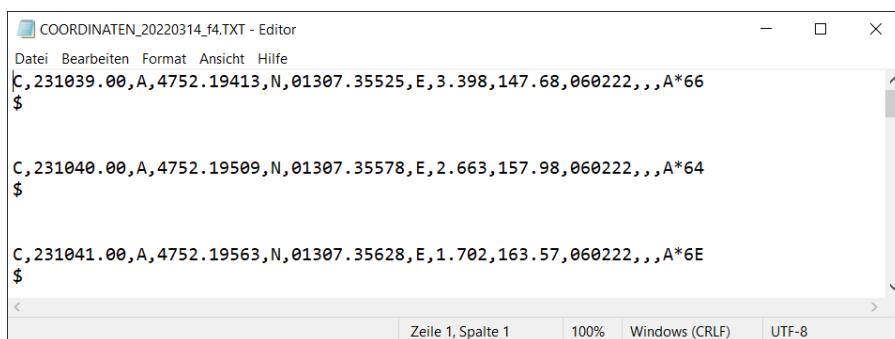


Abbildung 3.5.2: Sicherungsdatei mit Zeit- und Standortdaten am Tracking-Modul

## 3.6 Produktleistungen

### ***/L001/ Handhabung***

Das Bedienen des Systems soll intuitiv und leicht zugänglich gestaltet werden. Weiters sollen Eingabefelder und Knöpfe auf ein Minimum reduziert werden, um unnötige Verwirrungen zu verhindern und die Software möglichst einfach bedienen zu können.

### ***/L002/ Fahrzeugtracking***

Alle Fahrzeuge im Verein sollen jederzeit getrackt werden können. Dies dient zum einen zur Abrechnung der Fahrten, da die Kosten an die gefahrenen Kilometer gebunden sind und Benutzer daher nicht jede Fahrt umständlich, händisch in einem analogen Fahrtenbuch dokumentieren müssen. Zum anderen schützt es vor Missbrauch und soll Diebstahl vorbeugen, da ein Administrator jederzeit den Standort der Fahrzeuge abfragen kann.

### ***/L003/ Schlüsselbox***

Die Schlüsselbox kann einfach überall aufgestellt werden, wo es einen Stromanschluss gibt. Aufgrund des ausgeklügelten Konzepts, dass der Code zum Öffnen über die Uhrzeit kalkuliert wird, benötigt diese keine Verbindung mit dem Internet und kann so auch in Tiefgaragen installiert werden.

### 3.7 Qualitätszielbestimmungen

	sehr wichtig	wichtig	weniger wichtig	unwichtig
Benutzerfreundlichkeit	X			
Korrektheit	X			
Effizienz		X		
Kompatibilität		X		
Portierbarkeit				X

Tabelle 3.7.1: Die 5 Qualitätsbestimmungen von LPCS

- **Benutzerfreundlichkeit**

Da dieses System von allen Altersgruppen bedient werden soll, ist es sehr wichtig die Oberfläche einfach und schlicht zu gestalten. Man sollte sich auf der Website und in der App leicht zurechtfinden, um Buchungen unkompliziert abschließen zu können. Weiters soll es mühelos möglich sein, anfallende Kosten einzusehen und seine personenbezogenen Daten zu editieren.

- **Korrektheit**

Eine hohe Priorität liegt auch auf der Korrektheit. Es dürfen keine Daten verloren gehen oder falsch interpretiert werden, da ansonsten sensible Daten offengelegt werden könnten oder Kosten inkorrekt berechnet werden. Daher ist es sehr wichtig auf diesen Punkt ein erhöhtes Augenmerk zu legen.

- **Effizienz**

Wenn die Anzahl an Fahrzeugen im Verein wächst und die Zugriffe auf das System ansteigen, soll die Performance nicht darunter leiden.

- **Kompatibilität**

Besonders das Trackingmodul in den Fahrzeugen soll mit jedem kompatibel sein, heißt es soll in jedem Auto mit Strom versorgt werden können. Angedacht ist hier eine Lösung mit einer USB-Schnittstelle, da diese es auch ermöglicht, das Modul über einen Adapter und den Zigarettenanzünder, in seltenen Fällen mit einer Steckdose im Fahrzeug, zu versorgen.

- **Portierbarkeit**

Es wird bei diesem System nicht darauf geachtet plattformunabhängig zu sein, da sich die Software zum größten Teil auf einem Server befindet und auf diesen von überall zugegriffen werden kann.

## 3.8 Globale Testszenarien und Testfälle

### **/T0010/ Anmeldung bei dem Benutzersystem**

Registrierung und Login im System ist simpel gestaltet und übersichtlich.

### **/T0020/ Konto bearbeiten**

Die bei der Registrierung angegebenen Daten können im Nachhinein jederzeit abgeändert und auch gelöscht werden.

### **/T1010/ Reservierung eines Fahrzeuges**

Nur freigeschaltete Nutzer mit einer berechtigten Rolle haben die Möglichkeit ein Fahrzeug zu buchen und einen Schlüsselboxcode zu bekommen. Diese Buchungen können auch storniert werden (entweder vom Nutzer oder von einem Administrator).

### **/T2010/ KFZ Verwaltung**

Nur ein Administrator kann neue Fahrzeuge anlegen, editieren, den aktuellen Standort einsehen und diese löschen. Weiters können vergangene und geplante Buchungen für die Fahrzeuge angezeigt werden.

### **/T3010/ Fahrtenbuch**

Das Tracking-Modul sendet während einer Fahrt regelmäßig Standortdaten an das System. Anhand dieser können die Kilometerkosten für die einzelnen Fahrten automatisiert kalkuliert werden.

### **/T3110/ Gruppennachrichten senden**

Systemnachrichten der Administratoren können an alle Nutzer oder nur an ausgewählte Nutzer versendet werden.

### **/T4010/ Responsive Ansicht (erreicht)**

Die Benutzeroberfläche ist für jeden Anwender einfach zu bedienen und gut leserlich. Auf kleineren Geräten (zum Beispiel: Smartphones) passt sich das Layout an, um dieser Anforderung gerecht zu werden.

## 3.9 Entwicklungsumgebung

### 3.9.1 Software

- DataGrip 2021.1
- WebStorm 2019.2
- PhpStorm 2021.1.4
- IntelliJ IDEA 2021.1.3
- FileZilla 3.46.3
- PuTTY 0.76
- Postman 8.7.0
- Xcode 13.1
- Android Studio 4.1.3
- iTerm 2.12
- Visual Studio Code 1.65.2
- MAMP 5.3
- Roundcube
- VestaCP

### 3.9.2 Hardware

- Microchip Studio 7.0.2542
- chip45boot2 GUI 1.11
- HTerm 0.8.1
- Arduino IDE 1.8.20
- Autodesk Eagle 9.6.2
- Autodesk Fusion 2.0.12392

### 3.9.3 Orgware

- Microsoft Word (Office 365)
- Microsoft PowerPoint (Office 365)
- Microsoft Excel (Office 365)
- Adobe Photoshop 2021
- madewithcode.at (GitLab)
- draw.io
- Trello
- Discord 0.0.266

## 4 Organisation - Projektmanagement

---

### 4.1 Projektteam

Individuelle Aufgabenstellungen der Teammitglieder

**Noah Laireiter:**

1. Dokumentation
2. KFZ-Tracking-Modul
  - o Sensorik
  - o Anbindung an das System
3. Keybox
  - o Gehäuseentwicklung
  - o Öffnungsmechanismus
  - o Anbindung an das System
4. Softwareentwicklung
  - o Kostenkalkulation
  - o Unit-Tests und App-Security

**Thomas Ploder:**

1. Projektmanagement
2. Backend
  - o Server-Management, DNS, SSL
  - o Datenbankverwaltung
  - o Benutzersystem
  - o Buchungssystem
  - o KFZ-Verwaltung
3. Frontend
  - o Website
  - o App-Entwicklung (Native oder Web)

## 4.2 Individuelle Aufgabenstellungen inkl. Arbeits- und Terminplan (GANTT-Diagramme)

### 4.2.1 Noah Laireiter

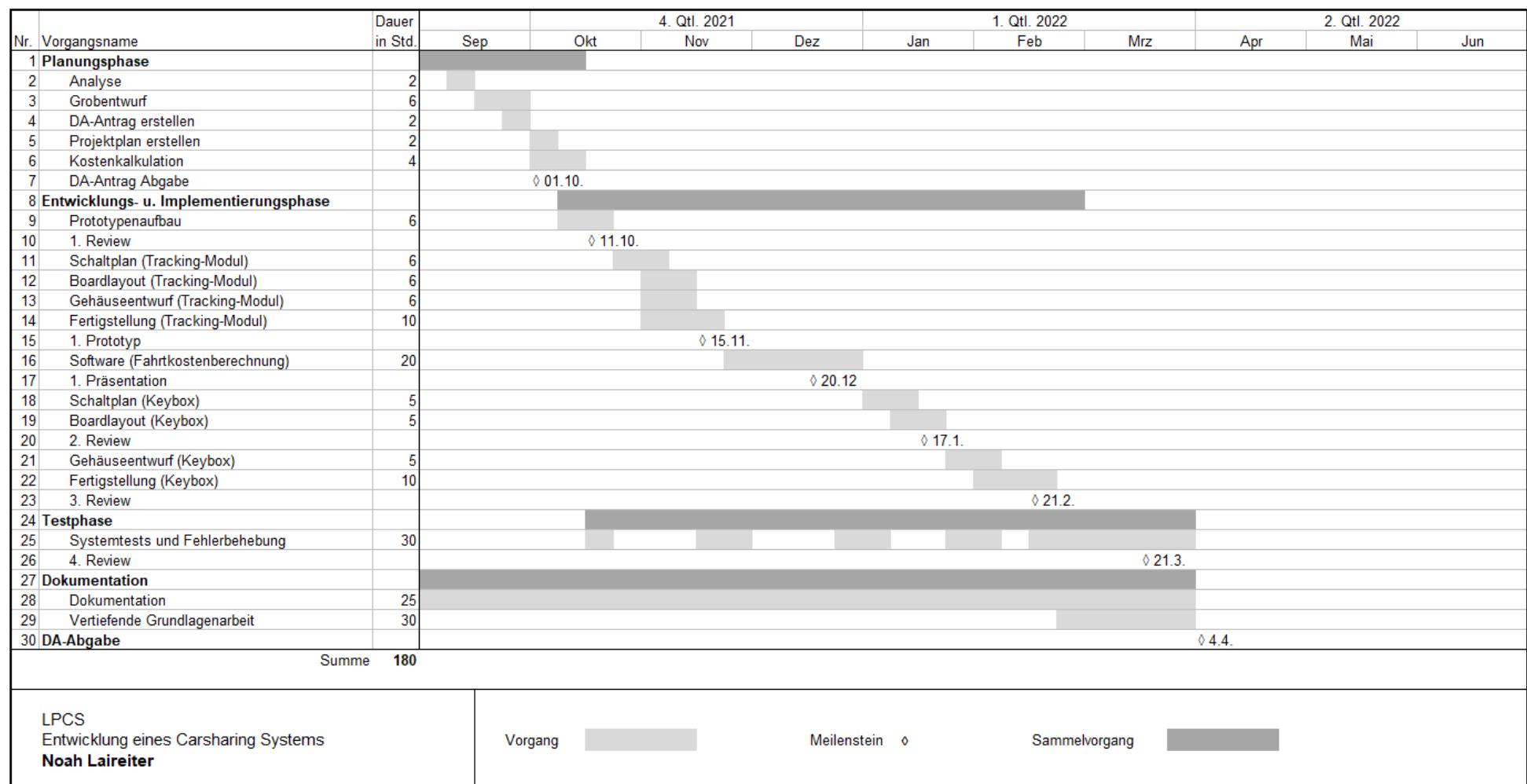


Abbildung 4.2.1: Gantt-Diagramm – Noah Laireiter

## 4.2.2 Thomas Ploder

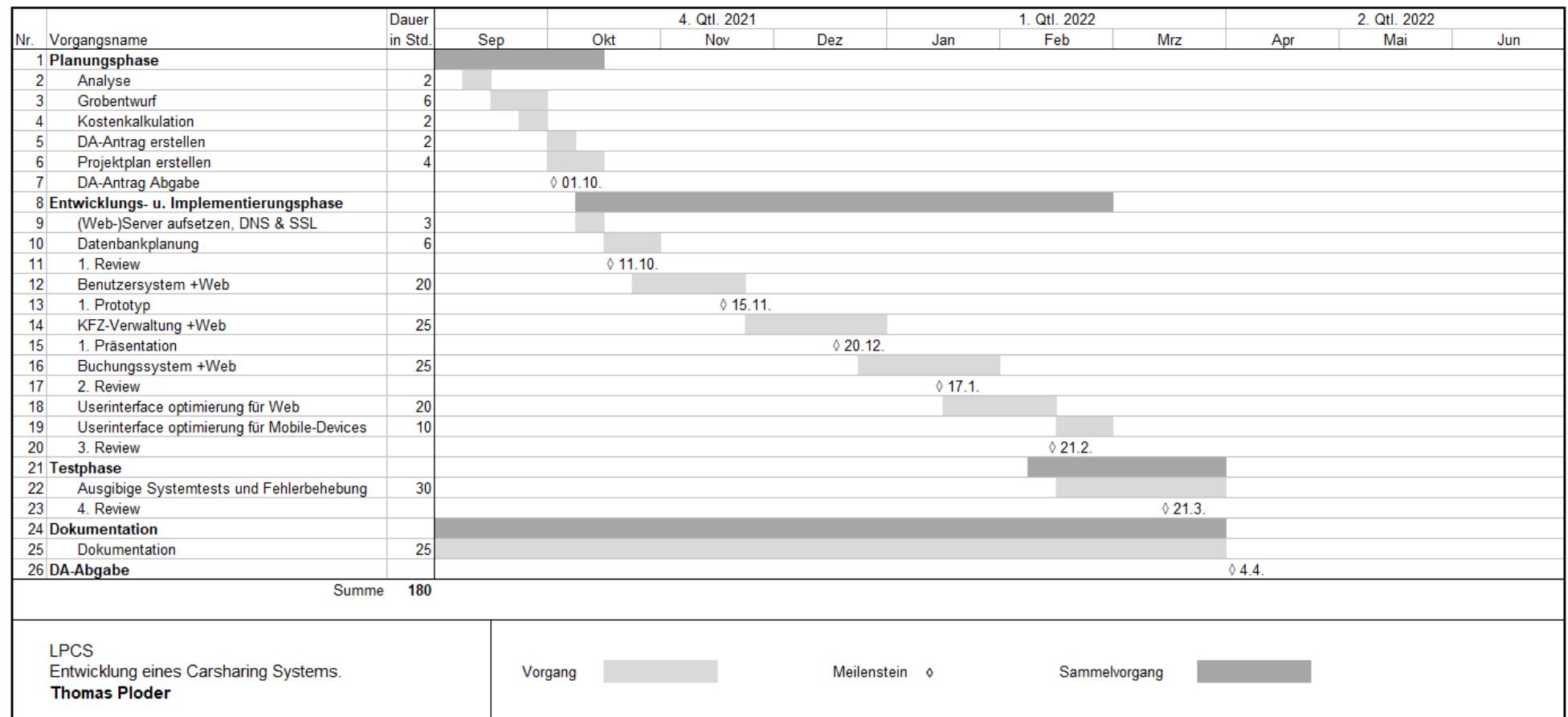


Abbildung 4.2.2: Gantt-Diagramm – Thomas Ploder

## 4.3 Scrum – agile Entwicklungsmethode

Bei diesem Projekt ist die Methodik des agilen Projektmanagements (Scrum) angewendet worden. Das Ziel dieser Herangehensweise ist es, den Arbeitsprozess für die Arbeitenden zu strukturieren. Dabei ist Scrum aber nur ein Gerüst, welches als Leitfaden für die geregelte Zusammenarbeit und Kommunikation dienen soll. Dieses legt Funktionen für Teammitglieder fest und beinhaltet einen strukturierten Ablauf. [2]

### Vorgehensweise:

Beim Projektstart wird jedem in der Arbeitsgruppe eine der folgenden Rolle zugewiesen:

- Product-Owner              (Vertreibt Interessen der Anwender / Stakeholder des Produkts)
- Scrum Master              (trägt Verantwortung, Projektunterstützer, Ansprechpartner)
- Entwicklungsteam        (selbstorganisierend in fünf bis zehner-Gruppen)

Sind diese vergeben wird das Product Backlog angelegt, welches die Produktanforderungen des Product-Owners auflistet. Diese werden nach ihrer Priorität sortiert, da die wichtigsten Funktionen mit höchster Kundenzufriedenheit zusammenhängen. Die restlichen Punkte werden zusammengelegt oder auf weiterführende Projekte verschoben.

Aus dem Product Backlog werden anschließend immer wieder neue Sprint Backlogs erstellt. Diese enthalten nur einzelne Aspekte des gesamten Projekts, welche innerhalb eines Sprints abgehandelt werden sollen. Ein Sprint beschreibt dabei einen fixen Zeitraum von maximal einem Monat. Er beginnt mit dem Planning, wo diese Teilziele aus dem Product Backlog formuliert und abgesteckt werden. Während des Monats finden dann täglich sogenannte „Stand-up-Meetings“, auch als „Daily-Scrum“ bezeichnet, statt. Darin beschreibt jedes Mitglied kurz seine Fortschritte und Pläne für den anstehenden Tag. Damit soll evaluiert werden, ob das Sprint-Ziel noch erreicht werden kann.

Nach Abschluss eines Sprints gibt es ein größeres Review. Hier werden die Ergebnisse dem Product-Owner vorgestellt, welcher überprüft, ob sie den Anforderungen entsprechen. Anhand dieses Fazits wird das Product Backlog aktualisiert, um anschließend den neuen Sprint planen zu können.

Zusätzlich zum Review setzt sich das Projektteam auch noch zu einer „Sprint Retrospective“ zusammen. Dabei soll geklärt werden, wie die teaminterne Kommunikation und Zusammenarbeit während des letzten Sprints gelungen ist und was verbessert werden kann. So soll der Lernprozess für kommende Sprints unterstützt werden. [3]

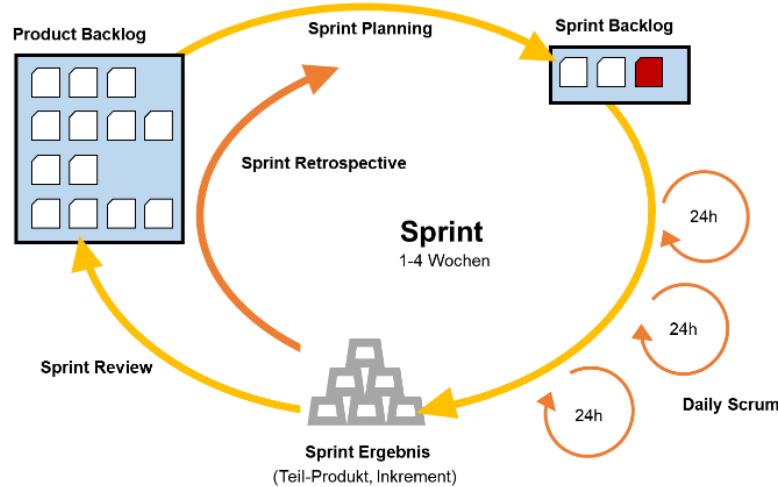


Abbildung 4.3.1: Scrum-Prozess [29]

## 5 Grundlagen und Methoden – Thomas Ploder

### 5.1 Aufgabenbereiche

#### 5.1.1 Projektverwaltung

Austausch und führen der Gespräche mit dem Auftraggeber *Freiraum Gneis mobil* sowie Festlegen der Ziele und dessen Umsetzung.

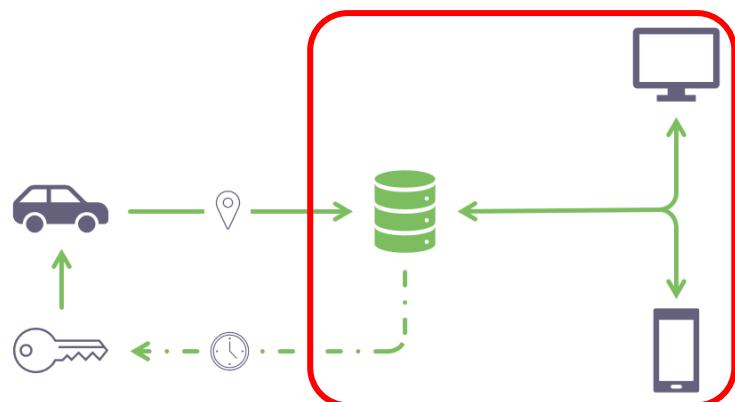


Abbildung 5.1.1: Systemaufbau und Aufgabenbereiche (Thomas Ploder)

Verwaltung des Linux-vServers, der SQL-Datenbank und Umsetzung des Benutzersystems, Buchungssystems und der KFZ-Verwaltung mit serverseitigem PHP-Script.

#### 5.1.2 Backend

Realisierung der Interaktionsmöglichkeiten mit dem System über eine Website oder Handy-App.

### 5.2 Environment

#### 5.2.1 Betriebssystem

Für die Entwicklung der Software wurde gezielt auf das Unix-artige Betriebssystem macOS 12.0.1 „Monterey“ des US-amerikanischen Tech-Unternehmens Apple Inc. gesetzt. Neben der persönlichen Präferenz jenes gegenüber dem Microsoft-Betriebssystem Windows vorzuziehen, bietet macOS zahlreiche Vorteile für Softwareentwickler. Eine integrierte Command Shell mit Linux/Unix-Befehlen ermöglicht eine optimale Integration von Git, Homebrew, CocoaPods, adb, npm und node. Das visuell ansprechende Userinterface ermöglicht außerdem einen verbesserten Workflow, auf den man nach mehreren Jahren Nutzung nicht mehr verzichten möchte.

#### 5.2.2 Server und Domain

Um einen Live-Betrieb zu simulieren, wurde die Domain [lpcs.at](https://lpcs.at) und ein Linux-Server angemietet. Auf dem Demosystem kann die Software umfangreich auf allen erdenklichen internetfähigen Geräten, von allen erdenklichen Nutzergruppen, getestet werden.

Für eine jährliche Gebühr von 11,99€ kann man eine beliebige, noch nicht vergebene, Domain mit der Top-Level-Domain „.at“ beim Prepaid-Hoster [nitrado.net](https://nitrado.net) anmieten. Da die Cloud-Hosting-Dienste bei diesem Anbieter mit Sommer 2022 eingestellt werden, wurde für den Server auf den Prepaid-Hoster [deinserverhost.de](https://deinserverhost.de) zurückgegriffen.

Da das Bereitstellen eines Webservers und einer Datenbank keine anspruchsvolle Hardware benötigt, fiel die Wahl auf einen Linux KVM vServer mit Debian 9. Das Prinzip eines vServer ist, dass mehrere Nutzer sich die Kapazitäten (RAM, CPU-Kerne usw.) eines Root-Servers teilen können, da vermutlich zu keinem Zeitpunkt jede Partei simultan die volle Leistung beanspruchen wird.

Folgende Konfiguration für einen KVM-Server bei [deinserverhost.de](https://deinserverhost.de) wurde somit von uns angemietet:

„KVM VSERVER SSD“ für 3€ pro Monat:

- SSH- und FTP-Zugriff
- Bis zu 8 Xeon Cores
- Bis zu 32GB DDR3/4 ECC RAM
- Bis zu 250GB SSD
- IPv4 und IPv6
- 1 Gbit/s Netzwerk
- 2 TB FairUse Traffic
- DDoS Protection

Damit in späterer Folge Aufrufe korrekt an den nginx-Webserver weitergeleitet werden können, müssen folgende DNS-Einträge in der Domainverwaltung hinterlegt werden:

Subdomain	Typ	Ziel
@	A-Eintrag	45.89.127.211 (Server-IP)
*	A-Eintrag	45.89.127.211 (Server-IP)

Tabelle 5.2.1: DNS-Einträge

Der \*-A-Eintrag betrifft alle Subdomains, inklusive „www“. Somit ist gewährleistet, dass die Website auch mit der Subdomain **www** (=> [www.lpcs.at](http://www.lpcs.at)) erreichbar ist.

### 5.2.3 VestaCP

Die Open-Source-Software „VestaCP“ nimmt einem Systemadministrator viele Prozesse wie das Konfigurieren eines Webservers oder erneuern der SSL-Zertifikate ab. Nach hinzufügen der Domain im Control-Panel hat man die Möglichkeit mühelos MySQL-Datenbanken anzulegen, den integrierten Roundcube-Mailserver zu nutzen und Let's Encrypt SSL-Zertifikate für die Domain und Subdomains auszustellen.

## 5.3 Datenbank & Datensicherheit

### 5.3.1 Entity-Relationship-Modell

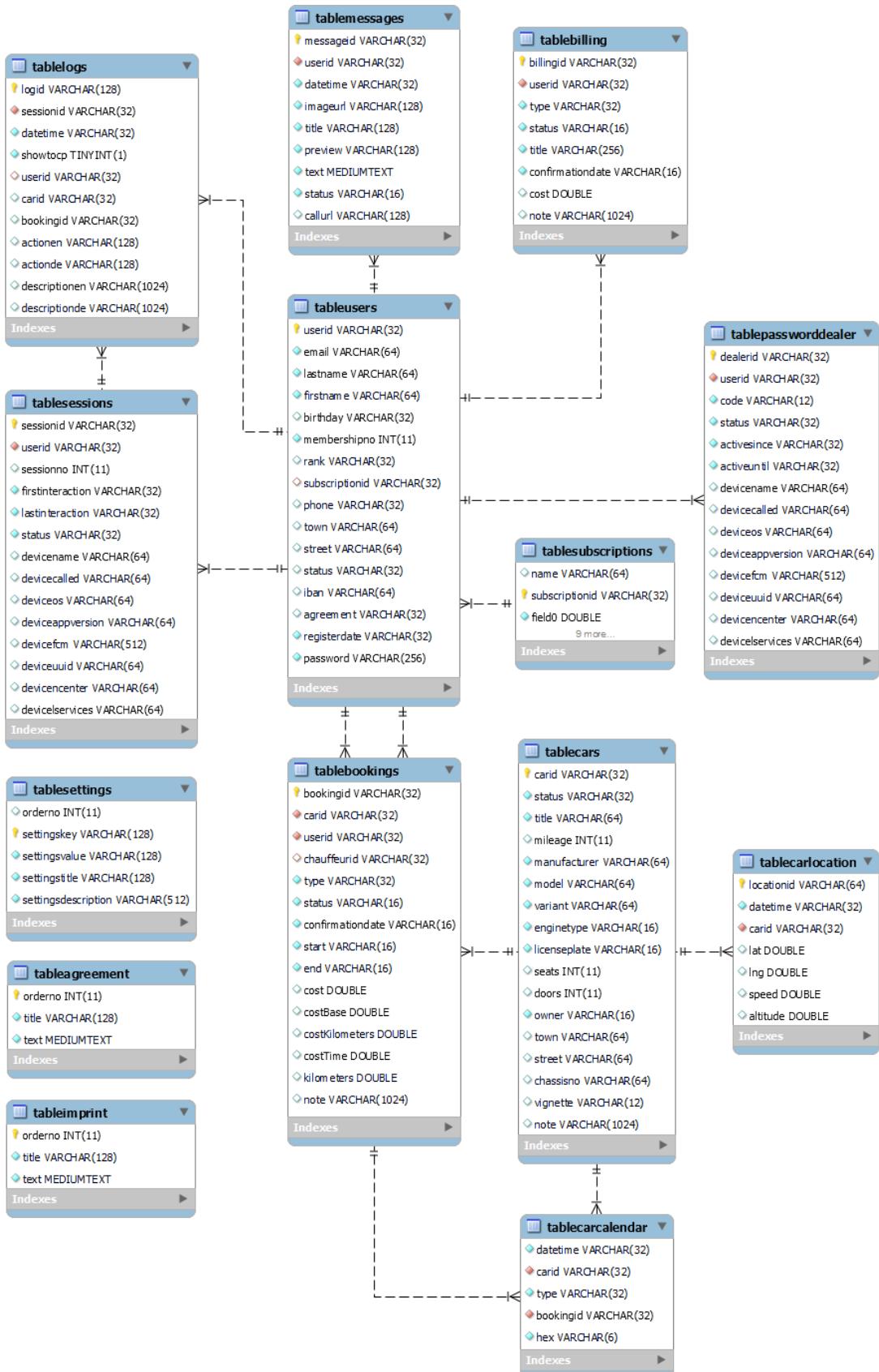


Abbildung 5.3.1: ER-Diagramm der entworfenen Datenbank

### 5.3.2 Ansprüche an eine Datenbank

Ein gutes Datenbankdesign zeichnet sich durch Erfüllen aller Normalformen aus. Der dynamische Prozess des Entwurfs einer Datenbank beherbergt viele potenzielle Risiken, welche dazu führen können, dass zahlreiche Datensätze mehrfach vorhanden sind (Redundanzfreiheit), Schlüssel ins nichts führen oder Felder nicht atomar sind.

Von großer Bedeutung war es daher mit Referenzen zu arbeiten. Jeder eindeutige Datensatz ist mit einem Identifier (z.B. *userid*, *sessionid*, *bookingid*, ...) als Primärschlüssel definiert. Somit ist gewährleistet, dass bei einer Referenz aus einer anderen Tabelle (Fremdschlüssel) der entsprechende Datensatz gefunden werden kann und nicht ins leere führt. Wichtig ist auch, festzulegen, was im Falle einer Löschung des Primärschlüssel-Eintrags mit den Fremdschlüssel-Einträgen passieren soll. Hier gibt es zwei Möglichkeiten: *on delete cascade* und *on delete restrict* – welches je nach Tabelle passend gewählt wurde. Wird zum Beispiel ein Nutzer komplett aus der Tabelle *tableUsers* gelöscht, so werden alle Log-Einträge aus der Tabelle *tableLogs* mitgelöscht (*on delete cascade*).

### 5.3.3 Maßnahmen zur Passwortsicherheit

Um die Privatsphäre der Nutzer zu gewährleisten, wird das von den Nutzern angegebene Passwort in einen md5-Hash verschlüsselt.

Wenn ein Nutzer das Passwort „**ilikefsst**“ angibt, könnte dieses nach Verschlüsselung als „**e20810ad17dcba8de3616ba355f42dcf**“ in der Datenbank gespeichert werden. Die Einwegfunktion von md5 garantiert eine nicht rücktransformierbare Zeichenkette. Doch würde aus unbekannter Ursache ein Datenleck auftreten und die über den Nutzer gespeicherten Daten gelangen in böse Hände, so ist das sensible Passwort trotzdem nicht sicher. Der Vorteil, dass die md5-Funktion nicht zeitabhängig ist und mit einem bestimmten Input immer denselben Output liefert, wird hier zum größten Nachteil. Vergleicht ein Angreifer nun md5-Hash verschlüsselte Zeichenketten (z.B. 10.000 meistbenutzten Passwörter) mit den durch das Datenleck ausgetretenen md5-Hash verschlüsselten Passwörtern, kann eventuell auf das tatsächliche Passwort einzelner Nutzer zurückgeschlossen werden.

Um dieses Risiko weiter zu minimieren, wird ein sogenannter Salt an den Passwortinput angehängt. Dieser Salt darf unter keinen Umstand öffentlich einsehbar sein. Wird bei LPCS beispielsweise der Salt „**2p{QN}**“ genutzt, lautet nun das in der Datenbank befindliche Passwort „**26ea0425427738adc1adc941c802d750**“ (Passwort + Salt = „**ilikefsst2p{QN}**“).

Ob der Salt vor, nach oder in das Passwort eingeschoben wird, ist dem Entwickler überlassen. Dadurch ist gewährleistet, dass das Passwort im Falle eines Datenlecks weiterhin sicher ist und nicht auf die Schnelle geknackt werden kann.

Das Risiko eines leicht zu erratenden Passwortes bleibt allerdings weiter bestehen. Es wäre theoretisch möglich, durch eine Brute-Force-Attacke sich in ein fremdes Nutzerkonto einzuloggen. Möglichkeiten dies zu verhindern wäre durch eine 2FA (Zwei-Faktor-Authentisierung) oder Implementierung einer CAPTCHA. Aus zeitlichen Gründen wurde allerdings auf eine Implementierung dieser verzichtet. Die Kriterien für ein Passwort wurden allerdings so festgelegt, dass der Nutzer eine Passwortlänge von mindestens sechs bis maximal 24 Symbolen angeben muss. Möglich sind dabei alle im ASCII-Code definierten Zeichen ( $2^7 = 128$  Symbole). Wenn eine durchschnittliche Login-Request  $50ms$  dauert, könnte eine Brute-Force-Attacke für das Passwort „**ilikefsst**“ bis zu  $128^n * t = 128^9 * 0,05s = 4,61 * 10^{17}s$  dauern. Dies entspricht ca. 14,6 Mrd. Jahre – zum Vergleich: Das Universum ist ca. 13,8 Mrd. Jahre alt.

```
function md5 (string $string, bool $binary = false): string {}
```

Abbildung 5.3.2: Implementierung der md5-Funktion.

### 5.3.4 Passwort zurücksetzen



Abbildung 5.3.3: Ablauf eines Passwort-Resets.

Um den Passwort-zurücksetzen-Prozess so einfach wie möglich zu gestalten, wird nach Eingabe der Email-Adresse ein sechsstelliger Zifferncode zugesendet. Es wurde bewusst auf einen Direktlink verzichtet, da ältere Menschen öfters Probleme damit haben, wenn sich nach Klicken plötzlich ein anderer Browser wie zuvor öffnet oder ein neues Fenster irgendwo aufspringt. Auch bietet der Code den Vorteil im Falle eines zurücksetzen aus der App heraus nach Klick nicht im Browser anstatt in der App zu landen.

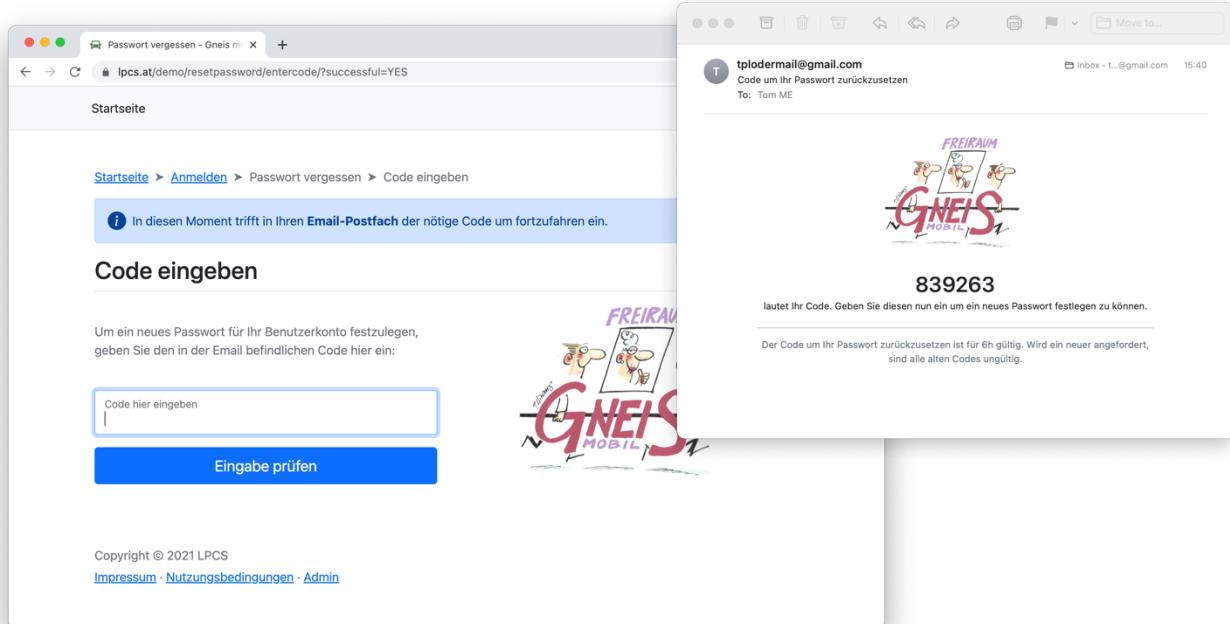


Abbildung 5.3.4: Passwort zurücksetzen mit Email-Code.

Durch die Email kann die Identität des Nutzers bestätigt werden. Der Code kann lediglich für einen Zeitraum von sechs Stunden genutzt werden, um das Passwort zurückzusetzen. Wird ein neuer angefragt, sind alle bis dahin bestehenden Zifferncodes automatisch ungültig und können nicht verwendet werden.

### 5.3.5 SQL-Injection

Eines der leichtesten Methoden ein System zu beeinflussen oder illegal an Daten zu gelangen ist über die gezielte Eingabe von speziellen Zeichensätzen, welche im späteren Verlauf Auswirkung auf das auszuführende SQL-Statement haben. Die Auswirkungen können für das System verheerend sein: von Löschen einer Tabelle bis hin zum Erschleichen einer Berechtigungsgruppe. Das folgende Beispiel veranschaulicht diese potenzielle Sicherheitslücke und zeigt mögliche Präventionsmöglichkeiten.

#### Exemplarische Veranschaulichung des Problems am Demosystem:

Der Nutzer mit der Email-Adresse [max.mustermann@lpcs.at](mailto:max.mustermann@lpcs.at) ist nicht freigeschalten. Er ist somit nicht berechtigt Buchungen durchzuführen. In der Datenbank wird dieser Berechtigungsstatus in der Spalte „status“ als „**[PENDING]**“ vermerkt:

```

SELECT
    tableUsers.email,
    tableUsers.lastname,
    tableUsers.firstname,
    tableUsers.rank,
    tableUsers.status
FROM
    tableUsers
WHERE email='max.mustermann@lpcs.at'
  
```

	email	lastname	firstname	rank	status
1	max.mustermann@lpcs.at	Mustermann	Max	[CP]	[PENDING]

Abbildung 5.3.5: Datenbankeintrag des Nutzers „max.mustermann@lpcs.at“

Unter dem Menüpunkt „Benutzerkonto“ ist der Nutzer in der Lage, je nach Systemeinstellung gewisse Angaben über sich zu ändern. In diesem Fall ist der Nutzer berechtigt seinen Vornamen zu ändern. Durch das gelbe Banner in der oberen Bildschirmhälfte (siehe Abbildung 5.3.4.2) wird klar zum Ausdruck gebracht, dass der Nutzer zu diesem Zeitpunkt nach wie vor sein Konto nur eingeschränkt nutzen kann.

Ihr Benutzerkonto wurde noch nicht freigeschalten. Wir bitten Sie um etwas Geduld bis Ihre Angaben geprüft wurden.

Hallo, Max!

Datenverwaltung

In dieser Übersicht können Sie Ihre personenbezogenen Daten ändern. Nachdem Sie ein Feld editiert haben, drücken Sie "Speichern".

Ihr Nachname Mustermann	Ihr Vorname Max
Ihre Email-Adresse max.mustermann@lpcs.at	
PLZ und Ort 5020 Salzburg	Straße und Hausnummer Itzlinger Hauptstrasse 30
Ihre Telefonnummer 066056132684	

Ich möchte mein Passwort ändern

Nach editieren und Speichern der Daten erfolgt eine Überprüfung der Änderungen durch unseren Systemadministrator.

**Speichern**

Abbildung 5.3.6: Eingeloggter Nutzer „max.mustermann@lpcs.at“ betrachtet Benutzerkonto.

Ohne Maßnahmen zur Prävention einer SQL-Injection würde der Input direkt in das auszuführende Statement eingefügt werden. Vereinfacht dargestellt:

```
$newFirstname = "Max"; // Input from Textfield
$sql = "UPDATE tableUsers SET firstname='' . $newFirstname . '';";
//      UPDATE tableUsers SET firstname='Max'; // Will be executed in the database
```

Abbildung 5.3.7: Veranschaulichung der Datenänderung

Nun wird dieses Wissen genutzt und die Eingabe so abgeändert, dass das SQL-Statement nicht nur den Vornamen ändert, sondern auch die Spalte „status“.

Ihr Vorname  
Thomas', status='[ACCESS]'

Abbildung 5.3.8: Manipulierte Eingabe im Textfeld

Tippt der Anwender an Stelle von „Max“ nun „Thomas', status='[ACCESS]“ in das Eingabefeld, so würde sich dies wie folgt auswirken:

```
$newFirstname = "Thomas', status=' [ACCESS]"; // Manipulated Input
$sql = "UPDATE tableUsers SET firstname='' . $newFirstname . '';";
//      UPDATE tableUsers SET firstname='Thomas', status='[ACCESS]'; // Will be executed
```

Abbildung 5.3.9: Prozess einer Injection

Nachdem dieses Statement in der Datenbank ausgeführt wird, ist zu erwarten, dass der Vorname sich von „Max“ auf „Thomas“ ändert und der Eintrag in der Spalte „status“ sich von „[PENDING]“ auf „[ACCESS]“ ändert. Der Nutzer hat sich somit selbst Zugriff auf das System gewährt und kann Buchungen durchführen.

The screenshot shows a database management interface with two tabs: 'Database' and 'Services'. In the 'Database' tab, the 'lpcs db' database is selected, showing a list of tables including 'tableAgreement', 'tableBilling', 'tableBookings', 'tableCarCalendar', 'tableCarLocation', 'tableCars', 'tableImprint', 'tableLogs', 'tableMessages', and 'tablePasswordDealer'. In the 'Services' tab, there are two entries: 'lpcs db' and 'console\_16'. The 'lpcs db' entry shows a query execution: 'admin\_lpcsdbs.tableUsers 1 s 121 ms'. The 'console\_16' entry shows a query execution: 'console\_16 373 ms'. The main window displays a SQL query in the editor:

```
1 ✓ SELECT
    tableUsers.email,
    tableUsers.lastname,
    tableUsers.firstname,
    tableUsers.rank,
    tableUsers.status
  FROM
    tableUsers
 WHERE email='max.mustermann@lpcs.at'
```

Below the editor, the results pane shows a table with one row:

	email	lastname	firstname	rank	status
1	max.mustermann@lpcs.at	Mustermann	Thomas	[CP]	[ACCESS]

Abbildung 5.3.10: Datenbankeintrag des Nutzers „max.mustermann@lpcs.at“ nach der SQL-Injection

#### Getroffene Maßnahme zur Risikominimierung:

Die Eingabemöglichkeit des Nutzers beherbergt zahlreiche potenzielle Schwachstellen. Einfachstes Mittel ist das Festlegen einer Zeichenobergrenze, da dadurch komplexere Anweisungen nicht möglich sind. Aber vor allem kann mit regex (**Regular Expression**) jedes nicht notwendige Zeichen gefiltert werden. Von allen möglichen Symbolen werden somit nur a-z, A-Z, 0-9 und wenige weitere Satzzeichen als gültige Zeichenkette im Code weiterverarbeitet.

Folgende Methode erfüllt diesen Zweck:

```
<?php
/* ... */
class CoreMethods /* ... */ {

    /**
     * Returns content without any symbols that could cause a sql-injection or xss-attack
     *
     * @param content The content which may contain harmful chars
     * @return Returns regex-Filtered Content
     */

    function handleParameter($content) {
        if (is_string($content)) { // Check for strings
            $content = preg_replace("/[^A-Za-z0-9 ,()._üüäööß@-]/", ' ', $content); // regex
            return $content;
        }

        if (is_int($content) || is_float($content) || is_double($content) || is_bool($content)) {
            return $content; // Return numbers as they are.
        }

        return null; // Any other data types are not allowed.
    }
} /* ... */
?>
```

Abbildung 5.3.11: Möglichkeit zur Prävention von SQL-Injection und XSS in der Sprache PHP

Dadurch, dass die Methode in dem File *CoreMethods.php* geschrieben wurde, welche in jedem anderen File implementiert wurde, kann diese universell eingesetzt werden. Beispiel:

```
include '../backend/CoreMethods.php';
$coreMethods = new CoreMethods();

$inputFirstname = "Thomas', status='[ACCESS]"; // Input sample
$newFirstname = $coreMethods->handleParameter($inputFirstname);

$sql = "UPDATE tableUsers SET firstname='".$newFirstname . "'";;
//      UPDATE tableUsers SET firstname='Thomas, status=[ACCESS]'; // Not the desired effect
```

Abbildung 5.3.12: Anwendung der Methode „handleParameter“ in der Sprache PHP

### 5.3.6 XSS (Cross-Site-Scripting)

Die Eingabefelder auf einer Website könnten auch potenzielles Risiko für eine Cross-Site-Scripting-Attacke werden. Bei jener wird Schadcode (in Form von html- oder script-Tags) in ein Textfeld injiziert und durch die natürliche Logikverarbeitung in weiterer Folge in einer Datenbank oder einem lokalen Speicher gespeichert. Dieser Zeichensatz wird nun zu einem beliebigen Zeitpunkt von einem anderen Nutzer abgefragt und in Form von Inhalt auf der Website dargestellt. Ein normaler Anwender kann dabei allerdings nicht unterscheiden, ob es sich um validen Content handelt oder nicht.

Doch neben der Darstellung von gefälschten Inhalten ist es einem Angreifer bei einem ungeschützten System auch möglich, JavaScript-Code auf einem fremden Gerät ausführen zu lassen. Dadurch ist es möglich, Interaktionen auf einer Website im Namen des Users zu bewirken, ohne dass dieser davon etwas mitbekommt. Auch das Einbinden von iframes kann Nutzer täuschen und den Verlust von persönlichen, sensiblen oder systeminternen Daten bedeuten.

### Aspekte der alert-Methode:

Für unerfahrene Entwickler ist „`alert(1)`“ eine gängige Variante XSS Vulnerabilitäten herauszufinden. Doch „`alert(1)`“ ist eine schlechte Wahl um genau bestimmen zu können, welche Auswirkung der Input auf das System hat. Natürlich veranschaulicht diese JavaScript-Methode visuell deutlich, dass das System zumindest in irgendeiner Weise anfällig für Cross-Site-Scripting ist, vor allem da die alert-Methode ein Teil des `window`-Objekts (`window.alert`) ist, welche sensible Daten wie `window.document.cookie` oder `window.localStorage` beinhaltet. Wird `alert` ausgeführt, kann man daraus schließen, dass es eventuell möglich ist auf jenen sensibleren Part des `window`-Objekts zuzugreifen.

Aber nur weil man einen `alert` erfolgreich getriggert hat, stellt dies noch nicht ausdrücklich eine Sicherheitslücke im System dar. Aussagekräftiger ist es, „`alert(document.domain)`“ oder „`alert(window.origin)`“ zu nutzen, da dies die wirkliche Umgebung in der der Code ausgeführt wird zum Vorschein bringt. Viele Online-Dienste nutzen daher Subdomains für nutzergenerierten Content und binden diesen über einen `iframe` in die Hauptseite ein. Dadurch ist sichergestellt, dass der ausgeführte Algorithmus keinen Zugriff auf die eigentlichen User-Daten hat. „`alert(window.origin)`“ macht diese Vorgehensweise im Gegensatz zu „`alert(1)`“ für den Bug-suchenden Entwickler sichtbar. Doch selbst wenn `window.origin` die tatsächliche URL wiedergibt, muss man noch nicht automatisch von einer kritischen Sicherheitslücke ausgehen. Viele JavaScript-Frameworks wie AngularJS bieten die Limitierung, eingebundenen Content (wie es bei XSS oft der Fall ist) in einem Sandbox-Environment zu implementieren und in späterer Folge auszuführen. Dabei ist es dem Script nur möglich auf scope-Variablen und Mathematik-Operatoren zuzugreifen.

```
<div style="text-align: center">
  <h1>{{ pageTitle }}</h1>
  <p>Folgender Inhalt wird extern geladen: {{ extContent }} </p>
  <div>{{ footer }}</div>
</div>
```

Abbildung 5.3.13: Ausschnitt aus einem AngularJS-File

Es ist zwar möglich einen sogenannten Sandbox-Escape unter Angular anzuregen, allerdings ist jener sehr vertiefend in der Thematik von Cross-Site-Scripting wie folgende Beispiele verdeutlichen:

```
{{ 
  []{{{toString:[]}.join,length: 1,0:'__proto__'}}].assign=[].join;
'a'.constructor.prototype.charAt=''.valueOf;
$eval('x=alert(1)//');
}}
```

Abbildung 5.3.14: Sandbox-Escape unter AngularJS. [12]

```
{{{a='constructor';b={};a.sub.call.call(b[a].getOwnPropertyDescriptor(b[a].getPrototypeOf(a.sub),a).value,0,'alert(1)')()}}}
```

Abbildung 5.3.15: Sandbox-Escape unter AngularJS. [12]

Sandbox-(Sub-)Domains für nutzergenerierten Content zu nutzen, um diesen zu isolieren und den möglichen Zugriff zu beschränken, ist somit die empfohlene Vorgehensweise, um XSS-Attacken vorzubeugen.

### Exemplarische Veranschaulichung des Problems am Demosystem:

Wie bereits unter Punkt 5.3.5 (*SQL-Injection*) manipuliert der Nutzer seine Angabe zum Vornamen, indem er ein Script-Tag gezielt einsetzt:

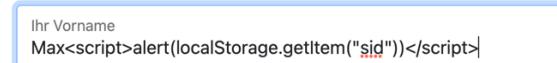


Abbildung 5.3.16: Ausnutzen der XSS-Sicherheitslücke

Klickt der Nutzer nun auf Speichern, wird der neue Vorname ohne Überprüfung in die Datenbank geschrieben. Anschließend wird der Seiteninhalt neu geladen und der Vorname in der Überschrift angezeigt. Dadurch wird der injizierte Code direkt ausgeführt:

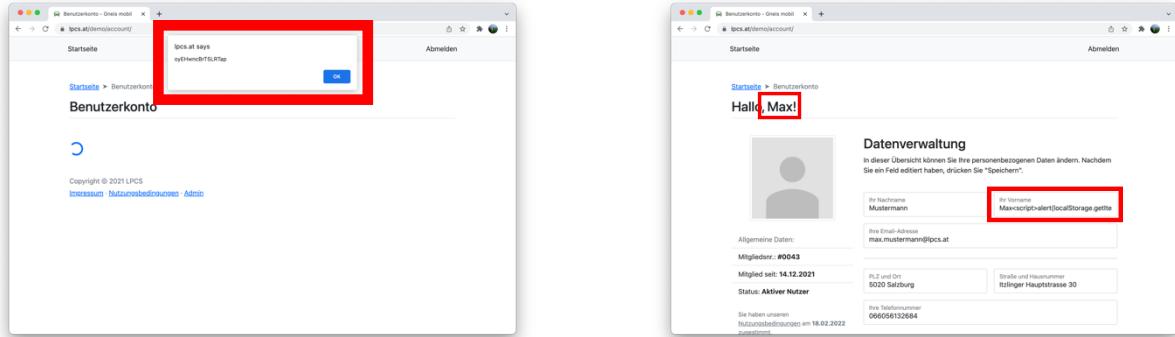


Abbildung 5.3.17: Auswirkung der XSS-Sicherheitslücke

Zunächst unscheinbar hat diese Aktion allerdings auch weitreichendere Auswirkungen: Möchte der Systemadministrator die Nutzerübersicht betrachten, wird ebenfalls der Vor- und Nachname in der Liste aller Nutzer geladen und angezeigt. Dies bewirkt, dass auch im Administrations-Control-Panel der Code ausgeführt wird.

Doch auch im Sinne von Phishing stellt diese Sicherheitslücke ein großes Risiko dar. Ein URL-Redirect könnte den Nutzer im Glauben lassen, er sei nach wie vor auf [lpcs.at](https://lpcs.at), obwohl er diese Seite bereits unwissentlich verlassen hat. Dadurch würden Nutzer leichter dazu neigen, Zugangs- oder Bankdaten einzugeben.



Abbildung 5.3.18: Ausnutzen der XSS-Sicherheitslücke

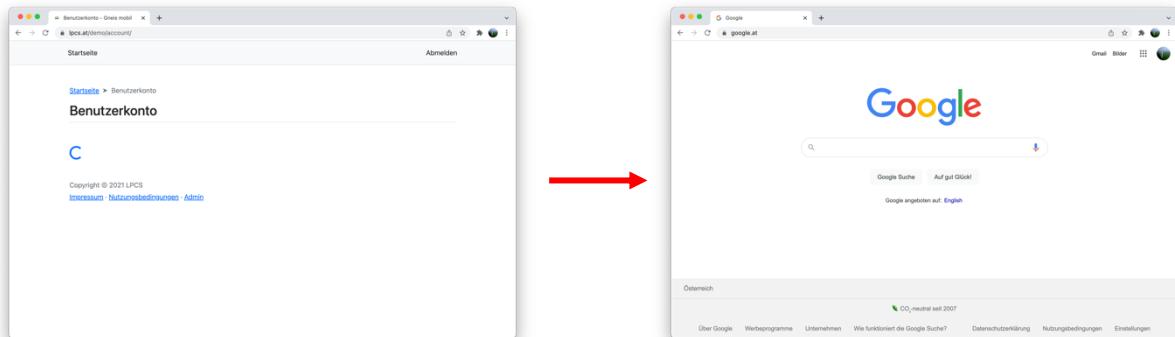


Abbildung 5.3.19: Auswirkung der XSS-Sicherheitslücke

### Getroffene Maßnahmen zur Risikominimierung:

Die in Punkt 5.3.5 (*SQL-Injection*) getroffenen Maßnahmen berücksichtigen auch die grundlegende Prävention von Cross-Site-Scripting, indem alle nicht notwendigen Symbole (vor allem „<“, „>“, „“ und „“) gefiltert werden.

### 5.3.7 uXSS (Universal-Cross-Site-Scripting)

Der Browser garantiert, dass JavaScript-Code nur auf Elemente zugreifen kann, in dessen Umgebung er gerade ausgeführt wird. Das heißt, dass JavaScript-Code welcher auf [lpcs.at](#) ausgeführt wird nicht auf Inhalt von [uxsstest.lpcs.at](#) zugreifen kann (sog. „Origin-check“). Wird ein JavaScript-Element allerdings auf [lpcs.at/demo](#) verändert, kann von [lpcs.at](#) sehr wohl darauf zugegriffen werden, da dies aus Sicht des Browsers dieselbe Domain ist.

Versagt ein Browser nun beim *origin-check*, wäre es einem Angreifer möglich, von einer externen Seite Nutzerdaten wie z.B. die *sessionid*, welche in *localStorage* gespeichert ist, zu stehlen. Die JavaScript-Methode „*history.replaceState*“, welche die angezeigte URL im Browser modifiziert, könnte bei Google Chrome 66.0.3359.122 (stable) auf iOS für eine uXSS-Attacke als Schwachstelle ausgenutzt werden.

```
history.replaceState('','','...;@lpcs.at:%3443/');
```

Abbildung 5.3.20: Ausnutzen einer uXSS-Vulnerability

Würde man dieses Codesegment nun in eine fremde JS-File implementieren, lässt man den Browser glauben, er sei auf [lpcs.at](#), obwohl er dies nicht ist. Dadurch gäbe es die Möglichkeit auf gespeicherten Inhalt, Variablen oder Elemente im RAM von [lpcs.at](#) zuzugreifen. Diesen Bug zu fixen liegt allerdings nicht in der Verantwortung und im Bereich des Möglichen eines Webentwicklers. Später konnte die Chromium-Organisation feststellen, dass dieser Bug von der WKWebView im Apple WebKit ausging. [4]

### Hintergrundinformation:

Diese Vulnerability wird als sogenannter „Sandbox Escape“ eingestuft und würde von Mozilla im „*Bug bounty program*“ mit einer Summe von bis zu \$10,000 belohnt werden.

	High Quality Report	Baseline
Highest Impact		
<ul style="list-style-type: none"> <li>• UXSS<sup>0</sup> ←</li> <li>• Sandbox Escape<sup>1</sup> ←</li> <li>• Bypassing WebExtension install prompts<sup>2</sup></li> </ul>	\$10,000	\$8,000
High Impact	\$5,000	\$3,000
<ul style="list-style-type: none"> <li>• Memory Corruption</li> <li>• Same Origin Bypass leaking high-impact user data</li> <li>• Obtaining the user's IP address if a proxy is configured</li> <li>• Other vulnerabilities not fitting 'Highest Impact', but still receiving a <a href="#">sec-high rating</a></li> </ul>		
Moderate Impact, at the discretion of the committee	\$2,500 - \$500	Typically \$1000
<ul style="list-style-type: none"> <li>• Persistent-DOS of browser across restarts or a DOS requiring reboot of user's computer<sup>3</sup></li> </ul>		

Abbildung 5.3.21: „*Bug Bounty Program*“ von Mozilla. [31]

### 5.3.8 Sicherheitsmaßnahmen

Kritische Teile der Logikverarbeitung werden ausschließlich in serverseitigen PHP-Code abgearbeitet, wodurch keine Möglichkeit besteht für Angreifer zu intervenieren. Würde man gewisse Prozesse, wie z.B. das Eintragen in den Buchungskalender, Client-seitig abarbeiten, setzt man sich einen enormen Risiko aus. Abgesehen davon würden dadurch vermutlich früher oder später Komplikationen durch verschiedene Client-Software-Versionen entstehen, welche sich nicht ausbessern ließen. Kritische Prozesse wären z.B.:

- Prüfprozess der Login-Daten und ausstellen einer *sessionid*.
- Durchführen einer Buchung.
- Prüfen der Berechtigung bei Zugriff auf das Control Panel.

## 5.4 Webentwicklung

### 5.4.1 Prinzip

Nach erfolgreichem Login wird dem Browser eine *sessionid* mitgeteilt. Diese wird im *localStorage* clientseitig gespeichert und beim Aufruf jeder Seite an das Backend weitergegeben. Das Backend kann anhand der *sessionid* auf die Berechtigungsgruppe und andere personenbezogenen Daten des Nutzers zurückschließen. Versucht der Nutzer eine Seite aufzurufen, zu welcher er nicht berechtigt ist, wird die Anfrage direkt mit einer Fehlermeldung abgebrochen. Um den Zeitraum dieser Überprüfung für den Nutzer visuell darstellen zu können, wird standartmäßig bei Aufruf einer Seite der *loading*-Div angezeigt. Empfängt das Script nun eine Antwort des Servers, wird zunächst geprüft, welches der zwei Szenarien aufgetreten ist:

Die Response ist ein String:

Der *loading*-Div wird mit *display: none;* unsichtbar gemacht. Da die Response zugleich eine Fehlermeldung ist, wird der *error*-Div mit *display: block;* sichtbar gemacht und das entsprechende Textfeld mit der Fehlermeldung bestückt. Der Nutzer hat daraufhin einen Button, um zur Startseite zurückzukehren.

Da es viele zahlreiche unterschiedliche Fehlermeldungen geben kann, wäre es zu aufwendig diese speziellen Szenarien auf jeder einzelnen Unterseite zu implementieren. Daher wird nach zurückkehren auf die Startseite jene Fehlermeldung dort besonders behandelt. Handelt es sich z.B. um den Fehler „Ihre Sitzung wurde beendet“, wird dem Nutzer auf der Startseite der Button „Jetzt ausloggen“ angezeigt.

Die Response ist kein String:

In diesem Fall passiert zunächst visuell Garnichts. Nach Empfangen der Response wird diese seitenspezifisch Verarbeitet. Dabei kann es sich um ein einfaches Array handeln, oder auch um ein komplexes Dictionary. Nach erfolgreicher Datenverarbeitung wird der *loading*-Div versteckt und der mit Daten bestückte *content*-Div angezeigt.

### 5.4.2 Adaptierbarkeit

Das System wurde so entwickelt, dass es zu jedem Zeitpunkt an eine andere Domäne angepasst werden kann. Dadurch ist es auch ohne großen Aufwand möglich, die URL eines bestehenden Systems zu ändern oder die Software einen weiteren Carsharing-Verein zur Verfügung zu stellen. Dafür muss lediglich die im Rootfolder befindliche File *Configuration.php* editiert werden. In ihr verstecken sich die dynamischen Parameter *BaseURL*, *Firma*, *LogoURL*, *ThemeColor*, *FaviconURL* und *Footer* welche von jeder Seite im System abgefragt werden.

### 5.4.3 Konfigurierbarkeit

Damit der Administrator das System nach seinen Wünschen und Vorstellungen anpassen kann, gibt es 20 Systemeinstellungsmöglichkeiten und sieben pro angelegtem Fahrzeug.

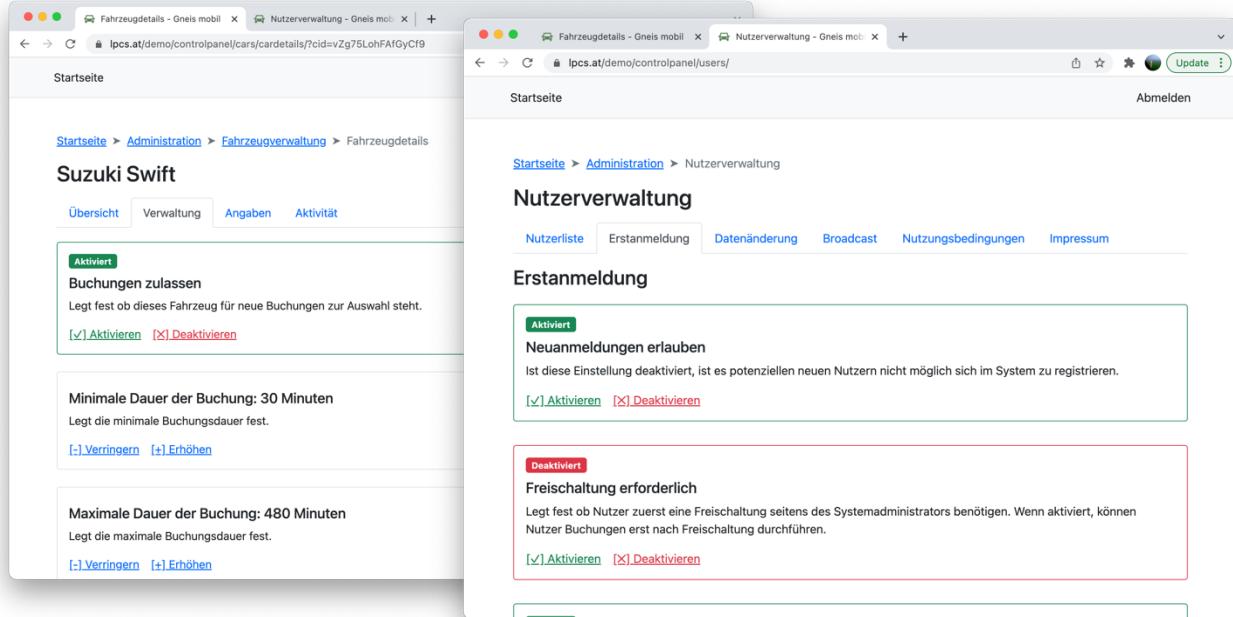


Abbildung 5.3.22: Einstellungsmöglichkeiten im LPCS Control-Panel.

Doch neben spezifischen Systemeinstellungen ist es auch von Relevanz die vom Gesetzgeber vorgeschriebene Offenlegungspflicht zu berücksichtigen. Deshalb sind das Impressum und die Nutzungsbedingungen-Seite in Form von Sektionen dynamisch erweiterbar, editierbar und formatierbar:

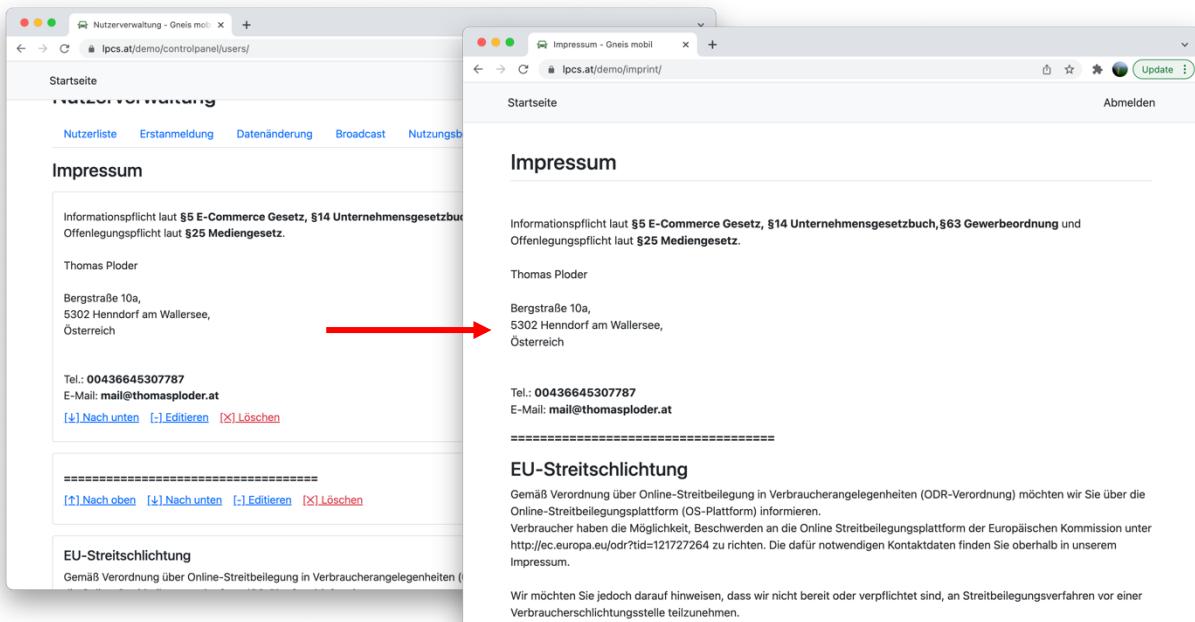


Abbildung 5.3.23: Konfigurierbarkeit des Impressums.

## 5.4.4 Besonderheiten

### 5.4.4.1 Startseite

Ein Verein ändert von Zeit zu Zeit seine Tarife sowie sein Angebot und das damit verbundene Auftreten gegenüber der Öffentlichkeit. Die Startseite prägt den ersten Eindruck eines potenziellen neuen Mitglieds und wird dadurch regelmäßig angepasst. Damit auch zukünftig der Verein für kleine Änderungen nicht auf das Wissen eines Softwareprofis angewiesen ist, welcher in den Tiefen der Server-Filestrukturen eine PHP-File abändert, wurde beschlossen, dies über das Control-Panel zu ermöglichen. Freigeschaltene Nutzer mit der Berechtigungsstufe *Administrator* besitzen dadurch die Möglichkeit, den auf der Startseite dargestellten Inhalt mit der Seitenbeschreibungssprache HTML zu editieren.

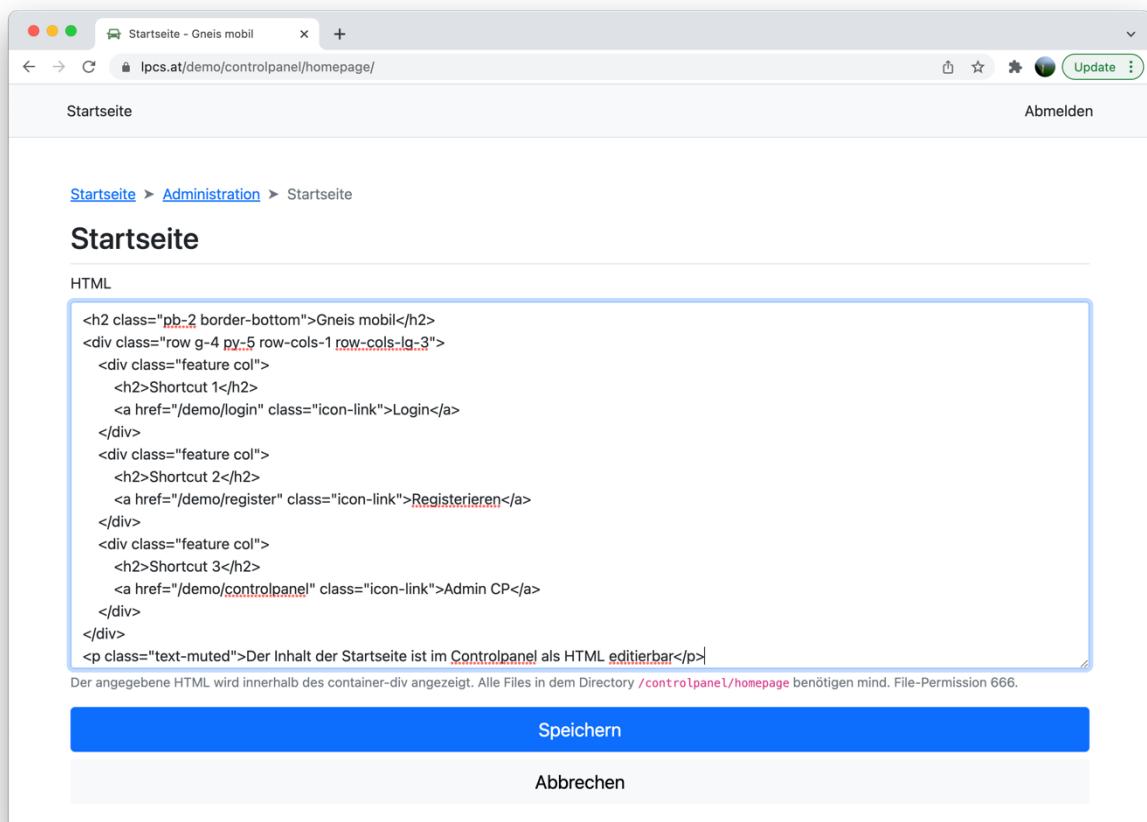


Abbildung 5.4.1: Screenshot aus dem Control-Panel

Die HTML-Sequenzen werden aufgrund der Datengröße sowie potenziellen Sondersymbolen allerdings nicht in einer Datenbank gespeichert, sondern lokal in einer HTML-File. Damit Änderungen vorgenommen werden können, ist es von Nöten dem anschließend aufgerufenen PHP-Skript mindestens *File-Permission 666* zu geben (CLI: „`chmod 666 Homepage.php`“). Diese Berechtigungsstufe legt fest, dass das File berechtigt ist, Lese- und Schreibvorgänge durchzuführen.

### 5.4.4.2 Nachrichtensystem

Der Systemadministrator steht im ständigen Austausch mit den Vereinsmitgliedern. Dabei ist es leicht möglich, den Überblick zu verlieren. Um einfach, schnell und zuverlässig alle Nutzer mit Informationen erreichen zu können, wurde ein systeminternes Kommunikationssystem entworfen. Mit dem Broadcast-Tools ist es dem Administrator im Control-Panel möglich, per direkter Nachricht, App-Notification oder Email einen gewissen Nutzer, eine gewisse Nutzergruppe oder alle Nutzer zu erreichen. Aber auch

vollautomatisiert werden diese Nachrichten entsendet. So wird bei einer Buchungsstornierung, Nutzerregistrierung und weiteren relevanten Interaktionen jeder Administrator über eine direkte Nachricht und App-Notification über das Ereignis informiert.

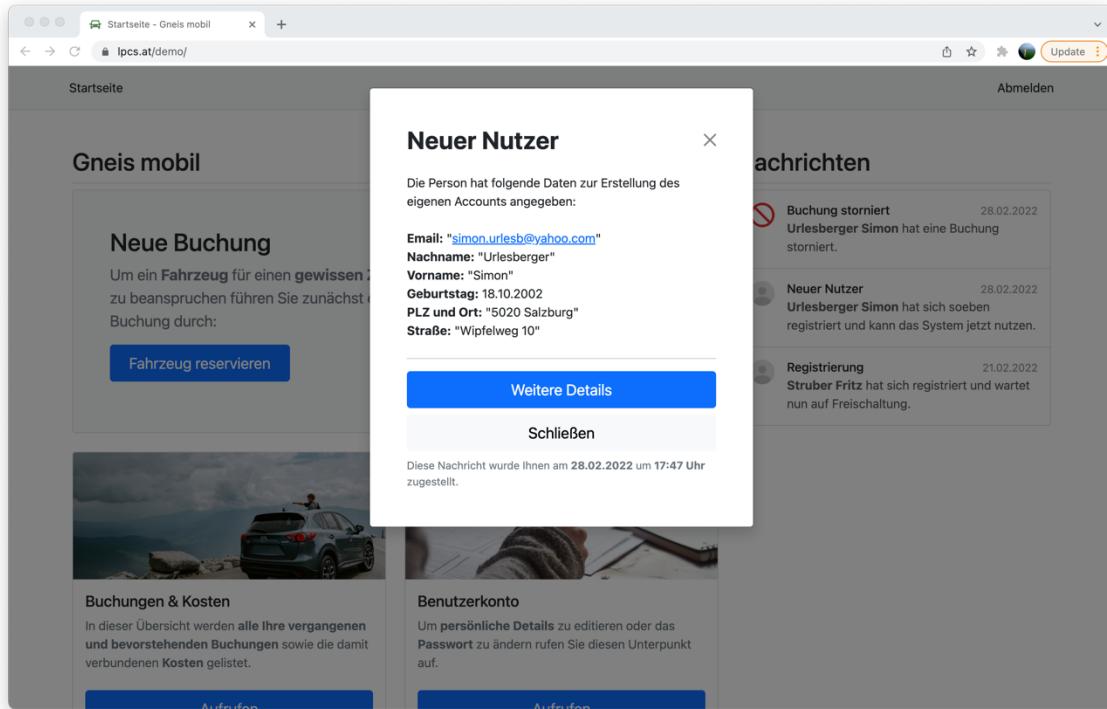


Abbildung 5.4.2: Beispiel einer systemgenerierten Nachricht.

Doch neben *messageid*, Zeitstempel, Überschrift, Vorschautext und Bodytext wurde auch eine Button-Action-URL generiert. Um das anschließende Einsehen zu vereinfachen, wurde direkt ein URL-Pfad mitgegeben, welche direkt in die Nutzerdetails im Control-Panel weiterleitet.

#### 5.4.4.3 Buchungsprozess und Fahrzeugkalender

Eine der wichtigsten Rollen im System spielt jene Seite, auf welcher man eine Buchung tätigen kann. Umso wichtiger war es, diese visuell besonders ansprechend und intuitiv zu gestalten.

Anfangs wurde eine tabellarische Ansicht entwickelt, in welcher man grafisch die belegten Zeiträume sichtbar macht. Nach ausgiebigen Tests stellte sich allerdings heraus, dass diese Art der Darstellung nicht intuitiv ist und für Verwirrung sorgt. Außerdem konnte diese Ansicht nicht entsprechend an kleine Displaygrößen angepasst werden. Daher wurde die Seite neugestaltet, beginnend mit der Wahl des Datums in einem Kalenderelement. Mit Klick auf das gewünschte Datum bewegt sich die blaue Fortschrittsanzeige vor und ein Ladesymbol bei „Details“ symbolisiert, dass das System eine Anfrage der verfügbaren Fahrzeuge an das Backend gesendet hat.

Die Reponse des Servers beinhaltet daraufhin alle an diesem Tag verfügbaren Fahrzeuge (eventuell ist eine Zeitraumsperre bei einem Fahrzeug aktiv => Fahrzeug steht nicht zur Auswahl). Nach Wahl des Fahrzeugs und der geschätzten Fahrstrecke werden die verfügbaren Uhrzeiten angezeigt. Dabei sind die fahrzeugspezifischen Einstellungen wie minimale und maximale Buchungsdauer berücksichtigt. Eine Auswahl einer Uhrzeit, welche zu knapp oder im Zeitraum einer bestehenden Buchung ist, ist somit nicht möglich. Auch der Puffer, welcher ebenfalls fahrzeugspezifisch eingestellt werden kann, hat Einfluss auf die zur Auswahl stehenden Uhrzeiten und kann vor allem für Fahrzeuge mit Elektroantrieb sinnvoll sein.

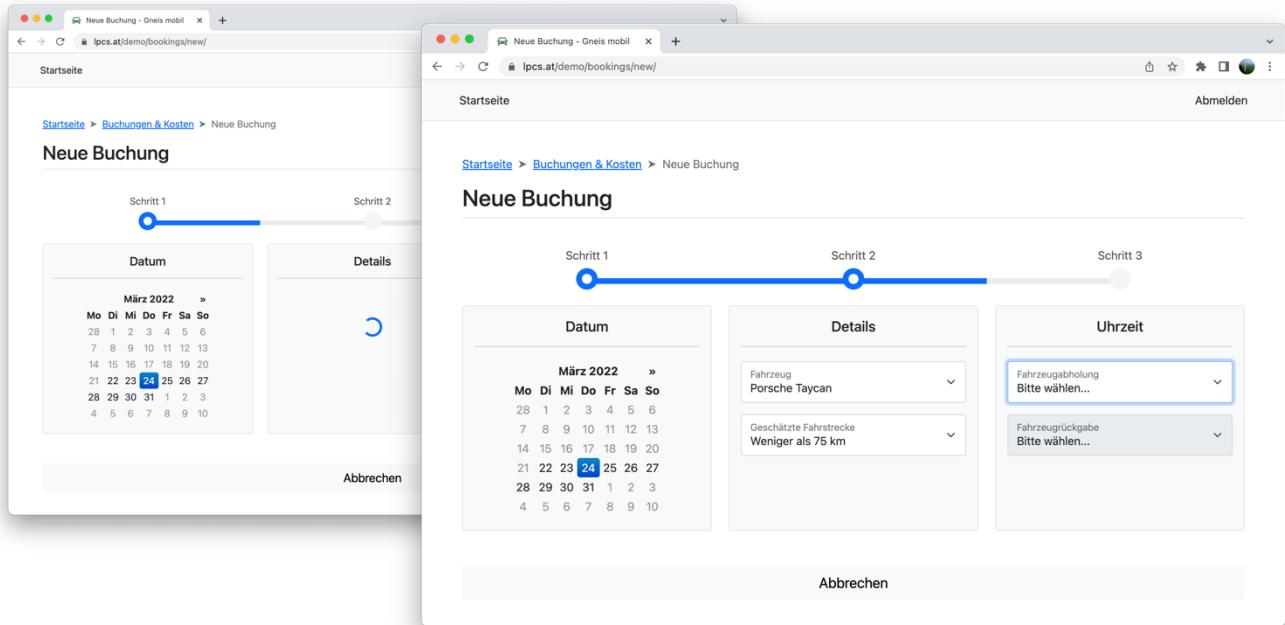


Abbildung 5.4.3: Prozess der Buchung.

#### 5.4.4.4 Finanzsystem

Es ist von großer Bedeutung, dass das systeminterne Finanzsystem keinen Ausfall erleidet. Dieser würde die Verrechnung der Mitgliedschaftsgebühr oder Kostenkalkulation der Buchungen betreffen. Daher wurde ein Scheduler vermieden, da dieser nach Serverneustart (z.B. Wartung oder Stromausfall) erneut gestartet werden müsste und dadurch fehleranfällig wäre. Daher wurde ein anderer Ansatz gewählt: Bei jedem Aufruf einer Seite, welche in Verbindung mit Buchungen steht, fungiert dieser als Trigger für die Überprüfung der Buchungen. Aus der Datenbank werden alle Buchungen abgefragt welche *null* in der Kostenpalte aufweisen. Stellt das System dabei fest, dass die Buchung bereits in der Vergangenheit liegt und abgeschlossen ist, wird eine Kalkulation der angefallenen Kosten angeregt. Dabei greift das System auf den Tarif des Nutzers zu, um Grundgebühr, Zeitgebühr und Kilometergeld mit den gefahrenen Kilometern zu evaluieren. Dieser Wert wird anschließend wieder in die entsprechende Spalte geschrieben und die Kalkulation ist abgeschlossen.

Dasselbe Prinzip wurde auch auf die Verrechnung der monatlichen Mitgliedschaftsgebühr angewandt. Ein Aufruf einer Seite im Bezug mit Nutzerdaten fungiert als Trigger für die Prüfung der zu verrechneten Kosten.

#### 5.4.4.5 Logging

Neben den im Control-Panel sichtbaren protokollierten Interaktionen, werden auch noch alle restlichen Seitenaufrufe des Nutzers festgehalten. Dabei werden diese Daten auf die *sessionid* des Nutzers bezogen, um im Falle einer Aufarbeitung genau feststellen zu können, von welchem Gerät der Aufruf stattfand. Um genauer Statistik führen zu können, werden alle beteiligten ID's (z.B.: *sessionid*, *userid*, *carid*, *bookingid*) mitprotokolliert, wodurch sich z.B. auf Browsetyp, Browsersversion, Betriebssystem, Appversion, Name des Geräts, FCM-Token, Device-UUID und einiges mehr schließen lässt.

## 5.5 Appentwicklung

Seit Jahren wird das Smartphone immer wichtiger – der Trend weg vom Desktop-PC hin zum Handy in der Hosentasche geht an keinem Spurlos vorbei. Das Smartphone bietet eine riesige Chance, in Form von Apps Millionen von Nutzer mit dem eigenen Dienst zu erreichen. Diese Chance möchten auch wir für unser LPCS Carsharing System nutzen, indem wir eine Anwendung für iOS- und Android-Geräte im jeweiligen Store den Nutzern mit einem einfachen Klick zum Download bereitstellen.

### 5.5.1 Aspekte der Appentwicklung

Vor jedem Start eines neuen Projekts muss man sich als Entwickler(team) die Frage stellen, mit welcher Technologie man sein Produkt umsetzen will. Hier muss man die Entscheidung zwischen der nativen- und der hybriden-Appentwicklung treffen. Je nach Qualitätsansprüchen, Entwicklungspräferenzen und verfügbarer Zeit gibt es hier verschiedene Vor- und Nachteile, welche man in Betracht ziehen muss. Die Wahl fiel nach Abwägen der folgend aufgeführten Gründe auf Swift mit Xcode für die iOS-Appentwicklung (UIKit), und Java mit Android Studio für die Android-Appentwicklung.

#### 5.5.1.1 Native iOS-Appentwicklung mit Objective-C

Die Entwicklung einer iOS-App mit der Programmiersprache Objective-C in Xcode bietet eine ausgereifte, allerdings etwas in die Jahre gekommene Plattform. Bereits seit Anbeginn der mobilen Appentwicklung gibt es Objective-C, was es zu einer der meistbenutzten Programmiersprachen im mobile-App Segment machte. Doch neben einer etwas gewöhnungsbedürftigen Syntax ist diese Programmiersprache bereits in die Jahre gekommen und durch die „Nachfolgerprogrammiersprache“ Swift ersetzt worden. Bestehende Obj.-C Apps lassen sich zwar nach wie vor Kompilieren und weiterentwickeln, allerdings wird seitens Apple empfohlen, bestehenden Code in Swift umzuschreiben. Objective-C wird nach und nach fallen gelassen – das fundamentale Framework „UIKit“, welches alle Userinterface Komponenten beherbergt, wurde bereits 2014 vollkommen von Objective-C zu Swift transformiert.

```
#import <Foundation/Foundation.h>

int main () {
    NSString *greeting = @"Greetings";
    NSLog(@"Willkommensnachricht: %@", greeting); // Willkommensnachricht: Greetings
    return 0;
}
```



Abbildung 5.5.1:  
Xcode Logo

Abbildung 5.5.2: Einfaches Code-Schnippel der Programmiersprache Objective-C

Heute ist die Entwicklung von Objective-C-Apps zwar noch möglich, allerdings nicht zukunftsorientiert.

### 5.5.1.2 Native iOS-Appentwicklung mit Swift (UIKit)

Die 2014 von Apple vorgestellte Programmiersprache Swift, welche als moderner Ersatz zu Obj.-C dienen soll, legte anfangs einen holprigen Start hin. Entwickler wurden in den ersten Jahren von zahlreichen Syntaxänderungen geplagt, welche nicht von großer Beliebtheit waren und sich bis heute auf Code-Schnipsel und stackoverflow-Antworten auswirken. Beim Einbinden mancher älterer Libraries ist es bis heute der Fall, dass man verbunden mit großem Arbeitsaufwand leichte Änderungen quer durch den ganzen Source-Code vornehmen muss, um den Syntax Compilergerecht zu machen.



Abbildung 5.5.3:  
Swift Logo

```
array.removeAtIndex(3); // Old Swift Syntax
array.remove(at: 3) // New Swift Syntax

NSJSONSerialization.JSONObjectWithData(data, options: []); // Old Swift Syntax
JSONSerialization.jsonObject(with: data, options: []) // New Swift Syntax
```

Abbildung 5.5.4: Code-Schnippel mit Vergleich von Swift 2 und Swift 3

Swift bietet eine große Community und einen modernen Syntax. Bestes Beispiel dafür ist das Handling von sogenannten „Optionals“.

```
// Declaration of opt. var. sessionid in AppHandler-class
var sessionid: String? {
    get { return UserDefaults.standard.string(forKey: "sessionid") } // Could be nil
    set { UserDefaults.standard.set(newValue, forKey: "sessionid") }
}
```

Abbildung 5.5.5: Deklaration einer Optional-Variable.

Jeder Datentyp kann bei der Deklaration mit einem „?“ versehen werden, was andeutet, dass der Datentyp unter Umständen „nil“ (bekannt als „null“ aus anderen Sprachen) annehmen kann. Der folgende Code-Ausschnitt aus der LPCS-iOS-App veranschaulicht das Handling mit Optionals, indem die Klasse Messages aufgerufen wird, welche die Nachrichten vom Server abfragt:

```
if let sessionid = AppHandler.instance.sessionid {
    Messages.instance.getMessages(sessionid: sessionid) { error, alerts, messages in
        if let error = error { // Error is declared as "String?" -> Could be nil
            print(error);
            // Further error handling
        }

        guard let messages = messages else { /* handle unknown error */ }

        self.state = .presentContent;
        DispatchQueue.main.async {
            // Do sth with messages array
            // If working with UI it's necessary to do it on the main-thread!
        }
    } else {
        // Handle user that is not logged in
    }
}
```

Abbildung 5.5.6: Veranschaulichung von Optionals.

Für Anfänger bietet Xcode in Kombination mit Swift die Möglichkeit „Storyboards“ zu nutzen. Dabei gestaltet man das UI wie in einem Editor und muss sich über Constraints nur wenig Gedanken machen. Arbeitet man allerdings mit dynamischen Content oder an einer größeren App wie es bei LPCS der Fall ist, ist es aufgrund der Organisation und Professionalität empfehlenswert, alles im Code oder mit „.xib“-Files umzusetzen.

### 5.5.1.3 Native iOS-Appentwicklung mit Swift (SwiftUI)

Bis vor wenigen Jahren waren macOS- und iOS-Apps noch strikt getrennt. Auf iOS wurde das UIKit benutzt und auf macOS das AppKit. Zwei Frameworks welche sich fundamental unterscheiden. Doch das Prinzip von Cross-Platform-Apps beschäftigte auch die Entwickler bei Apple und so wollte man die Möglichkeit bieten, eine App nicht nur auf dem iPhone laufen zu lassen, sondern auch auf einen Mac – nicht zuletzt da die neuen Generationen von Macs nicht mehr mit x86-, sondern mit ARM-basierten Prozessoren ausgestattet sind.

Auf der World-Wide-Developer-Conference 2019 stellte man also das Framework „SwiftUI“ vor, welches eine starke Ähnlichkeit zu Flutter und React Native aufweist. Mit dem UIKit arbeitet man mit wirklichen Objekten (z.B. UIButton, UILabel, UITextField, ...) – in SwiftUI hingegen arbeitet man mit Views in Layout-Strukturen.

```
struct User: Identifiable {
    let id = UUID()
    var name: String
    var isContacted = false
}

struct ContentView: View {
    @State private var users = [
        User(name: "Taylor"),
        User(name: "Justin")
    ]

    var body: some View {
        List($users) { $user in
            Text(user.name)
            Spacer()
            Toggle("User has been contacted", isOn: $user.isContacted).labelsHidden()
        }
    }
}
```

Abbildung 5.5.7: Beispiel einer Liste mit SwiftUI [5]

Objekte werden durch Structs ersetzt und die Logik wird versucht so weit wie möglich zu verdrängen. Mit diesem Programmierstil können sich viele iOS-Appentwickler, mich eingeschlossen, nicht anfreunden – was dazu führt, dass wenig über SwiftUI im Internet aufzufinden ist. Auf stackoverflow.com sind unter dem Framework-Tag „swiftui“ 24.500 Fragen angeheftet. Für das UIKit (bezeichnet als Swift) sind hingegen 310.500 Fragen angeheftet. Daraus lässt sich ableiten, dass für jeden erdenklichen Fehler eine Lösung im Internet auffindbar ist und es viele Experten auf dem Swift-UIKit-Gebiet gibt. Bei einer solch neuen Technologie wie SwiftUI ist dies noch nicht der Fall, was die Entwicklung erschwert.

### 5.5.1.4 React Native

Zeit ist Geld. Meta (ehemalig Facebook) stand vor der Frage, wie man kostengünstiger Apps entwickeln kann. Da man damals jede App doppelt, also separat für iOS und Android entwickelte, wollte man sich die Idee von Cross-Platform zunutze machen. Das Prinzip ist, Code einmalig zu schreiben und in nativen iOS- und Android-lauffähigen Code übersetzen zu lassen.

Man entwickelte somit basierend auf dem Framework React „React Native“. Mit JavaScript oder TypeScript geschriebener Skriptcode kann somit einfach übersetzt werden und wird auf iOS und Android mit nativen UI-Elementen dargestellt. Bereits viele bekannte Apps wie Facebook, Instagram, Discord, Skype oder der XXXLutz-Preisepass sind damit entwickelt worden.

Mit jährlichen Software-Updates von Betriebssystemen werden allerdings auch jährlich neue native Features wie Widgets, Haptic Feedback oder 3D-Touch eingeführt. Da diese OS-spezifisch sind, kann man diese Features unter React Native nur bedingt implementieren. Die Rede ist hier von Packages, welche von der Community bereitgestellt werden und man mittels NPM in das Projekt implementieren kann. Mit dem entsprechenden Code in JavaScript oder TypeScript wird dann ein natives Codesegment in Java bzw.

Swift oder Obj.-C aufgerufen. Das Übergeben von Parametern ist dabei etwas kompliziert – da ein Handshake zwischen einer Skriptsprache und einer kompilierten Software durchgeführt werden muss. Bei Packages ist man somit von den Qualitäten und Schnelligkeiten sogenannter „3rd-Paties“ abhängig. Doch hat man Zeit und vor allem genug Nerven, kann man auch selbst eines entwickeln: Da kein Package für Rich-Notifications (Notifications mit zusätzlichen Informationsdaten wie Bilder, Interaktionsmöglichkeiten usw.) zum damaligen Zeitpunkt auffindbar war, entwickelte ich im Zuge meines Praktikums bei einem Softwareunternehmen aus Henndorf jenes kurzerhand selber. Fazit: Das Entwickeln eines Packages ist zeitintensiv, sehr kompliziert und die Fehlersuche ist mit großem Aufwand verbunden.

Doch auch wenn React Native viele Vorteile bietet, zählt das Endprodukt. Und bei genauerer Analyse der Elemente kann man gewisse Qualitätsunterschiede auf iOS zu nativen Apps feststellen. Auch wenn React Native mit nativen Elementen wirbt, sind nur einige UI-Komponenten wirklich auf beiden Plattformen im Endeffekt nativ (Buttons, Labels, Textfelder und wenige weitere). Möchte man ein fundamentales Element wie einen NavigationController oder eine TableView auch auf Android implementieren, sucht man vergebens nach einem Pendant. Daher sind diese meist nur ein nachprogrammiertes, zu dem Original ähnliches Duplikat mit Qualitätsabstrichen.

```
import React from 'react';
import {Text, View} from 'react-native';
import {Header} from './Header';
import {heading} from './Typography';

const WelcomeScreen = () =>
<View>
  <Header title="Welcome to React Native"/>
  <Text style={heading}>See Your Changes</Text>
  <Text>
    Press Cmd + R inside the simulator to reload
    your app's code.
  </Text>
  <Text style={heading}>Debug</Text>
  <Text>
    Press Cmd + M or Shake your device to open the
    React Native Debug Menu.
  </Text>
  <Text style={heading}>Learn</Text>
  <Text>
    Read the docs to discover what to do next:
  </Text>
</View>
);
```

Abbildung 5.5.8: Einsteigerbeispiel in React Native. [6]

### 5.5.1.5 WebView-App

Eine WebView-App zu entwickeln ist ein kostengünstiges und einfaches Unterfangen, da keine besonderen Kenntnisse für die Entwicklung nativer Apps erforderlich sind. Es ist lediglich von Bedeutung, die anzuseigende Website optimal an kleine Gerätegrößen anzupassen (Responsiveness). Doch heutzutage werden aus zahlreichen Gründen keine WebView-Apps mehr entwickelt und angeboten:

#### 1. Sicherheit

Besteht eine App rein aus einer WebView, ist die Wahrscheinlichkeit, dass diese im AppStore Prüfverfahren abgelehnt wird, ziemlich hoch. Grund dafür ist, dass nicht garantiert werden kann, dass der angezeigte Inhalt zu einem späteren Zeitpunkt derselbe ist. Eine Weiterleitung zu einer unseriösen Website, welche Malware beinhaltet oder versucht Nutzer zu täuschen, wäre dem Betreiber ohne weiteres möglich.

## 2. Geräteanpassung

Den vollen Umfang möglicher Hard- und Softwarefeatures zu nutzen ist mit einer WebView nur bedingt machbar. Das Implementieren von Widgets, Haptic Feedback, Gestures und vieles mehr ist nur mit großem Aufwand möglich – da ein Handshake zwischen Website und einem nativen Codesegment erfolgen muss.

## 3. Offline-Funktionalität

Möchte ein Nutzer das Produkt offline nutzen, ist dies mit WebViews leider nicht möglich, da diese eine aktive Internetverbindung voraussetzen. Die App wäre somit in einem Funkloch funktionslos.

## 4. UI & UX

Auch wenn eine reine WebView-Anwendung den Zweck einer App erfüllt, bietet diese nicht ansatzweise das Feeling wie es ein normaler Nutzer erwartet. Die üblichen Animationen, Interaktionsmöglichkeiten und Gesten werden bei einer WebView nicht ansatzweise mit jene einer nativ entwickelten App vergleichbar sein.

### 5.5.2 Qualitätsmerkmale

#### 5.5.2.1 App-Icon

Eine gute Gestaltung bzw. Wahl des App-Icons ist essenziell für das Image des anzubietenden Produkts. Das App-Icon ist das Gesicht der App. Es soll nicht nur unter den vorgeschlagenen Apps auf der AppStore-Startseite, sondern auch in der Suche und im Store-Eintrag selbst, sowie in der Liste aller Apps hervorstechen – um nicht nur den ersten Eindruck, sondern auch den Wiedererkennungswert maßgeblich positiv zu beeinflussen.

Aus psychologischer Sicht gibt es eine Reihe an Merkmalen, wie beispielsweise die Farbe, welche man mit gewissen Eigenschaften assoziiert. Das gesättigte Grün bei LPCS wurde bewusst gewählt, da man diese Farbe mit Klarheit, Frische, Offenheit, Bewusstsein und Erholung unterbewusst in Verbindung setzt. [7]

Aber auch das Icon an sich sollte aussagekräftig sein und mit dem angebotenen Dienst in Verbindung stehen. Das Icon ist flat, um modern und poppig zu wirken. Da das Exportieren aus Grafiktools wie Photoshop für ein App-Icon mit verschiedenen Dimensionen aufwändig ist, empfehle ich das Onlinetool Appicon.co.

#### 5.5.2.2 Feedback

Interagiert ein Nutzer mit einer Software auf einem Bildschirm, ist es besonders wichtig den Nutzer den Eindruck zu verleihen, dass die Software den Input registriert hat (und ggf. diesen gerade verarbeitet). Bestes Beispiel für fehlendes Feedback sind Soft-Touch-Geräte, welche kein akustisches und visuelles Signal von sich geben. Als (unerfahrener) Nutzer fragt man sich nach berühren, ob das zu bedienende System den Input überhaupt erkannt hat und klickt oft hektisch und genervt noch einmal.

In Apps ist das Laden von Content oft mit Wartezeiten, abhängig von der Netzwerkverbindung und der Serverkapazität, verbunden. Um diesen Prozess für den Anwender so intuitiv wie möglich zu gestalten, wird hierbei auf Haptic Feedback der in den iPhones befindlichen *Taptic Engine* und Loading/Progress Indicators zurückgegriffen.

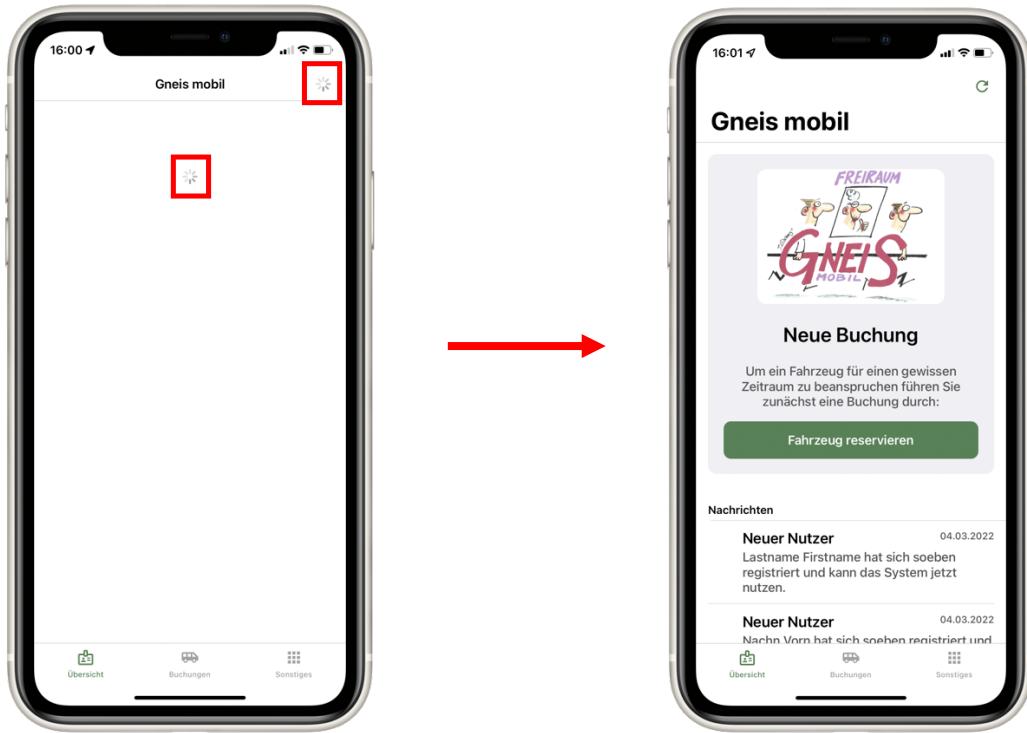


Abbildung 5.5.9: Ladeanimation in der Übersicht.

Tritt aus irgendeinem Grund ein Fehler beim Laden auf, wird dies dem Nutzer mit einem Alert klar kommuniziert. Zum Zeitpunkt, wo dieser eingeblendet wird, wird zeitgleich ein leichtes haptisches Vibrieren durch die *Taptic Engine* durchgeführt. Der Hinweis verschwindet erst, wenn der Nutzer jenen aktiv mit Klick auf „Schließen“ zur Kenntnis nimmt.

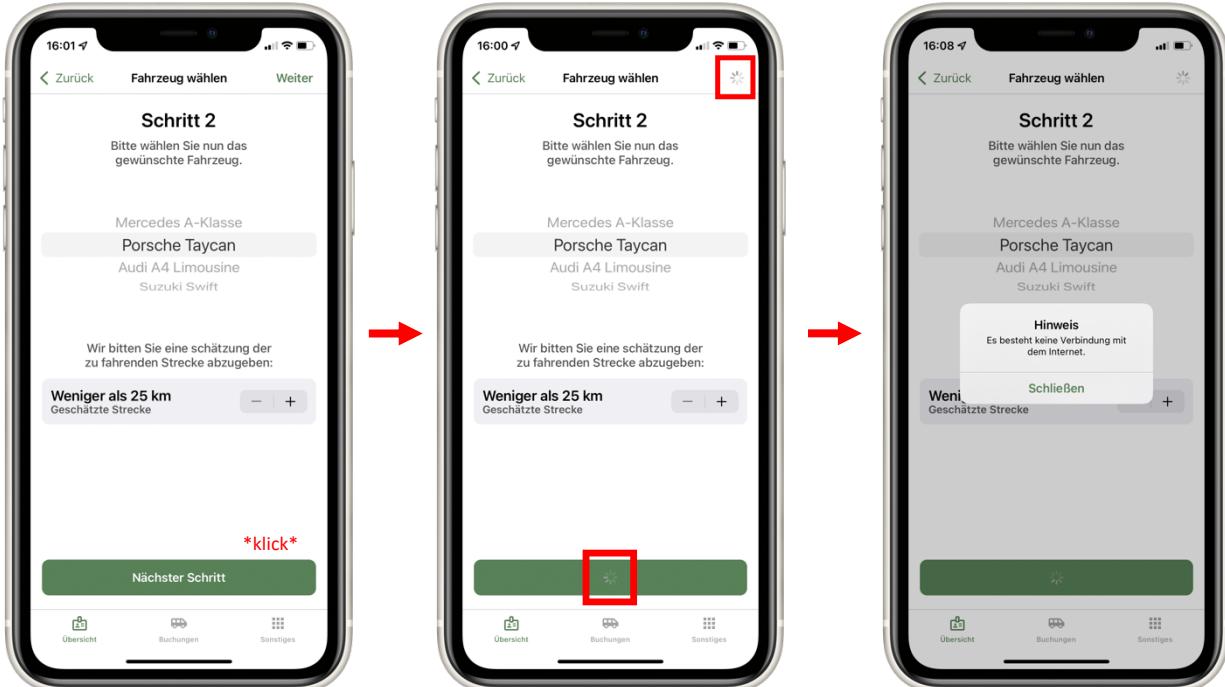


Abbildung 5.5.10: Veranschaulichung von Fehlern beim Laden.

### 5.5.2.3 Gestures

Jedes Betriebssystem bringt seine eigene Bedienweise mit sich. Nutzer können nicht sichtbare Wischaktionen intuitiv durchführen, da es an jeder Stelle im System gleich funktioniert. Ein klassisches Beispiel dafür ist der 3-finger-swipe nach links oder rechts am Mac.

Auf iOS ist das Wischen vom linken Bildschirmrand zur rechten Seite immer mit einer Animation zurück zur vorherigen Seite verbunden – Systemweit. Diese universellen Bedienungen sind für Nutzer nicht wegzudenken und müssen unbedingt in eine App implementiert werden.

Auf Android sind solche Gesten nicht so sehr vertreten, allerdings gibt es auf alten Android-Geräten (z.B. Samsung Galaxy S6) Zurück-Buttons neben dem physischen Home-Button. Auch diese müssen softwaremäßig in der entsprechenden Activity implementiert werden:

```
@Override
public boolean onSupportNavigateUp() {
    super.onBackPressed();
    return true;
}
```

Abbildung 5.5.11: Beispiel einer Implementierung in Java

Aber auch die Interaktion mit UI-Elementen beherbergt nicht sichtbare, aber dennoch intuitive Möglichkeiten. So ist auf iOS bei einer *UITableViewCell* oft die Option jene nach rechts oder links zu wischen, um weitere Funktionen hervorzurufen.

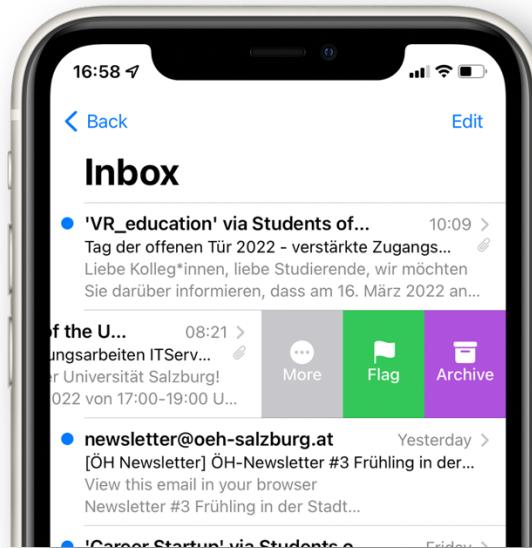


Abbildung 5.5.12: Veranschaulichung von *UITableViewCellSwipeActions*.

Auch Long-Press-Gestures sind auf Android als auch auf iOS weit verbreitet, um weitere Dialogmöglichkeiten zu zeigen. Wenn der Nutzer in der LPCS-App einen Long-Press auf seine Buchungsnummer macht, könnte ein Fenster mit Buchungsdetails eingeblendet werden, ohne dass der Nutzer diese Seite extra aufrufen muss. Da die LPCS-App allerdings vermehrt von älteren Personen bedient wird, welche oftmals mit jenen scheinbar intuitiven Gesten nicht so vertraut sind, wurde auf eine Implementierung aus zeitlichen Gründen verzichtet.

#### 5.5.2.4 Dark Mode und High Contrast

Seit wenigen Jahren hält bei den gängigsten Betriebssystemen ein systemweiter „dunkler Modus“ Einzug. Das Feature erfreut sich auch bei iOS und Android zunehmender Beliebtheit und so adaptieren Entwickler nach und nach ihre Apps, um auch im dunklen Design überzeugen zu können.

Wenn man die Farbwahl nicht statisch, sondern in Kombination mit den xcassets realisiert, bietet Xcode ganz einfach, sogar ohne nur eine Zeile Code schreiben zu müssen, die Möglichkeit die bestehende Farbpalette auf das Dark-Mode-Schema speziell zu erweitern. Aber auch auf Menschen mit einer Sehschwäche sollte man als Entwickler Rücksicht nehmen: Die Farbpalette ist ebenfalls einfach erweiterbar auf *High Contrast*-Einstellungen für Personen mit Sehbeeinträchtigungen. Für hohen Kontrast werden üblicherweise stärker gesättigte Farben mit weniger Transparenz gewählt.

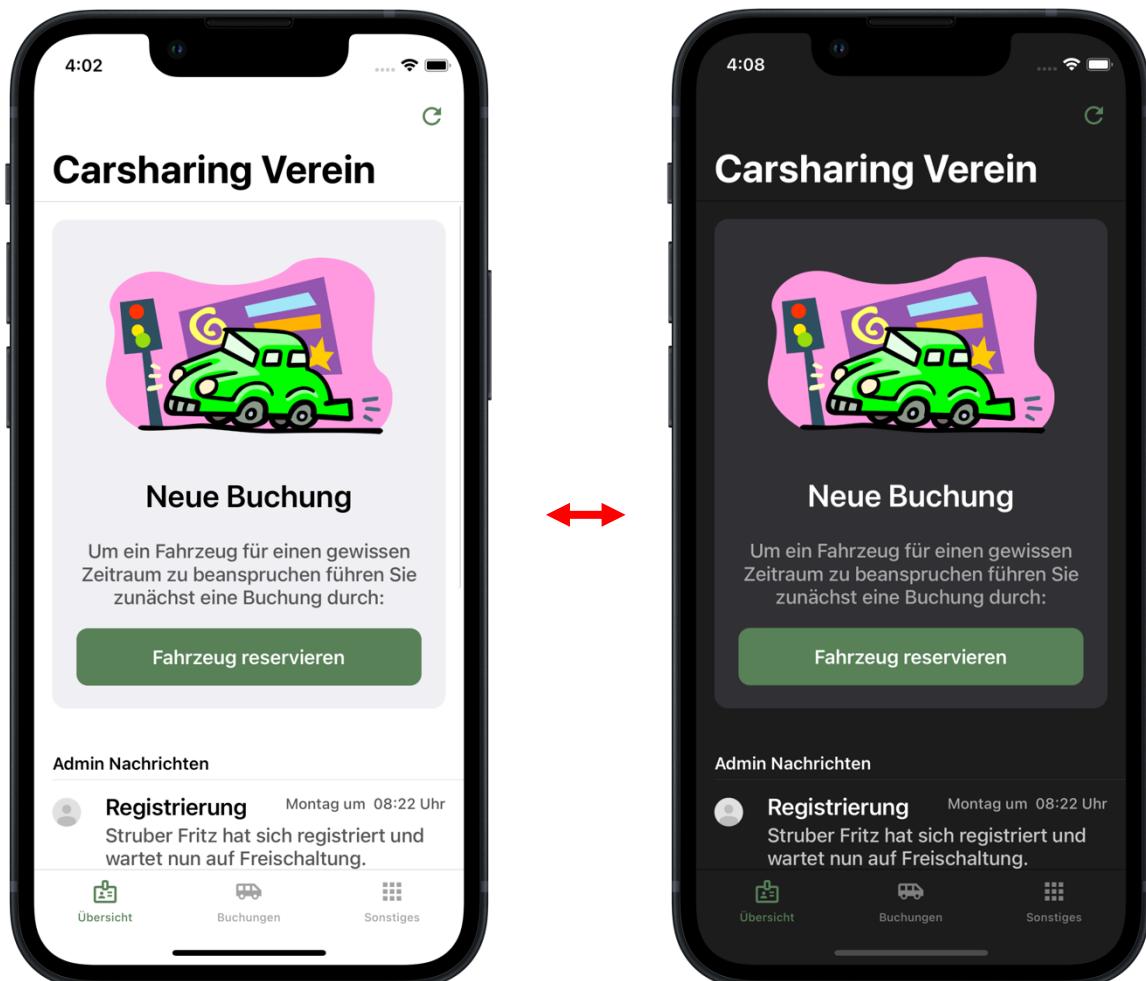


Abbildung 5.5.13: Gegenüberstellung Light- und Dark-Mode der LPCS iOS-App.



Abbildung 5.5.14: Farbpaletteneinstellung einer definierten Farbe in Xcode.

### 5.5.2.5 Versionsbezeichnung

Jeder App-Release wird mit einer Appversion und einer Buildnummer versehen. Dabei beruht die Wahl auf dem Prinzip „Semantic Versioning“, welches gefolgt lautet: **MAJOR.MINOR.PATCH.** [8]

1. MAJOR wird erhöht, wenn API-inkompatible Änderungen veröffentlicht werden,
2. MINOR wird erhöht, wenn neue Funktionalitäten, welche kompatibel zur bisherigen API sind, veröffentlicht werden, und
3. PATCH wird erhöht, wenn die Änderungen ausschließlich API-kompatible Bugfixes umfassen.

### 5.5.3 Push-Notifications

Um Nutzer mit aktuellen Informationen am laufenden zu halten, wurden bereits 2009 mit iOS 3.0 sogenannte Push-Notifications eingeführt. Der Unterschied zu herkömmlichen Notifications besteht darin, dass Push-Notifications nicht vom Gerät selbst getriggert werden, sondern von einer serverseitigen Applikation. Damit ist es dem Backend zu jedem beliebigen Zeitpunkt möglich, dem Nutzer eine Notification zu senden. Die kleinen Nachrichten, welche im Benachrichtigungszentrum am Sperrbildschirm oder in der oberen Bildschirmhälfte auftauchen, bieten in herkömmlicher Form zwei Attribute in Form von Title und Body.



Abbildung 5.5.15: Veranschaulichung einer (Push-)Notification.

#### 5.5.3.1 Rich-Push-Notifications

Eine erweiterte Form von (Push-)Notifications sind die Rich-(Push-)Notifications. Durch gezielte Erweiterung der Payload ist es möglich, ein Vorschaubild, einen anderen Sound, Icon-Badges oder eine andere Vibration einzustellen. Mit Extensions ist es auf iOS außerdem möglich, QuickReply oder ActionButtons zu implementieren. Letztere erfordern allerdings native Codesequenzen in Form von Targets. Für die LPCS-App bestand kein Bedarf Notifications auf Rich-Push-Notifications aufzuwerten.

### 5.5.3.2 Das Prinzip



Abbildung 5.5.16: Veranschaulichung von Push-Notifications

Damit Push-Notifications funktionieren, sind drei Teilnehmer erforderlich:

#### 1. Empfänger:

Der Empfänger fragt beim zuständigen Push-Notification-Dienst einen Token an (1a/1b). Unter iOS wird dieser Token durch die **Apple Push Notification Services** ausgestellt (2b). Unter Android, einem Google Betriebssystem, gibt es seit geraumer Zeit nur mehr die Möglichkeit Push-Notifications mit einbinden der Google Firebase zu realisieren. Daher wird Android-Geräten direkt ein FCM-Token (**Firebase-Cloud-Messaging-Token**) ausgestellt (2a). Damit der Entwickler nicht zwischen APNs- und FCM-Token unterscheiden muss, generiert die Firebase CloudMessaging-Library, welche in der iOS-App eingebunden sein muss, ebenfalls einen FCM-Token. Die Firebase-Dienste können aus diesem FCM-Token einen APNs-Token ableiten.

Der Empfänger hat allerdings auch die Möglichkeit sich in „Topics“, einer Art Verteiler der Firebase, einzuschreiben. Sendet der Server eine Notification an ein Topic, wird diese allen Geräten zugestellt, welche sich in jenes Topic eingeschrieben haben. Dadurch wäre es möglich, eine bestimmte Nutzergruppe zu erreichen welche spezifische Interessen aufweist (Target Audience).

#### 2. Firebase und Apple Push-Notification-Services (APNs):

Der Firebase Cloud-Messaging-Dienst, welcher in der Firebase-Console aktiviert werden kann, bietet die Möglichkeit, den Client FCM-Tokens auszustellen oder ihn mit Hilfe dieser mit einer Payload bis zu 4000 Bytes direkt zu benachrichtigen. Dabei muss nicht zwingend eine Notification gesendet werden, es könnte auch ein App-Interner Background-Task angeregt werden.

Die Firebase nimmt Entwicklern den Mehraufwand der serverseitigen Konfiguration für den Apple Push-Notification-Service. Aus Sicht des Servers ist es somit völlig irrelevant, ob der Token einem iOS- oder Android-Gerät zugeordnet ist.

cM8i4Pif24c:APA91bFch0dNZi1pLBV\_pmxbLPBkseCParexNS6wTqWxu7yCAR6o4M0p\_6I\_YP4\_SjEWZqo5azs7Vnm3BpkFU  
NC-dP5GLmLF\_r0grg251mnF45zVf5QheEjaR5kSSWV5XmbxdJcGEYpu

Abbildung 5.5.17: Beispiel für einen FCM-Tokens (152 Symbole).

### 3. Server:

Damit man einen Nutzer zu jedem beliebigen Zeitpunkt benachrichtigen kann, muss natürlich der FCM-Token bekannt sein. Empfängt die App jenen von der Firebase(-Library), wird dieser direkt dem Server mitgeteilt und in einer Datenbank gespeichert. Allerdings ist zu beachten, dass ein FCM-Token auch ungültig werden kann, wenn der Nutzer zum Beispiel die App deinstalliert, neu installiert oder der App die Berechtigung für Benachrichtigungen entzieht. Der Server kann dies erst feststellen, wenn er versucht eine Notification zu senden und eine ungültige Response bekommt.

#### 5.5.3.3 Hook Model

Junge Menschen schauen durchschnittlich 56-mal am Tag auf ihr Mobiltelefon. Dieses Potential kann man nutzen, zum Beispiel wenn man als Systembetreuer oder -entwickler feststellt, dass die Nutzerbindung bzw. Anzahl der Interaktionen in der App abnimmt. Mit Notifications kann man gezielt entgegenwirken, und Anwender zum Öffnen der App animieren. Dies ist zugleich der Einstieg in das Hook Model, wo eine Notification am Handy als externer Trigger fungieren kann:

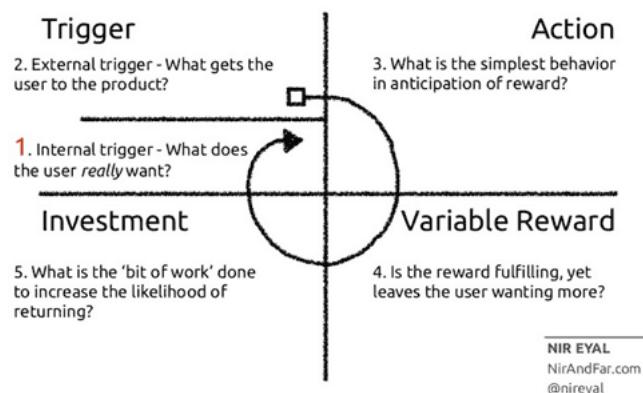


Abbildung 5.5.18: Das Hook-Modell.

Notifications sind also Auslöser für Nutzerinteraktionen. Beliebte Inhalte wären z.B.: Sportereignisse, Wetterberichte, Nachrichten oder Verkehrsinfos [9].

In manchen Artikeln wird aber auch genannt, dass das Kaufverhalten von Nutzern durch Werbung in Notifications angeregt werden kann. Doch dieser Punkt ist mit Vorsicht zu genießen, denn in den AppStore-Guidelines ist dies ausdrücklich verboten und kann einen Ausschluss der App aus dem AppStore bewirken:

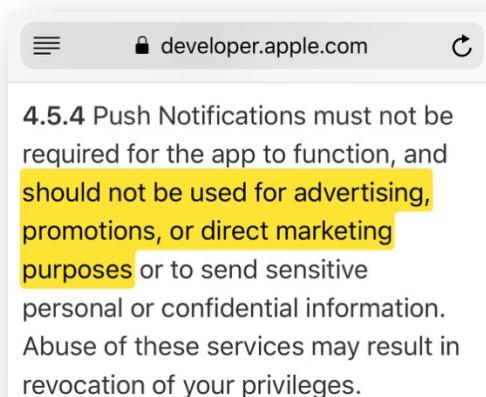


Abbildung 5.5.19: Ausschnitt aus den AppStore-Guidelines [10].

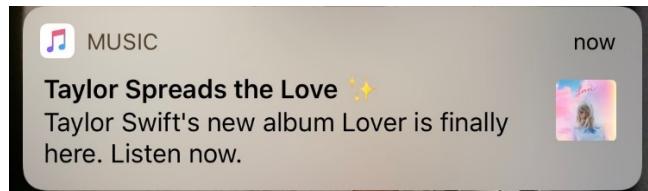


Abbildung 5.5.20: Apple verstößt gegen die eigenen Richtlinien.

Auch wenn Apple die AppStore-Guidelines mit harter Faust durchsetzt, neigt man beim hauseigenen Dienst „Apple Music“ anscheinend dazu, ein Auge zuzudrücken, wie Abbildung 5.5.20 veranschaulicht. [10]

#### 5.5.3.4 SDK einbinden

Nach Anlegen des Projekts in der Google Firebase Console muss das Firebase/Core- und Firebase/Messaging-SDK (Software Development Kit) in das Android Studio- bzw. Xcode-Projekt hinzugefügt werden.

##### Implementieren in das Xcode-Projekt:

Da die Firebase nach wie vor nicht über den Swift Package Manager implementiert werden kann, muss auf das traditionelle Verfahren mit dem Depenencymanager CocoaPods zurückgegriffen werden. Dieser kann über Homebrew auf macOS installiert werden. Um das Projekt zu initialisieren, muss im Projektfolder folgender Befehl im CLI ausgeführt werden:

```
pod init
```

Abbildung 5.5.21: Initialisieren einer CocoaPod-File.

Daraufhin wird eine sogenannte Podfile im Directory generiert, welche für CocoaPods relevante Informationen wie die einzubindenden Libraries enthält. Bei der LPCS-iOS-App wurde zusätzlich der Firebase Analytics-Dienst und die Library *SwiftyJSON* implementiert:

```
target 'lpcs' do
  use_frameworks! /* important for 3rd-party-libraries */

  pod 'Firebase/Core' /* implement firebase core sdk */
  pod 'Firebase/Analytics'
  pod 'Firebase/Messaging'
  pod 'SwiftyJSON' /* swiftyjson library */

end
```

Abbildung 5.5.22: Inhalt der „Podfile“.

Die angegebenen Libraries werden anschließend in das Projekt hinzugefügt mit:

```
pod install
```

Abbildung 5.5.23: Installieren der in der Podfiles gelisteten Libraries.

Innerhalb weniger Minuten generiert CocoaPods eine neue Xcode-Projektfile mit der Extension *.xcworkspace* welche fortan statt der ursprünglichen *.xcodeproj*-File geöffnet werden muss. Eine Workspace-Projektfile berücksichtigt beim Kompilieren und Arbeiten in der IDE die zuvor installierten Dependencies. Arbeitsabläufe bleiben in der IDE unverändert.

Da die Firebase nun erfolgreich zum Projekt hinzugefügt wurde, gilt es nun diese zu konfigurieren. Dabei muss in der Firebase Console die iOS-App mit dem Bundle-Identifier hinzugefügt werden. Man erhält dadurch die *GoogleServices-Info.plist*-Konfigurationsdatei, welche in das Xcode-Projekt implementiert werden muss (siehe grüner Pfeil in Abbildung 5.5.24). Dieses File beherbergt jede Menge wichtiger Informationen (API-Key, Client-ID, Project-ID, ...) um den Austausch mit den Firebase-Servern zu ermöglichen.

Beim Hinzufügen einer Datei in ein Xcode-Projekt gilt es zu vermeiden, dass nur referenziert wird, da dies zu Problemen bei Teamarbeiten am Projekt führen kann. Deshalb ist es notwendig, den Hacken bei *Copy items if needed* und dem richtigen Projekt-Target zu setzen.

Um die APNs-Dienste nutzen zu können ist es nun notwendig die App für jene zu registrieren. Der erste Schritt dazu ist es, die App entsprechend vorzubereiten. Dafür muss die Capability *Push Notifications* und *Background Modes* hinzugefügt werden. Dadurch ist die App in der Lage, Push-Notifications zu empfangen.

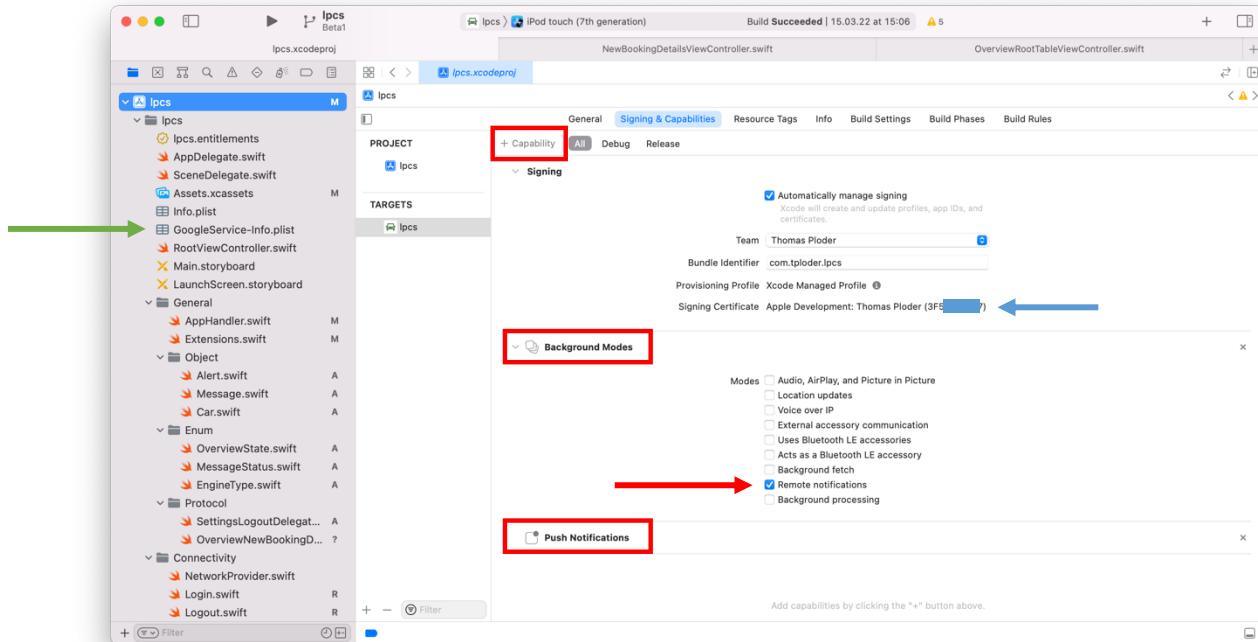


Abbildung 5.5.24: Projektkonfiguration.

Um die nächsten Schritte überhaupt durchführen zu können, ist es notwendig, dass die App im Apple-Developer-Portal des jeweiligen Entwicklers mit einer kostenpflichtigen Developer-Mitgliedschaft mit dem Bundle-Identifier registriert ist (siehe blauer Pfeil in Abbildung 5.5.24).

Damit die Firebase später Notifications beim APNs anregen kann, wird ein .p8-Zertifikat benötigt, welches unter [developer.apple.com](https://developer.apple.com) im Untermenü „Keys“ für die jeweilige App ausgestellt werden kann. Dieses Zertifikat beinhaltet einen private-Key und ist daher nur einmalig herunterladbar und darf unter keinen Umständen in fremde Hände gelangen!

The screenshot shows the 'Certificates, Identifiers & Profiles' section of the Apple Developer portal. A key named 'LPCS' is being configured. The 'Apple Push Notifications service (APNs)' checkbox is checked and highlighted with a red box. Other options like 'DeviceCheck', 'MapKit JS', and 'Media Services' are listed but not selected. A 'Continue' button is visible at the top right.

Abbildung 5.5.25: Projektkonfiguration.

Das heruntergeladene Zertifikat muss nun in der Firebase Console zum zuvor hinterlegten iOS-Projekt hinzugefügt werden. Dadurch ist es der Firebase möglich, Notifications an die App zu senden.

The screenshot shows the 'Projekteinstellungen' (Project Settings) screen in the Firebase console. The 'Cloud Messaging' tab is selected. Under 'Apple-App-Konfiguration' (Apple App Configuration), there is a table for 'Apple-Apps'. One entry for 'LPCS IOS com.tploder.lpcs' is shown. In the 'APNs-Authentifizierungsschlüssel' (APNs Authentication Key) section, a file icon labeled 'APNs-Authentifizierungsschlüssel' is highlighted with a red box. Below it, 'Schlüssel-ID' and 'Team-ID' fields are shown, along with a 'Löschen' (Delete) button.

Abbildung 5.5.26: Firebase Console.

Da nun alle formellen Schritte erledigt sind, gilt es noch den NotificationService in der App entsprechend aufzusetzen. Dafür wird die *AppDelegate.swift* (vergleichbar mit *build.gradle* unter Android) um folgendes erweitert:

```

import UIKit;
import Firebase; /* Import Firebase */
import UserNotifications; /* Import UN */
import SwiftyJSON;

@main
class AppDelegate: UIResponder, UIApplicationDelegate, MessagingDelegate,
UNUserNotificationCenterDelegate { /* Add corresponding delegates to handle events! */

    func application(_ application: UIApplication, didFinishLaunchingWithOptions launchOptions:
[UIApplication.LaunchOptionsKey: Any]?) -> Bool {
        FirebaseApp.configure(); // Configure with previously added GoogleServices-Info.plist */
        Messaging.messaging().delegate = self; // Assign delegate
        Messaging.messaging().subscribe(toTopic: "global"); // Subscribe to sample topic

        UNUserNotificationCenter.current().delegate = self; // Assign delegate

        let authOptions: UNAuthorizationOptions = [.alert, .badge, .sound];
        UNUserNotificationCenter.current().requestAuthorization(
            options: authOptions,
            completionHandler: { _, _ in }
        ) // Request permission to send notifications

        application.registerForRemoteNotifications(); // Trigger APNs registration

        return true;
    }

    /* ... */

    func messaging(_ messaging: Messaging, didReceiveRegistrationToken fcmToken: String?) {
        AppHandler.instance.fcmToken = fcmToken; // Could be nil!
        print("=====");
        print(fcmToken); /* FCM-Token! Send this to the server! */
        print("=====");

        /* ... */
    }

    func application(_ application: UIApplication, didReceiveRemoteNotification userInfo: [AnyHashable
: Any], fetchCompletionHandler completionHandler: @escaping (UIBackgroundFetchResult) -> Void) {
        /* Handle push-notification if user is In-App */
        print("=====");
        print("Received Message:");
        print(userInfo); /* Be careful with optionals (dict any:any)! */
        print("=====");

        /* ... */
    }
}

```

Abbildung 5.5.27: Ausschnitt der *AppDelegate*.

Da nun alle notwendigen Schritte gesetzt sind, um Push-Notifications auf iOS empfangen zu können, ist der letzte Schritt die Genehmigung vom Nutzer einzufordern. Dies ist dynamisch und kann je nach App zu einem individuellen Zeitpunkt geschehen. Klickt der Nutzer auf „Erlauben“ können dem Nutzer jederzeit entsprechend Notifications gesendet werden.

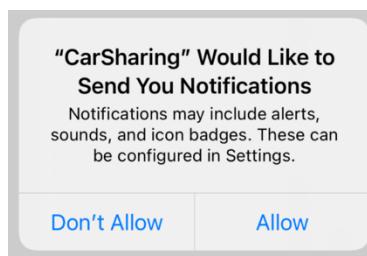


Abbildung 5.5.28: Dialogfenster zu Notifications.

### Implementieren in das Android Studio-Projekt:

Etwas einfacher hingegen ist das Einbinden unter Android mit Android Studio. Es ist weder eine Mitgliedschaft noch das Nutzen externer Tools notwendig. Navigiert man unter den Menüpunkt „Tools“ auf „Firebase“, öffnet sich im Assistance-Fenster die Konfiguration des Projekts mit der Firebase. Hier ist lediglich ein Klick auf „Set up“ mit anschließenden Google Login notwendig, um die Android Application in der Firebase Console zu hinterlegen.

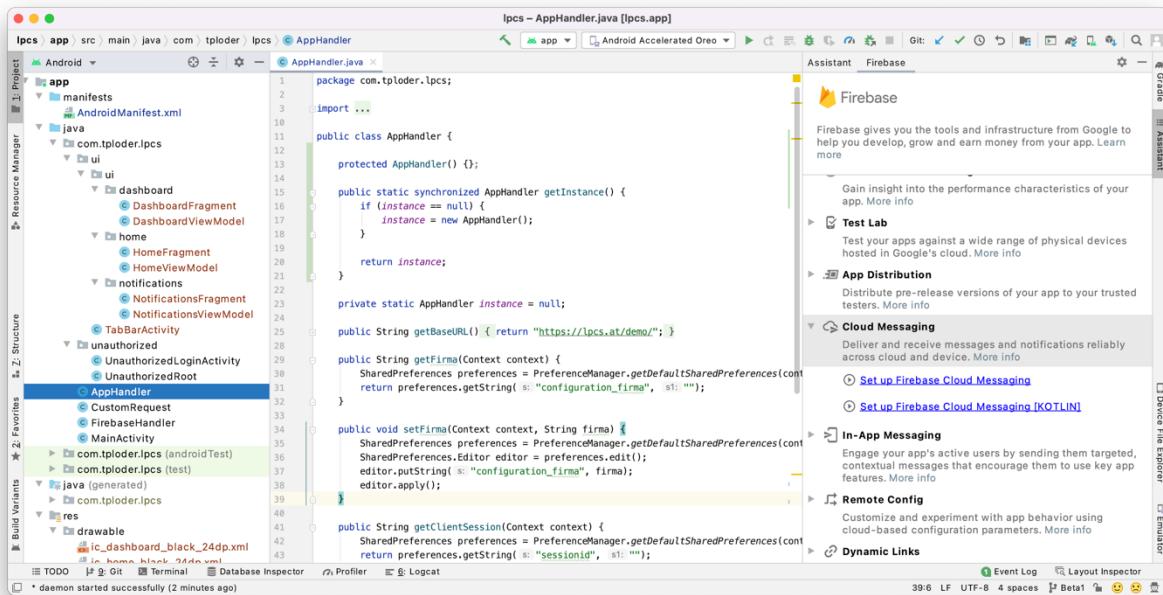


Abbildung 5.5.29: Firebase in Android Studio-Projekt integrieren.

Damit auch die Anwendung in der Lage ist Notifications zu empfangen, ist im *AndroidManifest.xml* unter *application* folgende Codesequenz einzufügen:

```
<service
    android:name=".java.MyFirebaseMessagingService"
    android:exported="false">
    <intent-filter>
        <action android:name="com.google.firebase.MESSAGING_EVENT" />
    </intent-filter>
</service>
```

Abbildung 5.5.30: Ausschnitt aus der *AndroidManifest.xml*-File.

Um den FCM-Token allerdings entsprechend an den Server zu senden, wurde noch die Klasse `FirebaseHandler.java` ins Leben gerufen:

```
package com.tploder.lpcs;

import com.google.firebase.messaging.FirebaseMessagingService;
import com.google.firebase.messaging.RemoteMessage;

/* ... */

public class FirebaseHandler extends FirebaseMessagingService {

    @Override
    public void onNewToken(@NonNull String s) {
        super.onNewToken(s);
        /* s == FCM-Token! Send this to server */
        Log.e("LPCS_LOG", s);
        /* ... */
    }

    @Override
    public void onMessageReceived(RemoteMessage remoteMessage) {
        /* Handle push-notification if user is In-App */
        Log.d("LPCS_LOG", "From: " + remoteMessage.getFrom());

        if (remoteMessage.getData().size() > 0) {
            /* Message has payload */
            Log.d("LPCS_LOG", "Message data payload: " + remoteMessage.getData());

            String notificationTitle = remoteMessage.getNotification().getTitle();
            String notificationBody = remoteMessage.getNotification().getBody();

            /* ... */
        }
    }
}
```

Abbildung 5.5.31: Ausschnitt aus `FirebaseHandler.java`.

### 5.5.3.5 Vorsichtsmaßnahmen

Wenn das Produkt bereits bei tausenden Anwendern installiert ist, ist zur Vorsicht geraten, denn man kann durch eine falsche Zeile Code oder einen falschen Klick potenziell tausende Anwender mit einer Notification erreichen. Die folgenden Ausschnitte machen es als Negativbeispiel vor:

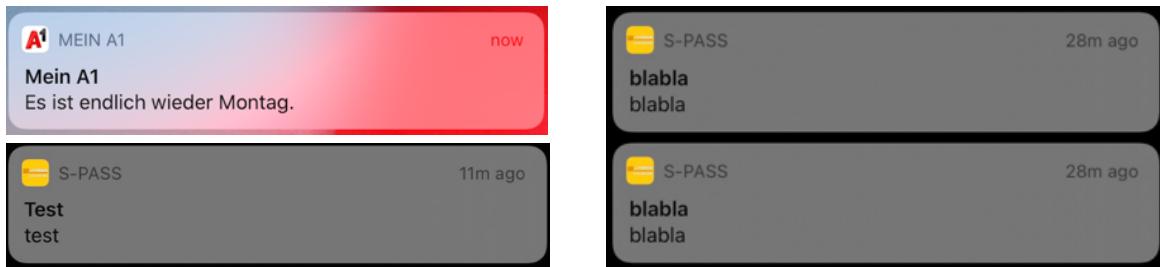


Abbildung 5.5.32: Sinnlose Notifications.

Test is working! Allerdings muss das nicht jeder Nutzer wissen. Einem professionellen Entwickler unterläuft so ein Fehler nicht – und erst recht nicht mehrere Monate später erneut. Doch diese peinliche Situation lässt sich ganz einfach vermeiden, indem man Production- und Test-Certificates nutzt.

### 5.5.3.6 Triggern einer Notification

Das Senden einer Notification ist mit einer einfachen **Post-Anfrage** an die Server der Google Cloud-Dienste möglich:

```
POST https://fcm.googleapis.com/fcm/send
```

Abbildung 5.5.33: Aufzurufender Endpoint

Serverseitig eignet sich dafür cURL in PHP oder zum Testen und Debuggen das API-Tool „Postman“. Damit die Firebase weiß, welchem Projekt die Request zuzuordnen ist, ist es von Nöten den Serverschlüssel mit anzugeben. Dieser kann in den Firebase Projekteinstellungen unter Cloud-Messaging gefunden werden. Des Weiteren gilt es noch den Content-Type zu definieren:

<code>Authorization</code>	<code>key=AAAFmLD1Xk:APA91tRTh34HrhtrGaRh_f...</code>
<code>Content-Type</code>	<code>application/json</code>

Abbildung 5.5.34: Benötigten Header

### **Senden an einen FCM-Token:**

```
{
  "registration_ids": [
    "cM8i4Pif24c:APA91bFch0dNZi1pLBV_pmxblPBkseCParexNS6wTqWxu7yCAR6o4M0p_6I_YP4_SjEWZqo5azs7Vnm3BpkFUN
    C-dP5GLmLF_r0grg251mnF45zVf5QheEjaR5kSSWV5XmbxdJcGEYpu"/* can be multiple fcm tokens (array) */,
    "notification": {
      "title": "Titel der Notification",
      "body": "Body der Notification",
      "sound": "default"
    }
}
```

Abbildung 5.5.35: Body des Requests

### **Senden an ein Topic:**

```
{
  "to": "/topics/mytopic", /* replace "mytopic" with desired topic name */
  "notification": {
    "title": "Titel der Notification",
    "body": "Body der Notification",
    "sound": "default"
  },
  "priority": 10 /* define importance of the notification 1-10 */
}
```

Abbildung 5.5.36: Body des Requests

### **Interpretation der Response:**

Damit das System am Server auch zu Protokoll führen kann, ob eine Notification erfolgreich zugestellt wurde, ist es möglich die JSON-Response genauer zu analysieren.

```
{
  "multicast_id": 2196728584260647617,
  "success": 2,
  "failure": 0,
  "canonical_ids": 0,
  "results": [
    {
      "message_id": "1647343141330187"
    },
    {
      "message_id": "1647343141327771"
    }
  ]
}
```

Abbildung 5.5.37: Erfolgreicher Zustellversuch.

In diesem Beispiel ist erkennbar, dass die Nachricht erfolgreich an zwei Geräte zugestellt wurde. Die `message_id` könnte genutzt werden, um eine Notification zurückzurufen (verbreitet bei Messenger-Apps). Dabei ist allerdings nicht garantiert, dass der Nutzer diese noch nicht gesehen hat.

Der Server kann allerdings anhand dieser Response nicht feststellen, ob das Gerät im ausgeschalteten Zustand ist oder nicht. Der APNs nimmt die Anfrage der Firebase entgegen, stellt fest, dass das Gerät existiert und theoretisch erreichbar wäre und versucht die Nachricht so schnell wie möglich zuzustellen – spätestens, wenn das Gerät wieder eingeschalten ist und eine Internetverbindung aufweist.

```
{
  "multicast_id": 496673997895279092,
  "success": 0,
  "failure": 1,
  "canonical_ids": 0,
  "results": [
    {
      "error": "InvalidRegistration"
    }
  ]
}
```

Abbildung 5.5.38: Fehlgeschlagener Zustellversuch.

Anders sieht es hingegen in diesem Beispiel aus. Der Zustellversuch ist fehlgeschlagen, woraus das System feststellen kann, dass der angegebene FCM-Token ungültig ist.

#### 5.5.4 AppStoreConnect

Der AppStore ist bekannt durch qualitative Apps. Die Gründe dafür sind allerdings weniger bekannt. Apple kontrolliert den eigens geschaffenen Markt sehr streng: Ohne einer jährlich-kostenpflichtigen Apple-Developer-Membership für 99€ ist das Anbieten einer App im AppStore oder auf TestFlight erst gar nicht möglich. Alternativen zum AppStore gibt es nicht, denn eine nicht aus dem AppStore heruntergeladene App (Extension „.ipa“) kann nur über ein Profile, verbunden mit mehreren Warnhinweisen, installiert werden. Auch Hobby-Entwickler sind somit gezwungen eine stattliche Gebühr an Apple abzudrücken.

Doch selbst nach abschließen einer Developer-Mitgliedschaft gibt es noch Prüfverfahren. Jede App wird mit jedem Update vor Veröffentlichung von Apple selber geprüft. Stellen diese Prüfer einen Verstoß gegen deren Richtlichen fest (z.B. sind Firmeninterne Apps im AppStore verboten) wird die App nicht veröffentlicht. Apple hält sich sehr bedeckt mit Informationen zu den Prüfteams, doch durch protokollieren jedes Vorgangs in der App konnte ich feststellen, dass diese die Apps auf mehreren Geräten mit verschiedenen Displaygrößen und Prozessoren testen. Selbst für den Public-Beta-Test auf TestFlight, wo man sich eine App nur per direktlink-Einladung herunterladen kann, wurde die App getestet. Selbstverständlich muss man bei Apps wie LPCS mit Login-verfahren einen Demo-Account zur Verfügung stellen. Durch die Protokollierung unseres Systems stellte ich fest, dass die LPCS-App mit folgenden Geräten in einem Zeitraum von 20 Minuten getestet wurde:

- Der erste Login wurde auf einem iPhone SE (2nd Gen) mit 4,7“ und einem A9-SOC verzeichnet.
- Der zweite Login wurde auf einem iPad Pro (5th Gen) mit 12,9“ und einem M1-ARM-Prozessor verzeichnet.
- Der dritte Login wurde auf einem iPad mini (6th Gen) mit 7,7“ und einem A15-SOC verzeichnet.

Interessant zu beobachten ist die Wahl der Geräte. Apple deckt das volle Spektrum der angebotenen Displaygrößen ab, um sicherzustellen, dass die App an jedes Gerät angepasst ist. In den drei verwendeten Geräten waren auch alle Altersstufen der aktuell am Markt vorhandenen Prozessoren abgedeckt (Apple A9, Apple A15 und Apple ARM M1). Ein Kriterium für das Ablehnen einer App wäre, wenn eine Richtlinie nicht erfüllt wird oder die App abstürzt.

Neben der App an sich überprüft Apple allerdings auch noch den Bundle-Identifier, die Vorschau-Screenshots und angegebenen Texte für den AppStore-Eintrag. Werden dabei nicht die strengen Vorgaben eingehalten, wird die App ebenfalls abgelehnt.

Falls man mit einer App im AppStore kommerziellen Gewinn erzielen möchte, gibt es allerdings ein paar Rückschläge, welche man in Kauf nehmen muss: Von jedem erzielten Umsatz (App-Kauf oder In-App-Kauf) zieht Apple einen Prozentsatz von 30 % ein.

Auch wenn es für Entwickler viele unangenehme Aspekte am AppStore gibt, garantieren diese Verfahren eine hohe Qualität der Apps. Sinnlose, Phishing oder veraltete Apps suchen Nutzer im AppStore vergebens.

### 5.5.5 Play Console

Um seine App den Großteil der österreichischen Smartphone-Nutzer bereitstellen zu können, ist es von Nöten jene im Google Play Store für Android-Geräte anzubieten. Dafür ist lediglich das Abschließen einer einmaligen Developer-Mitgliedschaft in Höhe von 20 € notwendig. Doch dieser geringe Aufwand Apps auf Android zu veröffentlichen schlägt sich unweigerlich auf die Qualität, das Angebot und die Experience der angebotenen Apps nieder. Während im AppStore unter iOS eine App mit nur wenigen Schlagwörtern eindeutig gefunden werden kann, sucht man im PlayStore auf Seite fünf noch vergebens nach der App, welche man mit exakten Namen gesucht hat, da diese unter zahlreichen anderen untergeht.

Auch Google schreibt in der Dokumentation von strengen Überprüfungen der Richtlinien und genauen Prüfungsverfahren der eingereichten Apps vor Veröffentlichung – in der Realität sieht dies basierend auf meinen Beobachtungen allerdings anders aus. Viele Hobbyentwickler geben bei Angaben wie den Publishernamen oft Fantasienamen an, das Impressum wird gelegentlich zur Gänze weggelassen und Angaben über Datenschutz sucht man oft vergebens. Reicht man eine App oder ein App-Update zur Veröffentlichung ein, wird der Status zwar auf „Warte auf Überprüfung“ und später auf „In Überprüfung“ geändert, allerdings wurde bei meinen bisher eingereichten Apps noch nie eine Anmeldung von einem Testgerät verzeichnet. Auch dauert das „Prüfverfahren“ nicht wie bei Apple durchschnittlich einen Tag, sondern meist nur wenige Stunden.

## 6 Grundlagen und Methoden – Noah Laireiter

### 6.1 Aufgabenbereiche

#### 6.1.1 KFZ-Tracking-Modul

Das Tracking-Modul sendet Standort und Fahrtdataen an den Server.

#### 6.1.2 Keybox

Die Keybox dient zur Aufbewahrung der Autoschlüssel.

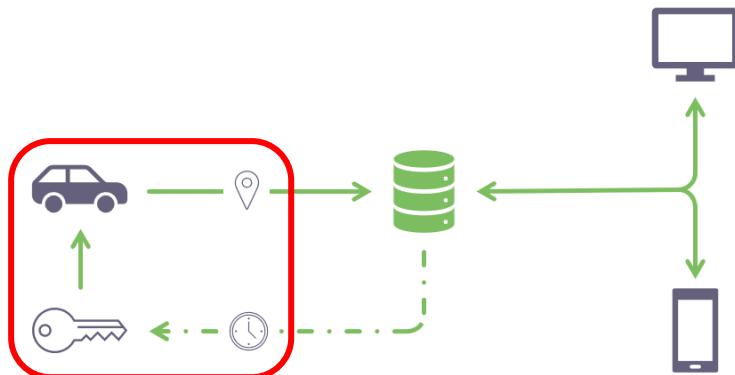


Abbildung 6.1.1 Systemaufbau und Aufgabenbereiche (Noah Laireiter)

#### 6.1.3 Unit-Tests und App-Security

Zur Gewährleistung einer fehlerfreien Software müssen in einem Projekt sogenannte Unit-Tests durchgeführt werden. Weiters wird auch auf die Sicherheit in der Anwendung geachtet, sodass nur autorisierte Nutzer auf ihre eigenen Daten zugreifen können.

#### 6.1.4 Dokumentation

Schriftführung und Protokollierung der Projektbesprechungen, sowie der durchgeführten Arbeiten.

## 6.2 Hardwareentwicklung

Für den Entwurf des Tracking-Moduls und der Schlüsselbox wird auf die Software-Produkte der Autodesk GmbH zurückgegriffen. Diese bringen den Vorteil mit sich, dass das Designen von Leiterplatten und die Konstruktion eines dazu passenden Gehäuses gemeinsam durchgeführt können.

Für die Entwicklung dieses Projektes werden die Programme EAGLE und Fusion benötigt.

### 6.2.1 Autodesk EAGLE

EAGLE steht für „Einfach Anzuwendender Grafischer Layout-Editor“. Dieser verknüpft die Gestaltung von Stromlaufplänen, das Platzieren von Elektronikkomponenten und das Routen von Leitungen auf Leiterplatten.

#### 6.2.1.1 Bibliotheken

EAGLE bietet umfassende Bibliotheken an. Sollten diese nicht alle benötigten Komponenten enthalten, findet man im Internet zumeist gewünschte Vorlagen. In den seltenen Fällen, wenn ein spezieller Baustein verbaut wird und es keine vorgefertigten Bibliotheken gibt, kann diese im „Package Editor“ auch manuell erstellt werden.

### 6.2.1.2 Schaltplanentwurf

Stehen alle notwendigen Bibliotheken für das Projekt zu Verfügung, kann der Schaltplan gezeichnet werden. Dabei können Elemente hinzugefügt und verbunden werden.

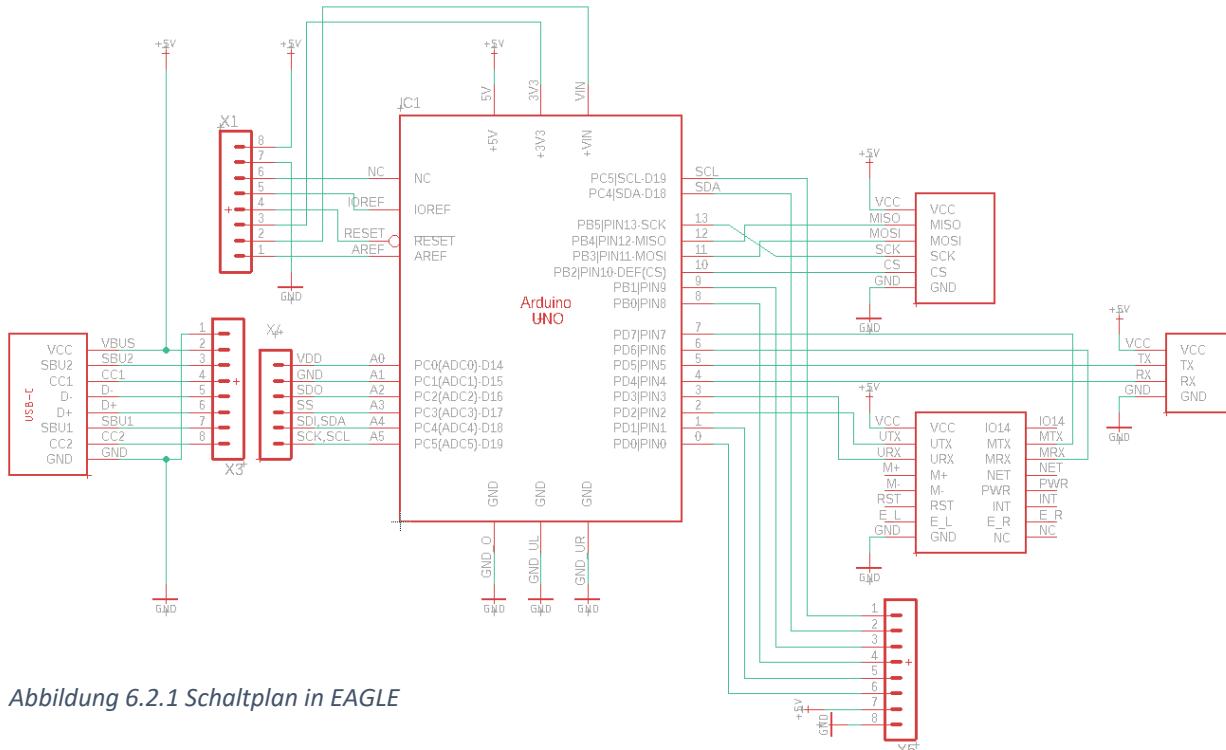


Abbildung 6.2.1 Schaltplan in EAGLE

### 6.2.1.3 Layout-Entwurf

Aus einem Schaltplan kann in EAGLE automatisch eine Platine erstellt werden. Auf dieser werden die Komponenten aus dem Schaltplan (mit realen Dimensionen) platziert und mit Leiterbahnen verbunden.

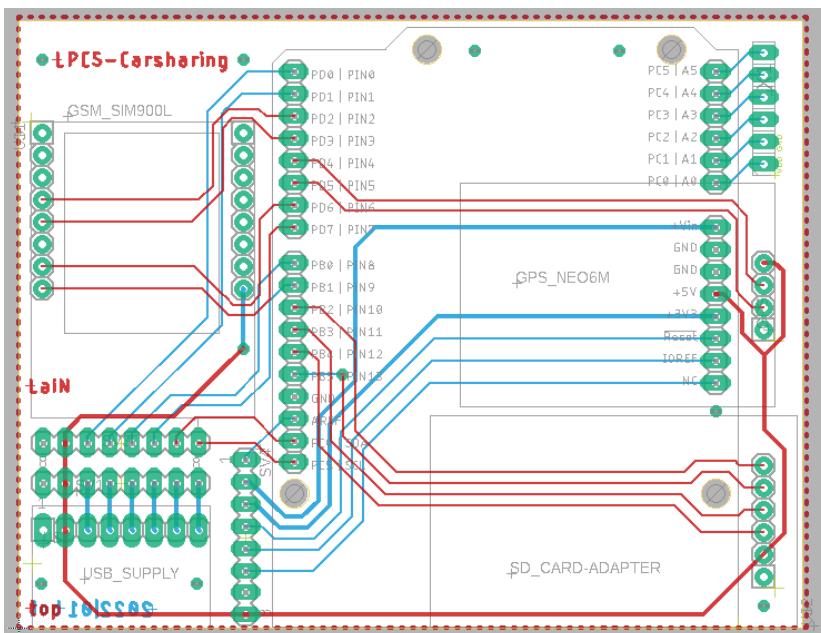


Abbildung 6.2.2 Leiterplattenlayout in EAGLE

## 6.2.2 Autodesk Fusion 360

Ein weiteres Programm von Autodesk ist Fusion 360, welches es ermöglicht 3D-Modelle zu gestalten. Von kreativen Ideen bis hin zur Gehäuseentwicklung ist dabei alles möglich. Vorwiegend werden darin aber Gehäuse zu den in EAGLE erstellten Platinen konstruiert.

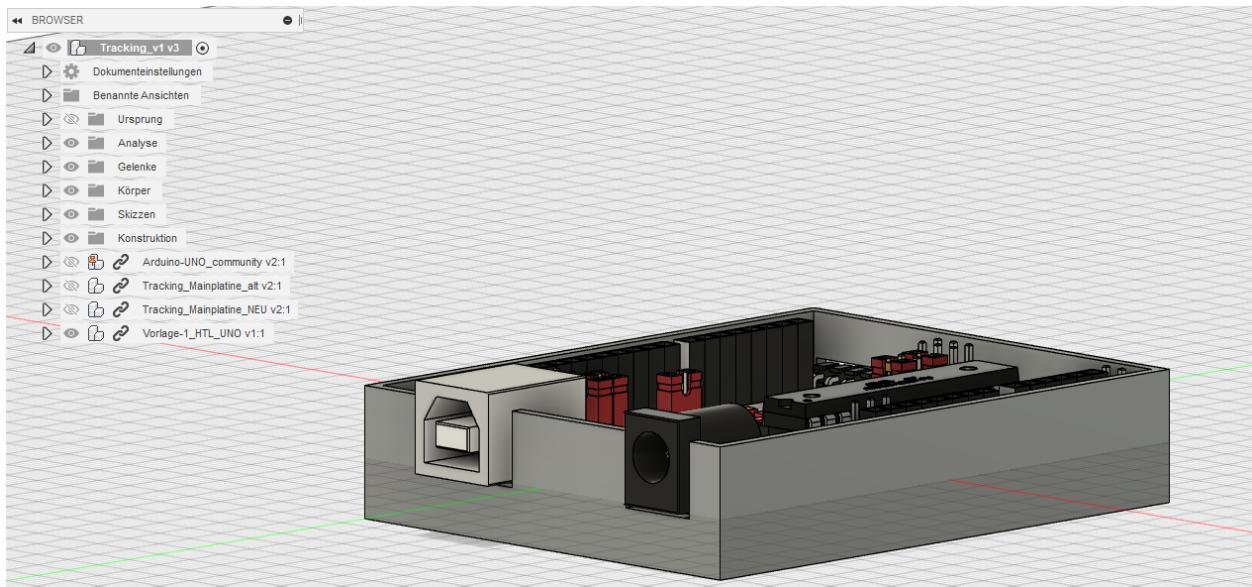


Abbildung 6.2.3 Gehäuseentwurf in Fusion360

## 6.3 Hardwareprogrammierung

### 6.3.1 Microchip Studio

Microchip Studio (vormals ATTEL Studio) ist eine Entwicklungsumgebung für die Erstellung von AVR und SAM Mikrokontroller-Programmen. Der Code kann dabei entweder in C/C++ oder in Assembler geschrieben werden und anschließen in ein .hex-File kompiliert werden. [11]

Weiters ermöglicht es Microchip Studio eine Simulation mit dem geschriebenen Code am Computer laufen zu lassen. In einem graphischen Interface kann man dabei Bits in den Registern setzen und löschen, um so das Verhalten der Software beurteilen zu können und mögliche Fehler zu finden, bevor das Programm auf der Hardware ausgeführt wird.

Es werden jedoch keine Programmier- und Kommunikationsschnittstelle zur Hardware angeboten, weshalb dafür zwei weitere Programme benötigt werden.



Abbildung 6.3.1 Atmel-Studio Logo [11]

### 6.3.2 chip45boot2 GUI

Dieses kleine Programm wird zum flashen von .hex-Files (kompilierter Code für Mikrokontroller) auf einen Mikroprozessor verwendet. Über eine USB-Schnittstelle wird die gewünschte Hardwarekomponente mit dem Computer verbunden und programmiert.

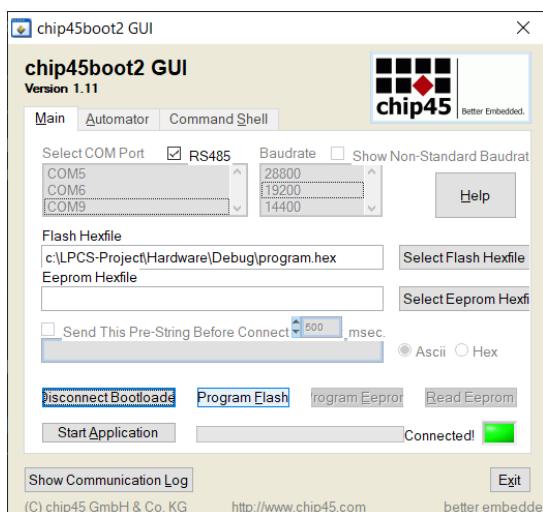


Abbildung 6.3.2 chip45 Bootloader

### 6.3.3 HTerm

HTerm stellt eine serielle Verbindung mit einem Device über USB her und ermöglicht die Kommunikation mit diesem, so dass Ausgaben einer Hardware am Bildschirm des Computers angezeigt werden können. So kann nach möglichen Fehlern in der Software, die bereits auf einem Chip in Betrieb ist, gesucht werden.

### 6.3.4 Arduino IDE

Diese Entwicklungsumgebung wird von der Firma Arduino zur Verfügung gestellt. Sie beinhaltet Bibliotheken für einen Großteil der Hardwarekomponenten, was es zu einer häufig genutzten Software im Bereich der Hardwareprogrammierung macht. Vom Auslesen von Sensoren bis zur Ansteuerung eines Displays lassen sich Vorlagen finden, mit welchen die Umsetzung eines Projekts schnell und unkompliziert ablaufen kann.

#### 6.3.4.1 GNU General Public License

Die veröffentlichten Bibliotheken und Vorlagen unterliegen dabei der „GNU General Public License“, welche es Nutzern ermöglicht, geistiges Eigentum anderer für seine eigenen Projekte zu verwenden, abzuändern und weiterzuverbreiten. [12]

#### 6.3.4.2 Vorteil gegenüber Microchip-Studio

Im Gegensatz zu Microchip-Studio vereint Arduino den Quelltexteditor, die Programmierschnittstelle und einen seriellen Monitor in einem Programm. Somit ist ein schnelleres und praktischeres Arbeiten möglich.

### 6.3.5 IntelliJ IDEA und Visual Studio Code

Die IntelliJ IDEA von JetBrains ist eine integrierte Entwicklungsumgebung, vorwiegend für die Programmiersprache Java. Weiters unterstützt IntelliJ Tools zur Versionsverwaltung wie Git, wodurch es sich als ideale Schnittstelle zu GitLab angeboten hat. [14]



Abbildung 6.3.3 IntelliJ Logo [14]

Visual Studio Code (VS Code) der Firma Microsoft ist ein Quelltexteditor für die drei gängigsten Betriebssysteme Windows, macOS und Linux. [15] Es bietet die Möglichkeit C und C++ Codeteile zu kompilieren und auszuführen, was es zum Mittel der Wahl gemacht hat, um Teile der Software zu testen. Da die Fehlersuche auf der Hardware nur erschwert möglich ist, da die Aktionen des ausgeführten Programms nur schwer über den seriellen Monitor mitverfolgt werden können, wird VS Code im Vorfeld als Verifikationssoftware eingesetzt. Einzelne Codeabschnitte können schnell kompiliert und auf etwaige Fehler kann schnell und einfacher reagiert werden, wodurch die Fehlersuche auf der Hardware auf ein Minimum reduziert wird und Zeit eingespart werden kann.



Abbildung 6.3.4 VSCode Logo [15]

## 6.4 Graphische Visualisierung der Hardware

### 6.4.1 Draw.io

Das Online-Tool Draw.io bietet eine Großzahl an Vorlagen an, mit welchen Abhängigkeiten in einem System graphisch dargestellt werden können. Dadurch ist es möglich die Zusammenhänge der Hardware in sich und mit der Software übersichtlich zu veranschaulichen.

## 6.5 Hardware

### 6.5.1 Wahl des Mikrokontrollers

Zu Beginn des Projekts ist das crumb644-Modul von ATMEL als zentrales Steuerelement für jeden einzelnen Hardwareteil gewählt worden, da dieser kostengünstig ist und eine Vielzahl an nutzbaren PINS besitzt. Vor Allem bei der Schlüsselbox ist dies von Vorteil, da Bauteile wie das Eingabefeld sehr viele PINS zum Auslesen beanspruchen.

Im Laufe des Projekts ist aber auf einen Arduino UNO gewechselt worden. Ausschlaggebend für einen Wechsel sind die Performanceeinbrüche des crumb644-Moduls gewesen. Mit der größer werdenden Software und mehr Daten, welche vom Chip bearbeitet werden mussten, ist dieser langsamer und fehleranfälliger geworden. Ein weiterer Grund ist, dass das Arduino Framework sehr viele Bibliotheken beinhaltet, welche grundlegende Aufgaben abnehmen. Weiters stellen Hersteller von Hardwarebaugruppen zumeist nur Bibliotheken für Arduino und Raspberry-Pis zur Verfügung. Da es ein unnötiger Aufwand wäre all diese Vorlagen umzaprogrammieren, damit sie kompatibel mit dem crumb644-Modul sind, ist auf den Arduino gewechselt worden. Und auch in Hinsicht auf das Flashen des Mikrocontrollers und die Fehlersuche bietet die Arduino IDE ihre Vorteile. Bei Microchip Studio wird ein externes Programm zum Programmieren des Controllers und ein weiteres zum Kommunizieren und zur Fehlersuche benötigt. Die Arduino IDE bringt all diese Funktionen in einem Programm zusammen und ermöglicht so ein schnelleres und praktischeres Arbeiten.

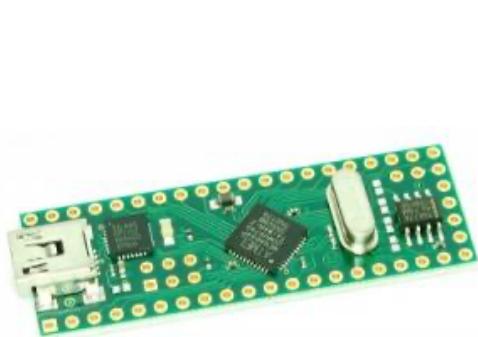


Abbildung 6.5.1 crumb644-Modul [37]

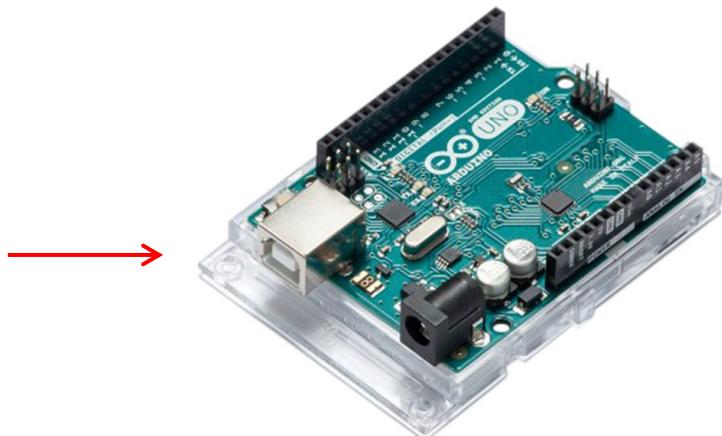


Abbildung 6.5.2 Arduino UNO [36]

## 6.6 Bussysteme

Die Kommunikation und der Datenaustausche zwischen Hardwarebausteinen erfolgt in den meisten Fällen über Bussysteme. Die folgen drei (UART, SPI, I2C) sind die meist genutzten.

### 6.6.1 UART

UART steht für Universal Asynchronous Receiver Transmitter und ist ein Bussystem, welches unter anderem bei USB verwendet wird.

Es ist ein Bussystem, welches ohne Takteleitung auskommt – daher der Name „Asynchronous“ – und im Vollduplex-Betrieb Daten austauschen kann.

Da es keine Takteitung gibt, muss im Vorhinein die sogenannte Baudrate festgelegt werden. Diese bestimmt wie viele bits pro Sekunde gesendet, beziehungsweise empfangen werden. Beim Verbinden der Leitungen ist zu beachten, dass die RxD und TxD PINS kreuzweise miteinander verbunden werden müssen und beide Bausteine mit dem gleichen Massepotential verbunden sind, da es sonst zu einer Zerstörung der Chips kommen kann. [16]

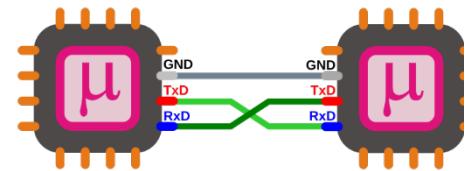


Abbildung 6.6.1 UART-Verbindung [16]

### 6.6.2 SPI

Das Serial Peripheral Interface ist ein Bussystem welches zusätzlich zu den beiden Datenleitungen noch eine Takteitung hat. Zusätzlich wird für jeden einzelnen angeschlossenen „Slave-Chip“ ein Chip-Select-Leitung benötigt. Diese signalisiert einem Chip, dass er angesprochen wird und Daten senden darf. [17]

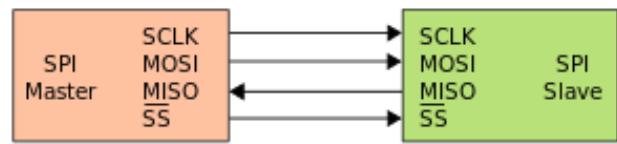


Abbildung 6.6.2 SPI-Verbindung [35]

Im Gegensatz zu UART werden die beiden Datenleitungen hier nicht kreuzweise verbunden.

Nachteil dieses Busses ist, dass bei steigender Anzahl der Chips, die benötigte Menge an PINS und Kabeln für die Chip-Select-Leitungen ebenso schnell ansteigt.

### 6.6.3 I2C

Das „I quadrat C“-Interface beschränkt sich auf zwei Leitungen. Daher wird es oft auch als TWI (Two Wire Interface) bezeichnet. Der Vorteil dieses Bussystems ist es, dass eine Takteitung und eine Datenleitung gibt. Diese übermittelt nicht nur die Daten, sondern adressiert auch die Bausteine, wodurch es einfacher möglich ist, mehrere Chips an einen Master zu hängen. [18]

Der Nachteil bei diesem System ist, dass es nicht im Vollduplex-Betrieb Daten austauschen kann. Weiters wird zu Beginn jeder Verbindung Laufzeit in Anspruch genommen, um den gewünschten Chip zu adressieren. In dieser Zeit ist kein Datenaustausch möglich.

Bei der Verkabelung ist zu beachten, dass die I2C-PINS der Mikrokontroller Open-Collector-Ausgänge sind. Diese haben keine internen Pull-up-Widerstände an den Transistoren, weshalb beim Verbinden der Bausteine nicht auf externe Pull-ups vergessen werden darf. Werden diese nicht implementiert, kommt es zu unbestimmten Logikpegeln, was zufällig erscheinende Datenströme zu Folge hat.

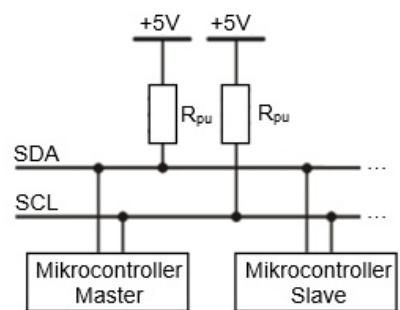


Abbildung 6.6.3 I2C-Verbindung [34]

## 6.7 Tracking-Modul

Um den Fahrzeugstandort jederzeit abrufen zu können, wird ein Tracking-Modul darin verbaut. Dieses hat die Aufgabe in regelmäßigen Abständen Fahrtinformationen, wie Geschwindigkeit und gefahrene Kilometer an den Server zu übermitteln. Für die Versorgung ist eine Verbindung mit USB gewählt worden, da die meisten Autos einen USB-Anschluss haben. Alternativ kann das Modul auch über den Zigarettenanzünder versorgt werden. Dann wird jedoch noch ein weiterer Adapter benötigt.

Aufgebaut ist diese Einheit aus dem zentralen Steuerelement, einem Arduino UNO, einem GPS-Modul, welches die Standort- und Geschwindigkeitsdaten liefert, einer Speichereinheit zum Zwischenspeichern der empfangenen Daten und einem GSM-Modul, welches die Daten an den Server weiterleitet.

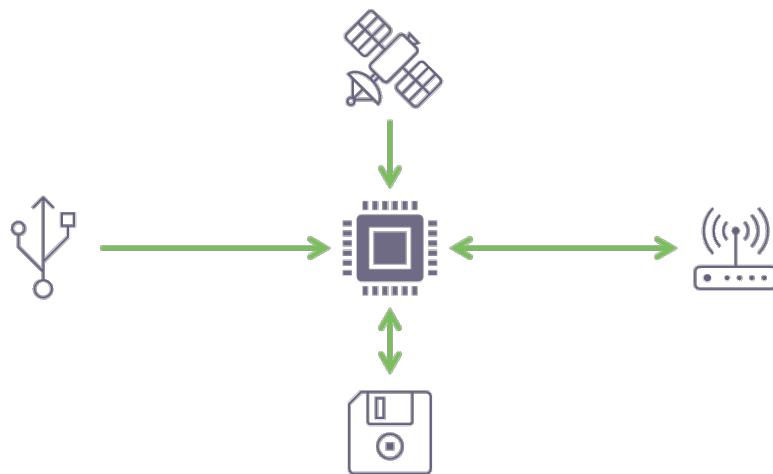


Abbildung 6.7.1 Aufbau des Tracking-Moduls

### 6.7.1 Steuerelement

Gesteuert wird das Tracking-Modul von einem Arduino. Dieser übernimmt die Aufgaben der Datenauswertung und Weiterverarbeitung. Die Standortdaten, welche das GPS-Modul empfängt, werden zerlegt und in die gewünschte Form zur Weiterverarbeitung umgewandelt, gespeichert und an den Server weitergeleitet.

### 6.7.2 Energieversorgung

Als Versorgung ist eine Verbindung mit USB zum Fahrzeug gewählt worden, da die meisten Autos heutzutage einen USB-Port verbaut haben. Sollte keiner vorhanden sein kann alternativ auf den Zigarettenanzünder mit einem Adapter zurückgegriffen werden.

## 6.7.3 GPS-Funktionsweise

### 6.7.3.1 Schnittpunkt mehrerer Signale

Für die Bestimmung des Standorts werden folgende Komponenten benötigt:

- Satelliten / geostationäre Satelliten
- Bodenstationen
- Ein GPS-Gerät zum Bestimmen

Die Stationen senden regelmäßig ihren Standort und die Uhrzeit als codierte Radiowellen aus. Aus diesen Signalen und dem Abstand zu den Sendern kann das GPS-Modul den Standort und weitere Daten, wie Geschwindigkeit und Seehöhe kalkulieren. [19]

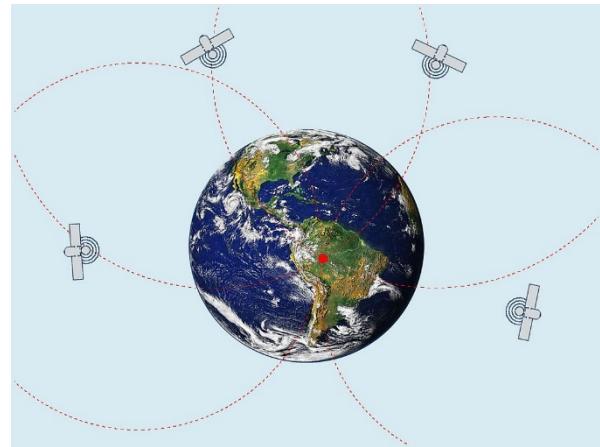


Abbildung 6.7.2 Schnittpunkt zur Positions berechnung [19]

### 6.7.3.2 Fehlerquellen

Ungenauigkeiten treten durch Rechenfehler der Satellitenumlaufbahn oder durch ungünstige Positionierung der Satelliten relativ zum GPS-Modul auf. Sollten die Satelliten genau in einer Linie sein, nimmt die Genauigkeit um mehrere Meter ab. [19]

Weitere Abweichungen treten durch Reflektionen und Laufzeitenunterschiede in der Atmosphäre auf. Durch diese Phänomene kann ein ausgesendetes Signal über mehrere, unterschiedlich lange Wege vom Modul empfangen werden. Dieser sogenannte „Multipath-Effekt“ kann zu Ungenauigkeiten von bis zu 150 Metern führen. [19]

Der größte Fehler wird allerdings durch Abweichungen der Uhrzeit verursacht. Eine Microsekunde reicht da schon aus, um den Standort nur mehr auf 300 Meter genau berechnen zu können. [19]

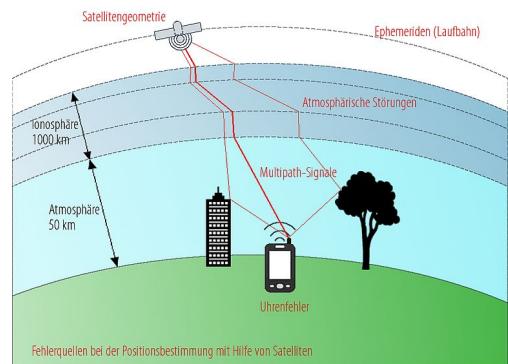


Abbildung 6.7.3 GPS-Fehlerquellen [18]

### 6.7.3.3 Fehlerkorrekturen

Es gibt zwei bedeutende Korrekturmethoden:

1. Differential Global Positioning System (DGPS)
2. Real Time Kinematic – Korrektur (RTK)

Davon ist DGPS die einfachere, billigere und ungenauere Variante im Vergleich mit RTK. [20]



Abbildung 6.7.4 Präzision der GPS-Korrekturmethoden [20]

DGPS verwendet Signale von Bodenstationen, von welchen der exakte Standort bekannt ist, wodurch die Positionsbestimmung deutlich verbessert wird und Fehler korrigiert werden können. Mit diesem Verfahren kann die Abweichung auf einige Meter verringert werden. [20]

Die zweite Möglichkeit, RTK, verspricht hingegen eine Genauigkeit von wenigen Zentimetern. Dabei werden nicht die gesendeten Daten in den Radiowellen der Satelliten zur Berechnung herangezogen, sondern die Wellen an sich. Die Wellenlängen mehrerer Signale werden miteinander verknüpft, wodurch dann möglich ist, den Standort mit dieser hohen Genauigkeit zu kalkulieren. [20]

## 6.7.4 GPS-Modul

### 6.7.4.1 Datenstring

Das GPS-Modul liefert über die UART-Schnittstelle einen langen Datenstring an den Arduino. Dieser enthält 19 Teile mit unterschiedlichen, teils ähnlichen Daten. Die drei wichtigsten Abschnitte für dieses Projekt sind \$GPGLL, \$GPRMC und \$GPGGA.

#### \$GPGLL – Geographic Position, Latitude / Longitude

\$GPGLL, Latitude, N/S, Longitude, E/W, UTC-Time, A

Hier können Längen- und Breitengrad herausgelesen werden. Um negativen Zahlen zu vermeiden, wird nach den Koordinaten noch angegeben in welche Himmelsrichtung zu gehen ist. Westliche und südliche Koordinaten würden hier ansonsten ein negatives Vorzeichen haben. [21]

#### \$GPRMC – Recommended minimum specific GPS data

\$GPRMC, UTC-Time, A, Latitude, N/S, Longitude, E/W, speed,,,E/W\*checksum

Hier kann neben Standort und Uhrzeit auch noch die Geschwindigkeit des Moduls entnommen werden. Diese werden in Knoten über Grund angegeben, weshalb sie zur Weiterverarbeitung noch umgewandelt werden müssen. [21] (1 Knoten entspricht dabei in etwa 1,852 km/h)

#### \$GPGGA – Global Positioning System Fix Data

\$GPGGA, UTC-Time, Latitude, N/S, Longitude, E/W, quality, satellites,,height,M,,,station-ID\*checksum

Diese Zeile liefert eine höhere Genauigkeit, abhängig des „quality“-Wertes. Mit 0 hat man keine Daten, mit 1 normales GPS und mit 2 das zuvor erwähnte DGPS. Außerdem gibt dieser Teil auch Auskunft über die Meereshöhe, auf der sich das Modul befindet. [21]

Dieser lange String kann zum einen händisch ausgewertet werden, was in C aber aufwendig und fehleranfällig ist.

Hier kommen die Bibliotheken von Arduino in Verwendung (mit GNU Public Licence zugänglich). Es ist auch erlaubt diese abzuändern und neu weiterzuverbreiten. [12]

Ideal bietet sich hier für die TinyGPSPuls-Bibliothek an. Diese zerteilt den String und gibt einzelnen Daten im gewünschten Format zurück. [22]

#### 6.7.4.2 TinyGPSPlus-Bibliothek

```

while (gps_lpcs.available() > 0) { // solange das GPS-Modul Daten sendet
    gpsparser.encode(gps_lpcs.read()); // TinyGPSPlus liest daten von GPS-Modul

    /*CREATING URL*
    String date_val = String("");
    String lat_val = String("");
    String lng_val = String("");
    String altitude_val = String("");
    String speed_val = String("");

    /*create Date-Value (Format: yyyy-mm-dd hh:mm:ss)*

    date_val.concat(gpsparser.date.year()); // add year
    if(gpsparser.date.month() < 10) date_val.concat("0"); date_val.concat(gpsparser.date.month()); // add month
    if(gpsparser.date.day() < 10) date_val.concat("0"); date_val.concat(gpsparser.date.day()); // add day
    if(gpsparser.time.hour() < 10) date_val.concat("0"); date_val.concat(gpsparser.time.hour()); // add hour
    if(gpsparser.time.minute() < 10) date_val.concat("0"); date_val.concat(gpsparser.time.minute()); // add minute
    if(gpsparser.time.second() < 10) date_val.concat("0"); date_val.concat(gpsparser.time.second()); // add second

    // create Lat/Lng-Value and altitude/speed-Value
    lat_val.concat(gpsparser.location.lat()); // add latitude
    lng_val.concat(gpsparser.location.lng()); // add longitude

    altitude_val.concat(gpsparser.altitude.meters()); // add altitude
    speed_val.concat(gpsparser.speed.kmph()); // add speed
}

```

Abbildung 6.7.5 Auslesen der Daten aus dem NMEA-String

Dieser Codeausschnitt verwendet die TinyGPSPlus-Bibliothek im „gpsparser“. Es müssen zu Beginn nur die 19 Zeilen eingelesen werden. Daraus können danach einfach Datum, Standort, Höhenmeter und die Geschwindigkeit entnommen werden. [22]

#### 6.7.4.3 Erstellen des Sendestrings

```

// create whole URL
String url = String("");
url.concat(_urlbase); // konstant
url.concat(_carid); // konstant
url.concat(_carid_val); // konstant
url.concat(_date); // konstant
url.concat(date_val);
url.concat(_lat); // konstant
url.concat(lat_val);
url.concat(_lng); // konstant
url.concat(lng_val);
url.concat(_altitude); // konstant
url.concat(altitude_val);
url.concat(_speed); // konstant
url.concat(speed_val);

```

Abbildung 6.7.6 Erstellung einer URL aus den Daten

Die ausgelesenen Daten werden am Ende zu einem langen String zusammengefügt, welcher wie eine URL mit Query-Strings aufgebaut ist. Damit können die Daten später an den Server übertragen werden.

### 6.7.5 Zwischenspeicher und Logging

Der Speicher ist in diesem Modul ein Sicherheitsfeature. Sollte es dazu kommen, dass die Stromversorgung für einen Zeitraum nicht genug Strom für das GSM-Modul zum Senden bereitstellt, dann gehen keinen Daten verloren, wenn diese noch zusätzlich auf einer SD-Karte zwischengespeichert werden. Auch wenn keine Funkverbindung für das GSM-Modul möglich ist, gehen damit keine Daten verloren.

In regelmäßigen Abständen wird dann versucht die Daten im Zwischenspeicher zu versenden. Ist das mindestens zweimal möglich, gelten die Daten als versendet und werden gelöscht.

## 6.7.6 GSM-Funktionsweise

### 6.7.6.1 Netzstruktur

Das GSM-Netz (Global System for Mobile Communications) in Österreich ist in Form einer Zellstruktur aufgebaut. Dies ermöglicht die begrenzte Anzahl an Funkkanälen optimal ausnutzen zu können, da in zwei separaten Zellen der gleiche Kanal genutzt werden kann, ohne Datenverlust und Übersprechen zu generieren. Verbunden werden diese Zellen über Basisstationen, das Service Netzwerk und den Heimatverteiler. [23]

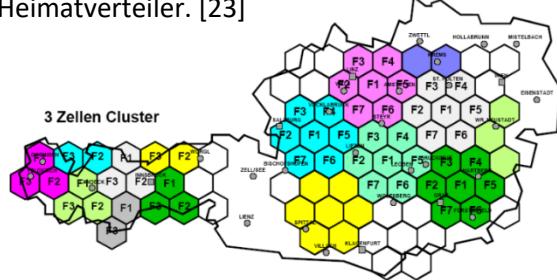


Abbildung 6.7.7 Zellenstruktur in Österreich [23]

### 6.7.6.2 GSM-Verbindungsauflaufbau

In einen Bereich verbinden sich Geräte über die Luftschnittstelle mit ihrer Basisstation, dem Basis Switching Centre. Dieses leitet dann an das Mobile Switching Centre (MSC) weiter, welches auf das Visitor und Home Location Register zugreifen kann. Aus diesen Registern kann ausgelesen werden, wohin die Verbindung weitergeleitet werden muss, um am Ziel anzukommen, das der Nutzer am Beginn angegeben hat (Telefonnummer, URL, ...). [23]

### 6.7.6.3 Authentifizierung

Die Verbindungen zu den Home Location Registers (HLR) sind verschlüsselt und verlangen eine Authentifizierung. Möchte sich ein Device mit einem bestimmten HLR verbinden, bekommt es eine Zufallszahl zugesendet, welche mit dem Authentifizierungsschlüssel verknüpft werden muss und zurückgesendet wird. Das HLR verknüpft die zufällige Zahl ebenfalls mit dem Schlüssel und vergleicht seinen Ausgang mit der Antwort des anfragenden Geräts. Stimmen diese überein, kann das Device auf das Register zugreifen, die gewünschten Zieladresse aufrufen und so den Verbindungsauflaufbau abschließen. [23]

### 6.7.6.4 GSM-Burst

Während die Geräte verbunden sind, können sie Daten austauschen. Da aber mehrere Devices von einer Funkzelle aus gleichzeitig kommunizieren wollen, werden diese mit Zeitmultiplex ineinander verschachtelt. Dabei stehen jedem Gerät innerhalb eines „time slots“ zweimal 57bit für die Datenübertragung zur Verfügung. Gesendet werden müssen diese bits dann inklusive eines vorgegebenen GSM-Rahmens in weniger als 0,6 Millisekunden. [23]

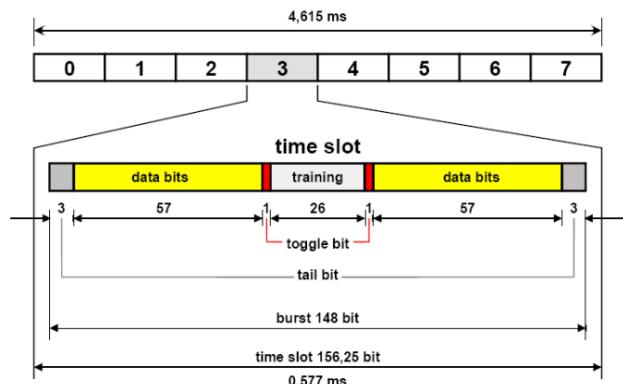


Abbildung 6.7.8 Zeitmultiplex und Rahmenstruktur bei GSM [23]

## 6.7.7 GSM-Modul

### 6.7.7.1 Wahl des GSM-Moduls

Für das Projekt ist das SIM800L GSM/GPRS Board gewählt worden, da es im Vergleich zu anderen weniger Platz in Anspruch nimmt und kostengünstiger ist. Zu Beginn ist auf die Standartausführung gesetzt worden, welche jedoch eine Spannung von 4,1V und einen Spitzenstrom von 1A voraussetzt. Da das Tracking-Modul aber nur mit 5V betrieben wird, ist es technisch nur schwierig möglich einen Spannungswandler zu entwerfen. Gründe dafür sind die niedrigen Drop-out-Voltage von 0,9V und der hohe Strom von 1A. Nach längerer Suche ist die Wahl dann auf ein typgleiches Board gefallen, welches aber mit 5V versorgt werden kann. Somit ist es nicht mehr nötig einen Spannungswandler zu entwerfen.

### 6.7.7.2 URL-Aufbau



Abbildung 6.7.9 Aufbau einer Internetadresse [24]

Zu Beginn befindet sich das gewählte Protokoll, die Domain und der Pfad zu der Datei, welche geöffnet und ausgeführt werden soll. Anschließend der Query-String mit zwei Parametern. In der obigen Abbildung wird ein Land und die Sprache übergeben.

Für die Datenübermittlung an den Server werden diese Query-Strings verwendet. Sie sind ein Teil des Uniform Resource Locators (kurz URL) und bieten die Möglichkeit Daten in Form von Parametern bei Aufruf einer Website mitzugeben. Diese Daten können von einem Script dann ausgelesen, weiterverarbeitet und beispielsweise in eine Datenbank geschrieben werden.

Eingeleitet wird der Query-String mit einem ? und endet mit einem #. Das Endzeichen wird aber nur benötigt, wenn die URL, wie in der Abbildung oben, nicht mit der Query-Komponente endet.

Die einzelnen Parameter und Datensätze darin, werden mit dem kaufmännischen & separiert. [24]

### 6.7.7.3 Senden an den Server

Zusammensetzung des Query-Strings, für das Senden der Fahrtinformationen an den Server:

- 1) carID, damit die Daten dem korrekten Fahrzeug in der Datenbank zugeordnet werden können
- 2) Datum und Uhrzeit, um die Werte der Buchung zu diesem Zeitpunkt zuordnen zu können
- 3) Längengrad in dezimaler Schreibweise
- 4) Breitengrad in dezimaler Schreibweise
- 5) Meereshöhe in Metern
- 6) Geschwindigkeit in km/h

Eine URL, welche vom GSM-Modul aufgerufen wird, kann dann folgendermaßen aussehen:

<https://lpcs.at/receive/GSMBridge.php?carid=GCDE59IkvQetYw9z&date=20220317122200&lat=47.821661&lng=13.044436&altitude=421.0&speed=27.3>

Hier wird das „GSMBridge.php“-File aufgerufen. Ihm werden eine carID, das Datum 17.03.2022 – 12:22, Längen- und Breitengrad, sowie eine Meereshöhe von 421 Metern und eine Fahrtgeschwindigkeit von 27,3 km/h als Parameter im Query-String übergeben.

Damit die Daten vom Mikrocontroller an das GSM-Modul gelangen, werden diese mit UART verbunden.

### 6.7.8 Gehäuse

Für das Tracking-Modul wird ein spezielles Gehäuse angefertigt. Dieses besteht nicht, wie gewöhnlich aus einem Boden und einem Deckel, sondern hat zusätzlich zwei weitere Schichten dazwischen. Ursache dafür ist, dass es drei Lagen an Platinen gibt. In der untersten Lage befindet der Arduino, welcher dann zwischen Boden und dem zweiten Gehäuseteil eingeklemmt und somit befestigt wird. Auf den Arduino wird dann die Platine gesteckt, welche die einzelnen PINS mit denen der Module (GSM, GPS, SD) verbindet. Dieser Print wird dann mit der dritten Schicht befestigt. Auf dieser liegen dann die einzelnen Module auf. Damit diese befestigt werden und niemand hineingreifen kann, kommt darauf dann der Deckel, welcher das Gehäuse abschließt.

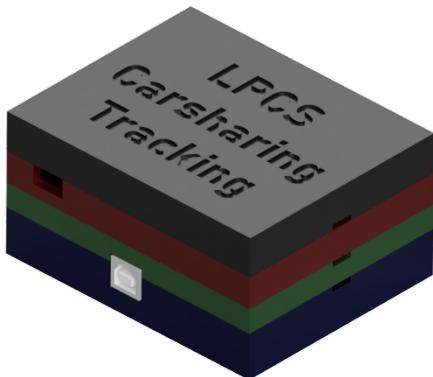


Abbildung 6.7.10 Tracking-Gehäuse vorne

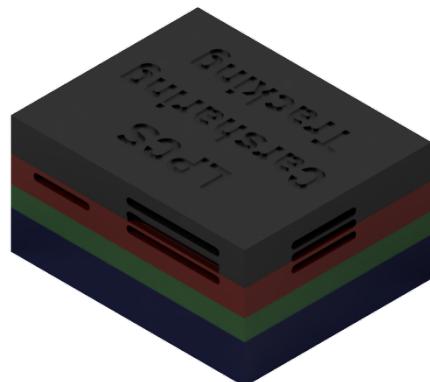


Abbildung 6.7.11 Tracking-Gehäuse hinten

## 6.8 Schlüsselbox

Die Schlüsselbox ist ein modular erweiterbarer Aufbewahrungsplatz für Autoschlüssel.

Aufgebaut ist diese Einheit aus dem zentralen Steuerelement, einem Arduino, der Real-Time-Clock, einem Eingabefeld und einer Anzeige, sowie der Speichereinheit für die Timecodes. Da es modular erweiterbar sein muss, sind diese Elemente in einem Würfel verbaut. An diesen können dann nach oben, unten, links oder rechts immer wieder neue Boxen für die Schlüssel angesteckt werden. Diese beinhalten nur mehr einen kleinen Chip, welcher die Aufgabe hat, seine Box zu öffnen, wenn er das Signal dazu bekommt.

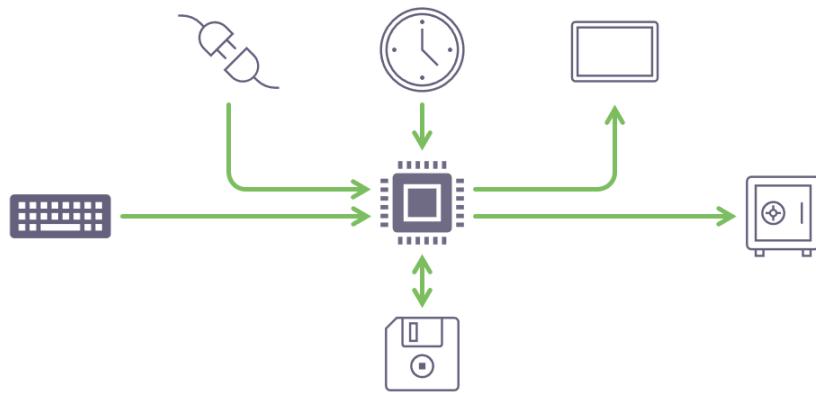


Abbildung 6.8.1 Aufbau der Schlüsselbox

### 6.8.1 Steuerelemente

Die Schlüsselbox wird von einem Arduino gesteuert. Dieser generiert und verwaltet die Zeitcodes, liest das Eingabefeld aus und sendet einer Box ein Signal, wenn sich diese öffnen sollte. In jeder Box befindet sich noch ein kleiner Mikrocontroller, welcher lediglich dafür zuständig ist, diese Box zu öffnen, wenn er vom Hauptcontroller das Signal dazu bekommt.

## 6.8.2 Energieversorgung

Versorgt wird die Schlüsselbox über ein Netzteil, welches an einer handelsüblichen Steckdose eingesteckt werden kann. Dieses liefert der Box eine Gleichspannung von 12V, die später über interne Regler auf die gewünschten Spannungen umgewandelt werden. Die meisten Module brauchen hier 5V Gleichspannung. Die Verriegelungseinheit benötigt jedoch eine Versorgung von 6V DC.

## 6.8.3 Öffnungscode-Generierung

Der Arduino liest die aktuelle Uhrzeit über I2C von der Real Time Clock (DS3231) bis auf eine Minute genau aus und wandelt es in den Unixtimestamp um. Das sind die vergangenen Sekunden seit dem 01. Jänner 1970 um 00:00. Daraus wird dann der 6-stellige Schlüsselboxcode generiert.

### 6.8.3.1 Algorithmus

Der Zeitcode wird nach dem folgenden Algorithmus berechnet:

- a) Ein Datum zum Berechnen des Unixtimestamp muss gewählt werden. Für den 01.Mai 2025 – 14:14 zum Beispiel ist dieser Wert **1746101640**.
- b) Diese Zahl wird anschließend in drei Teile zerlegt.
  1. Letzten beiden Stellen (Einer und Zehner) – **1746101640**
  2. Tausender und Zehntausenderstelle – **1746101640**
  3. Millionen und Zehnmillionenstelle – **1746101640**
  4. Die vierte Zahl ergibt sich aus der ID der Box – Box-ID (z.B.: 04)
- c) Zu diesen Zahlen wird 7 addiert und ein Produkt gebildet:
$$(40 + 7)^3 \times (01 + 7)^2 \times (46 + 7) \times (04 + 7) = 3\ 873\ 843\ 776$$
- d) Zuletzt wird dieses Ergebnis durch 10.000 dividiert. Der Rest dieser Division ist dann der Zeitcode.
$$3\ 873\ 843\ 776 \text{ modulo } 10\ 000 = 3776$$

### 6.8.3.2 Schlüsselboxcode

Der endgültige Code ergibt sich dann aus der Box-ID für die ersten zwei Stellen im Code und dem generierten Zeitcode.

$$\text{Öffnungscode} = \text{BoxID} \times 10\ 000 + \text{Zeitcode}$$

$$\text{Öffnungscode} = 04 \times 10\ 000 + 3776 = 043776$$

So ergibt sich für den 01.Mai 2025 um 14:14 für die Box mit der ID 4 der Schlüsselboxcode 0 4 3 7 7 6

## 6.8.4 Codeeingabe

Bucht ein Nutzer ein Fahrzeug, so erhält er einen Code, welcher im Bereich des Beginns seiner Buchung gültig ist. Das heißt, gibt er im Ziffernfeld den Code ein, der zu diesem Zeitpunkt gültig ist, so öffnet sich die entsprechende Box, welche den Schlüssel für das gewählte Auto enthält.

Damit bei der Eingabe erkannt werden kann, was gedrückt worden ist, werden die Zahlen auf einem sechsstelligen Siebensegment-Display ausgegeben. Ein anspruchsvolleres Display wird nicht benötigt, da Buchstaben oder andere Sonderzeichen nicht angezeigt werden müssen.

## 6.8.5 Schlüsselabholung und Rückgabe

### 6.8.5.1 Verriegelungseinheit

Als Schloss für die einzelnen Boxen wird auf ein elektronisches zurückgegriffen, welches im Leerlauf standardmäßig ausgefahren ist, sodass die Box ohne Strom immer verriegelt ist. In die nähere Auswahl sind dafür diese beiden Einheiten gefallen, da sie nur 6V als Versorgungsspannung beanspruchen:

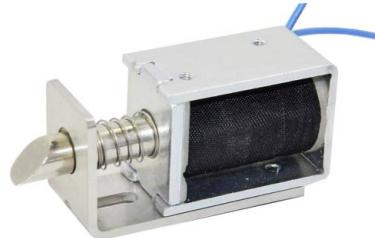


Abbildung 6.8.2 Schloss A [26]



Abbildung 6.8.3 Schloss B [25]

Der Vorteil von Schloss B ist, dass es billiger ist und weniger Strom zum Öffnen verbraucht. [25]

Schloss A hat im Gegenzug den Vorteil, dass es weiter eingefahren werden kann. Während Variante B nur etwa 2mm Unterschied zwischen dem eingefahrenen und dem ausgefahrenen Zustand hat, schafft es das andere auf 8mm. [26]

Aus diesem Grund wird Schloss A gewählt, da dadurch die Fläche erhöht wird, die zwischen dem Bolzen und der Wand der Box ist. Und eine größere Fläche bedeutet weniger Druck, sollte jemand an der Tür anziehen, wenn diese geschlossen ist. Der geringere Druck bringt dann wiederum den Vorteil, dass die Wand mehr Kraft aushält, als würde Schloss B verwendet werden.

### 6.8.5.2 Speichereinheit

Wenn ein gültiger Code eingegeben wird und eine Box geöffnet wird, speichert sich das System diesen Code. Dieser wird so lange gespeichert, bis er erneut eingegeben wird. Der Grund für diese Speicherung ist, dass die Schlüsselbox keine Internetverbindung hat und somit nicht weiß, wie lange eine Buchung dauern wird. Da der Nutzer den Schlüssel am Ende der Buchung aber mit dem gleichen Code zurückgibt, muss dieser gespeichert werden, da zu diesem späteren Zeitpunkt ansonsten ein anderer Zeitcode berechnet worden wäre und die Box mit dem alten Code nicht mehr geöffnet werden könnte.

Um den Speicher nicht zu überfüllen, werden Codes, die älter als 72 Stunden sind, gelöscht.

## 6.8.6 Gehäuse

Das Gehäuse der Schlüsselbox muss robust sein, dass es nicht einfach aufgebrochen werden kann. Weiters ist es modular erweiterbar, weshalb Vertiefungen, ähnlich eines LEGO®-Steins, am Gehäuse zu finden sind. Diese gewährleisten einen zusätzlichen Halt für die Verbindungen.

## 6.9 UNIT-Test

UNIT-Tests prüfen Units, das sind einzelne Komponenten (Bestandteile) des gesamten Projekts. Für gewöhnlich werden diese Tests auf Software angewendet und sind vollkommen automatisiert. Grund dafür ist, dass die Überprüfung des Codes keinen größeren Aufwand mehr verursachen und mit einem Knopfdruck abgehandelt werden soll. Es bleibt jedoch nicht erspart, diese Tests einmal zu formulieren und in ein Programm zu übertragen. Später muss man dann aber nicht mehr jeden Test händisch wieder und wieder durchführen, bis jede Komponente einwandfrei funktioniert, was enorm Zeit einspart und trotzdem die Qualität sichert. [27]

Bei diesem Projekt könnte ein Unit-Test zum Beispiel die Generierung des Zeitschlüssels für die Keybox überprüfen. Dafür könnte man eine Vielzahl an Zeitstempeln, für die man die dazugehörigen Codes kennt, in das System schicken und im Anschluss überprüfen, ob dieses die gleichen Codes generiert hat.

Der Vorteil eines automatisierten Tests ist hier klar erkennbar. Ein solches Programm würde für 100 Test, inklusive Validierung des Systemoutputs nur wenige Sekunde benötigen. Nimmt man im Vergleich einen Menschen, der schnellstens alle 10 Sekunden eine Zeiteingabe schafft, so würde er um die 16 Minuten für 100 Eingaben benötigen. In dieser Zeit hat er die kalkulierten Codes des Systems allerdings noch nicht überprüft.

## 6.10 App-Security

In der heutigen Zeit werden App- und Datensicherheit immer wichtige Themen. Je mehr digitalisiert wird und je mehr personenbezogene Daten dadurch im Internet landen und gespeichert werden, desto wichtiger ist es diese Daten auch zu schützen. Ansonsten können sensible Daten für Hacker leicht zugänglich sein.

Um keine Schlupflöcher offenzulassen, sind geeignete Gegenmaßnahmen zu treffen. Dafür sollte man zuerst die größten Sicherheitsrisiken im Internet evaluieren, um zu Beginn diese abzusichern. Dafür lohnt es sich, die Einschätzung des OWASP anzusehen. Diese stufen alle 4 Jahre die 10 aktuell schwerwiegendsten Sicherheitslücken neu ein. Weiters beschreiben sie, wie diese Zustände kommen und bieten Lösungsvorschläge an. [28]

## 7 Ergebnisse – Abnahme

Der jetzige Projektstand beinhaltet eine funktionierende Website mit Buchungssystem, Fahrzeug- und Nutzerverwaltung. Nutzer können sich anmelden oder registrieren und im System Fahrzeuge buchen, ihre Kontodaten bearbeiten und die monatlichen Kosten und wie diese zustandekommen einsehen. Administratoren haben die Möglichkeit, Fahrzeuge anzulegen und zu editieren. Weiters können sie auch die Aktivitäten der Benutzer ansehen, deren Daten bearbeiten und sie aus dem System entfernen. Im Control-Panel können außerdem noch diverse Einstellungen getroffen werden – Wie lange darf eine Buchung sein? Wie viel kostet ein Fahrtkilometer? Können Fahrzeuge momentan gebucht werden? – Um drei zu nennen.

Die Anzeige des Fahrzeugstandorts ist zum aktuellen Zeitpunkt noch nicht möglich, da das Tracking-Modul für die Fahrzeuge bisher nur den Standort empfangen kann, diesen aber noch nicht an den Server übermittelt.

The screenshot shows a web browser window with the URL [lpcs.at/demo/bookings/new/](http://lpcs.at/demo/bookings/new/). The page title is "Neue Buchung". A progress bar at the top indicates "Schritt 1" (Step 1) is completed, "Schritt 2" (Step 2) is in progress, and "Schritt 3" (Step 3) is pending. The main form has three sections: "Datum" (Date) showing a calendar for March 2022 with the 24th selected; "Details" showing a dropdown for "Fahrzeug" set to "Porsche Taycan" and another dropdown for "Geschätzte Fahrstrecke" set to "Weniger als 75 km"; and "Uhrzeit" (Time) showing dropdowns for "Fahrzeugabholung" and "Fahrzeugrückgabe", both currently set to "Bitte wählen...".

Abbildung 7.1.1: Aktueller Stand der Website



Abbildung 7.1.2: Tracking-Modul in einem Gehäuse

Anforderung	Durchgeführte Tests	Testergebnis	Freigabe (J/N)?
Anmeldung und Registrierung auf das Benutzersystem	T0010	Es ist möglich sich anzumelden oder sich neu zu registrieren.	J
Kontodaten bearbeiten oder löschen	T0020	Ein angemeldeter Nutzer kann seine Daten bearbeiten und sein Konto löschen, wenn er dies wünscht.	J
Reservierung und Stornierung eines Fahrzeuges für berechtigte Nutzer	T1010	Nur einem freigeschalteten Nutzer mit einer berechtigten Rolle ist es möglich, Fahrzeuge zu buchen oder diese auch wieder zu stornieren. Ein Administrator kann Buchungen ebenfalls stornieren.	J
Verwaltung der Fahrzeuge durch einen Administrator	T2010	Nur der Administrator kann ein neues Fahrzeug anlegen, dessen Daten editieren oder es aus dem System löschen.	J
Systembenachrichtigungen vom Administrator	T3110	Im Control-Panel kann der Admin Nachrichten verfassen, die an Anwender der Software gesendet werden. Dabei können diese an alle versandt werden, oder nur an eine ausgewählte Gruppe.	J
Responsive Ansicht	T4010	Egal ob die Website auf einem großen oder kleinen Bildschirm aufgerufen wird – sie passt sich immer darauf an. So kann von einem Smartphone, sowie von einem Tablet und Computer auf das System zugegriffen werden.	J
Fahrtenbuch	T3010	Standort wird empfangen. Das Senden an den Server bereitet jedoch noch Schwierigkeiten.	N

Tabelle 7.1.1: Anforderungen und Testergebnisse des Projekts

## 8 Quellen- und Literaturverzeichnis

---

- [1] B. Kahlbrandt, Software-Engineering mit der Unified Modeling Language (2. Aufl.), Berlin: Springer-Verlag, 2013.
- [2] A. Diehl, „Was ist Scrum? - Eine kompakte Einführung in die Scrum Methode,“ Digitale Neuordnung, 12 2019. [Online]. Available: <https://digitaleneuordnung.de/blog/scrum-methode/>. [Zugriff am 17 03 2022].
- [3] J. Dr. Fleig, „Agiles Projektmanagement - So funktioniert Scrum,“ Business-Wissen, 27 08 2021. [Online]. Available: <https://www.business-wissen.de/artikel/agiles-projektmanagement-so-funktioniert-scrum/>. [Zugriff am 17 03 2022].
- [4] thomi@inbox.com, „Issue 841105: Security: uXSS in Chrome on iOS,“ 09 05 2018. [Online]. Available: <https://bugs.chromium.org/p/chromium/issues/detail?id=841105>. [Zugriff am 14 03 2022].
- [5] P. Hudson, „How to create a List or a ForEach from a binding,“ Hacking with Swift, 19 07 2021. [Online]. Available: <https://www.hackingwithswift.com/quick-start/swiftui/how-to-create-a-list-or-a-foreach-from-a-binding>. [Zugriff am 13 03 2022].
- [6] „React Native,“ Meta Platforms, Inc., [Online]. Available: <https://reactnative.dev/>. [Zugriff am 02 03 2022].
- [7] J. Bauerfeind, „Grün,.. – Farben und Symbolik – Wirkung und Bedeutung,“ solaga, 02 11 2020. [Online]. Available: <https://www.solaga.de/2020/11/gruen-farben-und-symbolik-wirkung-und-bedeutung-2/>. [Zugriff am 13 03 2022].
- [8] T. Preston-Werner, „Semantic Versioning 2.0.0,“ [Online]. Available: <https://semver.org/lang/de/>. [Zugriff am 08 03 2022].
- [9] C. Weber, „Was sind eigentlich Push Notifications?,“ VERTUS, 15 12 2021. [Online]. Available: <https://vertus.co/push-notifications/>. [Zugriff am 27 03 2022].
- [10] @JonyIveParody, 23 08 2019. [Online]. Available: <https://twitter.com/jonyiveparody/status/1164931733889748992?s=21>. [Zugriff am 07 03 2022].
- [11] „Nächste Milliardenfusion in Chipbranche steht,“ Handelsblatt, 20 01 2016. [Online]. Available: <https://www.handelsblatt.com/unternehmen/it-medien/microchip-technology-uebernimmt-atmel-naechste-milliardenfusion-in-chipbranche-steht/12854458.html>. [Zugriff am 22 03 2022].
- [12] Free Software Foundation, Inc., „GNU General Public License 3,“ GNU, 29 06 2007. [Online]. Available: <https://www.gnu.org/licenses/gpl-3.0.html>. [Zugriff am 22 03 2022].

- [13] S. Labesius, „Unterschied Open Source und Freeware,“ ifrOSS, 23 01 2010. [Online]. Available: <https://www.ifross.org/?q=unterscheidet-sich-open-source-software-freeware-public-domain-und-shared-source>. [Zugriff am 23 03 2022].
- [14] „IntelliJ IDEA - Leistungsfähige und ergonomische IDE für JVM,“ JetBrains, [Online]. Available: <https://www.jetbrains.com/de-de/idea/>. [Zugriff am 23 03 2022].
- [15] „Visual Studio Code,“ Microsoft, [Online]. Available: <https://code.visualstudio.com/docs>. [Zugriff am 23 03 2022].
- [16] J. Weissgärber, „Die UART Schnittstelle,“ 02 03 2019. [Online]. Available: [http://www.mathe-mit-methode.com/schlaufuchs\\_web/elektrotechnik/mikrocontroller\\_lernmaterial/microcontroller\\_allgemein/mikrocontroller\\_ext\\_hardware/mikrocontroller\\_uart.html](http://www.mathe-mit-methode.com/schlaufuchs_web/elektrotechnik/mikrocontroller_lernmaterial/microcontroller_allgemein/mikrocontroller_ext_hardware/mikrocontroller_uart.html). [Zugriff am 27 03 2022].
- [17] „Serial Peripheral Interface,“ mikrocontroller.net, [Online]. Available: [https://www.mikrocontroller.net/articles/Serial\\_Peripheral\\_Interface](https://www.mikrocontroller.net/articles/Serial_Peripheral_Interface). [Zugriff am 27 03 2022].
- [18] „I2C,“ mikrocontroller.net, [Online]. Available: <https://www.mikrocontroller.net/articles/I%C2%B2C>. [Zugriff am 27 03 2022].
- [19] „Wie funktioniert Satellitennavigation?,“ magicmaps, [Online]. Available: <https://www.magicmaps.de/gnss-wissen/wie-funktioniert-gps/?L=0>. [Zugriff am 25 03 2022].
- [20] „Präzise Positionsbestimmung mit Hilfe von GPS / GNSS,“ magicmaps, [Online]. Available: <https://www.magicmaps.de/gnss-wissen/praezise-gps-messungen-mit-hilfe-von-dgps-und-rtk/?L=0>. [Zugriff am 25 03 2022].
- [21] G. Baddeley, „GPS - NMEA sentence information,“ aprs.gids, 20 07 2001. [Online]. Available: <http://aprs.gids.nl/nmea/>. [Zugriff am 27 03 2022].
- [22] M. Hart, „TinyGPSPlus,“ GitHub, 24 01 2022. [Online]. Available: <https://github.com/mikalhart/TinyGPSPlus>. [Zugriff am 2022 04 01].
- [23] A. Prof. DI Jerey, „GSM-KSN\_05Jahrgang,“ in *GSM.pdf*, PDF-Dokument.
- [24] K. Pfliegl, „Internetadressen — so sind sie aufgebaut,“ com!, 01 07 2013. [Online]. Available: <https://www.com-magazin.de/praxis/profi-wissen/internetadressen-so-aufgebaut-158054.html>. [Zugriff am 24 03 2022].
- [25] Intertec, „ITS-VEL-1510B-06VDC Verriegelungseinheit,“ conrad, [Online]. Available: <https://www.conrad.at/de/p/intertec-its-vel-1510b-06vdc-verriegelungseinheit-6-v-dc-1-8-w-2109465.html>. [Zugriff am 25 03 2022].
- [26] Intertec, „ITS-VEL-2620B-06VDC Verriegelungseinheit,“ conrad, [Online]. Available: <https://www.conrad.at/de/p/intertec-its-vel-2620b-06vdc-verriegelungseinheit-6-v-dc-3-84-w-2109460.html>. [Zugriff am 25 03 2022].

- [27] „Was ist ein Unit-Test?“ it-agile, [Online]. Available: <https://www.it-agile.de/agiles-wissen/agile-entwicklung/unit-tests/>. [Zugriff am 01 04 2022].
- [28] „OWASP Top Ten,“ OWASP, 2021. [Online]. Available: <https://owasp.org/www-project-top-ten/>. [Zugriff am 01 04 2022].
- [29] „Scrum-Prozess,“ Business-Wissen, [Online]. Available: [https://www.business-wissen.de/res/artikel/Scrum-Prozess\\_business-wissen.png](https://www.business-wissen.de/res/artikel/Scrum-Prozess_business-wissen.png). [Zugriff am 18 03 2022].
- [30] J. Buis, „Angular Sandbox Escape,“ 03 2016. [Online]. Available: <https://gist.github.com/jeremybuis/38c01acae19fc2ac6959>. [Zugriff am 15 03 2022].
- [31] „Client Bug Bounty Program,“ Mozilla, [Online]. Available: <https://www.mozilla.org/en-US/security/client-bug-bounty/>. [Zugriff am 14 03 2022].
- [32] J. Barnes, „Why is kotlin a bad programming language?,“ Quora, 11 11 2018. [Online]. Available: <https://www.quora.com/Why-is-kotlin-a-bad-programming-language/log>. [Zugriff am 14 03 2022].
- [33] C. Weber, „Vertus,“ 15 12 2021. [Online]. Available: <https://vertus.co/push-notifications/>.
- [34] R. Rahner, „Der I2C Bus,“ [Online]. Available: <https://www.rahner-edu.de/grundlagen/signale richtig-verstehen/i2c-besser-verstehen-1/>. [Zugriff am 01 04 2022].
- [35] Axel, „Das Serial Peripheral Interface,“ Attraktor e.V., 04 06 2012. [Online]. Available: <https://docplayer.org/8255678-Serielle-kommunikation-mit-dem-arduino-teil-1-das-serial-peripheral-interface-spi.html>. [Zugriff am 01 04 2022].
- [36] „Arduino Uno SMD,“ cablematic, [Online]. Available: <https://cablematic.com/en/products/arduino-uno-smd-development-board-AR107/>. [Zugriff am 01 04 2022].
- [37] S. F., „32 GPIO Sender und Empfänger via Bus/Ethernet,“ mikrocontroller, 14 03 2021. [Online]. Available: <https://www.mikrocontroller.net/topic/514768>. [Zugriff am 01 04 2022].

## 9 Verzeichnis der Abbildungen und Tabellen

---

Tabelle 3.7.1: Die 5 Qualitätsbestimmungen von LPCS.....	21
Tabelle 5.2.1: DNS-Einträge.....	29
Tabelle 7.1.1: Anforderungen und Testergebnisse des Projekts.....	82
 Abbildung 1.1.1: LPCS Carsharing System Logo .....	4
Abbildung 1.1.2: LPCS Carsharing System logo .....	6
Abbildung 2.2.1: Aufbau des LPCS Carsharing Systems .....	13
Abbildung 3.5.1: Anzeige der Nutzeraktivitäten im Admin-Control-Panel.....	19
Abbildung 3.5.2: Sicherungsdatei mit Zeit- und Standortdaten am Tracking-Modul.....	19
Abbildung 4.2.1: Gantt-Diagramm – Noah Laireiter .....	25
Abbildung 4.2.2: Gantt-Diagramm – Thomas Ploder .....	26
Abbildung 4.3.1: Scrum-Prozess [29] .....	27
Abbildung 5.1.1: Systemaufbau und Aufgabenbereiche (Thomas Ploder) .....	28
Abbildung 5.3.1: ER-Diagramm der entworfenen Datenbank.....	30
Abbildung 5.3.2: Implementierung der md5-Funktion. ....	31
Abbildung 5.3.3: Ablauf eines Passwort-Resets. ....	32
Abbildung 5.3.4: Passwort zurücksetzen mit Email-Code. ....	32
Abbildung 5.3.5: Datenbankeintrag des Nutzers „max.mustermann@lpcs.at“.....	33
Abbildung 5.3.6: Eingeloggter Nutzer „max.mustermann@lpcs.at“ betrachtet Benutzerkonto. ....	33
Abbildung 5.3.7: Veranschaulichung der Datenänderung .....	34
Abbildung 5.3.8: Manipulierte Eingabe im Textfeld.....	34
Abbildung 5.3.9: Prozess einer Injection .....	34
Abbildung 5.3.10: Datenbankeintrag des Nutzers „max.mustermann@lpcs.at“ nach der SQL-Injection .	34
Abbildung 5.3.11: Möglichkeit zur Prävention von SQL-Injection und XSS in der Sprache PHP.....	35
Abbildung 5.3.12: Anwendung der Methode „handleParameter“ in der Sprache PHP .....	35
Abbildung 5.3.13: Ausschnitt aus einem AngularJS-File.....	36
Abbildung 5.3.14: Sandbox-Escape unter AngularJS. [12].....	36
Abbildung 5.3.15: Sandbox-Escape unter AngularJS. [12].....	36
Abbildung 5.3.16: Ausnutzen der XSS-Sicherheitslücke .....	37
Abbildung 5.3.17: Auswirkung der XSS-Sicherheitslücke .....	37
Abbildung 5.3.18: Ausnutzen der XSS-Sicherheitslücke .....	37
Abbildung 5.3.19: Auswirkung der XSS-Sicherheitslücke .....	37
Abbildung 5.3.20: Ausnutzen einer uXSS-Vulnerability.....	38
Abbildung 5.3.21: „Bug Bounty Program“ von Mozilla. [31] .....	38
Abbildung 5.3.22: Einstellungsmöglichkeiten im LPCS Control-Panel.....	40
Abbildung 5.3.23: Konfigurierbarkeit des Impressums.....	40
Abbildung 5.4.1: Screenshot aus dem Control-Panel.....	41
Abbildung 5.4.2: Beispiel einer systemgenerierten Nachricht.....	42
Abbildung 5.4.3: Prozess der Buchung.....	43
Abbildung 5.5.1: Xcode Logo.....	44
Abbildung 5.5.2: Einfaches Code-Schnippel der Programmiersprache Objective-C.....	44
Abbildung 5.5.3: Swift Logo.....	45
Abbildung 5.5.4: Code-Schnippel mit Vergleich von Swift 2 und Swift 3.....	45
Abbildung 5.5.5: Deklaration einer Optional-Variable.....	45
Abbildung 5.5.6: Veranschaulichung von Optionals.....	45

Abbildung 5.5.7: Beispiel einer Liste mit SwiftUI .....	46
Abbildung 5.5.8: Einsteigerbeispiel in React Native. [6] .....	47
Abbildung 5.5.9: Ladeanimation in der Übersicht.....	49
Abbildung 5.5.10: Veranschaulichung von Fehlern beim Laden. ....	49
Abbildung 5.5.11: Beispiel ein jener Implementierung in Java .....	50
Abbildung 5.5.12: Veranschaulichung von UITableViewCellStyleActions. ....	50
Abbildung 5.5.13: Gegenüberstellung Light- und Dark-Mode der LPCS iOS-App.....	51
Abbildung 5.5.14: Farbpaletteneinstellung einer definierten Farbe in Xcode. ....	51
Abbildung 5.5.15: Veranschaulichung einer (Push-)Notification. ....	52
Abbildung 5.5.16: Veranschaulichung von Push-Notifications.....	53
Abbildung 5.5.17: Beispiel für einen FCM-Tokens (152 Symbole). ....	53
Abbildung 5.5.18: Das Hook-Model. ....	54
Abbildung 5.5.19: Ausschnitt aus den AppStore-Guidelines [10]. ....	54
Abbildung 5.5.20: Apple verstößt gegen die eigenen Richtlinien. ....	55
Abbildung 5.5.21: Initialisieren einer CocoaPod-File. ....	55
Abbildung 5.5.22: Inhalt der „Podfile“. ....	55
Abbildung 5.5.23: Installieren der in der Podfiles gelisteten Libraries.....	55
Abbildung 5.5.24: Projektkonfiguration.....	56
Abbildung 5.5.25: Projektkonfiguration. ....	57
Abbildung 5.5.26: Firebase Console. ....	57
Abbildung 5.5.27: Ausschnitt der AppDelegate. ....	58
Abbildung 5.5.28: Dialogfenster zu Notifications.....	58
Abbildung 5.5.29: Firebase in Android Studio-Projekt integrieren. ....	59
Abbildung 5.5.30: Ausschnitt aus der AndroidManifest.xml-File.....	59
Abbildung 5.5.31: Ausschnitt aus FirebaseHandler.java. ....	60
Abbildung 5.5.32: Sinnlose Notifications. ....	60
Abbildung 5.5.33: Aufzurufender Endpoint .....	60
Abbildung 5.5.34: Benötigten Header .....	61
Abbildung 5.5.35: Body des Requests .....	61
Abbildung 5.5.36: Body des Requests .....	61
Abbildung 5.5.37: Erfolgreicher Zustellversucht. ....	61
Abbildung 5.5.38: Fehlgeschlagener Zustellversuch. ....	62
Abbildung 6.1.1 Systemaufbau und Aufgabenbereiche (Noah Laireiter).....	64
Abbildung 6.2.1 Schaltplan in EAGLE.....	65
Abbildung 6.2.2 Leiterplattenlayout in EAGLE .....	65
Abbildung 6.2.3 Gehäuseentwurf in Fusion360 .....	66
Abbildung 6.3.1 Atmel-Studio Logo [11] .....	67
Abbildung 6.3.2 chip45 Bootloader.....	67
Abbildung 6.3.3 IntelliJ Logo [14] .....	68
Abbildung 6.3.4 VSCode Logo [15] .....	68
Abbildung 6.5.1 crumb644-Modul [37] .....	69
Abbildung 6.5.2 Arduino UNO [36] .....	69
Abbildung 6.6.1 UART-Verbindung [16] .....	70
Abbildung 6.6.2 SPI-Verbindung [35] .....	70
Abbildung 6.6.3 I2C-Verbindung [34] .....	70
Abbildung 6.7.1 Aufbau des Tracking-Moduls .....	71
Abbildung 6.7.2 Schnittpunkt zur Positions berechnung [19] .....	72
Abbildung 6.7.3 GPS-Fehlerquellen [18] .....	72
Abbildung 6.7.4 Präzision der GPS-Korrekturmethoden [20] .....	72

Abbildung 6.7.5 Auslesen der Daten aus dem NMEA-String.....	74
Abbildung 6.7.6 Erstellung einer URL aus den Daten.....	74
Abbildung 6.7.7 Zellenstruktur in Österreich [21].....	75
Abbildung 6.7.8 Zeitmultiplex und Rahmenstruktur bei GSM [21] .....	75
Abbildung 6.7.9 Aufbau einer Internetadresse [20].....	76
Abbildung 6.7.10 Tracking-Gehäuse vorne .....	77
Abbildung 6.7.11 Tracking-Gehäuse hinten .....	77
Abbildung 6.8.1 Aufbau der Schlüsselbox.....	77
Abbildung 6.8.2 Schloss A [24] .....	79
Abbildung 6.8.3 Schloss B [23] .....	79
Abbildung 7.1.1: Aktueller Stand der Website .....	81
Abbildung 7.1.2: Tracking-Modul in einem Gehäuse .....	81

## 9.1 Glossar

• adb	Android Debug Bridge
• Ajax	Asynchronous JavaScript and XML
• Android Studio	Entwicklungsumgebung
• Android Volley	Networking-Library
• apache2	Webserver
• APNs	Apple Push Notification Services
• Appicon.co	xcassets Generierungstool
• AppStoreConnect	AppStore Zugriffsportal für Publisher
• AVR	8-bit Mikrocontroller-Familie von Microchip
• Bootstrap	CSS Framework
• Bootstrap DatePicker	CSS Framework
• Chrome Developer Tools	Entwicklertool
• CocoaPods	Dependencymanager
• C/C++	Programmiersprachen für Mikrocontroller
• CSS	Cascading Style Sheet
• DataGrip	Datenbankverwaltungstool
• DGPS	Differential Global Positioning System
• Drop-out-Voltage	Differenzspannung: Ein- zu Ausgang des Spannungsreglers
• Duplex-Betrieb	richtungsunabhängige Kommunikation (halb, duplex, voll)
• FileZilla	(S)FTP Dateübertragungsprogramm
• Firebase	Entwicklungstool für Web- & Appanwendungen
• Firebase Messaging	Dienst der Firebase
• flashen	programmieren eines Mikrocontrollers mit einem .hex-File
• Git	Datenverwaltungstool
• GitLab	Webanwendung zur Versionsverwaltung
• GNU	GNU General Public License
• Google Analytics	Dienst der Google Cloud
• GPRS	General Packet Radio Service
• GPS	Global Positioning System
• GSM	Global System for Mobile Communications
• .hex	Kompiliertes C/C++ File für Mikrocontroller
• HLR	Home Location Register
• Homebrew	Paketmanager
• HTML	Hyper Text Markup Language
• I2C	Inter-Integrated Circuit
• IDE	Integrated Development Environment
• IDEA	Integrated Development Environment by IntelliJ
• iTerm	Terminal für macOS
• Java JDK	Java Development Kit
• JavaScript	Skriptsprache
• jQuery	JavaScript Library
• JSON	JavaScript Object Notation
• Kotlin	Programmiersprache
• Let's Encrypt	SSL Zertifizierungsstelle
• Linux/Debian 9	Betriebssystem

• macOS	Betriebssystem
• MAMP	Webservertool
• Master / Slave	Hierarchische Verwaltung des Zugriffs auf gemeinsame Speicher
• Material.io	Designlanguage
• Material Icons	Icons entsprechend dem Material Design
• md5	Einweg-Verschlüsselungsversion
• moment.js	JavaScript Library
• MSC	Mobile Switching Center
• mySQL	Datenbankverwaltungssystem
• nginx	Webserver
• Node.js	JavaScript Laufzeitumgebung
• npm	JavaScript Paketmanager für Node.js
• Objective-C	Programmiersprache
• OhMyZsh	Framework zur Zsh-Konfiguration
• OWASP	The Open Web Application Security Project
• PHP	Hypertext Preprocessor
• PHPMailer	Library für PHP
• phpmyadmin	Verwaltungstool für Datenbanken
• PhpStorm	Entwicklungsumgebung
• PIN	Ein- Ausgang auf einem Hardware-Element
• Play Console	PlayStore Zugriffspunkt für Publisher
• Postman	API Platform
• regex	Regular Expression
• Roundcube	Mailserver
• RTK	Real-Time Kinematic
• SAM	Smart ARM based Microcontroller
• SDWebImage	Library für iOS
• SF Symbols	Icons entsprechend dem UIKit-Design
• SPI	Serial Peripheral Interface
• Swift	Programmiersprache
• SwiftyJson	Library für iOS-Entwicklung
• TestFlight	Beta-Test-Plattform
• UART	Universal Asynchronous Receiver-Transmitter
• UIKit	Basisframework für iOS-Entwicklung
• URL	Uniform Resource Locator
• USB	Universal Serial Bus
• VestaCP	Hosting Control-Panel
• Visual Studio Code	Entwicklungsumgebung
• WebStorm	Entwicklungsumgebung
• XAMPP	Webservertool
• Xcode	Entwicklungsumgebung

# 10 Begleitprotokoll gemäß § 9 Abs. 2 PrO

---

## 10.1 Begleitprotokoll Noah Laireiter

**Name:** Noah Laireiter  
**Diplomarbeitstitel:** LPCS – Carsharing System

KW	Beschreibung	Zeitaufwand
37	Endgültige Festlegung des Projektthemas	5h
38	Diplomarbeitsantrag erstellen	5h
39	Diplomarbeitsantrag einreichen, Besprechung	7h
40	Projektplanung, Aufgabenverfeinerung, Ganttdiagramm	5h
41	Pflichtenheft, Big Picture, Kostenkalkulation, Besprechung	8h
42	Blockschaltbilder Hardware, Auswertung NumPad-Matrix, Anfang Code	8h
43	NumPad-Code weiterarbeiten, Einlesen GPS-Code (AZ-Delivery)	9h
44	RTC und GPS auf crumbl programmieren, Besprechung	12h
45	GPS fertigstellen (crumbl), GSM-Modul und SD-Card-Reader einlesen	6h
46	GSM programmieren, Bausteine bestellen, Besprechung	5h
47	GSM-Modul Fehlersuche (+Step Down Wandler)	6h
48	Step-Down Wandler weiter entwerfen	4h
49	SD-Card-Reader auf Arduino programmieren, Besprechung	7h
50	SD-Card-Reader fertigstellen	4h
51	GPS-Modul auf Arduino umprogrammieren	6h
52	Wechsel auf neues GSM-Modul mit 5V Versorgungsspannung	5h
01	Einarbeiten in neues GSM-Modul	8h
02	Erster Test neues GSM-Modul	4h
03	Fehlersuche GSM-Modul, Besprechung	6h
04	Platine zusammenstellen (Tracking-Modul)	4h
05	Platine in Auftrag geben, löten und testen	5h
06	Platine Fehlersuche und Fehlerbehebung	5h
07	GPS-Modul testen an verschiedenen Positionen	4h
08	GPS-Modul Output optimieren, Tracking-Gehäuse beginnen, Besprechung	6h
09	GPS-Modul Fehlersuche	9h
10	GPS-Modul Fehlersuche, Tracking-Gehäuse weiterarbeiten	11h
11	Besprechung, Verfassen der Diplomarbeit und Refactoring	17h
12	Verfassen der Diplomarbeit und Refactoring	20h
13	Verfassen der Diplomarbeit und Refactoring	9h
14	DA-Abgabe am 04.04.	-

KW ...Kalenderwoche

----- ...Semesterabschnitt

## 10.2 Begleitprotokoll Thomas Ploder

**Name:** Thomas Ploder  
**Diplomarbeitstitel:** LPCS – Carsharing System

KW	Beschreibung	Zeitaufwand
37	Endgültige Festlegung des Projektthemas	5h
38	Diplomarbeitsantrag erstellen	10h
39	Diplomarbeitsantrag einreichen, Besprechung	10h
40	Projektplanung, Aufgabenverfeinerung, Ganttdiagramm	10h
41	Benutzersystem Login, Account editieren; Besprechung	10h
42	Registrieren, Control-Panel Nutzerverwaltung	22h
43	Berechtigungssystem	1h
44	KFZ anlegen; Besprechung	18h
45	Kalenderansicht entwerfen	5h
46	Buchungssystem: Buchung tätigen; Besprechung	9h
47	Buchungssystem: Buchung tätigen	5h
48	Buchungssystem Verwaltung	9h
50	Tarifsystem	2h
02	Buchungsdetails, Kostenkalkulation	6h
03	Buchungen und Verwaltung im Control-Panel	12h
04	Logging, Kalenderansichten	8h
05	Fahrzeugverwaltung, Zeitraumsperren	9h
06	Injection, Cross-Site-Scripting	10h
07	Finanzübersichten	4h
08	iOS-App Buchungen	14h
09	Bugfixing	8h
10	Diverse kleine Erweiterungen	3h
11	Verfassen der Diplomarbeit und Refactoring	13h
12	Verfassen der Diplomarbeit und Refactoring	11h
13	Verfassen der Diplomarbeit und Refactoring	6h
14	DA-Abgabe am 04.04.	-

**KW** ...Kalenderwoche

----- ...Semesterabschnitt

# 11 Besprechungsprotokolle

## 11.1 Besprechungsprotokoll 27.09.2021

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg

**Datum:** 27.09.2021

**Dauer:** 11:30 – 12:40 Uhr

**Schriftführer:** Thomas Ploder

**Teilnehmer:** Hr. Lehenauer (Auftraggeber)  
Prof. Hofmann (Betreuer)

Thomas Ploder (Projektteam)  
Noah Laireiter (Projektteam)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Übersicht der Situation. Besprechung bezüglich geteiltem Zugriff auf Besprechungsprotokolle usw.		Alle
2	I	<b>Offenen Fragen</b> Aufklärung und Besprechung über die offenen Fragen seitens Herrn Lehenauers.		Alle
3	A	<b>Lastenheft Besprechung</b> <ul style="list-style-type: none"> <li>- Klärung zum Thema Buchung ohne Konto (KANN).</li> <li>- Berechnung Ladungsdauer von E-Autos (KANN aufgrund Umwelteinflüsse etc.)</li> <li>- Durchbesprechung aller restlichen Punkte.</li> <li>- Genaue Evaluierung ob Kalenderansicht oder Tabelle in Zukunft.</li> <li>- „Statistik“ (z.B. wer am meisten fährt): SOLL</li> <li>- Klärung des Begriffs „Konfigurierbarkeit“ (z.B. Tarifkosten konfigurierbar)</li> <li>- Ränge: Systemadministrator, Chauffeur, Normaler Nutzer, Unterstützendes Mitglied, (Deaktivierter Nutzer)</li> <li>- Mitgliederliste soll in irgendeiner Form exportierbar sein. Für Excel – CSV? (Vor- &amp; Nachname, Email Geb.d., Tel., Eintrittsdatum)</li> <li>- Kostenübersicht im System (Betrag, Grund, Datum)</li> </ul>		Alle
4	I	<b>Kosten</b> Evaluierung der Betriebs- & Entwicklungskosten ausstehend. Weitere Besprechung beim nächsten Treffen.	15.10.2021	Projektteam, Hr. Lehenauer
5	I	<b>Lizenfrage</b> Dieses Thema wird nach Fertigstellung des Projektes genauer besprochen.		Projektteam, Hr. Lehenauer
6	B	Monatliches Treffen an der HTBLuVA Salzburg bezüglich den Projektfortschritt und Kontaktierung bei ersten nutzbaren Prototypen. Nächstes Treffen: 15.10.2021 um 13:45 Uhr	15.10.2021	Projektteam, Hr. Lehenauer

**Art:**

I	...Information
A	...Auftrag
B	...Beschluss

## 11.2 Besprechungsprotokoll 15.10.2021

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 15.10.2021 **Dauer:** 13:45 – 15:45 Uhr  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Thomas Ploder (Projektteam)  
 Prof. Hofmann (Betreuer) Noah Laireiter (Projektteam)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Einwände zur letzten Besprechung (Lizenzfrage beantworten) Übersicht der Situation. Besprechung bezüglich geteilten Zugriffes auf Besprechungsprotokolle usw.		Alle
2	B	<b>Lizenzfrage</b> Nutzung und Handhabung nach Vollendung des Maturaprojekts  <u>Vorschlag:</u> closed source, kostenfreie Nutzungsrechte für den Verein für eine gewisse Dauer, für neue Features muss neu besprochen werden  Projekt soll nicht von anderen kopiert werden können. <u>Vorschlag:</u> Lizenzrechte ausschließlich für Verein, Ploder, Laireiter  Lizenzfrage muss in nächsten Besprechungen weiterverfolgt werden.	12. Nov.	Alle
3	I	<b>Big Picture (Überblick)</b> Ersten Überblick über das Projekt geben		Projektteam, Hr. Lehenauer
4	I	<b>Pflichtenheft Besprechung</b> Einwände zur Tarifgestaltung: Probleme, wenn man deutlich vor Buchungsende zurückkommt zB.: Kosten minimieren, Buchung mit Begründung abbrechen <u>Vorschlag:</u> Man betrachtet zuerst einen Monat lang die Buchungen und evaluiert im Anschluss, ob diesem Problem nachgegangen werden muss.  Genaues Format der monatlichen Kostenabrechnung definiert  Absegnung des Pflichtenheftes		Projektteam, Hr. Lehenauer
5	I	<b>Vorführung bisheriger Arbeiten</b> Vorführung Benutzersystem Texte der Website werden in Zusammenarbeit mit Herrn Lehenauer formuliert.  Einbringung: Administratoren sollte sehen, was Benutzer kürzlich geändert haben		Projektteam, Hr. Lehenauer
6	I	<b>GitLab/Trello Zugriff</b> Einleitung in GitLab und Trello		Projektteam, Hr. Lehenauer
7	I	<b>Kosten</b> Erste grobe Evaluierung der Betriebs- & Entwicklungskosten.		Projektteam, Hr. Lehenauer
8	B	Nächstes Treffen am: 12 November.2021 / 13.45		Projektteam, Hr. Lehenauer

**Art:** I ...Information  
 A ...Auftrag  
 B ...Beschluss

### 11.3 Besprechungsprotokoll 05.11.2021

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 05.11.2021 **Dauer:** 13:45 – 15:00 Uhr  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Thomas Ploder (Projektteam)  
 Prof. Hofmann (Betreuer) Noah Laireiter (Projektteam)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Einwände zur letzten Besprechung. Übersicht der Situation.		Alle
2	I	<b>Lastenheft Review</b> Keine weiteren Einwände		Alle
3	I	<b>Vorführung bisheriger Arbeiten</b> Vorführung Benutzersystem (Normaler Nutzer) Vorführung Control Panel (Nutzerverwaltung, Systemprotokoll, Rangverwaltung, 16 Einstellungsmöglichkeiten)  Export der Nutzerliste nicht erforderlich (nur Webview)  Fortschritt (Planung) Hardware		Alle
4	I	<b>Neue Homepage (entwickelt von ids.co.at)</b> Weitere Vorgehensweise <ul style="list-style-type: none"> <li>- Wie verknüpfen</li> <li>- Gibt es Schnittstellen?</li> <li>- Kompatibilität</li> <li>- Auf einen Server zusammenlegen?</li> </ul>		Auftraggeber Projektteam
5	A	<b>Spezielle Randbedingungen</b> die bei Buchung eintreten können, niederschreiben <ul style="list-style-type: none"> <li>- Dürfen Nutzer Buchungen stornieren (Grund, ...)</li> <li>- Was, wenn man zu früh zurückkommt (Abrechnung?)</li> <li>- ...</li> </ul>		Auftraggeber Projektteam
6	B	Nächstes Treffen am: 19.11.2021 / 13.45 mit Ing. Gerhard Huber Verknüpfung mit „statischer“ Homepage		Alle Ing. G. Huber

Art:    I    ...Information  
       A    ...Auftrag  
       B    ...Beschluss

## 11.4 Besprechungsprotokoll 19.11.2021

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 19.11.2021 **Dauer:** 13:45 – 15:45 Uhr  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Thomas Ploder (Projektteam)  
 Ing. Gerhard Huber (Begleitung) Noah Laireiter (Projektteam)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Einwände zur letzten Besprechung. Übersicht der Situation.		Alle
2	B	<b>Homepage (mit ids.co.at)</b> Integration der One-Page Seite von Ing. Huber in das bestehende System besprochen.		Alle
3	A	<b>Rechnung ausstellen</b> Bis 12. Dezember müssen Rechnungen eingereicht werden, damit der Verein eine Förderung für das Projekt erhält.	10.12.	Alle
4	B	<b>Buchungssystem</b> Fragen dazu was der Nutzer alles darf, welche Informationen er sehen kann.  Editieren einer Buchung: Was, wenn Nutzer Buchung verlängern möchte. Vorschlag (Ing. Huber): einfach eine zweite Buchung im Anschluss machen. Beschlossen, dass die Buchung nach Stornierung neu erstellt wird. Beim nächsten Treffen soll dies genauer besprochen werden.  Ladezeit für E-Auto: Wie soll die Zeit vor und nach einer Buchung kalkuliert werden, damit sich das Auto genug aufladen kann. (Möglichkeit: ungefähre Fahrtstrecke zu Beginn der Buchung angeben)		Alle
5	B	<b>Keybox</b> Es gibt keinen Netzstromanschluss für die Keybox. Beschluss: Herr Lehenauer wird sich nach einer Alternative umsehen.		Alle
6	B	<b>Online Meeting</b> Da ab 22.11. in ganz Österreich ein Lockdown gilt, werden Treffen in diesem Zeitraum mit Möglichkeit online abgehalten.		Alle
7	B	Nächstes Treffen am: 10.12.2021 / Uhrzeit noch nicht fixiert (je nach Lage auch online)		Alle

Art: I ...Information  
 A ...Auftrag  
 B ...Beschluss

## 11.5 Besprechungsprotokoll 10.12.2021

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 10.12.2021 **Dauer:** 12:40 – 13:40 Uhr  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Thomas Ploder (Projektteam)  
 Ing. Gerhard Huber (Begleitung) Noah Laireiter (Projektteam)  
 Prof. Aichriedler (Betreuer)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Einwände zur letzten Besprechung. Übersicht der Situation. Hintergründe zur Projektbetreuung.		Alle
2	B	<b>Schlüsselbox</b> 1) Stromversorgung Energieverbrauch abschätzen, damit die Betriebskosten ermittelt werden können.  2) Wasserdicht Die Schlüsselbox kann im Projektrahmen nicht wasserdicht entwickelt werden. Möglichkeit: Professor Strasser nimmt Verbindung mit einem Unternehmen auf, welches möglicherweise eine wasserdichte Box herstellen kann. (genauere Informationen folgen noch)		Alle
3	B	<b>Rechnung</b> Die Rechnung für die im Projekt anfallenden Aufwand- und Barauslagen ist an Hr. Lehenauer übergeben worden.		Hr. Lehenauer, Projektteam
4	I	<b>Buchungssystem</b> Vorschläge: Verfügbare Buchungszeit über Pulldown-Menü auswählen (auch am Smartphone einfacher) – Kalenderansicht bei der Buchung entfernen, soll nur mehr beim Admin sichtbar sein Den Tag der Buchung über einen Kalender auswählen  Vereinbarung: Es soll ein gemeinsamer Mailserver auf dem Server der „easynome GmbH“ genutzt werden.		Alle
5	B	<b>Tarifgestaltung</b> Es soll verschiedene Tarife geben. Um Diskussionen zu meiden, sollen Nutzer keine anderen Tarife sehen können.  Einigung: Bei Registrierung -> Standardtarif ausgewählt. Admin kann dann individuell anderen Tarif einstellen.		Alle
6	B	Nächstes Treffen am: 21.01.2022 / 13.45		Alle

Art: I ...Information  
 A ...Auftrag  
 B ...Beschluss

## 11.6 Besprechungsprotokoll 21.01.2022

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 21.01.2022 **Dauer:** 13:40 – 14:50 Uhr  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Thomas Ploder (Projektteam)  
 Ing. Gerhard Huber (Begleitung) Noah Laireiter (Projektteam)  
 Prof. Aichriedler (Betreuer)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<p><b>Einleitung</b>          Einwände zur letzten Besprechung. Übersicht der Situation.          Hintergründe zur Projektbetreuung.</p>		Alle
2	I	<p><b>Buchungssystem &amp; Tarifsystem</b>          Fortschritt der Entwicklung des Buchungssystems.</p> <p><u>Einbringungen:</u></p> <ul style="list-style-type: none"> <li>• Monatlicher Mitgliedsbeitrag sollte auf 0 gesetzt werden können.</li> <li>• Möglichkeit für Nutzer über eine Online-Plattform kommunizieren zu können, um z.B. Fahrtgemeinschaften organisieren zu können.</li> <li>• Frage, ob bei den Kosten für den Nutzer aufgelistet werden soll, wofür diese genau aufgekommen sind.</li> </ul> <p><u>Beschluss:</u>          Aus zeitlichen Gründen wird es nicht möglich sein, alles realisieren zu können.          Bei den nächsten Treffen wird noch genauer besprochen, wie dieses Projekt nach Abgabe weitergeführt werden könnte.</p>		Alle
3	B	<p><b>Testbetrieb</b>          Bis zum nächsten Treffen sollte ein erster Live-Test betriebsbereit sein.          Dafür werden die Zugangsdaten für den Server noch via E-Mail übermittelt.</p>	21.02.	Alle
4	I	<p><b>Rechnung</b>          Keine weiteren Fragen zur Rechnung so weit.</p>		Hr. Lehenauer, Projektteam
5	I	<p><b>Keybox</b>          Professor Strasser hat hierzu auf die Firma Masunt (Resatur GmbH) verwiesen, welche fertige Schlüsselboxen anbietet.          Diese würden allerdings Extrakosten pro Öffnung mit sich bringen, weshalb diese Option nicht weiter in Betracht gezogen wird.</p> <p>Es gebe eventuell die Möglichkeit die MYFLEXBOX auch mit dynamischen Codes zu nutzen.</p>		Alle
6	B	<p>Nächstes Treffen am:          21.02.2022 / 08.00</p>		Alle

Art:    I    ...Information  
           A    ...Auftrag  
           B    ...Beschluss

## 11.7 Besprechungsprotokoll 21.02.2022

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 21.02.2022 **Dauer:** 08:00 – 09:30 Uhr  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Thomas Ploder (Projektteam)  
 Ing. Gerhard Huber (Begleitung) Noah Laireiter (Projektteam)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Einwände zur letzten Besprechung. Übersicht der Situation.		Alle
2	I	<b>Testbetrieb</b> Fortschritt der Entwicklung des Buchungssystems.  Der Live-Test-Betrieb kann mit dem heutigen Datum beginnen.  Bis zum nächsten Treffen sollen auftretende Bugs zusammengeschrieben werden. Weiters sollen die Texte im System untersucht werden, damit etwaige Verbesserungen gemacht werden können.	18.03.	Alle
3	B	<b>Einbringungen zur Software</b> <ul style="list-style-type: none"> <li>• Können Notizen bei den Autos hinzugefügt werden, wo Nutzer Schäden am Auto notieren können.</li> <li>• Angabe der IBAN bei Registrierung neuer Nutzen zur Abrechnung</li> <li>• Gewisse Autos nur für gewisse Adressen zur Verfügung stellen</li> <li>• Sollte Admin stornierte Buchungen reaktivieren können (könnte Konflikte mit Nutzern erzeugen)</li> </ul> <u>Beschluss:</u> Aus zeitlichen Gründen wird es nicht möglich sein, alles realisieren zu können. Die IBAN wird nicht vom System automatisch geprüft (muss händisch vom Admin vor der Freischaltung erfolgen) Einschränkungen für verschiedene Adresse wird nicht implementiert Notizen der User können in der ersten Version noch nicht verfasst werden.		Alle
5	I	<b>Keybox</b> Ein Techniker bei der MYFLEXBOX meint, dass es möglich sei den Code über das Internet zu editieren.		Alle
6	B	Nächstes Treffen am: 18.03.2022 / 13.30		Alle

Art:    I    ...Information  
        A    ...Auftrag  
        B    ...Beschluss

## 11.8 Besprechungsprotokoll 18.03.2022

**Titel:** Projektbesprechung „LPCS – Carsharing System“ in der HTBLuVA Salzburg  
**Datum:** 18.03.2022 **Dauer:** 13.30 – 14.10  
**Schriftführer:** Noah Laireiter

**Teilnehmer:** Hr. Lehenauer (Auftraggeber) Noah Laireiter (Projektteam)  
 Ing. Gerhard Huber (Begleitung) Prof. Hofmann (Betreuer)

Pkt	Art	Beschreibung	Bis wann	Wer
1	I	<b>Einleitung</b> Einwände zur letzten Besprechung. Übersicht der Situation. Hintergründe zur Projektbetreuung		Alle
2	I	<b>Einwände</b> Eine stornierte Buchung, die reaktiviert wird, soll im Systemprotokoll auftauchen (Wer hat wann was reaktiviert?)  Schriftgröße der Fahrzeugnotizen, die nach abgeschlossener Buchung angezeigt werden, soll größer werden, da wichtige Nachrichten darin enthalten sein könnten.  Im Login Fenster soll ein Link hinzugefügt werden, der auf die Registrieren-Website weiterleitet.		Alle
3	I	<b>Projektstatus</b> Schriftliche Diplomarbeiten werden verfasst und am 04.04. abgegeben. Danach folgt die Matura – zieht sich bis zum 15. Juni (Tag der Projektverteidigung)		Alle
5	I	<b>Testbetrieb</b> Beim Testbetrieb wird noch gewartet bis zumindest das Tracking-Modul für das Fahrzeug funktioniert. Es wird nach der Maturareise Ende Juni Kontakt aufgenommen, um ein neues Treffen zu vereinbaren, an dem die weiteren Schritte besprochen werden. Bis Ende Juni könnten aber vertraute Betatester die Software einem ersten Test unterziehen.		Alle
6	B	Nächstes Treffen am: Wird im Juni 2022 beschlossen	Juni 2022	Alle

Art: I ...Information  
 A ...Auftrag  
 B ...Beschluss