
Meilenstein 2

Dokumentation

Entwicklung des IS basierend auf relationalem DBMS



Team 5

Nikola Babic, 1501205

Cordula Eggerth, 0750881

Denise Gall, 2996142

Gregor Langner, 1502605

BSc Wirtschaftsinformatik

VU Information Management & Systems Engineering

SoSe 2017

Inhaltsverzeichnis

1. Wahl des Realitätsausschnitts für die Entwicklung des Informationssystems	3
2. Infrastruktur	5
3. Relationales Datenbankdesign	5
4. Datenimport	7
5. Erstellung des Websystems (basierend auf relationalem DBMS)	9
6. Anhang: Klassendiagramm	11
7. Anhang: Login-Hilfe	12

1. Wahl des Realitätsausschnitts für die Entwicklung des Informationssystems

Für die Entwicklung des Informationssystems wurde das Projekt *CarGoDS* gewählt, das ein *Informationssystem (IS) für die Fahrschule CarGo* basierend auf einem relationalen Datenbankmanagementsystem (DBMS) vorschlägt¹. Das IS erfasst Informationen, die für den Betrieb der Fahrschule notwendig sind, sowohl aus Kundensicht als auch aus Mitarbeitersicht. Die *Kunden* (*Test-Account-Daten siehe Fußzeile*)² können einen Kunden-Account erstellen, den Kurskatalog anschauen und über diesen einen Kurs buchen. Jeder Kunde hat seine eigene Profil-Seite, auf der er seine Kundendaten und gebuchten Kurse einsehen kann. Bei Bedarf kann auch das Passwort geändert werden und eine nicht mehr gewünschte Kursteilnahme storniert werden. Die untenstehende Abbildung zeigt die den eingeloggten Kunden zur Verfügung stehenden Funktionalitätsbereiche. Der „Kurskatalog“ enthält innerhalb den Bereich zur Kursbuchung und Einzelansicht von Kurse. Der Bereich „Mein Profil“ enthält die Funktionalität zur Änderung des Passworts, zur Einsicht in die eigenen Kundendaten und gebuchten Kurse sowie die Möglichkeit der Stornierung einer Kursteilnahme.



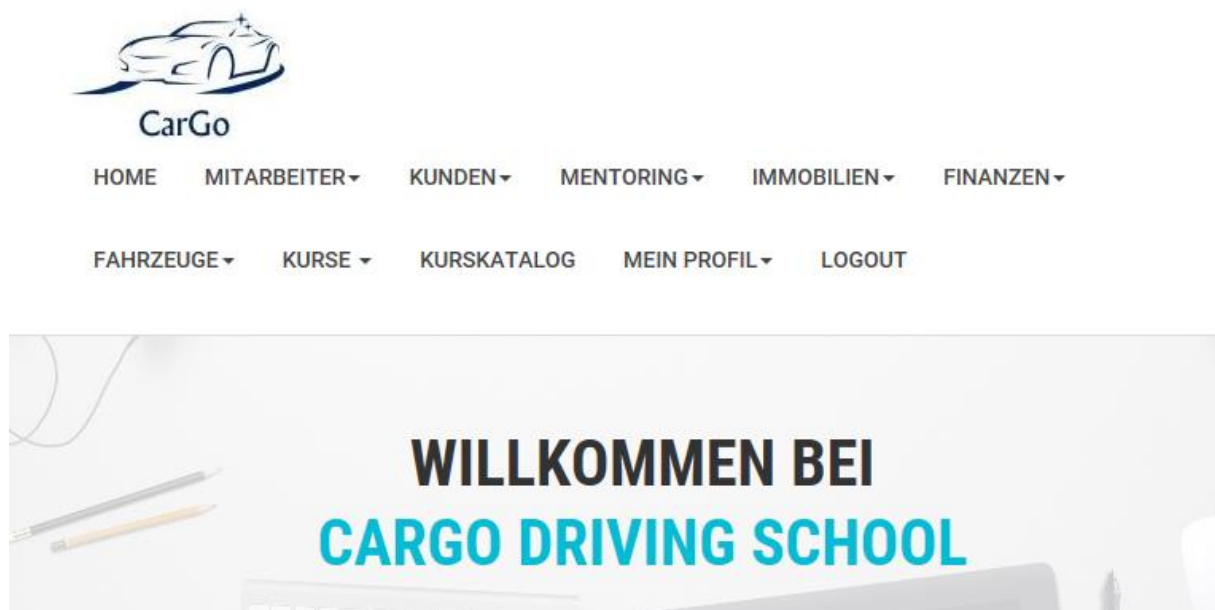
*Mitarbeiter*³ können nur von einem bestehenden Mitarbeiter, der eingeloggt ist, angelegt werden, und haben ebenfalls eine eigene Profil-Seite, auf der sie ihre Mitarbeiterinformationen einsehen und ändern können. Der Mitarbeiterbereich umfasst

¹ Für Details zum Projektvorschlag und den verschiedenen Diagrammtypen siehe Meilenstein 1 und 2 von Cordula Eggerth (0750881).

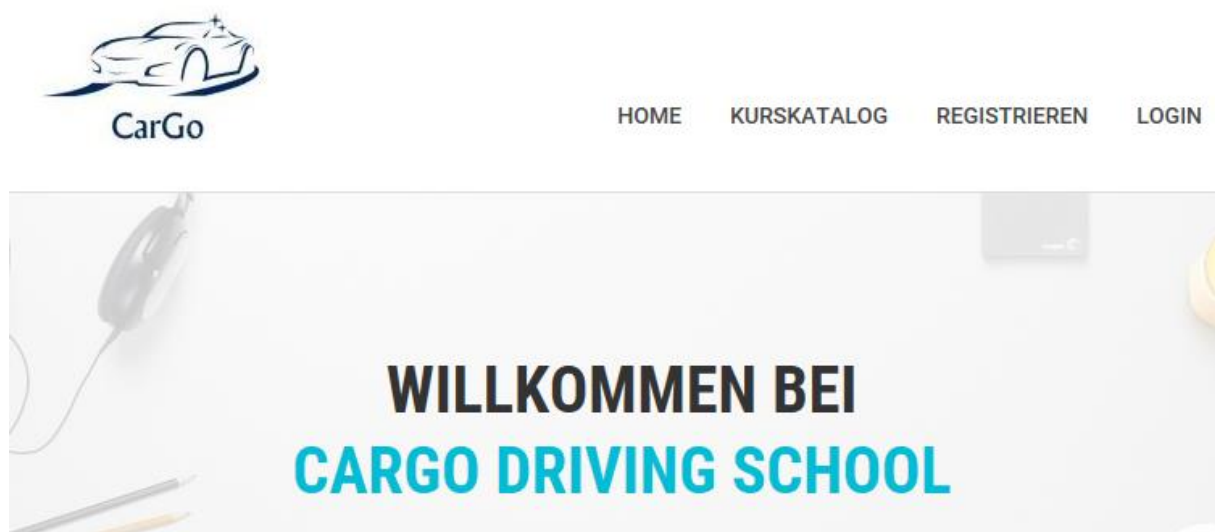
² Vorgeschlagene Testdaten für einen existierenden Kunden-Account: username: cargokunde, pw: cargokunde

³ Vorgeschlagene Testdaten für einen existierenden Kunden-Account: username: gregor, pw: db

weitere Immobilien, Fahrzeuge, Kurse, Mentoring und Mitarbeitermanagement, die jeweils bei Klick auf den Pfeil verschiedene Funktionalitäten bieten (siehe untenstehende Abbildung).



Nicht auf der Website registrierte/eingeloggte Benutzer können den Kurskatalog ansehen, Kursdetails einsehen, sich registrieren/einloggen und allgemeine Informationen über die Fahrschule wie z.B. den Lageplan einholen (siehe untenstehende Abbildung).



Für die Anzeige der Website im Browser wurde Firefox benützt und das Layout wird dort bestmöglich angezeigt.

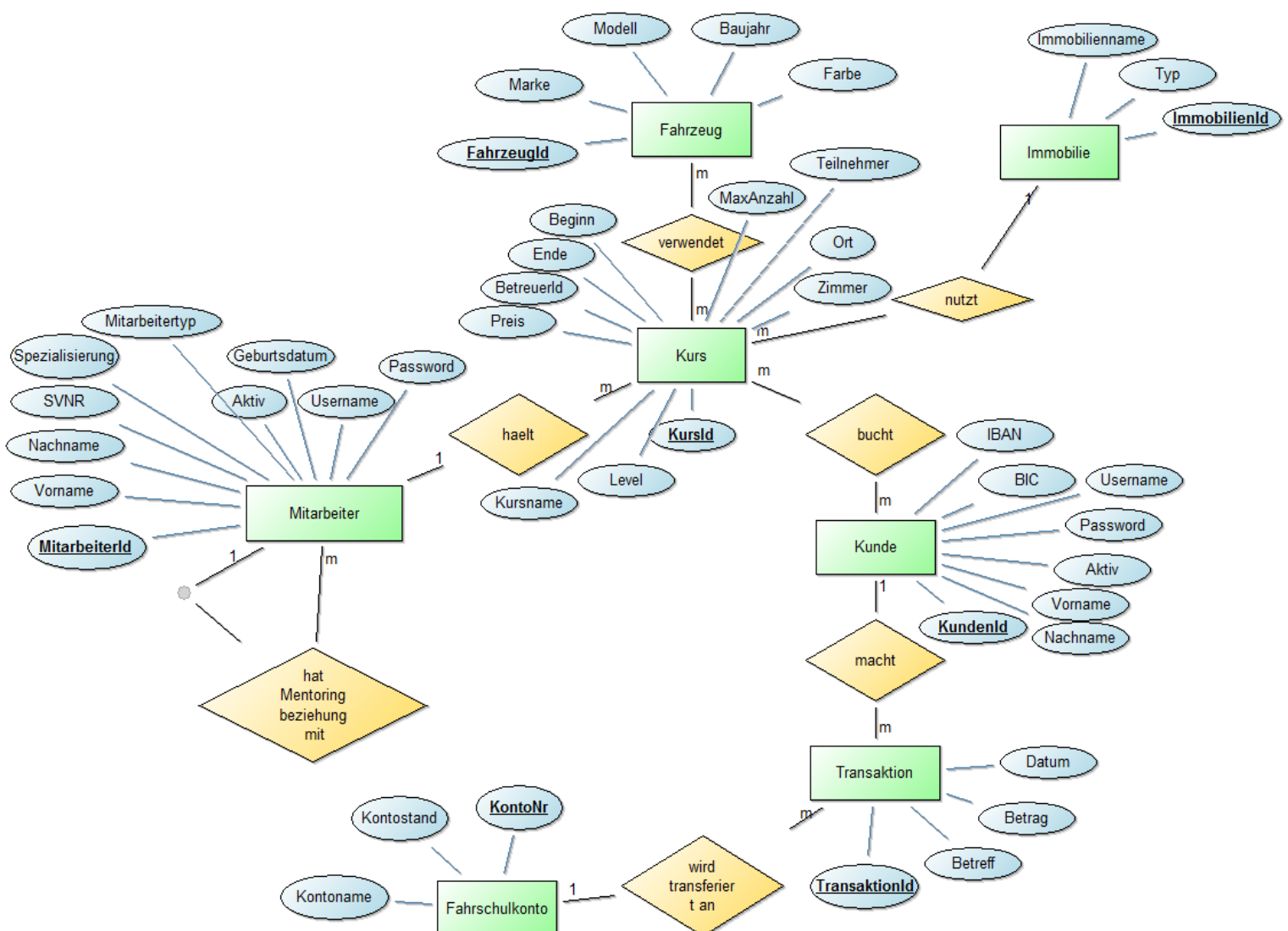
2. Infrastruktur

Das Betriebssystem, auf dem das Projekt entwickelt wurde, ist Windows. Als relationales DBS wurde MariaDB/SQL verwendet. Die Programmierung wurde auf lokalen MariaDB-Datenbanken und der von der Universität Wien zur Verfügung gestellten MariaDB-Datenbank getestet – die Angaben zur Datenbank befinden sich bereits in den Klassen DatabaseHandler.java bzw. in den jeweiligen DatabaseDAO-Klassen im Package repository. Als Programmiersprache wurde Java mit JDBC für die SQL-Statements gewählt. Als Webserver wird Apache Tomcat 8.0 verwendet.

Das Starten des Programms erfolgt über Ausführen des index.jsp (im „Webcontent“) am Tomcat-Server, und die weiteren Bereiche können über die Navigationsleiste erreicht werden.

3. Relationales Datenbankdesign

Die Datenbank (MariaDB) wurde basierend auf dem untenstehenden ER-Diagramm angelegt:



Die jeweiligen Tabellen wurden in der MariaDB (sowohl lokal zum Testen als auch auf der von der Universität zur Verfügung gestellten Datenbank) angelegt:

Host: 127.0.0.1	Datenbank: cargo	Tabelle: kunde	Daten	Abfrage
Name	Zeilen	Größe	Erstellt	Aktualisiert
fahrschulkonto	0	16,0 KiB	2017-05-10 00:42:41	
fahrzeug	0	16,0 KiB	2017-05-10 00:48:25	
immobilie	4	16,0 KiB	2017-05-10 00:51:59	
kunde	7	16,0 KiB	2017-05-10 00:59:54	
kurs	5	16,0 KiB	2017-05-09 21:36:28	
mentoring	0	16,0 KiB	2017-05-10 01:01:16	
mitarbeiter	2	16,0 KiB	2017-05-10 01:03:08	
teilnehmer	5	16,0 KiB	2017-05-14 20:01:05	

Die folgenden zwei Abbildungen zeigen anhand des Beispiels der Tabelle „kunde“, wie die jeweiligen Entities in der MariaDB angelegt wurden. In der ersten Abbildung wird das Create-Table-Statement gezeigt, mit dem die Tabelle angelegt wurde. Im zweiten Screenshot werden Daten angezeigt, mit denen die Tabelle „kunde“ bereits befüllt wurde.

Host: 127.0.0.1	Datenbank: cargo	Tabelle: kunde	Daten	Abfrage	
Allgemein	Optionen	Indizes	Fremdschlüssel	Partitionen	CREATE code

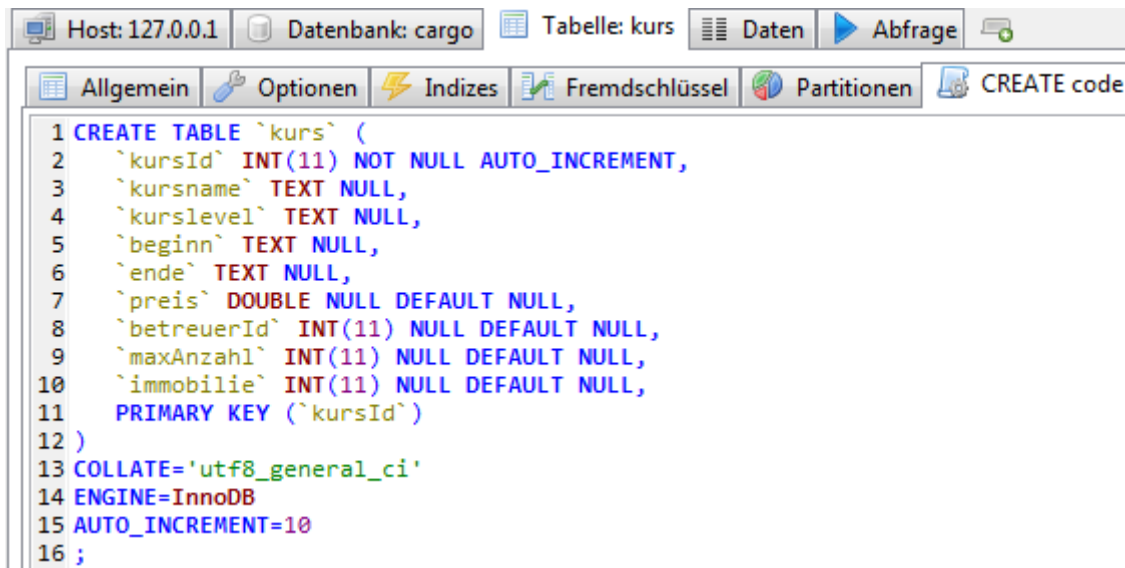
```
1 CREATE TABLE `kunde` (  
2   `kundenId` INT(11) NOT NULL AUTO_INCREMENT,  
3   `vorname` TEXT NULL,  
4   `nachname` TEXT NULL,  
5   `iban` INT(11) NULL DEFAULT NULL,  
6   `bic` TEXT NULL,  
7   `username` TEXT NULL,  
8   `passw` TEXT NULL,  
9   `active` TINYINT(1) NULL DEFAULT NULL,  
10  PRIMARY KEY (`kundenId`)  
11 )  
12 COLLATE='utf8_general_ci'  
13 ENGINE=InnoDB  
14 AUTO_INCREMENT=11  
15 ;
```

Host: 127.0.0.1 | Datenbank: cargo | Tabelle: kunde | Daten | Abfrage

cargo.kunde: 8 Zeilen gesamt (ungefähr) | Nächste Zeile | Alle Zeilen | Sortierung (1) | Spalten (8/8) | Filter

kundenId	vorname	nachname	iban	bic	username	passw	active
7	Tara	Tisch	12.141	meinBIC	tarat	taraUpdated	1
2	VornameUpdate	NachnameUpdate	11.111	BICupdate	usernameUpdate	pwUpdate	1
8	Maia	Meier	131	nkhn14	maiam	maia	1
9	maia	meier	12.323	kmlöwmk2	maiam	maia	1
1	Martin	Maier	12.341.234	RLNWKELA	martinm	martin	1
4	Lisa	Maier	111.111	RLNWKELA	lisam	lisa	1
5	Kathi	Katze	101.010	meinBIC	kathik	kathi	1
10	Katrin	Kater	1.235	BANKCODE2335	katrink	katrin	1

Die nächste Abbildung zeigt das Beispiel des Anlegens der Tabelle „kurs“, in der die Fahrkurse für die Fahrschule gespeichert werden. Die zweite Abbildung zeigt Daten der bereits erstellten und im Informationssystem gespeicherten Daten.



```

1 CREATE TABLE `kurs` (
2   `kursId` INT(11) NOT NULL AUTO_INCREMENT,
3   `kursname` TEXT NULL,
4   `kurslevel` TEXT NULL,
5   `beginn` TEXT NULL,
6   `ende` TEXT NULL,
7   `preis` DOUBLE NULL DEFAULT NULL,
8   `betreuerId` INT(11) NULL DEFAULT NULL,
9   `maxAnzahl` INT(11) NULL DEFAULT NULL,
10  `immobilie` INT(11) NULL DEFAULT NULL,
11  PRIMARY KEY (`kursId`)
12 )
13 COLLATE='utf8_general_ci'
14 ENGINE=InnoDB
15 AUTO_INCREMENT=10
16 ;

```



kursId	kursname	kurslevel	beginn	ende	preis	betreuerId	maxAnzahl	immobilie
6	MeinKurs1	Anfaenger	23.05.2017	25.05.2017	1	3	1	1
8	seinKurs1	Anfaenger	10.05.2017	11.05.2017	1	3	1	2
9	ABC Fahrkurs PKW	Anfaenger	27.05.2017	28.05.2017	93,5	1	22	1
5	Kursname3	Level3	02.06.2017	20.06.2017	111	1	11	1
2	NameUpdate	LevelUpdate	12.07.2017	12.07.2017	22	1	5	1
1	TestkursnameNeuNochmals	Fortgeschritten	14.05.2017	14.05.2017	33	1	20	5

4. Datenimport

Das Füllen der Tabellen erfolgt(e) mittels der Programmiersprache Java unter Verwendung von JDBC (Java Database Connectivity). Die Eingaben werden über die jeweiligen Formulare (in den JSPs⁴) auf der Website entgegengenommen, von den Servlets im Package controller verarbeitet, und über die Mgmt-Klassen im Package logic an die DatabaseDAO-Klassen im Package repository weitergereicht, in denen sich schließlich die SQL-Statements für die Datenbank-Operationen befinden.

Die untenstehende Abbildung zeigt anhand des Beispiels des Anlegens eines neuen Kunden die Methode addKunde, die das Interface KundenDAO im Package repository implementiert, und sich in der Klasse DatabaseKundenDAO befindet. In derartigen Methoden der DAO-Klassen werden unter Verwendung von JDBC die SQL-Statements in der Datenbank ausgeführt und die jeweiligen Anweisungen durchgeführt.

⁴ In den JSP-Files werden die Sprachen Java, HTML und JavaScript verwendet.

```
@Override
public void addKunde(Kunde kunde) {
    Connection con = null;
    try {
        Class.forName("org.mariadb.jdbc.Driver");
        con = DriverManager.getConnection(DBAdresse, username, password);

    } catch (ClassNotFoundException e) {
        e.printStackTrace();
    } catch (SQLException e) {
        e.printStackTrace();
    }

    try {

        // insert statement
        String query = " insert into Kunde (vorname, nachname, iban, bic, username, passw, active)"
            + " values (?, ?, ?, ?, ?, ?, ?)";
        // create insert preparedstatement
        PreparedStatement preparedStmt = con.prepareStatement(query);
        preparedStmt.setString (1, kunde.getVorname());
        preparedStmt.setString (2, kunde.getNachname());
        preparedStmt.setInt (3, kunde.getIban());
        preparedStmt.setString(4, kunde.getBic());
        preparedStmt.setString(5, kunde.getUsername());
        preparedStmt.setString(6, kunde.getPassword());
        preparedStmt.setBoolean(7, kunde.isActive());

        // execute the preparedstatement
        preparedStmt.execute();
        con.close();
    } catch (SQLException e) {
        e.printStackTrace();
    }finally {
        try {
            /*
                if (preparedStmt != null)
                    preparedStmt.close();
            */
            if (con != null)
                con.close();
        } catch (SQLException e) {
            e.printStackTrace();
        }
    }
}
```


5. Erstellung des Websystems (basierend auf relationalem DBMS)

Das untenstehende Use-Case-Diagramm zeigt die realisierten Funktionalitäten des Informationssystems für die Fahrschule CarGo.⁵ Die *Aufteilung der Use-Cases* unter den Team-Mitgliedern wurde mittels *Farbcodierung* gekennzeichnet, und in einer zusätzlichen Notiz im Diagramm wurden gemeinsame Bearbeitungen eines Use-Case sowie die Realisierung von Login und Logout vermerkt. Für *nähere Informationen* zu den *Aufgabenbereichen der einzelnen Teammitglieder* und wer, wann, woran gearbeitet hat, siehe *M2_Team5_TeamArbeitsprotokoll.pdf* in der Abgabe.

Die Datenbank ist nur mittels SSH-Verbindung zu erreichen und der Webserver wird mit selbst-signiertem Zertifikat abgesichert (mit https).

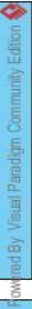
Das Projekt wurde auf dem Tomcat-Server deployed:

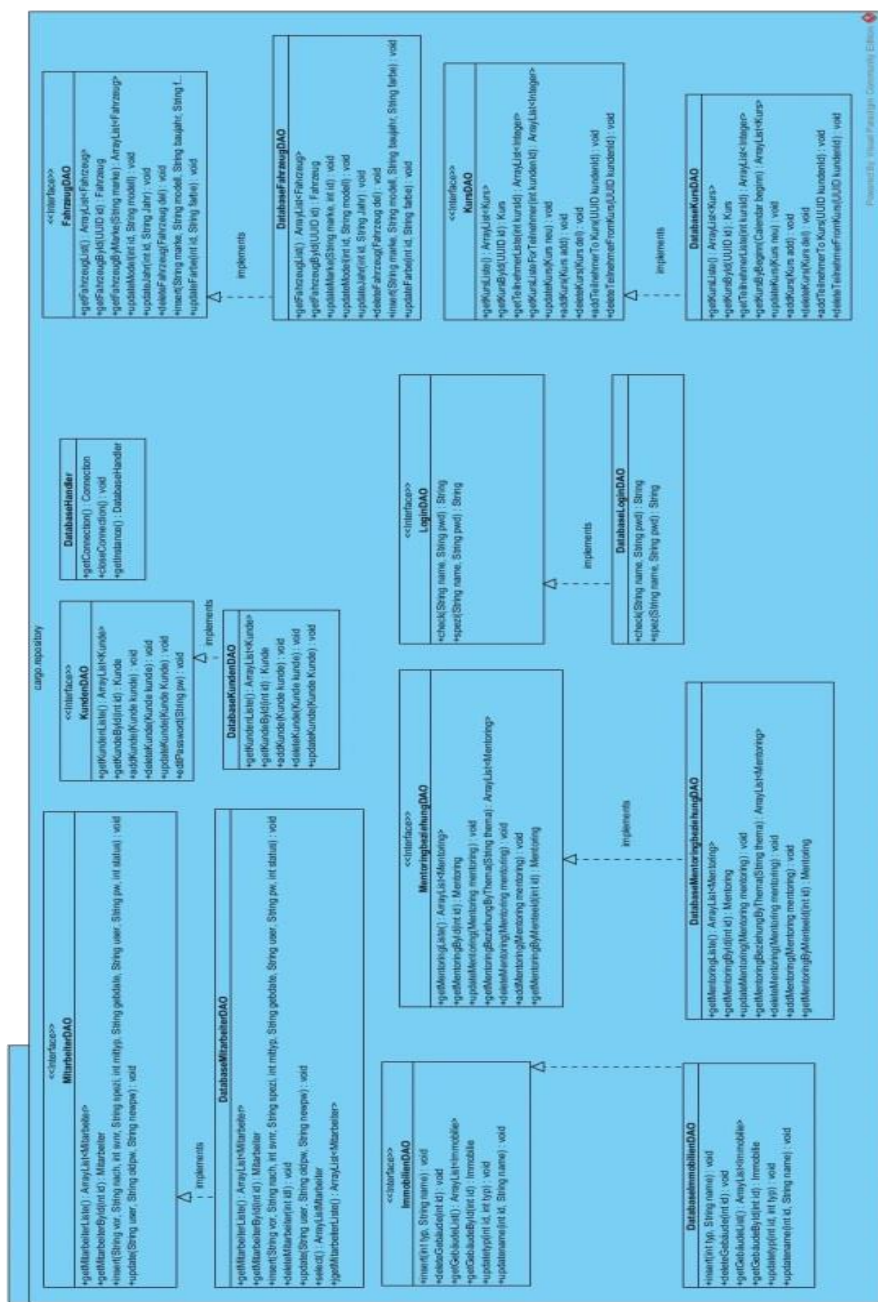
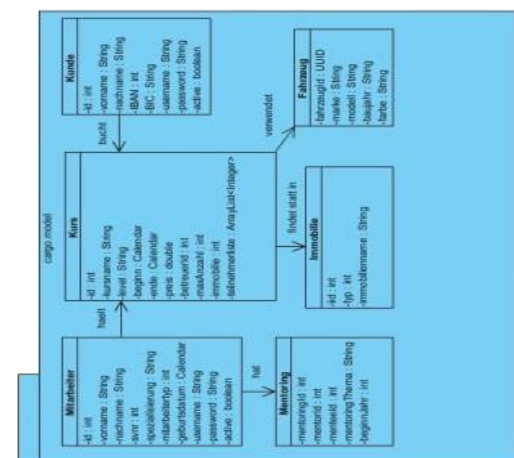
https://tomcat01lab.cs.univie.ac.at:31354/ise_Projekt/

Zur Farbcodierung des Use-Case-Diagramms auf der nächsten Seite:

- Nikola Babic: grün
- Cordula Eggerth: gelb
- Denise Gall: rot
- Gregor Langner: blau

⁵ Die Funktionalitäten des Login und Logout wurden nicht im Diagramm inkludiert, da sie lt. SWE VO (Prof. Benkner) nicht als „Use-Cases im engeren Sinn“ zu zählen sind.






7. Anhang: Login-Hilfe

Login als Kunde: (implementiert von Cordula Eggerth)

Kunden-Account-Daten: Username: cargokunde; Password: cargokunde



HOME KURSKATALOG REGISTRIEREN LOGIN

LOGIN ALS KUNDE:

USERNAME:

PASSWORD:

Login

Hauptseite, wenn als Kunde eingeloggt:



HOME KURSKATALOG MEIN PROFIL ▾ LOGOUT

WILLKOMMEN BEI CARGO DRIVING SCHOOL

SIE SIND EINGELOGGT ALS: CARGOKUNDE

WIR FREUEN UNS UEBER IHREN BESUCH BEI DER CARGO FAHRSCHULE IN WIEN!
BEI UNS KOENNEN SIE SICH SCHNELL UND EINFACH UEBER DIE FAHRKURSE
IHRER WAHL INFORMIEREN, KURSE SUCHEN UND BUCHEN.
NATUERLICH KOENNEN SIE UNS AUCH VOR ORT BESUCHEN, WO WIR SIE PERSOENLICH
BERATEN.

KURSKATALOG

Login als Mitarbeiter: (implementiert von Gregor Langner)

Eingabe der Test-Mitarbeiterdaten:

Username: gregor

Passwort: db

Mittels dieser Login-Daten kommt man zur Hauptseite für die Mitarbeiter. Die Navigation erfolgt über die Navigationsleiste oben bzw. über die Buttons in den jeweiligen Seiten.

