

# Conference Paper Title\*

\*Note: Sub-titles are not captured in Xplore and should not be used

1<sup>st</sup> Justin Frommberger  
*Interaktionstechnik und Design*  
*Hochschule Hamm-Lippstadt*  
City, Country  
email address or ORCID

2<sup>nd</sup> Jonas Gerken  
*Interaktionstechnik und Design*  
*Hochschule Hamm-Lippstadt*  
City, Country  
email address or ORCID

3<sup>rd</sup> Benedikt Lipinski  
*Interaktionstechnik und Design*  
*Hochschule Hamm-Lippstadt*  
Soest, Deutschland  
benedikt.lipinski@stud.hshl.de

4<sup>th</sup> Phillip Wagner  
*Interaktionstechnik und Design*  
*Hochschule Hamm-Lippstadt*  
City, Country  
email address or ORCID

**Abstract**—This document is a model and instructions for  $\LaTeX$ . This and the IEEEtran.cls file define the components of your paper [title, text, heads, etc.]. \*CRITICAL: Do Not Use Symbols, Special Characters, Footnotes, or Math in Paper Title or Abstract.

**Index Terms**—component, formatting, style, styling, insert

## I. PARTS OF INTEGRATION

### II. SERVER

In diesem Kontext spielt der Server eine sehr wichtige rolle in der Kommunikation zwischen den ausführenden parts des Projektes. Durch den Server und seine strukturen wird letztendlich erst eine plattform geschaffen, die Allen fahrzeuge und den Kunden (Usern) eine Möglichkeit bietet eine verbindung unter einander zu schaffen und weitere Aufgaben zu erledigen. Konkret waren die aufgaben des Servers:

Anmeldung von clients  
Das nächstgelegene Fahrzeug finden  
Jedem client eine eindeutige ID zuordnen  
Übermittelt fahrzeugdaten an den Kunden  
Interne verarbeitung einer fahrzeug bestellung

#### A. Anmedlung von Clients

Aufgabe des Servers war es eine Anmeldung von clients zu ermöglichen um einerseits nur aus dem pool der aktuell aktiven, freie fahrzeuge auszuwählen und andererseits einer unbekannten menge an Fahrzeugen und kunden die Möglichkeit zu bieten am anbot teilzuhaben. Zu den Clients gehören sowohl die Kunden (User), wie auch alle Fahrzeug typen , damit sind alle Servicefahrzeuge mit den unterkategorien: Police, Firefighter, ambulance gemeint und zuletzt auch Fahrzeuge der Kategorie Taxi.

Umgesetzt wurde die anmeldung der Clients in den Funktionen *registrierunUser*

---

```
#registration for user
def registrationUser(data):
    inliste = False
    for i in range(0, len(user)):
#alredy exists Requirement: F-S04
        if str(data[0]) == str(user[i][0]):
            inliste = True
    if inliste == False:
# no ? add! Requirement: F-S05
        user.append(data)
```

Identify applicable funding agency here. If none, delete this.

```

elif inliste == True:
#yes ? print(user exists alredy !)
    print("#user bereits vorhanden")

```

---

und für die Fahrzeuge wurde eine Registrierung in der Funktion *registrationCar* umgesetzt.

---

```

def registrationCar(data ,type):
    car = []
    inliste = False
    if type == 0:
        car = taxi
    elif type == 1:
        car = police
    elif type == 2:
        car = firefighter
    elif type == 3:
        car = ambulance
    elif type == 99:
        car = testcar
    for i in range(0 ,len(car)):# Requirement: F-S07
        if str(data[1]) == str(car[i][1]):
            inliste = True
    if inliste == False:
        if type == 0:
            data.append("free")
            taxi.append(data)
        elif type == 1:
            data.append("free")
            police.append(data)
        elif type == 2:
            data.append("free")
            firefighter.append(data)
        elif type == 3:
            data.append("free")
            ambulance.append(data)
        elif type == 99:
            data.append("free")
            testcar.append(data)
    elif inliste == True: #Requirements: F-S12
        print("#Car already registered")
#####

```

---

## REFERENCES

[1]