

Conference Paper Title*

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

1st Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

2nd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

3rd Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

4th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

5th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
City, Country
email address or ORCID

6th Given Name Surname
dept. name of organization (of Aff.)
name of organization (of Aff.)
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. EINLEITUNG

Im Laufe der Jahre findet MQTT vielseitigen Einsatz, für Sensordaten, klassische Nachrichten, Aktienkurse oder Kurzmitteilungen bei der Facebook Mobil App.

Dies weckt Interesse bei vielen Nutzern, sich mit dem Thema MQTT vertraut zu machen.

MQTT ist ein Client-Server-Protokoll.

Clients senden dem Server (Broker) nach Verbindungsaufbau Nachrichten mit einem Topic, welches die versendeten Nachrichten strukturiert.

Zum Beispiel Haus, Wohnzimmer, Sofa.

So hat sich unsere Gruppe die Aufgabe gestellt, das MQTT Projekt MQTTexercise2021 zu

Programmieren und so Kenntnisse zum Thema Mqtt zu sammeln.

In dem Projekt haben wir uns die Aufgabe gestellt, einen Server zu Programmieren der Nachrichten von anderen Clients (User, Services, Taxi) empfängt und Nachrichten versenden kann.

A. Aufgaben

Server: Registriert alle Clients, verteilt IDS, übermittelt Nachrichten, erhält Nachrichten, leitet/organisiert den Ablaufplan.

User: Anmelden beim Server, order Fahrzeug, sendet Koordinaten, Fahrzeug kommt, wird zum Zielort gebracht, setzt Car free

Taxi: Anmeldung Server, fährt zu den Koordinaten,

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

übermittelt Koordinaten, übermittelt Nachrichten, erhält Nachrichten.

Services: Anmelden beim Server, sendet Koordinaten, warten auf Nachricht, bekommt Koordinaten von User, zum User fahren, Fahrzeug ist bussy, User zum Ziel fahren, Fahrzeug wieder free

B. Ziel

Das Ziel des Projektes ist, ein Programm zu schreiben, welches ein MQTT Ablauf zeigt und eine funktionierende Kommunikation zwischen den Clients enthält.

C. Ablauf

Zunächst wird in Kapitel 2 die Projektaufgabe beschrieben. In Kapitel 3 werden die Vorbereitungen und Recherche dargestellt und in Kapitel 4 der Aufbau des Projektes. Die Arbeit endet mit einem Fazit.

II. DIAGRAMME

Bevor wir mit der Implementation des Projektes starten, führten wir ein Gruppen Brainstorming durch um festzulegen, welche Aufgabe jeder einzelne für das Projekt erfüllen muss.

A. Requirements

Das Ergebnis des Brainstormings, verfasste jeder in einer Requirements Tabelle zusammen, um später zu verhindern das Probleme zwischen den einzelnen Clients auftreten können.

So wurde festgelegt das jeder Client seine eigene ID und Namen besitzen muss.

Zudem sind alle Clients mit dem Server verbunden, dort sollen sie sich mit ihrem eigenen Namen anmelden.

Ein weiterer Ansatz ist, dass jeder Client, Nachrichten Senden und Empfangen soll.

So müssen die Standorte oder

Aufgaben zum Server gesendet und dort verarbeitet werden.

Die Nachrichten sollen innerhalb von 5 Sekunden zwischen den Clients ausgetauscht werden.

Abschließend wurde festgelegt, dass jeder sein Programm mit Python programmieren soll, um später alles reibungslos zusammenfügen zu können.

Taxi requirements

Typ	ID.	Description	
FA	01	Das Car muss eine eindeutige ID besitzen.	✓
FA	02	Das Car soll mit dem Server verbunden sein.	✓
FA	03	Das Car soll sich beim Server anmelden.	✓
FA	04	Das Car soll das Ziel vom User erhalten.	✓
FA	05	Das Car soll auf Anfrage vom Server seinen Standort mitteilen.	✓
FA	06	Das Car soll vom Server ausgewählt und dann auf busy gesetzt werden.	✓
NF	01	Das Car soll innerhalb von 5s dem Server eine Nachricht senden.	✓
NF	02	Das Car soll innerhalb von 5s die Nachricht vom Server erhalten.	✓
NF	03	Das Car soll innerhalb von 5s mitteilen ob es verfügbar ist und seinen aktuellen.	✓
NF	11	Die Software soll mit Python programmiert werden.	✓

Fig. 1. Requirments

B. Use-Case

Der Nächste Schritt im Projekt, ist aus dem gesammelten Informationen der Requirements, Use-Case Diagramme für jeden Client zu erstellen.

Use-Case Diagramme verdeutlichen die Aufgaben und Verbindungen zwischen den einzelnen Clients.

Vorab Registrieren sich alle Clients beim Server, so ist der Server mit jedem Client verbunden.

Nun bekommt der Server eine Message vom User, 'request car'. Folgend wird 'select closest car' vom Server durchgeführt und dem nächsten gelegenen Car mitgeteilt, dass eine anfragende vom User eingegangen ist.

Zudem sind in der Message die Koordinaten des Users enthalten, woraufhin das Car zu diesem Standort fährt.

Der Server setzt das bestellte Car auf Busy, 'change car status'.

Danach erhält das Car eine Message vom User mit dem gewünschten Zielort und bringt ihn dort hin.

Am Zielort setzt der User dann das Car beim Server wieder auf free.

Zum Schluss erfragt der Server den neuen Standort des Cars und schließt die Handlung damit ab.

III. TAXICODE

Die Aufgabe des Taxis ist es über den MQTT Server mit dem Server und dem User zu kommunizieren.

Über die erhaltenen Messages fährt das Taxi zum User und bringt ihn zu seinem Zielort.

Der erste Schritt vom Taxi ist, sich beim Server mit seinem Namen zu registrieren, folgend weiß der Server dann dem Taxi seine ID zu.

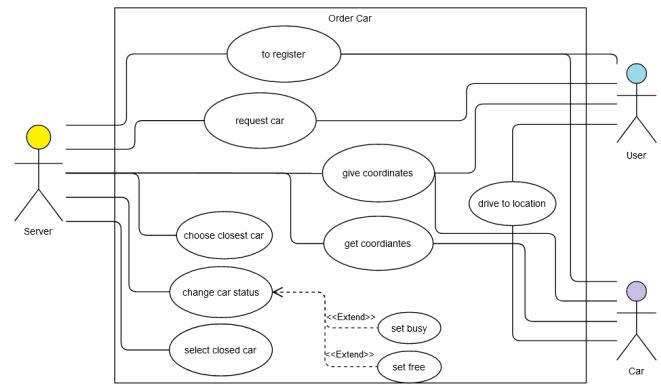


Fig. 2. Use-Case

```
data = {
    "id": "register",
    "name": name,
    "coordinates": coor
}
```

Daraufhin erhält das Taxi eine Message vom Server, dass ein User ein Taxi bestellt hat. Nun liest das Taxi die Koordinaten vom User aus.

```
data = {
    "id": id,
    "msg": "Arrival",
    "coordinates": js['coordinates']
}
```

Mit den erhaltenen Koordinaten, weiß das Taxi an welcher Stelle sich der User befindet und fährt zu seiner aktuellen Position. Am Zielort angekommen, wartet das Taxi auf die neue Message vom User, mit seinem neuen Zielort.

```
data={
    "id":id,
    "msg": "Arrival at destination",
    "coordinates": js['destination']
}
```

Zum Schluss erhält das Taxi eine Anfrage vom Server, mit seiner neuen Position und sendet diese dann zum Server.

```
data={
    "id":id,
    "name":name,
    "coordinates":coor
}
send(json.dumps(data),
    "hshl/mqtt_exercise/set_position")
```

For papers published in translation journals, please give the English citation first, followed by the original foreign-language citation [6].

REFERENCES

- [1] G. Eason, B. Noble, and I. N. Sneddon, "On certain integrals of Lipschitz-Hankel type involving products of Bessel functions," Phil. Trans. Roy. Soc. London, vol. A247, pp. 529–551, April 1955.
- [2] J. Clerk Maxwell, A Treatise on Electricity and Magnetism, 3rd ed., vol. 2. Oxford: Clarendon, 1892, pp.68–73.

- [3] I. S. Jacobs and C. P. Bean, "Fine particles, thin films and exchange anisotropy," in *Magnetism*, vol. III, G. T. Rado and H. Suhl, Eds. New York: Academic, 1963, pp. 271–350.
- [4] K. Elissa, "Title of paper if known," unpublished.
- [5] R. Nicole, "Title of paper with only first word capitalized," J. Name Stand. Abbrev., in press.
- [6] Y. Yorozu, M. Hirano, K. Oka, and Y. Tagawa, "Electron spectroscopy studies on magneto-optical media and plastic substrate interface," *IEEE Transl. J. Magn. Japan*, vol. 2, pp. 740–741, August 1987 [Digests 9th Annual Conf. Magnetism Japan, p. 301, 1982].
- [7] M. Young, *The Technical Writer's Handbook*. Mill Valley, CA: University Science, 1989.

IEEE conference templates contain guidance text for composing and formatting conference papers. Please ensure that all template text is removed from your conference paper prior to submission to the conference. Failure to remove the template text from your paper may result in your paper not being published.