Interaktionskonzept Mqtt\_exercise\_2021  
  
Dadurch das Mqtt heutzutage vielseitig Einsatz findet, wie für Sensordaten, klassische Nachrichten, Aktienkurse oder Kurzmitteilungen, wie bei Facebooks Mobiler 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 Nachrichten strukturiert, z.B. Haus/Wohnzimmer/Sofa.  
  
So hat sich unsere Gruppe die Aufgabe gestellt, das Mqtt Projekt "mqtt\_exercise\_2021 " zu Programmieren und so Kenntnisse zum Thema Mqtt zu sammeln.  
  
In dem Projekt haben wir die Aufgabe erhalten, einen Server zu Programmieren der Nachrichten von anderen Clients (User, Services, Taxi) empfangen und diesen auch etwas zusenden kann.  
Server: Registriert alle Clients, verteilt IDS, übermittelt Nachrichten, erhält Nachrichten, leitet/organisiert den Ablaufplan.  
  
User: anmelden bei Server, order Fahrzeug, sendet Koordinaten, Fahrzeug kommt, wird zum Zielort gebracht  
  
Taxi: Anmeldung Server, fährt zu den Koordinaten, übermittelt Koordinaten, übermittelt Nachrichten, erhält Nachrichten.  
  
Services: anmelden bei Server, sendet Koordinaten, warten auf Nachricht, bekommt Koordinaten von User, zum User fahren, Fahrzeug ist bussy, User zum Ziel fahren, Fahrzeug wieder free  
  
Das Ziel des Projektes ist es, ein Programm zu schreiben, welches ein Mqtt Ablauf zeigt und eine funktionierende Kommunikation zwischen den Clients enthält.  
  
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.   
  
  
  
  
  
  
  
  
  
  
  
  
  
  
Installation:  
Um MQTT in Python benutzen zu können, sind zunächst ein paar Installationen nötig.  
Hier zu gehören zum einen die Installation von Python version 2.7.9+. Um Python zu installieren, wird über diese Webseite <https://www.python.org/downloads/> Python heruntergeladen und danach via .exe Datei installiert.  
MQTT wird über die Windows Console „cmd“ mit dem Befehl „pip install paho-mqtt“ installiert.  
Im Code muss dann noch der import Befehl „import paho.mqtt.client as mqtt“ hinzugefügt werden, damit   
  
  
  
**Vorbereitung/Recherche**   
Server:  
Um Nachrichten zu verwalten wird ein Server mit einem zugehörigen Port benötigt.  
Hierzu gibt es in MQTT ein paar kostenlose und öffentliche Server/Broker, mit denen der erstellte Code getestet werden kann. Hierzu gehören:  
test.mosquitto.org, mqtt.eclipse.org und broker.hivemq.com. Über diese Adressen sollten keine sensiblen Daten versendet werden. Dann ist es besser einen eigenen MQTT-Broker zu benutzen.   
Jeder dieser Test-Server funktioniert nach dem publish, subscribe Prinzip.  
Also ein MQTT-Client published eine Nachricht an einen Broker und andere Clients können den Broker subscriben, um Nachrichten zu empfangen.  
Wir haben für unser Projekt die Adresse „test.mosquitto.org“ mit dem Port: 1883 verwendet.  
  
  
Java ist eine statisch typisierte und kompilierte Sprache, und Python ist eine dynamisch typisierte und interpretierte Sprache. Dieser einzelne Unterschied macht Java zur Laufzeit schneller und einfacher zu debuggen, aber Python ist einfacher zu verwenden und leichter zu lesen.  
Weil die Verwendung von Python einfacher ist, habe wir Python verwendet.  
  
Es gibt verschiedenste Python-MQTT-Open-Source-Clientbibliotheken, wie paho-mqtt, HBMQTT und gmqtt. Die HBMQTT und gmqtt Bibliothek unterstützen nicht die neuste MQTT version.  
Deswegen ist die beste Bibliothek, um MQTT in Python zu programmieren die paho-mqtt Clientbibliothek.