Hospitals in Smart Cities

Smart City Project Using MQTT

MELANIE LÖBEL
MATRIKEL-NR. 2170582
INTERAKTIONSTECHNIK UND DESIGN, 6. SEMESTER

14. Juli 2020

Motivation

Fig. 1: Accident cases

Motivation

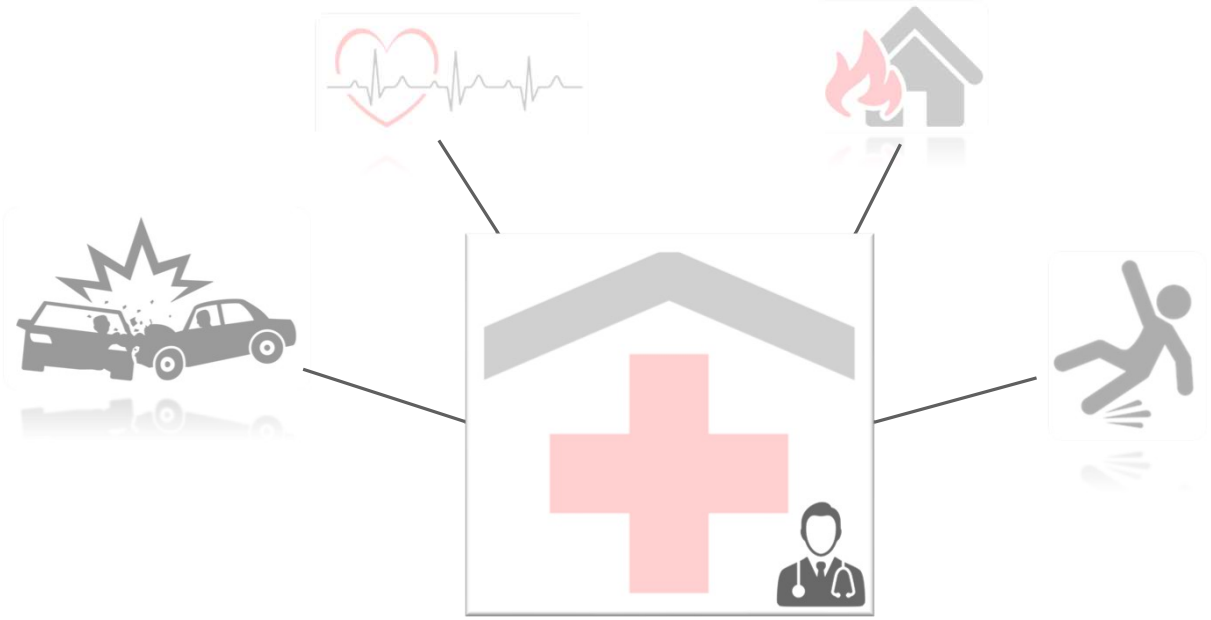


Fig. 1: Accident cases

- Which hospital is the nearest one?
- What kind of specialists does it have?
- Is there the respective doctor which is needed?

Availability info is needed:

- Does the hospital has free rooms?
- Does the doctor has free appointments?

HOCHSCHULE
HAMM-LIPPSTADT

Agenda

- Product overview and context (using MQTT)
- Concept
  - Requirements
  - Use Case
- Implementation in relationship to the requirements
  - Definition of Classes, attributes and functions
  - Objects with their relationships
- Conclusion and Results
- References and Bibliography

Agenda

- Product overview and context (using MQTT)
- Concept
  - Requirements
  - Use Case
- Implementation in relationship to the requirements
  - Definition of Classes, attributes and functions
  - Objects with their relationships
- Conclusion and Results
- References and Bibliography

Product overview and context

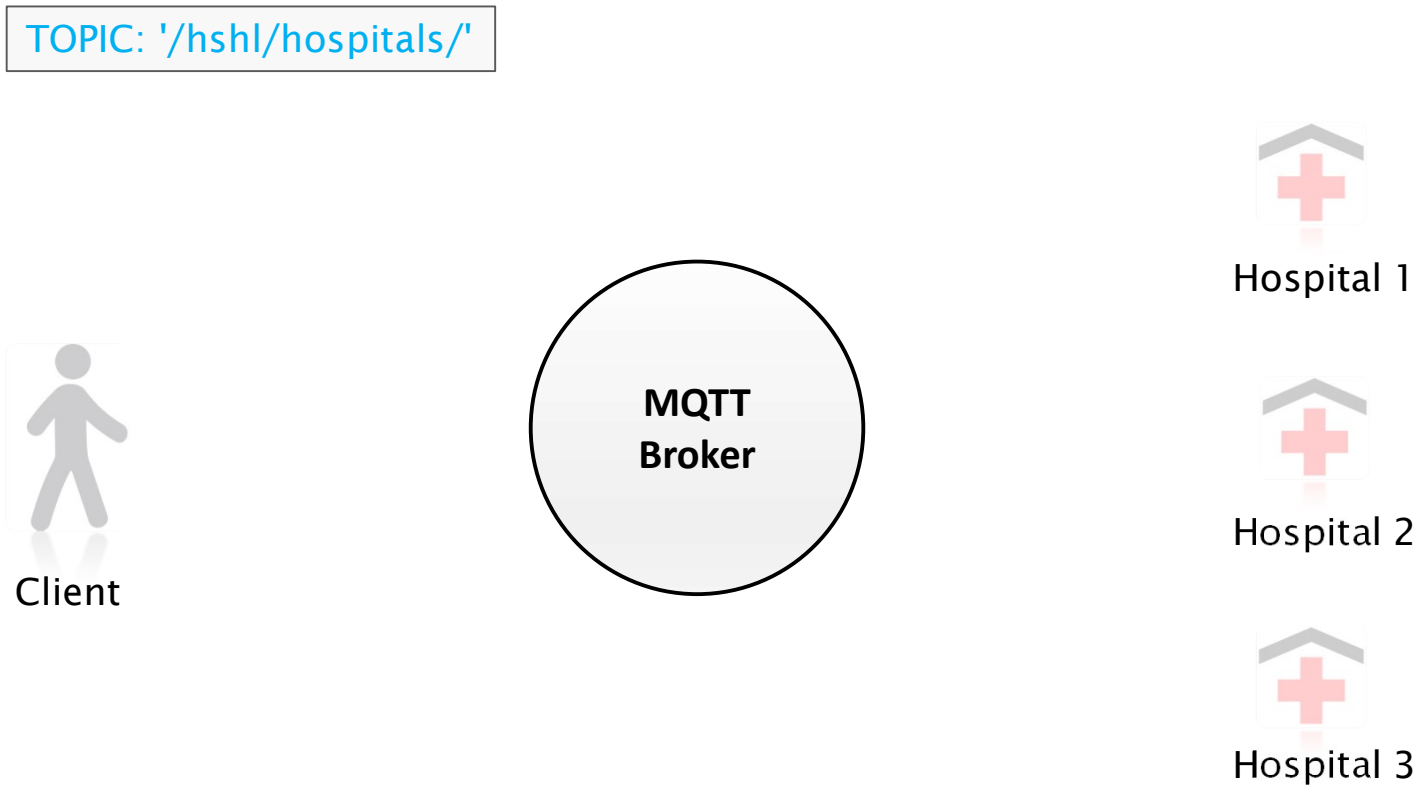


Fig. 2: Product overview

Product overview and context

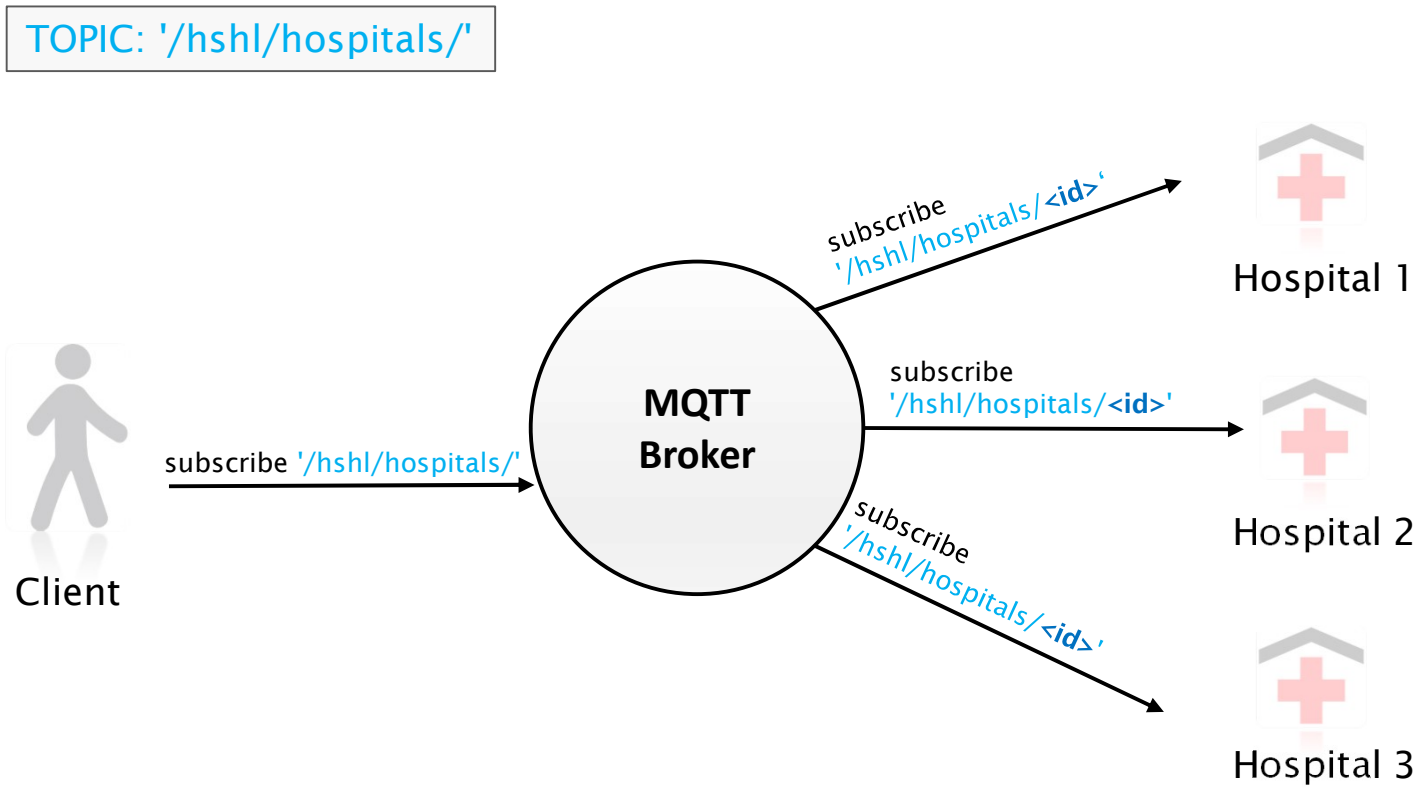


Fig. 2: Product overview

Product overview and context

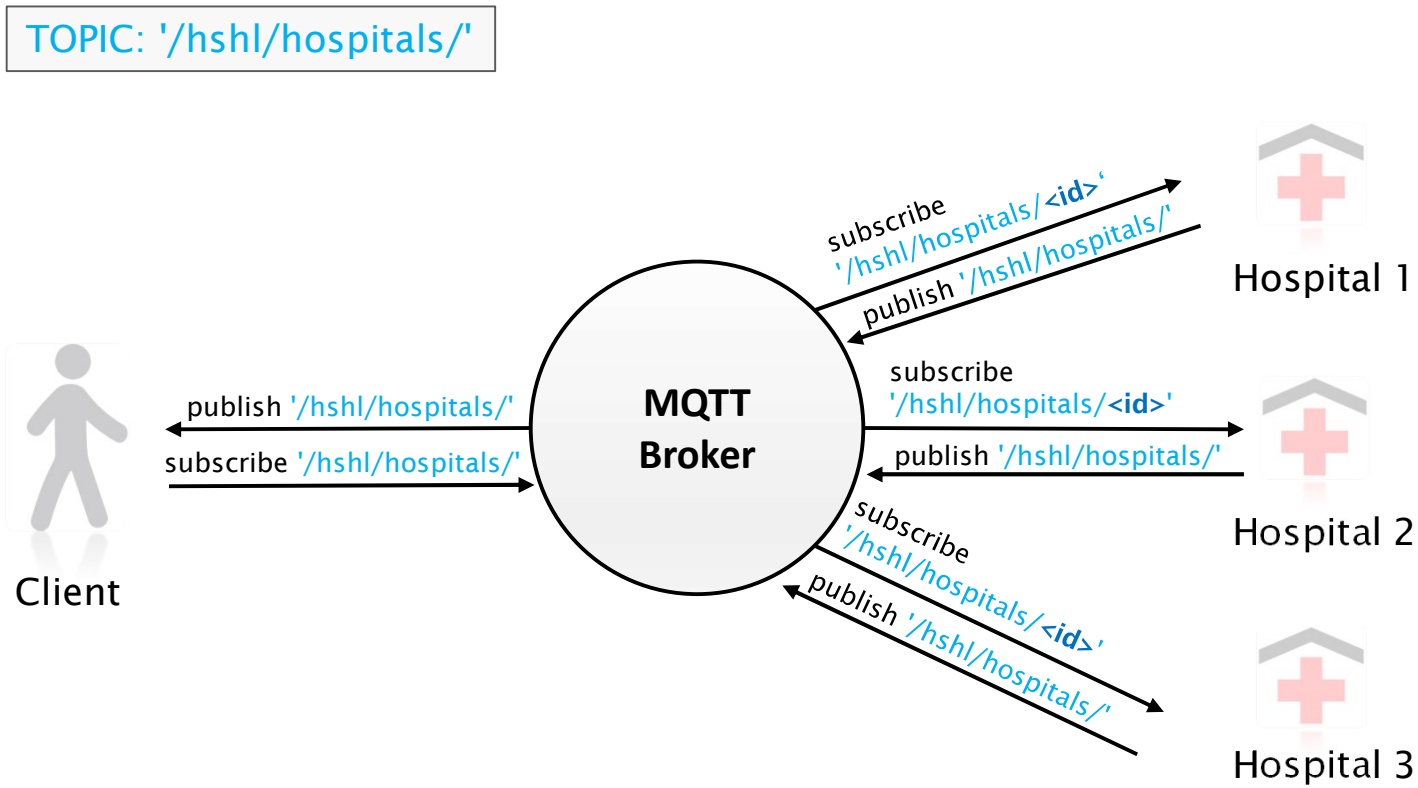


Fig. 2: Product overview

Agenda

- ✓ Product overview and context (using MQTT)
- Concept
  - Requirements
  - Use Case
- Implementation in relationship to the requirements
  - Definition of Classes, attributes and functions
  - Communication with the server
- Conclusion and Results
- References and Bibliography

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Attributes

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Attributes

Functions

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Attributes

Functions

Functions regarding MQTT

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Attributes

Functions

Functions regarding MQTT

Non Functional Requirements

Tab. 1: Requirements

Use Case

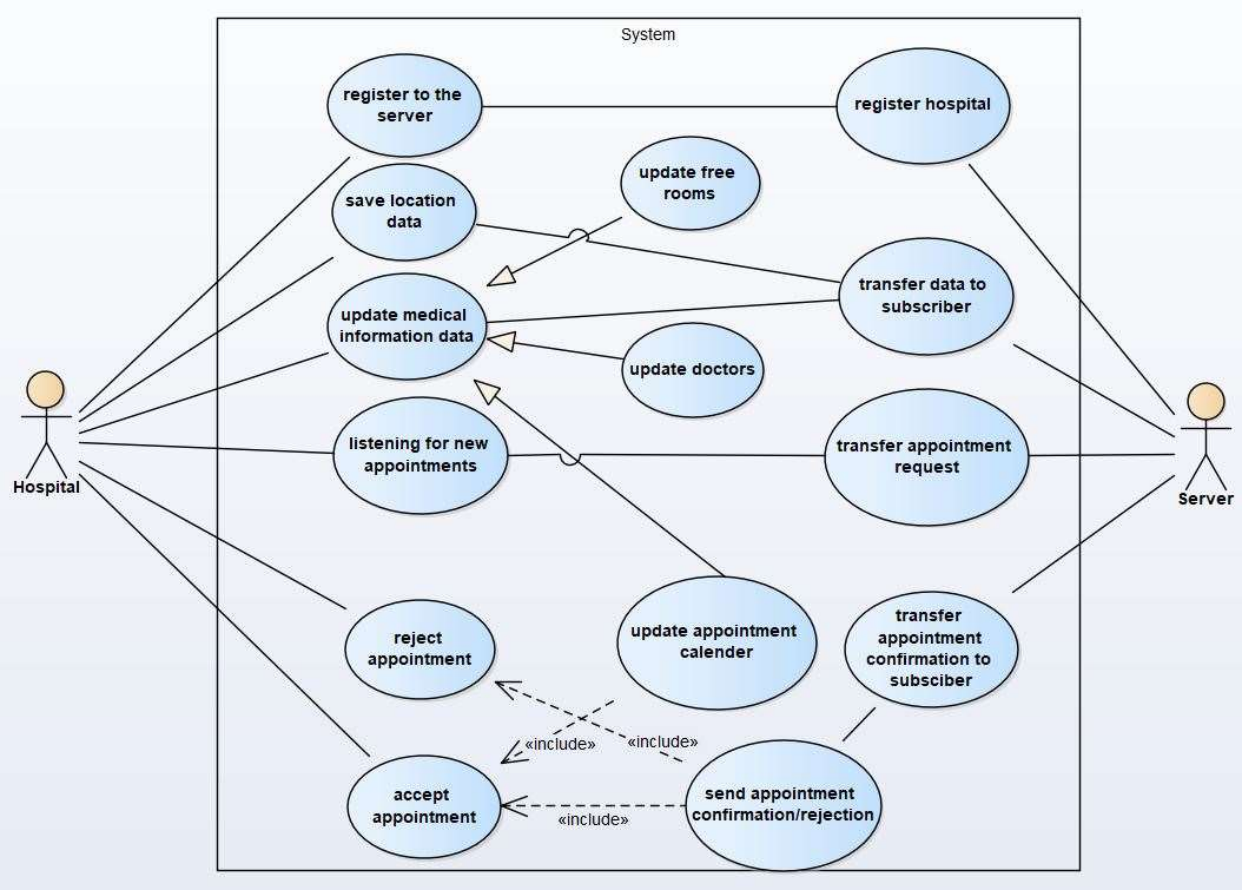


Fig. 3: Use Case Diagram

Agenda

- ✓ Product overview and context (using MQTT)
- ✓ Concept
  - ✓ Requirements
  - ✓ Use Case
- Implementation in relationship to the requirements
  - Definition of Classes, attributes and functions
  - Objects with their relationships
- Conclusion and Results
- References and Bibliography

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

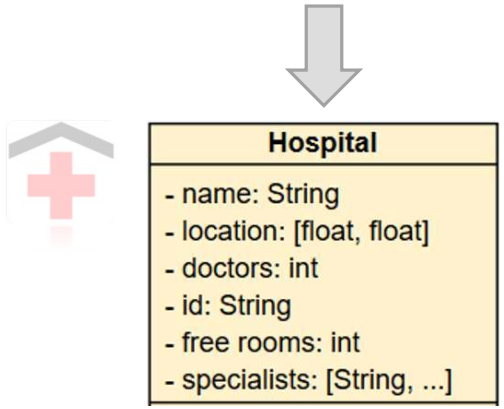
Tab. 1: Requirements

Functional Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.

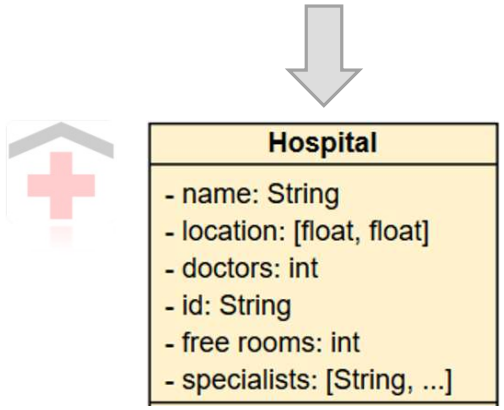
Functional Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.



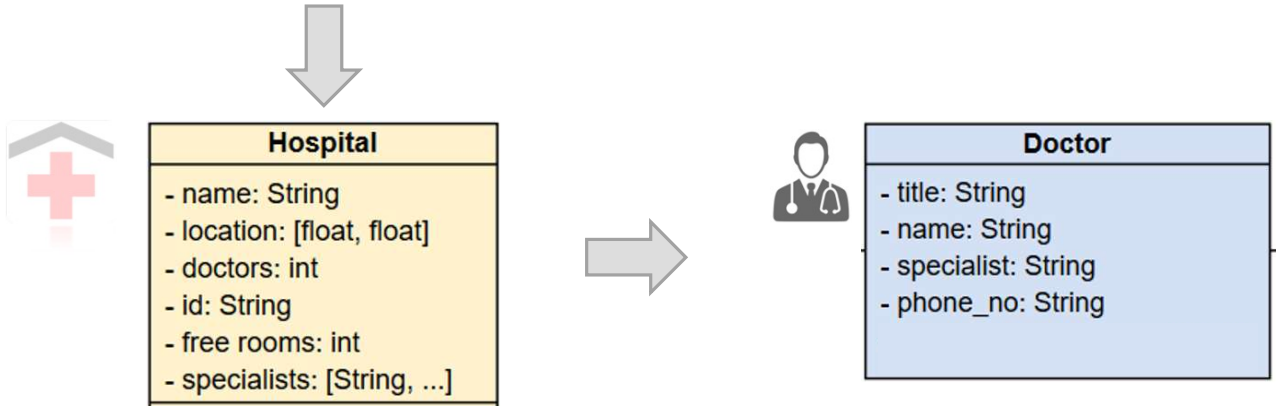
Functional Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.



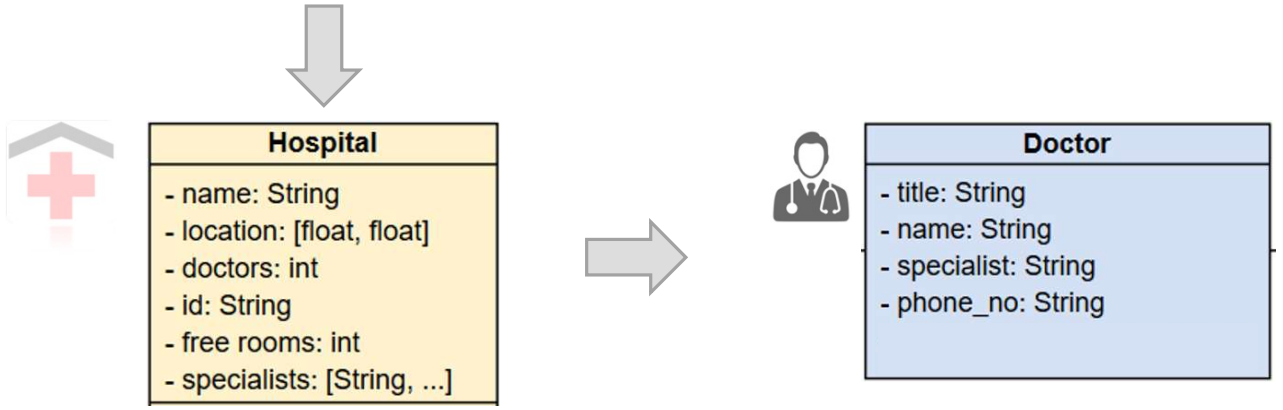
Functional Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.



Functional Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓



Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

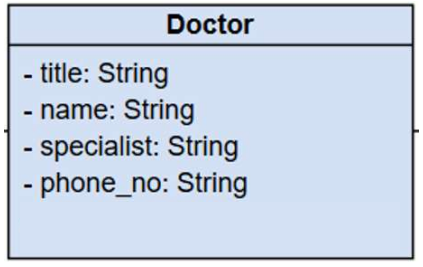
Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Functional Requirements

Requirements	
ID.	Description
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".



Functional Requirements

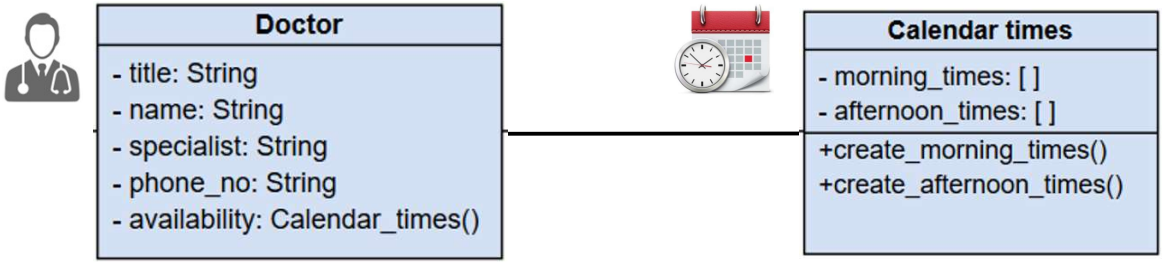
Requirements	
ID.	Description
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".

Doctor

- title: String
- name: String
- specialist: String
- phone_no: String
- availability: Calendar_times()

Functional Requirements

Requirements	
ID.	Description
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".



Make an appointment

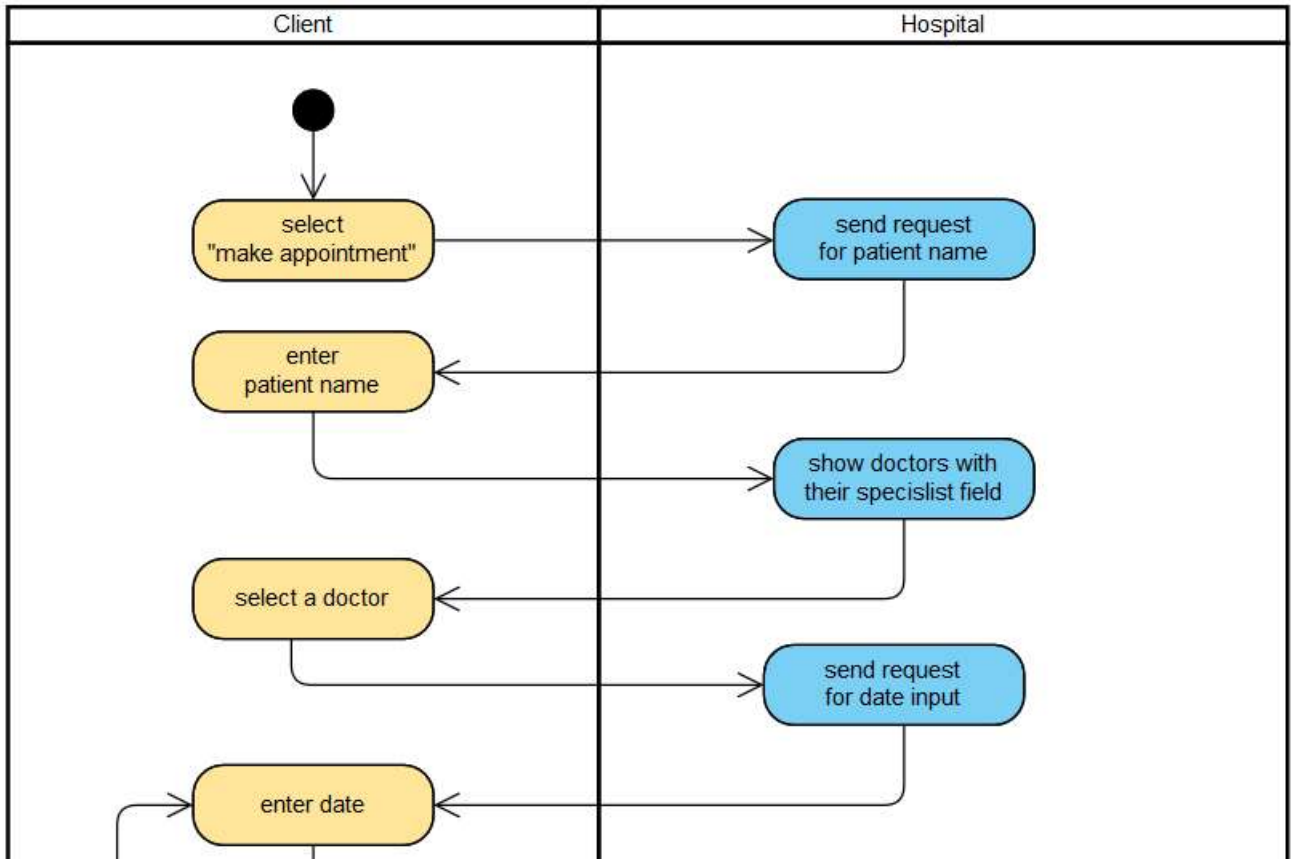


Fig. 4: Activity Diagram “Make an appointment”

Make an appointment

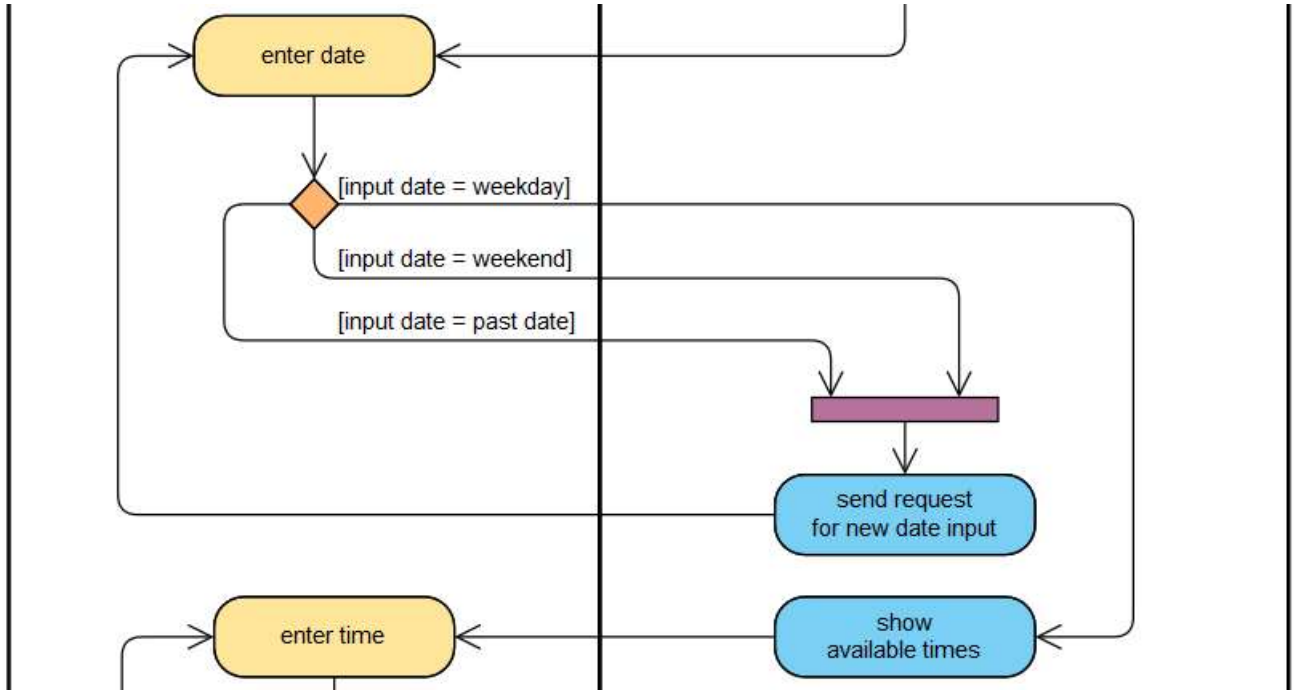


Fig. 4: Activity Diagram “Make an appointment”

Make an appointment

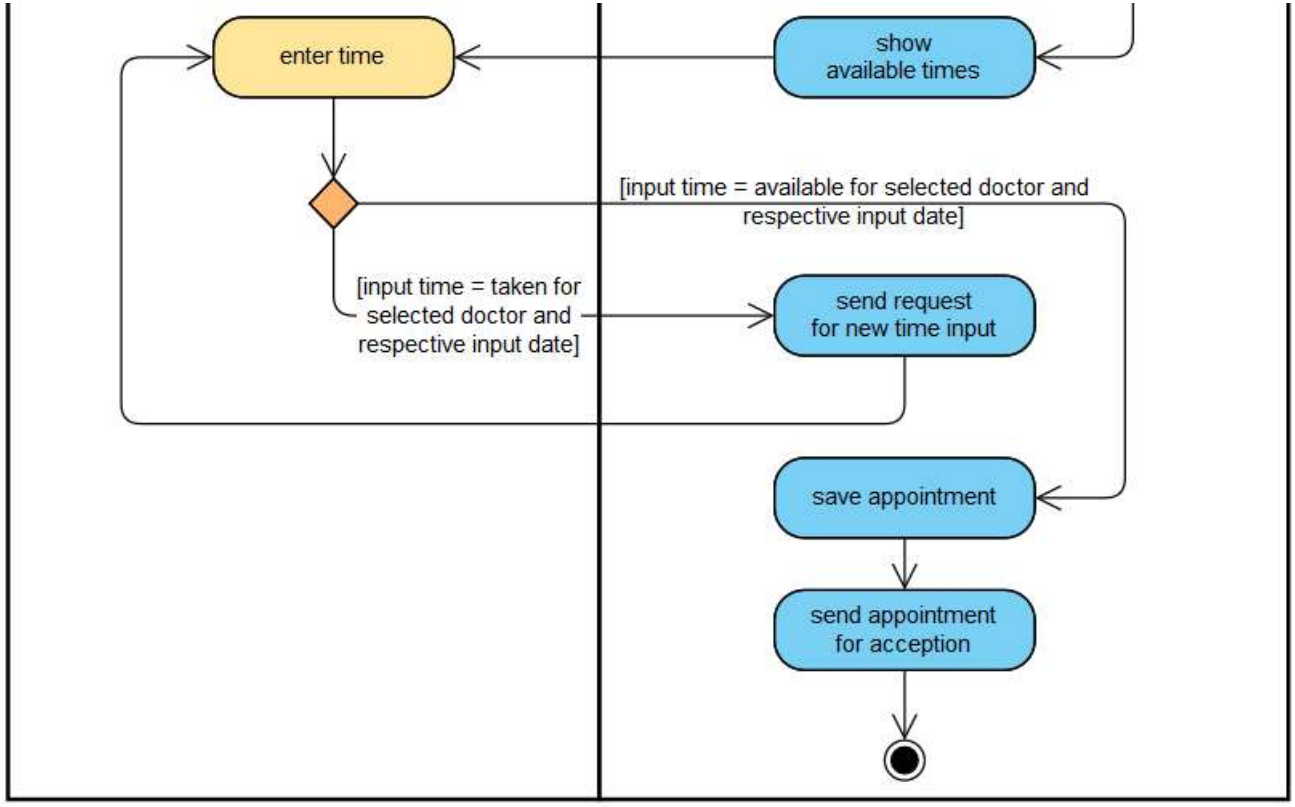


Fig. 4: Activity Diagram "Make an appointment"

Functional Requirements

Requirements	
ID.	Description
FR 4	The system must check availability of doctors. ✓
FR 4.1	The system must get a request for free appointment of a doctor. ✓
FR 4.2	The system must check the date which was given as input. ✓
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday. ✓
FR 4.3	The system must send a message which times are available for the respective doctor. ✓
FR 4.4	The system must check the time which was given as input. ✓
FR 4.4.1	The system must check if the input time for the respective date and doctor is free. ✓
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor. ✓
FR 4.6	The system must send a message info "Accepted appointment". ✓

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors. ✓
FR 4.1	The system must get a request for free appointment of a doctor. ✓
FR 4.2	The system must check the date which was given as input. ✓
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday. ✓
FR 4.3	The system must send a message which times are available for the respective doctor. ✓
FR 4.4	The system must check the time which was given as input. ✓
FR 4.4.1	The system must check if the input time for the respective date and doctor is free. ✓
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor. ✓
FR 4.6	The system must send a message info "Accepted appointment". ✓
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital.
FR 1.1	The system must save the name, the ID and the total number of free rooms.
FR 2	The system must save location for each hospital via GPS coordinates.
FR 3	The system must save medical information about the hospital.
FR 3.1	The system must save the medical specialist fields.
FR 3.2	The system must save the number of doctors.
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors.
FR 4.1	The system must get a request for free appointment of a doctor.
FR 4.2	The system must check the date which was given as input.
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday.
FR 4.3	The system must send a message which times are available for the respective doctor.
FR 4.4	The system must check the time which was given as input.
FR 4.4.1	The system must check if the input time for the respective date and doctor is free.
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor.
FR 4.6	The system must send a message info "Accepted appointment".
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.

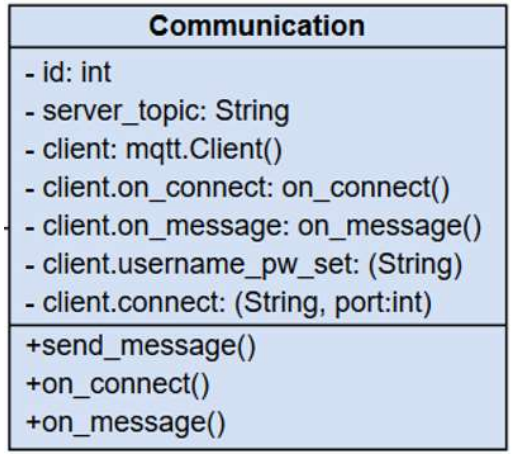
Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.

Communication
- id: int - server_topic: String - client: mqtt.Client() - client.on_connect: on_connect() - client.on_message: on_message() - client.username_pw_set: (String) - client.connect: (String, port:int)
+send_message() +on_connect() +on_message()

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.

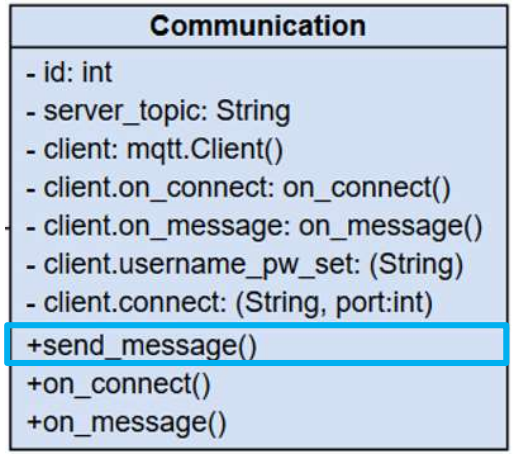



```
174     def get_hospital_info(self):
175         specialists = self.get_doctors_specialist()
176         hospital_info = {
177             'hospital_name': self.name,
178             'location': self.coordinates,
179             'doctors': str(len(self.doctors)),
180             'id': self.id,
181             'freeRooms': str(self.free_rooms),
182             'specialists': specialists
183         }
184         return hospital info
```

Fig. 5: Hospital info defined in class "Hospital"

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.

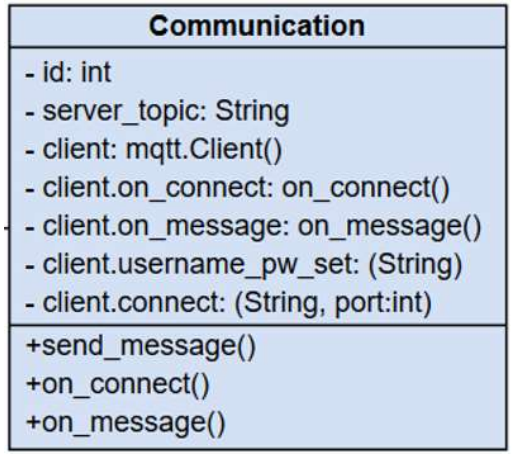



```
174     def get_hospital_info(self):
175         specialists = self.get_doctors_specialist()
176         hospital_info = {
177             'hospital_name': self.name,
178             'location': self.coordinates,
179             'doctors': str(len(self.doctors)),
180             'id': self.id,
181             'freeRooms': str(self.free_rooms),
182             'specialists': specialists
183         }
184         return hospital info
```

Fig. 5: Hospital info defined in class "Hospital"

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker. ✓
FR 7	The system must listen to messages from the server. ✓
FR 8	The system must send messages like the hospital info to the server. ✓




```
174     def get_hospital_info(self):
175         specialists = self.get_doctors_specialist()
176         hospital_info = {
177             'hospital_name': self.name,
178             'location': self.coordinates,
179             'doctors': str(len(self.doctors)),
180             'id': self.id,
181             'freeRooms': str(self.free_rooms),
182             'specialists': specialists
183         }
184         return hospital info
```

Fig. 5: Hospital info defined in class "Hospital"

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors. ✓
FR 4.1	The system must get a request for free appointment of a doctor. ✓
FR 4.2	The system must check the date which was given as input. ✓
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday. ✓
FR 4.3	The system must send a message which times are available for the respective doctor. ✓
FR 4.4	The system must check the time which was given as input. ✓
FR 4.4.1	The system must check if the input time for the respective date and doctor is free. ✓
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor. ✓
FR 4.6	The system must send a message info "Accepted appointment". ✓
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker. ✓
FR 7	The system must listen to messages from the server. ✓
FR 8	The system must send messages like the hospital info to the server. ✓
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Requirements

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken.
FR 4	The system must check availability of doctors. ✓
FR 4.1	The system must get a request for free appointment of a doctor. ✓
FR 4.2	The system must check the date which was given as input. ✓
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday. ✓
FR 4.3	The system must send a message which times are available for the respective doctor. ✓
FR 4.4	The system must check the time which was given as input. ✓
FR 4.4.1	The system must check if the input time for the respective date and doctor is free. ✓
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor. ✓
FR 4.6	The system must send a message info "Accepted appointment". ✓
FR 5	The system must check availability of rooms in a medical specialist field.
FR 5.1	The system must get a request for availability in one medical specialist field.
FR 5.2	The system must send a message how many rooms are free or if there is no room available.
FR 6	The system must register to the server, the MQTT Broker. ✓
FR 7	The system must listen to messages from the server. ✓
FR 8	The system must send messages like the hospital info to the server. ✓
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)
NFR 4	Privacy potection
NFR 5	Safety

Tab. 1: Requirements

Non Functional Requirements

Requirements	
ID.	Description
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)

Non Functional Requirements

Requirements	
ID.	Description
NFR 1	Usability
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment".
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)

Non Functional Requirements

Usability

- menu item to choose options like ‘send data to server’ or ‘make an appointment’

```
25          # possible options saved in a dictionary
26          options = {
27              '1': self.show_doctors,
28              '2': self.show_free_rooms,
29              '3': self.update_free_rooms,
30              '4': self.make_appointment,
31              '5': self.show_appointments,
32              '6': self.send_data_to_server,
33          }
```

Fig. 7: Implementation of menu item, Cut out from class “Hospital”

Non Functional Requirements

Requirements	
ID.	Description
NFR 1	Usability ✓
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment". ✓
NFR 2	Efficiency
NFR 3	Performance
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification)

Non Functional Requirements

Requirements	
ID.	Description
NFR 1	Usability ✓
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment". ✓
NFR 2	Efficiency ✓
NFR 3	Performance ✓
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2 ✓
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification) ✓

Objects with their relationships

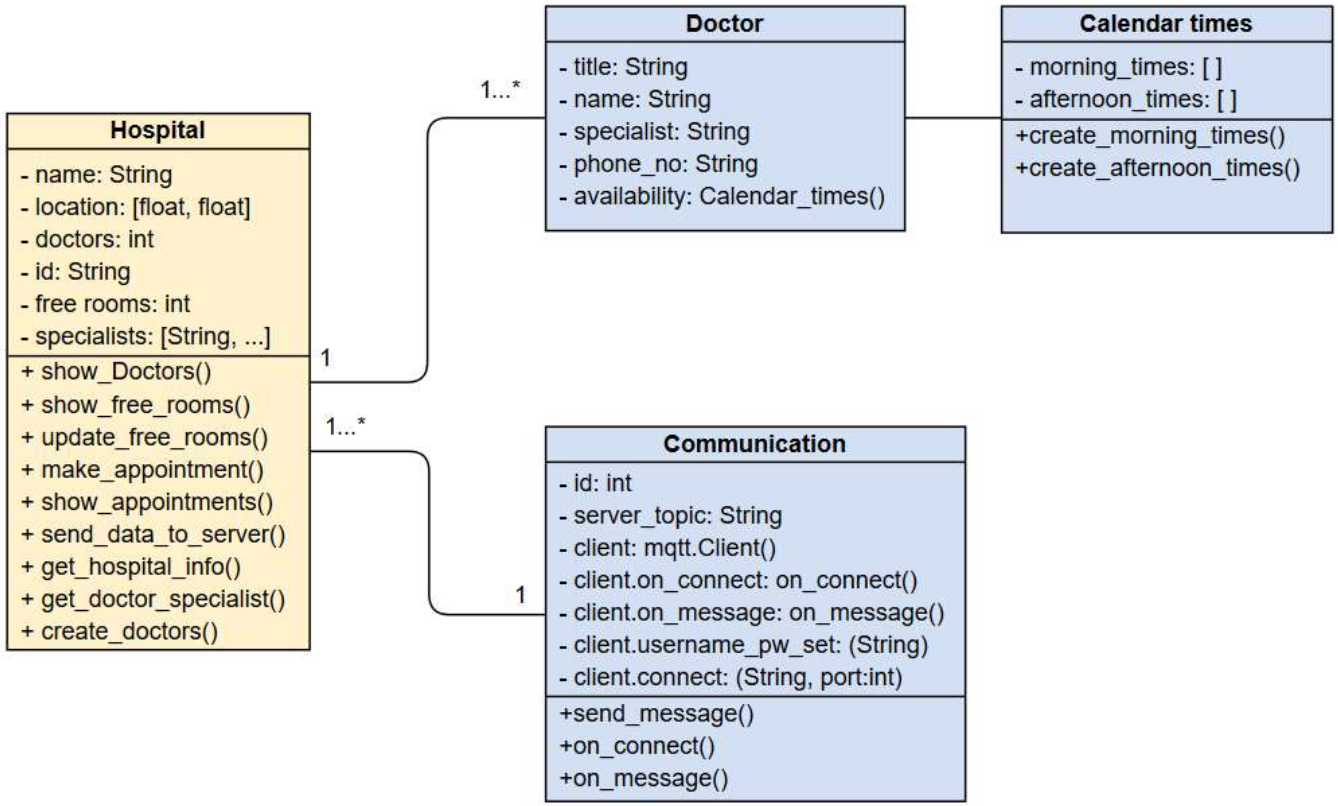


Fig. 8: Class Diagram

Agenda

- ✓ Product overview and context (using MQTT)
- ✓ Concept
  - ✓ Requirements
  - ✓ Use Case
- ✓ Implementation in relationship to the requirements
  - ✓ Definition of Classes, attributes and functions
  - ✓ Objects with their relationships
- Conclusion and Results
- References and Bibliography

Conclusion and Results

Requirements	
ID.	Description
FR1	The system must save user data for each hospital. ✓
FR 1.1	The system must save the name, the ID and the total number of free rooms. ✓
FR 2	The system must save location for each hospital via GPS coordinates. ✓
FR 3	The system must save medical information about the hospital. ✓
FR 3.1	The system must save the medical specialist fields. ✓
FR 3.2	The system must save the number of doctors. ✓
FR 3.3	The system must save number of rooms within each specialist fields and their status, free or taken. tbd
FR 4	The system must check availability of doctors. ✓
FR 4.1	The system must get a request for free appointment of a doctor. ✓
FR 4.2	The system must check the date which was given as input. ✓
FR 4.2.1	The system must check if the input date is an available date (today or in future) and if it is a weekday. ✓
FR 4.3	The system must send a message which times are available for the respective doctor. ✓
FR 4.4	The system must check the time which was given as input. ✓
FR 4.4.1	The system must check if the input time for the respective date and doctor is free. ✓
FR 4.5	The system must save the appointment with patient name, date and time into the calendar from the respective doctor. ✓
FR 4.6	The system must send a message info "Accepted appointment". ✓
FR 5	The system must check availability of rooms in a medical specialist field. tbd
FR 5.1	The system must get a request for availability in one medical specialist field. tbd
FR 5.2	The system must send a message how many rooms are free or if there is no room available. tbd
FR 6	The system must register to the server, the MQTT Broker. ✓
FR 7	The system must listen to messages from the server. ✓
FR 8	The system must send messages like the hospital info to the server. ✓
NFR 1	Usability ✓
NFR 1.1	Menu item to choose between options like "Show free rooms", "Show specialists" and "Make an appointment". ✓
NFR 2	Efficiency ✓
NFR 3	Performance ✓
NFR 3.1	Guarantee that each message is received only once by the intended recipients: QoS (Quality of service) level 2 ✓
NFR 3.2	Usage of a simple and language independent data exchange format: JSON (Java Script Object Notification) ✓
NFR 4	Privacy protection ✓
NFR 5	Safety ✓

Tab. 1: Requirements

Hospitals in Smart Cities

Makes your search smarter and more efficient !

References

Fig. 1	Self made figure by using sources: *http://www.attorneyganim.com*, *https://www.bing.com*, *https://image.shutterstock.com*
Fig. 2	Product overview: Self made by using sources *https://www.bing.com*
Fig. 3	Use Case Diagram: Self made by using Enterprise Architecture
Fig. 4	Activity Diagram “Make an appointment”: Self made by using *https://online.visual-paradigm.com*
Fig. 5	Definition of hospital info, Cut Out of Class “Hospital”: Self made (cut out of Python code)
Fig. 6a	Cut Out of Class “Communication”: Self made (cut out of Python code)
Fig. 6b	Access from Class “Hospital” to class “Communication”: Self made
Fig. 7	Implementation of menu item, cut out of class “Hospitals”: Self made (cut out of Python code)
Fig. 8	Class Diagramm: Self made by using *https://online.visual-paradigm.com*
Title Fig.	*http://anastasiyazakharova.com/wp-content/uploads*
Tab. 1	Requirements: Self made

Bibliography

[1]
Paul Ferguson and Geoff Huston. Quality of service: delivering QoS on the Internet and in corporate networks. John Wiley & Sons, Inc., 1998

[2]
Taewoo Nam and Theresa A Pardo. “Conceptualizing smart city with dimensions of technology, people, and institutions”. In: Proceedings of the 12th annual international digital government research conference: digital government innovation in challenging times. 2011, pp. 282–291

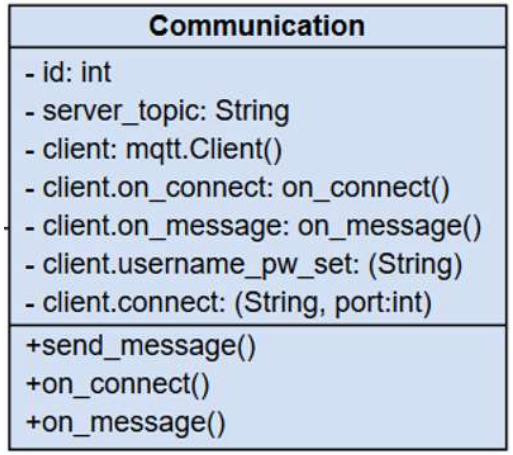
[3]
Anatolijs Zabasta et al. “MQTT Service Broker for Enabling the Inter-operability of Smart City Systems”. In:2018 Energy and Sustainability for Small Developing Economies (ES2DE). IEEE. 2018, pp. 1–6.

[4]
Manu Sporny et al. “JSON-LD 1.1–a JSON-based serialization forLinked Data”. PhD thesis. W3C, 2019

Back up

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.

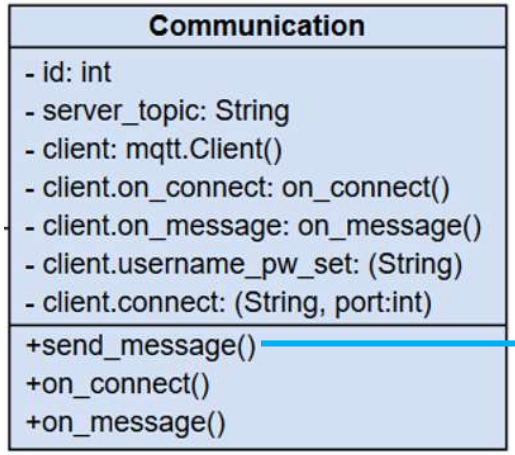



```
174     def get_hospital_info(self):
175         specialists = self.get_doctors_specialist()
176         hospital_info = {
177             'hospital_name': self.name,
178             'location': self.coordinates,
179             'doctors': str(len(self.doctors)),
180             'id': self.id,
181             'freeRooms': str(self.free_rooms),
182             'specialists': specialists
183         }
184         return hospital info
```

Fig. 5: Hospital info defined in class "Hospital"

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker.
FR 7	The system must listen to messages from the server.
FR 8	The system must send messages like the hospital info to the server.




```
20      def send_message(self, message):
21          self.client.publish(
22              self.server_topic,
23              json.dumps(message)
24          )
```

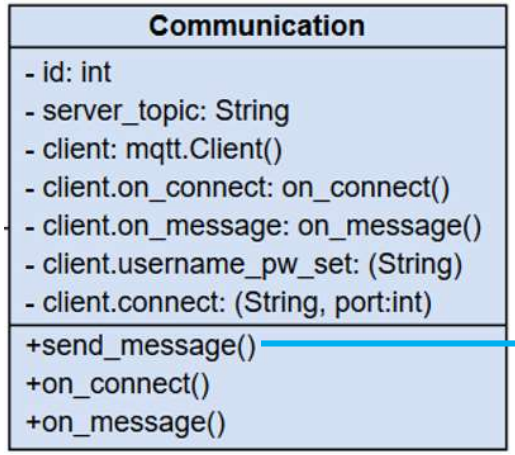
Fig. 6a: Cut Out of Class "Communication"

```
170     def send_data_to_server(self):
171         message = self.get_hospital_info()
172         self.communication.send_message(message)
```

Fig. 6a: Access from Class "Hospital" to class "Communication"

Functional Requirements

Requirements	
ID.	Description
FR 6	The system must register to the server, the MQTT Broker. ✓
FR 7	The system must listen to messages from the server. ✓
FR 8	The system must send messages like the hospital info to the server. ✓




```
20      def send_message(self, message):
21          self.client.publish(
22              self.server_topic,
23              json.dumps(message)
24          )
```

Fig. 6a: Cut Out of Class "Communication"

```
170     def send_data_to_server(self):
171         message = self.get_hospital_info()
172         self.communication.send_message(message)
```

Fig. 6a: Access from Class "Hospital" to class "Communication"