PERMISSION AND ACCESS SUBSYSTEM

Users:

- Student
- Professor
- Technical support unit
- Supervisor
- Security unit
- Interested person(any user from this list)

Role:

- ✔ Provide ability to access certain room (lecture hall, laboratory, etc.) using personal pass card.
- ✔ Provide ability to gain permission to access certain room (lecture hall, laboratory, etc.) using personal pass card.

Usecases:

- ➢ Register in the system
- ➢ Add new permission
- ➢ Request permission to access room
- ➢ Gain permission to access room
- ➢ Access room
- ➢ Open room
- ➢ Lock room

Register in the system usecase

Condition:
Interested person is permitted to register by Supervisor.

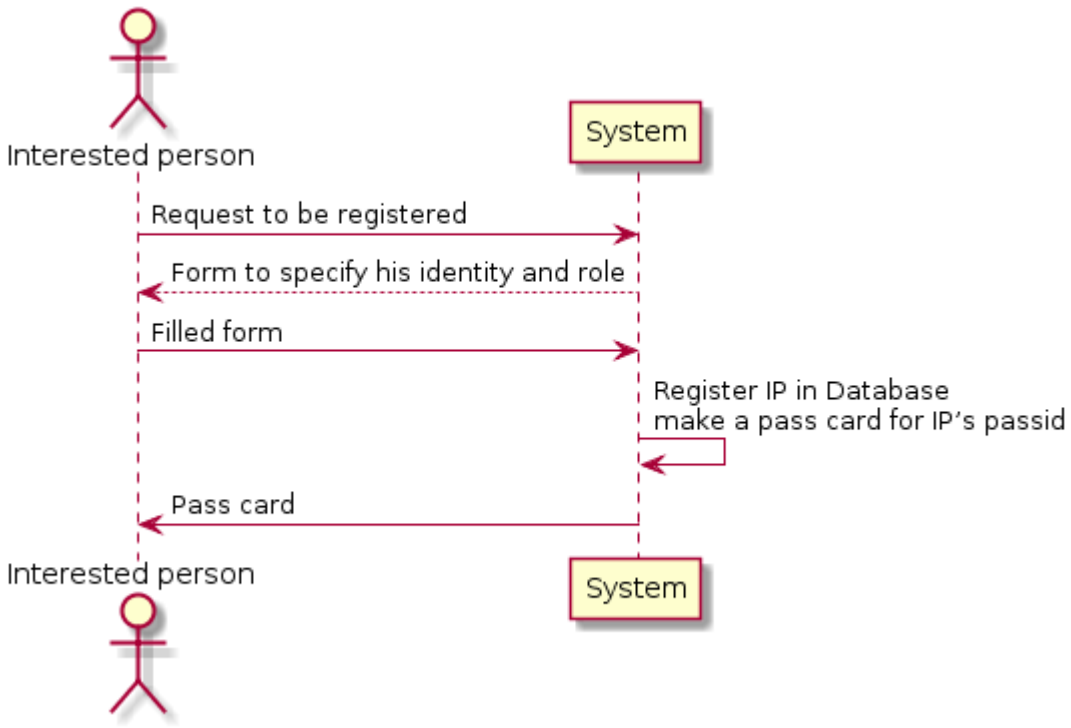
Actors:
- Interested person (IP)
- System

Goal:
Interested person needs to register.

Main scenario:
1. Interested person requests to be registered.
2. System sends form to specify his identity and role.
3. IP submits filled form.
4. System registers IP in Database.
5. System makes a pass card for IP's passid.
6. System gives pass card to IP.

Result:
IP is registered in the system and has pass card.

Diagram:



Plantuml code:

```
@startuml
actor "Interested person" as iper
participant System as sys
iper -> sys : Request to be registered
iper <-- sys : Form to specify his identity and role
iper -> sys : Filled form
sys -> sys : Register IP in Database \nmake a pass card for IP’s passid
sys -> iper : Pass card
@enduml
```

Add new permission usecase

Actors:

- Supervisor (SV)
- System

Goal:

Add new permission to access a room for an interested person (IP).

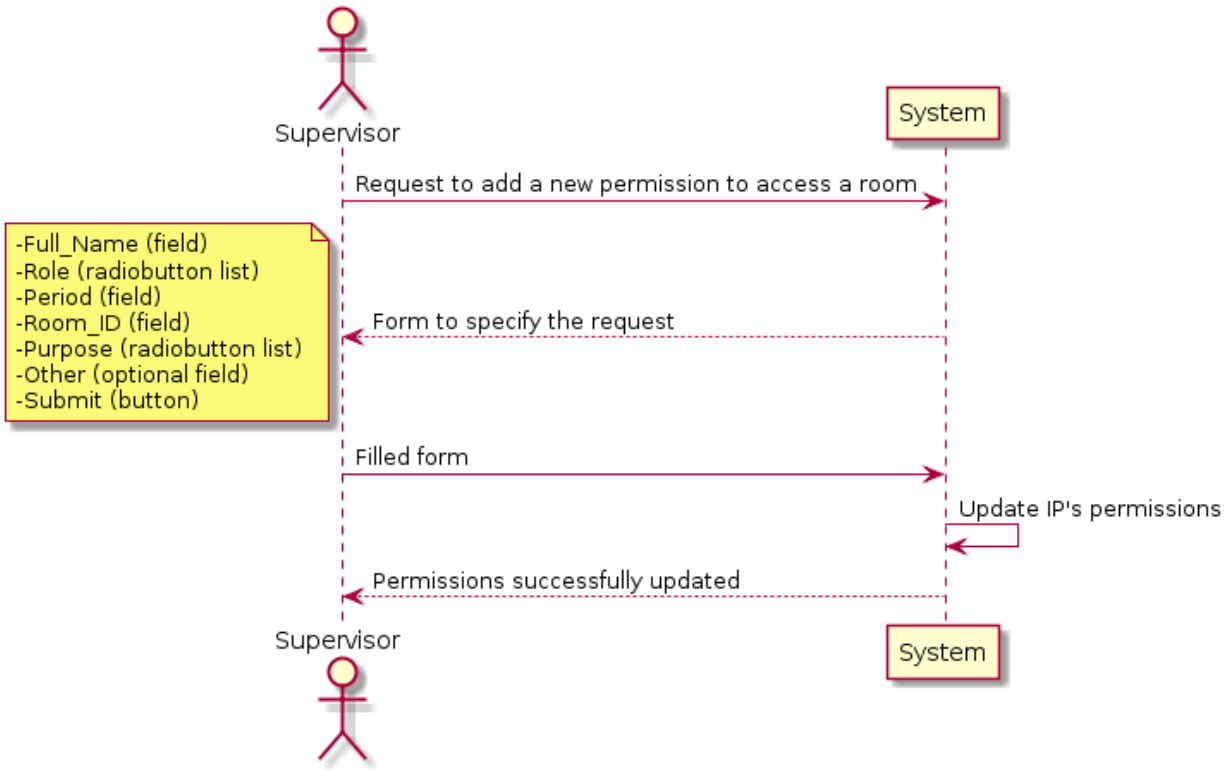
Main scenario:

1. SV sends request to add a new permission to access a room.
2. System sends form *(Full_Name (field), Role (radiobutton list), Period (field), Room_ID (field), Purpose (radiobutton list), Other (optional field), Submit (button))* to specify the request.
3. SV submits filled form by pressing “submit” button.
4. System updates IP’s permissions.
5. System report about the successful update.

Result:

Permission is granted.

Diagram:



Plantuml code:

```
@startuml
actor Supervisor as sv
participant System as sys
sv -> sys : Request to add a new permission to access a room
sv <-- sys : Form to specify the request

note left
  -Full_Name (field)
  -Role (radiobutton list)
  -Period (field)
  -Room_ID (field)
  -Purpose (radiobutton list)
  -Other (optional field)
  -Submit (button)
end note

sv -> sys : Filled form
sys -> sys : Update IP's permissions
sys --> sv : Permissions successfully updated
@enduml
```

Request permission to access room usecase

Actors:

- Interested person (IP)
- System

Goal:

Supervisor will consider granting a permission.

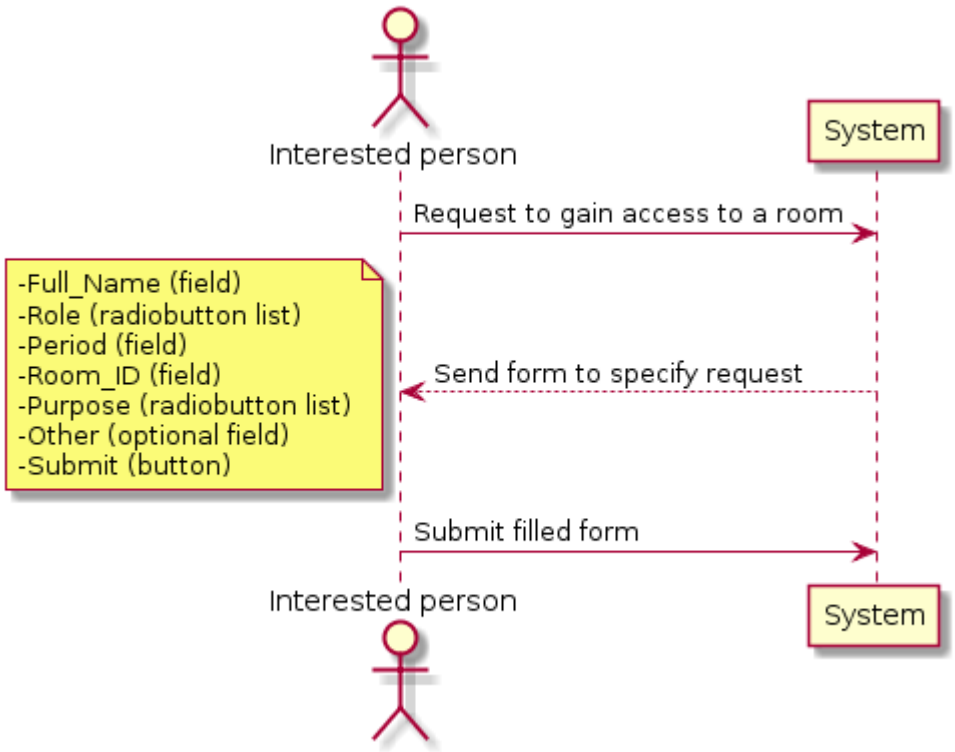
Main scenario:

1. IP sends request to gain access to a room.
2. System sends form *(Full_Name (field), Role (radiobutton list), Period (field), Room_ID (field), Purpose (radiobutton list), Other (optional field), Submit (button))* to specify the request.
3. IP submits filled form by pressing “submit” button.

Result:

Request is created.

Diagram:



Plantuml code:

```
@startuml
actor "Interested person" as iper
participant System as sys
iper -> sys : Request to gain access to a room
sys --> iper : Send form to specify request

note left
  -Full_Name (field)
  -Role (radiobutton list)
  -Period (field)
  -Room_ID (field)
  -Purpose (radiobutton list)
  -Other (optional field)
  -Submit (button)
end note

iper -> sys : Submit filled form
@enduml
```

Gain permission to access room usecase

Conditions:

- Interested person made a request.
- Supervisor looked into the request and approved it.

Actors:

- Interested person (IP)
- System

Goal:

Interested person can access a certain room.

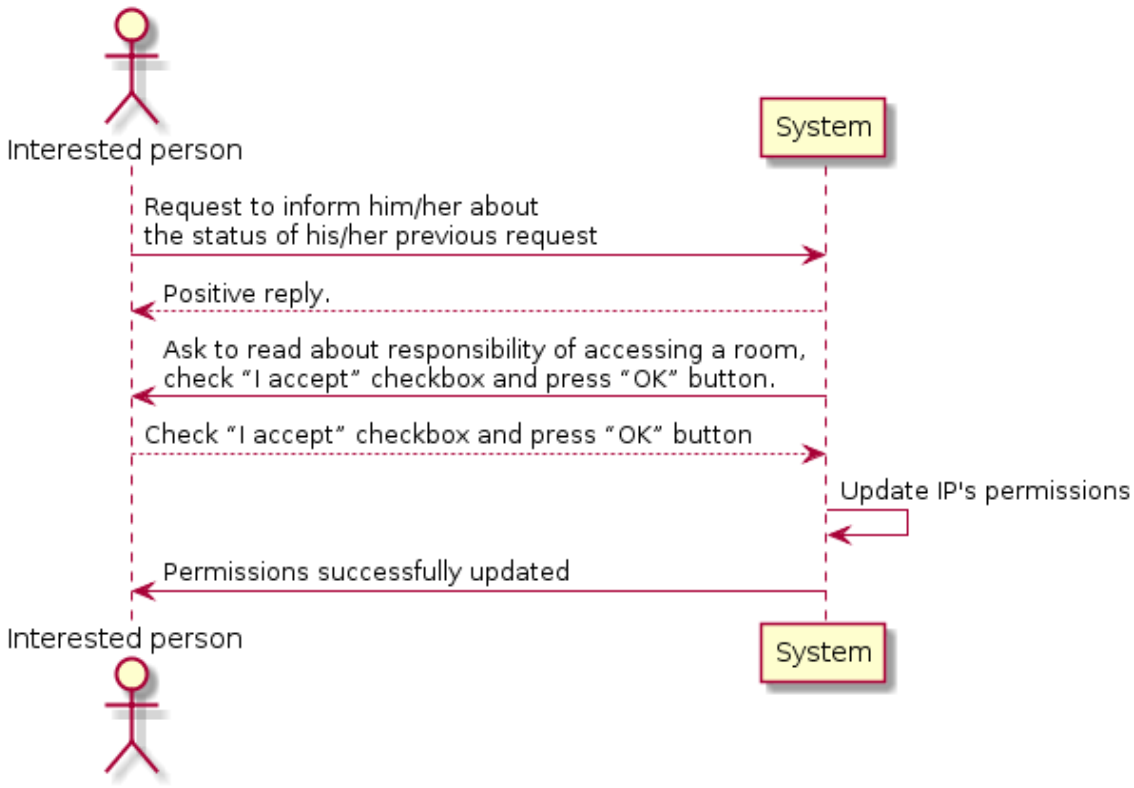
Main scenario:

1. IP requests system to inform him/her about the status of his/her previous request.
2. System sends a positive reply and ask IP to read about responsibility of accessing a room, check “I accept” checkbox and press “OK” button.
3. IP perform given instructions.
4. System updates IP’s permissions.
5. System report about the successful update.

Result:

Permission is granted.

Diagram:



Plantuml code:

```
@startuml
actor "Interested person" as iper
participant System as sys
iper -> sys : Request to inform him/her about \nthe status of his/her previous request
sys --> iper : Positive reply.
sys -> iper : Ask to read about responsibility of accessing a room, \ncheck “I accept” checkbox and press “OK” button.
sys <-- iper : Check “I accept” checkbox and press “OK” button
sys -> sys : Update IP's permissions
sys -> iper : Permissions successfully updated
@enduml
```

Access room usecase

Conditions:
Permission is granted to the interested person.

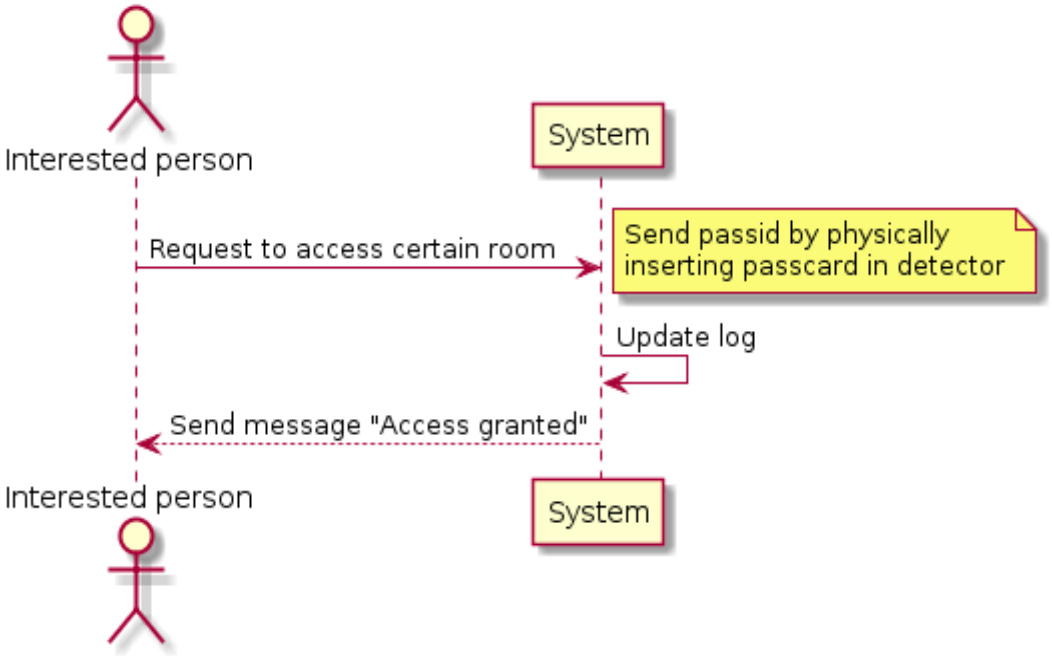
Actors:
- Interested person (IP)
- System

Goal:
Interested person can open a certain room.

Main scenario:
1. IP sends request to access a certain room by inserting his pass card in the room's detector.
2. System creates a log entry about accessing the room.
3. System sends message "Access granted".

Result:
The access is granted. Log is updated.

Diagram:



Plantuml code:

```
@startuml

actor "Interested person" as iper
participant System as sys
iper -> sys : Request to access certain room
note right : Send passid by physically\ninserting passcard in detector
sys -> sys : Update log
iper <-- sys : Send message "Access granted"

@enduml
```

Open room usecase

Conditions:
Access granted to the room.

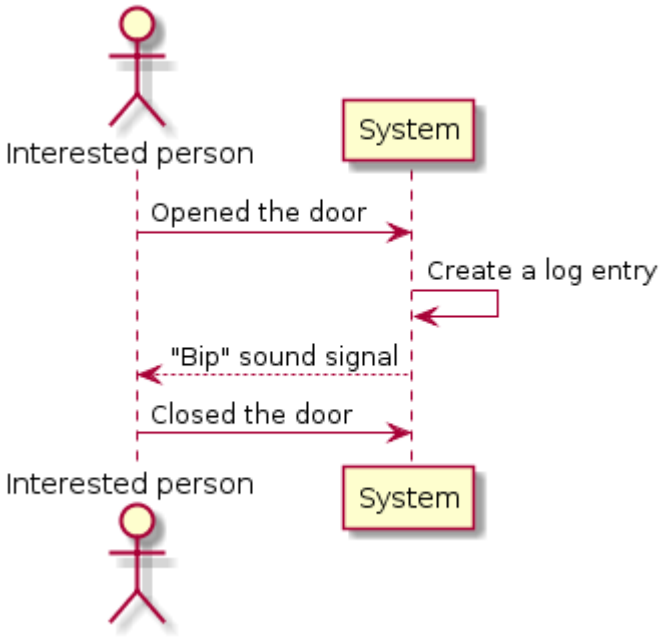
Actors:
- Interested person (IP), not only the person, whom the system granted access.
- System

Goal:
Log the opening process.

Main scenario:
1. IP opens a door.
2. System create a log entry about opening.
3. Release "Bip" sound signal.
4. IP closes a door.

Result:
Log is updated.

Diagram:



Plantuml code:

```
@startuml
actor "Interested person" as iper
participant System as sys
iper -> sys : Opened the door
sys -> sys : Create a log entry
sys --> iper : "Bip" sound signal
iper -> sys : Closed the door
@enduml
```

Lock room usecase

Conditions:

- Access is granted to the interested person.
- The door is closed.

Actors:

- Interested person
- System

Goal:

Lock the door, make open room usecase unavailable.

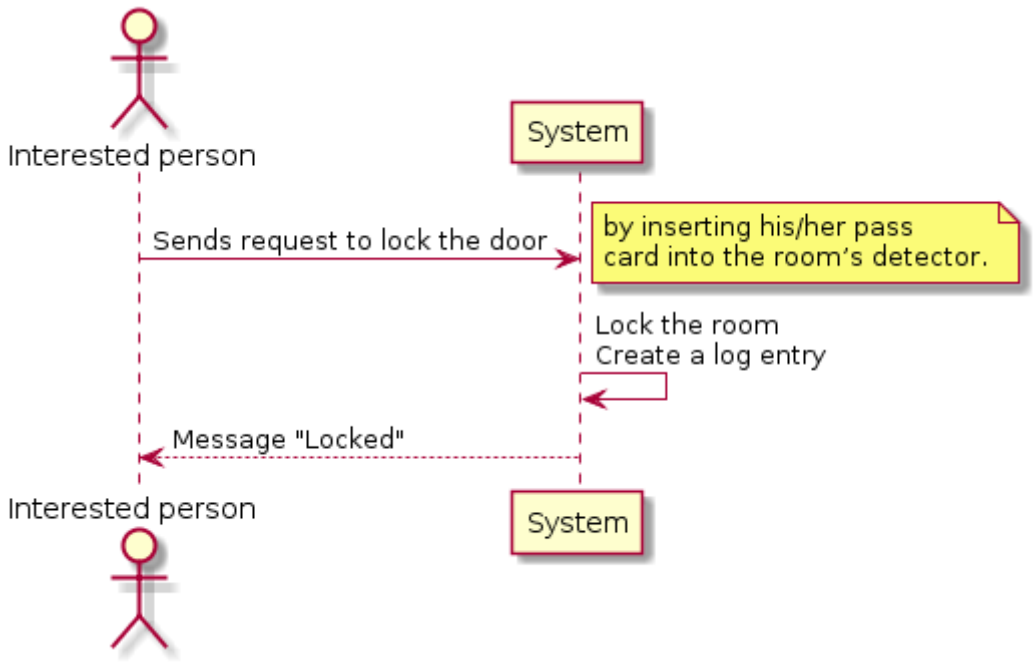
Main scenario:

1. IP sends request to lock the door by inserting his/her pass card into the room's detector.
2. System lock the room.
3. System create a log entry about locking the room.
4. System sends message "Locked".

Result:

Room is locked. Log is updated.

Diagram:



Plantuml code:

```
@startuml
actor "Interested person" as iper
participant System as sys
iper -> sys : Sends request to lock the door
note right : by inserting his/her pass \ncard into the room’s detector.
sys -> sys : Lock the room \nCreate a log entry
sys --> iper : Message "Locked"
@enduml
```