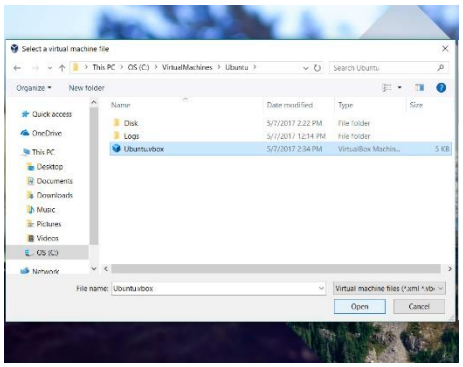
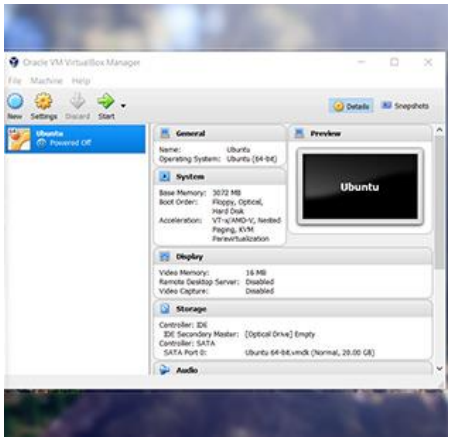
Manual Immersive Experience

Installation



1. Copy the folder VirtualMachines from SICHH USB to C:/
2. Install Oracle VM Virtualbox from https://www.virtualbox.org/wiki/Downloads
3. Open Oracle VM Virtualbox, select Machine > Add, and choose C:/VirtualMachines/Ubuntu/ubuntu.vbox
4. Run the virtual machine to check everything works

Execution of virtual machine



1. Run Oracle VM virtualbox
2. Select Ubuntu from the list
3. Click on Start, this will start the virtual machine
4. When prompted from Ubuntu login screen, select user and insert password:

   Username: `sichh`

   Password: `immersivex`

5. The virtual machine hosting the server is now running

Start the system – Client side

- Each antenna has to be started and correctly placed before running Immersive Experience, to start the antennas the following steps are necessary:
  - On the Raspberry Pi:
    1. Check that the Wifi dongle is connected

    2. Connect the USB of the antenna cable (green) to the USB port of the Raspberry Pi (green)
    3. Connect cable to power Raspberry Pi (blue) to the Power in of Raspberry Pi (blue)

- On the battery pack:
    1. Connect USB of the antenna (red) to the battery (red)
    2. Connect USB for powering the Raspberry Pi (Yellow) to the battery (yellow)
    3. Press button on battery to start

- Put the antennas in the desired place to be tracked

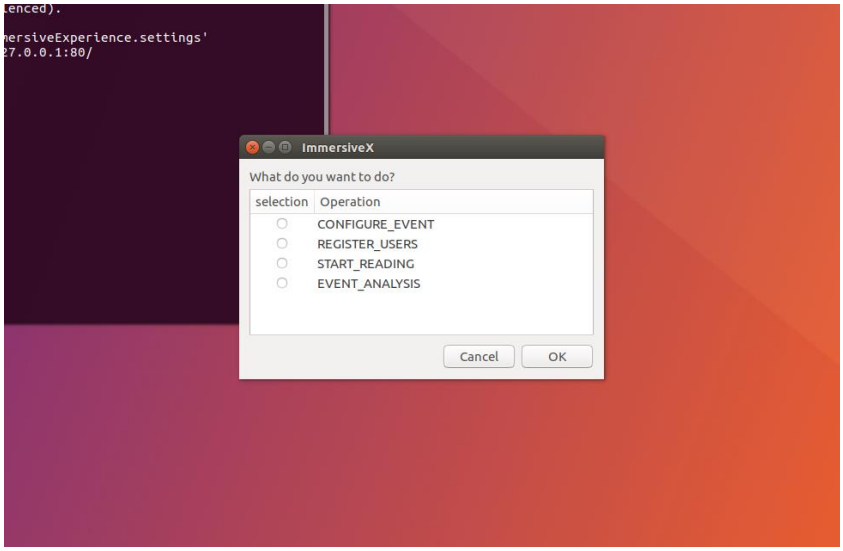
! Try to prevent as much as possible occlusions between the antenna and the tags to be tracked, the optimal positioning of an antenna would be under the top of a door frame)

! Max range of the antennas: 0.9 meters

Start the system – Server side

- Insert Wifi dongle into host computer and check that Wifi is recognized by the virtual machine (Ralink 802.11n WLAN (0101) ->
- Open Oracle VM Virtualbox and start the Ubuntu virtual machine
- On the desktop of the virtual machine, double click on immersiveX.sh to start the User Interface for choosing which operation to execute

- If everything is setup correctly, a message “Server running on http://127.0.0.1” is displayed. After that, the following options show up:

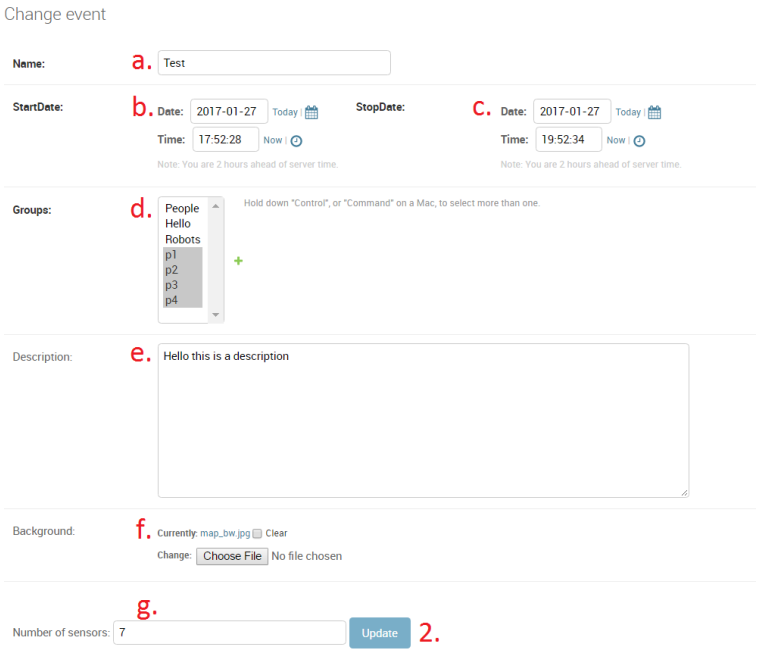


- CONFIGURE_EVENT:

Function:

Used to set up a new event or modify one. On the web page that is displayed, choose if to create a new event or to edit an existing one (to edit click on “Events” and then on the desired event to edit.

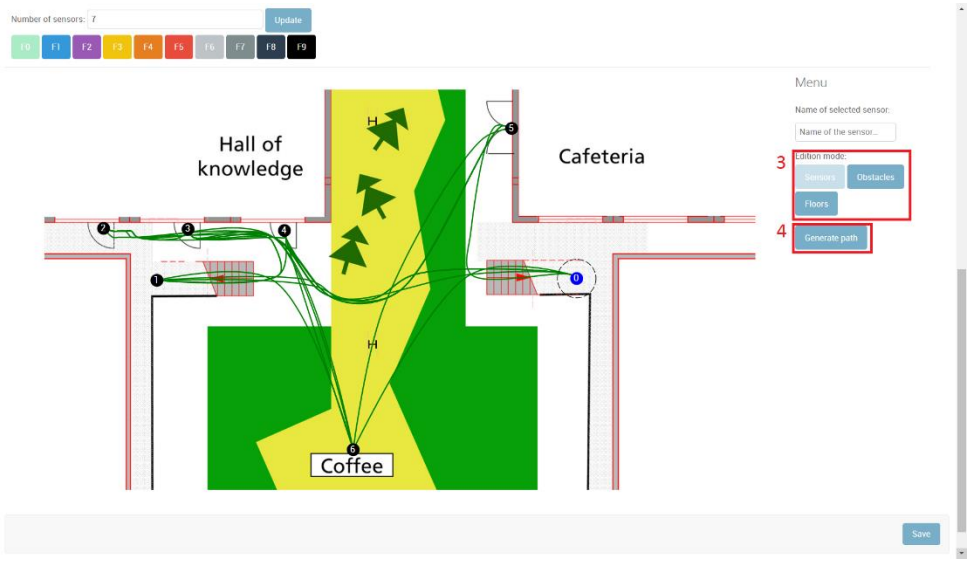
Event features:



1. Pick:
    a. A name,
    b. a start time,
    c. an end time,

d. the list of groups participating to the event (create new groups if necessary),
e. a description for the event (optional)
f. an image to show as background in the various visualizations -> created externally with an image editor (optional)
g. the number of sensors to be used for the event.

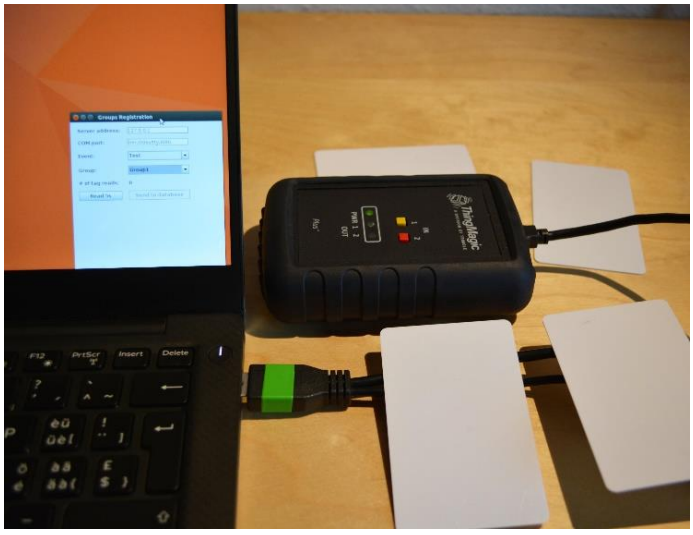
2. Click on Update
3. For the rest of the event's details, there are 3 edition modes:



1. Sensors:
   - Move sensors on the map by drag and drop
2. Obstacles:
   - Draw the walls or the zones where the users are not allowed to pass
3. Floors:
   - Can be used to subdivide a map into different floors:
     - Select floor number
     - Draw corners of that floor in the map

4. When finished setting everything up, click on "Generate path" and check if the paths between the sensors corresponds to the desired ones

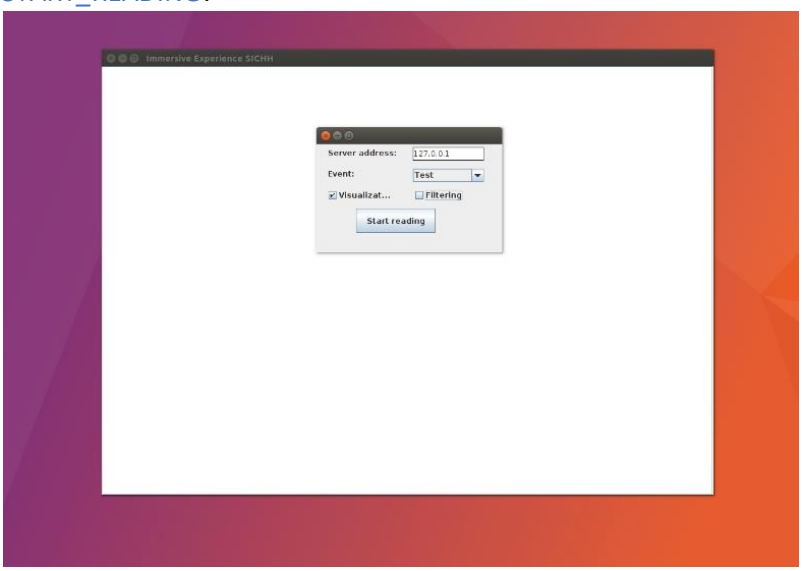
- *REGISTER_USERS:*



1. Before recording data, one has to register the users allowed to the event (association RFID tag-group)
2. Connect an RFID antenna (green cable) to an USB port of the server
3. In the “Group registration” application, select the desired event
4. Put the tags that you want to register next to the RFID antenna and click on “Read 5s”
5. When reading is finished, select the group you want to register the RFID tags to, and click on “Send to database”
6. Repeat for each different group

❗ In case some problems occur, try to remove all USB devices (e.g. Wifi dongle) from the pc and connect only the green cable of the antenna

- *START_READING*:



Function:
Start data collection

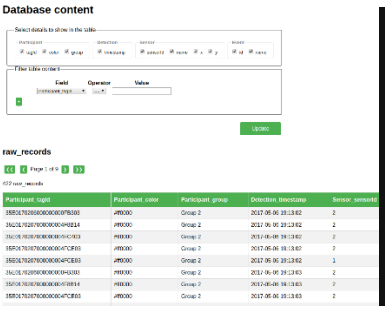
❗ Make sure to have started all antennas, created the wanted event, and registered the participants to the event before starting data collection.

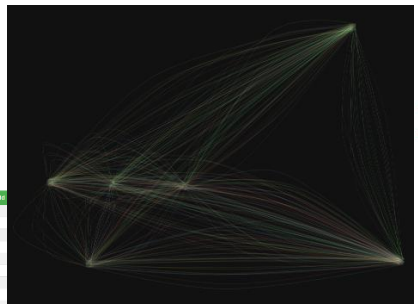
1. In the “Immersive Experience SICHH” application, select the wished event, and choose if you want to display the live visualization, and if to filter the data (in general it is better not to filter it, so deselect it if you don’t have particular needs)
2. Click on “Start reading”
3. A log window is available to check if the antennas could connect to the server, if some have not been detected, one can click on “Stop reading” and then on “Start reading” again

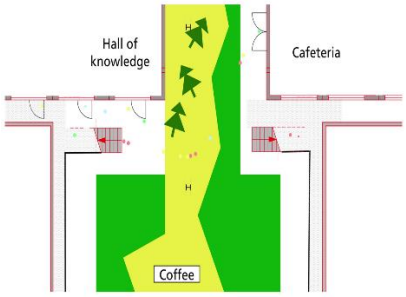
❗ To change window, use Alt + Tab⇆

❗ To stop data collection correctly, click on “Stop reading”

- *EVENT_ANALYSIS*:

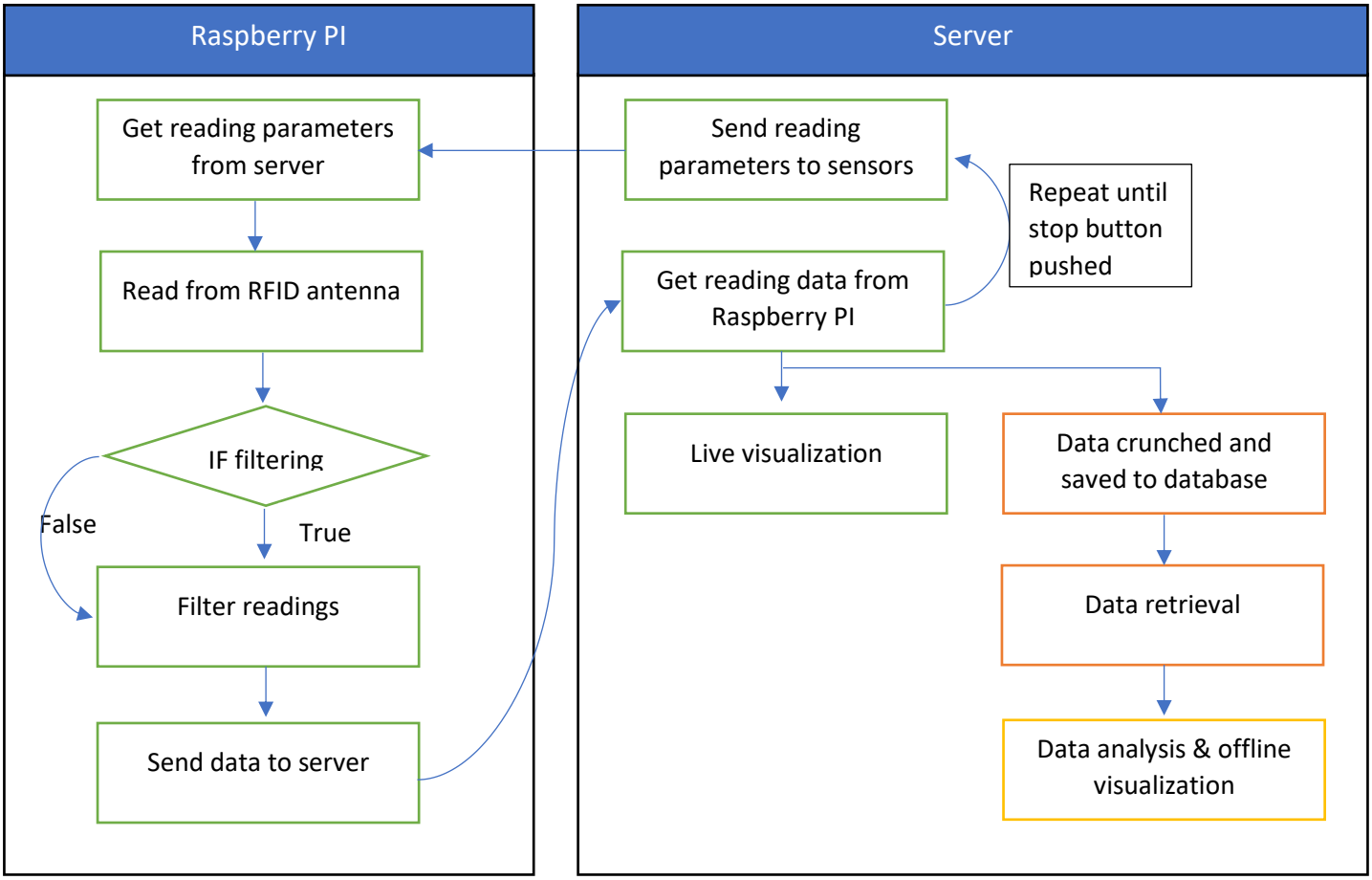






1. Open a web page where you can pick different visualizations, raw data, analysis of time spent in a room by a user, analysis of the movements between different antennas by a user.
    1. Tables:
        - Raw records: the recordings as retrieved from the antennas
        - Occupations: a list of the users that have stayed inside a room, with the time they spent in each room
        - Movements: a list of the movements of registered users from a room to another
    2. Data visualization:
        - Paths: a static visualization of movements between different antennas
        - Movements: a dynamic visualization of movements between different antennas

Software architecture



- Java
- Django (Python)
- HTML + Javascript + CSS (in browser)

Source code

Raspberry Pi

- ### IE2.0-client - java

  `Client.java`: class which handles connections between RPi and server via socket, handle RFID antenna setup, readings and filtering of data from RFID reader.

  `OldRecords.java`: helper class to handle filtering of readings.

  `Config.json`: Configuration file, it has to contain two options “raspberryId”, the identifier number of the RPi being configured, and “comPort”, the serial port the RFID antenna is communicating to, usually /dev/ttyUSB0.

Server

- ### IE2.0-server - java

  `GUI.java`: Create Graphical User Interface to choose some parameters for the data collection and to launch/stop it. On start button clicked, launch a new thread and a socket connection for each RPi.

  `SocketFactory.java`: start socket connection between server and RPi, and launch visualizations.

  `ServerThread.java`: a thread for each RPi. Receive data from RPis and send it both to the live visualization classes and to the Django database.

  `Visualization.java | GradientsVisualization.java`: Classes handling live visualization. The first one shows the color of the detected RFID tag next to each sensor on a map, the other the concentration of people inside a room with a sensor placed at its entrance.

  `Ch.sichh.helpers.*`: some helper classes for the previous listed classes.

- ### ~/Documents/immersiveX – Django (python + HTML + javascript)

  `ImmersiveExperience/settings.py`: Settings of the webserver.

  `hubnet/templates/*`: HTML templates to show webpages with custom content in a browser, where the offline visualizations are defined.

  `Hubnet/models.py`: A description of the objects to be saved in the database, with their attributes.

  `Hubnet/urls.py`: Used to associate some URLs to a view, with some optional parameters.

  `Hubnet/views.py`: Python functions to elaborate data got via URLs, or data from database, and save it to the database or use it as a parameter to be passed to a template.

- ### IE2.0-UserRegistration - java

  Handle the connection between Server and RFID reader, to read some RFID tags. Then associate them to a specific group and store this association to the database.

Raspberry Pi Setup

1. If Raspberry Pi has no operating system yet, install Raspbian
2. Turn Raspberry Pi on
3. Insert Wifi dongle
4. Connect to a wifi
5. Open a terminal window and enter:
    a. `sudo apt-get update`
    b. `sudo apt-get install batmand`
    c. `sudo apt-get install default-jre`
    d. `sudo nano /etc/network/interfaces`
    then replace the text with

```
auto wlan0
iface wlan0 inet static
     address 192.168.1.X
     netmask 255.255.255.0
     wireless-channel 1
     wireless-essid raspberry-network
     wireless-key 1234567890
```

    Replace `X` with the desired number for the Raspberry Pi (use increasing numbers 1-99, without repeating numbers)

    Ctrl + O to save and Ctrl + X to close

6. Copy Client folder from SICHH USB stick to Desktop
7. Open a terminal window and enter:
    a. `sudo nano /.config/lxsession/LXDE_pi/autostart`
    at the end of the file, add the following line:

```
@lxterminal -e- sudo sh
/home/pi/Desktop/Client/autostart.sh
```

    Ctrl + O to save and Ctrl + X to close

    b. `sudo nano ./Desktop/Client/autostart.sh`

    and change the ip address (in the form `192.168.1.1`) to `192.168.1.X`, where `X` is the Raspberry Pi's number.

    c. `sudo nano ./Desktop/Client/config.json`

    and change `raspberry_id` to `X`

8. Reboot. The Raspberry Pi has been configurated.