ICT in Building Design

Final Project

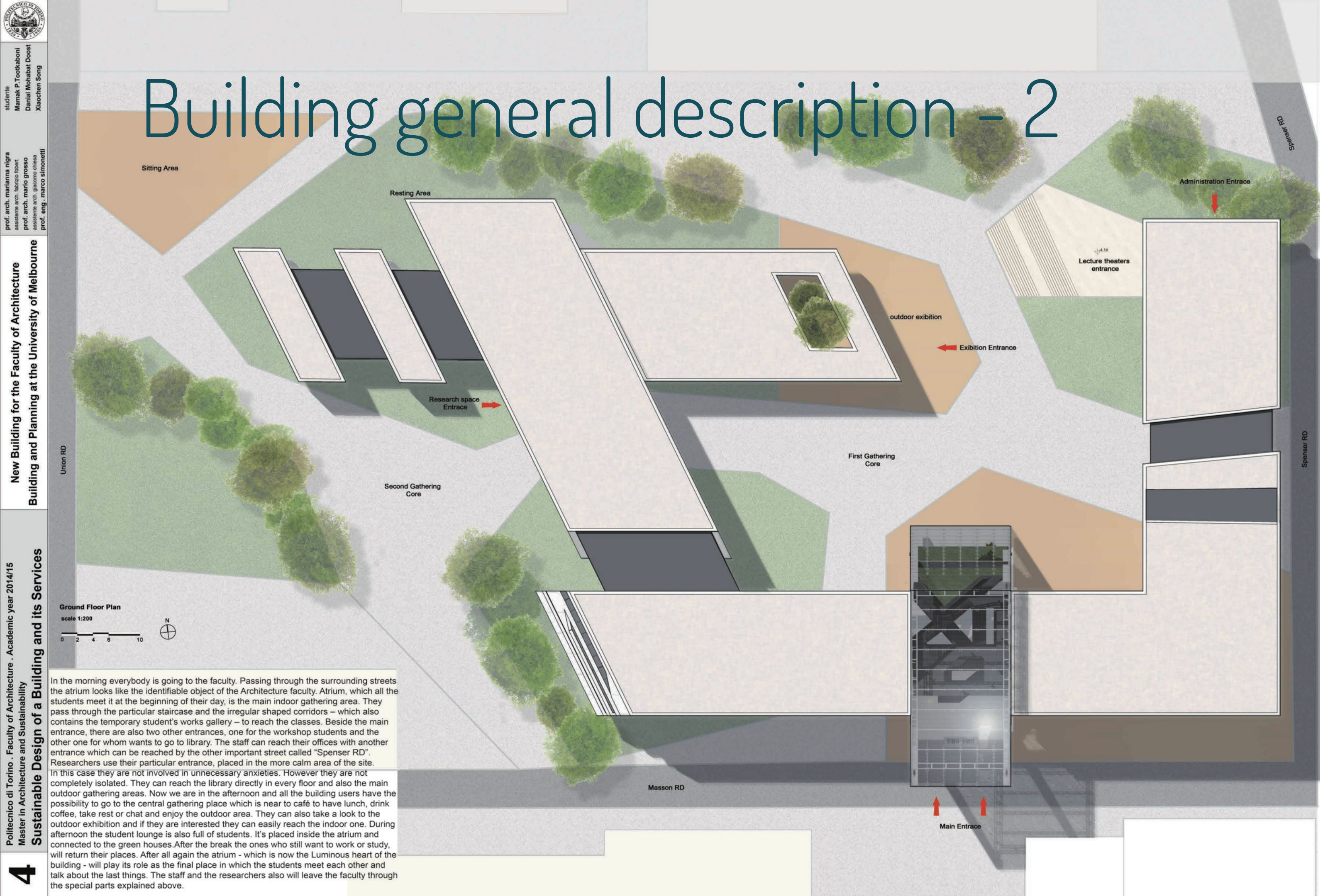
Alessandro Ciociola s220698

OUTLINE

- Building general description
- Software general description
- Specific modules
  - Shading
  - Lighting
  - Air quality

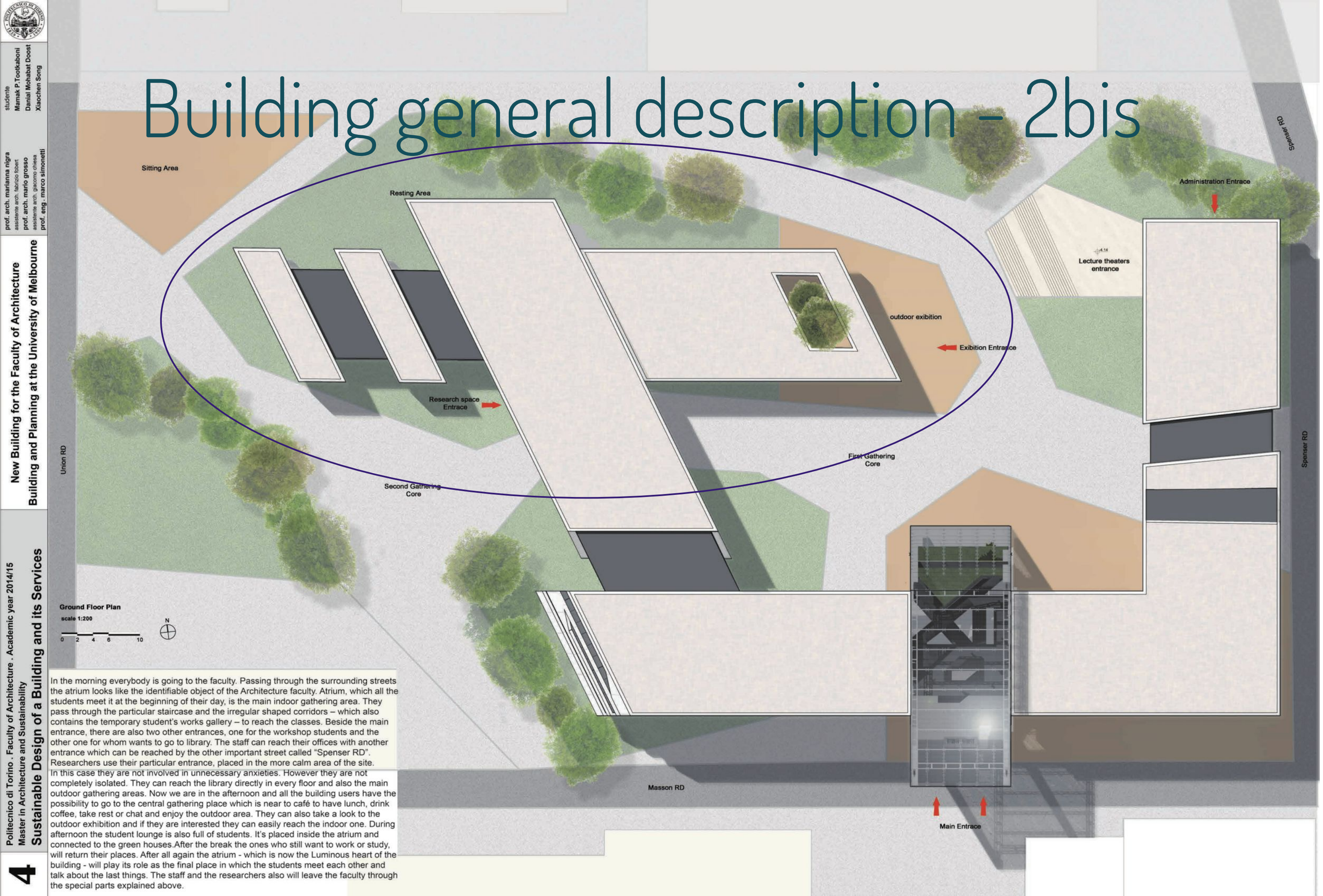
Building general description – 1

Building general description – 2



In the morning everybody is going to the faculty. Passing through the surrounding streets the atrium looks like the identifiable object of the Architecture faculty. Atrium, which all the students meet it at the beginning of their day, is the main indoor gathering area. They pass through the particular staircase and the irregular shaped corridors – which also contains the temporary student's works gallery – to reach the classes. Beside the main entrance, there are also two other entrances, one for the workshop students and the other one for whom wants to go to library. The staff can reach their offices with another entrance which can be reached by the other important street called "Spenser RD". Researchers use their particular entrance, placed in the more calm area of the site. In this case they are not involved in unnecessary anxieties. However they are not completely isolated. They can reach the library directly in every floor and also the main outdoor gathering areas. Now we are in the afternoon and all the building users have the possibility to go to the central gathering place which is near to café to have lunch, drink coffee, take rest or chat and enjoy the outdoor area. They can also take a look to the outdoor exhibition and if they are interested they can easily reach the indoor one. During afternoon the student lounge is also full of students. It's placed inside the atrium and connected to the green houses.After the break the ones who still want to work or study, will return their places. After all again the atrium - which is now the Luminous heart of the building - will play its role as the final place in which the students meet each other and talk about the last things. The staff and the researchers also will leave the faculty through the special paths explained above.

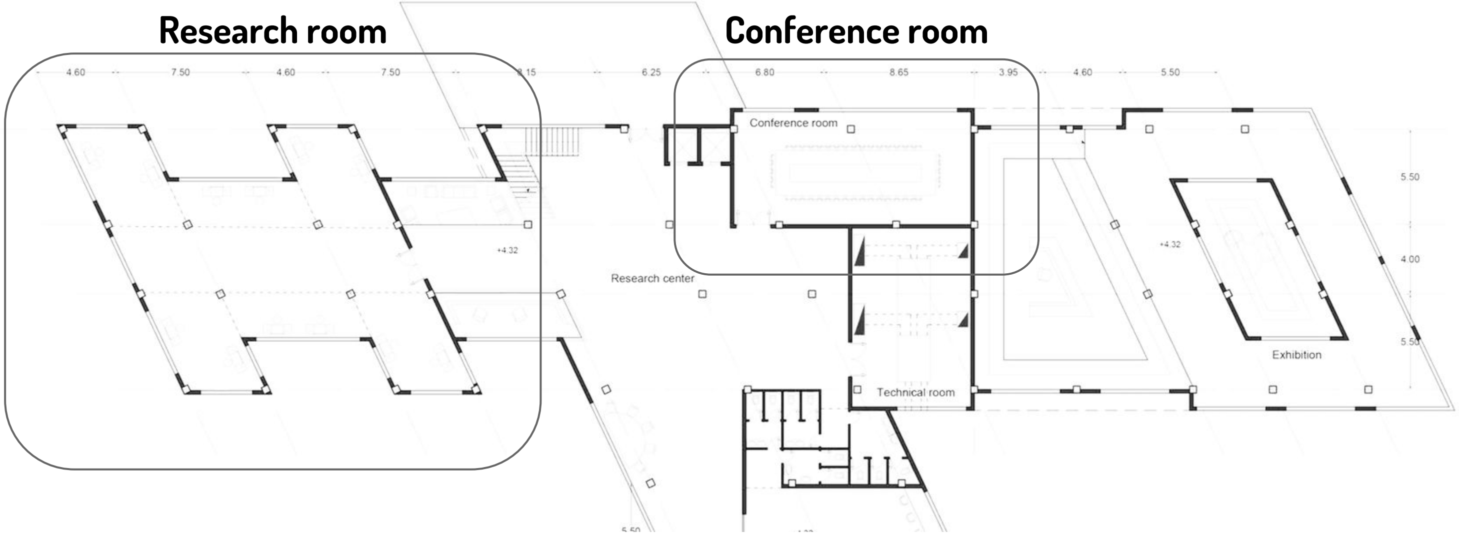
Building general description – 2bis



In the morning everybody is going to the faculty. Passing through the surrounding streets the atrium looks like the identifiable object of the Architecture faculty. Atrium, which all the students meet it at the beginning of their day, is the main indoor gathering area. They pass through the particular staircase and the irregular shaped corridors – which also contains the temporary student's works gallery – to reach the classes. Beside the main entrance, there are also two other entrances, one for the workshop students and the other one for whom wants to go to library. The staff can reach their offices with another entrance which can be reached by the other important street called "Spenser RD". Researchers use their particular entrance, placed in the more calm area of the site.

In this case they are not involved in unnecessary anxieties. However they are not completely isolated. They can reach the library directly in every floor and also the main outdoor gathering areas. Now we are in the afternoon and all the building users have the possibility to go to the central gathering place which is near to café to have lunch, drink coffee, take rest or chat and enjoy the outdoor area. They can also take a look to the outdoor exhibition and if they are interested they can easily reach the indoor one. During afternoon the student lounge is also full of students. It's placed inside the atrium and connected to the green houses.After the break the ones who still want to work or study, will return their places. After all again the atrium - which is now the Luminous heart of the building - will play its role as the final place in which the students meet each other and talk about the last things. The staff and the researchers also will leave the faculty through the special parts explained above.

Building general description – 3



Software general description – 1

Language, data sources and tools

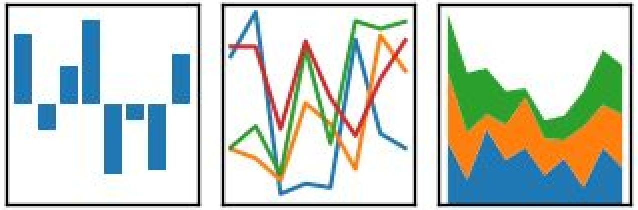
python™

spyder

CITY OF MELBOURNE

pandas

$y_{it} = \beta' x_{it} + \mu_i + \epsilon_{it}$



Software general description – 2

Project Directory

- **Configuration**
  - research_room.py
  - conference_room.py
- **Shading**
  - Design.py
  - requirements.py
- **Lighting**
  - Design.py
  - requirements.py
- **AirQuality**
  - Design.py
  - Requirements.py

- **ControlSimulation.py**
- **(main.py)**

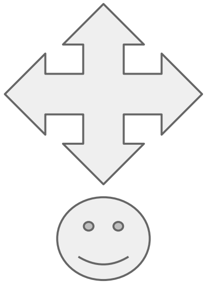
Software general description – 2bis

Project Directory

BUILDING SPECIFIC INFORMATIONS

- **Configuration**
  - **research_room.py**
  - **conference_room.py**
- **Shading**
  - **Design.py**
  - **requirements.py**
- **Lighting**
  - **Design.py**
  - **requirements.py**
- **AirQuality**
  - **Design.py**
  - **requirements.py**

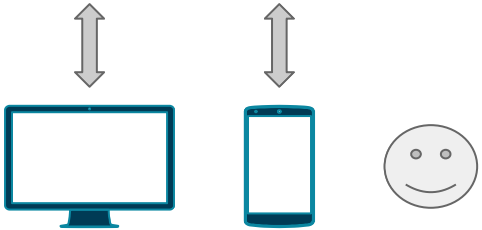
CONTROL MODULES



Developers
Architects
Analysts
(…)

SIMULATION OF CONTROL AND
COMMUNICATION OVER TIME

- **ControlSimulation.py**
- **(main.py)**



Users

Software general description – 2bis

Project Directory

BUILDING SPECIFIC INFORMATIONS

- **Configuration**
  - **research_room.py**
  - **conference_room.py**
- **Shading**
  - **Design.py**
  - **requirements.py**
- **Lighting**
  - **Design.py**
  - **requirements.py**
- **AirQuality**
  - **Design.py**
  - **requirements.py**

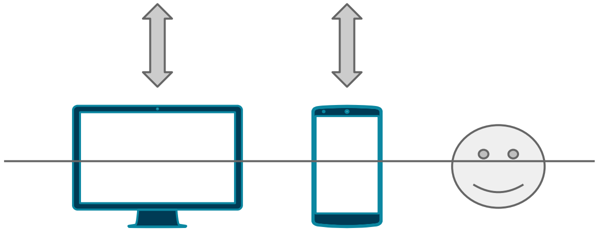
CONTROL MODULES

Developers
Architects
Analysts
(...)

-> <u>DEMO</u> USING THE *SPYDER* IDE AND *PANDAS* DATAFRAMES

SIMULATION OF CONTROL AND ~~COMMUNICATION~~ OVER TIME

- **ControlSimulation.py**
- **(main.py)**



Users

Shading – 1

Goals

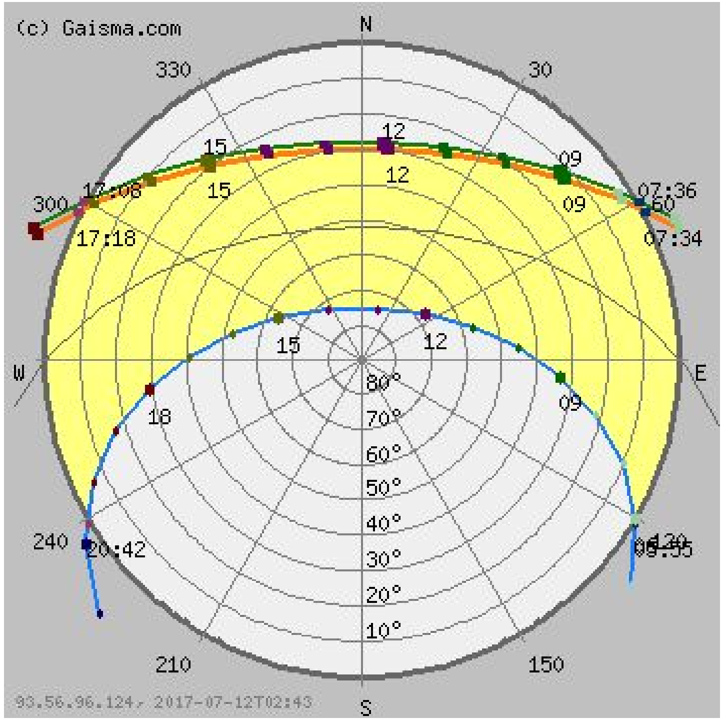
- Satisfy requirements
- Adapt to building specific conditions
- Save energy
- Ensure thermal comfort
- Low cost
- Low power

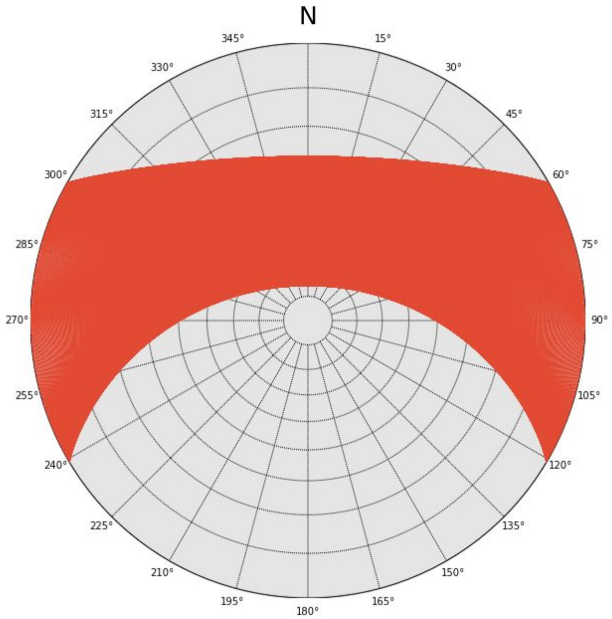
Shading - 2
Methodology

- Get geographical information of building
- Model sunpath
- Size up shading devices
- Define control strategy
- Simulate control strategy

Shading – 3

Melbourne sunpath

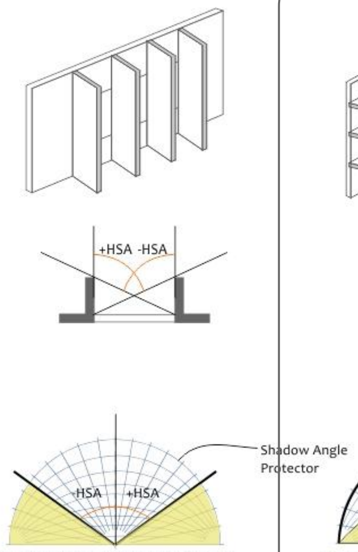




Shading – 4

Shading angles and shading devices

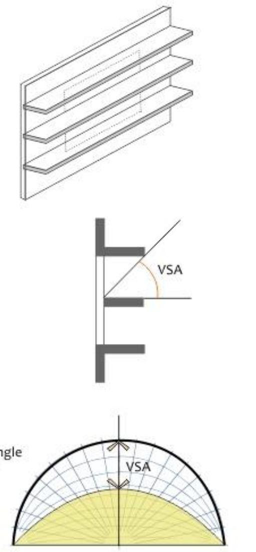
Vertical Shading



Shading mask of vertical shading device

vertical shading devices protect from sun at sides of the elevation such as east and west side

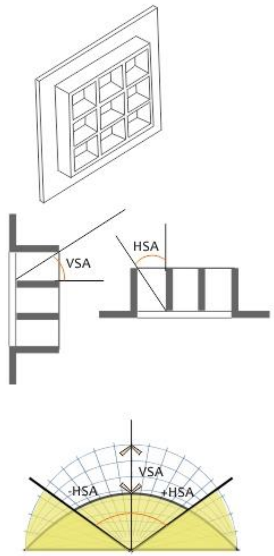
Horizontal Shading



Shading mask of horizontal shading device

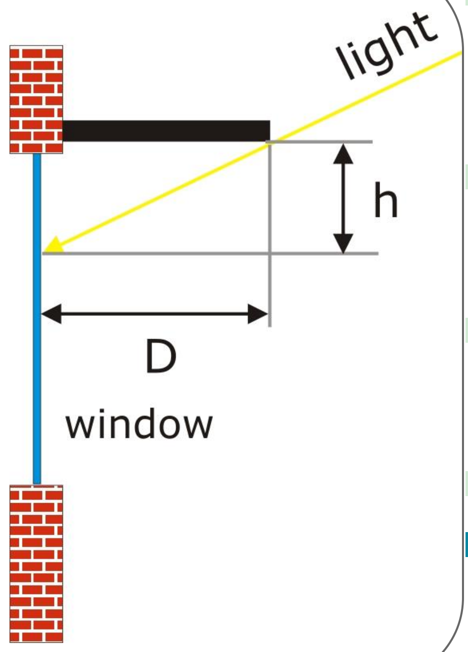
horizontal shading devices protect from sun at high angles and opposite to the wall to be shaded such as north and south sides

Horizontal & Vertical Shading

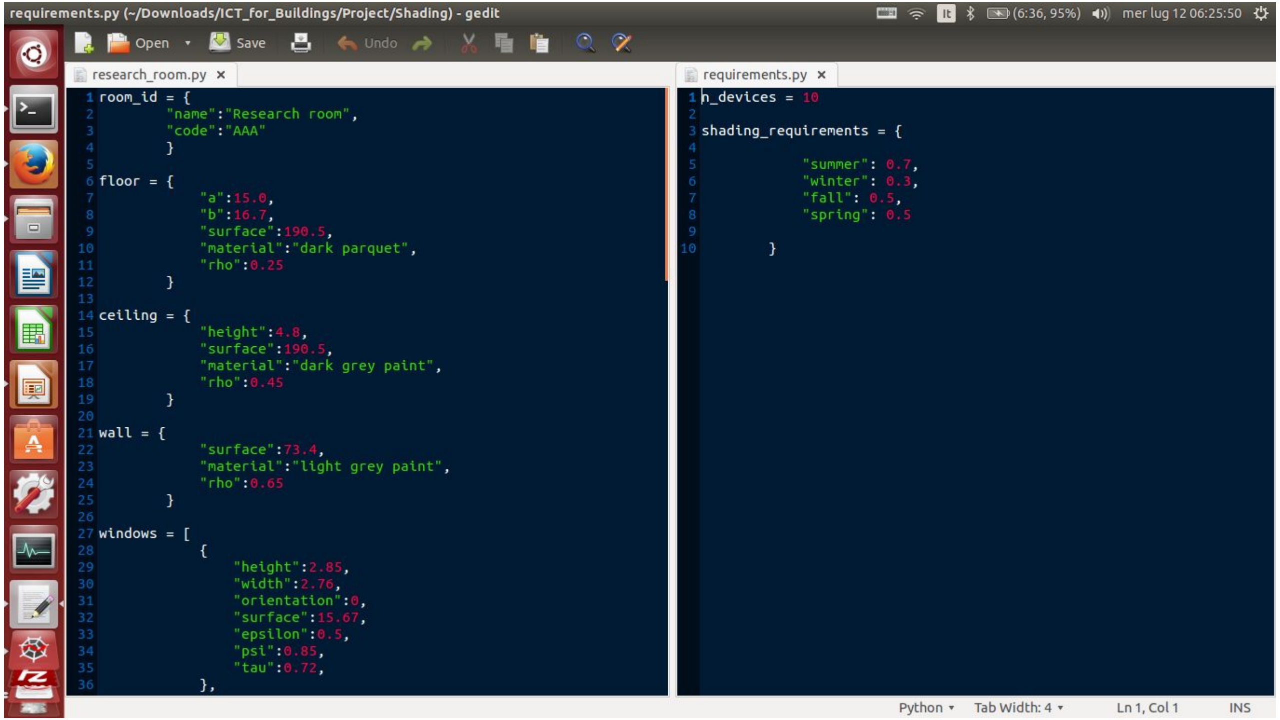


Shading mask of egg crate shading device

comination of horizontal and vertical shading devices protect from sun in all orientations



Shading – 5
Configuration example



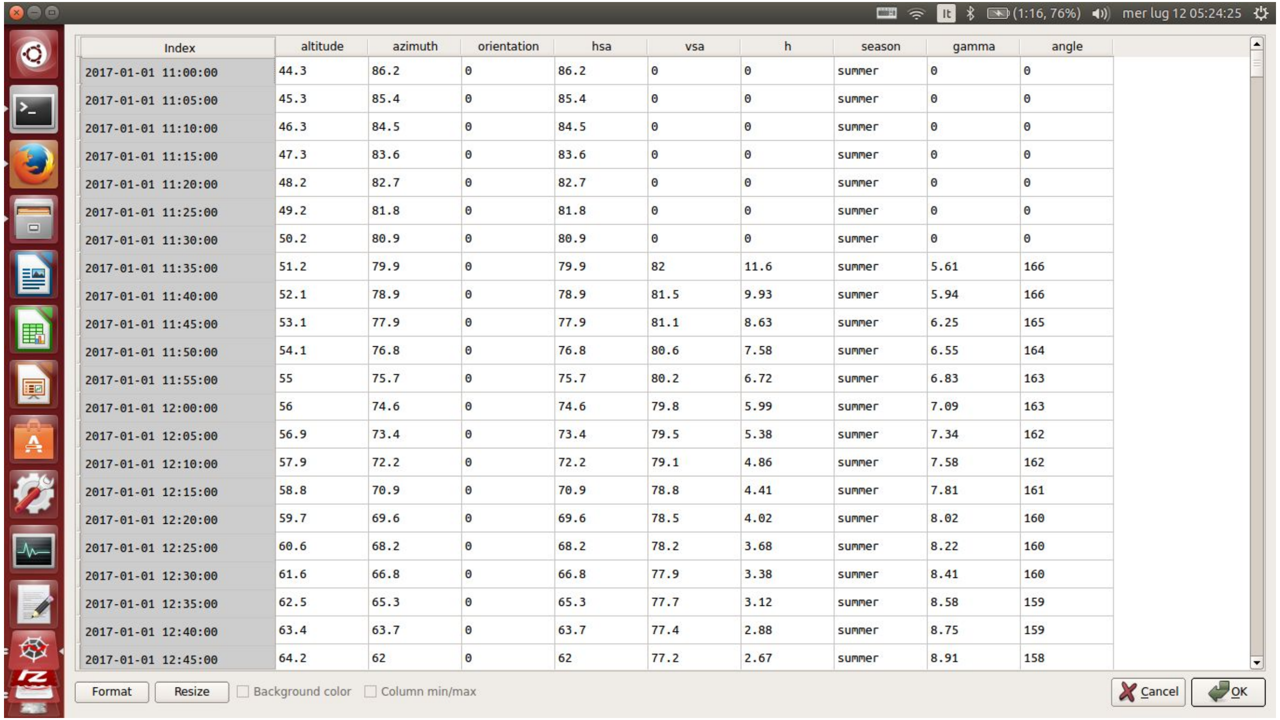
Shading – 5bis

Requirements algorithm

Design.py (~/Downloads/ICT_for_Buildings/Project/Shading) - gedit

```python
        self.df["orientation"] = window["orientation"]
        self.df["hsa"] = self.df["azimuth"] - self.df["orientation"]

        self.shade_needed_index = self.df.loc[(self.df.altitude > 0)\
                                        & (self.df.azimuth > -80)\
                                        & (self.df.azimuth < 80)].index

        self.df["vsa"] = 0.0
        self.df.loc[self.shade_needed_index, "vsa"] = \
                            (self.df["altitude"].apply(np.deg2rad).apply(np.tan)\
                            /self.df["hsa"].apply(np.deg2rad).apply(np.cos))\
                            .apply(np.arctan).apply(np.rad2deg)


        self.df["h"] = 0
        self.df.loc[self.shade_needed_index, "h"] = \
                    (self.device_depth * self.df["vsa"].apply(np.deg2rad).apply(np.tan))\
                    / (self.df["hsa"].apply(np.deg2rad).apply(np.cos))

        self.df["season"] = ""
        self.df.loc["2017-1-1":"2017-3-21", "season"] = "summer"
        self.df.loc["2017-12-21":"2017-12-31", "season"] = "summer"
        self.df.loc["2017-3-21":"2017-6-21", "season"] = "fall"
        self.df.loc["2017-6-21":"2017-9-21", "season"] = "winter"
        self.df.loc["2017-9-21":"2017-12-21", "season"] = "spring"

        def compute_angle (season, req):
            self.df.loc[self.df.season == season, "gamma"] = \
                        (req * (90.0 - self.df.vsa).apply(np.sin)).apply(np.deg2rad)\
                        .apply(np.arcsin).apply(np.rad2deg)
            self.df.loc[self.df.season == season, "angle"] =\
                       180.0 - (90.0 - self.df.vsa)\
                       - self.df.loc[:, "gamma"]

        for season in ["summer", "winter", "fall", "spring"]:
            compute_angle(season, requirements[season])
```

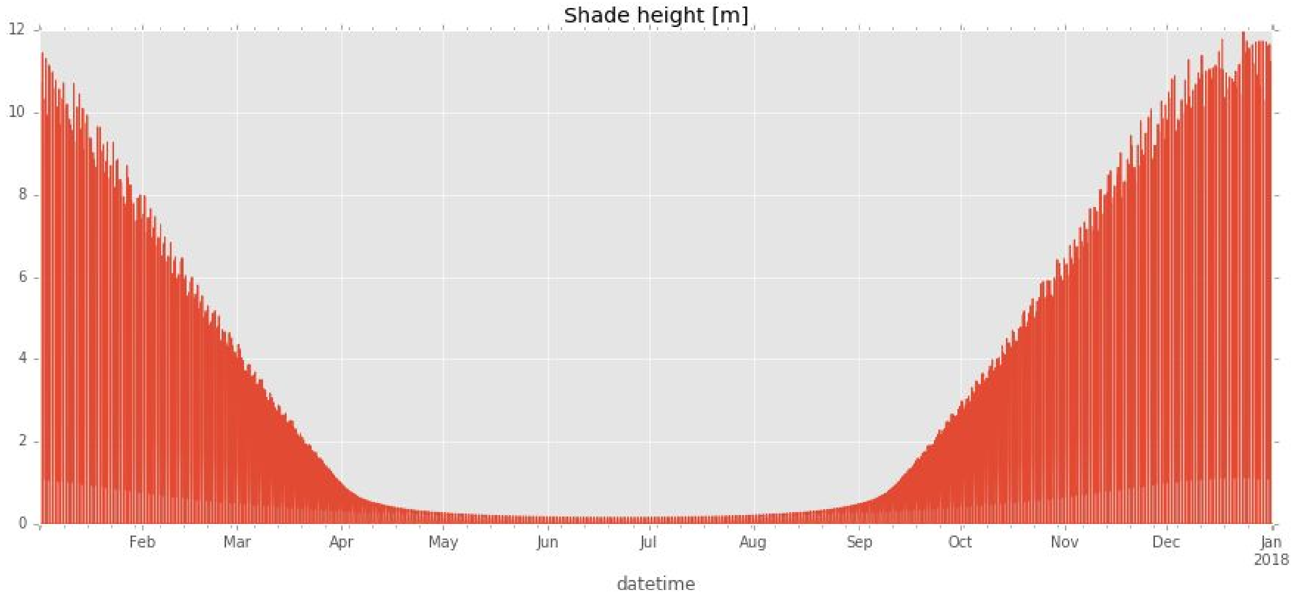
Shading – 6
Dataframe example



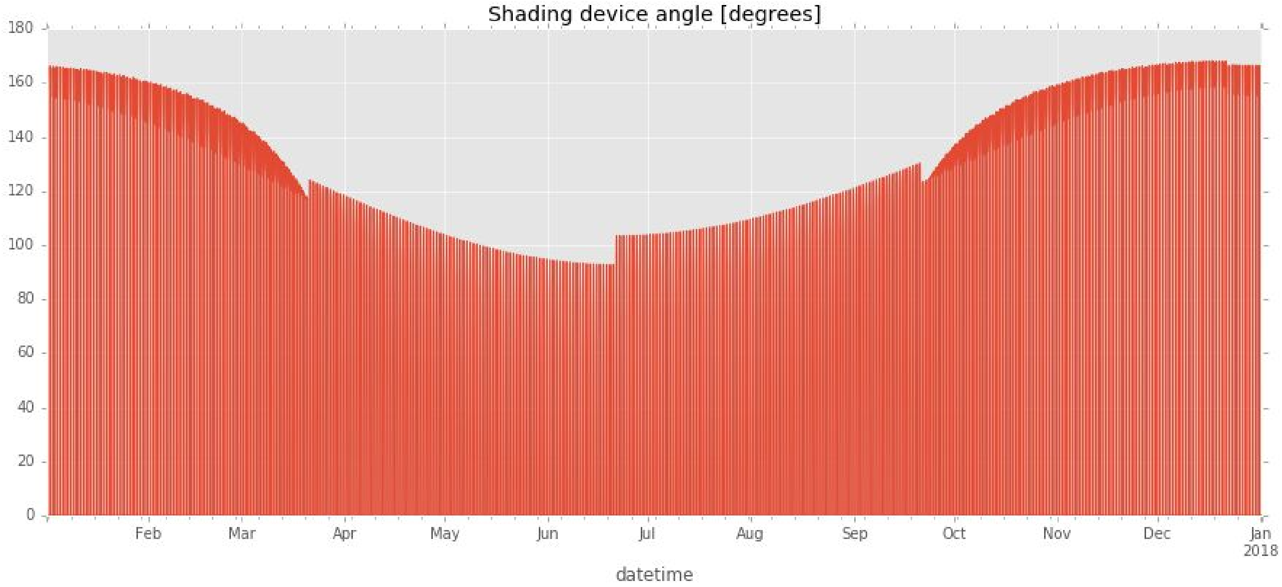
Index	altitude	azimuth	orientation	hsa	vsa	h	season	gamma	angle
2017-01-01 11:00:00	44.3	86.2	0	86.2	0	0	summer	0	0
2017-01-01 11:05:00	45.3	85.4	0	85.4	0	0	summer	0	0
2017-01-01 11:10:00	46.3	84.5	0	84.5	0	0	summer	0	0
2017-01-01 11:15:00	47.3	83.6	0	83.6	0	0	summer	0	0
2017-01-01 11:20:00	48.2	82.7	0	82.7	0	0	summer	0	0
2017-01-01 11:25:00	49.2	81.8	0	81.8	0	0	summer	0	0
2017-01-01 11:30:00	50.2	80.9	0	80.9	0	0	summer	0	0
2017-01-01 11:35:00	51.2	79.9	0	79.9	82	11.6	summer	5.61	166
2017-01-01 11:40:00	52.1	78.9	0	78.9	81.5	9.93	summer	5.94	166
2017-01-01 11:45:00	53.1	77.9	0	77.9	81.1	8.63	summer	6.25	165
2017-01-01 11:50:00	54.1	76.8	0	76.8	80.6	7.58	summer	6.55	164
2017-01-01 11:55:00	55	75.7	0	75.7	80.2	6.72	summer	6.83	163
2017-01-01 12:00:00	56	74.6	0	74.6	79.8	5.99	summer	7.09	163
2017-01-01 12:05:00	56.9	73.4	0	73.4	79.5	5.38	summer	7.34	162
2017-01-01 12:10:00	57.9	72.2	0	72.2	79.1	4.86	summer	7.58	162
2017-01-01 12:15:00	58.8	70.9	0	70.9	78.8	4.41	summer	7.81	161
2017-01-01 12:20:00	59.7	69.6	0	69.6	78.5	4.02	summer	8.02	160
2017-01-01 12:25:00	60.6	68.2	0	68.2	78.2	3.68	summer	8.22	160
2017-01-01 12:30:00	61.6	66.8	0	66.8	77.9	3.38	summer	8.41	160
2017-01-01 12:35:00	62.5	65.3	0	65.3	77.7	3.12	summer	8.58	159
2017-01-01 12:40:00	63.4	63.7	0	63.7	77.4	2.88	summer	8.75	159
2017-01-01 12:45:00	64.2	62	0	62	77.2	2.67	summer	8.91	158

Shading – 7

Height of produced shades during the year



Shading – 8
Device angles during the year



Lighting – 1

Goals

- Satisfy requirements
- Adapt to building specific conditions
- Save energy
- Ensure light comfort
- Low cost
- Low power

Lighting – 2

Methodology

- Choose lighting device
- Get space index and total flux
- Compute number of sources
- Define control strategy
- Simulate control strategy

Lighting – 3

Choosen LED lamp

GE
Lighting

WORLDWIDE PARTNER

DATA SHEET

Lumination™ LED Luminaires

Suspended 1200mm x 300mm

IP30 <7 KG CE

Product information

This 1200mm x 300mm luminaire showcases everything that's innovative and exciting about Lumination™. Suspended from the ceiling, this strikingly beautiful luminaire combines brilliant aesthetics with space-filling light, all contained within a premium-quality aluminium. And when it's switched off, clear lens means there's no visible light source.

Features & Benefits

LED Technology

- Long life (50,000hrs @ L85)
- 76lm/W delivered at 4000K
- Fully dimmable as standard (1-10V or DALI)
- Rohs Compliant, mercury free

Uniform illuminated surface

- A Choice of Design Aesthetic and Superior performances
- A Uniform illuminating surface: no led "dots"
- Enhanced visual performance
- Enhanced emotional light experience
- Direct and Indirect Light

Green solutions

- A sustainable green choice for Energy Saving
- Durable and Reliable and Superior performances

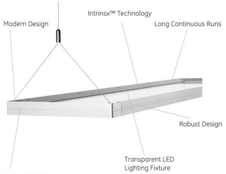
Compliant to photo biological safety standard

No IR or UV radiation

Ease of install & maintenance

Applications

- General Lighting
- Office, Retail, Healthcare, Education



Advanced design provides efficient functional downlighting and ambient uplighting.

GE imagination at work

Basic data

Product Code	Product Description	CCT	Beam Pattern	Dimming Controller	Certification
67646	EP147A1CVSLVR	3000K	Medium	1-10V	CE
67644	EP147A2CVSLVR	3500K	Medium	1-10V	CE
67645	EP147A3CVSLVR	4000K	Medium	1-10V	CE
67649	EP147A1CDSLVR	3000K	Medium	DALI	CE
67647	EP147A2CDSLVR	3500K	Medium	DALI	CE
67648	EP147A3CDSLVR	4000K	Medium	DALI	CE

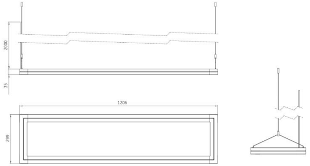
Specifications

Input Voltage	120-277V
Input Power	55W
Input Frequency [Hz]	50/60Hz
Power Factor	> 0.9
Lumen Output [lm]	3900lm (3000K) 4000lm (3500K) 4200lm (4000K)
Efficiency [lm/W]	71 (3000K) 73 (3500K) 76 (4000K)
CCT	3000, 3500, 4000
Min CRI	80
Control	1-10V or DALI
Life (L85, h)	50,000
Warranty	5 years
Files Available	LM79, LM80, IES

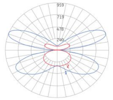
Physical Specifications	
Light Fixture Dimensions	1206mm x 299mm x 35mm
Driver Enclosure Dimensions:	128.1mm x 380mm x 49.8mm
Light Fixture Weight	6.5kg
Driver Enclosure Weight:	2.0kg
Environmental Specifications	
Environmental Operating Temperature Range	-10C to +25C
Environmental Humidity (non-condensing)	20 to 80% Non-condensing, dry & damp location rated
Environmental Storage Temperature Range	-40C to +60C
IP Rating	IP30

GE Lighting reserves the right to amend the technical data and the drawings.

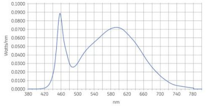
Dimensions [mm]



Light Output



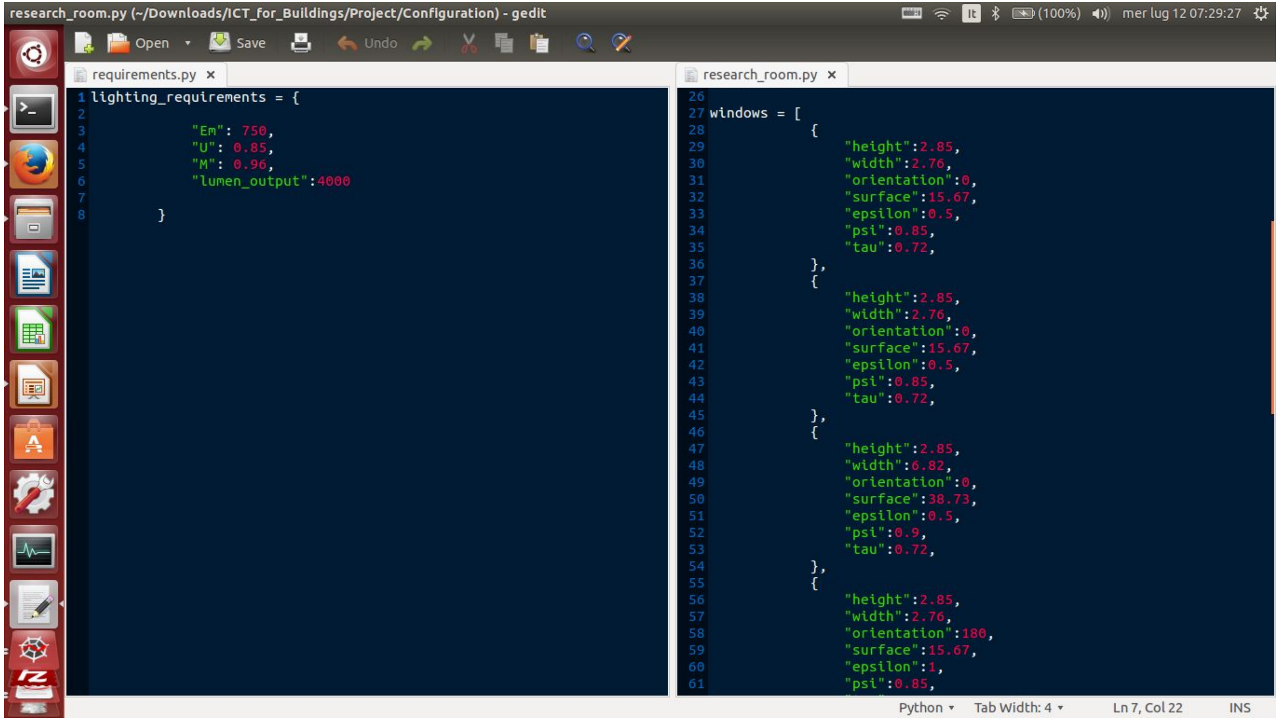
Spectral Distribution



www.gelighting.com/eu

Lumination® LED Luminaires 1200x300mm Suspended Data Sheet – October, 2012

Lighting – 4
Requirements and configuration



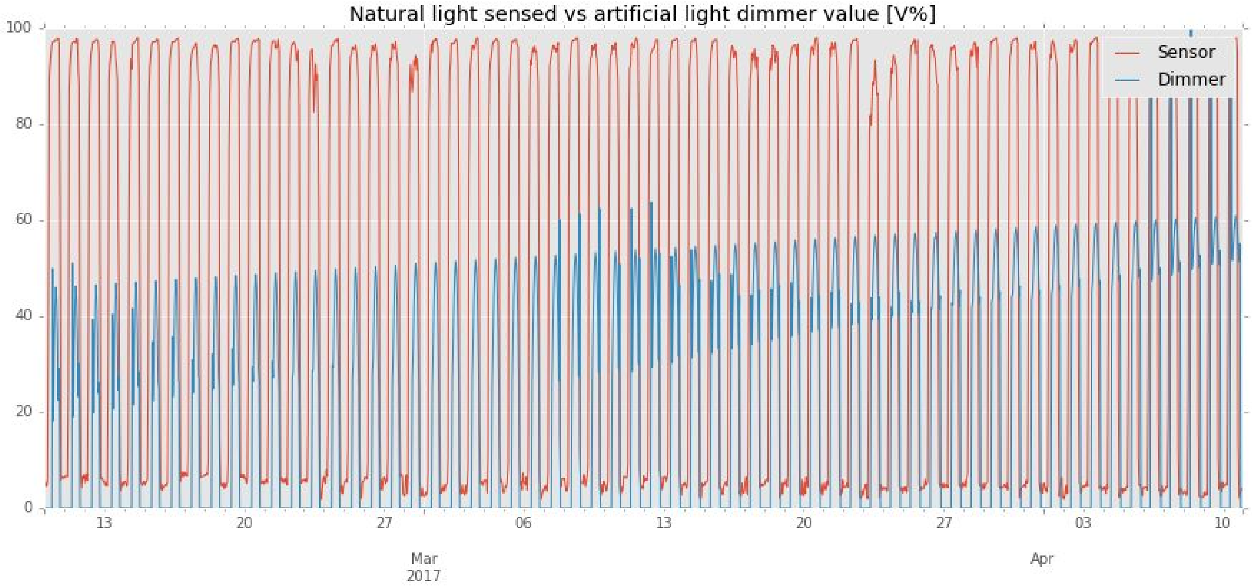
Lighting – 5

Requirements algorithm

Design.py (~/Downloads/ICT_for_Buildings/Project/Lighting) - gedit

```python
import sys
sys.path.append("/home/alecioc/Downloads/ICT_for_Buildings/Project")
sys.path.append("/home/alecioc/Downloads/ICT_for_Buildings/Project/Configuration")

import datetime
import pandas as pd

import matplotlib.pylab as plt
plt.style.use('ggplot')

from requirements import lighting_requirements

from research_room import floor
from research_room import ceiling
from research_room import wall
from research_room import windows
from research_room import desk

rho_m = (floor["surface"] * floor["rho"]\
 + ceiling["surface"] * ceiling["rho"]\
 + wall["surface"] * wall["rho"])\
 / (floor["surface"] + ceiling["surface"] + wall["surface"])


for w in windows:
    w["eta"] = w["height"] * w["width"] * w["psi"] * w["epsilon"] * w["tau"]
eta_m = pd.Series([w["eta"] for w in windows]).sum()\
/ ((1.0 - rho_m) * (floor["surface"] + ceiling["surface"] + wall["surface"]))

space_index = (floor["a"]*floor["b"])\
               / ((floor["a"]+floor["b"])*(ceiling["height"] - desk["height"]))


total_flux = (lighting_requirements["Em"] * floor["surface"])\
              / (lighting_requirements["U"] * lighting_requirements["M"])
n_sources = total_flux / lighting_requirements["lumen_output"]
```

Lighting – 6
Control simulation results



Air Quality – 1

Goals

- Satisfy requirements
- Adapt to building specific conditions
- Save energy
- Limit health risk
- Low cost
- Low power

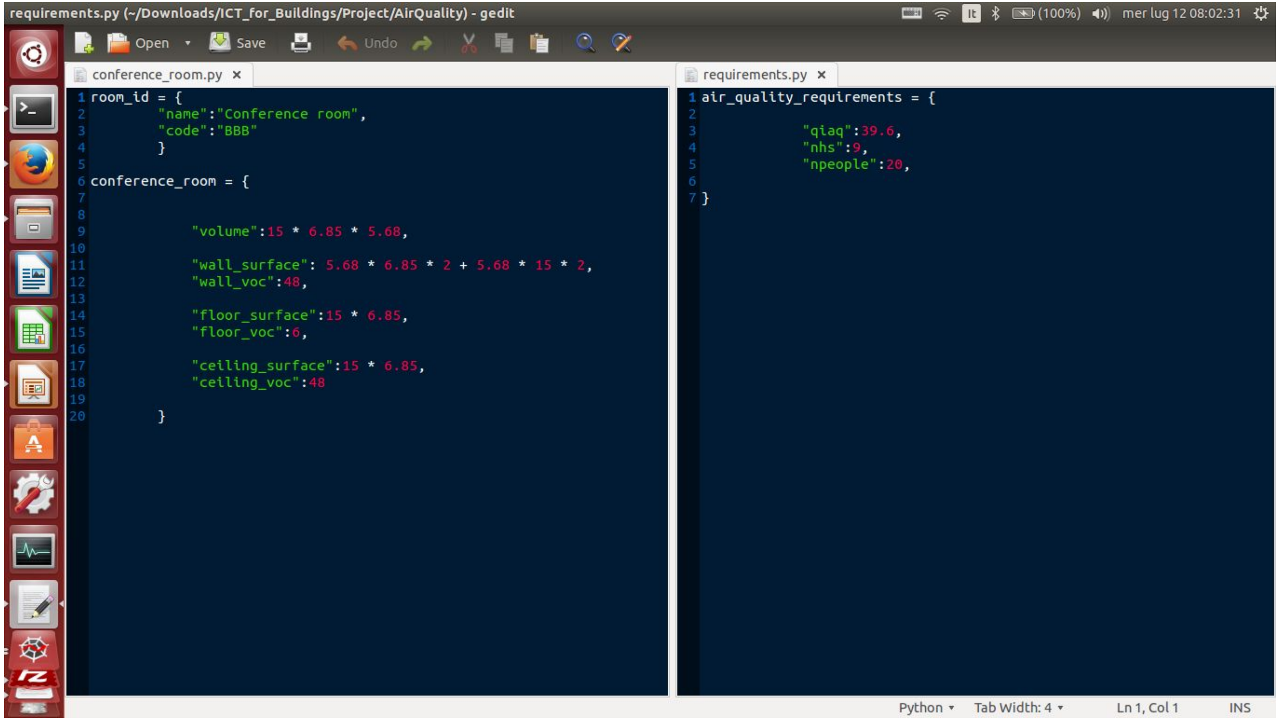
Air Quality - 2

Methodology

- Choose materials for room
- Compute room occupation factors
- Define control strategy
- Simulate control strategy

Air Quality – 3

Requirements and configuration



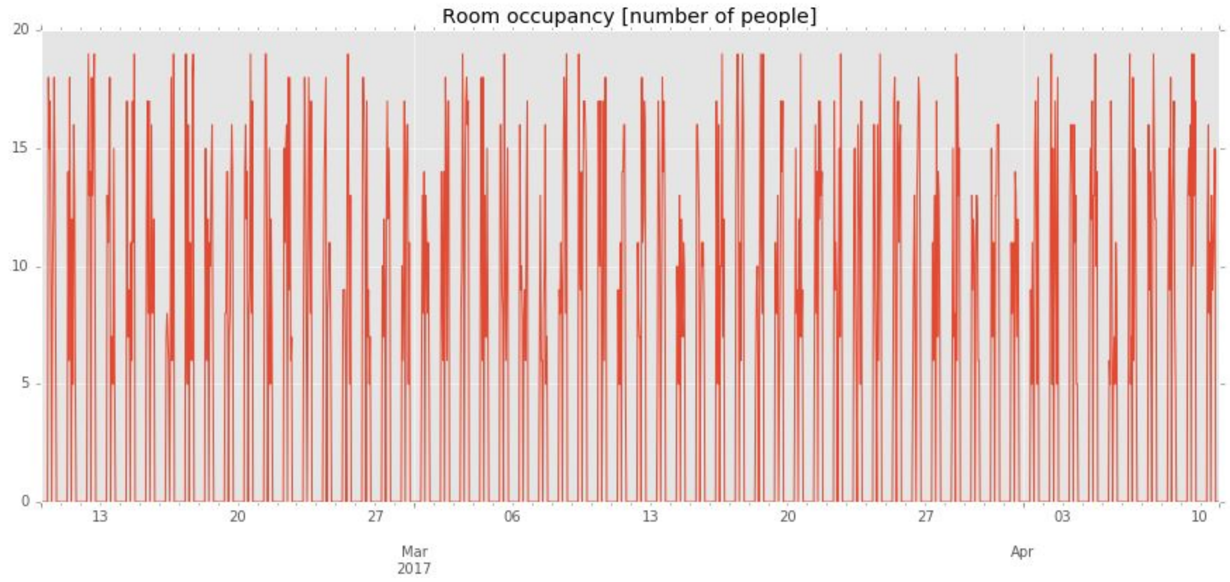
Air Quality – 4

Requirements algorithm

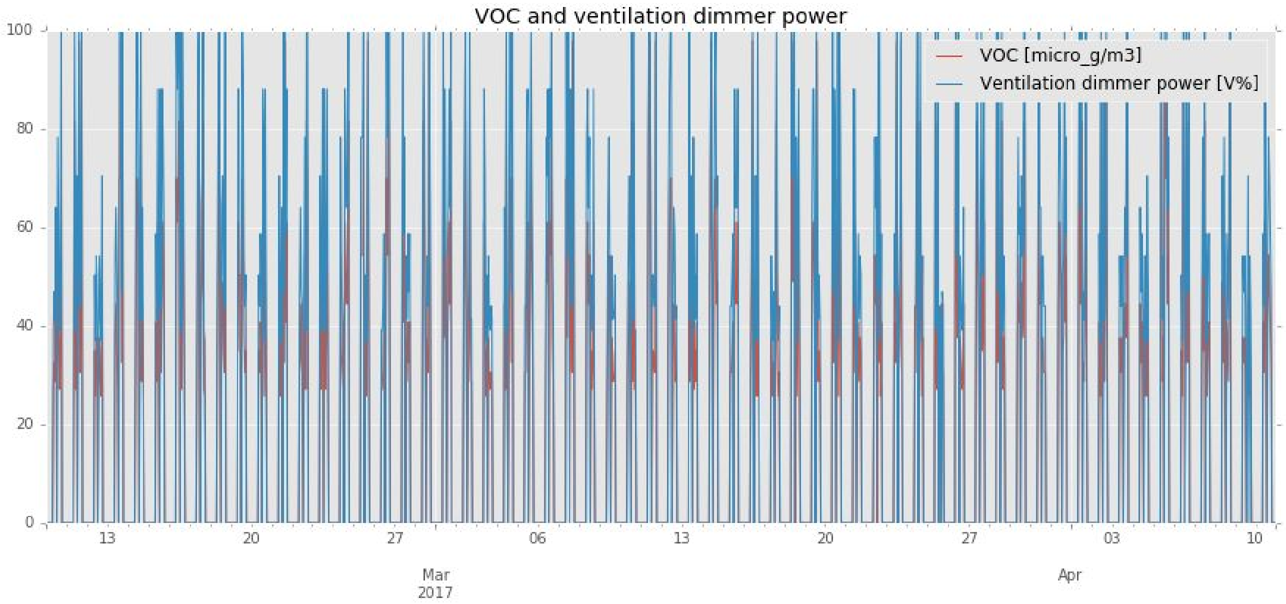
Design.py (~/Downloads/ICT_for_Buildings/Project/AirQuality) - gedit

```python
                .drop(["boardid", "boardtype", "elevation", "Position"], axis=1)
df["timestamp"] = df.index.values

df["occupancy"] = pd.Series(np.random.randint(5,20,size=len(df)), index=df.index)
df.loc[df.timestamp.apply(lambda d:d.hour) < 9, "occupancy"] = np.NaN
df.loc[df.timestamp.apply(lambda d:d.hour) > 18, "occupancy"] = np.NaN
df.loc[df.timestamp.apply(lambda d:d.hour) == 13, "occupancy"] = np.NaN

df["f_inst"] = df["occupancy"]
df["iaq_inst"] = air_quality_requirements["qiaq"] * df["f_inst"] / conference_room["volume"]

df["voc_wall"] = (conference_room["wall_voc"] * (conference_room["wall_surface"]/(0.9*conference_room["volume"])))\
                 / df["iaq_inst"]

df["voc_floor"] = (conference_room["floor_voc"] * (conference_room["floor_surface"]/(0.9*conference_room["volume"])))\
                 / df["iaq_inst"]

df["voc_ceiling"] = (conference_room["ceiling_voc"] * (conference_room["ceiling_surface"]/(0.9*conference_room["volume"])))\
                 / df["iaq_inst"]

df["voc"] = (df["voc_ceiling"] + df["voc_floor"] + df["voc_wall"]).fillna(0)
df["ventilation_power"] = (df.voc.clip(lower=0.0, upper=100.0)).fillna(0)

plt.figure(figsize=(15,6))
plt.title("Room occupancy [number of people]")
df.occupancy.fillna(0).plot(drawstyle="step")
plt.savefig("occupancy.png")

plt.figure(figsize=(15,6))
plt.title("VOC and ventilation dimmer power")
df.voc.fillna(0).plot(drawstyle="step", label="VOC [micro_g/m3]")
df.ventilation_power.fillna(0).plot(drawstyle="step", label="Ventilation dimmer power [V]")
plt.legend()
plt.savefig("ventilation_power.png")

df.loc[:,["timestamp","voc", "ventilation_power"]].to_csv("df.csv")
```

Air Quality – 5
Control simulation results



Air Quality - 5bis
Control simulation results



THANKS!

Any questions?

DEMO

Live demonstration of the implemented software