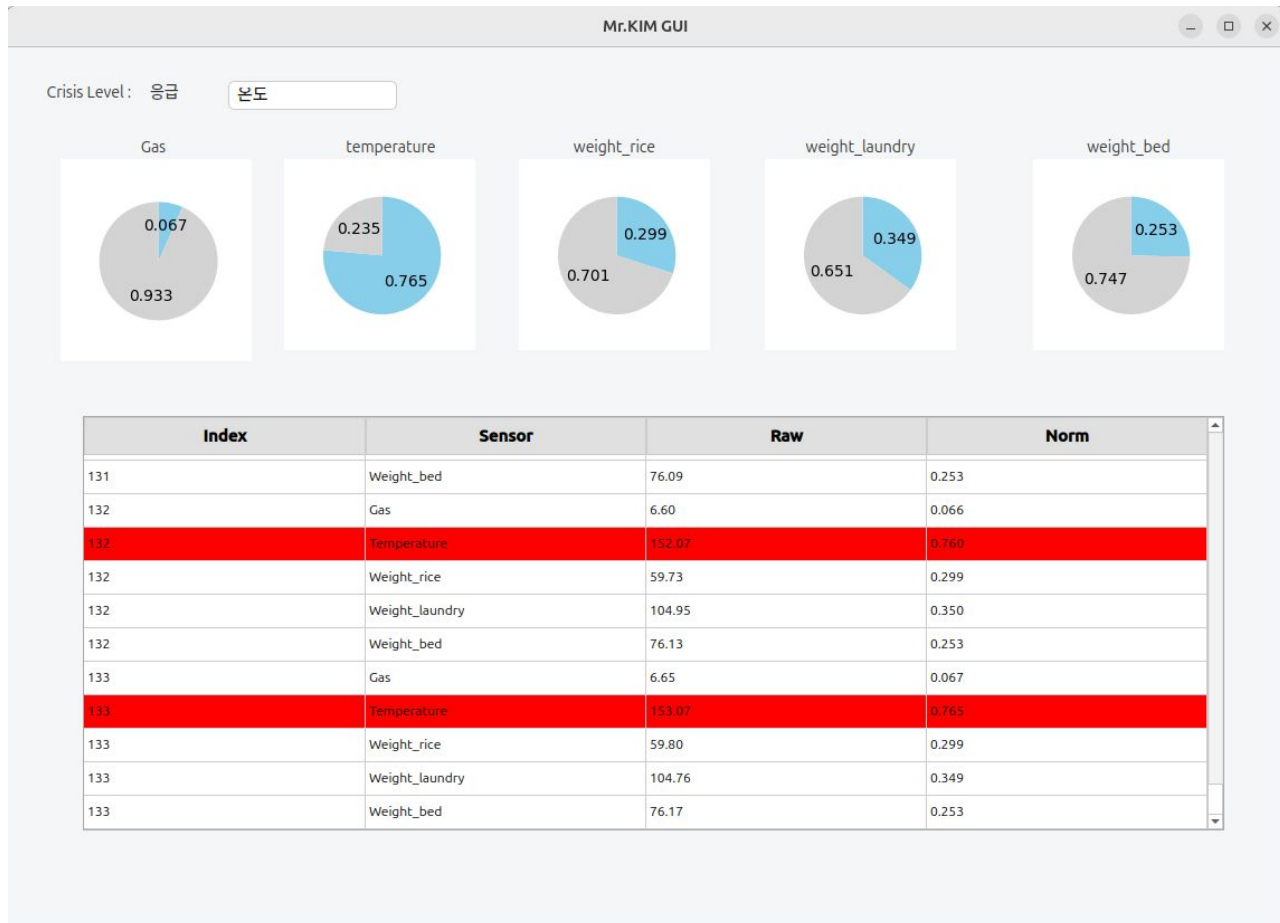


GUI 실행화면(미완)



GUI 코드(미완)

```
import sys
import numpy as np
from PyQt6.QtWidgets import *
from PyQt6.QtCore import QTimer
from PyQt6 import uic
from PyQt6.QtGui import QColor

from matplotlib.backends.backend_qt5agg import FigureCanvasQTAgg as FigureCanvas
from matplotlib.figure import Figure

import warnings
warnings.filterwarnings("ignore", category=UserWarning,
                        message="Ignoring fixed x limits.*")

from class = uic.loadUiType("graph.ui")[0] # UI 파일 경로

class PieCanvas(FigureCanvas):
    def __init__(self, parent=None):
        self.fig = Figure(figsize=(2, 2))
        self.ax = self.fig.add_subplot(111)
        super().__init__(self.fig)
        self.setParent(parent)
        self.draw_pie(0)

    def draw_pie(self, value):
        self.ax.clear()
        self.ax.set_aspect("equal", adjustable="box")

        val = [value, 1 - value]

    def make_autopct(values):
        def my_autopct(pct):
            total = sum(values)
            val = pct * total / 100.0
            return f'{val:.3f}'
        return my_autopct

    self.ax.pie(val,
                colors=["skyblue", "lightgray"],
                startangle=90,
                autopct=make_autopct(val),
                counter-clockwise=True)

    self.draw()

class WindowClass(QMainWindow, from_class):
    def __init__(self, parent=None):
        super().__init__()
        self.setupUi(self)
        self.setWindowTitle("Mr.KIM GUI")

        self.x = np.linspace(0, 200, 2000)
        self.gas = np.linspace(0, 100, len(self.x))
        self.temperature = 20 + 10 * self.x
        self.weight_rice = 50 + 30 * np.sin(self.x / 40)
        self.weight_bed = 100 + 20 * np.cos(self.x / 10)
        self.weight_laundry = 70 + 10 * np.sin(self.x / 20)

        self.table_log.setColumnCount(4)
        self.table_log.setHorizontalHeaderLabels(["Index", "Sensor", "Raw", "Norm"])
        self.table_log.setEditTriggers(QAbstractItemView.EditTrigger.NoEditTrigger)
        self.table_log.verticalHeader().setVisible(False)
        self.table_log.horizontalHeader().setStretchLastSection(True)

        header = self.table_log.horizontalHeader()
        for i in range(4):
            header.setSectionResizeMode(i, QHeaderView.ResizeMode.Stretch)
```

```
for i in range(4):
    header.setSectionResizeMode(i, QHeaderView.ResizeMode.Stretch)

self.sensors = {
    "Gas": {
        "widget": self.widgetPlot_pie_1,
        "value": self.gas,
        "min": 0,
        "max": 100,
        "threshold": 100
    },
    "Temperature": {
        "widget": self.widgetPlot_pie_2,
        "value": self.temperature,
        "min": 0.5,
        "max": 200,
        "threshold": 100
    },
    "Weight_rice": {
        "widget": self.widgetPlot_pie_3,
        "value": self.weight_rice,
        "min": 0,
        "max": 200,
        "threshold": 100
    },
    "Weight_laundry": {
        "widget": self.widgetPlot_pie_4,
        "value": self.weight_bed,
        "min": 0,
        "max": 300,
        "threshold": 100
    },
    "Weight_bed": {
        "widget": self.widgetPlot_pie_5,
        "value": self.weight_laundry,
        "min": 0.2,
        "max": 300,
        "threshold": 100
    }
}

self.sensor_canvases = {}
for name, info in self.sensors.items():
    canvas = PieCanvas(info["widget"])
    layout = QVBoxLayout(info["widget"])
    layout.setContentsMargins(0, 0, 0, 0)
    layout.addWidget(canvas)
    info["canvas"] = canvas # 캔버스를 센서 정보에 저장

self.index = 0

self.timer = QTimer()
self.timer.setInterval(100)
self.timer.timeout.connect(self.update_pies)
self.timer.start()

self.setStyleSheet("""
QMainWindow {
    background-color: #f4f6f8;
}

QLabel {
    font-size: 14px;
    color: #333;
}
```

```
font-size: 14px;
color: #333;

class WindowClass(QMainWindow, from_class):
    color: # def append_log(self, sensor_name, index, raw, norm):
        # highlight = QColor(173, 216, 230) # 연한 파랑
    else:
        highlight = QColor(255, 255, 255)

    item_index.setBackground(highlight)
    item_sensor.setBackground(highlight)
    item_norm.setBackground(highlight)
    item_raw.setBackground(highlight)

    # 표에 추가
    self.table_log.setItem(row, 0, item_index)
    self.table_log.setItem(row, 1, item_sensor)
    self.table_log.setItem(row, 2, item_raw)
    self.table_log.setItem(row, 3, item_norm)

    self.table_log.scrollToBottom()

    def decide_crisis_level(self):
        if True:
            self.label_crlevel.setText("응급")
        elif True:
            self.label_crlevel.setText("주의")
        elif True:
            self.label_crlevel.setText("관심")
        else:
            self.label_crlevel.setText("보통")

        self.show_crisis_contents()

    def show_crisis_contents(self):
        self.line_crcontents.setText("온도")

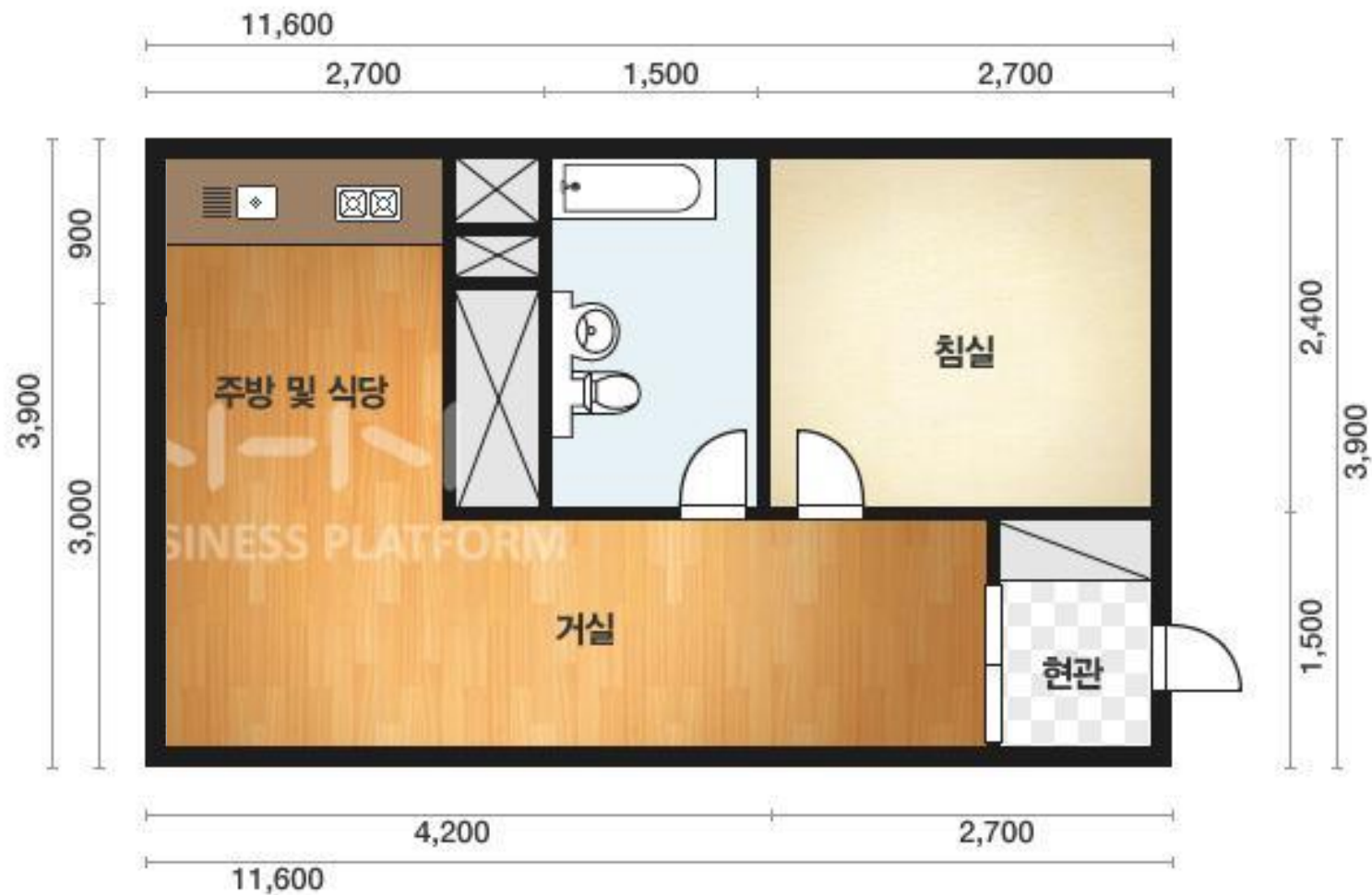
    def normalize(self, val, min_v, max_v):
        if max_v == min_v:
            return 0.0
        return min(max((val - min_v) / (max_v - min_v), 0.0), 2.0)

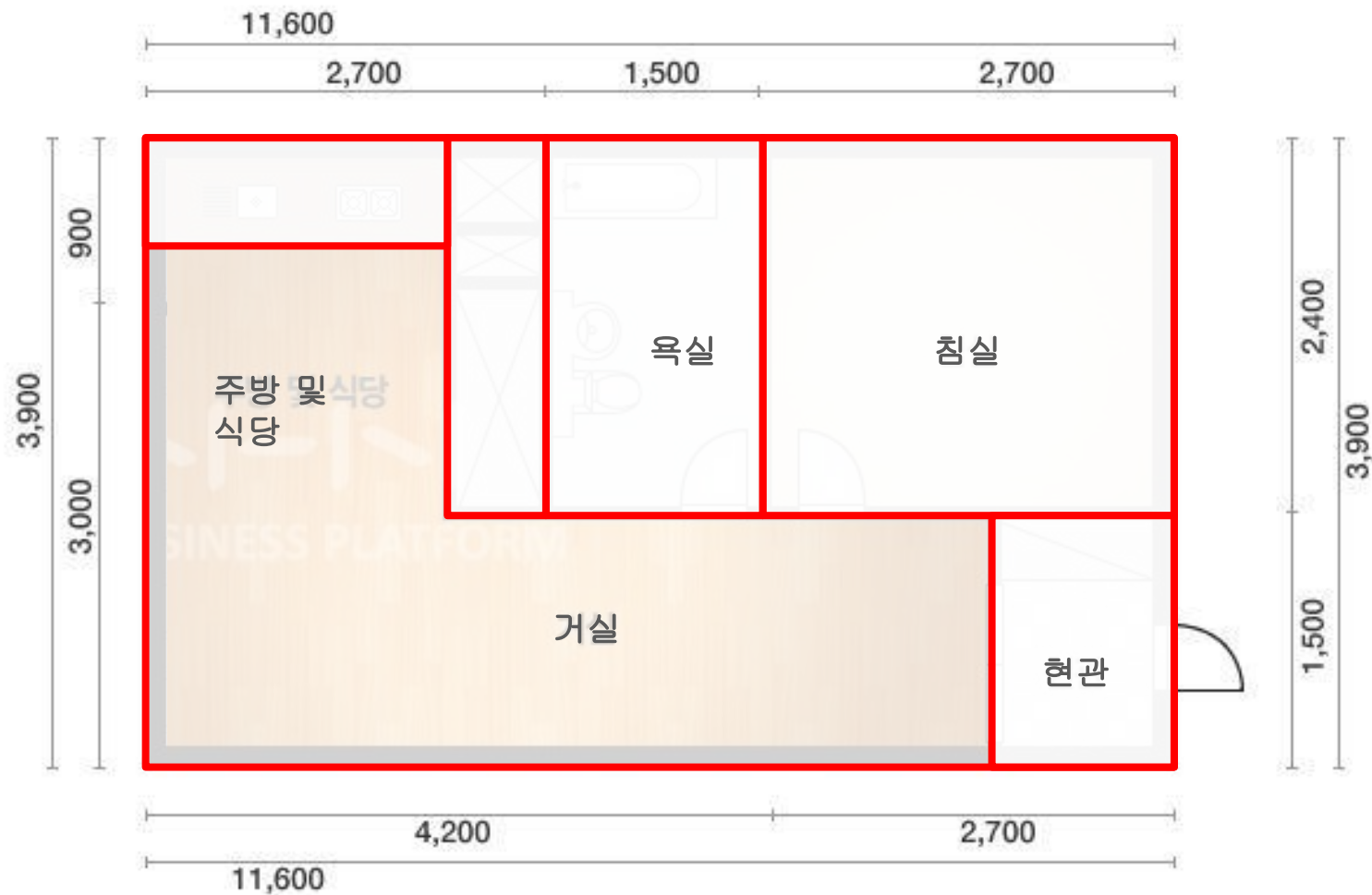
    def update_pies(self):
        for name, info in self.sensors.items():
            data = info["value"]
            if self.index < len(data):
                raw_val = data[self.index]
                norm_val = self.normalize(raw_val, info["min"], info["max"])
                info["canvas"].draw_pie(norm_val)

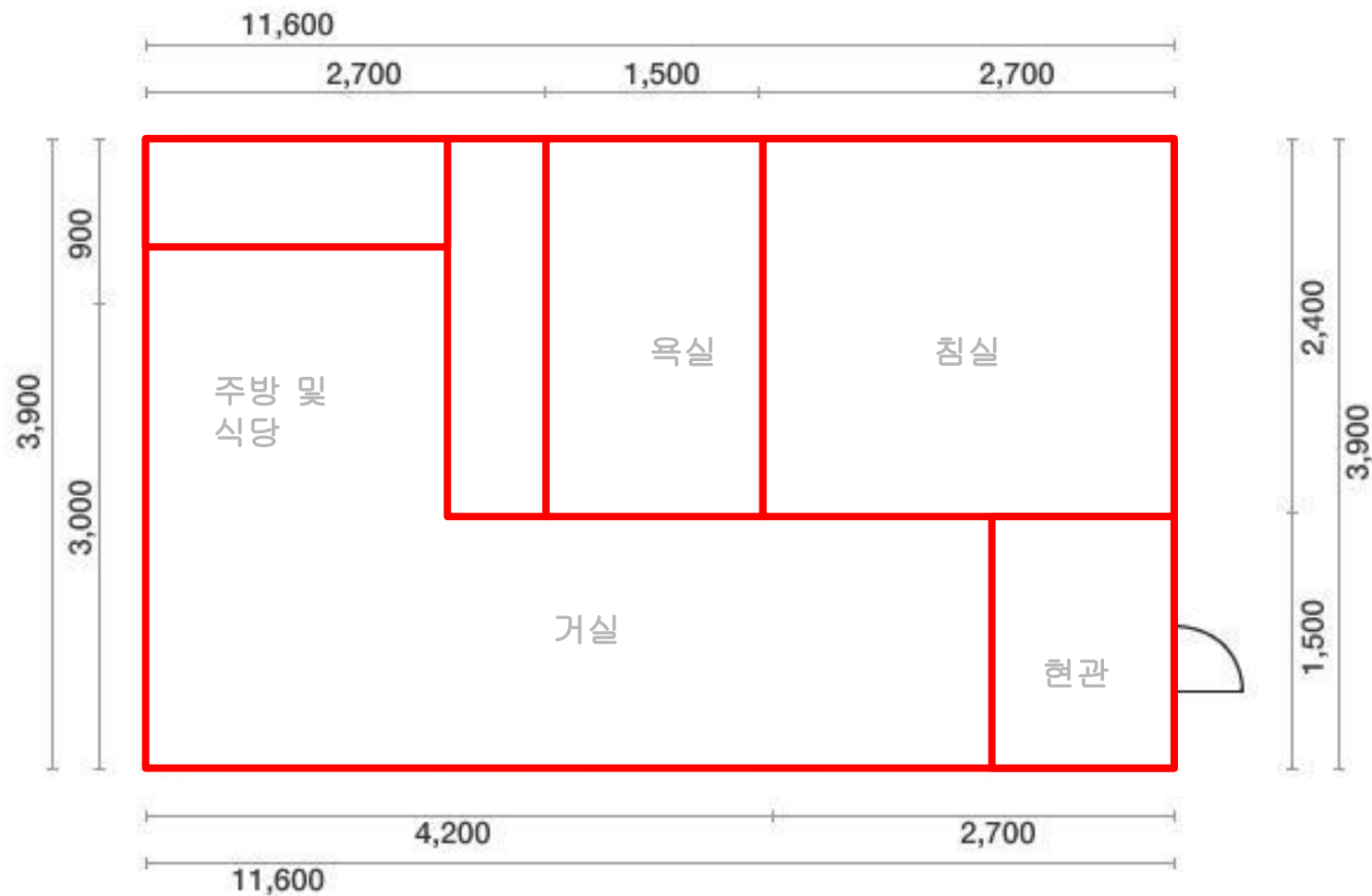
            # level = self.get_crisis_level_text(raw_val)
            self.append_log(name, self.index, raw_val, norm_val)
            self.decide_crisis_level()

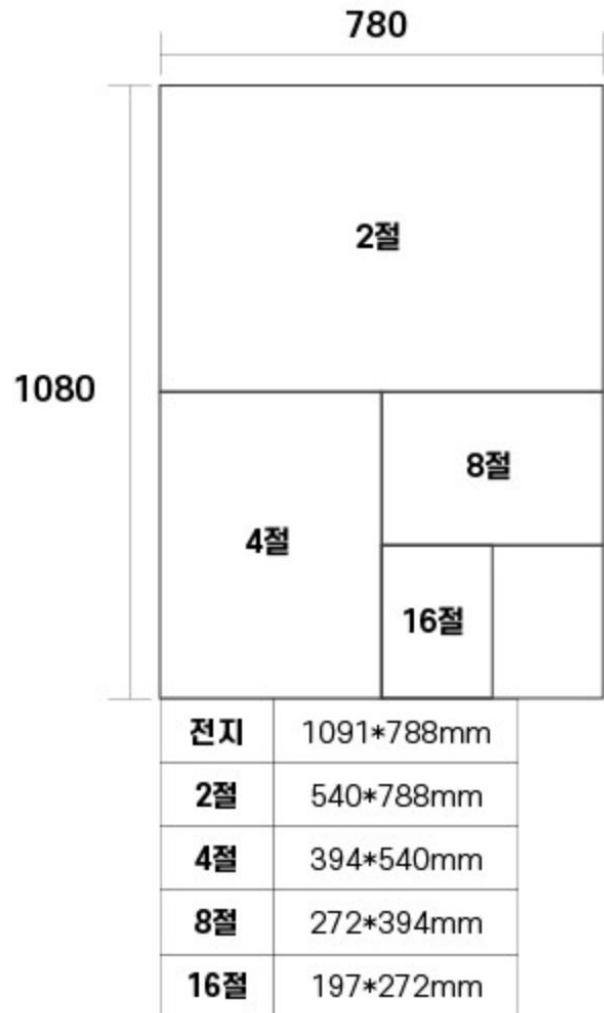
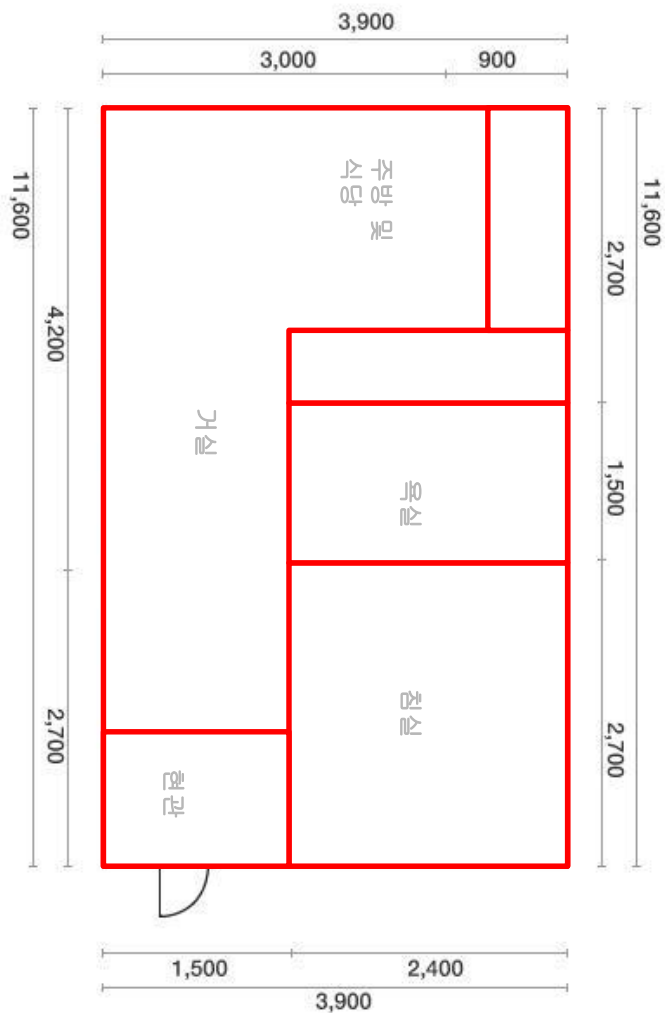
            self.index += 1

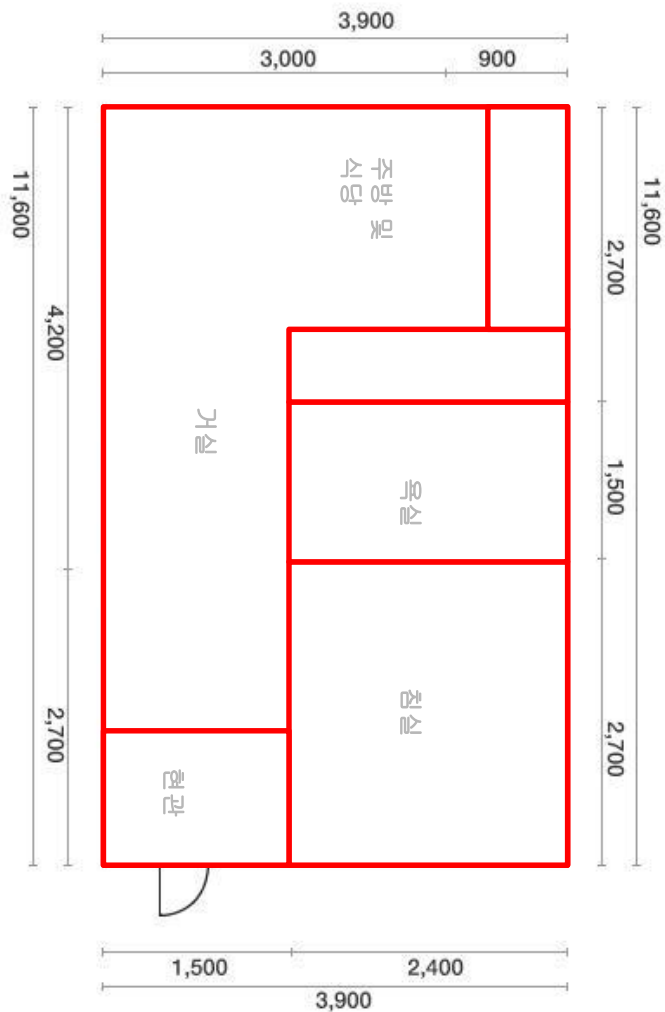
    def __name__ == "__main__":
        app = QApplication(sys.argv)
        myWindow = WindowClass()
        myWindow.show()
        sys.exit(app.exec())
```



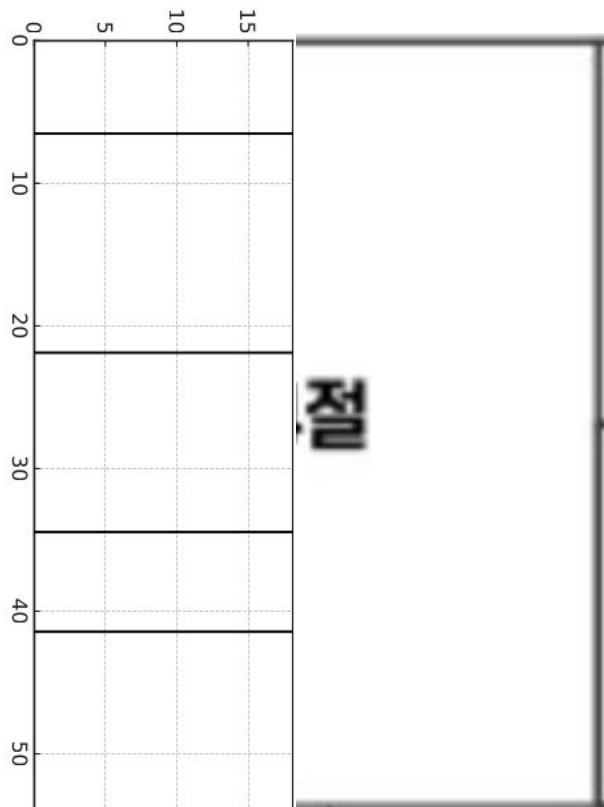
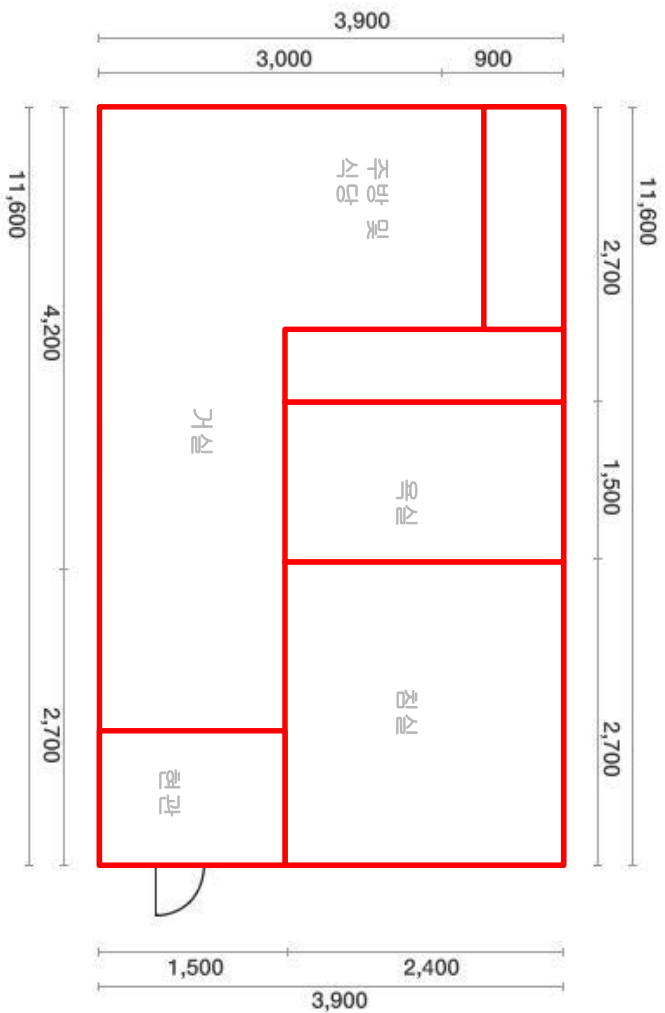




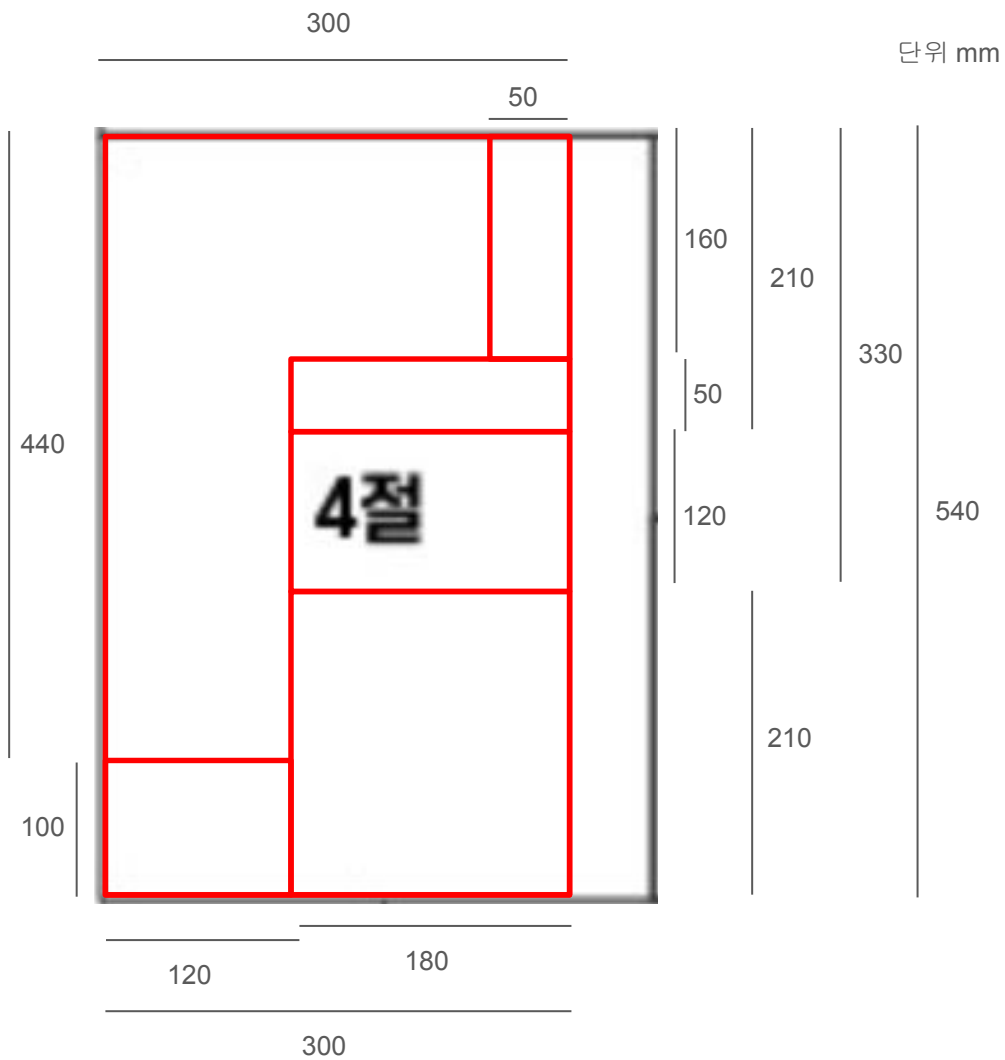
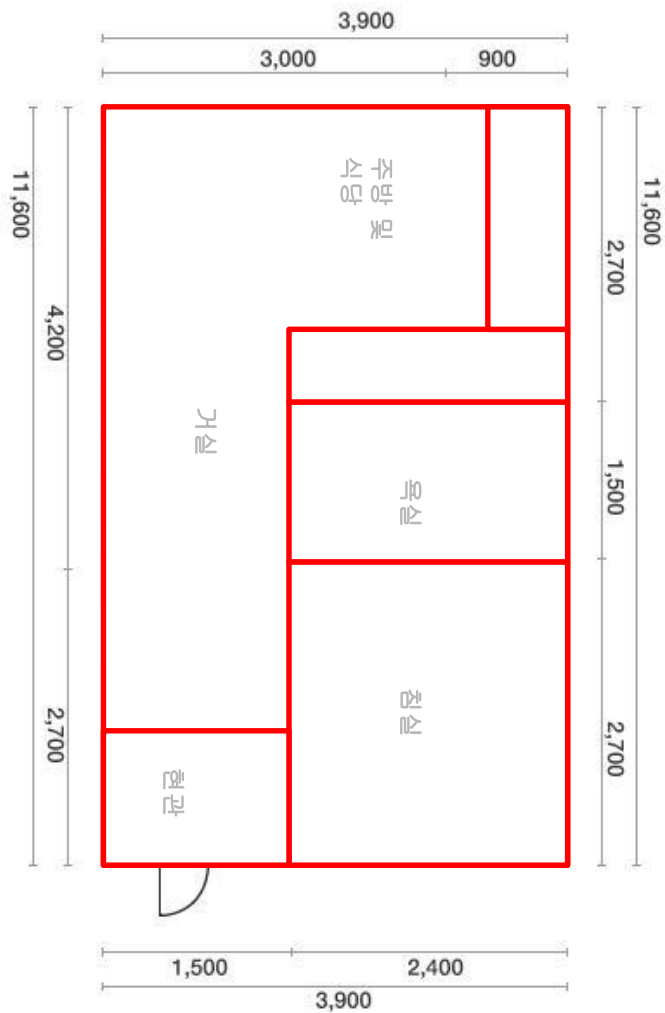


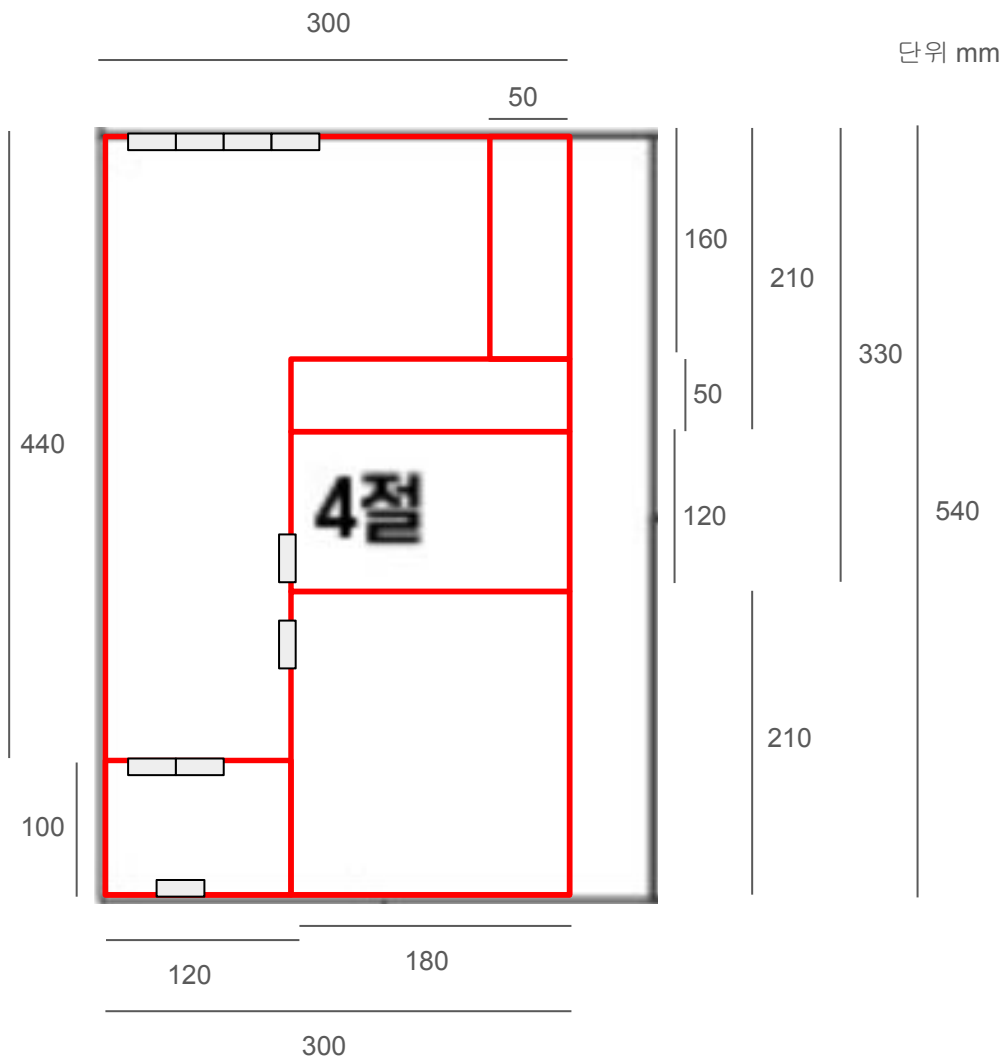
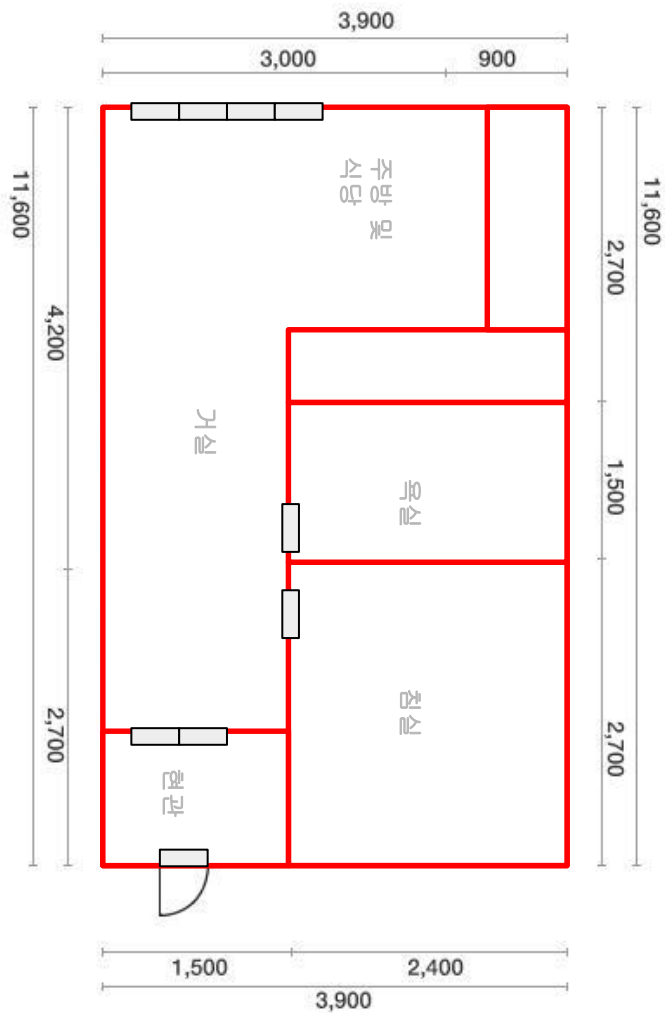


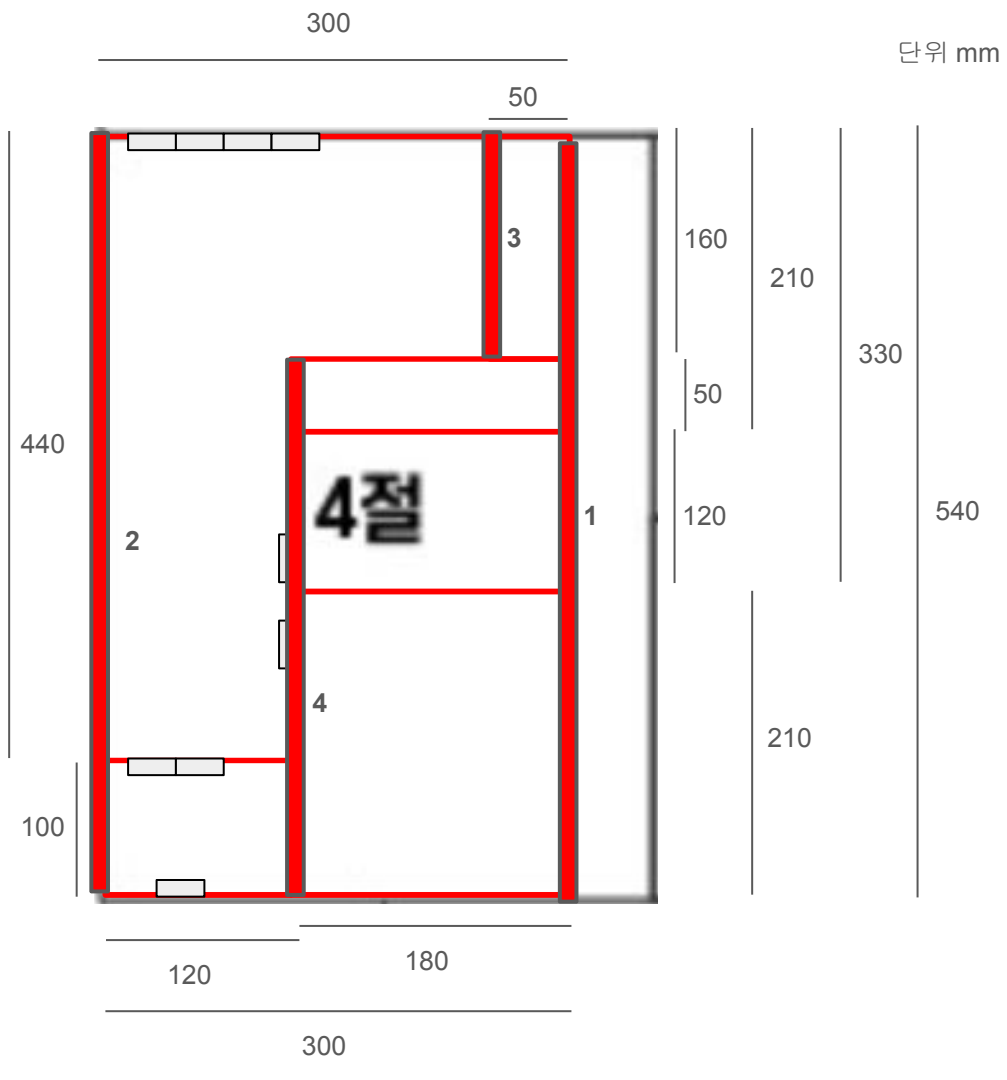
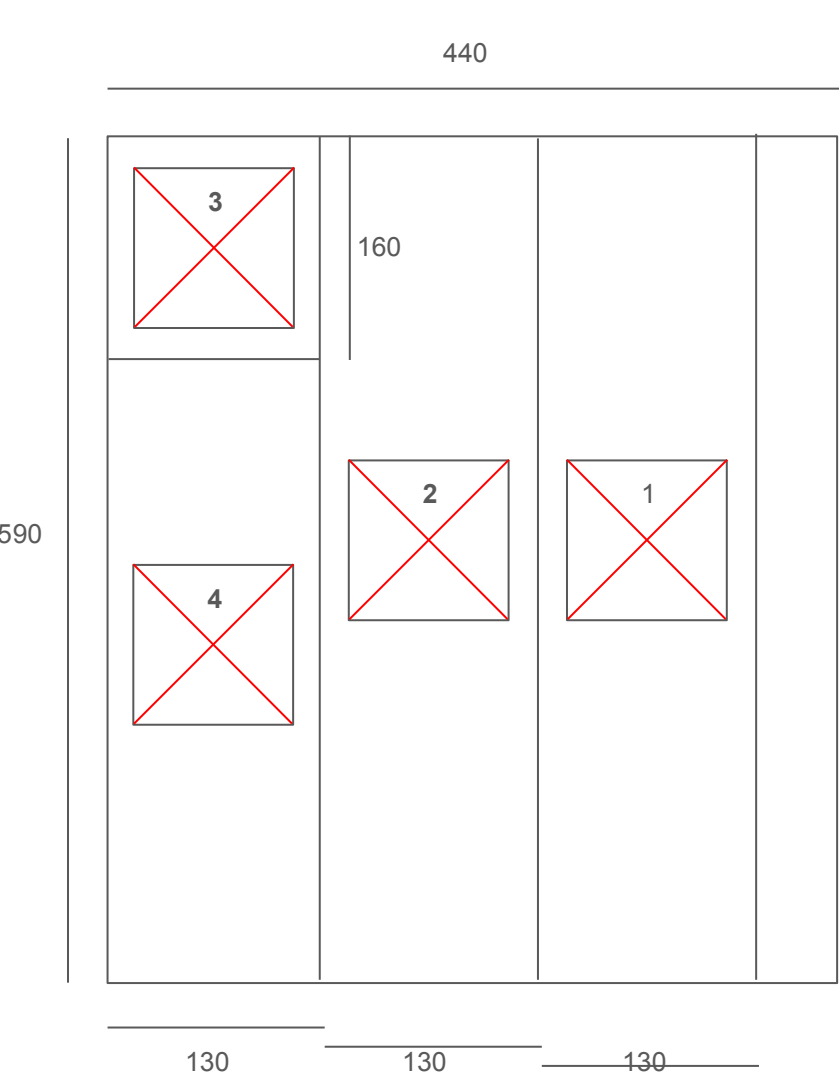
전지	1091*788mm
2절	540*788mm
4절	394*540mm
8절	272*394mm
16절	197*272mm

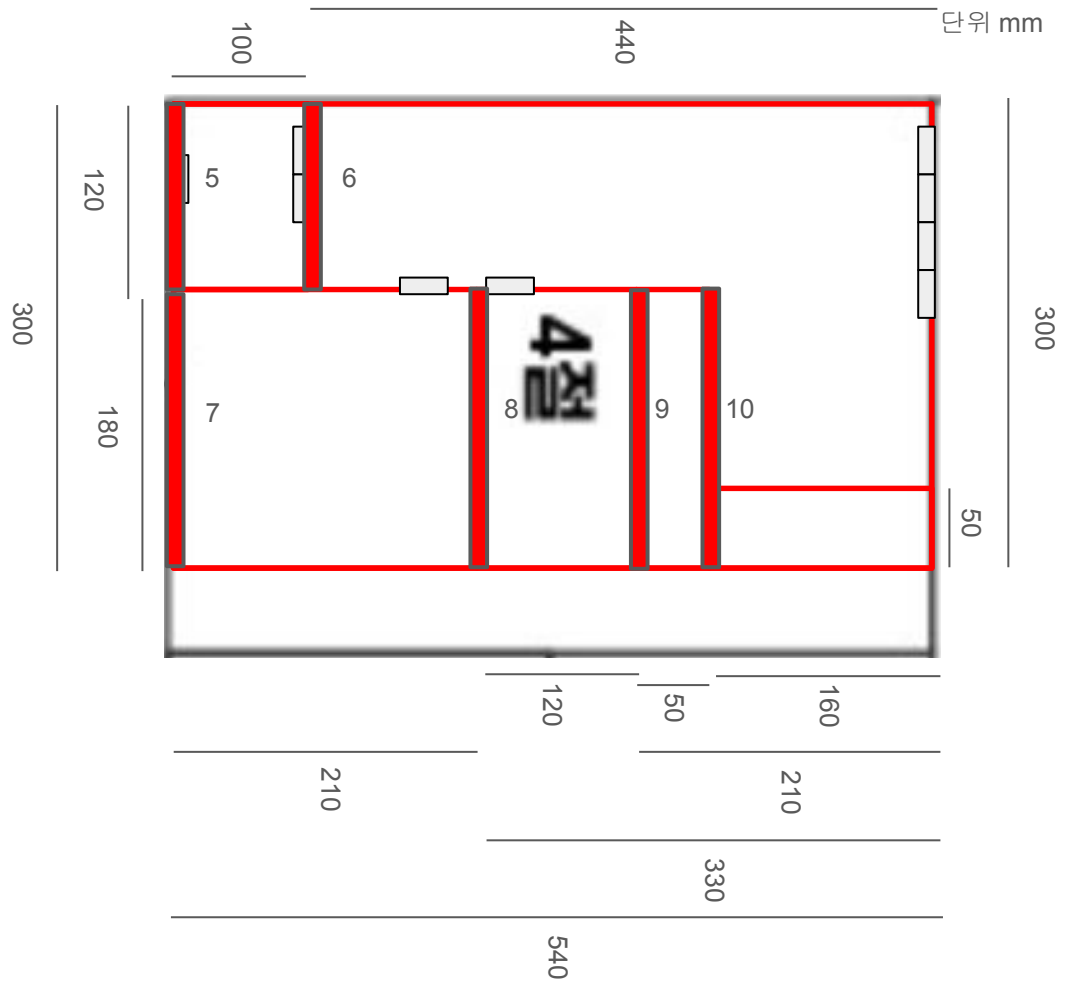
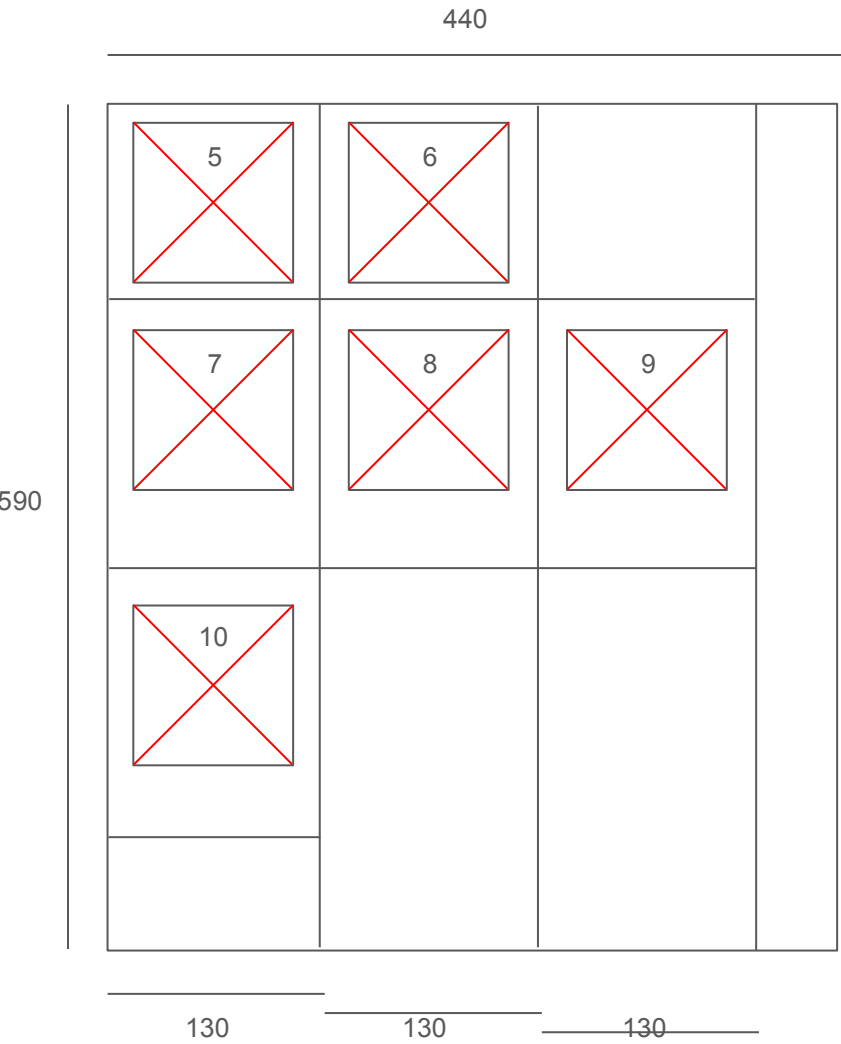


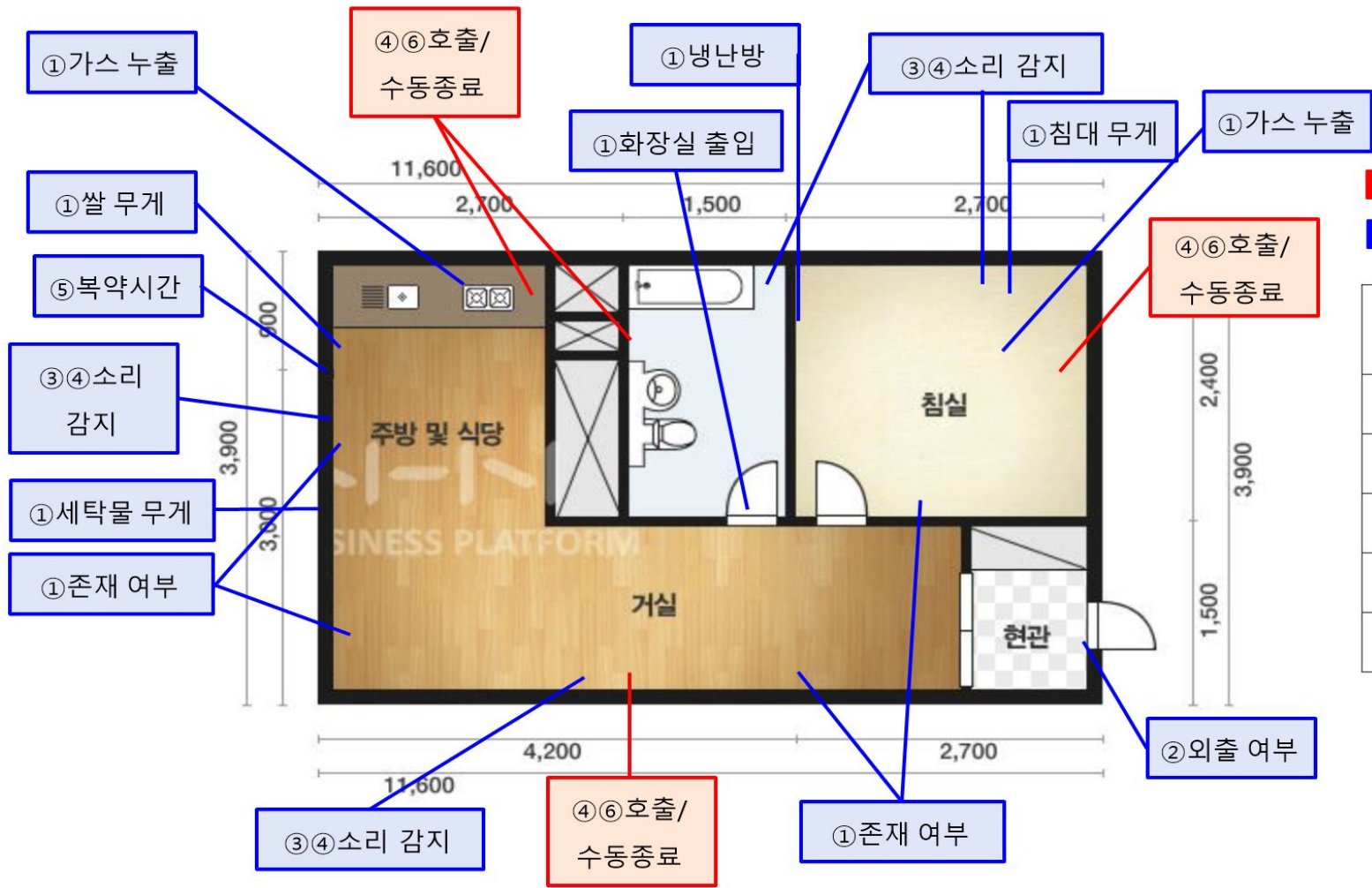
전지	1091*788mm
2절	540*788mm
4절	394*540mm
8절	272*394mm
16절	197*272mm







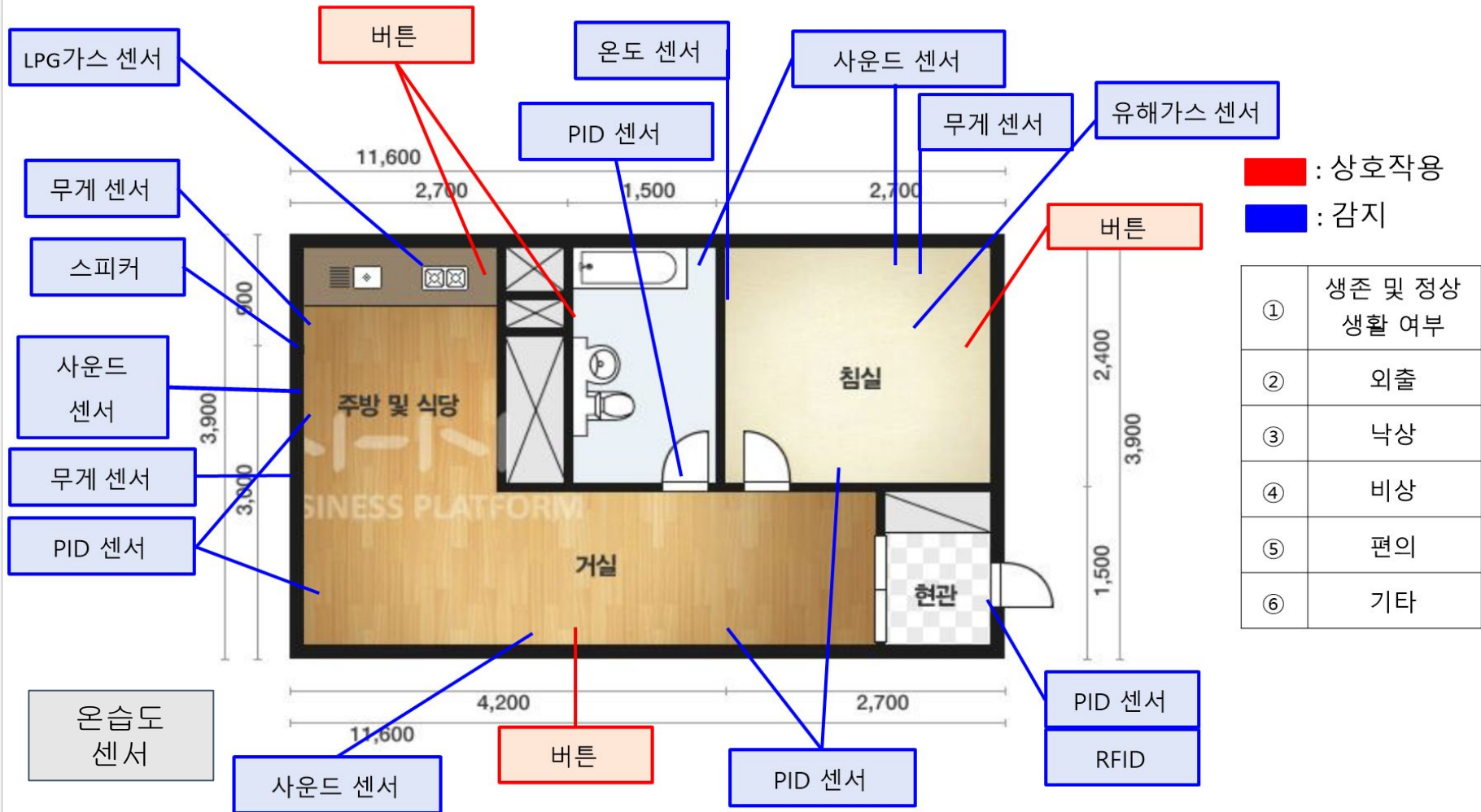




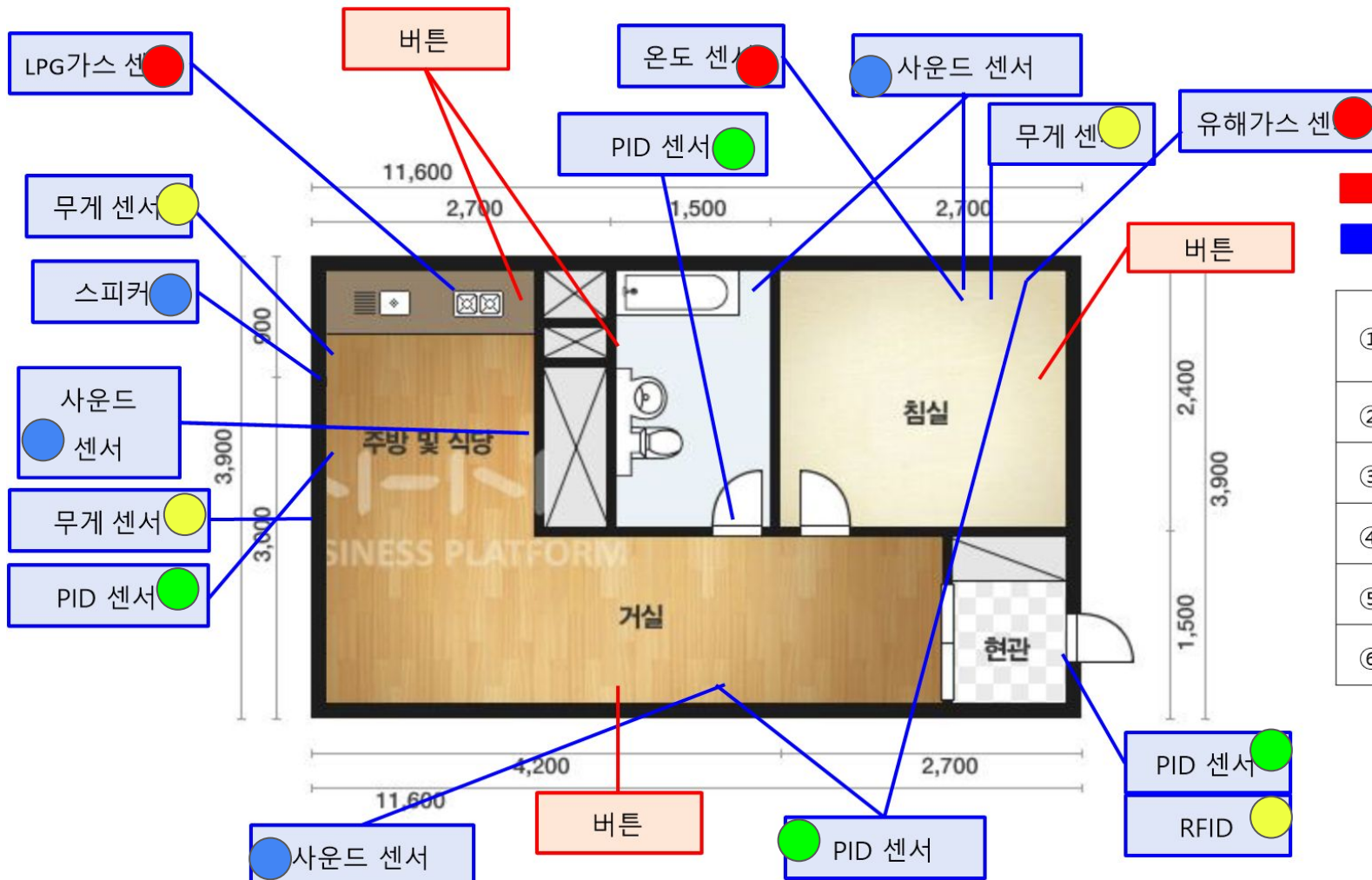
■ : 상호작용

■ : 감지

①	생존 및 정상 생활 여부
②	외출
③	낙상
④	비상
⑤	편의
⑥	기타



①	생존 및 정상 생활 여부
②	외출
③	낙상
④	비상
⑤	편의
⑥	기타



— : 상호작용
— : 감지

①	생존 및 정상 생활 여부
②	외출
③	낙상
④	비상
⑤	편의
⑥	기타

