

Course Project

Обектно Ориентирано Програмиране (OOP with C#.NET)

Име на Проекта: ИНФОРМАЦИОННА СИСТЕМА ЗА МОНИТОРИНГ И АНАЛИЗ УСЛОВИЯТА ЗА ЖИВОТ В СТУДЕНТСКИ ОБЩЕЖИТИЯ

Acceptable Programming Languages:
Deadline:
Instructor

C#.NET
Февруари, 2018(*on the final exam date*)
Dr. Evgeny Krustev

Problem Statement:

1 Project description

ICB received a request from a client to develop an information system that collects data from various sensors located in college dormitories all over the world. Data will be analyzed by independent research organization to evaluate the living conditions and map them to the performance of the students. Participants should have installed specific sensor equipment supplied by the organization and register to smartDormitory.

2 Sensor types

The following sensor types should be supported:

- Temperature, measured in °C
- Humidity sensor, measured in percent
- Electric power consumption sensor, measured in Watts
- Window /door sensor. Allowed values: true/false(True – when the window/door is open; False – when the window door is closed)
- Noise sensor, measured in Decibels

Note: ICB will provide web API to fetch sensors data.

3 Client GUI

The Graphical User Interface should support the following functionality:

3.1 Landing page

Provide relevant information about the system, its purpose and features. This page should also include a map with all sensors (see 3.6) and access to register new sensor (3.2), modify existing sensor (3.3), View all sensors (3.4), Graphical representation of the sensors (3.5) forms.

3.2 Register new sensor

The newly created sensor should have:

- Name
- Description
- Sensor Type
- Polling interval which specifies the amount of time to refresh sensor data (Optional)
- Latitude and longitude
- Range of acceptable values (e.g. -40 °C to +100 °C)
- Tick-off option for receiving sensor alarm (Out of acceptable range)

3.3 Modify existing sensor

The user should be able to edit sensors.

3.4 View list of all sensors

The user should see a list of registered sensors. For each sensor the list should include:

- Name
- Description
- Current value
- Link to graphical representation of the sensor

This page should contain a link to “View sensors on a map” (see 3.6).

3.5 Graphical representation of sensor

The user should have a detailed sensor’s view which includes a graphical representation of data collected from sensor.

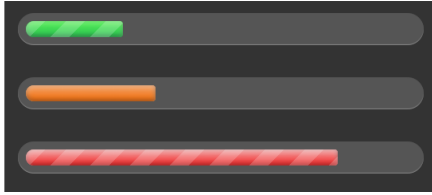
Note: The system should handle when the sensor is offline and show this to the users.

Possible graphical representations are:

Gauge - examples



Bar – examples



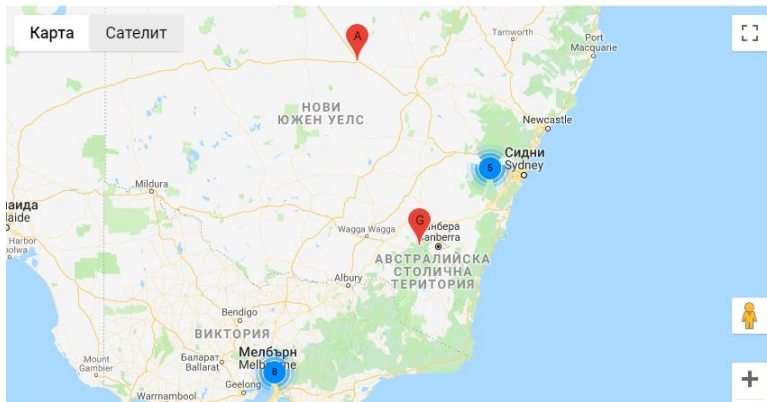
State indicator – examples



3.6 View sensors on a map

This page should provide a map view with sensor markers according to their lat/long. You can use Google Maps API and Marker clustering library.

On the landing page display all sensors for the users.



3.7 Reports

On the dashboard view, the user should be able easy to view the data out of acceptable range values:

4 General requirements

Use the following technologies, frameworks and development techniques:

1. Use **WPF** for the GUI
2. Data storage
 - a. Use **XML file format** to store sensors definition
3. Graphical representation of sensor data
 - a. Use **Telerik WPF** (<https://demos.telerik.com/wpf/>)

https://www.telerik.com/login/v2/download-b?ReturnUrl=https%3a%2f%2fwww.telerik.com%2fdownload-trial-file%2fv2-b%2fui-for-wpf%3futm_medium%3dtelerik%26utm_source%3dqsf%26utm_campaign%3dwpf-trial#register

4. Apply **error handling** and client **data validation** to avoid crashes when invalid data is entered
5. **Documentation** of the project and project architecture (including screenshots)
6. Use **caching** of data where it makes sense (e.g. landing page) (optional)

Evaluation:

Your project will be evaluated on the following general points:

☐ Sophistication/complexity/originality of the problem being solved/investigated and of the solution(s)/approaches considered.

☐ Clarity of explanations, and for implementations programming skill/quality. Your report (in Bulgarian!) should be well written and free of grammatical and spelling errors. Programs must be well-commented and in a professional style.

☐ Awareness of related work. Others have considered the same or similar problems before you. Your work does not have to be novel, but you should be able to contextualize your approach. Be sure to explain how each referenced work is related to your work. Note that a 5-minute Google search will not be adequate; if you are unfamiliar with the required textbooks for the course:

☐ Completeness of the project.

Deliverables: The files with :

1. the source code
2. the executable code
3. the instructions for compiling your source code
4. the report explaining the data structures and the algorithm implementation, describe things such as how your code has been tested, limitations of your code, problems encountered, and problems remaining
5. any files used to test the implementation of the program with an explanation about it included in the report.