



**Saguaro Technology**  
**UPT Senior Scholarship**  
*Project Descriptions*

The Engineering  
Talent You Need.

The Business  
Model You Want.

# 2016 Senior Projects overview; detailed descriptions on pages 4-17

Project	#	Proposed modules
Smart Infrastructure	1	Ultra-low power data acquisition
	2	Environmental control based on sensors
Smart administration	3	Smart administration core overview
	4	HR hiring system module
	5	Online evaluation system module
	6	Visitor access system module
BitVote network	7	P2P network for electronic voting system

# Administrative Schedule

Date		Activity
October 5, 2016		Program presentation
October 6, 2016	→ October 17, 2016	Apply for the project you want
October 18, 2016	→ October 21, 2016	Evaluation & selection of projects and student teams
October 24, 2016		Teams start
October 24, 2016	→ October 28, 2016	Administrative paperwork (sign contracts)
October 31, 2016	→ November 4, 2016	Training offered by SAGUARO
November 7, 2016	→ April 28, 2017	Project design and implementation
May 2, 2017	→ May 19, 2017	Begin presentation preparation
May 19, 2017		Submit documentation for approval
May 31, 2017		Project evaluation by Saguaro

# SmartOps 2018: Ultra-low power data acquisition (1 of 2)

## Module overview and scope

- The purpose of this project is to create an ultra-low power data acquisition system designed to collect data from a number of different sensors with minimal power consumption
- Using MSP432 ultra-low power MCU, the team will design a system (battery + charger + MCU) which can be used as a data logger for long periods of time without charging
- The architecture needs to be created so that adding different sensors can be done without changing the code on MCU; the architecture has to be modular to accommodate different types of sensor interfaces (i2c, adc, spi)
- Data storage should support 24 hours of data for three different sensors
- Sensor types which will have to be integrated:
  - Temperature
  - Accelerometer / Gyroscope
  - Current/Voltage
  - Humidity
- For data interpretation, the project will have to include a standalone application (WPF or web based) which will take the logged data and show it as graphs
- The system will be implemented using an existing “vibration measurement” project which was a diploma project in 2016

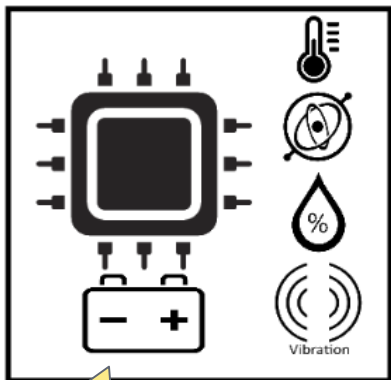
## Team

- Mentor: Deian Cosas
- Team size: 2 students

## Minimum requirements

- Basic C / C++ programming

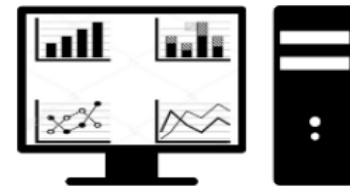
## Illustration of ultra-low power data acquisition module



The MCU system contains sensors to detect temperature, acceleration and movement (accelerometer, gyroscope), current/voltage and humidity



The MCU system will be placed on robot and log/monitor it during motion testing.



The data will be then interpreted and analyzed using a WPF or web based application.

# SmartOps 2018: Environmental control (1 of 2)

## Module overview and scope

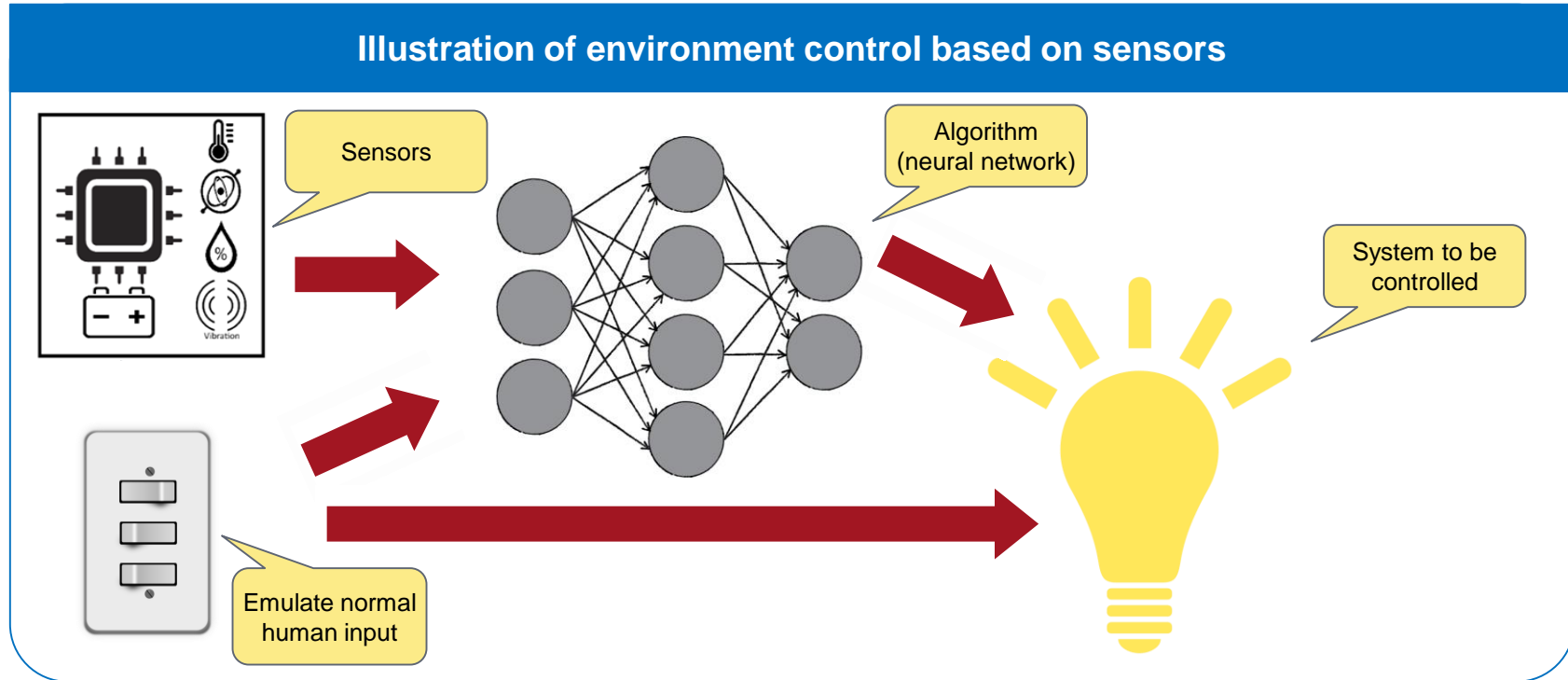
- Analyze the data received from sensors and human actions and correlate them
- After a period of training, the system will attempt to guess the human actions and execute them in anticipation of the need
  - ❑ A light sensor and a LED switch / dimmer can be used for implementation
  - ❑ The solution must be generic, so it can be adapted to any inputs
- Development platform and required components should be chosen together with the student

## Team

- Mentor: Nicolae Adrian Gavrila
- Team size: 1 student

## Minimum requirements

- Basic C / C++ programming



# SmartOps 2018: Smart administration core overview

## Overview of the Core

- Web / cloud platform for centralized SmartOps modules management, authorization and security
- Key components:
  - ❑ Will contain the main SmartOps DB / DB cluster in which the other modules will plug their own tables
  - ❑ Will contain the user interface in which the other modules will plug their own user interface
  - ❑ An SDK for module development will be created for facilitating the creation of new modules that can be integrated in the SmartOps platform.

## Team

- Mentors: Serban Cristea, Paul Boghiu
- Team size: 2 students

## Minimum requirements

- C#, Javascript
- .Net, Angular JS
- MS SQL
- Strong design patterns skills



# SmartOps 2018: HR hiring system module (1 of 2)

## Module overview and scope

- Module that will enable HR and Evaluator persons to collect, share and keep centralized data about the candidates
- The system will act as a web application where the company should be able to perform the following:
  - ❑ Allow tracking of the interviews (evaluation, interview history, etc.), particularly the individuals who were contacted for a discussion and / or interviewed
  - ❑ Provide a tablet-based note-taking application for visual effect during the interview
  - ❑ Retain a status for each individual (e.g., already employed at saguaro, rejected, get in touch after 4 months, etc.) and provide a ranking
  - ❑ Offer an interview schedule calendar that would make the Evaluator's agenda visible to enable more efficient interview scheduling

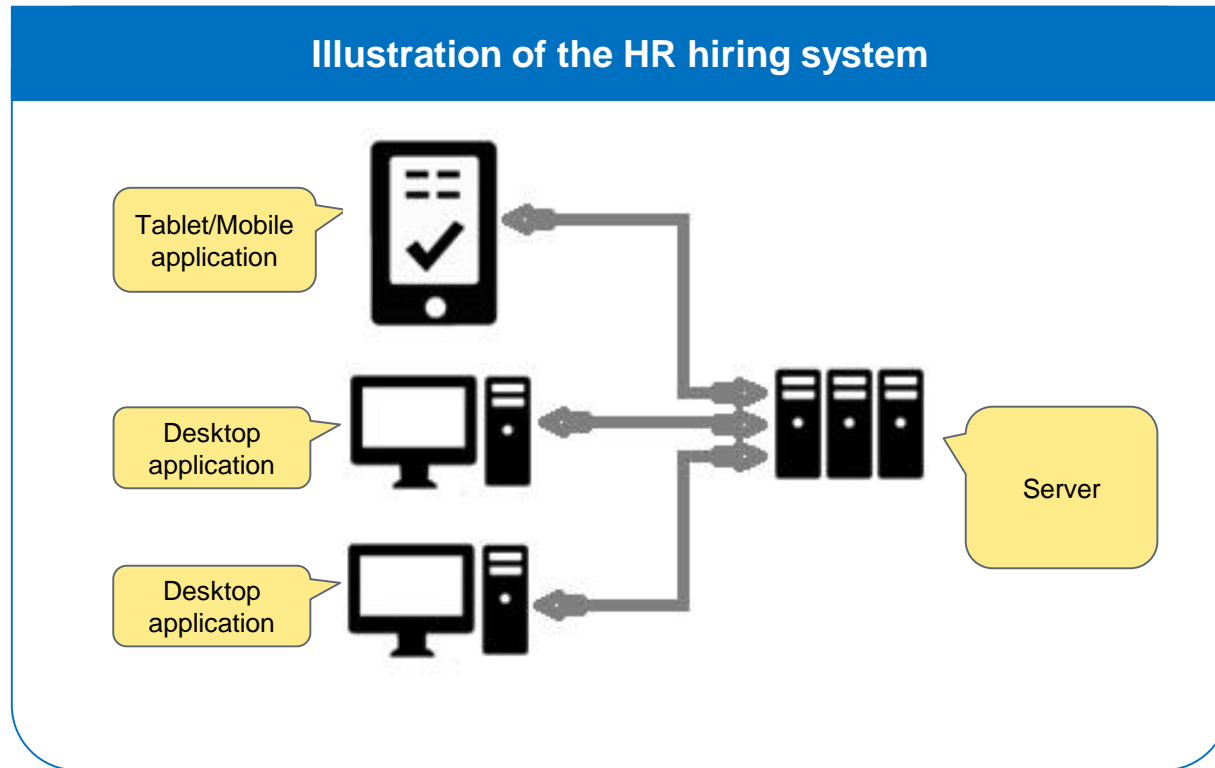
## Team

- Mentors: Vlad Constantinescu, Serban Cristea, Istvan Nuszl
- Team size: 1 student

## Minimum requirements

- C# .net
- JavaScript
- SQL

# SmartOps 2018: HR hiring system module (2 of 2)



# SmartOps 2018: Online evaluation system module (1 of 2)

## Module overview and scope

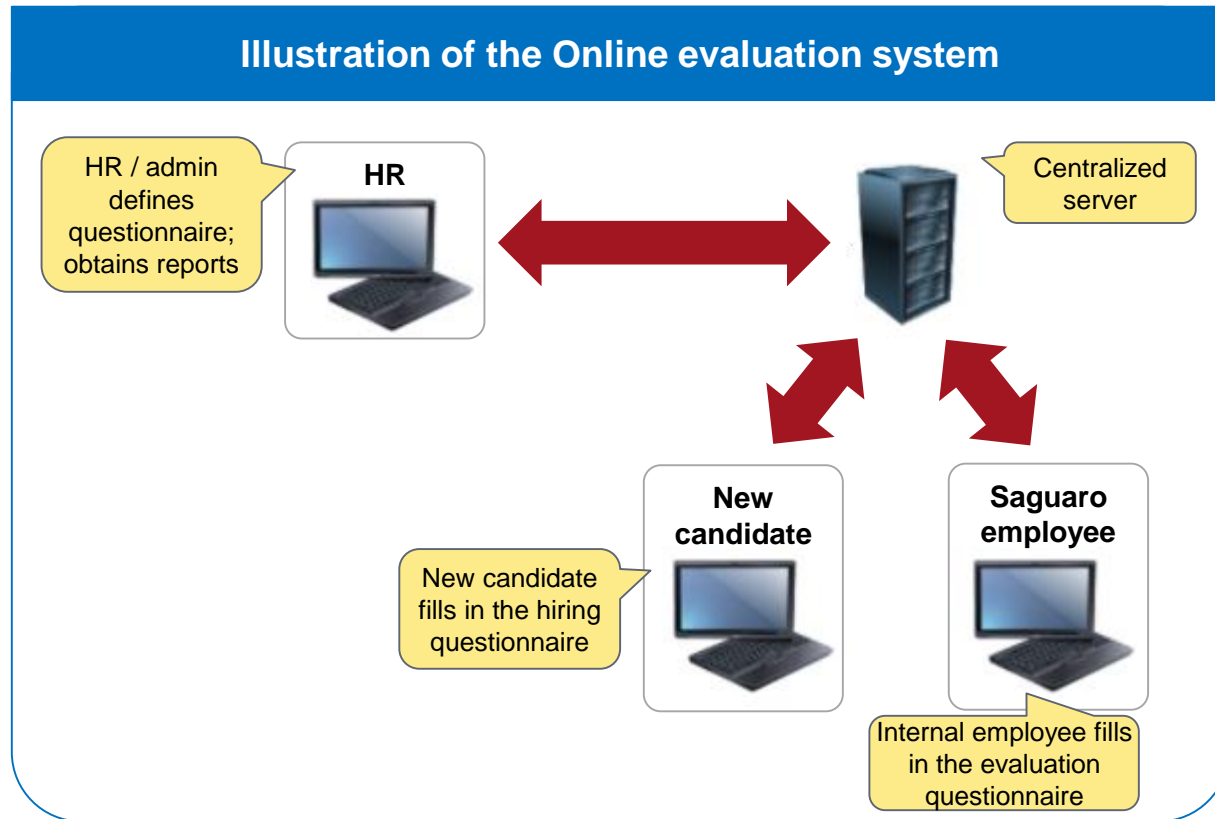
- Module where the company should be able to perform the following:
  - ❑ Online candidate evaluation
  - ❑ Internal questionnaires for different admin / HR topics
- Team leaders / HR members can define the tests / questions using a dedicated administrative area; for each possible / actual employee, an individual link is generated for the evaluation form
- The system should be able to accommodate static questions with simple / multiple answers and dynamic content (graphs, drag and drop, pictures, movies, etc.)
- After the form is completed, an overview / result is displayed in real-time for the employee and is displayed in the administrative area to the person defining the questions
- The system will be comprised of:
  - ❑ A SQL database controlled by an ASP.NET server technology
  - ❑ Frontend UI, driven by JavaScript that communicates with the ASP backend server to interact with the database
  - ❑ An external 'www' domain that can be used externally

## Team

- Mentors: Lucian Florea, Serban Cristea
- Team size: 1 student

## Minimum requirements

- C# .net
- JavaScript



# SmartOps 2018: Visitor access system module (1 of 2)

## Module overview and scope

- Module for registering visitors at reception
- Key components:
  - ❑ A tablet at reception where the visitors will input their info (name, company, visit purpose, ID number, host), take a photo, sign NDA, etc.
  - ❑ A backend system with database logging and email / SMS / app notification for the host; this will include:
    - ❑ Server side that will expose some web services for different functions (like register a new visitor, etc.)
    - ❑ Web application that can be accessed through intranet by employees
    - ❑ Mobile application installed on the mobile device for employees that are typically hosts to receive notifications
  - ❑ A printer that can print sticky labels with the visitor's name, photo and host name
  - ❑ Provide a special visitor card and associate the visitor with the card only for access to restricted areas (e.g., lab)
- Because it involves server, web, and mobile application, this project will allow the student to learn and experiment with different technologies and also will allow the mentors to better evaluate the student

## Team

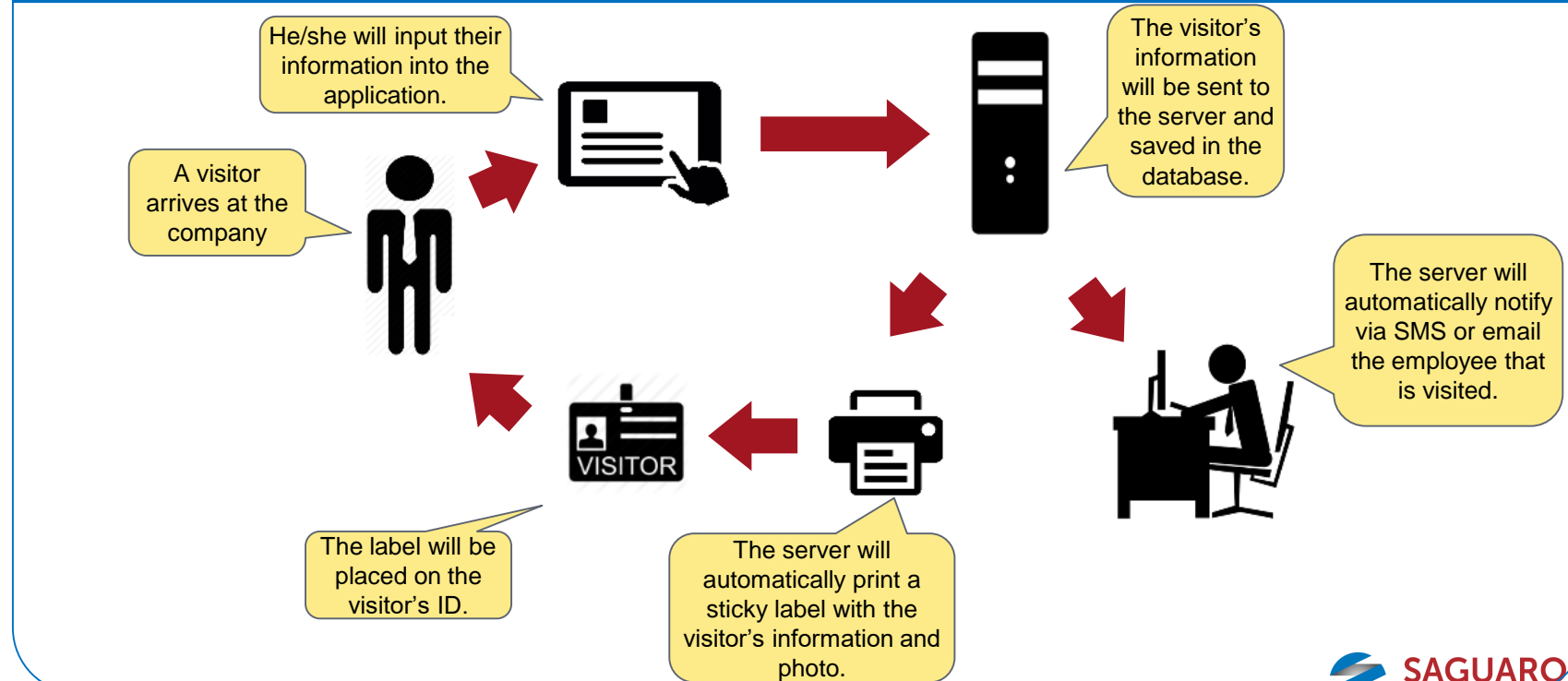
- Mentors: Eduard Niesner, Vlad Constantinescu
- Team size: 1 student

## Minimum requirements

- C# .net, Java
- JavaScript
- SQL

# SmartOps 2018: Visitor access system module (2 of 2)

## Illustration of the Visitor access system



# BitVote – A P2P electronic voting system

## Problem statement and description

- BitVote is a distributed voting system, that is secure, private, transparent and very scalable. The cost of implementation is low, being deployable on available public networks. Another gain of such system is its viability to be used in overlapping voting/polls/etc. sessions.
- Key features:
  - ❑ Similar to BitCoin architecture, having the benefits of a cryptocurrency: “Transactions are recorded into a distributed, replicated public [database](#) known as the [blockchain](#), with consensus achieved by a [proof-of-work](#) system ..” *Wikipedia*. [Wikipedia.org](#). n.p. Web.
  - ❑ Peer-to-peer network, where voting shall operate on a cryptographic protocol.
  - ❑ Central authority to issue new votes. A voter would need to authenticate to the issuing system, and one new vote will be transacted to the voter, as a result.
  - ❑ Privacy. Similar to the BitCoin model, where transaction issuers are anonymous, voters options are hidden and private.
  - ❑ Voting Verification Receipt: Each voter has the possibility of explicit verification of the voting process.
  - ❑ Proof-of-work with a similar implementation of Bitcoin’s “mining”.

## Modules

- The system would have the following modules:
  - ❑ BitVote network (define transactions, blockchain and implement transaction validators).
  - ❑ Central authority
  - ❑ BitVote clients
  - ❑ BitVote anomaly detection

# BitVote – A P2P electronic voting system project module

## BitVote network – A P2P electronic voting system

- Transactions
  - ❑ Transfer of a vote between two clients( e.g. voter and candidate) that gets included in the blockchain
- Blockchains
  - ❑ Individual blocks that contain the proof-of-work result
  - ❑ Similar to Bitcoin, it resembles a “share public ledger”
- Transaction validators
  - ❑ Similar implementation of Bitcoin’s “mining”.
  - ❑ Software running in the cloud validates transactions.

## Team

- Mentor: Bogdan Lile
- Team size: 2 students

## Minimum requirements

- C/C++
- Basic cryptography concepts
- Basic Networking knowledge
- Basic Network programming knowledge



# BitVote – A P2P electronic voting system

