

# Smart Irrigation System

---

## *Arhitecture*

*Team ADMIN*

---

*Robert-Ștefan Muscaliuc*

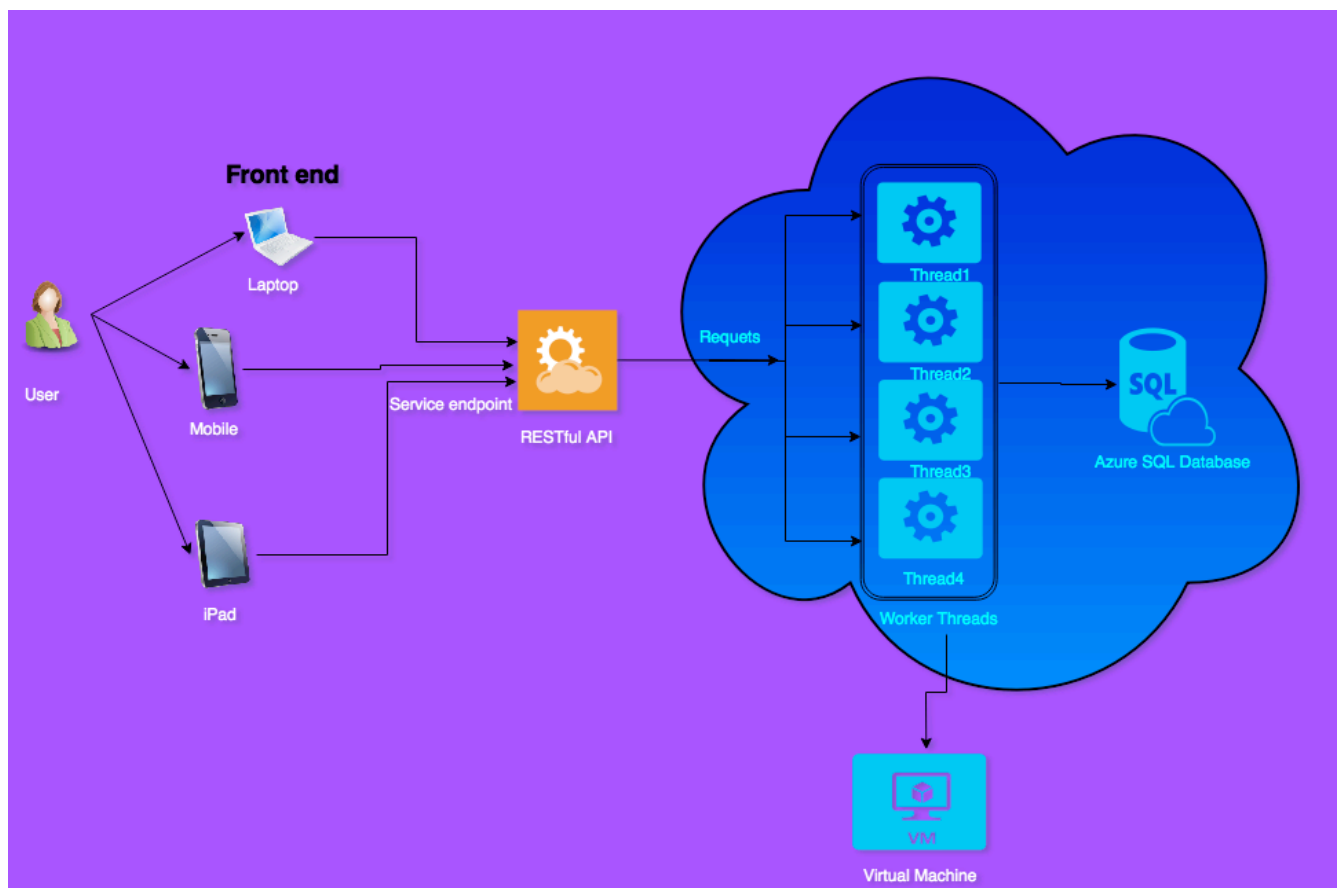
*Daniela Pena*

*Mădălina-Luciana Stănuț*

*Larisa-Cristina Trancă*

*Anul IV, Ingineria Sistemelor*

## 1.Schema aplicației



## 2. Scurtă descriere componente

- **Worker thread**  
Pentru ca sistemul sa fie cat mai actual vom folosi un worker care ruleaza in Azure Cloud si manage-ue toate procesele din aplicatia noastra. Accesul la acest worker se face prin intermediul unei masini virtuale.
- **Data generation**

Pentru generarea datelor se va construi un API RESTful care in primul rand ne va furniza resursa cu informatiile despre vreme( temperatura, umiditate, etc.). Vom folosi un API public care ne va da informatiile necesare pentru a putea manipula

obiectele JSON ce ulterior vor fi transmise front end-ului  
(<https://www.wunderground.com/weather/api/d/docs?d=index>).

Endpoints:

1. Get resource
  2. Create new resource
  3. Update resource
  4. Delete resource
- Front end  
Aplicatia o sa fie disponibila in varianta mobile. Aceasta va fi dezvoltata prin intermediul librăriei de JavaScript, React.js. Request-urile catre API vor fi trimise prin intermediul protocolului HTTP, iar in functie de raspunsul primit de la API vom afla daca zona in care suntem pozitionati este irigata sau nu. Afisajul facandu-se cu ajutorul unui layout de la Google.
  - GitHub repository: [https://github.com/LarisaTranca/DATC-2017/tree/master/TRANCA\\_LARISA\\_CRISTINA/PROIECT](https://github.com/LarisaTranca/DATC-2017/tree/master/TRANCA_LARISA_CRISTINA/PROIECT)

### 3. Development

- *Robert-Ștefan Muscaliuc – Worker & Database & Azure Services*
- *Daniela Pena – Worker & Database & Azure Services*
- *Mădălina-Luciana Stănuț - RESTful API*
- *Larisa-Cristina Trancă – Front end*