

# Ingineria Programării – Curs I

---



UNIVERSITY OF  
BUCHAREST

---

VIRTUTE ET SAPIENTIA

# Motivația cursului

- Obiectivul cursului este să vă învețe să integrați diferite cunoștințe dobândite în facultate pentru a duce un proiect de la 0 la o formă finală, livrabilă.
- Cunoștințelor pe care le-ați dobândit până acum le vor fi adăugate în cadrul cursului cele de analiză, proiectare, gestiune a activităților și lucrul în echipă în cadrul unui proiect software.

# Planul semestrului

S1 - prezentarea motivației cursului, a obiectivelor cursului, și a metodelor de evaluare

Temă - veți forma o echipă de 4-6 persoane și veți complete un formular pe care îl voi posta pe Teams

S2 - prezentarea documentației de analiză - strategii de planificare a aplicației

Temă - inițierea documentului de analiză

S3 - feedback documentația de analiză - strategii de planificare a lucrului în echipă

S4 - S14

- Feedback aplicație, dezvoltare Q&A
- Diferite prezentări în funcție de nevoile echipelor (folosirea OpenAPI, AsyncAPI, principii de testare, documentarea pentru utilizator, etc.)

# Proiect

- Fiecare echipă (de 4 – 6 studenți) va avea ca obiectiv să creeze un **program relativ simplu, dar documentat, și testat**.
- Colecția de aplicații dezvoltate de echipele voastre în acest semestru vor contribui la construirea unui dataset de programe mock IoT.
- Ce este un program IoT?
  - Internet of Things este o rețea de obiecte fizice („things”) care dispun de senzori, software și alte tehnologii pentru a le interconecta, și a schimba informații cu alte dispozitive, sau cu internetul.
  - Nu va trebui să integrați dispozitive hardware.

# Specificațiile programului - 1

- Programul pe care voi îl dezvoltați ar trebui să fie „creierul” unui device smart.
- Va prelua date „din mediul înconjurător” – informații despre lumină, temperatură, perioadă a anului, oră, umiditate ș.a.m.d.
- Va prelua date/setate de utilizator – setări, sau date transmise de utilizator / de un hub central teoretic
- Va transmite starea în care dispozitivul se află, diferitele operațiuni pe care dispozitivul le face

# Specificațiile programului - 2

- Programul trebuie scris în **C++ / Java / Python / Golang**
- Puteți folosi **orice librărie** de C++/Java/Python/Golang, **orice compiler, orice IDE**.
- Programul va trebui să comunice folosind două protocoale: HTTP și MQTT.

\*Doar să nu fie o librărie super OS-specific, sau cu multe condiții/dubioasă

# Evaluare - 1

- Nota finală va fi data de două elemente:
- 1 punct din oficiu yey
- 9 puncte - programul realizat

# Evaluare - 2

- Programul trebuie să respecte următoarele cerințe (2.5p):
  1. Expune un Rest API HTTP - documentat folosind Open API (Swagger)
  2. Expune un API MQTT - documentat folosind AsyncAPI
  3. Aplicația să aibă minim 5 funcționalități - puteți să vă gândiți la ele ca sell points ale aplicației. Depinde de aplicația pe care v-ați propus să o faceți, dar chestii de genul o funcționalitate e scăderea, o altă funcționalitate e adunarea, nu înseamnă chiar că sunt diferite
  4. Tot ce faceți să se găsească într-un singur repo.

Pentru puncte:

1. Toate funcționalitățile și/sau toate endpoints au unit teste asociate. +1.5p
2. Documentația de analiză este up to date + 1p
3. Documentația de utilizare reflectă aplicația reală + 1p



# Evaluare - 3

- Programul trebuie să respecte următoarele cerințe:
  7. Să prelucreze date reale (fie că accesează un alt api pentru a prelua date, fie că descărcați un set de date pe care îl dați apoi aplicației) + 1p
  8. Utilizarea unui tool de testare automată (gen RESTler) pentru a identifica buguri. +1.5p
  9. Integration tests +1p
  10. Coverage al testelor de 80%+ 0.5p

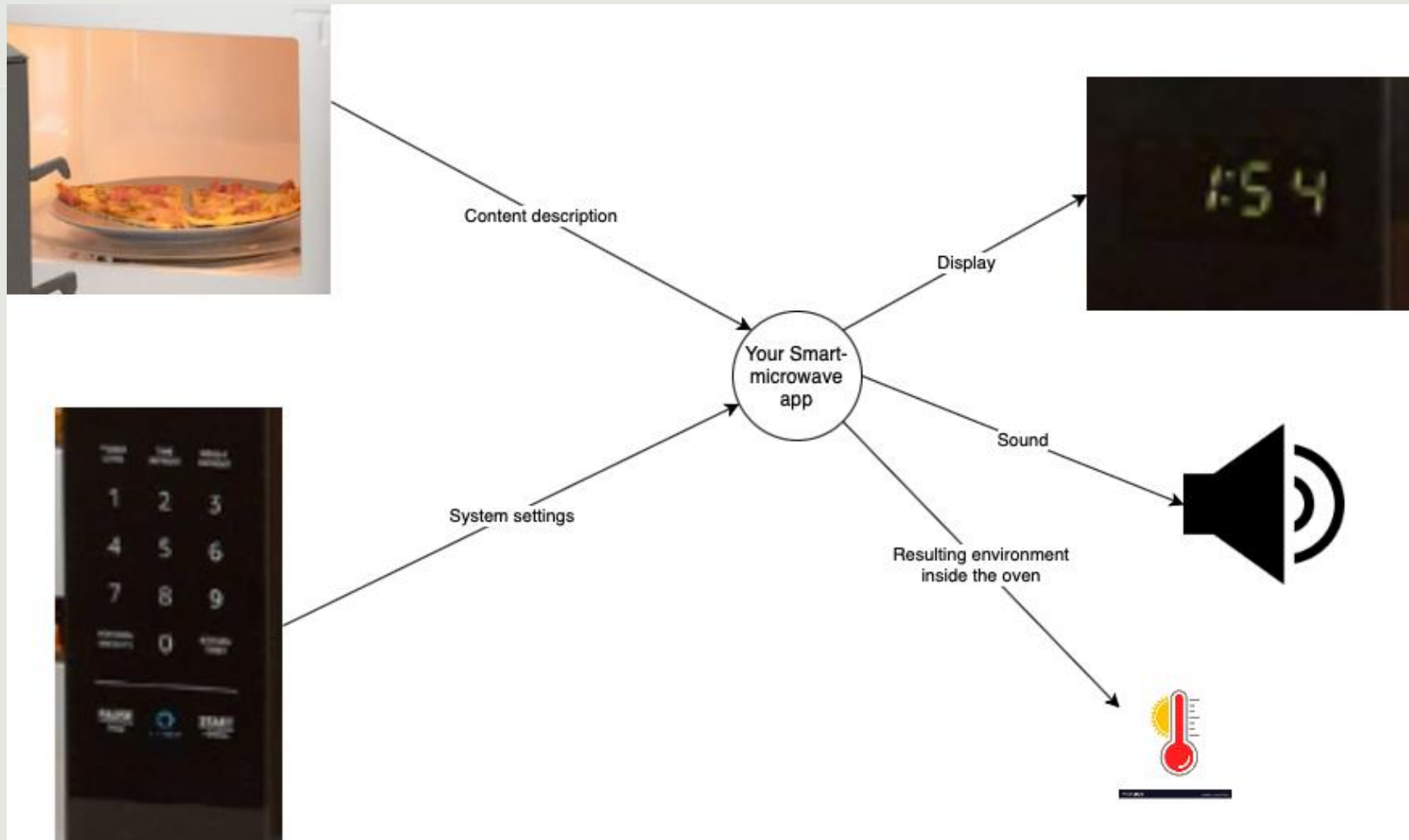
# FAQ

- Cum să ne vină o idee de proiect?
  - Uitați-vă în jurul vostru, găsiți device-uri dumb din casa voastră. Puneți-vă în rolul de a crea un software al acelui device care să îl facă „smart”. Precum: cuptor cu microunde, espresso, mașină de spălat etc.

# Exemplu de aplicație



# Exemplu de aplicație



# Context

- În ianuarie 2021, am scris un articol științific pentru Workshopul [SERP4IoT'21](#) numit „*RiverIoT - a framework proposal for fuzzing IoT applications*”.
- Aplicația River a fost dezvoltată de mai mulți între cercetătorii Facultății de Matematică și Informatică cât și colaboratori externi, și este o aplicație de fuzzing.
- Fuzzing-ul este un mod de a testa o aplicație software prin generarea unui număr cât mai mare de date de intrare, rularea unei aplicații cu respectivele date, și urmărirea erorilor, sau crashurilor aplicației. RiverIoT are la bază execuția concolică și Reinforcement Learning (linkuri la final pentru cei curioși).
- Tocmai pentru că e vorba de device-uri IoT, avem nevoie ca aceste programe să fie compilate într-un binar.

# Context

- La sfârșitul semestrului programele care vor fi făcute bine, vor face parte din datasetul de programe IoT, iar echipele respective vor face parte din acknowledgements în paperul pe care îl vom scrie despre acest dataset.
- Dataset-ul pe care îl construim va fi Open-Source, și vrem să constituie primul benchmark pentru fuzzing-ul sistemelor IoT.