**Studiu de caz**

# Introducere

Sumar: Proiectul a fost inceput in primavara lui 2020, de catre Filos Gabriel si Moise Valeriu, cu scopul de a simplifica dezolva o solutie pentru a simplifica procesul de management al prezentelor studentilor la cursuri, seminarii si laboratoare.

Solutie: Echipa noastra a realizat o platforma web capabila sa gestioneze simultan nenumarate “cataloage”, oferind atat profesorilor cat si studentilor experienta digitala pe care o merita. In urmatorul an, acestia planifica extinderea pe platforme mobile si modalitati de integrare cu sisteme universitare.

Rezultat: Minimium Viable Product-ul nostru a fost testat in versiunile alfa si beta in cadrul facultatii la care studiau membrii echipei la momentul inceperii proiectului, dupa care s-a extins la nivel de universitate, pentru a testa scalabilitatea platformei si adaptabilitatea dezvoltatorilor la noi cerinte si medii.

# Conceptualizare

Nevoile profesorului: Produsul a fost initial conceptualizat sub forma unui catalog online pentru profesor, in care acesta putea adauga elevi si marca prezenta acestora. Livrabilul din prima iteratie nu s-a dovedit a fi potrivit nevoilor utilizatorilor, deoarece delega majoritatea interactiunii catre profesor. Acest lucru a semnalat faptul ca “terenul de joaca” al utilizatorilor trebuie sa fie cat de bine nivelat posibil si procesele au nevoie de un grad nivel ridicat de automatizare. Asadar, urmatorul pas a fost intervievarea potentialilor utilizatori cu privire la asteptarile unei astfel de platforme. Astfel, am descoperit faptul ca unul dintre cei mai importanti actori din procesul nostru, tot o data si cei ce vor incuraja cel mai mult folosirea aplicatiei, sunt profesorii, iar acestia au mai multe asteptari de la un astfel de produs. Ca de exemplu, foarte multi profesori isi imaginau ca la inceputul cursului sau seminarului, elevii sa se poata marca pe sine ca fiind prezenti, acestia adaugand date personale intr-o lista, pe care ulterior profesorul sa o poata consulta si compara cu realitatea. Desigur, aceasta metodologie ridica probleme, deoarece o verificare manuala trebuie realizata. Alti profesori nu au fost interesati de procesul de marcare a prezentelor, ci de rezultatul obtinut de acestia in urma acestui proces; si anume un fisier Excel, in care sa isi centralizeze aceste lucruri, deoarece acolo pot face statistici si sa decida care dintre studenti au numarul minim de prezente, sau cat de mare este rata de prezenta pentru fiecare laborator.

Nevoie studentului: Conform subiectilor, cea mai des auzita fraza de la inceputul seminarului este “Faceti o foaie de prezenta”; modalitate care a functionat decent pana la ora actuala. Insa, studentii au remarcat ca aceasta metodologie acapareaza pana la 10 minute din timpul seminarului, timp in care se plimba acea foaie din mana in mana, pana la ultimul student. Timp in care, se puteau face explicatii suplimentare cu privire la materia din curs, sau alte activitati non-administrative. Nici contra-partea, care a stabilit aceasta metodologie nu este satisfacuta de aceasta, deoarece intelegerea scrisului de mana al studentilor reprezinta o dificultate recurenta. O problema comuna a celor doi actori este sincronizarea acestor date, deoarece deseori se intampla ca studentului sa nu ii fie marcata prezenta la un seminar sau doua, iar profesorul este pus in situatia de a decide daca acest lucru s-a intamplat fie din greseala sa, sau este vorba de o lipsa de sinceritate din partea studentului.

Produsul: Constientizand dificultatile fiecarei parti, am proiectat “Attendance”, o platforma capabila sa gestioneze prezentele studentilor la orele profesorilor, ce rezolva problemele de securitate, sincronizare si ii lipseste pe ambii de munca manuala din trecut. Acest soft se bazeaza pe coduri numite “tokens”, pentru a asigura securitatea si consistenta datelor. Un token este aferent unei ore la care se face prezenta,

Procesul: In “Attandance”, utilizatorul se poate inregistra sau loga prin diferite platforme de autentificare. Dupa autentificare, Profesorul poate crea o “clasa”, pentru care poate face prezenta. Aceasta actiune genereaza un token care va fi distriuit catre studenti. Acestia pot introduce acel token in aplicatie, astfel inregistrandu-si prezenta. Dupa realizarea acestor actiuni, profesorul poate genera si vizualiza raportul excep aferent acelei clase.

**Detalii technice**

In momentul de fata nu exista multe cataloage online de prezente in Romania, insa majoritatea dintre aceastea sunt facute in special pentru invatamantul gimnazial si liceal, nefiind niciunul facut pentru necesitatiile universitare. Cataloagele online precum “Catalog-scolar.ro”, „Adservio” sunt facute in special pentru invatamantul pre-universitar si astfel acestea nu rezolva probleme precum prezente optionale, notari particularizate ,calcularea de credite sau praguri minime diferite de trecere.

Daca ar fii sa comparam din punct de vedere architectural solutii precum “Catalog-scolar.ro” cu solutia noastra ar fii o mare diferenta in ceea ce priveste modul in care este conceput. O solutie clasica ar fii ceva de genu urmator:



Aceasta solutie fiind conceputa intr-o maniera simpla. Pornind de la serverul de date, in general se foloseste o baza de date SQL, iar serverul fiind unul foarte “gras” continand toata logica si managmentul datelor.

Dupa cum se poate vedea aceasta solutie are diverse probleme precum:

1. Se foloseste in general un server simplu fara un loader balancer ceea ce ar duce la o problema majora in cazul universitatiilor, deoarece poate aparea o incarcare majora in cazul in care s-ar face prezenta la curs in masa in contextual in care sunt peste 150 de studentii.
2. Fisierele sunt stocate sistemul de fisiere al serverului ceea ce ar duce la probleme de performanta

Technologiile folosite in general fiind cele simple:

* Servere proprii cu stocare locala in server

Din punct de vedere al marketingului nici o solutie nu se adreseaza universitatiilor, ci doar scolilor pre-universitare, ceea ce constituie un avantaj in cazul nostru, cat si un dezavantaj deorece nu se poate compara sau intemeia un plan de marketing bazat pe greselile/probleme altora.

**Overview al technologiilor**

Ca technologii pe care le dorim sa le folosim in crearea solutiei noastre se numara:

1. Cosmos DB este o serviciu de baze de date distribuita global, multi-model care se scaleaza in functie de necesitatii. De asemenea acesta este un serviciu complet gestionat de catre cei de la Azure, avand suport pentru diverse API precum Mongo, Cassandra, SQL.

Motivele pentru care am ales acest serviciu sunt urmatoarele:

* Latenta mica
* Disponibilitatea la nivel global
* Scalabiliatea

1. Azure Storage este un serviciu de stocare de fisiere, scalabil, durabil si cu o disponibilitate foarte ridicata. Acest serviciu este impartit in mai multe categorii precum File, Disk, Blob, Data Lake Storage, s.a. Motivul pentru care am ales acest serviciu este costul redus si disponibilitatea ridicata de aproximativ 99.99%.
2. Azure App Service este un serviciu bazat pe HTTP pentru gazduire de aplicatii web care se ocupa singur de gestionarea sa, adica scalarea si actualizarea automata, permitand astfel si integrarea sa cu alte technologii.

Alegand acest serviciu ne permite sa rezolvam probleme precum scalabilitatea serviciului nostru si disponibilitatea acestuia. Un alt motiv ar fii ca are suport nativ pentru Java.

1. Azure Api gateway este un serviciu gestionat complet de cei de la Azure care permite clienților să publice, să securizeze, să transforme, să întrețină și să monitorizeze API-urile. Acest serviciu lucreaza de obicei in colaborare cu alte servicii precum cele de mai joc, astfel motivul pentru care am ales sa folosim acest serviciu este deoarece ne permite sa securizam API foarte simplu daca il folosim impreuna cu alt serviciu precum cel de mai jos.
2. Azure Active Directory este un serviciu cloud Microsoft de identitate si management al servicilor web, care se scaleaza in functie de necessitate. Motivul pentru care dorim sa folosim acest serviciu este deoarece acesta se integreza foarte bine cu celelalte servicii pe care le folosim.
3. Virtual network este un block fundamental in constructia unei retele private in Azure. Aceasta permite in general conectarea intre diverse resource de pe Azure, astfel incat acestea sa comunice in mod sigur. Motivatia alegeri acestui serviciu este aceea ca resursele interne sa functioneze in mod sigur.
4. Spring Boot este un framework Java enterprise, care se axeaza pe flexibilitate si scurtarea codului scris, astfel incat sa se poata scrie o aplicatie functionala intru-un timp cat mai scurt.