

UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ
MANAGEMENTUL LUCRĂRILOR DE LICENȚĂ

Tanasă Nicoleta

Sesiunea: **iulie, 2018**

Coordonator științific

Colab. Olariu Florin

UNIVERSITATEA „ALEXANDRU IOAN CUZA” IAȘI
FACULTATEA DE INFORMATICĂ

LUCRARE DE LICENȚĂ
MANAGEMENTUL LUCRĂRILOR DE LICENȚĂ

Tanasă Nicoleta

Sesiunea: **iulie, 2018**

Coordonator științific

Colab. Olariu Florin

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA

DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „*Managementul lucrărilor de licență*” este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau străinătate. De asemenea, declar că toate sursele utilizate, inclusiv cele preluate de pe Internet, sunt indicate în lucrare, cu respectarea regulilor de evitare a plagiatului:

- toate fragmentele de text reproduse exact, chiar și în traducere proprie din altă limbă, sunt scrise între ghilimele și dețin referința precisă a sursei;
- reformularea în cuvinte proprii a textelor scrise de către alți autori deține referința precisă;
- codul sursă, imaginile etc. preluate din proiecte *open-source* sau alte surse sunt utilizate cu respectarea drepturilor de autor și dețin referințe precise;
- rezumarea ideilor altor autori precizează referința precisă la textul original.

Iași,

Absolvent *Tanasă Nicloleta*

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „*Managementul lucrărilor de licență*”, codul sursă al programelor și celelalte conținuturi (grafice, multimedia, date de test etc.) care însoțesc această lucrare să fie utilizate în cadrul Facultății de Informatică.

De asemenea, sunt de acord ca Facultatea de Informatică de la Universitatea „Alexandru Ioan Cuza” Iași să utilizeze, modifice, reproducă și să distribuie în scopuri necomerciale programele-calculator, format executabil și sursă, realizate de mine în cadrul prezentei lucrări de licență.

Iași,

Absolvent Tanasă Nicoleta

Cuprins

1. Introducere	6
1.1 Motivație	6
1.2 Context	6
1.3 Cerințe funcționale	7
1.4 Abordare tehnică	10
1.4.1 Limbaje de programare	10
1.4.2 Instrumente software	13
2. Contribuții	14
3. Proiectare.....	16
1. Arhitectura soluției.....	16
2. Modelare datelor	16
3. Comunicarea server-client.....	16
4. Interfata cu utilizatorul	16
Concluzii	16
Bibliografie	16

1. Introducere

1.1 Motivație

Susținerea lucrării de licență reprezintă un pas important în viața fiecărui student, aceasta dându-i încă o șansă de a pune în practică majoritatea cunoștințelor acumulate pe parcusul înregii facultăți și de a demonstra faptul că și le-a asumat în mod conștiincios.

Din cauza modului curent de viață, de cele mai multe ori, nici viitorul absolvent, nici coordonatorul științific al acestuia nu dispun de o perioadă de timp suficient de mare încât să asigure o colaborare care să acopere în totalitate nevoile studentului și așteptările coordonatorului. Din acest motiv am ales să realizez o aplicație web care să faciliteze înreg procesul de supraveghere a dezvoltării acestei lucrări pentru ambele părți implicate.

1.2 Context

În vremurile în care s-au pus bazele primelor universități din lume, în jurul anului 1100, profesorii erau forțați de către asociațiile studențești, formate în urma unor discuții cu cei aflați la conducerea orașelor, să respecte o serie de reguli, un exemplu fiind să nu lipsească de la cursuri fără aprobarea studenților. Aceștia trebuiau să se supună regulilor deoarece singurele lor venituri erau banii pe care studenții îi investeau în învățământ. Simțindu-se excluși din procesul de construire al acestor reguli, profesorii au început să impună de asemenea anumite reguli care să fie aplicate studenților, una dintre acestea fiind elaborarea unei lucrări de licență la finalul studiilor.

Astfel, în scurt timp, mediul academic universitar a prins o formă asemănătoare cu cea din zilele noastre, necesitatea dezvoltării unei lucrări de licență datând deci de acum aproximativ 900 de ani iar importanța acesteia rămânând aceeași.

În mod convențional, procesul de elaborare a lucrării de licență și, în cazul în care este necesar, a unei componente practice care să stea la baza acestei lucrări se realizează pe parcursul mai multor luni sub supravegherea unui coordonator, urmărind îndeaproape sfaturile acestuia. În acest sens au loc întâlniri periodice între cele două părți implicate în proces. Aceste perioade pot

varia în funcție de student sau coordonator și se stabilesc într-un interval din zi în care ambii participanți sunt prezenți în facultate și au la dispoziție suficient timp pentru a discuta eventualele nelămuriri sau următorii pași ce trebuie urmați.

În acest moment nu există o aplicație folosită pe scară largă și care să poată să îmbunătățească acest proces iar parcursul urmat depinde de fiecare fiecare coordonator științific, acesta alegându-și metoda prin care comunică studenților aspecte cum ar fi: următoarea întâlnire pentru o nouă discuție, conținutul ce trebuie modificat, corectat sau eventual îmbunătățit, sfaturi pentru a-i ușura munca și pașii pe care trebuie să îi parcurgă până la următoarea lor discuție. Există totuși un site care găzduiește o platformă numită EduSoft¹, aceasta fiind o asociație de tip ONG coordonată de către un specialist, sub îndrumarea căruia studenții pot alege să își elaboreze lucrarea de licență. Neajunsul acestei platforme este acela că numărul studenților care au folosit-o este extrem de redus iar din anul 2015 nu mai pare să existe activitate.

Am ales deci să construiesc un mod de a le ușura munca atât coordonatorilor cât și studenților prin a reduce cantitatea de timp pierdută prin aglomerația orașelor, prin așteptarea reciprocă pentru a avea o întâlnire, prin discuțiile care pot divaga de la subiect și prin neînțelegerile ce pot apărea cu privire la anumite secțiuni din lucrare din cauza unor eventuale exprimări neclare. Aplicația este adresată studenților ciclului de licență și coordonatorilor acestora, venind în ajutorul lor cu tot felul de funcționalități gândite pentru a le ușura în special modul de manageriere a timpului și pentru a asigura o mai bună comunicare între cei doi, fiecare dintre aceștia putând actualiza datele și/sau fișierele în orice moment le permite timpul.

1.3 Cerințe funcționale

Având ca principal mediu de utilizare cel universitar, studenții își vor putea crea un cont fiind restricționați de apartenența lor la o facultate. Deci aceștia vor folosi email-ul facultății curente pentru a se putea înregistra și mai apoi loga. De crearea conturilor coordonatorilor se va ocupa o persoană desemnată pentru acest lucru.

Un student va avea la dispoziție o listă care va conține numele coordonatorilor disponibili pentru o anumită sesiune de licență și va putea aplica la unul dintre acei profesori urmând ca

¹ <https://www.edusoft.ro/>

cererea lor să fie acceptată sau respinsă de către coordonator. În același timp, dacă un profesor are o listă prededefinită de idei o va putea face publică astfel încât studenții interesați de subiecte/domenii să poată lua la cunoștință aceste aspecte.

Aplicarea unui student la un îndrumător va presupune, de asemenea, scrierea unui paragraf în care să ofere câteva detalii despre o posibilă idee pentru viitoarea temă ce va fi mai departe dezbătută de cei doi participanți sau va putea cere o sugestie în caz că nu are încă ceva clar în minte.

Acceptarea sau respingerea cererii unui student de către un coordonator va ține stict de acesta. El va avea posibilitatea să își păstreze criteriile de departajare pe care le-a utilizat până în momentul de față. De asemenea, va avea acces la mediile studentului din fiecare semestru în caz ca acesta va fi considerat un factor important sau în caz că dorește să își contureze o idee despre nivelul actual la care se află studentului.

Un coordnator își va putea alocă un număr maxim de studenți, sub o limită stabilită în prealabil de reprezentanții facultății, pentru fiecare sesiune de prezentări în funcție de planurile proprii iar după cel acel număr va fi depășit niciun alt student nu va mai putea aplica pentru îndrumarea lui.

Dacă cererea unui student este acceptată el nu va mai putea aplica pentru un alt coordonator iar în caz contrar aplicabilitatea rămâne valabilă până când acesta își va găsi îndrumătorul, procesul de selectare a unui îndrumător continuând în același mod: aplicare cu o primă idee despre temă sau cererea unui sfat și mai apoi acceptare/respingere din partea coordonatorului.

În perioada dintre momentul aplicării, din partea studentului, și momentul acceptării/respingerii, din partea cadrului didactic, studentul va fi redirectionat către aceeași pagină în care sunt listate opțiunile domnilor profesori, dar opțiunea de a aplica din nou va fi blocată. În cazul în care la un termen hotărât de către reprezentanții facultății vor exista studenți care nu au un profesor coordonator asociat atunci respectivii se vor adresa secretariatului și își vor prezenta motivele pentru care doresc să fie coordonați de un cadru didactic dar nu au reușit să se înscrie în timp util, și vor rămâne să susțină lucrarea de licență într-o sesiune ulterioară.

După acest pas va începe efectiv interacțiunea coordonator-student având ca primă etapă stabilirea mai clară a temei lucrării. În acest sens vor fi disponibile anumite câmpuri ce vor putea

fi completate de ambele părți și în care se vor completa tot felul de detalii depre ideile fiecăruia până când ambii vor fi de acord cu propunerile făcute. Aceste detalii vor putea fi dezvoltate oricât de mult pentru a fi clar ambelor părți ceea ce urmează a fi făcut.

Dacă cei doi o să își dorească totuși o întâlnire față în față vor trebui doar să se puna de acord cu data și ora întâlnirii, aceste date fiind completate într-un loc special rezervat care va fi prestabilit de coordonator în funcție de orele sale disponibile.

După ce o temă va fi stabilită fiecare student va putea accesa pagina coordonatorului său unde acesta va posta periodic anunțuri, termene limită și orice alt tip de conținut dorește să îl aducă la cunoștință studenților săi. Studenții vor putea, deasemenea, posta întrebări/nelămuriri pe această pagină, aceasta fiind publică pentru toți studenții înscriși la respectivul coordnator.

Când studentul începe munca propriu-zisă, prima versiune a lucrării sale va fi încărcată într-un sistem de versionare și mai apoi un link către acest sistem va fi pus la dispoziție, urmând ca profesorul coordonator să o verifice și să acorde sfaturi de care studentul să țină cont mai departe. Pe parcursul evoluției lucrării, studentul nu va trebui decât să se asigure că toate modificările lui sunt actualizate în sistemul de versionare folosit.

Ceilalți studenți înscriși la același profesor coordonator vor putea de asemenea să vadă lucrările tuturor colegilor și eventual să își spună părerea pentru a îl ajuta pe studentul posesor al lucrării să o relizeze cât mai bine cu putință.

Atât coordonatorul cât și studentul vor avea acces la o serie de grafice realizate pe baza sistemului de versionare folosit de către student care vor furniza informații de tipul: de câte ori un student și-a actualizat lucrarea raportat la săptămână/lună, compararea a două versiuni consecutive ale lucrării sau proporția limbajelor de programare folosite. De asemenea, aceștia vor putea adăuga comentarii pe conținutul lucrării.

Fiecare student și coordonator va avea acces la două pagini special construite pentru a le furniza acestora informații despre perioada susținerii lucrării de licență, perioadele în care se desfășoră înscrierile, documentele necesare și sfaturi utile pentru a putea fi siguri că totul va merge bine pe parcursul prezentării lucrării.

1.4 Abordare tehnică

1.4.1 Limbaje de programare

.NET Core este versiunea complet rescrisă a principalului framework Microsoft, .NET Framework. Acesta are la bază limbajul C# și are ca principale avantaje faptul că poate rula pe mai multe platforme (nu doar Windows, cum eram obișnuiți, ci și pe Linux sau MacOS), compatibilitatea totală cu vechiul framework, menționat anterior și faptul că există posibilitatea accesării codului sursă, acesta fiind public.

ASP.NET Core este un framework folosit pentru crearea aplicațiilor de tip web sau cloud. Acesta este complet open-source și disponibil pe site-ul GitHub, site de găzduire al proiectelor software. Utilitatea principală a acestui framework este că aplicațiile dezvoltate cu ajutorul lui pot rula atât pe tehnologia .NET Core, cât și pe .NET Framework (fapt vizibil în *Figura 1*).

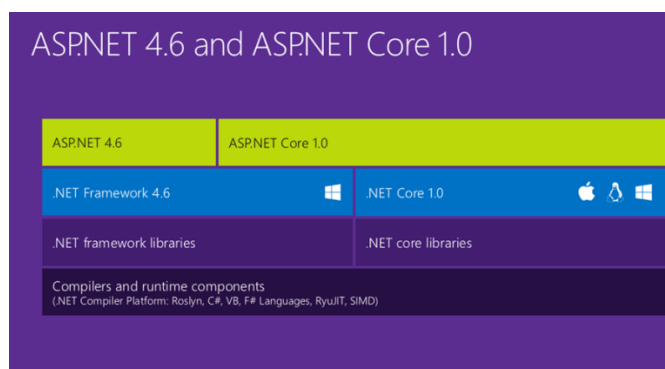


Figura 1: ASP.NET vs. ASP .NET Core²

Entity Framework Core este versiunea independentă de platformă al cadrului Entity Framework, acesta fiind un ORM (Object Relational Mapping) care permite dezvoltatorilor .NET să lucreze cu o bază de date folosind obiecte .NET. Există două abordări posibile ale acestei versiuni, și anume Code-First (mai întâi se scriu modelele și mai apoi se generează tabelele bazei de date cu ajutorul lor) și Database-First (mai întâi se crează tabelele bazei de date și mai apoi se generează modelele aferente), după cum se observă în *Figura 2*.

² Sursa: <http://www.jomendez.com/2017/02/15/asp-net-core-1-0/>

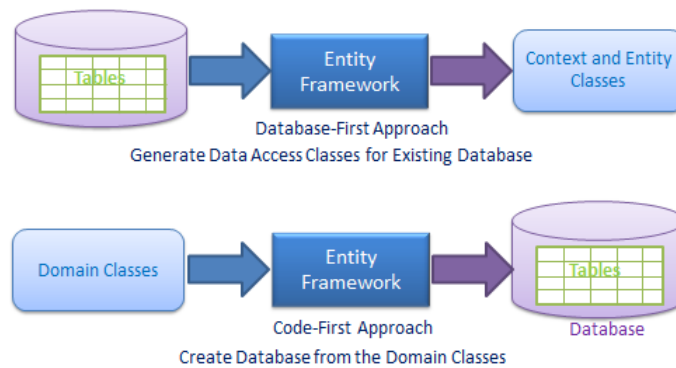


Figura 2: Abordări Entity Framework Core³

Din aceste două abordări, cea utilizată este Entity Framework Core Code-First datorită faptului că permite utilizarea propriilor clase pentru a reprezenta modelul pe care se bazează Entity Framework pentru a efectua orice fel de operații la baza de date, codul fiind scris mai întâi și apoi generându-se modelul pe baza acestuia.

HTML (HyperText Markup Language) este un limbaj de marcare utilizat pentru crearea paginilor web ce pot fi afișate într-un browser scopul acestuia fiind de a prezenta informațiile (font, tabelă, paragraf) și putând fi construit folosind un simplu editor de texte. Elementele acestuia, reprezentate prin structuri numite tag-uri, alcătuiesc scheletul unei pagini, unele exemple fiind <head>, <body>, <header>, <footer>⁴.

SCSS este un superset al limbajului CSS (Cascading Style Sheets) care se scrie în același mod dar permite folosirea caracteristicilor limbajului Sass (Syntactically Awesome Style Sheets), acesta fiind un preprocesor CSS. Utilitatea acestuia în construirea aplicației mele constă, în principal, în utilizarea variabilelor pentru memoria diferitelor elemente (culoare, font) care ajută la scrierea unui cod curat, ușor mentenabil și fără părți duplicate, având două exemple expuse în Figura 3, respectiv Figura 4.

```
$content-container-bg-color: rgba(222, 218, 191, 0.9);
$text-font: 'Lucida Sans';
```

Figura 3: Declararea unor variabile SCSS

³ Sursa: <http://www.entityframeworktutorial.net/efcore/entity-framework-core.aspx>

⁴ HTML pentru începători: <https://www.w3schools.com/html/>

```
.numberOfstudents{
  background-color: $content-container-bg-color;
  border: solid $input-border-color-default 1px;
```

Figura 4: Folosirea unor variabile SCSS

Angular este un framework structural construit în întregime cu ajutorul limbajului TypeScript, care este un superset al limbajului JavaScript ce poate fi folosit pentru a crea partea de client a unei aplicații. Avantajul major pe care l-am avut folosind ambele tehnologii a fost ușurința transmiterii informațiilor între server și pagina web, fiind posibilă stocarea datelor în obiecte ușor manipulabile, validarea facilă a datelor și eliminarea codului duplicat prin utilizarea de metode. În figura de mai jos (Figura 5) se găsesc menționate alte patru avantaje ale utilizării Angular.

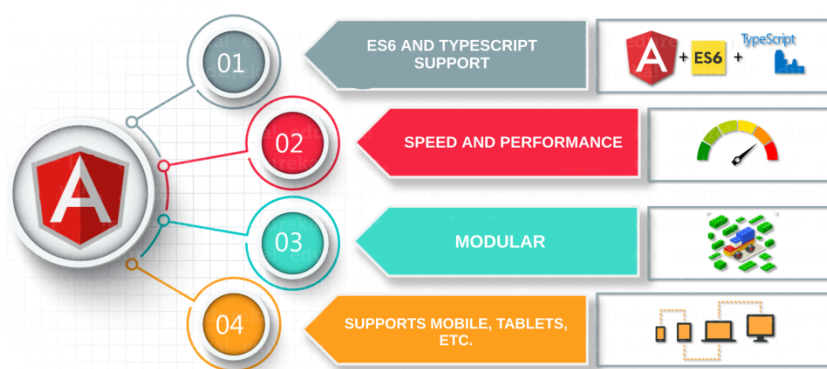


Figura 5: Avantajele utilizării Angular⁵

C# este un limbaj de programare orientat-obiect conceput de Microsoft la sfârșitul anilor 90. .NET Core permite dezvoltarea aplicațiilor folosind oricare din limbajele C#, F# sau Visual Basic. Eu am ales folosirea primului dintre ele deoarece este unul dintre cele mai puternice, pe care îl utilizează de milioane de alți oameni după cum poate fi observat în Figura 7.

⁵ Sursa: <https://softwaredevelopment.ae/angular-best-solution-2018-web-app/>

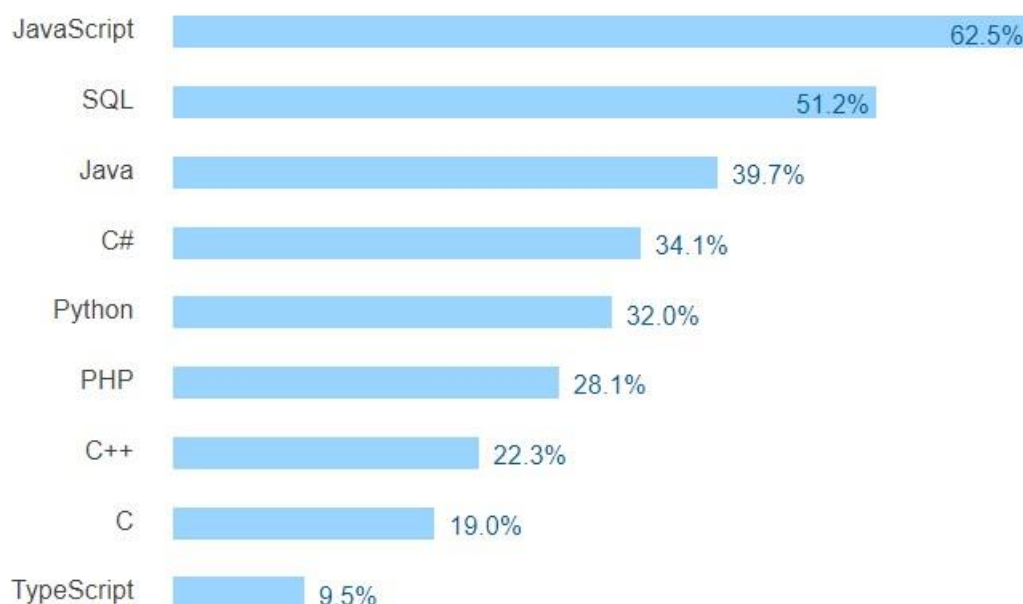


Figura 7: Fragment dintr-o imagine care conține cele mai populare tehnologii în 2017⁶

1.4.2 Instrumente software

Visual Studio Code este un editor de cod sursă ușor, dar puternic, care rulează pe desktop și este disponibil pentru Windows, MacOS și Linux⁷. Dispune de suport încorporat pentru TypeScript, acesta fiind principalul motiv pentru care am ales să îl folosesc pentru dezvoltarea interfeței cu utilizatorul.

Visual Studio este, de asemenea, un editor de cod sursă, cu ajutorul căruia pot fi contruite diferite tipuri de aplicații (desktop, servicii, mobile). L-am utilizat datorită multitudinii de avantaje pe care acesta le oferă. Unele dintre ele sunt: sugestiile pe care acesta le oferă în timp real, modul în care sunt afișate și generate erorile de compilare și ușurința creării unei structuri optime de pentru aplicație.

Postman este o unealtă software care poate fi utilizată pentru a simula cererile HTTP⁸ care ar fi în mod normal trimise de către un client către server. Utilitatea lui a fost posibilitatea testării serviciilor oferite de către server, după cum se poate observa în *Figura 8*.

⁶ Studiu, imagine: <https://insights.stackoverflow.com/survey/2017#mostpopular-technologies>

⁷ Documentație Visual Studio Code: <https://code.visualstudio.com/docs>

⁸ Hypertext Transfer Protocol: <https://developer.mozilla.org/en-US/docs/Web/HTTP>

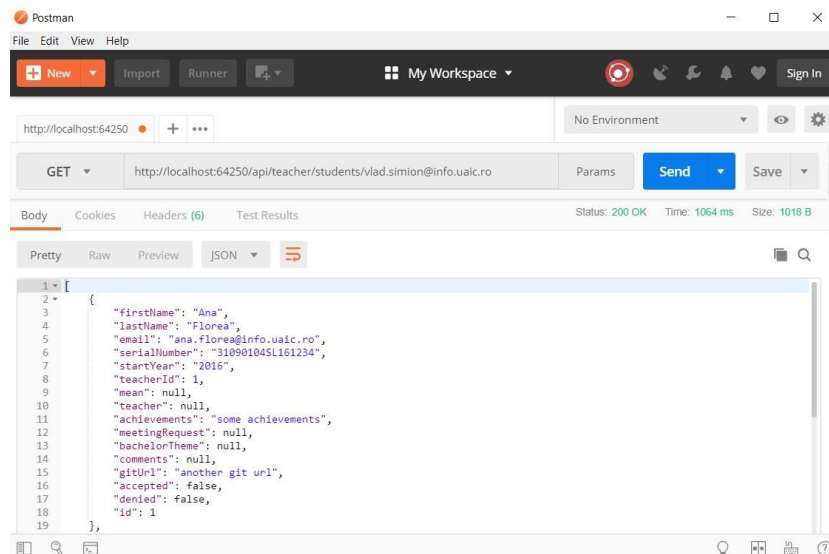


Figura 8: Cerere de tip GET și răspuns de la server, cu ajutorul Postman

SourceTree este o aplicație care joacă rolul de client pentru sistemul de versionare Git⁹. Lucrul cu sistemul de versionare este deci ușurat și încurajat cu ajutorul acestei interfețe grafice. Două din avantajele majore ale folosirii acestora este siguranța pe care o oferă în stocarea codului sursă (și a orice altor documente) și posibilitatea folosirii oricărei versiuni anterioare de cod.

2. Contribuții

Contribuția personală în realizarea prezentei lucrări a constat în alegerea temei, propunerea ei către domnul profesor coordonator și mai apoi discutarea ideilor. Petrecând niște ani superbi în cadrul studenției mi-am propus ca lucrarea mea de licență să poată fi baza unei aplicații mai generice care să aibă scopul de a ajuta studenții. În acest sens m-am gândit inițial la o platformă pentru organizarea și gestionarea materialelor aferente cursurilor și seminariilor predate în cadrul facultății dar, având în vedere că domnii profesori deja folosesc anumite spații pentru a-și pune resursele la îndemâna studenților, m-am reorientat spre un tip de aplicație ce nu există încă în cadrul facultății și nici măcar în cadrul universității.

Pentru realizarea obiectivului, am realizat o aplicație de tip client-server care are următoarea componență:

⁹ Detalii Git: <https://www.atlassian.com/git/tutorials/what-is-git>

- Aplicația server: reprezentată sub forma unui API REST care expune extern informații, prin intermediul serviciilor.
- Aplicația client: reprezentată de o interfață grafică, ce utilizează serviciile oferite de către aplicația server, apelându-le prin intermediul protocolului HTTP, și le expune informațiile obținute către utilizatori.

Prin utilizarea unei multitudini de tehnologii moderne (spre exemplu Angular 4 și .NET core 2.0), respectarea bunelor practici atât în arhitectura cât și în implementarea aplicației și prin ușurința cu care aceasta poate fi extinsă, am reușit să creez o aplicație care să ajute studenții și chiar să îi provoace în a continua munca începută de mine.

3. Proiectare

Lucrarea de licență va conține capitole numerotate crescător, fiecare putând să aibă, în partea finală, o secțiune de concluzii, care să sintetizeze informațiile și/sau rezultatele prezentate în cadrul aceluși capitol.

1. Arhitectura soluției

descriere pagini, diagrame uml,

2. Modelare datelor

diagrama db + explicații și relații

3. Comunicarea server-client

4. Interfața cu utilizatorul

screen shots și explicații pentru diferite scenarii

screen shots on laptop/phone size

Concluzii

cele mai importante concluzii din lucrare,

opinia personală privind rezultatele obținute în lucrare

potențiale direcții viitoare

Bibliografie

lista tuturor surselor de informație utilizate