

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI
FACULTATEA DE INFORMATICĂ



LUCRARE DE LICENȚĂ
eSIMS Bot - Utilitar pentru situația universitară

Propusă de
Bogdan-Gabriel Mihai

Sesiunea: **februarie, 2018**

Coordonator științific
Lector, dr. Mădălina Răschip

UNIVERSITATEA "ALEXANDRU IOAN CUZA" DIN IAȘI
FACULTATEA DE INFORMATICĂ

eSIMS Bot - Utilitar pentru situația universitară

Bogdan-Gabriel Mihai

Sesiunea: **februarie, 2018**

Coordonator științific

Lector, dr. Mădălina Răschip

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „eSIMS Bot - Utilitar pentru situația universitară”, este scrisă de mine și nu a mai fost prezentată niciodată la o altă facultate sau instituție de învățământ superior din țară sau din 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, 13.02.2018

Absolvent Bogdan-Gabriel Mihai

(semnătura)

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „eSIMS Bot - Utilitar pentru situația universitară”, 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” din 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, 13.02.2018

Absolvent Bogdan-Gabriel Mihai

(semnătura)

Cuprins

Cuprins	4
Terminologie	6
Introducere	7
Ce este un chatbot?	7
Motivația	8
Capitolul I: Chatbot-uri	11
Evoluția chatbot-urilor	11
De ce acest interes în chatbot-uri?	12
Tipuri de chatbot-uri	13
Potrivire de șabloane	13
Algoritmi de învățare automată pentru clasificarea textului	15
Clasificarea textului utilizând rețele neuronale	16
Messenger Chatbot	17
Capitolul II: Web Scraping	18
Ce este?	18
Este legal?	18
Cum este folosit în eSIMS Bot?	19
Capitolul III: Proiectarea aplicației	20
Tehnologii	20
Firebase	20
Heroku	20
Git	21
Facebook webhooks	21
MomentJS	22
Lodash	23
Async	23
CheerioJs	24
Cheerio TableParser	25
Capitolul IV: Aplicația	25
Comunicarea	25
Autentificarea cu eSIMS	28
Folosirea aplicației	31

Idei de extindere	38
Bibliografie	40

Terminologie

În redactarea acestor pagini, am fost nevoit să folosesc anumiți termeni specifici acestui domeniu. Pentru unii termeni, am preferat să le păstrez forma acceptată de comunitate, care este în limba engleză. Explicația lor o veți găsi mai jos.

- *Chatbot*: Este un serviciu bazat pe reguli sau, câteodată, pe inteligență artificială, pe care îl accesezi printr-o interfață de mesaje. El poate fi încorporat într-o aplicație de chat existentă (Facebook, Slack, Skype...).
- *Webhook*: sau *web callback*, este o metodă prin care o aplicație poate trimite altor programe diverse informații, în timp real.
- *Messenger*: Aplicația de mesagerie folosită de platforma Facebook.
- *Web scraping*: Procesul prin care se pot extrage programatic diverse informații de pe un website, descompunându-i pagina HTML și preluând conținutul, indiferent de natura acestuia (text, imagine, video).

Introducere

În zilele noastre, chatbot-urile au câștigat teren considerabil pe teritoriul WWW¹, fiind acum o utilitate (și, de ce nu, uneori chiar o necesitate) pentru diverse aplicații și servicii Web. Îl putem identifica sub mai multe forme și denumiri, cum ar fi asistent sau agent virtual. Se pare că previziunile marilor vizionari cum ar fi Nikolas Tesla² s-au adevărit, anticipând încă dinaintea erei calculatoarelor că omul va putea interacționa cu mașina, că se va ajunge la stadiul de interpretare și asimilare rațională a mesajelor transmise de oameni, venind în schimb cu răspunsuri sau chiar recomandări pertinente. Închipuirile acestea mai mult sau mai puțin fantasmagorice s-au concretizat într-o formă neașteptat de elegantă și capabilă, prin interacțiuni om-mașină limitate pentru ca mai apoi să apară primul robot capabil să interpreteze textul și, în cele din urmă, prin definitivarea conceptului de chatbot.

Ce este un chatbot?

Chatbot-ul este un program care mimează conversația dintre oameni, folosindu-se de inteligența artificială. Conceptul de chatbot este destul de vechi, fiind introdus de către Alan Turing, când a propus testul numit *The imitation game* [2]. În forma lui inițială, jocul presupunea trei participanți: un bărbat, o femeie și încă o persoană numită judecător/judecătoare, situați în trei camere diferite, conectați printr-un calculator. Rolul judecătorului era să își dea seama care dintre cei doi oameni cu care vorbea prin calculator era bărbatul. Apoi, Turing a propus o modificare jocului: în loc să participe un bărbat și o femeie, acum va participa un om (indiferent de sex) și un calculator. Astfel, scopul judecătorului a devenit să își dea seama când vorbește cu omul și când vorbește cu calculatorul. S-a convenit că, dacă judecătorul nu își poate da seama când vorbește cu ființa umană în mai mult de 50% din cazuri, calculatorul poate fi perceput ca

¹ World Wide Web

² "Innumerable activities still performed by human hands today will be performed by automatons. At this very moment scientists working in the laboratories of American universities are attempting to create what has been described as a 'thinking machine.' I anticipated this development. I actually constructed 'robots.' Today the robot is an accepted fact, but the principle has not been pushed far enough. In the twenty-first century the robot will take the place which slave labor occupied in ancient civilization. There is no reason at all why most of this should not come to pass in less than a century, freeing mankind to pursue its higher aspirations." [1]


un înlocuitor al omului. Astăzi, jocul a mai suferit o schimbare și nu mai presupune trei participanți, ci doi: judecătorul și omul/calculatorul.

Motivația

Trăim într-o eră care este caracterizată prin accesibilitate, inovație și optimizare a tot ceea ce vizează informația, o eră în care viteza cu care circulă informația a ajuns la cote maxime, o eră în care utilizatorul vrea să facă orice, acum, și cu minimum de efort. Acest lucru a devenit incontestabil valabil și zona noastră, cu precădere în rândul tinerilor. Nu este o noutate că oricine a recurs la procurarea și utilizarea zilnică a smartphone-urilor pentru activități de divertisment, dar mai ales pentru nevoi de bază (obținerea de informații atât din apeluri, mesaje, cât și prin intermediul internetului). Carduri bancare, notițe, bilete de acces, poze, persoane de contact – acestea sunt doar câteva elemente pe care cu toții le stocăm pe internet. Securizarea și autentificarea se realizează cu ajutorul amprenteii sau chiar a ochilor în câteva milisecunde, conectarea la unele platforme putând fi făcută prin introducerea unui cod de câteva cifre primit prin SMS, având la dispoziție o suită întreagă de metode pentru potențiala recuperare a unei parole uitate. S-a reușit afișarea oricărei informații, de la ecrane de câțiva metri, până la display-ul unui telefon sau chiar al unui ceas.

Acesta este contextul actual, context în care inclusiv studenții de regăsesc și își desfășoară activitățile, chiar și în cadrul facultății. Din păcate însă, în acest context, una din verigile în care am întâmpinat numeroase dificultăți o reprezintă procesul de aflare a situației universitare. Mai exact, la momentul aflării notelor, a eventualelor taxe de achitat sau chiar a orarului de curs, prin intermediul eSIMS.

De-a lungul anilor de facultate, întotdeauna mi s-a părut destul de dificil și neintuitiv folosirea platformei eSIMS.



universitatea alexandru ioan cuza iași
departamentul de statistică și informatizare

Logout «31090104SL130210»

Pagina principală

Studenti

Recuperare parolă

Noutăți
» de interes general

Legături rapide
» Wheather, 1
» Utile, loco
» MEC
» Bologna
» Euro Univs, Top 100
» Euro Univs, All
» USA Univs
» Canada Univs
» World Univs OL
» Dictionaries
» Sfatul medicului

Componentă Web destinată studenților.

- Înregistrarea se poate face doar cu NUMARUL MATRICOL având structura standard: cel puțin 16 caractere alfanumerice!


Pentru cei interesați!
La fiecare 15 minute sunt radiate conturile blocate.

- NU rezolvăm prin telefon probleme de conectare la eSIMS!
- Echipa eSIMS asigură funcționarea siteului. Studenții își creează, activează, blochează contul, recuperează parola etc.
- Îndemnul "Mergeți la eSIMS!" luați-l drept "Conectați-vă la eSIMS!" Cei care preferă să meargă la eSIMS beneficiază de radierea, instant, a contului.
- după 6 luni de neutilizare a contului are loc radierea lui.
- după un număr apreciabil de tentative de conectare eșuate contul este blocat, analizate cauzele eșuării și apoi radiat.
- tentativele "off road" de înregistrare/conectare vor fi tratate cu atenție.
- utilizați adrese de email active unde să primiți mesajul cu coordonatele de conectare. Dacă nu primiți mesajul, încercați o conectare după 15 minute de la crearea contului.
- creați-vă personal înregistrarea și nu via colegi, mătuși etc. Folosiți Internet Explorer, Mozilla ...
- luați în serios întrebarea de securitate, dar mai ales răspunsul!
- înregistrările cu adrese de email invalide/inactive sunt radiate din bază.

După radiere se poate face alt cont.

eSIMS

« »


» ...

Având în vedere că scopul aplicației este de a vedea situația notelor, restanțelor și a taxelor, ele sunt ascunse și greu accesibile, fiind dispuse la distanță de câteva click-uri bune. În plus, informația este structurată în mod ambiguu și deloc ușor de asimilat, întrucât pe o singură pagină sunt numeroase tabele (din cauza acestei dispunerii a informației, este foarte greu de găsit un anumit semestru, nota la o materie, suma care trebuie plătită vs. suma care a fost deja achitată etc.), iar navigarea între semestre sau ani de studiu nu este deloc intuitivă.

Traectoria studentului...					
an scolar	an studiu	sem.	grupa	specializare	
2013	1	1	Semian A\A3	Informatică	[detalii]
2013	1	2	Semian A\A3	Informatică	[detalii]
2014	2	3	Semian A\A3	Informatică	[detalii]
2014	2	4	Semian A\A3	Informatică	[detalii]
2015	3	5	Semian A\A3	Informatică	[detalii]
2015	3	6	Semian A\A3	Informatică	[detalii]

Detalii despre student	
ID	326484
Nume	MIHAI L. BOGDAN-GABRIEL
Matricol	31090104SL130210
Telefon	728859561
e-mail	bogdan.mihai@info.uaic.ro
Data nasterii	22-Oct-94
mama	SILVIA
tata	LIVIU
nationalitatea	Română
cetatenia	Română

Notele studentului						
	AnUniv	Semestru	Denumire disciplina	Nota finala	Credite	Data
[detalii]	2013	1	Arhitectura calculatoarelor și sisteme de operare	8.00	5.00	21.01.2014
[detalii]	2013	1	Introducere în programare	9.00	5.00	22.01.2014
[detalii]	2013	1	Structuri de date	8.00	5.00	22.01.2014
[detalii]	2013	1	Psihologia educației	7.00	5.00	24.01.2014
[detalii]	2013	1	Logică pentru informatică	6.00	5.00	24.01.2014
[detalii]	2013	1	Matematică	7.00	5.00	25.01.2014
[detalii]	2013	1	Limba engleză I	10.00	5.00	23.01.2014

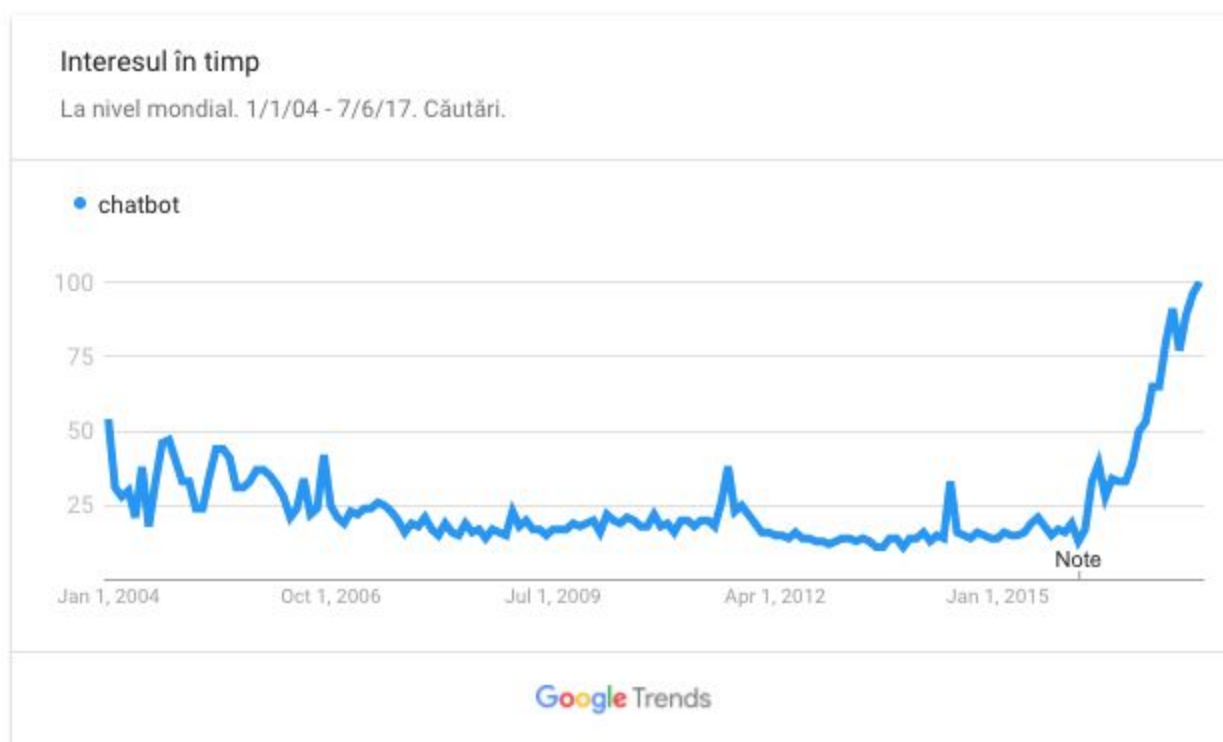
Un alt aspect notabil în care am identificat dificultăți îl reprezintă conectarea la platformă. Aceasta făcându-se cu numărul matricol (un cod greu de memorat și de găsit în alte părți), însoțit de o parolă pe care nu o poți reseta, ea fiind trimisă în clar de către administratorul sistemului, atunci când acesta vede solicitarea.

Gândindu-mă că e dificilă crearea unei noi platforme, sau regândirea celei actuale, și coroborat cu apariția platformei create de către Facebook, am decis să construiesc peste eSIMS un chatbot care să vină în ajutorul studenților care vor să-și afle situația anului universitar, prin interogarea în chat a unui set de informații prin câteva comenzi predefinite, vizibile și ușor de utilizat, în timp ce modalitatea de conectare poate fi facilitată. Pe lângă funcționalitățile de bază, aflarea notelor, taxelor, restanțelor, am considerat că poate fi util și integrarea orarului. Așadar, studentul poate ști și orarul de astăzi, de mâine, sau pe toată săptămâna.

Capitolul I: Chatbot-uri

Evoluția chatbot-urilor

În acest moment, chatbot-urile reprezintă un subiect destul de actual în mediul online. Se pare că ele au devenit obiect de interes pentru companii precum Facebook, Microsoft, Amazon, ele urmărind revoluționarea modului de comunicare, de transmitere a mesajelor dar și de a obține informații printr-o modalitate mai rapidă, ele fiind chiar capabile să rezolve anumite task-uri, să rețină date despre utilizator și să îi propună sugestii.



Sursă imagine: <http://blog.aylien.com/interactive-history-chatbots/>

De ce acest interes în chatbot-uri?

Din punctul meu de vedere, apariția acestui curent în ultime vreme provine dintr-o dorință de resuscitare a inteligenței artificiale și de utilizare a acesteia în industrie, dar și din obsesia noastră de a avea servicii care funcționează permanent. Un studiu recent³ arată că 44% din consumatorii din SUA preferă să folosească un chatbot în detrimentul conversației cu un om, iar 61% din cei care au luat parte la acest studiu folosesc deja un chat cel puțin o dată pe lună. Acest fapt se datorează disponibilității non-stop a chatbot-ului și vitezei cu care poate răspunde la întrebări.

Mari companii au anticipat acest moment și deja pun la dispoziție aplicații de mesagerie și asistenți cât mai „inteligenți” pentru a oferi o experiență plăcută utilizatorilor, dar și pentru optimizarea și îmbunătățirea calității serviciului de relații cu clienții, întrucât acest serviciu permite soluționarea multor probleme comune care apar la mai mulți utilizator, acest lucru întâmplându-se instant și natural, fără existența vreunui timp de așteptare sau a unei persoane desemnată să investigheze situația. Acest lucru poate fi pus în practică mai ales de când Facebook a lansat Messenger Chatbot⁴. Deși acest concept a crescut mai mult în ultima perioadă, el nu este deloc nou, fiind prezent încă de acum 50 de ani în comunitatea NLP (*Natural Language Processing*) și nu este decât o etapă în atingerea adevăratului scop: înțelegerea limbajul uman de către calculator.

De-a lungul timpului [2], au fost construite mai multe astfel de chatbot-uri:

- 1966: Eliza –A fost primul chatbot creat vreodată, bazat pe potrivire de șabloane, dând astfel iluzia de empatizare cu utilizatorul, dar nu avea capacitatea de a înțelege contextul;
- 1972: Perry –A fost considerat drept „Eliza cu atitudine”. Simula un personaj cu schizofrenie;
- 1995: A.L.I.C.E (Artificial Linguistic Internet Computer Entity) – A fost un robot care procesa limbajul natural;

³https://www.aspect.com/globalassets/2016-aspect-consumer-experience-index-survey_index-results-final.pdf

⁴ <https://messenger.fb.com/>

- 2010: Siri –Asistent personal care face parte din sistemul de operare iOS. Se folosește de procesarea limbajului natural pentru a răspunde la întrebări și pentru a face căutări. A pus bazele la tot ceea ce urmează a fi considerat mai târziu un asistent personal;
- 2012: Google Now –La fel ca Siri, procesează limbajul natural și se folosește de servicii web pentru a da răspunsuri și recomandări. El este încorporat în aplicația de căutare Google.
- 2016: Chatbot-uri pentru Facebook Messenger – În aprilie 2016, Facebook a lansat o platformă destinată programatorilor, cărora le permite să interacționeze cu utilizatorii prin intermediul ferestrei de convorbire din Facebook. Până în prezent, s-au dezvoltat peste 11.000 de chatbot-uri create cu ajutorul acestei platforme.

Tipuri de chatbot-uri

Clasificarea *chatbot*-urilor [4] se poate realiza în funcție de maniera în care interpretează textul. În mod abstract, el primește un text de la utilizator pe care îl procesează cu ajutorul unei funcții care se numește *classifier*. Această funcție atașează propoziției un *intent*, iar pe baza acestuia se formează un răspuns. În funcție de acest *classifier*, putem diferenția trei tipuri de chatbot-uri:

Potrivire de șabloane

Primele chatbot-uri foloseau această tehnică pentru a clasifica textul, dar și pentru a produce un răspuns. Mai este cunoscută sub numele de „forță brută” deoarece creatorul chatbot-ului trebuie să introducă manual fiecare șablon și semnificația lui. Pentru construcția acestor șabloane se folosește limbajul AIML⁵.

⁵ Artificial Intelligence Markup Language

Exemplu de șablon:

```
<aiml version = "1.0.1" encoding = "UTF-8"?>
  <category>
    <pattern> WHO IS ALBERT EINSTEIN </pattern>
    <template>Albert Einstein was a German physicist.</template>
  </category>

  <category>
    <pattern> WHO IS Isaac NEWTON </pattern>
    <template>Isaac Newton was a English physicist and
mathematician.</template>
  </category>

  <category>
    <pattern>DO YOU KNOW WHO * IS</pattern>
    <template>
      <srai>WHO IS <star/></srai>
    </template>
  </category>
</aiml>
```

Sursă imagine: https://medium.com/@gk_/how-chat-bots-work-dfff656a35e2

Răspuns:

Human: Do you know who Albert Einstein is
Robot: Albert Einstein was a German physicist.

Sursă imagine: https://medium.com/@gk_/how-chat-bots-work-dfff656a35e2

Algoritmi de învățare automată pentru clasificarea textului

Mecanismul „forță brută” are un dezavantaj: pentru fiecare situație, trebuie să existe un șablon care să îi permită chatbot-ului clasificarea propoziției utilizatorului. Din această cauză, a apărut un alt tip de clasificare, cea prin algoritmi de Învățare Automată. Acest lucru se poate face cu ajutorul unei ecuații:

$$\hat{P}(t|c) = \frac{T_{ct} + 1}{\sum_{t' \in V} (T_{ct'} + 1)} = \frac{T_{ct} + 1}{(\sum_{t' \in V} T_{ct'}) + B'}$$

Sursă imagine: https://medium.com/@gk_/how-chat-bots-work-dfff656a35e2

Formula de mai sus este mai simplă decât pare. Algoritmul presupune existența unui set de propoziții asiguate unei clase. Dată o nouă propoziție, putem extrage fiecare cuvânt și putem număra aparițiile acestuia în fiecare clasă. De asemenea, pentru fiecare cuvânt, ar trebui să avem un scor asociat, deoarece, spre exemplu, prepozițiile pot fi mult mai des întâlnite decât cuvinte ca „vreme”, „stare”, „preponderență” ș.a.m.d. În forma mai extinsă a algoritmului, înainte de a căuta cuvintele în setul de bază, ele trebuie aduse la forma lor inițială, fără derivări sau acorduri. Exemplu set de învățare:

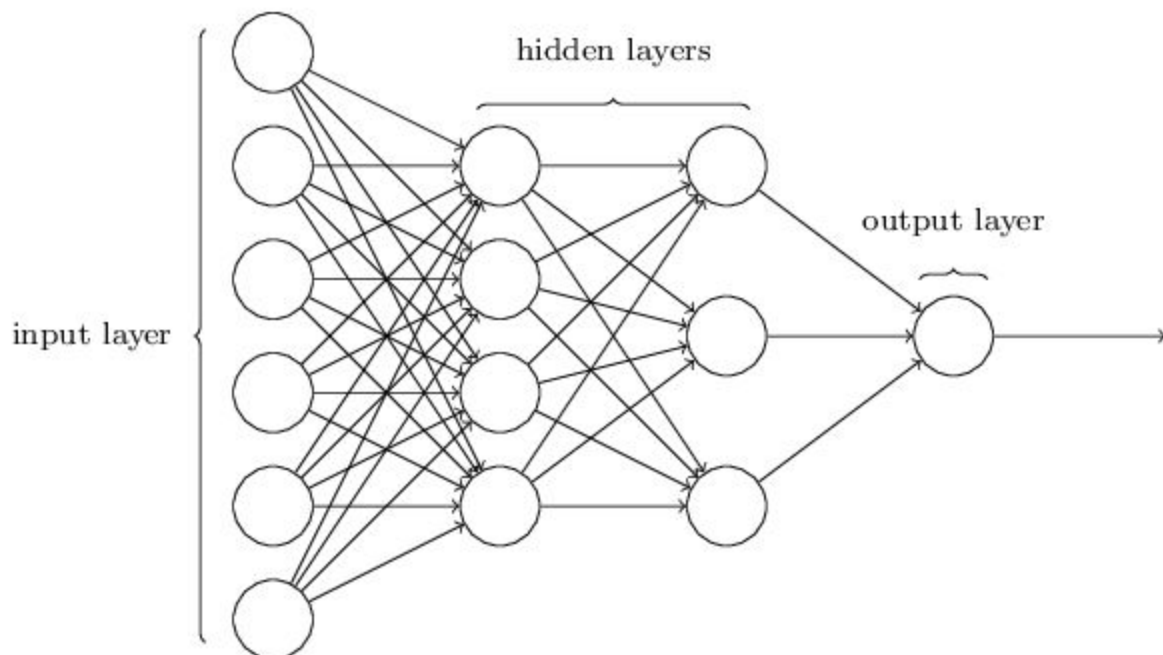
```
class: weather
    "is it nice outside?"
    "how is it outside?"
    "is the weather nice?"
class: greeting
    "how are you?"
    "hello there"
    "how is it going?"
```


Clasificarea unei propoziții:

```
input: "Hi there"
term: "hi" (no matches)
term: "there" (class: greeting)
classification: greeting (score=1)
input: "What's it like outside?"
term: "it" (class: weather (2), greeting)
term: "outside" (class: weather (2) )
classification: weather (score=4)
```

Clasificarea textului utilizând rețele neuronale

Rețelele neuronale au fost inventate în 1940, ca o modalitate de a oferi un răspuns pe baza unor date de antrenament. Ele calculează răspunsul cu ajutorul unor conexiuni care au asociate un scor, numite sinapse. Fiecare iterație a rețelei produce o schimbare a acestor scoruri și conduce, în acest fel, la un răspuns cu acuratețe mai mare.



Sursă imagine: https://medium.com/@gk_/how-chat-bots-work-dfff656a35e2

Ceea ce știm despre ele nu s-a schimbat, îmbunătățirea venind din faptul că puterea de procesare a informațiilor a crescut. În acest domeniu, viteza și puterea de calcul sunt cruciale, ele fiind folosite la înmulțirea de matrici. Asemănător metodei precedente, fiecare clasă are un număr de propoziții asociat. Din nou, fiecare propoziție este împărțită în cuvinte, care devin date de intrare pentru rețea. Ponderile sinaptice sunt calculate iterând de mii de ori, de fiecare dată ajustându-se pe baza pasului anterior. Aceste scoruri reprezintă o măsură a importanței informației: cu cât a fost accesată de mai multe ori, cu atât e mai importantă. Similar cu viața reală, cu cât ai văzut un lucru de mai multe ori, cu atât este mai probabil să ți-l amintești.

Messenger Chatbot

În aprilie 2016, Facebook a lansat posibilitatea de a programa chatbot-uri chiar în interiorul platformei lor. În momentul de față ea reprezintă cea mai folosită platformă pentru a crea astfel de servicii, câștigând avans față de ceilalți competitori (Microsoft, Slack) datorită fluxului mare de utilizatori pe care îl are, dar și a multitudinii de funcționalități pe care le pune la dispoziție. Iată câteva dintre ele:

- Identificarea utilizatorului după un ID unic;
- Conectarea cu aplicații externe;
- Salvarea fișierelor atașate, în vederea utilizării lor pe viitor (pentru a salva timp);
- Efectuare de plăți direct din chatbot;
- Procesare de limbaj natural și clasificarea replicilor în peste 20 de limbi;
- Încărcarea de pagini web în interiorul ferestrei de conversație.

Capitolul II: Web Scraping

Ce este?

[5] Procesul automat de colectare a datelor de pe Internet este aproape la fel de vechi ca însuși Internetul. Cu toate că termenul *web scraping* nu este nou, el este mai mult cunoscut sub numele de *screen scraping*, *data mining* sau *web harvesting*. În teorie, *web scraping* se referă la metoda automată de a colecta diferite informații prin alte mijloace decât cele uzuale: printr-un API⁶ sau prin navigarea într-un browser. Acest lucru este posibil prin crearea unui algoritm care execută interogări unei pagini HTML și extrage informațiile de care are nevoie, indiferent de natura lor (text, imagini, video, script-uri).

Este legal?

[6][7] Cred că este destul de dificil să afirmăm concret dacă este legal sau nu. Teoretic, dacă informațiile se află în spațiul public, nu ar trebui să fie nicio problemă în a le salva la noi în calculator. Însă, de-a lungul timpului au fost cazuri în care oamenii au fost dați în judecată pentru că au făcut *web scraping*:

- În cazul LinkedIn Corporation⁷ vs. Robocog Inc, Robocog Inc a fost somat să plătească 40.000\$ pentru extrageri neautorizate de conținut din LinkedIn.
- În cazul Facebook vs. Pete Warden, avocatul trimis de Facebook l-a amenințat pe domnul Warden că îl va da în judecată dacă va publica seturile de date rezultate din colectarea a milioane de informații despre profilurile utilizatorilor

Având în vedere aceste antecedente, cred că întrebarea capătă mai degrabă conotații etice decât juridice, și anume, cum plănuim să folosim informațiile obținute în urma acestei practici. După cum am mai spus, având în vedere că informația se găsește în spațiul public, nu ar trebui să constituie o problemă salvarea ei, dar în cazul în care ne însușim informația și o distribuim fără a

⁶ *Application Programming Interface*

⁷https://www.docketalarm.com/cases/California_Northern_District_Court/5--14-cv-00068/LinkedIn_Corporation_v._Robocog_Inc/docs/8.pdf

credita sursa, s-ar putea să fie greșit și chiar ilegal, în cazul în care există drepturi de autor asumate.

De asemenea, având în vedere că extragerea de date folosind un algoritm este mult mai rapidă și mai eficientă decât cea prin accesare și preluare manuală, trebuie să avem grijă ca procesul de *web scraping* să nu afecteze performanța site-ului care este interogat. În cele mai multe cazuri, serverele vor bloca automat IP-ul și vor restricționa accesul la pagină.

Cum este folosit în eSIMS Bot?

În cadrul aplicației, *web scraping*-ul este folosit pentru conectarea cu platforma eSIMS și afișarea informațiilor cerute de către utilizator. Ele nu vor fi stocate în nicio bază de date, ci vor persista doar în istoricul conversației, putând fi șterse de către utilizator oricând.

Capitolul III: Proiectarea aplicației

Tehnologii

În realizarea aplicației, am folosit cu preponderență ca limbaj de programare **JavaScript**, întreaga logică de manipulare a datelor fiind realizată pe partea de back-end (la care se adaugă utilizarea **NodeJS**). Am mai folosit, pentru interfața de autentificare, **HTML** și **CSS3**.

1. Firebase

Firebase⁸ este o platformă oferită de Google care stochează și sincronizează datele unei aplicații (iOS, Android sau Web) cu o bază de date în *cloud* de tip NoSQL. Datele sunt sincronizate în timp real și pot rămâne disponibile și *offline*. Ele sunt stocate sub forma unui JSON.

```
licenta-ff387
├── users
│   └── 1435516829841173
│       ├── batch: "A"
│       ├── credentials: "{ \"username\": \"e351d94814f1782773f42a1da63261bffa...\" }"
│       ├── group: "1"
│       └── year: "1"
```

2. Heroku

Hosting-ul și *deploy*-ul (automat, prin intermediul Git) au fost realizate pe platforma Heroku⁹, foarte cunoscută pentru ușurința de utilizare și configurare (este specifică și recomandată *startup*-urilor).

⁸ <https://firebase.google.com/>

⁹ <https://www.heroku.com/>

3. Git

Pentru versionare, *deploy* și *remote-tracking*, am folosit Git¹⁰.

4. Facebook webhooks

Platforma Messenger trimite evenimente către webhook-ul nostru pentru a notifica robotul dacă apar anumite schimbări, spre exemplu când un utilizator trimite un mesaj. Evenimentele sunt trimise de către platformă sub forma unui *request* de tip *POST*, către webhook-ul setat în aplicație. Iată evenimentele gestionate de chatbot:

Eveniment	Scop
<i>messages</i>	Indică prezența unui mesaj nou
<i>messaging_account_linking</i>	Apare când utilizatorul inițiază o acțiune de autentificare sau deconectare.

Secvența de cod care setează webhook-ul și se ocupă cu tratarea evenimentelor:

```
router.post('/webhook', function (req, res) {  
  var data = req.body;  
  if (data.object === 'page') {
```

¹⁰ <https://github.com/>

```

data.entry.forEach(function(entry) {
  entry.messaging.forEach(function(event) {
    if (event.message) {
      if (event.message.quick_reply) {
        fb_bot_service.handleQuickReply(event);
      } else {
        fb_bot_service.matchMessage(event);
      }
    }
    } else if (event.account_linking) {
      if (event.account_linking.status === 'linked') {
        var newUser = {};
        newUser[event.sender.id] = { credentials:
event.account_linking.authorization_code };

        firebase.database.ref('users').set(newUser);
      } else {

firebase.database.ref('users').child(event.sender.id).remove();
      }
    } else {
      console.log("Webhook received unknown event: ", event);
    }
  });
});
res.sendStatus(200);
});
});

```

5. MomentJS

Este utilizat în cadrul aplicației pentru identificarea orarului, atunci când utilizatorul dorește orarul de astăzi, sau de mâine. Folosindu-l, reușesc să accesez denumirea zilei. Exemplu de utilizare:

```
moment().format('ddd')
```

6. Lodash

Lodash¹¹ este un utilitar JS care facilitează operații pe *array*-uri, obiecte, *string*-uri sau numere. M-a ajutat la iterarea *array*-urilor și la transformarea unor obiecte:

```
result = _.transform(result, function(result, value, key) {  
    result.push({day: key, schedule: value});  
}, []);
```

7. Async

De cele mai multe ori, programatorii sunt obligați să scrie cod care se va executa asincron în Javascript. Până nu demult, de fiecare dată se ajungea la așa numitul *callback hell*, termen folosit atunci când prelucrarea unor date era condiționată de finalizarea unor acțiuni (colectarea de date prin servicii web, generarea unor statistici, etc.). În aplicație, ordinea răspunsurilor este importantă, mai ales în secțiunea de orar, acolo unde programul trebuia afișat în ordinea zilelor săptămânii și a orelor. Aici a intervenit Async.js, un modul utilitar care oferă posibilitatea de a scrie cod sincron într-o manieră elegantă și simplu de înțeles. Se poate observa acest lucru în secvența de mai jos:

```
function showDaySchedule(senderID, obj) {  
    communication_service.sendMessage(senderID,  
    obj.day).then(function() {  
        async.eachSeries(obj.schedule, function iteree(line,  
        scheduleCallback) {  
            string = line.start + ' - ' + line.end + ' ' +  
            line.discipline + ' ' + line.type + ' ' + line.room;  
  
            communication_service.sendMessage(senderID,  
            string).then(function() {  
                scheduleCallback(null);  
            });  
        });  
    });  
}
```

¹¹ <https://lodash.com/>


```

        }, function done() {
            resolve();
        });
    });
}}

```

8. CheerioJs

Este o bibliotecă ce m-a ajutat în parsarea HTML-ului rezultat în urma procesului de web scraping. Secvență luată din preluarea notelor:

```

getMarks: function(semester, payload) {
    var marks = [];
    var url = 'http://simsweb.uaic.ro/eSIMS/Members/StudentPage.aspx';
    payload.__EVENTARGUMENT = 'Select$' + semester;
    var options = {
        method: 'post',
        form: payload,
        url: url
    };

    return new Promise(function(resolve, reject) {
        request(options, function(err, resp, body) {
            if (err)
                throw err;
            var $ = cheerio.load(body);
            var discipline =
                $('#ctl00_WebPartManagerPanel1_WebPartManager1_wp523396956_wp729632565_Grid
                ViewNote tr');
        })
    })
}

```

9. Cheerio TableParser

O extensie a bibliotecii Cheerio, care ajută la convertirea tabelelor din HTML în *tablouri* de obiecte.

```
$ = cheerio.load(body);  
var table = $('p:first-of-type').html();  
cheerioTableparser($);  
var data = $(table).parsetable(false, false, true);
```

Capitolul IV: Aplicația

Comunicarea

Fundamentul interacțiunii dintre un chatbot și un utilizator este schimbul de replici. Putem identifica, deci, mai multe sarcini ale robotului:

1. Recepționarea mesajelor – Se face pe baza webhook-ului setat. Aici, Facebook-ul trimite notificări oricând cineva scrie, încarcă o imagine sau o înregistrare vocală.
2. Clasificarea lor – Am considerat că, fiind un chatbot care are ca scop informarea cu privire la informații exacte despre utilizator, clasificarea mesajelor prin potrivire de șabloane este cea mai bună metodă de abordare. Ca mecanism, am atașat fiecărui șablon un sens, acesta urmând să fie interpretat pentru construcția unui răspuns. Tiparele pot fi văzute mai jos:

```
var regularExpressions = [
{
  regexp: /(note( +)an( +)\d( *)?( *)seimestru(1?)( +)\d)/gi,
  means: 'note_seimestru'
},
{
  regexp: /(note( +)an( +)\d)/gi,
  means: 'note_an'
},
{
  regexp: /(not(a|ă)( +)*\w+)/gi,
  means: 'nota'
},
{
  regexp: /(restante( +)an( +)\d)/gi,
  means: 'restante_an'
},
{
  regexp: /(restante)/gi,
  means: 'restante'
}
```

```

    },
    {
        regexp: /(taxe)/gi,
        means: 'taxe'
    },
    {
        regexp: /(orar( +)m(a|â)ine)/gi,
        means: 'orar_maine'
    },
    {
        regexp: /(orar( +)azi)/gi,
        means: 'orar_azi'
    },
    {
        regexp: /(orar( +)reset)/gi,
        means: 'orar_reset'
    },
    {
        regexp: /(orar)/gi,
        means: 'orar'
    },
    {
        regexp: /(login)/gi,
        means: 'login'
    },
    {
        regexp: /(logout)/g,
        means: 'logout'
    }
];

```

3. Prelucrarea răspunsului – După ce este atribuit un sens mesajului, acesta este preluat și acționat asupra lui. O parte din secvența căreia se datorează acest lucru:

```

switch (meaning) {
    case 'login':
        communication_service.sendLoginButton(senderID);
        break;
    case 'logout':
        communication_service.sendLogoutButton(senderID);
        break;
}

```

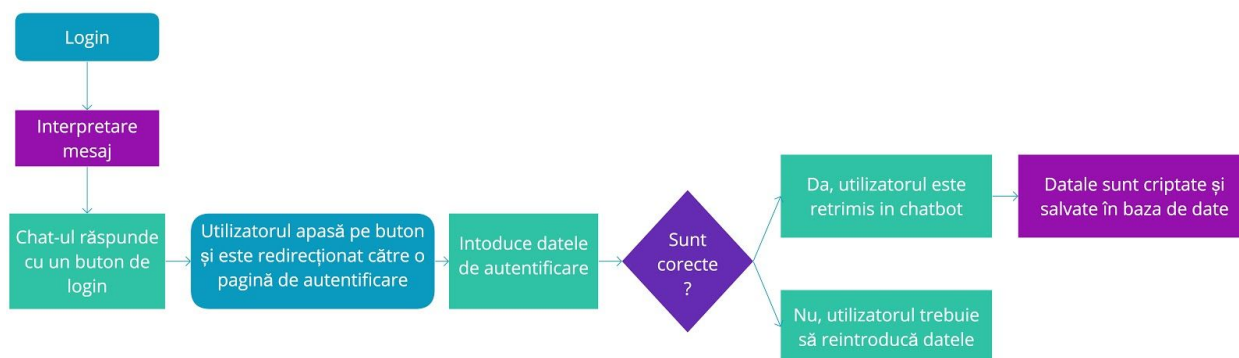
4. Trimiterea lui. În cadrul webhook-ului definit, pe lângă mesajul trimis de către utilizator, primim și un ID unic atașat. Pe baza acestuia, știm cui să răspundem. Pentru a trimite un mesaj înapoi, trebuie să efectuăm un apel către Facebook, atașând câțiva parametri, ca în secvența de mai jos:

```
callSendAPI: function(messageData) {
  return new Promise(function(resolve, reject) {
    request({
      uri: 'https://graph.facebook.com/v2.6/me/messages',
      qs: { access_token: PAGE_ACCESS_TOKEN },
      method: 'POST',
      json: messageData

    }, function (error, response, body) {
      if (!error && response.statusCode == 200) {
        var recipientId = body.recipient_id;
        var messageId = body.message_id;

        console.log("Successfully sent generic message with id %s
to recipient %s",
          messageId, recipientId);
        resolve();
      } else {
        console.error("Unable to send message.");
        console.error(response);
        console.error(error);
        reject(error);
      }
    });
  })
}
```

Autentificarea cu eSIMS



Fiind vorba despre o aplicație web cu circuit închis (studenții nu își pot vedea situația anului universitar decât dacă se vor autentifica), conectarea la platformă este un pas esențial. Am considerat că dacă la orice interacțiune cu chatbot-ul, utilizatorul va fi nevoit să introducă datele de conectare, aplicația își pierde scopul deoarece este un proces anevoios, id-ul de conectare fiind greu de reținut, iar fiind vorba de o aplicație mobilă, acest lucru va fi și mai deranjant. Am optat, așadar, către o autentificare *one-time*, înainte de orice interacțiune dintre *user* și robot. Până la pasul de implementare efectivă, am cercetat *request*-urile făcute de browser când un utilizator se conectează prin platformă. Am ajuns la concluzia că fiecare apel are atașat un set de parametri care pot fi împărțiți în două tipuri:

- Generați: Am observat în DOM-ul¹² site-ului că, de fiecare dată când cineva accesează pagina de conectare, o serie de attribute se generează automat de pe server.

¹² DOM - Document Object Model, este arborele atașat structurii HTML

```

<input type="hidden" name="__WPPSP" id="__WPPSP" value="s">
<input type="hidden" name="__LASTFOCUS" id="__LASTFOCUS" value>
<input type="hidden" name="ctl00_mainCopy_ScriptManager1_HiddenField" id="ctl00_mainCopy_ScriptManager1_HiddenField" value>
<input type="hidden" name="__EVENTTARGET" id="__EVENTTARGET" value>
<input type="hidden" name="__EVENTARGUMENT" id="__EVENTARGUMENT" value>
<input type="hidden" name="ctl00_subnavTreeView_ExpandState" id="ctl00_subnavTreeView_ExpandState" value>
<input type="hidden" name="ctl00_subnavTreeView_SelectedNode" id="ctl00_subnavTreeView_SelectedNode" value>
<input type="hidden" name="ctl00_subnavTreeView_PopulateLog" id="ctl00_subnavTreeView_PopulateLog" value>
<input type="hidden" name="__VIEWSTATE" id="__VIEWSTATE" value="/
wEPaA8FDzhkNTZkNjMxMmY2NzI3NkgBBS5fX0NvbnRyb2xzUmVxdWLyZV8vc3RCYWNrS2V5X18wBQUYY3RsMDAKTG9naW5TdGF0dXMxJGN0bDAXBRhjdGwMcRmb2dpb1N0YXR1czEkY3R5MDMF
IGN0bDAwJG1haW50b3B5JExvZ2luMSRSZW1lbWJlck1lBSZjdGwMcRtYwLUQ29weSRMb2dpbjEKTG9naW5JbWFnZUJ1dHRvbGUUY3RsMDAKc3VibmF2VHJlZXpZXdpYU0P/
PfflQhMPp62AZc/nRkq4Q==">

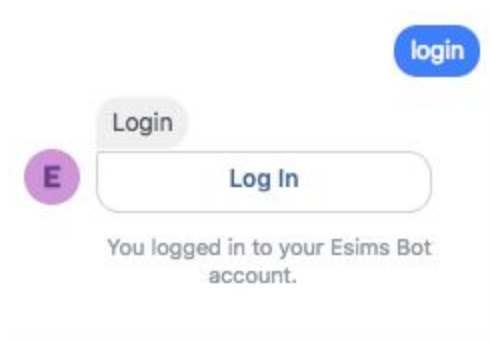
```

- Definiți de utilizator: Sunt cei pe care-i completează în formular (Numărul matricol, parola și, eventual, bifa de ținere minte).

Considerând afirmațiile de mai sus, procesul de autentificare va fi realizat în doi pași. În primul pas, se va face un simplu *request* de tip GET, pentru a colecta informațiile generate de server. Apoi, folosind informațiile obținute la pasul anterior și concatenat cu datele introduse de către utilizator, vom face un nou *request* de tip POST la link-ul

<http://simswb.uaic.ro/eSIMS/MyLogin.aspx>. Pentru a verifica dacă autentificarea a avut succes, analizăm rezultatul întors și vedem dacă în componența lui mai găsim formularul de introducere a datelor.

Procesul de autentificare începe o dată cu introducerea cuvântului „login” de către utilizator. În acest caz, chatbot-ul răspunde printr-un mesaj ce conține un buton, având atașat un link cu doi parametri: o adresă pentru *redirect* (folosit pentru a întoarce utilizatorul în aplicație) și un id unic). La apăsarea acestuia, utilizatorul este redirecționat într-o pagină unde găsește un formular cu două câmpuri - unul pentru introducerea numărului matricol și unul pentru introducerea parolei. După completarea câmpurilor și apăsarea butonului „Autentificare”, se execută pașii de colectare a datelor generate de server, și, în caz de succes, se preia *link*-ul din parametrul de *redirect*, i se atașează un *authorization_code* (un obiect JSON¹³ care conține numărul matricol și parola introdusă, ambele criptate) și se accesează adresa.



¹³ JSON - Javascript Object Notation

Această secvență are ca rezultat trimiterea unui mesaj de tip *messaging_account_linking* către webhook-ul setat de noi. Aici se preia *authorization code*-ul și se salvează în Firebase. De acum înainte, pentru fiecare cerere de la utilizator, se vor lua mai întâi datele din baza de date, se va face autentificarea, după care va fi posibilă extragerea de date din eSIMS.

Persistența conexiunii se datorează, pe de o parte, opțiunii `request = request.defaults({jar: true})`, care menține id-ul sesiunii pe apelurile web, și pe de altă parte funcției `keepConnectionAlive`, care are încapsulată toată logica de verificare a autentificării:

```
keepConnectionAlive: function (userID, request) {

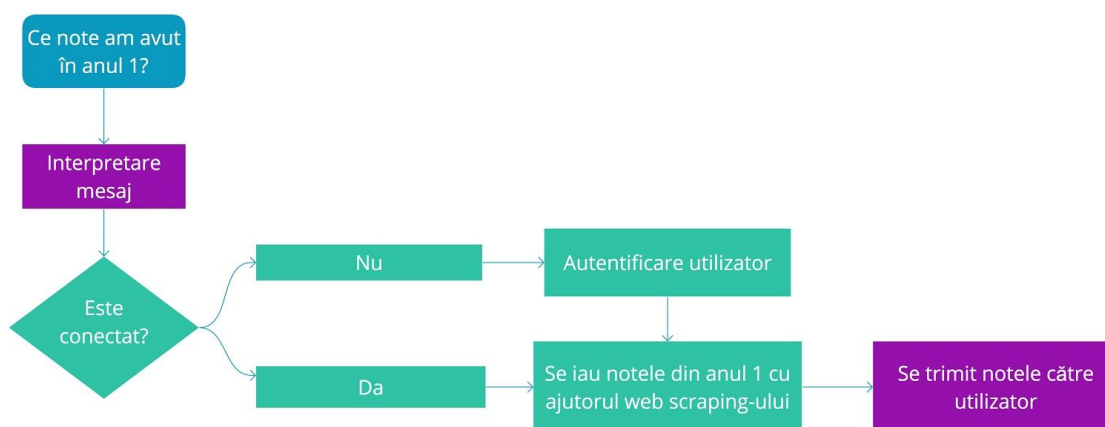
    module.exports.verifyIfLoggedIn(request).then(function(loggedIn) {
        if (!loggedIn) {
            firebase.database.ref('users/' + userID).once('value',
function(snapshot) {
                var user = snapshot.val();
                if (!user) {
                    communication_service.sendLoginButton(userID);
                    reject();
                } else {
                    var passKey = new
Buffer(process.env.JWT_SECRET).toString('base64');
                    var decrypted =
cryptoJSON.decrypt(JSON.parse(user.credentials), passKey, {
                        keys: []
                    });
                    module.exports.login(decrypted.username,
decrypted.password, request).then(function() {});
                }
            })
        })
    })
},
```

Pentru deconectare, va fi necesară comanda „logout”.

Folosirea aplicației

Chatbot-ul are patru funcții principale: vizualizarea notelor, a restanțelor, a taxelor și a orarului. Fiecare funcționalitate poate fi accesată prin intermediul unor șabloane ce trebuie să existe în replicile utilizatorului. Vom prelua spre analiză fiecare funcționalitate în parte:

1. Notele



Înainte de implementare, am studiat puțin pagina eSIMS-ului care se ocupă de gestionarea și afișarea notelor. Ulterior, am ajuns la concluzia că acestea nu sunt grupate pe ani și semestre, ci doar pe semestre, de la 1 la 6. La fel ca în cazul autentificării, o dată cu inițierea *web scraping*-ului trebuie trimisă o serie de parametri, fiind nevoie de un *request* în plus pentru a colecta parametri generați de către server, la care se adaugă și parametrul `__EVENTARGUMENT`, cel care indică numărul semestrului ale cărui note dorim să le accesăm. Prin urmare, am delimitat trei modalități de a cere notele – notele pe ani, pe semestre sau note individuale pe materii – care au condus la următoarele șabloane:

- note an X: Oferă toate notele, la toate materiile din anul X.

```
case 'note_an':  
    var year = parseInt(params[0]);  
    if (validateYear(year)) {  
        auth_service.keepConnectionAlive(senderID, request)  
            .then(function() {
```

```

marks_service.scrapeMarks(senderID, year, [(year * 2) - 2,
(year * 2) - 1], request);
    })
  } else {
    communication_service.sendTextMessage(senderID, errors.year);
  }
  break;

```

Codul de mai sus este creat în scopul de a colecta toate notele dintr-un an și afișarea acestora utilizatorului. În primul rând, se validează anul introdus (el trebuie să fie 1, 2 sau 3), după care se reface autentificarea cu eSIMS-ul. Aici, se verifică dacă *user*-ul este deja conectat, iar în caz contrar, i se iau datele de conectare din Firebase și se începe procesul de autentificare. După ce ne-am asigurat că nu a expirat sesiunea, începem procesul de colectare a notelor, specificând semestrelor pe care le dorim.

```

scrapeMarks:function(senderID, year, semesters) {
  async.eachSeries(semesters, function semesterIteree(semester,
semesterCallback) {
    communication_service.sendTextMessage(senderID, '-----An ' + year
+ ', semestrul ' + ((semester % 2) + 1) + '-----');

    module.exports.getPayload().then(function(payload) {
      module.exports.getMarks(semester, payload).then(function(marks)
{
        async.eachSeries(marks, function markIteree(mark,
markCallback) {
          communication_service.sendTextMessage(senderID,
mark.name + ' ' + mark.value).then(function() {
            markCallback(null);
          });
        }, function done() {
          semesterCallback(null);
        });
      });
    });
  }, function done() {})}

```

Se iterează prin fiecare semestru, se face apelul către pagina eSIMS pentru colectarea datelor și se trimit utilizatorului, rând pe rând, în ordinea în care sunt găsite în platformă.

- note an X, semestrul Y – Chatbot-ul colectează notele unui semestru precizat. Exact ca în cazul notelor pe ani, singura schimbare apare doar la apelul funcției *scrapeMarks*, ea având acum ca parametru un singur semestru;
- nota ABREVIERE_MATERIE – În facultatea de informatică, există un catalog nescris al tuturor abrevierilor materiilor, a căror acustică sună bine. Așadar, materii precum Structuri de date, Rețele de calculatoare, Inteligență Artificiă, s-au ales, în limbajul studenților, cu prescurtări precum SD, RC sau IA. Cunoșcând acest fapt, am ales să introduc această facilități în aplicație. Pentru a realiza acest lucru, a fost nevoie de o listă cu legături în abrevieri și materiile asociate. Am ținut cont că prescurtare poate proveni de la mai multe materii (de exemplu, „ag”, poate proveni atât de la Algoritmica Grafurilor, cât și de la Algoritmi genetici). În acest caz, am ales să afișez ambele materii.



2. Restanțele

Asemănător notelor, aflarea restanțelor se poate face în mai multe moduri: în mod general, în care se caută restanțele pe toată durata a celor trei ani de studiu, sau în mod indicat, pe un singur an.



Algoritmul se folosește de modulul de colectare a notelor explicat mai sus. În funcție de perioada selectată, se verifică fiecare notă care există, să nu fie sub 5. În caz contrar, materia aferentă notei este considerată restanță, și afișată către utilizator.

3. Taxele

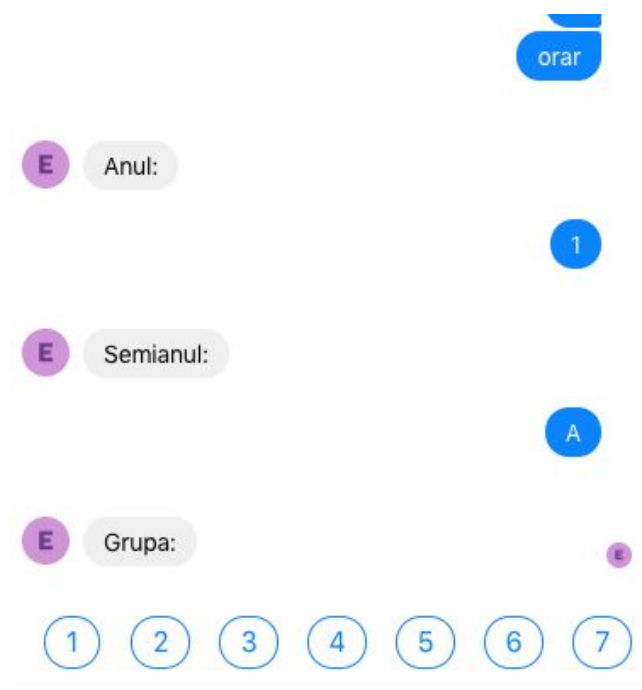
Taxele, precum notele, sunt împărțite pe semestre, de la 1 la 6. Ele se află sub formă de tabel, sub secțiunea de note, și conțin linii cu denumirea taxei, suma care trebuie plătită și cât s-a plătit din ea. În cazul în care suma ce a fost plătită nu coincide cu suma care trebuie plătită sau lipsește complet, linia respectivă va fi contorizată ca taxă restantă.

Secvența care iterează prin liniile tabelului de taxe:

```
var debts =  
$('#ctl00_WebPartManagerPanel1_WebPartManager1_wp1896648950_wp1261240874_GridViewTaxe tr');  
for(var i = 0; i < debts.length; i++) {  
    if (i > 0) {  
        if (!debts[i].children[1].children[0].children[0].data) {  
            communication_service.sendMessage(senderID,  
debts[i].children[1].children[0].children[0].data + ' ' +  
debts[i].children[2].children[0].children[0].data);  
            atLeastOne = true;  
        }  
    }  
}
```

4. Orarul

Independent de platforma eSIMS, orarul presupune deținerea a trei informații despre utilizator: anul, semianul și grupa din care face parte. Acest lucru se face la detectarea cuvântul „orar”, rezultatul fiind aruncarea utilizatorului într-un *flow* de selectare a celor menționate mai sus.



În urma setării celor trei informații, utilizatorul are acces la trei moduri de vizualizare:

- Orarul pe toată săptămâna – Dacă *user*-ul va scrie din nou „orar”, de data aceasta îi va fi afișat orarul pe toată săptămâna. Pentru a fi posibil acest lucru, mai întâi se preiau informațiile introduse de către el la pasul de configurare a anului, semianului și grupei din care face parte, apoi se formează un *link* ce va conține prefixul „https://profs.info.uaic.ro/~orar/participanti/orar_I”. De exemplu, dacă un student se află în anul 1, semianul B și grupa 6, *link*-ul va arăta în felul următor: https://profs.info.uaic.ro/~orar/participanti/orar_I1B6.html. Urmează pasul de transpunere a orarului în replici către utilizator, care este realizat cu ajutorul web scraping-ului. O parte din orarul afișat:

Luni

10:00 - 12:00 Limba engleza I Seminar C903

12:00 - 14:00 Matematica Curs C2

18:00 - 20:00 Matematica pentru concursuri Consultatii C2

Marti

08:00 - 10:00 Arhitectura calculatoarelor si sisteme de operare Curs C2 , video

10:00 - 12:00 Structuri de date Curs C2 , video

12:00 - 14:00 Introducere in programare Seminar C413

16:00 - 18:00 Logica pentru informatica Curs C2 , video

Miercuri

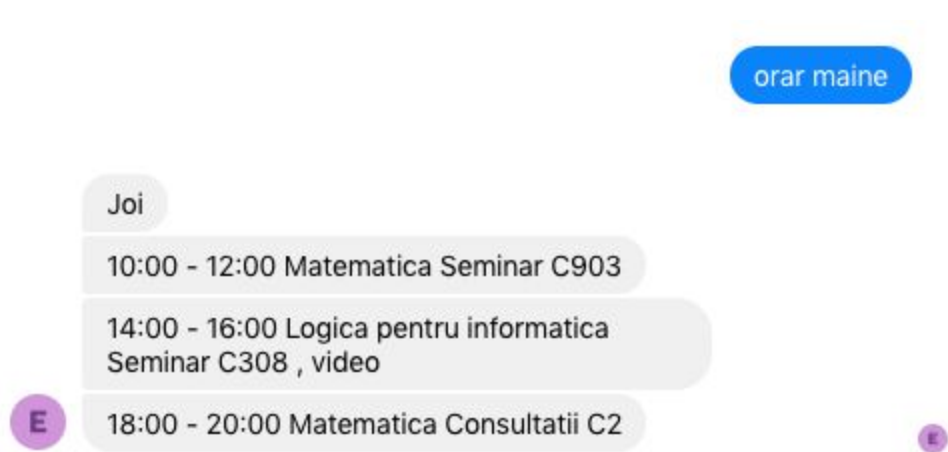


14:00 - 16:00 Structuri de date Seminar C905

- Orarul de astăzi – Este posibil prin introducerea textului „orar azi”. În acest caz, chatbot-ul identifică, în limba engleză, cu ajutorul bibliotecii MomentJs care este ziua actuală, o traduce în limba română și o caută în tabelul rezultat în urma web scraping-ului. Secvența care face iterarea printre liniile tabelului:

```
for (var i = 1; i < data[0].length; i++ ) {
  if (data[0][i].indexOf(daysRO[moment().format('ddd')]) !== -1) {
    start = i;
  } else {
    if (data[0][i].indexOf(nextDay[moment().format('ddd')]) !== -1) {
      end = i;
    }
  }
}
//in data[0] stocam toate liniile tabelului orarului
//moment().format('ddd') returnează ziua curentă
```

- Orarul de mâine. Asemănător orarului de astăzi, singura diferență fiind modul de obținere, și anume prin introducerea textului „orar maine”.



Idei de extindere

La începuturile ei, aplicația a fost mai mult o provocare, o cercetare asupra integrării platformei eSIMS cu un serviciu extern, având, cred eu, o concretizare finalizată cu succes. Întrucât ea ar putea fi folosită în momentul de față de către orice student ce are date de conectare la eSIMS. Cu toate acestea, cred că poate fi îmbunătățită destul de mult pe viitor plecând de la ideile următoare:

- Am ales platforma Facebook pentru a-mi susține ideea deoarece aceasta are cei mai mulți utilizatori (9,6 milioane doar în România), fiind greu de crezut că mai există vreun student care nu deține un cont. Totuși, pentru cei care nu au cont, chatbot-ul poate fi extins către orice alt instrument deja existent și cu flux mare de utilizatori (Skype, Slack, WhatsApp), sau, de ce nu, se poate opta pentru construirea unei aplicații proprii a universității. Indiferent de opțiunea care s-ar alege, algoritmul din spatele aplicației poate fi atașat și folosit oriunde, singura diferență fiind modalitatea în care primește mesajele de la utilizator.
- O altă idee ar fi ca chatbot-ul să conțină o parte de notificări. Există posibilitatea ca studentul să aștepte confirmarea unei note prin punerea acesteia în eSIMS. Bot-ul ar putea verifica în permanență când se schimbă ceva în platformă, și, sesizând modificarea, să trimită o notificare studentului. Acest lucru este posibil prin folosirea *cron job*-urilor¹⁴ care să ruleze o dată pe zi, eventual, noaptea. Tot în partea de notificări ar putea intra urmărirea orarului și înștiințarea studentului când s-a afișat programul examenelor, sau ar putea chiar să îl atenționeze de fiecare dată când se apropie un examen și chiar să îi sugereze resurse de pe care poate învăța pentru materia respectivă.
- Venind vorba de recomandări, ea poate fi prezentă și în partea de restanțe a aplicației. Când studentul dorește să afle restanțele pe care le are, chatbot-ul poate să-i afișeze, împreună cu lista cerută, o serie de resurse care vor putea servi drept material de studiu. Eventual, alte materiale decât pagina cursului.
- Având în vedere că am putut integra platforma eSIMS cu un serviciu total extern, de ce nu am putea face același lucru și cu alte utilitare ale universității? De pildă, platforma de

¹⁴ Cron job - o secvență de cod care se execută la un interval de timp.

mail-uri. Cred că partea de autentificare implementată pentru eSIMS ar putea fi foarte bine pusă în aplicare și la platforma de email. Impactul pe care această integrare l-ar aduce ar putea fi reperat în zona de notificări. În loc ca studentul să fie nevoit să intre de fiecare dată în aplicație pentru a vedea dacă a primit un email nou, acesta ar putea fi notificat în timp real de către chatbot, afișându-i-se și conținutul mesajului și, eventual, o opțiune de răspuns.

- Ultima idee provine din faptul că chatbot-ul, în forma actuală, este destul de rigid când vine vorba despre modul în care utilizatorul poate cere informațiile. Pentru a îmbunătăți acest lucru, ar putea fi introdus un modul de *Natural Language Processing*, care să ajute la înțelegerea contextului din mesajele primite. În varianta oferită de Facebook, acest lucru este deja posibil, și chiar există predefinit un set de concepte, de la formule de salut până la vreme, timp, dată.

Bibliografie

- [1] “Nikola Tesla on Artificial Intelligence”,
blog.habrador.com/2015/12/nikola-tesla-on-artificial-intelligence.html
- [2] “The Turing Test ”, <http://www.psych.utoronto.ca/users/reingold/courses/ai/turing.html>
- [3] “An Interactive History of Chatbots”, blog.aylien.com/interactive-history-chatbots/;
- [4] “Soul of the Machine: How Chatbots Work”, medium.com/@gk_/how-chat-bots-work-dfff656a35e2
- [5] Ryan Mitchel, “Web scraping with
Python”,zempirians.com/ebooks/Ryan%20Mitchell-Web%20Scraping%20with%20Python_%20Collecti
ng%20Data%20from%20the%20Modern%20Web-O'Reilly%20Media%20(2015).pdf
- [6] “Web Scraping and Crawling Are Perfectly Legal, Right?”,
benbernardblog.com/web-scraping-and-crawling-are-perfectly-legal-right/
- [7] “Is Web Scraping Legal ?”, webharvy.com/articles/is-scraping-legal.html