

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, 2017**

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, 2017**

Coordonator științific

Lector, dr. Mădălina Răschip

DECLARAȚIE PRIVIND ORIGINALITATE ȘI RESPECTAREA DREPTURILOR DE AUTOR

Prin prezenta declar că Lucrarea de licență cu titlul „”, 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, 30.06.2017

Absolvent Bogdan-Gabriel Mihai

(semnătura)

DECLARAȚIE DE CONSIMȚĂMÂNT

Prin prezenta declar că sunt de acord ca Lucrarea de licență cu titlul „”, 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, 30.06.2017

Absolvent Bogdan-Gabriel Mihai

(semnătura)

Cuprins

Cuprins	5
Terminologie	7
Introducere	8
Ce este un chatbot?	8
Motivația	9
Ideea de aplicație	9
Capitolul I: Chatbot-uri	10
Evoluția chatbot-urilor	10
De ce acest interes în chatbot-uri?	11
Tipuri de chatbot-uri	12
Clasificare prin potrivire de șabloane	12
Clasificare prin algoritmi	14
Clasificare prin rețele neuronale	15
Messenger Chatbot	16
Capitolul II: Web Scrapping	17
Ce este?	17
Este legal?	17
Cum este folosit în aplicație?	18
Capitolul III: Proiectarea aplicației	19
Ideea	19
Tehnologii	20
Firebase	20
Heroku	21
Git	21
Facebook webhooks	21
MomentJS	22
Lodash	23
Async	23
CheerioJs	24
Cheerio TableParser	24
Capitolul IV: Aplicația	25
Comunicarea	25
Autentificarea cu eSIMS	28

Folosirea aplicației	31
Idei de extindere	37
Bibliografie	39

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 în orice aplicație de chat care există deja (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 scrapping*: 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).
- *User*: Utilizatorul care folosește aplicația

Introducere

În zilele noastre, chatbot-urile au câștigat teren considerabil pe teritoriul WWW, 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 Tesla¹ 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, recomandări sau chiar decizii concludente. 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*. În forma lui inițială, jocul presupunea trei participanți: un băbat, 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, în mai mult de 50% din cazuri nu își poate da seama cine este cine, calculatorul poate fi perceput ca un înlocuitor al omului.

¹ "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."

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

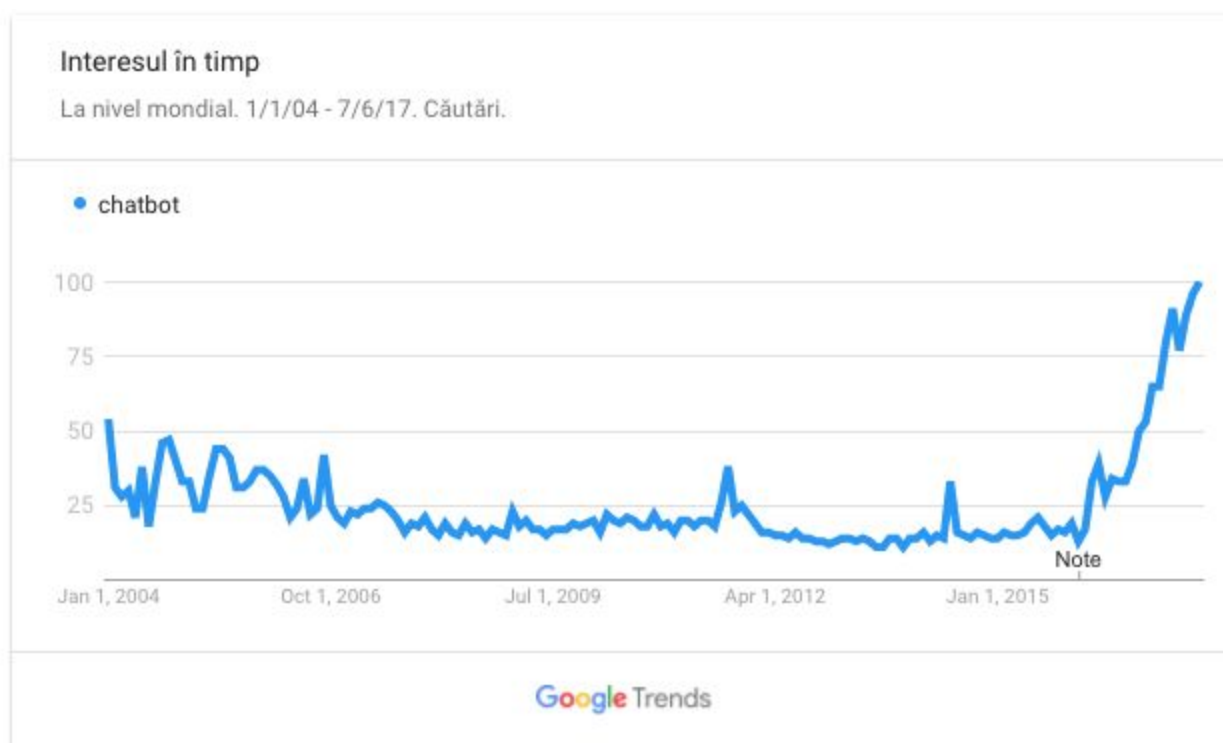
Motivația

Ideea de aplicație

Capitolul I: Chatbot-uri

Evoluția chatbot-urilor

În acest moment, chatbot-urile reprezintă un subiect destul de prezent. Sunt în centrul schimbării a modului de comunicare, dar și a modului de creștere a unor companiilor precum Facebook, Microsoft.



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

De ce acest interes în chatbot-uri?

Cred că el provine dintr-o dorință de resuscitare a inteligenței artificiale și aplicabilitatea acesteia în industrie, dar și din obsesia noastră de a avea servicii care funcționează non-stop. Un studiu recent² arată că 44% din consumatorii din SUA preferă să folosească un chatbot în detrimentul conversației cu un om și 61% din cei care au luat parte la cercetare, folosesc deja un chat de cel puțin o dată pe lună. Acest fapt se datorează disponibilității non-stop a chatbot-ului și viteza cu care poate răspunde la întrebări.

Mari companii au anticipat acest moment și deja se bazează pe aplicații de mesaje și asistenți cât mai „intelenți” pentru a oferi o experiență plăcută utilizatorilor. Acest lucru poate fi pus în practică mai ales de când Facebook a lansat Messenger Chatbot³. Deși acest concept a crescut mai mult în ultimul timp, el nu este deloc nou. El a fost prezent încă de acum 50 de ani în comunitatea NLP (Natural Language Processing) și nu este decât o etapă către adevăratul scop: înțelegerea limbajul uman de către calculator.

De-a lungul timpului au fost construite mai multe astfel de chatbot-uri:

- 1966: Eliza. A fost primul chatbot creat vreodată bazat pe potrivire de șabloane și înlocuire în replicile utilizatorilor, 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.
- 2010: Siri. Asistent personal care face parte din sistem 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 un asistent personal

²Link:https://www.aspect.com/globalassets/2016-aspect-consumer-experience-index-survey_index-resul ts-final.pdf

³ Link: <https://messenger.fb.com/>

- 2012: Google Now. La fel ca Siri, procesează limbajul natural și se folosește de servicii web pentru a da răspunsuri, recomandări. El este încorporat în aplicație de căutare de la Google.
- 2016: Chatbot-uri pentru Facebook Messenger. În aprilie 2016, Facebook a lansat o platformă destinată programatorilor care le permite să interacționeze cu utilizatorii prin intermediul ferestrei de convorbire din Facebook. Până în prezent, sunt peste 11.000 de chatbot-uri create cu ajutorul acestei platforme.

Tipuri de chatbot-uri

Clasificarea unui chatbot provine din felul 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:

Clasificare prin potrivire de șabloane

Primele chatbot-uri foloseau această tehnică pentru a clasifica textul și pentru a produce un răspuns. Mai este cunoscută sub numele de „forță brută” deoarece creatorul chatbot-ului trebuie să introducă fiecare șablon și semnificația lui. Pentru aceste șabloane se folosește AIML⁴.

⁴ 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

Clasificare prin algoritmi

Mecanismul prin forță brută are o parte descurajantă: pentru fiecare situație trebuie să existe un șablon pentru ca chatbot-ul să poate clasifica propoziția utilizatorului. Din această cauză, un alt tip de clasificare a apărut, cea prin algoritmi. Acest lucru se poate face cu ajutorul unei ecuații. Cea mai întâlnită algoritmul pentru clasificarea textului este „*Multinomial Naive Bayes*”⁵

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

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

Formula de mai sus este mai simplă decât pare. Algoritmul presupune un set de propoziții asignate unei clase. Dacă 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, deoarece, spre exemplu, propozițiile pot fi mult mai des întâlnite decât cuvinte ca „vreme”, „stare”, „preponderență” șamd. În forma mai complicată a algoritmului, înainte de a căuta cuvintele în setul de bază, ele trebuie aduse la forma lor inițială, fără derivări, 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?"
```

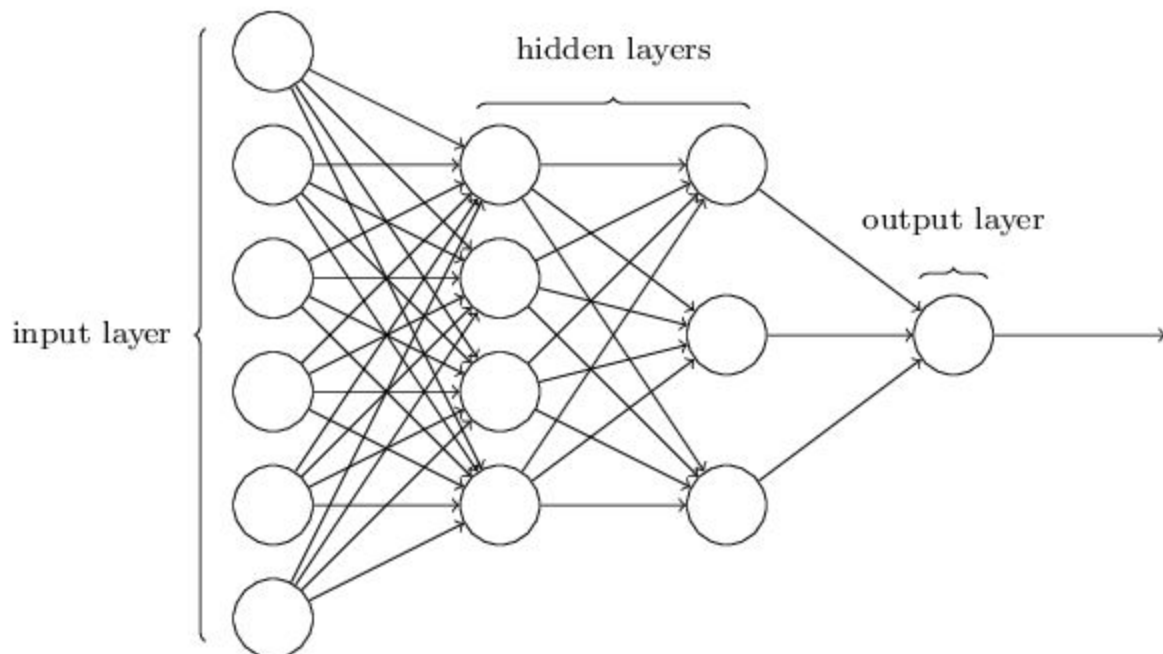
⁵Link: <https://nlp.stanford.edu/IR-book/pdf/13bayes.pdf>

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)
```

Clasificare prin rețele neuronale

Rețelele neuronale au fost inventate în 1940, ca o modalitate de a da 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. Din nou, fiecare propoziție este bucățelită în cuvinte, care devin date de intrare pentru rețea. Scorurile sinaptice sunt calculate iterând de mii de ori, de fiecare dată ajustându-se pe baza pasului anterior. Aceste scoruri reprezintă importanța informației: cu cât a fost accesată de mai multe ori, cu atât e mai importantă. Comparativ 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 platforma lor. În momentul de față ea fiind cea mai folosită platformă pentru a crea astfel de servicii, câștigând avans față de ceilalți competitori (Microsoft, Slack) datorită fluxului mare de utilizator pe care îl are, dar și a multitudinii de funcționalități pe care le pune la dispoziție. Câteva dintre ele ar fi:

- Identificarea utilizatorului după un ID unic
- Conectarea cu aplicații externe
- Salvarea atașamentelor 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 Scrapping

Ce este?

Procesul automat de colectare a datelor de pe Internet este aproape la fel de vechi ca Internetul însuși. 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 face interogări unei pagini html și extrage ce informații are nevoie, indiferent de natura lor (text, imagini, video, script-uri).

Este legal?

Cred că este destul de dificil să spună 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 *web scraping-ului*. După cum am mai spus, având în vedere că informația este în spațiul public, nu ar trebui să

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

constituie o problemă salvarea ei, dar în cazul în care ne însușim informația, și o distribuim fără a credita sursa, s-ar putea să fie greșit, și chiar ilegal, în cazul în care există drepturi de autor.

De asemenea, având în vedere că extragerea de date folosind un algoritm este mult mai rapidă și mai eficientă decât cea prin accesare manuală, trebuie avut 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 preveni accesul la pagină.

Cum este folosit în aplicație?

În toată aplicația el este folosit pentru conectarea cu platforma eSIMS și afișarea informațiilor cerute de către utilizator. Ele nu sunt stocate în nicio bază de date, ci persistă doar în istoricul conversației, putând fi șterse de către utilizator oricând.

Capitolul III: Proiectarea aplicației

Ideea

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

The screenshot shows the eSIMS website interface. At the top, there is a header with the university logo and name: "universitatea alexandru ioan cuza iași" and "departamentul de statistică și informatizare". A "Logout «31090104SL130210»" link is visible on the right. Below the header is a navigation bar with "Pagina principală" (highlighted in orange), "Studenti", and "Recuperare parolă". The main content area is titled "Componentă Web destinată studenților." and contains a list of instructions for students. On the left, there is a sidebar with "Noutăți" and "Legături rapide" (including links to weather, MEC, Bologna, Euro Univs, etc.). On the right, there is a section for "eSIMS" with a small image of a bird.

universitatea alexandru ioan cuza iași
departamentul de statistică și informatizare

Logout «31090104SL130210»

Pagina principală Studenti Recuperare parolă

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.

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

eSIMS
« »


» ...

Având în vedere scopul eSIMS-ului, de a vedea situația notelor, restanțelor și a taxelor, ele sunt ascunse și nu poți ajunge la ele decât după câteva click-uri. 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. Pe lângă funcționalitățile de bază,

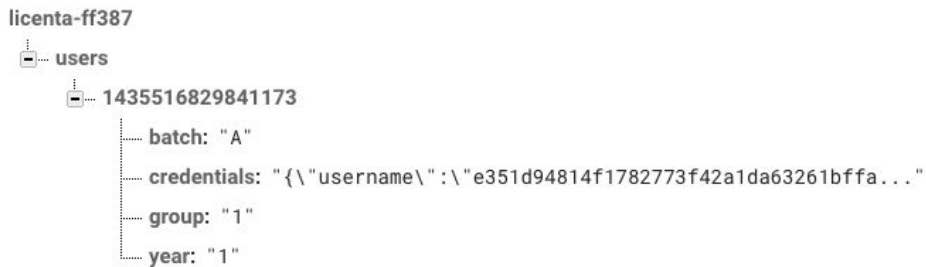
aflarea notelor, taxelor, restanțelor, am considerat că poate fi util și integrarea orarului. Așadar, studentul poate ști și orarul de astăzi, de mâine, sau pe toată săptămâna.

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.



Iată secvența responsabilă cu partea de înregistrare a unui student:

```
var newUser = {};  
newUser[event.sender.id] = { credentials: event.account_linking.authorization_code  
};  
firebase.database.ref('users').set(newUser);
```

⁸ Link: <https://firebase.google.com/>

2. Heroku

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

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 când o diversitate de schimbări apar, 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. Lista evenimentelor 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.

⁹ Link: <https://www.heroku.com/>

¹⁰ Link: <https://github.com/>

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') {

    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

Mă ajută la partea de orar, 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 de mult timp, de cele mai multe ori se ajungea la așa numitul *callback hell*, termen folosit când prelucrarea unor date era condiționată de finalizarea unor acțiuni (colectare de date prin servicii web, generarea unor statistici, șamd...). În aplicație, ordinea răspunsurilor este importantă, mai ales în secțiunea de orar, 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 aceasta în secvența de mai jos:

```
function showDaySchedule(senderID, obj) {  
    communication_service.sendTextMessage(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.sendTextMessage(senderID,  
string).then(function() {  
                scheduleCallback(null);  
            });  
        }, function done() {  
            resolve();  
        });  
    });  
}
```

¹¹ Link: <https://lodash.com/>

```
});  
}}}
```

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');  
        })  
    }  
}
```

Cheerio TableParser

O extensie a bibliotecii Cheerio, care ajută la convertirea tabelelor din HTML în *array*-uri 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, trimite 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( *) (, ?) ( *)semintru(1 ?) ( +)\d)/gi,  
    means: 'note_semintru'  
  },  
  {  
    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. Prelucra răspunsului. După ce este atribuit un sens mesajului, acesta este preluat și acționat asupra lui. O parte din secvența care face posibil acest lucru:

```

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

```

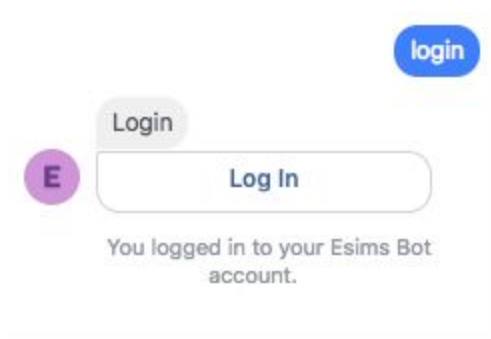
4. Trimiterea lui. În cadrul webhook-ului definit, pe lângă mesajul trimis de către utilizator, primim și un ID unic atașat. Pe baza acestuia, știm cui să răspundem. Pentru a trimite un mesaj înapoi, trebuie să efectuăm un apel către Facebook, atașând câțiva parametri, ca în secvența de mai jos:

```
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);
      }
    });
  });
}
```


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, criptate) și se accesează adresa.



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:

¹³ JSON - Javascript Object Notation

```

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 a se deconecta, utilizatorul va fi nevoit să scrie „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 lua și vom discuta fiecare funcționalitate în parte:

1. Notele

Înainte de implementare am fost nevoit să studiez puțin pagina eSIMS-ului care se ocupa de gestionarea și afișarea notelor. Așadar, am constatat că ele 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 trimiși o serie de parametri, fiind nevoie de un *request* în plus pentru a colecta parametrii generați de către server, la care se adaugă parametrul `__EVENTARGUMENT`, 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.sendMessage(senderID, errors.year);
    }
    break;
```

Codul de mai sus este creat cu 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 semestrele 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 trimite 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 a tuturor abrevierilor materiilor a căror acustică sună bine. Așadar, materii precum Structuri de date, Rețele de calculatoare, Inteligență artificială, 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. A trebuit să am grijă atunci când o prescurtare poate conține 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 cinci. În caz contrar, materia aferentă notei este considerată drept restanță, și, afișată către utilizator.

3. Taxele

Datoriile, 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;  
        }  
    }  
}
```

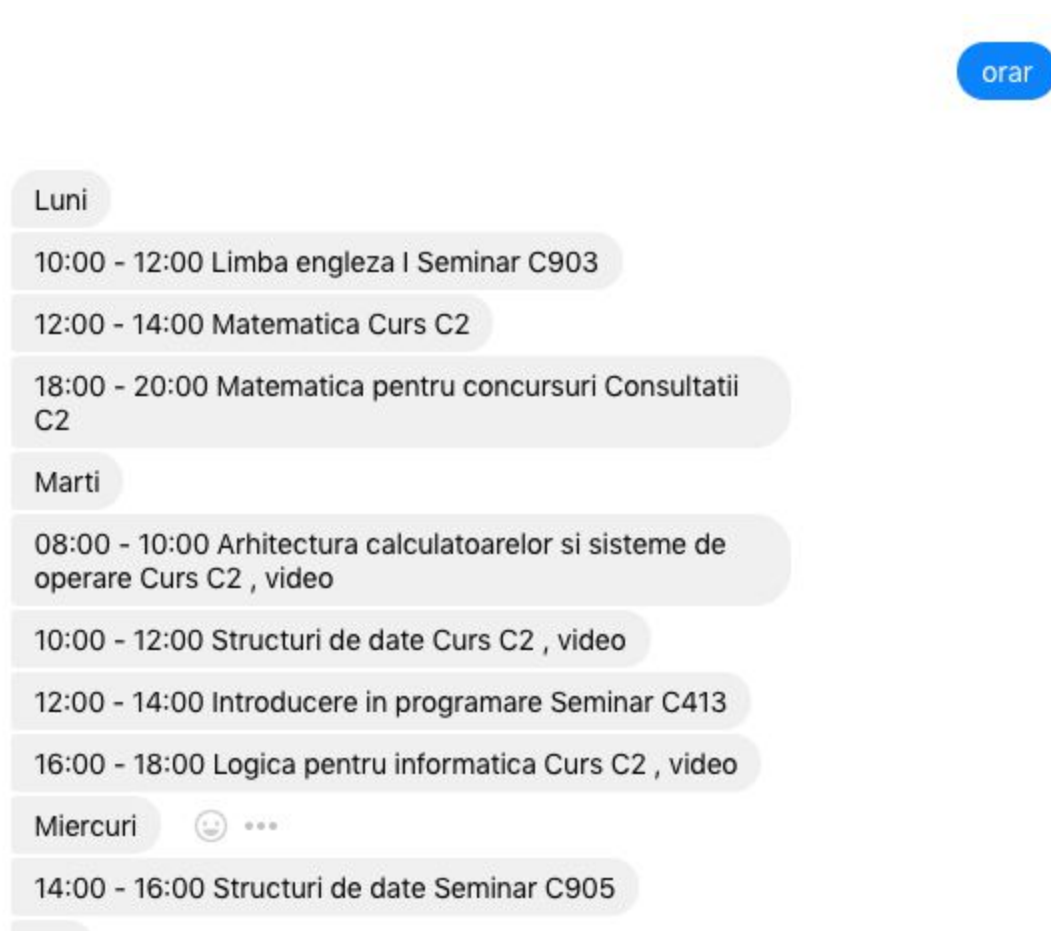
4. Orarul

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”, rezultat fiind aruncarea utilizatorului într-un *flow* de selectare a celor menționate mai sus.

The image shows a user interface for selecting academic information. It features three input fields labeled "Anul:", "Semianul:", and "Grupa:". To the right of these fields are three blue circular buttons labeled "1", "A", and "E" respectively. At the bottom, there is a horizontal row of seven blue circular buttons labeled "1" through "7". A blue button labeled "orar" is located at the top right of the interface.

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 aceeași îi va fi afișat orarul pe toată săptămâna. Pentru a fi posibil acest lucru, mai întâi se iau informațiile introduse de către el la pasul de configurare a anului, semianului și grupei din care face parte, apoi se formează un *link* cu 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 tranpunere a orarului în replici către utilizator, care este realizat cu ajutorul web scraping-ului. O parte din orarul afișat:



- 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”.

orar maine

Joi

10:00 - 12:00 Matematica Seminar C903

14:00 - 16:00 Logica pentru informatica Seminar C308 , video

E

18:00 - 20:00 Matematica Consultatii C2

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, ca rezultat reușita, 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 ea poate fi îmbunătățită destul de mult plecând de la ideile următoare:

- Am ales platforma Facebook pentru a-mi susține ideea deoarece ea are cei mai mulți utilizatori (9,6 milioane doar în România), fiind greu de crezut că mai există cineva fără cont. Totuși, pentru cei care nu au cont, chatbot-ul poate fi extins către orice platformă deja existentă 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 oriunde, singura diferență fiind felul 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 notifice 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 ar putea să-i afișeze, împreună cu lista cerută, o serie de resurse de pe care se poate inspira pentru a învăța. 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 mail-uri. Cred că partea de autentificare implementată pentru eSIMS ar putea fi foarte bine pusă în aplicare și la platforma de mail-uri. Impactul pe care această integrare l-ar aduce, ar putea veni 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 mail nou, acesta ar putea fi

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

notificat în timp real de către chatbot, afișându-i-se și conținutul mesajului, și, eventual, cu opțiune de răspundere.

- 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 au predefinit un set de concepte, de la formule de salut până la vreme, timp, dată.

Bibliografie

- [1] Oisin Muldowney, “Chatbots: An Introduction And Easy Guide To Making Your Own ”,
- [2] Inbenta, “The ultimate guide to chatbots for business”
- [3] “Soul of the Machine: How Chatbots Work”, medium.com/@gk_/how-chat-bots-work-dfff656a35e2
- [4] “An Interactive History of Chatbots”, blog.aylien.com/interactive-history-chatbots/;
- [5] Ryan Mitchel, “*Web scraping with Python*”, [zempirians.com/ebooks/Ryan%20Mitchell-Web%20Scraping%20with%20Python_%20Collecting%20Data%20from%20the%20Modern%20Web-O'Reilly%20Media%20\(2015\).pdf](http://zempirians.com/ebooks/Ryan%20Mitchell-Web%20Scraping%20with%20Python_%20Collecting%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
- [8] “Nikola Tesla on Artificial Intelligence”, blog.habrador.com/2015/12/nikola-tesla-on-artificial-intelligence.html