

ROBOT DE CHAT PENTRU SOLUȚIONAREA PROBLEMELOR STUDENȚILOR - RusBoT

Autori: Andrei – Cătălin FARCAȘ ¹, Denisa – Mihaela CIORNILĂ ²

andrei4ie@gmail.com, denisamihaelaciornila@gmail.com

Coordonator: Șef lucr.dr.ing. Cosmin RUS³

¹Universitatea din Petroșani, Facultatea I.M.E., specializarea: TTIA , anul 2

² Universitatea din Petroșani, Facultatea I.M.E., specializarea: Calculatoare , anul 2

³Universitatea din Petroșani, Facultatea I.M.E., Departamentul: A.C.I.E.E.

Rezumat:

Această lucrare investighează implementarea unui robot de chat, "RusBot", la Universitatea din Petroșani, cu scopul de a facilita accesul studenților la informații vitale, cum ar fi orarele și localizarea sălilor de curs. Motivată de adaptarea la era digitală și dorința de a îmbunătăți eficiența administrativă și academică, lucrarea abordează mai multe aspecte esențiale. Se evidențiază cât de profundă este integrarea tehnologiei în activitățile zilnice, propunând utilizarea acestei tendințe pentru îmbunătățirea vieții academice prin roboți de chat accesibili pe tablete în universitate. Lucrarea descrie tehnologia roboților de chat, de la sisteme simple la asistenți digitali avansați, folosind tehnici de IA, NLP și ML, subliniind diferența dintre roboții orientați pe activități și cei pe date și predicții, cu capacități mai complexe și personalizate. De asemenea, se detaliază utilizarea limbajului de programare C# și platforma .NET Framework, discutând despre componentele esențiale cum ar fi Common Language Runtime (CLR) și biblioteca de clase .NET, care suportă executarea și gestionarea memoriei aplicațiilor. Arhitectura aplicației este bazată pe modelul MVVM, facilitând separarea logicii de afișare de cea de interfață și permițând navigarea ușoară între diferite ferestre, cu un sistem de navigare ce păstrează istoricul interacțiunilor.

Cuvinte cheie:

robot de chat, automatizare, tehnologie digitală, inteligență artificială, interfață utilizator

1. Introducere

Știm cu toții că trăim în era digitală, eră în care mare parte din acțiunile desfășurate de către noi, zi de zi, se efectuează sau chiar automatizează de către calculator. Fiind studenți ai aceste prestigioase Universități, am dorit să ne gândim în afara cutiei și să căutăm moduri în care am putea ajuta ceilalți studenți din Universitate să ducă o viață de student cât mai simplă. Astfel, am ajuns la concluzia finală, după care am creat această lucrare. În această eră a vitezei, studenții au nevoie de un asistent mereu la dispoziția lor.

De aceea, ne-am gândit să realizăm un robot de chat, ce va putea fi prezent mereu pe o tabletă ce va fi situată în interiorul Universității, unde studenții vor putea veni rapid pentru a găsi soluția la impedimentele acestora. Un robot de chat este un program de computer care simulează și procesează conversația umană (scrisă sau vorbită), permițând oamenilor să interacționeze cu dispozitivele digitale, ca și cum ar comunica cu o persoană reală. Roboții de chat pot fi la fel de simpli precum programele rudimentare care răspund la o întrebare cu o singură propoziție sau la fel de sofisticate ca asistenții digitali care învață și evoluează, pentru a oferi niveluri de personalizare în creștere, pe măsură ce colectează și procesează informații.

Spre deosebire de acestea, roboții de chat permit companiilor să interacționeze cu un număr nelimitat de clienți într-un mod personalizat și se pot scala în sus sau în jos în funcție de cererea și cerințele afacerii. Prin utilizarea roboților de chat, o companie poate oferi simultan servicii similare cu cele umane, personalizate și proactive pentru milioane de persoane.

2. Caracteristici generale ale roboților de chat

Bazați pe IA, reguli automate, procesare a limbajului natural (NLP) și machine learning (ML), roboții de chat procesează datele pentru a oferi răspunsuri la solicitări de toate tipurile.

Există două tipuri principale de roboți de chat.

- Roboții de chat orientați pe activități (declarativi) sunt programe cu un scop unic, care realizează o singură funcție. Utilizând reguli, NLP și foarte puțin ML, aceștia generează răspunsuri automatizate, dar conversaționale la întrebările utilizatorilor. Interacțiunile cu acești roboți de chat sunt foarte particularizate și structurate, fiind utilizate în special pentru funcții de asistență și service - gândire robustă, întrebări frecvente interactive. Roboții de chat orientați pe activități pot gestiona întrebări comune, precum cele privind orele de program sau tranzacțiile simple, care nu implică o multitudine de variabile. Deși utilizează NLP, astfel încât utilizatorii finali să poată interacționa în mod conversațional, funcționalitățile acestor roboți sunt în principiu de bază. Aceștia sunt în prezent cei mai utilizați roboți de chat.
- Roboții de chat orientați pe date și previziuni (conversaționali) sunt numiți adesea asistenți virtuali sau asistenți digitali și sunt mult mai sofisticăți, interactivi și personalizați decât cei orientați pe activități. Acești roboți de chat sunt conștienți contextual și utilizează o înțelegere a limbajului natural (NLU), NLP și ML pentru a învăța din mers. Aceștia aplică informațiile predictive și analizele, pentru a permite personalizarea bazată pe profilurile și pe comportamentul din trecut ale utilizatorilor. Asistenții digitali pot învăța în timp preferințele unui utilizator, pot oferi recomandări și chiar pot anticipa cerințele acestuia. Pe lângă monitorizarea datelor și a intențiilor, pot iniția și conversații. Siri de la Apple și Alexa de la Amazon sunt exemple de roboți de chat orientați către consumatori, date și predicții.

3. Noțiuni generale despre limbajul de programare C#

Limbajul C# este unul dintre cele mai utilizate limbaje de programare multiparadigmă din lume. Este un limbaj simplu, modern, cu o flexibilitate foarte mare în ceea ce privește dezvoltarea de aplicații și portabilitatea acestora (deși există unele voci care spun că portabilitatea este limitată, vom demonstra în continuare lipsa de veridicitate a acestei afirmații). Istoria acestui limbaj își are originile la începutul mileniului în care ne aflăm. În iulie 2000 a fost lansată prima distribuție a limbajului C#, deși zvonurile apariției unui limbaj puternic au apărut încă de prin anul precedent. Microsoft a fost corporația care a decis să intre mult mai în forță în lumea dezvoltatorilor, întrucât limbajul Basic își pierduse popularitatea ceea ce a dus la o ușoară scădere a nivelului economic al companiei.

Unul dintre principalele motive care au inițiat crearea acestui limbaj de programare a fost legat de faptul că în anul 1995, cei de la Sun Microsystems au creat limbajul Java, primul limbaj multiparadigmă cu o portabilitate foarte mare. A înregistrat un nivel de popularitate exponențial în primii ani de la lansare, și chiar în zilele noastre, este pe primul loc mondial în ceea ce privește popularitatea. Astfel, Microsoft a decis să creeze un limbaj care să ofere mult mai multe elemente dezvoltatorilor. Portabilitatea limbajului se leagă în mod direct de platforma pe care rulează. Așa cum Java rulează în mediul său special, numit JVM (Java Virtual Machine), C# rulează în .NET Framework. Vom discuta acum despre această platformă, întrucât înțelegerea mecanismului care lucrează în spatele acesteia contribuie substanțial la înțelegerea funcționalității limbajului precum și la portabilitatea acestuia. Common Language Runtime este probabil cea mai importantă componentă a platformei, responsabilă de executarea fiecărei aplicații. Practic CLR-ul este mediul de execuție al platformei. Acesta conține toate tool-urile care permit executarea raportată la sistemul de operare a aplicației dezvoltate, printre care: garbage collector-ul, adică mecanismul de gestionare automată a memoriei – spre deosebire de C sau C++, acolo unde programatorul trebuia să decidă în ce punct al aplicației să aloce o zonă de memorie, și în ce punct să elibereze acea zonă, C# are acest feature foarte important, care eliberează automat memoria ocupată de obiecte atunci când știe că acestea nu vor mai fi folosite în cadrul aplicației; mecanismul de tratare automată a excepțiilor, adică a erorilor runtime, cele care apar în timpul execuției programului. Desigur că programatorul trebuie să își creeze propriul sistem de tratare a erorilor de execuție, numai că există situații în care nu se acoperă toate cazurile în care execuția poate fi întreruptă, de aceea acest mecanism de tratare automată a excepțiilor face medierea între sistemul de operare și platforma .NET. De asemenea o parte foarte importantă a platformei, alături de CLR, este și biblioteca de clase .NET sau base class library, pe scurt BCL. Practic, aici este tot codul creat de Microsoft pentru a construi limbajele .NET. Toate funcțiile, clasele, metodele, sunt aici în stare precompilată, ceea ce ușurează substanțial execuția programelor, timpul alocat pentru compilarea lor fiind practic inexistent.

4. Realizarea aplicației RusBoT

Aplicația descrisă în acest proiect constă în folosirea unei interfețe digitale pentru a putea răspunde întrebărilor generale ale unui student din Universitatea din Petroșani. În momentul actual, proiectul are ca și scop eliminarea a două probleme majore ale studenților, constatate de către studenți. Aceste două probleme sunt: obținerea orarelor semestrului curent și găsirea sălilor de curs. Robotul de chat “RusBot” este aici pentru a sări în ajutorul studenților.

Codul este scris în limbajul C#, utilizând subsistemul grafic WPF (Windows Presentation Foundation), dezvoltat de către Microsoft, ce oferă dezvoltatorilor posibilitatea de a crea interfețe grafice ușor și eficient (Fig. 1).

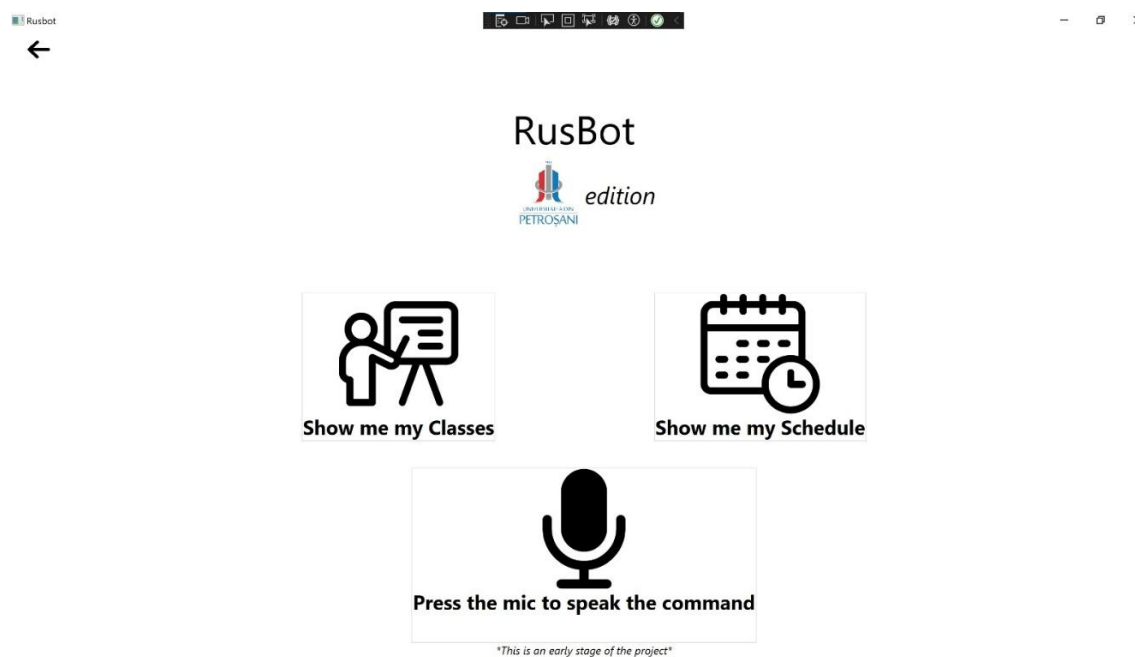


Fig. 1 Pagina de start a aplicației de tip chatbot - RusBoT

Codul aplicației constă în crearea unor ferestre ce sunt vizibile utilizatorului, prin care acesta poate interacționa cu programul. Utilizatorul poate tranzita de la o fereastră la alta utilizând butoanele ce le sunt atribuite diverse comenzi. De asemenea, pentru navigarea de la o fereastră la alta, a fost creat un sistem de navigare ce salvează istoricul de navigare al utilizatorului.


Această aplicație folosește șablonul arhitectural de tip MVVM (Model – View – ViewModel), șablon utilizat în programare ce a fost creat de către Microsoft și este specializat în șablonul de design Model Presentation. Modelul de prezentare extrage starea și comportamentul View-ului (Fereștrei de prezentare) într-o clasă de model ce face parte din prezentare. Modelul de prezentare se coordonează cu stratul de domeniu și oferă o interfață pentru vizualizare care minimizează luarea deciziilor în fereastra de prezentare. View-ul fie stochează toată starea sa în Modelul de prezentare, fie își sincronizează frecvent starea cu Modelul de Prezentare.

Arhitectura MVVM este bazată pe arhitectura de tip MVC (Model View Controller) și se concentrează pe platformele de dezvoltare a interfeței grafice moderne, cum ar fi și WPF. Crearea de ferestre pentru utilizator este realizată prin scrierea codului într-un limbaj declarativ, denumit XAML (Extensible Application Markup Language). View-ul este definit în limbajul XAML și ca o regulă de bază, acesta nu ar trebui să aibă logică în spate, fiind utilizat strict pentru interfața grafică.

Pentru a putea comunica cu backend-ul, View-ul ar trebui să fie conectat cu ViewModel-ul pentru a putea transfera datele necesare. Model-ul este responsabil pentru afișarea și structurarea datelor într-un mod în care este simplu de consumat de către WPF. Pentru acest lucru, ar trebui să implementeze interfața INotifyPropertyChanged, ce ajută la actualizarea și transmiterea datelor actualizate în timp real. ViewModel-ul

este un model pentru View-ul din aplicație, sau mai bine spus, o abstractizare a view-ului. Este utilizat pentru a transmite informațiile necesare către view și de a expune acțiunile executate prin comenzi.

Robotul de chat are posibilitatea de a afișa orarele și sălile aferente universității, pentru ca studentul ce vine și interacționează cu acesta să le poată găsi cu ușurință (Fig. 2).



RusBot
edition
PETROȘANI

An univ. 2023-2024, sem. II

LUNI		MARȚI		MIERCURI		JOI		VINERI	
8 ⁰⁰ - 9 ⁰⁰	Circuit electronic Lupuș L Pădureanu, N. L&A	Grădă științifică C - Petrosani N. A4 Kali A. A2	Circuit electronic C - Petrosani N. A4 Kali A. A2	Informatică aplicată L Pădureanu, N. L&A	Grădă științifică L Pădureanu, N. L&A				
10 ⁰⁰ - 11 ⁰⁰	Fizică C - Bănuș C. A2 +C11+C12+A1+A2	Informatică aplicată C Lupuș M. A3 +C11+C12	Circuit electronic C Pădureanu, N. A4 +C11+A2+A3+A4+A5		Grădă științifică L Lupășescu, E. L&A			Informatică aplicată A Pădureanu, N. L&A	
12 ⁰⁰ - 13 ⁰⁰		Engleză B Dănilăuș, G. L&A	Limbaj de programare II C Lupuș M. +C11+C12+A2+A3+A4		Limbaj de programare II L Olea M. L&A				
14 ⁰⁰ - 15 ⁰⁰			Limbaj de programare II C Lupuș M. A3 +C11+C12+A3+A4+A5		Limbaj de programare II L Olea M. L&A				
16 ⁰⁰ - 17 ⁰⁰	Fizică L Bănuș C. F1								
18 ⁰⁰ - 19 ⁰⁰		Fizică L Bănuș C. F1							

Legendă: Orele notate cu font roșu corespund săptămânii timpore (1, 3, ..., 13), iar cele notate cu font albastru corespund săptămânii pare (2, 4, ..., 14).
Atenție: Fiecare oră cuprinde 50 min. activitate didactică și 10 min. pauză.

Fig. 2 Fereastră afișare orar

Totodată, există și opțiunea de a utiliza comenzi vocale pentru a putea găsi informațiile necesare, însă este momentan în curs de dezvoltare, deoarece se dorește implementarea Inteligenței Artificiale în acest robot de chat, pentru a-l dezvolta mai departe și transforma într-un Asistent Virtual.

Concluzii

În concluzie, abordarea noastră inovativă privind integrarea unui robot de chat în mediul universitar evidențiază adaptabilitatea și aplicabilitatea tehnologiei de automatizare pentru îmbunătățirea experienței studențești. Utilizând roboți de chat bazați pe IA și procesare a limbajului natural, am demonstrat că este posibil să simplificăm semnificativ interacțiunile zilnice ale studenților cu administrația universității, permițându-le să acceseze rapid informații esențiale, cum ar fi orele și localizările sălilor de curs.

Implementarea acestui sistem nu numai că eficientizează gestionarea timpului studenților, dar oferă și o platformă accesibilă și interactivă care îmbunătățește comunicarea între studenți și facultate. Pe măsură ce acești roboți de chat evoluează, capacitatea lor de a învăța și de a personaliza interacțiunile va continua să crească, oferind servicii tot mai complexe și adaptate nevoilor individuale ale utilizatorilor.

Mai mult decât atât, folosirea limbajului C# și a platformei .NET în dezvoltarea acestui robot de chat demonstrează flexibilitatea și eficiența acestor instrumente în crearea de aplicații robuste, scalabile. Aceste tehnologii oferă o bază solidă pentru dezvoltarea continuă și îmbunătățirea robotului de chat, permițând adaptarea ușoară la cerințele și schimbările dinamicului mediu academic.

Robotul de chat RusBot este destinat în special studenților ce se află la început de drum în Universitatea din Petroșani. Crearea unui asistent digital ce va fi mereu la dispoziția lor va ajuta la automatizarea și eficientizarea lucrurilor, creând un loc special în Universitate unde informația va fi mereu disponibilă. Mai mult decât atât, asistentul RusBot va putea fi dezvoltat mai departe pentru a include mult mai multe informații ce le-ar servi studenților, precum informații legate de perioada de practică, informații cu privire la burse, sau hotărâri din structurile universitare.

Bibliografie:

- 1. <https://www.oracle.com/ro/chatbots/what-is-a-chatbot/>
- 2. <https://code-it.ro/tutoriale-c-prezentarea-limbajului/>
- 3. <https://www.c-sharpcorner.com/UploadFile/raj1979/simple-mvvm-pattern-in-wpf/>
- 4. <https://martinfowler.com/eaDev/PresentationModel.html>