**Documentul de specificare a cerințelor**

**Software Requirements Specification**

**(SRS) Document**

**<Chatbot AI pentru un magazin online >**

**<16.04.2025><1.0.0>**

**<Universitatea Tehnica de Constructii Bucuresti>**

**<Automatica si Informatica Aplicata>**

|  |
| --- |
| **Istoricul versiunilor** |

| Versiune | Autor(i) principali | Descriere versiune | Dată |
| --- | --- | --- | --- |
|  |  |  |  |

|  |
| --- |
| **Revizuiri și aprobări** |

Istoric aprobări

| Aprobă | Versiune | Semnătură | Dată |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |

Istoric revizuiri

| Revizor | Versiune | Semnătură | Dată |
| --- | --- | --- | --- |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

|  |
| --- |
| **Cuprins** |

[1. Introducere 4](#_Toc190689617)

[1.1 Scopul 4](#_Toc190689618)

[1.2 Convenții ale documentului 4](#_Toc190689619)

[1.3 Audiență țintă 4](#_Toc190689620)

[1.4 Sfera de aplicare 4](#_Toc190689621)

[1.5 Referințe 4](#_Toc190689622)

[**2 Descriere generală 4**](#_Toc190689623)

[**2.1 Perspectiva produsului 4**](#_Toc190689624)

[**2.2 Caracteristici ale produsului 4**](#_Toc190689625)

[**2.3 Clase și caracteristici ale utilizatorilor 4**](#_Toc190689626)

[**2.4 Mediul de operare 5**](#_Toc190689627)

[**2.5 Constrângeri de proiectare și de implementare 5**](#_Toc190689628)

[**2.6 Presupuneri și dependențe 5**](#_Toc190689629)

[**3 Cerințele sistemului 5**](#_Toc190689630)

[**3.1 Funcționalitatea 1 5**](#_Toc190689631)

[**3.1.1 Descriere generală 5**](#_Toc190689632)

[**3.1.2 Flux de interacțiune (scenarii de utilizare) 5**](#_Toc190689633)

[**3.1.3 Condiții prealabile și constrângeri 6**](#_Toc190689634)

[**3.1.4 Detaliere cerință 6**](#_Toc190689635)

[**3.1.5 Scenarii de eroare și gestionarea excepțiilor 6**](#_Toc190689636)

[**3.1.5 Dependențe și interacțiuni cu alte funcționalități 6**](#_Toc190689637)

[**3.2 Cerința funcțională 2 6**](#_Toc190689638)

[3.3 .... 6](#_Toc190689639)

[4 Cerințe pentru interfețe 6](#_Toc190689640)

[4.1 Interfețe cu utilizatorul 6](#_Toc190689641)

[4.2 Interfețe hardware 6](#_Toc190689642)

[4.2.1 Configurații Minime Recomandate 7](#_Toc190689643)

[4.2.2 Dispozitive Externe Compatibile 7](#_Toc190689644)

[4.3 Interfețe de comunicare 7](#_Toc190689645)

[4.3.1 Protocoale și Standarde de Comunicare 7](#_Toc190689646)

[4.3.2 Cerințe de Securitate în Comunicare 7](#_Toc190689647)

[4.4 Interfețe software 7](#_Toc190689648)

[4.4.1 Tehnologii Utilizate 7](#_Toc190689649)

[4.4.2 Servicii Externe și API-uri 7](#_Toc190689650)

[5 Cerințe non-funcționale 8](#_Toc190689651)

[5.1 Cerințe de performanță 8](#_Toc190689652)

[5.2 Cerințe de siguranță 8](#_Toc190689653)

[5.3 Cerințe de securitate 8](#_Toc190689654)

[5.4 Atribute de calitate ale software-ului 8](#_Toc190689655)

[6 Alte cerințe 8](#_Toc190689656)

[7 Anexe 8](#_Toc190689657)

[7.1 Anexa A: Glosar 8](#_Toc190689658)

[7.2 Anexa B: Modele de Analiză 8](#_Toc190689659)

[7.3 Anexa C: Listă de Probleme 9](#_Toc190689660)

# Introducere

## Scopul

## Acest document are ca scop definirea cerințelor funcționale și non-funcționale pentru dezvoltarea unui chatbot inteligent pentru un magazin online de sneakeri. Documentul furnizează specificațiile tehnice necesare pentru dezvoltarea, testarea și implementarea acestui sistem. Scopul principal al acestui chatbot este de a oferi asistență clienților prin răspunsuri automate și precise referitoare la produsele disponibile, stocuri și informații suplimentare legate de articolele comercializate.

## Textul scris cu font standard reprezintă cerințele generale. Abrevierile precum NLP (Natural Language Processing), API (Application Programming Interface) și CSV (Comma Separated Values) sunt explicate în glosar. Codul sursă și identificatorii sunt redați cu font monospace.

## Acest document este destinat echipei de dezvoltare (frontend și backend), echipei de testare QA, managerului de proiect, clientului (magazinul de sneakeri) și personalului de suport care va întreține chatbotul după lansare.

## Proiectul urmărește implementarea unui asistent AI pentru magazinul online de sneakeri, care poate răspunde automat la întrebări despre produse, disponibilitate și detalii despre articole. Acesta va ajuta la reducerea volumului de muncă pentru personalul uman și va îmbunătăți experiența utilizatorului.

## - Documentația oficială FastAPI:

## https://fastapi.tiangolo.com - Documentația spaCy:

## https://spacy.io/usage -Cerințe proiect licență, facultate - Fișierul CSV cu produsele magazinului: sneakeri.csv

# Descriere generală

## Perspectiva produsului

## Acest chatbot este o aplicație software inteligentă ce are rolul de asistent virtual pentru un magazin online de sneakeri. El permite clienților să adreseze întrebări despre produse, iar sistemul răspunde folosind un model NLP (spaCy) aplicat asupra unui fișier CSV care conține catalogul de articole. Aplicația funcționează local și comunică prin API REST (FastAPI).

## Caracteristici ale produsului

## Acceptă întrebări în limbaj natural despre:

## brand (Nike, Adidas etc.)

## model (Air Force 1 etc.)

## culoare, preț, mărime, stoc

## Oferă răspunsuri structurate în format JSON sau text

## Utilizează scoruri de similaritate pentru potrivirea răspunsurilor

## Funcționează local fără conexiune la internet

## Nu are nevoie de baze de date relaționale

## Clase și caracteristici ale utilizatorilor

|  |  |
| --- | --- |
| Tip utilizator | Descriere |
| Clientii | Pun întrebări despre produse în interfața chatbotului |
| Administratorii | Gestioneaza baza de date si verifica toate functionalitatile platformei |

## Mediul de operare

 Sistem de operare: Windows 10+, Linux (Ubuntu), MAC

 Limbaj: Python 3.10+

 Biblioteci: FastAPI, pandas, spaCy, uvicorn

 Interfață: locală, accesibilă prin browser sau consolă (localhost:8000)

## Constrângeri de proiectare și de implementare

## Proiectul este limitat la utilizarea unui fișier CSV ca sursă de date, fără integrare cu baze de date relaționale sau API-uri externe. Chatbotul trebuie să funcționeze complet offline, pe un server local sau pe o stație de lucru individuală. Din punct de vedere al tehnologiei, procesarea limbajului natural trebuie realizată exclusiv cu biblioteca spaCy, iar backend-ul aplicației va fi dezvoltat folosind FastAPI. Nu se permit soluții comerciale externe sau servicii cloud, iar codul trebuie să fie scris clar, comentat și modular.

## Presupuneri și dependențe

Se presupune că fișierul sneakeri.csv este disponibil și va fi actualizat manual, periodic, de către administratorii magazinului. Se consideră că utilizatorii vor formula întrebările în limbaj natural simplificat, fără ambiguități lingvistice complexe. Aplicația va rula exclusiv local, iar mediul de dezvoltare va avea preinstalate toate pachetele necesare în ecosistemul Python, precum spaCy, FastAPI și pandas. Nu se presupune existența unei conexiuni active la internet în timpul rulării, iar datele vor fi gestionate doar local.

# Cerințele sistemului

## Funcționalitatea 1

### Descriere generală

Această funcționalitate permite utilizatorilor să adreseze întrebări în limbaj natural despre produsele disponibile în magazinul online (de exemplu: brand, model, culoare, preț, mărime, stoc). Sistemul va interpreta întrebarea, va extrage entitățile relevante și va returna un răspuns structurat și clar. Funcționalitatea este esențială pentru rolul principal al chatbotului ca asistent AI

### Flux de interacțiune (scenarii de utilizare)

 Utilizatorul accesează interfața chatbotului.

 Introduce o întrebare precum: „Aveți Adidas Superstar pe alb, mărimea 42?”

 Chatbotul interpretează întrebarea și caută în fișierul CSV produsul potrivit.

 Sistemul răspunde cu detalii despre produs: „Avem Adidas Superstar alb, mărimea 42, în stoc (3 bucăți), la prețul de 450 RON.”

### Condiții prealabile și constrângeri

 Fișierul CSV trebuie să fie prezent și accesibil.

 Întrebările trebuie să conțină suficiente detalii (minim brand și model) pentru a permite filtrarea.

 NLP trebuie să funcționeze offline (doar cu spaCy, fără API-uri externe).

### Detaliere cerință

Aceasta este lista cerințelor care definesc exact comportamentul funcționalității. Fiecare cerință trebuie să fie:

* **Clară** (fără ambiguități)
* **Testabilă** (poate fi verificată printr-un test).

*<Fiecare cerință ar trebui identificată în mod unic printr-un număr de secvență sau o etichetă semnificativă:>* REQ-1, REQ-2, etc.

REQ-2:

*Exemplu:*

|  |  |  |  |
| --- | --- | --- | --- |
| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| **REQ-1** | Sistemul trebuie să accepte întrebări despre brand/model/culoare/mărime/preț/stoc | Ridicată | Orice întrebare validă returnează un răspuns în < 2 secunde |
| **REQ-2** | |  | | --- | |  |  |  | | --- | | Sistemul trebuie să extragă automat entitățile din întrebarea utilizatorului | | Ridicata | Entitățile sunt corect identificate și evidențiate |
| **REQ-3** | |  | | --- | |  |  |  | | --- | | Sistemul trebuie să returneze un răspuns structurat (text JSON sau natural) | | Ridicata | Răspunsul conține toate detaliile relevante despre produs |
| **REQ-4** | Dacă nu există produsul căutat, sistemul trebuie să ofere un mesaj clar | Medie | Mesaj: „Nu am găsit produsul căutat” |

### ****Scenarii de eroare și gestionarea excepțiilor****

 Dacă utilizatorul introduce un brand/model inexistent → se returnează un mesaj de informare.

 Dacă fișierul CSV nu poate fi încărcat → se returnează o eroare internă: „Eroare server – date indisponibile.”

 Dacă întrebarea este incompletă → chatbotul solicită mai multe detalii: „Te rog să menționezi brandul/modelul.”

### ****Dependențe și interacțiuni cu alte funcționalități****

 Această funcționalitate depinde direct de încărcarea fișierului sneakeri.csv.

 Este corelată cu funcționalitatea de procesare NLP (recunoașterea entităților).

 Funcționează împreună cu API-ul FastAPI care primește și prelucrează interogările.

## Cerința funcțională 2

## 3.2.1 Descriere general

## Această funcționalitate permite utilizatorilor să afle dacă un anumit produs se mai află în stoc, pe baza întrebărilor formulate natural.

## 3.2.2 Flux de interacțiune

## Utilizatorul întreabă: „Mai aveți Nike Air Max 270, mărimea 43?”

## Sistemul extrage entitățile, caută produsul și verifică stocul.

## Răspunsul este: „Da, avem 2 bucăți în stoc.” sau „Produsul nu mai este disponibil.”

# Cerințe pentru interfețe

## Interfețe cu utilizatorul

## Sistemul oferă o interfață minimă de tip REST API, accesibilă printr-un browser sau aplicație externă. Utilizatorul interacționează cu chatbotul prin transmiterea unui query, de exemplu:

## http://localhost:8000/chat/?query=Mai%20aveti%20Nike%20Air%20Max%20alb%20marimea%2042

## Răspunsul este afișat sub formă de text sau JSON, în funcție de modul de rulare. Într-o variantă extinsă, chatbotul poate fi integrat într-o interfață web (HTML + CSS + JavaScript) care imită o fereastră de chat (similară cu Telegram/WhatsApp).

## Exemple de interacțiune:

## Input: „Ce mărimi sunt disponibile pentru Adidas Superstar?”

## Output: „Pentru Adidas Superstar avem mărimile 40, 41, 42 și 44.”

## Interfețe hardware

Nu există cerințe hardware speciale. Proiectul rulează pe orice sistem desktop/laptop cu:

* Procesor minim Intel i3 sau echivalent
* 4 GB RAM
* 300 MB spațiu liber pentru fișiere și biblioteci

## Interfețe de comunicare

## Comunicarea dintre componente se face prin apeluri HTTP locale. Utilizatorul transmite un mesaj către endpoint-ul /chat/, iar serverul FastAPI procesează întrebarea, extrage entitățile cu spaCy, caută în fișierul CSV folosind pandas și returnează răspunsul în format JSON.

### ****Cerințe de Securitate în Comunicare****

Utilizatorul fa fi nevoit sa isi faca un cont si sa se autentifice intr-o pagina de login sau pe o platforma web aferenta aplicatiei. Deasemenea, datele din baza de date sau din fisierul csv sunt criptate cu ajutorul unei aplicatii ce foloseste protocolul auth 2.0.

## Interfețe software

Aplicația chatbot utilizează un set minim, dar eficient, de componente software pentru a funcționa local și offline. Backend-ul este construit folosind limbajul de programare Python, în versiunea 3.10 sau superioară, ales datorită ecosistemului vast de biblioteci și comunității de suport. Pentru crearea și expunerea rapidă a API-urilor REST, se folosește framework-ul **FastAPI**, care oferă viteză, simplitate și compatibilitate excelentă cu procesarea asincronă.

Rularea aplicației se face local cu ajutorul serverului de dezvoltare **Uvicorn**, compatibil cu FastAPI. Pentru procesarea datelor și încărcarea fișierului sneakeri.csv, se utilizează biblioteca **pandas**, iar pentru interpretarea limbajului natural – identificarea entităților precum brand, model sau culoare – se folosește **spaCy**, cu modelul lingvistic en\_core\_web\_sm, preantrenat pentru limba engleză.

Chatbotul poate fi testat fie printr-un browser, accesând endpoint-ul /chat/, fie prin instrumente CLI precum curl, fie printr-o interfață web HTML-CSS locală, creată opțional pentru un aspect vizual prietenos.

### Tehnologii Utilizate

* **Backend:**
* FastAPI – pentru construcția rapidă a unui API RESTful modern
* Uvicorn – server ASGI pentru rulare locală
* spaCy – pentru NLP (Natural Language Processing), utilizând modelul lingvistic en\_core\_web\_sm
* pandas – pentru citirea și manipularea datelor din fișierul CSV
* Python Standard Library – pentru operațiuni uzuale precum filtre, transformări de text și gestionarea tipurilor de date
* **Frontend:** CSS/HTML/JAVASCRIPT
* **Bază de date:** SQL Oracle si fisiere csv

### ****Servicii Externe și API-uri****

| **Serviciu/Interfață** | **Utilizare** |
| --- | --- |
| **[Serviciu/API 1]** | FastAPI |
| **[Serviciu/API 2]** | Uvicorn |
| **[Serviciu/API 3]** | spaCy |

# Cerințe non-funcționale

## Cerințe de performanță

Sistemul trebuie să răspundă la întrebările utilizatorilor în cel mai scurt timp posibil, fără întârzieri semnificative. De asemenea, trebuie să gestioneze simultan mai mulți utilizatori fără a afecta performanța. Se impune ca timpul de răspuns al aplicației să nu depășească 2 secunde pentru fiecare interogare. În cazul în care există volume mari de cereri, sistemul trebuie să fie capabil să le proceseze fără a depăși un număr de 100 de cereri simultane pe minut.

## Cerințe de siguranță

Sistemul trebuie să asigure confidențialitatea și integritatea datelor transmise între utilizator și server. Toate comunicările între utilizator și server trebuie să fie criptate folosind protocoale de securitate precum **HTTPS** sau **TLS** (Transport Layer Security) pentru a proteja datele sensibile (de exemplu, întrebările utilizatorilor). În plus, trebuie implementate măsuri de securitate pentru a preveni atacurile de tip **SQL injection**, **Cross-site scripting (XSS)** și **Cross-site request forgery (CSRF)**.

De asemenea, orice acces la fișierele de date și configurare ale aplicației va fi restricționat la utilizatori autorizați, iar orice modificare a fișierului **sneakeri.csv** va fi monitorizată și înregistrată pentru a asigura integritatea acestuia.

## Cerințe de securitate

Sistemul trebuie să respecte standardele de securitate pentru protejarea datelor utilizatorilor și a comunicărilor între client și server. Aceste cerințe de securitate vor include măsuri de protecție împotriva accesului neautorizat, protecția datelor în tranzit și prevenirea atacurilor cibernetice.

1. **Criptarea comunicațiilor**  
   Toate datele transmise între utilizator și server trebuie să fie criptate utilizând protocoale securizate, cum ar fi **TLS 1.2** sau versiuni ulterioare. Astfel, comunicările sensibile, precum întrebările utilizatorilor și răspunsurile sistemului, vor fi protejate împotriva interceptării.
2. **Autentificare și autorizare**  
   Deși sistemul nu include autentificare utilizator, se recomandă ca viitoarele implementări să prevadă autentificare prin **OAuth 2.0** sau alte mecanisme de autentificare securizată. Utilizatorii autorizați vor avea acces doar la funcționalitățile permise în cadrul aplicației.
3. **Prevenirea atacurilor**  
   Sistemul trebuie să fie protejat împotriva atacurilor comune, cum ar fi **SQL injection**, **Cross-Site Scripting (XSS)** și **Cross-Site Request Forgery (CSRF)**. Validarea corectă a intrărilor utilizatorilor și sanitizarea acestora va preveni executarea de cod malițios.
4. **Gestionarea accesului la fișierele sensibile**  
   Accesul la fișierele de configurare, inclusiv **sneakeri.csv**, trebuie să fie restricționat. Doar utilizatorii autorizați (de exemplu, administratorii sistemului) vor avea permisiunea de a modifica fișierul CSV. Orice modificare a acestuia va fi monitorizată și jurnalizată pentru a asigura integritatea datelor.
5. **Audit și jurnalizare**  
   Toate accesările și modificările fișierelor și datelor sensibile vor fi înregistrate într-un sistem de audit. Acesta va permite monitorizarea activității și detectarea unor posibile breșe de securitate. În cazul unui incident, datele din jurnal vor fi utilizate pentru a investiga sursa atacului.
6. **Backup și recuperare**  
   Sistemul trebuie să implementeze proceduri de backup și recuperare pentru fișierele esențiale (de exemplu, fișierul CSV) și să permită restaurarea rapidă a acestora în cazul unor evenimente neprevăzute, cum ar fi pierderea datelor.

.

## Atribute de calitate ale software-ului

Software-ul trebuie să îndeplinească următoarele atribute de calitate pentru a asigura o experiență optimă a utilizatorilor:

* **Fiabilitate**: Sistemul trebuie să funcționeze corect și continuu fără întreruperi neprevăzute, având un timp de disponibilitate de 99% sau mai mult.
* **Ușurința în întreținere**: Codul sursă trebuie să fie bine documentat și să permită actualizări și modificări ușoare în viitor, fără a afecta funcționalitatea existentă.
* **Flexibilitate**: Aplicația trebuie să permită adăugarea de noi funcționalități sau modificarea celor existente fără a necesita schimbări majore ale structurii de date sau ale interfeței.
* **Portabilitate**: Sistemul trebuie să fie compatibil cu diverse sisteme de operare (Windows, Linux) și să poată fi implementat pe diferite platforme hardware fără probleme majore.
* **Scalabilitate**: Deși proiectul nu are cerințe de scalabilitate pe termen scurt, arhitectura trebuie să permită extinderea aplicației pentru a gestiona un număr mai mare de utilizatori și date, dacă este necesar în viitor.

# Alte cerințe

Proiectul trebuie să respecte reglementările legale referitoare la protecția datelor cu caracter personal, cum ar fi **Regulamentul General privind Protecția Datelor (GDPR)**, având în vedere că chatbotul ar putea interacționa cu utilizatori care își împărtășesc informații personale în cadrul conversațiilor. Aceste reglementări includ:

* **Confidențialitatea datelor utilizatorilor**: Toate informațiile introduse de utilizatori trebuie procesate și stocate conform reglementărilor de confidențialitate, iar utilizatorii trebuie să fie informați despre modul în care sunt gestionate datele lor.
* **Protecția datelor**: Sistemul nu va colecta date personale fără consimțământul utilizatorului. Datele colectate de la utilizatori vor fi folosite doar pentru a răspunde la întrebările legate de produsele din magazin.
* **Ștergerea datelor**: Utilizatorii trebuie să aibă posibilitatea de a solicita ștergerea datelor lor conform reglementărilor GDPR.

# Anexe

## Anexa A: Glosar

|  |  |
| --- | --- |
| API | Application Programming Interface |
| CSV | Comma Separated Values |
| NLP | |  | | --- | |  |  |  | | --- | | Natural Language Processing | |
| JSON | JavaScript Object Notation |
| REST | Representational State Transfer |
| HTTP | |  | | --- | |  |  |  | | --- | | Hypertext Transfer Protocol | |
| HTTPS | |  | | --- | |  |  |  | | --- | | Hypertext Transfer Protocol Secure | |
|  |  |

## 

## Anexa B: Listă de Probleme

| **Problemă** | **Status** | **Comentarii** |
| --- | --- | --- |
| Integrarea cu API extern | Deschis | Necesită decizii privind utilizarea API-urilor externe pentru actualizări în timp real. |
| Procesarea complexă a întrebărilor complexe | În așteptare | Trebuie testat sistemul cu întrebări mai complexe pentru a evalua performanța NLP-ului. |
| Îmbunătățirea interfeței grafice | Rezolvat | S-au adus îmbunătățiri pentru a face interfața mai intuitivă. |

# Exemplu (parțial) de completare

## ****Funcționalitate: Autentificare Utilizator****

### ****3.1.1 Descriere Generală****

Această funcționalitate permite utilizatorilor să se autentifice în sistem folosind o adresă de email și o parolă. Este o funcționalitate esențială pentru gestionarea accesului utilizatorilor.

### ****3.1.2 Flux de Interacțiune****

1. Utilizatorul introduce email-ul și parola
2. Sistemul validează datele
3. Dacă datele sunt corecte → utilizatorul este autentificat
4. Dacă datele sunt greșite → apare un mesaj de eroare

### ****3.1.3 Condiții Prealabile și Constrângeri****

* Utilizatorul trebuie să aibă un cont activ
* Sistemul trebuie să fie conectat la baza de date pentru validarea acreditărilor

### ****3.1.4 Cerințe Funcționale Detaliate****

| **Cerință** | **Descriere** | **Prioritate** | **Criterii de acceptanță** |
| --- | --- | --- | --- |
| **REQ-1** | Constrangere caractere speciale destinate introducerii unui format corect de email | Mediu | Campul accepta doar date introduce ce respecta formatul email@test.ro |
| **REQ-2** | Verificarea in baza de date a existentei datelor introduse pentru autentificare | Ridicata | Se verifica daca in baza de date exista inregistrari specifice numelui si parolei din campurile de logare |

### ****3.1.5 Scenarii de Eroare și Gestionare a Excepțiilor****

* Dacă utilizatorul introduce o parolă greșită de 5 ori → cont blocat temporar
* Dacă baza de date nu este accesibilă → mesaj de eroare „Serviciu indisponibil”

### ****3.1.6 Dependențe și Interacțiuni cu Alte Funcționalități****

* Depinde de sistemul de gestionare a utilizatorilor
* Se integrează cu sistemul de recuperare a parolei

**4.1 Interfețe cu utilizatorul**

* Sistemul trebuie să ofere o interfață grafică intuitivă, optimizată pentru utilizare pe desktop și mobil.
* Ecranele principale trebuie să includă:
  + **Ecran de autentificare:** câmp pentru email/parolă, buton „Login”
  + **Ecran principal:** navigare prin meniul aplicației
  + **Ecran de setări:** opțiuni de personalizare a contului

**4.2 Interfețe Hardware**

* Aplicația trebuie să ruleze pe următoarele configurații minime:
  + **PC/Laptop:** Procesor minim Intel i5, 8GB RAM, Windows 10/Linux
  + **Smartphone:** Android 9+ sau iOS 14+
* Dispozitive externe compatibile: cititoare de amprente, module NFC

**4.3 Interfețe de Comunicare**

* Aplicația va folosi următoarele protocoale:
  + **HTTP/HTTPS** pentru interacțiunea client-server
  + **WebSockets** pentru notificări în timp real
  + **MQTT** pentru transmiterea datelor IoT
* Datele trebuie să fie criptate utilizând **TLS 1.2+** pentru a asigura securitatea comunicațiilor.

**4.4 Interfețe Software**

* Backend-ul aplicației se va baza pe **Django + MySQL**.
* Aplicația va interacționa cu următoarele API-uri externe:
  + **Google Maps API** pentru localizare
  + **Stripe API** pentru procesarea plăților
  + **Firebase Authentication** pentru gestionarea conturilor de utilizatori