



***Documentul de Proiectare a
Soluției Aplicației Software
- Software Design Document -
COVID INFERENCE***

Membrii echipei:

Cimbir Luciana 341C1

Badea Dan Gabriel 343C2

Manea Andrei 343C2

Monea Damian 343C2

Negru Mihai Vlad 343C2

Roșianu Sergiu Constantin 343C2

Îndrumător:

Laura Antonache

CUPRINS

1. Scopul documentului	3
2. Obiective	3
3. Conținutul documentului	3
4. Modelul datelor	3
4.1. Structuri de date globale.....	3
4.2. Structuri de date de legătura	3
4.3. Structuri de date temporare.....	4
4.4. Formatul fișierelor utilizate	4
5. Modelul arhitectural	4
5.1 Diagrama Arhitecturala	4
6. Modelul interfeței cu utilizatorul	5
7. Elementele de testare	5

1. Scopul documentului

Acest document are rolul de a descrie detaliat soluția proiectată pentru detectarea șansei de infectare cu COVID a pacienților folosind informațiile medicale primite de la aceștia.

Documentul oferă o imagine de ansamblu asupra structurii, componentelor sistemului software, legăturii între componentele arhitecturale cât și a modului de utilizare al soft-ului.

2. Obiective

- Vizualizarea evoluției proiectului de la faza de început până la faza finală.
- Evaluarea schimbării principiilor în funcție de problemele intervenite pe parcursul dezvoltării produsului.
- Analiza modelului arhitectural.
- Analiza performanțelor.

3. Conținutul documentului

Documentul conține patru părți:

- Modelul datelor- cuprinde structurile de date folosite în implementare
- Modelul arhitectural și modelul componentelor – cuprinde descrierea componentelor arhitecturale precum și întreaga arhitectură a sistemului
- Modelul interfață cu utilizatorul – cuprinde descrierea interfeței cu utilizatorul
- Elemente de testare – conține o serie de orientări de testare

4. Modelul datelor

4.1. Structuri de date globale

- **Simptom_dict**: dictionarul de simptome, folosit în parsarea datelor.

4.2. Structuri de date de legătura

- Se folosește un fișier de configurare(**config.json**), care conține modul de rulare și path-ul către dataset-ul de antrenare.
- Pentru citirea datelor din fișierul de intrare se folosește un obiect de tip **DataFrame** (din biblioteca **Pandas**). Acesta este folosit în interiorul clientului și trimis mai departe către server prin intermediul unui **JSON Array**.

4.3. Structuri de date temporare

- Aplicatia foloseste un dictionar in care pastreaza feature-urile, fiind util in antrenarea modelului.

4.4. Formatul fișierelor utilizate

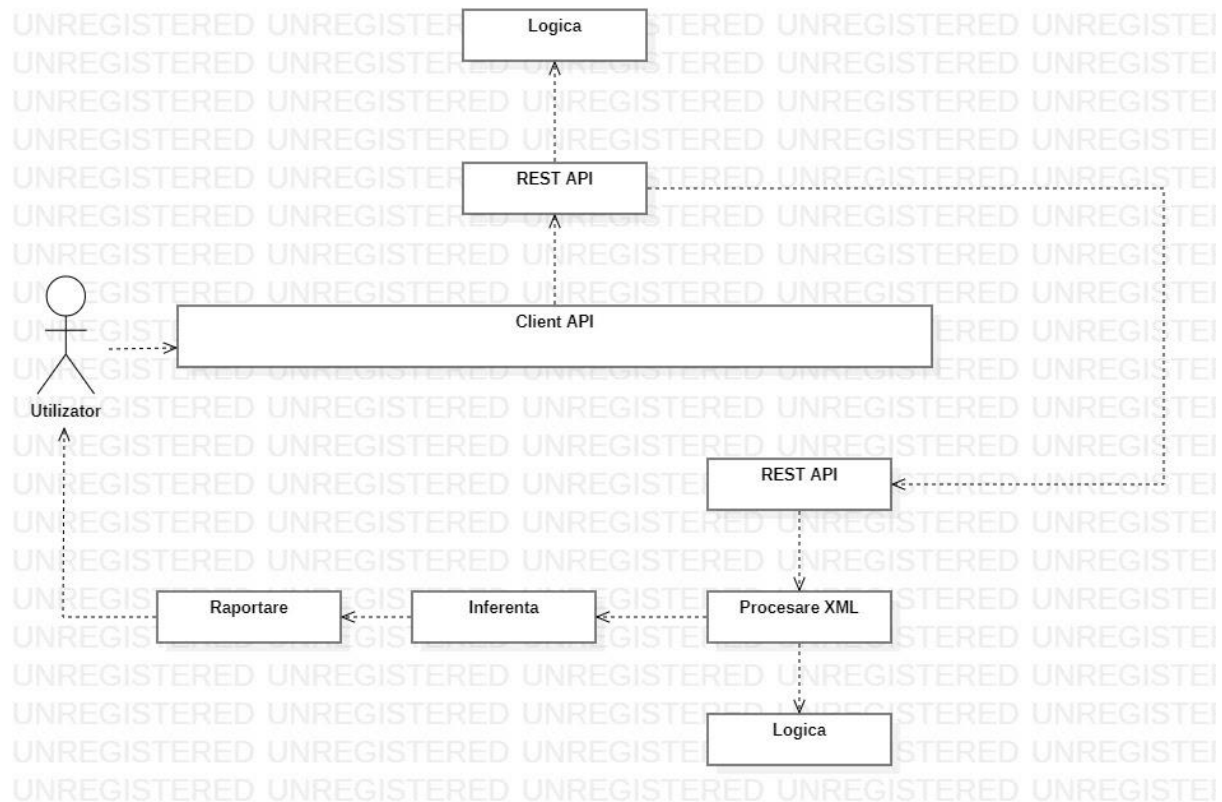
- Aplicatia foloseste fisiere in format .py, ce implementeaza functionalitatile aplicatiei.
- Pentru fisierul de input se folosesc fisiere in format .xls.
- Aplicatia pastreaza un fisier cu date de configurare, in format .json.

5. Modelul arhitectural

Arhitectura sistemului nostru de inferenta este bazata pe indeplinirea a trei obiective: primul este de a parse set-ul de date pentru obtinerea unor date mult mai “curate” si mai usor de utilizat si impartirea acestora in date de invatare si date de testare, al doilea este de a utiliza datele parseate pentru a putea antrena un algoritm ce ofera ca rezultat probabilitatea infectarii cu COVID si al treilea obiectiv este de a include functiile proiectului si in principal cea de inferenta intr-o componenta software de tip API.

Arhitectura folosita in aceasta aplicatie este una de tip back-end - api client. In cadrul acestei aplicatii componenta back-end returneaza pe baza unei retele neurale antrenate si a unui set de date trimis, probabilitatea de infectare cu COVID. Partea de client api ofera o interfata cu ajutorul careia utilizatorul poate printr-un set de date furnizat sa obtina probabilitatea de infectare.

5.1 Diagrama Arhitecturala



5.2 Descriere Componente

Aplicația constă din următoarele module interconectate:

Client:

- **Modulul Client API** - Trimite un json Array către modulul serverului
- **Modulul REST API** - Interfața pentru comunicarea client-server
- **Modulul Logic** - Este responsabil de coordonarea celorlalte module astfel încât rezultatele rulării lor să reflecte logica de baza a aplicației.

Server:

- **Modulul REST API** - Interfața pentru comunicarea client-server
- **Modulul Procesare XML** - Este responsabil de operațiile ce vizează direct fișierele XML, precum operații de validare, parsare, creare și editare de fișiere XML.
- **Modulul Logic** - Este responsabil de coordonarea celorlalte module astfel încât rezultatele rulării lor să reflecte logica de baza a aplicației.
- **Modulul Inferenta** - Primește json Array-ul ce trebuie parcurs și returnează inferența sansei de infectare cu COVID
- **Modulul Raportare** - Este responsabil cu exploatarea informațiilor primite în vederea generării de statistici, analize și predicții medicale. De asemenea trimite către client json array-ul cu răspunsul.

6. Modelul interfeței cu utilizatorul

Modelul presupune un REST API Client Server prin care sunt publicate date în server, acesta se ocupa prin modulul de parsarea și antrenare a datelor care la final va transmite către Client un JSON ARRAY cu predicțiile oferite pentru toți pacienții primiți.

7. Elementele de testare

7.1 Componente Critice

Performanța modulului de Procesare XML are o influență decisivă asupra performanței globale a aplicației, alături de performanța API-ului utilizat. Astfel, viteza de procesare a documentelor XML și transmiterea datelor prin rețea între componente sunt principalele părți ce influențează performanța soluției noastre software.

7.2 Alternative

Se poate utiliza un parser implementat de către un third-party care este specializat în documente medicale. În vederea scăderii timpilor de rețea, se pot utiliza algoritmi eficienți de comprimare a informațiilor transmise pe rețea.