Università degli Studi di Napoli Federico II

Scuola Politecnica e delle Scienze di Base

Dipartimento di Ingegneria Elettrica e Tecnologie dell’Informazione

Corso di Laurea Magistrale in Ingegneria Informatica

Immagine che contiene cerchio, simbolo, schizzo

Descrizione generata automaticamente

**ELABORATO DI AI SYSTEM ENGINEERING**

**HEALTHGATE: Un Sistema LLM per il Supporto Decisionale e la Riduzione del Sovraffollamento nei Pronto Soccorso**

*Prof.re Roberto Pietrantuono*

A.A. 2024-25

Studenti:

Campanella Alessandro M63001697

Castaldi Rita M63001667

Di Fraia Simone M63001678

Indice

[Business Idea 3](#_Toc209626851)

[Requirement Analysis 3](#_Toc209626852)

[Stakeholders 3](#_Toc209626853)

[Requisiti Funzionali 3](#_Toc209626854)

[Requisiti Non Funzionali 4](#_Toc209626855)

[Casi d’uso 5](#_Toc209626856)

[Use Case Diagram 9](#_Toc209626857)

[Design 10](#_Toc209626858)

[Prototype Development 10](#_Toc209626859)

[Continuous Monitoring & Testing, (Local) Deployment 10](#_Toc209626860)

# Business Idea

Uno dei problemi più rilevanti del sistema sanitario odierno è rappresentato dall’elevato numero di accessi impropri alle strutture di emergenza. Molti pazienti si rivolgono al pronto soccorso anche in presenza di condizioni cliniche non gravi, spesso per mancanza di informazioni, per timore o per difficoltà ad accedere ad altre forme di assistenza. Questo fenomeno determina un sovraccarico del sistema, con conseguenti rallentamenti nella gestione dei casi realmente urgenti, incremento dei tempi di attesa e aumento dei costi complessivi, oltreché un aggravio del carico di lavoro per il personale sanitario. Tale fenomeno incide negativamente sia sulla qualità dell’assistenza fornita ai pazienti in condizioni critiche, sia sulla sostenibilità complessiva del sistema sanitario.

Dall’altro lato, in molte situazioni, i pazienti si trovano in condizioni di incertezza circa la gravità dei propri sintomi. Tale incertezza può portare da un lato a sottovalutare situazioni cliniche che richiederebbero invece un intervento tempestivo, dall’altro a ricorrere inutilmente al pronto soccorso per disturbi lievi, come citato precedentemente.

Un sistema in grado di analizzare i sintomi descritti dal paziente e di fornire un’indicazione chiara e immediata può contribuire a ridurre questa incertezza, aumentando la consapevolezza del cittadino e promuovendo un comportamento più appropriato e responsabile nell’uso delle strutture sanitarie.

# Requirement Analysis

## Stakeholders

Gli **stakeholder** sono tutte le figure che hanno un interesse o un impatto nell’uso e nello sviluppo del sistema. Nel nostro contesto, gli stakeholder sono due:

* I **Pazienti** che rappresentano i principali beneficiari, in quanto ricevono supporto decisionale e un orientamento sull’eventuale necessità di recarsi al pronto soccorso.
* **L’Operatore Sanitario del pronto soccorso** che può utilizzare i report clinici generati dal sistema per velocizzare la presa in carico del paziente.

## Requisiti Funzionali

Di seguito si riportano i requisiti funzionali, i quali descrivono le funzionalità e i servizi offerti al sistema:

1. Il sistema deve essere in grado di consentire ai pazienti e al personale sanitario di potersi registrare sulla piattaforma.
2. Il sistema deve prevedere meccanismi di autenticazione sicura per i *pazienti* e per il *personale sanitario*.
3. Il sistema deve essere in grado di consentire ai *pazienti* di registrare i propri sintomi mediante input vocale.
4. Il sistema deve consentire ai *pazienti* di riportare i propri sintomi in formato testuale.
5. Il sistema deve poter consentire ai *pazienti* di correggere manualmente il risultato di una trascrizione vocale.
6. Il sistema deve essere in grado di elaborare le informazioni fornite dal paziente e determinare se sia necessario recarsi al Pronto Soccorso, basandosi su criteri clinici predefiniti.
7. Il sistema deve generare automaticamente un report clinico relativo alle condizioni del paziente.
8. Il sistema deve permettere all’*operatore sanitario* di reperire e consultare i report clinici dei pazienti.
9. Il sistema deve permettere agli utenti di accedere alla cronologia completa delle consultazioni effettuate in precedenza.
10. Il sistema deve permettere all’*operatore sanitario* di aggiornare i report clinici con informazioni relative alle cure somministrate e al piano terapeutico.

## Requisiti Non Funzionali

Di seguito si riportano i requisiti non funzionali che definiscono o limitano le proprietà del sistema:

1. Il sistema deve garantire la memorizzazione sicura dei dati sensibili dei pazienti.
2. Il sistema deve determinare la scelta relativa all’accesso al Pronto Soccorso in tempi relativamente bassi, garantendo reattività.
3. Il modello deve essere leggero (*lightweight*) per consentire l’esecuzione delle funzionalità anche su dispositivi mobili o con risorse limitate.
4. Il modello deve fornire spiegazioni precise e comprensibili riguardo le decisioni prese dal sistema.
5. Il sistema deve essere scalabile per supportare un numero crescente di utenti e dati clinici senza degrado delle prestazioni.
6. Il sistema deve essere facilmente manutenibile ed estensibile per consentire aggiornamenti di documenti clinici ufficiali o modifiche funzionali.
7. Il sistema deve essere robusto alla presenza di drift nei dati clinici o nelle abitudini degli utenti.

## Casi d’uso

|  |  |
| --- | --- |
| Caso d’uso: | Registrazione sintomi e classificazione tramite input vocale |
| Attore primario | Paziente |
| Attore secondario | - |
| Descrizione | Il paziente registra i propri sintomi descrivendoli al microfono del dispositivo. Il sistema trascrive ed elabora l’input vocale salvando le informazioni. |
| Pre-condizioni | Il paziente è autenticato nel sistema |
| Sequenti di eventi principali | 1. Il paziente accede alla funzione “Registra sintomi vocali”. 2. Il sistema avvia la registrazione audio. 3. Il paziente descrive i sintomi a voce. 4. Il sistema converte il parlato in testo. 5. **IF** il paziente ritiene errata la trascrizione   5.1 Il paziente corregge manualmente della trascrizione vocale. 6. Il paziente conferma la trascrizione. 7. Il sistema memorizza i sintomi trascritti o corretti. 8. Il sistema elabora automaticamente i sintomi e genera una valutazione clinica. 9. Il sistema crea un report clinico sulle condizioni del paziente e lo archivia nel fascicolo elettronico. 10. Il paziente riceve conferma dell’avvenuto salvataggio. |
| Post-condizioni | I sintomi vengono trascritti, eventualmente corretti, valutati e salvati nel profilo del paziente. |
| Sequenza di eventi alternativi | 5.1 |

|  |  |
| --- | --- |
| Caso d’uso: | Registrazione sintomi e classificazione tramite input testuale |
| Attore primario | Paziente |
| Attore secondario | - |
| Descrizione | Il paziente registra i propri sintomi mediante un input testuale. Il sistema elabora l’input salvando le informazioni. |
| Pre-condizioni | Il paziente è autenticato nel sistema |
| Sequenti di eventi principali | 1. Il paziente accede alla funzione “Trascrivi sintomi”. 2. Il paziente descrive i sintomi mediante input testuale. 3. Il paziente revisiona e conferma l’elaborazione del sistema. 4. Il sistema memorizza i sintomi. 5. Il sistema elabora automaticamente i sintomi e genera una valutazione clinica. 6. Il sistema crea un report clinico sulle condizioni del paziente e lo archivia nel fascicolo elettronico. 7. Il paziente riceve conferma dell’avvenuto salvataggio. |
| Post-condizioni | I sintomi vengono valutati e salvati nel profilo del paziente |
| Sequenza di eventi alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Registrazione |
| Attore primario | Paziente, Operatore sanitario |
| Attore secondario | - |
| Descrizione | L’utente accede al sistema per potersi registrare. |
| Pre-condizioni | Nessuna |
| Sequenti di eventi principali | * 1. L’utente seleziona il suo ruolo (Paziente / Operatore Sanitario).   2. **IF** L’utente ha selezionato il ruolo di Paziente  2.1 L’utente inserisce Nome, Cognome, Sesso, Data di Nascita e Luogo di Nascita, Password.   **ELSE**   2.2 L’utente inserisce Nome, Cognome, Numero di Iscrizione all’Albo professionale, Password.   * 1. Il sistema valida i dati   2. **IF** I dati sono stati inseriti in maniera scorretta  4.1 Il sistema procede a richiedere nuovamente i dati all’utente   3. L’utente viene correttamente registrato |
| Post-condizioni | L’utente viene correttamente registrato nella piattaforma |
| Sequenza di eventi alternativi | 2.1/2.2/4.1 |

|  |  |
| --- | --- |
| Caso d’uso: | Autenticazione sicura |
| Attore primario | Paziente, Operatore sanitario |
| Attore secondario | - |
| Descrizione | L’utente accede al sistema tramite credenziali. |
| Pre-condizioni | L’utente si è precedentemente registrato alla piattaforma. |
| Sequenti di eventi principali | 1. L’utente inserisce le credenziali. 2. Il sistema valida i dati. 3. **IF** I dati sono stati inseriti in maniera scorretta  3.1 Il sistema procede a richiedere nuovamente i dati all’utente 4. L’utente ottiene l’accesso in base al profilo. |
| Post-condizioni | L’utente accede al sistema con il proprio ruolo. |
| Sequenza di eventi alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Consultazione cronologia personale |
| Attore primario | Paziente |
| Attore secondario | - |
| Descrizione | Il paziente consulta la cronologia dei propri referti. |
| Pre-condizioni | Il paziente è autenticato. |
| Sequenti di eventi principali | 1. Il paziente accede alla sezione “Cronologia”. 2. Il sistema mostra la lista completa dei report precedenti relativi all’utente. 3. Il paziente può selezionare un report per leggerlo. |
| Post-condizioni | La cronologia viene mostrata sul dispositivo. |
| Sequenza di eventi alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Consultazione report pazienti |
| Attore primario | Operatore sanitario |
| Attore secondario | - |
| Descrizione | L’operatore sanitario consulta un report clinico. |
| Pre-condizioni | Report clinico già generato. Operatore sanitario autenticato e autorizzato. |
| Sequenti di eventi principali | 1. L’operatore sanitario seleziona un paziente dalla lista. 2. L’operatore sanitario seleziona un report del paziente. 3. Il sistema mostra a video il report selezionato dall’operatore sanitario. |
| Post-condizioni | Il report viene mostrato sull’interfaccia |
| Sequenza di eventi alternativi | - |

|  |  |
| --- | --- |
| Caso d’uso: | Aggiornamento report |
| Attore primario | Operatore sanitario |
| Attore secondario | - |
| Descrizione | L’operatore sanitario aggiorna un report clinico aggiungendo informazioni relative a cure e piano terapeutico. |
| Pre-condizioni | Report clinico già generato. Operatore sanitario autenticato e autorizzato. |
| Sequenti di eventi principali | 1. L’operatore sanitario seleziona un report. 2. Aggiunge note su cure somministrate e terapie. 3. Salva le modifiche. |
| Post-condizioni | Il report aggiornato viene salvato nel sistema. |
| Sequenza di eventi alternativi | - |

## Use Case Diagram

Di seguito vi è il diagramma dei casi d’uso:

Immagine che contiene testo, diagramma, disegno

Il contenuto generato dall'IA potrebbe non essere corretto.

# Design

## Architettura a microservizi

Il design del sistema è stato realizzato seguendo un’architettura a microservizi, controllata dall’utente mediante un’applicativo mobile e web. In particolare, l’utente effettua sostanzialmente delle chiamate API, gestite da un API Gateway che provvede a chiamare il microservizio di interesse.

* 1. **API Gateway**: microservizio centrale, per orchestrare e astrarre l’accesso ai vari servizi del sistema complessivo;
  2. **Identity and Authenticator**:
  3. **User Profile and Consent Service (User Management)**: gestisce i profili dei pazienti e degli operatori sanitari;
  4. **Ingestion Service (Audio & Text)**: riceve file audio, metadati e input testuali;
  5. **Speech-to-Text Service**: in caso di audio upload, gestisce la conversione da parlato a testo libero mediante un modello preaddestrato;
  6. **NLP/Symptoms Extraction**: microservizio incaricato di trasformare il testo libero in entità strutturate. Output: symptoms extracted;
  7. **Clinical Decision Engine**: consuma I symptom extracted dall’NLP. Output: triage result;
  8. **Report Service**: genera e salva i report clinici [annota API e vedi differenza sincrono-asincrono];
  9. **Admin/Document Updater**:
  10. **Database Manager (?)**: se scegliamo di usare un solo db
  11. **Model Management & Monitoring Service (?)**: nella traccia si parla chiaramente di monitoraggio prestazioni, drift detection e aggiornamento modello. Ti servirebbe un microservizio dedicato al lifecycle del modello ML (training, retraining, validazione, monitoraggio). un servizio che opera sulla drift detection ecc potrebbe avere senso (e se ci mettiamo troppo nei guai facendolo, leviamo pure quello che abbiamo scritto nei requisiti, però qualcosa giusto per evidenziare che esiste un microservizio del genere e che ci abbiamo pensato ci starebbe metterla).
  12. **Data Validation & Quality Service (?)**: visto che il sistema si basa su input sensibili e rumorosi (audio/testo), potresti prevedere un microservizio che fa controllo di qualità sui dati prima che passino all’NLP. alla fine noi dobbiamo valutarli i dati, possiamo includere un microservizio che fa semplicemente questo controllo qualità (anche magari separando i prompt degli LLM, uno che si occupa solo di creare la categorizzazione e uno invece incaricato di "preparare" il testo con termini più tecnici)

Ogni microservizio, per coerenza, avrà accesso a un database apposito e specifico per la funzione svolta, come mostrato nel diagramma di seguito:

Immagine che contiene schermata, Modellazione 3D, diagramma, testo

Descrizione generata automaticamente

# Prototype Development

Best effort demo (codice Python)

# Continuous Monitoring & Testing, (Local) Deployment

Test di qualità (ad esempio, sicurezza, correttezza, protezione)…