



**Politecnico  
di Torino**

**Politecnico di Torino**

Corso di Laurea Magistrale in Ingegneria Biomedica

A.a. 2024/2025

Sessione di Laurea Marzo 2025

**Progettazione e sviluppo di  
un'applicazione web per la raccolta di  
dati clinici a supporto delle attività di  
ricerca**

Relatore:

Prof. FILIPPO MOLINARI

Candidata:

CATIANA GIORGIO

Correlatore:

Dott.ssa GIULIANA FIORELLA



# Sommario

La ricerca scientifica e la rivoluzione digitale stanno trasformando significativamente il settore dell'healthcare, da applicazioni di AI alla gestione di database clinici. Uno degli aspetti più significativi in fase di trasformazione riguarda la centralizzazione dei dati, sia in ambito clinico che di ricerca.

Si ha l'esigenza di memorizzare e organizzare i dati per consentire ai ricercatori, medici specializzandi o data manager di avere il controllo del carico e delle categorie di dati raccolti, agevolando l'interoperabilità degli utenti nelle attività di ricerca.

Al fine di garantire una raccolta dati omogenea e coerente, è necessario un processo di standardizzazione, che miri a migliorare l'affidabilità e la qualità dei dati, facilitando l'analisi e l'utilizzo delle informazioni a supporto dell'attività di ricerca.

Il progetto di tesi, svolto in collaborazione con l'azienda Tecnologie Avanzate, si è incentrato sulla progettazione e sviluppo di un database relazionale per la raccolta di dati clinici e di un'applicazione web destinata a più utenti per la gestione di progetti con scopi clinici differenti. Inoltre, ha perseguito tre principali obiettivi. Il primo ha riguardato la progettazione e costruzione di un database relazionale per l'immagazzinamento di dati destinati a progetti con scopi clinici differenti.

La costruzione del database è stata fondamentale per la progettazione e lo sviluppo dell'applicazione web interattiva, dedicata alla gestione dei progetti nelle attività di ricerca e alla gestione del database, in particolare per gli utenti che non hanno conoscenze sulle basi di dati.

Il secondo obiettivo si è focalizzato sull'implementazione di un approccio retrospettivo, per avere una panoramica dei dati clinici relativi a specifici progetti di ricerca e guidare gli utenti a comprendere i passaggi da intraprendere.

Il terzo ed ultimo riguarda un approccio prospettico, implementato nell'applicazione web per popolare il database con nuovi dati clinici.

Nelle fasi progettuali ci si è serviti di diagrammi UML per le fasi di definizione e modellizzazione delle specifiche per l'applicazione web e per la progettazione concettuale relativa al database.

---

Per quanto riguarda la progettazione logica e fisica del database, è stato scelto il DBMS MySQL.

La fase di testing ha dimostrato che i casi d'uso progettati e successivamente sviluppati hanno raggiunto i requisiti prefissati e funzionano correttamente.

# Indice

<b>Introduzione</b>	<b>7</b>
Il contesto . . . . .	7
Il progetto . . . . .	8
Obiettivo tesi . . . . .	9
Struttura tesi . . . . .	9
<b>1 Strumenti a supporto della ricerca clinica</b>	<b>11</b>
1.1 EDC ed eCRF: digitalizzazione della raccolta dati clinici . . . . .	11
1.2 Modelli di sviluppo del software . . . . .	13
1.3 I database . . . . .	15
1.3.1 Il diagramma Entità-Relazione . . . . .	16
1.3.2 Principali modelli di database . . . . .	17
1.3.3 Il Database Management System . . . . .	20
1.3.4 Il linguaggio del database: SQL . . . . .	21
<b>2 Progettazione e sviluppo del software</b>	<b>24</b>
2.1 Approccio di sviluppo, software e linguaggi utilizzati . . . . .	26
2.2 Progettazione applicazione web . . . . .	27
2.2.1 Definizione delle specifiche . . . . .	28
2.2.2 Modellizzazione delle specifiche . . . . .	33
2.3 Progettazione e costruzione del database . . . . .	46
2.3.1 Progettazione concettuale . . . . .	46
2.3.2 Progettazione logica e fisica . . . . .	50
2.4 Costruzione dell'applicazione web . . . . .	52
2.5 Testing dell'applicazione web . . . . .	57
<b>3 Applicazione pratica della piattaforma</b>	<b>64</b>
3.1 Progetto TCGA . . . . .	64
3.2 Form di caricamento dati . . . . .	66
3.3 Dashboard . . . . .	68

## INDICE

---

<b>Conclusioni e sviluppi futuri</b>	<b>76</b>
<b>Appendice A</b>	<b>79</b>
<b>Bibliografia</b>	<b>92</b>

# Introduzione

## Il contesto

Negli ultimi decenni, la digitalizzazione ha rivoluzionato profondamente il settore sanitario, incidendo in particolare nella raccolta e centralizzazione dei dati clinici e riducendo il rischio di errori legati alla gestione manuale [1].

La crescita esponenziale di cui ha giovato negli ultimi anni si evince dal grafico in Figura 1.

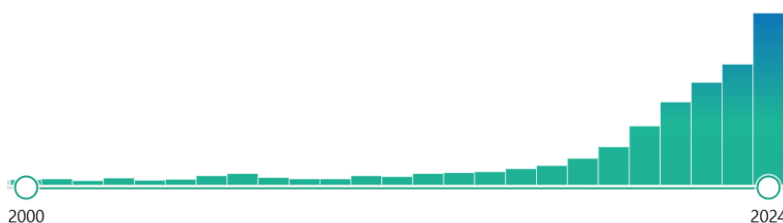


Figura 1: Distribuzione per anno dello sviluppo della digitalizzazione, raccolta e integrazione di dati clinici. Fonte: [2].

In riferimento al grafico in Figura 1, si può notare come la transizione digitale nella raccolta e gestione dei dati clinici è rimasta relativamente limitata nei primi anni 2000, registrando una crescita graduale.

Tuttavia circa dal 2015, si osserva un'accelerazione significativa, con un incremento ancora più marcato negli ultimi anni, segno della maggiore importanza attribuita alla digitalizzazione in ambito sanitario. Il picco raggiunto nel 2024 suggerisce che questa tendenza sia destinata a proseguire, evidenziando la necessità di strumenti sempre più avanzati per la raccolta, la gestione e l'analisi dei dati clinici.

Un problema persistente è relativo ai dati clinici dispersi o che vengono raccolti ed archiviati in diversi database. Con l'aumento della quantità di dati sanitari, anche il numero di database isolati è destinato a crescere, aggravando il problema

dell'inaccessibilità ai dati clinici, rendendo difficile una visione completa delle informazioni [3]. La dispersione dei dati causa una mancata interoperabilità e dati eterogenei rendono difficile il confronto tra loro, motivo per cui c'è l'esigenza di renderli omogenei e standardizzati [4].

In questo contesto e grazie all'evoluzione dell'informatica medica, il mio lavoro di tesi si inserisce come contributo alla digitalizzazione della raccolta dei dati clinici e all'integrazione di dati provenienti da diverse fonti. Attraverso la progettazione e lo sviluppo di un database relazionale, si punta alla loro centralizzazione.

Con la progettazione dell'applicazione web, si intende ottimizzare la gestione e l'importazione dei dati da immagazzinare, viene garantita l'accessibilità, l'integrità e la standardizzazione dei dati e viene agevolato il lavoro dei clinici nella ricerca, favorendo analisi più complete e accurate.

## Il progetto

Il progetto nasce da una collaborazione con l'azienda Tecnologie Avanzate per un'esigenza emersa da numerosi esperti del settore, con l'obiettivo di realizzare una soluzione innovativa che integri informazioni di diversa natura a supporto dei clinici e li possa agevolare nella conduzione di attività di ricerca.

Lo sviluppo del software da l'opportunità all'utente di gestire diversi progetti e per ognuno di essi centralizzare i dati nel database, consentire un approccio prospettico mediante la compilazione di un form per l'inserimento di dati e ottenere una panoramica delle informazioni relative al progetto.

Per quanto riguarda la compilazione del form, è stata richiesta la realizzazione di template personalizzati in base al progetto, così da garantire una raccolta accurata ed efficiente dei dati e consentire agli utenti di ottenere informazioni rilevanti per il progetto preso in considerazione. Nella sezione relativa alla visualizzazione generale dei dati, è emersa la volontà di poter settare dei filtri relativi agli specifici campi di interesse per agevolare il lavoro di ricerca per l'analisi e la visualizzazione del carico di dati.

Per l'aspetto relativo alla sicurezza, poiché le informazioni personali dei pazienti coinvolti nei progetti non erano rilevanti per l'attività di ricerca, i loro dati sono stati anonimizzati, evitando l'impiego di protocolli di sicurezza. Invece, le informazioni relative agli utenti che hanno accesso all'applicazione web sono state protette mediante tecniche di crittografia.



## Obiettivo tesi

Il progetto si propone di sviluppare un'applicazione web in grado di integrare e organizzare dati relativi a diversi progetti, con il fine di agevolare nella ricerca, servendosi della costruzione di un database.

Sulla base dei requisiti richiesti dall'azienda Tecnologie Avanzate, sono state progettate diverse funzionalità disponibili per gli utenti e alcune di queste sono state sviluppate:

- visualizzazione da parte dell'utente delle informazioni di un database già esistente, modifica della struttura o creazione di uno nuovo;
- integrazione di nuovi dati, mediante la compilazione di un form
- visualizzazione e modifica dei progetti di cui l'utente è proprietario;
- visualizzazione della dashboard per avere una panoramica dei dati clinici;

## Struttura tesi

Il lavoro di tesi è organizzato seguendo il naturale svolgimento delle attività svolte. La struttura che è stata seguita è la seguente:

### **Capitolo 1** : Strumenti a supporto della ricerca clinica

Questo capitolo fornisce una panoramica sugli strumenti utilizzati nella ricerca clinica, con particolare attenzione ai database e ai sistemi di Electronic Data Capture (eCRF). Vengono analizzate le caratteristiche principali di queste tecnologie, il loro ruolo nella gestione dei dati clinici ed in particolare l'importanza che hanno nell'obiettivo del lavoro di tesi.

### **Capitolo 2** : Progettazione e sviluppo dell'applicazione web

Si procede con la fase più pratica del lavoro, mostrando i vari passaggi relativi alla fase progettuale e di sviluppo per la costruzione del database e dell'applicazione web. In questo capitolo inoltre, sono descritti gli strumenti utilizzati.

### **Capitolo 3** : Applicazione pratica della piattaforma

Consiste nella descrizione dello sviluppo dei due principali approcci richiesti come funzionalità accessibili agli utenti.



# Capitolo 1

## Strumenti a supporto della ricerca clinica

### 1.1 EDC ed eCRF: digitalizzazione della raccolta dati clinici

Negli ultimi anni, i sistemi di Electronic Data Capture (EDC) sono stati ampiamente utilizzati per migliorare la gestione dei dati nei trial clinici decentralizzati, ottimizzandone la raccolta, l'archiviazione e l'analisi [4]. Tuttavia, in questo progetto di tesi, l'EDC non viene impiegato tradizionalmente per trial clinici, ma per la gestione dei dati a supporto della ricerca clinica in generale.

Gli EDC sono software complessi, progettati per migliorare la qualità dei dati raccolti rispetto alla raccolta manuale, per proteggere la loro integrità, rendendo la ricerca sempre più affidabile.

Ci sono diverse tipologie di EDC, tra i più utilizzati ci sono gli EDC basati su cloud, che permettono l'archiviazione e l'accesso ai dati attraverso infrastrutture cloud, offrendo scalabilità, flessibilità e accessibilità da qualsiasi luogo con connessione a Internet. Questa tendenza all'adozione del cloud sta trasformando il modo in cui i dati vengono raccolti e gestiti, rendendo i sistemi EDC più efficienti ed economici [5].

Gli EDC offrono agli utenti diverse funzionalità, tra cui:

- la possibilità di immettere, salvare, archiviare e recuperare informazioni in modo sicuro da varie fonti;
- la possibilità di analisi, revisione e monitoraggio dei dati relativi allo studio, consentendo di valutare se adottare approcci diversi per la specifica ricerca [4];

- consentono lo scambio sicuro ed efficiente di dati tra diversi sistemi compatibili tra loro, con la possibilità di poterli interpretare correttamente.

Inoltre, gli EDC devono essere intuitivi, modificabili, scalabili e flessibili.

I tradizionali EDC presentano anche delle limitazioni:

- potrebbero non consentire eventuali modifiche al database, complicando il processo;
- in alcuni casi non sono integrati con altri sistemi di raccolta dati per il trasferimento automatizzato di dati da altre fonti e richiedono l'inserimento manuale da parte dell'utente [4].

A tal proposito, nel caso in cui l'utente avesse la necessità di aggiungere nuovi dati, si ricorre a una soluzione che facilita l'inserimento: l'eCRF.

L'eCRF o electronic Case Report Form è un sistema integrato nel software EDC per l'inserimento automatico dei dati all'interno del database, consentendo all'utente di compilare un form personalizzato [4].

Inizialmente, per verificare la qualità dei dati nei trial clinici inseriti, per i quali gli EDC vengono principalmente impiegati, si confrontavano i moduli cartacei dei report clinici, anche detti CRF, con i dati inseriti nel database, passaggi che rallentavano il processo.

Inoltre, questo metodo spesso non teneva conto degli errori che potevano emergere durante il processo di estrazione e trascrizione delle informazioni a causa dell'intervento umano, in cui interpretazioni errate, omissioni o imprecisioni nella trascrizione all'interno del database potevano compromettere la vericità delle informazioni [6].

La trascrizione manuale dei dati richiedeva inoltre un dispendio in termini temporali, dovuto alla fase di pulizia e preparazione dei dati [7].

Grazie all'avvento dell'eCRF, con la compilazione del form si susseguono diverse azioni:

- raccolta immediata dei dati all'interno del database consentendo una centralizzazione all'interno dell'EDC;
- standardizzazione dei dati, che facilita l'accesso ai dati e lo scambio tra utenti;
- pre-processing dei dati, convalida automatizzata e controlli di qualità, assicurano che i dati siano privi di errori, migliorando l'accuratezza e l'affidabilità delle informazioni [7].

## 1.2 Modelli di sviluppo del software

L'Ingegneria del Software è emersa come disciplina scientifica indipendente negli anni '70, con l'obiettivo di creare modelli di sviluppo software per identificare l'approccio migliore da utilizzare durante la realizzazione [8].

Tra le varie definizioni di Ingegneria del Software, emerge la necessità di costruire software che siano economici, affidabili ed efficienti su sistemi reali [9].

È utile definire i principali modelli del ciclo di vita del software, accumulate dalle stesse macrofasi e che si distinguono per l'approccio utilizzato.

### 1. Modello a cascata:

il primo modello sviluppato e caratterizzato da una sequenza lineare di fasi, mostrate in Figura 1.1.

Attribuisce una grande importanza alla raccolta dei requisiti, trattandola come una fase fondamentale che deve essere completata in modo esaustivo prima di procedere con le fasi successive di progettazione totale e successivo sviluppo [10].

Un aspetto critico di questo modello è la rigidità dei cambiamenti nei requisiti, poichè il modello non prevede il ritorno a fasi precedenti. Questo implica un aumento di costi, compromettendo il flusso del progetto [8].

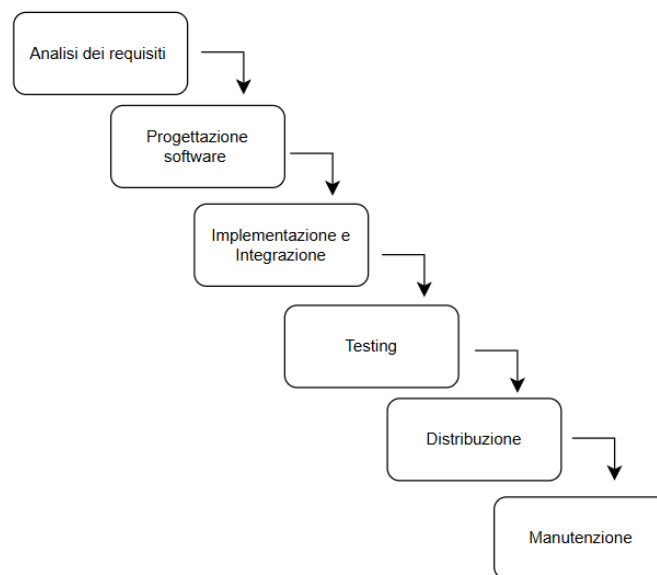


Figura 1.1: Modello di sviluppo del software: modello a cascata. Fonte: [8]

## 2. Modello a spirale:

è stato sviluppato per ovviare alle criticità riscontrate con il modello a cascata.

È basato su rippezioni cicliche che iterano finchè non si raggiunge l'obiettivo, determinato nella prima fase, come si evince in Figura 1.2.

L'approccio iterativo, pur richiedendo tempi più lunghi, consente un controllo periodico dei rischi, dando la possibilità di modificare la strategia, di svilupparla e pianificare l'obiettivo successivo [11].

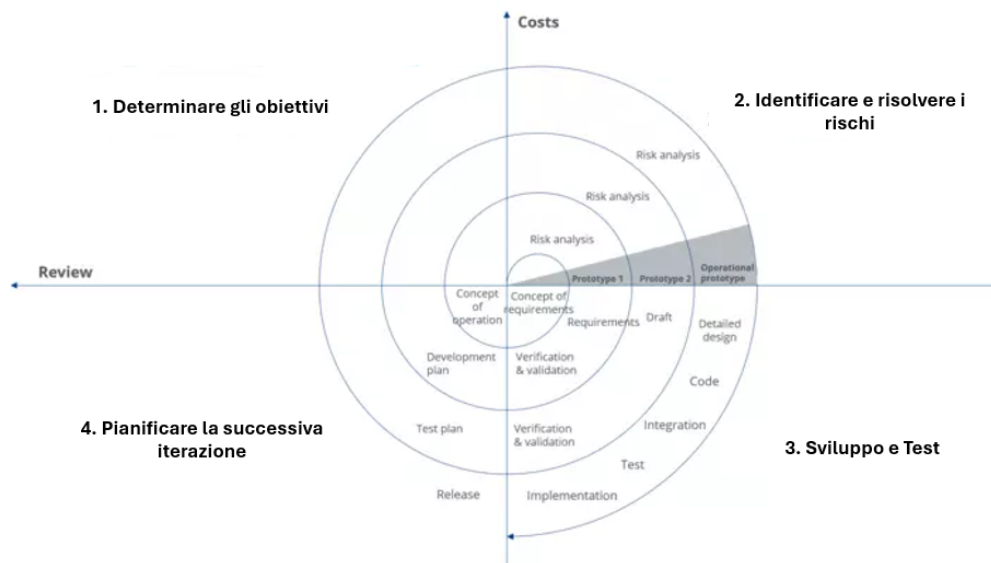


Figura 1.2: Modello di sviluppo del software: modello a spirale. Fonte: [8]

## 3. V-model:

il modello deve il nome alla struttura a forma di V, estensione del tradizionale modello a cascata. In Figura 1.3 è mostrata la stretta correlazione tra ogni fase di progettazione, a sinistra del modello, di sviluppo alla base e le corrispondenti attività di verifica e validazione, a destra.

L'approccio utilizzato è utile al fine di evitare errori, in particolare i test dei moduli dovrebbero essere eseguiti il più possibile parallelamente alle fasi di sviluppo [12].

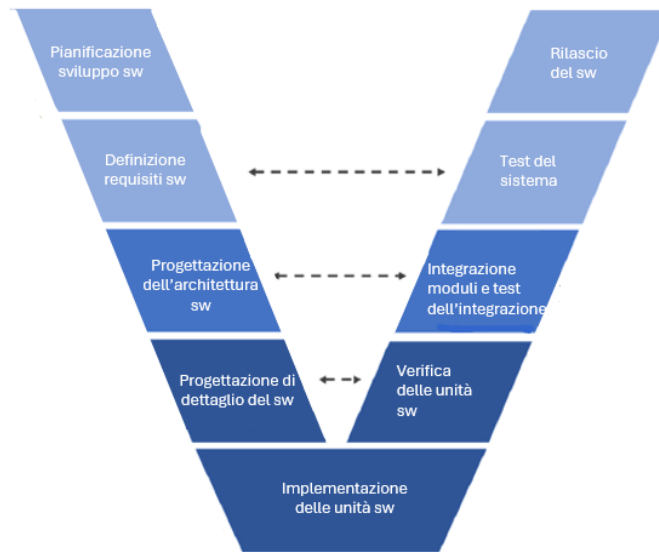


Figura 1.3: Modello di sviluppo del software: il V-model. Fonte: [13]

## 1.3 I database

Il database rappresenta un archivio strutturato di informazioni con lo scopo di garantire un'efficiente archiviazione, aggiornamento ed estrazione dei dati mediante un generatore di report [14].

È organizzato in una collezione di file, a sua volta il file è suddiviso in record logici, ognuno dei quali rappresenta un insieme di dati [14]. Questa struttura facilita l'accesso alle informazioni archiviate.

Il database, rispetto ai file, offre diversi vantaggi come metodo di archiviazione dei dati [14]:

- Astrazione dei dati: un modello di dati offre una rappresentazione astratta e concettuale dei dati e consente di nascondere i dettagli relativi alla memorizzazione e di offrire agli utenti una visione concettuale del database;
- supporto per viste multiple dei dati: ogni utente può accedere a una vista personalizzata del database, contenente esclusivamente le informazioni rilevanti per le sue esigenze;
- condivisione dei dati ed elaborazione delle transazioni multiutente: consente la gestione dei dati a più utenti contemporaneamente e in modo indipendente dalle applicazioni;

- gestione di grandi quantità di dati in maniera sicura e scalabile: sono garantiti controlli di accesso, crittografia e autenticazione, che garantiscono la protezione dei dati. Inoltre, i database possono essere facilmente scalati per supportare un aumento del volume di dati, assicurando buone prestazioni anche con un gran numero di utenti che operano simultaneamente [15].

Per poter sviluppare un database, ci sono diverse fasi di progettazione da seguire:

1. Progettazione concettuale: consiste nella rappresentazione astratta e di alto livello dei dati, senza considerare ancora la parte concreta. Descrive i dati e le loro relazioni, fornendo una panoramica chiara della struttura [16]. Questa fase di progettazione è descritta mediante il Diagramma Entità-Relazione, che verrà approfondito nel paragrafo 1.3.1.
2. Progettazione logica: i dati vengono organizzati secondo un modello formale. Questa fase rimane indipendente dalle scelte tecnologiche e si inizia la fase di normalizzazione e ottimizzazione dei dati per garantire efficienza e ridurre ridondanze ed errori. Il modello logico scelto al termine di questa fase di progettazione, è realizzato a seconda del tipo di database [17] e i vari modelli sono analizzati nel paragrafo 1.3.2.
3. Progettazione fisica: definisce come e dove vengono memorizzati i dati [16].

È fondamentale tenere traccia delle decisioni e delle motivazioni adottate in ogni fase della progettazione, per facilitare eventuali modifiche future.

### 1.3.1 Il diagramma Entità-Relazione

La progettazione concettuale consiste nella creazione di un diagramma che mostra la correlazione tra le entità e le loro relazioni, detto Diagramma Entità-Relazione o E-R.

I principali componenti del Diagramma E-R sono mostrati in Figura 1.4:

- entità: è qualsiasi oggetto di cui il sistema deve memorizzare informazioni, rappresentata da un rettangolo;
- relazione: descrive il modo in cui le entità interagiscono tra loro, rappresentata da un rombo. Ogni relazione può avere una cardinalità, che indica il numero di entità coinvolte e possono essere di tre tipi:
  - Relazione 1:1 (uno a uno)
  - Relazione 1:N (uno a molti)
  - Relazione N:N (molti a molti)



- attributi: descrivono le proprietà relative ad una specifica entità, rappresentati da un ellisse al quale corrisponde uno specifico valore.
- chiave: è uno specifico attributo utilizzato per identificare in modo univoco un'istanza dell'entità. Una chiave è considerata primaria se ha un valore unico e non nullo per ogni istanza. La chiave garantisce l'integrità e l'univocità dei dati, evitando duplicazioni [14].

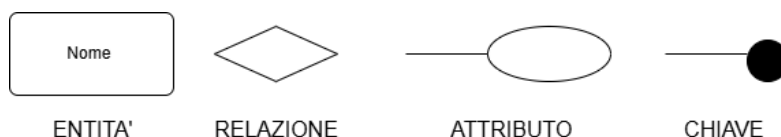


Figura 1.4: Componenti principali del Diagramma E-R. Fonte: [14]

Per comprendere meglio l'applicazione pratica del Diagramma E-R, in Figura 1.5 è rappresentata una parte del diagramma relativo al dataset del progetto di tesi e che mostra due entità principali, le loro relazioni e gli attributi chiave.

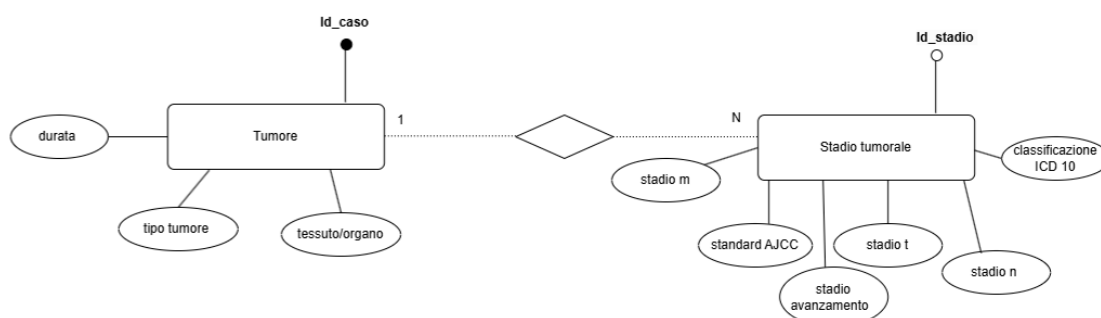


Figura 1.5: Esempio di Diagramma E-R rappresentata nella fase di progettazione concettuale.

### 1.3.2 Principali modelli di database

Nel corso degli anni, sono stati sviluppati vari modelli di database, ciascuno dei quali apportava miglioramenti rispetto al precedente. Ogni modello è stato progettato per soddisfare esigenze specifiche nell'organizzazione dei dati.

Tra i principali sistemi che si sono evoluti dal 1960 ai primi anni '90, spiccano tre modelli fondamentali.

1. **Modello gerarchico:** sviluppato agli inizi degli anni '70, si basa su una struttura ad albero in cui il punto di partenza, chiamato root o radice, rappresenta il nodo padre e non ha predecessori ma può avere più figli. Tutti

gli altri nodi o record hanno un unico predecessore e possono a loro volta generare più figli [18]. Il modello, mostrato in Figura 1.6, pur garantendo una chiara organizzazione dei dati, presenta limiti dovuti alla sua rigidità, che impedisce di ridurre la ridondanza delle informazioni [19].

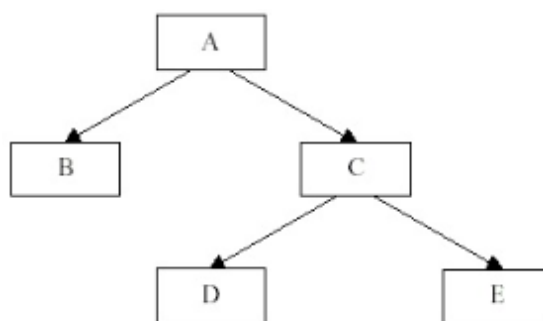


Figura 1.6: Struttura del modello gerarchico di un database. Fonte: [20].

2. **Modello reticolare:** sviluppato successivamente al gerarchico per ovviare al limite della ridondanza, si basa sull'uso di grafi ed è caratterizzato da record, registrazione logica composta da più campi e dal set che definisce le relazioni tra i record all'interno del database. Questo modello a differenza del gerarchico ha il vantaggio di formare una struttura a rete che consente ad ogni record di avere più predecessori e di poter accedere ai dati da più percorsi [18]. La struttura è mostrata in Figura 1.7

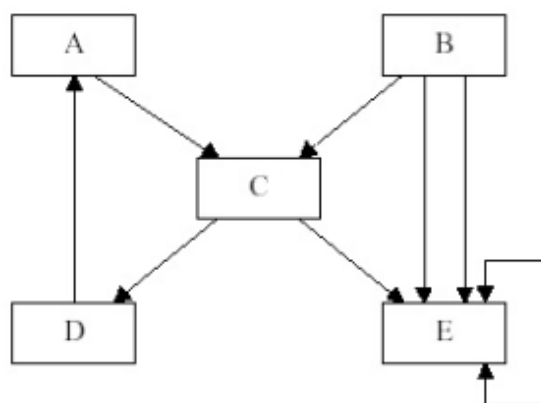


Figura 1.7: Struttura del modello reticolare di un database. Fonte: [20].

3. **Modello relazionale:** definito agli inizi degli anni '70 e affermatosi negli anni '80, attualmente è il modello più diffuso[19].

Basato su tabelle in cui le righe rappresentano i record, mentre le colonne corrispondono agli attributi. Le tabelle sono delle entità legate ad altre entità da relazioni che permettono di organizzare e recuperare i dati in modo efficiente senza ridondanza, grazie all'uso di chiavi primarie e secondarie[18]. La rappresentazione di questo modello è mostrata nel diagramma E-R, in particolare rappresentato nella fase di progettazione concettuale in Figura 1.5. Invece, la struttura del modello relazionale in maniera più ampia è mostrata mediante l'Enhanced Entity-Relationship diagram o Diagramma EER, mostrato in Figura 1.8 e approfondito nel Capitolo 2.

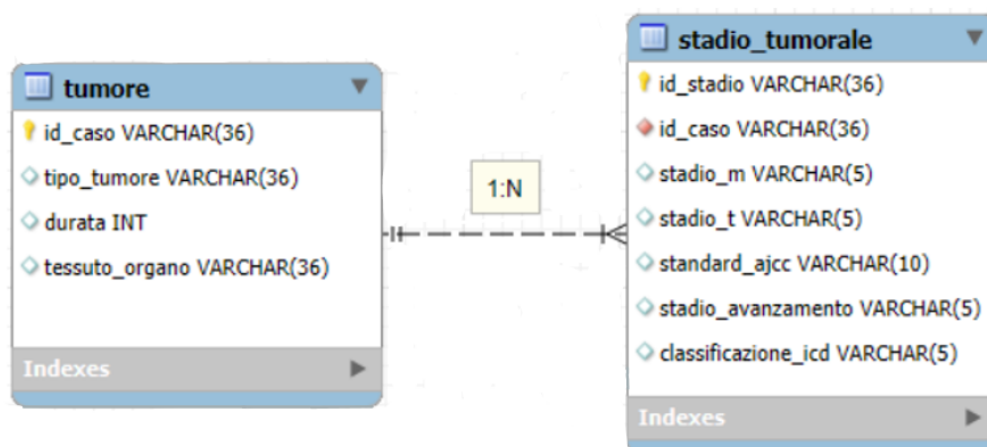


Figura 1.8: Struttura del modello relazionale di un database rappresentata attraverso il Diagramma EER.

Inoltre, esistono altre tipologie di modelli meno utilizzati, tra questi:

4. **Modello a oggetti:** è stato sviluppato come evoluzione del modello relazionale alla fine degli anni '80 principalmente per contenuti multimediali [16]. Il modello è strutturato in classi, identificate in maniera univoca da un identificatore numerico e caratterizzate da istanze o oggetti. Un aspetto distintivo di questo modello è che ogni oggetto contiene non solo gli attributi, ma anche i metodi ovvero le operazioni che si possono eseguire sugli attributi di quell'oggetto.

Sebbene sia particolarmente adatto alla gestione di problemi complessi, presenta una scarsa compatibilità con applicazioni e una bassa efficienza nelle query semplici, rendendolo quindi meno adatto ad usi quotidiani [18].

### 1.3.3 Il Database Management System

Il Database Management System o DBMS è un sistema software progettato per gestire grandi collezioni di dati, facendo da tramite tra database, applicazione e utenti. Ha diverse funzioni, tra cui [14]:

- gestisce la sicurezza e il controllo degli accessi attraverso diversi livelli detti privilegi, determinando i livelli di autorizzazione degli utenti, in collaborazione con il Database Administrator (DBA) che decide se concedere o revocare i privilegi per garantire la protezione del database;
- gestisce l'accesso simultaneo ai dati da parte di più utenti contemporaneamente, consentendo modifiche simultanee agli stessi dati, preservando l'integrità dei dati attraverso meccanismi di controllo della concorrenza e di eliminazione di inconsistenza per eventuali duplicati [21];
- garantisce l'indipendenza dei dati dalle applicazioni, senza svelare dettagli relativi all'archiviazione e al recupero e grazie all'indipendenza dei dati viene garantita la flessibilità, in quanto le modifiche al database o all'applicazione non generano conseguenze;
- gestisce il backup e il rapido recupero dei dati anche in caso di perdita a causa di guasti imprevisti [14].

Tra i principali DBMS open source più diffusi c'è MySQL, MariaDB e PostgreSQL, invece tra quelli commerciali c'è Oracle, Microsoft SQL Server e IBM DB2 [22].

Per comprendere come il DBMS gestisce e organizza i dati all'interno del database, è utile descrivere l'architettura a tre livelli di astrazione proposta dal modello ANSI/SPARC <sup>1</sup>, mediante lo schema mostrato in Figura 1.9:

- Livello dell'Utente o Esterno: rappresenta una visione personalizzata del sistema in base alle esigenze dei singoli utenti. Ha la capacità di non essere influenzato dai cambiamenti apportati allo schema logico;
- Livello Logico o Concettuale: rappresenta un'unica visione centralizzata del database. Questo livello è cruciale per una gestione efficiente delle informazioni, prendendo in considerazione le esigenze degli utenti, senza che gli utenti debbano preoccuparsi dei dettagli fisici e implementativi.

---

<sup>1</sup>L'acronimo ANSI-SPARC sta per American National Standards Institute - Standards Planning and Requirements Committee. Si riferisce alla proposta di modello di architettura a 3 livelli fatta nel 1975 da parte del comitato SPARC dell'ANSI.

- Livello di Sistema o Fisico: rappresenta una descrizione dei dettagli implementativi legati alla memorizzazione fisica dei dati [23].

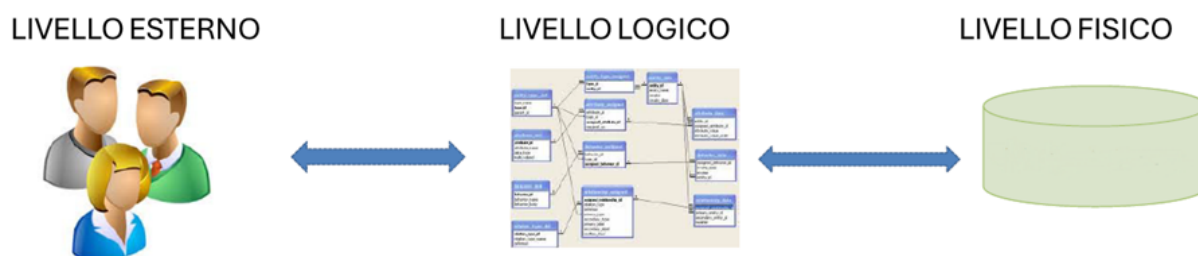


Figura 1.9: Architettura a 3 livelli. Fonte:[14].

### 1.3.4 Il linguaggio del database: SQL

In riferimento ai modelli di database descritti nel Paragrafo 1.3.2, nel progetto di tesi è stato adottato un modello di database relazionale.

Di conseguenza, verrà approfondito il linguaggio specifico di questo modello.

Lo Structured Query Language o SQL è un linguaggio di programmazione, sviluppato da IBM negli anni '70 e successivamente standardizzato dall'ANSI - American National Standards Institute e dall'ISO - International Organization for Standardization [15].

Il linguaggio SQL consente di interagire con i database attraverso comandi specifici, suddivisi in sottolinguaggi in base alla loro funzione.

Di seguito sono elencati i comandi più utilizzati delle diverse categorie [24]:

**DDL o Data Definition Language** utilizzato per definire la struttura del database, comprende comandi che consentono [24]:

- la selezione di uno specifico database con il comando *USE*;
- la creazione di un database con il comando *CREATE DATABASE* e la creazione di una specifica tabella con *CREATE TABLE*, specificando le colonne e la tipologia di dati che ci si aspetta di avere come formato o come *PRIMARY KEY* o *FOREIGN KEY*;
- la modifica di una specifica tabella con il comando *ALTER TABLE* o di una specifica colonna *ALTER COLUMN*;
- l'eliminazione di un database con il comando *DROP DATABASE*, l'eliminazione delle tabelle che costituiscono il database con *DROP TABLE* o delle colonne con *DROP COLUMN* per specifiche condizioni e l'eliminazione delle voci di una tabella mantenendone la struttura con il comando *TRUNCATE TABLE*;

**DML o Data Manipulation Language** utilizzato la gestione dei dati all'interno delle tabelle del database, comprende comandi che consentono:

- l'inserimento di un dato in una specifica tabella con il comando *INSERT INTO*;
- la modifica dei campi di uno o più record con il comando *UPDATE*;
- l'eliminazione dei record da una tabella con il comando *DELETE FROM*

**DQL o Data Query Language** utilizzato per interrogare il database e recuperare specifici dati, comprende un comando *SELECT*.

**DCL o Data Control Language** utilizzato per controllare gli accessi a diversi database e tabelle in maniera indipendente, comprende comandi che consentono:

- l'assegnazione dei permessi agli utenti con il comando *GRANT*
- la revoca dei permessi agli utenti con il comando *REVOKE*

SQL permette di applicare condizioni, come *WHERE* o *IF EXISTS* per controllare se un determinato elemento esiste prima di eseguire un'operazione, evitando errori quando si tenta di modificare dati inesistenti.



## Capitolo 2

# Progettazione e sviluppo del software

La realizzazione del software comprende l'applicazione web e il database. In particolare, è prevista la progettazione dell'architettura, mostrata come fase preliminare alla progettazione vera e propria del software in Figura 1.3 e comprende:

- il frontend dell'applicazione, ovvero la parte accessibile all'utente mediante le interfacce utente e il form con il quale l'utente interagisce;
- il backend dell'applicazione, ovvero la logica applicativa che si occupa della gestione interna nell'interazione con il database;
- il database, per l'organizzazione e la gestione dei dati in modo che possano essere manipolati dal backend.

L'applicazione web è progettata per ospitare progetti di diversi utenti.

In particolare, per il lavoro di tesi è stato utilizzato un dataset relativo al progetto TCGA della NCI<sup>1</sup> e nelle prime fasi è stata fondamentale la raccolta, l'analisi e la preparazione dei dati.

La fase di analisi dei dati ha lo scopo di trovare le relazioni che li legano e capire che nonostante i dati siano di tipo clinico, ogni progetto possiede dati specifici.

A tal proposito, si è scelto di immagazzinare i dati in un database unico che possa ospitare tutti i progetti, ma che abbia elementi caratterizzanti degli specifici progetti, per poter creare template personalizzati dell'applicazione.

Lo sviluppo di un software è un processo complesso che trasforma i requisiti iniziali in un prodotto finito, potenzialmente pronto per la commercializzazione.

---

<sup>1</sup>L'acronimo NCI sta per National Cancer Institute. È uno dei più antichi istituti della National Institutes of Health o NIH che sostiene la ricerca del cancro.



Il processo comprende sei fasi principali, descritte nel flowchart in Figura 2.1.

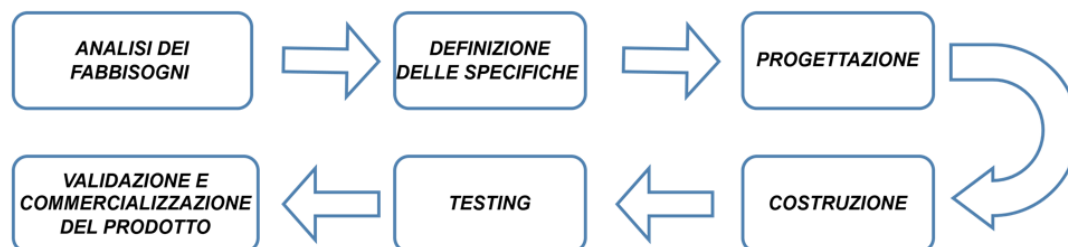


Figura 2.1: Flowchart delle fasi principali. Fonte: [25]

L' *Analisi dei Fabbisogni* è una fase che è stata principalmente affrontata dall'azienda Tecnologie Avanzate. Prevede la definizione dei requisiti del sistema, mediante la raccolta di interviste e osservazioni per comprendere le necessità degli utenti a cui è destinata l'applicazione web, al fine di stabilire i processi da sviluppare, modellarli per poi validarli mediante un report.

Successivamente, è necessaria la fase di *Definizione delle Specifiche*, in cui si modellizzano i processi che si vogliono avere nel sistema e le funzionalità nell'interazione tra gli utenti e il sistema.

La fase di *Progettazione del Software* comprende sia la modellizzazione delle specifiche relativa all'applicazione, sia la progettazione concettuale relativa al database. In particolare, per la definizione e modellizzazione delle specifiche è stato utilizzato l'Unified Modeling Language o UML, utile per monitorare lo sviluppo e per documentare le attività di progettazione [25].

Successivamente si passa allo *Sviluppo del Software*, sia del database che dell'applicazione e alla fase di *Testing*, in cui si preparano delle prove che verificano il corretto funzionamento del sistema.

Il testing massimizza la capacità di identificare anomalie nel software costruito, consente la tracciabilità completa delle attività svolte e permette di ripercorrere ogni fase del processo.

Infine, c'è la fase di *Validazione e Commercializzazione del Software*, ma il lavoro di tesi non ha portato a un prodotto pronto per la commercializzazione.

Entrando nel cuore del progetto, sono state progettate tutte le funzionalità dell'applicazione relative alla gestione del database e dei progetti relativi agli utenti. Tuttavia, la fase di sviluppo si è focalizzata principalmente sulla gestione dei progetti.

Sono stati realizzati la dashboard e le funzionalità relative ad uno specifico progetto, inclusa la possibilità di scaricare un file riassuntivo.

Questo capitolo ha lo scopo di mostrare tutte le fasi progettuali e di sviluppo del database e dell'applicazione web, includendo il testing relativo ad uno dei principali casi d'uso sviluppati.

## 2.1 Approccio di sviluppo, software e linguaggi utilizzati

Il modello che è stato seguito per il processo di sviluppo software è il V-model, mostrato in Figura 1.3.

Nel primo step di pianificazione, è importante definire le motivazioni che hanno portato alla scelta del modello. È stato ritenuto più adatto allo scopo, in quanto:

- i requisiti erano chiari, perchè definiti fin dall'inizio dall'azienda;
- ha una struttura abbastanza rigida che richiede una particolare precisione ai dettagli, per rendere l'applicazione il più affidabile possibile, riducendo errori soprattutto nella progettazione del database.

Successivamente, sulla base delle esigenze degli utenti e degli obiettivi da perseguire, sono stati definiti i processi principali utili allo sviluppo:

- creazione database;
- creazione progetto;
- visualizzazione dashboard

i cui risultati e output attesi sono descritti successivamente nella fase di definizione delle specifiche, relativa al flowchart in Figura 2.1.

Tra i requisiti ad alto livello del sistema, si prevede che il sistema debba gestire progetti di ricerca, raccogliendo dati e visualizzando una panoramica delle informazioni principali e debba gestire anche i database che agevolino l'immagazzinamento dei dati.

Tra i requisiti tecnici del sistema, per la costruzione dell'applicazione web è stato utilizzato Streamlit, un framework open-source in linguaggio Python, ideale per la creazione e lo sviluppo di applicazioni web personalizzate, per visualizzare dati mediante grafici interattivi [26].

Si è scelto un database relazionale, gestito dal DBMS MySQL per la necessità di adottare un database facile da progettare e costruire, per garantire una struttura scalabile e un sistema che necessita di integrità e affidabilità per la gestione di dati clinici.

Per le fasi progettuali del database e dell'applicazione web, sono state utilizzate due software:

1. *Diagrams.net* per la progettazione concettuale del database e per la costruzione dei diagrammi UML relativi alle fasi di definizione e modellizzazione delle specifiche per l'applicazione web;

2. *MySQL Workbench 8.0 CE*, per la progettazione logica e fisica del database. In particolare, per la progettazione logica è stato costruito il diagramma EER mediante un tool integrato del software e da quest'ultimo, attraverso la funzionalità 'Forward Engineer to Database' è stato possibile ottenere il database fisico.

Nella fase di definizione dei requisiti software, sono stati scelti i requisiti relativi alle interfacce utente. Devono risultare semplici e intuitive, graficamente accattivanti, fruibili, navigabili e flessibili per poter avere accesso alle interfacce da diversi percorsi [25].

L'applicazione web è destinata a utenti come il Coordinatore di Ricerca Clinica (CRC), Data Manager, medici specializzandi o ricercatori clinici.

Le fasi successive di progettazione, sviluppo e testing sono approfonditi nei successivi paragrafi.

## 2.2 Progettazione applicazione web

Il progetto di tesi consiste nella progettazione e nello sviluppo del software, fase cruciale per soddisfare le esigenze degli utenti a cui l'applicazione è destinata. Nello schema in Figura 2.2 è descritto il flusso del lavoro di tesi, e descrive cosa riceve in input l'EDC e cosa restituisce in output:

- in input, il Data Entry consente di importare manualmente i dati nel database, mentre il Data Import permette di caricare dati preesistenti in uno specifico formato direttamente nel database;
- l'EDC che elabora i dati e integra l'eCRF per l'inserimento dei dati in maniera automatizzata, mediante un form;
- in output, un file riassuntivo scaricabile e una dashboard per ottenere una panoramica personalizzata dei dati mediante il settaggio di specifici filtri.

Rispetto ai tradizionali EDC, il prototipo costruito è accessibile e adatto ai cambiamenti da parte degli utenti.

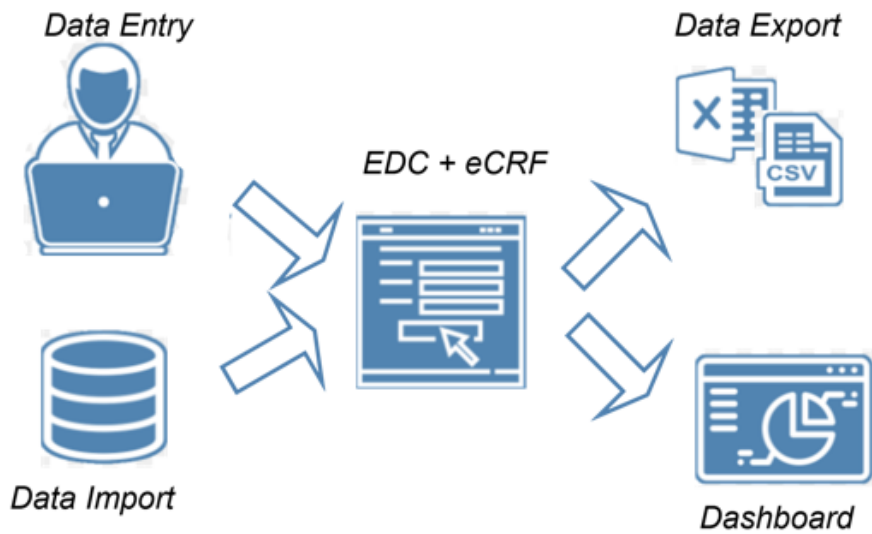


Figura 2.2: Flusso del lavoro di tesi

Non essendo un software medicale, l'applicazione ha lo scopo di agevolare le attività di ricerca. È stata pensata principalmente per dare accesso ai progetti di cui sono proprietari gli utenti, svolgendo diverse azioni:

- visualizzazione delle informazioni relative al progetto ed esportazione di un file riassuntivo;
- modifica ed eliminazione di un progetto;
- visualizzazione di una dashboard, settando dei filtri per avere una panoramica delle informazioni principali.

Inoltre, l'applicazione è pensata per agevolare la gestione dei database, rendendola accessibile a chiunque, in particolare agli utenti che non hanno familiarità con il linguaggio SQL. Tra le varie funzionalità che offre, consente:

- la modifica della struttura di un database già esistente e l'eventuale eliminazione;
- la creazione di un nuovo database, dando la possibilità di definire la struttura dall'applicazione web;
- l'inserimento dei dati clinici nel database, mediante compilazione di un'eCRF.

### 2.2.1 Definizione delle specifiche

La fase introduttiva alla vera e propria progettazione dell'applicazione web consiste nell'individuazione dei processi, per seguire il flusso delle attività svolte dagli attori.

Avendo già costruito un database generico che possa ospitare diversi progetti di ricerca ma che contiene già un progetto specifico relativo alla gestione di dati radioterapici, i processi individuati hanno lo scopo di:

- creare un nuovo database;
- gestire un nuovo progetto, visualizzando una panoramica dei dati relativi al particolare scopo clinico.

Il primo scopo è stato solamente progettato, il secondo invece partendo da un progetto preesistente è stato anche sviluppato nell'applicazione web. In questa fase sono stati realizzati *Synopsis Diagram* e *Workflow Diagram* per ogni processo.

Il Synopsis Diagram è stato scelto per mettere in evidenza in maniera immediata il contesto dei processi, in particolare l'evento trigger che fa partire il processo, i dati in input presenti, cosa ci si aspetta in output, il risultato e gli attori coinvolti. Il Workflow Diagram invece è stato preferito allo Swimlane Diagram, in quanto ogni processo non necessitava di un flusso di attività troppo dettagliato, ma il Workflow Diagram ha avuto l'obiettivo di dare una visione delle macroattività.

Di seguito sono mostrati i diagrammi relativi ai tre processi individuati.

### **Creazione database**

Il processo di *Creazione database* ha inizio nel momento in cui un utente decide di creare un database senza aggiornare o modificare la struttura di un database preesistente, se per esempio dovesse creare un progetto in un ambito di ricerca completamente diverso. Il processo ha come evento trigger il caricamento di un file per avere una struttura di default del database, o in alternativa la struttura può essere definita arbitrariamente.

La definizione arbitraria consiste nella compilazione di schede che hanno lo scopo di guidare l'utente nella scelta della struttura specifica del database, simulando un DBMS per coloro che non conoscono il linguaggio SQL.

L'inserimento dei dati mediante eCRF per popolare il database è un passo importante per raggiungere la completezza del processo ma come si vedrà nella fase di modellizzazione delle specifiche, l'importazione di un file o la compilazione di un form per popolare il database è un caso d'uso a parte.

Per accedere alle funzionalità dell'applicazione web, si forniscono in input le credenziali di accesso dell'utente.

Questo passaggio è fondamentale per mantenere la sicurezza del database e garantire che solo chi è autorizzato possa eseguire il processo.

In Figura 2.3 e 2.4 sono riportati rispettivamente il synopsis diagram e il workflow diagram relativi al processo *Creazione database*.

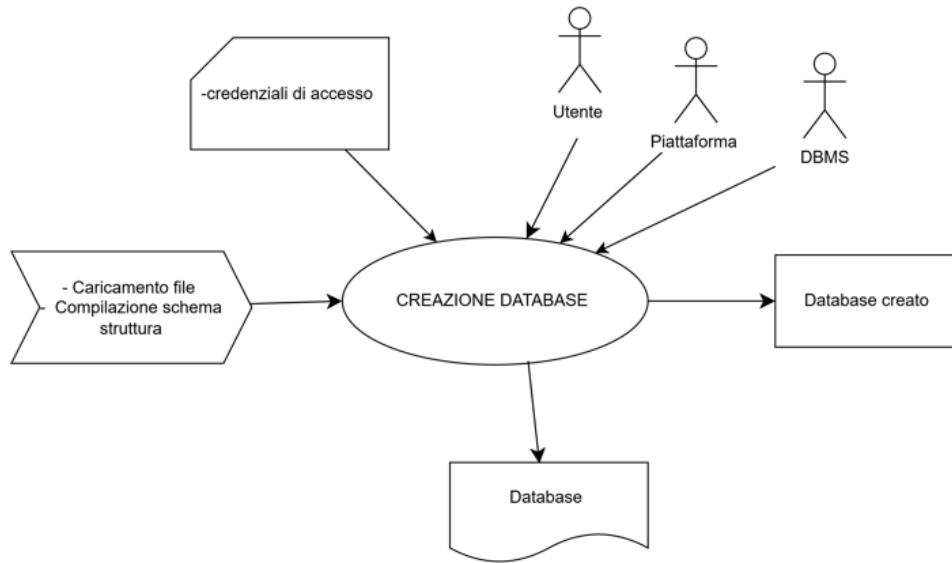


Figura 2.3: Synopsis Diagram relativo al processo 'Creazione database'

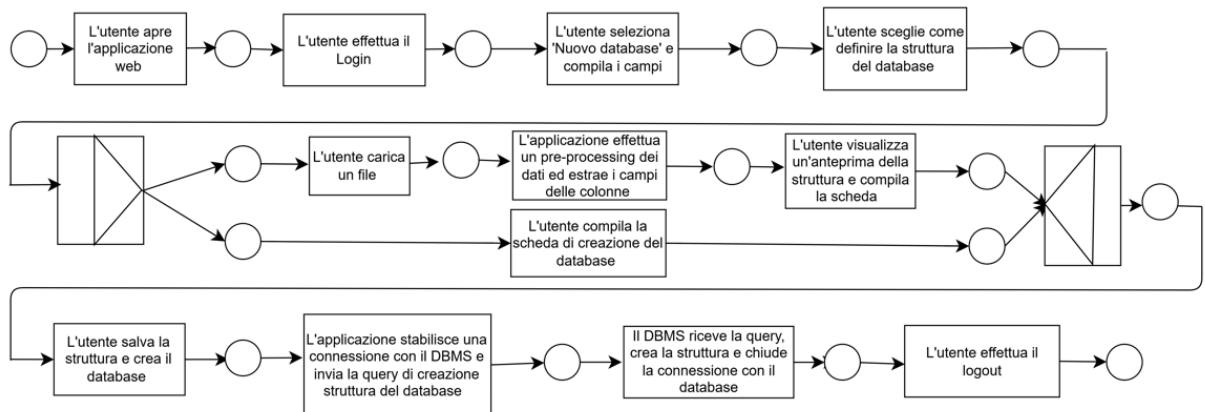


Figura 2.4: Workflow Diagram relativo al processo 'Creazione database'

Il workflow diagram, descrive più nel dettaglio le attività in cui viene mostrata l'interazione tra l'applicazione web e il DBMS. Viene messo in evidenza anche il pre-processing effettuato sui dati per creare la struttura del database partendo da un file.

## Creazione progetto

Il processo di *Creazione progetto* descrive un passo fondamentale nel ciclo di vita di un'applicazione.

Ha lo scopo di creare progetti con scopi clinici differenti, in quanto agevola l'integrazione dei dati in modo strutturato all'interno del database.

Il processo segue il flusso descritto nel workflow diagram in Figura 2.6, descrivendo le interazioni tra l'applicazione e il database. In input al processo, nel synopsis diagram in Figura 2.5 viene fornito un file riassuntivo del progetto, che servirà per l'importazione e la raccolta dei dati all'interno del database.

L'uso del file riassuntivo è cruciale per ottimizzare e automatizzare il processo, serve a centralizzare tutte le informazioni in un formato standardizzato.

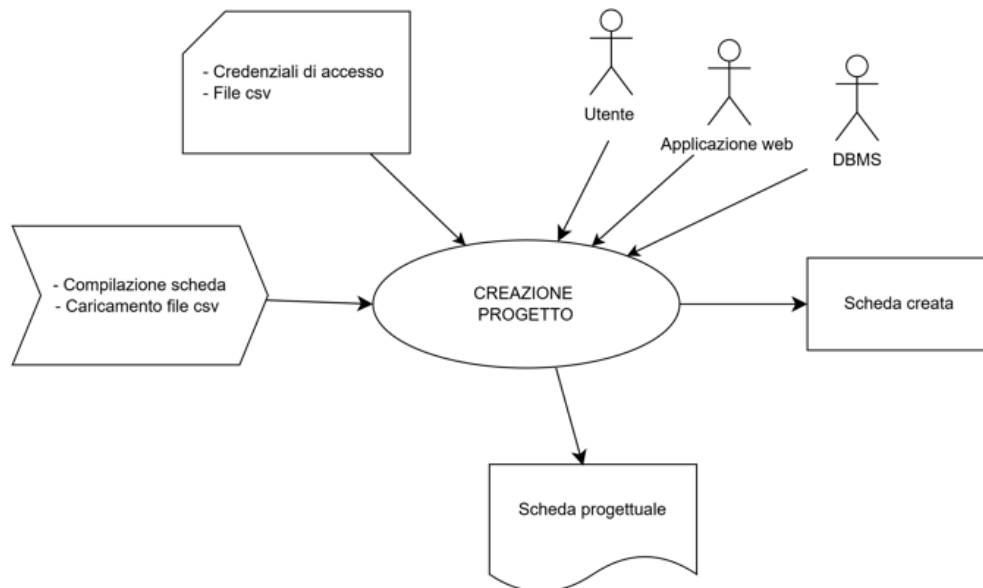


Figura 2.5: Synopsis Diagram relativo al processo 'Creazione progetto'

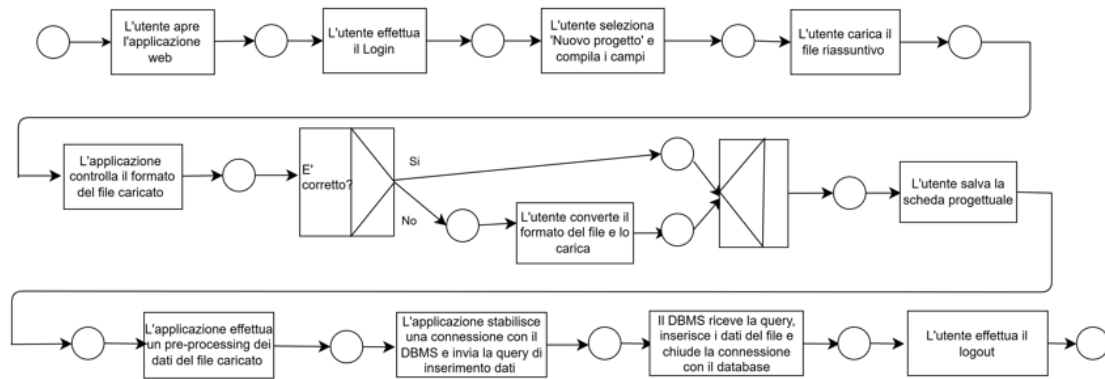


Figura 2.6: Workflow Diagram relativo al processo 'Creazione progetto'

## Visualizzazione dashboard

Il processo di *Visualizzazione della dashboard*, avviene in modo fluido e naturale, una volta che i dati sono stati inseriti nel database ed è basato su una rappresentazione accurata delle informazioni che provengono direttamente dai dati raccolti relativi al progetto di cui si vuole avere una panoramica.

L'evento che scatena il processo è il settaggio di filtri che implica l'interazione dell'applicazione web con il DBMS mediante delle query per poter ricevere informazioni utili alla realizzazione dei grafici richiesti.

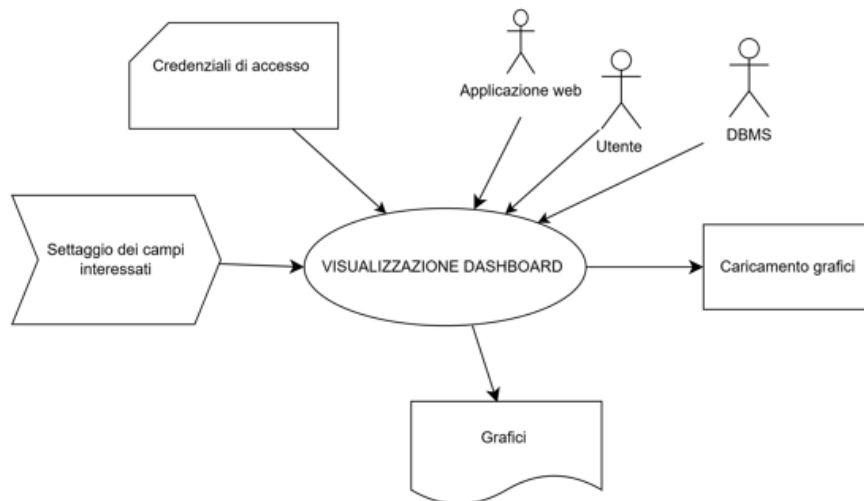


Figura 2.7: Synopsis Diagram relativo al processo 'Visualizzazione dashboard'



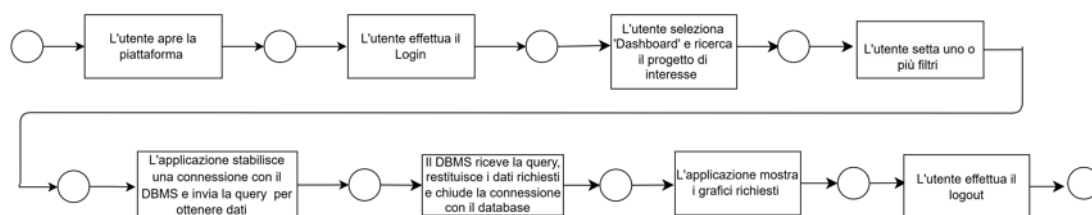


Figura 2.8: Workflow Diagram relativo al processo 'Visualizzazione dashboard'

## 2.2.2 Modellizzazione delle specifiche

La modellizzazione delle specifiche è uno step interno alla progettazione dell'applicazione web, che rientra nella progettazione di dettaglio descritta nel V-model che è stato seguito.

È una fase cruciale per tenere traccia dei passaggi implementati, permettendo di replicare lo sviluppo in futuro.

In questa fase si lavora parallelamente sulla progettazione dell'applicazione e del database, garantendo che le loro componenti operino in sinergia, per un funzionamento armonioso.

La modellizzazione consiste nella realizzazione dello *Use Case Diagram*, degli *oggetti entità* e delle *interfacce utente*.

Per ogni caso d'uso, il flusso delle attività è descritto mediante *Activity Diagram*, che elenca tutti i possibili percorsi ad alto livello che l'utente può intraprendere, senza entrare nel dettaglio poiché durante lo sviluppo effettivo verranno mostrate le varie attività associate ad ogni caso d'uso.

I casi d'uso individuati e mostrati nello use case diagram in Figura 2.9 rappresentano le funzionalità fornite dall'applicazione web agli attori e in particolare nello use case diagram si mostra l'interazione tra loro [25].

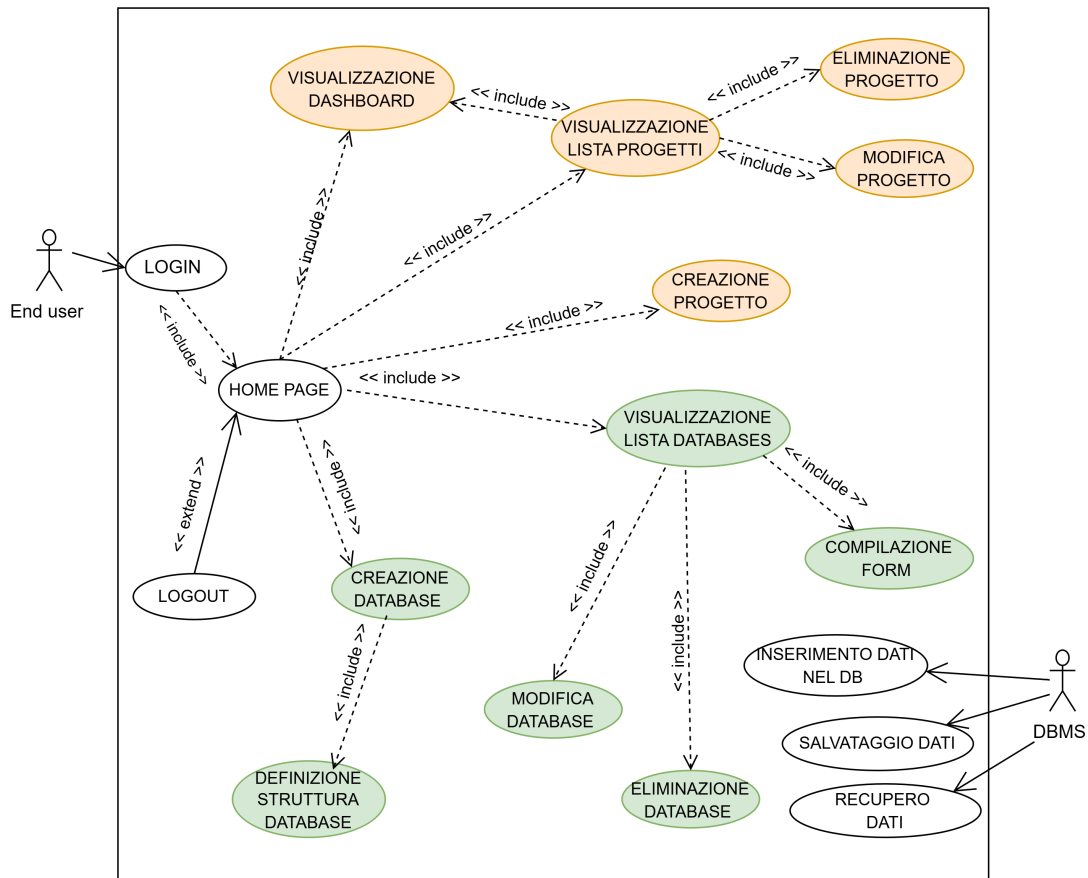


Figura 2.9: Use case Diagram dell'applicazione web

Gli attori sono sia entità umane, in generale definiti come utenti, sia software o altri sistemi come il DBMS e l'applicazione web stessa.

L'applicazione web è rappresentata dalla system boundary box, viene considerato come utente anche il DBMS, in quanto interagisce con l'applicazione al fine di inserire, salvare dati nel database e recuperare dati dal database per fornirli all'applicazione web.

Il caso d'uso 'Home page' include i casi d'uso principali ovvero 'Creazione database', 'Creazione progetto' e 'Visualizzazione dashboard', scelti come processi nelle fasi precedenti.

Sono stati individuati altri due casi d'uso 'Visualizzazione lista progetti' e 'Visualizzazione lista databases', avendo pensato ad una applicazione web che potesse ospitare più progetti e che desse la possibilità di gestire un database di raccolta dati relativi ai progetti creati.

Questi ultimi due casi d'uso a loro volta includono altri casi d'uso, che svolgono delle funzioni aggiuntive di modifica ed eliminazione dei progetti o dei database già esistenti.

A tal proposito, per agevolare la comprensione dello use case diagram in Figura 2.9, si è scelto di evidenziare gli use case relativi ai progetti e al database utilizzando colori differenti.

Il caso d'uso 'Compilazione form' è stato identificato come caso incluso di 'Visualizzazione lista databases', anziché come caso incluso di 'Creazione database', pur essendo implicito che nella maggior parte dei casi la creazione di un database comporti anche il suo popolamento. Questa scelta è stata fatta per permettere all'utente di inserire un singolo dato da importare, con la possibilità di caricare anche un file per importare più dati contemporaneamente, così da rendere l'applicazione web più user-friendly e flessibile.

Inoltre, il caso d'uso 'Creazione database' include il caso d'uso 'Definizione struttura database', per rendere accessibile l'applicazione web a chiunque, in particolare agli utenti che non hanno familiarità con il linguaggio SQL.

La costruzione degli oggetti entità, caratteristica della fase di modellizzazione delle specifiche, verrà approfondita durante la progettazione del database relazionale, poiché utile alla costruzione del diagramma E-R, del modello relazionale e del database fisico.

Successivamente alla realizzazione dello Use Case Diagram, si è scelto di progettare le interfacce utente, pensate e abbozzate nel dettaglio per seguire il più possibile i requisiti emersi nella fase di definizione dei requisiti software.

È stato ritenuto opportuno disegnare le interfacce utente prima della costruzione degli activity diagram relativi ad ogni use case, per dare forma visiva a ciò di cui gli utenti dovranno usufruire. Perciò, una volta disegnate le interfacce è stato necessario definire anche il flusso di tutte le funzionalità rese disponibili da ogni use case durante l'utilizzo dell'applicazione web.

Nel contesto degli use case, 'Home page' include il caso d'uso di 'Login' perché per accedere alle funzionalità principali dell'applicazione web, è obbligatoria l'autenticazione per consentire l'accesso solo agli utenti autorizzati. A sua volta, lo use case di 'Home page' estende il caso d'uso di 'Logout' perché è un caso d'uso opzionale a cui l'utente può accedere dalla Home page.

Di seguito sono mostrate l'activity diagram e l'interfaccia utente del caso d'uso 'Login' in Figura 2.10 e quelli del caso d'uso 'Home page' in Figura 2.11.

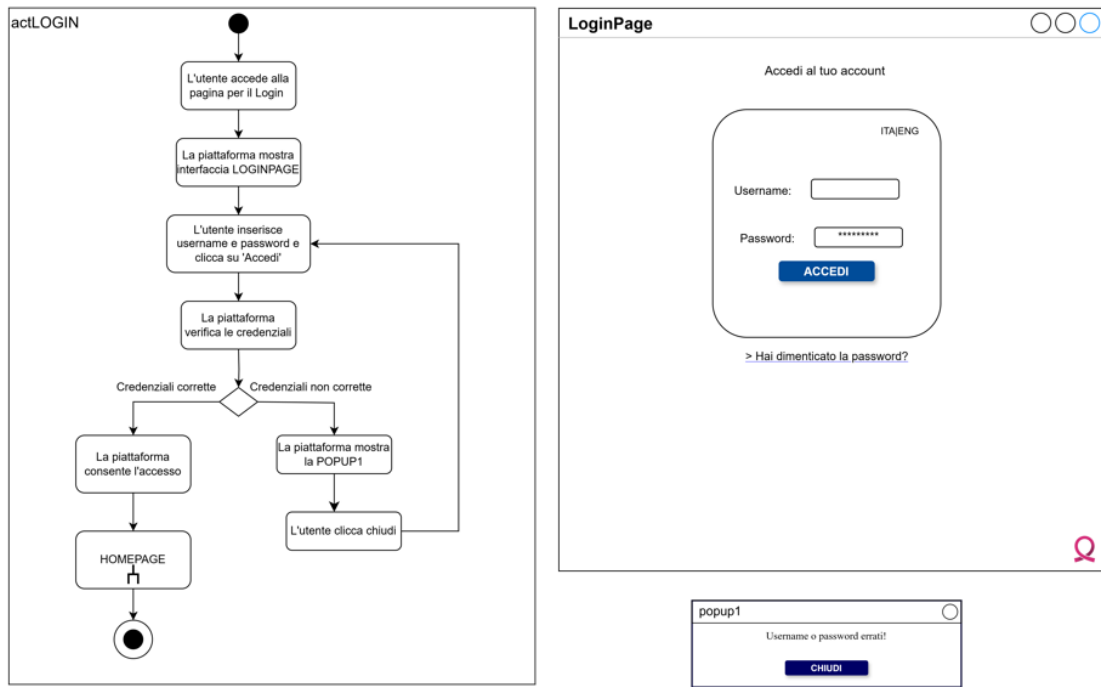


Figura 2.10: Activity diagram e interfaccia utente di *Login* con relativa popup

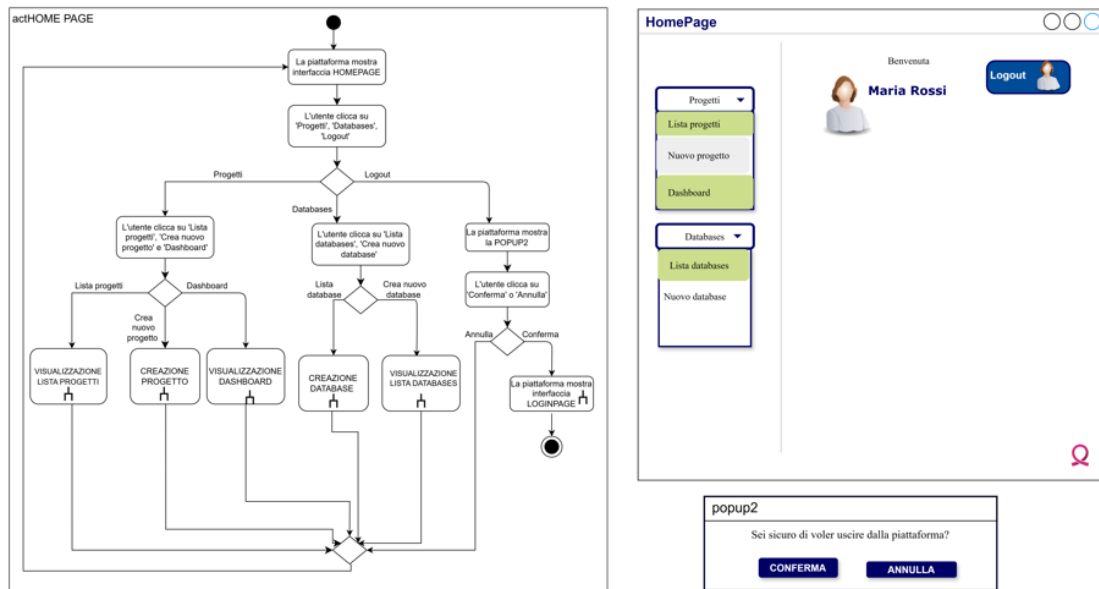


Figura 2.11: Activity diagram e interfaccia utente di *Home page* con relativa popup

Partendo dagli use case relativi alla gestione dei progetti, di seguito sono mostrate le interfacce e gli activity diagram relativi a due tra i principali processi individuati nella fase precedente di definizione delle specifiche. In particolare, per il caso d'uso di 'Creazione progetto' sono riportati activity diagram e interfacce con popup relative nelle Figure 2.12 e 2.13 rispettivamente.

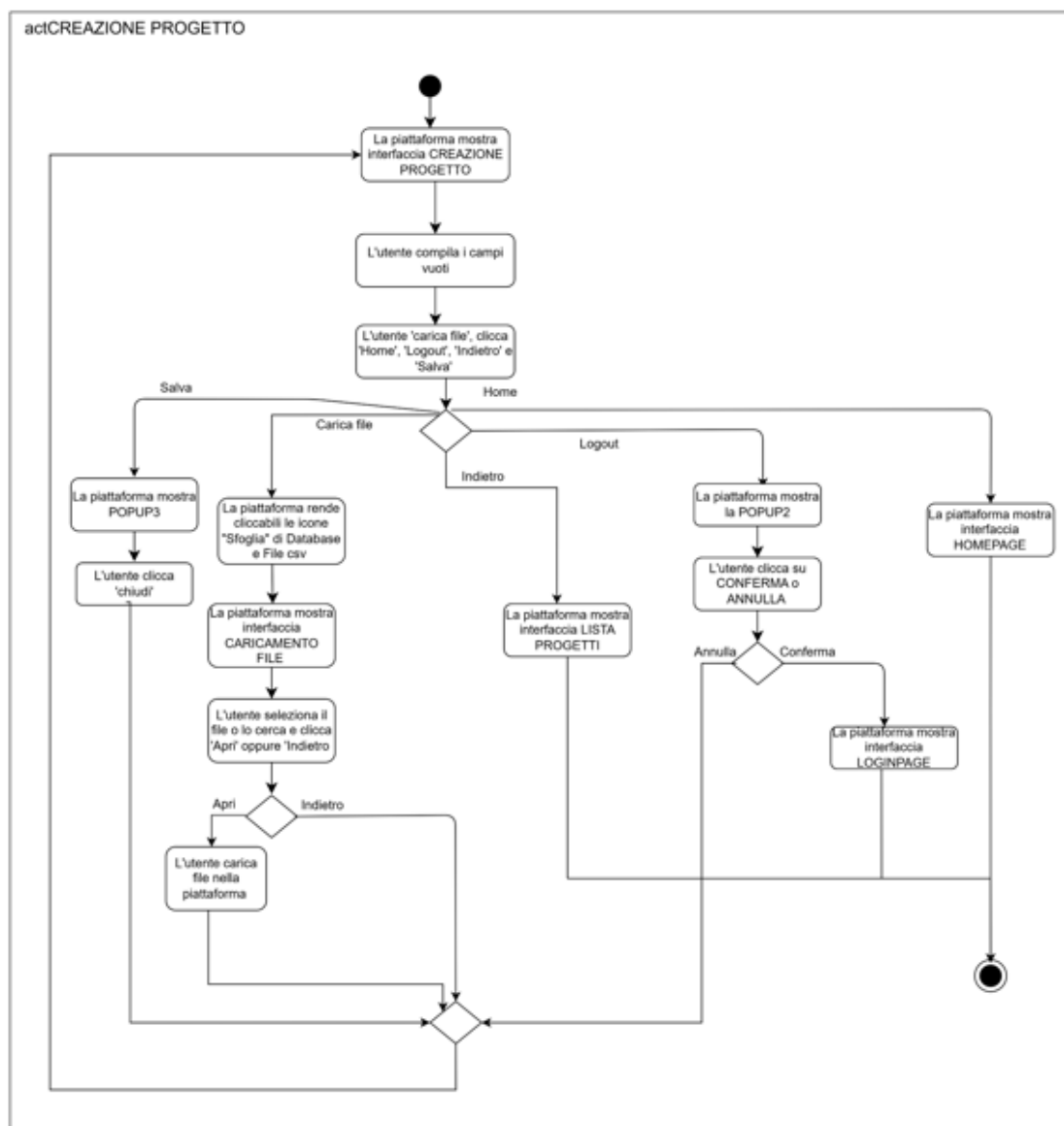


Figura 2.12: Activity diagram di *Creazione progetto*

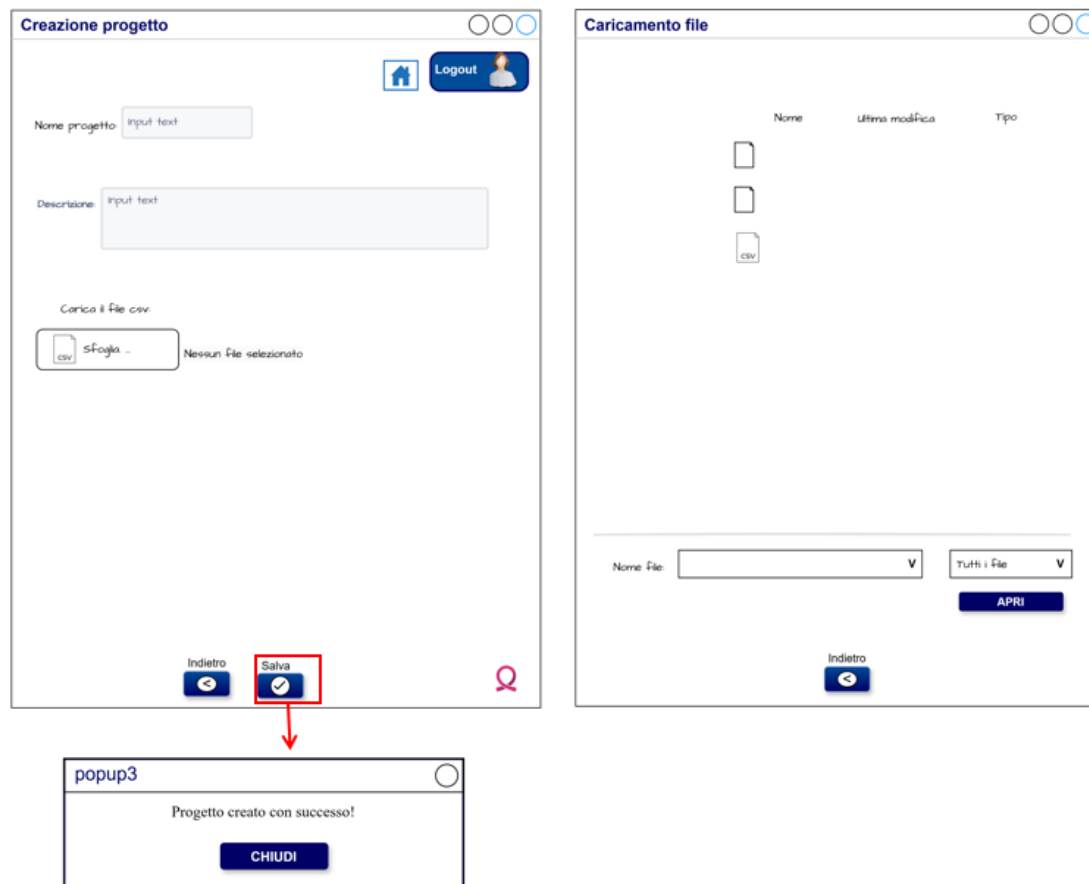


Figura 2.13: Interfacce utente di *Creazione progetto*

Per il caso d'uso 'Visualizzazione dashboard' sia gli activity che le interfacce utente sono riportati in Figura 2.14.

In particolare, il caso d'uso 'Visualizzazione dashboard' è già specifico del progetto sul quale il lavoro di tesi si è basato e verrà approfondito nel prossimo capitolo, essendo uno dei principali obiettivi richiesti per l'applicazione web.

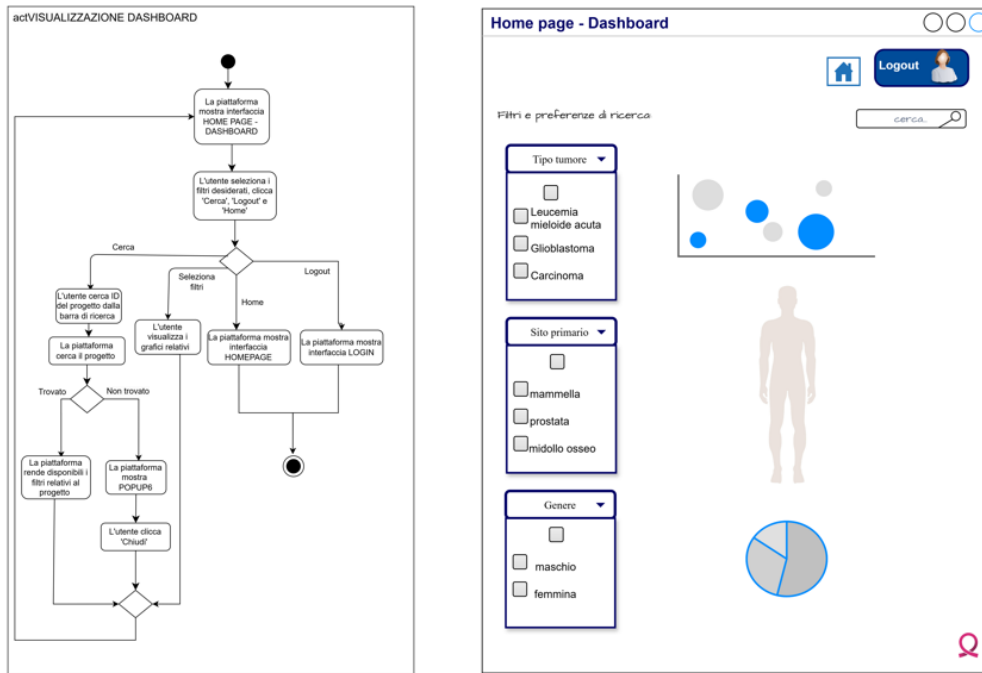


Figura 2.14: Interfaccia e Activity diagram di *Visualizzazione dashboard* rispettivamente

Poiché si è deciso di focalizzarsi su un solo caso d'uso per testarne lo sviluppo, è stato selezionato il caso d'uso 'Visualizzazione lista progetti'.

L'interfaccia utente relativa a questo caso d'uso è stata progettata con una struttura diversa rispetto alle altre, poiché durante lo sviluppo è emerso che la libreria Streamlit consente di mantenere sempre accessibile il menù con le funzionalità principali visibili in ogni interfaccia e viene consentita la visualizzazione dei cambiamenti nelle interfacce del caso d'uso stesso, senza la creazione di interfacce secondarie.

È emersa l'utilità della presenza del menù principale, perchè rende flessibile l'applicazione web e consente all'utente una facile navigazione tra le schede.

L'activity diagram del caso d'uso è mostrato in Figura 2.15 e le relative interfacce utente in Figura 2.16.

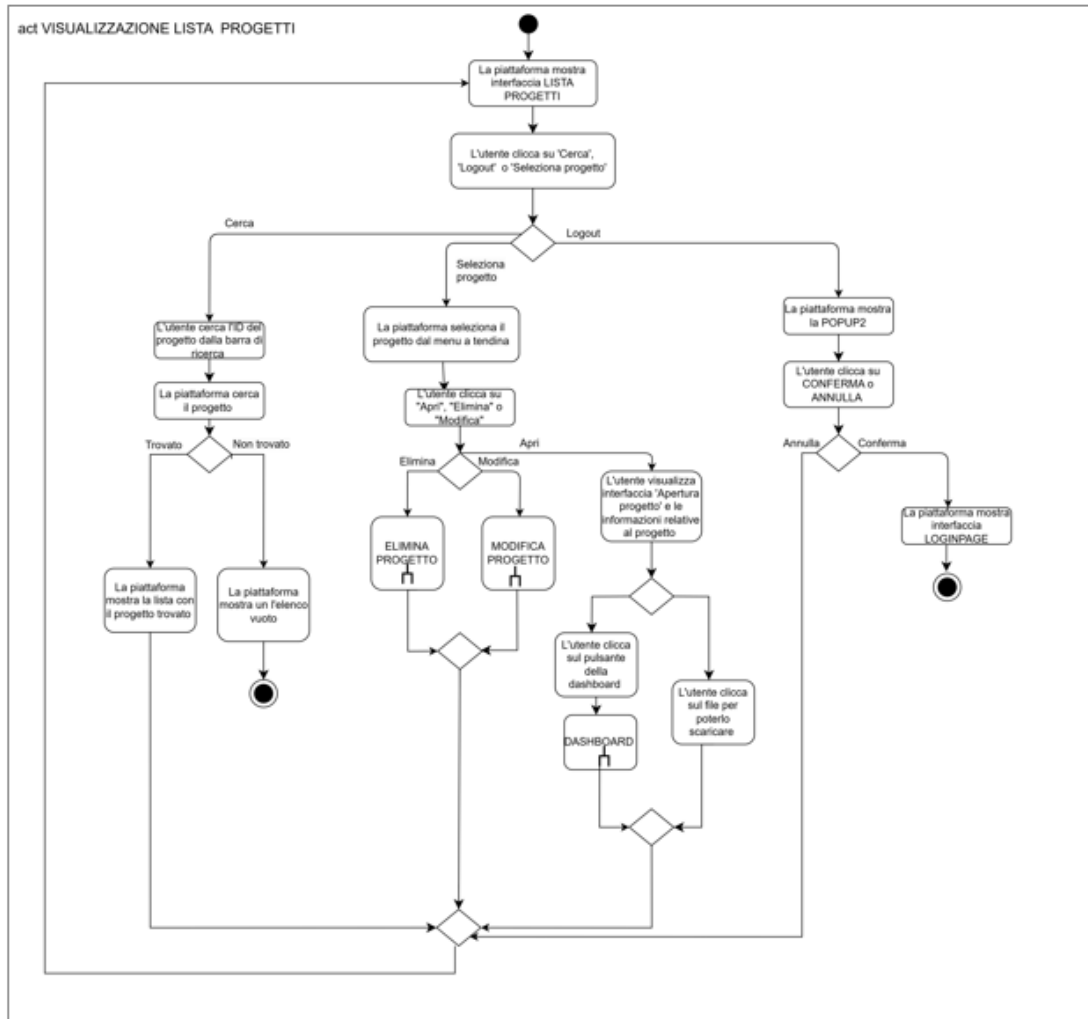


Figura 2.15: Activity diagram di *Visualizzazione lista progetti*



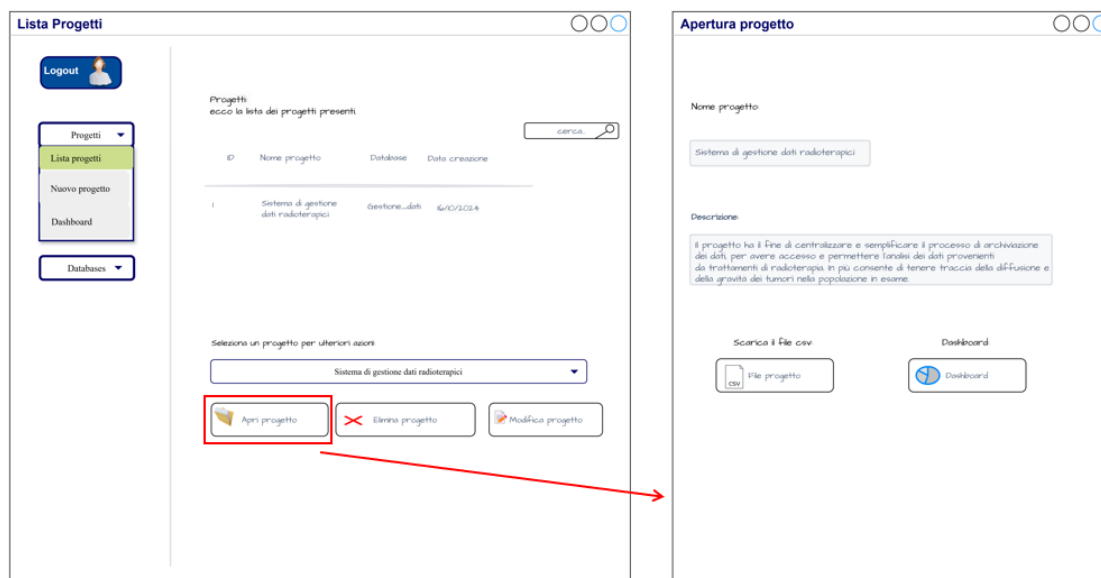


Figura 2.16: Interfacce utente di *Visualizzazione lista progetti* e la relativa interfaccia 'Apertura progetto'

Per il caso d'uso 'Visualizzazione lista progetti' è prevista un'altra interfaccia utente per aprire la scheda relativa ad uno specifico progetto e per ottimizzare l'aspetto visivo, però è considerata parte integrante del caso d'uso stesso.

L'interfaccia utente di apertura di un progetto è realizzata basandosi sui dati usati come esempio e a sua volta consente con un percorso secondario di visualizzare una dashboard del progetto in questione. Rimanda ad un caso d'uso che ha lo stesso obiettivo del caso d'uso 'Visualizzazione dashboard', mostrato nell'interfaccia in Figura 2.14.

L'interfaccia per la visualizzazione della dashboard partendo dall'interfaccia di apertura di un progetto specifico differisce per alcuni particolari da quella dello use case 'Visualizzazione dashboard'.

Per questo motivo, l'interfaccia utente e l'activity diagram sono diversi e sono mostrati in Figura 2.17.

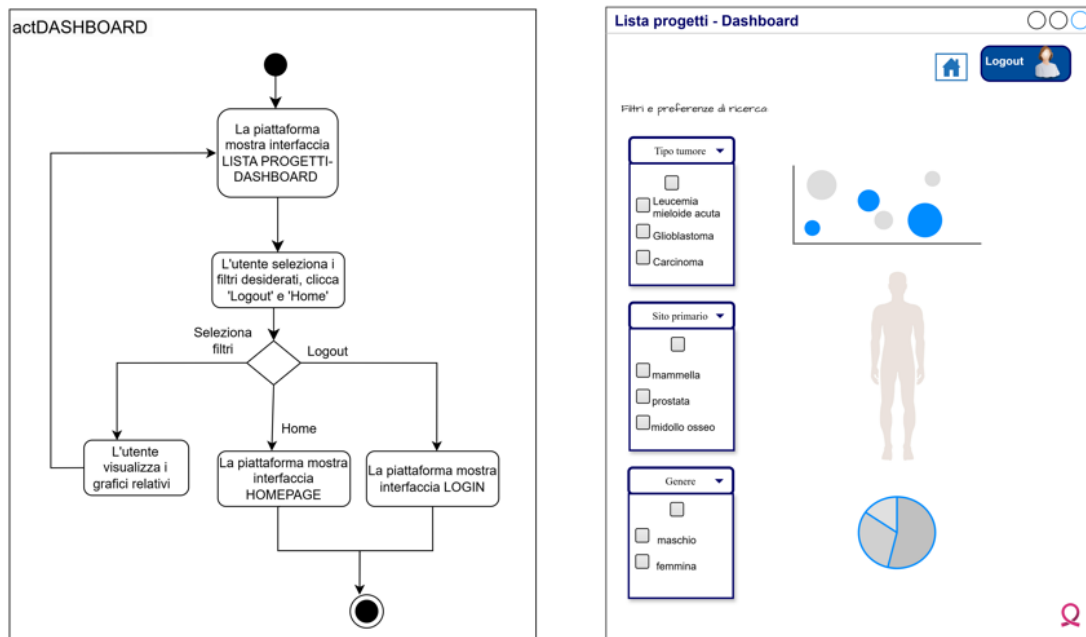


Figura 2.17: Interfaccia e Activity diagram per visualizzare la dashboard dall'interfaccia Apertura progetto

Come mostrato nell'interfaccia utente 'Lista progetti' in Figura 2.16, l'utente ha la possibilità di accedere ai casi d'uso di modifica ed eliminazione di uno specifico progetto.

In particolare, per il caso d'uso 'Eliminazione progetto', è stato necessario partire dall'interfaccia relativa al caso d'uso 'Visualizzazione lista progetti' per agevolare la comprensione del flusso delle attività.

In Figura 2.18 è possibile visualizzare gli activity diagram e le interfacce utente relative.

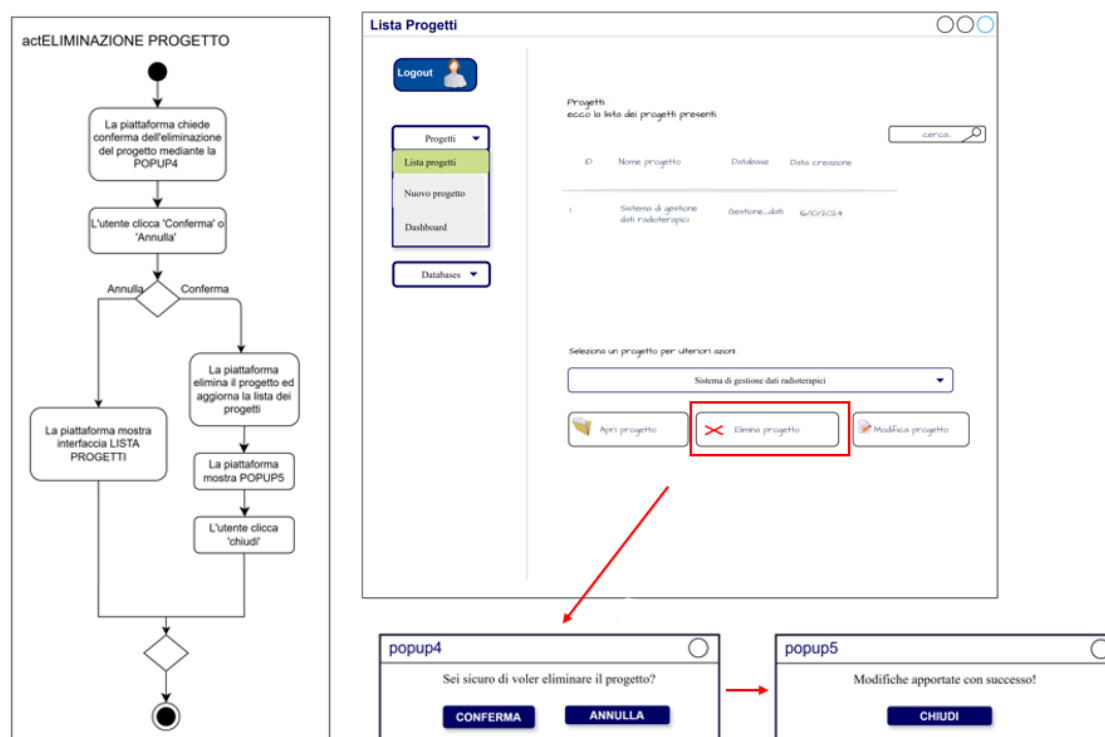


Figura 2.18: Activity diagram di *Eliminazione progetto*, interfaccia di 'Visualizzazione lista progetti' con le popup relative al caso d'uso *Eliminazione progetto*

In Figura 2.19 è mostrato l'activity diagram relativo allo use case 'Modifica progetto'. Per mostrare il percorso effettuato, è stato necessario partire dall'interfaccia relativa al caso d'uso 'Visualizzazione lista progetti', mostrata in Figura 2.18, per poi accedere all'interfaccia specifica dello use case 'Modifica progetto', compresa l'interfaccia di caricamento di un file riassuntivo, mostrate in Figura 2.20.

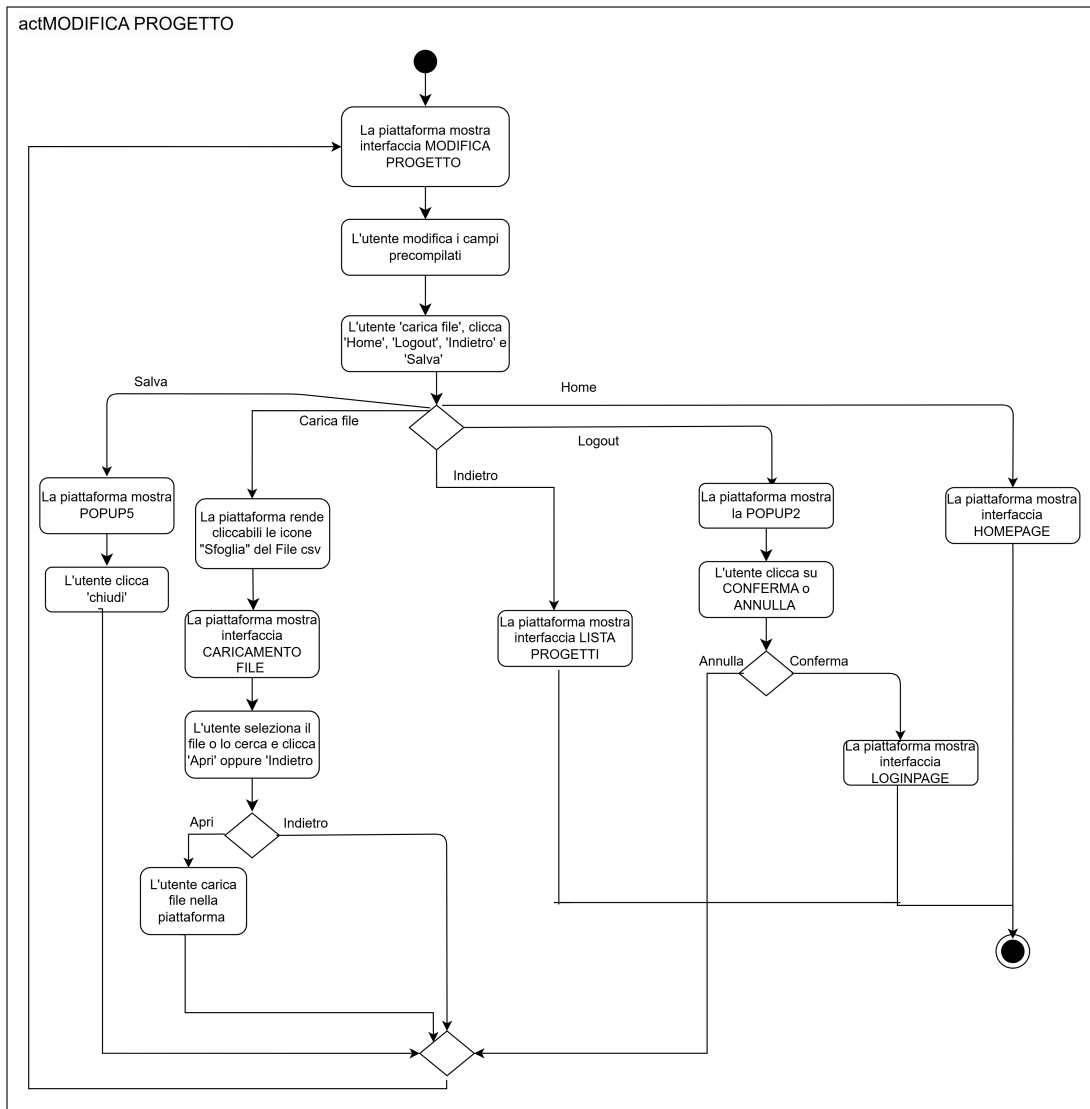


Figura 2.19: Activity diagram di *Modifica progetto*

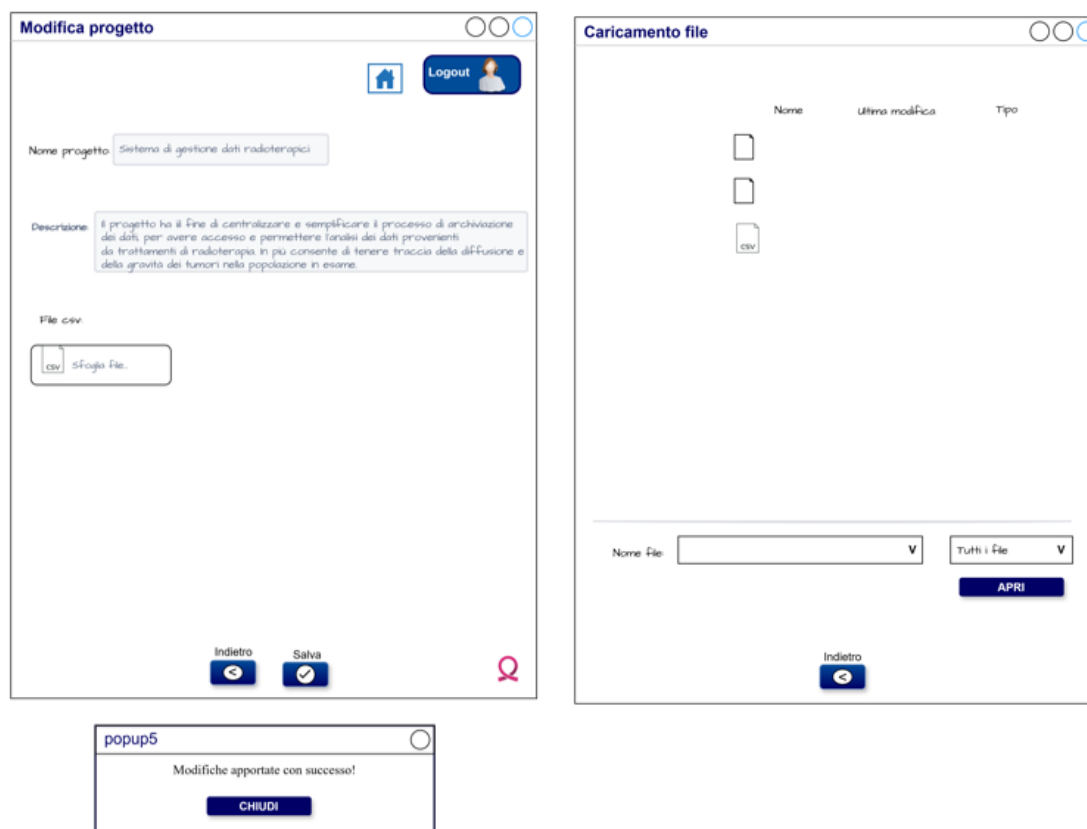


Figura 2.20: Interfacce di *Modifica progetto*

Per quanto riguarda la sezione relativa alla gestione di database per la raccolta dati di diversi progetti, le schede relative ai database presenti nella piattaforma sono molto incentrate sulla costruzione e la definizione della struttura di un nuovo database.

Al contrario nel caso delle interfacce relative alla gestione dei progetti, il fulcro è relativo alla visualizzazione della dashboard per avere una panoramica del progetto in questione.

Avendo deciso di sviluppare solo la sezione dell'applicazione web relativa ai progetti, la sezione dedicata alla gestione del database è stata solamente progettata. Perciò, è possibile trovare le figure relative agli activity diagram e alle interfacce utente nell'Appendice A dell'elaborato di tesi.

## 2.3 Progettazione e costruzione del database

La progettazione e costruzione di un software comprende sia la componente front-end, ovvero l'applicazione web con il quale l'utente interagisce, sia la componente back-end, ovvero il database e la logica applicativa che gestisce l'interazione con i dati.

Tutti gli aspetti relativi alla progettazione e alla costruzione sono stati affrontati in modo integrato, con l'obiettivo di sviluppare un sistema armonizzato, in cui ogni parte contribuisce a una visione unitaria.

Si è optato per l'adozione di un database relazionale perché è una soluzione semplice e flessibile, soprattutto considerando possibilità dell'utente di definire e modificare la struttura del database tramite l'applicazione web.

Inoltre, un database relazionale supporta facilmente eventuali modifiche alla struttura senza compromettere l'integrità complessiva dei dati.

### 2.3.1 Progettazione concettuale

Il modello concettuale di una base dati è descritto mediante il diagramma E-R, basato sugli oggetti entità specifici della fase di modellizzazione delle specifiche.

Questo diagramma consente di avere una visione immediata dei dati che il database dovrà ospitare, descritti mediante gli oggetti entità e avere una visione delle relazioni che li legano. Perciò, verrà mostrato direttamente il diagramma realizzato mediante il software Diagrams.net, con gli oggetti entità contenuti all'interno.

Gli oggetti entità individuati sono stati raccolti partendo dai dati relativi al progetto TGCA. Sono stati individuati altri oggetti entità relativi alla gestione dell'applicazione web, tra i principali ci sono gli utenti che possono avere accesso all'applicazione e i progetti che l'applicazione web ospita.

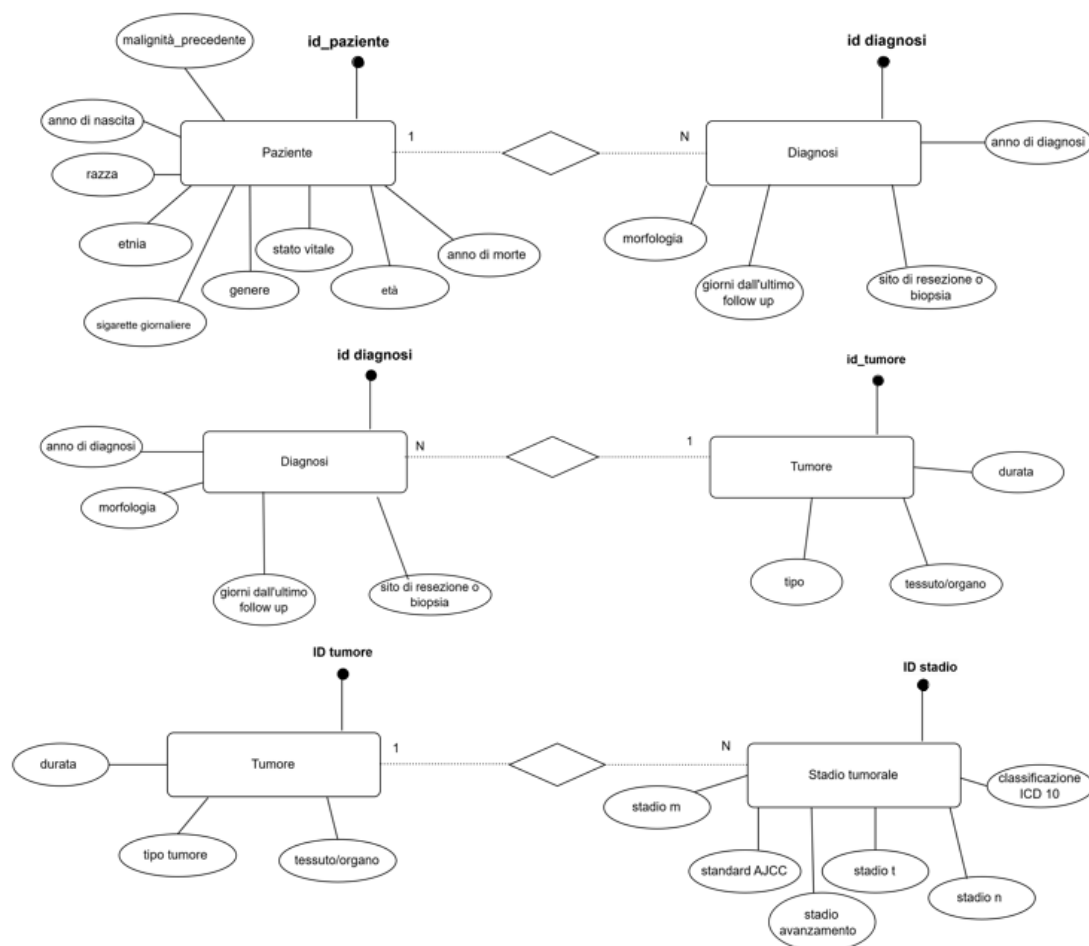
Per semplicità di comprensione e interpretazione delle relazioni che legavano le entità, si è scelto di relazionarli a coppie.

Partire con relazioni chiare e dirette ha agevolato la comprensione delle connessioni principali, senza introdurre connessioni più complesse che non sono evidenti ma che possono emergere man mano che il modello evolve nelle successive fasi di progettazione.

In questa fase progettuale, il modello costruito rispetto alle fasi successive è astratto, flessibile e indipendente dalle specifiche tecniche del database che verrà usato per implementarlo, questo per rendere il sistema più facilmente adattabile a future implementazioni.

Nello specifico, i dataset utilizzati per lo sviluppo dell'applicazione web e per avere dei dati di partenza per il riempimento del database presentano tutti la stessa struttura, ciò ha permesso di individuare le entità e le relazioni con una struttura omogenea per tutti i dataset raccolti.

In Figura 2.21 sono riportati i diagrammi E-R che presentano dati relativi ai progetti, invece in Figura 2.22 sono mostrati i diagrammi E-R relativi all'applicazione web.



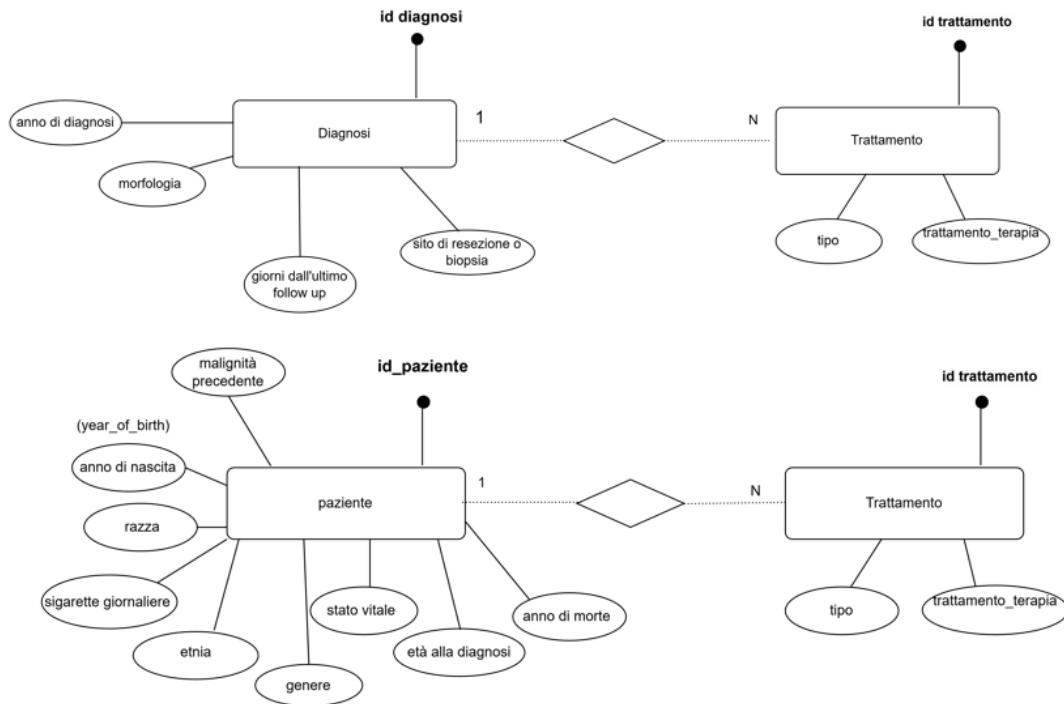


Figura 2.21: Diagrammi E-R relativi ai dati dei progetti

Per quanto riguarda i dati relativi ai progetti, ci si è basati sul set di dati specifico fornito ma la scelta degli attributi è stata piuttosto ampia, senza decidere a priori gli attributi utili, per poi selezionare quelli più pertinenti man mano che il progetto prende forma.

Dalla rappresentazione fornita e dato il livello di specificità degli oggetti entità, si può notare come le entità per tutte le combinazioni siano legate tramite una relazione uno-a-molti (1:N), in cui ciascuna entità secondaria mantiene un legame chiaro e diretto con quella principale che invece può essere legata a più entità correlate.



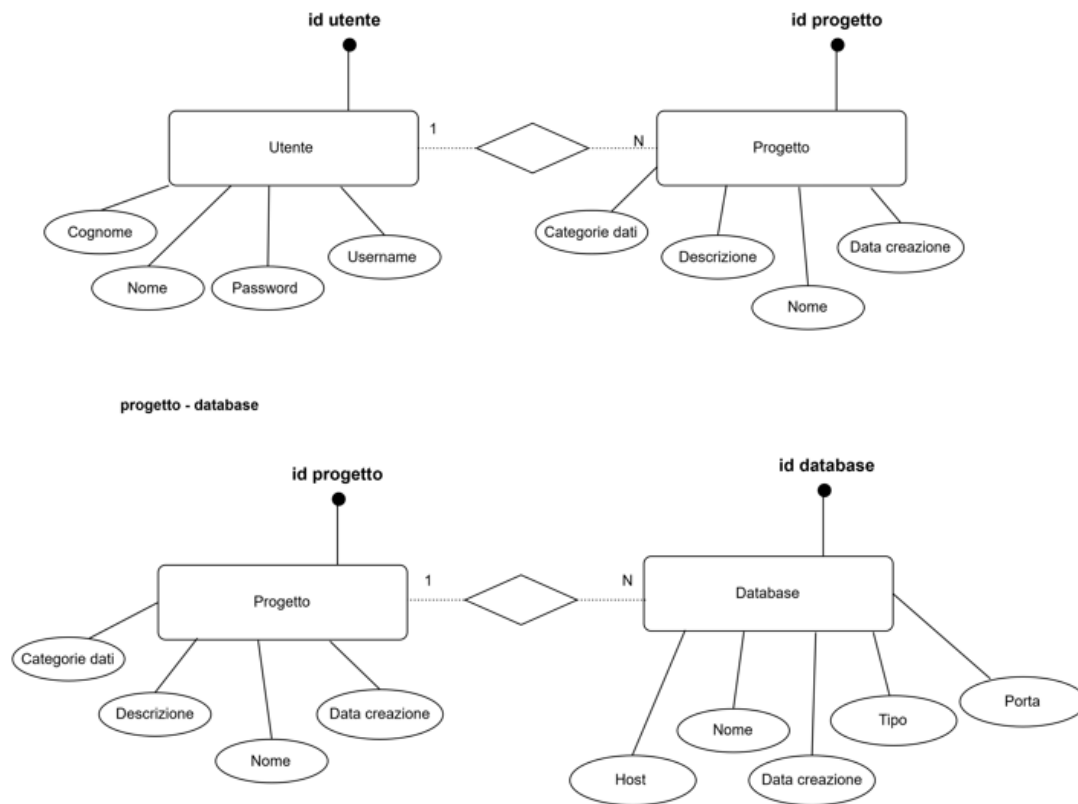


Figura 2.22: Diagrammi E-R relativi ai dati utili all'applicazione web

I dati relativi all'applicazione web, mostrati in Figura 2.22 riguardano gli aspetti legati alla gestione dell'accesso, alle interazioni degli utenti e alla loro relazione con i progetti.

Le tabelle specifiche per l'applicazione web sono necessarie per separare e organizzare in modo chiaro e strutturato le informazioni relative agli utenti e alle loro azioni all'interno del sistema, assicurando una gestione efficiente delle informazioni per facilitare la scalabilità dell'applicazione web.

Per quanto riguarda la gestione delle autorizzazioni all'accesso, i diagrammi E-R sono stati realizzati abbastanza generici e a tutti gli utenti registrati viene data la possibilità di accedere ai progetti specifici, crearne nuovi, apportare modifiche o prendere visione dei dati.

### 2.3.2 Progettazione logica e fisica

La progettazione logica del database consiste nella costruzione del diagramma EER o Enhanced Entity-Relationship, ottenuto dai diagrammi E-R o Entità-Relazione della progettazione concettuale realizzati con il software MySQL Workbench.

Anche in questa fase progettuale si è scelto di distinguere la parte di gestione dei progetti dalla parte di gestione dell'applicazione web.

Il diagramma EER a differenza di quello E-R converte gli oggetti entità in tabelle, avendo scelto un database relazionale. Inoltre, le relazioni non sono più legami diretti uno-a-uno, ma vengono definite in modo più articolato, creando connessioni complesse tra tutte le tabelle. In questo modello oltre alle chiavi primarie, PK o Primary Keys sono state inserite le chiavi esterne, FK o Foreign Keys, fondamentali per instaurare le relazioni e le dipendenze tra tabelle. Inoltre, sono specificati i tipi di dati per ogni attributo di tutte le tabelle, indicando la lunghezza massima o la precisione.

L'introduzione di questi vincoli consente di definire uno scheletro del database, fornendo una base solida per la progettazione fisica.

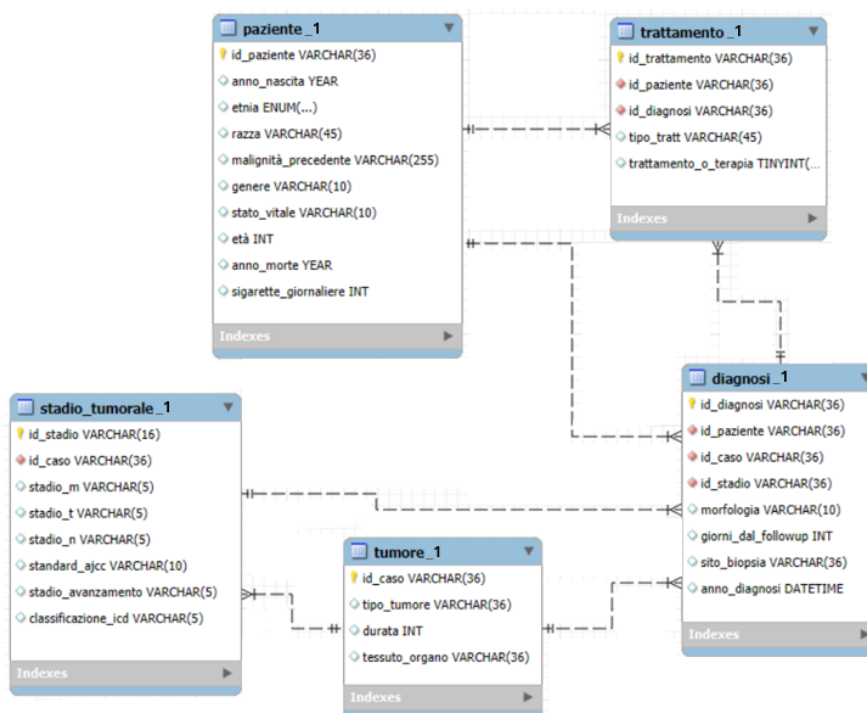


Figura 2.23: Modello relazione EER del database sulla sezione relativa alla gestione di un progetto specifico

Come si evince dal diagramma EER relativo ai dati utili all'applicazione web, per la tabella *progetto* si è scelto di inserire l'attributo 'categorie dati' in formato json caratterizzato da una chiave che contiene la lista delle tabelle relative agli specifici progetti.

Questo approccio è stato utilizzato per unire la sezione relativa ai progetti con quella dedicata alla gestione dell'applicazione web, motivo per cui non è stato necessario inserire una relazione fisica tra la tabella *progetto* e tutte le altre. Inoltre, è stata pensata per poter gestire progetti diversi che possono avere entità simili, ma con attributi specifici per ciascun progetto.

Inoltre, le tabelle relative alla gestione dell'applicazione web presentano un nome caratterizzato da un suffisso numerico che corrisponde all'id del progetto in questione, per differenziare versioni di una tabella in cui il nome è lo stesso ma la struttura e i suoi attributi variano in base al progetto.

Un esempio è la tabella *paziente* che potrebbe essere ricorrente nei vari progetti, ospitando diversi progetti e principalmente nel settore clinico.

La tabella *db* è stata creata perchè l'applicazione web contiene l'opzione di creazione database e l'utente potrebbe voler creare un database specifico per un progetto se riguarda un ambito completamente diverso da quello del database esistente.

Nell'esempio specifico sviluppato con l'applicazione web, è stato ritenuto opportuno creare un solo database relazionale anche per le informazioni riguardo l'applicazione web perchè avere più di un database potrebbe risultare dispendioso, poco flessibile e poco ottimizzata.

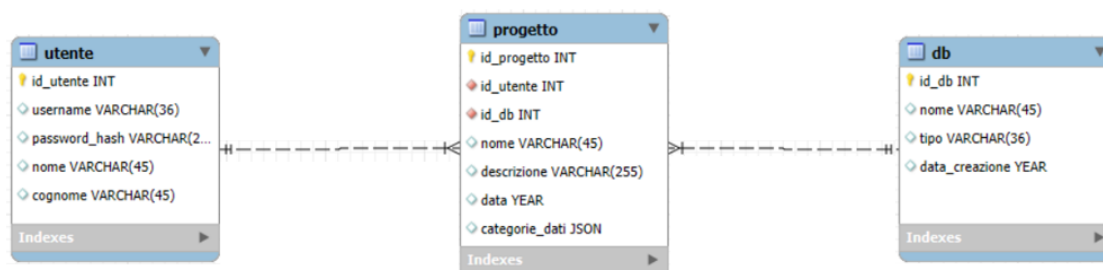


Figura 2.24: Modello relazione EER del database sulla sezione relativa alla gestione dell'applicazione web

È stato utilizzato il tool *forward engineering* di MySQL Workbench per convertire il diagramma EER in un database fisico. Inoltre, il software utilizzato offre l'opzione di reverse engineering, che consente di eseguire la conversione inversa.

Durante la progettazione fisica, in concomitanza con la realizzazione del diagramma EER sono stati analizzati i dati presenti nei dataset forniti per lo specifico progetto usato per lo sviluppo della piattaforma.

Sono stati utilizzati formati per stringhe di testo o di lunghezza stabilita o con la possibilità di scelta dei valori predefiniti, numeri interi e infine l'anno per le date. La scelta dei formati verrà motivata nel successivo paragrafo relativo alla costruzione dell'applicazione web.

## 2.4 Costruzione dell'applicazione web

Di rilevante importanza è chiarire le intenzioni iniziali di sviluppare tutto ciò che è stato progettato a livello di applicazione web, ma che non è stato possibile realizzare in maniera completa.

Una revisione completa di tutte le interfacce utente e relativi flussi delle attività avrebbe richiesto un impegno significativo in termini di tempo e risorse, perciò si è deciso di privilegiare l'implementazione di funzionalità più rilevanti per l'esperienza dell'utente, ponendo l'attenzione sulla modifica della progettazione relativa ai casi d'uso relativi ai progetti che ho sviluppato e testato contemporaneamente, senza apportare modifiche a tutte le interfacce.

Questa scelta è stata fatta per le funzionalità relative ai progetti che sono state sviluppate, così da avere coerenza con la fase di progettazione annessa.

Entrando nel cuore dello sviluppo dell'applicazione web, quest'ultima è stata costruita parallelamente alla sua progettazione, mediante le bozze delle interfacce utente e parallelamente alla progettazione del database fisico.

Partendo dai dati dello specifico progetto usato come esempio nello sviluppo dell'applicazione web, per avere una base è stato necessario seguire alcuni step importanti:

### **Implementazione di un pre-processing e standardizzazione dei dati**

Mediante uno script Python è stato possibile automatizzare questa fase per avere i dati puliti, normalizzati e standardizzati prima di essere utilizzati nell'applicazione web.

Sono stati seguiti diversi passaggi mediante l'utilizzo di dataframe, ed è stato necessario:

- un filtraggio dei campi utili selezionati nelle fasi di progettazione del database, partendo dai file csv raccolti;
- un'omologazione tra i nomi dei campi tra i diversi file csv;
- generazione dinamica di codici identificativi univoci per le stadiazioni relative alla tabella *stadio tumorale*;
- conversione dei NaN in None per evitare possibili problemi di compatibilità durante l'inserimento nel database, proiettando il lavoro di pre-processing dei dati sull'importazione di questi ultimi nel database progettato e sviluppato;

- l'eliminazione delle righe che presentano degli identificatori univoci nulli per le tabelle che costituiranno il database relazionale;
- il riempimento dei campi vuoti con valori nulli e conversione dei valori decimali in interi per campi contenenti date, in particolare anni.

### **Importazione dei dati relativi al progetto**

Avendo diversi dataset, è stato necessario creare uno script Python per automatizzare l'importazione dei dati all'interno del database creato.

In particolare, è necessario configurare e stabilire la connessione con il database. Questo è un passaggio utile all'importazione automatica dei dati e sia la connessione che l'importazione sono possibili grazie a dei metodi di librerie Python.

### **Generazione random dei dati utili agli accessi all'applicazione web**

Dopo aver importato i dati relativi al progetto, non avendo un dataset di utenti già autenticati, è stato deciso di creare uno script per generare automaticamente le credenziali degli utenti, inclusi username, password e hash della password, per garantirne la sicurezza.

### **Sviluppo applicazione web**

Una volta gestiti i dati utili allo sviluppo dell'applicazione web, si è passati all'implementazione degli script per crearla, tenendo fede alle bozze delle interfacce progettate.

In questa fase è stata importante la connessione e l'interazione dell'applicazione web con il database relazionale, mediante query in linguaggio SQL.

Si è deciso di utilizzare la libreria Streamlit per la creazione dell'applicazione web interattiva, perchè mette a disposizione diversi widget che sono stati presi in considerazione in fase di progettazione dell'applicazione web.

Tra i vari widget, per la realizzazione dell'applicazione web sono stati utilizzati i button, i select box o multiselect box e il text o number input [26].

Nella progettazione relativa alle funzionalità per i database sono stati previsti i checkbox e i radio button.

Inoltre, Streamlit supporta diverse librerie per la rappresentazione grafica dei dati attraverso dei chart interattivi, utili per la visualizzazione della panoramica relativa ai dati di uno specifico progetto.

Per quanto riguarda il layout dell'applicazione web, si è scelto di utilizzare due aree:

- la sidebar, ovvero la barra laterale a sinistra della pagina che contiene il menù principale realizzato mediante selectbox, evidenziato con un riquadro rosso in Figura 2.25 e che contengono le funzionalità a cui l'utente deve poter accedere in ogni momento;

- il contenuto principale, che cambia dinamicamente in base alle azioni selezionate nella sidebar, visualizzando i risultati o le interazioni richieste dall'utente.

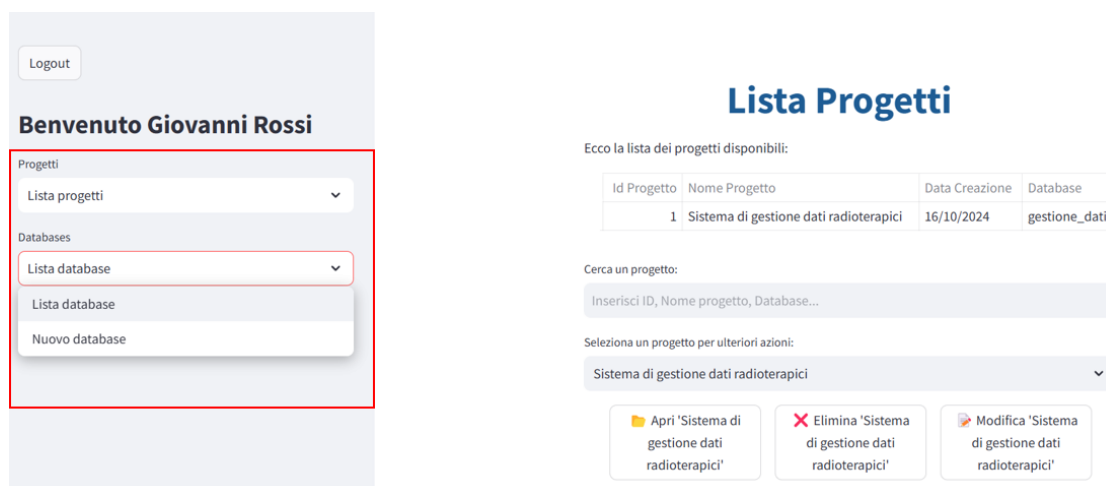


Figura 2.25: Esempio della sidebar nell'applicazione web caratterizzata da due sezioni di selectbox

Inoltre, Streamlit fornisce una flag da linea di comando che consente di eseguire lo script principale relativo all'applicazione web ed avere accesso direttamente dal proprio PC.

Inoltre, l'applicazione web è stata pensata nell'ottica di poter essere utilizzata anche da altri utenti, quindi eseguendo il comando viene garantita l'accessibilità tramite un indirizzo di rete, consentendo l'accesso anche da dispositivi esterni.

In Figura 2.26, è mostrato un esempio di funzionamento dell'applicazione web nell'interazione con il database relazionale in cui viene effettuato il Login all'applicazione web, come nella bozza di interfaccia mostrata in Figura 2.10, mostrando l'interazione con il database. In particolare, si evince la corrispondenza con lo username e con l'hash della password.

Cliccando il tasto *Accedi*, in automatico viene inviata una query per verificare la validità dello username e della password in chiaro sulla base dell'hash salvato nel database.

## Accedi al tuo account

Username

marbia11

Password

.....

ACCEDE

	id_utente	username	password_hash	nome	cognome
▶	1	marbia11	4c883c290def946f30ca7bb507f35459b73880f5...	Giovanni	Rossi
	2	marbia75	d82a4dfabead9fd637d61c9974656dade6df708...	Giovanni	Bianchi
	3	sargia26	74ab8070662d9e24e8b21c119a1ae00d524de1...	Giovanni	Verdi
	4	franer35	6046bcc578b7cc3aa2ccd6ee3849e17147fcc700...	Giovanni	Gialli
	5	marros36	e807de7ae3429512a6786fabe4c06c12b0fff246...	Giovanni	Neri
	6	gioner51	f930cc883349aaef8c6abcaea462da924069cbb...	Maria	Rossi
	7	elegia61	e74cef586c6e7c2440539b443f88c871b8c3bc1e...	Maria	Bianchi

Figura 2.26: Esempio di autenticazione di un utente tramite interfaccia di Login

Al contrario in caso di errore nelle credenziali, è previsto un errore come mostrato in Figura 2.27 e come progettato in Figura 2.10, il che implica che non c'è corrispondenza con le credenziali salvate nel database.



Figura 2.27: Esempio di autenticazione errata di un utente

Inoltre, una volta effettuato l'accesso, dalla sidebar a sinistra è sempre possibile uscire dall'applicazione web.

In Figura 2.28 è mostrata l'interfaccia con la popup che consente di effettuare il Logout.

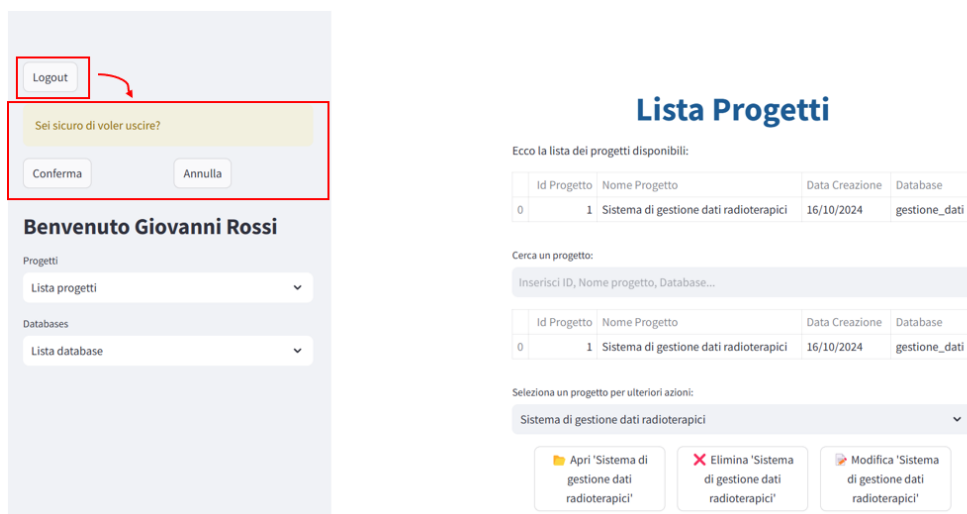


Figura 2.28: Esempio di Logout

La restante parte dell'applicazione web sviluppata verrà mostrata durante il testing e durante il capitolo di approfondimento relativo alla dashboard realizzata.



## 2.5 Testing dell'applicazione web

Seguendo l'approccio del V-model, è importante testare l'efficienza delle funzionalità sviluppate volta per volta, potendo usufruire dei dataset forniti dall'azienda del progetto TCGA di cui si parlerà più nel dettaglio nel capitolo successivo.

L'UML descrive tre fasi di testing: si individuano gli errori del sistema e si correggono, iniziando con delle operazioni di *Unit testing*, dedicate a testare le singole routine del codice e di *Integration testing*, tramite le quali si valuta l'unione tra le varie funzionalità, fino a raggiungere il *System Testing* in cui le attività di test sono divise in *Verifica* e *Validazione* [27].

È previsto un Testing Plan costruito a partire dallo Use Case diagram, considerando ogni caso d'uso e costruendo tutte le prove relative ai vari percorsi, sia principali sia alternativi contemplati negli activity diagram. Inoltre, è prevista una checklist, tramite la quale tenere traccia dei risultati delle prove.

Per ogni possibile percorso ideato negli activity diagram deve essere eseguita una prova, per verificare che tutto funzioni. Nello specifico, si è deciso di implementare una sola prova per un singolo caso d'uso, *Visualizzazione lista progetti* e viene testato il funzionamento partendo dal relativo activity diagram mostrato in Figura 2.15 e dalle bozze di interfacce progettate.

Di seguito in Figura 2.29, è riportato l'activity diagram del caso d'uso 'Visualizzazione lista progetti', con il percorso specifico evidenziato che si è deciso di testare.

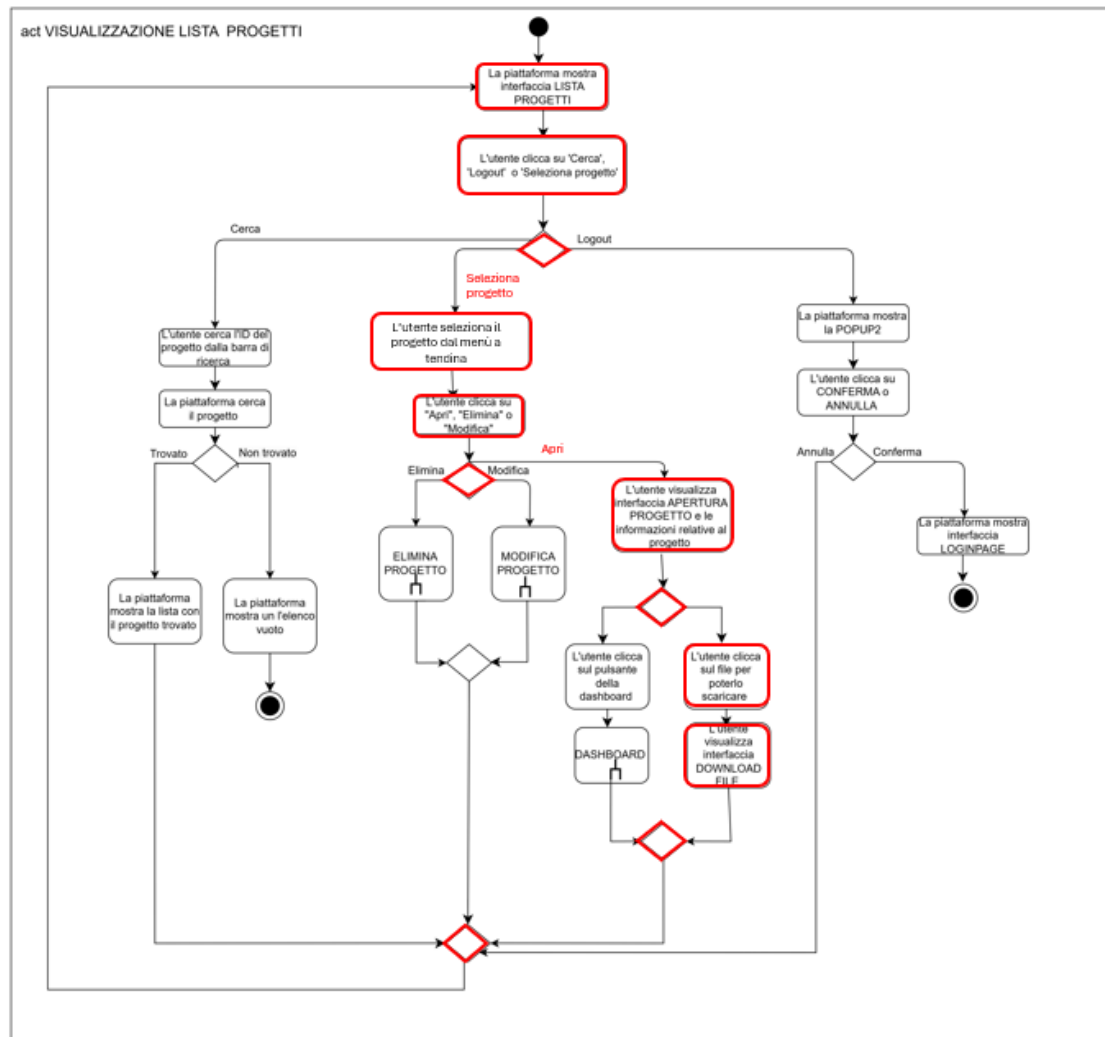


Figura 2.29: Activity diagram del caso d'uso *Visualizzazione lista progetti* con persorso relativo alla prova di testing

Valutando il percorso di apertura di uno specifico progetto, la sequenza delle attività da verificare sarebbe la seguente:

**PROVA 1 - APERTURA DI UNO SPECIFICO PROGETTO:**

1. Deve essere visualizzata l'interfaccia *Lista progetti*
2. Selezionare il progetto dal menù a tendina
3. Cliccare su APRI *nome progetto*
4. Deve essere visualizzata la scheda relativa al progetto
5. Selezionare FILE PROGETTO
6. Deve essere visualizzata interfaccia *Download file*
7. Selezionare cartella nel quale si vuole salvare il file
8. Cliccare su SALVA

Nel descrivere la prova, la dicitura *nome progetto* del pulsante indica il nome del progetto che viene selezionato nell'attività precedente e che varia in base a ciò che viene selezionato.

Le interfacce sviluppate che descrivono l'esecuzione delle azioni sono mostrate dalla Figura 2.30 alla 2.34.

Questa prova utilizzata per mostrare il testing è molto semplice e descrive uno dei percorsi relativo al caso d'uso scelto. Ogni volta che è stata implementata una piccola funzionalità, ne è stata verificata l'efficienza.

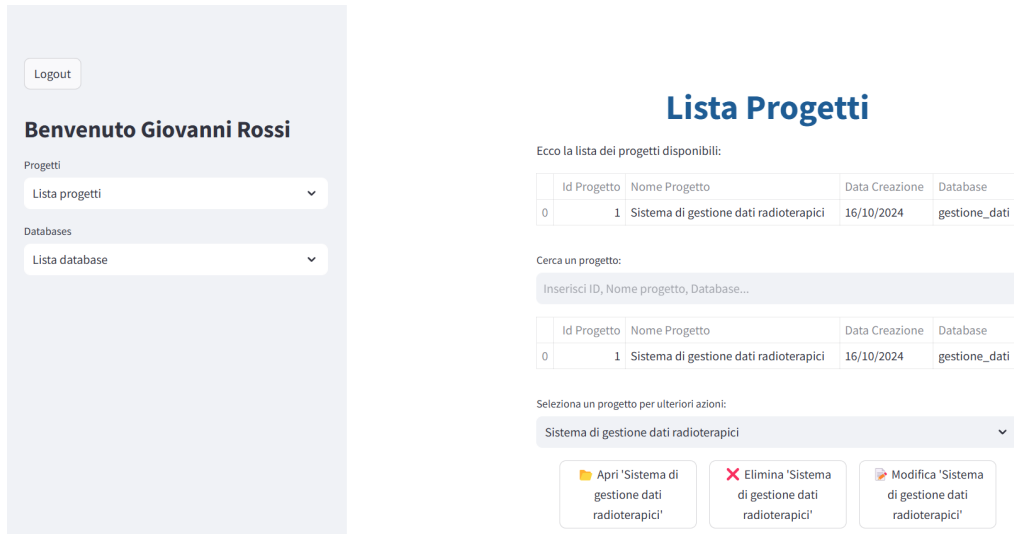


Figura 2.30: Prova 1 - Apertura di uno specifico progetto: step 1

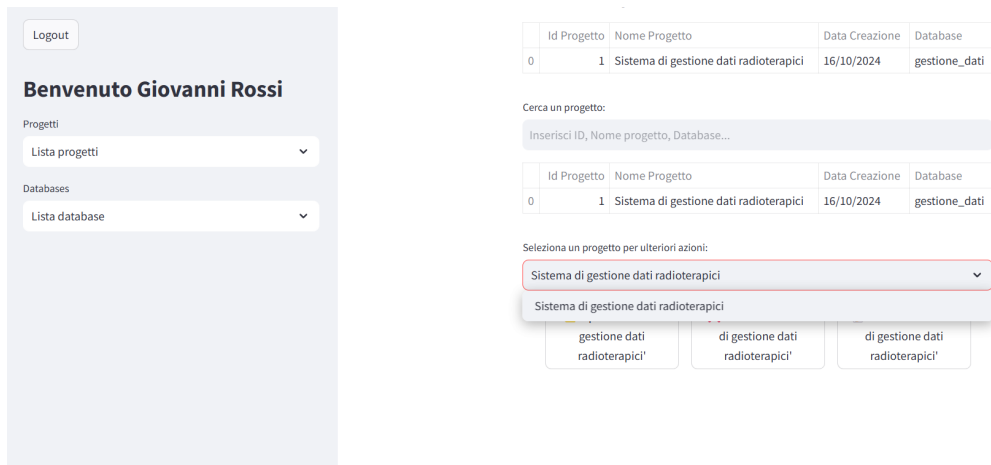


Figura 2.31: Prova 1 - Apertura di uno specifico progetto: step 2

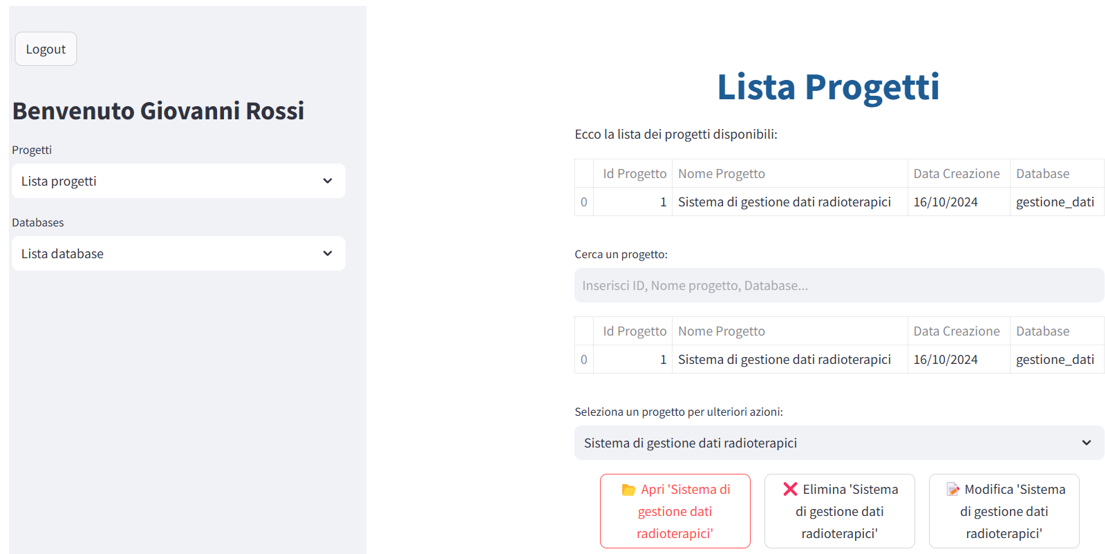


Figura 2.32: Prova 1 - Apertura di uno specifico progetto: step 3

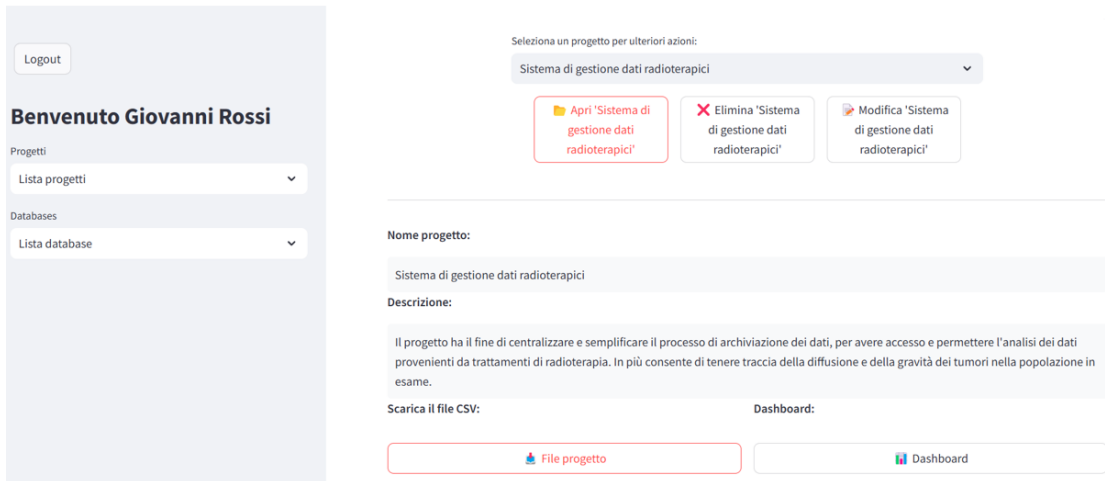


Figura 2.33: Prova 1 - Apertura di uno specifico progetto: step 4 e 5

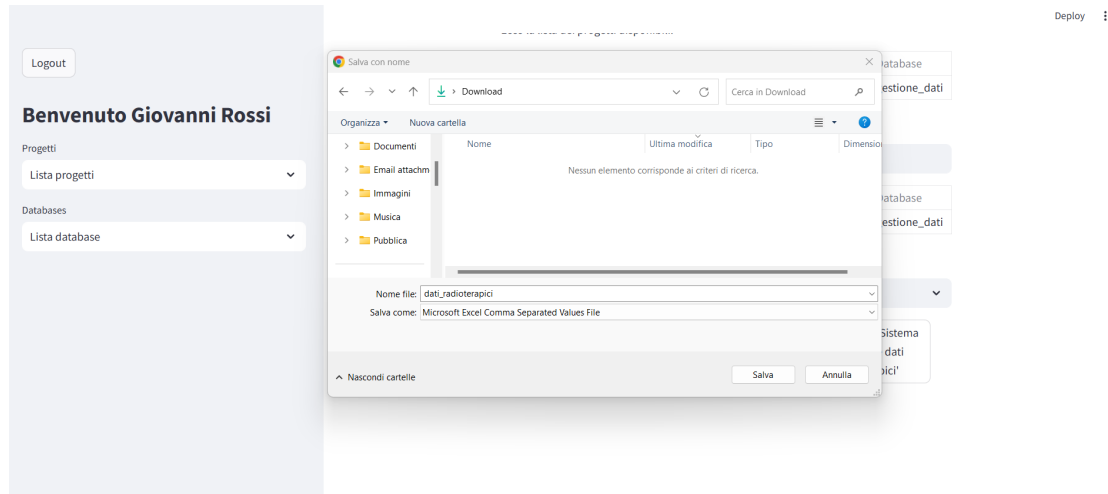


Figura 2.34: Prova 1 - Apertura di uno specifico progetto: step 6, 7 e 8

Dunque per seguire il V-model scelto come approccio, il testing non è stato eseguito alla fine della procedura di costruzione, ma ogni volta che veniva realizzato un percorso precedentemente progettato, testando contemporaneamente anche le operazioni di interfacciamento con il database.

Nella fase di testing è stato scelto un caso d'uso che includesse l'interfacciamento con la parte di database relativa all'applicazione web, invece nel capitolo successivo viene approfondita la realizzazione della Dashboard e quindi include l'interfacciamento con la parte di database relativa ai dati dei progetti che ospita l'applicazione web.

L'obiettivo del caso d'uso testato è quello di dare una visione generale dei progetti di cui l'utente autenticato è proprietario e avere accesso alle schede specifiche dei singoli progetti. Inoltre, per ogni scheda, gli utenti hanno a disposizione il file riassuntivo dei dati relativi al progetto o possono accedere alla dashboard per un resoconto generale del progetto, di cui si parlerà nel capitolo successivo.

Il tutto per rendere più funzionale e veloce la visione delle informazioni progettuali, motivo per cui è stata realizzata un'applicazione web dedicata.



# Capitolo 3

## Applicazione pratica della piattaforma

### 3.1 Progetto TCGA

Nella fase di *Analisi dei fabbisogni*, sono stati raccolti dati provenienti dal NCI<sup>1</sup>, utilizzati come caso di esempio pratico nello sviluppo dell'applicazione web.

In particolare il NIH<sup>2</sup> ospita uno dei tanti progetti inerenti alla genomica del cancro, tra cui il TCGA<sup>3</sup>.

Il TCGA ha lo scopo di catalogare le alterazioni genetiche responsabili di tanti tumori, in particolare sono stati analizzati diversi campioni tumorali associati ai tessuti sani per un confronto genetico e molecolare.

I dati sono resi disponibili pubblicamente per supportare i ricercatori e contribuire al miglioramento delle tecniche di diagnosi, trattamento e prevenzione del cancro [28].

In questa fase, sono stati selezionati dataset relativi allo stesso progetto ma di distretti corporei o tessuto d'origine differenti oppure relativi a progetti con diagnosi primarie differenti.

---

<sup>1</sup>L'acronimo NCI sta per National Cancer Institute. È uno dei più antichi istituti della National Institutes of Health o NIH che sostiene la ricerca del cancro.

<sup>2</sup>L'acronimo NIH sta per National Institute of Health ed è l'agenzia federale statunitense responsabile della ricerca biomedica e sanitaria.

<sup>3</sup>L'acronimo TCGA sta per The Cancer Genome Atlas Program, è un progetto di ricerca relativo alle alterazioni genetiche nei tumori umani, con l'obiettivo di migliorare la comprensione del cancro e favorire lo sviluppo di trattamenti mirati.



Per comprendere meglio come relazionare i dati, come passaggio fondamentale per la progettazione del database, si è pensato di dividere i dati presenti nei dataset in macro categorie:

- dati demografici;
- diagnosi;
- trattamenti.

Successivamente, è stato necessario decodificare alcuni codici alfanumerici appartenenti al sistema ICD-10<sup>4</sup> e al sistema TNM<sup>5</sup>. Contemporaneamente è stata necessaria la consultazione del dizionario dei dati, reso disponibile dal GDC o Genomic Data Commons, una piattaforma per la condivisione di dati genomici e clinici.

Il dizionario ha lo scopo di aiutare i ricercatori a comprendere il significato e l'uso di ogni variabile, parametro o campo nei dataset [29], in modo da fornire un'interpretazione univoca.

Nello specifico, nella tabella in Figura 3.1 sono riportati alcuni codici numerici e alfanumerici presenti nei dataset, usati come esempi nello sviluppo dell'applicazione web. Queste variabili sono utili per entrare nel merito del progetto e comprendere l'utilità della dashboard realizzata.

Standard AJCC	Stadio T	Stadio N	Stadio M	Morfologia	Classificazione ICD
I	T1	N1	M0	8140/3	C18.9
II	T2	N1b	M1	8211/3	C19
IIA	T3	N1c	MX	8255/3	C20
IIB	T4	N2		8263/3	C80.1
IIC	T4a	N2a		8480/3	C49.4
III	T4b	N2b		8020/3	
IIIA		NX		8310/3	
IIIB				8380/3	
IIIC				8382/3	
IV					
IVA					

Figura 3.1: Parametri di classificazione dei tumori

---

<sup>4</sup>L'acronimo IDC10 sta per International Classification of Disease ed è un sistema di codifica internazionale gestito dal World Health Organization

<sup>5</sup>L'acronimo TNM sta per Tumor-Nodes-Metastasis ed è un sistema di classificazione dello stadio del cancro.

In particolare, analizzando questi parametri lo standard AJCC (American Joint Committee on Cancer) classifica le stadiazioni per i tumori dai più localizzati ai più avanzati.

Nello standard TNM, ci sono tre principali categorie:

- la categoria T in cui sono classificati i tumori sulla base dello spessore e dell'estensione nei tessuti circostanti, definiti dai numeri subito dopo la T e c'è l'aggiunta di lettere a seconda della lacerazione presente;
- la categoria N in cui si analizza se il tumore si è diffuso nei linfonodi, il numero definisce i linfonodi nei quali si è diffuso il tumore e c'è l'aggiunta di lettere per fornire ulteriori dettagli sulla diffusione ai linfonodi;
- la categoria M in cui sono classificate le metastasi e dove eventualmente sono diffuse[30];
- per la categoria Morfologia è stato necessario consultare il sistema ICD-O-3 [31], in generale il primo numero indica la categoria morfologica del tumore, il secondo numero la malignità tumorale.

## 3.2 Form di caricamento dati

Nella modellizzazione delle specifiche è stato previsto il caso d'uso 'Compilazione del form' per consentire un approccio prospettico all'utente mediante l'inserimento di un singolo dato da importare nel database, con l'obiettivo di rendere più semplice il popolamento da parte dell'utente.

Questo caso d'uso è stato solamente progettato e non sviluppato, perciò le interfacce sono state pensate ad hoc per l'esempio del progetto usato nella fase di sviluppo, con campi specifici. Le bozze sono mostrate in Figura 3.2 e l'activity diagram del caso d'uso è riportato in Figura 7 nell'Appendice A dell'elaborato di tesi.

The image shows two screenshots of a web application interface for data entry. The first screenshot, titled "form inserimento", displays a form for entering patient data. It includes a dropdown menu for "Seleziona un database" (currently showing "gestione\_dati"), a "Logout" button, and a "CARICA FILE" button. The form is divided into sections: "Dati anagrafici" (Anabolic data) with fields for "Eta alla diagnosi", "Anno di nascita", "Etnia", "Razza", "Genere" (M/F), "Malignita precedente" (Si/No), "Sigarette giornaliere", "Stato vitale" (A/N A), and "Anno morte"; "Diagnosi" (Diagnosis) with fields for "Anno diagnosi", "Data Follow up", "Sito Incepis", and "Mor-Fologia"; and "Trattamento o Terapia" (Treatment or Therapy) with a "Spunta la casella" checkbox and a "Tipo" dropdown. The second screenshot, titled "form inserimento 2", displays a form for entering tumor data. It includes fields for "Tipo", "Durata (giorni)", "Tessuto o Organo", "Stadio tumorale" (Stadio M, Stadio T, Stadio N, Standard AJCC, Stadio avanzamento), and "Classificazione ICD". It also has a "Spunta la casella" checkbox and a "Tipo" dropdown. Both screenshots feature "Indietro" and "Avanti" buttons at the bottom.

Figura 3.2: Interfaccia utente del caso d'uso *Compilazione form*

In particolare, l'importazione dei dati è accessibile dall'interfaccia principale relativa al caso d'uso di 'Visualizzazione lista database'. La scelta di dare accesso a questa funzionalità dalla sezione dei database nasce dalla logica intuitiva che associa immediatamente l'importazione dei dati alla loro gestione all'interno di un database. In questo modo si rende l'operazione più naturale e diretta, semplificando il flusso di lavoro.

Inoltre, per lasciare invariata la scelta delle tre azioni principali associate anche alla sezione relativa ai progetti, è stato scelto di aggiungere l'opzione di importazione separatamente alle altre azioni disponibili.

La bozza di interfaccia consente innanzitutto il settaggio del database nel quale importare i dati e offre la possibilità di caricare un file per importare più dati contemporaneamente. L'interfaccia che ne consegue è mostrata in Figura 8.

In alternativa al caricamento di un file c'è la compilazione di un form per importare i dati relativi ad un singolo paziente e la popup che ne consegue è riportata in Figura 9.

Scegliendo il database da popolare, di default viene mostrata l'interfaccia suddivisa in base alle tabelle e ai relativi campi, pensate sulla base del progetto e considerando i widget che Streamlit mette a disposizione.

Per i dati anagrafici dei pazienti sono stati previsti text o number input, pensando che la compilazione possa avvenire in concomitanza con una intervista ad

uno specifico paziente che si vuole faccia parte di un progetto clinico o estraendo informazioni da cartelle cliniche di diverso tipo.

Inoltre, in questa sezione è stata prevista l'aggiunta di radio button, che consentono all'utente di effettuare una scelta esclusiva tra le alternative proposte, principalmente per opzioni che si basano su due sole alternative come il genere, lo stato vitale o se presenti malignità precedenti a quella da registrare.

Per i campi elencati nella tabella in Figura 3.1, essendo relativi a sistemi di codifica sono stati scelti i select box a tendina in cui l'utente può selezionare tra diverse alternative disponibili, senza dover inserire una risposta manuale. Inoltre, è previsto un radio button per l'assunzione di una terapia e nel caso in cui venga spuntato è prevista un'ulteriore select box.

### 3.3 Dashboard

Tra i principali obiettivi del progetto di tesi c'è l'implementazione di un approccio retrospettivo, mediante lo sviluppo di una dashboard personalizzata in base agli specifici progetti. Per lo sviluppo dell'applicazione la dashboard è stata personalizzata utilizzando il progetto TCGA come esempio.

L'obiettivo è consentire agli utenti di visualizzare in maniera diretta una panoramica su diverse tipologie di dati.

La dashboard è accessibile da due diversi percorsi:

**Home page - Dashboard:** partendo dalla Home Page nella sezione relativa ai progetti nel menù a sinistra dell'applicazione web.

L'interfaccia della Home Page riportata in Figura 2.11 mostra il menù principale, invece l'interfaccia della dashboard relativa a questo percorso è mostrata in Figura 2.14;

**Visualizzazione Lista progetti - Apertura progetto - Dashboard:** visualizzando la scheda riassuntiva delle principali informazioni di uno specifico progetto, oltre a poter scaricare un file riassuntivo, c'è l'opzione di accedere alla dashboard relativa. Le interfacce che mostrano come accedere alla dashboard da questo percorso sono mostrate in Figura 2.16.

La differenza tra le interfacce per la visualizzazione della dashboard per i due percorsi sta nel dover cercare o meno il progetto del quale si vuole visualizzare i dati. Nel primo percorso è necessario inserire l'ID o il nome del progetto accedendo dalla home page, mentre nel secondo percorso non è richiesto, poiché ci si trova già nella scheda relativa al progetto.

Per la maggior parte dei grafici realizzati per la dashboard, sono stati scelti i mul-

tiselect come widget forniti da Streamlit, che consentono di scegliere alternative multiple tra quelle proposte e per ciascuna selezione effettuata, l'applicazione web interagisce dinamicamente con il database per recuperare e aggiornare i dati in base alle scelte fatte.

Per la creazione della dashboard sono state scelte le informazioni principali provenienti dai dati del progetto usato per lo sviluppo, sfruttando i dati che possono risultare molto utili per gli utenti a cui l'applicazione web è destinata. Per la creazione dei grafici interattivi sull'applicazione web è stata utilizzata la libreria Matplotlib, che consente di generare grafici e diagrammi personalizzati in Python. Nella creazione dei diagrammi si è scelto di mostrare le distribuzioni in percentuali ma anche di mostrare i numeri, che consentono di vedere l'entità effettiva delle osservazioni. Questo potrebbe rendere i dati più concreti e comprensibili, così da interpretare in maniera diretta l'ampiezza di ogni categoria e poiché le percentuali da sole potrebbero risultare poco significative se non contestualizzate rispetto al numero totale.

Il primo grafico è relativo ai tipi di tumori presenti tra i dati del progetto in questione. L'applicazione web mediante una query seleziona di default tutti i tipi di tumore dalla tabella *tumore* del database, mostrata in Figura 3.3, li raggruppa e li contegga al fine di calcolare le percentuali utili al diagramma a torta.

Questo grafico potrebbe agevolare l'utente nella scelta delle terapie più indicate ai tumori più frequenti.

	id_caso	tipo_tumore	durata	tessuto_organo
	001e0309-9c50-42b0-9e38-347883ee2cd3	Endometrioid adenocarcinoma, NOS	0	Endometrium
	0022478c-4dfd-4cbe-a05e-fb20310844e3	Clear cell adenocarcinoma, NOS	563	Kidney, NOS
	002724fa-7051-49fa-9c58-4bcb7eba4ac6	Serous cystadenocarcinoma, NOS	671	Endometrium
	0045349c-69d9-4306-a403-c9c1fa836644	Adenoid cystic carcinoma	0	Breast, NOS
	005a6517-2e5a-4ea3-ab36-531522723607	Serous cystadenocarcinoma, NOS	74	Ovary
	00630714-7ab3-44e1-afff-186300edae44	Serous cystadenocarcinoma, NOS	1156	Ovary

Figura 3.3: Esempio di dati provenienti dalla tabella *tumore*, in particolare l'attributo *tipo tumore* utile alla query

In Figura 3.4 è mostrata la dashboard relativa ai tipi di tumori, con le percentuali relative ai dati immagazzinati nel database. In particolare nella seconda immagine si può notare il menù a tendina che consente di inserire più opzioni contemporaneamente, grazie appunto al widget multiselect.

Quando è selezionata l'opzione predefinita 'all', il gran numero di tipi di tumori rende difficile associare un colore unico a ciascuno. Per questo motivo, alcuni tumori sono stati associati a colori già utilizzati. Il widget multiselect è stato

introdotto per permettere all'utente di selezionare un sottoinsieme di dati, riducendo così il numero di categorie visualizzate contemporaneamente e facilitando una visualizzazione più chiara delle differenze tra i dati scelti.

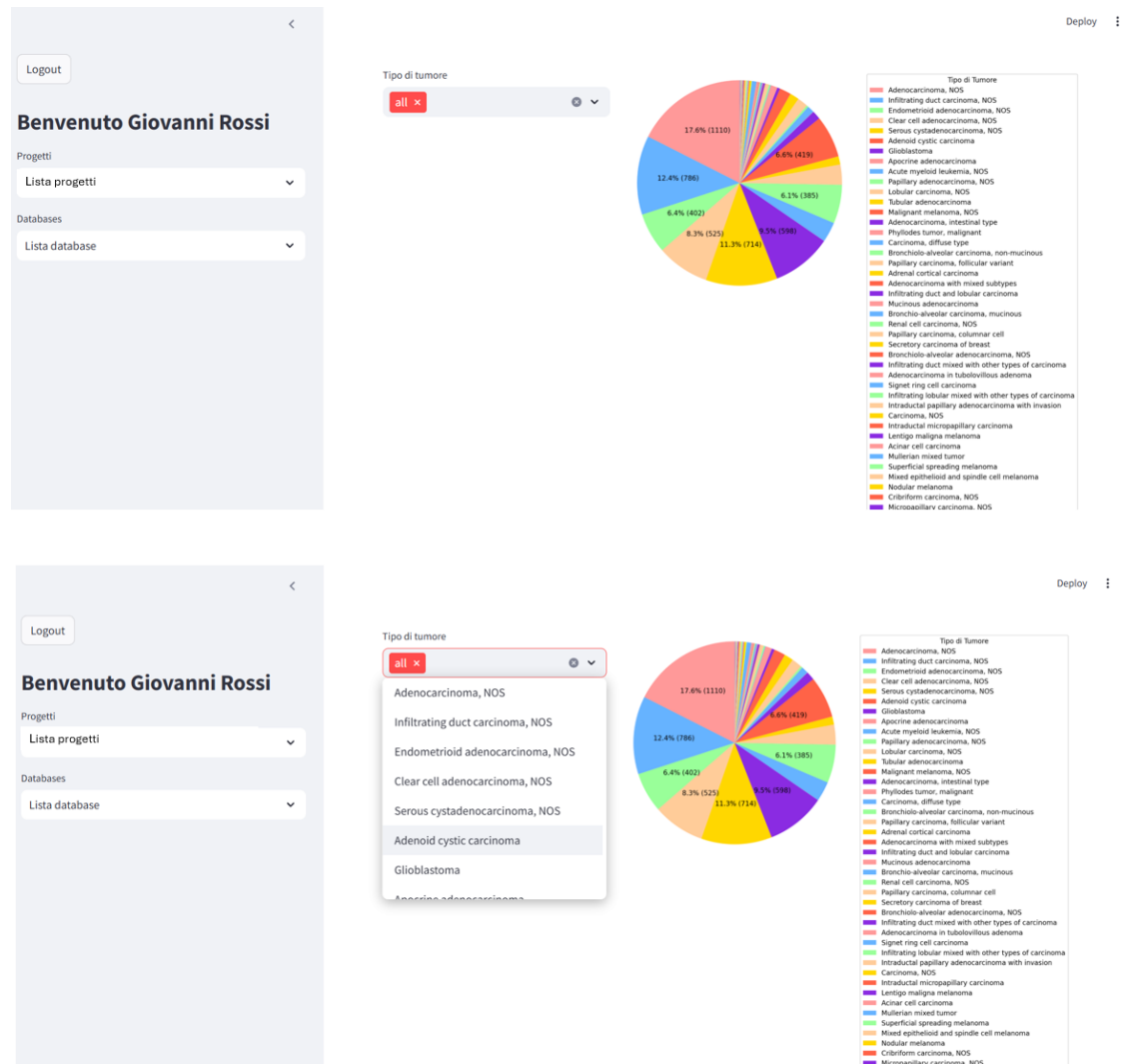


Figura 3.4: Dashboard dell'applicazione web relativa al tipo di tumore

Inoltre, si è pensato di creare un grafico per conoscere la distribuzione del rischio di tumore tra i due generi, relativamente ai dati dello specifico progetto.

L'applicazione web esegue una query per contare il numero di pazienti appartenenti ai due generi affetti da tumore, selezionando i dati relativi alla tabella *paziente* in Figura 3.5.

## Applicazione pratica della piattaforma

id_paziente	anno_nascita	etnia	razza	malignità_precedente	genere	stato_vitale	età
15d0d9d7-5e56-5802-908f-ad25112f7b06	1982	not hispanic or latino	black or african american	no	male	Alive	27
cbfec53d-d54c-50bc-8c35-a743e72a7d2a	1948	not hispanic or latino	white	no	female	Dead	57
de343f63-94d4-5792-9fb6-fed3833a6221	1939	not hispanic or latino	white	no	male	Dead	65
59b71222-4d3b-5103-8313-7d9d41e3ebb6	1955	not hispanic or latino	black or african american	no	female	Alive	51
fe1105ef-a4ae-5eb1-ae9a-efb1d46e40dd	1946	not hispanic or latino	white	no	male	Dead	60
1ee87765-cf4d-5069-bae9-594aaefcbe4	1925	not hispanic or latino	white	no	male	Dead	83

Figura 3.5: Esempio di dati provenienti dalla tabella *paziente*, in particolare l'attributo genere utile alla query

In Figura 3.6 è mostrata la dashboard relativa ai generi con il menù a tendina che consente di inserire entrambe le opzioni contemporaneamente o le due opzioni singolarmente, grazie al widget multiselect .

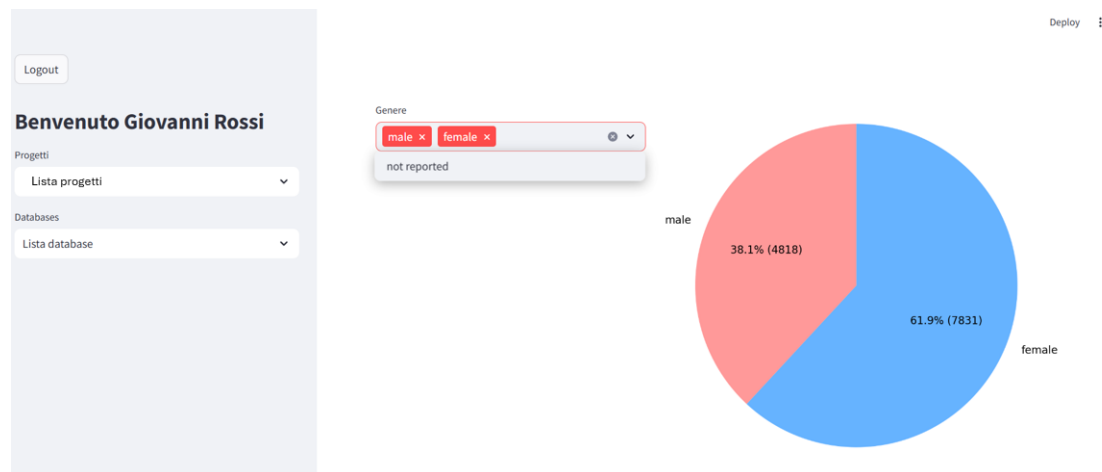


Figura 3.6: Dashboard dell'applicazione web relativa al genere

Per quanto riguarda le stadiazioni tumorali, la dashboard è stata costruita mediante grafico a barre considerando tutti gli standard elencati nella tabella riassuntiva in Figura 3.1.

A differenza degli altri grafici realizzati, in questo caso è stato utilizzato come widget un select box al posto del multiselect, in quanto permettere la selezione di più attributi contemporaneamente avrebbe reso il grafico confusionario e visivamente poco chiaro.

L'applicazione web esegue una query per selezionare dalla tabella *stadio tumorale* le varie stadiazioni, come mostrato in una parte della tabella *stadio tumorale* in Figura 3.7.

id_stadio	id_caso	stadio_m	stadio_t	stadio_n	standard_ajcc	stadio_avanzamento	classificazione_icd
2691	25131445-d3cf-46af-b3c5-ca1d1e3561bb	M1	T3a	NX	Stage IV	5th	C64.9
2692	265f1d03-fa56-4f9a-a1d2-33f123558729	M0	T1a	NX	Stage I	6th	C64.9
2693	267ff78b-bceb-466e-8582-c560fe227ff0	M1	T3a	N0	Stage IV	5th	C64.9
2694	26cc2c84-aaec-4cd1-9c45-dd08d372128d	M0	T3a	NX	Stage III	7th	C64.9
2695	270c9a0b-d71a-4d4e-8686-f183412159f2	M0	T3a	N1	Stage III	6th	C64.9
2696	2711b4a4-9e6f-4a12-add6-c10f72e97d3c	MX	T1a	NX	Stage I	7th	C64.9
2697	28011111-4a01-4cdc-8d6b-7223fb2c501b	MX	T2a	NX	Stage IV	7th	C64.9

Figura 3.7: Esempio di dati provenienti dalla tabella *stadio tumorale*

In Figura 3.8 è mostrata la dashboard relativa ai vari tipi di stadiazioni, senza aggiunta di nessuna spiegazione dei significati delle varie stadiazioni, presupponendo che l'utente a cui è destinata l'applicazione web sia informato a riguardo.

Nella prima immagine è mostrato il select box con le varie tipologie di stadiazioni selezionabili, invece nella seconda immagine è mostrato nello specifico il grafico a barre relativo allo stadio M del sistema TNM.

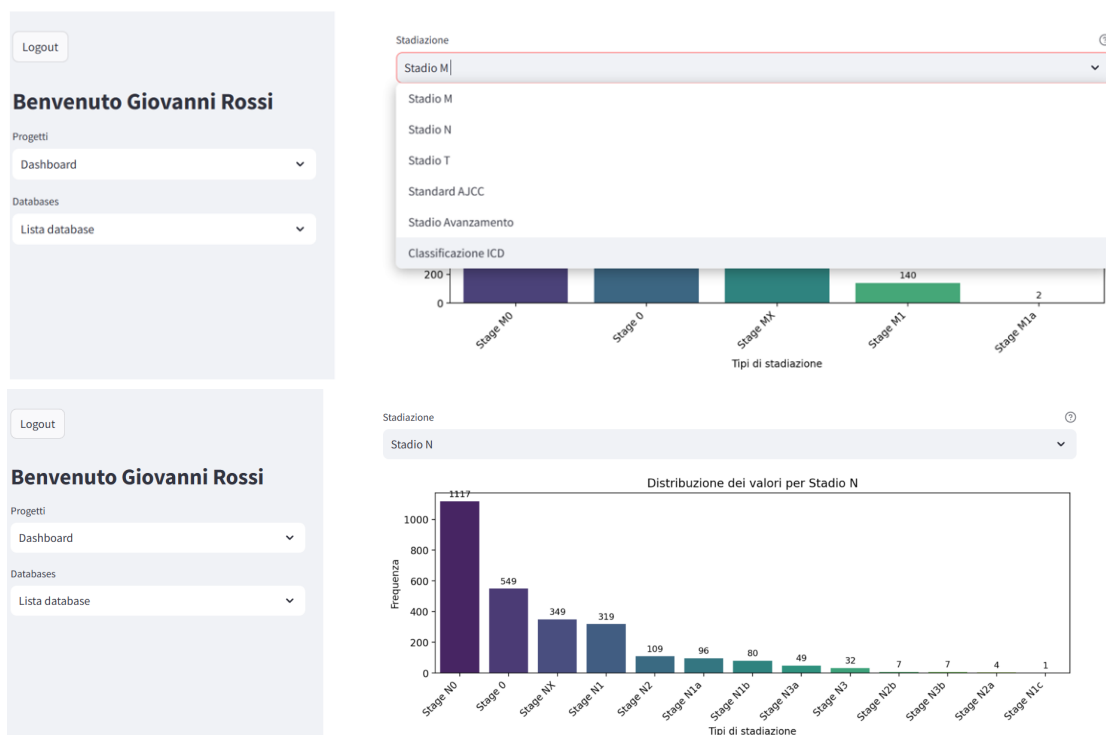


Figura 3.8: Dashboard dell'applicazione web relativa alle varie stadiazioni

La dashboard dell'applicazione web inoltre comprende un grafico a torta per mostrare all'utente le percentuali dei trattamenti somministrati ai pazienti dello specifico progetto in questione.

L'applicazione web mediante una query seleziona di default tutti i tipi di trattamento dalla tabella *trattamento* del database mostrata in Figura 3.9, li raggruppa



scegliendo solamente i casi in cui si sia deciso di somministrare il trattamento e li conteggia al fine di calcolare le percentuali utili al diagramma a torta.

id_trattamento	tipo_tratt	trattamento_o_terapia	id_diagnosi	id_paziente
53cc6bcd-34c3-5f1e-a51c-9faee225286f	Radiation Therapy, NOS	yes	e638aea5-f8a7-596d-b34...	d7591f2f-2ec7-5982-811c-ffd6398f62ee
5299e865-99b0-5f13-bdb8-6a94ce585918	Pharmaceutical Therapy, NOS	no	e7b56ec3-8d23-5b45-b8...	a47477fa-7bc0-54dd-bccb-0ae576a33...
7f032f19-ff0a-5680-a285-c8697020948d	Pharmaceutical Therapy, NOS	yes	44e41836-b8ff-57e6-938...	09ee1fbb-9a2d-588d-b9fc-36d013f76719
50c2f3ba-5d99-5aba-b303-af50434c5d2c	Radiation Therapy, NOS	no	c3c5473b-a3cc-588f-a71...	b8563d53-5515-53de-82e7-1db2d83cb...
a5d7a476-cfce-5ece-97c6-0eaac818f01d	Pharmaceutical Therapy, NOS	yes	1529a83e-2a66-56dd-b5...	3d3765a2-73f4-5aa7-948a-9edb569cccc9
b27ab3b0-2baa-59b6-b414-07bd25bbde5c	Radiation Therapy, NOS	yes	45ddd85c-041a-569c-ad3...	ee70946d-9ef6-50f3-8c47-8db91326f6b0
c781ad83-f022-516a-a9c3-9b182087c9ce	Pharmaceutical Therapy, NOS	yes	7c93752e-1375-56d3-8b...	5d65a8b2-9550-58c8-8346-b2054bdee...

Figura 3.9: Esempio di dati provenienti dalla tabella trattamento

In Figura 3.10 è mostrata la dashboard relativa ai vari tipi di trattamento.

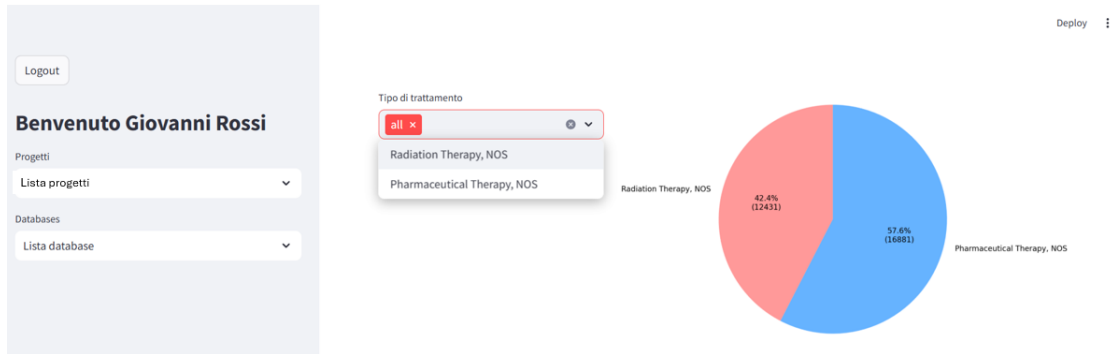


Figura 3.10: Dashboard dell'applicazione web relativa ai tipi di trattamento

Infine, la dashboard dell'applicazione web comprende un grafico a dispersione che aiuta l'utente a visualizzare in maniera immediata se c'è una maggiore concentrazione di pazienti in determinate fasce d'età nel momento in cui viene diagnosticato il tumore.

L'applicazione web mediante una query seleziona di default tutte le fasce d'età dalla tabella *paziente* del database mostrata in Figura 3.5, li raggruppa e li conteggia al fine di calcolare le percentuali utili al grafico a dispersione.

In Figura 3.11 è mostrata la dashboard relativa all'età alla diagnosi dei pazienti.

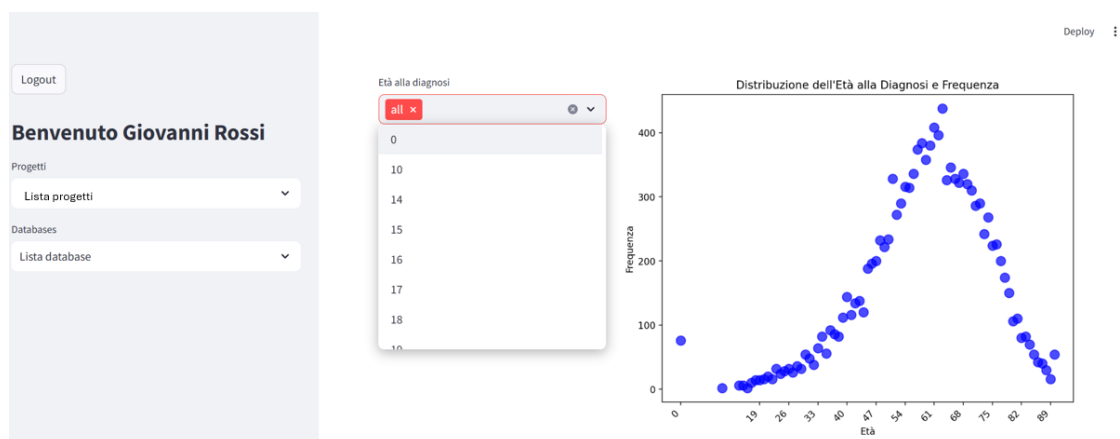


Figura 3.11: Dashboard dell'applicazione web relativa all'età alla diagnosi



# Conclusioni e sviluppi futuri

Il presente lavoro di tesi ha permesso di sviluppare una soluzione innovativa per la gestione dei dati clinici, rispondendo ai requisiti richiesti dall'azienda Tecnologie avanzate e puntando alla centralizzazione e standardizzazione dei dati appartenenti a diversi progetti in ambito di ricerca.

Il database relazionale progettato e sviluppato, insieme all'applicazione web interattiva, costituiscono la base per una gestione più efficiente e centralizzata dei dati.

L'utilizzo di MySQL come DBMS ha garantito scalabilità, progettando funzionalità che consentano di ampliare l'applicazione web con nuovi progetti di ricerca.

Tra i principali obiettivi posti inizialmente, l'adozione di un approccio retrospettivo per una visualizzazione dei dati disponibili è stata portata a termine così come era stata progettata, invece l'approccio prospettico per la raccolta di nuovi dati è stato progettato ma non ancora sviluppato.

La fase di testing ha confermato che lo sviluppo di un percorso di un caso d'uso sviluppato rispecchiasse i requisiti prefissati durante la progettazione, verificando che le funzionalità fossero correttamente implementate.

Questo progetto ha posto le basi per sviluppi futuri, in particolare per il supporto alle attività di ricerca che potrebbero includere l'integrazione di nuove funzionalità e l'espansione dell'applicazione web mediante adozione di tecniche di intelligenza artificiale per l'analisi predittiva dei dati clinici con l'obiettivo di agevolare il lavoro dei clinici, fornendo supporto nella diagnosi e di migliorare la qualità delle terapie somministrate in ambito clinico.

In particolare, analizzando i limiti relativi all'applicazione web, è necessario:

- gestire la parte di creazione del database, generando degli errori in caso di utilizzo di uno stesso nome di tabella per mantenere l'integrità e l'univocità del database, per evitare conflitti nell'accesso e nel recupero dei dati o nell'inserimento di nuovi;
- espandere il sistema per la gestione di grandi volumi di dati;
- integrare l'applicazione web sviluppata con altri sistemi clinici, come le cartelle cliniche per rendere l'applicazione web più versatile;

- migliorare l'applicazione web, ampliando le informazioni nella dashboard che l'utente può visualizzare, relativamente all'esempio di progetto usato nella fase di sviluppo.  
Per esempio, aggiungendo i distretti corporei coinvolti nel tumore in entrambi i sessi;
- rendere l'applicazione web il più possibile intuitiva e accessibile per l'importazione dei dati, sia tramite caricamento di un file, sia tramite compilazione di un eCRF, permettendo l'importazione direttamente dal menù dell'applicazione web;
- gestire la parte relativa alla compilazione di un form, offrendo all'utente la possibilità di configurare liberamente la struttura del form di inserimento dati, così come è stata gestita la creazione della struttura del database.

In conclusione, il progetto rappresenta un passo importante nell'evoluzione delle tecnologie digitali nell'ambito dell'healthcare, contribuendo a migliorare la gestione dei dati clinici e la qualità delle attività di ricerca.



# Appendice A

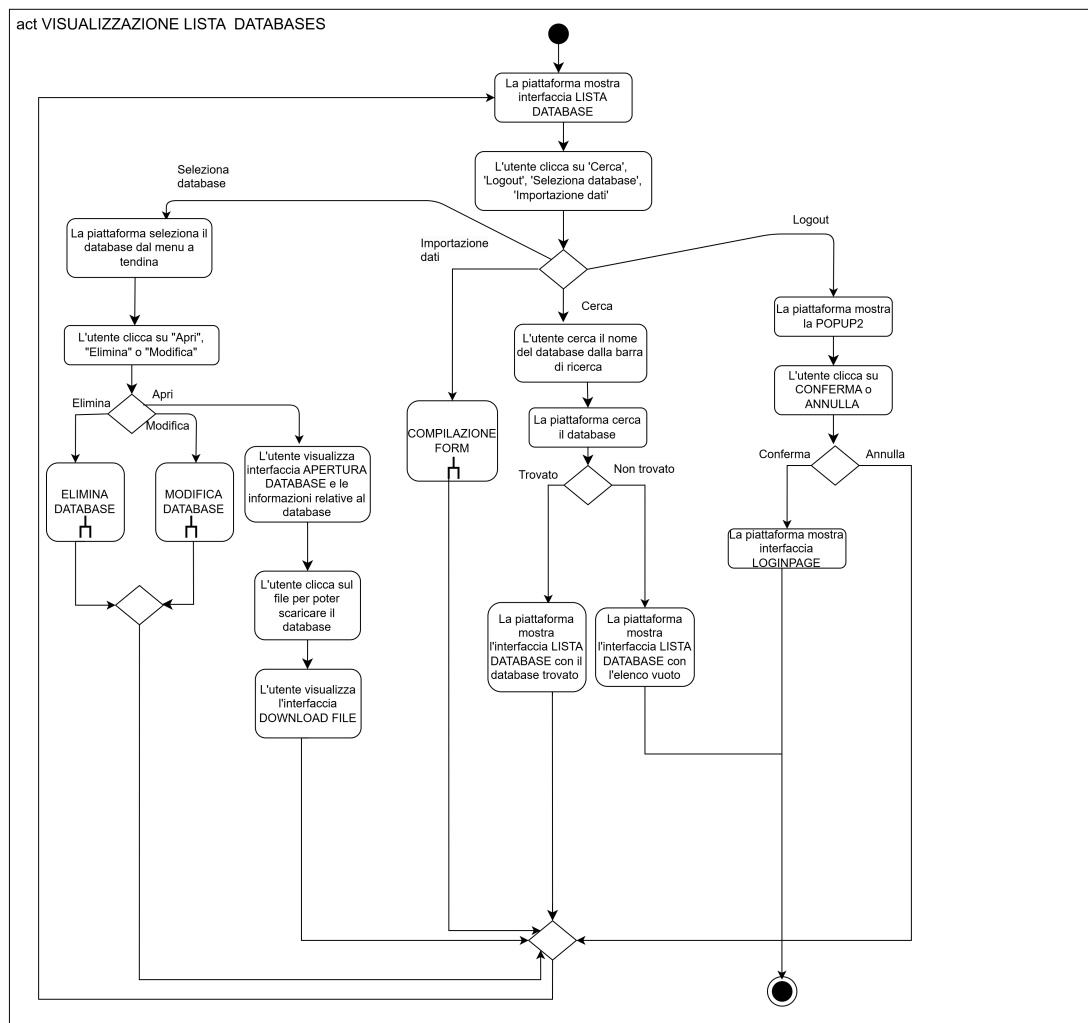


Figura 1: Activity diagram di *Visualizzazione lista databases*

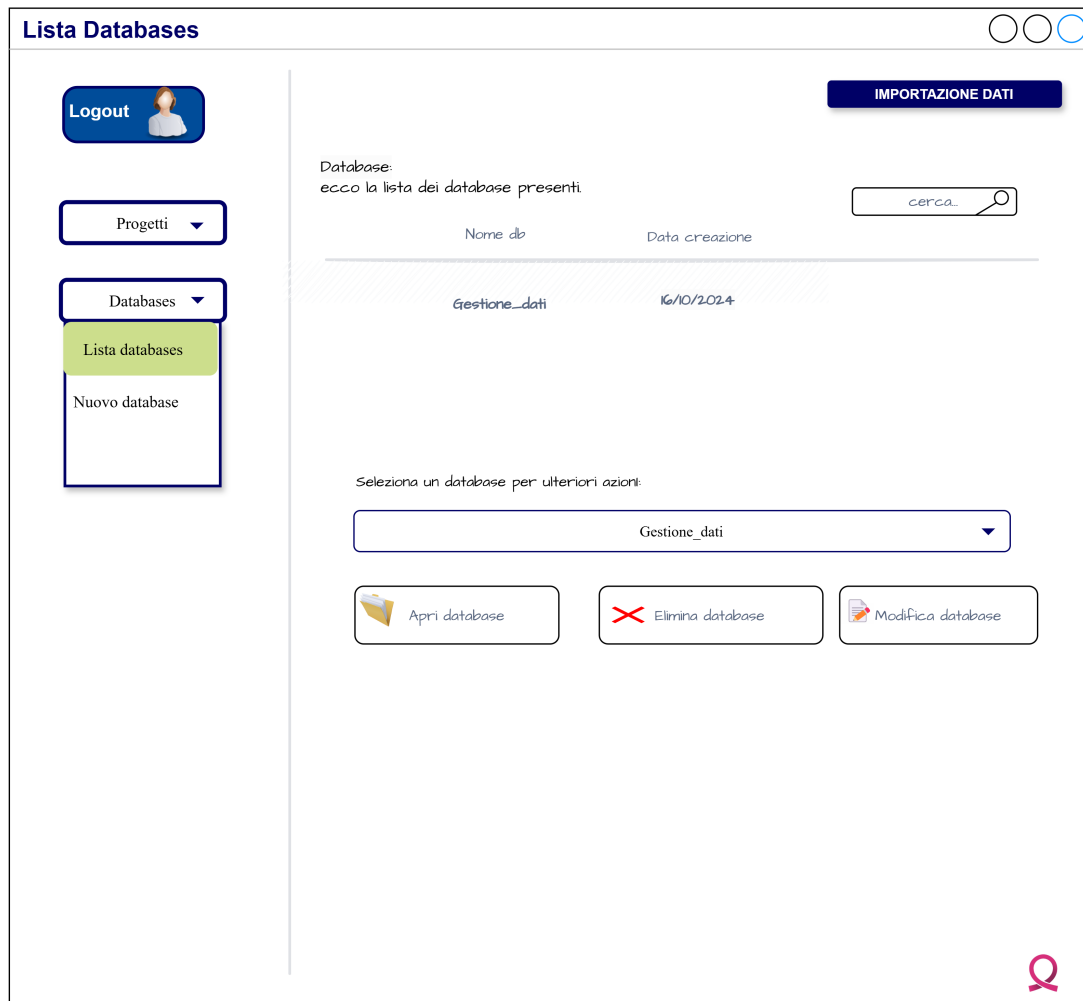


Figura 2: Interfaccia utente di *Visualizzazione lista databases*



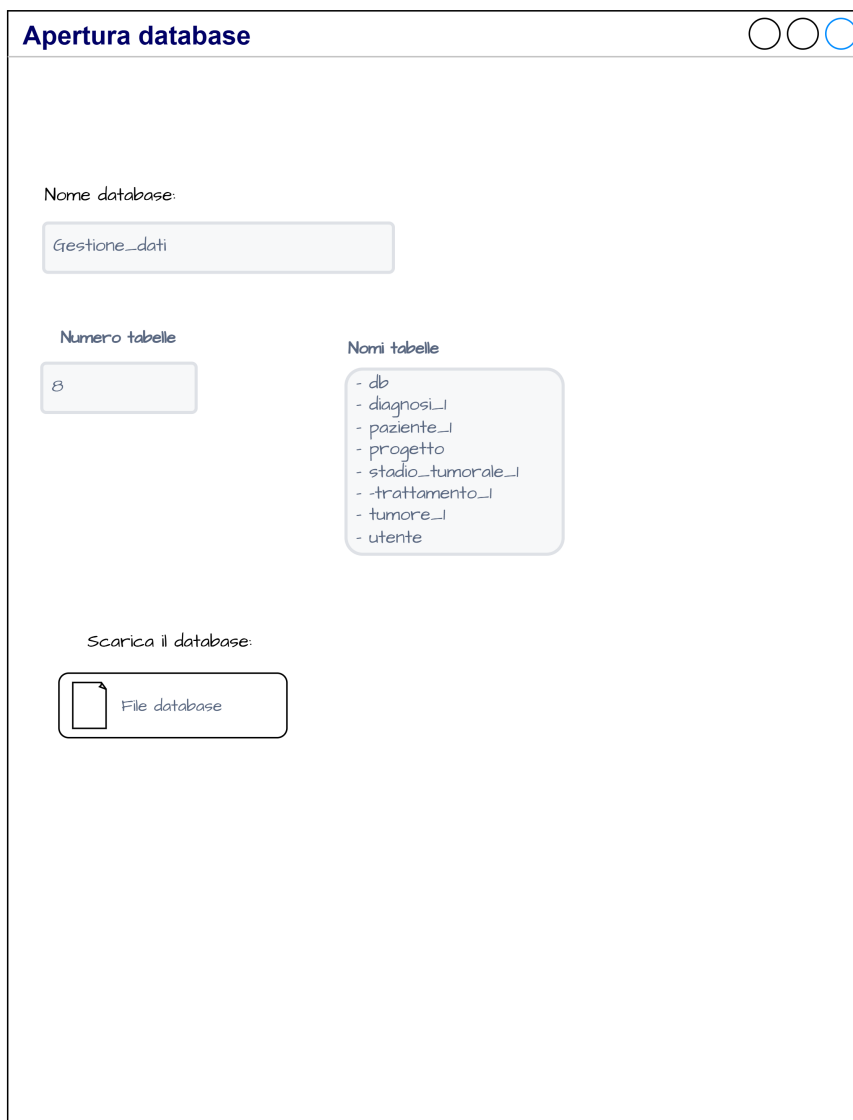


Figura 3: Interfaccia utente proveniente dall'apertura della scheda database del caso d'uso *Visualizzazione lista databases*

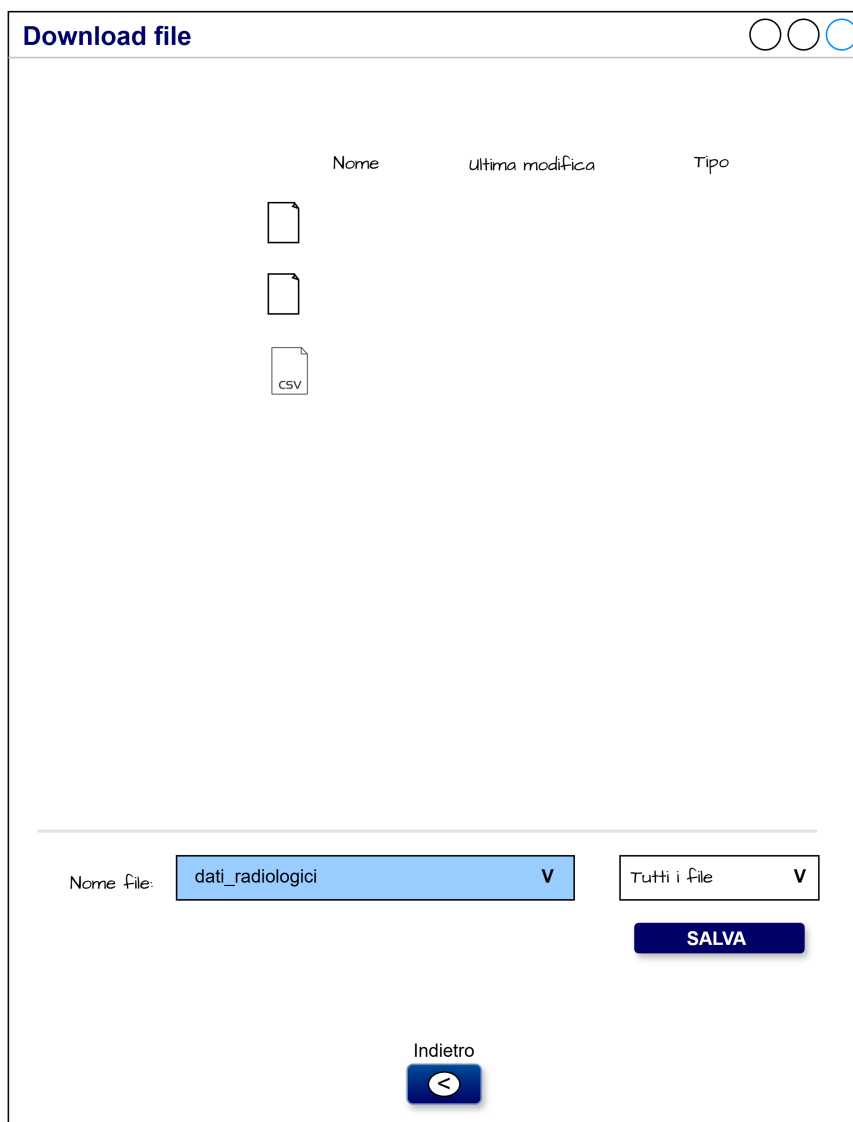


Figura 4: Interfaccia utente proveniente dal download del database proveniente dall'apertura della scheda database

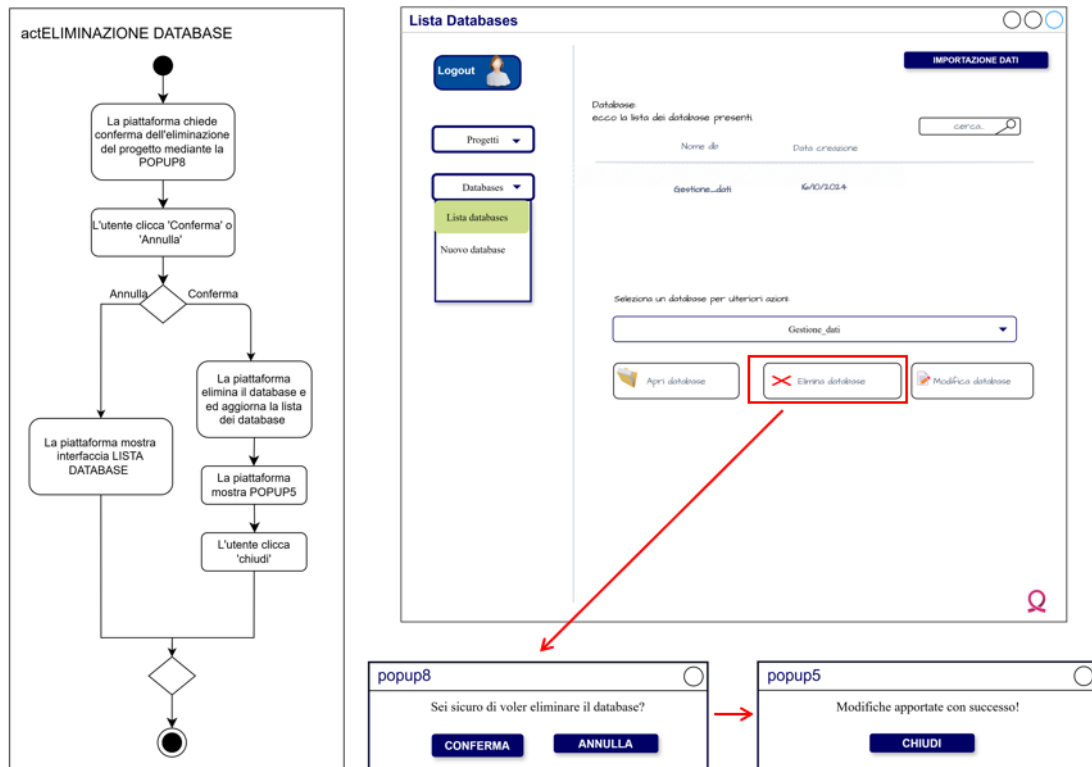


Figura 5: Activity diagram di *Eliminazione databases*, interfaccia di *Visualizzazione lista databases* con le popup relative al caso d'uso *Eliminazione databases*

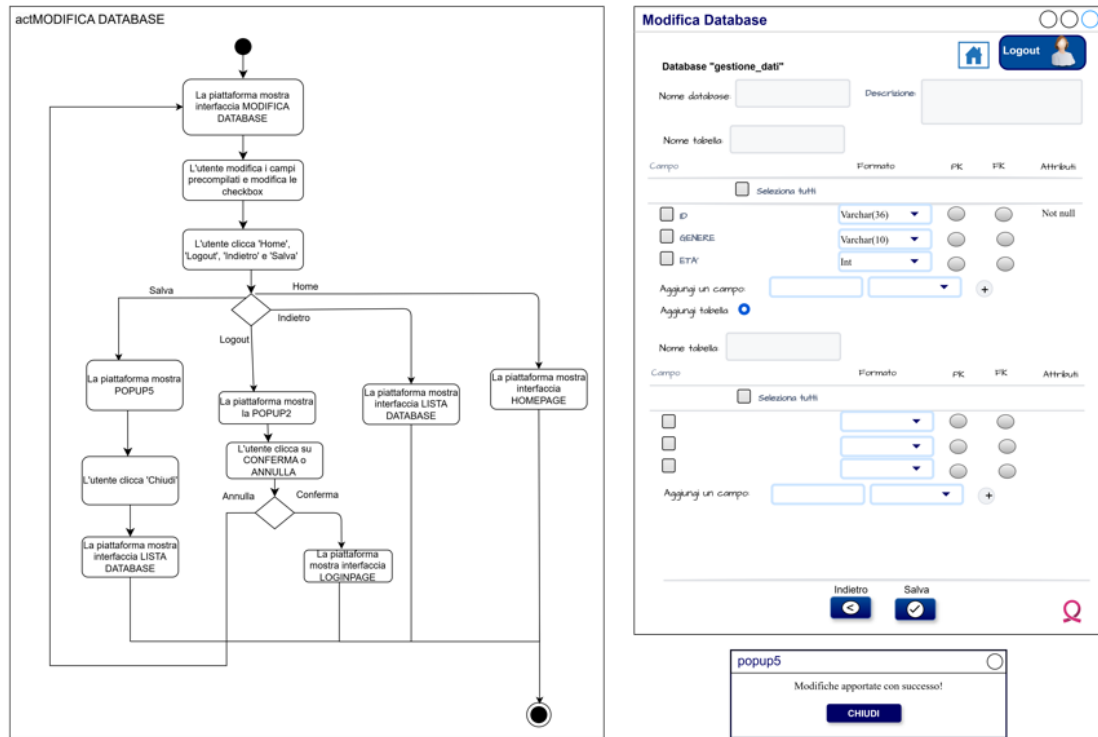


Figura 6: Activity diagram e interfaccia utente di *Modifica database* con relativa popup

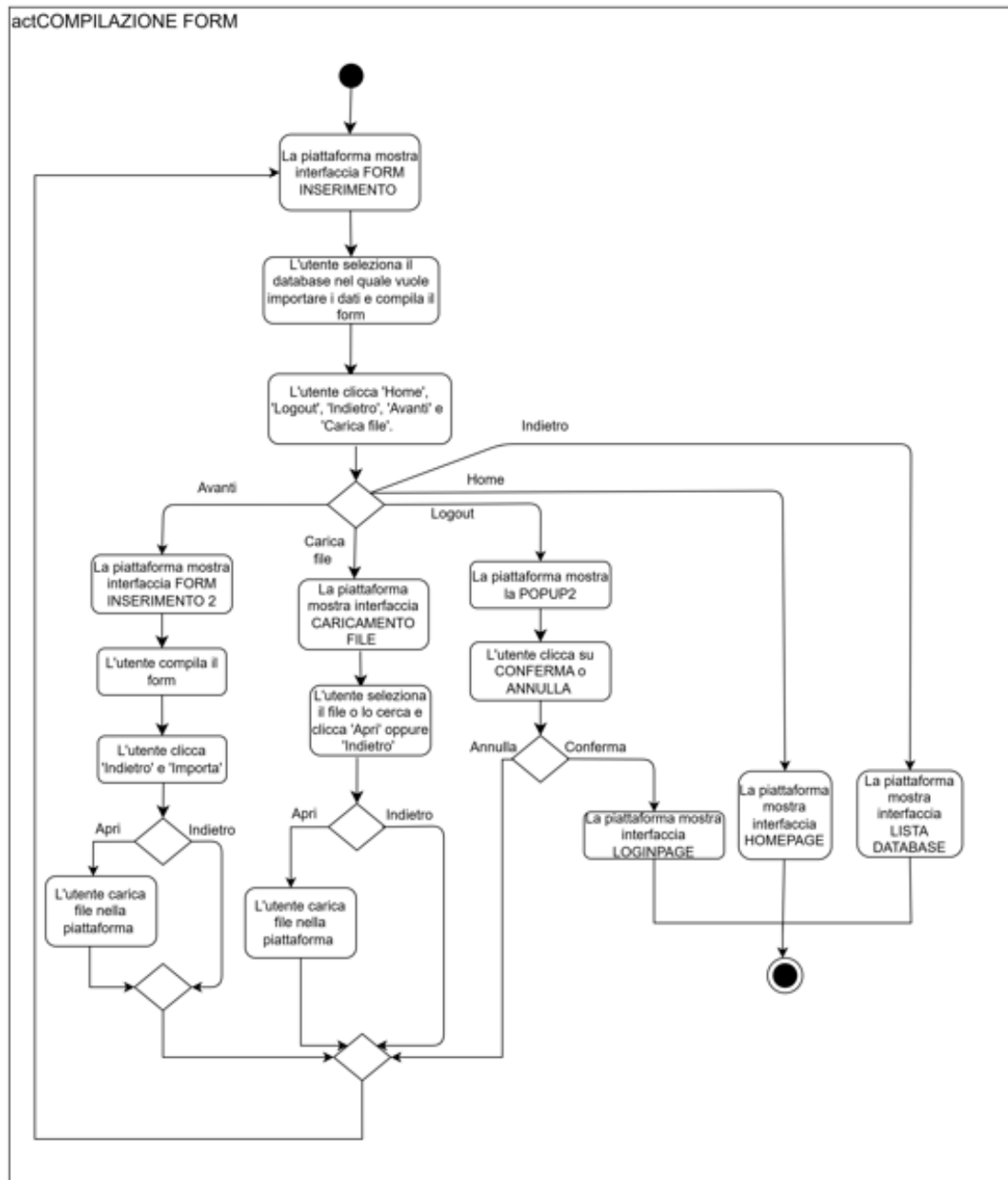


Figura 7: Activity diagram di *Compilazione form*

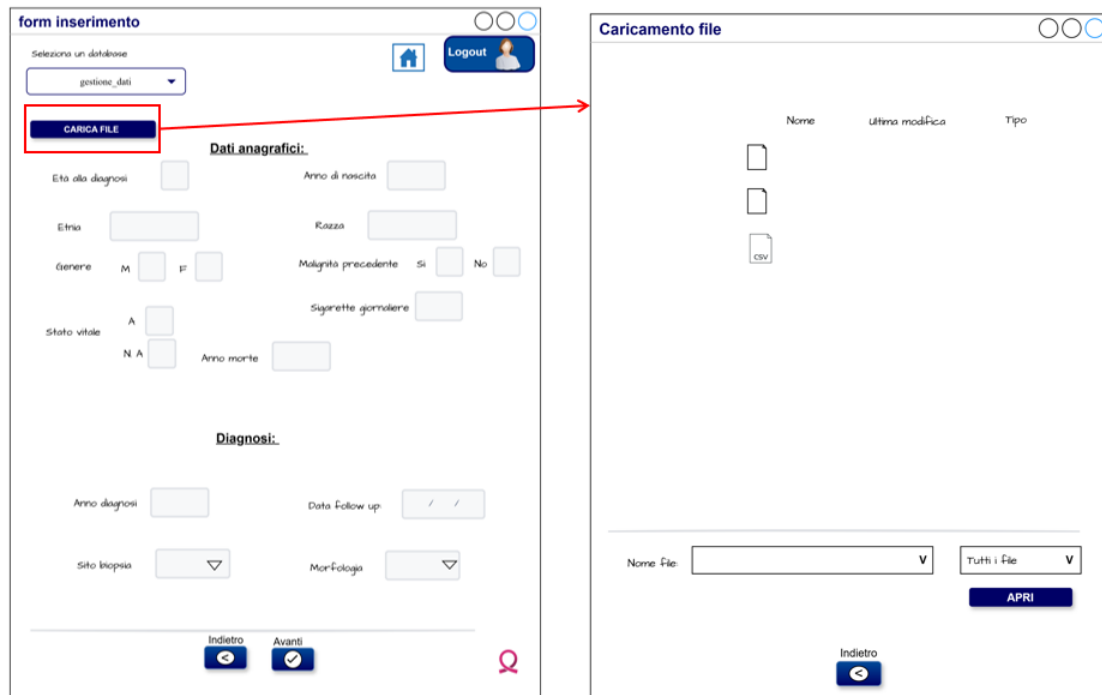


Figura 8: Interfacce utente di *Compilazione form*

The image shows a web application interface for data entry. The main window is titled "form inserimento 2" and contains the following sections:

- Tumore:** Includes input fields for "Tipo", "Durata (giorni)", and "Tessuto o Origine".
- Stadio tumorale:** Includes dropdown menus for "Stadio M", "Stadio T", "Stadio N", "Standard AJCC", "Stadio avanzamento", and "Classificazione ICD".
- Trattamento o Terapia:** Includes a "Spunta la casella" checkbox and a "Tipo" dropdown menu.

At the bottom of the form, there are two buttons: "Indietro" (Back) and "Importa" (Import). The "Importa" button is highlighted with a red box. A red arrow points from this button to a separate dialog box titled "popup8".

The "popup8" dialog box contains the message "Dati correttamente importati nel database!" and a "CHIUDI" (Close) button.

Figura 9: Interfaccia utente per la seconda parte dello use case *Compilazione form*, con relativa popup

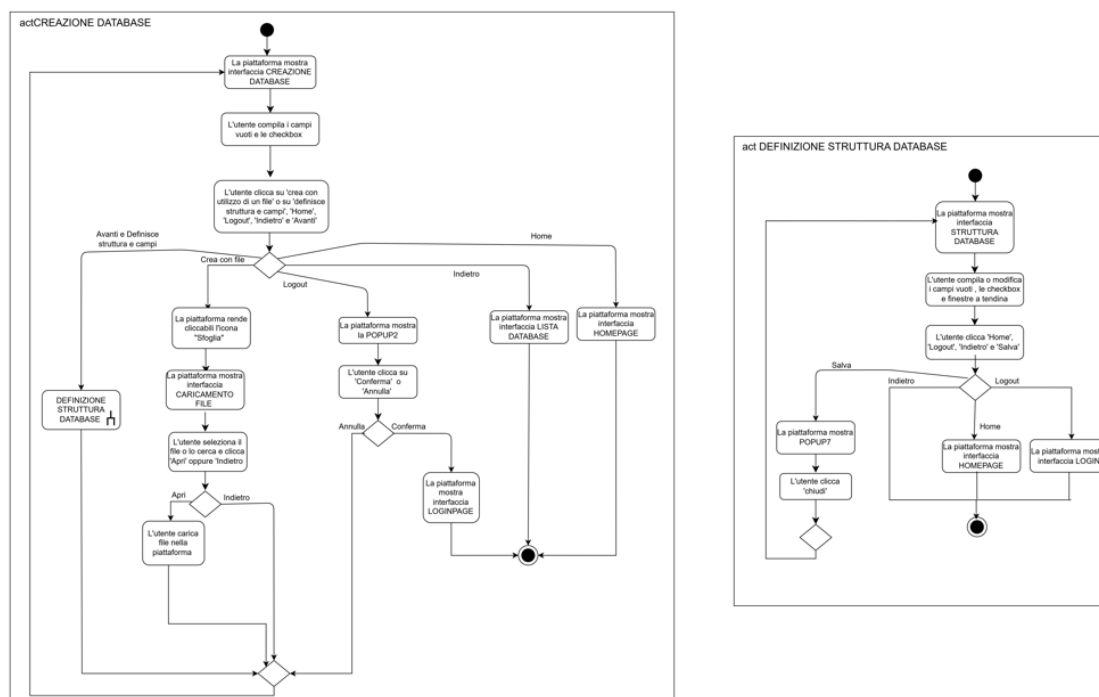


Figura 10: Activity diagram di *Creazione database* e *Definizione struttura database*



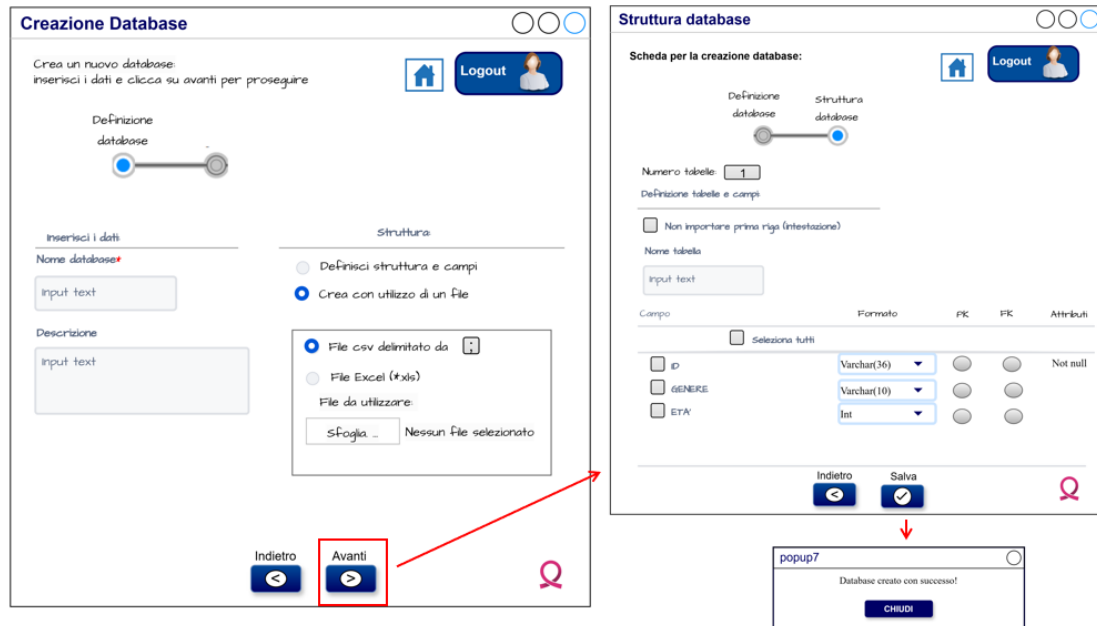


Figura 11: Interfacce utente di *Creazione database* e *Definizione struttura database* con utilizzo di un file e relativa popup.

È utilizzata anche l'interfaccia di *Caricamento file*, presente in Figura 8

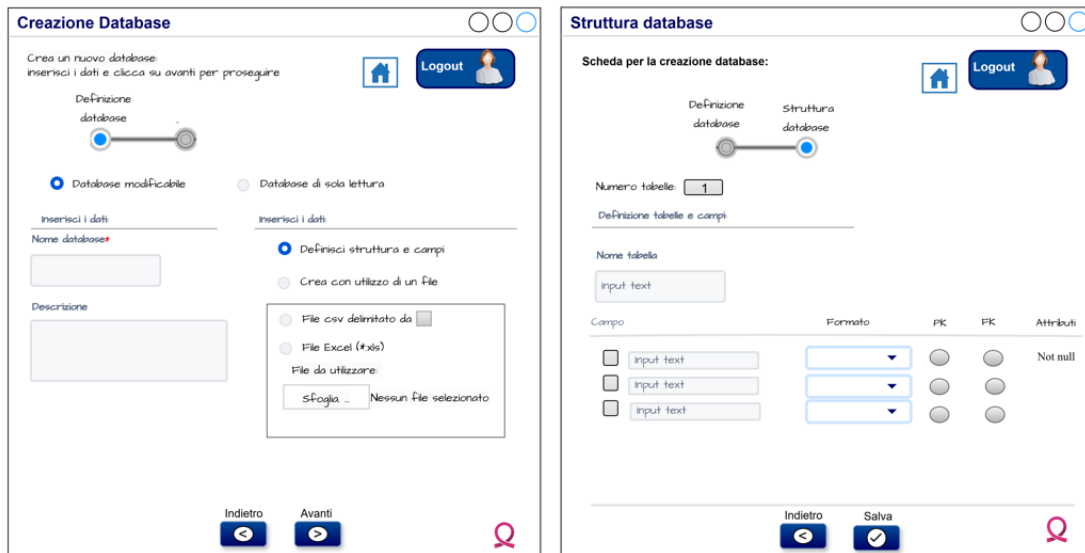


Figura 12: Interfacce utente di *Creazione database* e *Definizione struttura database* con definizione manuale della struttura



# Bibliografia

- [1] Kais Tahar, Christoph Müller, Andreas Dürschmid, Silke Haferkamp, Kutaiba Saleh, Patrick Jürs, Sebastian Stäubert, Jan Erik Gewehr, Sven Zenker, Danny Ammon, and Thomas Wendt. Integrating heterogeneous data sources for cross-institutional data sharing: Requirements elicitation and management in SMITH. *Stud. Health Technol. Inform.*, 264:1785–1786, August 2019.
- [2] PubMed. <https://pubmed.ncbi.nlm.nih.gov>.
- [3] Benjamin Kinast, Hannes Ulrich, Björn Bergh, and Björn Schreiweis. Functional requirements for medical data integration into knowledge management environments: Requirements elicitation approach based on systematic literature analysis. *J. Med. Internet Res.*, 25:e41344, February 2023.
- [4] Jakob Vielhauer, Ujjwal Mukund Mahajan, Kristina Adorjan, Christopher Benesch, Bettina Oehrle, Georg Beyer, Simon Sirtl, Anna-Lena Johlke, Julian Allgeier, Anna Pernpruner, Johanna Erber, Parichehr Shamsrizi, Christian Schulz, Fady Albashiti, Ludwig Christian Hinske, Julia Mayerle, and Hans Christian Stubbe. Electronic data capture in resource-limited settings using the lightweight clinical data acquisition and recording system. *Scientific Reports*, 14(1):1–10, 2024. Received: 13 June 2024; Accepted: 6 August 2024; Published online: 17 August 2024.
- [5] Market Research Intellect. Revolutionizing data collection: The power of edc (electronic data capture) systems, 2024. Accessed: 2024-02-03.
- [6] Caleb J. Ruth, Samantha Lee Huey, Jesse T. Krisher, Amy Fothergill, Bryan M. Gannon, Camille Elyse Jones, Elizabeth Centeno-Tablante, Laura S. Hackl, Susannah Colt, Julia Leigh Finkelstein, and Saurabh Mehta. An electronic data capture framework (connedct) for global and public health research: Design and implementation. *Journal of Medical Internet Research*, 22(8):e18580, 2020.
- [7] G2. Electronic case report forms (ecrf): A complete guide, 2023. Accessed: 2025-02-07.

- [8] Balestra G. Construction process: software development models. Materiale didattico, 2022/23. Politecnico di Torino, Bio-Lab, DET - Dipartimento di Elettronica e Telecomunicazioni.
- [9] Balestra G. Software development. Materiale didattico, 2022/23. Politecnico di Torino, Bio-Lab, DET - Dipartimento di Elettronica e Telecomunicazioni.
- [10] Wikipedia. Modello a cascata, 2025. Accesso: 12 febbraio 2025.
- [11] IONOS Startup Guide. Modello a spirale: come funziona e quando conviene utilizzarlo, 2023. Accesso il 12 febbraio 2025.
- [12] IONOS. Il v-model: un modello per lo sviluppo del software, 2024. Ultimo accesso: 13 febbraio 2025.
- [13] Balestra G. Ciclo di vita del software - processo di sviluppo. Materiale didattico, 2022/23. Politecnico di Torino, Bio-Lab, DET - Dipartimento di Elettronica e Telecomunicazioni.
- [14] Balestra G. Construction process: biomedical data storage. Materiale didattico, 2022/23. Politecnico di Torino, Bio-Lab, DET - Dipartimento di Elettronica e Telecomunicazioni.
- [15] Aulab. Cos'è un database? guida introduttiva, 2025.
- [16] Andrea Minini. Progettazione di una base di dati, 2025. 2025-01-26.
- [17] Docsity. Costruzione e progettazione di un database locale, 2025. Accessed: 2025-01-27.
- [18] IONOS Digital Guide. Database: Definizione e tipologie, 2024. Ultimo accesso: 27 gennaio 2025.
- [19] Vito Lavecchia. Differenza tra modello relazionale, gerarchico e reticolare di un database, 2024. Accessed: 2024-01-27.
- [20] Il Prof d'Informatica. Organizzazione dei database, 2025. Ultimo accesso: 27 gennaio 2025.
- [21] LearnSQL.it. Che cos'è un dbms?, 2024. Accessed: 2025-01-29.
- [22] GeekandJob. Database, 2024. Accessed: 2025-01-29.
- [23] G. Lorenzi and M. Cavalli. *Progettazione dei database, linguaggio SQL, dati in rete*. Atlas, 2025. Parte prima: Progettazione della base di dati, Capitolo 1: Organizzazione degli archivi e basi di dati.

- [24] IONOS Digital Guide. Comandi sql: panoramica e spiegazione dei comandi più importanti, 2024. Accesso: 30 gennaio 2025.
- [25] Balestra G. Requirements analysis:uml. Materiale didattico, 2022/23. Politecnico di Torino, Bio-Lab, DET - Dipartimento di Elettronica e Telecomunicazioni.
- [26] Baralis E. Cerquitelli T. Applicazioni web - introduzione a streamlit. Materiale didattico, 2023/24. Politecnico di Torino, DAUIN - Dipartimento di Automatica e Informatica.
- [27] Balestra G. Testing process. Materiale didattico, 2022/23. Politecnico di Torino, Bio-Lab, DET - Dipartimento di Elettronica e Telecomunicazioni.
- [28] National Cancer Institute. The cancer genome atlas (tcga), 2025.
- [29] Genomic Data Commons. Gdc data dictionary viewer, 2025.
- [30] Melanoma Italia. Determinare lo stadio del melanoma, 2025.
- [31] Registro Tumori. Icd-o-3: Classificazione dei tumori in base alla morfologia e alla sede anatomica, 2005.