APPLICAZIONE SISTEMA VACCINI

Documentazione DB

Università degli Studi dell'Insubria – Laurea Triennale in Informatica

Progetto Laboratorio B: Applicazione per prenotazione vaccini

Sviluppato da: Lorenzo Bruni, Francesco Clary, Anna Lutsyshyna, Alessio Panarese

SOMMARIO

RACCOLTA E ANALISI DEI REQUISITI

L'obiettivo è realizzare un'applicazione Client-Server in grado di simulare in modo realistico un sistema avanzato di inserimento degli eventi avversi, visibili nella ricerca dei centri in modo tale che gli utenti possano vedere in quale centro ci sono stati più effetti collaterali dovuti al vaccino.

Sia l'utente che l'operatore vaccinale avranno lo stesso client. Il Server è usato solo per salvare tutti i dati inseriti dagli utenti e mostrare i centri vaccinali registrati dagli operatori.

Per prima cosa l'operatore (munito di username e password autorizzate) deve registrare un centro vaccinale inserendo: Nome centro vaccinale, Comune, CAP, Via, Nome via, Civico, Provincia e Tipologia.

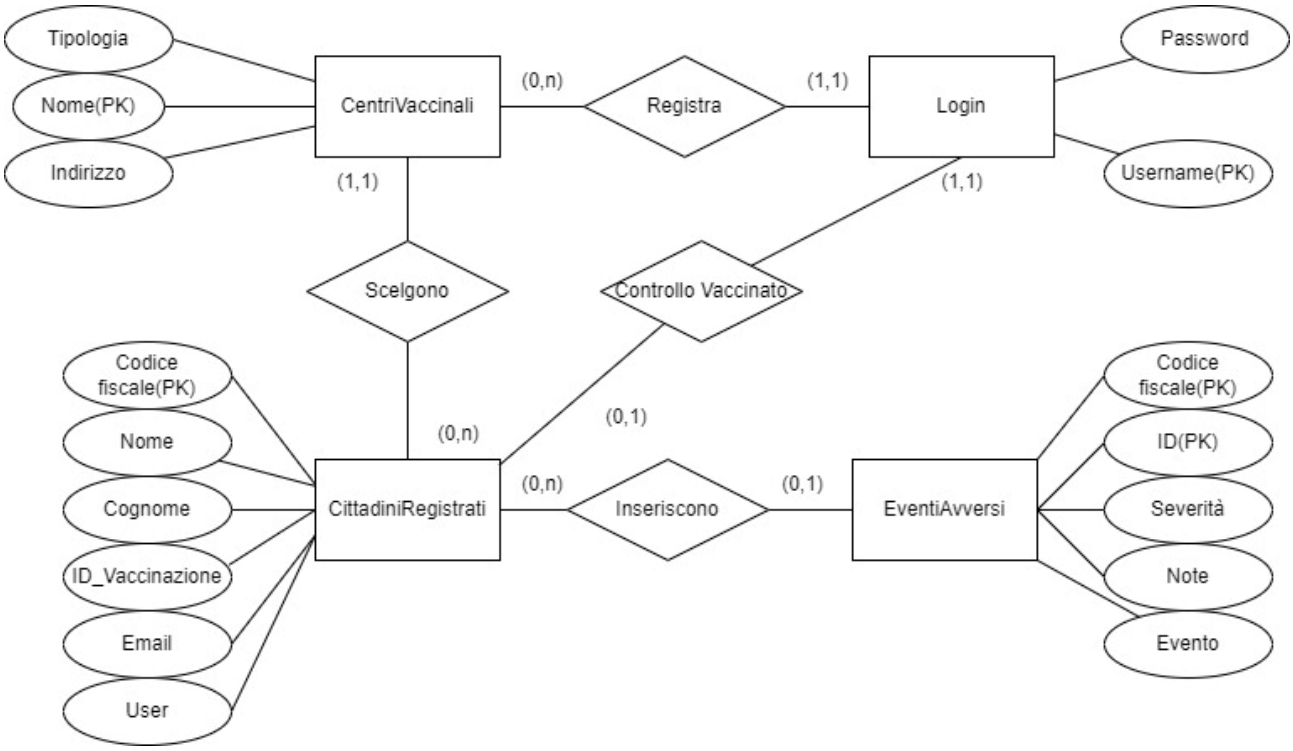
Successivamente, l'operatore dovrà inserire il cittadino vaccinato inserendo: Nome Cognome, Data di nascita, Genere, Codice Fiscale, Data di somministrazione, Vaccino, Nome centro vaccinale, Comune centro vaccinale, CAP, Via, Nome Via, Civico, Tipologia e Provincia.

Una volta registrato il cittadino vaccinato, l'operatore dovrà fornire al cittadino l'ID univoco generatosi per verificare effettivamente se tale si è vaccinato.

L'utente, per registrarsi, dovrà inserire: Nome, Cognome, Data di nascita, Genere, E-Mail, Codice Fiscale, ID Univoco (fornito dall'operatore vaccinale una volta che il cittadino si è vaccinato), Comune del centro vaccinale scelto, Provincia, Nome del centro vaccinale, Cap, Userid, Password e la conferma della Password scelta.

PROGETTAZIONE CONCETTUALE

Di seguito è riportato lo schema concettuale:



NORMALIZZAZIONE

Data la ridotta complessità della struttura del database, non si ritiene necessaria.

PROGETTAZIONE LOGICA

CREAZIONE TABELLE

CentriVaccinali

```
CREATE TABLE IF NOT EXISTS public."CentriVaccinali"
(
    nome character varying(30) COLLATE pg_catalog."default" NOT NULL,
    indirizzo "IndirizzoCentro" NOT NULL,
    tipologia character varying(30) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "CentriVaccinali_pkey" PRIMARY KEY (nome)
)
```

NB IndirizzoCentro è formato da Via, Nome via, Civico, Comune, Provincia e CAP (Vedi Indirizzo.java nel package com.example.models)

CittadiniRegistrati

```
CREATE TABLE IF NOT EXISTS public."CittadiniRegistrati"
(
    codice_fiscale character varying(16) COLLATE pg_catalog."default" NOT NULL,
    nome character varying(30) COLLATE pg_catalog."default" NOT NULL,
    cognome character varying(30) COLLATE pg_catalog."default" NOT NULL,
    id_vaccinazione character varying(16) COLLATE pg_catalog."default" NOT NULL,
    email character varying(30) COLLATE pg_catalog."default" NOT NULL,
    "user" character varying(30) COLLATE pg_catalog."default" NOT NULL,
    password character varying(30) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "CittadiniRegistrati_pkey" PRIMARY KEY (codice_fiscale)
)
```

Questa tabella è dinamica in quanto per ogni centro vaccinale inserito, si creerà una tabella associata dove verranno inseriti i vaccinati di quel centro vaccinale in questione. Per fare ciò è stato utilizzato tale codice:

```
String query = "CREATE TABLE IF NOT EXISTS
public.\"CittadiniRegistrati_"+datiVaccinato.get(8)+"\""+
                "( nome character varying(30) COLLATE pg_catalog.\"default\" NOT NULL," +
                "  cognome character varying(30) COLLATE pg_catalog.\"default\" NOT NULL," +
                "  data_nascita date NOT NULL," +
                "  genere character varying(30) COLLATE pg_catalog.\"default\" NOT NULL," +
                "  cod_fiscale character varying(30) COLLATE pg_catalog.\"default\" NOT NULL," +
                "  data_somministrazione date NOT NULL," +
                "  nome_vaccino character varying(30) COLLATE pg_catalog.\"default\" NOT NULL," +
```

```
"  nome_centro character varying(30) COLLATE pg_catalog.\"default\" NOT NULL," +
"  id_vaccinazione character varying(30) COLLATE pg_catalog.\"default\" NOT NULL,"
+
"  CONSTRAINT \"CittadiniRegistrati"+datiVaccinato.get(8)+"p_key\" PRIMARY
KEY(id_vaccinazione),"+
"  CONSTRAINT \"CittadiniRegistratiNome"+datiVaccinato.get(8)+"p_key\" FOREIGN
KEY(nome_centro) REFERENCES public.\"CentriVaccinali\" (nome))"
```

EventiAvversi

```
CREATE TABLE IF NOT EXISTS public."EventiAvversi"
(
    id integer NOT NULL DEFAULT nextval('"EventiAvversi_id_seq"'::regclass),
    evento character varying(50) COLLATE pg_catalog."default" NOT NULL,
    note character varying(256) COLLATE pg_catalog."default" NOT NULL,
    "severità" integer NOT NULL,
    codice_fiscale character varying(16) COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT "EventiAvversi_pkey" PRIMARY KEY (id),
    CONSTRAINT "EventiAvversi_fkey" FOREIGN KEY (codice_fiscale)
        REFERENCES public."CittadiniRegistrati" (codice_fiscale) MATCH SIMPLE
        ON UPDATE NO ACTION
        ON DELETE NO ACTION )
```

Login

```
CREATE TABLE IF NOT EXISTS public."Login"
(
    username character varying(20)[] COLLATE pg_catalog."default" NOT NULL,
    password character varying(20)[] COLLATE pg_catalog."default" NOT NULL,
    CONSTRAINT id_login PRIMARY KEY (username)
)
```

Tali dati sono inseribili solamente tramite database in quanto solo gli operatori possono avere le credenziali per accedere al sistema per creare centri vaccinali e inserire i cittadini vaccinati.

QUERY (SERVER)

Seguono le interrogazioni, inserite nei metodi, presenti nella classe Server.java che ci permettono di cercare i centri vaccinali per nome e per tipologia:

getCentriVaccinaliByName()

SELECT nome, (indirizzo).via, (indirizzo).nome, (indirizzo).numero_civico, (indirizzo).comune, (indirizzo).sigla_provincia, (indirizzo).\"CAP\", tipologia FROM public.\"CentriVaccinali\" WHERE nome LIKE '%" + nomeCentro + "%'

getCentriVaccinaliByType()

SELECT nome, (indirizzo).via, (indirizzo).nome, (indirizzo).numero_civico, (indirizzo).comune, (indirizzo).sigla_provincia, (indirizzo).\"CAP\", tipologia FROM public.\"CentriVaccinali\" WHERE tipologia='" + tipoCentro + "' AND (indirizzo).comune LIKE '%" + comune + "%'"

Quest'altra interrogazione invece ci permette di recuperare i codici fiscali dei cittadini vaccinati presso un centro vaccinale preciso.

getCittadiniVaccinati()

SELECT cod_fiscale FROM public.\"CittadiniRegistrati_" + centro + "\""

La query successiva invece ci permette di recuperare i codici univoci di vaccinazione di tutti i cittadini vaccinati presso un preciso centro vaccinale.

getCittadiniVaccinatiId()

SELECT id_vaccinazione FROM public.\"CittadiniRegistrati_" + centro + "\"

Il metodo successivo si occupa di recuperare, tramite query i dati di login presso la tabella login.

getDatiLogin()

SELECT * FROM public.\"Login\";"

Il metodo che segue ci restituisce i cittadini registrati al sistema.

getCittadiniRegistrati()

SELECT * FROM public.\"CittadiniRegistrati\";

Questo metodo inserisce nel database un nuovo centro vaccinale

setCentroVaccinale()

INSERT INTO public.\"CentriVaccinali\" (nome,indirizzo.via,indirizzo.nome,indirizzo.numero_civico,indirizzo.comune,indirizzo.sigla_provincia,indirizzo.\"CAP\",tipologia) VALUES ('"+ nomCentro +"'::character varying, '"+ qualCentro +"'::character varying, '"+ indirizzoCentroString +"' ::character varying, '"+ civicoCentro +"' , '"+ comuneCentro +"'::character varying, '"+ siglaCentro +"'::character varying, '"+ capCentro +"','"+ tipoCentro +"'::character varying);"

Quest'altro metodo invece, inserisce, nella tabella dinamica creata per l'apposito centro vaccinale, il cittadino vaccinato.

setVaccinato()

INSERT INTO public.\"CittadiniRegistrati_"+datiVaccinato.get(8)+"\"(nome,cognome,data_nascita,genere,cod_fiscale,data_somministrazione,nome_vaccino,nome_centro,id_vaccinazione)"+

```
"VALUES ('"+ datiVaccinato.get(0) +"'::character varying," +
    "'"+ datiVaccinato.get(1) +"'::character varying," +
    "'"+ datiVaccinato.get(2) +"'::date," +
    "'"+ datiVaccinato.get(3) +"'::character varying," +
    "'"+ datiVaccinato.get(4) +"'::character varying," +
    "'"+ datiVaccinato.get(5) +"'::date," +
    "'"+ datiVaccinato.get(6) +"'::character varying," +
    "'"+ datiVaccinato.get(8) +"'::character varying," +
    "'"+ datiVaccinato.get(7) +"'::character varying)
```

Con la seguente query possiamo recuperare i centri vaccinali registrati nel sistema.

getCentriVaccinali()

SELECT nome, (indirizzo).via, (indirizzo).nome, (indirizzo).numero_civico, (indirizzo).comune, (indirizzo).sigla_provincia, (indirizzo).\"CAP\", tipologia FROM public.\"CentriVaccinali\"

Tale metodo inserisce nella tabella “CittadiniRegistrati” il cittadino appena registrato.

registraCittadino()

INSERT INTO public.\"CittadiniRegistrati\" (codice_fiscale, nome, cognome, id_vaccinazione, email, \"user\", password) VALUES ('" + registrato.getCodiceFiscale() + "'::character varying, \'" + registrato.getNome() + "\'::character varying, '" + registrato.getCognome() + "'::character varying, '" + registrato.getIdVaccinazione() + "', '" + registrato.getEmail() + "'::character varying, '" + registrato.getUserid() + "'::character varying, '" + registrato.getPassword() + "'::character varying);

Inserimento eventi avversi nella tabella eventi avversi associati al codice fiscale.

registraEventiAvversi()

INSERT INTO public.\"EventiAvversi\" (evento, note, \"severità\", codice_fiscale) VALUES ('" + ev.getEvento() + "'::character varying, '" + ev.getNoteOpzionali() + "'::character varying, '" + ev.getSeverità() + "'::integer, '" + codiceFiscale + "'::character varying);"

Recupero eventi avversi da database associati sempre al codice fiscale.

getEventiAvversi()

SELECT * FROM public.\"EventiAvversi\" WHERE codice_fiscale='" + codiceFiscale + "' AND severità > 1"