SDD

System Design Document – Dati Persistenti

Partecipanti

Cognome	Nome	Matricola
Campofreda	Alessio	05121 05492
Caccia	Raffaele	05121 05514
Pincivalle	Rolando Antonio	05121 05192
Spera	Francesco	05121 05408

Indice

1. DESCRIZIONE DEI DATI PERSISTENTI

1.1 PAZIENTE

- Email: string (PK)
- Password: string
- Domicilio: string
- Residenza: string

La tabella Paziente contiene informazioni riguardo il paziente che vorrà interfacciarsi con i medici presenti in piattaforma.
Il paziente è identificato univocamente tramite la sua email.

.

1.2 MEDICO

- Email: string (PK)
- Password: string
- Telefono: string
- IndirizzoStudio: string
- ComuneStudio: string
- MediaRecensioni: float
- Tipo: string

La tabella Medico contiene informazioni riguardo il medico e il ruolo che svolge sulla piattaforma (medico di base oppure una specializzazione).
Il medico è identificato univocamente tramite la sua email.

il sistema dovrà essere disponibile h24 in modo da permettere agli utenti di usufruirne costantemente.

1.3 ACCOUNT

- IDAccount: int (PK)
- Nome: string
- Cognome: string
- DataDiNascita: date
- CF: string
- Foto: longblob
- Paziente: string (FK)
- Medico: string (FK)

La tabella Account contiene informazioni generali riguardo l'account creato sulla piattaforma. L'account è identificato univocamente dal campo IDAccount ed è associato al tipo di account (paziente o medico) attraverso i campi Paziente e Medico (chiavi esterne).

1.4 VOTAZIONE

- IDVotazione: int (PK)
- Data: date
- Voto: int
- Paziente: string (FK)
- Medico: string (FK)

La tabella Votazione contiene informazioni riguardo i voti di un medico.
La tabella è identificata univocamente dal campo IDVotazione. È associato al paziente che ha recensito tramite il campo Paziente (chiave esterna) e associato al medico recensito tramite il campo Medico (chiave esterna).

1.5 APPUNTAMENTO

- IDAppuntamento: int (PK)
- Data: date
- Ora: string
- Stato: int
- Medico: string (FK)
- Paziente: string(FK)

La tabella Appuntamento contiene informazioni riguardo l'appuntamento accordato tra medico e paziente.
La tabella è identificata univocamente dal campo IDRecensione. È associato al medico tramite il campo Medico (chiave esterna) e al Paziente(chiave esterna).

1.6 REFERTO

- IDReferto: int (PK)
- Oggetto: string
- Descrizione: string
- Data: date
- Paziente: string(FK)
- Medico: string (FK)

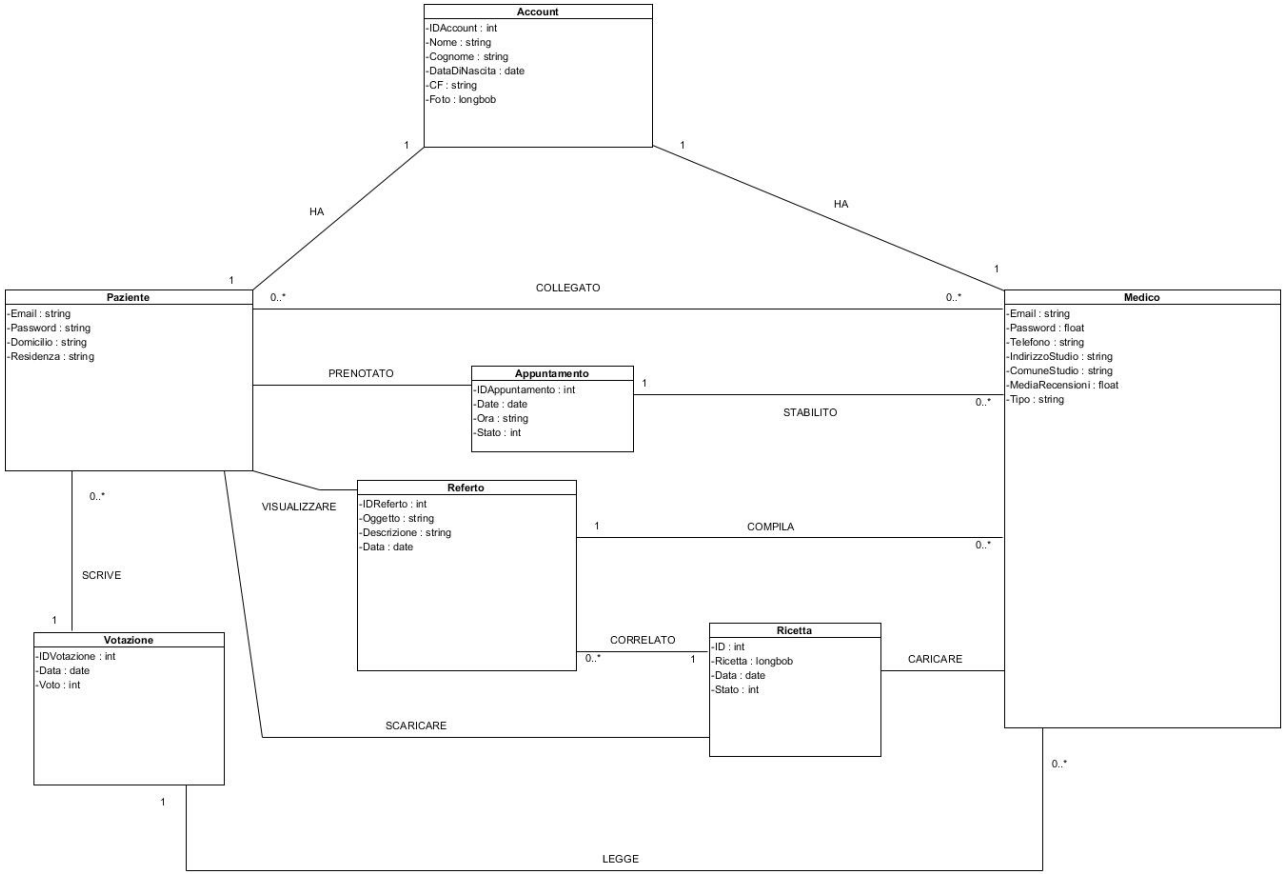
La tabella Referto contiene informazioni riguardo il referto scritto dal medico.
La tabella è identificata univocamente dal campo IDReferto. È associato al medico tramite il campo Medico (chiave esterna) e al paziente tramite il campo Paziente (chiave esterna).

1.7 RICETTA

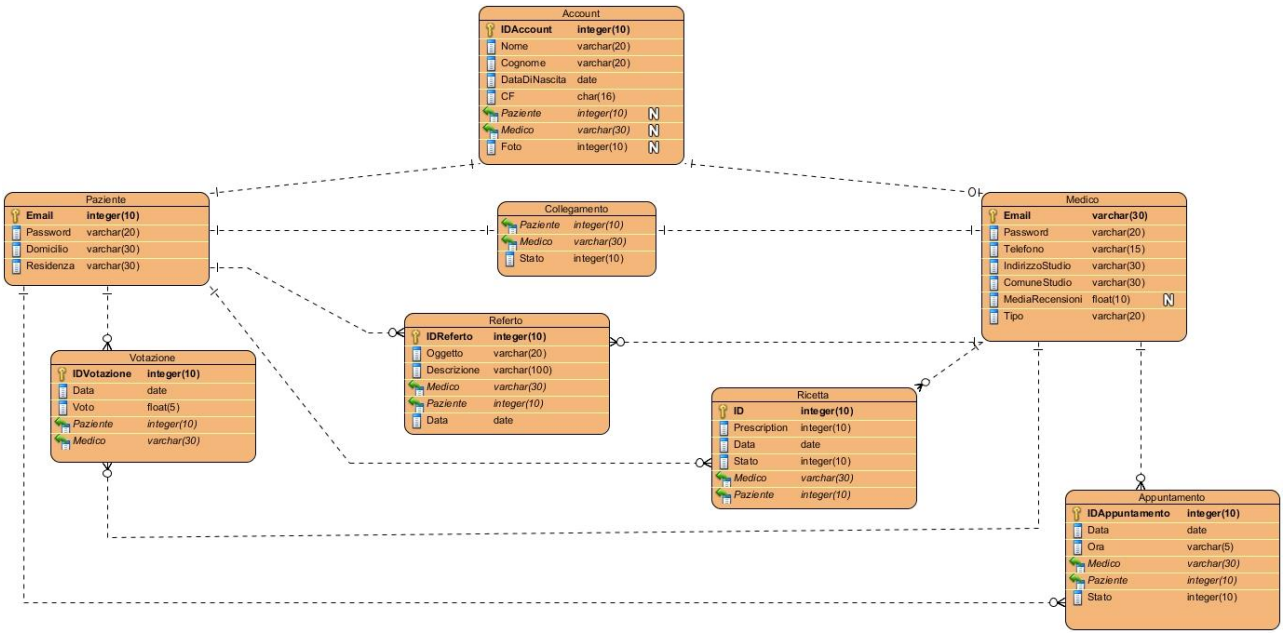
- ID: int (PK)
- Ricetta: longbob
- Data: date
- Stato: int
- Medico: string (FK)
- Paziente: string (FK)
- Referto: int (FK)

La tabella Ricetta contiene informazioni riguardo la ricetta prescritta dal medico. La tabella è identificata univocamente dal campo ID. È associato al referto tramite il campo Referto (chiave esterna), al medico tramite il campo Medico (chiave esterna) e al paziente tramite il campo Paziente (chiave esterna).

2. DIAGRAMMA DEI DATI PERSISTENTI



3. SCHEMA LOGICO



4. QUERIES

4.1 ProfiloManager

Autenticazione:

- SELECT * FROM patient WHERE email = ? AND password = ?;
- SELECT * FROM account WHERE patient = ?;
- SELECT * FROM doctor WHERE email = ? AND password = ?;
- SELECT * FROM account WHERE doctor = ?;

Registrazione:

- INSERT INTO patient (Email, Password, Domicile, Residence) VALUES (?, ?, ?, ?);
- INSERT INTO account (Name, Surname, BirthDate, Cf, Photo, Patient) VALUES (?, ?, ?, ?, ?, ?);
- INSERT INTO doctor (Email, Password, PhoneNumber, StudioAddress, Type, MunicipalityAddress) VALUES (?, ?, ?, ?, ?, ?);
- INSERT INTO account (Name, Surname, BirthDate, Cf, Photo, Doctor) VALUES (?, ?, ?, ?, ?, ?);

visualizzaPaziente:

- SELECT email, domicile, residence FROM patient WHERE email = ?;

visualizzaMedico:

- SELECT email, phonenumber, studioaddress, avgReviews, type, municipalityaddress FROM doctor WHERE email = ?;

modificaProfilo:

- UPDATE account SET Name = ? WHERE patient = ?;
- UPDATE account SET Surname = ? WHERE patient = ?;
- UPDATE account SET BirthDate = ? WHERE patient = ?;
- UPDATE account SET CF = ? WHERE patient = ?;
- UPDATE patient SET Password = ? WHERE email = ?;
- UPDATE patient SET Domicile = ? WHERE email = ?;
- UPDATE patient SET Residence = ? WHERE email = ?;
- UPDATE account SET Name = ? WHERE doctor = ?;
- UPDATE account SET Surname = ? WHERE doctor = ?;
- UPDATE account SET BirthDate = ? WHERE doctor = ?;
- UPDATE account SET CF = ? WHERE doctor = ?;
- UPDATE doctor SET Password = ? WHERE email = ?;
- UPDATE doctor SET PhoneNumber = ? WHERE email = ?;
- UPDATE doctor SET StudioAddress = ? WHERE email = ?;
- UPDATE doctor SET MunicipalityAddress = ? WHERE email = ?;

cercaAccount:

- SELECT name FROM account WHERE patient = ? OR doctor = ?;

4.2 RefertoManager

caricaReferto:

- INSERT INTO medicalreport(Object, Description, Doctor, Patient, Date) VALUES(?, ?, ?, ?, ?);

getRefertoById:

- SELECT * FROM medicalreport WHERE idReport = ?;

getRefertoByPaziente:

- `SELECT object, date, idreport FROM medicalreport m WHERE m.patient = ?;`

4.3 RicercaManager

`cercaMedicoZonaTipo:`

- `SELECT email, phonenumber, studioaddress, avgreviews, type FROM doctor d WHERE d.type = ? AND d.municipalityAddress = ?;`

`cercaMedicoNome:`

- `SELECT email, phonenumber, studioaddress, avgreviews, type FROM doctor d, account a, link l WHERE d.email = a.doctor AND l.state = 1 AND l.doctor = d.email AND l.patient = ?  AND CONCAT( name,  ' ', surname ) LIKE ?;`

`cercaPazienteNome:`

- `SELECT email, domicile, residence FROM patient p, account a, link l WHERE p.email = a.patient AND l.state = 1 AND l.patient = p.email AND l.doctor = ?  AND CONCAT( name,  ' ', surname ) LIKE ?;`

`getMedici:`

- `SELECT name, surname, doctor FROM account a WHERE a.doctor = (SELECT doctor FROM link l WHERE l.patient = ?);`

`getPazienti:`

- `SELECT name, surname, patient, birthdate FROM account a WHERE a.patient = (SELECT patient FROM link l WHERE l.doctor = ?);`

`cercaAccountMedico:`

- SELECT name, surname, birthdate, photo, doctor FROM account a WHERE a.doctor = ?;

cercaMedico:

- SELECT email, phonenumber, studioaddress, avgreviews, type, municipalityaddress FROM doctor d WHERE d.email = ?;

cercaAccountPaziente:

- SELECT name, surname, cf, birthdate, photo FROM account a WHERE a.patient = ?;

cercaPaziente:

- SELECT email, domicile, residence FROM patient p WHERE p.email = ?;

4.4 AppuntamentoManager

visualizzaAPPmedico:

- SELECT * FROM appointment WHERE Doctor = ?

visualizzaAPPpaziente:

- SELECT * FROM appointment WHERE Patient = ?

visualizzaAppuntamento:

- SELECT * FROM appointment WHERE IDAppointment = ?

controlloAppDisponibile

- SELECT * FROM appointment WHERE Doctor=? AND Date=? AND Time=?

inserisciApp:

- INSERT INTO appointment (Date,Time,Doctor,Patient,State) VALUES (?,?,?,?,?)

ritornoID:

- SELECT * FROM appointment WHERE Date = ? AND Doctor = ? AND Patient = ? AND state = ?

updateAppuntamentoState:

- UPDATE appointment SET state = ? WHERE IDAppointment =?

modificaAppuntamento:

- UPDATE appointment SET Date =?, Time=?,State = ? WHERE IDAppointment =?

4.5 CollegamentoManager

doRetrieveByPazienteCollegato:

- SELECT * FROM patient, account WHERE account.email = p.email AND account.email = ?

doRetrieveByMedicoCollegato:

- SELECT * FROM doctor, account WHERE account.email = doctor.email AND account.email = ?

doRetrieveAll:

- SELECT d.email, d.password, d.phonenumber, d.studioaddress, d.avgreviews, d.type FROM patient p, doctor d, link l WHERE p.email = l.patient AND l.doctor = d.email AND p.email = ?

creaCollegamento:

- INSERT INTO link (Patient, Doctor, State) VALUES (?,?,?)

modificaStatoCollegamento:

- UPDATE link SET state = ? where patient=? and doctor=?

4.6 RicettaManager

ricercaRicettabyID:

- SELECT * FROM prescription WHERE MedicalReport = ? ORDER BY ID DESC;

caricareRicetta:

- `INSERT INTO prescription (MedicalReport, prescription,date,Doctor,Patient,state) VALUES (?,?,?,?,?,?);`

`modificaStato:`

- `UPDATE prescription SET state = ? WHERE ID = ?;`

`ritornoID:`

- `SELECT * FROM prescription WHERE MedicalReport = ? AND doctor = ? AND patient = ? AND state = ?;`

`updateRicetta:`

- `UPDATE prescription SET state = -1, prescription = ?,date = ? WHERE ID = ?;`

4.7 VotazioneManager

`votaMedico:`

- `INSERT INTO voting (Date, Vote, Patient, Medico) VALUES(?,?,?,?);`

5. MOTIVAZIONI

Per la memorizzazione dei dati persistenti abbiamo optato per un database relazionale in modo che possa essere gestito agevolmente l'accesso concorrente ai dati e garantita la consistenza dei dati stessi.

La scelta di un DBMS porta i seguenti vantaggi:

- Affidabilità dei dati: un DBMS offre dei metodi per effettuare backup dei dati e per ripristinare i precedenti dati.
- Privatezza dei dati: un DBMS permette un accesso protetto ai dati.
- Imposizioni di vincoli di integrità sui dati: un DBMS permette di specificare diversi tipi di vincoli per mantenere l'integrità dei dati e garantisce che tali vincoli siano sempre soddisfatti.