MEDBASE

Giugno 2021
ITI “E. Medi”

Alessio Lustri
5° A informatica

Sommario

1. Prova d'esame a.s. 2020/21

1.1. Parte 1

Una struttura ospedaliera polispecialistica vuole registrare l'occupazione dei posti letto nei vari reparti, in modo da monitorare costantemente il numero dei posti disponibili.
Solo il personale dell'Ospedale può accedere al Data Base. con funzioni di accesso personalizzate in base alla mansione ricoperta nella struttura.

- Per ciascun Reparto (Cardiologia, Oncologia, Ostetricia, …) il direttore generale ha valutato il numero massimo di letti.
- L'ufficio accettazione provvede a ricoverare un Assistito (memorizzato con il solo Nome-Cognome) in un reparto previa verifica della disponibilità.
- Il medico può dimettere un assistito (registrando la data di fine del Ricovero), incrementando così i posti letto del reparto.

Lo studente, fatte le opportune ipotesi aggiuntive, sviluppi

1. Un'analisi della realtà di riferimento individuando la soluzione che a suo motivato giudizio è la più idonea a rispondere alle specifiche indicate;
2. Lo schema concettuale della base di dati;
3. Lo schema logico della base di dati;
4. La descrizione dei vincoli di integrità referenziale e/o vincoli di dominio, laddove presenti;
5. La definizione in linguaggio SQL delle relazioni della base di dati;
6. Le seguenti interrogazioni espresse in linguaggio SQL :
   a. prospetto dei Reparti con dati di occupazione dei letti;
   b. elenco dei reparti senza disponibilità;
   c. tempi medi di permanenza degli assistiti in ciascun reparto.
7. Realizzare un Sito Web che consenta l'interazione con il Data Base da parte dei membri dell'organizzazione e/o degli utenti esterni, utilizzando appropriati linguaggi, sia lato client che lato server. A tale scopo:
   a. Progettare la mappa delle pagine del sito, indicando per ciascuna le azioni svolte, l'eventuale passaggio di parametri e i privilegi richiesti per l'accesso.
   b. Suddividere gli utenti in gruppi, assegnando a ciascuno gli opportuni privilegi.
   c. Descrivere il metodo utilizzato per garantire un accesso controllato al sito.
   d. Costruire un'interfaccia completa per la gestione del sito.

Il sito permette la consultazione dei dati on line, quindi, per il rispetto della privacy, ovvero dei dati degli utenti registrati, per ciascuna query, indicare quali sono i privilegi necessari per l'esecuzione (amministratore, utente, visitatore…).

1.2. Parte 2

Il candidato esponga le diverse soluzioni per "ospitare" i contenuti e i servizi implementati nella prima parte, evidenziando i vantaggi e gli svantaggi.

Proponga poi una soluzione adatta al progetto realizzato evidenziando gli aspetti hardware, software e di sicurezza, anche con l'aiuto di diagrammi o schemi.

2. Cos'è MedBase

MedBase è il nome dato alla piattaforma per la gestione di strutture sanitarie del committente, questa richiede una serie di istanze fondamentali per il suo funzionamento:

- Il paziente – colui che viene ricoverato all'interno della struttura,
- Il reparto – dipartimento della struttura dove va ricoverato il paziente,
- Il personale – gli impiegati di ogni tipo della struttura, ognuno potenzialmente con mansioni diverse.

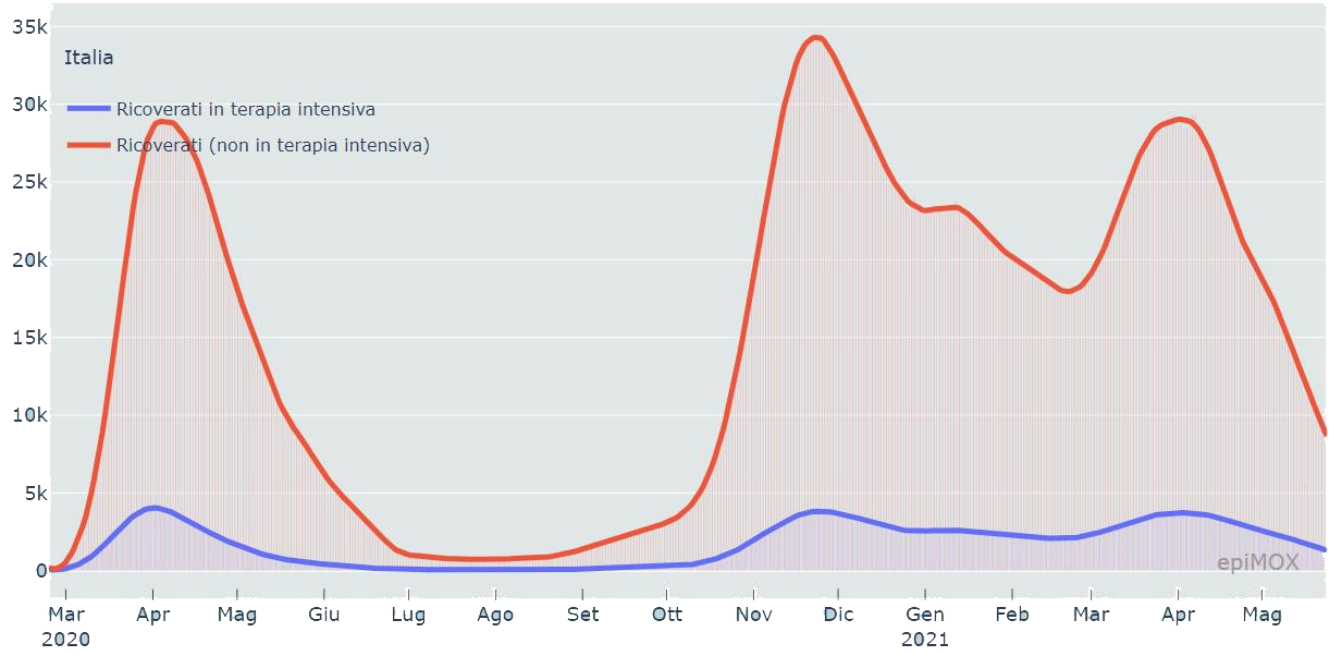
Le entità appena descritte conterranno tutte le informazioni essenziali quali il nome e il cognome dei pazienti, oltre che il codice fiscale. Il reparto avrà oltre che il nome, anche la capienza massima. Il personale invece avrà uno username e una password per effettuare l'accesso alla piattaforma, e un privilegio: ovvero tutte le azioni che un determinato utente può svolgere. Dalla richiesta emerge la necessità di un totale di cinque azioni diverse, ognuna delle quali associata ad un gruppo di utenti:

- Gli analisti, coloro che si occupano di analizzare le statistiche relative a reparti e singoli pazienti.
- I receptionists, il personale addetto all'inserimento dei pazienti non appena arrivano nella struttura, previa verifica di disponibilità di posti letto in quel reparto.
- I medici, il gruppo di utenti che si farà carico dei pazienti e provvederà a dimetterli quando lo riterranno opportuno.
- Il direttore sanitario, la figura che si occupa della gestione dei reparti, aggiungendone di nuovi o rimodulandoli con una diversa capacità di accoglienza.
- Il gestore del personale, la figura incaricata della gestione degli utenti e delle loro possibilità di azione all'interno della piattaforma.

Il sito web della piattaforma prevederà l'accesso direttamente dalla homepage tramite un riquadro a scomparsa. Effettuato l'accesso correttamente, si ritorna alla home e il tasto di login è stato rimpiazzato da un menu a tendina, presente in ogni pagina del sito, da cui si può accedere ad una serie di pagine, ognuna delle quali corrisponde un'azione. Gli utenti che non hanno accesso a determinate funzioni non solo non vedranno comparire i riferimenti a quelle pagine, ma se proveranno ad accedervi in qualunque modo verranno bloccati e visualizzeranno un messaggio di "Accesso non consentito".
Per ragione di sicurezza, tutte le password verranno criptate secondo l'algoritmo "**sha512**", che ne impedisce la decodifica e quindi impedisce la criptazione di tutti gli altri dati che necessitano di essere visualizzati così come sono.
MedBase si appoggia ad un database MariaDB, DBMS nato da un fork del più noto MySQL e si interfaccia con l'utente tramite pagine HTML, con l'ausilio dei linguaggi CSS per lo stile grafico dell'applicazione e JavaScript.
Per la creazione di pagine dinamiche e per comunicare con il database è stato utilizzato il linguaggio PHP, mentre per permettere l'invio di mail al team di supporto di Medbase si è utilizzata la libreria PHPmailer.
La piattaforma è indirizzata principalmente ad un target di utenti spesso non propriamente addentro a le dinamiche dei sistemi informatici, si tratta infatti del personale di strutture mediche, per cui la user experience dovrà tenerne conto.
Per permettere la maggior diffusione possibile e per fornire supporto, nella home del sito è stato posto un form di contatto da compilare a cui poi risponderà il team adatto, che si tratti di supporto in caso di problemi che di informazioni in caso qualcuno sia interessato ad implementarla nella propria struttura.

3. Il contesto

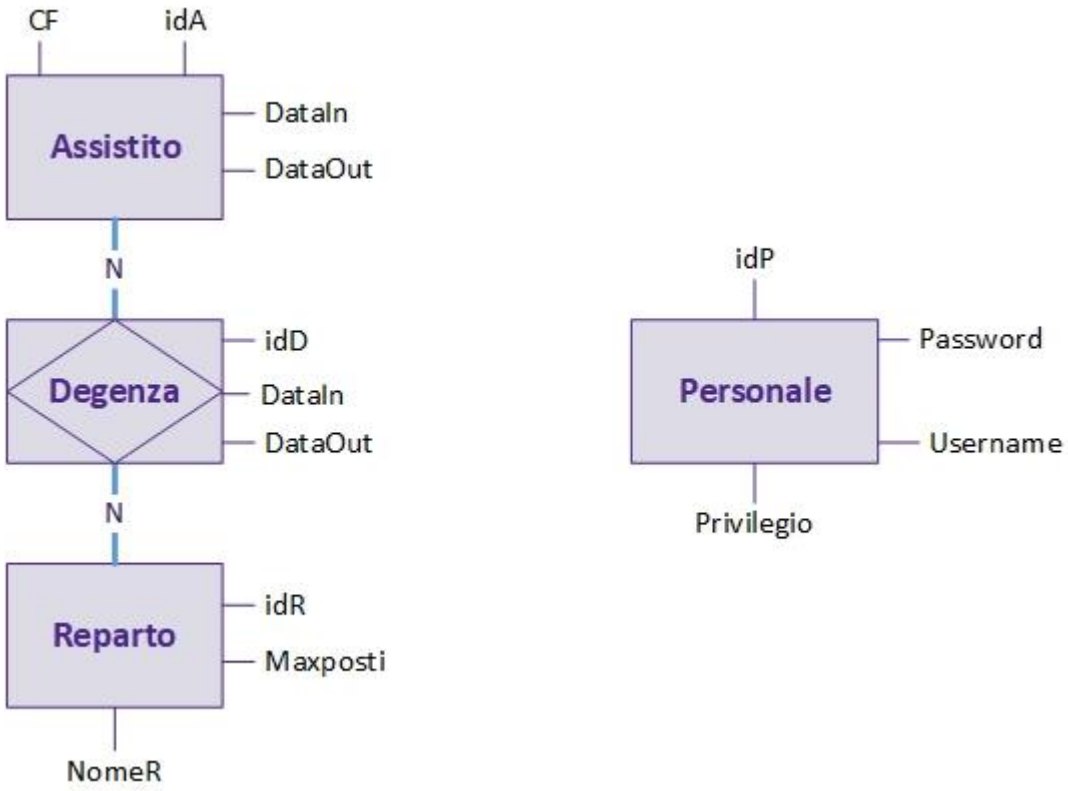
Medbase nasce in un periodo estremamente complicato per le strutture sanitarie di ogni tipo che si sono trovate ad affrontare un'emergenza pandemica senza precedenti in tempi recenti e hanno dovuto garantire una certa flessibilità nella gestione dei reparti in quanto qualora ce ne fosse stato bisogno ogni singolo posto doveva essere riconvertito da ricovero ordinario a un ricovero speciale covid.
L'evoluzione dell'infezione da Sars-COV-2 si è dimostrata estremamente rapida nel tempo ed è in grado di aumentare il numero in pochi giorni.
Questa è la curva dell'andamento dei ricoveri da marzo 2020 a maggio 2021 e come si può notare entrambe le curve hanno un andamento esponenziale.



Il dato fondamentale in un modello simile e per capire l'andamento epidemiologico è il cosiddetto R_0, un dato che indica il numero medio di persone infettate giornalmente da un individuo infetto. Calcolare questo dato è molto importante perché ci permette di capire quale sia la velocità dell'epidemia. In caso di $R_0 > 1$ significa che l'aumento dei casi avviene con una crescita esponenziale, mentre se è uguale a 1, in tal caso allora si parla di una curva piatta in cui l'andamento è costante nel tempo.

4. Schema concettuale

Dall'analisi fatta è emerso una associazione di tipo molti a molti tra l'assistito e il reparto in cui viene ricoverato, dato che se un paziente è stato ricoverato e dimesso, può anche tornare nuovamente nella struttura, magari in un nuovo reparto. Per cui, nella tabella nata per scomporre questa associazione, "Degenza", ad ogni record corrisponderà un ricovero di un paziente. Se lo stesso paziente viene ricoverato più volte, il suo ID sarà presente in tanti record tanti quanti sono i suoi ricoveri. Per tutti gli ID non è richiesto un ordine preciso in cui devono essere inseriti, per cui può pensare il DBMS a inserire il loro valore assegnando l'attributo AUTO_INCREMENT



5. Schema logico

*In **grassetto le chiavi primarie**, sottolineate le chiavi esterne.*

Reparto (**idR**, nomeR, maxPosti)
Personale (**idP**, username, password, Privilegio)
Assistito (**idA**, nomeA, cognomeA, CF)
Degenza (**idD**, DataIn, DataOut, idA, idR)

6. Vincoli

6.1. Vincoli intra-relazionali:

- **Obbligatorietà**: Tutti gli attributi del database sono obbligatori, ad eccezione di DataOut che non può essere riempito finché il paziente non viene dimesso.
- **Vincoli di dominio**:
  - Reparto:
    - NomeR – obbligatorio, massimo 32 caratteri;
    - MaxPosti – obbligatorio, intero;

- Personale:
  - Username – obbligatorio, massimo 32 caratteri;
  - Password – obbligatorio, lunghezza fissa di 128 caratteri;
  - Privilegio – obbligatorio, intero;
- Assistito:
  - NomeA - obbligatorio, massimo 32 caratteri;
  - CognomeA - obbligatorio, massimo 32 caratteri;
  - CF – obbligatorio, lunghezza fissa di 16 caratteri;
- Degenza:
  - DataIn – obbligatorio, data;
  - DataOut – data, di default *null*;

- **Vincoli di tupla**:
  - La DataOut non può essere antecedente a DataIn.
- **Vincoli di unicità**:
  - Un singolo paziente non può essere ricoverato in due reparti contemporaneamente, per cui non ci possono essere due record che fanno riferimento al medesimo assistito e che entrambi abbiano il campo DataOut vuoto.
  - Il campo CF (Codice fiscale) è univoco per ogni paziente.

6.2. Vincoli inter-relazionali:

- **Vincoli di integrità referenziale**:
  - L'ordine in cui le tabelle vengono create deve essere tale in modo che le chiavi esterne siano già esistenti alla creazione della tabella.
  - Il reparto non può più essere eliminato non appena avrà ospitato almeno un paziente, questo perché sarebbe poi impossibile recuperare i dati su quello specifico ricovero per eventuali ricerche statistiche.

7. Analisi delle associazioni

- L'associazione tra Assistito e Degenza è di tipo 1-N:
  - Un assistito può avere più degenze nella struttura.
  - Più degenze possono riguardare un singolo paziente.
- L'associazione tra Reparto e Degenza è di tipo 1-N:
  - Un reparto può accogliere più pazienti in diversi ricoveri.
  - Più degenze possono avvenire in un unico reparto.

8. Codifica in SQL

8.1. Reparti

```
CREATE TABLE IF NOT EXISTS reparti
(
    idR       INT AUTO_INCREMENT NOT NULL,
    NomeR     VARCHAR(32) NOT NULL,
    MaxPosti  INT NOT NULL,

    PRIMARY KEY (idR)
);

INSERT INTO reparti (NomeR, MaxPosti)
         VALUES ('Cardiologia', 25);
INSERT INTO reparti (NomeR, MaxPosti)
         VALUES ('Pneumologia', 5);
INSERT INTO reparti (NomeR, MaxPosti)
         VALUES ('Terapia intensiva', 1);
```

REPARTI		
idR	**NomeR**	**MaxPosti**
1	Cardiologia	25
2	Pneumologia	5
3	Terapia intensiva	1

8.2. Assistiti

```
create table if not exists assistiti
(
    idA       INT AUTO_INCREMENT NOT NULL,
    nomeA     VARCHAR(32) NOT NULL,
    cognomeA  VARCHAR(32) NOT NULL,
    CF        CHAR(16) NOT NULL,

    PRIMARY KEY (idA)
);

INSERT INTO assistiti (nomeA, cognomeA, CF)
         VALUES ('Alessio', 'Lustri', 'LSTLSS02B19F839U');
INSERT INTO assistiti (nomeA, cognomeA, CF)
         VALUES ('Gabriel', 'Amore', 'MRAGRL02P09H892Y');
INSERT INTO assistiti (nomeA, cognomeA, CF)
         VALUES ('Vittorio', 'Picone', 'PCNVTR02L30F839G');
```

ASSISTITI			
idA	**NomeA**	**CognomeA**	**CF**
1	Alessio	Lustri	LSTLSS02B19F839U
2	Gabriel	Amore	MRAGRL02P09H892Y
3	Vittorio	Picone	PCNVTR02L30F839G

8.3. Personale

```
CREATE TABLE IF NOT EXISTS personale
(
    idP         INT AUTO_INCREMENT NOT NULL,
    username    VARCHAR(32) NOT NULL,
    password    VARCHAR(128) NOT NULL,
    privilegio INT NOT NULL,

    PRIMARY KEY (idP)
);

INSERT INTO personale (username, password, privilegio)
         VALUES ('admin',
'c7ad44cbad762a5da0a452f9e854fdc1e0e7a52a38015f23f3eab1d80b931dd472634dfac71cd34eb
c35d16ab7fb8a90c81f975113d6c7538dc69dd8de9077ec', 31);
INSERT INTO personale (username, password, privilegio)
         VALUES ('analista',
'659b83afdc290b9e794af4fb2bec4ebb416ebeecc6efd3f0d957e1f1923e5b088e97bfa41e2385c77
d24da9e7f0fd0ca716213052154f7bbc58ecaef45e55843', 1);
INSERT INTO personale (username, password, privilegio)
         VALUES ('direttore',
'b6af56c356c4dc45fea6e06b3703cddca0bd1cba3dc919d71927c37639ec04a61688094334efdd911
dad7524cbaeffda2dc7add17459e20a0ab7d7da4880a1f2', 8);
```

PERSONALE			
idP	**username**	**password**	**privilegio**
1	admin	c7ad44cbad762a5da0a452f9e854fdc1e0e7a52a38015f23f3eab1d80b931dd472634dfac71cd34ebc35d16ab7fb8a90c81f975113d6c7538dc69dd8de9077ec	31
2	analista	659b83afdc290b9e794af4fb2bec4ebb416ebeecc6efd3f0d957e1f1923e5b088e97bfa41e2385c77d24da9e7f0fd0ca716213052154f7bbc58ecaef45e55843	1
3	direttore	b6af56c356c4dc45fea6e06b3703cddca0bd1cba3dc919d71927c37639ec04a61688094334efdd911dad7524cbaeffda2dc7add17459e20a0ab7d7da4880a1f2	8

8.4. Degenze

```
create table if not exists degenze
(
    idD INT AUTO_INCREMENT NOT NULL,
    DataIn DATE NOT NULL,
    DataOut DATE,
    idA INT NOT NULL,
    idR INT NOT NULL,

    PRIMARY KEY (idD),
    FOREIGN KEY (idA) REFERENCES assistiti (idA),
    FOREIGN KEY (idR) REFERENCES  reparti   (idR)
);

INSERT INTO degenze (DataIn, DataOut, idA, idR)
          VALUES ('2020/10/10', '2020/10/20', 1, 1);
INSERT INTO degenze (DataIn, DataOut, idA, idR)
          VALUES ('2020/11/20', '2020/12/10', 2, 2);
INSERT INTO degenze (DataIn, DataOut, idA, idR)
          VALUES ('2021/01/10', '2021/01/15', 3, 3);
```

DEGENZE				
idD	DataIn	DataOut	idA	idR
1	2020/10/10	2020/10/20	1	1
2	2020/11/20	2020/12/10	2	2
3	2021/01/10	2021/01/15	3	3

9. Analisi della qualità del database

9.1. Prima forma normale

Il database in oggetto è conforme alla prima forma normale in quanto ogni attributo è definito su un dominio di valori atomici, cioè indivisibili e ogni record ha una propria chiave primaria univoca.

9.2. Seconda forma normale

Il database risulta essere conforme anche seconda forma normale poiché rispetta gli standard del suddetto livello. Infatti, ogni attributo **non** chiave dipende esclusivamente dall'intera chiave primaria del record, e non ad una parte di essa.

9.3. Terza forma normale

Il database è infine conforme anche alla terza forma normale poiché ogni attributo non chiave non dipende in alcun modo da un altro attributo non chiave del record, eliminando così la proprietà della dipendenza transitiva degli attributi.

10. Query richieste

10.1. Prospetto dei reparti con dati di occupazione dei letti

Per ottenere la tabella richiesta, bisogna prima capire la condizione secondo la quale un ricovero è ancora in corso, oppure è stato terminato. Questa condizione è data dal campo DataOut della tabella Degenze, in caso questo sia *null* significa che il paziente è ancora presente nella struttura.
Applicata questa condizione si va a contare quante sono in quel momento le degenze in corso per ogni reparto, ottenendo il numero di posti letto occupati.

```
select R.NomeR, count(D.idD) as PostiOccupati, R.MaxPosti,
    (MaxPosti-count(D.idD)) as PostiDisponibili
from degenze D, reparti R
where D.DataOut is null
    AND R.idR = D.idR
group by(D.idR)
```

TABELLA RISULTANTE			
NomeR	**PostiOccupati**	**MaxPosti**	**PostiDisponibili**
Cardiologia	1	25	1
Pneumologia	1	5	2
Terapia intensiva	1	1	3

10.2. Elenco dei reparti senza disponibilità;

Similmente alla query precedente, si va a verificare quanti sono i posti letto occupati per ogni reparto e si va ad estrarre il nome del reparto solo dove il numero di posti massimi (MaxPosti) coincide con quello dei posti occupati.

```
select R.NomeR, R.MaxPosti, T1.PostiOccupati
from reparti R,
     (
        select R.idR, R.NomeR, count(D.idD) as PostiOccupati
        from degenze D, reparti R
        where D.DataOut is null
            AND R.idR = D.idR
        group by(D.idR)
     ) as T1
where R.MaxPosti = T1.PostiOccupati
    and T1.idR = R.idR
```

TABELLA RISULTANTE		
NomeR	**MaxPosti**	**PostiOccupati**
Terapia intensiva	1	1

10.3. Tempi medi di permanenza degli assistiti in ciascun reparto.

Per ottenere il prospetto richiesto, è necessario prima di tutto andare ad individuare la durata dei singoli ricoveri con il reparto associato, quando dato verrà poi raggruppato in base al reparto in cui è avvenuto il ricovero e viene calcolata la media con una precisione di due cifre decimali.

```
select DISTINCT NomeR, truncate(AVG(T1.Durata),2) as PermanenzaMedia
from reparti R,
    (
        select idR, DATEDIFF(D.DataOut, D.DataIn)+1 as Durata
        from degenze D
        where DataOut is not null
    ) as T1
where R.idR = T1.idR
group by T1.idR
```

TABELLA RISULTANTE	
NomeR	**PermanenzaMedia**
Cardiologia	11.00
Pneumologia	21.00
Terapia intensiva	6.00

11. Sito Web

11.1. Analisi del mercato e del target

Scopo della piattaforma:

La piattaforma si propone come uno strumento utile a semplificare la gestione di complesse strutture sanitarie polispecialistiche. La concorrenza, anche nel settore pubblico, è estremamente ricca e questo dovrebbe rappresentare un incentivo a sviluppare una piattaforma che abbia il più ampio target possibile, che riceva i feedback migliori possibili dagli utilizzatori ai quali risulti facile l'utilizzo dell'interfaccia.

Target:

La piattaforma va sponsorizzata ai dirigenti di strutture sanitarie e a tutti coloro che hanno potere decisionale su quest'ultime. D'altro canto, però la piattaforma va ad essere utilizzata principalmente da medici, che molto spesso non sono esperti in tecnologia e da personale addetto alla ricezione dei pazienti.
La progettazione dell'interfaccia grafica e quindi del design delle pagine deve essere il più user-friendly possibile.

11.2. Pianificazione e timeline del progetto

Chiarito lo scopo e il target della piattaforma web si è progettato la struttura del sito e si è optato per un sito a maglia intrecciata: questo consente all'utilizzatore di avere, tramite una barra di navigazione (navbar) posta in alto, un punto di riferimento in qualunque pagina egli si trovi.
Tramite la navbar appunto, si potrà poi raggiungere tutte le pagine a cui ha diritto di accesso in base al suo profilo tramite un menu a tendina oppure modificare la password del proprio account ed eventualmente effettuare il logout, evitando continui cambi di pagina per potersi muovere all'interno del sito.

Analizzate le richieste del committente si è convenuto che le azioni del sito si districheranno in un totale di 6 (sei) pagine, con le quali l'utente interagisce.
Le pagine sono le seguenti:

- **Home**, la quale farà da presentazione alla piattaforma e con un modulo di contatto per informazioni o problemi.
- **Statistiche**, tramite la quale gli addetti potranno analizzare informazioni statistiche riguardanti pazienti e reparti.
- **Inserimento paziente**, tramite la quale gli addetti alla ricezione possono inserire i pazienti non appena questi vengono accolti nella struttura.
- **Dimissione paziente**, con la quale i medici potranno, dopo aver inserito il codice fiscale del diretto interessato, verificare da quanto tempo il paziente è ricoverato, in quale reparto ed eventualmente dimetterlo.
- **Gestione reparti**, dove i direttori sanitari potranno effettuare modifiche ai reparti, visualizzare l'elenco dei ricoverati, rimuovere i reparti in caso non siano mai stati utilizzati o anche crearne di nuovi.
- **Gestione utenti**, la quale permetterà al gestore del personale medico di fornire il personale di account ma anche di limitare o ampliare le possibilità di azione all'interno della piattaforma.

Il sito sarà inoltre adattabile a tutti i tipi di risoluzione di monitor, essendo realizzato tramite il framework "Bootstrap" che permette tramite delle classi HTML di suddividere in 12 colonne la pagina e di indicare ogni singolo elemento quante di queste colonne deve occupare, in base alla grandezza dello schermo.

Scelta dei linguaggi

Per la codifica del sito sono stati adottati i seguenti linguaggi:

- **HTML**, per la creazione delle pagine Web con cui si interfaccia l'utente.
- **CSS**, per la personalizzazione delle pagine HTML.
- **PHP**, che svolge il ruolo di middleware tra il client e il DB server.
- **JavaScript**, per rendere le pagine HTML dinamiche e per la sua libreria jQuery.

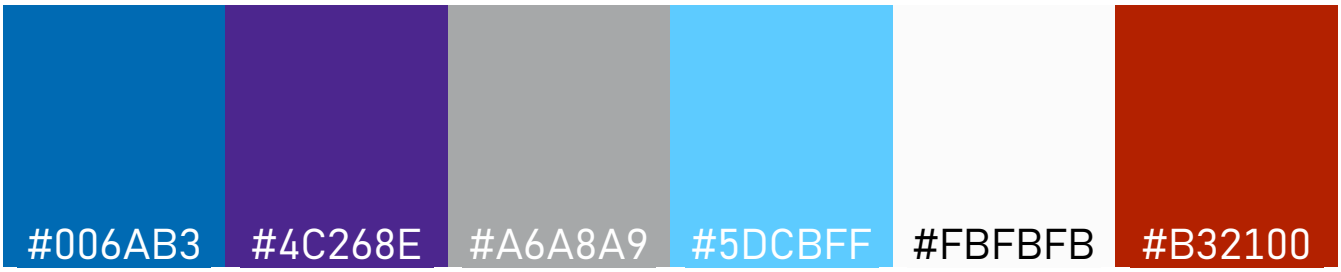
Timeline

L'ordine di sviluppo (timeline) dell'intera piattaforma viene ad essere di conseguenza il seguente:

1- Definizione delle tabelle del database in SQL.
2- Sviluppo della navbar.
3- Sviluppo delle pagine di login e logout
4- Sviluppo della pagina "Gestione reparti" con le relative funzioni back-end.
5- Sviluppo della pagina "Inserimento pazienti" con la relativa funzione back-end.
6- Sviluppo della pagina "Dimissione pazienti" e la relativa funzione back-end.
7- Sviluppo della pagina "Statistiche" e le relative funzioni back-end.
8- Sviluppo della pagina "Gestione account" e le relative funzioni back-end.
9- Sviluppo della pagina "Index" e del form di contatto.

12. Palette colori

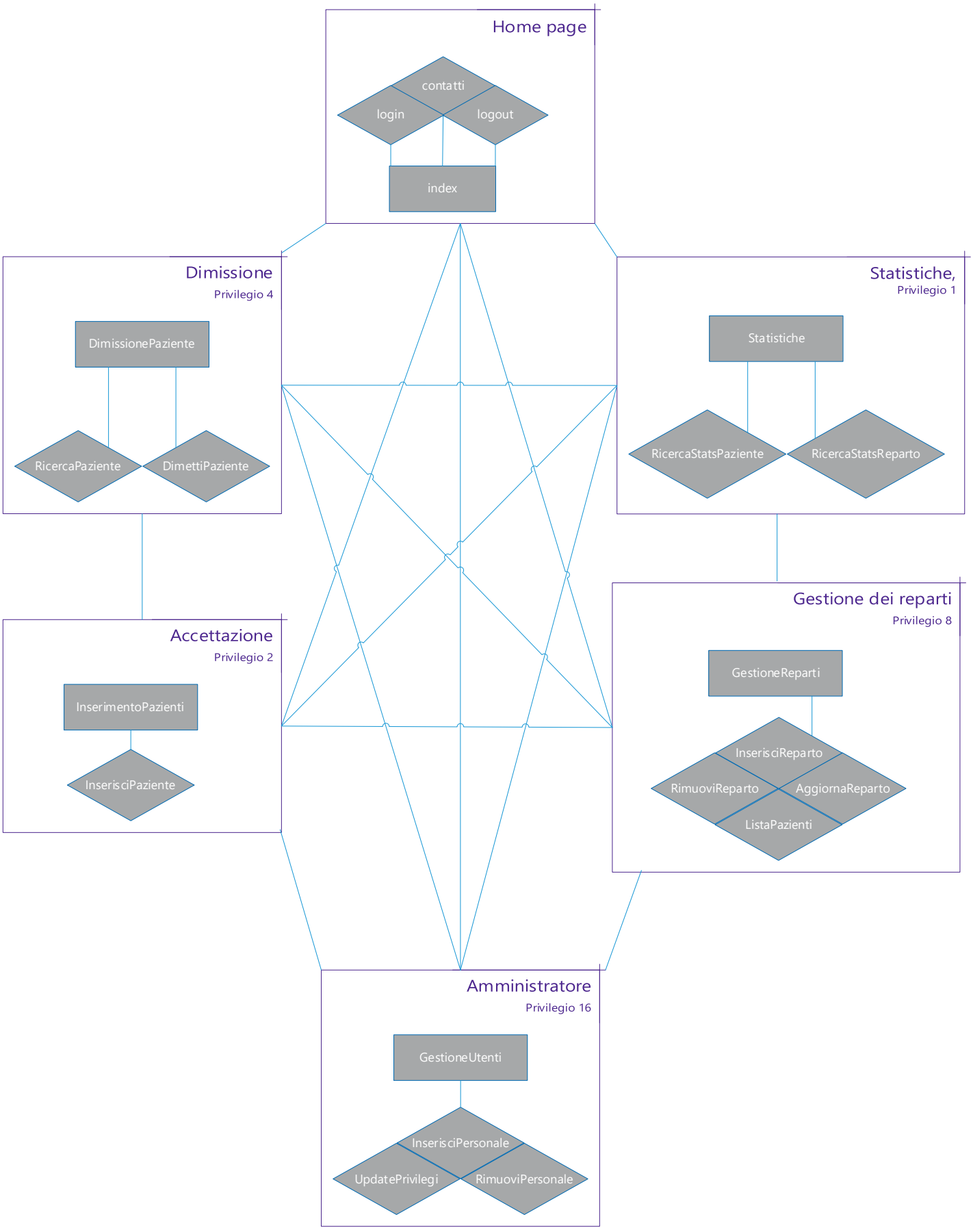
La scelta della palette da utilizzare è partita scegliendo i due colori primari, e successivamente tutti i colori che fanno da contorno.
Il blu primario è stato usato per il logo, mentre il viola è stato utilizzato per gli elementi caratteristici del sito web.
Di seguito, i codici esadecimali utili per una possibile campagna di marketing o di rebranding.



13. Privilegi

Id	Mansione	Descrizione
1	Analista	Visualizzazione e analisi delle statistiche relative a reparti e pazienti.
2	Receptionist	Inserimento dei pazienti all'interno della piattaforma.
4	Medico	Dimissione del paziente dai reparti.
8	Direttore sanitario	Impostazione posti letti massimi e precedenti.
16	Gestore del personale	Gestione degli account e dei relativi privilegi.

14. Mappa del sito



15. Garanzie di accesso controllato

La piattaforma prevede l'accesso alle pagine esclusivamente se il gestore del personale al momento della creazione dell'account ha impostato il "privilegio" necessario per accedere alla pagina.
Il controllo viene effettuato tramite il linguaggio PHP e la variabile di sessione dell'utente creata in fase di login. In caso non si possegga il privilegio necessario per accedere alla pagina, verrà mostrato un avviso e nient'altro.
Lo stesso sistema di protezione viene applicato anche nelle pagine che interagiscono con il database e non producono alcun output sullo schermo.
Così facendo si ha una protezione minima contro eventuali attacchi sia da soggetti terzi che da personale malintenzionato che vorrebbe apportare modifiche che non gli spettano.

15.1. Approfondimento tecnico

Il dato del privilegio è contenuto in un array di sessione che viene creato ogni volta che l'utente effettua l'accesso, insieme allo username, la sua password e il suo ID.
Il cosiddetto privilegio è un numero intero, espresso come somma di potenze di due, in quanto per ogni bit del codice binario posto ad 1 equivale al permesso di svolgere una determinata azione, in caso l'utente possa svolgere due o più azioni il dato è composto dalla somma dei privilegi.

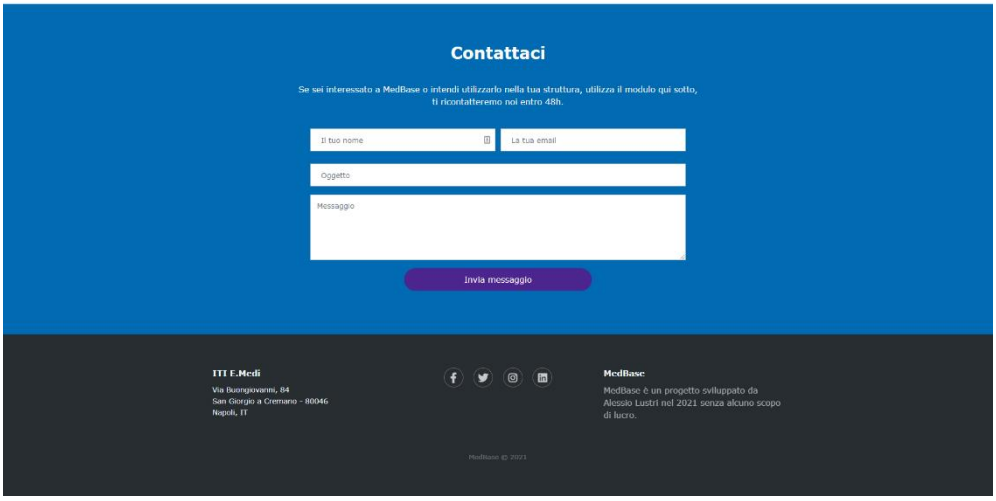
Per verificare se un utente può accedere ad una pagina, va quindi posto un controllo che tramite un AND bit a bit con il numero del privilegio richiesto, in caso entrambi i bit siano 1 allora l'utente potrà visitare la pagina, altrimenti non avrà alcuna possibilità di azione e visualizzerà un avviso.

Ad esempio, se ad una pagina è richiesta la carica di dirigente sanitario (privilegio 8) per accedere alla pagina e l'utente che tenta di accedervi ha come cariche quella di direttore sanitario e analista (8+1) allora potrà accedervi.

16. Pagine web

16.1. Home Page

La pagina è l'unica ad essere pubblica, quindi visibile anche a chi non ha un proprio account. Contiene una rapida introduzione alla piattaforma, cosa fa e cosa offre.
Immediatamente sotto vi è un form di contatto che tramite la libreria PHPmailer, invia alla mail "no-reply@alessiolustri.it" il testo scritto dall'utente formattato secondo quando scritto nel file "contatti.php".
Quest'ultima pagina riceve dal form il nome del mittente, la sua mail, l'oggetto e il corpo in sé, tutto tramite l'array POST per garantire maggiore privacy e per non far passare i dati in chiaro nell'URL.
Alla fine della pagina è presente un footer con alcune informazioni circa l'autore del sito, i suoi contatti e i profili social.
In testa alla pagina viene richiesta la connessione al DB server e poi al database specifico. In caso quest'ultimo non esista viene lanciata la funzione `creaDB` che va a crearlo, aggiungendo l'utente di admin con cui poi l'utente farà il primo accesso alla piattaforma e potrà poi configurarla secondo le sue esigenze.



16.2. Statistiche

È la pagina che richiede il privilegio di Analista per potervi accedere.
Non appena vi si accede ci si trova di fronte due "isole". La prima, a sinistra, permette di selezionare uno dei reparti e di visualizzare la capacità del reparto, il numero di posti occupati in quel momento e la durata media di un ricovero in quel reparto in base ai dati disponibili al momento della richiesta.
Questo è stato realizzato tramite la funzione "Ajax" contenuta nella libreria JQuery. La funzione prende come parametro esclusivamente l'ID del reparto, tramite il quale va ricavare i dati citati in precedenza con delle query direttamente dal database.
Una volta che ha ricevuto tutti i dati viene mostrato a schermo una tabella contenente tutte le info.

L'altra sezione, quella di destra, presenta un campo di inserimento testuale in cui inserire il codice fiscale del paziente che si vuole analizzare e similmente al caso citato in precedenza verrà mostrata una tabella contenente lo storico di tutti i ricoveri del paziente in questione prendendo le date di inizio ricovero, quelle di fine e il reparto in cui ha alloggiato.

16.3. Accettazione

La pagina, alla quale l'accesso è consentito solo al personale dell'accettazione presenta un singolo form in cui vanno inseriti nome, cognome, codice fiscale del paziente e il reparto in cui si desidera ricoverarlo. Il passaggio dei dati alla pagina "InserisciPaziente.php" avviene sempre tramite metodo POST, trattandosi di dati sensibili.
Il sistema provvederà prima di tutto a verificare se il paziente, e il suo codice fiscale, è già presente nel database, in quel caso va a prenderne l'identificativo. Altrimenti va ad inserirlo, sempre reperendone l'ID.

Fatta questa verifica, va a verificare se il paziente ha già ricoveri attivi in quel momento, come verifica anche della disponibilità del reparto indicato, in caso sia possibile allora va a creare una nuova degenza in quel reparto, inserendo la data odierna automaticamente come data di inizio e lasciando vuoto il valore della data di fine.

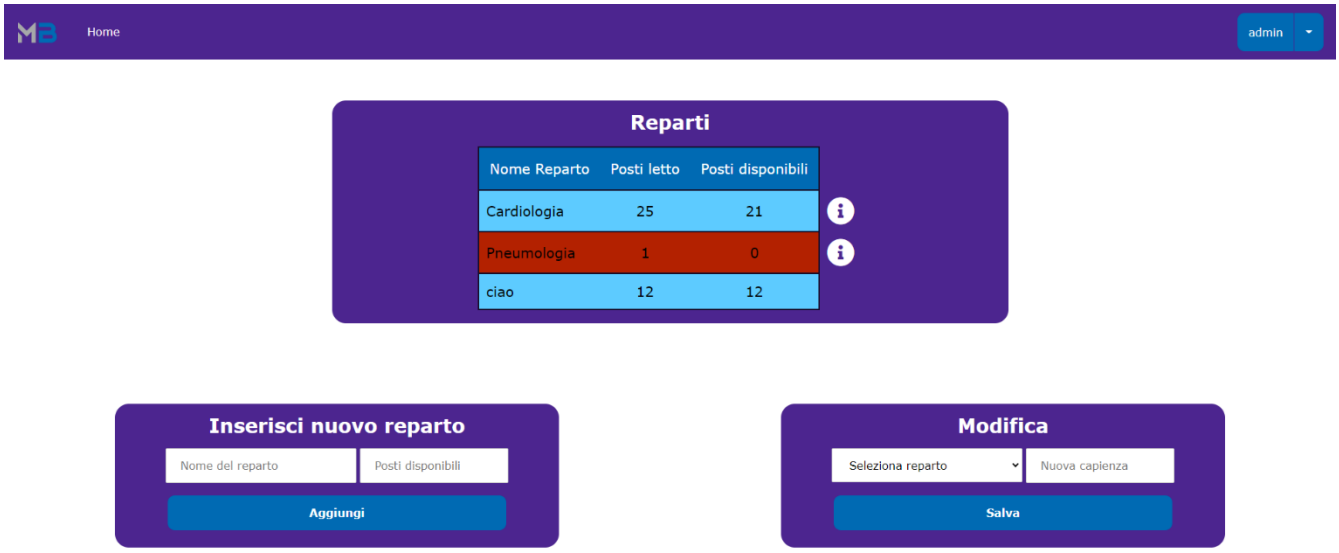
16.4. Dimissione

La pagina, alla quale ha accesso esclusivamente il personale medico e prevede un box di ricerca in cui inserire il codice fiscale del paziente. Il sistema cerca eventuali ricoveri attivi del paziente in questione e in tal caso mostra a schermo tutte le informazioni seguite da un pulsante “Dimetti” tramite la quale è possibile appunto dimettere il paziente ponendo fine al suo ricovero aggiornando il dato di fine ricovero con la data attuale.
Entrambe le azioni vengono svolte tramite la funzione Ajax e il passaggio dei parametri avviene anche questa volta con POST.

16.5. Gestione reparti

La pagina è divisa in tre zone:

- La prima in cui viene mostrata una tabella con l'elenco dei reparti, il numero di posti disponibili e il numero di quelli occupati. Accanto ad ogni riga della tabella è posto un pulsante per mostrare in sovrimpressione l'elenco dei ricoverati in quel reparto, in caso ce ne siano.
  Quando un reparto supera il 75% dei posti occupati, la riga si colorerà di rosso in modo da avere un'immediata idea sulle situazioni più critiche.
  In caso invece il reparto non sia mai stato utilizzato, viene mostrato un cestino che cliccandolo elimina appunto il reparto dal database.
  Per preservare l'integrità dei dati non sarà più possibile rimuovere un reparto una volta che viene utilizzato, anche solo per una persona. Di conseguenza, se il reparto dovesse essere dismesso sarebbe consigliato aggiornare la disponibilità dei posti a 0.
- La seconda area del sito invece riguarda la modifica della capienza dei posti che tramite una pagina "AggiornaReparto.php" esegue prima una verifica se la nuova capienza non è superiore al numero di persone attualmente ricoverate lì. Il passaggio dei dati (Identificativo del reparto e la nuova capienza) avviene sempre tramite POST.
- La terza area invece ha un semplice form che prevede l'inserimento del nome e la capienza di un nuovo reparto. Il passaggio dei dati inseriti avviene sempre con POST e in caso esista già un reparto con quel nome l'utente viene avvisato e il reparto non viene creato.



16.6. Gestione utenti

La pagina come primo elemento mostra a schermo una tabella con la lista degli utenti presenti nel database, ad eccezione dell'utente che apre la pagina.
Ogni account ha una propria serie di spunte che vanno ad ampliare o limitare il numero di pagine della piattaforma a cui quell'utente ha accesso. È inoltre anche qui possibile eliminare l'utente semplicemente cliccando sul cestino posto di fianco.

Immediatamente sotto vi è una sezione che prevede la possibilità di creazione di nuovi account, indicare i loro permessi, fornirli di un username e una password, questi ultimi due campi sono da inserire due volte per evitare errori.
Gli username degli utenti non possono essere doppi per cui il sistema esegue prima una verifica sul nuovo username, in caso non sia stato ancora utilizzato allora procede alla creazione. Tutti i dati vengono passati alla pagina "InserisciPersonale.php" tramite POST così da non rendere visibili i dati sensibili quali username e password.

16.7. Extra

All'interno del sito, in qualunque pagina l'utente si trovi, avrà la possibilità di cambiare la propria password, inserendo prima la password attuale e poi la nuova per due volte, in modo da assicurarsi che non ci siano intromissioni esterne volte a cambiare password ad un account di cui non si è proprietari.

17. Analisi delle soluzioni per la pubblicazione

Per la fruizione pubblica della piattaforma sono state trovate due soluzioni : quella di creare un proprio server da zero, partendo dall'hardware per poi configurare manualmente tutti i software, il cosiddetto "housing", oppure affidarsi ad una società terza, il provider, che offre servizi simili a pagamento.

	Vantaggi	Svantaggi
Housing	• Maggiore versatilità e adattabilità alle proprie esigenze. • Maggior controllo sulle risorse hardware. • Piena autonomia nelle decisioni relative alla sicurezza.	• Richiede competenze in vari ambiti per essere gestito. • Deve essere una macchina totalmente dedicata a quel compito, di conseguenza deve essere accesa h24, cosa che comporta dei costi e porta l'hardware ad un deterioramento prematuro. • Costi di gestione quali corrente e connessione ad Internet. • Richiede di rimanere aggiornati rispetto ai progressi tecnologici.
Hosting	• Non ci sono costi extra oltre la rata mensile, dato che sono a carico del provider. • Il provider offre anche un servizio clienti che può intervenire in caso di problemi. • Semplice da configurare, spesso tramite interfaccia grafica.	• A meno che non si tratti di un server dedicato di proprietà di terzi, che comporta costi ancor maggiori, solitamente si tratta di un host condiviso: un'unica macchina a disposizione di altre aziende, senza possibilità di sapere con chi condividiamo. • Poco flessibile riguardo le scelte hardware.

Per Medbase si è optato per un servizio di hosting proprio per la caratteristica di stabilità e sicurezza che la società tedesca Hetzner offre tramite i suoi cloud server (server condivisi ma comunque sicuri e protetti da attacchi informatici), completamente configurabili nel software, ne offre con varie caratteristiche in modo da adattarsi alle necessità di ogni persona.
Nel nostro caso il server è dotato del seguente hardware:

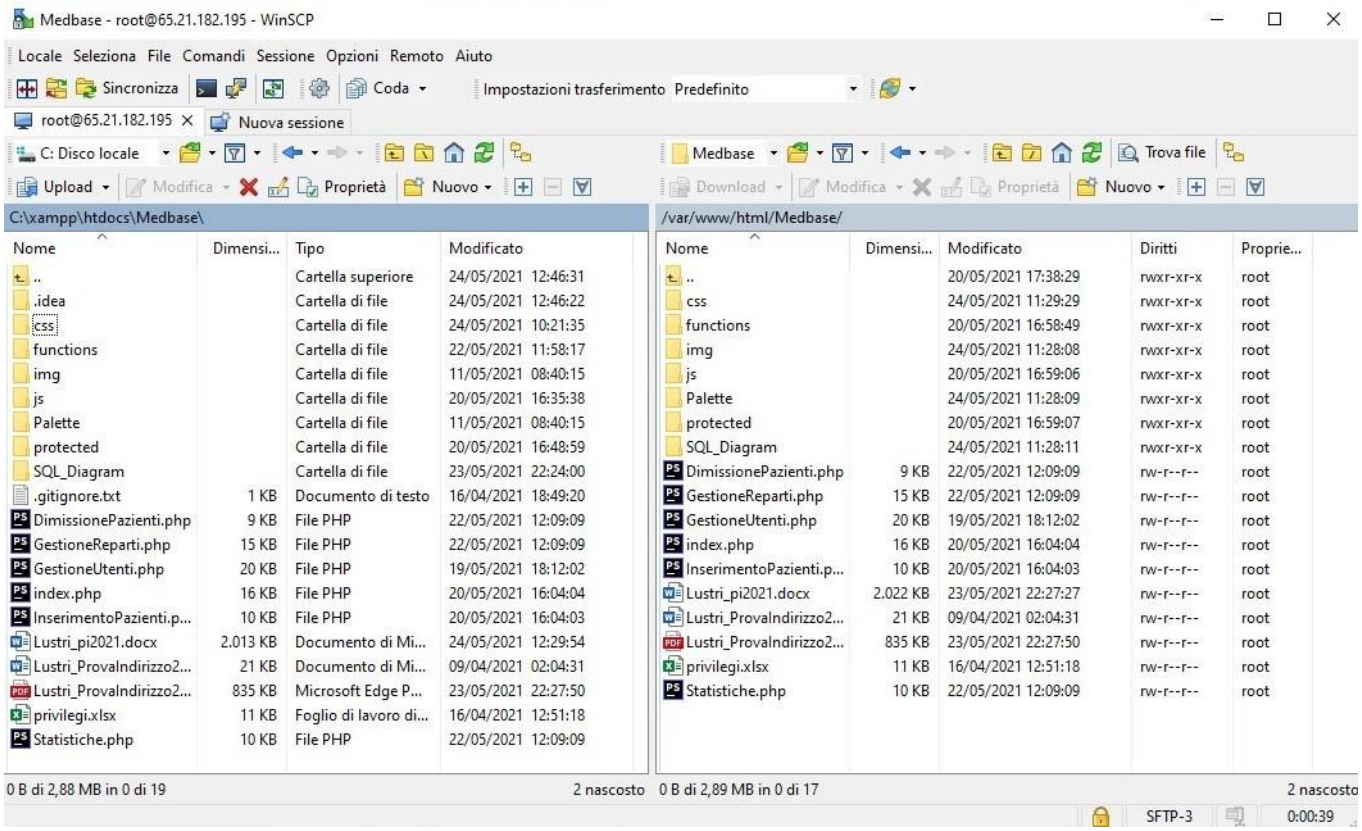
- CPU Intel Xeon Platinum 8180
- 2 GB di RAM
- 20 GB di SSD NVMe M.2
- 20 TB di traffico dati

Il server è configurabile anche nella scelta del sistema operativo, nel nostro caso si è optato per Fedora 34, distribuzione GNU/Linux sponsorizzata da Red Hat, particolarmente adatto all'utilizzo dei server per tutta la serie di tool utili a questo scopo.

Ottenuto l'IP del nostro server, che nel nostro caso è situato ad Helsinki, gli si può accedere tramite protocollo SSH, tramite il quale sono stati installati:

- PHP
- phpMyAdmin

- MariaDB
- MariaDB Server
- HTTPD (Apache)



Interfaccia grafica di WinSCP, a sinistra la cartella del progetto in locale, a destra

Configurato il database sul server, si è installato il software gratuito WinSCP, un client FTP che tramite interfaccia grafica permette di trasferire i file del sito web sul server.
Il dominio del sito web "alessiolustri.it" è stato acquistato dal registrar "register.it", il quale fornisce anche una web mail abbinata al dominio.
Per questioni di sicurezza e affidabilità si è deciso di trasferire il controllo del dominio alla società "Cloudflare" che fornisce i certificati SSL, protezione da attacchi DDOS oltre che uno dei DNS server più performanti sul mercato.
Tramite l'interfaccia di Cloudflare è stato definito il sottodominio "medbase.alessiolustri.it" ed è stato configurato il server Apache per reindirizzare tutti gli utenti che visitano dal sottodominio sopracitato alla sottocartella di Medbase subito sotto la root directory.

17.1. Protocollo FTP

Il File Transfer Protocol (abbreviato FTP) è uno dei primi protocolli, sviluppato dal MIT nel 1971 con lo scopo di permettere la condivisione di file sulla rete Internet attraverso la porta 21.
Originalmente non prevede alcuna cifratura per i dati scambiati, quindi i nomi utenti, password, comandi e codici di risposta viaggiano sulla rete in chiaro col rischio che vengano intercettati.
Il problema è stato risolto aggiungendo un layer di cifratura SSL, portando alla nascita del protocollo FTPS.

17.2. Protocollo SSH

La Secure Shell (SSH) è un protocollo che permette di stabilire una connessione remota cifrata con un altro host tramite riga di comando, sostituendo il poco sicuro Telnet.

È lo standard per l'amministrazione remota di sistemi GNU/Linux, a cui la IANA (Internet Assigned Numbers Authority) ha assegnato la porta 22 TCP e UDP.

17.3. HTTPS e SSL

Il protocollo TCP/IP sin dalla sua nascita non prevede un sistema di sicurezza a causa della natura chiusa della rete in origine, per cui il problema della sicurezza si è posto solo quando la rete internet ha iniziato ad entrare nelle case di tutti.
L'SSL è ad esempio applicato nel protocollo HTTPS, in cui tra il protocollo TCP e HTTP si intrapone un layer di crittografia o autenticazione come l'SSL appunto.
In sostanza il messaggio http viene criptato prima di essere poi inviato, e tra i due host tra i quali avviene la comunicazione viene stabilito un canale criptato in cui prima vengono scambiati i certificati che contengono le chiavi di codifica, poi tramite http avviene lo scambio di informazioni, a cui hanno accesso solo il client e il server coinvolti nella comunicazione.
Questo non rende HTTPS un protocollo a sé stante, ma è semplicemente il protocollo HTTP a cui viene aggiunto un layer di crittografia per migliorarne la sicurezza.

17.4. Attacchi DDoS

Il distributed denial of service (DDoS) è un attacco informatico volto a esaurire le risorse di un sistema che fornisce un servizio a dei client, come può essere un sito web.
A differenza dell'attacco DoS classico, nel DDoS la fonte da cui il server viene inondato è multipla e quindi è impossibile fermare un simile attacco.
Per evitare di essere individuati, gli attaccanti infettano preventivamente un numero elevato di computer con dei virus o worm che lasciano aperte delle backdoor a loro riservate, anche per avere a disposizione un numero sufficiente di computer per l'attacco. I computer che sono controllati dall'attaccante vengono chiamati zombie.

18. Primo accesso e configurazione

Quando la piattaforma viene aperta per la prima volta presso la struttura, sarà possibile accedervi esclusivamente con l'account che ha come username "**admin**" e password "**admin**". Da lì poi è possibile creare account a tutto il personale con le relative deroghe, aggiungere tutti i reparti.

19. Download

La piattaforma è disponibile al link "medbase.alessiolustri.it".
Il codice del sito è disponibile al download cliccando qui, o in allegato a questo documento.