

Elaborato per l'Esame di Stato di istruzione seconda
superiore, 2020-2021

Pardini Alessio, Classe 5A-INT
31 Maggio 2021

Indice

1 Modello relazionale	3
1.1 Ipotesi iniziali	3
1.2 Diagramma E-R	6
1.2.1 Entità	6
1.2.2 Relazioni Significative	7
2 Modello Logico	9
2.1 Diagramma Logico	9
3 DB schema MySQL	10
4 Implementazione Linguaggio PHP	16
4.1 Ipotesi iniziali	16
4.2 SQL Injection	18
4.3 Interfaccia per i pazienti	19
4.3.1 Connessione al DB (connessione_db.php)	19
4.3.2 File configurazione (config.php)	19
4.3.3 Area Paziente (areaPersonale.php)	23
4.3.4 Area Visualizzazione Cartella Clinica (privato.php)	29
4.3.5 Area Visualizzazione Esami Fatti (esamiFatti.php)	32
4.3.6 Area Visualizzazione Esami Da Fare (esamiDaFare.php)	34
4.3.7 Login (login.php)	37
4.3.8 Area Registrazione (registrazione.php)	38
4.3.9 Insert (Inasert.php)	50
4.3.10 Termini e condizioni d'uso (termini.php)	52
5 GDPR	53
6 Ipotesi APP	55
7 Web API	56
7.1 Progettazione	56
7.2 Implementazione	70

	2
8 Infrastruttura tecnologica ed informatica	78
8.1 Attori della rete	78
8.2 Collocazione Server	79
8.3 Soluzione on-premise	81
8.4 Soluzione cloud	85
8.5 Dispositivi mobili utilizzati dagli operatori addetti al servizio domiciliare . . .	85
8.6 Autenticazione	86
8.7 VPN	87

Capitolo 1

Modello relazionale

Il modello relazionale pone le sue basi sulla gestione dei dati ricondotta all'unico concetto chiave, inteso in senso algebrico, di <<relazione>>. Il modello fu introdotto da Edgar Codd nel 1970 ed ad oggi costituisce la base per molti DBMS tra i quali Oracle DB, IBM DB2, Microsoft SQL-server e MySQL. Nel contesto dell'elaborato, ovviamente, il modello relazionale è stato utilizzato per descrivere e rappresentare il quadro del problema fornito. Per quanto il testo del problema possa identificare uno scenario ben preciso, alcuni aspetti non risultano essere dettagliati, lasciando spazio così ad iniziative personali. Difatti per poter sviluppare una soluzione è necessario fare delle ipotesi ragionevoli sul contesto in analisi. La *sottosezione 1.1*¹ è perciò dedicata alla definizione delle ipotesi dalle quali dipendono il modello concettuale e logico della base di dati per la gestione del servizio.

1.1 Ipotesi iniziali

- Nello scenario delineato ogni giorno vengono programmati da parte del personale addetto presso la sede centrale dell'ASL gli interventi per ogni assistito ed i relativi operatori (medico e/o infermiere), ciò implica la definizione di entità quali <<esame>>, <<operatore>> e <<paziente>>.

L'entità esame si ipotizza che sia identificata da una coppia di chiavi primarie, 'id_esame' e 'data_prenotazione'.

Questa assunzione è giustificata dal fatto che se quest'ultima entità fosse identificata solo ed esclusivamente da un identificativo incrementale oppure da un codice univoco costituito da caratteri, nel lungo periodo si potrebbero riscontrare problematiche. Tutto ciò poiché la lunghezza sia nel caso di un identificativo incrementale che di un codice costituito da combinazioni di caratteri e numeri, potrebbe potenzialmente non essere adatta, oppure dispendiosa da memorizzare. Per questi determinati motivi si ipotizza una coppia 'id_esame', 'data_prenotazione', come chiave primaria dove il codice dell'esame incrementa in maniera sequenziale secondo la seguente sintassi('00001', '00002', ...), a fine giornata quest'ultimo si azzererà, ripartendo da '00000'.

Le entità <<operatore>> e <<paziente>> sono identificate rispettivamente da una 'matricola' per gli operatori e dal 'codice_tessera' per i pazienti. In entrambi viene tenuto conto di una generica anagrafica, che si differenzia nei paziente con l'aggiunta dei campi 'indirizzo_domicilio', 'cap_domicilio', relativi alla posizione nella quale sarà fornito l'ipotetico servizio.

¹ La sottosezione 1.1 riporterà spezzoni del testo dell'elaborato, delimitato da apposite virgolette, ed in base al quale saranno effettuate le eventuali ipotesi.

- “Le prestazioni da effettuare presso i vari assistiti sono opportunamente codificate e quantificate in un prontuario in termini di azioni previste e tempo occorrente per il loro espletamento.” A tale scopo è stata gestita un'entità denominata <<prontuario>>, identificata da 'id_prestazione' che descrive anche il prezzo e l'eventuale numero di medici e infermieri richiesti per una specifica tipologia di esame.

Dallo scenario emerso si può evidentemente notare come 'n' operatori partecipino a 'm' esami e come ognuno di questi abbia uno specifico ruolo nell'esecuzione di un determinato intervento (medico, infermiere, amministratore e per ultimo la mansione che vede protagonista l'analisi di eventuali campioni). Per questo motivo nella tabella di mezzo che si va a creare, viene memorizzato il ruolo assunto dallo specifico operatore durante l'esecuzione di un determinato esame.

- “Alcune di queste prevedono il rilievo di parametri quali la temperatura corporea, la pressione sanguigna, ecc. da inserire contestualmente nella cartella clinica elettronica del paziente ...” l'entità <<cartellaClinica>> è identificata da 'id_cartella_clinica' e dal 'codice_tessera', specifico per un determinato paziente, questa assunzione è data dal medesimo ragionamento spiegato precedentemente per la coppia di chiavi primarie presente in esame. L'entità in questione è delineata dai campi 'data_inizio' e 'data_fine', infatti un paziente detiene più cartelle cliniche che vengono aperte in una determinata data e chiuse a fine percorso. Inoltre in questa entità possiamo trovare i seguenti campi 'cf_medico_curante', 'temperatura_gradi', 'peso_kg', 'descrizione'.

- “Nel caso in cui sia prevista l'effettuazione di prelievi (sangue, urine, ecc.) i campioni sono raccolti in provette identificate da un codice a barre prestampato...” Si ipotizza quindi a tale scopo la delineazione di un'entità <<provetta>>, identificata dal campo 'id_provetta', costituito da un codice a barre prestampato. Per ogni provetta memorizzata all'interno della struttura dati, viene preso nota dell'esame nella quale eventualmente è stata utilizzata, del laboratorio al quale viene trasportata per le analisi, di conseguenza sarà presente anche un riferimento agli eventuali risultati delle stesse e di alcuni campi descrittivi della provetta. Come si può ben notare dal contesto sopra descritto entra in gioco una nuova entità quella del <<laboratorio>>, la quale risulta essere identificata da 'id_laboratorio' e viene tenuto conto nella struttura dati di alcune informazioni relativi alla posizione.

Nello scenario delineato dal testo del problema i risultati delle analisi saranno memorizzate nel database, sfruttando il sito Web reso fruibile dall'ASL in questione. Perciò si ipotizza la presenza nel modello relazionale di un'entità <<analisi>>, identificata semplicemente da 'id_analisi'.

- “...gli operatori redigono un report con i riferimenti temporali (data/ora) di inizio/fine intervento e una descrizione di ciò che è stato fatto.” Nel modello relazionale definito viene ipotizzata un'entità <<report>>, la quale è resa univoca dal medesimo identificativo dell'esame, presentando i riferimenti temporali, una piccola descrizione ed in particolare l'identificativo dell'operatore che ha redatto il report; infatti si ipotizza che anche in una circostanza che vede

protagonista la presenza di ‘*n*’ operatori sul luogo del domicilio e quindi operativi nel contesto di un determinato esame, uno singolo tra questi si assume la responsabilità di redigere il report.

La definizione di un'entità <<report>> è stata dettata da un ragionamento effettuato nel caso di ipotesi contraria. Difatti qualora non avessi un'entità <<report>>, si potrebbe aver ipotizzato di gestire il report come semplice campo dell'entità Esame. In questo caso però dato dallo scenario presentato dal problema che vede un evidente relazione M a N tra le entità <<esame>> e <<operatore>>, si otterrebbe il fatto che ogni operatore nel contesto di un singolo esame redigerebbe un report, quindi nel caso in cui vi fossero più operatori per un singolo esame, si avrebbero più report, scontrandosi così con la logica dettata dallo scenario presentato. Pertanto la presenza di questa entità, rappresenta la soluzione più ragionevole secondo anche le restanti ipotesi generali, poste alla base del modello relazionale fornito in seguito.

- “...il centro di controllo potrà, conoscendo in ogni istante la posizione sul territorio dei vari operatori, allertare quello/i più idoneo/i per professionalità e vicinanza per interventi estemporanei...”. Come si può osservare da quanto riportato, una delle funzionalità generali del servizio richiede la conoscenza della specializzazione professionale di un determinato medico/infermiere, a tale scopo nel modello logico è presente una relazione ‘*m*’ a ‘*n*’ tra l'entità <<prontuario>> e <<operatore>> che formerà a sua volta un'entità <<prontuarioOperatore>> .

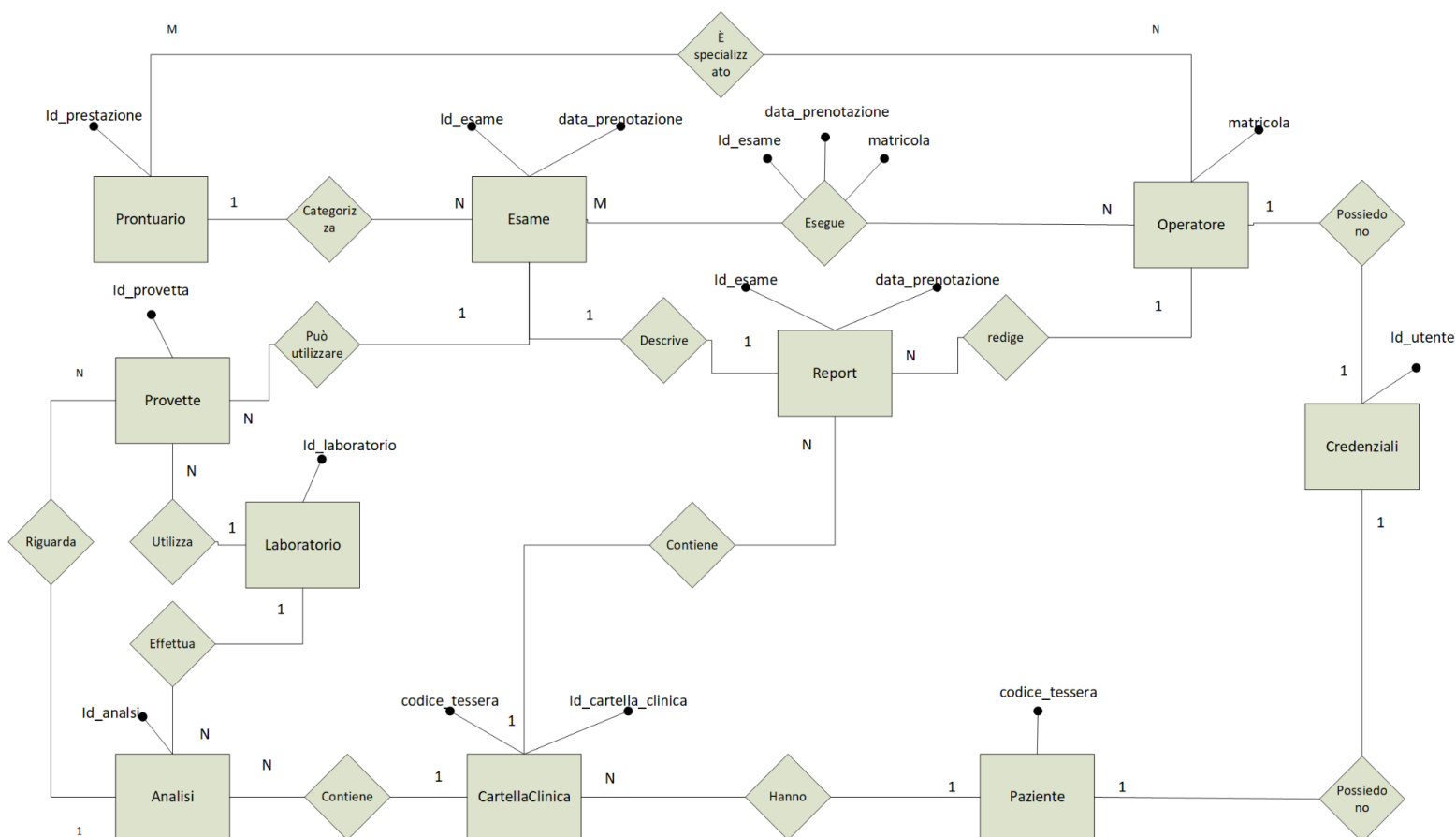
- Si ipotizza inoltre che l'asl in questione metta a disposizione un proprio sito fruibile dai clienti e dal personale, in modo da fornire un servizio migliore ed efficace. Per questo motivo il modello relazionale presenterà un'entità apposita, <<credenziali>>, nella quale saranno memorizzate l'email ed il risultato hashing della password, ciò rende la password effettiva non recuperabile da un ipotetico attacco hacker, che nella migliore delle ipotesi potrà risalire soltanto al valore dell'hash. Al momento del login, come potremmo vedere successivamente dal codice, sarà eseguito l'hash della password appena digitata e confrontato con il valore hash memorizzato per la verifica.

L'entità credenziali è identificata da un 'id_utente' che nel caso degli operatori sarà riconducibile semplicemente alla matricola, mentre nel caso del paziente sarà riconducibile al codice della tessera sanitaria. Proprio per questo motivo l'identificativo delle credenziali è stato definito come *varchar(20)*, per adattarsi sia al paziente dove il codice della tessera risulta essere un *varchar(20)* che all'operatore dove la matricola è un semplice *varchar(5)*.

- Si ipotizza che il modello non debba includere gli aspetti logistici legati al trasporto del personale sul luogo del domicilio.

1.2 Diagramma E-R

In questa sezione come si può vedere viene riportato il diagramma E-R che descrive il problema fornito.



1.2.1 Entità

- **Prontuario:** Rappresenta una lista contenente gli esami effettuabili, ordinato in modo da renderne agevole la ricerca e la consultazione.
- **Esame:** Rappresenta l'intervento a domicilio compiuto da determinati operatori e destinato ad un eventuale paziente.
- **Operatore:** Rappresenta lo specifico personale dell'asl, che assume un ruolo in un esame e viene descritto da una generica anagrafica.

- **ProntuarioOperatore:** Rappresenta l'entità formata dalla relazione M a N tra operatore e prontuario, che permette di memorizzare la specializzazione professionale di un determinato operatore utile alle funzionalità del servizio.
- **EsameOperatore:** Rappresenta l'entità formata dalla relazione M a N tra operatore ed esame, che descrive il ruolo che assume un operatore nel contesto di un esame.
- **Report:** Rappresenta il report descrittivo di ciò che è stato fatto, che a fine esame/intervento viene redatto da un operatore.
- **Paziente:** Rappresenta il cliente del servizio, che prenoterà uno specifico intervento.
- **Credenziali:** Rappresenta le credenziali di ogni paziente e operatore registrato al sito reso fruibile dell'asl.
- **CartellaClinica:** Rappresenta la cartella clinica di un paziente che viene aperta e chiusa in determinate date. Quest'ultima risulta essere dipendente dalla "traumatologia", ciò significa che in un contesto ipotetico, un determinato paziente potrà avere due cartelle cliniche aperte nel medesimo giorno, una dovuta ad una frattura della tibia ed un'altra dovuta ad una lesione al tendine della spalla.
- **Provetta:** Rappresenta la provetta fisica che viene utilizzata per contenere i campioni prelevati e trasportata ai laboratori per le analisi.
- **Laboratorio:** Rappresenta il laboratorio fisico nel quale vengono effettuate le analisi dei campioni.
- **Analisi:** Rappresenta il risultato delle analisi effettuate in un determinato laboratorio ed appartenenti ad una determinata cartella clinica.

1.2.2 Relazioni significative

In questa sezione vengono delineate le relazioni più significative.

- **Relazione 1 a N tra Cartella clinica e report:** In base anche allo scenario definito in precedenza, una cartella clinica si ipotizza che contenga al proprio interno più report dati dall'effettuazione di più esami dovuti ad una determinata "traumatologia". Su tale ipotesi nasce questa relazione che permetterà all'utente di controllare dalla sua cartella clinica gli eventuali report, gli esami svolti e da svolgere.

- **Relazione 1 a N tra operatore e report:** Un operatore non redige più report nel contesto di un esame, bensì è ovvio che può redigere più report nel contesto di più esami.
- **Relazione 1 a N tra Provette ed esame:** La relazione 1 a N tra le seguenti entità nasce dal fatto che per ogni singolo esame possono essere effettuati dei prelievi (sangue, urine, ecc.), dove i campioni sono raccolti in apposite provette, di conseguenza può essere previsto che ogni singolo esame possa richiedere l'utilizzo di 'n' provette.
- **Relazione 1 a N tra Analisi e Provette:** La relazione seguente deriva dalla considerazione per la quale ogni risultato di un'analisi è frutto di un'esaminazione di più provette. Questa relazione si crea anche per risalire, tramite interrogazione al database, al determinato esame nel quale sono stati effettuati i campioni analizzati in input.
- **Relazione M a N tra Operatore e Prontuario:** La relazione descritta nasce dalla considerazione che ogni operatore possa essere specializzato professionalmente in più tipologie di esame e che ogni tipologia possa essere oggetto di specializzazione per ogni operatore.

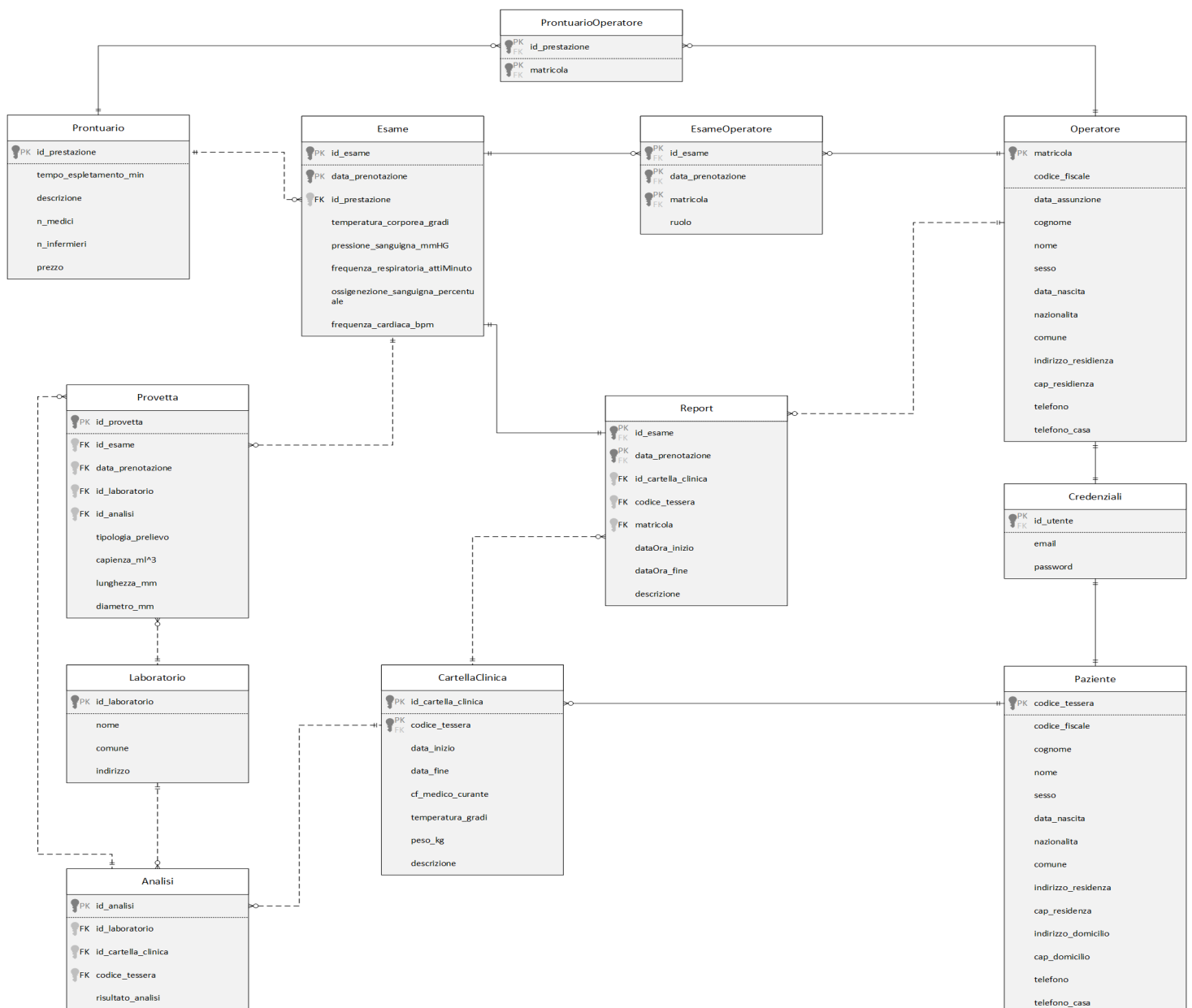
Capitolo 2

Modello logico

In questa sezione viene riportato il modello logico derivato dal precedente modello E-R.

Si può notare che i campi riguardanti la misurazione di un parametro, contengono nel loro nome la dicitura finale “_unitàDiMisura”, a scopo di dare un senso logico ai valori riportati e salvati all’interno della struttura dati.

2.1 Diagramma logico



Capitolo 3

DB schema MySQL

In questa sezione viene presentato il codice per la definizione della struttura dati in linguaggio SQL, facendo riferimento alle caratteristiche del DBMS MySQL/MariaDB.

```
--
-- Database: `asl`
--
-----
--
-- Struttura della tabella `Esame`
--
CREATE TABLE `esame` (
  `id_esame` varchar(5) NOT NULL DEFAULT '',
  `data_prenotazione` dateTime NOT NULL,
  `id_prestazione` varchar(4) NOT NULL,
  `temperatura_corporea_gradi` double DEFAULT NULL,
  `pressione_sanguigna_mmHG` double DEFAULT NULL,
  `frequenza_respiratoria_attiMinuto` double DEFAULT NULL,
  `ossigenazione_sanguigna_percentuale` int(11) DEFAULT NULL,
  `frequenza_cardiaca_bpm` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Prontuario`
--
CREATE TABLE `prontuario` (
  `id_prestazione` varchar(4) NOT NULL DEFAULT '',
  `tempo_espletamento_min` int(11) DEFAULT NULL,
  `descrizione` text DEFAULT NULL,
  `n_medici` int(11) DEFAULT NULL,
  `n_infermieri` int(11) DEFAULT NULL,
  `prezzo` double NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `EsameOperatore`
--
CREATE TABLE `esameoperatore` (
  `id_esame` varchar(5) NOT NULL DEFAULT '',
  `data_prenotazione` dateTime NOT NULL,
  `matricola` varchar(5) NOT NULL DEFAULT '',
  `ruolo` varchar(16) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Operatore`
--
CREATE TABLE `operatore` (
  `matricola` varchar(5) NOT NULL DEFAULT '',
  `codice_fiscale` varchar(16) DEFAULT NULL,
  `data_assunzione` date DEFAULT NULL,
  `cognome` varchar(40) DEFAULT NULL,
  `nome` varchar(40) DEFAULT NULL,
  `sesso` varchar(1) DEFAULT NULL,
  `data_nascita` date DEFAULT NULL,
```

```

    `nazionalita` varchar(30) DEFAULT NULL,
    `comune` varchar(20) DEFAULT NULL,
    `indirizzo_residenza` varchar(30) DEFAULT NULL,
    `cap_residenza` int(5) DEFAULT NULL,
    `telefono` varchar(10) DEFAULT NULL,
    `telefono_casa` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `ProntuarioOperatore`
--
CREATE TABLE `prontuariooperatore` (
  `id_prestazione` varchar(4) NOT NULL DEFAULT '',
  `matricola` varchar(5) NOT NULL DEFAULT ''
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Credenziali`
--
CREATE TABLE `credenziali` (
  `id_utente` varchar(20) NOT NULL DEFAULT '',
  `email` varchar(100) DEFAULT NULL,
  `password` varchar(150) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Paziente`
--
CREATE TABLE `paziente` (
  `codice_tessera` varchar(20) NOT NULL DEFAULT '',
  `codice_fiscale` varchar(16) DEFAULT NULL,
  `cognome` varchar(40) DEFAULT NULL,
  `nome` varchar(40) DEFAULT NULL,
  ` Sesso` varchar(1) DEFAULT NULL,
  `data_nascita` date DEFAULT NULL,
  `nazionalita` varchar(30) DEFAULT NULL,
  `comune` varchar(20) DEFAULT NULL,
  `indirizzo_residenza` varchar(30) DEFAULT NULL,
  `cap_residenza` int(5) DEFAULT NULL,
  `indirizzo_domicilio` varchar(30) DEFAULT NULL,
  `cap_domicilio` int(5) DEFAULT NULL,
  `telefono` varchar(10) DEFAULT NULL,
  `telefono_casa` varchar(10) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Report`
--
CREATE TABLE `report` (
  `id_esame` varchar(5) NOT NULL DEFAULT '',
  `data_prenotazione` dateTime NOT NULL,
  `id_cartella_clinica` varchar(5) NOT NULL,
  `codice_tessera` varchar(20) NOT NULL,
  `matricola` varchar(5) DEFAULT NULL,
  `dataOra_inizio` dateTime DEFAULT NULL,
  `dataOra_fine` dateTime DEFAULT NULL,
  `descrizione` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `provetta`
--
CREATE TABLE `provetta` (
  `id_provetta` varchar(16) NOT NULL DEFAULT '',

```

```

`id_esame` varchar(5) NOT NULL,
`data_prenotazione` dateTime NOT NULL,
`id_laboratorio` varchar(5) NOT NULL,
`id_analisi` varchar(10) NOT NULL,
`tipologia_prelievo` varchar(30) DEFAULT NULL,
`capienza_ml^3` double DEFAULT NULL,
`lunghezza_mm` double DEFAULT NULL,
`diametro_mm` double DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Laboratorio`
--
CREATE TABLE `laboratorio` (
  `id_laboratorio` varchar(5) NOT NULL DEFAULT '',
  `nome` varchar(30) DEFAULT NULL,
  `comune` varchar(30) DEFAULT NULL,
  `indirizzo` varchar(5) DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `Analisi`
--
CREATE TABLE `analisi` (
  `id_analisi` varchar(10) NOT NULL DEFAULT '',
  `id_laboratorio` varchar(5) NOT NULL,
  `id_cartella_clinica` varchar(5) NOT NULL,
  `codice_tessera` varchar(20) NOT NULL,
  `risultato_analisi` text NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Struttura della tabella `CartellaClinica`
--
CREATE TABLE `cartellaClinica` (
  `id_cartella_clinica` varchar(5) NOT NULL DEFAULT '',
  `codice_tessera` varchar(20) NOT NULL DEFAULT 0,
  `data_inizio` date DEFAULT NULL,
  `data_fine` date DEFAULT NULL,
  `cf_medico_curante` varchar(16) DEFAULT NULL,
  `temperatura_gradi` double DEFAULT NULL,
  `peso_kg` double DEFAULT NULL,
  `descrizione` text DEFAULT NULL
) ENGINE=InnoDB DEFAULT CHARSET=latin1;
--
-- Indici per le tabelle scaricate
--
-- Indici per le tabelle `esame`
--
ALTER TABLE `esame`
  ADD PRIMARY KEY (`id_esame`,`data_prenotazione`),
  ADD KEY `EsameProntuario` (`id_prestazione`);
--
-- Indici per le tabelle `prontuario`
--
ALTER TABLE `prontuario`
  ADD PRIMARY KEY (`id_prestazione`);
--
-- Indici per le tabelle `esameoperatore`
--
ALTER TABLE `esameoperatore`

```

```

    ADD PRIMARY KEY (`id_esame`,`data_prenotazione`,`matricola`);
--
-- Indici per le tabelle `operatore`
--
ALTER TABLE `operatore`
    ADD PRIMARY KEY (`matricola`);
--
-- Indici per le tabelle `prontuariooperatore`
--
ALTER TABLE `prontuariooperatore`
    ADD PRIMARY KEY (`id_prestazione`,`matricola`);
--
-- Indici per le tabelle `credenziali`
--
ALTER TABLE `credenziali`
    ADD PRIMARY KEY (`id_utente`);
--
-- Indici per le tabelle `paziente`
--
ALTER TABLE `paziente`
    ADD PRIMARY KEY (`codice_tessera`);
--
-- Indici per le tabelle `report`
--
ALTER TABLE `report`
    ADD PRIMARY KEY (`id_esame`,`data_prenotazione`),
    ADD KEY `ReportCartellaClinica` (`id_cartella_clinica`,`codice_tessera`),
    ADD KEY `ReportOperatore` (`matricola`);
--
-- Indici per le tabelle `provetta`
--
ALTER TABLE `provetta`
    ADD PRIMARY KEY (`id_provetta`),
    ADD KEY `provettaEsame` (`id_esame`,`data_prenotazione`),
    ADD KEY `provettaLaboratorio` (`id_laboratorio`);
--
-- Indici per le tabelle `laboratorio`
--
ALTER TABLE `laboratorio`
    ADD PRIMARY KEY (`id_laboratorio`);
--
-- Indici per le tabelle `analisi`
--
ALTER TABLE `analisi`
    ADD PRIMARY KEY (`id_analisi`),
    ADD KEY `analisiCartellaClinica` (`id_cartella_clinica`,`codice_tessera`),
    ADD KEY `analisiLaboratorio` (`id_laboratorio`);
--
-- Indici per le tabelle `cartellaClinica`
--
ALTER TABLE `cartellaClinica`
    ADD PRIMARY KEY (`id_cartella_clinica`,`codice_tessera`);
--
-- Limiti per le tabelle scaricate
--
--
-- Limiti per la tabella `esame`
--
ALTER TABLE `esame`

```

```

    ADD CONSTRAINT `EsameProntuario` FOREIGN KEY (`id_prestazione`) REFERENCES `prontuario`
(`id_prestazione`);
--
-- Limiti per la tabella `esameoperatore`
--
ALTER TABLE `esameoperatore`
    ADD CONSTRAINT `EsameoperatoreEsame` FOREIGN KEY (`id_esame`,`data_prenotazione`) REFERENCES `esame`
(`id_esame`,`data_prenotazione`),
    ADD CONSTRAINT `EsameoperatoreOperatore` FOREIGN KEY (`matricola`) REFERENCES `operatore`
(`matricola`);
--
-- Limiti per la tabella `operatore`
--
ALTER TABLE `operatore`
    ADD CONSTRAINT `operatoreCredenziali` FOREIGN KEY (`matricola`) REFERENCES `credenziali`
(`id_utente`);
--
-- Limiti per la tabella `paziente`
--
ALTER TABLE `paziente`
    ADD CONSTRAINT `pazienteCredenziali` FOREIGN KEY (`codice_tessera`) REFERENCES `credenziali`
(`id_utente`);
--
-- Limiti per la tabella `prontuariooperatore`
--
ALTER TABLE `prontuariooperatore`
    ADD CONSTRAINT `prontuariooperatoreProntuario` FOREIGN KEY (`id_prestazione`) REFERENCES `prontuario`
(`id_prestazione`),
    ADD CONSTRAINT `prontuariooperatoreOperatore` FOREIGN KEY (`matricola`) REFERENCES `operatore`
(`matricola`);
--
-- Limiti per la tabella `report`
--
ALTER TABLE `report`
    ADD CONSTRAINT `ReportCartellaClinica` FOREIGN KEY (`id_cartella_clinica`,`codice_tessera`)
REFERENCES `cartellaClinica` (`id_cartella_clinica`,`codice_tessera`),
    ADD CONSTRAINT `ReportOperatore` FOREIGN KEY (`matricola`) REFERENCES `operatore` (`matricola`),
    ADD CONSTRAINT `ReportEsame` FOREIGN KEY (`id_esame`,`data_prenotazione`) REFERENCES `esame`
(`id_esame`,`data_prenotazione`);
--
-- Limiti per la tabella `provetta`
--
ALTER TABLE `provetta`
    ADD CONSTRAINT `provettaEsame` FOREIGN KEY (`id_esame`,`data_prenotazione`) REFERENCES `esame`
(`id_esame`,`data_prenotazione`),
    ADD CONSTRAINT `provettaLaboratorio` FOREIGN KEY (`id_laboratorio`) REFERENCES `laboratorio`
(`id_laboratorio`);
--
-- Limiti per la tabella `analisi`
--
ALTER TABLE `analisi`
    ADD CONSTRAINT `analisiCartellaClinica` FOREIGN KEY (`id_cartella_clinica`,`codice_tessera`)
REFERENCES `cartellaClinica` (`id_cartella_clinica`,`codice_tessera`),
    ADD CONSTRAINT `analisiLaboratorio` FOREIGN KEY (`id_laboratorio`) REFERENCES `laboratorio`
(`id_laboratorio`);
--
-- Limiti per la tabella `cartellaClinica`
--
ALTER TABLE `cartellaClinica`

```

```
ADD CONSTRAINT `cartellaClinicaPaziente` FOREIGN KEY (`codice_tessera`) REFERENCES `paziente`  
(`codice_tessera`);  
COMMIT;
```


Capitolo 4

Implementazione Linguaggio PHP

In questo capitolo è riportato l'aspetto implementativo del linguaggio PHP, in tutti gli script che seguono, sono state appositamente omesse le porzioni riguardanti lo stile, in modo da rendere il codice più breve e leggibile.

Il codice completo è disponibile al seguente repository:

[<https://drive.google.com/drive/folders/1NtQDIVEVfi7q7zFAz-o8haj5CdkPqFaC?usp=sharing>]

4.1 Ipotesi iniziali

In questa sezione vengono poste le principali ipotesi relative alla descrizione del sito.

Come riportato in precedenza si ipotizza che l'ASL in questione metta a disposizione un proprio sito fruibile sia dai clienti, che dal personale, ospitato su un server web che consenta l'interazione con la base di dati e le varie web application disponibili.

Il sito provvederà una schermata iniziale nella quale si potranno vedere principalmente le novità relative al contesto dell'ASL ed i contatti, inoltre vi sarà una sezione apposita di login preceduta da una schermata nella quale si richiede se l'utente in questione del sito sia un medico/infermiere, un paziente, un operatore amministrativo oppure un operatore del laboratorio. Una volta effettuato l'accesso, l'utente che identifica il medico/infermiere avrà la possibilità di visualizzare gli eventuali interventi fatti/da fare, troverà i contatti per richiedere delle informazioni date da necessità specifiche ed eventualmente vi saranno anche delle sezioni relative alle informazioni di un determinato paziente. Queste funzionalità saranno rese disponibili anche attraverso l'utilizzo di apposite API.

Per quanto riguarda l'utente che identifica l'operatore del laboratorio, avrà principalmente la possibilità di memorizzare nella struttura dati predisposta i risultati delle analisi; anche questa funzionalità sarà resa disponibile tramite l'utilizzo di API.

Per quanto riguarda l'utente del sito che identifica il paziente, quest'ultimo potrà visualizzare la sua anagrafica, con la possibilità di modificarla in caso di campi inseriti in maniera scorretta durante la registrazione, gli esami fatti/da fare, i dati ed il contenuto relativo alle sue cartelle cliniche, quindi anche eventualmente il risultato delle analisi ed i vari report. Inoltre il paziente potrà prenotare un determinato intervento specificando la data di prenotazione, la fascia oraria e la tipologia. Ovviamente potrà scegliere la data e l'ora in base alla disponibilità.

Sarà possibile eccezionalmente richiedere un intervento urgente, in base a ciò vi sarà un software specifico che allerverà tramite APP il personale disponibile, adatto e più vicino; Utilizzando un sistema di georeferenziazione, in base al quale sarà calcolato il tempo impiegato dall'operatore per raggiungere il luogo di domicilio.

Si ipotizza inoltre che il paziente possa prenotare un determinato servizio anche attraverso contatti telefonici predisposti, in tal caso l'operatore amministrativo tramite l'accesso al sito potrà registrare un nuovo esame. Questa area permetterà all'operatore amministrativo le seguenti funzioni:

- Allo scopo di soddisfare l'esigenza dettata dal testo del problema dove ogni giorno vengono programmati da parte del personale addetto presso la sede centrale dell'ASL gli interventi per ogni assistito ed i relativi operatori (medico e/o infermiere), Il personale amministrativo potrà registrare nel database il ruolo che andrà assumere un determinato operatore per uno specifico esame. In seguito verrà avvisato in automatico il personale medico tramite email.
- L'operatore potrà registrare una nuova tipologia esame nel database.
- L'operatore potrà registrare una nuova provetta nel database.
- L'operatore potrà registrare una nuova cartella clinica nel database.
- L'operatore potrà registrare un nuovo laboratorio nel database.
- L'operatore potrà registrare un nuovo paziente, in caso quest'ultimo non riuscisse a registrarsi dal sito.
- L'operatore potrà registrare i risultati di eventuali analisi, nel caso di problematiche nelle sedi periferiche.
- L'operatore potrà inserire un report, nel caso in cui il dispositivo disponibile dall'operatore sanitario addetto al servizio domiciliare presenti un guasto temporaneo.
- L'operatore potrà richiedere i dati di tutti gli elementi appena citati.
- Quest'area permetterà anche altre funzioni più specifiche di acquisizione dati, perfino a scopo statistico, sfruttando rappresentazioni su excel.

L'area in questione si baserà sull'utilizzo di API specifiche che verranno presentate di seguito.

Considerando l'inserimento dei dati in due tabelle differenti e di conseguenza l'utilizzo di due query diverse, sarà necessario prevedere la possibilità di un eventuale errore in una delle due, che potrebbe portare a inconsistenze nei dati. Per evitare questa problematica, si prevede l'impiego di transazioni con le quali ci assicuriamo di eseguire le query in modalità all-or-nothing, facendo sì che i dati, se inseriti, siano sempre completi.

4.2 SQL Injection

La sicurezza è sicuramente uno degli aspetti fondamentali di un sistema informatico. E' per questo che è stata posta alla base anche in questo progetto, infatti lo script php che interagisce con la struttura dati utilizza delle specifiche metodologie per proteggersi da eventuali attacchi (SQL injection).

SQL injection è una delle tecniche di web hacking più comuni, nello specifico rappresenta una tecnica di code injection, utilizzata per attaccare applicazioni che gestiscono dati attraverso database relazionali sfruttando il linguaggio SQL. Sfrutta le vulnerabilità di sicurezza del codice applicativo che si collega alla fonte dati SQL, ad esempio, utilizzando il mancato filtraggio dell'input dell'utente (e.s. 'caratteri di escape' nelle stringhe SQL), difatti le stringhe SQL saranno come ben sappiamo eseguite dall'applicazione server; grazie a questo meccanismo è possibile far eseguire comandi SQL anche molto complessi, che permettono agli attaccanti di modificare dati esistenti, causare *repudiation issues*² come l'annullamento di transazioni o la modifica dei bilanci, ottenere tutti i dati sul sistema, eliminare o rendere inaccessibile dei dati, e creare o compromettere gli utenti (e.s. amministratore) del database server.

SQL Injection ad oggi è più conosciuto come un attacco destinato ad applicazioni web, ma è anche utilizzato per effettuare attacchi a qualsiasi altro tipo di applicazione che impieghi in modo non sicuro database SQL.

In *uno studio del 2012*³, è stato osservato che in media le applicazioni web ricevono 4 attacchi al mese, ed i rivenditori ricevono il doppio degli attacchi rispetto alle industrie.

La tecnica di difesa utilizzata per attenuare le conseguenze di un SQL injection all'interno del progetto, sono state le query parametrizzate. Con questo tipo di approccio prima di passare i parametri alla query viene definita la sintassi delle istruzioni, distinguendo il codice dai dati indipendentemente dall'input ricevuto. In tal modo si assicura che non sia possibile alterare la finalità di una query attraverso interventi malevoli esterni.

Tale format prevede tre passaggi principali :

1. la preparazione: viene creato un template di istruzione SQL (si utilizzano dei parametri segnaposto "?" "placeholder" per indicare i valori dei dati da specificare) ed inviato al database tramite `'mysql_stmt_prepare($stmt,$query)'`.
2. l'analisi: il template dell'istruzione sql ricevuto dal db viene analizzato, compilato e memorizzato senza eseguirlo, tramite `'mysql_stmt_bind_param($stmt,"ss",$parametro1,$parametro2)'`. Vengono associati come si

² Un attacco di ripudio si verifica quando un'applicazione o un sistema non adotta controlli per tracciare e registrare correttamente le azioni degli utenti, consentendo così la manipolazione dannosa o falsificando l'identificazione di nuove azioni.

³Uno studio del 2012: Imperva Web Application Attack Report.

può vedere le variabili con i segnaposti ed il tipo di dato atteso, rispettando rigorosamente l'ordine di posizione. In questo caso le variabili \$parametro1 e \$parametro2 attesi devono essere delle stringhe 'ss'. Altri tipi previsti sono: 'i' -> tipo dato numero intero, 'd' -> tipo dato numeri double, 'b' -> tipo di dato binario.

3. l'esecuzione: l'applicazione, una volta associati i valori ai parametri, esegue l'istruzione tramite `$result = mysqli_stmt_execute($stmt);`

4.3 Interfaccia per i pazienti

4.3.1 Connessione al DB (connessione_db.php)

In questa sezione è presentato il codice del file che permette la connessione al db.

```
<?php
$conn = mysqli_connect("localhost", "root", "", "asl");
if (mysqli_connect_errno($conn)){
    die("Errore di connessione al DBMS My-SQL.");
}
?>
```

4.3.2 File configurazione (config.php)

In questa sezione è riportato il codice del file di configurazione.

```
<?php
//dati generali
$sito_internet = "Targetweb";
$data =(date("d-m-y"));
$vers = "2.0";
//URL sito
$base_url = "http://localhost/test";
//connessione DB
$host = "localhost";
$db_user = "root";
$db_psw = "";
$db_name = "asl";
?>
```

4.3.3 Area Paziente (areaPersonale.php)

Il seguente codice, contenuto nel file areaPersonale.php, descrive la pagina principale del sito dell'asl orientato ai pazienti. Nella parte iniziale vi sarà un semplice controllo che permette nel

caso in cui un utente non abbia ancora effettuato l'accesso, di essere reindirizzato alla schermata di login. In seguito vi saranno due query, una per ottenere tutti i dati del paziente che ha effettuato l'accesso, in modo da visualizzare un'anagrafica, una per ottenere i dati relativi alle cartelle cliniche del paziente. I dati ricavati dalla seconda query permetteranno di visualizzare, tramite un reindirizzamento in altre pagine, gli eventuali report, risultati delle analisi, esami da fare ed esami già fatti.

Nell'implementazione sotto riportata non è presente l'aspetto che prevede una sezione specifica per il paziente dove quest'ultimo possa prenotare un esame, scegliendo la tipologia, la data e la fascia oraria. Tutto ciò sarebbe stato possibile interrogando opportunamente il database per verificare la disponibilità ed in seguito attraverso un ulteriore query inserire l'esame, all'interno del DB.

```
<?php
session_start();
//controllo
if($_SESSION['abilitato'] != 1){
    echo("<script language=javascript> alert('Effettua il login, prima di accedere alla pagina.');

```

```

else{
    mysqli_stmt_bind_param($stmtCartella,"s",$codice_tessera);
    mysqli_stmt_execute($stmtCartella);
}
$resultCartellaClinica = mysqli_stmt_get_result($stmtCartella);

?>

<html>
<head>
    <title>Area Personale</title>
</head>
<body>
<!--Visualizzazione anagrafica-->
    <h4>Area Personale <?php echo($riga['nome']. " "); echo($riga['cognome']);?></h4>
    <div class="datiPersonali">
        <label>Codice Tessera: <?php echo($riga['codice_tessera']. " ");?></label><br>
        <label>Codice Fiscale: <?php echo($riga['codice_fiscale']. " ");?></label><br>
        <label>Sesso: <?php echo($riga['sesso']. " ");?></label><br>
        <label>Data di nascita: <?php echo($riga['data_nascita']. " ");?></label><br>
        <label>Nazionalita: <?php echo($riga['nazionalita']. " ");?></label><br>
        <label>Comune: <?php echo($riga['comune']. " ");?></label><br>
        <label>Indirizzo di residenza: <?php echo($riga['indirizzo_residenza']. " ");?></label><br>
        <label>Cap di residenza: <?php echo($riga['cap_residenza']. " ");?></label><br>
        <label>Indirizzo di domicilio: <?php echo($riga['indirizzo_domicilio']. " ");?></label><br>
        <label>Cap di domicilio: <?php echo($riga['cap_domicilio']. " ");?></label><br>
        <label>Telefono: <?php echo($riga['telefono']. " ");?></label><br>
        <label>Telefono di casa: <?php echo($riga['telefono_casa']. " ");?></label><br>
    </div>
    <div>
    <br>
    <form action="privato.php" method="POST">
        <?php
            // nel caso in cui non vi siano cartelle cliniche presenti
            if (mysqli_num_rows($resultCartellaClinica)==0){
                echo("<label>Non è stata trovata nessuna cartella clinica </label><br>");
            }
            // nel caso in cui siano presenti cartelle cliniche
            else{
                echo("<label>Selezione la tua cartella clinica </label><br>");
                echo "<select name='id_cartella_clinica'>";
                while ($rigaCartellaClinica=mysqli_fetch_assoc($resultCartellaClinica))
                {
                    echo "<option
value=".$rigaCartellaClinica["id_cartella_clinica"].">".$rigaCartellaClinica["id_cartella_clinica"];
                } // chiusura while
                echo"<option value='*' selected>Visualizza tutte";
                echo"<input type='submit' value='Invio'>";
            }
        ?>
    </div>
</div>

```

```

<button>
    <a href="esamiFatti.php">Visualizza dati relativi agli esami fatti</a>
</button>
<button>
    <a href="esamiDaFare.php">Visualizza dati relativi agli esami da fare o fatti in data odierna</a>
</button>
</div>
<button>
    <a href="logout.php">Logout</a>
</button>
</body>
</html>

```

Area Personale Alessio Pardini

Codice Tessera: 80380000900101850626
 Codice Fiscale: PRDLSS02P03E625L
 Sesso: M
 Data di nascita: 2002-09-03
 Nazionalita: Italia
 Comune: Livorno
 Indirizzo di residenza: Viale giouse carducci 46
 Cap di residenza: 57124
 Indirizzo di domicilio: Viale giouse carducci 46
 Cap di domicilio: 57124
 Telefono: 3249052010
 Telefono di casa: 3480540446

Selezione la tua cartella clinica

Visualizza tutte ▼ Invio

Visualizza dati relativi agli esami fatti

Visualizza dati relativi agli esami da fare o fatti in data odierna

Logout

Figura 4.1

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui siano presenti cartelle cliniche per il paziente.

Area Personale Michela Mazzoni

Codice Tessera: 80380000900099515538
 Codice Fiscale: MZZMHL68H49E625X
 Sesso: F
 Data di nascita: 1968-06-09
 Nazionalita: Italia
 Comune: Livorno
 Indirizzo di residenza: Viale giouse carducci 46
 Cap di residenza: 57124
 Indirizzo di domicilio: Viale giouse carducci 46
 Cap di domicilio: 57124
 Telefono: 3249052010
 Telefono di casa: 3480540446

Non è stata trovata nessuna cartella clinica

Visualizza dati relativi agli esami fatti

Visualizza dati relativi agli esami da fare o fatti in data odierna

Logout

Figura 4.2

L'immagine presente in figura rappresenta la schermata

visualizzata tramite lo script PHP, nel caso in cui non siano presenti cartelle cliniche per il paziente.

4.3.4 Area Visualizzazione Cartella Clinica (privato.php)

Il codice riportato di seguito, descrive l'area che permette la visualizzazione del contenuto relativo alla cartella clinica, infatti si potranno visualizzare i dati di eventuali report e di ipotetici risultati delle analisi.

```
<?php
    session_start();
    include("connessione_db.php");
    include("config.php");
    $id_cartella_clinica=$_POST['id_cartella_clinica'];
    $codice_tessera = $_SESSION['cod'];
    // query per ottenere i dati relativi ad una cartella clinica
    $queryCartella = "SELECT cartellaClinica.*
    FROM cartellaClinica
    WHERE codice_tessera = ? ";
    //caso in cui si visualizza nello specifico
    if($id_cartella_clinica != '*'){
        $queryCartella.=" AND id_cartella_clinica=?";
        $stmtCartella = mysqli_stmt_init($conn);
        if(!mysqli_stmt_prepare($stmtCartella,$queryCartella)){
            echo "Errore SQL";
        }
        else{
            mysqli_stmt_bind_param($stmtCartella,"ss",$codice_tessera,
$id_cartella_clinica);
            $resultCartella=mysqli_stmt_execute($stmtCartella);
        }
        $resultCartella = mysqli_stmt_get_result($stmtCartella);
    }
    //caso in cui si visualizza tutto
    else{
        $stmtCartella = mysqli_stmt_init($conn);
        if(!mysqli_stmt_prepare($stmtCartella,$queryCartella)){
            echo "Errore SQL";
        }
        else{
            mysqli_stmt_bind_param($stmtCartella,"s", $codice_tessera);
            $resultCartella=mysqli_stmt_execute($stmtCartella);
        }
        $resultCartella = mysqli_stmt_get_result($stmtCartella);
    }
    // query per ottenere i dati relativi ai report
    $queryReport = "SELECT report.*, operatore.nome, operatore.cognome
```



```

        FROM report, operatore
        WHERE report.matricola = operatore.matricola
        AND codice_tessera = ?";
//caso in cui si visualizza nello specifico
if($id_cartella_clinica != '*'){
    $queryReport.=" AND id_cartella_clinica=?";
    $stmtReport = mysqli_stmt_init($conn);
    if(!mysqli_stmt_prepare($stmtReport,$queryReport)){
        echo "Errore SQL";
    }
    else{
        mysqli_stmt_bind_param($stmtReport,"ss",$codice_tessera,$id_cartella_clinica);
        $resultReport=mysqli_stmt_execute($stmtReport);
    }
    $resultReport = mysqli_stmt_get_result($stmtReport);
}
//caso in cui si visualizza tutto
else{
    $stmtReport = mysqli_stmt_init($conn);
    if(!mysqli_stmt_prepare($stmtReport,$queryReport)){
        echo "Errore SQL";
    }
    else{
        mysqli_stmt_bind_param($stmtReport,"s",$codice_tessera);
        $resultReport=mysqli_stmt_execute($stmtReport);
    }
    $resultReport = mysqli_stmt_get_result($stmtReport);
}
// query per ottenere il risultato analisi per un determinato esame
$queryAnalisi = "SELECT risultato_analisi, id_esame, data_prenotazione,
id_cartella_clinica, codice_tessera
FROM analisi, provetta
WHERE analisi.id_analisi = provetta.id_analisi
AND codice_tessera = ?";
//caso in cui si visualizza nello specifico
if($id_cartella_clinica != '*'){
    $queryAnalisi.=" AND id_cartella_clinica=?";
    $stmtAnalisi = mysqli_stmt_init($conn);
    if(!mysqli_stmt_prepare($stmtAnalisi,$queryAnalisi)){
        echo "Errore SQL";
    }
    else{
        mysqli_stmt_bind_param($stmtAnalisi,"ss",$codice_tessera,
$id_cartella_clinica);
        $resultAnalisi=mysqli_stmt_execute($stmtAnalisi);
    }
    $resultAnalisi = mysqli_stmt_get_result($stmtAnalisi);
}
//caso in cui si visualizza tutto
else{
    $stmtAnalisi = mysqli_stmt_init($conn);
    if(!mysqli_stmt_prepare($stmtAnalisi,$queryAnalisi)){
        echo "Errore SQL";
    }

```

```

    }
    else{
        mysqli_stmt_bind_param($stmtAnalisi,"s", $codice_tessera);
        $resultAnalisi=mysqli_stmt_execute($stmtAnalisi);
    }
    $resultAnalisi = mysqli_stmt_get_result($stmtAnalisi);
}
?>
<html>
<head>
    <title>Cartella clinica</title>
</head>
<body>
<!--Visualizzazione cartella clinica-->
<table border>
    <caption><b>Cartella clinica</b></caption>
    <thead>
        <tr>
            <th>Identificativo della cartella clinica</th>
            <th>Codice della tessera</th>
            <th>Data inizio</th>
            <th>Data fine</th>
            <th>codice fiscale del medico curante</th>
            <th>Temperatura (gradi)</th>
            <th>Peso (kg)</th>
            <th>Descrizione</th>
        </tr>
    </thead>
    <tbody>
        <?php

            while ($row = mysqli_fetch_assoc($resultCartella)){
                ?>
                <tr>
                    <td><?php echo ($row['id_cartella_clinica']); ?></td>
                    <td><?php echo ($row['codice_tessera']); ?></td>
                    <td><?php echo ($row['data_inizio']); ?></td>
                    <td><?php echo ($row['data_fine']); ?></td>
                    <td><?php echo ($row['cf_medico_curante']); ?></td>
                    <td><?php echo ($row['temperatura_gradi']); ?></td>
                    <td><?php echo ($row['peso_kg']); ?></td>
                    <td><?php echo ($row['descrizione']); ?></td>
                </tr>
            <?php
            }
            mysqli_free_result($resultCartella);
            echo "</tbody>\n";
            echo "</table>\n";
        ?>

<!--Visualizzazione Report-->
<?php
if (mysqli_num_rows($resultReport) == 0){

```

```

        echo("Nessun report trovato.");
        echo("<br>");
    }else{
?>
<table border>
<caption><b>Report</b></caption>
<thead>
<tr>
<th>Identificativo esame</th>
<th>Data prenotazione esame</th>
<th>Identificativo della cartella clinica</th>
<th>Codice della tessera</th>
<th>Identificativo operatore</th>
<th>Nome operatore</th>
<th>Cognome operatore</th>
<th>Data e ora inizio esame</th>
<th>Data e ora fine esame</th>
<th>Descrizione</th>
</tr>
</thead>
<tbody>
<?php

    while ($row = mysqli_fetch_assoc($resultReport)){
?>
        <tr>
            <td><?php echo ($row['id_esame']); ?></td>
            <td><?php echo ($row['data_prenotazione']); ?></td>
            <td><?php echo ($row['id_cartella_clinica']); ?></td>
            <td><?php echo ($row['codice_tessera']); ?></td>
            <td><?php echo ($row['matricola']); ?></td>
            <td><?php echo ($row['nome']); ?></td>
            <td><?php echo ($row['cognome']); ?></td>
            <td><?php echo ($row['dataOra_inizio']); ?></td>
            <td><?php echo ($row['dataOra_fine']); ?></td>
            <td><?php echo ($row['descrizione']); ?></td>
        </tr>
    <?php
    }
    mysqli_free_result($resultReport);
    echo "</tbody>\n";
    echo "</table>\n";
}
?>
<?php
if (mysqli_num_rows($resultAnalisi) == 0){
    echo("Nessun analisi trovata.");
    echo("<br>");
}else{
?>
<table border>
<caption><b>Analisi</b></caption>
<thead>

```

```

<tr>
  <th>Risultato delle analisi</th>
  <th>identificativo dell'esame</th>
  <th>Data prenotazione esame</th>
  <th>Identificativo della cartella clinica</th>
  <th>Codice della tessera</th>
</tr>
</thead>
<tbody>
  <?php
    while ($row = mysqli_fetch_assoc($resultAnalisi)){
      ?>
        <tr>
          <td><?php echo ($row['risultato_analisi']); ?></td>
          <td><?php echo ($row['id_esame']); ?></td>
          <td><?php echo ($row['data_prenotazione']); ?></td>
          <td><?php echo ($row['id_cartella_clinica']); ?></td>
          <td><?php echo ($row['codice_tessera']); ?></td>
        </tr>
      <?php
    }
    mysqli_free_result($resultAnalisi);
    echo "</tbody>\n";
    echo "</table>\n";
    mysqli_close($conn);
  }
  ?>
<label>Torna alla <a href="areaPersonale.php">home</a></label>
<button>
  <a href="logout.php">Logout</a>
</button>
</body>
</html>

```

Selezione la tua cartella clinica

AAA1 ▼ Invio

Figura 4.3

Cartella clinica									
Identificativo della cartella clinica	Codice della tessera	Data inizio	Data fine	codice fiscale del medico curante	Temperatura (gradi)	Peso (kg)	Descrizione		
AAA1	80380000900101850626	2020-03-11	2020-04-12	MRALSS80A16E62SR	36.7	58	Lesione del tendine della spalla		
Report									
Identificativo esame	Data prenotazione esame	Identificativo della cartella clinica	Codice della tessera	Identificativo operatore	Nome operatore	Cognome operatore	Data e ora inizio esame	Data e ora fine esame	Descrizione
00001	2020-03-14 15:15:00	AAA1	80380000900101850626	0001	Stefano	Bianchi	2020-03-14 15:15:00	2020-03-14 15:31:32	La mobilità della spalla risulta essere fortemente danneggiata
Analisi									
Risultato delle analisi		identificativo dell'esame	Data prenotazione esame	Identificativo della cartella clinica	Codice della tessera				
I risultati hanno rivelato x nel paziente		00001	2020-03-14 15:15:00	AAA1	80380000900101850626				
Torna alla home Logout									

Figura 4.4

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui siano presenti report ed eventuali risultati delle analisi all'interno di una specifica cartella clinica.

Selezione la tua cartella clinica

AAA2 ▼ Invio

Figura 4.5

Cartella clinica

Identificativo della cartella clinica	Codice della tessera	Data inizio	Data fine	codice fiscale del medico curante	Temperatura (gradi)	Peso (kg)	Descrizione		
AAA2	80380000900101850626	2021-03-11	2021-04-12	MRALSS80A16E62SR	36.7	58	Lesione del tendine della spalla		

Report

Identificativo esame	Data prenotazione esame	Identificativo della cartella clinica	Codice della tessera	Identificativo operatore	Nome operatore	Cognome operatore	Data e ora inizio esame	Data e ora fine esame	Descrizione
00001	2021-03-20 12:00:00	AAA2	80380000900101850626	0001	Stefano	Bianchi	2021-03-20 12:00:00	2021-03-20 12:15:00	La mobilità della spalla risulta essere fortemente danneggiata

Nessun analisi trovata.

Torna alla [home](#)

[Logout](#)

Figura 4.6

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui siano presenti report al contrario dei risultati delle analisi che risultano assenti, all'interno di una specifica cartella clinica.

Selezione la tua cartella clinica

AAA3 ▼ Invio

Figura 4.7

Cartella clinica							
Identificativo della cartella clinica	Codice della tessera	Data inizio	Data fine	codice fiscale del medico curante	Temperatura (gradi)	Peso (kg)	Descrizione
AAA3	80380000900101850626	2021-05-12	0000-00-00	MRALSS80A16E625R	36.7	58	Lesione del tendine subiacomale

Nessun report trovato.
 Nessun analisi trovata.
 Torna alla [home](#) [Logout](#)

Figura 4.8

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui non siano presenti ne report ne risultati delle analisi all'interno di una specifica cartella clinica.

Selezione la tua cartella clinica

Visualizza tutte

Figura 4.9

Cartella clinica							
Identificativo della cartella clinica	Codice della tessera	Data inizio	Data fine	codice fiscale del medico curante	Temperatura (gradi)	Peso (kg)	Descrizione
AAA1	80380000900101850626	2020-03-11	2020-04-12	MRALSS80A16E625R	36.7	58	Lesione del tendine della spalla
AAA2	80380000900101850626	2021-03-11	2021-04-12	MRALSS80A16E625R	36.7	58	Lesione del tendine della spalla
AAA3	80380000900101850626	2021-05-12	0000-00-00	MRALSS80A16E625R	36.7	58	Lesione del tendine subiacomale

Report									
Identificativo esame	Data prenotazione esame	Identificativo della cartella clinica	Codice della tessera	Identificativo operatore	Nome operatore	Cognome operatore	Data e ora inizio esame	Data e ora fine esame	Descrizione
00001	2020-03-14 15:15:00	AAA1	80380000900101850626	0001	Stefano	Bianchi	2020-03-14 15:15:00	2020-03-14 15:31:32	La mobilità della spalla risulta essere fortemente danneggiata
00001	2021-03-20 12:00:00	AAA2	80380000900101850626	0001	Stefano	Bianchi	2021-03-20 12:00:00	2021-03-20 12:15:00	La mobilità della spalla risulta essere fortemente danneggiata

Analisi				
Risultato delle analisi	identificativo dell'esame	Data prenotazione esame	Identificativo della cartella clinica	Codice della tessera
I risultati hanno rivelato x nel paziente	00001	2020-03-14 15:15:00	AAA1	80380000900101850626

Torna alla [home](#) [Logout](#)

Figura 4.10

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui si scelga di visualizzare tutti i dati di tutte le cartelle cliniche.

4.3.5 Area Visualizzazione Esami Fatti (esamiFatti.php)

Il codice riportato di seguito, descrive l'area che permette la visualizzazione degli esami fatti.

```
<?php
session_start();
include("connessione_db.php");
include("config.php");
$codice_tessera = $_SESSION['cod'];
//query esami
```

```

$queryEsami = "SELECT esame.*
FROM esame,(
SELECT id_esame, data_prenotazione
    FROM report
        WHERE codice_tessera=?
            AND data_prenotazione < CURRENT_DATE()) AS t1
WHERE esame.id_esame = t1.id_esame AND esame.data_prenotazione =
t1.data_prenotazione;";

$stmt = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmt,$queryEsami)){
    echo "Errore SQL";
}
else{
    mysqli_stmt_bind_param($stmt,"s",$codice_tessera);
    $result=mysqli_stmt_execute($stmt);
}
$result = mysqli_stmt_get_result($stmt);

if (mysqli_num_rows($result) == 0){
    echo("Nessun esame fatto in data odierna oppure da fare.");
    ?>
    <label>Torna alla <a href="areaPersonale.php">home</a></label>
    <button>
        <a href="logout.php">Logout</a>
    </button>
    <?php
        die();
    }
    ?>
<html>
<head>
    <title>Esami</title>
</head>
<body>

<table border>
<caption><b>Esami fatti</b></caption>
<thead>
<tr>
    <th>Identificativo esame</th>
    <th>Data prenotazione esame</th>
    <th>Identificativo prestazione</th>
    <th>Temperatura corporea (gradi)</th>
    <th>Pressione sanguigna (mmHG)</th>
    <th>Frequenza respiratoria (atti al minuto)</th>
    <th>Ossigenazione sanguigna (percentuale)</th>
    <th>Frequenza cardiaca (bpm)</th>
</tr>
</thead>
<tbody>
    <?php
        while ($row = mysqli_fetch_assoc($result)){

```

```

?>

<tr>
<td><?php echo ($row['id_esame']); ?></td>
<td><?php echo ($row['data_prenotazione']); ?></td>
<td><?php echo ($row['id_prestazione']); ?></td>
<td><?php echo ($row['temperatura_corporea_gradi']); ?></td>
<td><?php echo ($row['pressione_sanguigna_mmHG']); ?></td>
<td><?php echo ($row['frequenza_respiratoria_attiMinuto']); ?></td>
<td><?php echo ($row['ossigenazione_sanguigna_percentuale']); ?></td>
<td><?php echo ($row['frequenza_cardiaca_bpm']); ?></td>
</tr>

<?php
}
mysqli_free_result($result);
echo "</tbody>\n";
echo "</table>\n";
mysqli_close($conn);
?>
<label>Torna alla <a href="areaPersonale.php">home</a></label>
<button>
<a href="logout.php">Logout</a>
</button>
</body>
</html>

```



Figura 4.11

Esami fatti							
Identificativo esame	Data prenotazione esame	Identificativo prestazione	Temperatura corporea (gradi)	Pressione sanguigna (mmHG)	Frequenza respiratoria (atti al minuto)	Ossigenazione sanguigna (percentuale)	Frequenza cardiaca (bpm)
00001	2020-03-14 15:15:00	0001	37.2	60.2	30.2	80	80.1
00001	2021-03-20 12:00:00	0001	37.2	60.2	30.2	80	80.1

Torna alla [home](#) [Logout](#)

Figura 4.12

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui siano presenti esami fatti.

Nessun esame fatto. Torna alla [home](#)

Figura 4.13

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui non siano presenti esami fatti.

4.3.6 Area Visualizzazione Esami Da Fare (esamiDaFare.php)

Il codice riportato di seguito, descrive l'area che permette la visualizzazione degli esami da fare.

```
<?php
session_start();
include("connessione_db.php");
include("config.php");
$codice_tessera = $_SESSION['cod'];
//query esami
$queryEsami = "SELECT esame.id_esame, esame.data_prenotazione, esame.id_prestazione,
prontuario.descrizione
FROM esame, prontuario, (
SELECT id_esame, data_prenotazione
FROM report
WHERE codice_tessera=?
AND data_prenotazione>= CURRENT_DATE()) AS t1
WHERE esame.id_esame = t1.id_esame AND esame.data_prenotazione = t1.data_prenotazione AND
prontuario.id_prestazione = esame.id_prestazione;";
$stmt = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmt,$queryEsami)){
    echo "Errore SQL";
}
else{
    mysqli_stmt_bind_param($stmt,"s",$codice_tessera);
    $result=mysqli_stmt_execute($stmt);
}
$result = mysqli_stmt_get_result($stmt);

if (mysqli_num_rows($result) == 0){
    echo("Nessun esame fatto in data odierna oppure da fare.");
    ?>
    <label>Torna alla <a href="areaPersonale.php">home</a></label>
    <button>
        <a href="logout.php">Logout</a>
    </button>
<?php
die();
}
?>
<html>
```

```

<head>
  <title>Esami</title>
</head>
<body>

<table border>
  <caption><b>Esami da fare</b></caption>
  <thead>
    <tr>
      <th>Identificativo esame</th>
      <th>Data prenotazione esame</th>
      <th>Identificativo prestazione</th>
      <th>Tipologia esame</th>
    </tr>
  </thead>
  <tbody>
    <?php
      while ($row = mysqli_fetch_assoc($result)){
        ?>
        <tr>
          <td><?php echo ($row['id_esame']); ?></td>
          <td><?php echo ($row['data_prenotazione']); ?></td>
          <td><?php echo ($row['id_prestazione']); ?></td>
          <td><?php echo ($row['descrizione']); ?></td>
        </tr>
      <?php
    }
    mysqli_free_result($result);
    echo "</tbody>\n";
    echo "</table>\n";
    mysqli_close($conn);
    ?>
  <label>Torna alla <a href="areaPersonale.php">home</a></label>
  <button>
    <a href="logout.php">Logout</a>
  </button>
</body>
</html>

```



Figura 4.14

Esami da fare			
Identificativo esame	Data prenotazione esame	Identificativo prestazione	Tipologia esame
00001	2021-06-25 12:00:00	0001	Intervento x

Torna alla [home](#) [Logout](#)

Figura 4.15

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui siano presenti esami da fare.

Nessun esame fatto in data odierna oppure da fare. Torna alla [home](#) [Logout](#)

Figura 4.16

L'immagine presente in figura rappresenta la schermata visualizzata tramite lo script PHP, nel caso in cui non siano presenti esami da fare.

4.3.7 Login (login.php)

Il codice riportato di seguito, descrive l'area login.

Nell'implementazione non si presenta una funzionalità specifica per il recupero della password in caso di smarrimento. Si ipotizza dunque che questa pagina permetta anche questa funzionalità.

```
<?php include('config.php'); ?>
<!DOCTYPE html>
<script>
//permette di visualizzare la password
function showPwd() {
    var input = document.getElementById('pwd');
    if (input.type === "password") {
        input.type = "text";
    } else {
        input.type = "password";
    }
}
```

```

</script>

<html>
  <head>

    <title>Login</title>
    <link rel="stylesheet" href="styleL.css">
  </head>

  <body>
    <div class="container">
      <form id="login" action="verifica.php" method="post">
        <h4>Asl<span>Toscana</span></h4>
        <h5>Tutti i campi sono obbligatori.</h5>
        <div class="inputs">
          <input class="if--input" id="email" name="email" type="text"
            placeholder="Inserisci email" autofocus required>
          <label class="pass">
            <!-- Immagine vettoriale -->
            <svg class="vector-image" width="20px" height="30px" viewBox="0 0 488.85 488.85"
onclick="showPwd()">
              <g>
                <path
d="M244.425,98.725c-93.4,0-178.1,51.1-240.6,134.1c-5.1,6.8-5.1,16.3,0,23.1c62.5,83.1,147.2,134.2,240.6,134.
2
s178.1-51.1,240.6-134.1c5.1-6.8,5.1-16.3,0-23.1c422.525,149.825,337.825,98.725,244.425,98.725z
M251.125,347.025

c-62,3.9-113.2-47.2-109.3-109.3c3.2-51.2,44.7-92.7,95.9-95.9c62-3.9,113.2,47.2,109.3,109.3
C343.725,302.225,302.225,343.725,251.125,347.025z
M248.025,299.625c-33.4,2.1-61-25.4-58.8-58.8c1.7-27.6,24.1-49.9,51.7-51.7
c33.4-2.1,61,25.4,58.8,58.8c297.925,275.625,275.525,297.925,248.025,299.625z"/>
              </g>
            </svg>
          </label>
          <input id="pwd" class="if--input" id="password" name="password" type="password"
            placeholder="Inserisci password" required>
        </div>
        <div class="paragrafo">Non sei ancora registrato?<a href="registrazione.php"
id="back">Registrati!</a></div>
        <input type="submit" value="Login" class="btn1">

      </form>
    </div>
  </body>
</html>

```

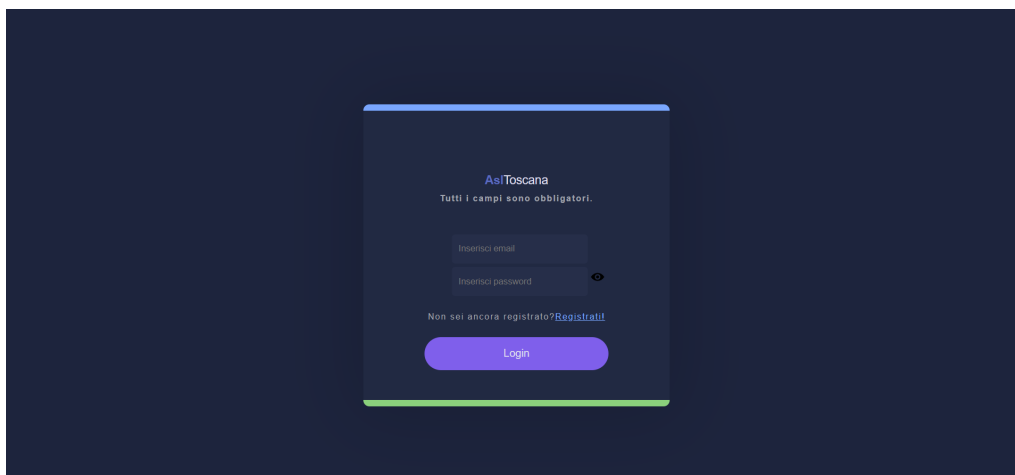


Figura 4.17

L'immagine presente in figura rappresenta la schermata iniziale visualizzata tramite lo script PHP.

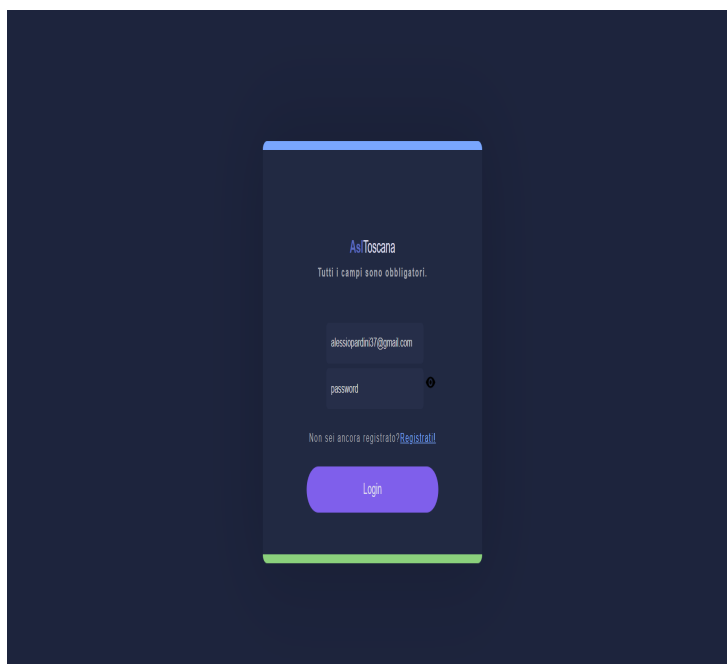
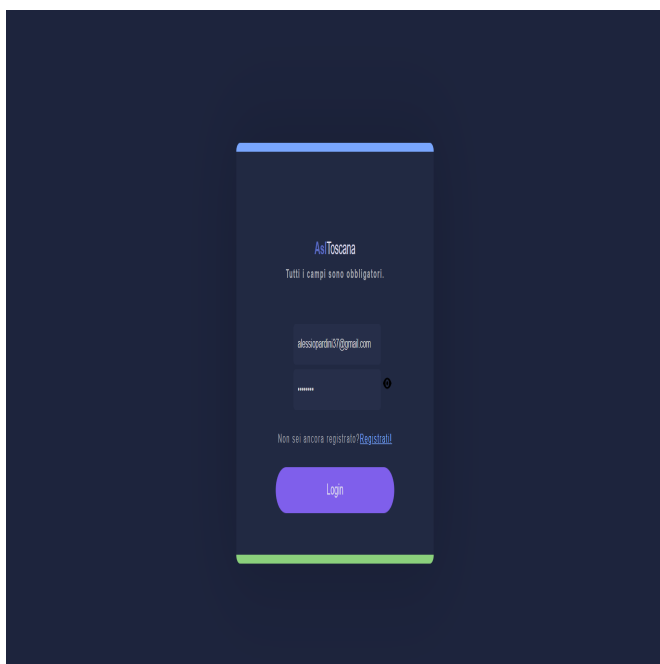


Figura 4.18

L'immagine presente in figura rappresenta la funzionalità resa disponibile dalla funzione *'showPwd'*, la quale permette di visualizzare la password inserita, per verificare eventuali errori in digitazione.

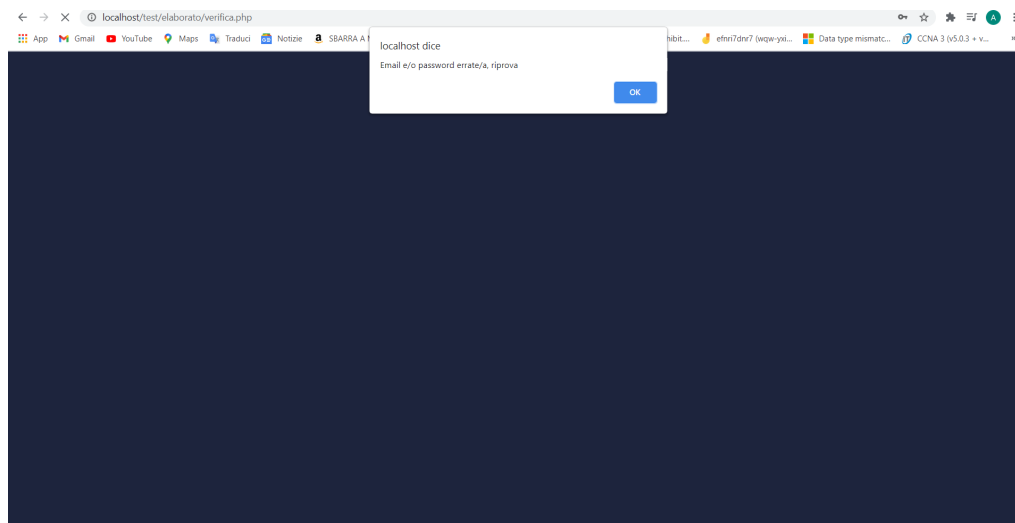


Figura 4.19

L'immagine presente in figura mostra come in caso di email o password errate, verrà visualizzato un alert di errore. Nel caso in cui si premesse il pulsante 'OK', l'utente sarà rimandato alla pagina di login. Questa funzionalità è resa disponibile dallo script PHP presente nel file 'verifica.php', il quale viene richiamato nel momento in cui l'utente dopo aver inserito l'email e la password, effettui un click sul bottone 'login'.

4.3.8 Verifica(verifica.php)

Il codice riportato di seguito, viene richiamato dallo script presente in 'login.php'.

```
<?php
session_start(); //inizio sessione
//inclusione file per collegamento al db con script di accesso
include("connessione_db.php");
include("config.php");
//conn
mysqli_select_db($conn, "$db_name");
//variabili POST con anti sqlInjection (escape caratteri
//potenzialmente dannosi)
$email=mysqli_real_escape_string($conn,$_POST['email']);
$password=mysqli_real_escape_string($conn,$_POST['password']);
$password=crypt($password,'$2a$07$usesomesillystringforsalt$');
$query = "SELECT *
FROM credenziali
WHERE email = ?;";
$stmt = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmt,$query)){
    echo "Errore SQL";
}
else{
```

```

mysqli_stmt_bind_param($stmt,"s",$email);
mysqli_stmt_execute($stmt);

}
$result = mysqli_stmt_get_result($stmt);

$riga = mysqli_fetch_array($result);
if (hash_equals($password, $riga['password'])){
    $trovato=1;
}
else
$trovato = 0 ;
if($trovato) { //Email e password corrette
    $_SESSION['abilitato'] = 1;
    $_SESSION['cod'] = $riga['id_utente']; //Registro il codice utente
    /*Redirect alla pagina riservata*/
    echo '<script
    language=javascript>document.location.href="areaPersonale.php"
    </script>';
}
else {
    /*Username e password errati, redirect alla pagina di login*/
    echo '<script
    language=javascript>alert("Email e/o password errate/a, riprova");
    document.location.href="login.php"
    </script>';
}
?>

```

4.3.8 Area Registrazione (registrazione.php)

Il codice riportato di seguito descrive la pagina di registrazione per i pazienti.

Lo script PHP è composto da due parti fondamentali, la prima che identifica una funzione javascript ‘Modulo’ la quale viene richiamata nel momento in cui l’utente effettua un click sul bottone ‘Registrati’; questa funzione ha lo scopo di controllare la validità dei campi inseriti.

La seconda parte fondamentale presenta una ‘Form’ che permette l’inserimento di input differenti e di passare questi dati alla pagina ‘Insert.php’.

```

<script >
function Modulo() {
    // Variabili associate ai campi del modulo
    var codice_tessera = document.modulo.codice_tessera.value;
    var codFisc = document.modulo.codFisc.value;
    var nome = document.modulo.nome.value;
    var cognome = document.modulo.cognome.value;
    var sesso = document.modulo.sesso.value;
    var data_nascita = document.modulo.data_nascita.value;

```

```

var nazionalita = document.modulo.nazionalita.value;
var comune = document.modulo.comune.value;
var indirizzo_residenza = document.modulo.indirizzo_residenza.value;
var cap_residenza = document.modulo.cap_residenza.value;
var indirizzo_domicilio = document.modulo.indirizzo_domicilio.value;
var cap_domicilio = document.modulo.cap_domicilio.value;
var telefono = document.modulo.telefono.value;
var telefono_casa = document.modulo.telefono_casa.value;
var email = document.modulo.email.value;
var password = document.modulo.password.value;
var conferma = document.modulo.conferma.value;
var agree_term = document.getElementById('agree');
// Espressione regolare dell'email
var email_reg_exp = /^[a-zA-Z0-9_-]+@([a-zA-Z0-9-]{2,})+\.([a-zA-Z0-9]{2,})+$/;
var lenCodiceTessera = codice_tessera.toString().length;
var lenCapResidenza = cap_residenza.toString().length;
var lenCapRDomicilio = cap_domicilio.toString().length;
var check=/^[A-Za-z]+[à,è,ì,ò,ù]$/;
//Effettua il controllo sul campo codice_tessera
if ((codice_tessera == "") || (codice_tessera == "undefined")) {
alert("Il campo Codice Tessera è obbligatorio.");
document.modulo.codice_tessera.focus();
return false;
}
if(isNaN(codice_tessera)){
alert("Il campo Codice Tessera è numerico.");
document.modulo.codice_tessera.focus();
return false;
}
if(lenCodiceTessera != 20){
alert("Inserire il Codice Tessera di dimensione corretta (20 cifre).");
//document.modulo.conferma.value = "";
document.modulo.codice_tessera.focus();
return false;
}
//Effettua il controllo sul campo codFisc
if ((codFisc == "") || (codFisc == "undefined")) {
alert("Il campo Codice Fiscale è obbligatorio.");
document.modulo.codFisc.focus();
return false;
}
if(codFisc.length != 16){
alert("Inserire il Codice Fiscale di dimensione corretta (16 caratteri).");
document.modulo.codFisc.focus();
return false;
}

```



```

}
if(codFisc != codFisc.toUpperCase()){
    alert("I caratteri del Codice Fiscale devono essere maiuscoli.");
    document.modulo.codFisc.focus();
    return false;
}

```

```

//Effettua il controllo sul campo nome
if ((nome == "") || (nome == "undefined")) {
    alert("Il campo Nome è obbligatorio.");
    document.modulo.nome.focus();
    return false;
}
if(!check.test(nome)){
    alert("Il campo Nome può contenere esclusivamente lettere.");
    document.modulo.nome.focus();
    return false;
}
if(nome != (nome.charAt(0).toUpperCase() + nome.slice(1))){
    nome = nome.charAt(0).toUpperCase() + nome.slice(1);
}

```

```

//Effettua il controllo sul campo cognome
if ((cognome == "") || (cognome == "undefined")) {
    alert("Il campo Cognome è obbligatorio.");
    document.modulo.cognome.focus();
    return false;
}
if(!check.test(cognome)){
    alert("Il campo Cognome può contenere esclusivamente lettere.");
    document.modulo.cognome.focus();
    return false;
}
if(cognome != (cognome.charAt(0).toUpperCase() + cognome.slice(1))){
    cognome = cognome.charAt(0).toUpperCase() + cognome.slice(1);
}

```

```

//Effettua il controllo sul campo sesso
if ((sesso == "") || (sesso == "undefined")) {
    alert("Il campo Sesso è obbligatorio.");
    document.modulo.sesso.focus();
    return false;
}

```

```

//Effettua il controllo sul campo data_nascita
if ((data_nascita == "") || (data_nascita == "undefined")) {
alert("Il campo Data Nascita è obbligatorio.");
document.modulo.data_nascita.focus();
return false;
}

//Effettua il controllo sul campo nazionalita
if ((nazionalita == "") || (nazionalita == "undefined")) {
alert("Il campo Nazionalita è obbligatorio.");
document.modulo.nazionalita.focus();
return false;
}
if(!check.test(nazionalita)){
alert("Il campo Nazionalita può contenere esclusivamente lettere.");
document.modulo.nazionalita.focus();
return false;
}
if(nazionalita != (nazionalita.charAt(0).toUpperCase() + nazionalita.slice(1))){
    nazionalita = nazionalita.charAt(0).toUpperCase() + nazionalita.slice(1);
}

//Effettua il controllo sul campo comune
if ((comune == "") || (comune == "undefined")) {
alert("Il campo Comune è obbligatorio.");
document.modulo.comune.focus();
return false;
}
if(!check.test(comune)){
alert("Il campo Comune può contenere esclusivamente lettere.");
document.modulo.comune.focus();
return false;
}
if(comune != (comune.charAt(0).toUpperCase() + comune.slice(1))){
    comune = comune.charAt(0).toUpperCase() + comune.slice(1);
}

//Effettua il controllo sul campo indirizzo_residenza
if ((indirizzo_residenza == "") || (indirizzo_residenza == "undefined")) {
alert("Il campo Indirizzo Residenza è obbligatorio.");
document.modulo.indirizzo_residenza.focus();
return false;
}

//Effettua il controllo sul campo cap_residenza

```

```

if ((cap_residenza == "") || (cap_residenza == "undefined")) {
    alert("Il campo Cap Residenza è obbligatorio.");
    document.modulo.cap_residenza.focus();
    return false;
}
if(isNaN(cap_residenza)){
    alert("Il campo cap residenza è numerico.");
    document.modulo.cap_residenza.focus();
    return false;
}
if(lenCapResidenza!=5){
    alert("Inserire il Cap Residenza di dimensione corretta (5 cifre).");
    document.modulo.cap_residenza.focus();
    return false;
}

//Effettua il controllo sul campo indirizzo_domicilio
if ((indirizzo_domicilio == "") || (indirizzo_domicilio == "undefined")) {
    alert("Il campo Indirizzo Domicilio è obbligatorio.");
    document.modulo.indirizzo_domicilio.focus();
    return false;
}

//Effettua il controllo sul campo cap_domicilio
if ((cap_domicilio == "") || (cap_domicilio == "undefined")) {
    alert("Il campo Cap Domicilio è obbligatorio.");
    document.modulo.cap_domicilio.focus();
    return false;
}
if(lenCapRDomicilio != 5){
    alert("Inserire il Cap Domicilio di dimensione corretta (5 cifre).");
    document.modulo.cap_domicilio.focus();
    return false;
}
if(isNaN(cap_domicilio)){
    alert("Il campo Cap Domicilio è numerico.");
    document.modulo.cap_domicilio.focus();
    return false;
}

//Effettua il controllo sul campo telefono
if ((telefono == "") || (telefono == "undefined")) {
    alert("Il campo Telefono è obbligatorio.");
    document.modulo.telefono.focus();
    return false;
}

```

```

}
if(isNaN(telefono)){
    alert("Il campo telefono è numerico.");
    document.modulo.telefono.focus();
    return false;
}
if(telefono.length!=10){
    alert("Inserire il Numero di Telefono di dimensione corretta (10 cifre).");
    document.modulo.telefono.focus();
    return false;
}

//Effettua il controllo sul campo telefono_casa
else if ((telefono_casa == "") || (telefono_casa == "undefined")) {
    alert("Il campo Telefono Casa è numerico ed obbligatorio.");
    document.modulo.telefono_casa.value = "";
    document.modulo.telefono_casa.focus();
    return false;
}
if(isNaN(telefono_casa)){
    alert("Il campo telefono casa è numerico.");
    document.modulo.telefono_casa.focus();
    return false;
}

//Effettua il controllo sul campo email
if ((email == "") || (email == "undefined")) {
    alert("Il campo Email è obbligatorio.");
    document.modulo.email.focus();
    return false;
}
if (!email_reg_exp.test(email)) {
    alert("Inserire un indirizzo email corretto.");
    document.modulo.email.select();
    return false;
}
//Effettua il controllo sul campo PASSWORD
if ((password == "") || (password == "undefined")) {
    alert("Il campo Password è obbligatorio.");
    document.modulo.password.focus();
    return false;
}
//controllo numeri
var x=0;

```

```

var check1 = /[([0-9])]/;
if(check1.test(password)){
    x = x + 20;
}
//controllo minuscole
var check2=/[a-z]/;
if(check2.test(password)){
    x = x + 20;
}
//controllo maiuscole
var check3=/[A-Z]/;
if(check3.test(password)){
    x = x + 20;
}
//controllo simboli
var check4=/[$-/:-?{-~!"^_`\[\\]]/;
if(check4.test(password)){
    x = x + 20;
}
// controllo lunghezza (minore o uguale a 10 caratteri)
if(password.length >=10){
    x = x + 20;
}
if(x<60){
    alert("password troppo debole, assicurati che quest'ultima contenga almeno un carattere
minuscolo, almeno un carattere maiuscolo, almeno un numero e una lunghezza maggiore di 9
caratteri.");
    document.modulo.conferma.focus();
    return false;
}
//Effettua il controllo sul campo CONFERMA_PASSWORD
if ((conferma == "") || (conferma == "undefined")) {
    alert("Il campo Conferma Password è obbligatorio.");
    document.modulo.conferma.focus();
    return false;
}
//Verifica l'uguaglianza tra i campi PASSWORD e CONFERMA PASSWORD
if (password != conferma) {
    alert("La password di conferma è differente da quella inizialmente inserito, per favore
riserisci.");
    document.modulo.conferma.value = "";
    document.modulo.conferma.focus();
    return false;
}
if(agree_term.checked == false){

```

```

alert("Prima di effettuare l'accesso, devi accettare i Termini e condizioni d'uso.");
document.modulo.agree_term.focus();
}

//INVIA IL MODULO
else{
document.modulo.action = "Insert.php";
document.modulo.submit();
}
}
function showPwd() {
    var input = document.getElementById('pwd');
    if (input.type === "password") {
        input.type = "text";
    } else {
        input.type = "password";
    }
}
function showPwd1() {
    var input = document.getElementById('pwd1');
    if (input.type === "password") {
        input.type = "text";
    } else {
        input.type = "password";
    }
}
</script>
<html>
<title>Registrazione</title>
<head><link rel = "stylesheet" href="styleR.css"> </head>

<div class="container">
    <form method="post" class="box" name="modulo">
        <h4>Asl<span>Toscana</span></h4>
        <h5>Tutti i campi sono obbligatori.</h5>
        <div class="row">
            <input class="if--input" type="text" name="codice_tessera" placeholder="Codice tessera sanitaria" >
            <input class="if--input" type="text" name="codFisc" placeholder="Codice fiscale" >
        </div>
        <div class="row">
            <input class="if--input" type="text" name="nome" placeholder="Nome" >
            <input class="if--input" type="text" name="cognome" placeholder="Cognome" >
        </div>
        <div class="row">

```

```

<select class="if--input" name="sesso" >
  <option value=""></option>
  <option value="M">Maschio</option>
  <option value="F">Femmina</option>
<input class="if--input" type="date" name="data_nascita" required>
<span class="placeholder"> </span>
</div>
<div class="row">
<input class="if--input" type="text" name="nazionalita" placeholder="Nazionalità" >
<input class="if--input" type="text" name="comune" placeholder="Comune" >
</div>
<div class="row">
<input class="if--input" type="text" name="indirizzo_residenza" placeholder="Indirizzo di
residenza" >
<input class="if--input" type="text" name="cap_residenza" placeholder="CAP di residenza" >
</div>
<div class="row">
<input class="if--input" type="text" name="indirizzo_domicilio" placeholder="Indirizzo di
domicilio" >
<input class="if--input" type="text" name="cap_domicilio" placeholder="CAP di domicilio" >
</div>
<div class="row">
<input class="if--input" type="text" name="telefono" placeholder="Telefono" >
<input class="if--input" type="text" name="telefono_casa" placeholder="Telefono casa" >
</div>
<div class="row-email-password">
<input class="if--input" type="text" name="email" placeholder="Email" >
<input id="pwd" class="if--input" type="password" name="password" placeholder="Password">
<label class="pass">
<svg class="vector-image" width="20px" height="30px" viewBox="0 0 488.85 488.85"
onclick="showPwd()">
  <g>
    <path
d="M244.425,98.725c-93.4,0-178.1,51.1-240.6,134.1c-5.1,6.8-5.1,16.3,0,23.1c62.5,83.1,147.2,134.2,24
0.6,134.2
s178.1-51.1,240.6-134.1c5.1-6.8,5.1-16.3,0-23.1c422.525,149.825,337.825,98.725,244.425,98.725z
M251.125,347.025
c-62,3.9-113.2-47.2-109.3-109.3c3.2-51.2,44.7-92.7,95.9-95.9c62-3.9,113.2,47.2,109.3,109.3
C343.725,302.225,302.225,343.725,251.125,347.025z
M248.025,299.625c-33.4,2.1-61-25.4-58.8-58.8c1.7-27.6,24.1-49.9,51.7-51.7
c33.4-2.1,61,25.4,58.8,58.8c297.925,275.625,275.525,297.925,248.025,299.625z"/>
  </g>
</svg>

```

```

</label>
</div>
<div class="row">
<input id ="pwd1" class="if--input" type="password" name="conferma" placeholder="Conferma
password" >
<label class="pass">
<svg class="vector-image" width="20px" height="30px" viewBox="0 0 488.85 488.85"
onclick="showPwd1()">
    <g>
        <path
d="M244.425,98.725c-93.4,0-178.1,51.1-240.6,134.1c-5.1,6.8-5.1,16.3,0,23.1c62.5,83.1,147.2,134.2,24
0.6,134.2

s178.1-51.1,240.6-134.1c5.1-6.8,5.1-16.3,0-23.1c422.525,149.825,337.825,98.725,244.425,98.725z
M251.125,347.025

c-62,3.9-113.2-47.2-109.3-109.3c3.2-51.2,44.7-92.7,95.9-95.9c62-3.9,113.2,47.2,109.3,109.3
    C343.725,302.225,302.225,343.725,251.125,347.025z
M248.025,299.625c-33.4,2.1-61-25.4-58.8-58.8c1.7-27.6,24.1-49.9,51.7-51.7
    c33.4-2.1,61,25.4,58.8,58.8c297.925,275.625,275.525,297.925,248.025,299.625z"/>
    </g>
</svg>
</label>
</div>
<input type="checkbox" name="agree_term" id="agree"/>
<label class="label-term" for="agree_term" ><span><span></span></span>Accetto le
dichiarazioni contenute nei <a href="termini.php">Termini e condizioni d'uso</a></label>
<div class="login-div">Hai già un account?<a href="login.php">Accedi.</a></div>
<input type="button" value="Registrati" onClick="Modulo()" class="btn1">
</form>
</div>

</html>

```


Figura 4.20

L'immagine presente in figura rappresenta la schermata iniziale visualizzata tramite lo script PHP.

Figura 4.21

L'immagine presente in figura rappresenta come in caso di parametri errati oppure non compilati, venga visualizzato un alert specifico.

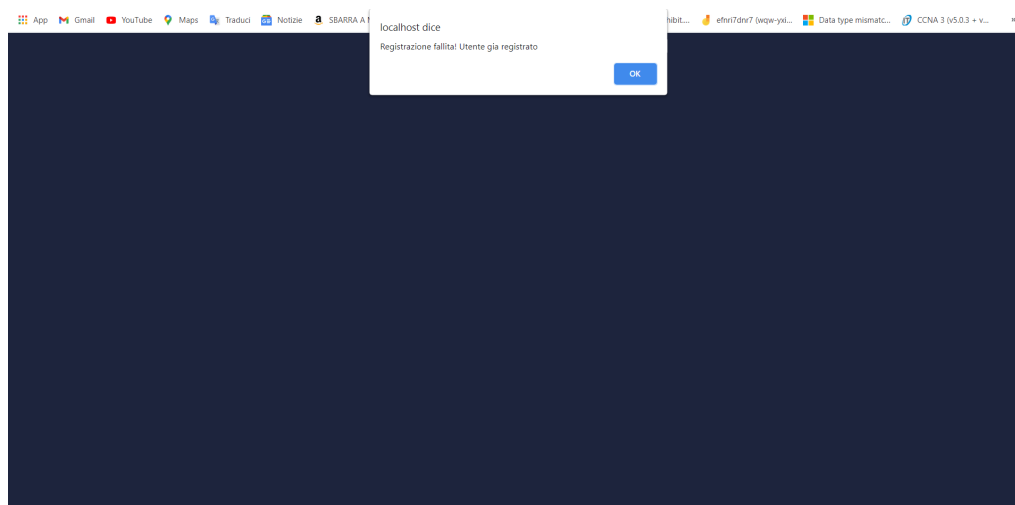


Figura 4.22

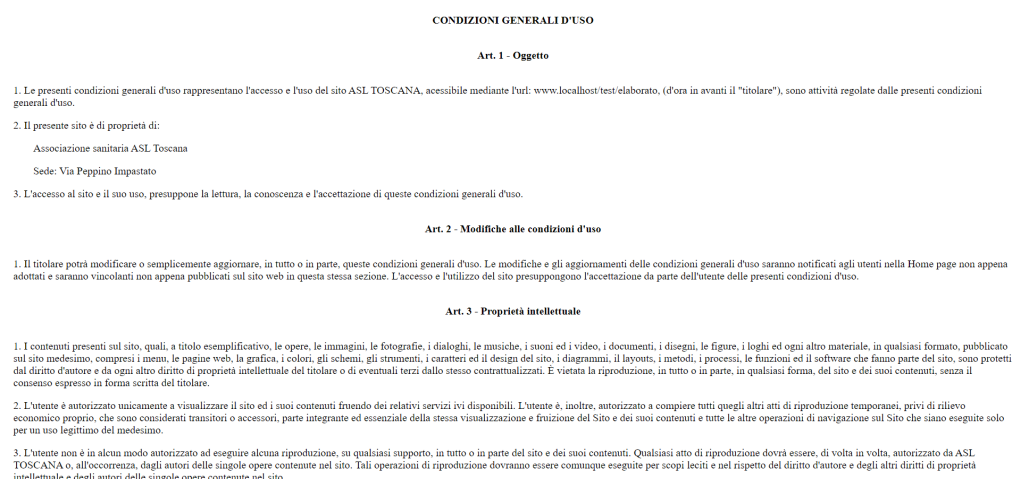
L'immagine presente in figura rappresenta come in caso un utente già registrato provi a registrarsi di nuovo, venga visualizzato l'alert in figura. Questa funzionalità è resa disponibile dallo script PHP presente nel file *'Insert.php'*.

Email già registrata, impossibile registrarsi usando questo indirizzo email.

Email già registrata, impossibile registrarsi usando questo indirizzo email.

Figura 4.23

L'immagine presente in figura rappresenta come in caso si provi ad utilizzare un email già registrata nella struttura dati, venga visualizzato un messaggio di errore. Questa funzionalità è resa disponibile dallo script PHP presente nel file *'Insert.php'*



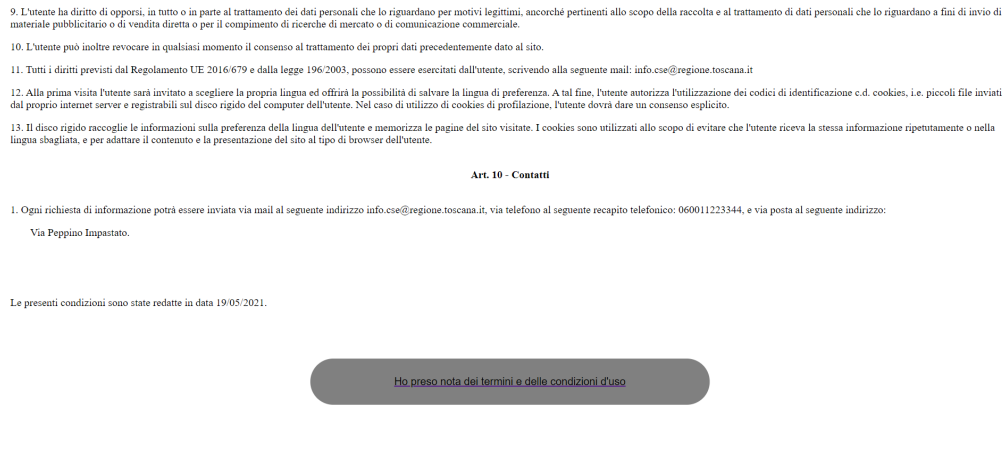


Figura 4.24

L'immagine presente in figura rappresenta i termini e condizioni d'uso che devono essere letti e accettati per potersi registrare al sito. Questa figura rappresenta ciò che viene rappresentato dallo script PHP presente nel file *'termini.php'*.

4.3.9 Insert (Inasert.php)

Il codice riportato di seguito, viene richiamato dallo script presente in *'registrazione.php'*.

```
<?php
include("connessione_db.php");
//gestione parametri per effettuare query
$codice_tessera = mysqli_real_escape_string($conn,$_POST['codice_tessera']);
$codice_fiscale = mysqli_real_escape_string($conn,$_POST['codFisc']);
$nome = mysqli_real_escape_string($conn,$_POST['nome']);
$cognome = mysqli_real_escape_string($conn,$_POST['cognome']);
$ sesso = mysqli_real_escape_string($conn,$_POST['sesso']);
$data_nascita = mysqli_real_escape_string($conn,$_POST['data_nascita']);
$nazionalita = mysqli_real_escape_string($conn,$_POST['nazionalita']);
$comune = mysqli_real_escape_string($conn,$_POST['comune']);
$indirizzo_residenza = mysqli_real_escape_string($conn,$_POST['indirizzo_residenza']);
$cap_residenza = mysqli_real_escape_string($conn,$_POST['cap_residenza']);
$indirizzo_domicilio = mysqli_real_escape_string($conn,$_POST['indirizzo_domicilio']);
$cap_domicilio = mysqli_real_escape_string($conn,$_POST['cap_domicilio']);
$telefono = mysqli_real_escape_string($conn,$_POST['telefono']);
$telefono_casa = mysqli_real_escape_string($conn,$_POST['telefono_casa']);
$email = mysqli_real_escape_string($conn,$_POST['email']);
$password = mysqli_real_escape_string($conn,$_POST['password']);

//conversione da Stringa a intero per il campo CAP, poichè all'interno del DB quest'ultimo è intero e
```

```

non string
$cap_residenza=(int)$cap_residenza;
$cap_domicilio=(int)$cap_domicilio;

// valutare opzione prioritata $priorita=1;
//cripto la password
$password=crypt($password,'$2a$07$usesomesillystringforsalt$');
//controllo email se esiste gia
$controllo_email = "SELECT email FROM credenziali WHERE email = ?";
$stmt = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmt,$controllo_email)){
    echo "Errore SQL";
}
else{
    mysqli_stmt_bind_param($stmt,"s",$email);
    mysqli_stmt_execute($stmt);
}
$resultControllo = mysqli_stmt_get_result($stmt);

$row = mysqli_fetch_array($resultControllo);
if (mysqli_num_rows($resultControllo) != 0) {
    echo "Email già registrata, impossibile registrarsi usando questo indirizzo email.";
    echo "<br>";
    echo "<br>";
    echo "<button><a href='registrazione.php'>Email già registrata, impossibile registrarsi usando
questo indirizzo email.</a></button>";
    die();
}
//query registrazione credenziali relative al paziente
$queryCredenziali = "INSERT INTO credenziali(id_utente, email, password) VALUES (?,?,:)";

$stmtCredenziali = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmtCredenziali,$queryCredenziali)){
    echo "Errore SQL";
}
else{
    mysqli_stmt_bind_param($stmtCredenziali,"sss",$codice_tessera,$email,$password);
    $resultCredenziali=mysqli_stmt_execute($stmtCredenziali);
}

//query registrazione del paziente nel database
$queryPaziente = "INSERT INTO paziente(codice_tessera, codice_fiscale, nome, cognome,
sesso, data_nascita, nazionalita, comune, indirizzo_residenza, cap_residenza,
indirizzo_domicilio, cap_domicilio,telefono,telefono_casa) VALUES (?,?,,?,,?,,?,,?,,?)";

$stmtPaziente = mysqli_stmt_init($conn);
if(!mysqli_stmt_prepare($stmtPaziente,$queryPaziente)){
    echo "Errore SQL";
}
else{
    mysqli_stmt_bind_param($stmtPaziente,"sssssssssisiss",$codice_tessera,$codice_fiscale,$nome,$cognome,$sesso
,$data_nascita,$nazionalita,$comune,$indirizzo_residenza,

```

```

$cap_residenza,$indirizzo_domicilio,$cap_domicilio,$telefono,$telefono_casa);
$resultPaziente=mysqli_stmt_execute($stmtPaziente);
}

if($resultCredenziali && $resultPaziente){
    echo '<script language=javascript>
    alert("Registrazione completata");
    document.location.href="login.php";
    </script>';
}
else{
    echo '<script language=javascript>
    alert("Registrazione fallita! Utente gia registrato");
    document.location.href="login.php";
    </script>';
}
?>

```

4.3.10 Termini e condizioni d'uso (termini.php)

Il codice riportato di seguito, viene richiamato dallo script presente in *'registrazione.php'*.

```

<html>
<head>
    <link rel = "stylesheet" href="styleT.css">
    <title>Termini e condizioni d'uso</title>
</head>

<body>
<!-- in seguito seguiva il testo dei termini e condizioni d'uso che è stato eliminato, per rendere più
leggero il seguente script.-->
<a href="registrazione.php">
<button class="bottone">Ho preso nota dei termini e delle condizioni d'uso </button>
</a>
</body>
</html>

```

Capitolo 5

GDPR

I dati saranno ovviamente trattati in conformità della normativa GDPR che ad oggi rappresenta la punta di diamante per la privacy.

[il testa di seguito rappresenta un estratto da: P. Ollari, “Normative relative allo sviluppo del software”

in “Manuale Cremonese di Informatica e Telecomunicazioni”, III edizione, Zanichelli, 2020] .

Dal 25 maggio 2018 in Italia viene [...] adottato il Regolamento dell'Unione europea n. 2016/679 Regolamento Generale sulla Protezione Dei Dati (GDPR, General Data Protection Regulation).

Il GDPR riprende molti contenuti del D.lgs. n. 196/2003, tra cui l’obiettivo di base: “Chiunque ha diritto alla protezione dei dati personali che lo riguardano”. Quindi estende questo diritto a tutta l’Unione Europea, disciplinando il trattamento dei soli dati personali unicamente delle persone fisiche, escludendo i dati identificativi di soggetti con personalità giuridica (aziende), ma includendo i dati personali delle ditte individuali. Inoltre, si applica anche a soggetti con sede legale fuori dall’UE ma che posseggono dati personali di cittadini con residenza UE.

Il GDPR definisce dato personale qualsiasi informazione sufficiente a identificare una precisa persona fisica: è dato personale un identificativo informatico (per es. un account), ma non lo è un dato riservato come l’importo dello stipendio. Tra i dati personali il GDPR distingue i dati particolari (ex dati sensibili, ovvero quei dati personali idonei a rivelare l’origine razziale ed etnica, le convinzioni religiose, filosofiche, politiche, l’adesione a partiti, sindacati, associazioni, l’orientamento e la vita sessuale) e, tra questi, anche i dati genetici, i dati biometrici e i dati relativi alla salute. Infine, riconosce specificatamente lo status di dato giudiziario come il dato personale relativo a reati o procedimenti giudiziari in genere. A differenza del precedente 196/03, che per i dati sensibili prevedeva il consenso scritto del soggetto, il GDPR per i dati particolari consente un generico consenso “inequivocabile”, che può essere scritto, ma non necessariamente.

Come precedentemente, i dati personali possono essere trattati da terzi solo con consenso esplicito della persona interessata, tranne tre casi:

1. dati sanitari, per finalità di ricerca medica o epidemiologica;
2. dati giudiziari, per le finalità proprie delle Autorità giudiziarie o relative all’Ordine pubblico;
3. dati personali in genere allorquando siano ritenuti utili a scopo di sicurezza nazionale.

Il GDPR prevede, oltre al Titolare (del trattamento dei dati personali) solo un’altra figura

coinvolta, ovvero il Responsabile, colui o coloro che sono autorizzati dal Titolare al trattamento dei dati posseduti dall'organizzazione; vengono introdotte le figure dei Contitolari (Joint Controller), laddove più entità condividono o si scambiano dati personali.

In effetti il GDPR aggiunge una figura inedita alla precedente normativa, il Responsabile per la Protezione dei Dati (DPO, Data Protection Officer) [...].

Il Titolare, ma soprattutto il DPO quando previsto, devono anche assicurare una specifica politica per la Sicurezza del trattamento dei dati, adeguato al rischio correlato, e che comprende:

- la pseudo-anonimizzazione e la cifratura dei dati personali;
- la capacità di assicurare su base permanente la riservatezza, l'integrità, la disponibilità e la resilienza dei sistemi e dei servizi di trattamento;
- la capacità di ripristinare tempestivamente la disponibilità e l'accesso dei dati personali in caso di incidente fisico o tecnico;
- una procedura per testare, verificare e valutare regolarmente l'efficacia delle misure tecniche e organizzative al fine di garantire la sicurezza del trattamento.

Se il GDPR mantiene il diritto alla cancellazione, alla limitazione e alla rettifica dei dati personali richieste dai soggetti interessati (diritto all'oblio), le due novità più importanti introdotte dal GDPR sono sicuramente quelle relative al cosiddetto data breach (violazione dei dati) e alle norme circa il diritto alla portabilità dei dati. Qualora fosse individuata una violazione dei dati personali e questa dovesse presentare un rischio per i diritti e le libertà delle persone fisiche coinvolte, il Titolare deve comunicare la violazione agli interessati "senza indebito ritardo", al fine di consentire agli stessi di adottare le necessarie precauzioni per tutelare i propri diritti. La mancata notifica implica sanzioni amministrative di una certa rilevanza.

Le violazioni citate potrebbero essere relative ai criteri di sicurezza classici, ovvero non solo violazioni di riservatezza (violazione della privacy), ma anche violazioni di integrità e di disponibilità (violazioni del RID).

Allo stesso modo un soggetto ha il diritto alla portabilità dei suoi dati personali ovvero di ricevere da una organizzazione che li detiene i propri dati personali in una forma idonea alla trasferibilità informatica verso una nuova organizzazione che li richiede.

Capitolo 6

Ipotesi APP

Nel caso di un dispositivo con sistema operativo Android le APP possono essere sviluppate in linguaggio JAVA con accesso alle API esposte dal sistema operativo per questo linguaggio.

Nello scenario descritto dal testo del problema, i sanitari addetti al servizio saranno dotati di dispositivi mobili attraverso i quali potranno utilizzare una specifica APP.

Le funzionalità riportate dal testo dall'elaborato sono le seguenti:

- Permette agli operatori di redigere un report in tempo reale con i riferimenti temporali (data/ora) di inizio/fine intervento e una descrizione di ciò che è stato fatto.
- Permette, se necessario, una consultazione mediante chat col personale medico e organizzativo presente presso il centro di controllo del servizio in remoto.
- Permette al centro di controllo di allertare il personale più idoneo/i per professionalità e vicinanza a causa di interventi estemporanei richiesti da urgenze specifiche segnalate dagli assistiti.

Si ipotizza che le eventuali funzionalità aggiuntive resi disponibili dall'applicazione per gli operatori addetti al servizio domiciliare siano le seguenti:

- Consentire la visualizzazione dell'elenco degli interventi da eseguire in giornata.
- Consentire la visualizzazione su mappa della posizione dei domicili relativi agli interventi.

L'applicazione in questione si baserà anche sull'utilizzo di API che verranno presentate di seguito.

Capitolo 7

Web API

7.1 Progettazione

Le seguenti API supportano e permettono le operazioni che il personale amministrativo o gli operatori sanitari possono utilizzare tramite APP o sito Web che interroghi un Web Service.

Richiesta	Metodo	Risposta
Registra nuovo operatore: /*ovviamente viene inserito l'hashing della password */	POST URL http://.../Elaborato/ServletElaborato/nuovoOperatore? <operatore> <matricola>...</matricola> <codice_fiscale>...</codice_fiscale> <data_assunzione>...</data_assunzione> <cognome>...</cognome> <nome>...</nome> < Sesso>...</ Sesso> <data_nascita>...</data_nascita> <nazionalita>...</nazionalita> <comune>...</comune> <indirizzo_residenza>...</indirizzo_residenza> <cap_residenza>...</cap_residenza> <telefono>...</telefono> <telefono_casa>...</telefono_casa> <email>...</email> <password>....</password> </operatore>	VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva status-code HTTP (200 se OK, opportuni errori altrimenti) 400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db

Registra nuovo esame	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoEsame?</p> <pre><esame> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_prestazione>...</id_prestazione> <temperatura_corporea_gradi>...</temperatura_corpo rea_gradi> <pressione_sanguigna_mmHG>...</pressione_sanguig na_mmHG> <frequenza_respiratoria_attiMinuto>...</frequenza_res piratoria_attiMi nuto> <ossigenazione_sanguigna_percentuale>...</ossigen azione_sanguig na_percentuale> <frequenza_cardiaca_bpm>...</frequenza_cardiaca_b pm> </esame></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
Registra nuovo report	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoReport?</p> <pre><report> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <matricola>...</matricola> <dataOra_inizio>...</dataOra_inizio> <dataOra_fine>...</dataOra_fine> <descrizione>...</descrizione> </report></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
Registra il ruolo di un determinato operatore in un determinato intervento	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoRuolo?</p> <pre><esameOperatore> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <matricola>...</matricola> <ruolo>...</ruolo> </esameOperatore></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>

Registra nuova tipologia di esame	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoProntuario?</p> <pre><prontuario> <id_prestazione>...</id_prestazione> <tempo_espletamento_min>...</tempo_espletamento_min> <descrizione>...</descrizione> <n_medici>...</n_medici> <n_infermieri>...</n_infermieri> <prezzo>...</prezzo> </prontuario></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
Registra nuova provetta	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoProvetta?</p> <pre><provetta> <id_provetta>...</id_provetta> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_laboratorio>...</id_laboratorio> <id_analisi>...</id_analisi> <tipologia_prelievo>...</tipologia_prelievo> <capienza_ml^3>...</capienza_ml^3> <lunghezza_mm>...</lunghezza_mm> <diametro_mm>...</diametro_mm> </provetta></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
Registra nuova cartella clinica	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoCartellaClinica?</p> <pre><cartellaClinica> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <data_inizio>...</data_inizio> <data_fine>...</data_fine> <cf_medico_curante>...</cf_medico_curante> <temperatura_gradi>...</temperatura_gradi> <peso_kg>...</peso_kg> <descrizione>...</descrizione> </cartellaClinica></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>

Registra nuovo laboratorio	<p>POST</p> <p>URL http://../Elaborato/ServletElaborato/nuovoLaboratorio?</p> <pre><laboratorio> <id_laboratorio>...</id_laboratorio> <nome>...</nome> <comune>...</comune> <indirizzo>...</indirizzo> </laboratorio></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
<p>Registra nuovo paziente:</p> <p><i>/*ovviamente viene inserito l'hashing della password */</i></p>	<p>POST</p> <p>URL http://../Elaborato/ServletElaborato/nuovoPaziente?</p> <pre><paziente> <codice_tessera>...</codice_tessera> <codice_fiscale>...</codice_fiscale> <cognome>...</cognome> <nome>...</nome> < Sesso>...</ Sesso> <data_nascita>...</data_nascita> <nazionalita>...</nazionalita> <comune>...</comune> <indirizzo_residenza>...</indirizzo_residenza> <cap_residenza>...</cap_residenza> <indirizzo_domicilio>...</indirizzo_domicilio> <cap_domicilio>...</cap_domicilio> <telefono>...</telefono> <telefono_casa>...</telefono_casa> <email>...</email> <password>...</password> </paziente></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
Registra nuova analisi	<p>POST</p> <p>URL http://../Elaborato/ServletElaborato/nuovoAnalisi?</p> <pre><analisi> <id_analisi>...</id_analisi> <id_laboratorio>...</id_laboratorio> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <risultato_analisi>...</risultato_analisi> </analisi></pre>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>

Registra nuova specializzazione per un determinato operatore	<p>POST</p> <p>URL http://.../Elaborato/ServletElaborato/nuovoProntuarioOperatore?</p> <p><prontuarioOperatore> <id_prestazione>...</id_prestazione> <matricola>...</matricola> </prontuarioOperatore></p>	<p>VIENE INVIATO SOLTANTO LO STATUS CODE + una stringa descrittiva</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p>400 richiesta errata 401 errore durante l'inserimento 402 XML parser error 500 errore connessione al db</p>
Richiesta dati relativi ad un paziente	<p>GET</p> <p>URL http://.../Elaborato/ServletElaborato/paziente?codice_tessera=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p><paziente> <codice_tessera>...</codice_tessera> <codice_fiscale>...</codice_fiscale> <cognome>...</cognome> <nome>...</nome> < Sesso>...</ Sesso> <data_nascita>...</data_nascita> <nazionalita>...</nazionalita> <comune>...</comune> <indirizzo_residenza>...</indirizzo_residenza> <cap_residenza>...</cap_residenza> <indirizzo_domicilio>...</indirizzo_domicilio> <cap_domicilio>...</cap_domicilio> <telefono>...</telefono> <telefono_casa>...</telefono_casa> <email>...</email> </paziente></p> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>

<p>Richiesta dati relativi ad una cartella clinica specifica di un determinato paziente</p>	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/cartellaClinica?id_cartella_clinica =&codice_tessera=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <pre><cartellaClinica> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <data_inizio>...</data_inizio> <data_fine>...</data_fine> <medico_curante>...</medico_curante> <temperatura_gradi>...</temperatura_gradi> <peso_kg>...</peso_kg> <descrizione>...</descrizione> </cartellaClinica></pre> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>
<p>Richiesta dati relativi ad un operatore</p>	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/operatore?matricola=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <pre><operatore> <matricola>...</matricola> <codice_fiscale>...</codice_fiscale> <data_assunzione>...</data_assunzione> <cognome>...</cognome> <nome>...</nome> <sex>...</sex> <data_nascita>...</data_nascita> <nazionalita>...</nazionalita> <comune>...</comune> <indirizzo_residenza>...</indirizzo_residenza> <cap_residenza>...</cap_residenza> <telefono>...</telefono> <telefono_casa>...</telefono_casa> <email>...</email> </paziente></pre> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>

Richiesta dati relativi al ruolo di un determinato operatore in un determinato intervento	<p>GET</p> <p>URL</p> <p>http://.../Elaborato/ServletElaborato/ruolo?id_esame=&data_prenotazione=&matricola=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <pre><esameOperatore> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <matricola>...</matricola> <ruolo>...</ruolo> </esameOperatore></pre> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>
Richiesta dati relativi ad un determinato esame	<p>GET</p> <p>URL</p> <p>http://.../Elaborato/ServletElaborato/esame?id_esame=&data_prenotazione=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <pre><esame> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_prestazione>...</id_prestazione> <temperatura_corporea_gradi>...</temperatura_corporea_gradi> <pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG> <frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto> <ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale> <frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm> </esame></pre> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>

<p>Richiesta dati relativi ad una determinata tipologia d'esame.</p>	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/prontuario?id_prestazione=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <pre><prontuario> <id_prestazione>...</id_prestazione> <tempo_espletamento_min>...</tempo_espletamento_min> <descrizione>...</descrizione> <n_medici>...</n_medici> <n_infermieri>...</n_infermieri> <prezzo>...</prezzo> </prontuario></pre> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>
<p>Richiesta dati relativi ad una provetta</p>	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/provetta?id_provetta=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <pre><provetta> <id_provetta>...</id_provetta> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_laboratorio>...</id_laboratorio> <id_analisi>...</id_analisi> <tipologia_prelievo>...</tipologia_prelievo> <capienza_ml^3>...</capienza_ml^3> <lunghezza_mm>...</lunghezza_mm> <diametro_mm>...</diametro_mm> </provetta></pre> <p>400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db</p>

Richiesta dati relativi ad un laboratorio	GET URL http://../Elaborato/ServletElaborato/laboratorio?id_labo ratorio=	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <laboratorio> <id_laboratorio>...</id_laboratorio> <nome>...</nome> <comune>...</comune> <indirizzo>...</indirizzo> </laboratorio> 400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db
Richiesta dati relativi ad una analisi	GET URL http://../Elaborato/ServletElaborato/analisi?id_analisi=	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <analisi> <id_analisi>...</id_analisi> <id_laboratorio>...</id_laboratorio> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> </analisi> 400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db
Richiesta dati relativi ad un report	GET URL http://../Elaborato/ServletElaborato/report?id_esame= &data_prenotazione=	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <report> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <matricola>...</matricola> <dataOra_inizio>...</dataOra_inizio> <dataOra_fine>...</dataOra_fine> <descrizione>...</descrizione> </report> 400 richiesta errata

		401 record non presente 402 parametri mancanti 500 errore connessione al db
Richiesta per ottenere gli operatori che sono specializzati in una determinata tipologia di esame.	GET URL http://../Elaborato/ServletElaborato/prontuarioOperatore?id_prestazione=	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <operatori> <matricola>...</matricola> <matricola>...</matricola> ... </report> 400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db
Lista report redatti da un determinato operatore	GET URL http://../Elaborato/ServletElaborato/listaReport?matricola=	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <lista_report> <report> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <matricola>...</matricola> <dataOra_inizio>...</dataOra_inizio> <dataOra_fine>...</dataOra_fine> <descrizione>...</descrizione> </report> <report> <id_esame>...</id_esame> <data_prenotazione>...</data_prenotazione> <id_cartella_clinica>...</id_cartella_clinica> <codice_tessera>...</codice_tessera> <matricola>...</matricola> <dataOra_inizio>...</dataOra_inizio> <dataOra_fine>...</dataOra_fine> <descrizione>...</descrizione> </report> ... </lista_report> 400 richiesta errata 401 record non presente 402 parametri mancanti

		500 errore connessione al db
Lista esami di un determinato operatore	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/listaEsami?matricola=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p><listaEsami></p> <p><esame></p> <p><id_esame>...</id_esame></p> <p><data_prenotazione>...</data_prenotazione></p> <p><id_prestazione>...</id_prestazione></p> <p><temperatura_corporea_gradi>...</temperatura_corporea_gradi></p> <p><pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG></p> <p><frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto></p> <p><ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale></p> <p><frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm></p> <p></esame></p> <p><esame></p> <p><id_esame>...</id_esame></p> <p><data_prenotazione>...</data_prenotazione></p> <p><id_prestazione>...</id_prestazione></p> <p><temperatura_corporea_gradi>...</temperatura_corporea_gradi></p> <p><pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG></p> <p><frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto></p> <p><ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale></p> <p><frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm></p> <p></esame></p>

<p>Lista esami in data odierna</p>	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/listaEsamiOdierni?</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p><listaEsami></p> <p><esame></p> <p><id_esame>...</id_esame></p> <p><data_prenotazione>...</data_prenotazione></p> <p><id_prestazione>...</id_prestazione></p> <p><temperatura_corporea_gradi>...</temperatura_corporea_gradi></p> <p><pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG></p> <p><frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto></p> <p><ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale></p> <p><frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm></p> <p></esame></p> <p><esame></p> <p><id_esame>...</id_esame></p> <p><data_prenotazione>...</data_prenotazione></p> <p><id_prestazione>...</id_prestazione></p> <p><temperatura_corporea_gradi>...</temperatura_corporea_gradi></p> <p><pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG></p> <p><frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto></p> <p><ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale></p> <p><frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm></p> <p></esame></p> <p>...</p> <p></listaEsami></p> <p>400 richiesta errata</p> <p>401 record non presente</p> <p>402 parametri mancanti</p> <p>500 errore connessione al db</p>
------------------------------------	--	---

<p>Lista esami in data odierna</p>	<p>GET</p> <p>URL</p> <p>http://../Elaborato/ServletElaborato/listaEsamiOdierni?</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p><listaEsami></p> <p><esame></p> <p><id_esame>...</id_esame></p> <p><data_prenotazione>...</data_prenotazione></p> <p><id_prestazione>...</id_prestazione></p> <p><temperatura_corporea_gradi>...</temperatura_corporea_gradi></p> <p><pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG></p> <p><frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto></p> <p><ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale></p> <p><frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm></p> <p></esame></p> <p><esame></p> <p><id_esame>...</id_esame></p> <p><data_prenotazione>...</data_prenotazione></p> <p><id_prestazione>...</id_prestazione></p> <p><temperatura_corporea_gradi>...</temperatura_corporea_gradi></p> <p><pressione_sanguigna_mmHG>...</pressione_sanguigna_mmHG></p> <p><frequenza_respiratoria_attiMinuto>...</frequenza_respiratoria_attiMinuto></p> <p><ossigenazione_sanguigna_percentuale>...</ossigenazione_sanguigna_percentuale></p> <p><frequenza_cardiaca_bpm>...</frequenza_cardiaca_bpm></p> <p></esame></p> <p>...</p> <p></listaEsami></p> <p>400 richiesta errata</p> <p>401 record non presente</p> <p>402 parametri mancanti</p> <p>500 errore connessione al db</p>
------------------------------------	--	---

<p>Lista cartelle cliniche di un determinato paziente</p>	<p>GET</p> <p>URL</p> <p>http://.../Elaborato/ServletElaborato/listaCartelleClinichePaziente?</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p><lista_cartellaCliniche></p> <p><cartellaClinica></p> <p><id_cartella_clinica>...</id_cartella_clinica></p> <p><codice_tessera>...</codice_tessera></p> <p><data_inizio>...</data_inizio></p> <p><data_fine>...</data_fine></p> <p><medico_curante>...</medico_curante></p> <p><temperatura_gradi>...</temperatura_gradi></p> <p><peso_kg>...</peso_kg></p> <p><descrizione>...</descrizione></p> <p></cartellaClinica></p> <p><cartellaClinica></p> <p><id_cartella_clinica>...</id_cartella_clinica></p> <p><codice_tessera>...</codice_tessera></p> <p><data_inizio>...</data_inizio></p> <p><data_fine>...</data_fine></p> <p><medico_curante>...</medico_curante></p> <p><temperatura_gradi>...</temperatura_gradi></p> <p><peso_kg>...</peso_kg></p> <p><descrizione>...</descrizione></p> <p></cartellaClinica></p> <p>...</p> <p></lista_cartellaCliniche></p> <p>400 richiesta errata</p> <p>401 record non presente</p> <p>402 parametri mancanti</p> <p>500 errore connessione al db</p>
<p>Numero esami per un determinato comune</p>	<p>GET</p> <p>URL</p> <p>http://.../Elaborato/ServletElaborato/numEsamiComune?comune=</p>	<p>VIENE INVIATO LO STATUS CODE</p> <p>status-code HTTP (200 se OK, opportuni errori altrimenti)</p> <p><num>..</num></p> <p>400 richiesta errata</p> <p>401 record non presente</p> <p>402 parametri mancanti</p> <p>500 errore connessione al db</p>

Numero esami effettuati di una determinata tipologia in uno specifico anno	GET URL http://../Elaborato/ServletElaborato/numEsamiAnno?anno=	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <num>...</num> 400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db
Numero esami effettuati a livello annuale.	GET URL http://../Elaborato/ServletElaborato/numEsamiAnnuali?	VIENE INVIATO LO STATUS CODE status-code HTTP (200 se OK, opportuni errori altrimenti) <num>..</num> 400 richiesta errata 401 record non presente 402 parametri mancanti 500 errore connessione al db

7.2 Implementazione

Nel codice riportato di seguito sono riportate le implementazioni delle richieste *'esame'*, *'listaReport'*, *'nuovoEsame'*.

```

/**
 * le import sono state eliminate, per rendere più leggere il codice
 * @author Alessio Pardini
 */
public class ServletElaborato extends HttpServlet {
    private Connection con = null;
    private String dbms_url = "jdbc:mysql://localhost:3306/asl";

    public void init() {
        try {
            Class.forName("com.mysql.jdbc.Driver");
            con = DriverManager.getConnection(dbms_url, "root", "");
            System.out.println("Connessione con dbms effettuata");
            // creare la connessione con il database
        } catch (SQLException ex) {
            con = null;
            System.out.println("Errore connessione DBMS");
        } catch (ClassNotFoundException ex) {
            System.out.println("Errore connessione DBMS");
        }
    }
    @Override
    protected void doGet(HttpServletRequest request, HttpServletResponse response)
        throws ServletException, IOException {
        //controllo se la connessione al db è andata a buon fine
        if(con == null){
            response.sendError(500,"Errore di connessio al db.");
            return;
        }
        //ottengo la url, per gestire le varie richieste
        String url = request.getRequestURL().toString();
        //richiesta esame
        if (url.endsWith("esame")) {
            String id_esame = request.getParameter("id_esame");
            String data_prenotazione = request.getParameter("data_prenotazione");
            //Controlla se i parametro siano presenti
            if (id_esame == null || id_esame.isEmpty() || data_prenotazione == null ||
data_prenotazione.isEmpty()){
                //Restituisce un errore nel caso in cui non lo siano.
                response.sendError(402, "Parametri mancanti");
                return; //serve il return altrimenti il metodo viene eseguito comunque
            }
            try {
                //query che permette di ottenere i dati di uno specifico esame
                String command = "SELECT esame.* FROM esame WHERE id_esame= ? AND data_prenotazione= ?";
                PreparedStatement stat = con.prepareStatement(command);
                stat.setString(1, id_esame);
                stat.setString(2, data_prenotazione);
                System.out.println(stat);
                //eseguo il comando
                ResultSet rs = stat.executeQuery();
                rs.next();
                //costruzione XML

```



```

response.setContentType("text/xml;charset=UTF-8");
PrintWriter w = response.getWriter();
w.println("<?xml version=\"1.0\" encoding=\"UTF-8\"?>");
w.println("<esame>");

    w.print("<id_esame>");
    w.print(rs.getString("id_esame"));
    w.println("</id_esame>");

    w.print("<data_prenotazione>");
    w.print(rs.getDate("data_prenotazione"));
    w.println("</data_prenotazione>");

    w.print("<id_prestazione>");
    w.print(rs.getString("id_prestazione"));
    w.println("</id_prestazione>");

    w.print("<temperatura_corporea_gradi>");
    w.print(rs.getDouble("temperatura_corporea_gradi"));
    w.println("</temperatura_corporea_gradi>");

    w.print("<pressione_sanguigna_mmHG>");
    w.print(rs.getDouble("pressione_sanguigna_mmHG"));
    w.println("</pressione_sanguigna_mmHG>");

    w.print("<frequenza_respiratoria_attiMinuto>");
    w.print(rs.getDouble("frequenza_respiratoria_attiMinuto"));
    w.println("</frequenza_respiratoria_attiMinuto>");

    w.print("<ossigenazione_sanguigna_percentuale>");
    w.print(rs.getInt("ossigenazione_sanguigna_percentuale"));
    w.println("</ossigenazione_sanguigna_percentuale>");

    w.print("<frequenza_cardiaca_bpm>");
    w.print(rs.getDouble("frequenza_cardiaca_bpm"));
    w.println("</frequenza_cardiaca_bpm>");

w.println("</esame>");
w.close();
//set status
response.setStatus(200);

} catch (SQLException ex) {
    //viene sollevata una eccezione se il record non è stato inserito in precedenza
    response.sendError(401, "Record non presente");
    return;
}
//richiesta listaReport
} else if (url.endsWith("listaReport")) {
    String matricola = request.getParameter("matricola");
    //Controlla se il parametro è presente
    if (matricola == null || matricola.isEmpty()) {
        //Restituisce un errore nel caso in cui non lo sia

```

```

response.sendError(402, "Parametri mancanti");
return; //serve il return altrimenti il metodo viene eseguito comunque
}
try {
    ///query che permette di ottenere i dati di tutti i report redatti da uno specifico operatore
    String command = "SELECT report.* FROM report WHERE matricola = ?";
    PreparedStatement stat = con.prepareStatement(command);
    stat.setString(1, matricola);

    //viene eseguito il comando
    ResultSet rs = stat.executeQuery();

    //costruzione XML
    response.setContentType("text/xml;charset=UTF-8");
    PrintWriter w = response.getWriter();
    w.println("<?xml version='1.0' encoding='UTF-8'?'>");
    w.println("<lista_report>");
    while(rs.next()){
        w.println("<report>");

        w.print("<id_esame>");
        w.print(rs.getString("id_esame"));
        w.println("</id_esame>");

        w.print("<data_prenotazione>");
        w.print(rs.getString("data_prenotazione"));
        w.println("</data_prenotazione>");

        w.print("<id_cartella_clinica>");
        w.print(rs.getString("id_cartella_clinica"));
        w.println("</id_cartella_clinica>");

        w.print("<codice_tessera>");
        w.print(rs.getString("codice_tessera"));
        w.println("</codice_tessera>");

        w.print("<matricola>");
        w.print(rs.getString("matricola"));
        w.println("</matricola>");

        w.print("<dataOra_inizio>");
        w.print(rs.getString("dataOra_inizio"));
        w.println("</dataOra_inizio>");

        w.print("<dataOra_fine>");
        w.print(rs.getString("dataOra_fine"));
        w.println("</dataOra_fine>");

        w.print("<descrizione>");
        w.print(rs.getString("descrizione"));
        w.println("</descrizione>");

        w.println("</report>");
    }
}

```

```

    }
    w.println("</lista_report>");
    w.close();
    //set status
    response.setStatus(200);

    } catch (SQLException ex) {
        //viene sollevata una eccezione se il record non è stato inserito in precedenza
        response.sendError(401, "Record non presente");
        return;
    }
}
else{
    //nel caso in cui la richiesta non corrisponde alle richieste gestibili
    response.sendError(400, "richiesta errata");
}
}

/**
 * Handles the HTTP <code>POST</code> method.
 *
 * @param request servlet request
 * @param response servlet response
 * @throws ServletException if a servlet-specific error occurs
 * @throws IOException if an I/O error occurs
 */
@Override
protected void doPost(HttpServletRequest request, HttpServletResponse response)
    throws ServletException, IOException {
    //controllo connessione db
    if(con == null){
        response.sendError(500,"Errore di connessio al db.");
        return;
    }
    //prendo la url, per gestire richiesta specifica
    String url = request.getRequestURL().toString();
    //gestione richiesta nuovoEsame
    if (url.endsWith("nuovoEsame")) {
        BufferedReader input = request.getReader();
        // estrazione dell'xml dal body della richiesta
        StringBuilder xml = new StringBuilder();
        String line;
        // BufferedWriter file = new BufferedWriter(new FileWriter("request.xml"));
        while ((line = input.readLine()) != null) {
            // file.write(line);
            // file.newLine();
            xml.append(line);
        }
        input.close();
        try {
            // parsing dell'xml ricevuto
            DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();

```

```

DocumentBuilder builder = factory.newDocumentBuilder();
Document document;
// il parsing non viene fatto a partire dal file ma dalla stringa
document = builder.parse(new ByteArrayInputStream(xml.toString().getBytes()));
Element root = document.getDocumentElement();
// parsing id_esame
NodeList list = root.getElementsByTagName("id_esame");
String id_esame = null;
if(list != null && list.getLength() > 0){
    id_esame = list.item(0).getFirstChild().getNodeValue();
}
// parsing data_prenotazione
NodeList list1 = root.getElementsByTagName("data_prenotazione");
String data_prenotazione = null;
if(list1 != null && list1.getLength() > 0){
    data_prenotazione = list1.item(0).getFirstChild().getNodeValue();
}
// parsing id_prestazione
NodeList list2 = root.getElementsByTagName("id_prestazione");
String id_prestazione = null;
if(list2 != null && list2.getLength() > 0){
    id_prestazione = list2.item(0).getFirstChild().getNodeValue();
}
// parsing temperatura_corporea_gradi
NodeList list3 = root.getElementsByTagName("temperatura_corporea_gradi");
String temperatura_corporea_gradi = null;
if(list3 != null && list3.getLength() > 0){
    temperatura_corporea_gradi = list3.item(0).getFirstChild().getNodeValue();
}
// parsing pressione_sanguigna_mmHG
NodeList list4 = root.getElementsByTagName("pressione_sanguigna_mmHG");
String pressione_sanguigna_mmHG = null;
if(list4 != null && list4.getLength() > 0){
    pressione_sanguigna_mmHG = list4.item(0).getFirstChild().getNodeValue();
}
// parsing frequenza_respiratoria_attiMinuto
NodeList list5 = root.getElementsByTagName("frequenza_respiratoria_attiMinuto");
String frequenza_respiratoria_attiMinuto = null;
if(list5 != null && list5.getLength() > 0){
    frequenza_respiratoria_attiMinuto = list5.item(0).getFirstChild().getNodeValue();
}
// parsing ossigenazione_sanguigna_percentuale
NodeList list6 = root.getElementsByTagName("ossigenazione_sanguigna_percentuale");
String ossigenazione_sanguigna_percentuale = null;
if(list6 != null && list6.getLength() > 0){
    ossigenazione_sanguigna_percentuale = list6.item(0).getFirstChild().getNodeValue();
}
// parsing frequenza_cardiaca_bpm
NodeList list7 = root.getElementsByTagName("frequenza_cardiaca_bpm");
String frequenza_cardiaca_bpm = null;
if(list7 != null && list7.getLength() > 0){
    frequenza_cardiaca_bpm = list7.item(0).getFirstChild().getNodeValue();
}

```

```

//Conversione di alcuni parametri nei tipi adatti
double temperatura_corporea_gradi_double=0.0;
double pressione_sanguigna_mmHG_double=0.0;
double frequenza_respiratoria_attiMinuto_double=0.0;
int ossigenazione_sanguigna_percentuale_int=0;
double frequenza_cardiaca_bpm_double=0.0;
try {
// Ovviamente la conversione avviene all'interno di un costrutto try-catch per gestire
eventuali eccezioni
temperatura_corporea_gradi_double = Double.parseDouble(temperatura_corporea_gradi);
pressione_sanguigna_mmHG_double = Double.parseDouble(pressione_sanguigna_mmHG);
frequenza_respiratoria_attiMinuto_double =
Double.parseDouble(frequenza_respiratoria_attiMinuto);
ossigenazione_sanguigna_percentuale_int= Integer.parseInt(ossigenazione_sanguigna_percentuale);
frequenza_cardiaca_bpm_double = Double.parseDouble(frequenza_cardiaca_bpm);
} catch (NumberFormatException e) {
response.sendError(400, "Parametri errati");
return;
}
//SQL per inserimento nuovo esame
String command = "INSERT INTO esame values (?, ?, ?, ?, ?, ?, ?, ?)";
PreparedStatement stat = con.prepareStatement(command);
stat.setString(1, id_esame);
stat.setString(2, data_prenotazione);
stat.setString(3, id_prestazione);
stat.setDouble(4, temperatura_corporea_gradi_double);
stat.setDouble(5, pressione_sanguigna_mmHG_double);
stat.setDouble(6, pressione_sanguigna_mmHG_double);
stat.setInt(7, ossigenazione_sanguigna_percentuale_int);
stat.setDouble(8, frequenza_cardiaca_bpm_double);
//eseguo il comando
stat.executeUpdate();
//INVIO STATUS
PrintWriter res = response.getWriter();
res.append("Inserimento riuscito");
response.setStatus(200);
//gestione delle varie eccezioni
} catch (SQLException ex){
response.sendError(401, "Errore durante l'inserimento!");
} catch (ParserConfigurationException ex){
response.sendError(402, "XML parser error!");
}
} catch (SAXException exception){
response.sendError(402, "XML parser error!");
}
}
else{
//nel caso in cui la richiesta non corrisponde
response.sendError(400, "richiesta errata");
}
}
}
/**

```

```
* Returns a short description of the servlet.  
*  
* @return a String containing servlet description  
*/  
@Override  
public String getServletInfo() {  
    return "Short description";  
} // </editor-fold>
```

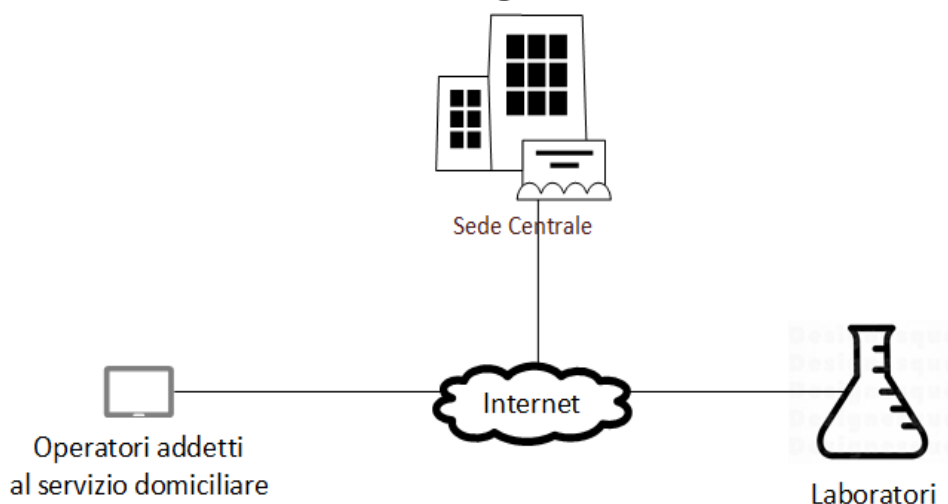
```
//disattivazione servlet (disconnessione da DBMS)
```

```
public void destroy(){  
    try{  
        con.close();  
    }  
    catch(SQLException exception){  
    }  
}
```

```
}
```

Capitolo 8

Infrastruttura tecnologica ed informatica



8.1 Attori della rete

La figura sopra riportata rappresenta i 3 attori principali dell'eventuale infrastruttura di rete:

- La sede centrale, che sarà la rete che ospita i server nel caso di soluzione on-premise, sarà connessa alla rete Internet tramite una connessione in fibra ottica ad alte prestazioni sia in downstream che in upstream (FTTB/FTTH) fornita da un ISP specializzato;
- I laboratori: i dispositivi del laboratorio si conatteranno alla rete internet mediante la connessione wired (xDSL/FTTX). I laboratori costituiscono le sedi periferiche che andranno a interagire con il server per l'aggiornamento dei dati relativi al risultato delle analisi. In questa rete sarà presente un server utile ad effettuare un backup di dati, nel caso in cui si presentino problematiche relative al funzionamento della rete.
- I dispositivi impiegati dagli operatori, che si ipotizza siano dei tablet, opereranno sia in ambienti esterni urbani che rurali e si conatteranno al web server attraverso la rete internet, sfruttando la connettività LTE offerta da operatori di telefonia mobile che rendono disponibili commercialmente SIM dedicate. A livello applicativo questo avverrà mediante un API HTTP esposta dal server sulla rete internet che a sua volta interagirà con il DBMS. Il router che consente l'accesso al server specifico sarà dotato di un indirizzo IPv4/IPv6 pubblico statico ed eventualmente associato ad un nome di dominio registrato e risolto dal servizio DNS.

8.2 Collocazione Server

La collocazione dei server che devono essere disponibili per l'erogazione del servizio, presenta 2 principali differenti soluzioni che verranno discusse e poste in analisi di seguito.

- Soluzione on premises: collocazione del server fisico nella sede centrale; questa soluzione presenta un vantaggio relativo di avere un controllo esclusivo su sistemi e dati. Bensì comporta significative problematiche, tra cui:

- La presenza di un costo dato dall' acquisto e dalla manutenzione dell'hardware, che eventualmente comprende quanto necessario per garantire il funzionamento anche in mancanza di alimentazione della rete elettrica;

- La presenza di un costo relativo alla connettività alla rete internet con una banda adeguata per l'accesso al server da parte dei vari utenti;

- La formazione sistemistica del personale dell'asl, per la configurazione e gestione della rete DMZ e del server, in particolare allo scopo di garantirne e monitorarne la sicurezza rispetto ad accessi non autorizzati da parte del personale aziendale e ad attacchi informatici finalizzati alla violazione della riservatezza e dell'integrità dei dati, oltre che alle potenziali cause di disservizio e perdita dei dati.

- Cloud: La soluzione precedentemente esposta presentava vari aspetti negativi, infatti risulta essere fortemente consigliabile adottare una soluzione cloud, che nel caso specifico dello scenario delineato, può essere di due tipologie, entrambe offerte commercialmente a fronte di contenuti canonici mensili:

- IaaS(Infrastructure as a Service, ad esempio Amazon AWS): permette la configurazione di un server virtuale su cui ospitare i servizi software sviluppati.

- PaaS(Platform as a Service, ad esempio Google APP engine): permette lo sviluppo di applicazioni web e dei servizi che espongono API HTTP con vari linguaggi di programmazione(tramite Java e PHP) utilizzando i servizi (ad esempio Google Maps, o un databaseSQL) integrati della piattaforma.

Nell'ipotesi in cui i server siano collocati fisicamente sulla rete della sede centrale dell'asl, essi avranno un'architettura PC dotata di sistema operativo Windows o Linux. Sia in caso di soluzione on premise e soluzione cloud, i server dovranno ospitare un SMTP, un DBMS MySQL, un web server Apache HTTP per l'esecuzione di una applicazione web realizzata in linguaggio PHP ed un'altro web server Apache Tomcat per la pubblicazione delle API HTTP a cui accede il sito e la APP sui dispositivi utilizzati dal personale durante il servizio a domicilio.

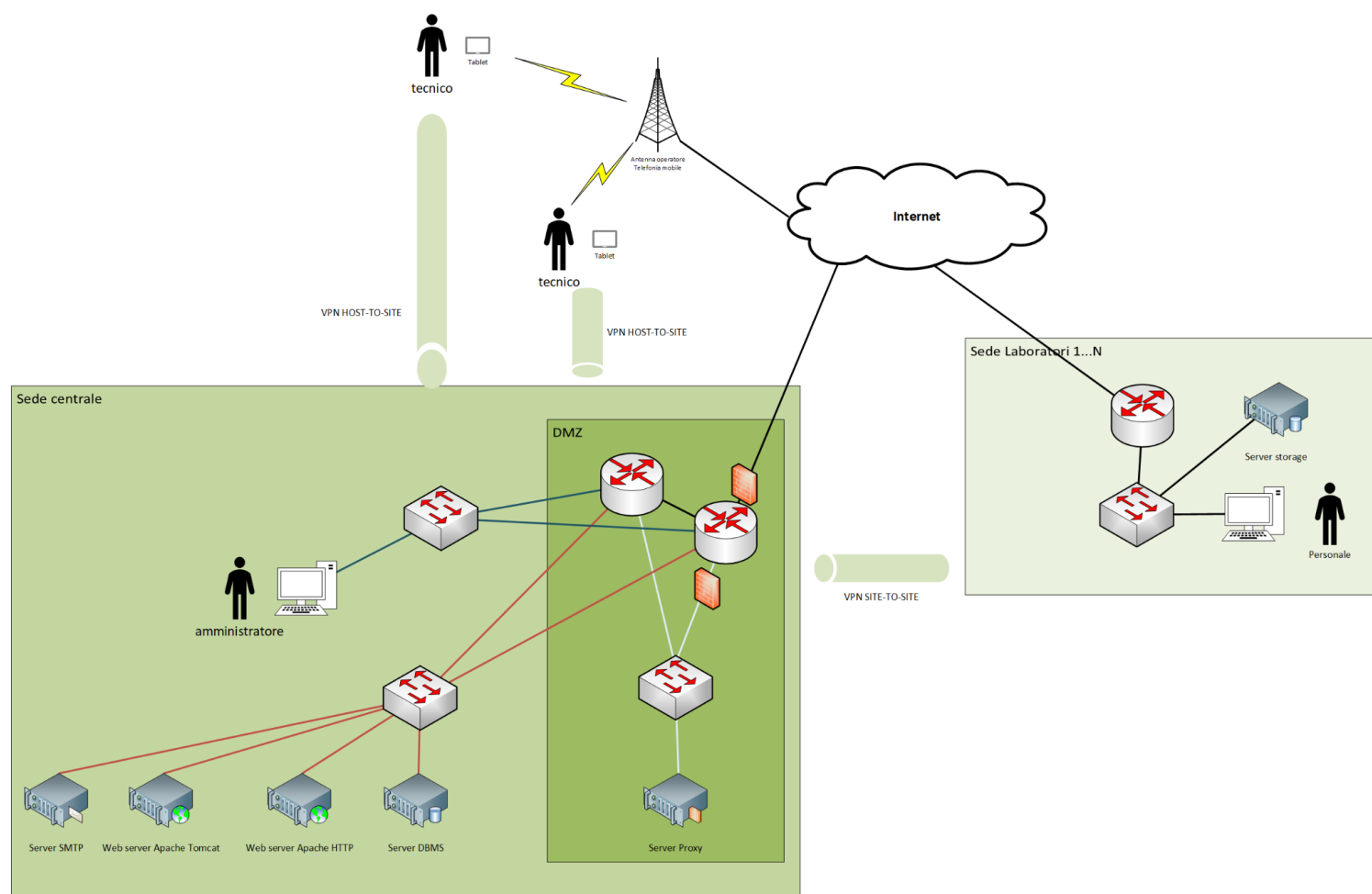
Inoltre il dimensionamento dei server, viste le “operatività” di quest'ultimi si ipotizza che avranno delle adeguate caratteristiche prestazionali e di robustezza: 2/4 processori, memoria

RAM di alcune decine di GB, dischi multipli con velocità di rotazione pari o superiore a 10Krpm accoppiati in modalità RAID, doppio alimentatore. Questi ultimi due aspetti permettono di implementare una ridondanza che riduca una probabilità di guasto.

Dovrà inoltre essere previsto un sistema UPS per la continuità dell'alimentazione elettrica dei server e dei dispositivi di rete che lo connettono alla rete Internet (switch, router, modem, ...) con una batteria la cui durata possa ovviare alle più comuni cause di mancanza dell'alimentazione primaria. il contratto stipulato con lo ISP dovrà prevedere specifici SLA che garantiscano la connettività alla rete Internet praticamente senza discontinuità apprezzabili nell'arco dell'anno solare. Ovviamente non vi è una ridondanza totale nella soluzione on-premise poiché quest'ultima porterebbe ad un costo non sensato per una organizzazione come l'ASL, questo è un motivo ulteriore che porta a favorire una soluzione cloud rispetto a quella di mantenere il server presso la sede aziendale.

8.3 Soluzione on-premise

Nella situazione in cui i server siano fisicamente interni, il possibile *layout* della rete da realizzare potrebbe essere il seguente:



Si può dunque notare due aspetti principali:

- La presenza di un server proxy.
- La presenza di ridondanza a livello del Default Gateway.

La ridondanza a livello del Default gateway garantirà nel caso di guasto di un singolo router, di non causare l'interruzione del servizio. Tutto ciò avviene mediante l'utilizzo di un router virtuale avente un MAC ed un IP virtuale. Gli eventuali protocolli utilizzati saranno facenti parti della famiglia di protocolli FHRP (First Hop Redundancy Protocol), in questo scenario si potrebbe utilizzare VRRP (Virtual Router Redundancy Protocol). Ovviamente creando ridondanza al Default gateway, si presenteranno più collegamenti fisici che saranno gestiti dallo specifico protocollo STP (Spanning Tree Protocol).

Inoltre si potrebbe presupporre un doppio contratto con due ISP differenti in modo da avere una “ridondanza” maggiore.

Per quanto riguarda il server proxy, sarà collocato nella DMZ, la quale rappresenta l’unica sezione della rete accessibile da Internet in base alle regole configurate sul router che agisce come firewall. La presenza di un server proxy consente di innalzare il livello di sicurezza del server. In tal caso tutte le richieste ricevute dagli operatori e dai pazienti, saranno dirette al server proxy che le dirigerà al server di gestione da cui riceve le risposte che inoltra al client che le ha richieste.

I server proxy di oggi giorno utilizzano delle tecniche algoritmiche di intelligenza artificiale, riuscendo ad ottenere una sicurezza maggiorata.

Nell’eventuale ipotesi di configurare sul router un NAT statico per associare l’indirizzo IP del server proxy a un indirizzo IP pubblico specifico, e un NAT/PAT dinamico per consentire la navigazione della rete internet da parte degli host della rete degli uffici, tutti gli indirizzi IP della rete possono essere privati; il seguente è un possibile schema:

Rete	Indirizzo/netmask	Indirizzi
DMZ	192.168.0.0/24	proxy: 192.168.0.1 Default gateway: 192.168.0.254
Server	192.168.1.0/24	server web server Apache HTTP: 192.168.1.1 server web server Apache Tomcat: 192.168.1.2 server DBMS: 192.168.1.3 server SMTP: 192.168.1.4 Default gateway: 192.168.1.254
Uffici	192.168.2.0/24	host: 192.168.2.1-192.168.2.253 Default gateway: 192.168.2.254

Il router sarà configurato come firewall con il compito di realizzare la rete DMZ che ospita il server proxy, in modo da difendere la rete che ospita il server di gestione del servizio da attacchi esterni e consentire la navigazione da parte degli host della rete degli uffici su internet; la tabella che segue riassume le regole di accesso configurate per questi scopi:

WAN-IN:

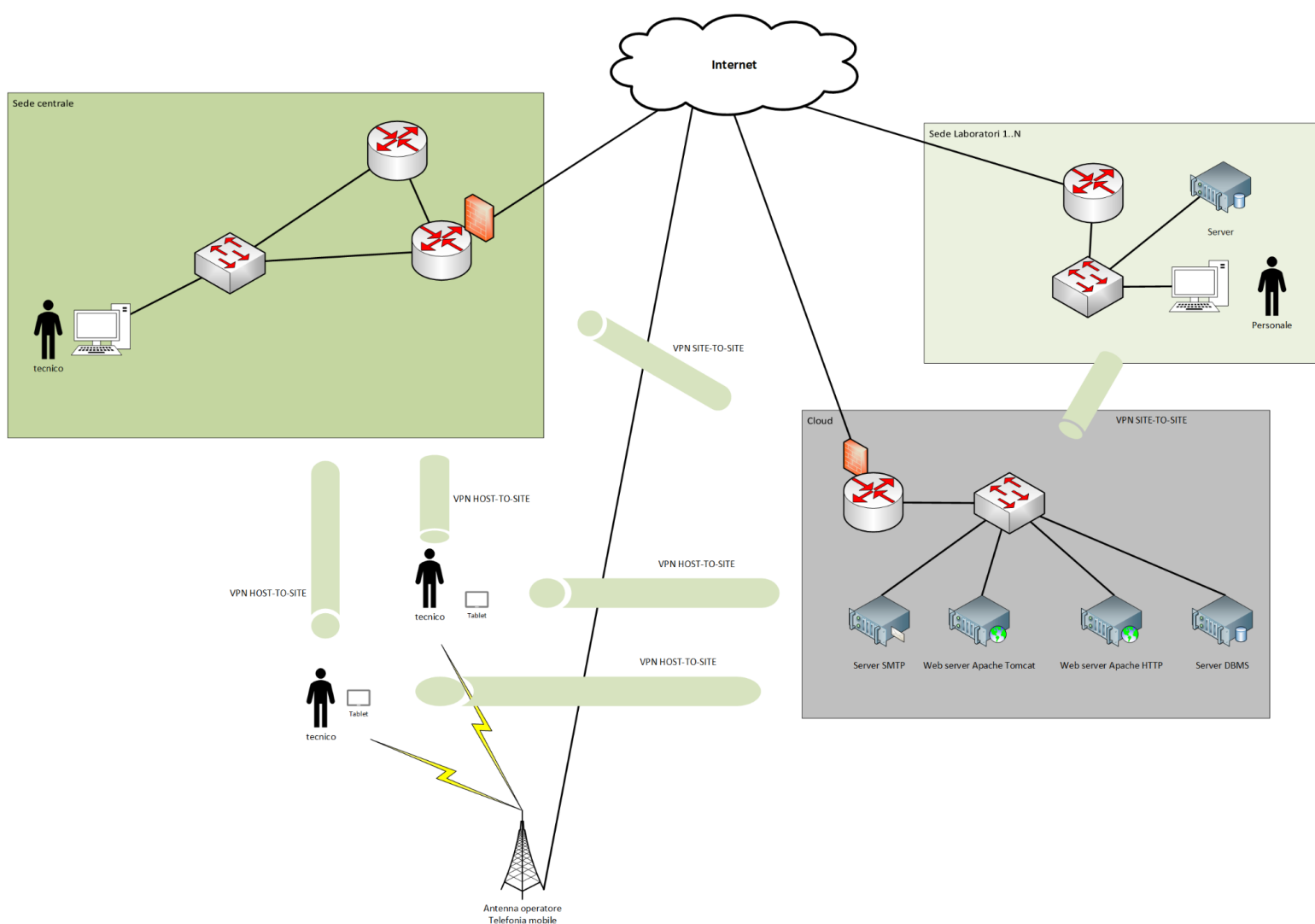
Protocollo	Origine		Destinazione		permesso
TCP	qualsiasi	qualsiasi	192.168.0.1 0.0.0.0	443(https) 8443(https)	concesso
TCP	qualsiasi	qualsiasi	192.168.0.0 0.0.0.255	qualsiasi	rifiuta
TCP	qualsiasi	qualsiasi	192.168.1.0 0.0.0.255	qualsiasi	rifiuta
TCP	qualsiasi	qualsiasi	192.168.2.0 0.0.0.255	qualsiasi	concesso solo per connessione stabilita
UDP	qualsiasi	53(DNS)	192.168.2.0 0.0.0.255	qualsiasi	concesso
UDP	qualsiasi	qualsiasi	192.168.1.0 0.0.0.255	qualsiasi	rifiuta
ICMP	qualsiasi	tipo 8 (ping)	qualsiasi	qualsiasi	rifiuta
ICMP	qualsiasi	qualsiasi	192.168.0.1 0.0.0.255	qualsiasi	concesso
qualsiasi	qualsiasi	qualsiasi	qualsiasi	qualsiasi	rifiuta

DMZ-IN:

Protocollo	Origine		Destinazione		permesso
TCP	192.168.0.1	qualsiasi	192.168.1.1 0.0.0.0	80(http)	concesso
TCP	192.168.0.1	qualsiasi	192.168.1.2 0.0.0.0	8080(Tomcat)	concesso
TCP	192.168.0.1	qualsiasi	192.168.1.3	3306(DBMS)	concesso

			0.0.0.0)	
TCP	192.168.0.1	qualsiasi	192.168.1.4 0.0.0.0	25(SMTP)	concesso
TCP	qualsiasi	qualsiasi	192.168.2.0 0.0.0.255	qualsiasi	rifiuta
TCP	192.168.0.1	443(https) 8443(https)	qualsiasi	qualsiasi	concesso solo per connessioni stabilite
qualsiasi	qualsiasi	qualsiasi	qualsiasi	qualsiasi	rifiuta

8.4 Soluzione cloud



Nell'ipotesi in cui il server sia collocato sul cloud, il contratto firmato dall'asl con l'azienda fornitrice del servizio, dovrà garantire una sicurezza tale per la quale la presenza di un eventuale server proxy, che fornisce una maggiore sicurezza risulti del tutto inutile.

8.5 Dispositivi mobili utilizzati dagli operatori addetti al servizio domiciliare

I tablet utilizzati dagli operatori addetti al servizio domiciliare saranno forniti in comodato d'uso dalla sede centrale, avranno una tipologia uniforme in relazione al sistema operativo (Android) e

alla dimensione/risoluzione dello schermo, in questo modo si potrà garantire una corretta esecuzione dell'APP. Le funzionalità dell'APP non richiedono caratteristiche particolari per il tablet che la eseguono, se non una connettività LTE, un dispositivo GPS in modo che gli operatori addetti al servizio domiciliare siano costantemente localizzati sia per far sì che nel caso di urgenze specifiche segnalate dai pazienti, la sede centrale riesca a fornire un servizio adatto, sia per controllare l'operatività del personale anche per una questione di sicurezza, essendo comunque l'operatore a stretto contatto con altre persone; Inoltre il dispositivo GPS permette l'uso di navigatore stradale per raggiungere i domicili. La visualizzazione di quest'ultimi su mappa sarà realizzata impiegando la specifica libreria per l'accesso ai servizi Google maps. Per quanto riguarda invece l'aspetto della sicurezza, considerando la tipologia dei dati che la APP gestisce e scambia con il server, si ritengono sufficienti le normali misure di protezione di un dispositivo personale (aggiornamento regolare del S.O. , autenticazione mediante password/PIN di accesso, impostazione dell'auto-disconnessione a tempo dell'utente, installazione/aggiornamento della APP tramite lo store,...).

Per quanto riguarda la continuità di servizio per i tablet, in particolare nelle zone rurali, la connettività alla rete Internet potrebbe non essere sempre assicurata: la APP a tal proposito dovrà di conseguenza memorizzare una copia limitata dei dati relativi al funzionamento di base in modo da poter essere utilizzata anche in assenza di connessione, prevedendo una successiva sincronizzazione dei dati con il server quando risulta essere di nuovo disponibile.

8.6 Autenticazione

Gli operatori del laboratorio, i clienti, gli operatori amministrativi e quelli relativi al servizio domiciliare interagiscono con il server mediante un'applicazione web eseguita nel contesto di un web-server, esposta mediante il protocollo applicativo HTTP e realizzata con tecnologie lato client convenzionali (linguaggio HTML, CSS e JavaScript), di conseguenza risulta accessibile mediante un qualsiasi browser. Questo fattore rende necessaria l'autenticazione e l'autorizzazione dei pazienti e degli operatori; A tale scopo sono praticabili in particolare due soluzioni:

- prevedere delle VPN end-to-end, realizzate ad esempio impiegando il software Open VPN, con credenziali specifiche per i singoli pazienti e i singoli operatori del laboratorio;
- generare e richiedere credenziali di accesso (email/password) per i singoli utenti.

La seconda soluzione prevede la gestione delle credenziali da parte dell'applicazione software che memorizzerà in un database un opportuno hashing delle password, associate all'indirizzo email che invece sarà memorizzato in chiaro. Nel caso in cui non si faccia utilizzo delle VPN, sarà però necessario esporre il servizio ricorrendo al protocollo applicativo HTTPS, che impiega il protocollo TLS sopra lo stack dei protocolli TCP/IP e consente l'autenticazione del server mediante un certificato X.509 rilasciato da un'autorità di certificazione.

Per l'interazione dei dispositivi utilizzati dagli operatori sul posto del domicilio, con il server sono essenzialmente possibili due alternative:

- L'implementazione di un protocollo applicativo proprietario utilizzando un protocollo del livello di trasporto (UDP o TCP)
- La configurazione di un web-service di tipo REST esposto mediante il protocollo HTTP. L'APP in esecuzione sui tablet deve essere autenticata ed autorizzata. nel caso in cui si adotti la prima soluzione potrebbe, ad esempio, essere impiegato il protocollo SSH per creare un tunnel di comunicazione cifrato e mutuamente autenticato tra client e server; nel caso invece si opti per la seconda opzione, deve essere impiegato il protocollo HTTPS autenticando il server mediante un certificato X.509 rilasciato da un'autorità di certificazione, mentre l'autenticazione client avverrà successivamente.

8.7 VPN

Data la necessità di avere una comunicazione privata, l'infrastruttura necessita di una rete privata virtuale (VPN - Virtual Private Network) che garantisca privacy e sicurezza tra due reti remote attraverso un canale di comunicazione logicamente riservato - chiamato tunnel VPN - che viene creato sopra un'infrastruttura di rete pubblica come Internet.

Le due tipologie di VPN utilizzate sono:

- Site-to-Site: in questo caso la VPN viene configurata sul router. Questo fa in modo che i due router abbiano una connessione cifrata e sicura. Nello scenario on-premise sono utilizzate per connettere i laboratori con la sede centrale. Questo permetterà di accedere al sito web e memorizzare il risultato delle analisi nel database. Nello scenario Cloud sono utilizzate per connettere la sede centrale ed i vari laboratori ai server presenti nel cloud.
- Host-to-Site: Sono utilizzate per connettere tipicamente 2 utenti in remoto. Nello scenario on-premise, permettono di connettere i dispositivi degli operatori addetti al servizio domiciliare alla sede centrale, in questo modo si supporteranno le funzionalità della APP (comunicazione mediante chat, possibilità di inserimento dei report nel db). Nello scenario Cloud sono utilizzate per connettere i dispositivi degli operatori addetti al servizio domiciliare sia con il cloud, in modo da permettere la memorizzazione dei report nel database, sia con la sede centrale per l'apposita sezione di live chat, messa a disposizione dalla APP in esecuzione sui tablet.

Uno dei protocolli principali per la realizzazione di VPN su Internet è IPsec, quest'ultimo fornisce due metodologie di configurazione dei tunnel VPN. Nel caso di una VPN Site-to-Site, verrà utilizzata una modalità di configurazione denominata «Tunnel Mode», ovvero una configurazione che permette la cifratura completa del pacchetto IPv4/IPv6. Mentre nel caso di una VPN Host-to-Site sarà possibile utilizzare servizi come OpenVPN per la realizzazione del tunnel VPN.

