

ALMA MATER STUDIORUM · UNIVERSITÀ DI BOLOGNA

SCUOLA DI SCIENZE
CORSO DI LAUREA IN INFORMATICA PER IL MANAGEMENT

**PROGETTAZIONE E IMPLEMENTAZIONE
DI UN WEB INFORMATION SYSTEM
PER IL MONITORAGGIO REMOTO
DI PAZIENTI CARDIOPATICI**

Relatore:
Chiar.mo Prof.
Marco Di Felice

Presentata da:
Martina Ianaro

Correlatore:
Dott.
Ivan Corazza

Sessione II
Appello II
Anno Accademico 2019-2020

"Non ho particolari talenti, sono soltanto appassionatamente curioso"

Albert Einstein

"Non fermatevi là dove siete arrivati"

Pitagora

Abstract

E' di importanza sempre crescente l'alleggerimento del sistema sanitario a causa del sovraffollamento degli ambienti sanitari e delle finite risorse ospedaliere.

La Smart Health fornisce un valido supporto a tale problematica, migliorando lo stile di vita dei pazienti e facilitando il lavoro dei team sanitari.

In questo contesto si inserisce il seguente lavoro di tesi che realizza un sistema back-end per la gestione dei dati dei pazienti in possesso di un app mobile per il monitoraggio giornaliero.

Il Web Information System realizzato è finalizzato al miglioramento della gestione del follow up dei pazienti cardiopatici in cura presso il Policlinico Sant'Orsola-Malpighi di Bologna e potrà essere utilizzato da altri team sanitari nella gestione di diverse patologie.

Tale sistema viene illustrato descrivendo tutte le varie fasi di sviluppo, dalla progettazione all'implementazione, in seguito all'esposizione dello stato dell'arte della Smart Health.

Indice

Introduzione	v
1 Stato dell'Arte	1
1.1 Smart Health	1
1.2 Diffusione della Smart Health a livello mondiale	2
1.2.1 eHealth	7
1.2.2 mHealth	8
1.3 Applicazioni in ambito biomedicale	12
1.3.1 Applicazione mHealth	13
1.3.2 Applicazioni eHealth	18
2 Progettazione	27
2.1 Specifiche	28
2.2 Progettazione dei dati	30
2.2.1 Decomposizione del testo	32
2.2.2 Tabella glossario	35
2.2.3 Dizionario delle Entità	36
2.2.4 Dizionario delle Relazioni	37
2.2.5 Business Rules	38
2.2.6 Diagramma E-R	38
2.2.7 Ristrutturazione dello schema E-R	39
2.2.8 Traduzione nel modello logico	40
2.3 Progettazione interfaccia utente - Use Case	42
2.4 Architettura	49

3	Implementazione	51
3.1	Tecnologie	51
3.1.1	Architettura di Angular	52
3.1.2	PHP	56
3.1.3	MySQL (Structured Query Language)	63
3.1.4	Apache	67
3.2	Implementazione dell'applicativo in Angular	67
3.3	Fase di Testing delle API REST	76
3.4	App Mobile	77
4	Validazione	79
4.1	Implementazione delle funzionalità	79
4.1.1	Casi D'uso	87
4.1.2	L'infermiere valida una rilevazione	87
4.1.3	Il medico inserisce una terapia	93
5	Conclusioni e sviluppi futuri	101
5.1	Sviluppi futuri	102
	Appendice	110
	Bibliografia	111
	Ringraziamenti	116

Elenco delle figure

1.1	Home view of Breast Cancer e-Support	4
1.2	smart health	11
1.3	Electric Jacket	13
1.4	Schema logico	14
1.5	Architettura hardware	15
1.6	Activity Diagram	16
1.7	View dati	17
1.8	View allarme	18
1.9	Architettura sistema gestionale	20
1.10	Logica del sistema gestionale	22
1.11	ML logic	23
1.12	Performance DL-ML	24
1.13	Performance NN-ML	24
2.1	Progettazione di una base di dati	30
2.2	Artefatti della progettazione della base di dati	31
2.3	WIS Architecture	41
2.4	Diagramma dei casi d'uso	44
2.5	Architettura Heart	49
3.1	Componente Angular	53
3.2	Architettura Angular	54
3.3	Ajax web application model	55
3.4	RDBMS	63

Introduzione

Le fonti Ministeriali affermano che *"le malattie cardiovascolari....sono tra le principali cause di morbosità, invalidità e mortalità in Italia....e rappresentano ancora la principale causa di morte nel nostro Paese, essendo responsabili del 38,5% di tutti i decessi".*[1]

Il 5 Ottobre 2020 si è tenuto al Senato il VI° Convegno Nazionale Annuale AISC (Associazione Italiana Scompensati Cardiaci) dal titolo *"La gestione dello scompenso cardiaco nell'era post COVID: parliamone con i pazienti"*. Durante il convegno è stato evidenziato che in Italia i pazienti con scompenso cardiaco sono oltre 1 milione ed in Europa 15 milioni. L'incidenza della malattia aumenta del 2% al crescere dell'età, fino a raggiungere il 10% nei pazienti over 70. Oltre il 25% degli affetti muore entro un anno dalla diagnosi e circa la metà entro 5 anni dal giorno in cui viene certificata la malattia. In più, ogni ricovero ospedaliero correlato allo scompenso cardiaco triplica il rischio di morte entro dodici mesi.[2]

A causa della pandemia è stata rivalutata la centralità della telemedicina che consente allo staff medico di comunicare con i pazienti senza far loro correre il rischio di un trasferimento in ospedale.

Il professore di Medicina dell'Università La Sapienza e responsabile emergenze territoriali dell'Asl di Latina Salvatore Di Somma afferma durante il VI°Convegno Nazionale Annuale AISC :

Nella fase iniziale della pandemia bisognava ridurre gli afflussi inappropriati al pronto soccorso. Mettere in sicurezza i pazienti positivi al COVID ed evitare il contagio dei negativi, specie se affetti da una condizione cronica e da fragilità che ne avrebbero peggiorato le condizioni. Era necessario inoltre alleggerire il carico delle strutture ospedaliere, completamente assorbite dall'e-

mergenza. Ecco, allora, i servizi di tele-monitoraggio domiciliare. Il progetto pilota che abbiamo messo in atto presso la Asl di Latina ha previsto la consegna di un Kit a domicilio del paziente per la rilevazione di quattro parametri da inviare due volte al giorno alla centrale operativa attiva h24. In caso di parametri alterati scattava un allarme che allertava il medico specialista in pneumologia e un relativo protocollo di assistenza.[2]

Per sei mesi sono stati presi in carica 780 pazienti tra cui 325 positivi al COVID con contestuali patologie, quali diabete e obesità. Di questi, 24 hanno avuto bisogno di cure urgenti erogate tempestivamente che hanno evitato il ricovero in terapia intensiva.[2]

Il dottor Giorgio Casati, direttore Generale dell'Asl di Latina, membro del VI° Convegno Nazionale Annuale AISC :

Il tele-consulto nei pazienti fragili può integrare il tessuto della medicina territoriale in un'ottica di innovazione per prevenire le complicanze di alterazioni delle condizioni dei pazienti. Il sistema ha il vantaggio della continuità e della possibilità di fornire un intervento personalizzato e rapido. Vorremmo inserire un sistema del genere per pazienti con scompenso cardiaco non solo per monitorare le condizioni in tempo reale ma anche per fornire le informazioni su terapie e stili di vita utili al controllo della malattia e della qualità dell'esistenza in un'ottica di medicina di iniziativa.[2]

Pertanto è evidente che, al fine di prevenire nuovi eventi cardiaci, i pazienti cardiopatici necessitano di un monitoraggio continuo che si realizza attraverso la misurazione frequente, almeno giornaliera, di alcuni parametri fisici.

Le attività di misurazione, fino ad alcuni decenni fa, dovevano essere eseguite presso gli ambulatori a causa della complessità di utilizzo ed il notevole costo degli strumenti di misurazione. Va tenuto in considerazione ,inoltre, che spesso il paziente cardiopatico ha una età avanzata ed ha difficoltà a muoversi autonomamente.

Negli ultimi decenni la diffusione di apparecchiature domestiche per la misurazione di parametri fisiologici ha consentito sempre di più ai pazienti di effettuare direttamente la misurazione di tali parametri. Inoltre, con la recente diffusione delle tecnologie Web e

dei Mobile Device è stato eliminato anche l'ultimo ostacolo alla condivisione in real-time delle misurazioni con gli ambulatori delle strutture sanitarie.

E' chiaro che l'automazione del processo di monitoraggio dei pazienti cardiopatici comporta notevoli vantaggi sia per il paziente che per la struttura sanitaria.

Di seguito si elencano alcuni aspetti:

- autonomia del paziente nelle attività di misurazione;
- limitazione degli spostamenti per recarsi presso le strutture sanitarie;
- analisi dei valori misurati in real-time;
- immediato feed-back per il paziente;
- limitati ambienti delle strutture sanitarie dedicate al monitoraggio;
- limitato impiego di personale sanitario.

I vantaggi sopra elencati comportano, senza alcun dubbio, da una parte un miglioramento della qualità della vita del paziente e dall'altra un servizio efficiente ed efficace con notevole riduzione di risorse sia economiche che umane.

A ciò va aggiunto che la disponibilità, da parte delle strutture sanitarie, di dati strutturati ed aggiornati consente di effettuare un'attività di prevenzione più efficace tramite l'impiego di algoritmi statistici e di intelligenza artificiale per la classificazione dei pazienti.

La tesi di laurea è finalizzata alla realizzazione di un Sistema Informatico Web sperimentale da utilizzare presso il reparto di cardiologia del Policlinico S.Orsola-Malpighi di Bologna. Esso si integra con precedenti implementazioni software realizzate dal DISI dell'Università di Bologna. Il servizio di cardiologia del Policlinico S.Orsola-Malpighi ha la necessità di controllare giornalmente i valori dei parametri sanitari di due tipologie di pazienti cardiopatici, con e senza dispositivo VAD. Il dispositivo VAD è una pompa meccanica che assiste uno o entrambi i ventricoli nello svolgimento della loro funzione.

Una prima fase per l'automazione del processo di monitoraggio ha visto la realizzazione di un App Mobile che consente ai pazienti di registrare giornalmente i valori dei

parametri sanitari ed inviarli periodicamente al reparto di cardiologia. I dati vengono ricevuti via e-mail ogni 15 giorni raggruppati in un file contenente le rilevazioni inserite dal paziente nelle ultime due settimane. La gestione in back-office dei dati è completamente manuale, pertanto è necessario estrapolarli manualmente dalla e-mail e memorizzarli in un apposito file per analizzarli, confrontandoli con i valori precedentemente comunicati.

L'obiettivo del progetto è fornire uno strumento che consenta la memorizzazione centralizzata ed immediata, direttamente da App Mobile, dei dati in modalità strutturata; fornire al personale ospedaliero delle funzionalità per il monitoraggio immediato dei dati del paziente con l'aiuto di rappresentazioni grafiche.

Inserendo, inoltre, note e terapie sempre aggiornate (farmacologiche o suggerimenti per lo stile di vita da seguire) è possibile monitorare in real-time la loro efficacia sull'evoluzione della malattia. Le informazioni sui pazienti e la loro evoluzione terapeutica vengono condivise tra i diversi operatori sanitari della struttura di monitoraggio.

Le funzionalità del sistema, per ora individuate, sono:

- Inserimento/Modifica Utilizzatore (Admin)
- Inserimento/Modifica Paziente (Infermiere)
- Validazione Rilevazioni (Infermiere/Medico)
- Gestione Terapie (Medico)
- Esportazione dati (Infermiere/Medico)
- Visualizza profilo e modifica password (Infermiere/Medico/Admin)

L'obiettivo di questo lavoro è sperimentale. In seguito ad una esposizione sullo stato dell'arte, ossia la letteratura a riguardo dello sviluppo della Smart Health, presenteremo in modo approfondito la progettazione e l'implementazione del Web Information System come mezzo per la gestione dello stato di salute di pazienti monitorati da remoto. Per concludere, saranno mostrati alcuni casi d'uso dell'applicazione con le relative interfacce come forma di validazione implicita del lavoro.

La Tesi è composta da 5 capitoli:

-
- nel **Capitolo 1-Stato dell'Arte** è descritto lo stato dell'arte della Smart Health;
 - nel **Capitolo 2-Progettazione** è descritta dettagliatamente l'attività di progettazione dei dati e delle diverse componenti del sistema;
 - nel **Capitolo 3-Implementazione** sono descritte le tecnologie impiegate per l'implementazione del sistema e la descrizione del sistema effettivamente realizzato con relativi frammenti di codice;
 - nel **Capitolo 4-Validazione** vi è una descrizione della fase di validazione che consiste nell'illustrazione delle interfacce utente e dell'utilizzo del sistema tramite due casi d'uso;
 - nel **Capitolo 5-Conclusioni e Sviluppi Futuri** si riassume il contenuto dell'elaborato e si accenna alle possibili evoluzioni a cui tale sistema potrà essere sottoposto.

Capitolo 1

Stato dell'Arte

1.1 Smart Health

La Smart Health che negli ultimi anni ha raggiunto un livello di primo piano tra gli obiettivi di innovazione in ambito ICT, si sta rivelando strategica con la pandemia COVID-19 in atto.

La Commissione Europea nel piano strategico di Marzo 2020 ha inserito la Smart Health tra gli indirizzi fondamentali del piano per la salute per i prossimi anni.

La Smart Health si compone di tutte quelle tecnologie di misurazione, elaborazione e comunicazione impiegate per lo sviluppo di soluzioni ICT per la persona che consentono di rispondere alle necessità di salute e benessere, necessità sempre più stringenti a causa dell'invecchiamento della popolazione europea. Essa consente alle strutture preposte alla cura delle persone di raggiungere elevati livelli di efficienza ed efficacia sia nelle attività cliniche che nella prevenzione. Le soluzioni tecnologiche combinano mobile/wearable device, connettività dei dati per mobile device, sviluppi software, tecnologie per sensori, analisi di Big Data, Cloud Computing consentendo nuove modalità di gestione dei pazienti, il monitoraggio della salute di remote comunità, la prevenzione di stili di vita dannosi.

E' pertanto facilmente prevedibile la capacità della Smart Health di trasformare profondamente il settore della healthcare.

I provider di soluzioni SH offrono agli ospedali ed alle organizzazioni healthcare non

solo l'opportunità di maggiore efficienza ed efficacia, ma soprattutto di rendere più moderni i loro sistemi dedicati alla salvaguardia della salute delle persone. Tali fornitori di soluzioni combinano la conoscenza delle tecnologie utilizzate nella SH con lo specifico Know-How in campo sanitario, del benessere e delle malattie.

Ciò porta alla realizzazione di nuove possibilità di cura e prevenzione. L'innovazione in ambito SH necessita di continui sviluppi nel campo ICT, tecnologie di e-location e connessioni broadband. Inoltre, le modifiche economiche e sociali dovute all'invecchiamento della popolazione europea incoraggiano sia il settore pubblico che privato ad esplorare le opportunità offerte da Smart Health. Gli ostacoli al suo sviluppo sono la disponibilità di tecnologie per infrastrutture non idonee alle strutture healthcare, un approccio non propenso all'introduzione delle innovazioni in ambito sanitario, mancanza di un approccio comune a livello europeo circa i dati sanitari da gestire e limitate risorse finanziarie a disposizione delle strutture sanitarie. L'approccio per superare questi ostacoli potrebbe essere quello di sostenere i pilastri tecnologici e finanziari sui quali si fonda la SH continuando ad investire in mobile ICT, connettività broadband, ed innovazione in SH. Incrementando gli investimenti nei settori pubblici e le opportunità di dimostrazioni sperimentali è possibile rimuovere le barriere che i fornitori di soluzioni SH incontrano nel mercato della salute europeo.[4]

1.2 Diffusione della Smart Health a livello mondiale

Di seguito sono stati riportati alcuni provvedimenti messi in atto da diversi paesi del mondo per favorire la divulgazione della telemedicina, da provvedimenti di natura legislativa ad invenzioni tecnologiche finalizzate ad innovare il sistema sanitario.

La legge tedesca Digitale-Versorgung-Gesetz, entrata in vigore nel 2020, si concentra sul miglioramento delle cure attraverso l'impiego della digitalizzazione e dell'innovazione includendo Health App, EHRs, cure virtuali e dati sanitari strutturati dei pazienti da condividere come base per DaaP (Data as a Platform).[5]

Gli Stati Uniti già a Febbraio 2009 hanno emanato l'HITECH Act (Health Information Technology for Economic and Clinical Health Act) per stimolare l'impiego di tecnologie ICT in campo sanitario (electronic health records-EHRs). Il provvedimento

incoraggia la sanità ad adottare sistemi per gestire gli EHRs assicurando la protezione dei dati sanitari. Tutto ciò attraverso incentivi fiscali ed aumentando le penalità nel caso di violazione delle regole HIPAA per la Privacy e la Sicurezza dei dati. Prima dell'introduzione dell'HITECH Act solo il 10% degli ospedali aveva adottato gli EHRs, pertanto per aumentare l'efficienza, il coordinamento e lo scambio dei dati sanitari si è ritenuto necessario estenderne l'adozione. L'HITECH Act introduce incentivi per incoraggiare gli ospedali ad attuare il cambiamento. Le strutture sanitarie che hanno adottato gli EHRs sono passate dal 3,2% del 2008 al 14,2% del 2015. Nel 2017, l'86% degli *office-based physician* (medico di base) avevano adottato un EHR ed il 96% degli ospedali per *acute care* (terapie intensive) non-federali avevano implementato sistemi di IT certificati per la gestione di dati sanitari. Inoltre gli incentivi sono stati utilizzati per assicurare il rispetto delle regole HIPAA relative alla Privacy ed alla Sicurezza, al fine di rendere confidenziali le informazioni sanitarie personali ed assicurare ai pazienti la possibilità di avere copia dei propri dati sanitari quando richiesti.[6]

Uno dei più grandi sviluppi è il Data Center nella provincia cinese dello Guizhou che completa la rete nazionale cinese di dati sanitari. La rete si compone di ben cinque Data Center. I ricercatori usano i dati ricevuti da tecnologie mediche e digitali per analizzarli utilizzando la statistica inferenziale. Sono state effettuate simulazioni per testare l'efficacia di alcuni interventi sanitari ipotizzati, utilizzando l'AI e Machine-learnig.

"Breast Cancer e-Support" è un'applicazione mobile per supportare le donne affette da cancro al seno sottoposte a chemioterapia. Donne reclutate da due ospedali affiliati all'università in Cina sono state inserite in un programma di gestione delle cure che combina cure di routine con l'accesso al programma e-support durante i 4 cicli di chemioterapia.[21]

Tale applicazione prevede 5 funzionalità:

- un forum di apprendimento;
- un forum di discussione;
- un forum Ask-the-Expert;
- un forum di storie personali.

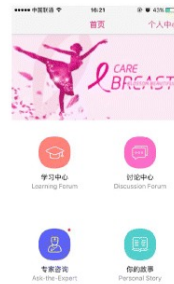


Figura 1.1: Home view of Breast Cancer e-Support

Wellthy Therapeutics è una Mobile APP, realizzata in India, che utilizza l'AI per aiutare i malati diabetici di Tipo-2 a cambiare il loro stile di vita attraverso ricerche di informazioni generate da dati personalizzati.[7] I ricercatori hanno fatto uso dei social media per monitorare le reazioni dannose dei medicinali sui pazienti. Ad esempio, alcuni ricercatori dell'Università Yonsey in Corea stanno sviluppando videogiochi che rilevano il rischio nei bambini di problemi cognitivi ed emozionali.[7]

A Singapore, la startup in tecnologie sanitarie RingMD adoperava i dispositivi indossabili per raccogliere i dati dei pazienti relativi ai parametri cardiaci ed alla temperatura corporea. I ricercatori stanno anche verificando come informazioni genetiche e mediche possono essere utili per ricerche sulla medicina di precisione, ovvero sviluppare farmaci specifici per un paziente. Inoltre, viene utilizzata l'AI, con il programma Smart Health-Assist, per sviluppare sensori domestici per monitorare i pazienti, ed i ricercatori stanno esplorando la possibilità di associare i dati forniti dai sensori con informazioni di tipo medico, economico, geografico o di identificazione personale.[7]

Il Giappone ha legalizzato l'uso dei Big Data nel 2017 e sta accelerando lo sviluppo dei Personal Health Record (PHR). L'accesso ai dati sanitari è ritenuto necessario per lo sviluppo di sistemi sanitari basati sull'AI e per aiutare l'epidemiologia, la clinica e le ricerche.[10]

Il sistema sanitario Israeliano negli ultimi anni è diventato uno dei migliori al mondo grazie allo sviluppo di tecnologie digitali applicate alla salute. Gli investimenti in startup nella Digital Health nel 2019 hanno superato i 500 milioni di dollari con un aumento del 32% in un anno. Ciò ha consentito di salvare il sistema nazionale sanitario messo

in crisi dalla mancanza di medici e dall'invecchiamento della popolazione. Dal 2019 il governo finanzia le aziende del settore (attualmente ci sono in Israele più di 1.000 startup) per realizzare progetti pilota negli ospedali. Tra i progetti sono da segnalare quelli finalizzati all'analisi immediata delle immagini per individuare i casi gravi prima che i pazienti lascino gli ospedali grazie all'impiego dell'AI ed alla realizzazione di piattaforme tecnologiche di telemedicina per connettere gli ospedali e per favorire l'uso secondario dei dati dei pazienti. I dati anonimi saranno raccolti e resi fruibili ad università e ricercatori per consentire loro di sviluppare nuovi trattamenti per centinaia di patologie, nonché di ottenere informazioni più accurate sulle loro cause.[8]

In questo scenario mondiale i servizi healthcare non potevano che attrarre anche i giganti digitali, forti dell'enorme mole di dati che posseggono. Essi stanno favorendo nuovi tipi di partnership Public-Private-Partnership (PPP) che spesso consentono di esportare servizi sanitari anche in altri paesi. In India le linee guida del 2018 per le PPP hanno previsto che le organizzazioni private possano diventare partner degli ospedali pubblici per fornire servizi in campo oncologico, cardiologico e altri. HealthMap Diagnostics Private Limited (HDPL) sta affrontando la carenza indiana di servizi di diagnostica per immagini a prezzi accessibili collaborando con ospedali statali che servono oltre 500 milioni di indiani. In molti paesi insulari, i PPP stanno contribuendo ad alleviare le pressioni finanziarie e logistiche sui sistemi sanitari pubblici. Gli esempi di progresso nella trasformazione digitale dell'assistenza sanitaria stanno proliferando e dovrebbero continuare a farlo anche nel 2020 e oltre.[9]

La stessa Huawei ha contribuito allo sviluppo della Smart Health, ad esempio attraverso l' Huawei Cloud EIHealth che è un'innovativa piattaforma gratuita che offre servizi come il rilevamento del genoma virale, l'identificazione di farmaci antivirali utilizzando simulazioni computerizzate e l'analisi della Tac assistita da intelligenza artificiale e l'Anti-COVID-19 Partner Program incentrato su telelavoro. Un esempio evidente di quanto la tecnologia sia sempre più necessaria nell'ambito sanitario è il caso delle équipe mediche che hanno sfruttato le videoconferenze e i cloud installati da Huawei e Retelit per scambiare i dati ed i risultati di esami diagnostici di pazienti a migliaia di chilometri, è il caso dell'ospedale Cotugno di Napoli e dell'ospedale Zhongshan della Fudan University di Shanghai durante la pandemia da COVID-19.[19]

In ambito sociale, l'introduzione delle tecnologie ICT utilizzate a scopo sanitario e benessere, ha portato allo sviluppo delle Smart Health Communities (SHC). Le SHC sono aggregati di persone, su base geografica, culturale o generazionale, focalizzate sulla prevenzione delle malattie e sul benessere che operano in gran parte al di fuori del sistema sanitario tradizionale. Molte SHC sono gestite internamente e rispondono ad un bisogno non soddisfatto dal sistema sanitario. Tali comunità già esistevano ed erano composte dalle persone incluse in progetti con attività finalizzate al benessere ed alla prevenzione. L'utilizzo delle tecnologie ICT ha consentito la loro espansione ed il raggiungimento di risultati migliori, consentendo di inglobare anche le fasce più svantaggiate. Le piattaforme sviluppate per gli SHC possono utilizzare tecnologie come l'IoT e la realtà aumentata/virtuale per consentire di prendere decisioni tempestive ed informate. Le informazioni condivise possono incorporare principi delle scienze comportamentali, anche attraverso giochi, per aiutare le persone a rimanere in linea con i propri obiettivi sanitari. Un esempio è il Villaggio della Salute (o Terveyskyla in Finlandia), una piattaforma online e mobile dedicata a portare le cure fuori dagli ospedali, nelle abitazioni. Consiste in 30 diverse "case" ognuna delle quali funge da hub digitale specifico per sintomi, malattie o parti del corpo. La piattaforma permette agli utenti di accedere e di creare un'identità ed ha una componente per i medici. I contenuti sono sviluppati dagli ospedali ed università ed hanno generato oltre 1,5 milioni di visite al sito web in poco più di un anno. Analogamente ParkinsonNet, una raccolta di 70 reti regionali nei Paesi Bassi, si concentra sulla fornitura di cure di alta qualità ai pazienti affetti dal morbo di Parkinson. Con una comunità di pazienti, medici, neurologi, accademici ed oltre 3.000 professionisti sanitari, la piattaforma ha contribuito a ridurre il costo del trattamento per paziente di quasi il 40% (circa 250 euro).[9]

La SH consente di informare, personalizzare, accelerare ed aumentare la capacità delle persone di prendersi cura di se stesse. Oltre ad abilitare le visite video o gli appuntamenti in teleconferenza, essa può consentire l'erogazione di cure mirate alle specifiche esigenze del paziente in funzione delle capacità dell'organizzazione sanitaria e della disponibilità delle risorse. I suoi obiettivi primari sono di espandere l'accesso di pazienti e medici ai servizi sanitari, migliorare i risultati clinici, aumentare l'impegno dei pazienti, migliorare il coordinamento delle cure, ridurre i costi e migliorare l'efficienza durante tutto il ciclo

delle cure.

Dal punto di vista tecnologico, i sistemi di Smart Health possono essere distinti in sistemi eHealth e sistemi mHealth.

1.2.1 eHealth

Il termine eHealth si riferisce all'innovazione dei servizi sanitari utilizzando Internet e le relative tecnologie digitali di backoffice. Include uno spettro di tecnologie digitali che hanno lo scopo di migliorare, mantenere o modificare le condizioni di salute dei pazienti attraverso interventi diagnostici, preventivi e terapeutici.[11]

Le tecnologie ICT sicuramente impiegate in eHealth vengono di seguito sommariamente riportate:

- Le soluzioni di **Cloud Computing** che creano maggiore flessibilità e mobilità rispetto alle piattaforme informatiche custom, aiutano i professionisti della sanità a snellire le attività che in passato richiedevano molto tempo, danno accesso ad applicazioni con funzionalità più ricche e forniscono ai medici che lavorano da soli ed in team nuove modalità di comunicazione istantanea.
- **Tecnologia di quinta generazione (5G)**, progettata per supportare comunicazioni ultra affidabili di dati, a bassa latenza e di notevole mole.
- **Interoperabilità e Piattaforme Aperte** che rendono le informazioni sanitarie più accessibili incoraggiando la condivisione dei dati tra le strutture sanitarie per migliorare la qualità e l'efficienza dell'assistenza sanitaria, e con i pazienti per aiutarli a prendere decisioni informate.
- **Intelligenza artificiale (AI)** che integra la memoria e l'esperienza per migliorare l'accuratezza diagnostica e l'efficacia del trattamento.
- **Elaborazione del linguaggio naturale (PNL)** che genera note mediche dettagliate e accurate attraverso frasi parlate o scritte.

- **Analisi dei Big Data**, in grado di esaminare grandi set di dati non strutturati per scoprire modelli o correlazioni e di visualizzare i dati nel modo più efficiente ed efficace per garantire risultati che guidino la strategia e il processo decisionale.
- **Data as a Platform (DaaP)** che estrae intuizioni sanitarie dai dati dei pazienti.
- **Robotica** che aiuta i medici ad eseguire azioni complesse con maggiore precisione e flessibilità offrendo ai chirurghi un facile accesso a immagini e dati digitali in modo da non dover distogliere lo sguardo dal paziente per raccogliere informazioni.
- **Applicazioni Aziendali** che sono progettate per integrare sistemi informatici che eseguono tutte le fasi del servizio sanitario in modo da facilitare la cooperazione e il coordinamento del lavoro nella struttura.
- **Sistemi Blockchain** che utilizzano una registrazione digitale delle transazioni per aiutare le organizzazioni del settore delle scienze della vita e della sanità a monitorare e tracciare i prodotti medici nella catena di fornitura, a condividere i dati generati dagli studi clinici e a mettere insieme le transazioni di più sistemi sanitari, farmacie e piani sanitari.
- **Piattaforme di Apprendimento** che incoraggiano l'apprendimento flessibile ed interattivo tra gli studenti di medicina e che permettono agli educatori di guidare una formazione mirata. Inoltre esse fungono da strumenti di collaborazione nei gruppi di lavoro per una condivisione efficiente delle informazioni migliorando il lavoro di squadra ed il controllo dei dati nella ricerca e nelle capacità accademiche.

Considerata la natura dei dati sanitari, classificati come dati sensibili, le infrastrutture per la trasmissione ed elaborazione devono assicurare un elevato livello di tutela della Privacy. Pertanto anche l'aspetto della Cybersecurity riveste un ruolo di estrema importanza nella Smart Health.

1.2.2 mHealth

La Mobile Health (mHealth) è il segmento della SH che consiste nell'impiego di dispositivi di comunicazione mobile per i servizi di tipo sanitario, di benessere e per scopi

informativi. Contribuisce all'autonomia dei pazienti, i quali possono gestire la loro salute in modo più attivo e di essere più indipendenti grazie a soluzioni di autovalutazione e di monitoraggio a distanza. L'mHealth può anche supportare gli operatori sanitari nel trattamento dei pazienti in modo efficace poiché le applicazioni mobili possono incoraggiare l'adesione di questi ultimi ad uno stile di vita sano.[12]

Si contano infatti più di 100.000 applicazioni negli store di Google ed Apple che promuovono la salute: alcune fungono come strumento di gestione e monitoraggio, altre informano l'utente riguardo malattie e rischi di vario genere.[3]

Per sanità mobile (mHealth) si intende l'utilizzo di dispositivi mobili per raccogliere dati sanitari in tempo reale dai pazienti, eventualmente memorizzandoli su server centralizzati collegati ad Internet. Gli ultimi sviluppi delle tecnologie micro e nano, nonché dell'elaborazione delle informazioni e della comunicazione wireless offrono la possibilità di misurazioni biomediche non invasive, nonché di sensori, di elaborazione e di comunicazione indossabili basati su tecnologia IoT. L'Internet degli Oggetti (Internet of Things - IoT) è un ambiente popolato da una varietà di componenti come tag RFID, dispositivi medici, telefoni cellulari, ecc. Questi dispositivi si connettono attraverso identificatori univoci (IPaddress) e trasferiscono tra di loro i dati. La tecnologia di rilevamento medico pervasivo e la tecnologia mobile implementata nei sistemi IoT portano alla raccolta di grandi quantità di dati, che vengono inviati ai server per la loro persistenza.

Essi forniscono informazioni in tempo reale sulla salute dei pazienti consentendo diagnosi, trattamenti o ospedalizzazione in caso di emergenza medica. Di conseguenza le infrastrutture IoT, composte dai dispositivi per il monitoraggio della salute e per le comunicazioni wireless, offrono enormi possibilità di migliorare la qualità dei servizi sanitari.

Le parti essenziali della mHealth e dell'IoT sono i dispositivi medici dotati di sensori e componenti di comunicazione. I dispositivi medici sono dispositivi intelligenti che possono essere utilizzati per monitorare vari parametri, come il livello di zucchero, la pressione sanguigna, la frequenza cardiaca, ecc. Gli algoritmi analizzano i dati in tempo reale per identificare determinati livelli di allarme a seconda delle condizioni dei pazienti osservati. Il volume di dati nei server dell'IoT è molto grande in quanto i dati sono raccolti da milioni di utenti o pazienti ad intervalli regolari, anche più volte al giorno.

Pertanto l'architettura IT di un sistema mHealth è multilivello e comprende livelli di data collection, data storage e data processing. Il livello di data storage deve essere in grado di memorizzare Big Data generati da differenti sorgenti. Il livello di data processing è responsabile dell'analisi intelligente dei dati, della generazione e visualizzazione dei report. I dati sanitari dei pazienti devono essere disponibili alle strutture sanitarie sempre ed ovunque.

Per i dispositivi medici, la dimensione, la mobilità e il comfort sono molto importanti. Ci sono molti dispositivi mhealth sul mercato con dimensioni e funzionalità diverse. Ogni dispositivo ha un identificativo univoco che gli consente di trasmettere e ricevere i dati tramite la rete di trasmissione di tipo wireless.

E' necessario utilizzare tecnologie standard per garantire l'interoperabilità dei dispositivi medici in una rete. Ci sono diversi standard per le connessioni wireless ma per questa tipologia di dispositivi sono fattori decisivi la bassa potenza, i bassi costi, la dimensione e la facilità di uso.

Gli attuali standard per dispositivi medici sono:

- CEN ISO/IEEE 11073 standard per la comunicazione tra dispositivi medici/wellness e sistemi di calcolo esterni.
- IEEE 802.11 a/b/g/n standards wireless local area network (WLAN) nella banda di frequenze 2.4 GHz e 5 GHz.

Di seguito vengono descritti, come esempio, alcuni dispositivi disponibili in ambito cardiologico:

- **Misuratore dei livelli di zuccheri nel sangue.** Un dispositivo mobile di auto-testing in grado di misurare il livello di zucchero nel sangue. E' importante per il piano di trattamento e per prevenire le complicanze a lungo termine del diabete. Google, in collaborazione con l'azienda farmaceutica svizzera Novartis, ha sviluppato lenti a contatto intelligenti per le persone che soffrono di diabete. Le lenti a contatto intelligenti prendono le lacrime delle persone e misurano i livelli di glucosio.
- **Misuratore pressione sanguigna.** La pressione sanguigna può variare da un minuto all'altro, quindi il monitoraggio continuo aiuta a comprendere meglio la

salute cardiovascolare del paziente. I misuratori della pressione sanguigna indossabili possono essere inseriti in una cintura e sono collegati ad un bracciale. Questi dispositivi sono automatici e registrano la pressione sanguigna per 24/7, mentre i pazienti possono svolgere le loro normali attività quotidiane.

- **Misuratore asma.** Il flussometro di picco è un semplice dispositivo portatile per il monitoraggio dell'asma. Il sistema che trasmette i segnali tramite un sensore wireless si attacca alla pelle, è costituito da un braccialetto e da un cerotto toracico misuratore degli impatti dell'ozono sulle condizioni di asma cronica.

I sistemi mHealth, basati sull'infrastruttura IoT, stanno rivoluzionando il modo in cui viene condotta la ricerca clinica e vengono erogate le terapie mediche. I dati in tempo reale vengono registrati e trasferiti tramite device indossabili e sofisticati dispositivi medici impiantabili. L'accesso alle informazioni dei pazienti da parte dei medici consente la diagnosi rapida, il monitoraggio remoto e la riabilitazione domiciliare. Inoltre, i sistemi mHealth riducono significativamente i costi complessivi dell'assistenza sanitaria e i ricoveri ospedalieri non necessari.[13]

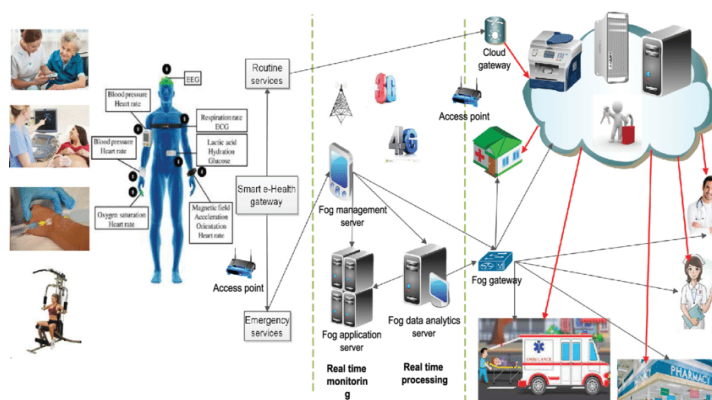


Figura 1.2: smart health

1.3 Applicazioni in ambito biomedicale

Di seguito vengono riportati alcuni esempi di applicazioni sperimentali che mettono in evidenza le modalità d'impiego delle suddette tecnologie per lo sviluppo di servizi Smart Health. In particolare verranno illustrati:

- un sistema sperimentale di tipologia strettamente mHealth finalizzato alla raccolta dei dati fisiologici in tempo reale su smartphone;
- un sistema eHealth, finalizzato all'elaborazione dei dati ricevuti da sistemi mHealth, con l'impiego di nuove tecnologie di analisi dei dati per il triage automatizzato e l'invio del paziente presso la struttura più idonea;
- i risultati conseguiti dall'applicazione di un sistema eHealth, per ottenere predizioni circa eventuali aggravamenti dei pazienti cardiopatici utilizzando algoritmi di Deep Learning.

1.3.1 Applicazione mHealth

Il progetto sperimentale riportato di seguito dal titolo "Design and Implementation of Hypothermia Symptoms Early Detection With Smart Jacket Based on Wireless Body Area Network", pubblicato in IEEE Access (Volume: 8), ha l'obiettivo di implementare un sistema per la raccolta dei dati fisiologici legati allo stato di ipotermia di una persona attraverso device indossabili e trasmissione dei dati allo smartphone.

Si è scelto di riportare questo progetto perché mostra un esempio di applicazione mobile completa viste le diverse tecnologie integrate nel sistema ed il suo recente sviluppo (2020).

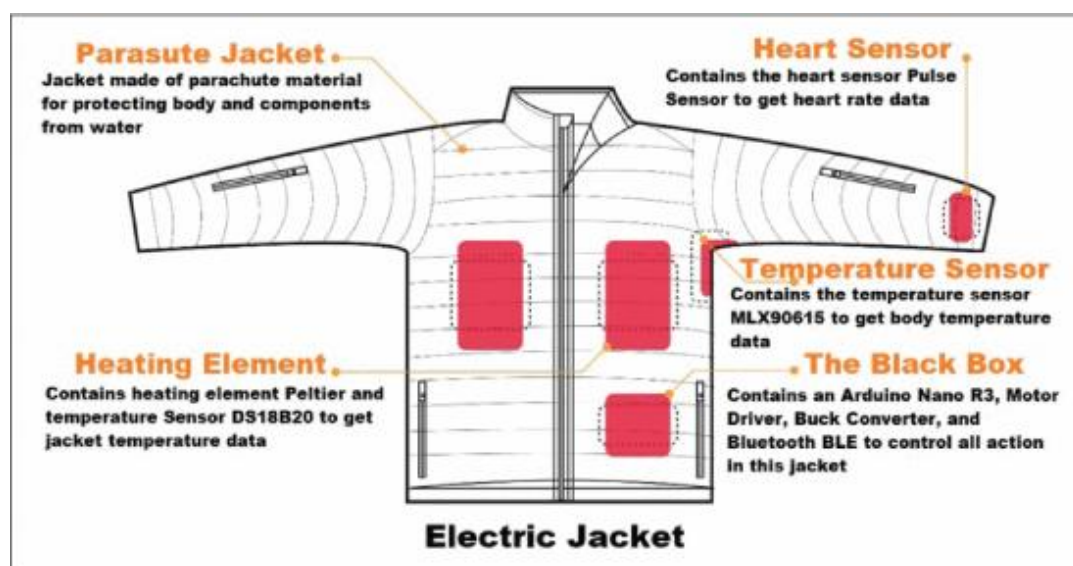


Figura 1.3: Electric Jacket

Il dispositivo indossabile capace di rilevare l'ipotermia è costituito da un sensore di temperatura, un sensore di impulsi ed elementi riscaldanti. I dispositivi dei diversi scalatori del gruppo sono collegati con una rete di area corporea senza fili (wireless body area network-WBAN). Questo dispositivo si connette allo smartphone dell'utente utilizzando il Bluetooth in modo che l'utente possa vedere le condizioni del suo corpo e regolare la temperatura dell'elemento riscaldante.

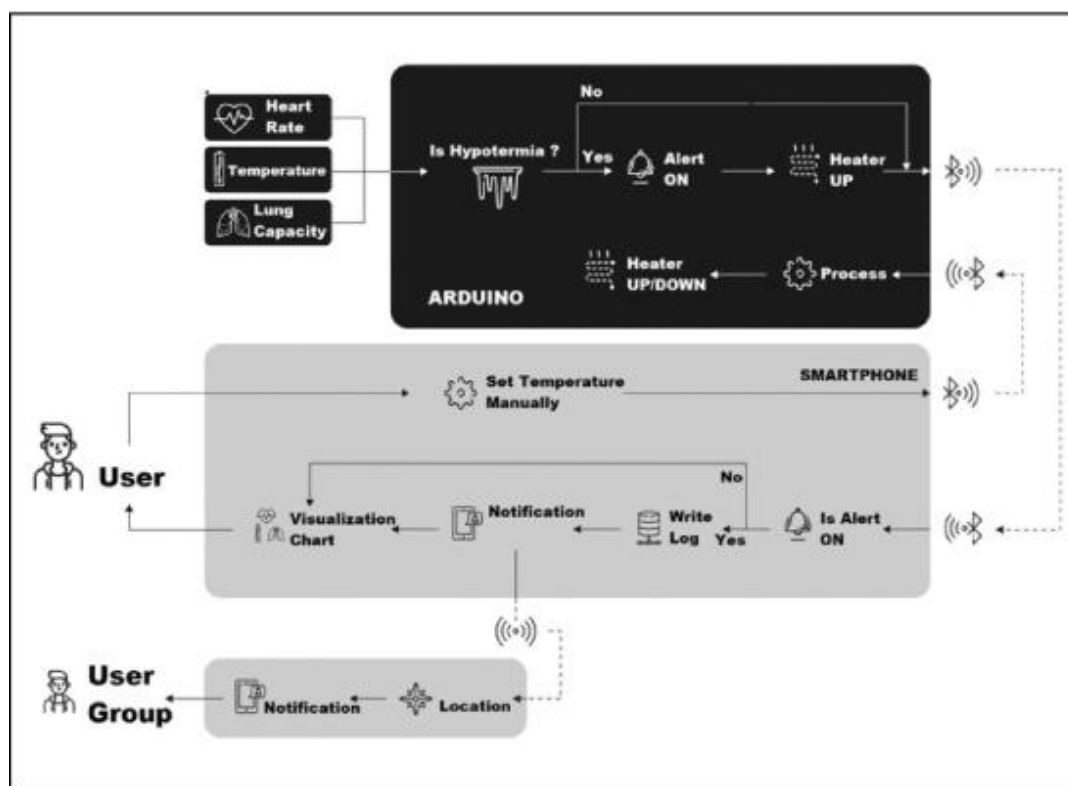


Figura 1.4: Schema logico

Il risultato dell'esperimento mostra che il dispositivo è in grado di rilevare i sintomi dell'ipotermia e di informare di questi sintomi gli utilizzatori sportivi ed i gruppi di utilizzatori dell'area WBAN, consentendo agli utenti di essere più vigili nella risposta all'ipotermia in modo da ridurre al minimo i casi di mortalità.

Viene proposto un sistema di dispositivi di monitoraggio della salute incorporato in una giacca per arrampicata. Il sistema ha utilizzato Arduino Nano come dispositivo centrale di raccolta dati dei sensori, mentre i sensori utilizzati sono di temperatura e di impulsi. Questo dispositivo è accompagnato da un elemento riscaldante, TEC12706 peltier. Il peltier si collega al driver del motore che serve a regolare la temperatura. Il Bluetooth utilizzato in questa ricerca è stato supportato con tecnologia a bassa energia.

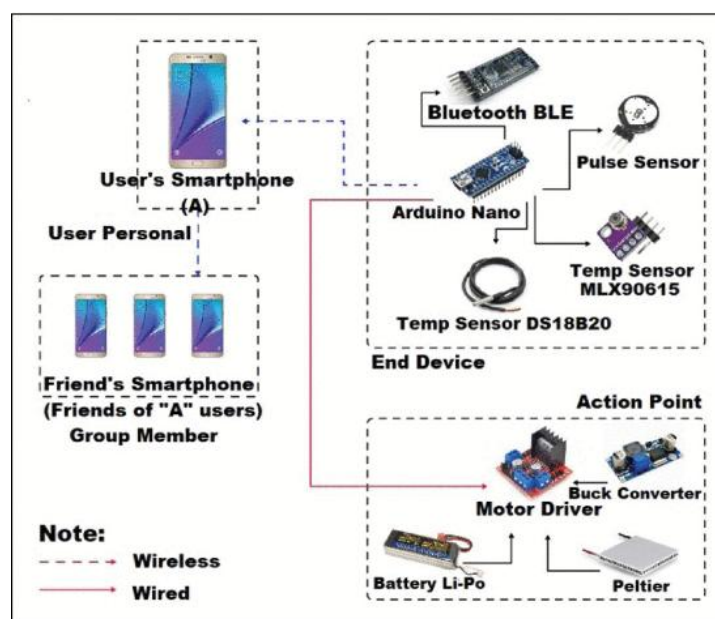


Figura 1.5: Architettura hardware

Attraverso il loro smartphone Android, gli scalatori possono vedere la temperatura e le condizioni del battito cardiaco insieme al loro stato corporeo. Con l'App, gli scalatori riceveranno informazioni su come comportarsi in base alle condizioni di salute del momento. Inoltre, si possono creare gruppi di arrampicatori utilizzando la tecnologia Wi-Fi Direct che permette ad Android di connettersi ad altri Android. Quando gli scalatori sono collegati al gruppo di arrampicata, possono vedere la posizione e le condizioni del corpo di un altro scalatore. Quando la condizione del corpo di uno di essi peggiora, gli altri membri riceveranno una notifica. I dati del peggioramento della condizione corporea saranno memorizzati su un database SQLite Android. Sono stati utilizzati diversi sensori ed altri componenti, in particolare il sensore di polso che è stato utilizzato per ottenere i dati della frequenza cardiaca degli arrampicatori, il sensore di temperatura MLX90615 che è stato utilizzato per ottenere la temperatura del corpo dell'arrampicatore, il sensore di temperatura DS18B20 che è stato utilizzato per la temperatura del riscaldatore e il peltier che è stato utilizzato per il riscaldamento della giacca. I sensori ed i componenti sono stati collegati con un microcontrollore Arduino ed il circuito è stato integrato in

una giacca. La trasmissione dei dati è stata effettuata in modalità wireless tramite una rete di comunicazione Bluetooth. Il processore Arduino riceve i dati dai sensori e li elabora per definire il livello di ipotermia del corpo. Successivamente, i dati numerici dei parametri fisiologici ed il risultato finale della loro valutazione vengono inviati all'App.

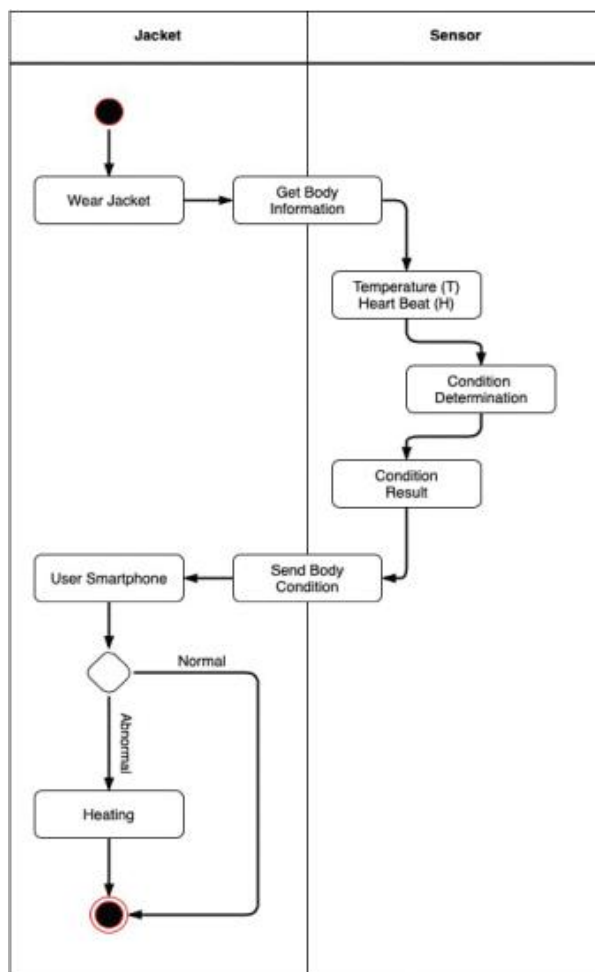


Figura 1.6: Activity Diagram

In questa ricerca l'Applicazione Android è collegata con i dispositivi hardware via Bluetooth e serve a visualizzare i risultati della raccolta dati effettuata dai sensori di temperatura e dai sensori di impulsi. Inoltre, l'applicazione Android può regolare il calore. I dati vengono memorizzati nel RDBMS SQLite embedded in Android e visualizzati

da tutti i membri del gruppo di scalatori, sotto forma di valori e grafici.



(a)



(b)

Figura 1.7: View dati

Inoltre in caso di allarme per ipotermia viene visualizzata la schermata di allarme.

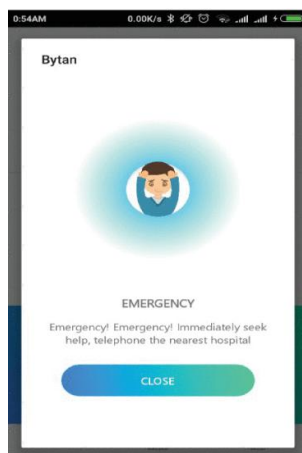


Figura 1.8: View allarme

E' possibile visualizzare anche i dati storici delle misurazioni memorizzati in SQLite e lo stato dei singoli partecipanti al gruppo. Il device indossabile per l'arrampicata può essere collegato, tramite Smartphone, ad un punto di accesso o gateway associato ad altri punti di accesso ed a sedi di monitoraggio. In questo modo, il personale che si trova nella stazione di monitoraggio otterrà in real-time le informazioni sugli scalatori individuando quelli che hanno l'ipotermia e la loro posizione.[14]

1.3.2 Applicazioni eHealth

Già in un articolo del 2007 si parlava di IT associata al termine sanità.[22]

Le prime applicazioni in ambito sanitario si limitavano semplicemente alla gestione dei dati sanitari dei pazienti ai fini della semplice fornitura dei servizi. L'introduzione del WWW ha consentito lo scambio dati tra il paziente e le strutture sanitarie portando ad una prima evoluzione i sistemi IT della sanità. Negli ultimi anni è sorto un nuovo trend in forte crescita, definito Sanità Personalizzata (pHealth), che riguarda l'individualizzazione delle procedure di prevenzione, cura e benessere disponibili attraverso il sistema sanitario. pHealth si realizza attraverso una serie di capacità, derivanti dalle micro e

nanoscienze, dalle tecnologie dell'informazione e della comunicazione (ICT). L'idea consiste nella gestione dei dati sanitari in ambito familiare grazie all'utilizzo di una cartella clinica familiare elettronica.

Attualmente i sistemi ICT sanitari stanno attraversando una nuova fase evolutiva che si concretizza con l'impiego di algoritmi che consentono il supporto alle decisioni sia in ambito gestionale che clinico.

Di seguito si riportano alcuni esempi di sperimentazioni recenti nei due ambiti.

Gestionale

L'elaborazione dei dati nei sistemi di telemedicina avviene attraverso tre livelli principali, i sensori indossabili (Tier 1), il gateway (Tier 2) e il server del centro medico (Tier 3). Il Tier 1 prevede la raccolta dei segnali vitali dei pazienti mediante l'utilizzo di dispositivi medici indossabili interoperabili. Alcuni dispositivi includono l'elettrocardiogramma (ECG), i sensori del livello di saturazione dell'ossigeno nel sangue (SpO2) e i monitor della pressione sanguigna (BP). Questi dispositivi trasferiscono i dati raccolti al Tier 2. Dopo la raccolta dei dati di tutti i dispositivi di monitoraggio, i dispositivi del Tier 2 li aggregano e li trasferiscono ad un server remoto tramite gateway esterno per garantire comunicazioni a lungo raggio. Il Tier 3, ovvero il server del centro medico è un elaboratore remoto che si trova in un istituto medico per monitorare i dati in tempo reale e fornire raccomandazioni per la salute dei pazienti. Il monitoraggio è effettuato dal personale sanitario direttamente o tramite algoritmi per il post-processing. Questo studio fornisce un metodo per migliorare il processo di triage e di fornitura di servizi sanitari.

Il sistema garantisce la fornitura di servizi sanitari continui per i pazienti bilanciando e controllando tali servizi tra gli ospedali in caso di disastri naturali e di invecchiamento della popolazione. L'attuale ricerca mira a (i) proporre un nuovo design dell'architettura della telemedicina per la fornitura di servizi sanitari; (ii) proporre un nuovo triage a distanza a quattro livelli e la localizzazione del pacchetto automaticamente proposto per i pazienti con malattie cardiache croniche; (iii) identificare una matrice decisionale (DM) per "servizi di assistenza sanitaria multipli" ed "ospedalieri" basata sul triage e sulla localizzazione del pacchetto automaticamente proposto; (iv) sviluppare un quadro

di raccomandazioni sanitarie intelligenti in tempo reale, basato su un sensore di dati sanitari indossabile; (v) convalidare il quadro sviluppato utilizzando metodi statistici e (vi) valutare il quadro sviluppato usando scenari e liste di controllo per il benchmarking.

L'implementazione del sistema è composta da due fasi:

1. Fase di Identificazione
2. Fase di Sviluppo

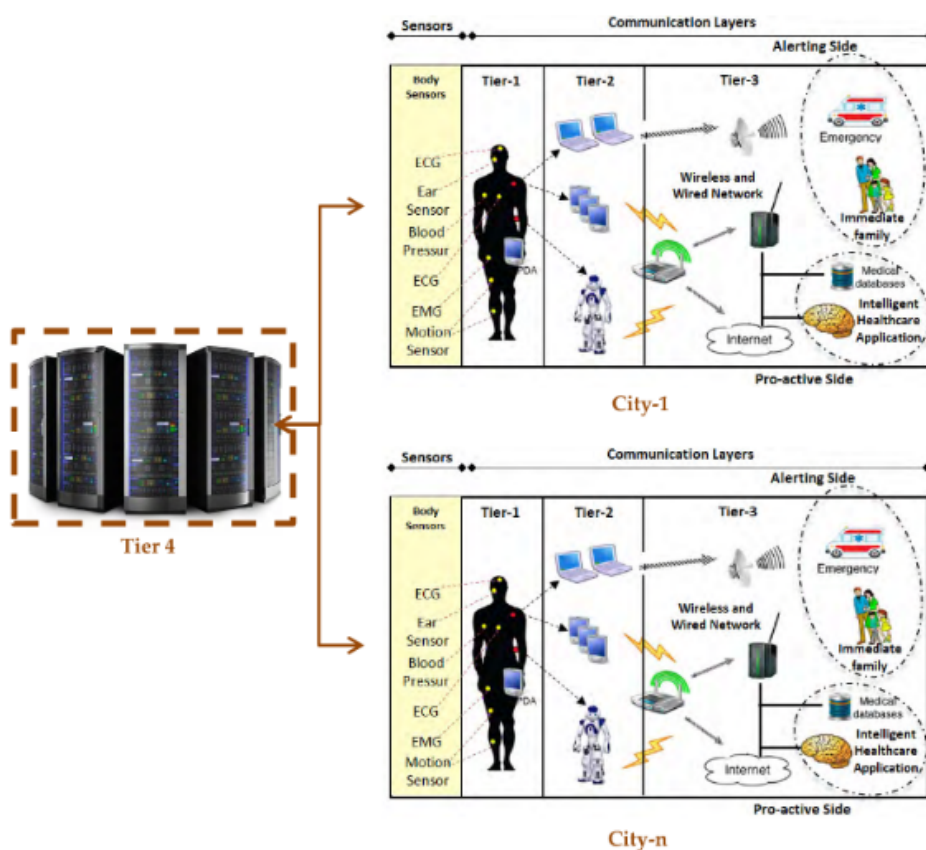


Figura 1.9: Architettura sistema gestionale

La Fase di Identificazione si sviluppa in sei passi:

- **Nuova Architettura per la Telemedicina:** centro di gestione intelligente dei dati e dei servizi (Tier 4). Il Tier 4 possiede la capacità di identificare un ospedale adatto a trattare il paziente ed è la parte in cui vengono elaborati tutti i processi e le decisioni.
- **Identificazione dei pazienti cardiopatici:** i pazienti da remoto inviano i segnali vitali raccolti dai sensori al Tier 4 per la valutazione e il monitoraggio.
- **Livelli di Triage remoto e localizzazione dei servizi sanitari:** l'algoritmo 4LRTPL classifica le condizioni dei pazienti e identificare i servizi sanitari adeguati. I tre tipi di decisioni risultanti dal 4LRTPL sono il "livello di triage", il "valore del codice di triage (TC)" ed i "pacchetti di servizi sanitari".
- **Identificazione Ospedali:** sono stati individuati 12 ospedali situati nella città di Baghdad che rappresentano le alternative nella Matrice Decisionale (MD). Ogni ospedale comprende tre pacchetti di servizi sanitari.
- **Identificazione dei pacchetti di servizi sanitari:** i sei servizi del pacchetto 1 sono: "preparare la sala operatoria" (PSR), "preparare l'équipe chirurgica" (PST), "preparare il medico di chirurgia" (PSD), "preparare il fornitore di O2" (POS), "inviare l'ambulanza" (SA) e "fornire le medicalizzazioni" (PM). I sei servizi (parametri) del pacchetto 2 sono "preparare il pronto soccorso" (PER), "preparare la consulenza (PCS)", "preparare il medico" (PD), POS, SA e (PM); i quattro servizi (parametri) del pacchetto 3 sono PCS, PD, POS e PM. Il processo di selezione degli ospedali si basa sul numero di servizi disponibili all'interno degli ospedali.
- **Matrici Decisionali:** le DM sono generate dal Tier 4 per i pacchetti 1, 2 e 3 in base al livello di triage dei pazienti. In queste DM, le alternative sono rappresentate dagli ospedali, mentre i criteri sono rappresentati da servizi sanitari utilizzati. Ciò consente la selezione automatica dell'ospedale per il paziente.

La fase di sviluppo comprende le tecniche del processo decisionale basato su criteri multipli (MCDM) per la classificazione degli ospedali nel Tier 4. I metodi MCDM

integrati richiedono l'AHP (processo analitico gerarchico) per il calcolo degli attributi. In questo caso gli attributi sono stati fissati come pesi per i servizi sanitari in modo da identificare ciascuno di essi e come contributo per prendere una decisione. In seguito il metodo VIKOR è stato utilizzato per classificare gli ospedali sulla base di informazioni quantitative. Infine, il numero di servizi è stato individuato come fattore chiave per classificare gli ospedali in ordine decrescente. I punteggi degli ospedali vengono elencati in ordine crescente e l'ospedale più adatto sarà selezionato secondo la tecnica VIKOR.[15]

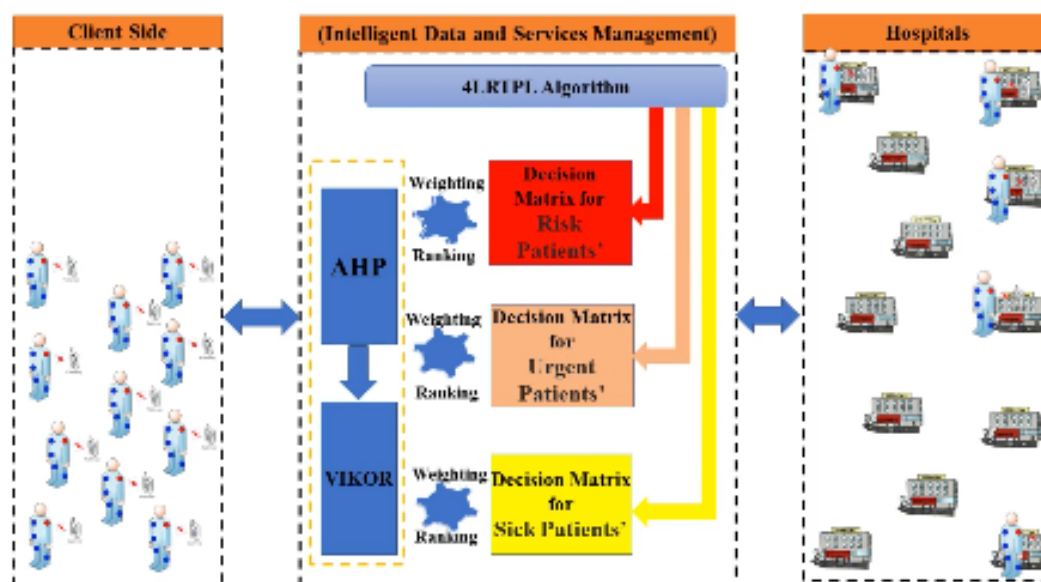


Figura 1.10: Logica del sistema gestionale

Diagnosi precoce delle malattie cardiache tramite Machine Learning

Partendo dalla consapevolezza che le malattie cardiache sono la causa principale di morte per tutte le fasce d'età, il progetto "Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform" illustrato di seguito è finalizzato a migliorare la capacità di prevedere gli attacchi di cuore utilizzando diverse tecniche di Deep Learning. La diagnosi precisa e accurata delle malattie cardiache dipende principalmente dalla conoscenza di precedenti

eventi patologici. I pazienti cardiopatici devono monitorare i parametri corporei come la pressione sanguigna, il fumo di sigaretta, il colesterolo, il diabete ed il sesso. Queste variabili sono indipendenti e sono una buona base per l'intelligenza artificiale (AI) e per i sistemi di apprendimento automatico.

La ricerca ha compiuto progressi significativi nel prevedere efficacemente le malattie cardiache, soprattutto tramite il Deep Learning e le tecniche di classificazione. I metodi attuali per diagnosticare la gravità delle malattie cardiache nei pazienti includono test da sforzo, radiografia del torace, angiogramma coronarico, risonanza magnetica cardiaca (RM), elettrocardiogramma (ECG), ecc. La scienza medica e le tecniche di Data Mining sono utilizzate per diagnosticare diversi tipi di segnali di sindromi metaboliche durante le attività fisiche come l'esercizio fisico, il riposo e il lavoro. I modelli di previsione sviluppati e la classificazione Deep Learning possono consentire diagnosi altamente precise e affidabili delle malattie cardiache e ridurre il numero di diagnosi errate che possono essere dannose per i pazienti. Le prestazioni sono state ulteriormente migliorate dalle tecniche *feature selection* che hanno contribuito ad aumentare l'accuratezza degli algoritmi di Deep Learning. Il progetto propone l'utilizzo di una Enhanced Deep Convolutional Neural Network (EDCNN) per la diagnosi precoce delle malattie cardiache. Il modello EDCNN è focalizzato su un'architettura più profonda che segue il modello del perceptrone multistrato con un approccio di regolarizzazione dell'apprendimento.

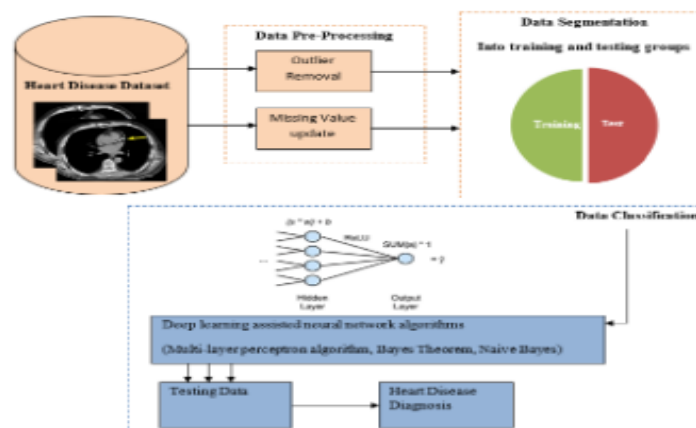


Figura 1.11: ML logic

L'uso delle tecniche di analisi dei dati come il Deep Learning riduce la necessità di competenza e la probabilità di errore umano aumentando, così, l'accuratezza delle previsioni. L'EDCNN mostra risultati promettenti nella progettazione e nella messa a punto di architetture per l'individuazione delle malattie cardiache basata su dati clinici di routine. I risultati dei test mostrano che una progettazione flessibile e la successiva messa a punto dell'EDCNN consentono di raggiungere una precisione del 99%.

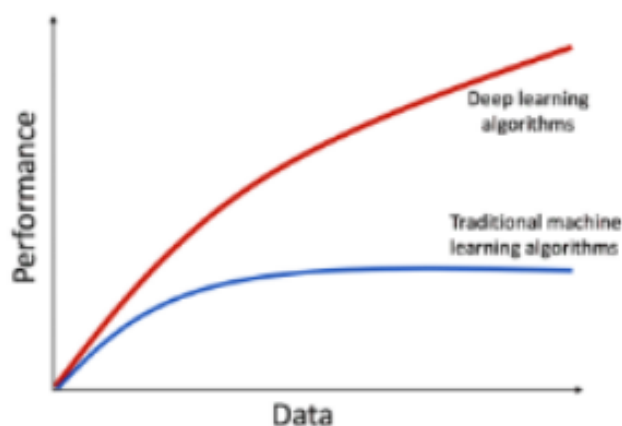


Figura 1.12: Performance DL-ML

Diversi metodi di data mining possono essere impiegati per prevedere le condizioni cardiache utilizzando una Artificial Neural Network (ANN).

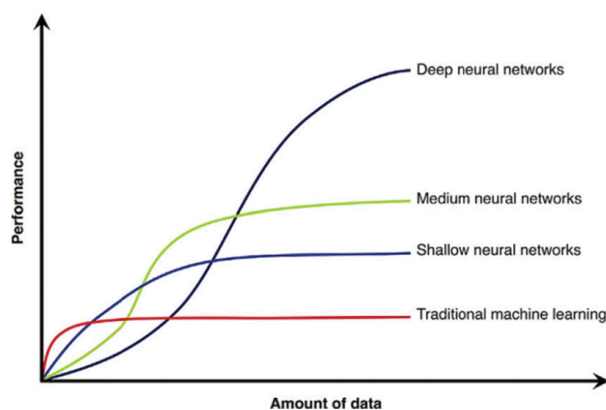


Figura 1.13: Performance NN-ML

Gli algoritmi possono quindi essere utilizzati per aiutare i pazienti e gli operatori sanitari di tutto il mondo a sostenere sia la salute globale che quella pubblica, in particolare nei paesi in via di sviluppo e nelle aree con risorse limitate e con un minor numero di specialisti cardiaci disponibili.[16]

Capitolo 2

Progettazione

Questa tesi vuole presentare il Web Information System "Heart" finalizzato alla gestione dei pazienti cardiopatici dell'Ospedale S.Orsola-Malpighi di Bologna.

Un Web Information System consiste in un'applicazione con componenti orientate a specifiche funzionalità e componenti di informazione. Il web browser è usato come front-end di un sistema dove il database è il back-end. I DBMS permettono di implementare un paradigma di separazione tra dati ed applicazione in modo che l'applicazione conosca la struttura logica dei dati e non la loro struttura fisica.

Heart è pensato per supportare l'équipe medica nell'assistere i pazienti di due tipologie: quelli con scompenso cardiaco e quelli portatori di LVAD. Il sistema è stato realizzato per dare risposta alle esigenze dello staff medico del reparto di cardiocirurgia nell'attività di gestione dei dati sanitari dei pazienti in cura presso tale reparto.

Heart è il back-end del sistema software che ha come front-end l'applicazione mobile frutto del lavoro recentemente realizzato dal dottor Lombardo. I termini inglesi front-end (FE) e back-end (BE) in informatica denotano, rispettivamente, l'interfaccia utente di un programma che consente l'interazione utente-sistema e la parte che permette l'effettivo funzionamento di queste interazioni. Il front-end, nella sua accezione più generale, è responsabile dell'acquisizione dei dati di ingresso e della loro elaborazione tale da renderli utilizzabili dal back-end. La distinzione di una parte di ingresso e di una parte terminale nei sistemi software è un genere di astrazione che aiuta a mantenere le diverse parti di un sistema complesso logicamente separate in modo da semplificare la loro manutenzione.

Si è partiti dalla raccolta e analisi dei requisiti per definire le funzionalità del sistema.

2.1 Specifiche

Nella fase di elaborazione delle specifiche di sistema si è descritto nel modo più dettagliato possibile il problema di dominio del medico/infermiere nella gestione delle rilevazioni inviate dai pazienti tramite App Mobile.

Le funzionalità individuate nascono da consultazioni con il team ospedaliero e da necessità di uso comune.

Il problema di dominio viene così descritto:

Il malato a seguito di visita specialistica e/o operazione chirurgica diventa un paziente con scompenso cardiaco oppure LVAD. L'utilizzatore è un medico o un infermiere. L'infermiere è registrato alla piattaforma e accede al sistema con username e password. Esso registra sulla piattaforma il nuovo paziente inserendo Nome, Cognome, Num Telefono, Sesso, Data di nascita, Luogo di nascita, Residenza, Altezza, Peso, Tipologia (LVAD o no LVAD), Dispositivo, Patologia di base, classe NYHA, classe INTERMACS. Viene associato ad ogni paziente un medico curante. Il medico è registrato alla piattaforma con Nome, Cognome, Num Tel, PID. Successivamente il paziente scarica l'app ed inserisce periodicamente la rilevazione dei dati. Se è un paziente No-LVAD inserisce Peso, Pressione arteriosa, Frequenza cardiaca, Passi, Nota. Il paziente di tipo LVAD inserisce dei campi aggiuntivi: Rpm, Watt, Flusso Medio, allarme. I pazienti LVAD si distinguono in pazienti con dispositivo HM3 e HVAD. I pazienti HM3 inseriscono PI, i pazienti HVAD inseriscono Picco e Depressione. Non tutti i campi sono obbligatori. L'infermiere può visualizzare tutte le rilevazioni in tempo reale che non sono state già validate. Le rilevazioni risultano validate dopo essere state visualizzate nel dettaglio e commentate dal punto di vista sanitario inserendo una nota associata alla rilevazione. Le rilevazioni validate sono ancora visualizzabili dal medico in una finestra ad hoc separate dalle rilevazioni non validate in modo da dare maggiore visibilità a queste ultime. L'infermiere può inserire una nota ad ogni rilevazione che visualizza vedendo, al con-

tempo, la cartella clinica completa del paziente. La cartella clinica del paziente contiene le informazioni anagrafiche, lo storico delle rilevazioni da validare, il medico curante e la note relative alle rilevazioni già validate. Ogni volta che il paziente inserisce i dati, automaticamente il sistema verifica lo storico dei dati del paziente e genera un allarme in caso di particolari scostamenti dei dati dai valori normali. In particolare, nel caso di pazienti LVAD si genera un'allarme se ci sono variazioni di +1 Watt o -1.5 litri di flusso sanguigno medio nell'arco di due settimane. Il medico visualizza le rilevazioni solo dei suoi pazienti in ordine cronologico sia quelle validate, con eventuale nota dell'infermiere, sia quelle non validate. Per quelle non ancora validate può inserire la relativa nota e per quelle già validate può inserire una terapia come un testo dettagliato delle attività e dei farmaci da assumere.

Il sistema prevede 3 tipologie di utenti:

1. Amministratore di sistema (Admin)
2. Infermiere
3. Medico

Le funzionalità che il sistema implementa sono così riassunte:

- Ogni utilizzatore può visualizzare il proprio profilo e modificare la propria password.
- L'amministratore può inserire un utilizzatore (amministratore/medico/infermiere).
- L'amministratore può modificare il profilo di un utilizzatore.
- L'infermiere può visualizzare le rilevazioni.
- L'infermiere può modificare/inserire un paziente.
- L'infermiere può validare le rilevazioni inserendo una nota.
- L'infermiere può esportare i dati in un file csv.
- Il medico può modificare/inserire un paziente.

- Il medico può visualizzare le rilevazioni.
- Il medico può validare le rilevazioni inserendo una nota.
- Il medico può esportare i dati in un file csv.
- Il medico può visualizzare tutte le terapie di un paziente e inserirne delle nuove.
- Login con PID e Password
- Logout

Successivamente si è passati alla fase di Progettazione dei dati.

2.2 Progettazione dei dati

In tale fase si individua la struttura dei dati e l'organizzazione che dovrebbero avere. La metodologia di progettazione prevede 3 fasi: progettazione concettuale, logica e fisica.

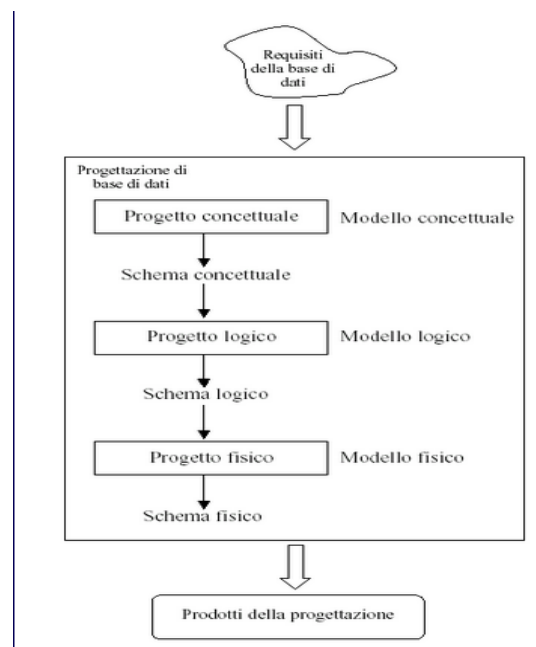


Figura 2.1: Progettazione di una base di dati

Nella fase di Progettazione concettuale si è cercato di rappresentare il contenuto informativo della base di dati e la loro organizzazione ad un alto livello di astrazione senza tener conto degli aspetti implementativi.

Nella fase di Progettazione logica lo schema concettuale viene convertito in schema logico della base di dati che ci consente di descrivere i dati secondo una rappresentazione concreta seppur indipendente dai dettagli fisici. Le scelte progettuali si basano su criteri di ottimizzazione delle operazioni da effettuare sui dati. Si fa uso di tecniche di verifica della correttezza dello schema logico come la normalizzazione.

In ultimo, si specificano i parametri fisici di memorizzazione dei dati nel modello fisico. Il modello fisico dipende dal sistema di gestione della base di dati.

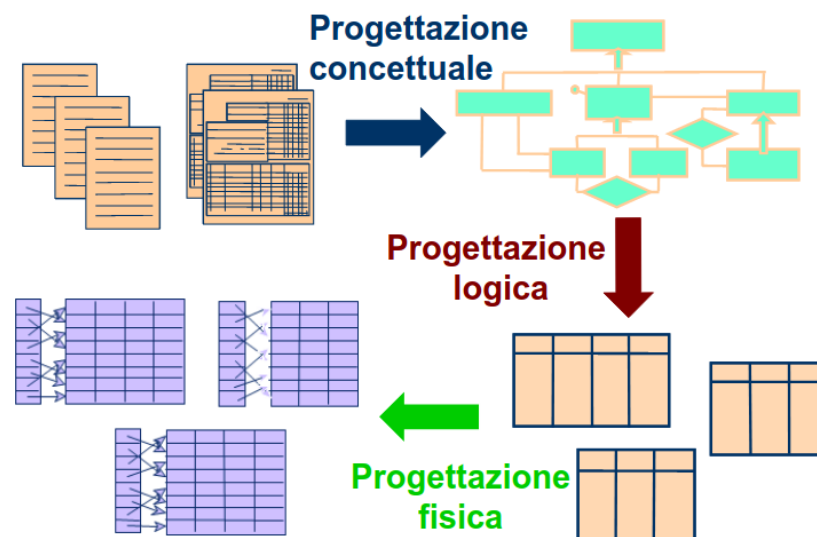


Figura 2.2: Artefatti della progettazione della base di dati

Un DBMS si distingue in base al modello logico scelto per la gestione dei dati:

- Relazionale
- Gerarchico
- Reticolare

- Ad Oggetti
- Approccio NoSQL

Il Modello relazionale è il modello logico scelto per la realizzazione del database poiché i dati sono organizzati in record di dimensione fissa e divisi in tabelle dette relazioni. Nel nostro problema di dominio, ogni paziente può inserire una rilevazione costituita da parametri fissi e strutturati. Anche le restanti entità individuate in tale dominio sono strutturate. Si procede con la realizzazione delle diverse fasi della progettazione concettuale.

- Analisi dei requisiti
- Decomposizione del testo
- Glossario dei termini
- Dizionario delle entità
- Dizionario delle relazioni
- Business rules
- Modello Entità-Relazione

L'intero processo di trasformazione delle specifiche in uno schema concettuale può essere svolto seguendo diverse strategie di progetto. Tra le tre strategie principali (strategia top-down, bottom-up e mista) è stata applicata la strategia mista.

2.2.1 Decomposizione del testo

- **Utilizzatore:**
l'utilizzatore è un medico o un infermiere. L'utilizzatore ha un nome, cognome, PID, email, password.
- **Infermiere:**
l'infermiere accede al sistema con PID e Password. L'infermiere del sistema (infermiere e/o medico) registra sulla piattaforma il nuovo paziente. L'infermiere

inserisce Nome, Cognome, Num Telefono, Sesso, Data di nascita, Luogo di nascita, Residenza, Altezza, Peso, Tipologia (LVAD o no LVAD), Dispositivo, Patologia di base, classe NYHA, classe INTERMACS. L'infermiere associa ad ogni paziente un medico curante. L'infermiere può visualizzare tutte le rilevazioni in tempo reale che non sono state già validate. L'infermiere può vedere la cartella clinica completa di ogni paziente in forma grafica.

- Medico:

viene associato ad ogni paziente un medico curante. Il medico è registrato alla piattaforma con Nome, Cognome, Num Tel, PID. Il medico visualizza in ordine cronologico le rilevazioni dei suoi pazienti sia quelle validate con nota dell'infermiere e sia quelle non validate e può inserire una nota. Il medico può inserire una terapia per ogni paziente.

- Paziente:

l'infermiere inserisce Nome, Cognome, Num Telefono, Sesso, Data di nascita, Luogo di nascita, Residenza, Altezza, Tipologia (LVAD o no LVAD), Dispositivo, Patologia di base, classe NYHA, classe INTERMACS. Il malato a seguito di visita specialistica e/o operazione chirurgica diventa un paziente con scompenso cardiaco oppure LVAD. Successivamente il paziente scarica l'app e dopo il primo accesso modificherà la password. Il paziente inserisce la rilevazione dei dati. Se è un paziente No LVAD inserisce Peso, Pressione arteriosa, Frequenza cardiaca, Passi, Nota. Il paziente di tipo LVAD inserisce dei campi aggiuntivi : Rpm, Watt, Flusso Medio, allarme. I pazienti LVAD si distinguono in pazienti con dispositivo HM3 e HVAD. I pazienti HM3 inseriscono il PI, i pazienti HVAD inseriscono Picco e Depressione. Non tutti i campi sono obbligatori. Il paziente visualizza la terapia dall'app e può ricevere notifiche ogni volta che la terapia viene modificata.

- Rilevazione:

il paziente inserisce la rilevazione dei dati. Se è un paziente No LVAD inserisce Peso, Pressione arteriosa, Frequenza cardiaca, Passi, Nota. Il paziente di tipo LVAD inserisce dei campi aggiuntivi: Rpm, Watt, Flusso Medio, Picco, Depressione, PI. Non tutti i campi sono obbligatori. L'infermiere può visualizzare tutte le rilevazioni

in tempo reale che non sono state già validate. Le rilevazioni risultano validate dopo essere state visualizzate nel dettaglio. L'infermiere può inserire una nota ad ogni rilevazione che visualizza.

- **Terapia:**
il medico può inserire le terapie per ogni paziente. La terapia è un testo dettagliato delle attività da svolgere e dei farmaci da assumere.

2.2.2 Tabella glossario

TABELLA GLOSSARIO			
Termine	Descrizione	Sinonimi	Collegamenti
Utilizzatore	Infermiere impegnato nel monitoraggio ed il medico curante.		Infermiere Medico
Infermiere	Infermiere addetto al monitoraggio delle rilevazioni		Paziente Medico Rilevazione
Medico	Medico curante che controlla lo stato di salute del paziente ed inserisce/aggiorna le terapie		Paziente Terapia Infermiere Rilevazione
Rilevazione	Dati inseriti dal paziente tramite app		Infermiere Medico Paziente (rilevazione)
Terapia	Testo inserito dal medico con attività e farmaci da assumere.		Medico Paziente
Paziente	Paziente con LVAD o scompenso cardiaco		Medico Rilevazione Terapia

2.2.3 Dizionario delle Entità

DIZIONARIO ENTITÀ'			
Entità	Descrizione	Attributi	Identificatore
Utilizzatore		email, pwd, nome, cognome, num-tel, PID, categoria	PID
Paziente		ID, Nome, Cognome, luogo-nascita, data-nascita, sesso, peso, residenza, altezza, tipologia (noLVAD/ LVAD-HM3/LVAD-HVAD), classe-INTERMACS, classe-NYHA	ID
Rilevazione		ID, Data, frequenza-arteriosa, nota, passi, pressione, peso, allarme-aut	ID
LVAD		Picco, rpm, watt, flusso-medio, PI, depressione, allarme	ID (rilevazione)
Terapia		Data, nota	Data IDpaziente

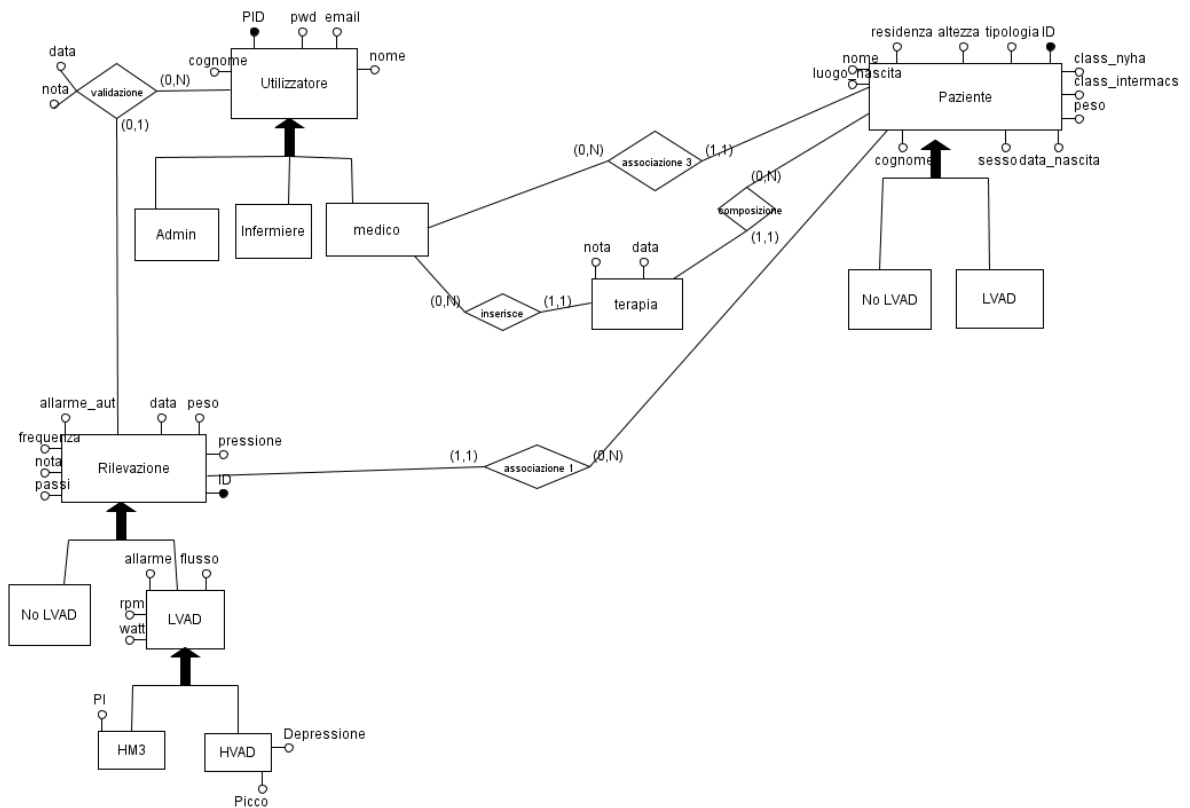
2.2.4 Dizionario delle Relazioni

DIZIONARIO RELAZIONI			
Relazione	Descrizione	Componenti	Attributi
Composizione	Un paziente è composto dalle terapie a lui prescritte	Terapia Paziente	
Validazione	Visualizzazione delle rilevazioni di un paziente da parte di un utilizzatore (medico o infermiere)	Utilizzatore Rilevazione	Data nota
Associazione 1	Ogni rivelazione è associato ad un paziente	Rilevazione Paziente	
Associazione 2	Associa ad ogni rilevazione di un paziente di tipo LVAD i campi aggiuntivi da lui inseriti	Rilevazione LVAD	
Associazione 3	Associazione tra il paziente e il suo medico curante	Utilizzatore (categoria medico) Paziente	data-inizio data-fine

2.2.5 Business Rules

-BUSINESS RULES	
REGOLE DI VINCOLO	
Il paziente può essere associato ad un solo utilizzatore di categoria "medico"	
-Nome, Cognome, Sesso, data_nascita, luogo_nascita, residenza, tipologia, dispositivo, patologia, pressione arteriosa, nota, allarme: String	
-altezza, NYHA, INTERMACS, peso, frequenza, passi, rpm, flusso medio: int	
-Watt, picco, depressione, PI: double	
Allarme_aut deve essere settata a "Watt alti" se i watt sono aumentati di +1 unità negli ultimi 15 giorni o a "Flusso basso" se il flusso sanguigno è diminuito di 1.5 litri negli ultimi 15 giorni.	
REGOLE DI DERIVAZIONE	

2.2.6 Diagramma E-R



2.2.7 Ristrutturazione dello schema E-R

Fase di ottimizzazione dello schema concettuale che si compone dei seguenti passi da effettuare in sequenza:

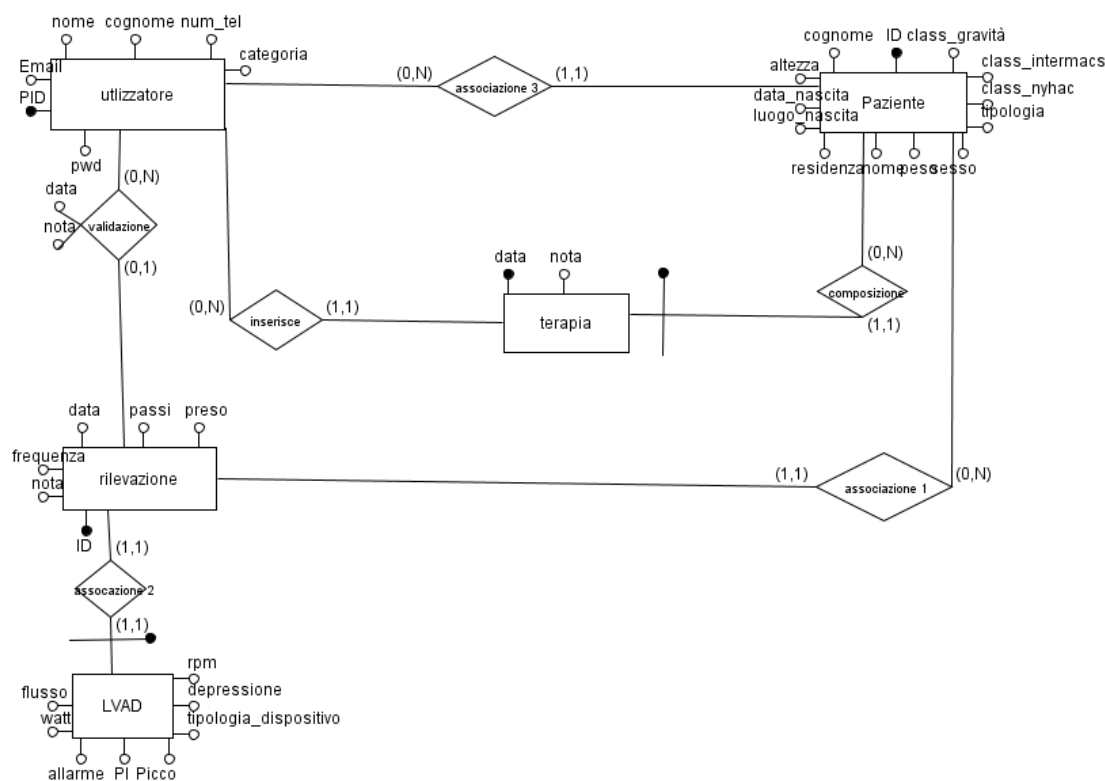
- Analisi delle ridondanze
- Eliminazione delle generalizzazioni
- Partizionamento di entità e associazioni
- Scelta degli identificatori principali

Nel nostro diagramma E-R sono state individuate 4 generalizzazioni. Ogni generalizzazione può essere risolta in 3 modi differenti in base al caso specifico.

- Accorpamento delle figlie della generalizzazione nel genitore
- Accorpamento del genitore della generalizzazione nelle figlie
- Sostituzione della generalizzazione con associazioni

L'eliminazione delle generalizzazioni e la loro conversione in costrutti del modello E-R per i quali esiste un'implementazione naturale ovvero sotto forma di entità o relazioni, è stata così risolta:

1. La generalizzazione relativa all'entità utilizzatore è stata risolta accorpando i figli nel padre aggiungendo l'attributo categoria.
2. La generalizzazione dell'entità rilevazione LVAD è stata eliminata accorpando i figli nel padre aggiungendo l'attributo tipologia_dispositivo .
3. La generalizzazione dell'entità paziente è stata risolta accorpando i figli nel padre aggiungendo l'attributo tipologia.
4. La generalizzazione dell'entità rilevazione è stata risolta trasformando la generalizzazione in una relazione.



2.2.8 Traduzione nel modello logico

Il diagramma E-R viene tradotto nel modello logico. Le tabelle realizzate saranno sei:

- **Paziente** (ID, nome, cognome, luogo_nascita, data_nascita, sesso, altezza, peso, residenza, class_gravità, class_NYHA, class_INTERMACS, tipologia, PID_medico).
- **Terapia** (data, ID_paz, nota, PID_medico).
- **Rilevazione** (data, ID_paz, peso, pressione, frequenza, nota, passi).
- **LVAD** (data_rilevazione, ID_pazR, rmp, watt, flusso, picco, depressione, PI, tipologia_dispositivo).
- **Validazione** (data_rilevazione, ID_pazR, data, nota, PID_utilizzatore).
- **Utilizzatore** (PID, email, pwd, nome, cognome, num_tel, categoria).

Vincoli di integrità

- LVAD.data_rilevazione → Rilevazione.data
- LVAD.ID_pazR → Rilevazione.ID_paz
- Rilevazione.ID_paz → Paziente.ID
- Validazione.data_rilevazione → Rilevazione.data
- Validazione.ID_pazR → Rilevazione.ID_paziente
- Validazioneee.PID_utilizzatore → Utilizzatore.PID
- Terapia.PID_medico → Utilizzatore.PID
- Terapia.ID_paz → Paziente.ID
- Paziente.PID_medico → Utilizzatore.PID

Terminata la progettazione della base dati, si passa alla progettazione dell'applicazione Web. Oggetto della tesi è la realizzazione di un Web Information System con architettura AMP (Apache MySQL PHP).

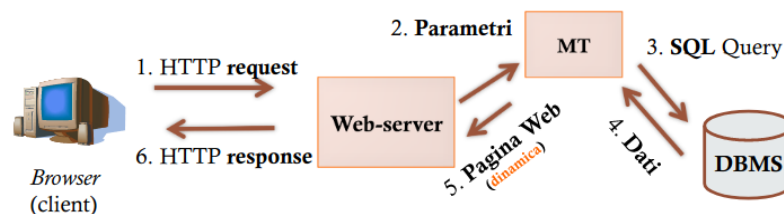


Figura 2.3: WIS Architecture

2.3 Progettazione interfaccia utente - Use Case

La necessità di realizzare un sistema informatico rispondente a quanto richiesto per un'effettiva automazione del processo di monitoraggio dei pazienti, comporta l'impiego di un approccio progettuale di Ingegneria del Software di tipo TOP-DOWN. In tale approccio assume fondamentale importanza l'analisi del dominio per giungere ad una definizione abbastanza precisa dei requisiti del sistema. Si deve comprendere cosa effettivamente deve fare un sistema software dal punto di vista dell'utilizzatore e non come lo deve fare.

Partendo, quindi, dalla comprensione degli aspetti più astratti del problema di dominio ed approfondendoli (approccio TOP-DOWN) si evita che incomprensioni del problema vengano trasferite alla fase implementativa del sistema software ottenendo come risultato la realizzazione di un sistema software di automazione non rispondente a quanto richiesto dall'utilizzatore.

L'analisi del dominio, che ha inizio con un approccio informale per la raccolta delle informazioni direttamente dall'utilizzatore e/o da documentazione, deve portare alla definizione delle specifiche del dominio e cioè alle funzionalità che il sistema informatico deve implementare.

Le specifiche devono tradursi in modelli concettuali che consentono di rappresentare il dominio applicativo ed il sistema senza possibilità di incomprensioni e fraintendimenti.

Considerata la dimensione del sistema software si è scelta una modalità di sviluppo che consente un veloce passaggio dalla fase di raccolta dei requisiti alla fase di sviluppo, limitandosi ad elaborare un numero strettamente sufficiente di modelli concettuali.

Inoltre ci si è avvalsi dell'ausilio dell'approccio basato sulla prototipazione per una migliore individuazione dei requisiti funzionali.

In ultimo, considerata la centralità dei dati rispetto alla logica ed alla riusabilità del software, si è preferito procedere secondo le modalità dell'Analisi Strutturata basata sulla separazione delle funzionalità dai dati.

Nei prossimi paragrafi verrà illustrato l'approccio Use Case per la modellazione dei requisiti funzionali che affianca il modello E-R finalizzato alla modellazione dei dati.

Caso d'Uso

Un Caso d'Uso è una sequenza di azioni che descrive l'interazione tra il sistema e l'utilizzatore per eseguire un'attività.

L'utilizzo dei casi d'uso consente di scoprire, chiarire e concordare i requisiti del sistema, ragionando con gli utilizzatori in termini di effettiva interazione con il sistema. Infatti l'elaborazione dei casi d'uso implica:

- individuare gli attori (utilizzatori);
- gli obiettivi che si intendono conseguire con l'utilizzo del sistema;
- approfondire le modalità di utilizzo.

Essi descrivono:

- il modo in cui inizia l'interazione con il sistema;
- le risposte che il sistema deve dare all'utilizzatore;
- la sequenza di passi del flusso logico;
- eventuali altri soggetti/sistemi esterni coinvolti.

I casi d'uso vengono descritti in forma testuale elencando i passi che il sistema compie. Costituiscono la base delle fasi di Analisi, Design, Implementazione e Test del sistema.

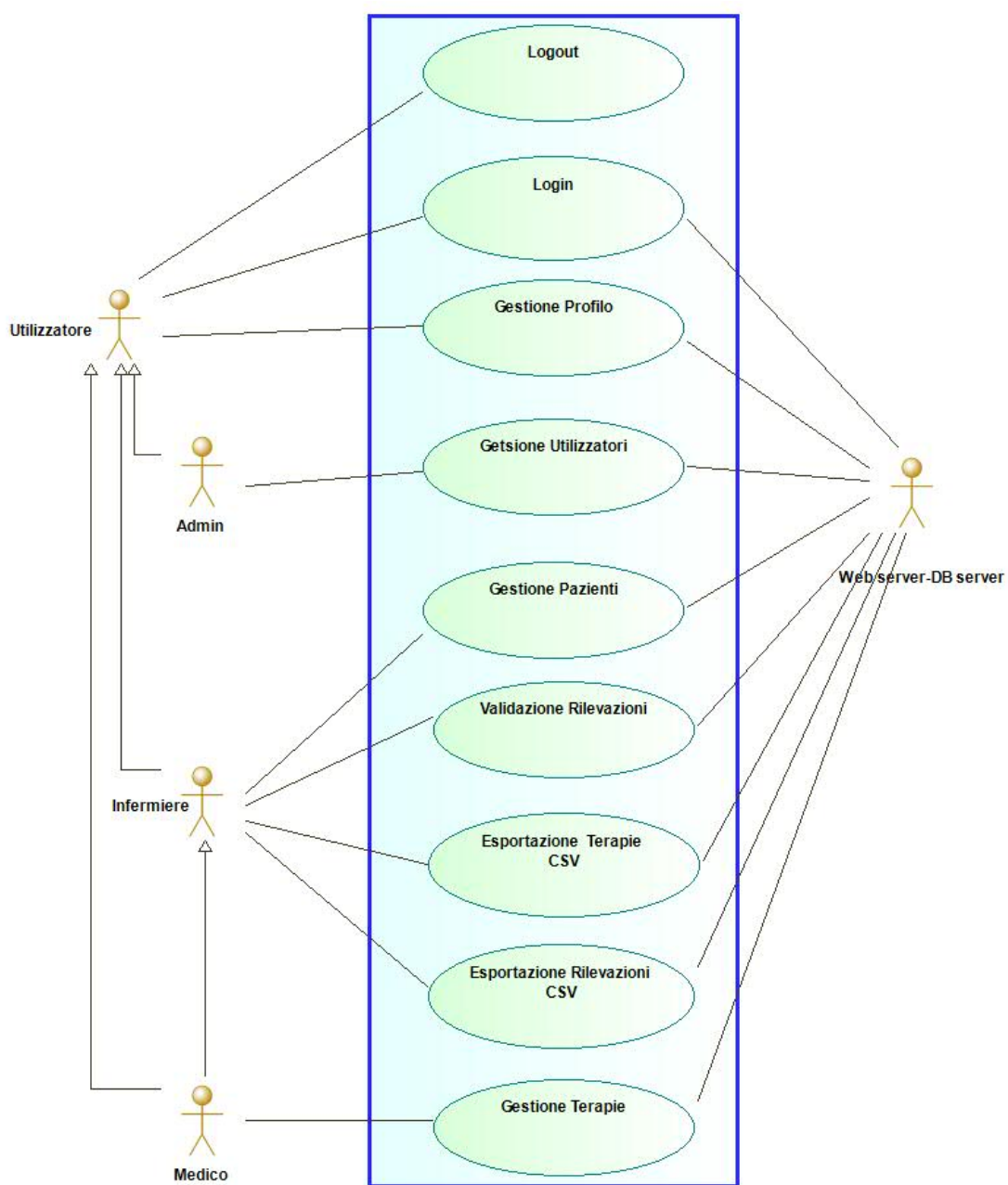


Figura 2.4: Diagramma dei casi d'uso

Casi d'Uso:

- Gestione Utilizzatori;
- Gestione Pazienti;
- Validazione Rilevazioni;
- Gestione Terapie;
- Gestione Profilo;
- Esportazione Rilevazioni CSV;
- Esportazione Terapie CSV;
- Login;
- Logout.

UC: Gestione Utilizzatori

Attore: Admin

Descrizione: Admin deve inserire/modificare un Utilizzatore

Precondizione: Admin si trova nella pagina iniziale

Scenario Principale

1. Admin inserisce il PID dell'Utilizzatore;
2. Il sistema verifica se il PID è già utilizzato;
3. Admin inserisce i dati dell'Utilizzatore.

Scenario Secondario

- 2.a.1 il sistema visualizza i dati dell'Utilizzatore con il PID inserito;
- 2.a.2 Admin modifica i dati;
- 2.a.3 Admin salva i dati dell'Utilizzatore;

Postcondizione: il sistema salva il dati dell'Utilizzatore oppure visualizza un messaggio di errore.

UC: Gestione Pazienti

Attore: Infermiere/Medico

Descrizione: L'Attore deve inserire/modificare un Paziente

Precondizione: L'Attore ha selezionato la voce "Gestione Paziente" del menu.

Scenario Principale

1. Infermiere inserisce l'ID del Paziente;
2. Il sistema verifica se l'ID è già utilizzato;
3. Infermiere inserisce i dati del Paziente;
4. Il sistema visualizza l'ID assegnato al Paziente.

Scenario Secondario

- 2.a.1 il sistema visualizza i dati del Paziente con l'ID inserito;
- 2.a.2 Infermiere modifica i dati;
- 2.a.3 Infermiere salva i dati del Paziente;

Postcondizione: il sistema salva il dati del Paziente oppure visualizza un messaggio di errore.

UC: Validazione Rilevazioni

Attore: Infermiere/Medico

Descrizione: L'Attore deve validare le rilevazioni inserendo una nota associata.

Precondizione: L'Attore ha selezionato la voce "Gestione Rilevazione" del menu ed è visualizzata la tabella con tutte le Rilevazioni ancora da validare.

Scenario Principale

1. L'Attore seleziona una riga della tabella;
2. Il sistema visualizza la pagina rappresentante lo stato del paziente;
3. L'Attore seleziona dalla tabella una Rilevazione da validare;
4. L'Attore inserisce una nota associata alla rilevazione;

5. Il sistema visualizza la tabella e la lista delle note aggiornate.

Postcondizione: il sistema salva la nota oppure visualizza un messaggio di errore.

UC: Gestione Terapie

Attore: Medico

Descrizione: Il Medico deve inserire una Terapia.

Precondizione: Il Medico ha selezionato la voce “Gestione Terapie” del menu.

Scenario Principale

1. Il Medico inserisce ID del paziente ;
2. Il sistema visualizza la pagina rappresentante lo stato del paziente;
3. Il medico inserisce una terapia;
5. Il sistema visualizza la lista delle terapie aggiornata.

Postcondizione: il sistema salva la terapia oppure visualizza un messaggio di errore.

UC: Gestione Profilo

Attore: Utilizzatore

Descrizione: Utilizzatore deve modificare la Password.

Precondizione: Utilizzatore ha selezionato l'icona “Profilo” della Tool-bar.

Scenario Principale

1. Il sistema visualizza i dati dell'Utilizzatore;
2. L'Utilizzatore inserisce nuova password;
4. il sistema verifica l'uguaglianza delle password inserite;
5. Il sistema salva la nuova password.

Scenario Alternativo

- 4.a.1 Il sistema visualizza un messaggio di errore se le password inserite non sono uguali.

Postcondizione: il sistema salva la nuova password oppure visualizza un messaggio di errore.

UC: Login

Attore: Utilizzatore

Descrizione: Utilizzatore deve accedere al sistema.

Precondizione: L'Utilizzatore ha inserito URL sistema.

Scenario Principale

1. Il sistema visualizza la pagina iniziale;
2. L'Utilizzatore inserisce PID e Password;
4. Il sistema verifica la correttezza dei dati inseriti;
5. L'Utilizzatore accede al sistema.

Scenario Alternativo

4.a.1 Il sistema visualizza un messaggio di errore se i dati inseriti sono errati.

Postcondizione: L'Utilizzatore visualizza la pagina del sistema con il proprio menu oppure un messaggio di errore.

UC: Logout

Attore: Utilizzatore

Descrizione: L'Utilizzatore deve abbandonare il sistema.

Precondizione: L'Utilizzatore ha eseguito il Login.

Scenario Principale

1. L'utilizzatore seleziona il bottone "Logout" della Tool-bar;
2. Il sistema visualizza la pagina di Login.

Postcondizione: L'Utilizzatore visualizza la pagina di Login.

2.4 Architettura

L'infrastruttura informatica prevista si articola nell'invio dei valori sanitari del paziente tramite apposita App Mobile e nella conservazione dei dati in un DB Relazionale disponibile nella LAN del Policlinico. L'accesso ai dati da parte dell'Utilizzatore può avvenire da qualsiasi postazione poiché l'interfaccia è di tipo Web pertanto accessibile grazie al solo client Browser. La componente server è a due livelli: Web server e DB server.

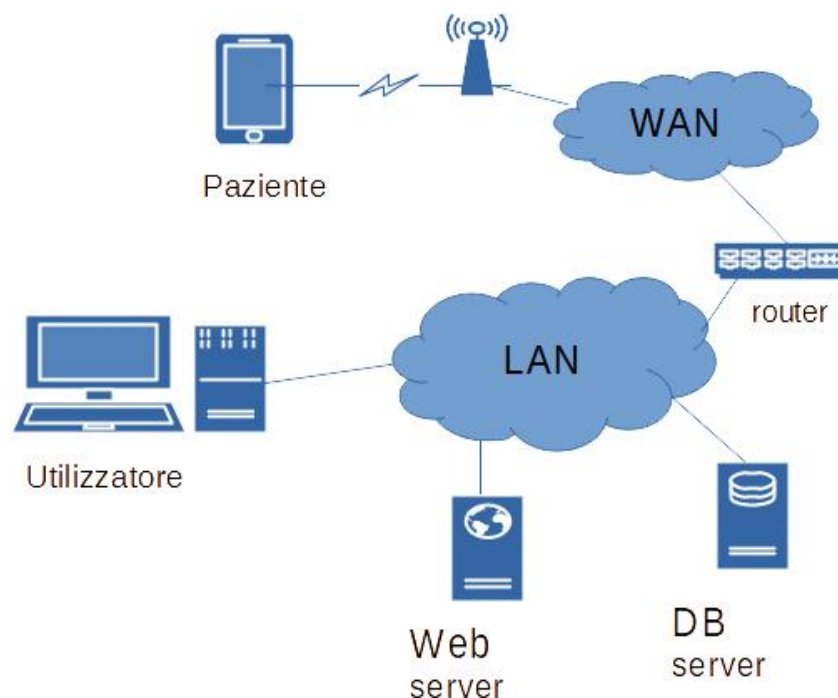


Figura 2.5: Architettura Heart

L'app mobile invia i valori dei parametri vitali via rete wireless con IP di destinazione il web server dell'ospedale utilizzando il protocollo HTTP. I dati vengono instradati dalla rete internet fino alla LAN dell'ospedale. I dati giungono al Web Server, il quale esegue la logica definita nei file PHP per svolgere le operazioni CRUD nello schema "Heart" del RDBMS MySQL, installato nel DB server dell'ospedale.

Capitolo 3

Implementazione

3.1 Tecnologie

Il Web Information System "Heart" è un RDBMS con architettura AMP ovvero è realizzato con le seguenti componenti tecnologiche:

- Apache web server
- MySQL
- PHP

Apache è uno dei server web più popolari e supporta il server-side scripting, in particolare il linguaggio PHP.

Si è optato per l'implementazione di una base dati relazionale vista predominanza di dati strutturati. L'RDBMS (Relational Database Management System) indica un database management system basato sul modello introdotto da Edgar F. Codd nel 1970 il quale presenta la sua visione di modello "relazionale" dei dati fondato sul concetto matematico di relazione tra insiemi.

Il linguaggio di scripting PHP è stato scelto perchè ha integrato il supporto per lavorare a stretto contatto con MySQL e perchè è multiplatforma.

L'interfaccia web è stata realizzata tramite il Rich Internet Application (RIA) framework lato client Angular.

Angular è un framework open source per lo sviluppo di applicazioni web dinamiche dove il linguaggio di programmazione principale è TypeScript. Angular sviluppa applicazioni web single-page client-side usando TypeScript e HTML. Le applicazioni sviluppate in Angular vengono eseguite interamente dal web browser dopo essere state scaricate dal web server (elaborazione lato client). Questo comporta una riduzione del carico di lavoro dato che non è più necessario scaricare una pagina web dal web-server ogni volta che c'è una richiesta di azione da parte dell'utente.

TypeScript è un linguaggio di programmazione open source sviluppato da Microsoft. Si tratta di un super-set di JavaScript che basa le sue caratteristiche su ECMAScript 6. TypeScript estende la libreria di JavaScript e prevede un sistema di annotazione dei tipi che consente di controllare i tipi durante la fase di compilazione. Questa caratteristica può essere ignorata se si vuole utilizzare la classica tipizzazione dinamica di JavaScript.

3.1.1 Architettura di Angular

Angular ha un'architettura CBA (Component Based Architecture) pertanto il concetto fondamentale su cui si basa il framework è il componente. Esso ha il controllo di una porzione dello schermo implementando sostanzialmente una View.

Un'applicazione è un insieme di componenti che interagiscono tra di loro. In particolare, i componenti possono essere combinati tra loro creando nuovi componenti organizzati in una struttura gerarchica. Ogni componente viene posto in una directory indipendente che contiene un frammento HTML, una dichiarazione di classe in TypeScript, e uno o più file CSS associati, ottenendo una totale autonomia ed incapsulamento del codice del componente. Un'applicazione avrà sempre un root component che costituisce in un certo senso il punto d'ingresso dell'applicazione stessa. Per convenzione al root component viene assegnato il nome di AppComponent. La root component connette una gerarchia di componenti con il Document Object Model (DOM). Angular costruisce da zero il DOM finale a partire da frammenti HTML e CSS indipendenti tra loro, inseriti ed integrati in appositi moduli TypeScript. L'avvio dell'applicazione Angular solitamente viene eseguita specificando il modulo AppModule.

Un modulo è un contenitore di funzionalità che consente di organizzare il codice di un'applicazione ed è caratterizzato dal decoratore @NgModule.

Le direttive sono componenti speciali introdotte dal decoratore @Directive che cambiano il comportamento del markup ed il DOM che viene generato.

Un decoratore è una funzione che arricchisce una classe di specifiche funzionalità.

Un componente Angular è un tipo di direttiva a cui viene applicato il decoratore @Component.

```
import { Component } from '@angular/core';
@Component({
  selector: 'app-root',
  templateUrl: './app.component.html',
  styleUrls: ['./app.component.css']
})
export class AppComponent {
  title = 'app works!';
}
```

Figura 3.1: Componente Angular

Al decoratore @Component vengono passate alcune metainformazioni che determinano come il componente si aggancia al markup.

L'applicazione può passare da un macro stato all'altro attraverso un meccanismo di routing interno che modifica anche l'URI visibile e gestisce lo scoping ed il passaggio dei dati da una parte all'altra dell'applicazione.

Il componente può essere usato all'interno di una pagina HTML inserendo l'elemento app-root. La definizione dei fogli di stile da applicare al markup determina l'aspetto grafico del componente. Questi hanno effetto soltanto sul componente e sui suoi discendenti, ma non influenzano il markup dei componenti esterni. La clausola export fa sì che il componente sia accessibile all'esterno del modulo corrente. Si accede dall'esterno al componente definito nel modulo utilizzando la clausola import.

Le metainformazioni del @Component sono :

- **selector**: elemento del markup a cui è agganciato il componente;
- **templateUrl**: indica il file html che descrive il markup del componente;
- **styleUrls**: elenco dei file CSS da applicare al markup.

I componenti utilizzano servizi che forniscono funzionalità specifiche non direttamente correlate alle visualizzazioni. I fornitori di servizi possono essere inseriti nei componenti come dipendenze, rendendo il codice modulare, riutilizzabile ed efficiente.

L'Angular Router NgModule fornisce un servizio che consente di definire un percorso di navigazione tra i diversi stati dell'applicazione e di visualizzare le gerarchie nell'applicazione. Per definire le regole di navigazione, si associano i percorsi di navigazione ai componenti. Un percorso utilizza una sintassi simile ad un URL che integra i dati del programma.

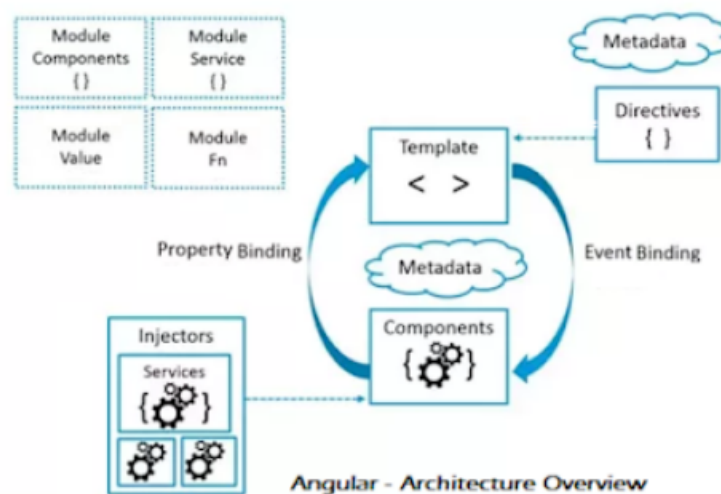


Figura 3.2: Architettura Angular

Una definizione di classe di servizio è immediatamente preceduta dal decoratore @Injectable. L'iniezione di dipendenza (DI) consente di mantenere le classi dei componenti snelle ed efficienti in quanto non recuperano dati dal server, non convalidano l'input dell'utente delegando tali compiti ai servizi.

Event binding consente all'app di rispondere all'input dell'utente nell'ambiente di destinazione aggiornando i dati dell'applicazione.[20]

Angular permette di effettuare chiamate Ajax (Asynchronous Javascript And XML) verso il web-server tramite le classi appartenenti al modulo HttpClient. Ajax consiste nella comunicazione asincrona con tra client e server ovvero una richiesta Ajax viene eseguita in maniera asincrona rispetto alla navigazione della pagina HTML. Il web server processa la chiamata ed invia la risposta al front-end. Questa risposta è essenzialmente XML o JSON, ovvero stringhe di testo che permettono di ottimizzare il consumo della banda. Il front-end Angular processa la risposta e modifica opportunamente la pagina web. Tramite Ajax è possibile inserire il risultato delle elaborazioni lato server in pagine web statiche lato client senza dover eseguire il refresh della pagina.[23]

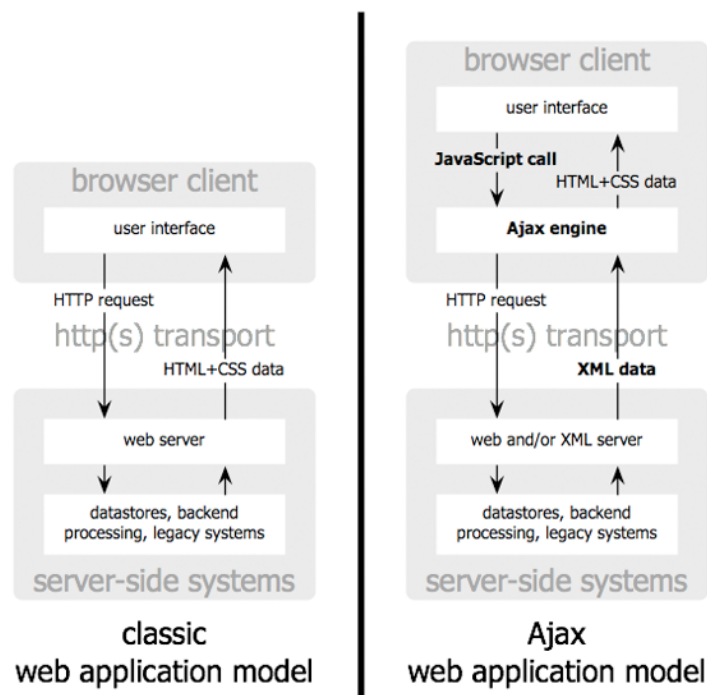


Figura 3.3: Ajax web application model

3.1.2 PHP

Lato server sono stati realizzati degli script in PHP. Il linguaggio di scripting viene eseguito dal server web prima di restituire il suo output al browser. L'utilizzo di script lato server comporta alcuni vantaggi.[24]

Gli script PHP vengono interpretati dal server web indipendentemente dal linguaggio interpretato dal browser, ciò non crea problemi di compatibilità con i browser. Il codice eseguito lato-server permette di ridurre il carico di lavoro sul client.

Il PHP è stato sviluppato principalmente per il Web, questo implica che la maggior parte delle sue caratteristiche sono state implementate in funzione di esso. Il linguaggio consente di accedere in maniera semplicissima alle richieste HTTP di tipo GET e POST.

Il PHP è multiplatforma, cioè può essere utilizzato sia in ambienti Unix-based (Linux, Mac OSX) che su Windows.

La combinazione più utilizzata e scelta per questa tesi, però, resta quella LAMP ovvero Linux come sistema operativo, Apache come Web server, MySQL per il database e PHP come linguaggio di scripting server-side.[17]

Angular fornisce un'API HTTP client semplificata, la classe di servizio `HttpClient`.

Il servizio client HTTP di Angular offre:

- Capacità di richiedere in risposta oggetti tipati.
- Gestione degli errori.
- Intercettazione di richieste e risposte.

Si è fatto uso della classe di servizio Angular `HttpClient` che implementa un meccanismo per la comunicazione tra il front-end ed il server remoto tramite il protocollo HTTP ed ogni suo metodo restituisce un oggetto di tipo `Observable`. L'oggetto di tipo `Observable` funge da supporto per il passaggio di dati tra diverse parti. L'Observer Pattern è un software design pattern in cui un oggetto, chiamato `subject`, gestisce una lista di dipendenti, chiamati `observers`, e gli notifica automaticamente dei suoi cambi di stato. La classe di servizio `mysql.service.ts` gestisce il collegamento tra front-end e back-end. Ogni metodo implementato invoca un metodo GET/POST della classe `HttpClient` che

richiede in argomento il path dello script PHP e, in caso di metodo POST, un JSON contenente i dati della query.

```
mysql.service.ts

@Injectable({
  providedIn: 'root'
})
export class MysqlService {

  public user : User=null;

  httpOptions = {
    headers: new HttpHeaders({
      'Content-Type': 'application/json',
    })
  };

  options: {
    headers?: HttpHeaders | {[header: string]: string | string[]},
    observe?: 'body' | 'events' | 'response',
    params?: HttpParams|{[param: string]: string | string[]},
    reportProgress?: boolean,
    responseType?: 'arraybuffer'|'blob'|'json'|'text',
    withCredentials?: boolean,
  };
};
```

```
mysql.service.ts

constructor(private http: HttpClient) { }

getuser(g: FormGroup) : Observable<User> {
  console.warn(g.value);
  const log=this.http.post("/api/log.php", { data:'', PID: g.get('pid').value, action:'login'}, this.httpOptions);
  log.subscribe();
  return this.http.post<User>("/api/login.php", g.value, this.httpOptions).pipe(catchError(this.handleError));
}

getAllmed() : Observable<User []> {
  const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, action:'getAllUser'}, this.httpOptions);
  log.subscribe();
  return this.http.get<User[]>("/api/getallmed.php", {observe: 'body', responseType: 'json'}).pipe(catchError(this.handleError));
}

allpaz() : Observable<Paz []> {
  const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, action:'allpaz'}, this.httpOptions);
  log.subscribe();
  return this.http.get<Paz[]>("/api/allpaz.php", {observe: 'body', responseType: 'json'}).pipe(catchError(this.handleError));
}

uppwd (g:Np) : Observable<any> {
  console.warn(g);
  const log=this.http.post("/api/log.php", { data:'', PID: g.PID, action:'updatepwd'}, this.httpOptions);
  log.subscribe();
  return this.http.post<any>("/api/uppwd.php", g, this.httpOptions).pipe(catchError(this.handleError));
}
```

Ecco un esempio di script PHP *allril.php* che gestisce la connessione con il db MySQL ed invoca una select per estrarre tutte le rilevazioni, di un paziente specifico se l'ID è diverso da 0 oppure di tutti i pazienti se il valore dell'ID è uguale a 0.

Di seguito è riportato il codice.

```
allr1.php
1 <?php
2 $postdata = file_get_contents("php://input");
3 $request = json_decode($postdata);
4 header('Content-type: application/json');
5 header("Access-Control-Allow-Origin: *");
6 header('Access-Control-Allow-Headers: X-Requested-With, content-type, access-control-allow-origin,
7 access-control-allow-methods, access-control-allow-headers');
8 try {
9     //connessione al db mysql
10    $pdo=new PDO('mysql:host=localhost;dbname=heart','user','heart99');
11    //Sollevare un'eccezione se l'oggetto PDO non riesce ad eseguire un'operazione sul DB
12    $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
13    $pdo->exec("set names utf8");
14    $pd=$request->idpaz;
15    $dt1=json_encode($request->dt1);
16    $dt2=json_encode($request->dt2);
17    //select sql per estrarre le rilevazioni del paziente con ID!=0
18    if($pd != 0){
19        $sql="SELECT r.ID, r.ID_paz, r.data, r.ID, r.peso, r.frequenza, r.pressione, r.passi, r.nota,
20        r.allarme_aut, l.rpm, l.watt, l.flusso, l.picco, l.depressione, l.PI, l.allarme, v.data as datav, v.nota as notav, v.PID_u as PIDv
21        FROM Rilevazione as r LEFT JOIN Validazione as v ON v.dataR=r.data
22        LEFT JOIN LVAD as l ON l.ID_rilev=r.ID where (r.ID_paz=$pd) AND (r.data >= $dt1 AND r.data <= $dt2)";
23    }
24    //select per estrarre le rilevazioni di tutti i pazienti
25    else {
26        $sql="SELECT r.ID, r.ID_paz, r.data, r.ID, r.peso, r.frequenza, r.pressione, r.passi, r.nota,
27        r.allarme_aut, l.rpm, l.watt, l.flusso, l.picco, l.depressione, l.PI, l.allarme, v.data as datav, v.nota as notav, v.PID_u as PIDv
28        FROM Rilevazione as r LEFT JOIN Validazione as v ON v.dataR=r.data
29        LEFT JOIN LVAD as l ON l.ID_rilev=r.ID where (r.data >= $dt1 AND r.data <= $dt2)";
30    }
31    //invio della query al db
32    $stmt = $pdo->prepare( $sql );
33    $stmt->execute();
34 }
```

```

allril.php
10 //Sollevare un'eccezione se l'oggetto PDO non riesce ad eseguire un'operazione sul DB
11 $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
12 $pdo->exec("set names utf8");
13 $pd=$request->idpaz;
14 $dt1=json_encode($request->dt1);
15 $dt2=json_encode($request->dt2);
16 //select sql per estrarre le rilevazioni del paziente con ID!=0
17 if($pd != 0){
18     $sql="SELECT    r.ID, r.ID_paz, r.data, r.ID, r.peso, r.frequenza, r.pressione, r.passi, r.nota,
19     r.allarme_aut, l.rpm, l.watt, l.flusso, l.picco, l.depressione, l.PI, l.allarme, v.data as datav, v.nota as notav, v.PID_u as PIDv
20     FROM Rilevazione as r LEFT JOIN Validazione as v ON v.dataR=r.data
21     LEFT JOIN LVAD as l ON l.ID_rilev=r.ID where (r.ID_paz=$pd) AND (r.data >= $dt1 AND r.data <= $dt2)";
22 }
23 //select per estrarre le rilevazioni di tutti i pazienti
24 else {
25     $sql="SELECT    r.ID, r.ID_paz, r.data, r.ID, r.peso, r.frequenza, r.pressione, r.passi, r.nota,
26     r.allarme_aut, l.rpm, l.watt, l.flusso, l.picco, l.depressione, l.PI, l.allarme, v.data as datav, v.nota as notav, v.PID_u as PIDv
27     FROM Rilevazione as r LEFT JOIN Validazione as v ON v.dataR=r.data
28     LEFT JOIN LVAD as l ON l.ID_rilev=r.ID where (r.data >= $dt1 AND r.data <= $dt2)";
29 }
30 //invio della query al db
31 $stmt = $pdo->prepare( $sql );
32 $stmt->execute();
33 }
34 catch(PDOException $e) {
35     echo("[ERRORE] Select al DB non riuscita. Errore: ".$e->getMessage()."<br>");
36     exit();
37 }
38 //vengono prelevate le righe del result set
39 $row = $stmt->fetchAll(PDO::FETCH_ASSOC);
40 //invia al browser un oggetto json contenente dalla variabile $row
41 echo json_encode($row);
42 ?>

```

Lo script PHP *insril.php* gestisce l'inserimento di una rilevazione nel database. In seguito alla connessione al database, esegue una prima operazione di select per estrarre il paziente con l'ID corrispondente alla variabile \$pd ricevuto lato client. Viene verificata l'effettiva corrispondenza dei dati anagrafici quali nome e cognome del paziente individuato dal DB ed i dati ricevuti in argomento alla chiamata. In caso di esattezza dei dati,

viene invocata la stored procedure *InsRil* che gestisce l'inserimento della rilevazione per un dato paziente nello script SQL.

```
insril.php
1 <?php
2 $postdata = file_get_contents("php://input");
3 $request = json_decode($postdata);
4 header('Content-type: application/json');
5 header("Access-Control-Allow-Origin: *");
6 header('Access-Control-Allow-Headers: X-Requested-With, content-type, access-control-allow-origin,
7 access-control-allow-methods, access-control-allow-headers');
8
9 $pd=json_encode($request->id);
10 $pd1=$request->id;
11 $peso = $request->peso;
12 $frequenza= $request->frequenza;
13 $pressione= json_encode($request->pressione);
14 $passi= $request->passi;
15 $nota= json_encode($request->nota);
16 $rpm= $request->rpm;
17 $watt= $request->watt;
18 $flusso= $request->flusso;
19 $picco= $request->picco;
20 $depressione= $request->depressione;
21 $pi= $request->pi;
22 $allarme= json_encode($request->alarm);
23 $nome= json_encode($request->nome);
24 $cognome= json_encode($request->cognome);
25
26 try {
27     $pdo=new PDO('mysql:host=localhost;dbname=heart','user','heart99');
28     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
29     $pdo->exec("set names utf8");
30     //select per estrarre paziente con ID dichiarato in argomento al metodo.
31     $sql="SELECT nome, cognome FROM Paziente WHERE ID=$pd";
32     $stmt = $pdo->prepare( $sql );
33     $stmt->execute();
34     $row = $stmt->fetch(PDO::FETCH_ASSOC);
```

```
insril.php
21 $pi= $request->pi;
22 $allarme= json_encode($request->alarm);
23 $nome= json_encode($request->nome);
24 $cognome= json_encode($request->cognome);
25
26 try {
27     $pdo=new PDO('mysql:host=localhost;dbname=heart','user','heart99');
28     $pdo->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
29     $pdo->exec("set names utf8");
30     //select per estrarre paziente con ID dichiarato in argomento al metodo.
31     $sql="SELECT nome, cognome FROM Paziente WHERE ID=$pd";
32     $stmt = $pdo->prepare( $sql );
33     $stmt->execute();
34     $row = $stmt->fetch(PDO::FETCH_ASSOC);
35     //verifica che il nome ed il cognome ricevuti in argomento alla chiamata http corrispondono ai valori
36     //del paziente con l'ID dichiarato in argomento alla chiamata.
37     if(json_encode($row["nome"]) == strtoupper($nome) && json_encode($row["cognome"]) == strtoupper($cognome)){
38         //chiamata alla procedura in sql InsRil che inserisce la rilevazione nel db.
39         $sql="CALL InsRil($pd,$peso,$frequenza,$pressione,$passi,$nota,'',$rpm,$watt,$flusso,$picco,$depressione,$pi,$allarme)";
40         $stmt1 = $pdo->prepare( $sql );
41         $stmt1->execute();
42     }
43     else {
44         echo json_encode('ID/nome/cognome errato');
45         exit();
46     }
47 }
48 catch(PDOException $e) {
49     echo("[ERRORE] Select al DB non riuscita. Errore: ".$e->getMessage()."<br>");
50     exit();
51 }
52 echo json_encode($row["nome"]);
53 ?>
```

I due script PHP sopra riportati sono rappresentativi di ognuno dei 21 script PHP realizzati per gestire le operazioni CRUD necessarie al funzionamento del sistema.

3.1.3 MySQL (Structured Query Language)

Nello script SQL è stato implementato un trigger in SQL-3. Il trigger è un costrutto potente in grado di reagire ad eventi. I trigger, detti regole, seguono il paradigma Evento-Condizione-Azione rendendo la base di dati *attiva*. L'esecuzione delle regole avviene sotto il controllo di un sotto-sistema autonomo detto processore delle regole (rule engine) che tiene traccia degli eventi permettendo l'esecuzione alternata di transazioni lanciate dagli utenti e di regole lanciate dal sistema.[25]

I trigger hanno due livelli di granularità:

- di tupla (row-level)
- di primitiva (statement-level)

e due modalità :

- immediata (opzione after)
- differita (opzione before)

Il trigger di tupla in modalità immediata inserisce nell'attributo `allarme_aut` dei pazienti LVAD la stringa "Watt alti" o "Flusso basso" se, rispettivamente, il valore dei watt è aumentato di 1 unità o il flusso sanguigno si è ridotto di 1.5 litri rispetto alla rilevazione meno recente inserita nei 15 giorni precedenti.

Una stored procedure gestisce l'inserimento della rilevazione sia in caso di paziente LVAD che non. Le stored procedure, così chiamate perché vengono memorizzate nella base di dati come parte dello schema, permettono di associare un nome ad un'istruzione CRUD.[18]

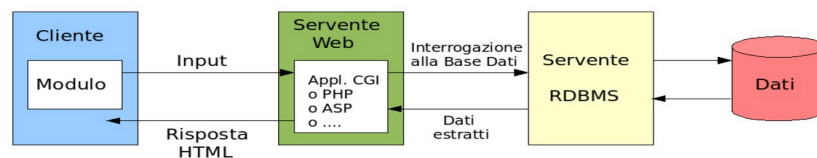


Figura 3.4: RDBMS

Script SQL

Sono riportati alcuni frammenti di codice dello script SQL implementato utilizzando il GUI tool MySQL Workbench.[28]

Implementazione tabella Utilizzatore:

#Creazione tabelle con relativi vincoli di integrità referenziale e chiave primaria

```
CREATE TABLE Utilizzatore(  
    PID varchar(200) primary key not null,  
    Email varchar(200)not null,  
    Password varchar(200) not null,  
    nome varchar(200) not null,  
    cognome varchar(200) not null,  
    num_tel varchar(200) not null,  
    Categoria enum("Medico","Infermiere","admin") not null  
)engine=innodb;
```

Implementazione tabella Paziente e Rilevazione:

```
CREATE TABLE Paziente(  
    ID int auto_increment primary key,  
    nome varchar(200) not null,  
    cognome varchar(200) not null,  
    num_tel varchar(200) not null,  
    luogo_nascita varchar(200) not null,  
    data_nascita date not null,  
    residenza varchar(200) not null,  
    sesso enum("M","F") not null,  
    peso int,  
    altezza int,  
    class_gravita enum ("alta","media","bassa"),
```



```
class_NYHA int not null,  
class_INTERMACS int not null,  
tipologia enum("LVAD HM3","LVAD HVAD","NO LVAD"),  
PID_medico varchar(200),  
foreign key (PID_medico) REFERENCES Utilizzatore(PID)  
on delete set null  
)engine=innodb;  
  
CREATE TABLE Rilevazione(  
ID int auto_increment primary key,  
data datetime not null,  
ID_paz int,  
peso int default 0,  
frequenza int default 0,  
pressione varchar(200) default "",  
passi int default 0,  
nota varchar(1000) default "",  
allarme_aut varchar(200) default "",  
foreign key (ID_paz) REFERENCES Paziente(ID) on delete set null  
)engine=innodb;
```

Implementazione trigger:

```
#Creazione trigger allarme_aut su flusso e watt pazienti LVAD  
/*Allarme viene settato dopo aver confrontato il flusso  
e/o watt della rilevazione inserita con  
la rilevazione più recente tra quelle più vecchie di 15gg */  
DELIMITER |  
create trigger AllarmeFlusso  
after insert on LVAD  
for each row  
begin
```

```
declare paziente int;
declare data_ril datetime;
declare allarme_autV varchar(200);
set paziente= (SELECT ID_paz from Rilevazione
where (Rilevazione.ID=new.ID_rilev));
set data_ril=(SELECT data from Rilevazione
where (Rilevazione.ID=new.ID_rilev) );
if(new.flusso <= ((SELECT flusso
from Rilevazione R join LVAD L on R.ID=L.ID_rilev
where (R.ID_paz=paziente) AND (R.data<date_add(data_ril,interval -15 DAY))
order by R.data desc LIMIT 1)) -1.5)
then
update Rilevazione set allarme_aut="Flusso basso"
where (Rilevazione.ID=new.ID_rilev);
end if;

set allarme_autV=(SELECT allarme_aut FROM Rilevazione
where (ID=last_insert_id()));
if(new.watt >=((SELECT watt
from Rilevazione R join LVAD L on R.ID=L.ID_rilev
where (R.ID_paz=paziente)
AND (R.data<date_add(data_ril,interval -15 DAY))
order by R.data desc LIMIT 1)) +1)
then
update Rilevazione set allarme_aut =CONCAT("Watt alti - ",allarme_autV)
where (Rilevazione.ID=new.ID_rilev);
end if;
end|
DELIMITER ;
```

Per lo script completo vedi Appendice.

3.1.4 Apache

Apache è uno dei server web più usati al mondo ed è il più importante tra i progetti posti sotto l'egida dell'omonima Apache Software Foundation, una delle più grandi comunità distribuite di sviluppatori che lavorano su progetti open source. [26] Un server web è un sistema informatico che si occupa di rispondere a richieste di risorse usando il protocollo HTTP (acronimo che sta per HyperText Transfer Protocol). Apache supporta anche il cosiddetto server-side scripting, ciò permette di generare dinamicamente i documenti da servire. Tra i linguaggi di scripting supportati il più popolare è il PHP.

3.2 Implementazione dell'applicativo in Angular

Il progetto ha una struttura composta dal file *index.html* che a sua volta si connette alla componente *app.component.html*. Ogni componente è associato al file in TypeScript e al foglio di stile. In *app.component.ts* è inserito il tag `<router-outlet>` che si ricollega alla classe *app-routing.module.ts*, descritta successivamente. La classe *app-routing.module.ts* associa ad ogni path il componente Angular corrispondente.

```
app-routing.module.ts
20 const routes: Routes = [
21   { path: 'login', component: LoginComponent },
22   { path: 'admin',
23     component: AdminComponent,
24     canActivate: [AdminGuard],
25   },
26   { path: 'inf',
27     component: InfComponent,
28     canActivate: [InfGuard],
29     canActivateChild: [InfGuard],
30     children: [
31       { path: 'gestpaz', component: GestpazComponent },
32       { path: 'visrilinf', component: VisrilinfComponent },
33       { path: 'pazst/:id', component: PazstComponent },
34       { path: 'estrdati', component: EstrdatiComponent },
35     ],
36   },
37   { path: 'med',
38     component: MedComponent,
39     canActivate: [MedGuard],
40     canActivateChild: [MedGuard],
41     children: [
42       { path: 'visrilmed', component: VisrilmedComponent },
43       { path: 'visrilinf', component: VisrilinfComponent },
44       { path: 'estrdati', component: EstrdatiComponent },
45     ],
46   },
47   { path: 'profilo',
48     canActivate: [ProfilGuard],
49     component: ProfiloComponent },
50   { path: '**', component: LoginComponent },
51 ];
52
53 @NgModule({
54   declarations: [],
55   imports: [
56     RouterModule.forRoot(routes),
57   ],
58   exports: [RouterModule],
59 })
60 export class AppRoutingModule { }
61
```

La classe di servizio *mysql.service.ts* implementa i metodi che invocano metodi (Get/Post) della classe *HttpClient* di Angular che inviano richieste HTTP allo script PHP definito nell'argomento del metodo.

```
mysqlservice.ts
14 @Injectable({
15   providedIn: 'root'
16 })
17 export class MysqlService {
18
19   public user : User=null;
20
21   httpOptions = {
22     headers: new HttpHeaders({
23       'Content-Type': 'application/json',
24     })
25   };
26   options: {
27     headers?: HttpHeaders | {[header: string]: string | string[]},
28     observe?: 'body' | 'events' | 'response',
29     params?: HttpParams | {[param: string]: string | string[]},
30     reportProgress?: boolean,
31     responseType?: 'arraybuffer'|'blob'|'json'|'text',
32     withCredentials?: boolean,
33   };
34   private handleError(error: HttpResponse) {
35     let errorMessage='';
36     if (error.error instanceof ErrorEvent) {
37       // Si è verificato un errore sul lato client o di rete. Gestirlo di conseguenza.
38       errorMessage='Errore client ${error.message}';
39     } else {
40       //Il backend ritorna un code di errore
41       // Il corpo della risposta può contenere indizi su cosa è andato storto.
42       errorMessage=
43         `Errore server: ${error.status}, ` +
44         `descrizione: ${error.message}`;
45     }
46     //Messaggio di errore
47     window.alert(errorMessage);
48     return throwError(errorMessage);
49
50
51 //metodo per la ricerca di un utilizzatore tramite pid
52 srcpid(g: FormGroup) : Observable<User> {
53   console.warn(g.value);
54   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID_u:g.value, action:'srcutil'}, this.httpOptions);
55   log.subscribe();
56   return this.http.post<User>("/api/srcpid.php", g.value, this.httpOptions).pipe(catchError(this.handleError));
57 }
58 //Metodo per estrarre tutte le rilevazioni
59 allril(g: FormGroup) : Observable<Rilev> {
60   console.warn(g.value);
61   console.log(g.value);
62   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID_u:g.value, action:'allril'}, this.httpOptions);
63   log.subscribe();
64   return this.http.post<Rilev>("/api/allril.php", g.value, this.httpOptions).pipe(catchError(this.handleError));
65 }
66 //Metodo per estrarre tutte le terapie
67 alltrp(g:number) : Observable<Terap> {
68   console.warn(g);
69   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID_u:g, action:'alltrp'}, this.httpOptions);
70   log.subscribe();
71   return this.http.post<Terap>("/api/alltrp.php", g, this.httpOptions).pipe(catchError(this.handleError));
72 }
73 //Metodo per inserire un utilizzatore
74 isrutl (g: User) : Observable<any> {
75   console.warn(g);
76   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, PID_ins:g.PID, action:'insutil'}, this.httpOptions);
77   log.subscribe();
78   return this.http.post<any>("/api/isrutl.php", g, this.httpOptions).pipe(catchError(this.handleError));
79 }
80 }
```

```

mysql.service.ts
115 //Metodo per la ricerca di un paziente tramite il suo ID
116 srcpaz(g: FormGroup) : Observable<Paz> {
117   console.warn(g.value);
118   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID:g.get('id').value, action:'srcpaz'}, this.httpOptions);
119   log.subscribe();
120   return this.http.post<Paz>("/api/srcpaz.php", g.value, this.httpOptions).pipe(catchError(this.handleError));
121 }
122 //Metodo per l'inserimento di un paziente
123 isrpaz (g: Paz) : Observable<any> {
124   console.warn(g);
125   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID:g.ID, action:'inspaz'}, this.httpOptions);
126   log.subscribe();
127   return this.http.post<any>("/api/isrpaz.php", g, this.httpOptions).pipe(catchError(this.handleError));
128 }
129 //Metodo per la modifica di un paziente
130 mdpaz (g: Paz) : Observable<any> {
131   console.warn(g);
132   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID:g.ID, action:'modpaz'}, this.httpOptions);
133   log.subscribe();
134   return this.http.post<any>("/api/mdpaz.php", g, this.httpOptions).pipe(catchError(this.handleError));
135 }
136 //Metodo per estrarre rilevazioni
137 srcril() : Observable<Rilev []> {
138   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, action:'srcril'}, this.httpOptions);
139   log.subscribe();
140   return this.http.get<Rilev []>("/api/srcril.php", {observe: 'body', responseType: 'json'}).pipe(catchError(this.handleError));
141 }
142 //Metodo per cercare le rilevazioni da validare di un paziente definito da ID
143 srcrilid (g: number) : Observable<Rilev []> {
144   console.warn(g);
145   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID:g, action:'srcrilid'}, this.httpOptions);
146   log.subscribe();
147   return this.http.post<Rilev []>("/api/srcrilid.php", g, this.httpOptions).pipe(catchError(this.handleError));
148 }
149 }
150 }

mysql.service.ts
149 //Metodo per la ricerca delle rilevazioni di un paziente (ID) con nota ovvero validate
150 srcrilidcn (g: number) : Observable<Rilev []> {
151   console.warn(g);
152   return this.http.post<Rilev []>("/api/srcrilidcn.php", g, this.httpOptions).pipe(catchError(this.handleError));
153 }
154 //Metodo per la ricerca di rilevazioni di un paziente (ID) inserite negli ultimi 30 giorni. Utilizzate per i grafici
155 srcrilidgf (g: number) : Observable<Rilev []> {
156   console.warn(g);
157   return this.http.post<Rilev []>("/api/srcrilidgf.php", g, this.httpOptions).pipe(catchError(this.handleError));
158 }
159 //Metodo per l'inserimento di una nota inerente ad una rilevazione
160 insnt (g: Valid) : Observable<any> {
161   console.warn(g);
162   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID:g.ID_pazR, dataR:g.dataR, action:'insnt'}, this.httpOptions);
163   log.subscribe();
164   return this.http.post<any>("/api/insnt.php", g, this.httpOptions).pipe(catchError(this.handleError));
165 }
166 //Metodo per la ricerca di una terapia
167 srctrp (g: number) : Observable<Terap []> {
168   console.warn(g);
169   return this.http.post<Terap []>("/api/srctrp.php", g, this.httpOptions).pipe(catchError(this.handleError));
170 }
171 //Metodo che inserisce una terapia
172 instrp (g: Terap) : Observable<any> {
173   console.warn(g);
174   const log=this.http.post("/api/log.php", { data:'', PID:this.user.PID, ID:g.ID_paz, action:'instrp'}, this.httpOptions);
175   log.subscribe();
176   return this.http.post<any>("/api/instrp.php", g, this.httpOptions).pipe(catchError(this.handleError));
177 }

```

La classe *app-routing.module.ts* invoca il componente *aut* che funge da schermata di login. La form per il login di *aut.component.html* invoca il metodo "getUser()" della classe *mysqlservice* che restituisce un oggetto Observable sul quale, invocando il metodo "subscribe", viene attivato l'observer pattern. Se l'utente che ha tentato il login è riconosciuto come di categoria "medico" viene invocato il metodo della classe *app-routing.module.ts* `router.navigate(['/med'])`, se infermiere `router.navigate(['/inf'])`, se admin `router.navigate(['/admin'])`.

```
aut.component.html
1 <div class="logtxt">
2   
3   <h1>HEART</h1>
4   <h2>Sistema Web gestione pazienti cardiopatici Policlinico S.Orsola-Malpighi</h2>
5   <h3>D.I.S.I.</h3>
6   <br>
7   <br>
8   <form [formGroup]="userForm" (ngSubmit)="onSubmit()">
9     <label>
10      PID:
11      <input type="text" FormControlName="pid">
12    </label>
13    <label>
14      Password:
15      <input type="password" FormControlName="pwd">
16    </label>
17    <button type="submit" [disabled]="!userForm.valid">Login</button>
18  </form>
19 </div>
20
```

```
aut.component.ts
7
8 @Component({
9   selector: 'app-aut',
10  templateUrl: './aut.component.html',
11  styleUrls: ['./aut.component.css']
12 })
13
14 export class AutComponent implements OnInit {
15   constructor(private mysqlservice: MySQLService, private router: Router) { }
16   ngOnInit(): void {
17     const pid = this.userForm.get('pid');
18     pid.valueChanges.subscribe(() => {
19       pid.patchValue(pid.value.toUpperCase(), {emitEvent: false});
20     });
21   }
22   private user: User;
23   userForm = new FormGroup({
24     pid: new FormControl(''),
25     pwd: new FormControl(''),
26   });
27
28   onSubmit(): void {
29     //invoca il metodo subscribe sul metodo getuser della classe mysqlservice
30     this.mysqlservice.getUser(this.userForm).subscribe((user: User | any)=>{
31       this.user=user;this.mysqlservice.user=user;console.log(this.mysqlservice.user);
32       //se pid o pwd errati
33       if (user==false){
34         window.alert("Utente non presente o Password errata");};
35       if(this.user.Categoria=="admin"){this.router.navigate(['/admin']);};
36       if(this.user.Categoria=="Infermiere"){this.router.navigate(['/inf']);};
37       if(this.user.Categoria=="Medico"){this.router.navigate(['/med']);}; });
38   }
39 }
```

Nel caso del login come admin si aggiunge il componente Admin che contiene una form per la ricerca di un utilizzatore e due tag `<app-mdfutl>` e `<app-isrutl>` che sono associati a due diversi componenti, il primo per la modifica dell'utilizzatore con PID digitato in caso esso già esista e il secondo per l'inserimento di un nuovo utilizzatore se non ne esiste alcuno con il PID inserito nella form.

Nell'immagine a seguire è riportato il codice del "admin.component" per l'interfaccia dedicata all'amministratore.


```
admin.component.html
1 <app-top-bar></app-top-bar>
2 <div class="logtxt">
3   <h5>Inserire PID dipendente per modifica o inserimento</h5>
4   <form [formGroup]="utilForm" (ngSubmit)="onSubmit()" >
5     <label>
6       PID:
7       <input type="text" FormControlName="pid" required>
8     </label>
9     <button type="submit" [disabled]="!utilForm.valid">Cerca</button>
10  </form>
11  <app-mdfctl *ngIf="src&&this.util!=null" [util]="util"></app-mdfctl>
12  <app-isrctl *ngIf="src&&this.util==null"></app-isrctl>
13 </div>

admin.component.ts
/
8 @Component({
9   selector: 'app-admin',
10  templateUrl: './admin.component.html',
11  styleUrls: ['./admin.component.css']
12 })
13 export class AdminComponent implements OnInit {
14
15   util :User;
16   src=false;
17   utilForm = new FormGroup({
18     pid: new FormControl(''),
19   });
20
21   constructor(private mysqlservice: MySQLService) { }
22   ngOnInit(): void {
23     const pid = this.utilForm.get('pid');
24     pid.valueChanges.subscribe() => {
25       pid.patchValue(pid.value.toUpperCase(), {emitEvent: false});
26     };
27   }
28
29   onSubmit(): void {
30     this.util=null;
31     this.src=true;
32     this.mysqlservice.srcpid(this.utilForm).subscribe((user:User| any)->{
33       if (user!=false) {this.util=user;}
34       else {this.util=null;};
35       console.log(this.util);
36       if (!this.util){
37         window.alert("Utilizzatore non presente");}} });
38   }
39
40 }
```

Nell'immagine a seguire è riportato il codice del "inf.component" per l'interfaccia dedicata all'infermiere.

```
inf.component.html
1 <app-top-bar></app-top-bar>
2 <nav>
3   <ul>
4     <li><a routerLink="gestpaz">Gestione Pazienti</a></li>
5     <li><a routerLink="visrilinf">Gestione Rilevazioni</a></li>
6     <li><a routerLink="estrdati">Esportazione dati</a></li>
7   </ul>
8 </nav>
9 <router-outlet></router-outlet>

inf.component.ts
1 import { Component, OnInit } from '@angular/core';
2
3 @Component({
4   selector: 'app-inf',
5   templateUrl: './inf.component.html',
6   styleUrls: ['./inf.component.css']
7 })
8 export class InfComponent implements OnInit {
9
10  constructor() { }
11  ngOnInit(): void {
12  }
13 }
14
```

Nell'immagine a seguire è riportato il codice del "med.component" per l'interfaccia dedicata al medico.

```
med.component.html
1 <app-top-bar></app-top-bar>
2 <nav>
3   <ul>
4     <li><a routerLink='visrilmed'>Gestione Terapie</a></li>
5     <li><a routerLink="visrilinf">Gestione Rilevazioni</a></li>
6     <li><a routerLink="estrdati">Esportazione dati</a></li>
7   </ul>
8 </nav>
9 <router-outlet></router-outlet>
10

med.component.ts
1 import { Component, OnInit } from '@angular/core';
2 @Component({
3   selector: 'app-med',
4   templateUrl: './med.component.html',
5   styleUrls: ['./med.component.css']
6 })
7 export class MedComponent implements OnInit {
8   constructor() { }
9   ngOnInit(): void {
10  }
11
12 }
```

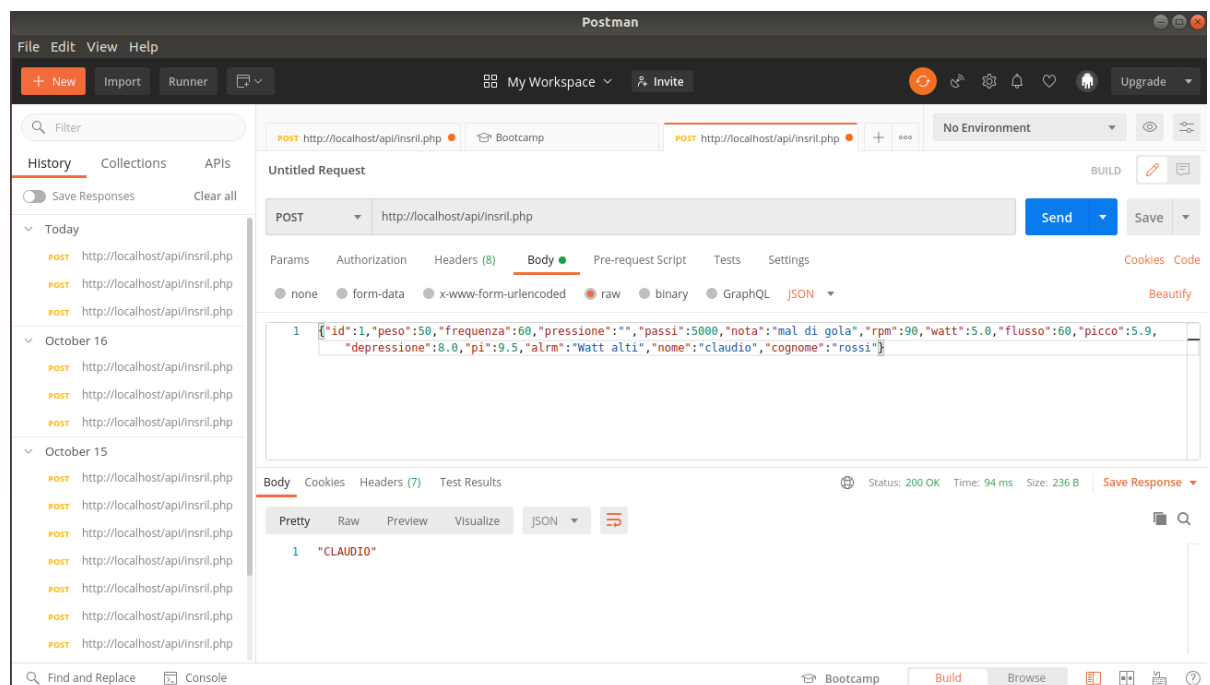
Per ogni funzionalità del sistema è stato realizzato un componente che ne gestisce l'interfaccia web e la logica.

3.3 Fase di Testing delle API REST

Le API REST implementate in PHP sono state sottoposte a testing tramite l'utilizzo del tool Postman.[27]

Postman è una piattaforma di collaborazione per lo sviluppo di API. Le funzionalità di Postman semplificano ogni passaggio della creazione di un'API e ottimizzano la collaborazione in modo da poter creare API migliori. E' uno strumento che permette di testare le nostre API in maniera molto semplice senza dover costruire un driver adeguato. Una risposta API è costituita dal corpo, dalle intestazioni e dal codice di stato. Postman organizza il corpo e le intestazioni in diverse schede e ci offre tre aree per visualizzare una risposta. La modalità "Pretty" compone la risposta in maniera JSON/XML in modo tale che sia più facile da leggere e da visualizzare.

Sono state simulate richieste HTTP di tipo GET e POST in base alla funzione fornita dall'API e le risposte sono state visualizzate in modalità Pretty ovvero in formato JSON. Nell'immagine seguente viene illustrato come è stata testata l'API per l'inserimento delle rilevazioni.



3.4 App Mobile

L'App Mobile funge da front-end finalizzato alla raccolta delle rilevazioni giornaliere inserite dai pazienti e il loro invio al back-end. E' già disponibile ed è stata sviluppata per sistemi operativi Android con il linguaggio di programmazione Kotlin.

L'applicazione Mobile consente l'assistenza di due tipologie di pazienti: quelli con scompenso cardiaco e quelli portatori di LVAD. Rientra nella categoria di App che ricopre un settore specifico dell'ambito biomedicale. L'App è stata pensata ponendo il paziente al centro del suo sviluppo diversamente dal back-end che è stato realizzato sulla base delle necessità dell'équipe medica.[3]

Capitolo 4

Validazione

4.1 Implementazione delle funzionalità

Il sistema web Heart consente l'archiviazione, l'analisi e la gestione dei dati sanitari dei pazienti con patologie cardiovascolari. I dati vengono inviati tramite apposita App ed inseriti in un database centralizzato. Lo scopo di Heart è consentire l'immediata visualizzazione, da parte del personale specialistico, sia paramedico che medico, dei dati ricevuti. In particolare l'utilizzatore può visualizzare i dati di tutte le rilevazioni, anche graficamente, ed inserire annotazioni relative ad ognuna di esse. Il medico, oltre a quanto detto per il personale paramedico, può inserire anche una o più terapie.

Accesso al sistema

L'accesso al sistema è possibile con PID e password attraverso la pagina di login. Il PID è un codice identificativo univoco dell'utilizzatore inserito in fase di registrazione da parte dell'Amministratore del sistema.

E' possibile inserire utilizzatori di tre categorie: Admin, Infermiere, Medico. In base alla categoria di appartenenza si disporrà di funzionalità appropriate.



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



HEART

Sistema Web gestione pazienti cardiopatici Policlinico S.Orsola-Malpighi

D.I.S.I.

PID: Password:

Inserimento Utilizzatore

Un utente con categoria Admin può inserire/modificare i dati relativi ad un utilizzatore. Nella apposita interfaccia è necessario inserire il PID identificativo del dipendente ed il sistema:

1. se è già presente un utilizzatore con il PID inserito, visualizza e consente di modificare i dati dell'utilizzatore (compresa la password) ad esclusione del PID;
2. se non è già presente un utilizzatore con il PID inserito, consente di inserire i dati dell'utilizzatore creando un'utenza con il PID e la password inseriti.

Inserire PID dipendente per modifica o inserimento

PID:

Modifica Utilizzatore

Cognome:

Nome:

Mail:

Telefono:

Categoria Utilizzatore:

Password:

Inserire PID dipendente per modifica o inserimento

PID:

Inserisci Utilizzatore

PID:

Cognome:

Nome:

Mail:

Telefono:

Categoria Utilizzatore:

Password:

Inserimento/Modifica Paziente

L'utilizzatore con categoria Infermiere o Medico, tramite il link "Gestione Pazienti", può inserire/modificare i dati di un paziente. L'interfaccia consente di inserire l'ID univoco del paziente.

Se:

1. l'ID inserito è già assegnato ad un paziente, il sistema consente la modifica dei dati del paziente con esclusione dell'ID;
2. l'ID non è già assegnato, l'interfaccia consente l'inserimento di un nuovo paziente. Al termine dell'inserimento il sistema visualizza l'ID del paziente inserito da utilizzare successivamente nell'App per l'invio dei dati.

Per inserire direttamente un nuovo utente è possibile utilizzare ID=0.

Attenzione: l'ID visualizzato nell>alert dopo l'inserimento del paziente va memorizzato ed inserito nell'App per l'invio dei dati.

The screenshot displays the HEART system interface. At the top, there is a red header with the text "HEART" on the left and a "Logout" button on the right. Below the header, a dark grey navigation bar contains two buttons: "Gestione Pazienti" and "Gestione Rilevazioni". The main content area features a search bar with the text "ID: 0" and a "Cerca" button. Below the search bar, the title "Inserimento Paziente" is centered. To the left of the form, a list of labels is provided: "Cognome", "Nome", "Luogo Nascita", "Residenza", "Data Nascita", "Telefono", "Sesso", and "Peso". The form itself consists of several input fields: a text field for "Cognome", a text field for "Nome", a text field for "Luogo Nascita", a text field for "Residenza", a date field for "Data Nascita" with a placeholder "gg / mm / aaaa", a text field for "Telefono", a dropdown menu for "Sesso", and a text field for "Peso" with a "0" value and a small icon on the right.

Modifica Password

La password di un utilizzatore può essere modificata dall'Admin con le modalità sopra descritte (modifica dati utilizzatore) oppure dall'utilizzatore stesso dall'interfaccia "Profilo", visibile tramite apposita icona disposta in alto a sinistra della top bar.



PID: 1

Cognome: verdi

Nome: stefano

Categoria: admin

Email: STEFANOVERDI@GMAIL.COM

Numero telefono: 3345424422

nuova password: nuova password:

Validazione Rilevazioni

L'utilizzatore con categoria Infermiere può selezionare il link "Gestione Rilevazioni" per visualizzare una tabella con tutte le rilevazioni, di tutti i pazienti, ancora non validate. La tabella è ordinabile in modo crescente o decrescente sulle colonne ID e data.

ID	ID Paz	data	peso	frequenza	pressione	passi	rpm	watt	flusso	picco	depressione	PI	allarme	allarme_aut	nota
1	1	2020-07-02 00:00:00	61	67	alta	300	4	4	5	1.1	1.2	1.3	Watt alti		
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt alti		
3	1	2020-08-03 00:00:00	61	67	alta	200	4	3	5	1.1	1.2	1.3	Watt alti		
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt alti		
5	1	2020-08-05 00:00:00	57	100	alta	100	2	5.1	6	2.1	1.7	1	Watt alti	Watt alti -	

Items per page: 5 1 - 5 of 9

Selezionata una riga, viene visualizzata la pagina dello stato del paziente scelto che presenta:

- sul lato sinistro i grafici dei dati sanitari delle rilevazione degli ultimo 30 giorni;
- al centro in alto una tabella con tutte le rilevazioni ancora non validate del paziente;
- sul lato destro tutte le note inserite per la validazione delle rilevazioni del paziente.

Per validare una Rilevazione è necessario selezionarla dalla tabella ed inserire la nota. Inserita la nota, la relativa rilevazione viene eliminata dalla tabella e sul lato destro della pagina compare l'elenco di tutte le note inserite.

Parametri ultimi 30 giorni

Rilevazioni da validare

ID	ID_paz	Data	Peso	Frequenza	Pressione	Passi	RPM	Watt	Flusso	Picco	Depressione	PI	Allarm
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt ai
3	1	2020-08-03 00:00:00	61	67	alta	200	4	3	5	1.1	1.2	1.3	Watt ai
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt ai

Note Rilevazioni

PID: 2
 Data val.: 2020-09-02 11:14:49
 Data ril.: 2020-07-02 00:00:00
 watt troppo alti

Inserimento Terapie

L'utilizzatore con categoria Medico dispone del link "Gestione Rilevazioni" (vedi paragrafo precedente) e del link Gestione Terapie per l'inserimento delle terapie. La relativa interfaccia presenta un campo di input dove inserire l'ID del paziente al quale si vuole prescrivere una terapia. Inserito l'ID, verrà visualizzata la pagina del paziente che riporta:

- sulla sinistra tutte le note inserite per la validazione delle rilevazioni del paziente;
- sulla destra l'elenco delle terapie già inserite per il paziente;

- in alto, al centro, i grafici relativi ai dati sanitari del paziente comunicati negli ultimi 30 giorni.



Esportazione dati

Sia il medico che l'infermiere hanno accesso alla funzionalità di esportazione dei dati in un file csv. E' possibile esportare le rilevazioni di tutti i pazienti o di uno soltanto, inserite nel sistema nel periodo definito tramite i due DatePicker. E' possibile esportare le terapie di tutti o di uno specifico paziente identificato tramite ID.

The screenshot shows the HEART application interface. At the top, there is a red header with the text 'HEART' on the left and a user profile icon, a 'Logout' button, and the text 'PID: 2' and 'CAT: Infermiere' on the right. Below the header is a dark navigation bar with three menu items: 'Gestione Pazienti', 'Gestione Rilevazioni', and 'Esportazione dati'. The main content area is white and features the title 'Esportazione Rilevazioni in file CSV' in red. Below the title, there are two date input fields: 'Periodo: da: 26 / 10 / 2020' and 'a: 26 / 10 / 2020'. Below these is a dropdown menu labeled 'Paziente:'. A red message below the dropdown reads: 'Se non viene selezionato alcun paziente, estre i dati di tutti i pazienti'. At the bottom of the form is a button labeled 'Estrai'.

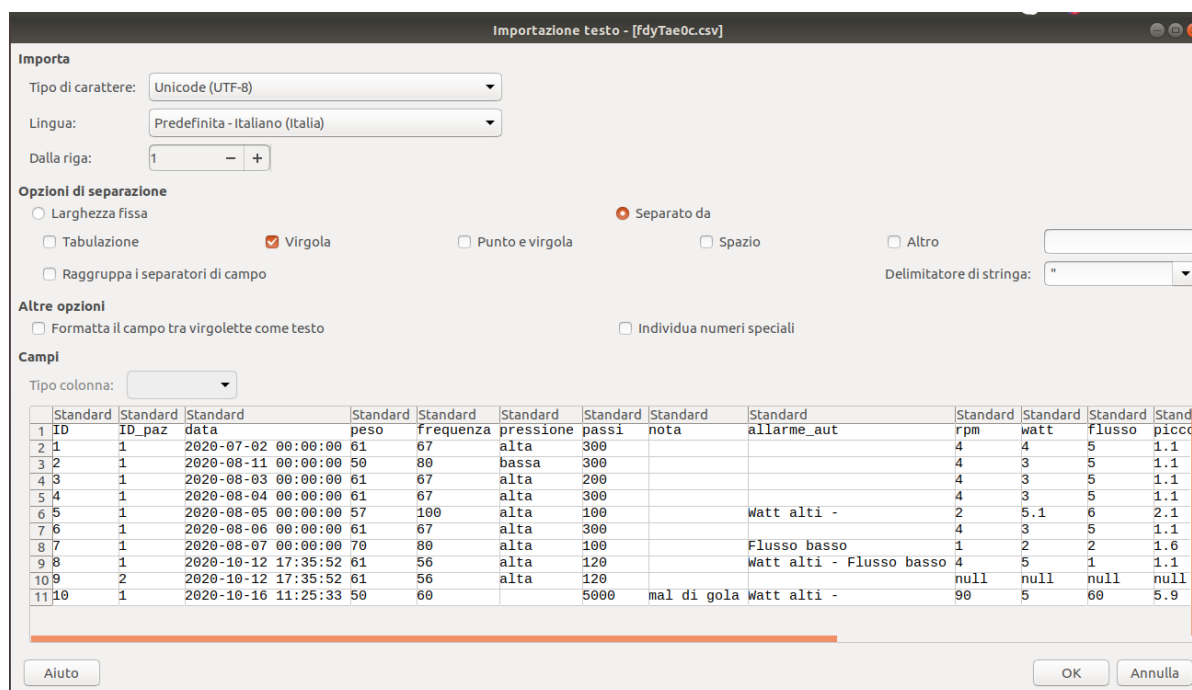
L'utilizzatore, medico/infermiere, può selezionare il periodo di suo interesse ed il paziente di cui vuole avere un prospetto delle rilevazioni inserite in tale periodo.

Esportazione terapie in file CSV

Paziente:

Se non viene selezionato alcun paziente, estre i dati di tutti i pazienti

Le terapie vengono estratte in base al paziente selezionato.



Viene generato un file csv. Nell'immagine mostrata sono esportate le rilevazioni di tutti i pazienti inserite dalla primavera 2020 ad oggi.

4.1.1 Casi D'uso

Mostriamo adesso una serie di casi d'uso dell'applicazione sviluppata, raggruppati in due user experience. Il primo, consiste nella validazione di una rilevazione da parte dell'utilizzatore infermiere con esportazione delle rilevazioni del paziente di interesse. Il secondo caso d'uso mostra i passi che il medico segue per inserire una terapia destinata ad un paziente dove, ad operazione conclusa, esporta i dati delle terapie ad egli prescritte.

4.1.2 L'infermiere valida una rilevazione

L'infermiere, già registrato dall'admin sulla piattaforma come utilizzatore con PID 2 e password "inf", esegue il login.



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA 

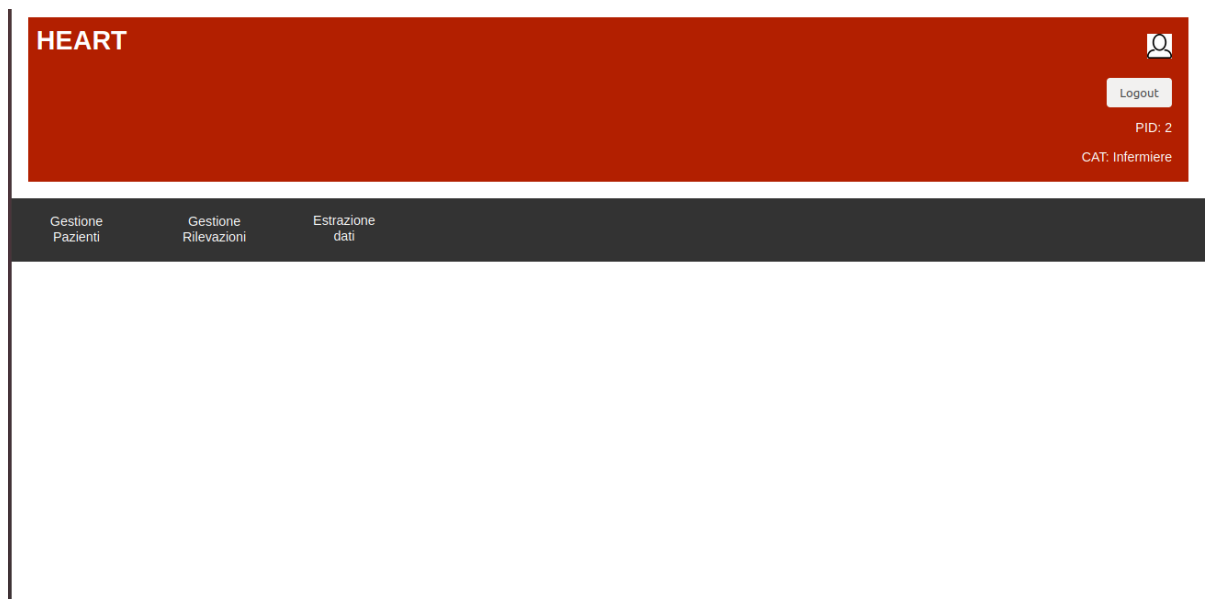
HEART


Sistema Web gestione pazienti cardiopatici Policlinico S.Orsola-Malpighi

D.I.S.I.

PID: Password:

Entra nella home page destinata all'utente della categoria infermiere.



HEART 

PID: 2

CAT: Infermiere

Gestione Pazienti Gestione Rilevazioni Estrazione dati

Seleziona la funzionalità nella top-bar "Gestione Rilevazioni" e visualizza una tabella contenente tutte le rilevazioni non validate in ordine, dalla meno recente alla più recente, in modo da evidenziare le rilevazioni da validare con più urgenza.

The screenshot shows the HEART application interface. At the top, there is a red header with the text "HEART" and a user profile icon. Below the header, there is a navigation bar with three tabs: "Gestione Pazienti", "Gestione Rilevazioni" (which is highlighted), and "Estrazione dati". The main content area displays a table of patient vital signs. The table has 15 columns: ID, ID Paz, data, peso, frequenza, pressione, passi, rpm, watt, flusso, picco, depressione, PI, allarme, allarme_aut, and nota. The table contains 5 rows of data. At the bottom right of the table, there is a pagination control showing "Items per page: 5" and "1 - 5 of 10".

ID	ID Paz	data	peso	frequenza	pressione	passi	rpm	watt	flusso	picco	depressione	PI	allarme	allarme_aut	nota
1	1	2020-07-02 00:00:00	61	67	alta	300	4	4	5	1.1	1.2	1.3	Watt alti		
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt alti		
3	1	2020-08-03 00:00:00	61	67	alta	200	4	3	5	1.1	1.2	1.3	Watt alti		
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt alti		
5	1	2020-08-05 00:00:00	57	100	alta	100	2	5.1	6	2.1	1.7	1	Watt alti	Watt alti -	

Seleziona una rilevazione del paziente di interesse.

The screenshot shows the HEART application interface. At the top, there is a red header with the text "HEART" and a user profile icon. Below the header, there is a navigation bar with three tabs: "Gestione Pazienti", "Gestione Rilevazioni" (which is highlighted), and "Esportazione dati". The main content area displays a table of patient vital signs. The table has 15 columns: ID, ID Paz, data, peso, frequenza, pressione, passi, rpm, watt, flusso, picco, depressione, PI, allarme, allarme_aut, and nota. The table contains 5 rows of data. The last row (ID 10) is highlighted in grey, indicating it is selected. At the bottom right of the table, there is a pagination control showing "Items per page: 5" and "6 - 10 of 13".

ID	ID Paz	data	peso	frequenza	pressione	passi	rpm	watt	flusso	picco	depressione	PI	allarme	allarme_aut	nota
6	1	2020-08-06 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt alti		
7	1	2020-08-07 00:00:00	70	80	alta	100	1	2	2	1.6	1	1.9	Watt alti	Flusso basso	
8	1	2020-11-12 11:10:39	61	56	alta	120	4	5	1	1.1	1.2	1.3	Watt alti	Watt alti - Flusso basso	
9	1	2020-11-12 11:10:39	67	58	alta	180	2	4	2	4.1	3.2	2.3	Watt alti	Watt alti - Flusso basso	
10	1	2020-11-12 11:10:39	57	68	alta	130	1	4.6	3	4.6	2.2	3.3	Watt alti	Watt alti - Flusso basso	

Si visualizzano tutte le rilevazioni appartenenti al paziente con lo stesso ID della rilevazione selezionata. L'infermiere seleziona la rilevazione che vuole validare e visualizza tutti i dati della rilevazione in fondo alla tabella. Si digita la nota da inserire nella text-box in basso.

CAT: Infermiere

Gestione Pazienti Gestione Rilevazioni Esportazione dati

Parametri ultimi 30 giorni

Rilevazioni da validare

ID	ID_paz	Data	Peso	Frequenza	Pressione	Passi	RPM	Watt	Flusso	Picco	Depressione	PI	Allar
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt :
3	1	2020-08-03 00:00:00	61	67	alta	200	4	3	5	1.1	1.2	1.3	Watt :
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt :

Nota:

Valore Pressione da tenere sotto controllo.

Note Rilevazioni

PID: 2
 Data val.: 2020-11-12 11:19:39
 Data ril.: 2020-07-02 00:00:00
 Valore Watt anomalo, aumentare i controlli.

CAT: Infermiere

Gestione Pazienti Gestione Rilevazioni Esportazione dati

Parametri ultimi 30 giorni

Rilevazioni da validare

ID	ID_paz	Data	Peso	Frequenza	Pressione	Passi	RPM	Watt	Flusso	Picco	Depressione	PI	Allar
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt :
3	1	2020-08-03 00:00:00	61	67	alta	200	4	3	5	1.1	1.2	1.3	Watt :
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt :

Nota:

Valore Pressione da tenere sotto controllo.

Note Rilevazioni

PID: 2
 Data val.: 2020-11-12 11:19:39
 Data ril.: 2020-07-02 00:00:00
 Valore Watt anomalo, aumentare i controlli.

Una volta effettuato correttamente l'inserimento della nota, questa è visibile nella colonna di destra.

CAT: Infermiere

Gestione Pazienti Gestione Rilevazioni **Esportazione dati**

Parametri ultimi 30 giorni

Rilevazioni da validare

ID	ID_paz	Data	Peso	Frequenza	Pressione	Passi	RPM	Watt	Flusso	Picco	Depressione	PI	Allar
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt :
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt :
5	1	2020-08-05 00:00:00	57	100	alta	100	2	5.1	6	2.1	1.7	1	Watt :

ID: 0 ID_paz: 0 Data: Peso: 0 Frequenza: 0 Pressione: 0
Watt: 0 Flusso: 0 Picco: 0 Depres.: 0 PI: 0 Passi: 0
RPM: 0 Allar.: Allar. aut:

Nota:

Inserisci nota

Note Rilevazioni

PID: 2
Data val.: 2020-11-12 11:19:39
Data ril.: 2020-07-02 00:00:00
Valore Watt anomalo, aumentare i controlli.

PID: 2
Data val.: 2020-11-12 11:21:25
Data ril.: 2020-08-03 00:00:00
Valore Pressione da tenere sotto controllo.

Dalla tabella delle rilevazioni è stata rimossa la rilevazione validata. L'infermiere ora esporta le rilevazioni del paziente con ID 1 rimaste invalidate.

HEART Logout PID: 2 CAT: Infermiere

Gestione Pazienti Gestione Rilevazioni **Esportazione dati**

ID	ID Paz	data	peso	frequenza	pressione	passi	rpm	watt	flusso	picco	depressione	PI	allarme	allarme_aut	nota
2	1	2020-08-11 00:00:00	50	80	bassa	300	4	3	5	1.1	1.2	1.3	Watt alti		
4	1	2020-08-04 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt alti		
5	1	2020-08-05 00:00:00	57	100	alta	100	2	5.1	6	2.1	1.7	1	Watt alti	Watt alti -	
6	1	2020-08-06 00:00:00	61	67	alta	300	4	3	5	1.1	1.2	1.3	Watt alti		
7	1	2020-08-07 00:00:00	70	80	alta	100	1	2	2	1.6	1	1.9	Watt alti	Flusso basso	

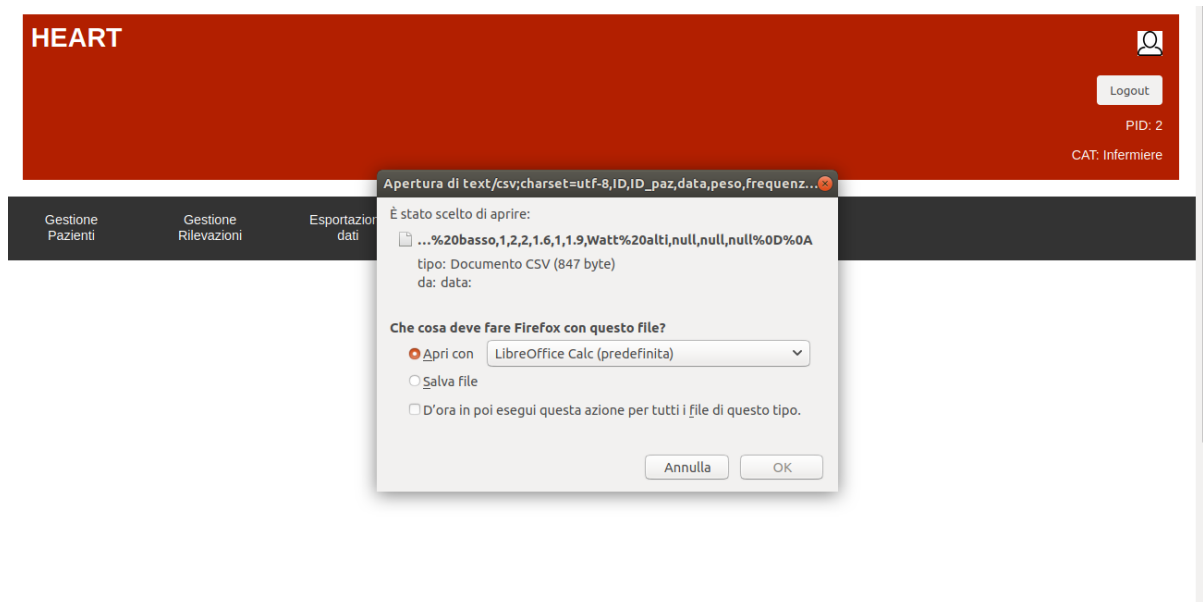
Items per page: 5 1 - 5 of 9

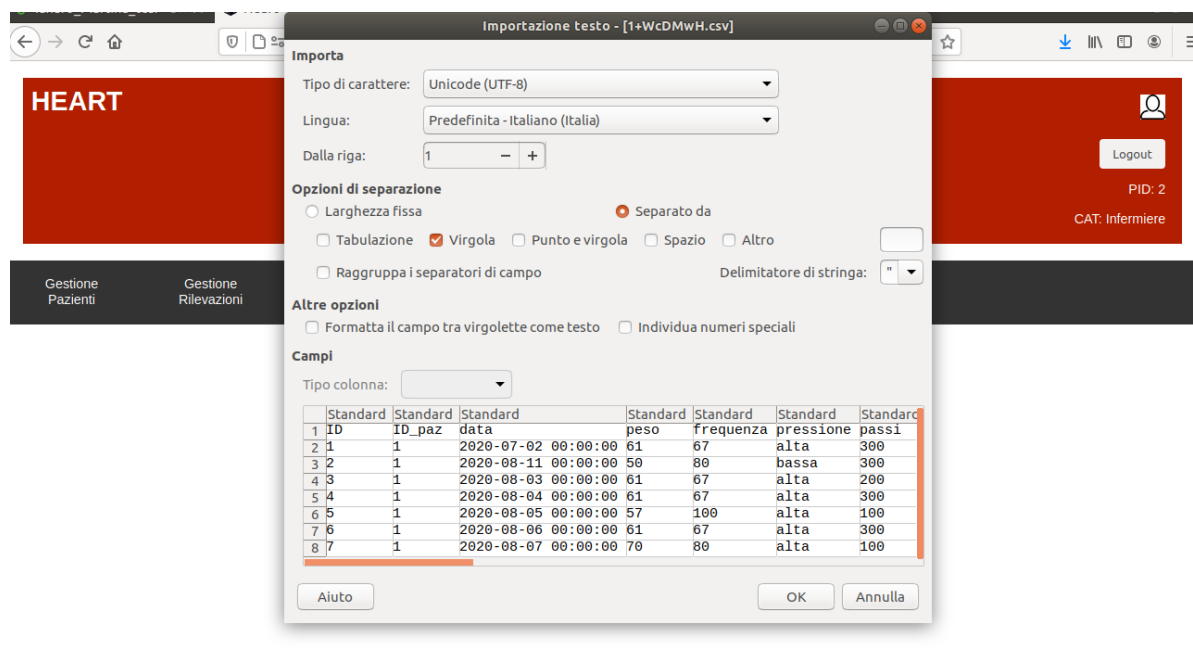
Seleziona il periodo di interesse ed il paziente di cui si vogliono ottenere le rilevazioni, nel caso mostrato sono selezionate tutte le rilevazioni inserite da 01/05/2020 al 12/11/2020

dal paziente con ID 1.



I dati vengono esportati in un file csv. Li visualizza con LibreOffice Calc.





Per terminare l'utilizzatore effettua il logout.

4.1.3 Il medico inserisce una terapia

Il medico, già registrato come utilizzatore dall'admin, ha PID 3 e password "med". Effettua il login e seleziona la funzionalità "Gestione terapie" destinata unicamente ad un utilizzatore della categoria "medico".



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA  *for Zaky*

HEART

Sistema Web gestione pazienti cardiopatici Policlinico S.Orsola-Malpighi

D.I.S.I.

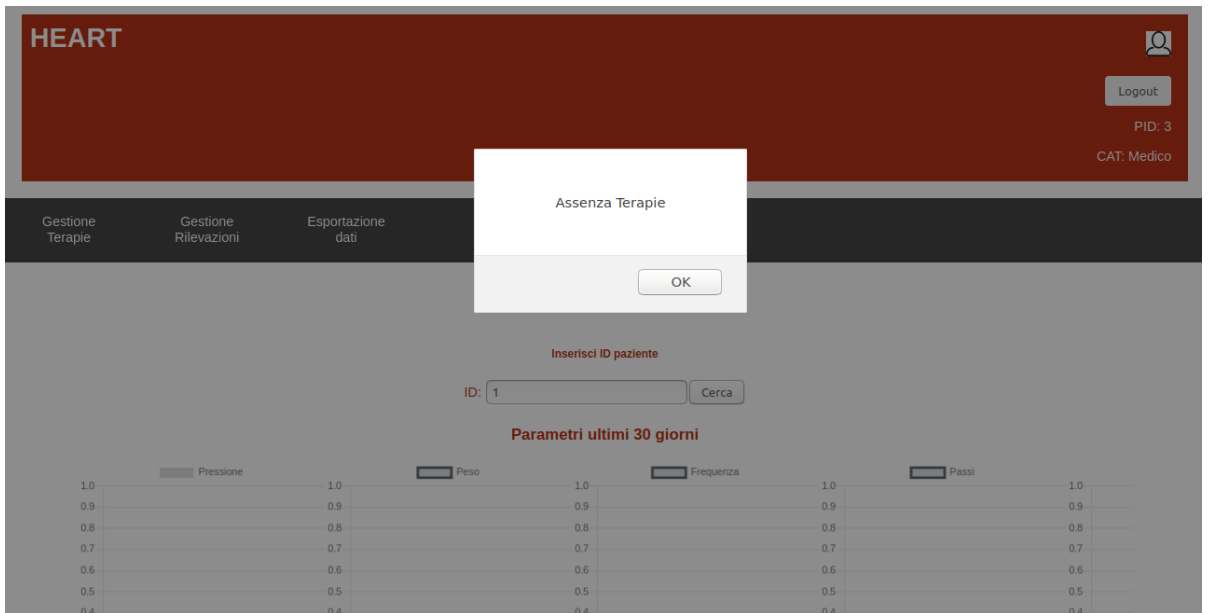
PID: Password:

Il medico inserisce l'ID del paziente per il quale si vuole prescrivere una terapia, in questo caso il paziente con ID 1.

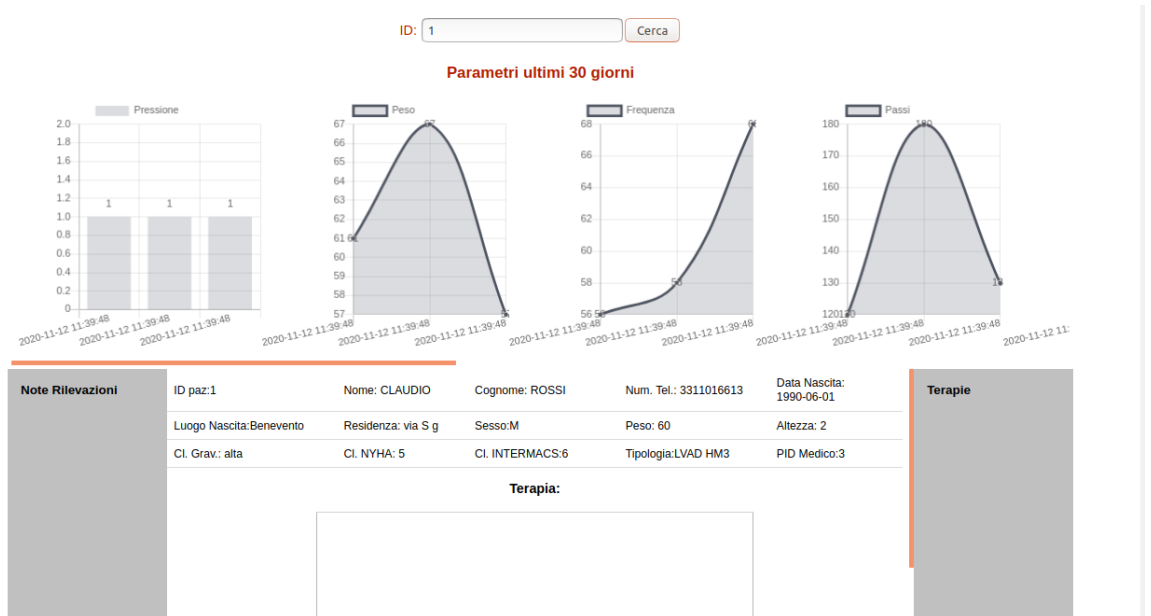


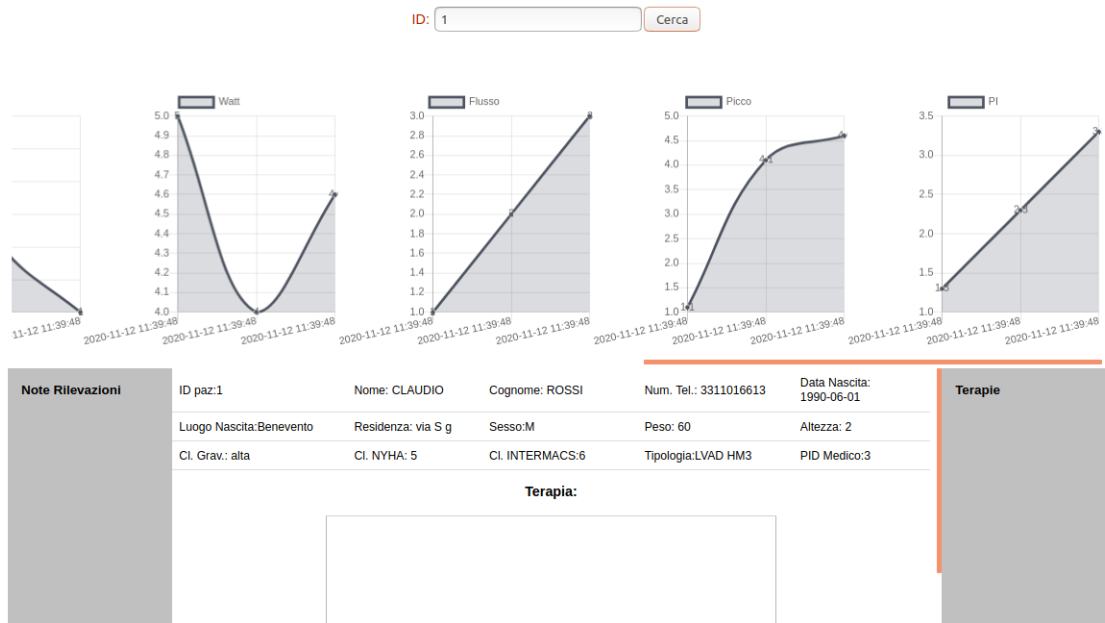
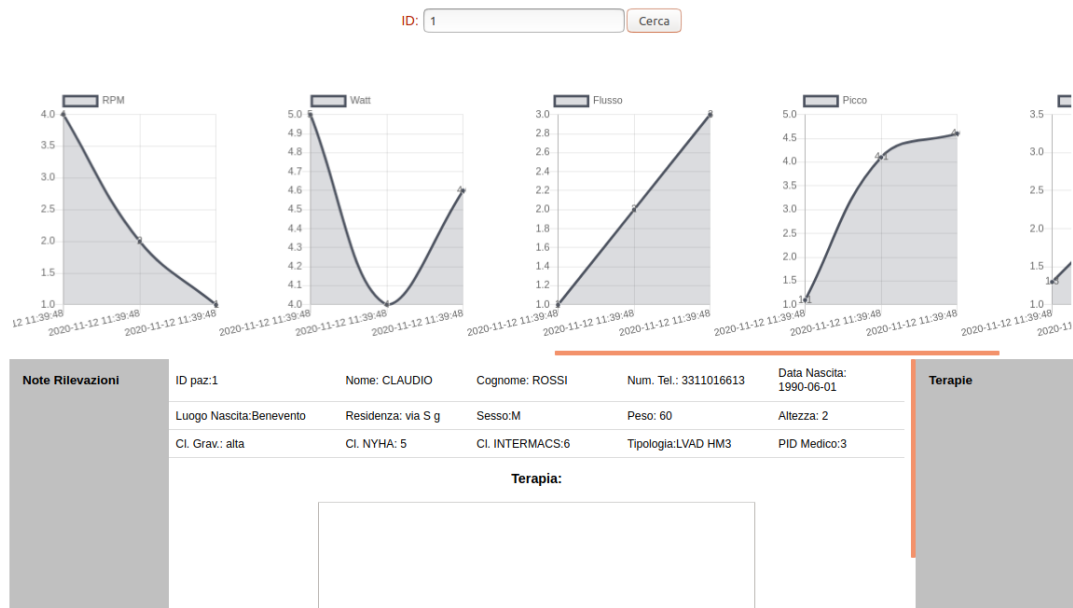
Inserisci ID paziente
ID:

Una finestra di alert annuncia la mancanza di terapie.

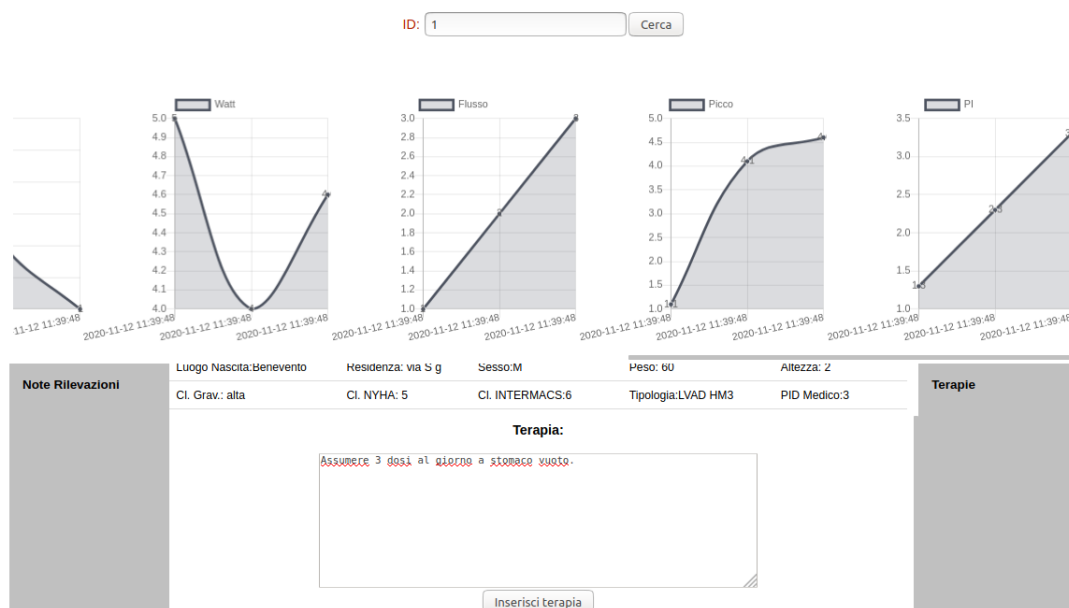


L'interfaccia mostra in alto una serie di grafici che descrivono l'andamento dei valori fisiologici del paziente negli ultimi 30 giorni.

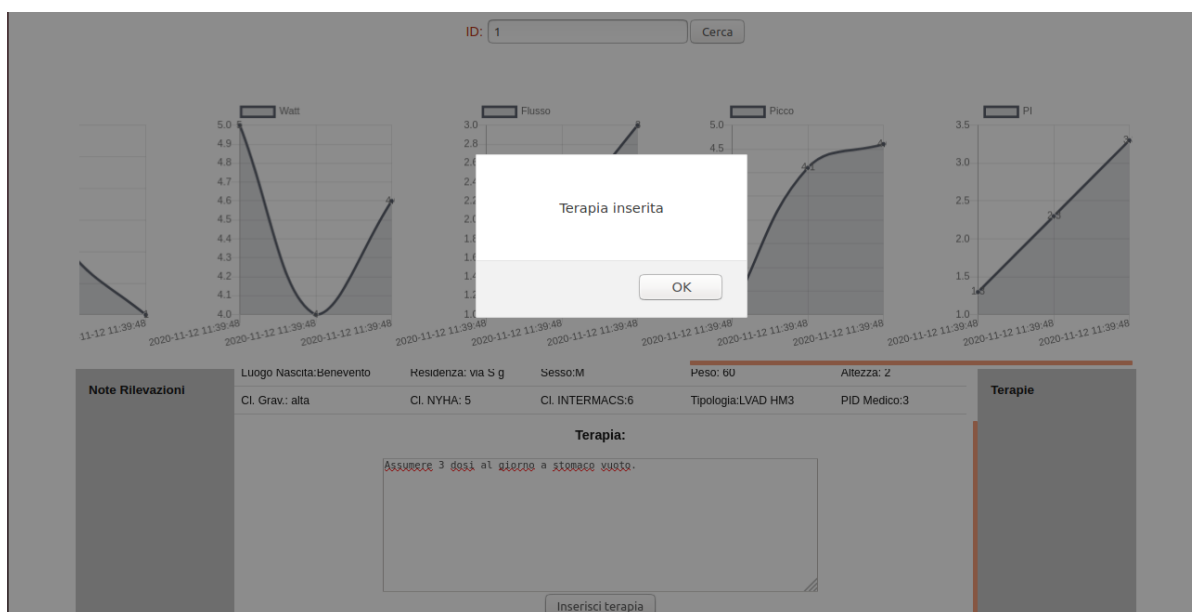




Il medico, dopo aver stabilito la terapia, scrive la prescrizione nella textbox in basso.



Un>alert notifica il riuscito inserimento della terapia.



La terapia, se inserita correttamente, è visibile in basso nella colonna di destra.



Ora il medico seleziona la funzione "Esportazione dati". Seleziona il paziente con ID 1 ed esporta in un file csv tutte le terapie riferite paziente 1 salvate nel sistema.

Gestione Terapie Gestione Rilevazioni **Esportazione dati**

Esportazione Rilevazioni in file CSV

Periodo: da: a:

Paziente:

Se non viene selezionato alcun paziente, estre i dati di tutti i pazienti

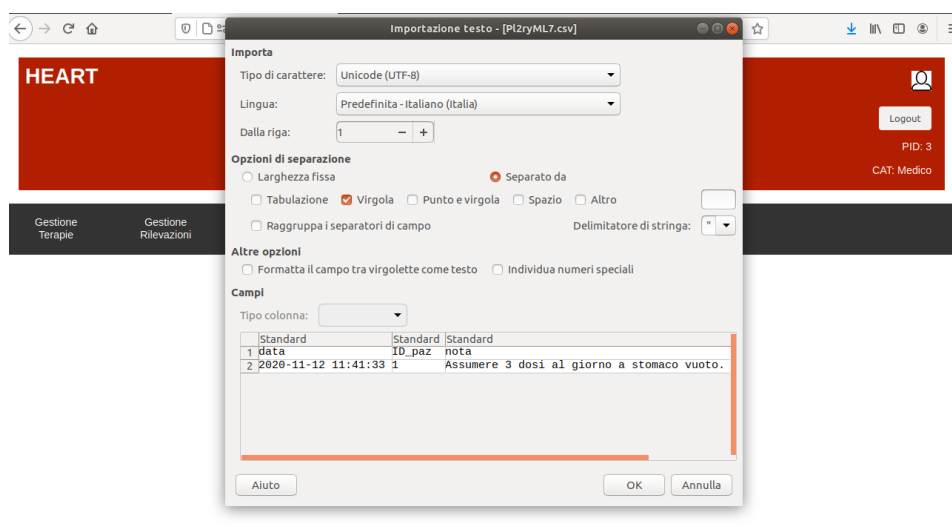
Esportazione terapie in file CSV

Paziente:

Se non viene selezionato alcun paziente, estre i dati di tutti i pazienti



Visualizza le terapie con LibreOffice Calc.



Per terminare l'utilizzatore effettua il logout.

Capitolo 5

Conclusioni e sviluppi futuri

Questa tesi è nata con lo scopo di presentare un Web Information System destinato allo staff medico che risponde all'esigenza del sistema sanitario di un maggiore supporto da parte delle ICT nella gestione delle terapie dei pazienti con scompenso cardiaco e portatori di LVAD.

La pandemia COVID-19 ha evidenziato maggiormente la necessità di sistemi come quello sviluppato con questo progetto, basti pensare alle modalità attuate per il monitoraggio dei pazienti infetti effettuato tramite verifica telefonica da parte dei centri di monitoraggio delle strutture sanitarie dei valori fisiologici misurati direttamente dai pazienti in quarantena.

Il sistema Heart dimostra come le moderne tecnologie ICT consentano in modo rapido l'implementazione di strumenti informatici in grado di automatizzare i processi aziendali, consentendo al contempo di raggiungere un'elevata qualità del servizio e un'elevato livello di ottimizzazione delle risorse a disposizione di qualsiasi azienda.

Riassumendo, si è discusso circa lo sviluppo della telemedicina sia in termini di sistemi back-end che di applicazioni mobili mostrando vari utilizzi delle tecnologie informatiche per il miglioramento della salute e del sistema sanitario. Si è descritta la risonanza che ha avuto negli ultimi anni in ambito socio-politico la Smart Health e i provvedimenti adottati da diversi paesi del mondo per promuovere la sua crescita. Si sono analizzate gli obiettivi posti, le specifiche richieste e le funzionalità da realizzare sviluppando, poi, ogni fase della progettazione della base dei dati passando dalla progettazione concettuale,

con i suoi artefatti, alla progettazione logica. Sono state individuate le funzionalità da realizzare tramite la descrizione dei possibili casi d'uso. Si è scelta l'architettura del software realizzato dopo una breve descrizione delle diverse tecnologie impiegate.

Nell'elaborato sono riportate le illustrazioni delle interfacce utente realizzate per le varie categorie di utilizzatori previsti dal sistema seguite, poi, dalla descrizione di due user experience costituite dall'infermiere che visualizza le rilevazioni di un paziente, validandone una e dal medico che inserisce la terapia destinata ad un paziente ed estrae i dati relativi ad esso in un file csv.

Per terminare sono stati ipotizzati alcuni possibili sviluppi futuri di tale sistema informatico volti al suo miglioramento in termini di funzionalità e performance.

In conclusione, questo lavoro ha consentito di realizzare un'applicazione web la cui validazione, a causa della emergenza COVID-19, sarà realizzata quanto prima sottoponendolo all'uso da parte dei medici.

5.1 Sviluppi futuri

Le possibili funzionalità da implementare nel prossimo futuro sono:

- Il recepimento delle terapie in modalità automatica da parte del paziente. Pertanto, tramite App Mobile, il paziente potrebbe ricevere in real-time la nuova terapia inserita dal medico a seguito dell'analisi del proprio stato di salute.
- L'applicazione di algoritmi che consentono la classificazione dei pazienti secondo categorie di gravità. La relazione "Paziente" potrebbe avere un attributo "classificazione_gravità". A tale classificazione potrebbe seguire un'assegnazione di priorità a ciascun paziente in modo da individuare quello che richiede un monitoring più accurato delle rilevazioni ricevute tramite App Mobile.
- Geolocalizzazione del paziente da parte dei sanitari nel caso si verifichi una situazione di emergenza. L'App Mobile potrebbe disporre di un sistema di "Emergenza" che invia in automatico la geolocalizzazione del paziente al momento dell'invio del segnale di allarme. Il sistema di back-end potrebbe ricevere tale notifica che assume una posizione di primo piano sulla schermata utente in modo tale che l'équipe

medica potrà intervenire il prima possibile. In questo modo il team sanitario avrà la possibilità di raggiungere il paziente nel luogo in cui si trova.

Appendice

Script SQL

```
DROP DATABASE IF EXISTS heart;
CREATE DATABASE heart;
DROP USER IF EXISTS 'user';
USE heart;
```

```
DROP DATABASE IF EXISTS heart;
CREATE DATABASE heart;
DROP USER IF EXISTS 'user';
USE heart;
```

#Creazione tabelle con relativi vincoli di integrità referenziale
e chiave primaria

```
CREATE TABLE Utilizzatore(
  PID varchar(200) primary key not null,
  Email varchar(200)not null,
  Password varchar(200) not null,
  nome varchar(200) not null,
  cognome varchar(200) not null,
  num_tel varchar(200) not null,
  Categoria enum("Medico","Infermiere","admin") not null
)engine=innodb;
```

```
CREATE TABLE Paziente(  
  ID int auto_increment primary key,  
  nome varchar(200) not null,  
  cognome varchar(200) not null,  
  num_tel varchar(200) not null,  
  luogo_nascita varchar(200) not null,  
  data_nascita date not null,  
  residenza varchar(200) not null,  
  sesso enum("M","F") not null,  
  peso int,  
  altezza int,  
  class_gravita enum ("alta","media","bassa"),  
  class_NYHA int not null,  
  class_INTERMACS int not null,  
  tipologia enum("LVAD HM3","LVAD HVAD","NO LVAD"),  
  PID_medico varchar(200),  
  foreign key (PID_medico) REFERENCES Utilizzatore(PID) on delete set null  
)engine=innodb;
```

```
CREATE TABLE Terapia(  
  data datetime not null,  
  ID_paz int,  
  nota varchar(1000) not null,  
  PID_medico varchar(200),  
  primary key (ID_paz, data),  
  foreign key (PID_medico) REFERENCES Utilizzatore(PID) on delete set null,  
  foreign key (ID_paz) references Paziente(ID)  
)engine=innodb;
```

```
CREATE TABLE Rilevazione(  

```

```
ID int auto_increment primary key,  
data datetime not null,  
ID_paz int,  
peso int default 0,  
frequenza int default 0,  
pressione varchar(200) default "",  
passi int default 0,  
nota varchar(1000) default "",  
allarme_aut varchar(200) default "",  
foreign key (ID_paz) REFERENCES Paziente(ID) on delete set null  
)engine=innodb;
```

```
CREATE TABLE LVAD(  
ID_rilev int primary key,  
rpm int default 0,  
watt double default 0,  
fLusso int default 0,  
picco double default 0,  
depressione double default 0,  
PI double default 0,  
allarme enum("Basso Flusso", "Watt alti", "Nessuno") default "Nessuno",  
foreign key (ID_rilev) REFERENCES Rilevazione(ID) on delete cascade  
)engine=innodb;
```

```
CREATE TABLE Validazione(  
dataR datetime not null,  
ID_pazR int not null,  
data datetime not null,  
nota varchar(1000),  
PID_u varchar(200),  
PRIMARY KEY (dataR, ID_pazR),
```

```
foreign key (PID_u) REFERENCES Utilizzatore(PID),
foreign key (ID_pazR) references Rilevazione(ID_paz) on delete cascade
foreign key (dataR) references Rilevazione(data) on delete cascade
)engine=innodb;

#Creazione trigger allarme_aut su flusso e watt pazienti LVAD
/*Allarme viene settato dopo aver confrontato il flusso e/o watt
della rilevazione inserita con la rilevazione più recente tra quelle
più vecchie di 15gg */
DELIMITER |
create trigger AllarmeFlusso
after insert on LVAD
for each row
begin
    declare paziente int;
    declare data_ril datetime;
    declare allarme_autV varchar(200);
    set paziente= (SELECT ID_paz
    from Rilevazione
    where (Rilevazione.ID=new.ID_rilev));
    set data_ril=(SELECT data from Rilevazione
    where (Rilevazione.ID=new.ID_rilev) );
    if(new.flusso <= ((SELECT flusso
    from Rilevazione R join LVAD L on R.ID=L.ID_rilev
    where (R.ID_paz=paziente) AND
    (R.data<date_add(data_ril,interval -15 DAY))
        order by R.data desc LIMIT 1)) -1.5)
    then
        update Rilevazione set allarme_aut="Flusso basso"
        where (Rilevazione.ID=new.ID_rilev);
    end if;
```

```
set allarme_autV=(SELECT allarme_aut FROM Rilevazione
where (ID=last_insert_id()));
if(new.watt >=((SELECT watt
from Rilevazione R join LVAD L on R.ID=L.ID_rilev
where (R.ID_paz=paziente) AND (R.data<date_add(data_ril,interval -15 DAY))
order by R.data desc LIMIT 1)) +1)
then
update Rilevazione set allarme_aut =CONCAT("Watt alti - ",allarme_autV)
where (Rilevazione.ID=new.ID_rilev);
end if;
end|
DELIMITER ;

#Procedura di inserimento rilevazione
DELIMITER |
create procedure InsRil(in ID_pazIn int , in pesoIn int, in frequenzaIn int,
in pressioneIn varchar(200), in passiIn int, in notaIn varchar(1000),
in allarme_autIn varchar(200), in rpmIn int , in wattIn double,
in flussoIn int, in piccoIn double, in depressioneIn double ,
in PIIn double,in allarmeIn varchar(200))
begin
start transaction;
insert into
Rilevazione(data,ID_paz,peso,frequenza,pressione,passi,nota,allarme_aut)
VALUES (now(),ID_pazIn,pesoIn,frequenzaIn,pressioneIn,passiIn,notaIn,
allarme_autIn);
if((SELECT tipologia FROM Paziente where (ID=ID_pazIn))!="NO LVAD" )
then
insert into LVAD (ID_rilev, rpm, watt, flusso, picco, depressione,
PI, allarme)
```

```
VALUES (last_insert_id(), rpmIn, wattIn, flussoIn, piccoIn,
        depressioneIn, PIIIn, allarmeIn);
commit work;
end if;
end|
DELIMITER ;

SET @pwad=md5("admin");
insert into Utilizzatore(PID,Email>Password,nome,cognome,num_tel,categoria)
values ("0","inf@",@pwad,"","","","admin");

CREATE USER 'user'@'%' IDENTIFIED BY 'heart99';
GRANT SELECT, INSERT, UPDATE, EXECUTE ON heart.* TO 'user'@'%' ;
FLUSH PRIVILEGES;
```

Bibliografia

- [1] <http://www.salute.gov.it/portale/donna/dettaglioContenutiDonna.jsp?area=Salute%20donnaid=4490lingua=italianomenu=patologie>
- [2] <http://www.associazioneaisc.org/news/video/487-5-ottobre-2020-6-convegno-nazionale-annuale-aisc-video-integrale-evento.html>;
<http://www.associazioneaisc.org/news/presentazioni.html>
- [3] Andrea Lombardo. *Tesi: PROGETTAZIONE E IMPLEMENTAZIONE DI UN'APPLICAZIONE MOBILE PER IL MONITORAGGIO DI PAZIENTI CON SCOMPENSO CARDIACO*. Università di Bologna ALMA MATER-SCUOLA DI SCIENZE-Corso di Laurea in Informatica per il Management, III Sessione Anno Accademico 2019/2020
- [4] European Commission March 2020. *Industrial Policy Business Innovation Observatory Trend: Internet of Things Case study title: Smart health* .
https://ec.europa.eu/growth/content/smart-health_en
- [5] *Bundesministerium für Gesundheit (Federal Ministry of Health), "Ärzte sollen Apps verschreiben können," accessed November 14, 2019*
- [6] HIPAAJournal. *HIPAA History*. <https://www.hipaajournal.com/hipaa-history/>
- [7] HEALTHCARE | Staff Reporter, Singapore. *Asia's healthcare industry turns to big data to streamline services*. Published: 05 Mar 20,
<https://healthcareasiamagazine.com/healthcare/news/asias-healthcare-industry-turns-big-data-streamline-services>

- [8] DIGITAL HEALTH. *Israele: digitale è salute*. 12 Settembre 2020, <https://www.onhealth.it/digital-health/israele-digitale-e-salute>
- [9] DELOITTE Insights. *2020 global health care outlook*. <https://documents.deloitte.com/insights/2020globalhealthcareoutlook>
- [10] Cabinet Secretariat, Office of Healthcare Policy, “*Next-generation Medical ICT Infrastructure Council*” accessed November 14, 2019
- [11] International Journal of Technology Assessment in Health Care. *Vis C, Bührmann L, Riper H, Ossebaard HC (2020). Health technology assessment frameworks for eHealth: A systematic review*. International Journal of Technology Assessment in Health Care 36,204–216. <https://doi.org/10.1017/S026646232000015X>
- [12] Shaping Europe’s digital future. *Policy mHealth*. Last update: 9 May 2017 Team responsible DG CONNECT. <https://ec.europa.eu/digital-single-market/en/mhealth>
- [13] *Mobile Health (m-health) System in the context of IoT*. Published in 2016 4th International Conference on Future Internet of Things and Cloud Workshops file:///C:/Users/amer/AppData/Local/Temp/2016-IEEE-FiCloudWConference.pdf
- [14] IEEE Access. *Design and Implementation of Hypothermia Symptoms Early Detection With Smart Jacket Based on Wireless Body Area Network*. This work was supported by the Ministry of Research and Technology of Indonesia, through scheme Higher Education Excellent Applied Research 2020. Received July 21, 2020, accepted August 13, 2020, date of publication August 24, 2020, date of current version September 3, 2020 <https://ieeexplore.ieee.org/document/9174736>
- [15] IEEE Access. *Based Multiple Heterogeneous Wearable Sensors: A Smart Real-Time Health Monitoring Structured for Hospitals Distributor*. Date of Publication: 11 February 2019 Date of <https://ieeexplore.ieee.org/document/8638775>
- [16] IEEE Access. *Enhanced Deep Learning Assisted Convolutional Neural Network for Heart Disease Prediction on the Internet of Medical Things Platform*. Date of Publication: 23 September 2020 <https://ieeexplore.ieee.org/abstract/document/9204704>

- [17] *"Sviluppare applicazioni con PHP e MySQL. Guida per imparare la programmazione web lato server"* febbraio 2019 Tom Butler ,Kevin Yank Apogeo-Idee Editoriali Feltrinelli
- [18] *Basi di dati* V edizione Paolo Atzeni, Stefano Ceri,Piero Fraternali,Stefano Paraboschi, Riccardo Torlone Copyright 2018 McGraw-Hill Education(Italy)
- [19] *La città intelligente cresce con Smart Healthcare e Smart Education* Il Sole 24ore 27 ottobre 2020 URL: https://www.ilsole24ore.com/art/la-citta-intelligente-cresce-smart-healthcare-e-smart-education-ADw4Nax?refresh_ce=1/
- [20] *Introduction to Angular concepts* URL: <https://angular.io/guide/architecture/>
- [21] *A mobile application of breast cancer e-support program versus routine Care in the treatment of Chinese women with breast cancer undergoing chemotherapy: study protocol for a randomized controlled trial* Jiemin Zhu, Lyn Ebert,Xiangyu Liu ,Sally Wai-Chi Chan In: BMC Cancer 2017 Apr 26;17(1):291 <https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5406970/>
- [22] *A Web-based system for family health record* Stefano Bonacina, Sara Marceglia, Michele Bertoldi, Francesco Pincioli Annu Int Conf IEEE Eng Med Biol Soc. 2007 <https://pubmed.ncbi.nlm.nih.gov/18002789/>
- [23] *Angular.io* <https://angular.io/>
- [24] *PHP: Hypertext Preprocessor* <https://www.php.net/>
- [25] *MySQL* <https://www.mysql.com/it/>
- [26] *Apache website* <http://www.apache.org/>
- [27] *Postman* <https://www.postman.com/>
- [28] *MySQL Workbench* <https://www.mysql.com/products/workbench/>

Ringraziamenti

Ringrazio il professore Marco Di Felice per avermi dato l'opportunità di partecipare alla realizzazione del lavoro che mi ha permesso di mettere a disposizione dell'équipe medica del Policlinico Sant'Orsola-Malpighi di Bologna le conoscenze acquisite durante il triennio per agevolare il monitoraggio dello stato di salute dei pazienti cardiopatici.

Ringrazio Andrea Lombardo per l'amichevole collaborazione offerta durante il processo di sviluppo del presente lavoro.

Sento di dover rivolgere un sentito ringraziamento ancora una volta al professore ed a tutto l'Ateneo per aver consentito a tutti noi studenti di proseguire il percorso di studi in questo delicato periodo storico segnato dall'emergenza epidemiologica da COVID-19.

Ringrazio tutte le persone con le quali ho condiviso questi tre anni, ognuna delle quali mi ha insegnato qualcosa.

Ringrazio tutti i compagni con i quali ho collaborato ai svariati progetti di studio e che mi hanno permesso di mettermi in discussione e di migliorare la gestione dei rapporti interpersonali nei team di lavoro.

Ringrazio gli amici, tutti, vicini e lontani, che non hanno mai smesso di volermi bene e di sostenermi, anche e soprattutto, nei momenti difficili.

Ringrazio la mia coinquilina se non miglior amica che mi ha sostenuto in tutti i mo-

menti di sconforto coccolandomi con preparazioni dei miei piatti preferiti.

Ringrazio i miei genitori che mi hanno sempre dato fiducia supportandomi in questi anni. Mi hanno dato l'opportunità di vivere questa esperienza che mi ha permesso di maturare e responsabilizzarmi senza farmi mancare nulla.

Ringrazio mio padre che mi ha sempre seguita e motivata a superare le mie paure ed insicurezze, oltre ad avermi permesso di vivere tutte le esperienze da me desiderate in questi anni.

Ringrazio mamma per esserci in ogni momento di sconforto e di conoscere sempre la soluzione migliore ad ogni mio malumore.

Ai miei nonni, per il sostegno silenzioso e tenace.

Ringrazio tutti coloro con cui ho passato questi anni che saranno indimenticabili e che, soprattutto in un momento storico come questo, già ricordo con nostalgia.