

**UNIVERSITÀ DEGLI STUDI DI BARI "ALDO MORO"**



**DIPARTIMENTO DI INFORMATICA**  
**Corso di laurea in**  
**Informatica**

---

**Gestione di dati ospedalieri in una  
applicazione mobile in ambito medico**

---

**Tesi in**  
**METODI PER IL RITROVAMENTO DELL'INFORMAZIONE**

**Relatore:**  
Prof. Pasquale Lops  
Dr. Marco Polignano

**Laureando:**  
Simone Gramegna

# Indice

<b>1</b>	<b>Introduzione</b>	<b>4</b>
1.1	Storia ed evoluzione dell'e-health . . . . .	5
1.2	Applicazioni e tecnologie nell'e-health . . . . .	7
1.3	Vantaggi e sfide dell'e-health . . . . .	9
1.4	Obiettivi . . . . .	12
<b>2</b>	<b>Stato dell'arte</b>	<b>13</b>
2.1	Applicazioni mobile e e-health . . . . .	13
2.2	Analisi delle applicazioni per il fitness nel Playstore . . . . .	16
2.3	Applicazioni mobile in ambito ospedaliero e farmaceutico . . . . .	24
2.4	Applicazioni mobile per il monitoraggio psicologico . . . . .	34
2.5	Tecnologie di supporto alle applicazioni e-health . . . . .	44
<b>3</b>	<b>Analisi e progettazione dell'applicazione Helena</b>	<b>51</b>
3.1	Introduzione al sistema operativo Android . . . . .	51
3.2	Architettura di Android . . . . .	52
3.3	Linguaggi ed ambiente di sviluppo Android . . . . .	53
3.4	Organizzazione applicazione Helena . . . . .	55
3.5	Progettazione applicazione Helena . . . . .	57
<b>4</b>	<b>Implementazione</b>	<b>67</b>
4.1	Bugfix . . . . .	67
4.2	Sviluppo sezione Temperatura corporea . . . . .	76
4.3	Sviluppo sezione Ospedalizzazioni . . . . .	85
4.4	Sviluppo sezione Ospedali . . . . .	101
4.5	Sviluppo sezione Farmacie . . . . .	111
4.6	Sviluppo sezione Stato Emotivo . . . . .	119
<b>5</b>	<b>Sperimentazione</b>	<b>126</b>
5.1	Protocollo sperimentale . . . . .	126
5.2	Analisi dei dati raccolti . . . . .	130
<b>6</b>	<b>Conclusioni e sviluppi futuri</b>	<b>139</b>

*Per aspera ad astra*

## 1 Introduzione

L'informatica e le tecnologie ICT (Information Communication Technology) hanno assunto negli ultimi decenni un ruolo importante in tutti gli aspetti della nostra vita quotidiana, dietro le quinte della maggior parte dei prodotti e servizi di cui usufruiamo c'è sicuramente almeno un sistema informatico più o meno complesso. Dai servizi bancari, al trasporto pubblico e privato, all'intrattenimento l'ICT è onnipresente anche un aspetto prettamente umano ovvero la salute non è stato escluso da questa rivoluzione. Possiamo riferirci alle applicazioni delle tecnologie ICT nell'ambito della salute umana con il termine **e-health**, termine che ha un significato molto ampio che denota un ambito nato recentemente. Una definizione formale di e-health è stata fornita, in uno studio pubblicato nel 2001 sul Journal of Medical Internet Research, da Gunther Eysenbach secondo il quale:

*L'e-health è un campo emergente nell'intersezione tra informatica medica, sanità pubblica e business, che si riferisce ai servizi e alle informazioni sanitarie fornite o migliorate attraverso Internet e le tecnologie correlate. In senso più ampio, il termine caratterizza non solo uno sviluppo tecnico, ma anche uno stato d'animo, un modo di pensare, un'attitudine e un impegno a pensare in rete e in modo globale, per migliorare l'assistenza sanitaria a livello locale, regionale e mondiale utilizzando le tecnologie dell'informazione e della comunicazione.* [E<sup>+01</sup>]

La definizione è molto ampia ed è applicabile ad un settore dinamico come l'informatica che è in continua evoluzione, nello studio Eysenbach definisce 10 caratteristiche fondamentali dell'e-health:

1. *Efficienza* - Una delle promesse dell'e-health è quella di aumentare l'efficienza dell'assistenza sanitaria, riducendo così i costi. Un possibile modo per ridurre i costi sarebbe quello di evitare la duplicazione o l'inutilità di interventi diagnostici o terapeutici, attraverso il miglioramento delle possibilità di comunicazione tra le strutture sanitarie e il coinvolgimento dei pazienti.
2. *Miglioramento la qualità delle cure* - L'aumento dell'efficienza comporta non solo la riduzione dei costi, ma anche il miglioramento della qualità. L'e-health può migliorare la qualità dell'assistenza sanitaria, ad esempio consentendo il confronto tra diversi fornitori, coinvolgendo i consumatori come ulteriore potere di garanzia della qualità e indirizzando i flussi di pazienti verso i fornitori di migliore qualità.
3. *Basarsi sulle evidenze* - Gli interventi di e-health dovrebbero essere basati sull'evidenza, nel senso che la loro efficacia ed efficienza non dovrebbero essere presunte, ma dimostrate da una rigorosa valutazione scientifica.
4. *Responsabilizzazione* dei consumatori e dei pazienti - rendendo accessibili ai consumatori via Internet le basi di conoscenza della medicina e le cartelle cliniche elettroniche personali, l'e-health apre nuove strade per una medicina centrata sul paziente e consente una scelta del paziente basata sulle prove.
5. *Incoraggiamento* di un nuovo rapporto tra paziente e professionista della salute, verso una vera e propria partnership, in cui le decisioni vengono prese in modo condiviso.
6. *Educazione* dei medici attraverso fonti online (formazione medica continua) e dei consumatori (educazione sanitaria, informazioni preventive su misura per i consumatori)
7. *Consentire* lo scambio di informazioni e la comunicazione in modo standardizzato tra le strutture sanitarie.

8. *Estendere la portata dell'assistenza sanitaria oltre i suoi confini convenzionali.* L'e-health consente ai consumatori di ottenere facilmente servizi sanitari online da fornitori globali. Questi servizi possono andare dalla semplice consulenza a interventi più complessi o a prodotti come i farmaci.
9. *Etica* - L'e-health comporta nuove forme di interazione paziente-medico e pone nuove sfide e minacce a questioni etiche come la pratica professionale online, il consenso informato, la privacy e le questioni di equità.
10. *Equità* - Per rendere l'assistenza sanitaria più equa è una delle promesse dell'e-health, ma allo stesso tempo c'è una notevole minaccia che l'e-health possa approfondire il divario tra "chi ha" e "chi non ha". Le persone che non dispongono di denaro, competenze e accesso a computer e reti non possono utilizzare i computer in modo efficace. Di conseguenza, queste popolazioni di pazienti (che in realtà trarrebbero i maggiori benefici dall'informazione sanitaria) sono quelle che hanno meno probabilità di beneficiare dei progressi della tecnologia informatica, a meno che le misure politiche non garantiscano un accesso equo per tutti. Il divario digitale attualmente esiste tra popolazioni rurali e urbane, tra ricchi e poveri, tra giovani e anziani, tra uomini e donne e tra malattie trascurate/rare e comuni.

La definizione e i punti dati da Eysenbach stabiliscono un riferimento formale alle applicazioni dell'ICT nella salute umana avvenute fino a quel momento, le prime applicazioni risalgono agli anni '50 del novecento.

## 1.1 Storia ed evoluzione dell'e-health

Un termine antecedente a "e-health" è il termine **telemedicina**, per telemedicina intendiamo in generale lo scambio a distanza di informazioni mediche (relative alla salute) attraverso i mezzi di comunicazione esistenti prima il telegrafo, poi il telefono ed infine internet e le moderne reti 5G. I primissimi scambi di informazioni di questo tipo avvengono a partire dal 1860, durante la guerra di secessione americana, il telegrafo fu usato per trasmettere messaggi sui soldati feriti alle squadre mediche in attesa. L'idea di trasmettere dati clinici e diagnostici è stata esplorata all'inizio del XX secolo. Nel 1905, William Einthoven sviluppò il primo elettrocardiogramma e trasmise con successo questo "TeleECG" e i suoni cardiaci tra il suo laboratorio e un centro accademico. Il passo successivo fu l'utilizzo del telefono (inventato intorno al 1870) e della radio (inventata intorno al 1900). L'uso della radio per i consulti medici è stato esplorato in Europa, Australia e Nord America, soprattutto per fornire assistenza medica agli equipaggi delle navi. Il Centro Internazionale Radio Medico (CIRM) in Italia e l'Aerial Medical Service in Australia sono esempi notevoli di consulti medici via radio. Hugo Gernsback, autore e scrittore di fantascienza, aveva previsto che la radio potesse essere utilizzata per fornire assistenza medica a distanza: "Il medico via radio". Negli anni '50 la televisione è stata utilizzata per l'assistenza clinica, l'educazione medica e la dimostrazione di procedure chirurgiche. Un esempio famoso fu l'operazione "Blue baby" eseguita al Johns Hopkins Hospital dal dottor Alfred Blalock e trasmessa tramite televisione a circuito chiuso a centinaia di chirurghi. Tra il 1970 e il 2000 avviene la rivoluzione, si passa da tecnologie analogiche al digitale, vi è la diffusione dei computer che diventano sempre più piccoli e sempre più potenti fino ad arrivare nei primi anni '90 con l'invenzione di Internet e del World Wide Web. Nel 1974 viene coniato da Bird il termine **telemedicina**, in quell'anno Murphy e Bird hanno pubblicato la loro esperienza di "Telediagnosi" in 1000 incontri con pazienti, una delle più grandi serie di pazienti dell'epoca. Il sistema prevedeva il collegamento tra il pronto soccorso del Massachusetts General Hospital e la stazione medica dell'aeroporto internazionale Logan di East Boston. Utilizzando circuiti televisivi

e audiovisivi, dimostrarono che era possibile esaminare in tempo reale i pazienti in luoghi distanti. Nel 1977 Grundy ha riportato il primo esempio di telemedicina sincrona per fornire accesso alle cure critiche a una popolazione poco servita. La rete di telemedicina prevedeva un collegamento tra l'University Hospital (UH) di Cleveland della Case Western Reserve University e il Forest City Hospital, un piccolo ospedale privo di specialisti in terapia intensiva. L'accesso agli specialisti di terapia intensiva tramite la telemedicina è stato fornito sia agli adulti che ai bambini. Il sistema è stato utilizzato anche per fornire visite al nido neonatale del Forest City Hospital da parte dei neonatologi dell'University Hospital. In questo studio Groundy trasse 7 importanti conclusioni riguardo la telemedicina:

1. Le consulenze periodiche in terapia intensiva possono essere fornite utilizzando un collegamento audiovisivo
2. La tecnologia attuale (del 1977) è adeguata ma costosa
3. Le consultazioni di telemedicina possono essere rese accettabili per utenti e fornitori
4. La telemedicina può essere una preziosa risorsa educativa
5. La telemedicina può influenzare il processo e probabilmente l'esito della cura del paziente
6. Per le consultazioni il collegamento audiovisivo è superiore al telefono
7. La telemedicina può fungere da importante collegamento tra un piccolo ospedale e un grande centro medico, influenzando favorevolmente la qualità dell'assistenza nell'unità di cure critiche del piccolo ospedale

La telemedicina ha ricevuto un ulteriore impulso dai progressi della tecnologia spaziale e dalla sua applicazione per promuovere l'accesso all'assistenza sanitaria nelle aree rurali. Nel 1975, un progetto chiamato STARPAHC (Space Technology Applied to Rural Papago Advanced Health Care) è stato sviluppato dalla National Aeronautics and Space Administration (NASA) per fornire l'accesso all'assistenza sanitaria nelle riserve indiane. Il sistema prevedeva un collegamento bidirezionale tramite radio televisione e telemetria a distanza tra un ospedale del Servizio Sanitario Indiano e un'unità sanitaria mobile nella riserva indiana con personale infermieristico e medico avanzato. Molti operatori ritengono che il programma abbia aumentato l'accesso all'assistenza sanitaria per la popolazione della riserva indiana, ma sono state riscontrate anche delle difficoltà nell'uso della tecnologia e nell'aumento del tempo dedicato alle consultazioni di telemedicina. Un passo da gigante nella connettività è stato fatto con lo sviluppo di ARPANET (Advanced Research Projects Agency Network), una rete creata per collegare vari computer mainframe nelle strutture del Dipartimento della Difesa in tutti gli Stati Uniti. Questa tecnologia si è poi evoluta nella moderna Internet alla fine degli anni '90 ed è stata resa accessibile al grande pubblico attraverso i fornitori di servizi Internet. Internet che ha rivoluzionato la telemedicina nelle due decadi successive tra il 2000 e il 2020, questi anni sono stati segnati da una rapida evoluzione della tecnologia: hardware (smartphone, tablet, dispositivi di telemedicina), software (applicazioni mobili), connettività (cellulare, banda larga). Sono stati sviluppati diversi dispositivi di telemedicina e periferiche che hanno facilitato l'assistenza a distanza. La rivoluzione degli smartphone ha permesso ai consumatori di accedere all'assistenza sanitaria attraverso i loro dispositivi portatili. La telemedicina è cresciuta in modo esponenziale e si sono sviluppate aziende di telemedicina con modelli direct-to-consumer come "TelaDOC Health". Come in molti altri settori, l'assistenza medica ha iniziato a passare dalle strutture in muratura agli ospedali virtuali.[\[JS21\]](#)

A partire dai primi mesi del 2020 caratterizzata dalla pandemia da Covid-19, le tecnologie e le applicazioni e-health sono state ulteriormente potenziate al fine di garantire il distanziamento sociale, ridurre il sovraccarico degli ospedali già in affanno salvando quante più vite possibili.

## 1.2 Applicazioni e tecnologie nell'e-health

Le applicazioni dell'e-health sono cresciute con lo sviluppo dell'informatica e dell'elettronica, beneficiando dell'evoluzione delle tecnologie a tutti i livelli. Dalle connessioni mediante linea telefonica si è passati alla banda larga e negli ultimi anni al 5g, dai primi portali web degli anni '90 alle applicazioni mobile che interpretano grandi quantità di dati provenienti da dispositivi smart mediante tecniche di intelligenza artificiale. In generale, l' e-health include servizi e tecnologie quali:

- Cartelle cliniche elettroniche
- Prenotazione di trattamenti e test in modo elettronico
- Diagnosi e monitoraggio di pazienti a distanza
- mHealth: informazioni cliniche su dispositivi mobile
- Telechirurgia
- Prescrizione di farmaci e terapie in modo elettronico
- Health knowledge management
- Robotica

Di seguito descrivo brevemente queste tecnologie e la relativa applicazione effettiva a vantaggio della salute umana, soffermandomi sulle tecnologie inedite adottate durante la pandemia.

### Intelligenza artificiale, big data ed e-health

Nell' e-health l'intelligenza artificiale è applicata alla predizione ed alla diagnosi di malattie con grande accuratezza. Nel caso delle analisi istopatologiche per rilevare il carcinoma delle cellule basali ,mediante l'analisi di oltre 1400 immagini, si è arrivati ad un'accuratezza di oltre il 90% , caso analogo, per la diagnosi del cancro al seno mediante 41000 immagini di tessuti, l'accuratezza è stata del 94%.[\[NSZZA19\]](#) Avere un'accuratezza molto alta è fondamentale nel identificazione e nel trattamento di malattie che sono la causa principale di disabilità e morte, le aree di ricerca e applicazione sono:

- Patologia: rilevazione di tessuti in metastasi
- Neurologia
- Predizione del rischio di malattie cardiovascolari
- Screening di casi di diabete

Un algoritmo non potrà mai sostituire l'esperienza e gli anni di studio di un medico specializzato ma la capacità di queste tecniche di analizzare, trovare pattern ed imparare in pochissimo tempo da grandi quantità di dati fornisce allo specialista uno strumento in più per il supporto alle decisioni critiche, decisioni da cui dipendono la vita e la morte dei pazienti. Questi algoritmi necessitano

di grandi quantità di dati, i cosiddetti *BIG DATA*, dati usati non solo per l'addestramento dei modelli predittivi ma la cui analisi aiuta a comprendere meglio il funzionamento del corpo umano. Ad esempio nella bioinformatica l'analisi dei dati del genoma è alla base della comprensione dei meccanismi presenti a livello molecolare, nella neuro-informatica le immagini dei tessuti celebrali aiutano a trovare correlazioni con gli eventi clinici. Big data non solo di dati clinici ma anche di dati provenienti da motori di ricerca e social media. L'analisi di ricerche su Google o di post su Twitter, Facebook è in grado di tracciare in modo abbastanza preciso un'epidemia, tra la fine del 2019 e l'inizio del 2020 con la diffusione del Coronavirus, lo studio di 38 query effettuate su Google negli Stati Uniti suddivise in sei categorie ha fornito una correlazione tra l'aumento dei casi dell'infezione e le ricerche effettuate.

### **m-health**

Per m-health intendiamo l'uso di applicazioni per dispositivi mobile quali smartphone e tablet per la gestione, l'utilizzo di dati clinici e il miglioramento di servizi relativi alla salute. Questi dispositivi sono dotati di connessione Wi-fi e bluetooth, numerosi sensori quali accelerometri, giroscopi, GPS oltre a prestazioni che si avvicinano ad i notebook in commercio. Si stima che le applicazioni e-health per questi dispositivi siano oltre 325000 tra gli stores dei sistemi Android ed iOS, l'utilizzo di queste applicazioni fornisce un contatto più diretto tra medici e pazienti, inoltre sfruttando i sensori presenti è possibile il monitoraggio e l'assistenza in tempo reale.[\[Iye20\]](#) Le tecnologie e-health non si limitano soltanto a migliorare l'esperienza dei pazienti ma anche per formare i professionisti in ambito sanitario. Sistemi di realtà virtuale permettono l'addestramento di medici, chirurghi ed infermieri in scenari realistici.[\[MCGR03\]](#)

### **Tecnologie robotiche**

Le tecnologie robotiche sono state adottate per fornire un supporto simultaneo e diretto ai pazienti incapaci in modo più rapido e per contribuire alla sicurezza degli operatori medici e dei volontari che assistono i pazienti affetti da COVID-19. Diverse tecnologie robotiche utilizzate per l'assistenza medica (ad es. riabilitazione, assistenza e robotica medica) forniscono un supporto per la diagnosi e la cura dei pazienti. Con l'avanzamento dell'intelligenza artificiale, i robot ora lavorano più velocemente e servono i pazienti in strutture di isolamento o centri di quarantena, come quelli utilizzati in paesi come la Cina per il trattamento della COVID-19. Inoltre, questi robot sono dotati di interpreti dell'umore integrati per comprendere le espressioni facciali dei pazienti e ricevere feedback, valutare il riconoscimento vocale e fornire la somministrazione di farmaci.

### **Social network**

Le applicazioni di social network vengono utilizzate per l'auto-quartierizzazione e l'isolamento sociale. Sebbene le piattaforme di eHealth non possano sostituire l'interazione faccia a faccia, forniscono un aiuto a coloro che si sentono soli e depressi a causa dell'ordine di rimanere a casa o dell'isolamento. Queste applicazioni possono essere installate e utilizzate su Windows phone, Android, Blackberry e iOS. Possono essere installate come software desktop nei sistemi Windows e Apple MacBook, tra gli altri.

### **App di contact tracing**

La tracciabilità dei contatti è un metodo importante per i medici e le amministrazioni comunali per gestire la diffusione della COVID-19. Attualmente, in tutto il mondo si utilizzano applicazioni per la ricerca dei contatti. Ad esempio, il governo di Singapore ha rilasciato l'applicazione mobile TraceTogether, sviluppata per aiutare i funzionari sanitari a rintracciare i soggetti infetti e le persone con cui potrebbero essere stati in contatto. In Israele, è stata approvata una legge che consente

al governo di tracciare i dati dei dispositivi mobili di individui con sospetta infezione A Taiwan, le istituzioni sanitarie hanno accesso ai dati di localizzazione del telefono di individui in quarantena.

### **Applicazioni in ambito sanitario**

Le applicazioni mediche comprendono programmi che forniscono ai pazienti servizi di assistenza sanitaria sia sincroni che asincroni. Le piattaforme e-health offrono guide e risorse mediche soprattutto ai pazienti che risiedono in aree non sviluppate dove l'accesso alle cure mediche è limitato. Inoltre, le piattaforme eHealth mettono in contatto i pazienti con medici remoti durante i disastri naturali o le emergenze, quando la richiesta di servizi medici aumenta. Esempi di applicazioni mediche sono Primary Care, K Health, Teladoc e Doctor on Demand, tutti disponibili per il download su Google Play e Apple Store.

### **Applicazioni per la salute e il fitness**

Le applicazioni per la salute e il fitness sono attualmente utilizzate a causa della quarantena e dell'isolamento sociale che hanno favorito i comportamenti sedentari e la diminuzione dell'attività fisica, un problema per la popolazione che trascorre il 60% del tempo impegnata in attività sedentarie. Pertanto, l'uso di applicazioni per la salute e il fitness potrebbe ridurre l'impatto negativo dei comportamenti sedentari, proteggere la salute psicologica e contribuire a migliorare le abitudini del sonno. Software come MyFitnessPal, che fornisce un piano dietetico e un contatore di calorie, vengono utilizzati per fornire benefici al benessere.[\[Bok21\]](#)

L'utilizzo di queste tecnologie offre numerosi vantaggi ma espone sia gli utenti che il personale competente a potenziali rischi se utilizzate in maniera inappropriata.

## **1.3 Vantaggi e sfide dell'e-health**

### **Vantaggi dell'e-health**

Le tecnologie ICT hanno permesso alle persone per la prima volta di accedere liberamente ad una quantità di informazioni mai vista, di connettersi e scambiarsi informazioni rapidamente da una parte all'altra del globo, cambiando le vite di tutti noi e rendendo pressoché impossibile in molti aspetti farne a meno. L' e-health quindi si inserisce all'interno di questa realtà, le cui opportunità sono potenzialmente infinite e le cui applicazioni crescono con l'evoluzione della tecnologia, ciò che venti, trenta anni fa sembrava impossibile, adesso è possibile. I vantaggi principali offerti dall'e-health sono i seguenti:

- Disponibilità delle informazioni h24
- Migliore monitoraggio dei pazienti
- Decision-making più facile per il personale sanitario
- Sanità più accessibile ed equa
- Maggiore efficienza di ospedali e cliniche
- Incoraggiamento di abitudini più salutari

[\[comb\]](#)

I vantaggi comprendono i pazienti ed il personale sanitario: medici, infermieri e specialisti il cui lavoro è coadiuvato dai dati clinici in formato elettronico, dati la cui elaborazione mediante modelli

predittivi garantisce diagnosi e trattamenti maggiormente efficaci. La digitalizzazione dei servizi relativi alla sanità comporta un risparmio non indifferente di tempo da parte di tutti, evitando lunghe attese presso le strutture e gli studi dei medici di base riducendone l'affollamento. Quest'ultimo aspetto dopo lo scoppio della pandemia da Coronavirus avvenuto in Italia nel Marzo del 2020 è divenuto fondamentale, l'utilizzo di software già esistenti quali Zoom, Teams, Skype o di applicazioni create ad hoc ha permesso la cura a distanza dei pazienti affetti da forme lievi dell'infezione. Nella regione dell'Africa sub-sahariana in cui la mortalità al parto ed infantile sono elevate, l'uso dell'e-health e delle tecnologie mobile hanno ridotto la mortalità nella fasi del parto e nei primi mesi di vita del neonato. Nell'Uganda orientale, un sistema di comunicazione radio è stato utilizzato dalle assistenti al parto tradizionali, per collaborare con le strutture sanitarie pubbliche fornendo assistenza sanitaria alle donne incinte. Allo stesso modo, in Egitto, l'uso dei telefoni cellulari ha contribuito a ridurre il tempo di risposta per le emergenze ostetriche , mentre in Nigeria, il programma di maternità sicura, Abiye, ed il Midwifery Service Scheme sono esempi notevoli dell'uso delle TIC da parte degli operatori sanitari per ridurre la mortalità materna.[[OML15](#)]

### Sfide dell'e-health

Le sfide della digitalizzazione in ambito sanitario sono molteplici, la prima sfida e forse la più importante è la sicurezza, diventato un fronte caldo negli ultimi anni a seguito dei numerosi attacchi che hanno colpito anche ospedali e cliniche. È fondamentale garantire la confidenzialità, l'integrità e la disponibilità di informazioni e servizi (CIA triad), i sistemi devono essere sicuri ed aggiornati alle più recenti minacce alla sicurezza. Non sempre ciò accade, molti ospedali hanno grandi problemi di budget da destinare all'infrastruttura informatica, in cui sono presenti una moltitudine di sistemi operativi, software e periferiche, spesso obsoleti, aggiunti nel corso degli anni alla rete interna.[[Calb](#)] Gli attacchi informatici sono un grave problema e mettono in pericolo la vita dei pazienti, è il caso dell'attacco ransomware che ha paralizzato l'ospedale di Dusseldorf, in Germania, causando il respingimento di un'ambulanza in servizio di emergenza e la morte della paziente colpita da un aneurisma.[[Cyb](#)] Recentemente un attacco analogo avvenuto nei confronti del sistema sanitario della regione Lazio ha paralizzato per diverse settimane la campagna vaccinale anti-covid mentre in Campania una falla nel sistema di prenotazione del vaccino ha esposto i dati personali dei cittadini. Un altro fattore importante è l'interoperabilità, ovvero la capacità di scambiare dati ed informazioni facilmente e secondo dei precisi standard tra le diverse parti coinvolte, in questo caso medici, farmacie , strutture ospedaliere e i cittadini. Attualmente a livello Europeo, non esiste ancora un'integrazione tra i sistemi informatici sanitari, al contrario vi è un'enorme frammentazione tra servizi, repository di dati, formati utilizzati. A fronte di questa situazione, nel 2017, è stato adottato il nuovo European Interoperability Framework (EIF), per favorire l'interoperabilità nel settore pubblico, incoraggiando le amministrazioni a progettare e a fornire sistemi:

- Digital-by-default: servizi e dati accessibili digitalmente
- Open-by-default: permettendo il riutilizzo, la partecipazione, l'accesso e la trasparenza
- Security-by-design: per assicurare che le infrastrutture ed i sistemi siano conformi ai requisiti relativi alla protezione ed alla privacy dei dati
- Interoperability-by-design: approccio standard per la progettazione dei servizi pubblici europei

L'interoperabilità coinvolge anche la governance tra le diverse strutture, integrando i processi organizzativi e di decision-making, progettare sistemi e-health interoperabili supporta lo sviluppo e l'innovazione dell'assistenza sanitaria. [[KK19](#)]

La diffusione di tecnologie e-health necessita di nuove competenze da parte del personale sanitario che deve essere in grado di utilizzare, sfruttando al massimo le potenzialità del software e della tecnologia a disposizione. Secondo Giancarlo De Leo, segretario dell'Osservatorio Sanità Digitale dell'Associazione Italiana Digital Revolution (AIDR), un operatore sanitario deve essere in grado di utilizzare:

1. Sistemi di refertazione vocale e software di trascrizione
2. Software per l'amministrazione dei pazienti
3. Sistemi di monitoraggio dei parametri vitali dei pazienti mediante sensori

[[Leo](#)] Non solo bisogna agire sulla formazione dei medici e di tutto il personale coinvolto ma anche è necessario rendere accessibili le tecnologie e-health anche da parte di fasce della popolazione meno avvezze all'uso della tecnologia.

## 1.4 Obiettivi

Helena è una applicazione per dispositivi Android, sviluppata dagli studenti dell’Università di Bari con lo scopo di fornire uno strumento utile e di facile utilizzo per il monitoraggio dello stato di salute. L’ applicazione è in grado di acquisire dati da bilance, saturimetri, glucometri e sfigmomanometri smart mediante Bluetooth inoltre include funzionalità quali:

- Consultare l’elenco dei medici nelle vicinanze
- Dati i sintomi, fornire una diagnosi il più possibile accurata
- Chatbot
- Inserimento e gestione di terapie medicinali
- Diario alimentare, con inserimento di alimenti e ricette, calcolo del fabbisogno calorico in base ai dati inseriti dell’utente
- Gestione di documenti clinici
- Dizionario medico

Funzionalità utili a tutti gli utenti che hanno intrapreso o vogliono intraprendere un percorso di corretta alimentazione con l’obiettivo di migliorare la propria salute o nel caso di sportivi (dilettanti o professionisti) aumentare le proprie performance. L’applicazione è utile anche per i medici e gli operatori in ambito sanitario in quanto include funzioni di diagnosi di possibili malattie a partire da i sintomi dell’utente ed inserimento di terapie farmacologiche.

Il mio lavoro si è concentrato nello sviluppare nell’applicazione inedite funzionalità in ambito medico-sanitario, in particolare nell’applicazione non vi era alcuna sezione in cui un utente può inserire un ricovero in un ospedale nonostante fossero presenti, come già detto, sezioni per l’inserimento di terapie, suggerimento medici e analisi dei sintomi. Assente anche una apposita sezione in cui l’utente una volta inserita la terapia può visualizzare farmacie e parafarmacie in cui può acquistare i medicinali e i dispositivi medici necessari. In aggiunta è stata anche ampliata la sezione monitoraggi con due distinte sezioni: la prima per l’inserimento e la visualizzazione della temperatura mentre la seconda per l’inserimento e la visualizzazione dello stato emotivo dell’utente. In dettaglio le sezioni sviluppate sono le seguenti:

- Sezione **Ospedalizzazioni**
- Sezione **Ospedali**
- Sezione **Farmacie**
- Sezione **Stato emotivo**

inoltre sono stati eseguiti diversi **bugfix**, la sezione **Documenti clinici** è stata rinnovata nell’interfaccia e infine è stata completata la sezione **Temperatura corporea**.

## 2 Stato dell'arte

In questo paragrafo analizzo e discuto lo stato dell'arte esaminando le varie tipologie di applicazioni mobile nell'e-health e le tecnologie di supporto quali dispositivi wearable, intelligenza artificiale e sistemi di raccomandazione.

### 2.1 Applicazioni mobile e e-health

Le applicazioni per dispositivi mobile nascono molto prima dell'avvento di moderni smartphone, nel 1997 il Nokia 6610 introduce una versione pre-installata del popolare gioco *Snake*, gioco che potremmo considerare la prima applicazione mobile in assoluto.[\[coma\]](#) Negli anni successivi non c'è un'evoluzione sostanziale, i telefoni cellulari hanno ancora display molto piccoli, capacità hardware limitate ed una ridotta connettività ma la svolta avviene nel 2007 in cui Apple presenta al mondo l'iPhone, un dispositivo che per quanto adesso possa sembrare semplice, rivoluzionò l'intero mercato spingendo la concorrenza allo sviluppo di dispositivi analoghi, nello stesso anno esattamente il 5 Novembre 2007 avviene la presentazione ufficiale del sistema operativo Android. Al momento il mercato dei sistemi operativi mobile è dominato da due grandi competitors: Apple e Google. Android detiene oltre il 75% del mercato ed è diffuso su i smartphone e tablet delle principali marche ovvero: Xiaomi, Samsung, Huawei inoltre presenta una versione per dispositivi wearable: WearOS ed è integrabile con i sistemi di infotainment delle automobili mediante Android Auto. L'ecosistema sviluppato negli anni da Google è diventato sempre più vasto ed ha attirato migliaia di sviluppatori ed aziende a sviluppare app, nel 2022 il numero di app disponibili nel play store di Google è di oltre due milioni e mezzo. Il sistema operativo iOS è presente solo su smartphone e tablet Apple, detiene circa il 30% del mercato globale e come il rivale presenta una variante chiamata watchOS per smartwatch e tramite CarPlay è integrabile con l'infotainment delle auto-vetture. Rispetto ad Android, il sistema operativo di Apple è più chiuso ed integrato meglio con i dispositivi, l'app store conta quasi due milioni di app. App che spaziano dall'intrattenimento, ai videogiochi, alla produttività (pensiamo alla suite di Office), ai social network ed alla salute. Più in dettaglio secondo il rapporto Digital Health Trends 2021 dell'iqvia institute, il numero di app e-health disponibile tra App Store e Play Store è di circa 350.000.

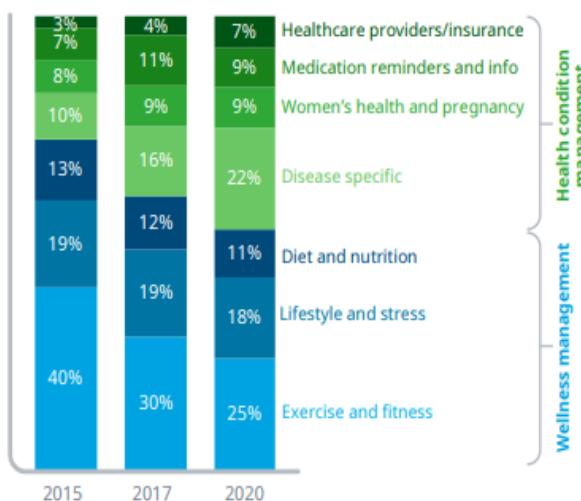


Figura 1: Percentuali delle varie tipologie di applicazioni e-health

Nello sport è importante la costanza, non bastano 2 o 3 giorni di attività per ottenere risultati, sotto questo punto di vista le app di fitness mediante diverse soluzioni e funzionalità possono stimolare e motivare l'utente a fare attività fisica con più frequenza e per un periodo di tempo più lungo, influenzando positivamente l'utente senza limitarsi ad essere un software che raccoglie ed elabora continuamente dati. Diversi studi hanno evidenziato la correlazione tra utilizzo delle app di fitness ed abitudini degli utenti, negli Stati Uniti un campione di 310 persone di età compresa tra i 18 ed i 25 anni è stata sottoposta ad un sondaggio con lo scopo di approfondire i legami e i fattori che intervengono in questo fenomeno. Ad ogni partecipante è stato somministrato il questionario IPAQ (International Physical Activity Questionnaire) da cui si ricava un valore numerico: il MET (metabolic equivalent minutes per week) che rappresenta la quantità di ossigeno che il corpo utilizza per 1Kg di peso corporeo in uno stato di riposo. Questo valore, che varia in un range compreso tra 1 e 20, da indicazioni sul consumo calorico e sull'intensità dell'attività fisica.[BHH<sup>+</sup>05] In aggiunta i partecipanti hanno compilato un sondaggio costituito da 9 domande con scala Likert che ha fornito la validazione *dell'identità di esercizio* di ogni utente, in questo modo sono stati ottenuti diversi parametri tra cui: utilizzo di app di fitness, attività fisica moderata/intensa e abitudini sedentarie. I risultati sono impressionanti, è emerso che oltre il 67% degli utenti con app di fitness compie attività fisica ad alta intensità la cui identità di esercizio è maggiore del 36% rispetto ad utenti che non utilizzano questo tipo di app. Più in generale il gruppo di partecipanti, utenti di app di fitness, è stato fisicamente più attivo e ciò mostra l'influenza positiva, a livello psicologico, di queste app di stimolare e motivare l'utente a compiere esercizio fisico.[BLS<sup>+</sup>20] Tutto questo in maniera indiretta, altri aspetti interni ad un'applicazione di questo tipo che possono stimolare l'attività sono: la presenza di una componente social con cui "competere" con i propri amici ed i membri della community oppure delle *social rewards*. È il caso dell'app "Storywell", rivolta alle famiglie con figli, basata sull'utilizzo di storie per sensibilizzare bambini ed adulti all'importanza dell'attività fisica. Ogni storia termina con una serie di domande a cui gli utenti rispondono in modo da far capire quali sono i fattori che possono influenzare positivamente l'attività fisica infine vi è la selezione di un obiettivo a sorpresa da completare. Lo studio di tre mesi condotto su 18 famiglie con almeno un figlio di età compresa tra 3 ed 8 anni ha fornito risultati incoraggianti, mediamente i caregivers hanno riportato un aumento di circa 30 minuti dell'attività fisica settimanale in famiglia per 1/2 allenamenti a settimana. I figli invece hanno riportato, in media, 6/7 allenamenti settimanali con una durata di circa 60 minuti.[SCSH<sup>+</sup>20]

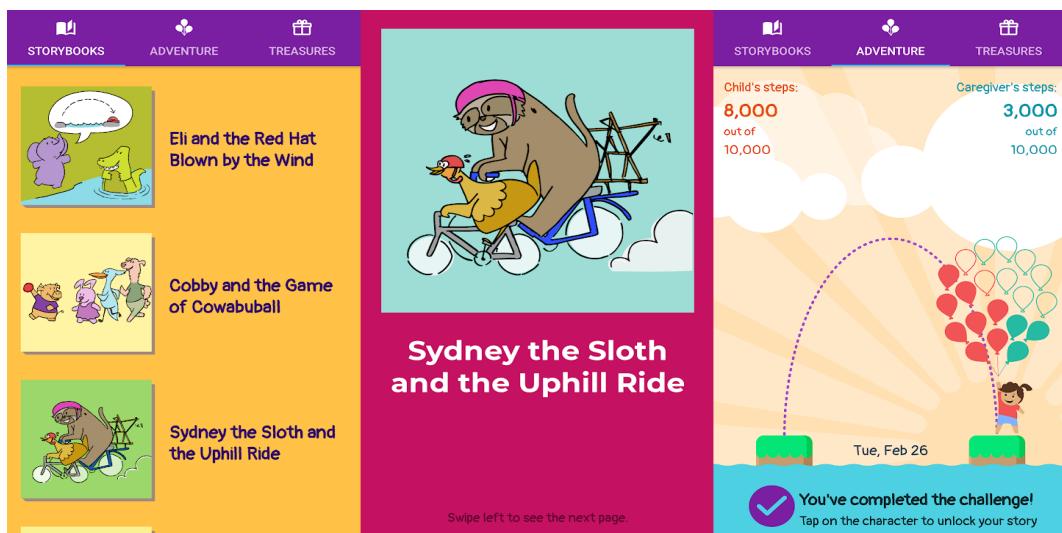


Figura 2: Storywell

La frequenza con cui un utente compie esercizio fisico è legata anche ad aspetti sociali, che avvengono anche tramite i social networks. Lo studio condotto nel 2016 in Corea del Sud ha mostrato che gli utenti più attivi nella condivisione di contenuti relativi al fitness ed alla salute sono maggiormente presi dal fitness, infatti molte app consentono agli utenti di condividere le statistiche dei propri allenamenti via social quali: Facebook, Instagram e Twitter. In particolare è stata raccolta una vasta collezione di tweet riportanti l'hashtag "myfitnesspal" postati da oltre 7000 utenti. Mediante le API di Twitter è stato possibile estrarre da questi profili features tra cui: numero di followers, diario personale di esercizi, numero di tweet con l'hashtag #myfitnesspal, profili seguiti. L'analisi di questi profili ha fornito risultati interessanti in quanto sono stati individuati dei profili seguiti fortemente discriminati ovvero profili i cui followers hanno una probabilità alta di essere utenti persistenti di applicazioni di fitness, questi ultimi hanno una maggiore propensione a postare tweet e seguire profili sul tema. Da questo studio sono state tratte le seguenti conclusioni:

- Gli utenti persistenti hanno maggiori probabilità di usare Twitter in un modo orientato alla salute.
- La condivisione di tweet relativi all'esercizio fisico è positivamente correlata con l'impegno a lungo termine
- Avere un contatto reciproco con gli utenti che condividono contenuti relativi alla salute e al fitness sui social media o che utilizzano pesantemente MyFitnessPal è positivamente correlato con la condivisione sociale persistente

I risultati indicano due linee-guida per la progettazione di applicazioni fitness:

1. Introduzione di un programma di ricompensa
2. Connessione di utenti simili mediante l'applicazione

Molte applicazioni analizzate in seguito presentano queste caratteristiche che contribuiscono al successo tra gli utenti.[[PWCL16](#)] In questo ambito non esistono soltanto applicazioni che si limitano al monitoraggio dei parametri vitali o dello stato emotivo, esistono numerose tipologie di applicazioni presenti sul mercato e negli app store che possiamo suddividere nelle seguenti categorie:

- Applicazioni cliniche per la diagnosi di malattie
- Applicazioni ospedaliere
- Applicazioni per l'assistenza da remoto
- Applicazioni per mantenere uno stile di vita sano (dieta, attività fisica, sonno..)
- Applicazioni per il tracciamento delle cure farmacologiche in corso
- Applicazioni per la ricerca di farmacie e parafarmacie
- Applicazioni per il monitoraggio dello stato psicologico

Molte di queste applicazioni integrano i dati provenienti dai sensori e le informazioni durante l'utilizzo con lo scopo non solo di ingaggiare maggiormente l'utente ma anche di fornire uno strumento valido, intelligente, personalizzabile che va quasi a sostituirsi alla figura di un personal trainer il tutto disponibile gratuitamente agli utenti. Molte di queste applicazioni includono anche funzionalità premium a pagamento destinate agli utenti più esperti. A livello tecnico queste applicazioni

utilizzano in maniera intensiva sensori integrati negli smartphone quali: accelerometri, giroscopi, sensori di prossimità e GPS il che influisce (in base al dispositivo) sulla durata della batteria, sono quasi tutte integrabili con dispositivi wearable mediante bluetooth in modo da avere dati più accurati da elaborare. Le applicazioni per Android sono scritte in Java oppure Kotlin: un nuovo linguaggio di programmazione nato nel 2011 e reso ufficiale da Google per lo sviluppo di applicazioni a partire dal 2019 il tutto integrato nell'ambiente di sviluppo *Android Studio* disponibile gratuitamente per tutte le piattaforme (Windows/Linux/macOS). Le applicazioni per iOS sono basate sui linguaggi Objective-C (basato su C e C++), linguaggio rimpiazzato dal più moderno Swift maggiormente diffuso al momento tuttavia l'ambiente di xCode risulta disponibile soltanto per utenti macOS. Nei successivi paragrafi, visti gli obiettivi della tesi, analizzerò dettagliatamente le seguenti tipologie di applicazioni:

- Applicazioni per il fitness
- Applicazioni in ambito ospedaliero
- Applicazioni per la ricerca di farmacie
- Applicazioni per il monitoraggio dello stato emotivo e psicologico

## 2.2 Analisi delle applicazioni per il fitness nel Playstore

In questo paragrafo analizzo le applicazioni per il fitness presenti nel Playstore, esaminando le caratteristiche, i pregi e i difetti di ognuna. Queste applicazioni non sono rivolte soltanto a utenti dilettanti con poca dimestichezza nel praticare attività fisica ma anche a sportivi o comunque utenti che praticano sport a livello agonistico al fine di misurare le performance durante le sessioni di allenamento e tenere traccia dei progressi.

### 1 - Google Fit



Figura 3: Google Fit

Google Fit è un'applicazione nata nel 2014, sviluppata da Google e disponibile gratuitamente per Android, WearOS e iOS. Oltre ad avere le funzionalità base delle app di fitness tracking quali: conteggio del numero di passi, delle calorie bruciate e della distanza percorsa, permette all'utente

di impostare obiettivi su base giornaliera, settimanale e mensile il cui raggiungimento incrementa il numero di Heart Points. In questo modo l'utente, magari sedentario, è stimolato a fare più movimento ed attività fisica. L'applicazione è compatibile con applicazioni esterne di terze parti, è integrata perfettamente con Google Calendar ed è compatibile con i dispositivi wearOS ma anche con Fitbit, Xiaomi, Withings, Polar.

Tuttavia l'applicazione non dispone di una sezione alimentare per il tracciamento esaustivo degli alimenti e dei nutrienti assunti, da migliorare anche il menù per il collegamento con i dispositivi wearable che risulta poco immediato da trovare all'interno dell'interfaccia. Complessivamente l'app risulta curata e ben fatta, nonostante la presenza dell'applicazione Fitbit, sotto la proprietà della compagnia di Mountain View data la recente acquisizione di Fitbit.

Punti di forza:

- Integrazione con dispositivi e app di terze parti
- Interfaccia piacevole, dark mode
- Integrazione con Google Calendar
- Semplicità di utilizzo anche per utenti non esperti
- Assenza di pubblicità

Punti di debolezza:

- Mancanza di una componente social, per condividere gli obiettivi
- Mancanza di programmi di allenamento dettagliati
- Aggiunta dei dispositivi wearable non immediata

## 2 - Fitbit

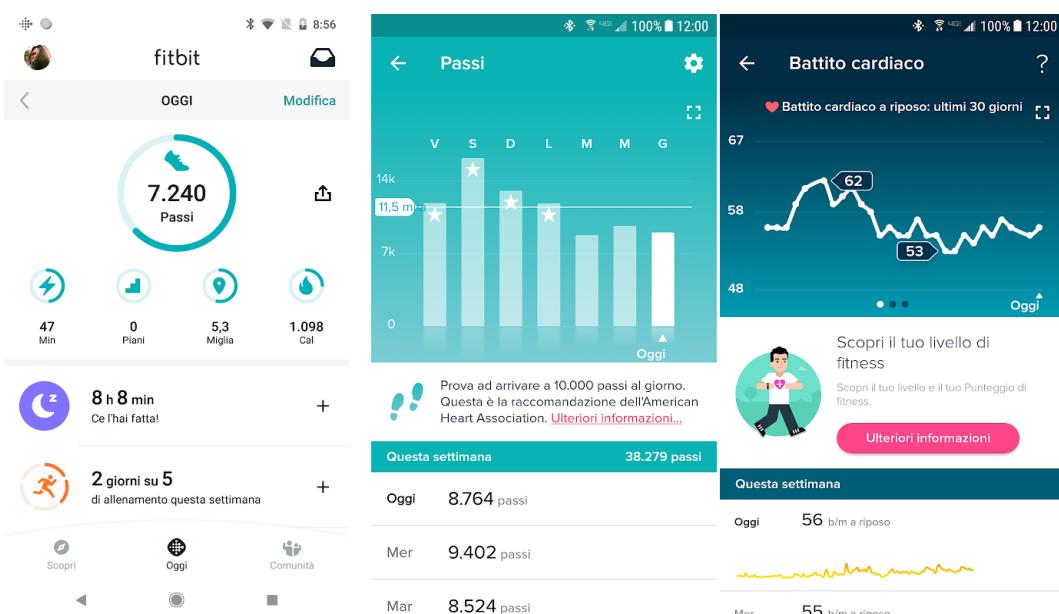


Figura 4: Fitbit

Applicazione ufficiale Fitbit, disponibile sia in versione gratuita che a pagamento, comprende numerose funzionalità essenziali per il tracciamento efficace delle attività sportive. Mediante l'integrazione di app esterne, è possibile tracciare l'attività fisica dell'utente, calcolando il fabbisogno giornaliero. Questo grazie ai dati provenienti da tutti i dispositivi Fitbit che è possibile collegare ed alla sincronizzazione con oltre 3000 applicazioni esterne. È possibile fissare obiettivi giornalieri, settimanali, mensili cercando programmi gratuiti di: allenamento, dieta, sonno e meditazione. Il design è curato, i dati sono facilmente accessibili anche da parte di utenti meno esperti e per gli utenti Android è disponibile un comodo assistente vocale.

Punti di forza:

- Design curato
- Numerose funzionalità
- Integrazione perfetta con dispositivi Fitbit
- Possibilità di sincronizzazione con moltissime app esterne

Punti di debolezza:

- Scarsa integrazione con device esterni

### 3 - MyFitnessPal

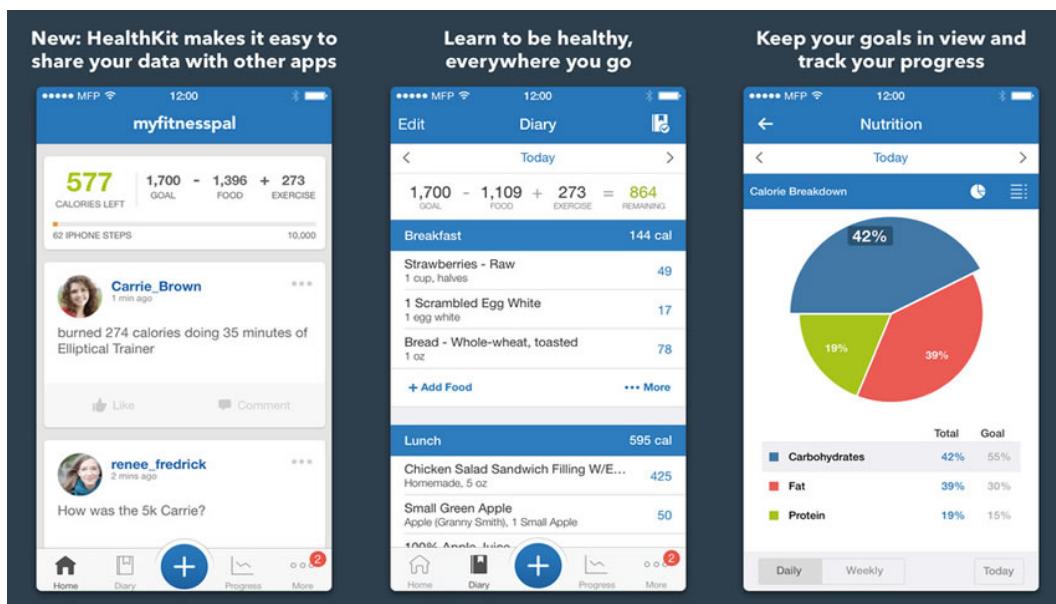


Figura 5: MyFitnessPal

L'applicazione, disponibile sia in versione gratuita che a pagamento, è stata sviluppata con l'obiettivo di fornire agli utenti un supporto per tracciare gli alimenti e le calorie bruciate dall'utente. Inoltre include funzionalità quali:

- Inserimento di alimenti e ricette mediante scansione QR, fonti esterne, database dell'applicazione
- Conteggio del numero di bicchieri di acqua

- Tracciamento del peso corporeo
- Diario dell'utente

L'applicazione si concentra più sull'aspetto alimentare che sull'aspetto sportivo dell'utente.

Punti di forza:

- Database degli alimenti molto ricco
- Compatibilità con molteplici applicazioni e dispositivi
- Eccellenti controlli di input

Punti di debolezza:

- I contenuti aggiuntivi, come le ricette e i video, non sono molto convincenti.
- Calcolo non sempre preciso delle calorie e del fabbisogno

#### **4 - Adidas Runtastic**

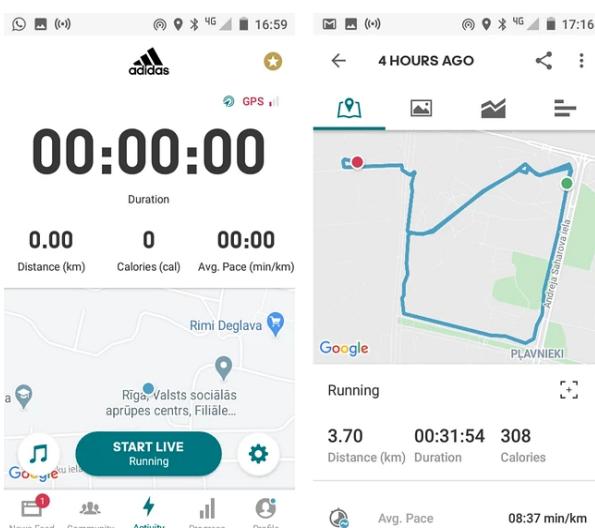


Figura 6: Runtastic

L'applicazione, sponsorizzata dal brand Adidas, è in grado di tracciare e monitorare l'attività sportiva dell'utente, impostando obiettivi, tracciando i progressi e condividerli con la community mediante l'integrazione con Facebook. L'applicazione è integrata con i dispositivi Garmin e Polar ed include un pratico assistente vocale per comunicare i dati parziali durante l'attività. Una funzionalità inedita, che non è presente nelle applicazioni concorrenti, è l'aggiunta del paio di scarpe indossato durante le attività.

Punti di forza:

- Integrazione perfetta con Facebook
- Integrazione con l'app Adidas Training
- Challenges della community

Punti di debolezza:

- Costo eccessivo della versione premium
- Interfaccia poco curata
- Assistente vocale non disponibile in italiano o in altre lingue diverse dall'inglese

## 5 - RunKeeper

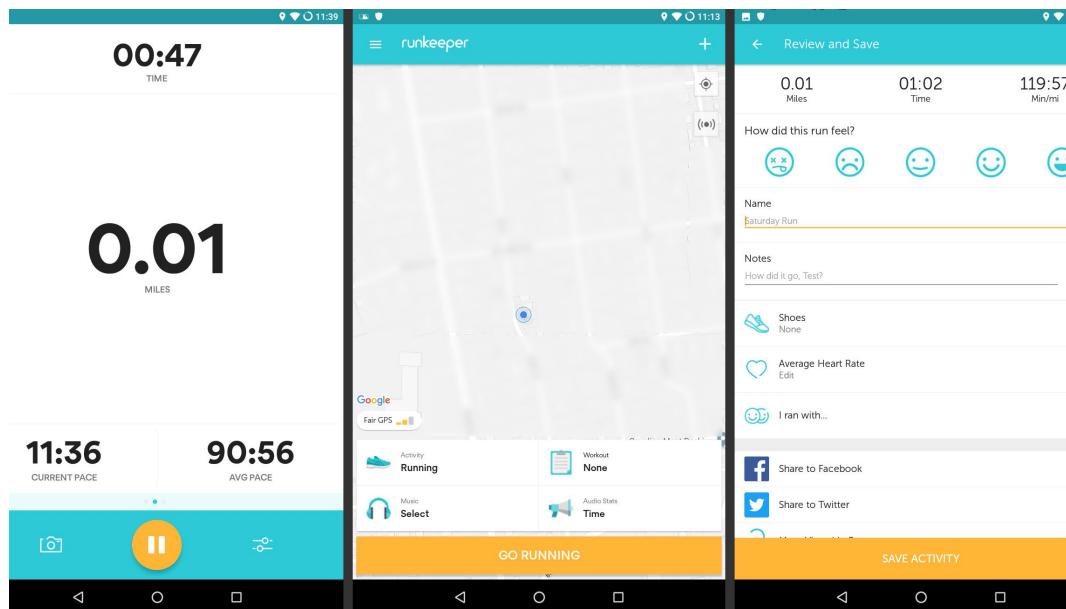


Figura 7: Runkeeper

Anche dietro questa applicazione vi è un brand sportivo: Asics e sostanzialmente offre funzionalità simili alla concorrenti ovvero impostazione di obiettivi, tracciamento e monitoraggio di diverse attività sportive e sfide atletiche mediante GPS. L'applicazione risulta piacevole da utilizzare, include la possibilità di creare un gruppo di amici, grazie all'integrazione con Facebook, per un programma di allenamento comune ed è presente anche un assistente vocale disponibile in diverse lingue. Tuttavia manca completamente l'integrazione con app e dispositivi esterni.

Punti di forza:

- Interfaccia semplice e piacevole
- Condivisione di obiettivi e sfide
- Assistente vocale

Punti di debolezza:

- Assenza di integrazione con servizi esterni
- Community piccola

## 6 - Zepp Life

L'applicazione ufficiale della Xiaomi si presenta con un'interfaccia colorata e curata sotto molti aspetti, includendo le tipiche funzionalità della app di fitness quali: monitoraggio delle attività sportive, impostazione degli obiettivi e creazione del profilo utente. Risultano piacevoli le funzionalità di aggiunta di amici mediante QR code e calcolo del PAI: un indice di attività fisiologica

personale. La perfetta integrazione con i dispositivi Xiaomi consente il monitoraggio del sonno e di parametri quali: ritmo cardiaco, livello di stress ed ossigenazione del sangue. Infatti tramite l'applicazione è possibile connettere e personalizzare le impostazioni e le configurazioni del proprio dispositivo. Tuttavia a differenza delle applicazioni precedenti non vi è un assistente vocale ed è assente l'integrazione con Facebook ed altre piattaforme social.

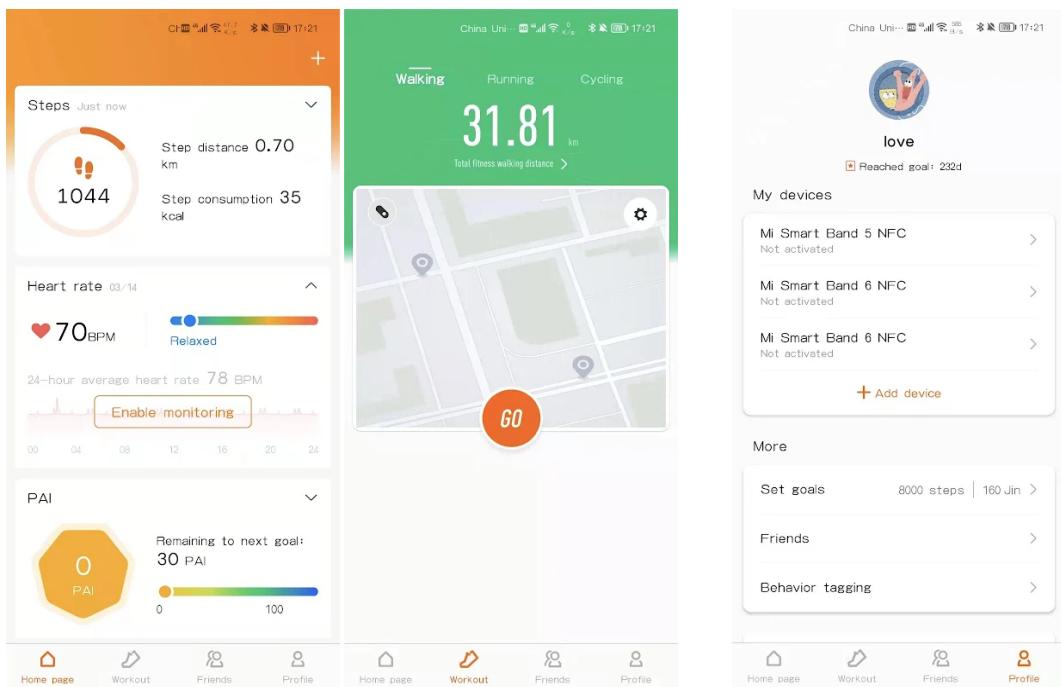


Figura 8: Zepp Life

#### Punti di forza:

- Interfaccia piacevole
- Integrazione con i dispositivi Xiaomi
- Monitoraggio di numerosi parametri

#### Punti di debolezza:

- Assenza di integrazione con dispositivi esterni
- Consumo eccessivo della batteria

#### 7 - Decathlon Coach

Applicazione creata dal colosso francese leader nella vendita di abbigliamento ed attrezzatura sportiva, è disponibile in versione gratuita ed offre funzionalità di monitoraggio di oltre 80 diversi sport. L'applicazione mediante i dati delle attività dell'utente elabora delle statistiche inoltre consente la creazione di programmi di allenamento personalizzati. Tuttavia si fa sentire la totale mancanza di integrazione di dispositivi ed app esterne inoltre non è possibile, come nelle app rivali, collegare alcun social.

#### Punti di forza:

- Numerosi sport supportati

- Interfaccia piacevole
- Statistiche accurate
- Assenza di pubblicità

Punti di debolezza:

- Assenza di integrazione con dispositivi esterni

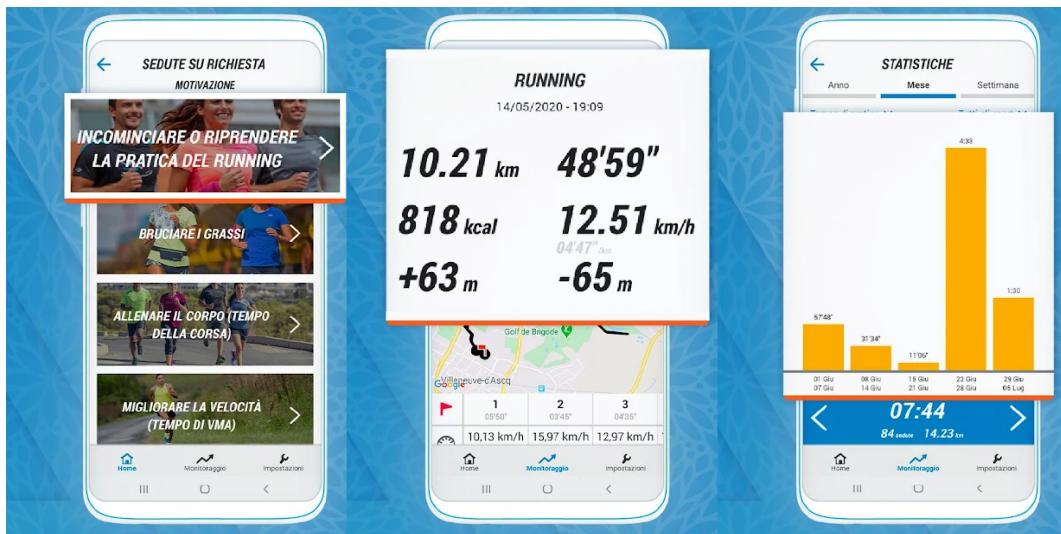


Figura 9: Decathlon Coach

## 8 - Samsung Health

L'applicazione ufficiale della Samsung per il fitness si presenta con un'interfaccia all'avanguardia compatibile con gli schermi di tutti i dispositivi, dagli smartphone ai tablet alle smart TV. Le funzionalità dell'app vanno dal classico monitoraggio delle attività sportive mediante GPS ed i vari sensori al tracciamento del sonno, del peso corporeo, del ritmo cardiaco e dei livelli di ossigenazione del sangue in base al dispositivo wearable collegato. Mediante questi dati, l'app calcola in tempo reale numerose statistiche esaustive anche per l'utente più esperto. Utile la sezione "Fitness", non sempre presente nelle applicazioni concorrenti, contenente programmi di workout coadiuvati da video esplicativi di 15 minuti di ciascun esercizio. È presente una forte componente social, infatti tramite la sezione "Together" è possibile confrontare le prestazioni ed il numero di passi con altri utilizzatori attraverso sfide aperte o sfide 1 contro 1. L'applicazione è compatibile soltanto con le app esterne quali: Strava e Technogym ma offre l'integrazione con moltissimi dispositivi smart, non solo smartwatch e fitness tracker Samsung ma anche glucometri, sensori per il monitoraggio del ritmo cardiaco, bilance e sensori per la bici.

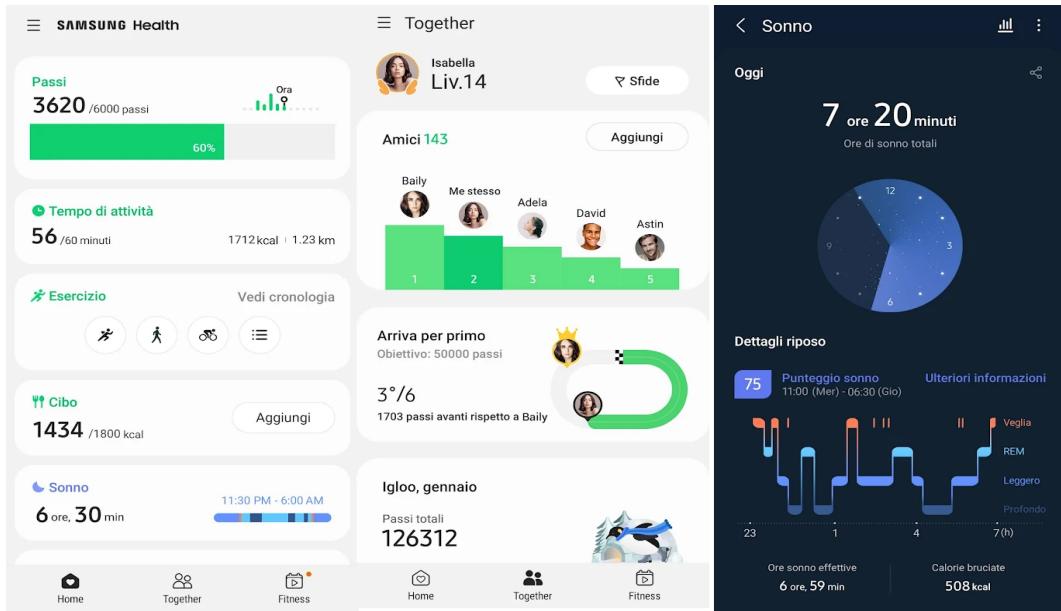


Figura 10: Samsung Health

#### Punti di forza:

- Interfaccia moderna
- Numerose funzionalità
- Integrazione con una grande varietà di dispositivi smart
- Componente social accattivante

#### Punti di debolezza:

- Scarsa integrazione con applicazioni esterne

## 2.3 Applicazioni mobile in ambito ospedaliero e farmaceutico

In questo paragrafo analizzo le applicazioni mobile in ambito farmaceutico/ospedalieri, come già accennato nell'introduzione, queste applicazioni durante il periodo pandemico sono state fondamentali per il monitoraggio a distanza dei pazienti positivi a distanza contribuendo a salvare migliaia di vite e riducendo il carico presso le strutture ospedaliere. In tal senso, nel mondo sono state sviluppate applicazioni per l'accesso rapido da parte del personale medico ma anche dei pazienti a dati clinici, per la consultazione di informazioni relative a ospedali e reparti e per la comunicazione tra medici e pazienti. Un esempio è l'applicazione moblMD sviluppata inizialmente nel 2018 da parte di un borsista in cardiologia dall'università di Chicago. In fase iniziale l'app forniva un elenco degli ospedali, aiutava a comporre i numeri di telefono degli ospedali e distribuiva le linee guida istituzionali tra gli utenti. Nel 2021 l'applicazione è stata estesa introducendo una funzionalità per la visualizzazione delle politiche relative al COVID-19 del team HICS e per aiutare i medici a cercare e comporre i numeri di telefono dei pazienti allettati nonché i numeri di telefono di ciascuna infermiera responsabile e della segretaria dell'unità, al fine di promuovere il distanziamento sociale e il giro dei pazienti a distanza.

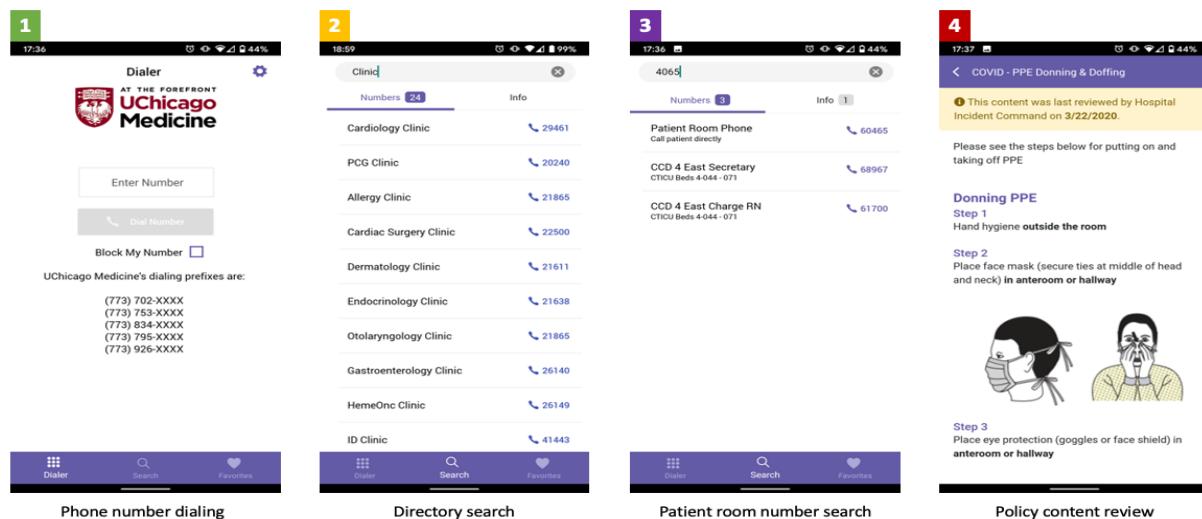


Figura 11: moblMD

L'applicazione nei quattro mesi dopo il rilascio è stata scaricata da 1004 utenti unici, con il 48% dei download avvenuto nelle prime 72 ore dopo l'annuncio, le funzionalità principali utilizzate dagli utenti sono le seguenti:

1. Composizione di numeri telefonici
2. Ricerca nelle directory
3. Ricerca del numero di camera
4. Revisione delle policy dell'ospedale

Il principale vantaggio dell'applicazione è stato centralizzare una serie di servizi, infatti prima dello scoppio della pandemia i medici utilizzavano i propri smartphone come punto di assistenza. Sebbene i numeri telefonici dei pazienti fossero precedentemente accessibili tramite il call center dell'ospedale o all'interno della cartella clinica elettronica, l'applicazione moblMD ha migliorato l'efficienza della comunicazione fornendo un'alternativa più rapida. Facilitando la comunicazione

mobile, è stata ridotta la necessità di comunicare di persona, in quanto l'assistenza remota ai pazienti è diventata la norma per ridurre al minimo l'uso dei DPI e l'esposizione dei medici.[[AWS+21](#)] I medici e gli infermieri sono i professionisti che più hanno sofferto la situazione pandemica, centinaia di loro hanno dato la vita pur di salvare vite umane, solo in Italia si stimano oltre 350 decessi [[dodmcedo](#)] tra il personale medico a causa del Coronavirus la cui diffusione negli ospedali è stata favorita in molti casi dalla scarsa organizzazione e dai pochi dispositivi di protezione individuale forniti dalle aziende sanitarie. Per arginare questo problema nel 2020, in piena pandemia, l'applicazione *HEpiTracker* ha aiutato a tracciare i casi di Covid-19 nel personale ospedaliero. L'applicazione è stata sviluppata secondo un approccio iterativo e multidisciplinare che ha incluso un team di medici, infermieri, fisici, informatici, matematici e statistici ma non sono stati coinvolti direttamente i pazienti. L'applicazione include un'autovalutazione dello stato di salute generale basata su una scala Likert ordinale da 0 a 10 ed è stata resa disponibile per i sistemi operativi Android e iOS, rispettivamente negli store di Google e Apple. Una volta che *HEpiTracker* è diventata operativa, è stato progettato uno studio pilota in condizioni reali per testare la fattibilità dell'applicazione. Il personale ospedaliero di 5 comunità autonome spagnole (Andalusia, Baleari, Catalogna, Galizia e Madrid) è stato invitato a scaricare l'applicazione sul proprio smartphone e a registrare la propria temperatura corporea giornaliera, i sintomi compatibili con il COVID-19 e il punteggio di salute generale, nonché i risultati di eventuali test PCR o sierologici. Tutto il personale degli ospedali partecipanti, ovvero medici, infermieri, tecnici, impiegati amministrativi, guardiani, addetti alle pulizie, dirigenti, personale della mensa, della sicurezza e altre professioni, sono stati invitati a partecipare, senza criteri di esclusione.

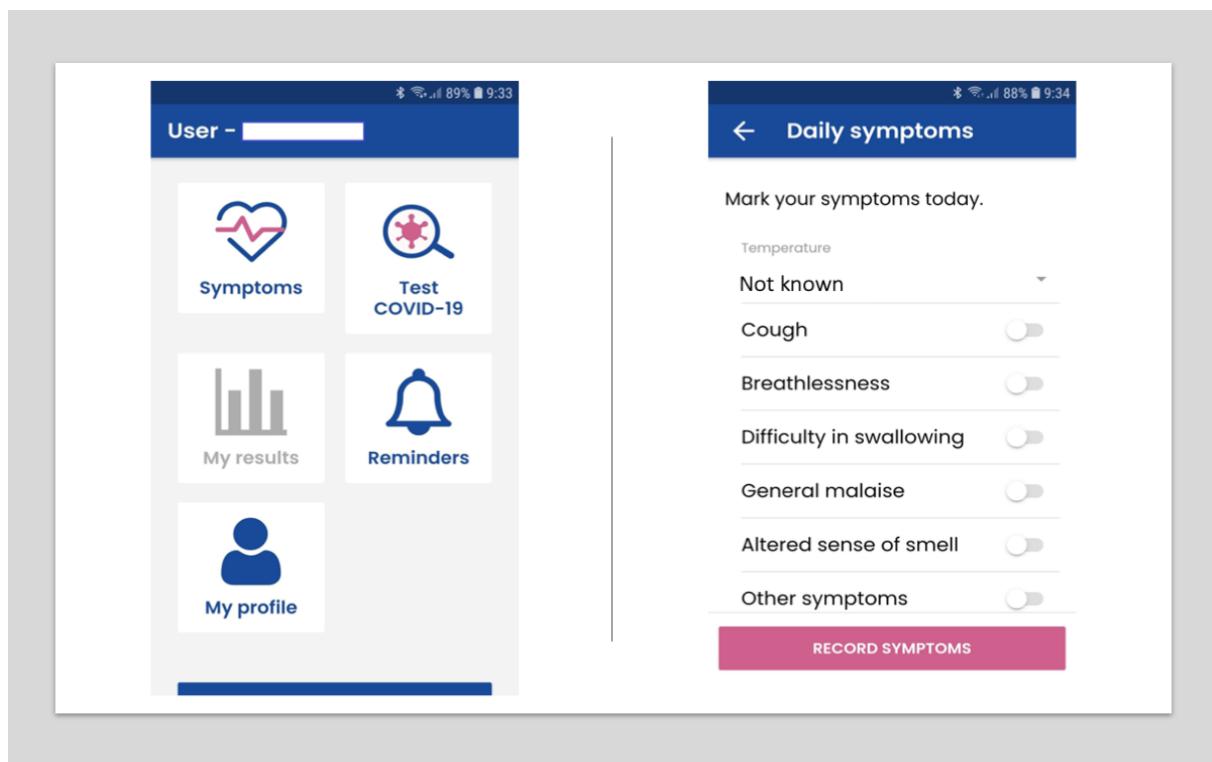


Figura 12: HEpiTracker

In totale sono stati coinvolti 477 partecipanti tra l'Aprile ed il Giugno del 2020, in media la partecipazione è stata bassa (intorno al 5%), infatti La segnalazione giornaliera dei sintomi è stata molto variabile tra i partecipanti; in generale, dal 2% al 6% della popolazione di origine in ogni ospedale ha utilizzato l'app. Sebbene si sia osservato un calo dell'adesione dopo un picco iniziale di

partecipazione in alcuni ospedali, altri siti sono stati caratterizzati da tassi di partecipazione scarsi sin dall'inizio e per tutto il periodo dello studio inoltre la maggior parte dei soggetti è femminile (67.7%) con un'età media di 45 anni. La scarsa partecipazione del personale sanitario non può smuovere i punti di forza di HepiTracker quali la flessibilità e la possibilità di modificarla rapidamente per includere nuovi aggiornamenti e informazioni, in particolare, l'applicazione è disponibile in diverse lingue (spagnolo, inglese e portoghese).[\[SFdA+20\]](#)

L'Italia è stato il primo paese occidentale colpito dalla pandemia, una situazione inedita che nessuno poteva prevedere e che è stata affrontata con coraggio e determinazione dal personale sanitario provato dai continui tagli al sistema sanitario[\[Gim\]](#), dal mancato aggiornamento del piano pandemico (risalente al 2006) [\[SZ\]](#) e dalla scarsa integrazione dei sistemi informatici. In questa situazione sono emersi esempi virtuosi, è il caso delle Provincia Autonoma di Trento che ha sviluppato tra la primavera e l'estate del 2020 un'app chiamata *TreCovid19* progettata per fornire al personale sanitario dedicato uno strumento pronto all'uso per il monitoraggio a distanza dei pazienti con sintomi progressivi di COVID-19, sottoposti a quarantena domiciliare durante la prima ondata dell'epidemia, e concentrarsi su quei pazienti che, in base ai dati clinici auto-riferiti, richiedevano una risposta rapida da parte degli operatori sanitari. Lo sviluppo è stato motivato dalla necessità di supportare il personale sanitario nell'affrontare l'improvviso aumento del carico di lavoro presentato durante la prima fase di trasmissione della malattia. Per raggiungere questo obiettivo, il sistema *TreCovid19* è stato impostato per:

- Raccogliere regolarmente e automaticamente i sintomi auto-riferiti dai pazienti con COVID-19 in quarantena domiciliare attraverso un'applicazione per smartphone
- Tradurre i report personali dei sintomi in scale numeriche
- Consentire una serie di allarmi basati su specifici cut-off, informando periodicamente il personale sanitario sullo stato dei pazienti e ottimizzando gli interventi e i contatti diretti se necessario.

L'applicazione mobile è stata integrata con una piattaforma di telemedicina già esistente, adottata dagli operatori di assistenza domiciliare. L'approccio principale adottato è stato quello di fondere le componenti organizzative e tecnologiche incorporando un servizio di telemedicina (supportato principalmente da un'app dedicata) nel quadro già esistente delle procedure cliniche per il monitoraggio dei pazienti in quarantena domiciliare. Il sistema è stato rapidamente progettato sviluppando due macro-componenti:

1. Un'applicazione per i pazienti per supportare l'auto-raccolta quotidiana dei sintomi
2. Una dashboard per medici e infermieri incaricati di monitorare i pazienti

La homepage dell'app *TreCovid19* presenta una serie di funzionalità che si dividono in due tipologie principali. La prima categoria riguarda la fornitura di informazioni ufficiali sulla pandemia COVID-19 e di informazioni generali per affrontarla, come ad esempio aggiornamenti recenti sui dati dell'epidemia, video tutorial forniti dall'APSS, suggerimenti e consigli su specifiche procedure di sicurezza, informazioni su decreti regionali e nazionali e normative relative alla pandemia COVID-19.

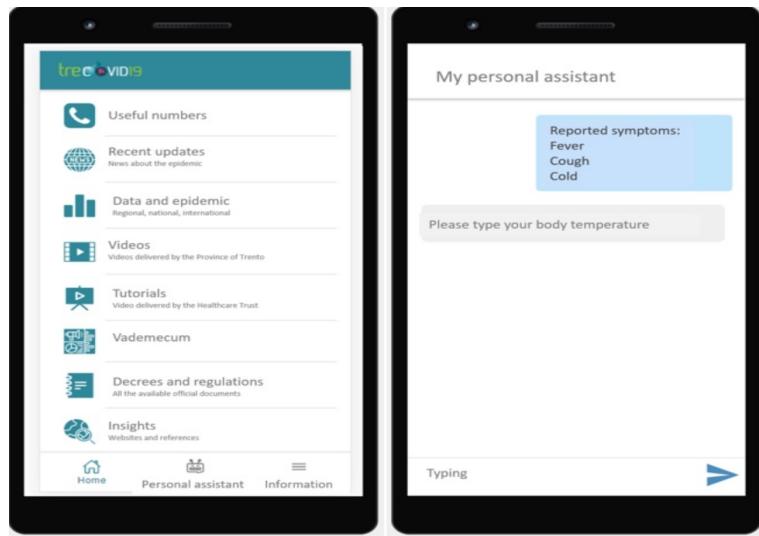


Figura 13: TreCovid19

La seconda categoria ha costituito lo strumento principale per i pazienti ed è rappresentata da una funzionalità di chatbot automatizzata. L'app attivava periodicamente una chat specifica con il paziente, somministrando una serie di item per raccogliere dati auto-riferiti sul suo stato di salute e sui sintomi correlati. Le diverse informazioni venivano raccolte due volte al giorno (una al mattino e una al pomeriggio) e comunicate in tempo reale al sistema centrale. Nel caso in cui il livello dei sintomi auto-riferiti superasse i valori limite stabiliti dal personale sanitario, veniva inviato un allarme specifico via e-mail all'Unità speciale COVID-19 per un'immediata (ri)azione.

La progettazione è stata effettuata secondo un approccio multidisciplinare coinvolgendo medici, infermieri, informatici per identificare i potenziali elementi da includere nel sistema, come le scale di monitoraggio. La sperimentazione che ha coinvolto i pazienti è stata una fase delicata ed importante, per ridurre al minimo i potenziali rischi sono stati scelti pazienti positivi ma in quarantena domiciliare, residenti nella provincia di Trento, in condizioni cliniche stabili e non critiche ed in grado di utilizzare uno smartphone correttamente. Inoltre i pazienti sono stati partizionati in due distinti gruppi:

1. *Gruppo Rosso (Acv19)*: pazienti considerati positivi a COVID-19, determinati in base a uno dei seguenti criteri: segnalazione di sintomi clinicamente rilevanti, risultato positivo del test del tampone o parametri clinici rilevanti basati sull'esame radiologico.
2. *Gruppo Blu (AOcv19)*: conviventi o familiari che vivono con un paziente risultato positivo alla COVID-19. Il gruppo blu è stato avviato specificamente per monitorare l'evoluzione delle condizioni dei conviventi, in particolare per segnalare la loro potenziale infezione da parte del virus.

In totale sono stati selezionati 170 pazienti, 107 pazienti sono stati assegnati al gruppo rosso (pari al 62.9%) mentre i rimanenti 63 pazienti (pari al 37.1%) sono stati assegnati al gruppo blu. Lo studio che ha portato allo sviluppo dell'app *TreCovid19* ha evidenziato una serie di problematiche e aspetti ai quali chi intende progettare applicazioni di questo tipo deve prestare attenzione. Come prima cosa è necessario identificare le priorità chiave da affrontare tra le questioni urgenti da gestire, il secondo aspetto è legato alla capacità di selezionare infrastrutture preesistenti, adattarle alla luce di nuovi fattori contestuali e fornire il servizio in tempi ragionevoli. La fornitura di un sistema di tele-monitoraggio era essenziale per poterlo realizzare nel più breve tempo possibile, considerando il rapido aumento dell'epidemia di COVID-19 che stava mettendo a dura prova

le istituzioni sanitarie provinciali, sviluppare da zero un nuovo strumento per il monitoraggio dei pazienti COVID-19-positivi si sarebbe rivelato un compito impossibile e dannoso. L'approccio multidisciplinare adottato in TreCovid19 è stata la chiave del successo, l'armonizzazione tra gli asset tecnologici e le procedure organizzative, così come la messa insieme del know-how clinico, delle competenze in materia di salute pubblica e delle conoscenze informatiche, è stata possibile solo grazie a un metodo di lavoro poliedrico e integrativo. Il fatto che medici, infermieri, project manager e tecnici informatici fossero già esposti a una collaborazione a lungo termine nel campo della salute digitale ha permesso una cooperazione rapida e senza intoppi in un contesto critico a causa della pandemia COVID-19. [GFM+21]

La provincia autonoma di Trento non è stata l'unica realtà a sviluppare appositamente un'app per il monitoraggio a distanza dei pazienti, nel Luglio del 2021 la regione Toscana ha lanciato l'app *Toscana Salute* per l'accesso semplice e veloce da parte dei cittadini ai servizi on-line della sanità toscana. Con l'app *Toscana Salute* è possibile:

- prenotare analisi del sangue
- effettuare una televisita
- consultare il libretto di gravidanza digitale
- chiamare il Pronto Badante
- consultare informazioni utili per stranieri
- scaricare la certificazione verde Covid 19
- consultare i tuoi referti
- scaricare il promemoria delle ricette per farmaci, visite ed esami
- consultare il borsellino elettronico per celiaci
- consultare gli accessi al pronto soccorso
- prenotare visite mediche ed esami
- pagare il ticket sanitario
- scegliere il medico di famiglia
- compilare e inviare l'autocertificazione per esenzione o per reddito per la compartecipazione alla spesa sanitaria
- gestire il taccuino personale
- trovare i numeri utili del Servizio sanitario e delle Aziende sanitarie

Gli utenti possono scaricare l'applicazione sia dal Play Store (per i dispositivi Android) che dall'App Store (per i dispositivi iOS), una volta scaricata basta effettuare l'accesso tramite le credenziali SPID per usufruire di tutti i servizi a disposizione. [Tos]

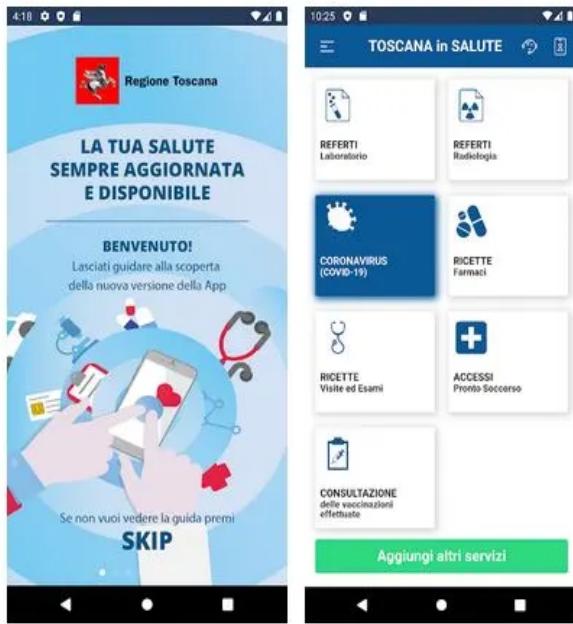


Figura 14: Toscana Salute

Toscana Salute non è l'unica app nel Play Store con cui gli utenti possono consultare informazioni ospedaliere, sempre in ambito regionale è stata sviluppata l'app *Salutile* dalla regione Lombardia. L'applicazione include funzionalità quali:

- Consultazione degli ospedali e di tutti i Pronto Soccorso del territorio Lombardo
- Visualizzazione del numero di persone in coda e dei tempi di attesa
- Visualizzazione dell'indice di affollamento

Nell'applicazione non solo è presente un elenco degli ospedali ma è possibile visualizzare gli ospedali su una mappa in modo interattivo.

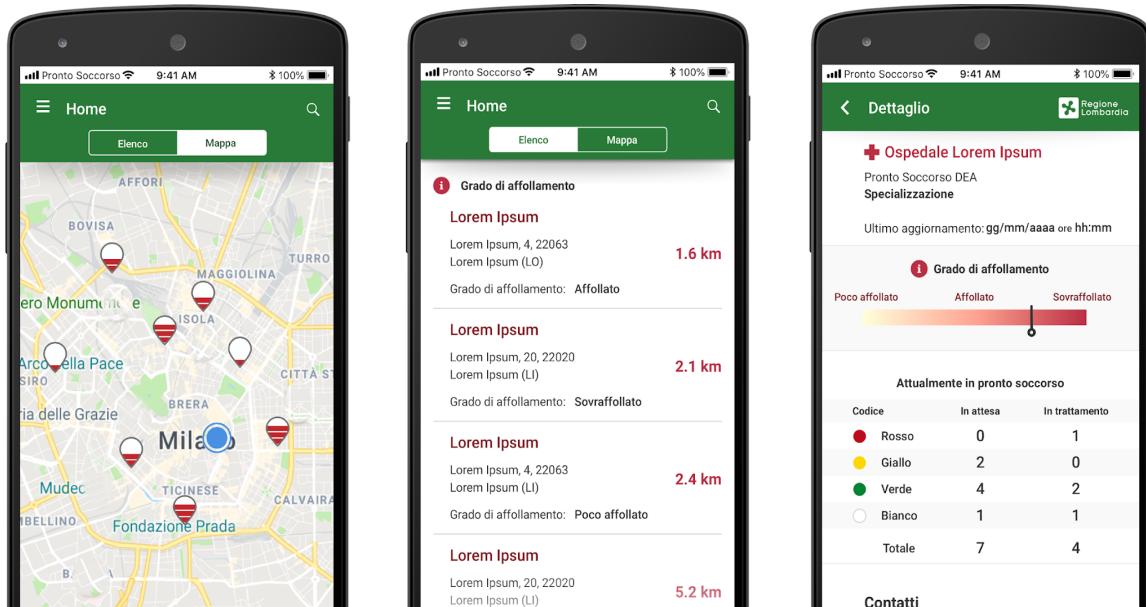


Figura 15: Salutile

Nel Play Store è presente una applicazione simile a Salutile, analizzata precedentemente, ma relativa alle strutture ospedaliere della regione Lazio. L'app *Pronto Soccorso Lazio Ospedali* include funzionalità quali:

- Visualizzazione di ospedali e pronto soccorso nella regione Lazio
- Visualizzazione su una mappa interattiva degli ospedali laziali
- Visualizzazione del numero di pazienti in attesa e in trattamento

La mappa è in grado di mostrare un percorso per raggiungere l'ospedale a partire dalla posizione dell'utente.

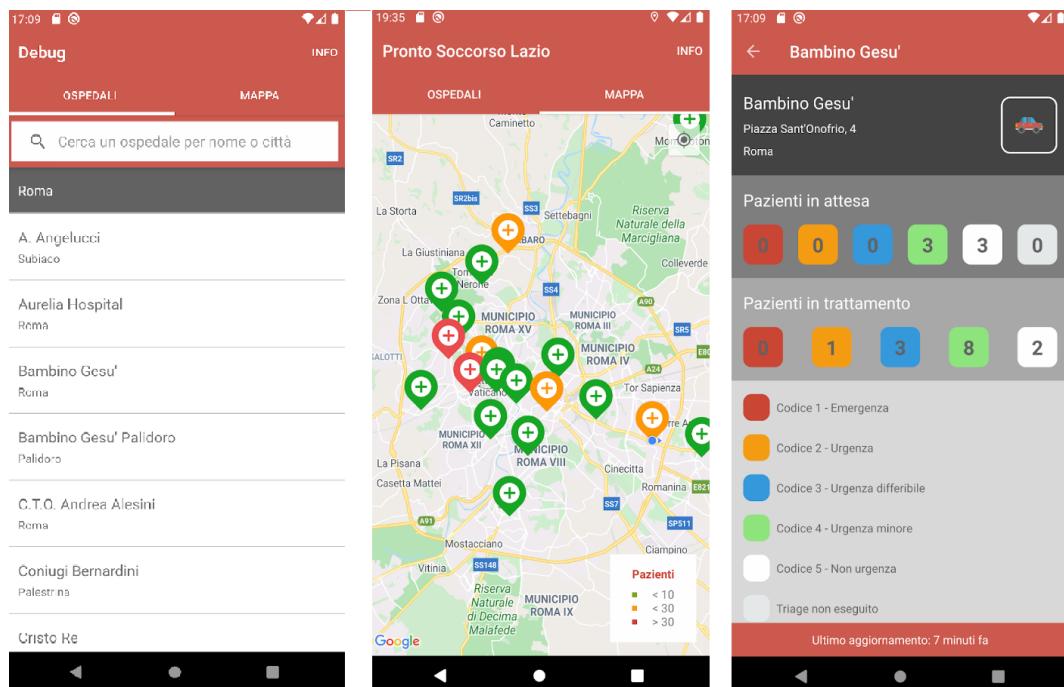


Figura 16: Pronto Soccorso Lazio Ospedali

Nonostante l'applicazione sia graficamente gradevole e facile da usare, molti utenti segnalano la presenza di numerosi bug che portano il rating complessivo nel playstore ad un valore pari a 2.9. Il principale limite delle app analizzate è la frammentazione sia a livello di informazioni che di funzionalità offerte, al momento non esiste alcuna app che include tutte le informazioni utili di tutte le strutture ospedaliere di tutte le regioni.

### **Applicazioni mobile per farmacie**

Un'altra tipologia di applicazioni da analizzare sono le app relative alla ricerca/suggerimento/visualizzazione di farmacie e parafarmacie, queste app sono usate molto più frequentemente rispetto alle app per gli ospedali e coprono una platea di utenti più ampia. Analizzo in dettaglio le app disponibili sul Play Store esaminando funzionalità, pregi e difetti.

#### **1 - Cerca Farmacie**

L'applicazione si presenta con un'interfaccia abbastanza basilare e consente di cercare farmacie in una città inserita dall'utente o rilevata tramite GPS. L'utente può scorrere la lista delle farmacie presenti nella città selezionata visualizzando per ciascuna informazioni quali: nome, indirizzo, numero di telefono e orari di apertura. Punti di forza:

- Semplice da usare
- Informazioni complete

Punti di debolezza:

- Design datato
- Richiede connessione a internet



Figura 17: Cerca Farmacie

## 2 - Farmacie di turno

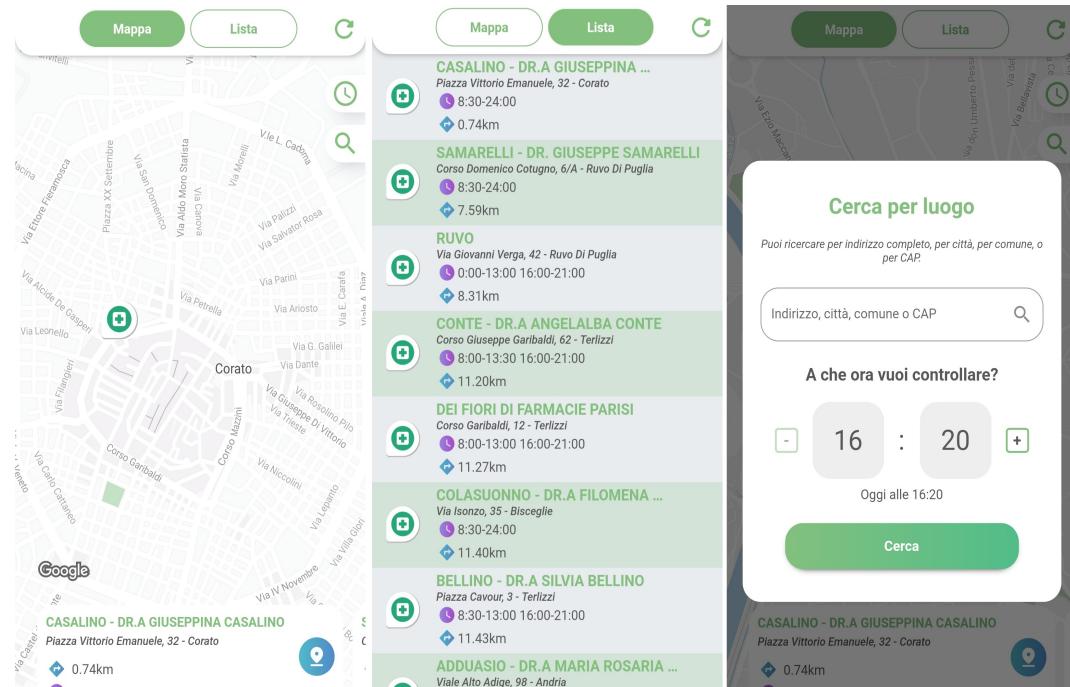
L'applicazione dal design moderno e accattivante mostra le farmacie di turno nelle vicinanze dell'utente acquisendo la posizione tramite GPS, l'app inoltre presenta una lista di farmacie ordinate per distanza crescente dalla posizione dell'utente. Per ciascuna farmacie sono disponibili informazioni quali: nome, indirizzo, numero di telefono e orari di apertura.

Punti di forza:

- Design minimale ed estetico
- Mappa interattiva
- Possibilità di cercare le farmacie anche in base all'orario

Punti di debolezza:

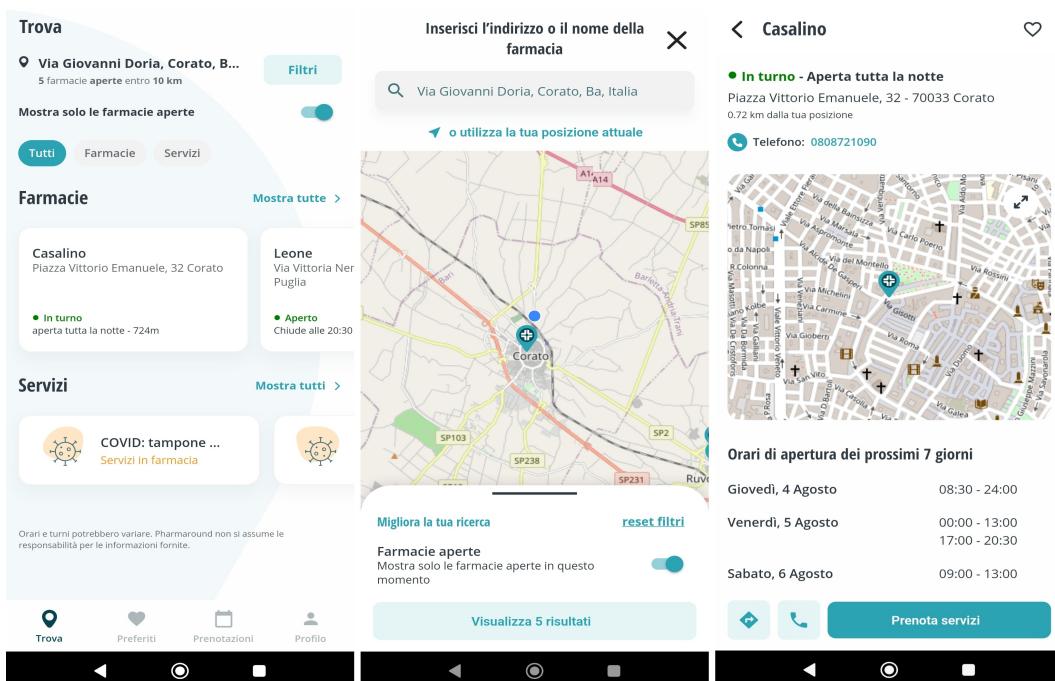
- Manca la funzionalità di cercare la farmacia su internet



**Figura 18: Farmacie di turno**

### 3 - Pharmaround

L'applicazione Pharmaround è una delle migliori applicazioni per il suggerimento delle farmacie presenti sul Play Store, le funzionalità non includono soltanto la ricerca/visualizzazione di farmacie ma è anche possibile la prenotazione di servizi quali: test per il Covid, vaccini, analisi del sangue e visite dal veterinario per gli animali domestici inoltre è in grado di filtrare le farmacie in base a prodotti e servizi offerti il tutto in una interfaccia moderna e minimale.



**Figura 19: Pharmaround**

**Punti di forza:**

- Design estetico e moderno
- Numerose funzionalità presenti
- Possibilità di creare un profilo

#### 4 - FarmacieSpecializzate

L'ultima applicazione che analizzo è l'applicazione FarmacieSpecializzate, un'app che presenta funzionalità quali ricerca/visualizzazione di farmacie nelle vicinanze dell'utente rilevando la posizione GPS oppure inserendo una città. In aggiunta l'applicazione consente all'utente l'inserimento di una farmacia mediante un codice univoco (fornito dal farmacista) o la scansione di un QR code. Punti di forza:

- Interfaccia minimale
- Possibilità di inserire farmacie

Punti di debolezza:

- Mancanza di una mappa interattiva

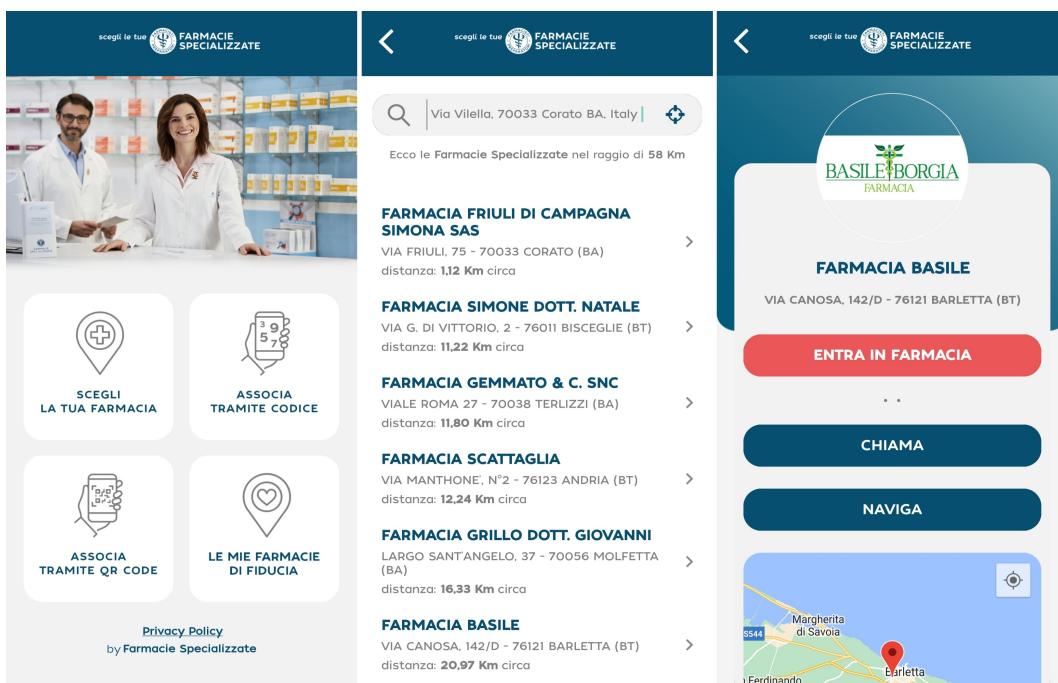


Figura 20: FarmacieSpecializzate

## 2.4 Applicazioni mobile per il monitoraggio psicologico

La pandemia ha avuto un impatto psicologico notevole sulle persone, solo nel primo anno della pandemia i disturbi da ansia e da depressione sono aumentati di ben il 25% in tutto il mondo. Ciò è dovuto a fattori di stress multipli, alla solitudine dovuta ai vari lockdown, alla paura di infettarsi, i più colpiti da questi disturbi sono le donne ma in particolare i giovani, da sempre trascurati da parte della politica. In questo quadro si aggiunge un notevole gap tra i servizi per la salute mentale disponibili e la richiesta crescente da parte dei cittadini, per colmare questo gap tra le misure adottate da parte di vari governi, in particolare dal governo italiano, c'è il *bonus psicologo* le cui richieste (oltre 113.000) sono esaurite nel giro di pochissime ore escludendo di fatto metà di chi ne ha fatto richiesta.[\[OMS, con\]](#) In questa situazione, sicuramente critica, possono le applicazioni mobile dare un contributo significativo al miglioramento della salute mentale degli utenti? E se sì, in che modo?

Una applicazione sviluppata specificatamente per il supporto di pazienti affetti da disturbi psicologici e disagio psichico dovuti alla pandemia è l'applicazione COVID Coach: uno strumento interattivo per affrontare al meglio lo stress e l'ansia. L'applicazione fornisce informazioni su come stare bene e funzionalità per il monitoraggio di sintomi legati alla salute mentale e il collegamento con vari tipi di supporti verificati e monitorati. L'app può essere utilizzata in modo indipendente o in combinazione con l'assistenza professionale per la salute mentale, ma non è un sostituto della terapia. Inoltre gli utenti non sono tenuti a creare un account o ad accedere a qualsiasi contenuto ed è compatibile con le tecnologie assistive. L'applicazione è stata rilasciata sia sul Play store, per smartphone e tablet Android, che sull'Appstore per dispositivi iOS alla fine dell'Aprile 2020 e stando ai dati raccolti in data 31 Ottobre 2020, il numero di download è stato pari a 143.097 con un rating di 4,8 su entrambi gli store. L'utenza media nei sei mesi considerati è stata di circa 49000 utenti unici, la maggior parte di questi ha usato l'app per i primi tre giorni e solo il 2% degli utenti ha usato l'applicazione per più di 15 giorni di fila.[\[JTR<sup>+21\]</sup>](#)

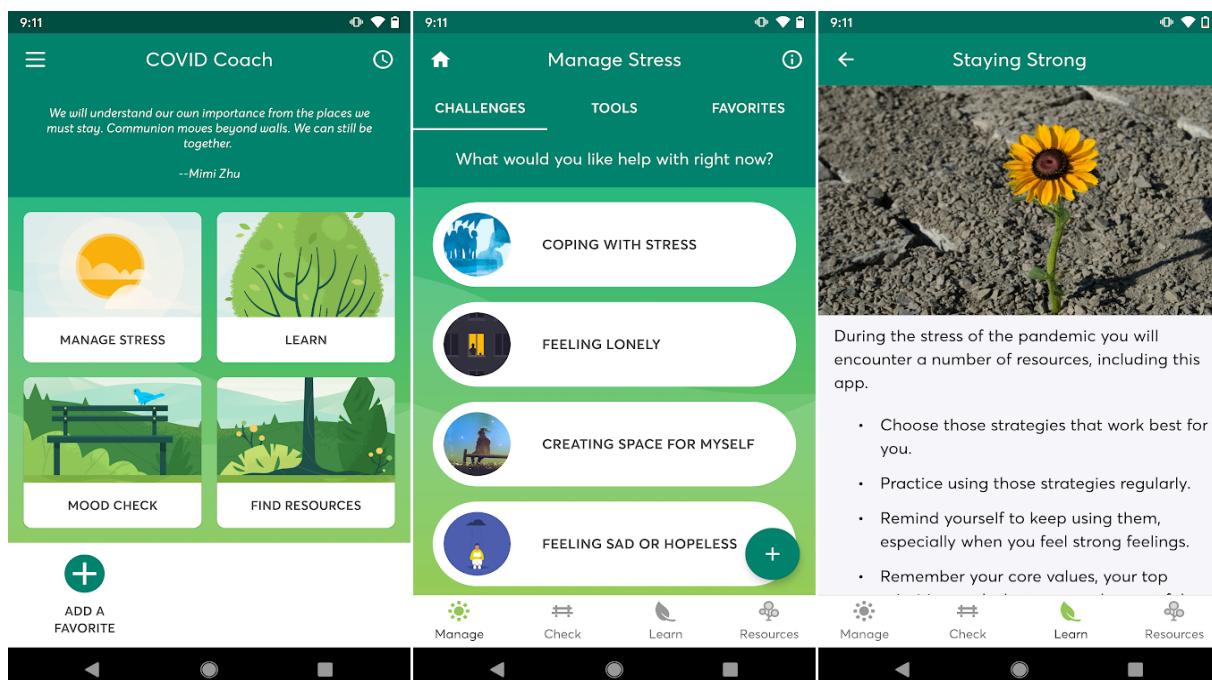


Figura 21: COVID Coach

Non essendoci stato un questionario non è possibile conoscere le motivazioni che hanno spinto

gli utenti ad abbandonare l'utilizzo dell'applicazione ma ciò dimostra l'importanza da parte degli sviluppatori di curare funzionalità che possano aumentare l'utilizzo e favorire l'engagement degli utenti.

Nel 2021 è stata creata una applicazione per il monitoraggio della salute mentale nel personale sanitario duramente provato dalla pandemia. L'applicazione *Wellness Hub* è stata progettata come strumento per il mantenimento della salute mentale generale e della resilienza e mette in contatto gli utenti con i servizi di salute mentale disponibili attraverso il Center for Stress, Resilience, and Personal Growth (CSRPG) e il Mount Sinai. Un team di psichiatri e psicologi affiliati al CSRPG ha fornito tutti i contenuti dell'applicazione e ha collaborato con il Digital Discovery Program dell'Hasso Plattner Institute for Digital Health del Mount Sinai per guidare lo sviluppo del software. L'applicazione contiene sondaggi sulla salute mentale presentati nella schermata principale, gli utenti vengono introdotti ai sondaggi in una modalità di base, in cui tutte le domande vengono proposte in una sola volta, in un unico sondaggio di base. Dopo il completamento, gli utenti vedranno ogni sondaggio individualizzato nella schermata iniziale e potranno rispondere a ciascun sondaggio tutte le volte che lo desiderano. I sondaggi offerti agli utenti sono misure convalidate comunemente utilizzate nella ricerca e nella pratica clinica, anche se è possibile rispondere in modo coerente all'interno dell'applicazione, anziché secondo un calendario convalidato. Questo perché i sondaggi sono pensati come un meccanismo di feedback per gli utenti, che possono valutare regolarmente il loro benessere e la loro salute mentale e ricevere un feedback coerente in base ai loro risultati. I sondaggi sono stati scelti in base alle conoscenze pregresse sui disturbi che si verificano nel contesto del trauma e ai dati ricavati da un precedente sondaggio distribuito tra gli operatori sanitari del Mount Sinai Hospital. *Wellness Hub* contiene i seguenti sondaggi:

1. Valutazione dell'umore: è stato incluso un misuratore di sentimenti giornalieri con emoticon che indicano tristezza e felicità su una scala da 1 a 7 per consentire ai partecipanti di valutare il loro stato d'animo giornaliero.
2. PHQ-8 (Patient Health Questionnaire a 8 voci) e GAD-2 (scala del disturbo d'ansia generalizzato a 2 voci) ("My Mood"). Il PHQ-8 e il GAD-2 rilevano rispettivamente la depressione e l'ansia generale nelle ultime due settimane.
3. PCL-5 (PTSD Checklist for DSM-5 [Diagnostic and Statistical Manual of Mental Disorders, Fifth Edition]) ("My Stress Reactions"). Il questionario è composto da 4 item per valutare le reazioni da stress post-traumatico.
4. CD-RISC2 (Connor-Davidson Resilience Scale a 2 elementi) ("My Coping Resources"). È stato utilizzato il CD-RISC2, una forma breve del CD-RISC precedentemente validata. La scala va da 0 (per niente vero) a 4 (quasi sempre vero), con punteggi più alti che indicano una maggiore resilienza.
5. WHO-5 (Indice di benessere dell'Organizzazione Mondiale della Sanità a 5 voci) ("Il mio benessere"). L'OMS-5 è una misura a 5 voci utilizzata a livello globale del benessere generale. Gli item sono presentati su una scala da 0 (mai) a 5 (sempre). Punteggi più alti indicano un maggiore benessere
6. AUDIT-C (Alcohol Use Disorders Identification Test-Concise) ("Il mio uso di alcol"). L'AUDIT-C è una misura di screening a 3 item ampiamente utilizzata per l'abuso di alcol. Gli item sono presentati su una scala da 0 a 4; i punteggi di 3 o superiori sono stati considerati suggestivi di problemi di abuso di alcol.

7. Item sulla lotta spirituale ("La mia spiritualità"). Sulla base di ricerche precedenti, nell'applicazione sono stati inseriti due item per misurare i problemi con la spiritualità: (1) "Lotta con la perdita di significato e di gioia nella sua vita?" e (2) "Ha attualmente delle lotte che definirebbe spirituali o religiose?". Questi item sono stati presentati su una scala Likert a 4 punti che va da 0 (per niente) a 3 (molto). L'approvazione di un valore pari o superiore a 1 su uno dei due item ha dato origine a un feedback relativo all'affrontare la perdita di significato o i problemi spirituali.

I questionari sono presentati in una interfaccia moderna e facile da usare come possiamo vedere dalla figura seguente.

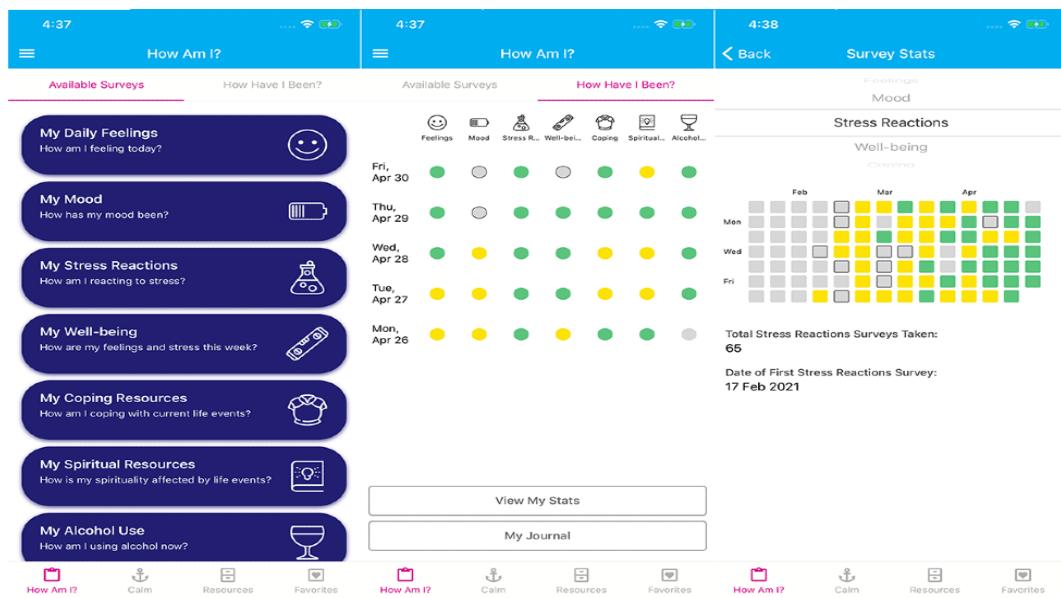


Figura 22: Wellness Hub

Per le persone che desiderano interagire direttamente con un esperto di salute mentale o con una rete di supporto, l'applicazione fornisce informazioni di contatto per il CSRPG e altri uffici e pratiche comportamentali del Mount Sinai. L'applicazione attraverso la sezione "Le mie risorse" include una funzione di diario, che consente agli utenti di annotare qualsiasi pensiero o esperienza che ritengono importante registrare. Queste annotazioni servono a monitorare i propri sentimenti e pensieri e possono essere sotto forma di testo, immagine o registrazione vocale e possono anche essere etichettate come una normale voce di diario o un atto di resilienza. Un'altra sezione presente è la sezione "My Calm" che offre agli utenti una libreria di video di rilassamento, alcuni provenienti dal web pubblico e altri progettati specificamente per i dipendenti del Mount Sinai. Questi video variano per argomento e durata, in modo che l'utente possa scegliere quello che fa al caso suo in quel momento.

La sperimentazione dell'applicazione è durata circa quattro mesi e mezzo coinvolgendo in totale 231 utenti distinti, di questi 173 utenti, pari al 74.0%, hanno completato la valutazione di base di tutti i parametri della salute mentale presenti nell'app. A tre mesi dal lancio circa il 55% degli utenti è rientrato nell'app dopo la prima apertura per esplorare ulteriori funzionalità, con una media di quattro aperture dell'app per utente. In media, gli utenti hanno effettuato 2,06 indagini dopo aver completato la valutazione di base. Di tutti i sondaggi disponibili sull'app, quelli relativi ai sentimenti quotidiani ("I miei sentimenti quotidiani"), al PHQ-8 ("Il mio stato d'animo") e alla PCL-5 ("Le mie reazioni allo stress") sono stati i più eseguiti, rispettivamente con 410, 368 e 318 complementi totali.[GZD<sup>+21</sup>]

Le applicazioni analizzate ovvero *Covid Coach* e *Wellness Hub* si basano sostanzialmente su una valutazione soggettiva dello stato emotivo e psicologico, alcuni sintomi di problemi più gravi potrebbero essere sottovalutati da parte degli stessi utenti il che crea un ostacolo alla valutazione ed alla predizione accurata dello stato psicologico. Uno studio, pubblicato nel 2021, denominato *A Mobile Sensing App to Monitor Youth Mental Health: Observational Pilot Study* si è posto l'obiettivo di correlare i dati raccolti in un lasso di tempo di 14 giorni da diversi sensori presenti in uno smartphone per effettuare una predizione su possibili sintomi di malattie psichiche in un target di utenti composto prevalentemente da giovani. I partecipanti, di età compresa tra 10 e 21 anni, hanno completato due misure di sintomi in formato digitale all'inizio dello studio di 14 giorni. I sintomi d'ansia sono stati valutati utilizzando lo Screen for Child Anxiety-Related Emotional Disorders (SCARED). Lo SCARED è uno strumento self-report per i bambini e i loro genitori, che comprende 41 item per lo screening di diversi tipi di disturbi d'ansia. Una volta completato il questionario preliminare gli utenti hanno installato una applicazione appositamente progettata per registrare i dati di interesse dello studio. L'app PROSIT (Predicting Risk and Outcomes of Social Interaction) raccoglie diversi indici di comportamento della vita di ogni giorno mediante l'uso dello smartphone. In questo studio sono state utilizzate le seguenti caratteristiche dell'app PROSIT: come prima cosa, l'app ha raccolto dati GPS ogni volta che i giovani si sono spostati in un raggio di 20 metri, impedendo l'identificazione dell'indirizzo di casa o della posizione geografica precisa per preservare la privacy. In secondo luogo, l'applicazione ha registrato gli eventi di chiamata (ossia, ora e data della chiamata, durata e contatto delle chiamate in entrata e in uscita) e gli eventi di accensione e spegnimento dello schermo (ossia, ora e data), infine, sempre tramite PROSIT, sono stati campionati i dati sulla luce ambientale come indicatore del sonno. Le chiamate e l'uso del telefono erano flussi di sensori basati su eventi, mentre i dati del GPS e della luce ambientale sono stati campionati come serie temporali. Dopo l'installazione, PROSIT è stato costantemente eseguita in background e ha raccolto i dati dei sensori e dell'utilizzo. Per proteggere i dati sensibili raccolti dall'applicazione sono stati resi anonimi mediante una funzione di hash crittografica già integrata.



Figura 23: Studio app PROSIT

In totale sono stati coinvolti 161 partecipanti, di questi:

- 11 partecipanti si sono ritirati a causa di dubbi riguardo la privacy

- 25 partecipanti hanno fornito dati per meno di 14 giorni come richiesto
- 3 partecipanti non hanno fornito dati di alcun sensore rilevanti ai fini dello studio

Quindi lo studio si è concentrato sul 75,7% dei partecipanti selezionati in fase iniziale, l'età media è pari a 18 anni e la maggior parte sono ragazze (96/122). In totale, il 24,5% (30/122) dei giovani partecipanti aveva una diagnosi di depressione, il 31,9% (39/122) aveva una diagnosi di disturbo d'ansia generalizzato e il 9,8% (12/122) aveva una diagnosi di fobia sociale in vita. Per quanto riguarda le misure self-report dei sintomi, lo SCARED ha una media di 33,02 (SD 14,79) e un range 3-62, mentre il CES-DC ha una media di 32,59 e un range 1-58. Durante il periodo di studio, al 9,8% (12/122) dei giovani è stata diagnosticata la depressione, all'11,4% (14/122) l'ansia generalizzata e al 4,9% (6/122) la fobia sociale.

Prima di correlare i dati raccolti dai questionari con i dati provenienti dai sensori su questi ultimi è stata effettuata una fase preliminare di estrazione di features e una analisi statistica. Nella feature extraction, per quanto riguarda i dati provenienti dal sensore GPS, sono state registrate le coordinate distanti tra loro almeno 20 metri e che sono state visitate per almeno due minuti da parte dell'utente estraendo la distanza totale (percorsa ogni giorno), l'entropia normalizzata (per la precisione della misurazione) ed il tempo passato in un certo punto geografico. Per ogni chiamata telefonica sono stati registrati il timestamp, il tipo di chiamata, e la durata. Gli eventi di accensione e spegnimento dello schermo del cellulare sono stati trasformati in due caratteristiche: (1) il numero totale di volte in cui lo schermo è stato acceso al giorno e (2) la quantità totale di tempo trascorso sullo schermo al giorno (calcolata come differenza tra il numero di eventi di accensione o spegnimento dello schermo). Dai dati sulla luce ambientale, sono state registrate le variazioni di intensità luminosa e calcolato il tempo trascorso a ciascun livello di intensità luminosa. Successivamente, è stata calcolata l'intensità luminosa media totale, il tempo trascorso nelle fasce di intensità luminosa più bassa e più alta durante le ore notturne del giorno (dalle 23 alle 7), che è stato utilizzato come indicatore della qualità del sonno. Completata la feature extraction, le analisi sono state regolate in base all'età, al sesso, allo status socio-economico e al sistema operativo dello smartphone.

Lo studio ha indicato l'esistenza di correlazioni significative tra i sintomi e i biomarcatori dello smartphone. Livelli più bassi di mobilità, un minor numero di interazioni sociali e una maggiore intensità della luce ambientale durante la notte, utilizzata come causa di una peggiore qualità del sonno, erano correlati a sintomi depressivi più elevati. Livelli più elevati di mobilità, minori interazioni sociali e un maggiore utilizzo dello schermo dello smartphone erano correlati a sintomi di ansia più elevati tra i giovani. Lo studio ha anche dimostrato che, aggiungendo ai modelli di previsione i dati raccolti passivamente dai sensori dello smartphone, siamo è stato possibile un migliore adattamento del modello. Infine, i risultati indicano il potenziale dei biomarcatori dello smartphone di avere una natura discriminante nel predire altri sintomi ma non i sintomi del disturbo da deficit di attenzione e iperattività negli adulti.

Gli studi analizzati dimostrano l'efficacia delle applicazioni nel monitoraggio dello stato emotivo e psicologico degli utenti sia in funzione attiva, raccogliendo e analizzando dati provenienti da sensori, che in funzione passiva, somministrando questionari e domande all'utente sul proprio stato mentale. Numerose sono le applicazioni di questo tipo presenti nel Playstore, che riscontrano un successo tra gli utenti non trascurabile, di seguito analizzo le principali applicazioni disponibili agli utenti.

## **1 - Amaha**

L'applicazione si presenta in una interfaccia moderna e accattivante, guidando l'utente in un per-

corso di cura e terapia.

Punti di forza:

- Interfaccia piacevole
- Percorso guidato per l'utente

Punti di debolezza:

- Poco scientifica

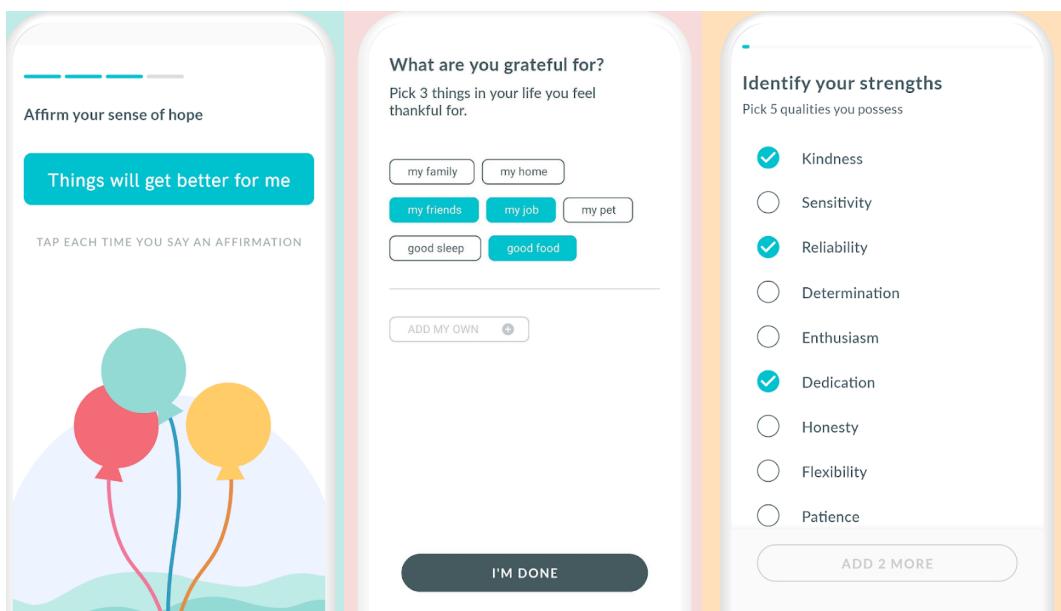


Figura 24: Amaha

## 2 - BetterMe

L'applicazione si focalizza sulla meditazione come attività per migliorare la salute mentale dell'utente, analizzando le informazioni inserite in fase iniziale relative ad abitudini, obiettivi e motivazioni elabora un piano settimanale per la meditazione.

Punti di forza:

- Applicazione originale, unica nel suo genere
- Elevata personalizzazione
- Interfaccia moderna

Punti di debolezza:

- Costo elevato delle versioni premium

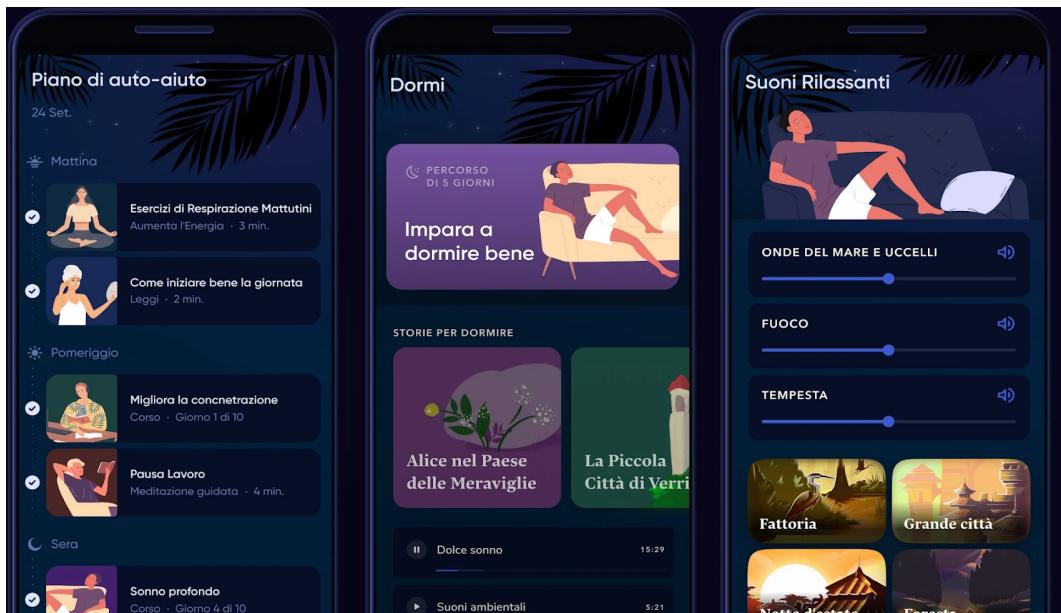


Figura 25: BetterMe

### 3 - Wysa

L'applicazione è user-friendly e offre numerose funzionalità, oltre al classico monitoraggio dello stato emotivo, include un chatbot intelligente per la CBT (Cognitive Behavioral Therapy) che insegna le tecniche CBT suggerendo attività quali: meditazione, yoga e attività per gestire l'ansia.

Punti di forza:

- Numerose funzionalità
- Chatbot

Punti di debolezza:

- Gran parte delle funzionalità sono a pagamento

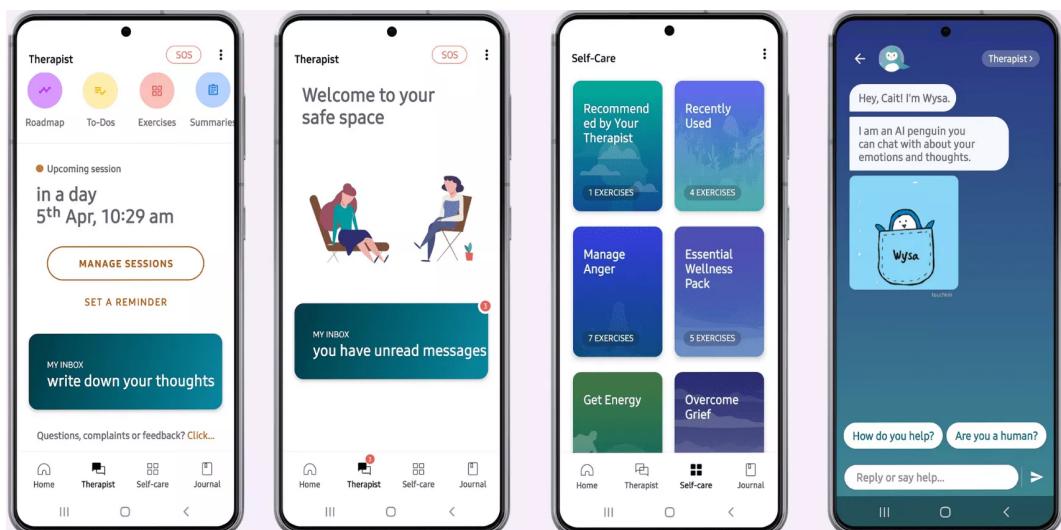


Figura 26: Wysa

#### 4 - MyPossibleSelf

L'applicazione utilizza contenuti riconosciuti a livello clinico, creati in collaborazione con esperti leader a livello mondiale presso Priory Healthcare, per aiutare i pazienti a migliorare la propria salute mentale. Gli strumenti e le tecniche interattive, che utilizzano la terapia cognitivo-comportamentale (CBT), sono stati personalizzati per l'uso digitale. L'app comprende sei serie guidate: Dormire bene, Superare l'ansia, Affrontare la depressione, Alleviare lo stress lavorativo, Bere in sicurezza e Giocare d'azzardo in sicurezza.

Punti di forza:

- Interfaccia intuitiva
- Approvata dall'NHS britannico
- Numerose funzionalità



Figura 27: MyPossibleSelf

#### 5 - MindDoc

MindDoc aiuta gli utenti a tracciare, monitorare e comunicare i sintomi della depressione a un operatore sanitario. L'applicazione chiede all'utente di rispondere ogni giorno, per due settimane, a domande sui propri sintomi e al termine delle due settimane, viene creato un riepilogo che gli utenti possono condividere con un operatore sanitario. Nell'app sono presenti quattro sezioni principali: Percorso dell'umore, Conoscenze, Statistiche, Profilo.

Punti di forza:

- Possibilità di condividere le misurazioni con professionisti
- Interfaccia minimale

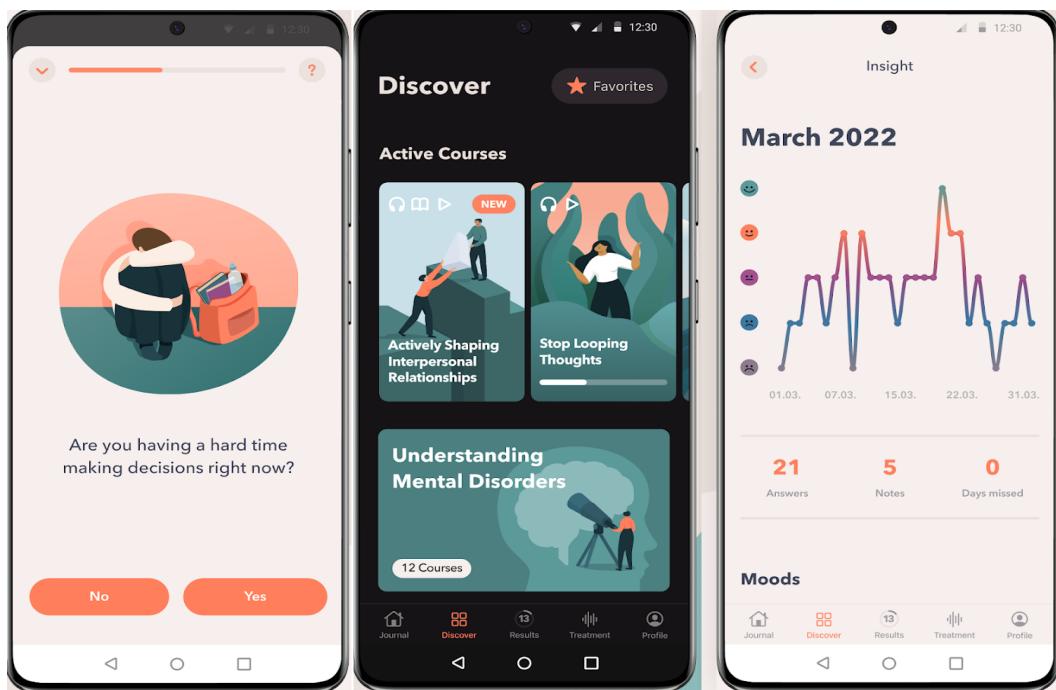


Figura 28: MindDoc

## 6 - Being

L'applicazione include oltre alle funzionalità per il monitoraggio dello stato emotivo, delle mini-sessioni di terapia di circa 10 minuti, delle musiche rilassanti ed una chat in cui un operatore umano risponde in poco tempo a qualsiasi domanda posta dall'utente.

Punti di forza:

- Presenza delle sessioni di terapia
- Chat in tempo reale
- Interfaccia piacevole

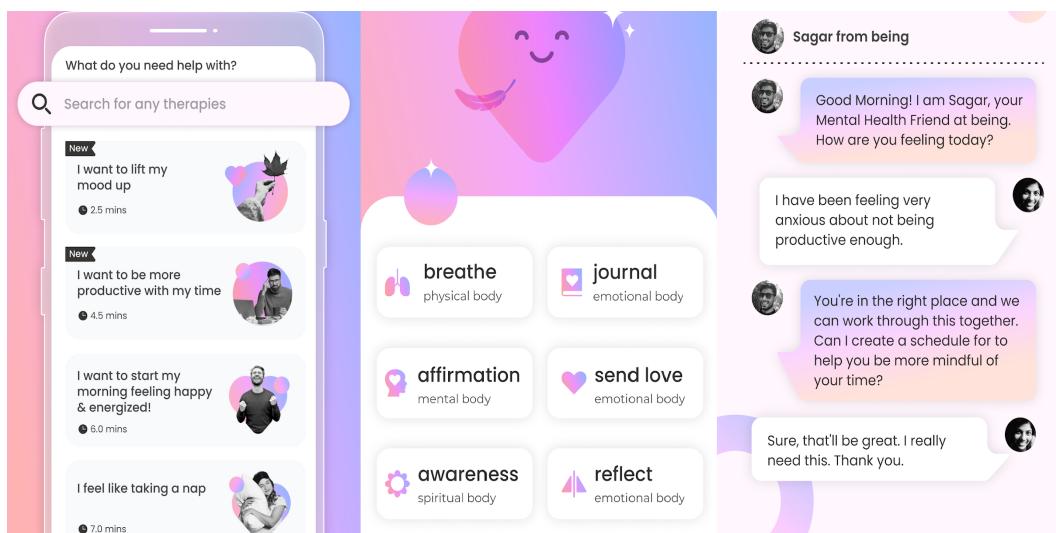


Figura 29: Being

## 7 - Reflectly

Reflectly è un'applicazione di mindfulness guidata dall'intelligenza artificiale, utilizza domande intelligenti per aiutare gli utenti a ottenere un modo semplice e diretto per eliminare i pensieri più importanti dalla loro testa, ogni giorno. L'applicazione include un diario in cui l'utente può descrivere lo stato emotivo, inserendo note vocali e fotografie inoltre sono presenti molte frasi motivazionali piacevoli.

Punti di forza:

- Funzionalità basate sull'ai
- Interfaccia piacevole
- Numerose funzionalità

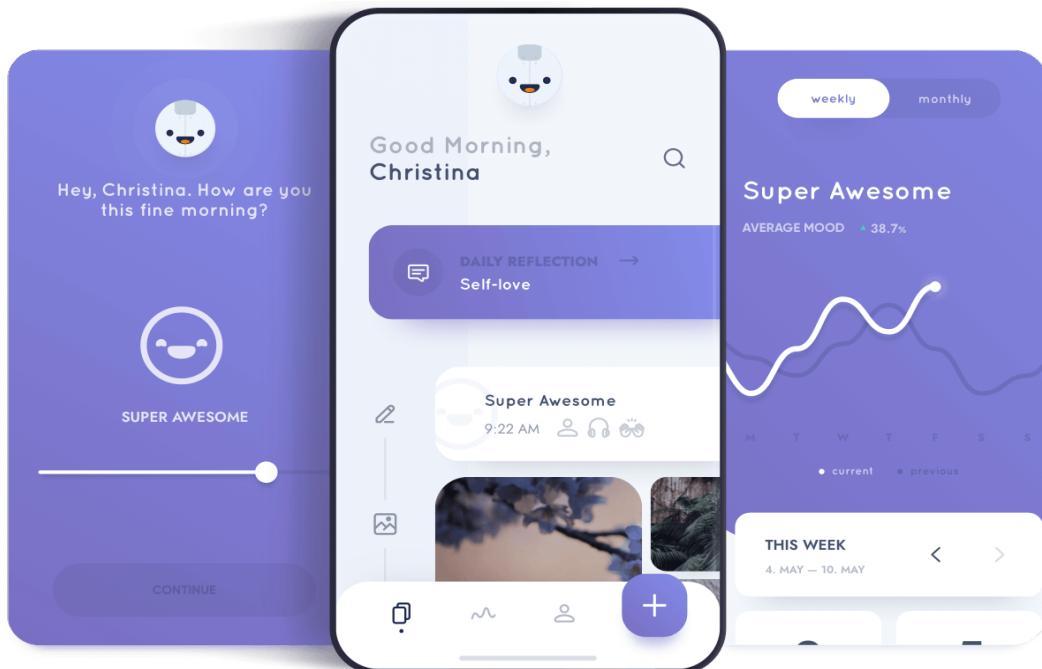


Figura 30: Reflectly

Le applicazioni presenti nel playstore che sono state analizzate non hanno lo scopo di diagnosticare malattie o problemi psichici quanto quello di fornire all'utente uno strumento semplice ed intuitivo per esprimere il suo stato emotivo e tenerne traccia nel tempo. Molte applicazioni includono funzionalità personalizzate e percorsi di auto-miglioramento per aiutare nei disturbi più comuni quali: ansia, stress, burnout, difficoltà nelle relazioni e non intendono sostituirsi ad alcun professionista.

## 2.5 Tecnologie di supporto alle applicazioni e-health

### Sistemi di raccomandazione ed assistenti virtuali nelle applicazioni e-health

Le applicazioni di questo tipo offrono numerose funzionalità, funzionalità che non si limitano ad avere un ruolo passivo nella raccolta dei dati e nella presentazione delle informazioni ma possono avere un ruolo attivo analizzando l'utente durante l'utilizzo in modo da fornire un'esperienza personalizzata ed unica.

I sistemi di raccomandazione sono sistemi in grado di fornire contenuti ed informazioni personalizzate in base al profilo dell'utente, profilo appreso mediante tecniche di machine learning e deep learning analizzando le informazioni personali, i suoi gusti dedotti dalle informazioni a cui accede con lo scopo di prevedere nel modo più accurato possibile ciò che potrebbe essere di suo interesse. In uno studio pubblicato nel settembre del 2017 dal titolo: *emHealth: Towards Emotion Health Through Depression Prediction and Intelligent Health Recommender System* i ricercatori hanno sviluppato un sistema di raccomandazione per un'applicazione rivolta ad utenti con disturbi mentali. L'applicazione analizzando lo stato mentale dell'utente mediante domande-e-risposte suggerisce il trattamento psicologico più appropriato usando alberi di decisione e support vector machines (SVM).[\[YZD<sup>+</sup>18\]](#) Sono stati sviluppati sistemi di raccomandazione per suggerire alimenti e ricette sulla tenendo conto della salute dell'utente, è il caso del recommender system sviluppato dall'Università di Bolzano descritto nel paper *Health-aware Food Recommender System*. Nelle fasi iniziali il sistema chiede all'utente quali sono i cibi preferiti, le abitudini in cucina (cosa cucina l'utente di solito) ed il suo profilo. Sulla base del bilancio calorico calcolato a partire dal profilo dell'utente, il sistema seleziona le ricette che hanno un apporto calorico che si avvicina di più a questo valore. Mediante questo criterio è effettuato un ranking delle ricette, per ogni ricetta l'utente può esprimere un giudizio di gradimento da 1 a 5 (scala Likert) che influirà sulle successive raccomandazioni.[\[GRM15\]](#) Gli assistenti virtuali tramite l'interazione con l'utente sono in grado di profilarlo in modo da fornire dati preziosi ad un sistema di raccomandazione. L'introduzione degli assistenti virtuali nelle app ha aperto una nuova frontiera dell'interazione tra gli utenti ed il software che andrà a rendere obsoleti menù e bottoni. Le tecniche di *natural language processing* (elaborazione del linguaggio naturale) consentono di comprendere il linguaggio naturale sia in forma verbale che in forma scritta con lo scopo di ottenere sistemi abili a dialogare con l'utente. Gli assistenti virtuali possono fornire una grande varietà di informazioni in modo semplice e veloce, caratteristica alla base del successo di assistenti quali: Alexa, Google Assistant e Siri, questi ultimi durante la pandemia da Covid-19 sono stati utilizzati dagli utenti per informazioni riguardanti: possibili sintomi, trattamenti e numero di casi giornalieri. Uno studio, intitolato "*Alexa, Am I pregnant?": A content analysis of a virtual assistant's responses to prenatal health questions during the COVID-19 pandemic*", condotto nel marzo 2021 si è focalizzato sull'analisi delle risposte di Alexa, l'assistente virtuale prodotto da Amazon, a 40 domande relative la salute prenatale da donne in gravidanza, di cui 3 domande relative al covid, formulate dal collegio americano degli ostetrici e dei ginecologi per poi classificare le risposte secondo l'accuratezza (accurata/non accurata) e la completezza (completa/non completa). L'analisi è stata effettuata su due dispositivi Alexa e ha dato i seguenti risultati:

- 36 su 40 risposte identiche
- 10% delle risposte sono state ritenute accurate
- Sul 35% delle domande (14 su 40) Alexa non è stata in grado di fornire una risposta
- Il 52% delle risposte (21 su 40) non erano accurate
- Il 15% delle risposte (6 su 40) erano irrilevanti per la domanda

Inoltre le risposte fornite alle tre domande relative al covid sono state ritenute sbagliate o non sufficienti.[SRE21] Questo studio ha mostrato la necessità di sviluppare assistenti virtuali più accurati e precisi nel fornire informazioni e risposte che potrebbero influenzare la salute ed il benessere dell'utente, con questo obiettivo in mente i ricercatori della University of South-Australia hanno sviluppato un assistente virtuale per la dieta e l'attività fisica in uno studio durato 12 settimane. L'assistente virtuale chiamato *Paola* ha fornito allenamenti e dieta personalizzati mediante:

1. Una sessione introduttiva che ha insegnato agli utenti l'aumento dell'attività fisica e la dieta mediterranea, comprese le porzioni giornaliere e settimanali raccomandate.
2. Ogni settimana dopo la sessione introduttiva, per 11 settimane, i partecipanti hanno ricevuto un' e-mail che li invitava di completare un check-in settimanale con Paola riguardo alla loro attività fisica e alla dieta per la settimana precedente. Paola ha fornito un feedback personalizzato e poi ha assistito gli utenti nell'impostare obiettivi personalizzati per la settimana successiva
3. Paola era disponibile h24 per rispondere alle domande degli utenti sull'attività fisica e la dieta mediterranea.

I 31 partecipanti, di età compresa tra 45 e 73 anni, hanno complessivamente promosso l'assistente virtuale, ritenendolo uno strumento utile, infatti l'89% ha ritenuto il linguaggio usato da Paola corretto, fattore che ha spinto il 67% dei partecipanti a raggiungere gli obiettivi per la dieta ed il 60% gli obiettivi del numero di passi ed esercizio fisico.[DMCM20] Assistenti virtuali e bot intelligenti possono essere usati anche in ambito ospedaliero, per l'assistenza dei pazienti. È il caso di *HealthAssistantBot*, un bot tutto italiano progettato per:

- Comprendere i sintomi dei pazienti.
- Suggerire medici/specialisti.
- Monitorare trattamenti e parametri vitali.
- Aumentare la consapevolezza dei pazienti relativa a sintomi e malattie.

fornendo un supporto ai pazienti nella prenotazione di visite e trattamenti. Per utilizzare il bot basta scaricare Telegram, una applicazione popolare di messaggistica con oltre 500 milioni di utenti, ed una volta avviato è in grado di interagire con l'utente mediante una chat. L'interazione risulta particolarmente efficace grazie al menù principale costituito da 6 sezioni quali:

1. Analisi dei sintomi
2. Glossario medico
3. Monitoraggio
4. Trattamenti
5. Profilo utente
6. Raccomandazione di dottori

in particolare per la funzionalità *Analisi dei sintomi* basta descrivere i sintomi mediante un messaggio ed il sistema fornisce la malattia compatibile con i sintomi descritti. Per fare questo il bot utilizza una base di conoscenza appositamente creata in italiano per malattie e relativi sintomi da cui un modello, addestrato mediante un dataset di pazienti, effettua la classificazione della malattia in base ai sintomi descritti dall'utente. La funzionalità *Raccomandazione di dottori* in cui il sistema in base ai sintomi suggerisce i medici analizzando le aree cliniche associate. Alla base vi è un sistema di raccomandazione *content-based* basato su informazioni estratte dal portale mediciitalia.it tra cui: aree di competenza, strutture mediche e grado di soddisfazione.[\[PNI<sup>+</sup>20\]](#)

L'integrazione di sistemi di raccomandazione ed assistenti digitali fornisce un valore aggiunto all'applicazione e consente di coinvolgere più a lungo gli utenti, aumentando la fidelizzazione, inoltre fornisce un'arma nel monitoraggio costante dei pazienti in modo che i medici ed il personale sanitario possano offrire le migliori cure.

Gli assistenti virtuali per funzionare al meglio necessitano dei dati dei pazienti, delle malattie e delle cure in formato elettronico machine-readable dati che devono essere disponibili agli specialisti. La grande mole di dati presenti in questi sistemi rende complesso ed oneroso in termini di tempo l'accesso ed il ritrovamento delle informazioni necessarie per questo sono nati dei motori di ricerca specializzati per l'information retrieval delle informazioni mediche. In generale possiamo suddividere i modelli per il ritrovamento dell'informazione in tre categorie:

1. Modelli probabilistici
2. Modelli basati su conoscenza
3. Modelli basati sull'apprendimento automatico

I modelli probabilistici si basano sul calcolo della probabilità della rilevanza tra un documento e la query dell'utente, i modelli basati su conoscenza sfruttano delle *knowledge base* di un dominio specifico mentre i modelli basati sull'apprendimento automatico utilizzano dati e features estratte algoritmicamente per trovare dei pattern. La progettazione di questi sistemi comporta numerose sfide infatti devono saper gestire la complessità del linguaggio medico nonché la moltitudine di informazione presenti la cui rilevanza potrebbe cambiare in base al contesto e l'ambiguità delle frasi presenti nelle query formulate magari da utenti non-experti, su quest'ultimo aspetto si è concentrato il motore di ricerca **MedSearch**. MedSearch è progettato per l'utilizzo non solo da parte di personale specializzato ma anche da parte di utenti che non hanno familiarità con la terminologia medica né un background in ambito medico permettendo la formulazione di lunghe query in linguaggio naturale non dettagliate. MedSearch utilizza un'ontologia la *Medical Subject Headings*: un vocabolario standard contenente diverse categorie tra cui: anatomia, organismi, malattie, farmaci utilizzata per indicizzare e catalogare documenti biomedici o relativi alla salute, nel caso di questo sistema l'ontologia è utilizzata per identificare nelle pagine maggiormente rilevanti le frasi mediche sulle quali è effettuato un ranking in base alla rilevanza con la query originale. Ogni query è processata in quattro passi:

1. Rimozione delle stopwords
2. Riscrittura della query con una lunghezza minore se è troppo lunga
3. Fornire i risultati di ricerca delle pagine
4. Generare snippets
5. Suggerire frasi mediche correlate

Quest'ultima feature è particolarmente importante in quanto riduce i tempi di ricerca fornendo risultati più accurati, il suggerimento delle frasi mediche avviene selezionando i termini delle N-migliori pagine ritrovate (in base alla query) con i valori del *tf-idf* (Term Frequency - Inverse Document Frequency) maggiori per poi effettuare un ranking con cui il sistema seleziona i termini migliori, inoltre per ogni pagina selezionata il sistema genera uno *snippet* ovvero seleziona tre termini che sono più vicini alla query.[[LYW08](#), [GT18](#)]

Questo tipo di motori di ricerca rendono la sanità e le informazioni ad essa correlate sempre più accessibili e fruibili da parte di tutti, la diffusione di questi sistemi unita alla condivisione reciproca delle informazioni tra ospedali consente ai medici non solo di trovare farmaci e cure adatta ma anche di integrare le diagnosi sui pazienti con casi clinici analoghi già trattati.

### I dispositivi wearable e l'e-health

Un dispositivo wearable, secondo il media lab del MIT di Boston, è un computer che può essere indossato da un utente. L'interazione avviene mediante dispositivi di input non invasivi, è dotato di connettività wireless e di una serie di sensori per rilevare il contesto agendo come un assistente virtuale.[[Lab](#)] In ambito sanitario, di nostro interesse, i *wearables* possono essere definiti come tecnologie usate per misurare vari parametri fisiologici essendo indossati dell'utente.[[LBL+20](#)] L'utilizzo per la prima volta, secondo questa definizione, avviene durante le missioni spaziali degli anni '60, in cui la NASA mise a punto l'equipaggiamento biomedicale per trasmettere i dati biometrici degli astronauti quali elettrodi per rilevare il battito cardiaco, un sensore per rilevare la respirazione sulla base della temperatura dell'aria nella bocca ed un termometro rettale per la misurazione della temperatura corporea. Sono dispositivi poco pratici, ingombranti ma che hanno contribuito a mandare l'uomo nello spazio.



Figura 31: Strumentazione medica missione Apollo

La ricerca si è concentrata sul ridurre l'intrusività, aumentando la comodità e la precisione, ricerca favorita dall'evoluzione dei materiali e dalla miniaturizzazione delle componenti hardware. In questo modo è possibile per un dispositivo dalle dimensioni ridotte di ricevere segnali, elaborare e memorizzare i dati mediante un microprocessore ed una memoria inviandoli a dispositivi esterni mediante bluetooth, NFC o RFID ad altri smart-device. La miniaturizzazione ha portato alla nascita di dispositivi applicabili in diverse parti del corpo in grado di registrare diversi parametri vitali che è possibile classificare in questo modo:

- Activity trackers: Dispositivi per raccogliere dati e monitorare l'attività fisica dell'utente.
- E-skin: Una "pelle artificiale" con proprietà meccaniche della pelle umana con funzioni tattili.
- Tessuti intelligenti: Tessuti che incorporano sensori e circuiti integrabili con gli indumenti

- Cinture EEG ed ECG: Cinture per monitorare lo stato di salute dell'utente.
- Tute aptiche: Tute per registrare dati biometrici e cinetici mediante sensori integrati nella tuta.
- Dispositivi inseribili e deglutibili: Oggetti che vanno dentro, attraverso e sotto il corpo umano dalle dimensioni di una compressa contenente sensori in grado di monitorare malattie ed assunzione di farmaci.
- Interfacce neurali: Dispositivi in comunicazione diretta con il cervello usati per pazienti con disabilità fisiche e comportamentali
- Indumenti smart: Vestiti intelligenti con sensori e funzionalità integrate quali: riscaldamento, ricarica, visualizzazione
- Cerotti smart: Cerotti usa-e-getta che aderiscono alla pelle per monitorare in tempo reale i parametri vitali.
- Anelli smart: anelli con sensori, senza una interfaccia o display, che inviano dati a dispositivi esterni.
- Tatuaggi smart: Biosensori impiantati sotto la pelle che misurano i livelli di glucosio e cambiano colore a seconda del risultato.

In base alla tecnologia ed ai materiali utilizzati, questi dispositivi possono essere applicati in varie parti del corpo (testa, petto, braccia, gambe) o arrivare perfino nell'apparato digerente nel caso dei dispositivi deglutibili. La zona in cui il dispositivo verrà applicato influenza sue dimensioni e sulla potenza di calcolo che avrà, fattore cruciale per i progettisti che incide sulle funzionalità, in base alla potenza di calcolo possiamo suddividere i dispositivi in:

1. Dispositivi a bassa potenza: costituiti da hardware con limitate capacità di calcolo, sono progettati per operare per molto tempo con lo scopo di acquisire e trasmettere dati.
2. Dispositivi a media potenza: hanno una potenza di calcolo maggiore, possono avere un display per mostrare le informazioni e molteplici sensori.
3. Dispositivi ad alta potenza: hanno processori potenti che devono elaborare ad alta frequenza grandi moli di dati.

[OSK<sup>+21</sup>] La tecnologia è in continua evoluzione, nei prossimi anni saranno sviluppati sicuramente nuovi tipi di dispositivi con nuove sfide da affrontare per i progettisti, magari in futuro sarà possibile usare questi device per aumentare le capacità fisiche e cognitive umane. Dalla fantascienza alla realtà, nell'agosto 2020 la startup Neuralink, fondata dall'imprenditore visionario Elon Musk, ha presentato una *brain-machine interface* (interfaccia uomo-macchina) progettata per essere impiantata nel cervello umano per migliorare la memoria e l'interazione con i sistemi informatici. Il dispositivo sviluppato da Neuralink adotta un approccio invasivo: delle sonde polimeriche ultrasottili (4-6 micrometri) intercettano i segnali celebrali e li inviano ad un circuito integrato (Neuralink ASIC) dove verranno amplificati e digitalizzati per poi essere inviati in modo wireless ad uno smartphone con l'app Neurallink. L'applicazione più importante di Neuralink è per i pazienti con lesioni dei nervi della spina dorsale, i quali possono controllare un mouse o una tastiera digitali con il solo pensiero. Lo sviluppo di questa tecnologia aprirà nuove applicazioni interessanti tra cui la cura del parkinson, il trattamento dell'epilessia e di disturbi quali l'autismo o la depressione ma al momento non è stata ancora testata sull'uomo in quanto manca l'approvazione della Food and Drug

Administration (FDA). [Gur21, M<sup>+</sup>19] FDA che ha dato l'autorizzazione nel 2017 alla pillola smart ingeribile sviluppata dall'azienda californiana Proteus Digital Health per monitorare con precisione il momento in cui è assunto qualsiasi farmaco. Una volta raccolti i dati, la pillola li trasmette ad un cerotto intelligente che fa da intermediario con lo smartphone del paziente.

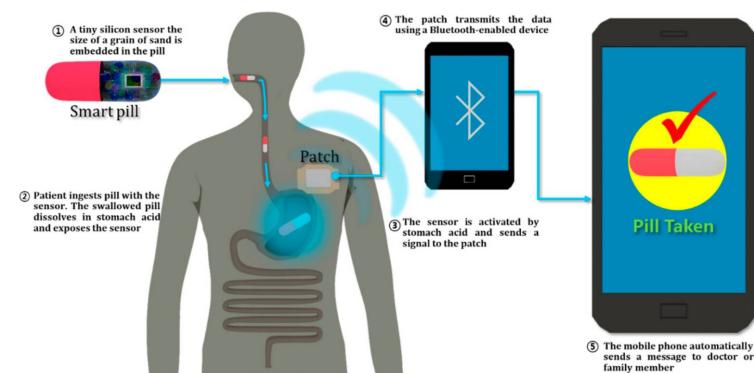


Figura 32: Proteus Digital Health

Quest'ultima è una delle tante applicazioni dei dispositivi ingeribili in quanto l'analisi delle sostanze chimiche presenti nel cavo orale, nell'esofago, nello stomaco e nell'intestino può fornire informazioni sulla presenza di malattie ed infezioni in atto. Con dei dispositivi dotati di telecamera è possibile catturare immagini di parti del tratto digerente non accessibili dalle normali endoscopie, oppure mediante appositi sensori presenti all'interno, è possibile misurare la temperatura interna, il pH dello stomaco fino a rilevare i gas e le sostanze presenti nell'intestino. La progettazione di queste "pillole" presenta numerosi requisiti da soddisfare e relative difficoltà quali le dimensioni che devono essere ridotte al massimo in modo che il dispositivo possa essere facilmente deglutito, deve consumare poca potenza in quanto non è possibile avere una batteria molto grande inoltre nei soggetti più fragili potrebbero esserci problemi di ritenzione.[KZHOB17] Il sistema progettato dalla Proteus Digital Helth sfrutta uno *smart patch* per trasmettere i dati, questo tipo di dispositivi sono molto meno invasivi ma altrettanto efficaci nel monitoraggio dei pazienti. Questi cerotti possono essere usati per la somministrazione mirata di farmaci tramite una membrana presente all'interno, mediante dei semiconduttori è possibile misurare: temperatura, pressione del sangue e ritmo cardiaco inoltre i sensori possono sfruttare proprietà biochimiche per l'analisi dei microfluidi presenti sulla pelle raccolti mediante dei canali presenti all'interno del cerotto. I cerotti più avanzati incorporano più funzionalità quali: il rilevamento, l'attuazione, la terapia, l'alimentazione, la comunicazione wireless e l'adesione cutanea reversibile biocompatibile all'interno di una configurazione a film sottile. Le sfide per questi cerotti riguardano i materiali che devono essere flessibili e resistenti nonché il processo di fabbricazione su larga scala data la complessità.[HKS<sup>+</sup>18] Sulla pelle è possibile non solo applicare cerotti ma anche tatuaggi con speciali caratteristiche diventando elementi non più puramente estetici ma funzionali al monitoraggio dello stato di salute dell'utente. Questi tatuaggi non sono altro che bio-sensori ovvero sensori costituiti da un recettore biologico o sintetico che si lega a un biomarcatore target per produrre cambiamenti molecolari o creare prodotti di reazione che vengono amplificati e trasdotti attraverso letture ottiche, elettriche o magnetiche. Maaplne ha sviluppato un nanosensore in grafene per rilevare la presenza di batteri nella saliva e nella respirazione mentre il gruppo Wang ha sviluppato un tatuaggio in grado di monitorare la presenza di lattato, glucosio, ammoniaca ed alcool nel corpo. Ad Harvard ed al MIT sono stati sviluppati dei bio-inchiostri in cui, per mezzo di una reazione chimica, avviene un cambiamento di colore che segnala la presenza, la concentrazione di sostanze ed il pH in base alla tonalità ed all'intensità sfruttando la pelle come interfaccia per trasmettere le informazioni.

[VJL<sup>+17</sup>, GHL<sup>+19</sup>] I nanomateriali che costituiscono i cerotti smart hanno anche applicazione negli *smart-clothes*. Questi ultimi rispetto ai bio-inchiostri ed ai cerotti smart di cui abbiamo parlato, non necessitano di particolari tecniche o metodi per aderire al corpo, basta indossare l'indumento che tramite i sensori di cui è dotato trasmette i dati. La progettazione di questi indumenti presenta requisiti di tipo:

- **Tecnico:** I materiali devono essere abbastanza resistenti ed elastici per l'utilizzo durante le attività, inoltre le batterie devono durare abbastanza per alimentare l'elettronica incorporata durante le attività da monitorare.
- **Funzionale:** Gli indumenti devono essere confortevoli da indossare, l'elettronica presente non deve limitare in alcun modo i movimenti né produrre surriscaldamento
- **Estetico:** L'estetica è un requisito essenziale nell'abbigliamento in questo caso è necessario considerare i materiali e l'elettronica da utilizzare
- **Culturale:** È necessario distinguere l'abbigliamento per età, genere, pubblico a cui è rivolto. Uno stesso indumento può essere gradevole o sgradevole in base a fattori culturali, tradizioni e dress code.

Soddisfatti questi requisiti, le applicazioni sono molteplici non solo nel fitness e nello sport ma anche per il trattamento di pazienti e nella riabilitazione. Un'applicazione interessante è lo sviluppo di pantaloni smart, dotati di attuatori collegati a dei sensori per rilevare il movimento, questi pantaloni sono in grado di applicare una forza che assiste e facilita il movimento della caviglia.[KKL<sup>+20</sup>] Per soggetti con problemi cardiaci o per atleti sottoposti per lungo periodo a sforzo, può essere utile monitorare il ritmo cardiaco in modo da segnalare eventuali eventi anomali. Tramite degli elettrodi presenti nella zona della schiena e del petto è possibile effettuare le misurazioni per poi essere memorizzate in una scheda SD ed inviate via bluetooth a dispositivi esterni. [MCT13] I dati raccolti mediante gli smart clothes possono essere integrati con quelli provenienti da uno smartwatch o fitness tracker. I primi smartwatch nascono tra gli anni '80 e gli anni '90, offrendo funzionalità di un micro-computer quali: semplici calcoli, giochi, invio di messaggi, alcuni erano perfino in grado di ricevere le immagini televisive. Nel 2003 Microsoft, in collaborazione con Citizen, Fossil, Suunto e Swatch, lancia SPOT: uno smartwatch in grado di offrire: news, meteo, risultati sportivi, un servizio di messaggistica e tanto altro. Nonostante il lancio in *pompa magna* avvenuto al Consumer Electronics Show di Las Vegas, nel 2005 erano stati venduti soltanto 100,000 orologi, risultato che spinse il CEO di Swatch a dichiarare il progetto *un totale fallimento* citando in causa Microsoft per un risarcimento di 14 milioni di dollari.[Thob] In generale questo tipo di dispositivi non ebbe molto successo, in quanto poco pratici ma tra 2008 e 2010 avviene una svolta, che cambia il mercato: nascono Fitbit e nuove startup da cui sono lanciati smartwatch integrabili con Android ed iOS (sistemi operativi mobile nati da pochi anni) ed in grado di misurare, mediante appositi sensori, dati relativi alle attività dell'utente. L'idea dietro cambia, si passa da orologi con l'ambizione di sostituire completamente computer e telefoni cellulari a smartwatch che puntano a fornire funzionalità aggiuntive, integrabili con sistemi esterni, con particolare attenzione al monitoraggio dell'attività fisica.

## 3 Analisi e progettazione dell'applicazione Helena

### 3.1 Introduzione al sistema operativo Android

Android è un sistema operativo mobile per dispositivi quali: smartphone, tablet e smartwatch (wearOS basato su Android) lanciato sul mercato nel settembre 2008 dalla big-tech Google con lo scopo di fornire un sistema operativo completo e funzionale per tutti i produttori di smartphone la cui diffusione sul mercato è avvenuta a partire dal 2007. Android è un sistema operativo *open-source* con componenti proprietarie basato su Unix. Ad Aprile 2022 in tutto il mondo, Android è diffuso su oltre il 71% dei dispositivi mobile, seguito solo da iOS con circa il 27% e da altri sistemi operativi quali: KaiOS, Windows 10 Mobile con percentuali trascurabili.[[Staa](#)] I fattori alla base del successo di Android sono molteplici, sicuramente la scelta di Google di rendere il sistema open-source e adattabile da parte dei vari produttori è stata una scelta vincente a differenza di Apple che ha mantenuto, secondo la filosofia aziendale, il sistema chiuso e non installabile su dispositivi di altre marche. Android è disponibile per una grande varietà di dispositivi con diverse prestazioni e prezzi sul mercato, l'utente quindi ha una vasta possibilità di scelta, a ciò si unisce un design del sistema minimale, la presenza dei servizi e software Google tra cui: Gmail, Google Maps, Google Drive, Google Lens ed il numero elevato di applicazioni disponibili sul PlayStore, a marzo 2022 erano disponibili oltre due milioni e mezzo di applicazioni sul Playstore.[[Stab](#)]

La storia di Android inizia nel 2003, ben prima della diffusione degli smartphones, *Android Inc.* fu fondata a Palo Alto da Rich Miner, Nick Seats, Chris White ed Andy Rubin per migliorare i sistemi operativi presenti nelle fotocamere digitali per poi concentrarsi su i sistemi operativi mobile pochi mesi dopo. Tutto cambia nel 2005 con l'acquisizione da parte di Google che rese disponibile il sistema operativo a tutti i produttori gratuitamente allo stesso tempo guadagnando sulla fornitura di servizi ed applicazioni. Il primo lancio della versione beta di Android ovvero *Android 1.0* avviene il 5 Novembre 2007 e nel settembre 2008 il primo smartphone con Android ovvero l'*HTC Dream* fa il suo debutto sul mercato. La prima versione includeva funzionalità quali: Google Maps, Youtube, un browser con supporto ad HTML5 ed una prima versione del Playstore denominato "Android Market". Nell'aprile 2009, la versione 1.5 "Cupcake" introduce migliorie quali il supporto a tastiere di terze parti, la rotazione automatica dello schermo e la possibilità di caricare video su Youtube dall'applicazione. Le versioni successive includono miglioramenti delle funzionalità e dell'interfaccia: nella versione 2.2 sono introdotte funzionalità quali notifiche push ed hotspot Wi-fi, nella versione 2.3 è introdotto il supporto allo standard di comunicazione di prossimità NFC, nelle versioni 4.x il sistema viene ottimizzato per dispositivi con 512 MB di memoria ram e supporto al browser Chrome. Le versioni più recenti tra la 6.0 e la 12.0 si sono concentrate sul miglioramento dell'esperienza utente per dispositivi con schermi e prestazioni maggiori introducendo features tra cui: la Vulkan API per il rendering 3d, un multitasking migliore delle applicazioni ed il supporto ai dispositivi pieghevoli.[[Cala](#)]

## 3.2 Architettura di Android

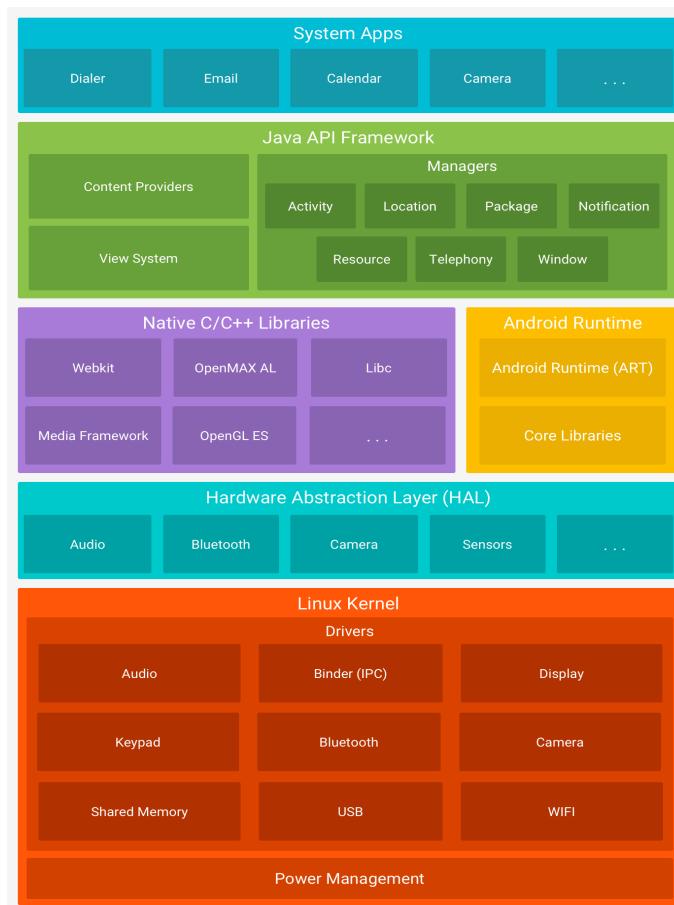


Figura 33: Stack Android

Alla base di Android vi è il Kernel Linux 2.6 che fornisce le funzionalità base del sistema quali: gestione di processi, memoria, batteria gestendo anche sensori e dispositivi i/o quali: fotocamera, sensori, schermo ed hardware esterno. L'*Hardware Abstraction Layer* - HAL fornisce un'interfaccia standard per utilizzare le funzionalità dell'hardware ad alto livello includendo diversi moduli, ciascuno dei quali implementa un'interfaccia per un componente o una funzionalità dell'hardware specifica. Nel livello superiore a questo ci sono le librerie ed il runtime Android (ART) per eseguire il codice delle applicazioni. A partire dalla versione 5.0 il processo ed i relativi threads di ogni app sono eseguiti in un'istanza dell'Android runtime progettato per far girare macchine virtuali multiple eseguendo files DEX: un formato di bytecode specifico per Android ottimizzato per avere un impatto minimo sulla memoria. Molte componenti e funzionalità del sistema richiedono librerie scritte in C/C++ tra cui:

- SQLite: database interno molto veloce.
- OpenGL: libreria per il rendering di elementi 2D e 3D.
- SSL: Libreria per fornire una comunicazione sicura con altri dispositivi.
- Webkit: Libreria open-source usata per i browsers.
- MediaFramework: Libreria che include il supporto ai codec audio e video

- Freetype: Font engine
- Libc: Librerie standard per C/C++

Android include anche una serie di librerie runtime di base che forniscono la maggior parte delle funzionalità del linguaggio di programmazione Java, comprese alcune caratteristiche del linguaggio Java 8, che il framework Java API utilizza. Questo framework include i blocchi con cui uno sviluppatore può creare un'applicazione Android includendo:

- View System: include componenti grafici quali pulsanti, liste, menu ..
- Resource Manager: fornisce l'accesso a risorse quali stringhe, files multimediali e files di layout.
- Activity manager: gestisce il ciclo di esecuzione delle applicazioni.
- Content Providers: permettono all'applicazione l'accesso ai dati di altre app quali contatti oppure di condividere i propri dati.

Le applicazioni di terze parti e le applicazioni di sistema di default quali e-mail, SMS, calendario, telefono, browser, si basano su questo framework.[\[And, Thoa\]](#)

### 3.3 Linguaggi ed ambiente di sviluppo Android

I linguaggi *ufficiali* per lo sviluppo di applicazioni Android sono due: Java e Kotlin. Java è un linguaggio di programmazione orientato ad oggetti semi-compilato creato a metà degli anni 90 da James Gosling della Sun Microsystems il cui scopo era creare un linguaggio veloce ma soprattutto *indipendente dalla piattaforma*, caratteristica quest'ultima che lo contraddistingue rispetto al C/C++ e che ne determinato il successo infatti a maggio 2022, secondo il TIOBE index, Java risulta il terzo linguaggio più utilizzato e diffuso. Android non utilizza la Java Virtual machine, il codice è compilato in bytecode Dalvik salvato in file .dex per poi essere interpretato ed eseguito sulla macchina.

#### Android studio

Android studio è un software per lo sviluppo di applicazioni Android gratuito, disponibile per i sistemi operativi Windows, MacOS e GNU/Linux. Rilasciato il 16 marzo 2013 durante la *Google I/O Conference* in fase beta, nel dicembre 2014 è rilasciata la versione stabile 1.0 rimpiazzando di fatto il precedente ambiente di sviluppo usato ovvero Eclipse con un plugin specifico per Android e diventando di fatto uno strumento indispensabile per gli sviluppatori.[\[Eme\]](#) In Android studio è possibile creare activity facilmente, gestire le risorse e i permessi dell'applicazione, creare l'interfaccia di activity e componenti sia in maniera visuale mediante un'interfaccia semplice ed intuitiva che editando file xml. In aggiunta è disponibile anche un emulatore virtuale con oppure attivando la modalità sviluppatore dal dispositivo, è possibile collegarlo per testare direttamente l'applicazione. Android studio utilizza Gradle: uno strumento di automazione della compilazione.

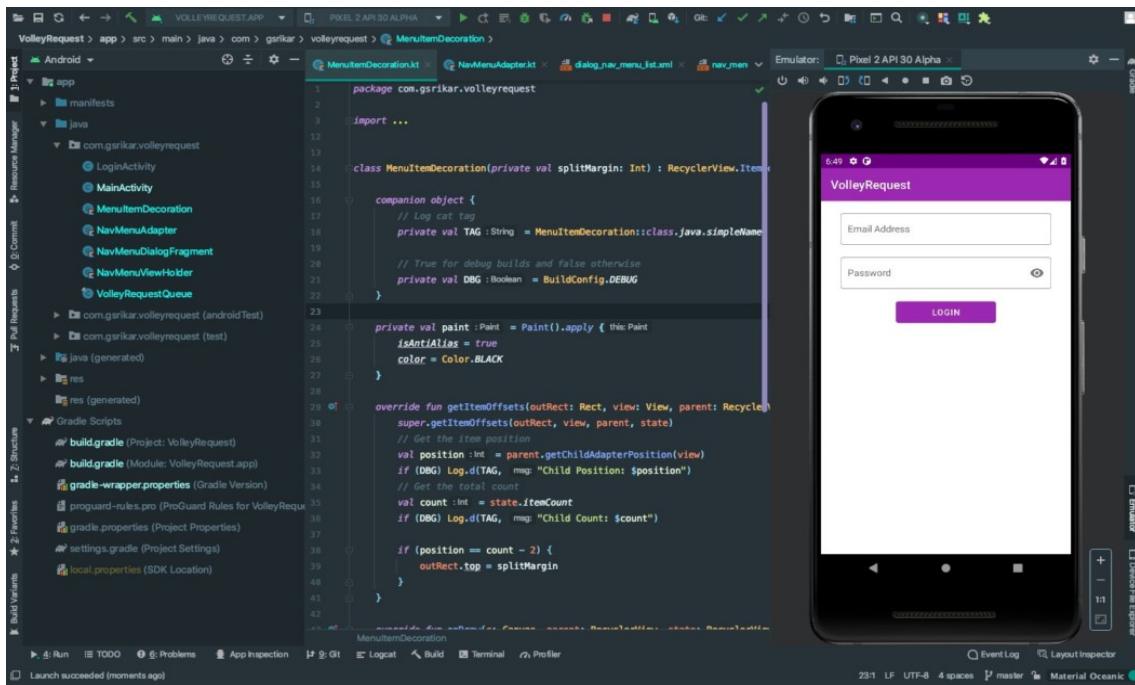


Figura 34: Android studio

## Gradle

Gradle è uno strumento di automazione della compilazione (build automation tool) che fornisce un linguaggio specifico di dominio (DSL) basato su Groovy. Viene eseguito su JVM, oltre a consentire il supporto per la creazione di progetti nativi, consente di utilizzare repository Maven, Ivy o altri definiti dall'utente per la gestione delle dipendenze, inoltre le funzionalità di Gradle possono poi essere estese grazie alla libreria di plug-in nativi e di terze parti. Gradle nasce 2007 con lo scopo di fornire un build-system semplice ed intuitivo per gli sviluppatori i quali spesso devono combattere con strumenti inefficienti e spesso integrati male. Gradle offre una serie di funzionalità tra cui:

- Script Groovy
- Convenzioni flessibili
- Build scalabili
- Gestione delle dipendenze robusto e potente
- Linguaggio di build espressivo
- Integrazione con altri build tools

La caratteristica principale di Gradle è Groovy, un linguaggio di programmazione basato su Java con cui è possibile definire le dipendenze e le librerie presenti nel progetto, ogni elemento nello script Gradle ha una rappresentazione uno ad uno in Java. È possibile integrare altri build tools già presenti quali Apache Ant, Maaven basati su files xml mediante una classe che li converte nel linguaggio specifico di dominio.[Mus14]

Mediante Gradle è stato possibile aggiungere facilmente librerie esterne utilizzate nelle funzionalità implementate quali: *Jackson* per la lettura di file Json e *Osmdroid* per la visualizzazione delle mappe.

### **Firebase**

I dati dell'applicazione Helena sono presenti in un database Firebase, un database NoSQL distribuito offerto da Google ed integrabile con applicazioni web e mobile. I dati in firebase sono memorizzati in formato Json (JavaScript Object Notation) ed ogni cambiamento dei dati è riportato mediante un'operazione sincrona sul db ed è possibile organizzarli sfruttando delle regole (di lettura e di scrittura) dette *Firebase Security Rules*.[\[Mor17\]](#)

### **Github**

Github è un servizio web cloud-based che aiuta gli sviluppatori ad archiviare e gestire il codice di progetti e controllare le modifiche apportate. Basato su Git, uno strumento di controllo di versione open-source creato nel 2005 da Linus Torvalds, il controllo delle versioni aiuta gli sviluppatori a tracciare e gestire le modifiche al codice di un progetto software. Man mano che un progetto software cresce, il controllo delle versioni diventa essenziale permettendo agli sviluppatori di lavorare in sicurezza attraverso features tra cui il *branching* e *merging*. [\[Kin\]](#)

### **Python**

Anche se non è presente codice Python nell'applicazione, sono stati utilizzati degli script separati per processare i dati in csv di ospedali e farmacie ed estrarre dei file Json presenti nell'applicazione.

Python è un linguaggio ad alto livello, orientato ad oggetti creato nel 1991 dal programmatore olandese Guido van Rossum, il linguaggio è caratterizzato da una sintassi semplice, un gestore di pacchetti efficiente e migliaia di librerie a disposizione degli sviluppatori.

## **3.4 Organizzazione applicazione Helena**

L'applicazione Helena, come tutte le applicazioni Android, oltre al codice sorgente ha al suo interno una serie di cartelle e files indispensabili per il suo funzionamento.

- **AndroidManifest.xml:** in questo file sono dichiarate tutte le componenti dell'applicazione, i servizi, i ricevitori di broadcast e i fornitori di contenuti. Inoltre sono presenti i permessi i permessi di cui necessita l'app (connessione ad internet, GPS ..) e le features hardware e software richieste.
- **build.gradle:** sono presenti due files di questo tipo, hanno lo scopo di definire le dipendenze e le librerie utilizzate per creare la build dell'applicazione
- **Cartella /res:** Presenti le risorse quali: animazioni, colori definiti dall'utente, stringhe, valori, layout e menu
- **Cartella /assets:** Contiene files testuali/json/csv accessibili dalle activity.
- **Cartella /java:** Contiene il codice sorgente in Java dell'applicazione

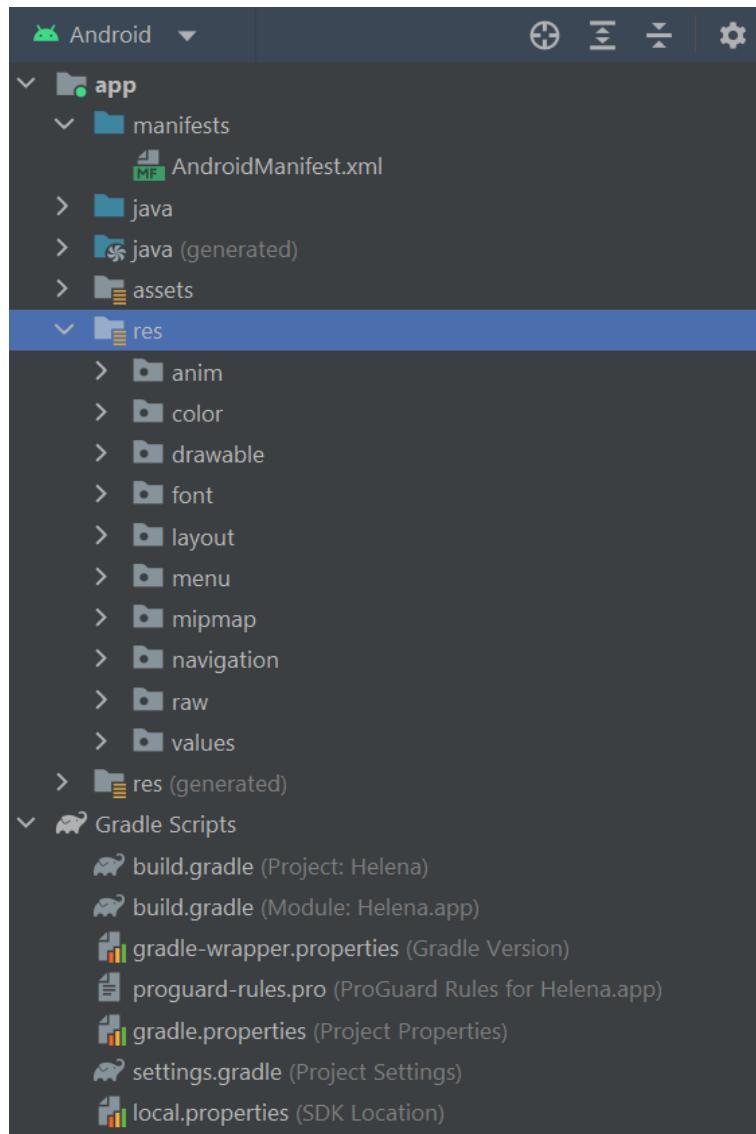


Figura 35: Organizzazione applicazione

Il codice dell'applicazione è organizzato in diversi package, ciascuno contenente funzionalità specifiche dell'applicazione. In dettaglio i package creati sono i seguenti:

- Package *Ospedalizzazioni*
- Package *Ospedali*
- Package *Farmacie*
- Package *SaluteMentale*

### 3.5 Progettazione applicazione Helena

Fino a questo punto abbiamo discusso delle tecnologie utilizzate nell'implementazione delle nuove funzionalità dell'applicazione e dell'importanza che hanno tutt'ora le applicazioni in ambito e-health diventante uno strumento imprescindibile non solo per gli utenti ma anche per il personale medico e sanitario. La fase di progettazione ha posto particolare attenzione su questo aspetto per certi versi carente nell'applicazione, le principali funzionalità presenti sono:

- Terapie
- Suggerimento medico
- Dizionario medico
- Chatbot
- Diario alimentare
- Profilazione utente
- Monitoraggio

In particolare nella sezione monitoraggio sono presenti quattro distinti monitoraggi:

1. Peso
2. Glucosio
3. Saturazione ossigeno
4. Documenti clinici

inoltre, da un mio collega, è stata aggiunta la sezione *Fitbit* con cui è possibile visualizzare i dati presi da un dispositivo wearable Fitbit dell'utente. Tuttavia l'utente non ha la possibilità di inserire un ricovero o un intervento presso una struttura ospedaliera associando uno o più documenti clinici ad un determinato ricovero.

#### Progettazione sezione Ospedalizzazioni e Ospedali

In questa sezione l'utente può inserire, visualizzare, modificare ed eliminare le ospedalizzazioni effettuando tutte le operazioni previste dalla *CRUD* (Create, Read, Update, Delete) nell'apposita sezione del database Firebase. Oltre a queste operazioni base l'utente può visualizzare l'ospedale associato al ricovero sulla mappa e i documenti clinici associati. La sezione Ospedalizzazione è legata a *doppio filo* con la sezione **Ospedali** che permette all'utente di inserire un nuovo ospedale nel database, visualizzare tutti gli ospedali italiani ordinandoli in base alla distanza dalla posizione dell'utente e alla città di residenza. In questo modo l'utente mentre sta inserendo i dati dell'ospedalizzazione, può creare un nuovo ospedale da associare all'ospedalizzazione che sta creando oppure selezionarne uno da una apposita sezione. La struttura della sezione Ospedalizzazioni è la seguente:

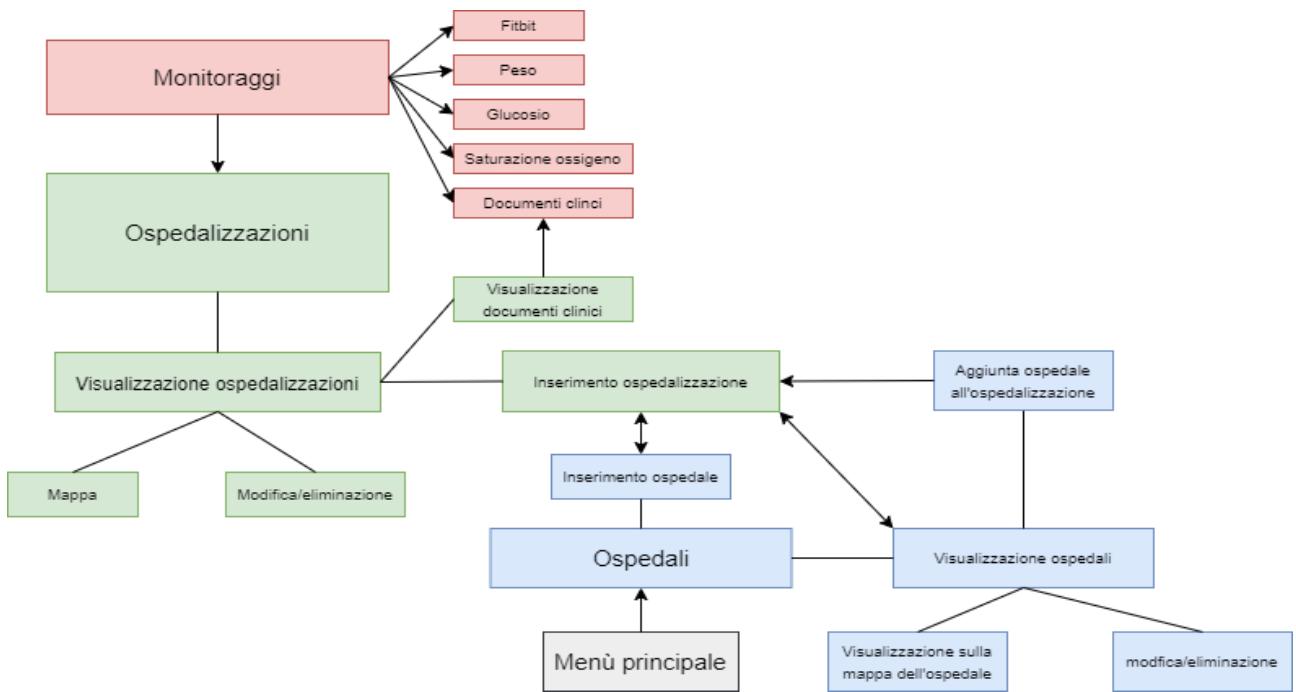


Figura 36: Struttura sezione monitoraggi

Nella sezione *Visualizzazione Ospedalizzazioni* è possibile effettuare una ricerca per nome dell'ospedale, per città dell'ospedale e per data di inserimento dell'ospedalizzazione mentre nella visualizzazione degli ospedali è possibile ricercarli per nome e per città. La navigazione tra le sezioni avviene mediante bottoni ben visibili.

Come prima cosa, è stato definito l'inserimento dell'ospedalizzazione che comprende informazioni quali:

- Data inizio ricovero
- Data fine ricovero
- Causa dell'ospedalizzazione
- Sintomi dell'utente
- Ospedale associato all'ospedalizzazione
- Reparto
- Nome del medico
- Eventuali note

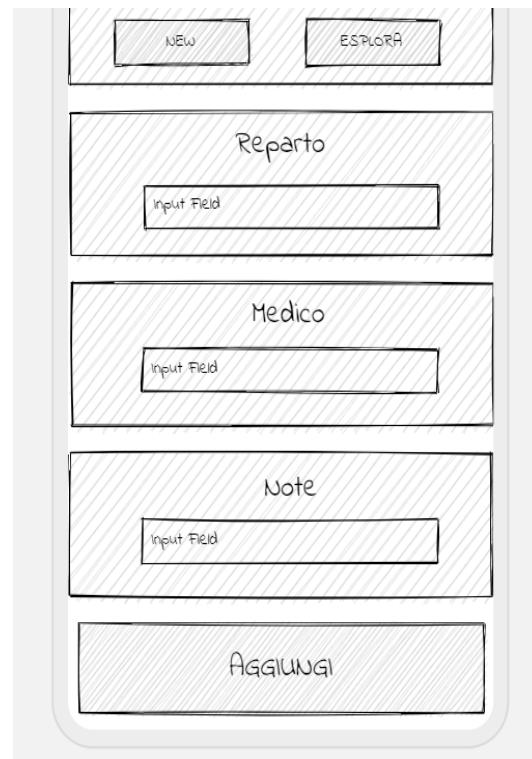
per ciascuno di questi campi vi è una apposita card in cui l'utente inserisce le informazioni relative.



The prototype shows a vertical stack of five input fields:

- Inizio Ricovero**: Date field containing "21 luglio 2022".
- Fine ricovero**: Date field containing "23 luglio 2022".
- Causa ricovero**: Input Field.
- Sintomi**: Dropdown menu.
- Struttura**: Input Field with "NEW" and "ESPLORA" buttons below it.

Figura 37: Prototipo inserimento ospedalizzazioni



The prototype shows a vertical stack of four input fields:

- Reparto**: Input Field.
- Medico**: Input Field.
- Note**: Input Field.
- AGGIUNGI**: Large button at the bottom.

Definita l'interfaccia di inserimento, definisco l'interfaccia relativa alla visualizzazione in cui non solo dovranno essere mostrate le informazioni di tutte le ospedalizzazioni ma anche i comandi per la modifica, eliminazione, ricerca, visualizzazione sulla mappa.



Figura 38: Prototipo visualizzazione ospedalizzazioni

Per quanto riguarda la sezione ospedali l'utente può visualizzare sia gli ospedali presi da una sorgente esterna (file, api ecc..) che gli ospedali inseriti mediante un'apposita interfaccia contenente i seguenti campi:

- Nome della struttura ospedaliera
- Città dell'ospedale
- Indirizzo dell'ospedale

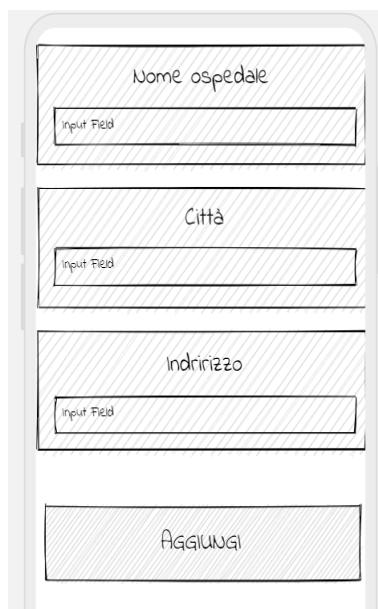


Figura 39: Prototipo inserimento ospedale

L'interfaccia di visualizzazione degli ospedali deve includere tutte le informazioni presenti relative agli ospedali con i comandi di modifica, eliminazione, visualizzazione sulla mappa, ricerca su internet, ordinamento in base alla posizione e ricerca per nome e città.

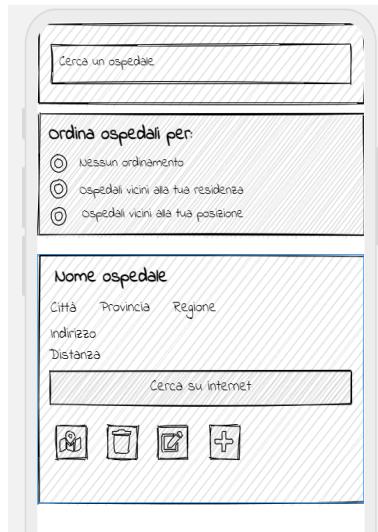


Figura 40: Prototipo visualizzazione ospedali

### Progettazione sezione Temperatura Corporea

Una sezione incompleta nei monitoraggi è la sezione **Temperatura corporea** in cui l'utente può inserire manualmente e visualizzare le temperature inserite, ad ogni misurazione sono associate le seguenti informazioni:

- Valore della temperatura
- Data della misurazione
- Orario della misurazione
- Eventuale descrizione

La sezione stata riprogettata ed è così strutturata:

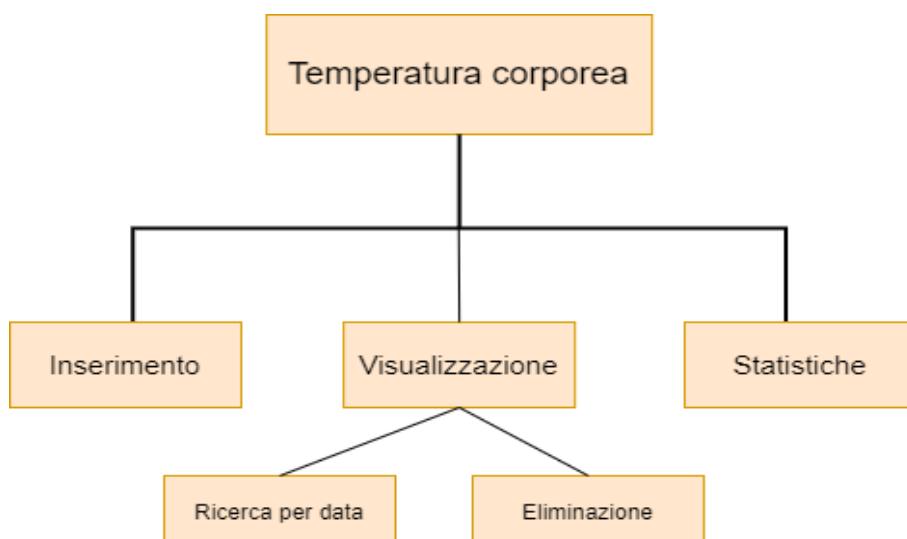


Figura 41: Struttura sezione Temperatura corporea

L'utente può passare velocemente da una sezione all'altra della sezione mediante un menù posto nella parte inferiore dello schermo



Figura 42: Menù sezione Temperatura corporea

La prima schermata visualizzata è la seguente per l'inserimento della temperatura

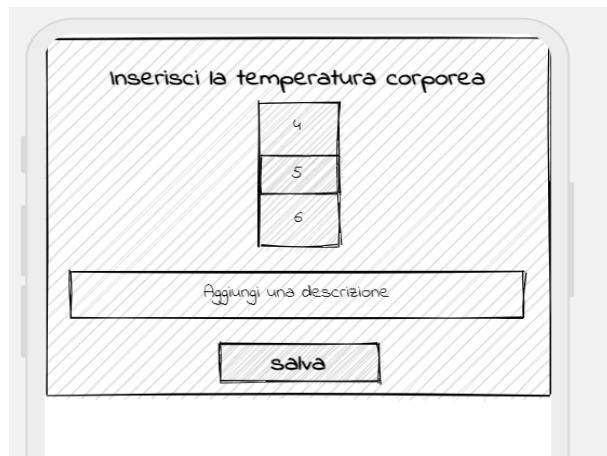


Figura 43: Layout inserimento misurazione

Tutte le misurazioni inserite sono visualizzabili nella seguente schermata:



Figura 44: Layout visualizzazione misurazioni

Nell'ultima schermata è possibile visualizzare in un grafico i valori delle misurazioni inseriti nel tempo.

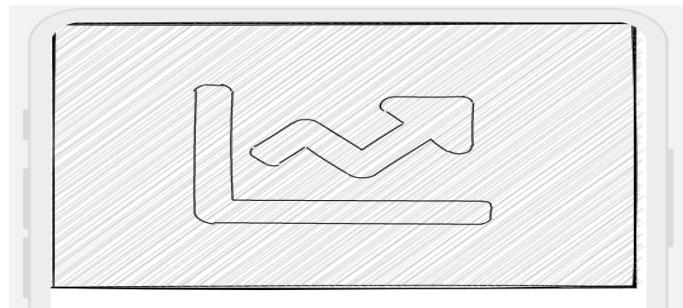


Figura 45: Layout visualizzazione grafico

### Progettazione sezione Farmacie

L'utente in questa sezione, accessibile dalla schermata principale dell'applicazione, può visualizzare tutte le farmacie e le parafarmacie italiane presenti, ordinandole per distanza dal punto in cui si trova e in base alla città di residenza, visualizzando sulla mappa tutte la farmacie/parafarmacie presenti in una data città inoltre l'utente può inserire una nuova farmacia/parafarmacia. La struttura della sezione farmacie è la seguente:

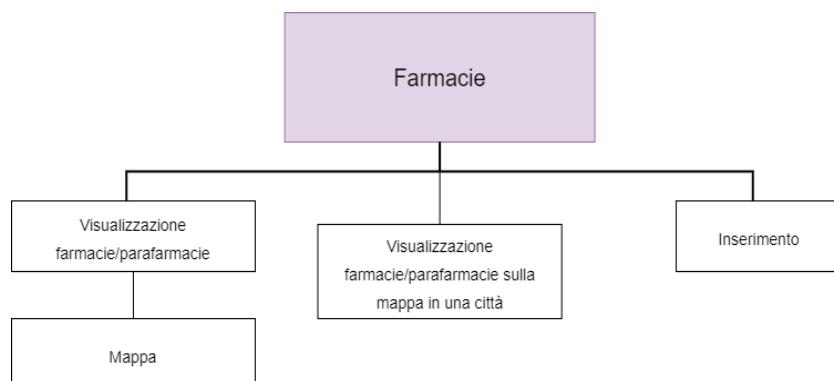


Figura 46: Struttura sezione Farmacie

L'utente può passare da una funzionalità all'altra della sezione mediante un menù posto nella parte inferiore



Figura 47: Menù inferiore Farmacie

Per ogni farmacia/parafarmacia sono visualizzate nel seguente layout informazioni quali:

- Nome
- Tipo (farmacia o parafarmacia)
- Città, Provincia, Regione

- Indirizzo
- Distanza



Figura 48: Layout visualizzazione farmacie

L'utente può visualizzare tutte le farmacie su una mappa inserendo il nome della città secondo il seguente layout:

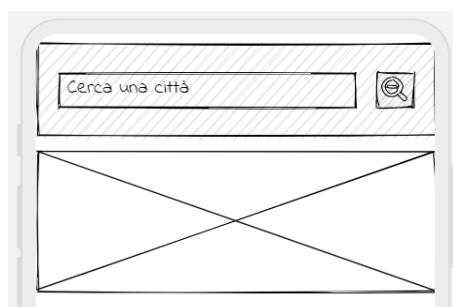


Figura 49: Layout visualizzazione farmacie sulla mappa

Infine vi è la funzionalità di inserimento con cui l'utente può inserire una nuova farmacie/parafarmacie che verrà visualizzata.

The layout for inserting a pharmacy consists of five stacked sections within a rounded rectangle frame:

- Nome**: An input field labeled "Input Field".
- Tipo**: A section with two radio button icons, each containing a square divided by a diagonal line. Below each icon is a label: "Farmacia" under the first and "Parafarmacia" under the second.
- Città**: An input field labeled "Input Field".
- Indirizzo**: An input field labeled "Input Field".
- INSERISCI**: A large rectangular button at the bottom.

Figura 50: Layout inserimento farmacie

### Progettazione sezione Stato Emotivo

In questa sezione l'utente può inserire il proprio stato emotivo inserendo informazioni quali:

- Titolo
- Emozione provata
- Intensità dell'emozione
- Diario

La sezione stato emotivo ha la seguente struttura:

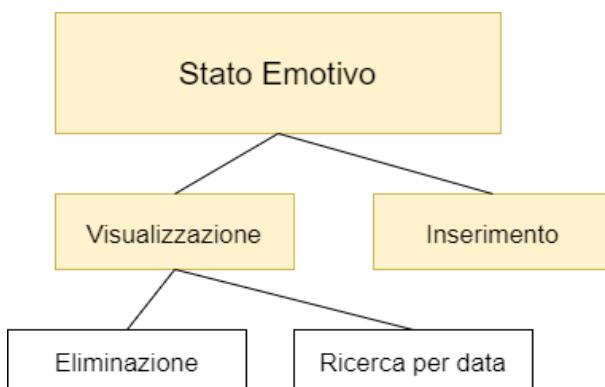


Figura 51: Struttura sezione Stato Emotivo

L'inserimento dello stato emotivo avviene nel seguente layout:

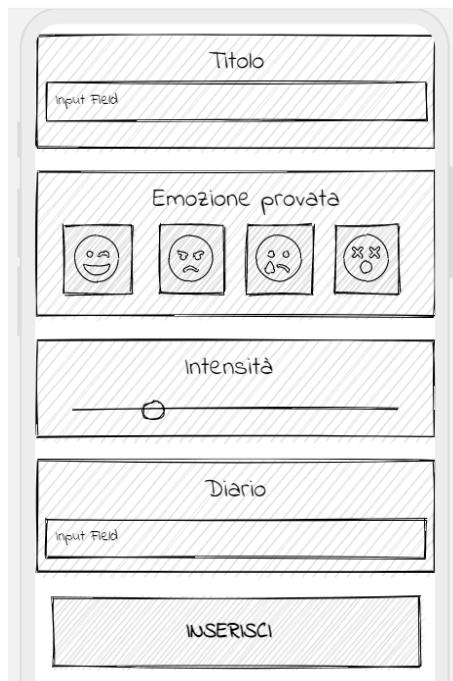


Figura 52: Layout inserimento stato emotivo

La visualizzazione e la ricerca dei vari stati emotivi inseriti dall'utente avviene in questa schermata con il seguente layout:



Figura 53: Layout visualizzazione stati emotivi

## 4 Implementazione

### 4.1 Bugfix

La fase di bugfix è stata una fase preliminare allo sviluppo delle nuove funzionalità, questa fase è stata utile innanzitutto per prendere familiarità con l'ambiente di sviluppo e con il codice sorgente dell'applicazione ma soprattutto per risolvere problemi, soprattutto relativi al layout, presenti nell'applicazione. Sono stati effettuati bugfix nelle seguenti activity dell'applicazione:

1. LoginRegisterChoice - (Prima schermata per la registrazione o il login)
2. Water\_Section - Sezione Diario alimentare
3. RicetteSection - Sezione Diario alimentare
4. RecalculateMacro - Sezione Diario Alimentare
5. HomeFoodSection - Sezione Diario Alimentare
6. CreateNewFood - Sezione Diario alimentare
7. EditProfile - Sezione Modifica Profilo

#### 1 - Bugfix LoginRegisterChoice

Il primo bug risolto è stato relativo a questa activity in cui il problema più evidente era il totale disordine degli elementi del layout ponendo il dispositivo in orizzontale (landscape) senza la possibilità di scrollare.

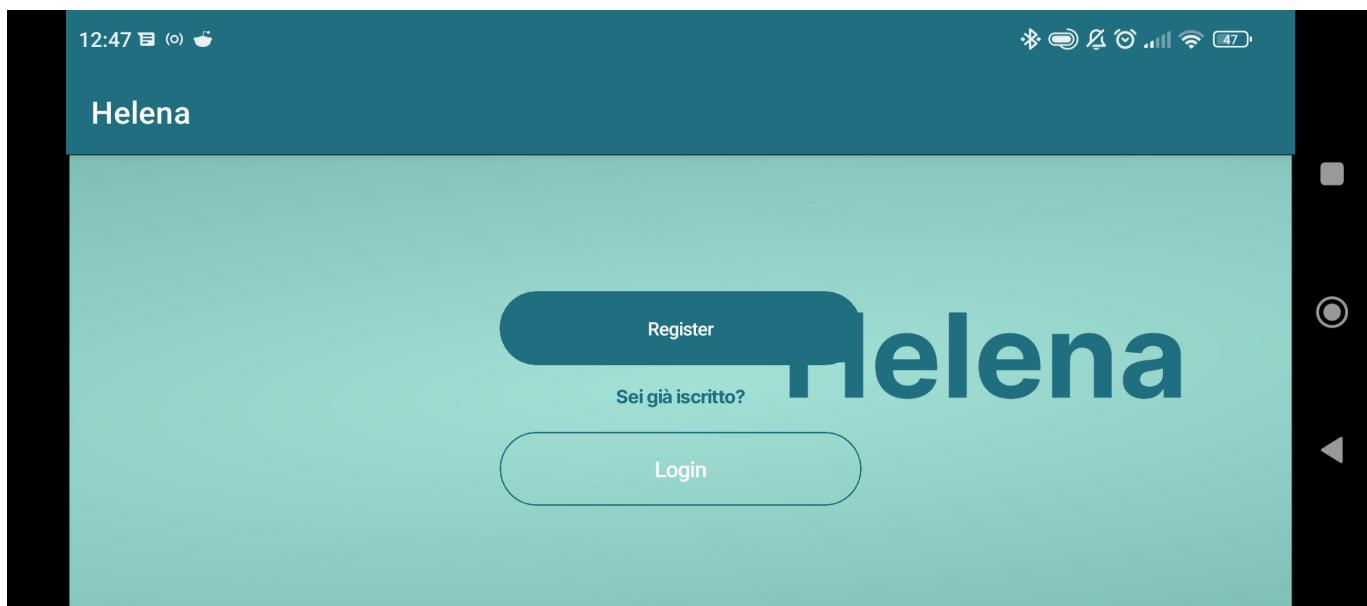


Figura 54: Activity LoginRegisterChoice prima del bugfix

Il layout è stato ordinato e reso scrollabile usando una *ScrollView* ovvero un componente che permette di posizionare gli elementi in modo gerarchico offrendo la possibilità di scrollarli sullo

schermo. Inoltre sono state usate delle *guidelines* nel layout per centrare gli elementi sullo schermo qualsiasi fosse la sua larghezza, infine sono stati inseriti i loghi relativi ad *AsterIA* e alla regione Puglia. Il risultato finale è il seguente:

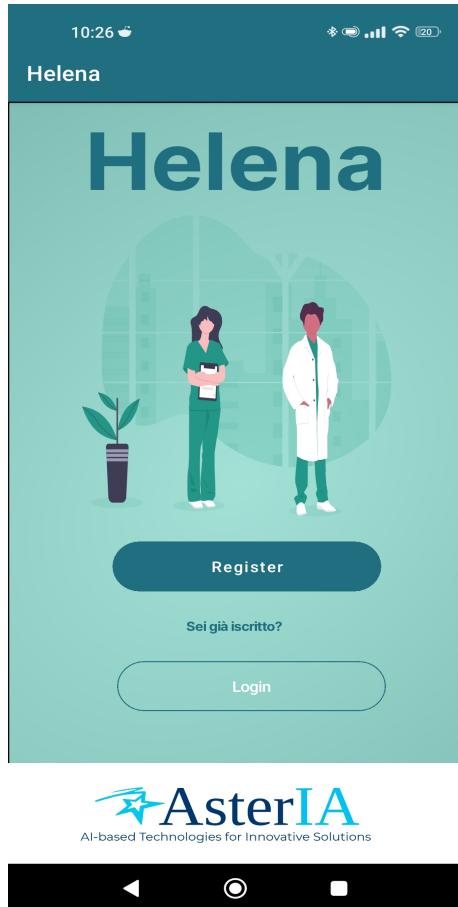


Figura 55: Activity LoginRegisterChoice Portrait

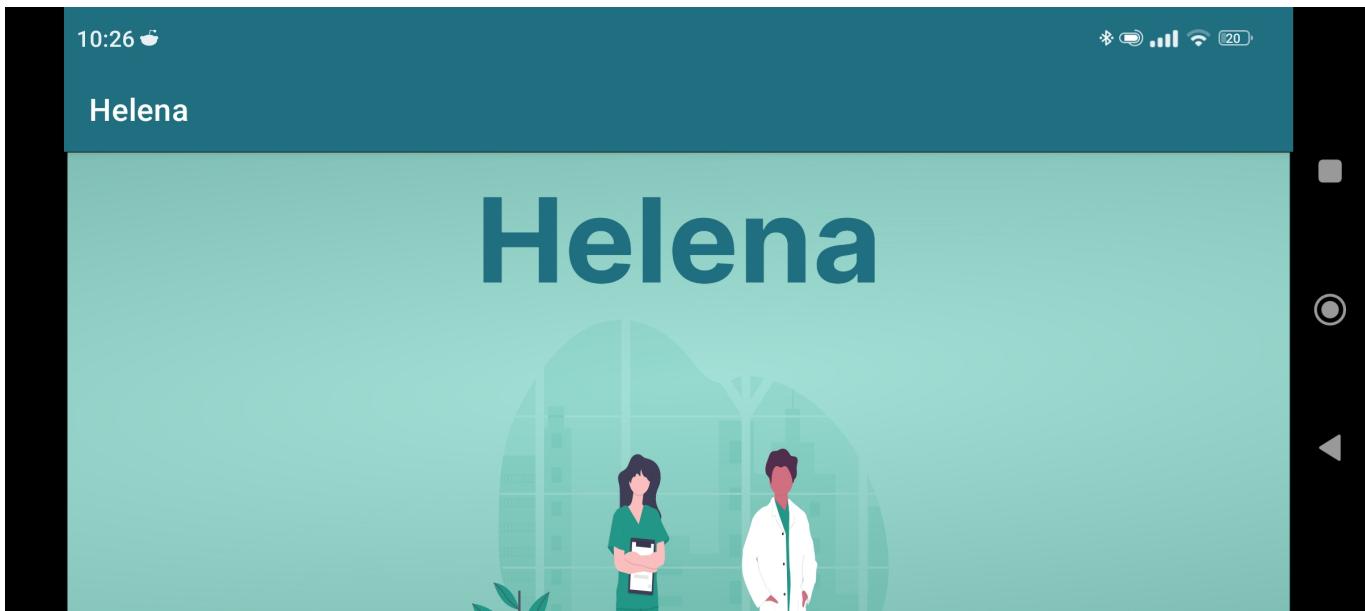


Figura 56: Activity LoginRegisterChoice Landascape (parte superiore)

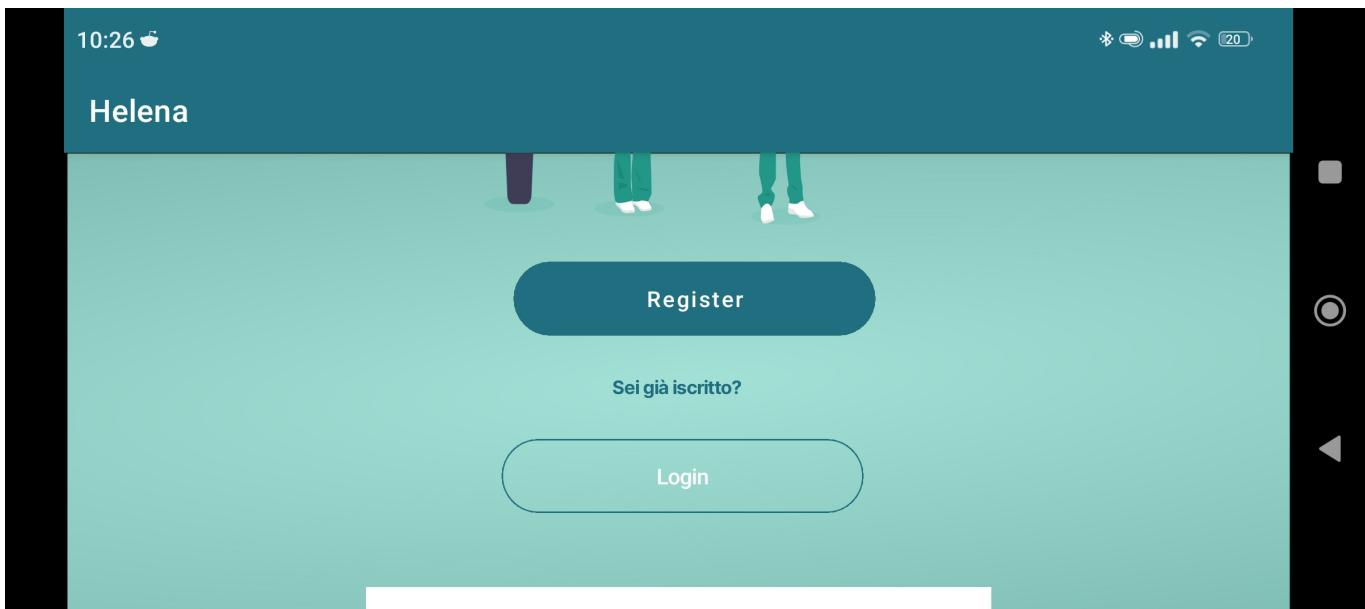


Figura 57: Activity LoginRegisterChoice Landascape (parte inferiore)

## 2 - Bugfix Water\_Section

Questa activity consente all’utente di impostare un fabbisogno giornaliero di acqua e tenere traccia dei bicchieri di acqua bevuti durante la giornata. Anche in questa activity era presente un problema analogo all’activity precedente, gli elementi del layout ponendo il dispositivo in orizzontale non mantenevano la posizione originaria rendendo di fatto la funzionalità inutilizzabile.

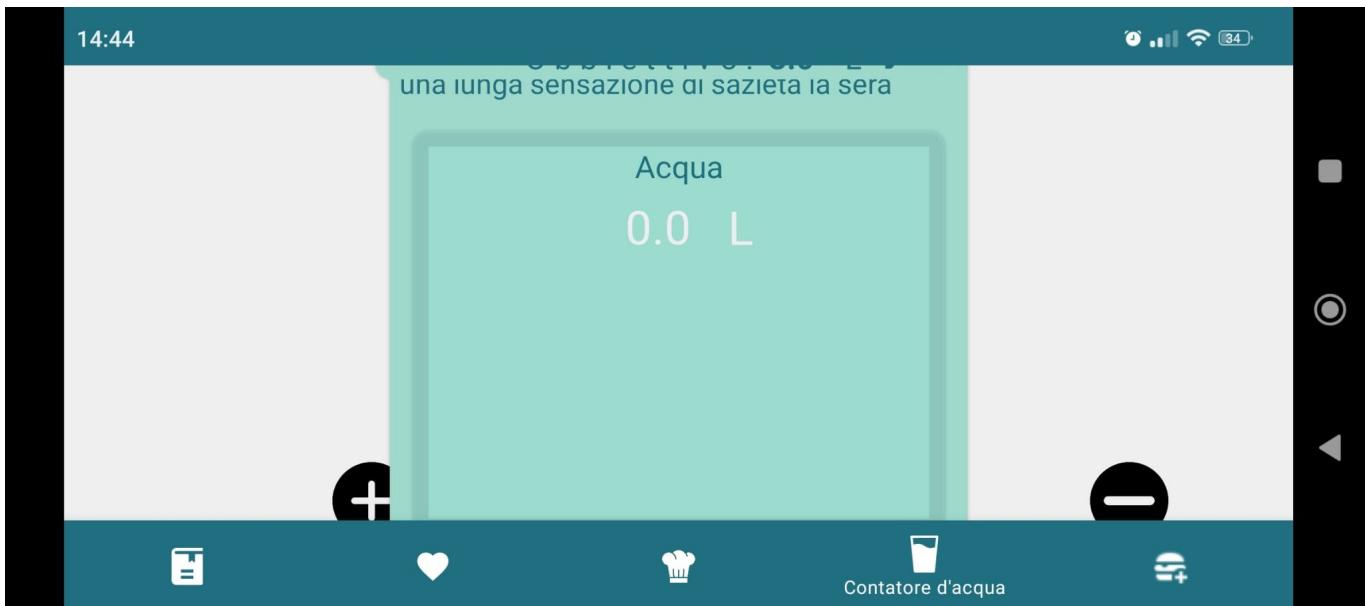


Figura 58: Water\_Section prima del bugfix

Il layout è stato reso scrollabile mediante la *ScrollView* inoltre è stato modificato il design dei buttoni di aggiunta ed eliminazione di bicchieri di acqua rendendoli conformi allo stile dell'activity.

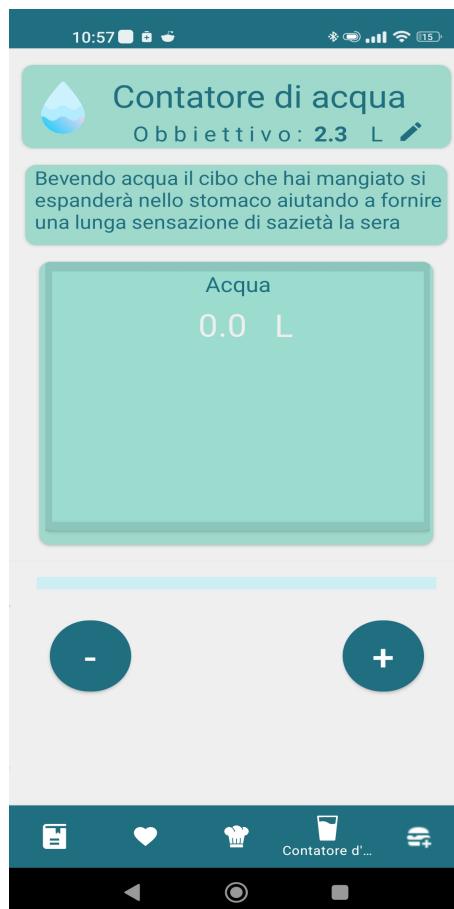


Figura 59: Activity Water\_Section Portrait

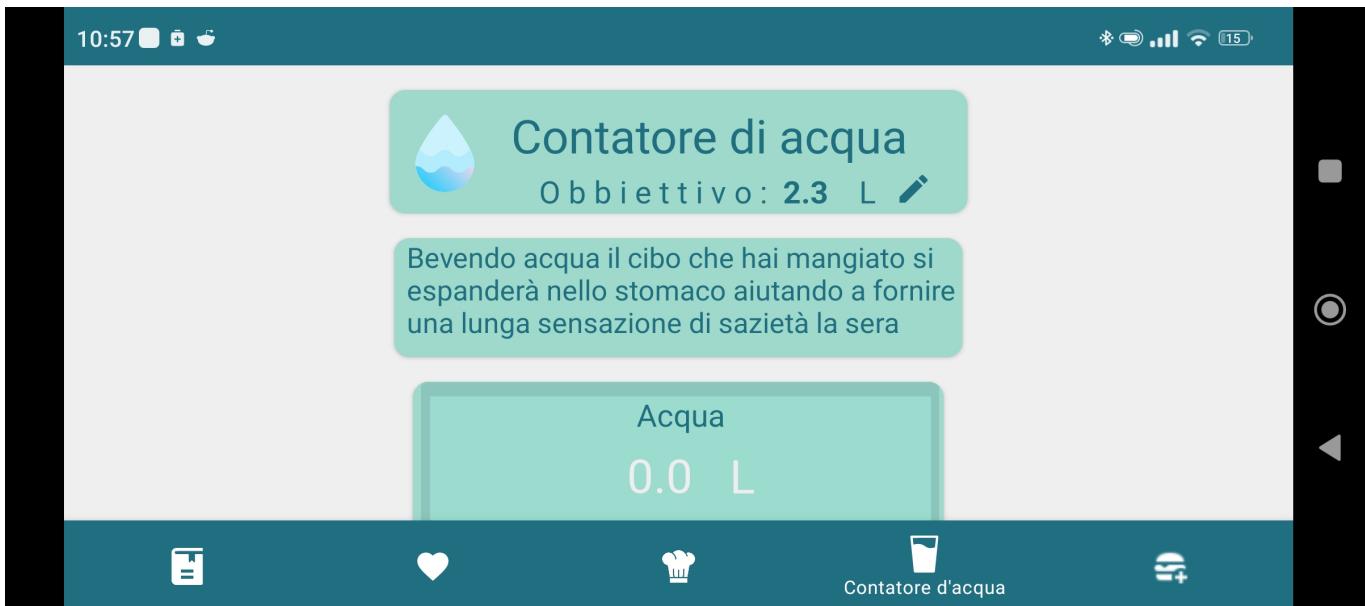


Figura 60: Activity Water\_Section Landascape (parte superiore)

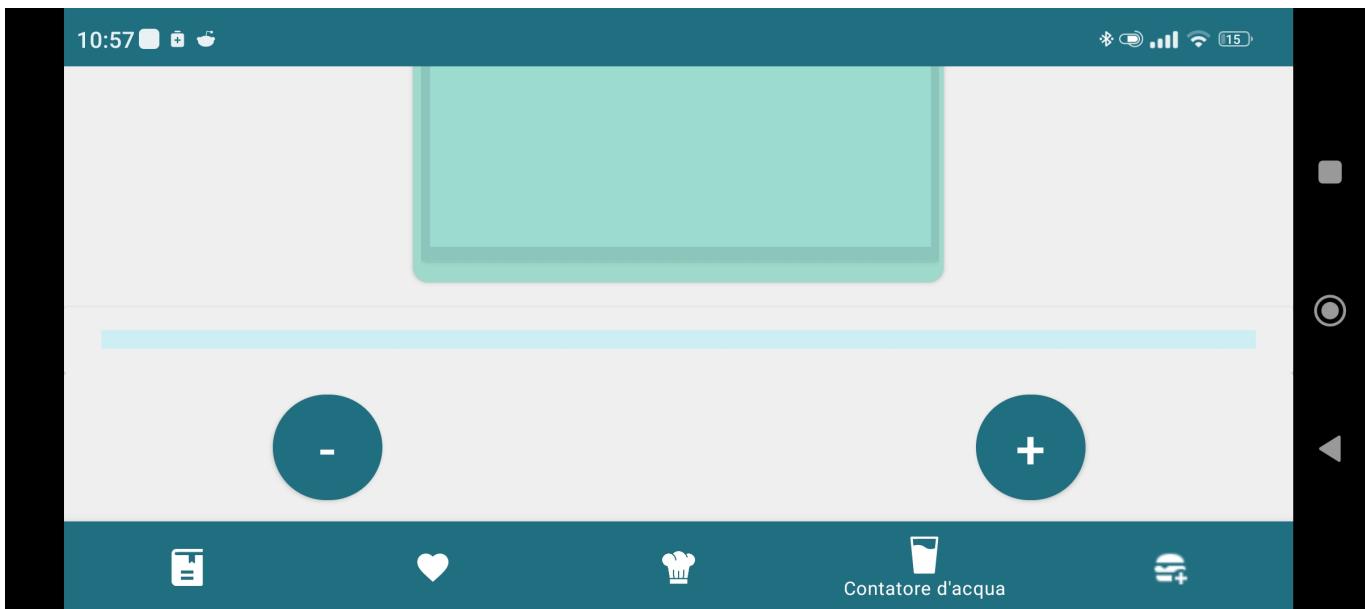


Figura 61: Activity Water\_Section Landascape (parte inferiore)

### 3 - Bugfix RicetteSection

Questa activity permette all’utente la creazione di ricette inserendo gli elementi già presenti nel database. Anche in questa activity il problema principale è dato dalla sovrapposizione delle card nella modalità *landscape* del dispositivo.

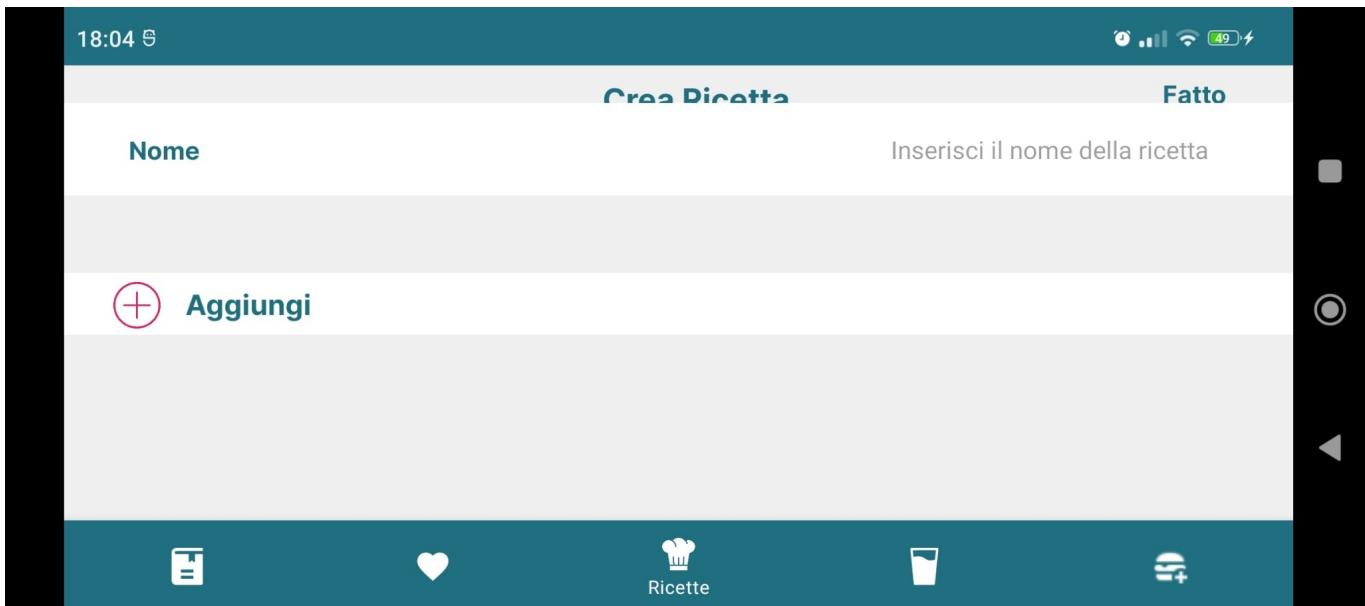


Figura 62: Activity RicetteSection prima del bugfix

Nell'activity è stata introdotta la *ScrollView*, sono stati impostati dei margini fissi tra le cards e sono stati modificati alcuni elementi per renderli più accessibili.

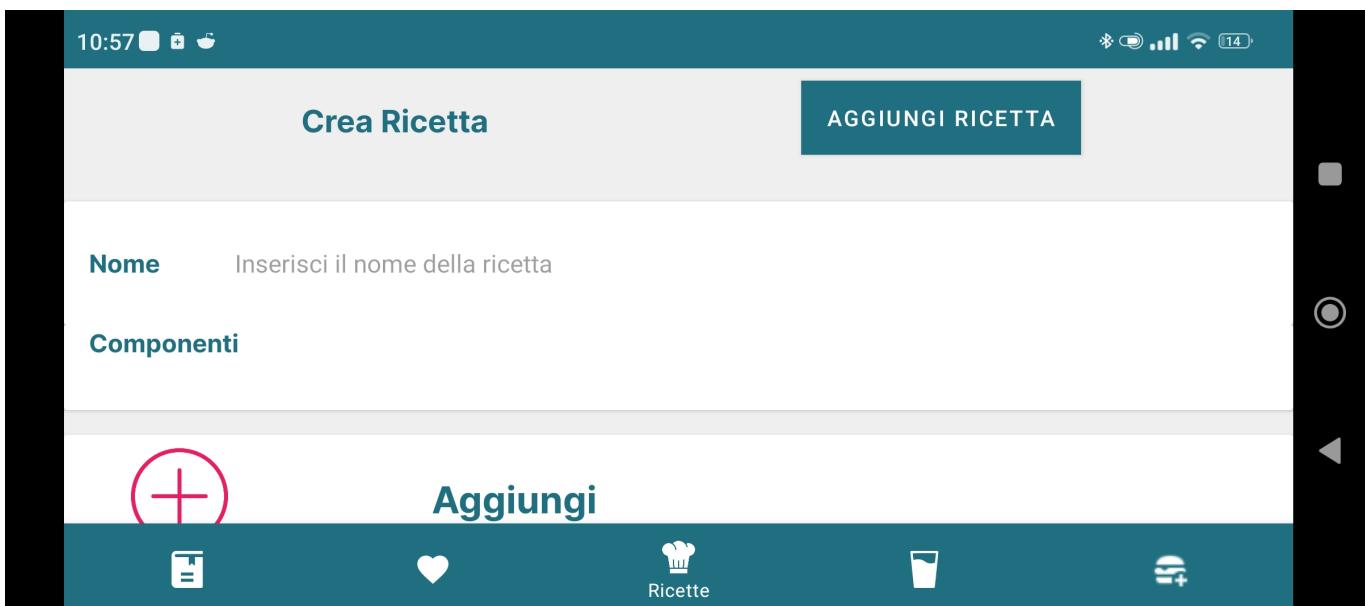


Figura 63: Activity RicetteSection

#### 4 - Bugfix RecalculateMacro

L'activity offre la possibilità all'utente di calcolare il fabbisogno calorico giornaliero, prendendo in input dati quali:

- Peso
- Età

- Sesso
- Obiettivo dell'utente (Perdita di peso, mantenimento del peso, aumento di peso)
- Livello di attività fisica (nessuno, leggero, moderato, intenso, molto intenso)

L'activity non solo non era scrollabile ma obbligava l'utente ad inserire nuovamente i dati ogni volta che accedeva l'activity. Per risolvere il primo problema, come nelle activity precedenti, è stata utilizzata la *ScrollView* nel layout mentre per il secondo problema sono state aggiunte delle funzioni al codice dell'activity. Tramite l'interfaccia *SharedPreferences* i dati inseriti per la prima volta dall'utente sono memorizzati in automatico in un file xml presente nello storage interno (*shared\_prefs.xml*) e sono accessibili a tutte le activity. Ogni volta che l'activity è avviata, se nelle *SharedPreferences* sono presenti dati relativi all'activity, sono caricati in automatico nell'activity.

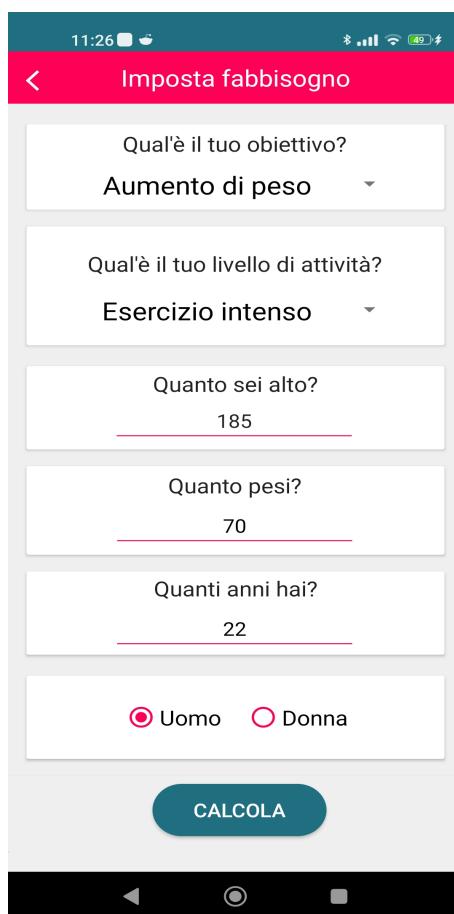


Figura 64: Activity RecalculateMacro

```
SharedPreferences sharedpreferences =
    this.getSharedPreferences("com.dev.helena.FoodSection", Context.MODE_PRIVATE);

sharedPreferences.edit().putString("PESO", peso.getText().toString()).apply();

sharedPreferences.edit().putString("ETA", eta.getText().toString()).apply();

sharedPreferences.edit().putString("ALTEZZA", altezza.getText().toString()).apply();
```

---

Codice 1: Modifica Shared Preferences

### 5 - Bugfix HomeFoodSection

L'activity HomeFoodSection è l'activity principale della sezione *Diario Alimentare* in cui è possibile visualizzare i pasti, calcolare il fabbisogno giornaliero ed accedere a funzionalità per la creazione di ricette, l'inserimento del fabbisogno di acqua, la creazione di alimenti, l'inserimento di pasti. L'activity, tuttavia, diventa quasi inutilizzabile ponendo il dispositivo in orizzontale in quanto vi è la sovrapposizione delle diverse card che compongono il layout.

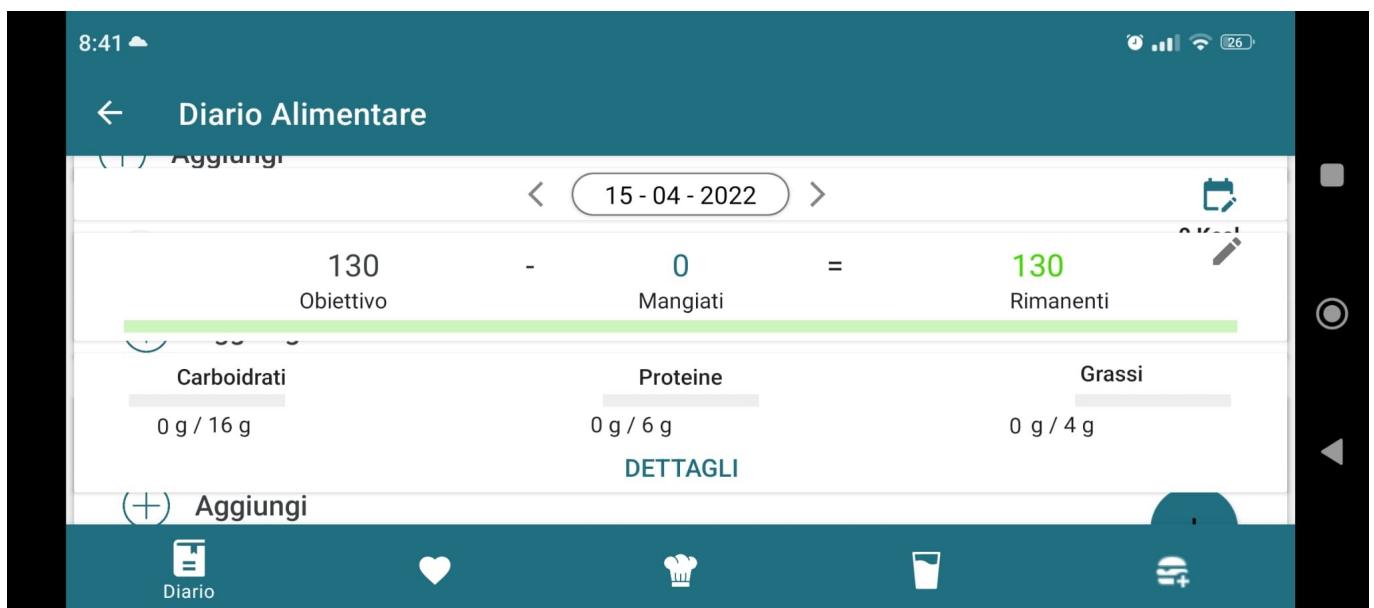


Figura 65: Activity HomeFoodSection prima del bugfix

Un altro problema piuttosto fastidioso era la sovrapposizione delle voci dei pasti cliccando il *FloatingActionButton* presente nell'activity.



Figura 66: Activity HomeFoodSection FloatingActionButton

Per risolvere questi problemi sono stati imposti dei vincoli rigidi tra le varie card in modo da non farle sovrapporre inoltre il FloatingActionButton è stato posto in una posizione in cui non risulta più parzialmente nascosto dal menù sottostante.



Figura 67: Activity HomeFoodSection FloatingActionButton

#### 6 - Bugfix CreateNewFood

La sezione CreateNewFood consente la creazione di nuovi alimenti da aggiungere al database Firebase memorizzati nella sezione *Lista\_alimenti*. Anche in questa activity, ponendo il dispositivo in orizzontale non era possibile scrollare i vari elementi perciò è stata inserita una *ScrollView* che ha risolto il problema.

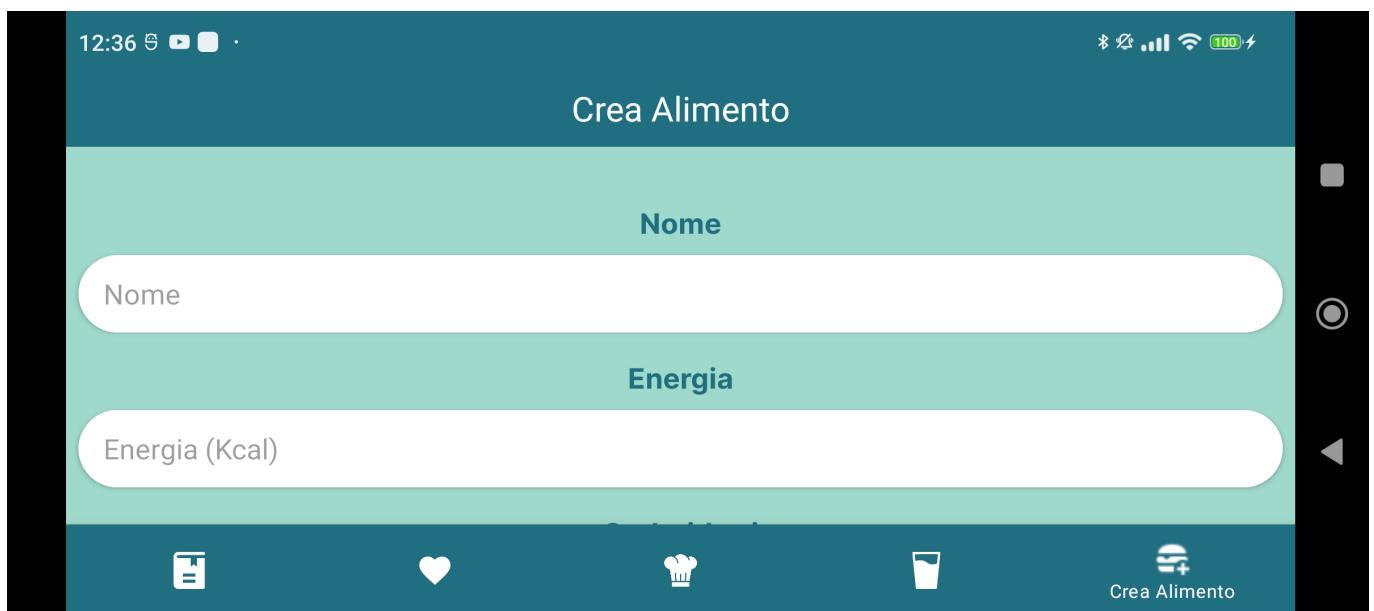


Figura 68: Activity CreateNewFood

#### 7 - Bugfix EditProfile

L'activity consente all'utente di modificare le informazioni relative al proprio profilo memorizzate nel database quali:

- Nome

- Cognome
- E-mail
- Sesso
- Numero di telefono
- Codice Fiscale
- Città di residenza
- Data di nascita

La modifica effettuata riguarda il campo *Città di residenza* in cui grazie ai dati dei comuni italiani presi dal file *comuni.json* presente nella cartella */assets* è stato inserito l'autocompletamento.

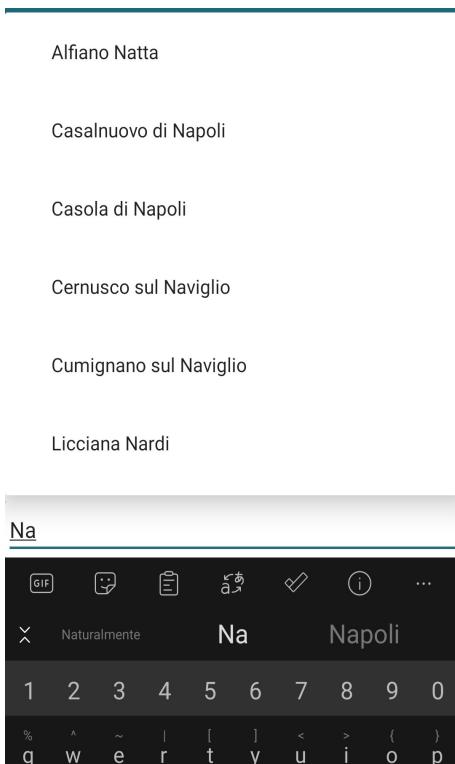


Figura 69: Auto-completamento del comune di residenza

L'informazione del comune di residenza dell'utente è molto importante per le nuove funzionalità implementate in quanto consente il suggerimento di ospedali e farmacie.

## 4.2 Sviluppo sezione Temperatura corporea

Una sezione presente nell'applicazione ma non completata è stata la sezione Temperatura corporea, sezione tramite la quale l'utente può inserire manualmente i valori della propria temperatura, visualizzando le misurazioni in un apposito grafico.



Figura 70: Sezione monitoraggi

L’activity *BodyTemperatureActivity* contiene tre distinti fragment, ciascuno con una funzionalità specifica:

1. *BodyTemperatureInputFragment* per l’inserimento della temperatura corporea.
2. *BodyTemperatureFramgent* per la visualizzazione delle temperature inserite.
3. *BodyTemperatureStatisticsFramgent* per la visualizzazione del grafico delle misurazioni inserite.

Mediante un menù presente nella parte inferiore è possibile passare velocemente da un fragment all’altro.

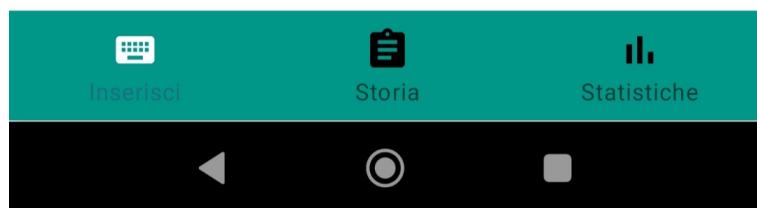


Figura 71: Menù inferiore

---

```

public class BodyTemperatureActivity extends AppCompatActivity {

    private BottomNavigationView bottomNavigationView;
    private FirebaseDatabase database;
    private DatabaseReference dbRef;

    private BodyTemperatureInputFragment fragment1 = new
        BodyTemperatureInputFragment();
    private BodyTemperatureFragment fragment2 = new BodyTemperatureFragment();
    private BodyTemperatureStatisticsFragment fragment3 = new
        BodyTemperatureStatisticsFragment();

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_body_temperature);

        bottomNavigationView = findViewById(R.id.bottom_navigation_view);
        database = FirebaseDatabase.getInstance();
        dbRef =
            database.getReference("Utenti_misurazioni").child(FirebaseAuth.getInstance()
                .getCurrentUser().getUid()).child("Temperatura");

        bottomNavigationView.setOnItemSelectedListener(item -> {
            switch (item.getItemId()){
                case R.id.inserimento_misurazioni:
                    getSupportFragmentManager().beginTransaction().replace(R.id.tempContainer,
                        fragment1).commit();
                    return true;

                case R.id.storia_misurazioni:
                    getSupportFragmentManager()
                        .beginTransaction().replace(R.id.tempContainer,
                            fragment2).commit();
                    return true;

                case R.id.statistiche_misurazioni:
                    getSupportFragmentManager()
                        .beginTransaction().replace(R.id.tempContainer,
                            fragment3).commit();
                    return true;
            }
            return true;
        });
    }
}
...
}

```

---

Codice 2: BodyTemperatureActivity

Nell'activity è presente il metodo *saveTemperatureDb()* con cui è possibile salvare la tempera-

tura nel database Firebase che è rappresentata dalla classe *BodyTemperature*.

### **Classe BodyTemperature**

La classe rappresenta la temperatura corporea e ha i seguenti attributi:

- key (String)
- temperature (float)
- measurementDate (String)
- measurementHour (String)
- description (String)

---

```
public class BodyTemperature {

    private String key;
    private float temperature;
    private String measurementDate;
    private String measurementHour;
    private String description;

    public BodyTemperature(float temperature, String measurementDate, String
        measurementHour, String description) {
        this.temperature = temperature;
        this.measurementDate = measurementDate;
        this.measurementHour = measurementHour;
        this.description = description;
    }
    ...
}
```

Codice 3: Classe BodyTemperature

### **Fragment BodyTemperatureInputFragment**

Questo fragment è il primo che appare appena l'activity *BodyTemperatureActivity* è avviata e consente all'utente di inserire manualmente, mediante uno spinner, la propria temperatura corporea aggiungendo una descrizione e salvando la misurazione in Firebase. Oltre a questi dati sono inseriti automaticamente la data e l'ora della misurazione tramite i metodi *getActualDateString()* e *getCurrentHourString()* della classe *DateController* memorizzandoli come stringhe. La temperatura è espressa in gradi centigradi (°C) e varia da un minimo di 33° ad un massimo di 45°, il valore di default è pari a 36°.

---

```
insertTempBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        int temperatureValue = tempValueInput.getValue();
        String day = dateController.getActualDateString();
        String hour = dateController.getCurrentHourString();
        String description = tempDescriptionTxt.getText().toString();
```

```

BodyTemperature savedTemperature = new BodyTemperature((float)
    temperatureValue, day, hour,
    description);
((BodyTemperatureActivity)
    getActivity()).saveTemperatureDb(savedTemperature);

((Vibrator)
    getActivity().getSystemService(Context.VIBRATOR_SERVICE)).vibrate(50);
Toast.makeText(getContext(), "Temperatura inserita con successo",
    Toast.LENGTH_LONG).show();
}
});

```

Codice 4: Inserimento della temperatura nel database



Figura 72: BodyTemperatureInputFragment

### Fragment BodyTemperatureFragment

In questo fragment è possibile visualizzare le misurazioni inserite, i dati di ciascuna misurazione sono inseriti in una card in cui vi è un bottone con cui è possibile eliminare la misurazione inserita. La visualizzazione delle cards è possibile grazie alla classe *BodyTemperatureAdapter* in cui ci sono i metodi per la visualizzazione dei dati di ogni misurazione e per l'eliminazione della misurazione. È presente una innerclass ovvero la classe *BodyTemperatureViewHolder* che fornisce il layout da po-

polare con i dati presenti nel DataSource e viene riutilizzato dal RecyclerView per ridurre il numero di layout da creare per popolare la lista.

```
public class BodyTemperatureViewHolder extends RecyclerView.ViewHolder {

    private TextView title, value, day, hour, description;
    private ImageView deleteBtn;

    public BodyTemperatureViewHolder(@NonNull View itemView) {
        super(itemView);

        title = itemView.findViewById(R.id.temperatureTitle);
        value = itemView.findViewById(R.id.temperatureValueDisplay);
        day = itemView.findViewById(R.id.temperatureDateDisplay);
        hour = itemView.findViewById(R.id.temperatureHourDisplay);
        description = itemView.findViewById(R.id.temperatureDescriptionDisplay);
        deleteBtn = itemView.findViewById(R.id.imageDeleteTemperature);
    }
}
```

Codice 5: BodyTemperatureViewHolder



Figura 73: BodyTemperatureInputFragment

In ogni card sono mostrate: la data, l'ora della misurazione, il valore, la descrizione ed il numero

della misurazione inserita. In basso a destra è presente un *FloatingActionButton* con cui è possibile mostrare le misurazioni inserite in una data specifica selezionabile tramite un menù apposito.

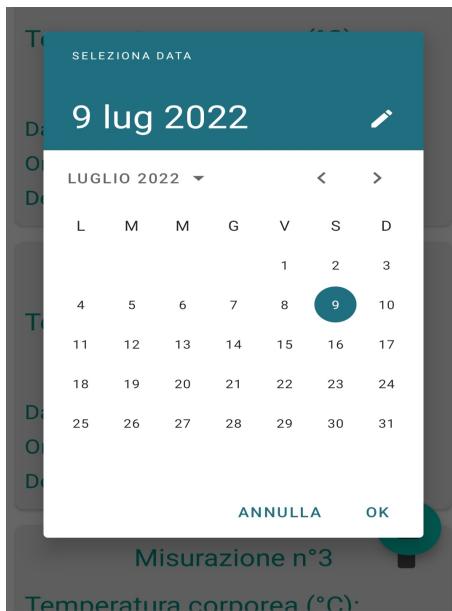


Figura 74: Menù per la selezione della data

Appena l'utente seleziona la data, la funzione *filterBodyTemperatureDate()* inserisce in una lista tutte le misurazioni corrispondenti alla data selezionata dall'utente ed è aggiornata *RecyclerView* del fragment.

```

@Override
public View onCreateView(LayoutInflater inflater, ViewGroup container,
                        Bundle savedInstanceState) {
    View view = inflater.inflate(R.layout.fragment_body_temperature, container,
        false);
    temperatureList = new ArrayList<>();

    dateController = new DateController();

    FirebaseUser user = FirebaseAuth.getInstance().getCurrentUser();
    database = FirebaseDatabase.getInstance();
    dbRef = database.getReference("Utenti_misurazioni")
        .child(user.getUid()).child("Temperatura");

    recyclerView = view.findViewById(R.id.recViewTemp);
    dateTemperatureBtn = view.findViewById(R.id.dateTemperatureBtn);

    adapter = new BodyTemperatureAdapter(this, temperatureList);
    recyclerView.setLayoutManager(new LinearLayoutManager(getActivity()));
    recyclerView.setAdapter(adapter);

    final MaterialDatePicker materialDatePicker =
        MaterialDatePicker.Builder.datePicker().build();

```

```

dbRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot dataSnapshot) {
        temperatureList.clear();
        for (DataSnapshot snapshot : dataSnapshot.getChildren()) {
            temperatureList.add(snapshot.getValue(BodyTemperature.class));
        }
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});

dateTemperatureBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        materialDatePicker.show(getFragmentManager(), "DATE_PICKER");
    }
});

materialDatePicker.addOnPositiveButtonClickListener(new
MaterialPickerOnPositiveButtonClickListener() {
    @Override
    public void onPositiveButtonClick(Object selection) {
        String normalizedSelectedDate =
            dateController.getDataStringDatePicker(materialDatePicker);
        filterBodyTemperatureDate(normalizedSelectedDate);
    }
});
return view;
}

private void filterBodyTemperatureDate(String date){
    ArrayList<BodyTemperature> bodyTemperaturesList = new ArrayList<>();

    for(BodyTemperature bodyTemperature : temperatureList){
        if(bodyTemperature.getMeasurementDate().equals(date)){
            bodyTemperaturesList.add(bodyTemperature);
        }
    }
    adapter.setTemperatureList(bodyTemperaturesList);
    adapter.notifyDataSetChanged();
}
    
```

Codice 6: Logica per la selezione delle misurazioni

### **Fragment BodyTemperatureStatisticsFragment**

In questo fragment è mostrato un grafico a linee con i valori della temperatura corporea inseriti, il grafico è un componente della libreria *MpAndroidChart*.

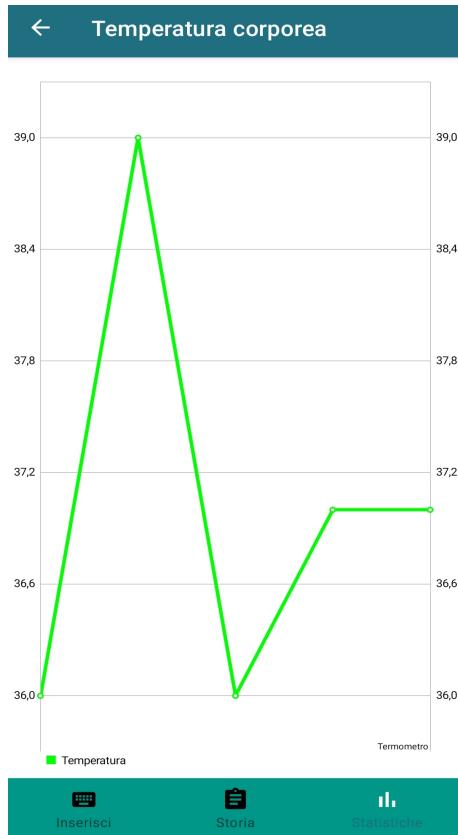


Figura 75: BodyTemperatureStatisticsFragment

Tramite la funzione `setDataLineChart()` il grafico di classe LineChart è disegnato nel fragment e sono inseriti i valori numerici della temperatura presi dal database.

```
private void setDataLineChart() {

    ArrayList<Entry> yMeasure = new ArrayList<>();

    for (int i = 0; i < temperatureList.size(); i++) {

        float temperature = temperatureList.get(i).getTemperature();
        yMeasure.add(new Entry(i, temperature,
            temperatureList.get(i).getMeasurementDate()));
    }
    setMeasure = new LineDataSet(yMeasure, "Temperatura");

    setMeasure.setLineWidth(3f);
    setMeasure.setDrawValues(false);
    setMeasure.setColor(Color.GREEN);
    setMeasure.setCircleColor(Color.GREEN);

    LineData data = new LineData(setMeasure);

    XAxis xAxis = lineChart.getXAxis();
    xAxis.setDrawLabels(false);
    xAxis.setDrawGridLines(false);
}
```

```

        lineChart.getAxisRight().setDrawGridLines(false);

        Description description = lineChart.getDescription();
        description.setText("Termometro");
        lineChart.setData(data);
    }

```

Codice 7: Codice per la creazione del grafico

### 4.3 Sviluppo sezione Ospedalizzazioni

La sezione Ospedalizzazioni è la sezione principale su cui il mio lavoro si è concentrato, questa sezione presenta al suo interno numerose funzionalità quali:

- Inserimento, modifica, eliminazione di ospedalizzazioni
- Visualizzazione sulla mappa della struttura ospedaliera associata all'ospedalizzazione
- Ricerca per nome, città delle ospedalizzazioni
- Ricerca per data delle ospedalizzazioni
- Inserimento, eliminazione di sintomi associati all'ospedalizzazione
- Visualizzazione ed integrazione dei documenti clinici associati all'ospedalizzazione
- Integrazione con la sezione **Ospedali**

Tutte queste funzionalità sono in un'interfaccia minimale, semplice da usare progettata secondo i principi dell'usabilità mentre i dati degli ospedalizzazioni e degli ospedali inseriti dagli utenti sono memorizzati in Firebase, in particolare i dati degli ospedali inseriti da un utente non sono visibili agli altri utenti ma restano privati e la gestione è a discrezione dell'utente che può eliminarli e/o modificarli. Alla sezione *Ospedalizzazioni* si accede, così come accade per le temperature, da un apposito bottone presente nella sezione monitoraggi facilmente riconoscibile.



Figura 76: Bottone per accedere alle ospedalizzazioni

#### ListHospitalizationsActivity

Dal bottone presente nella sezione monitoraggi si accede a questa activity in cui è possibile visualizzare le ospedalizzazioni inserite dall'utente, le informazioni associate ad ogni ospedalizzazione sono:

- Numero del ricovero
- Data di inizio del ricovero
- Data di fine del ricovero
- Causa del ricovero
- Sintomi associati
- Reparto ospedaliero
- Ospedale associato all'ospedalizzazione
- Città, provincia, regione dell'ospedale associato all'ospedalizzazione
- Nome del medico
- Note relative all'ospedalizzazione
- Monitoraggi associati all'ospedalizzazione

Per ciascuna ospedalizzazione, tramite la classi *HospitalizationAdapter* e *HospitalizationViewHolder* è creata una card al cui interno sono presenti quattro buttoni:

1. Visualizzazione sulla mappa dell'ospedale associato
2. Eliminazione dell'ospedalizzazione
3. Modifica dell'ospedalizzazione
4. Visualizzazione dei documenti clinici dell'ospedalizzazione (nel caso fossero associati uno o più documenti)

La *SearchView* posta nella parte superiore dell'activity consente di effettuare una ricerca testuale tra le ospedalizzazioni in base al nome dell'ospedale e alla città del ricovero richiamando la funzione *filterHospitalizations()* che prende in input una stringa appena il testo della *SearchView* cambia. La funzione inserisce in una lista di ospedalizzazioni le ospedalizzazioni il cui nome dell'ospedale e città dell'ospedale associato contengono la stringa presa in input.

---

```
searchHospitalization.setOnQueryTextListener(new SearchView.OnQueryTextListener() {  
    @Override  
    public boolean onQueryTextSubmit(String s) {  
        return false;  
    }  
  
    @Override  
    public boolean onQueryTextChange(String s) {  
        filterHospitalizations(s);  
        return false;  
    }  
});
```

---

Codice 8: Ricerca per nome e città dell'ospedale

```

private void filterHospitalizations(String search) {
    ArrayList<Hospitalization> foundHospitalizations = new ArrayList<>();
    search = search.toLowerCase(Locale.ROOT);

    for (Hospitalization hospitalization : hospitalizationsList) {
        String hospitalName =
            hospitalization.getHospital().getName().toLowerCase(Locale.ROOT);
        String hospitalizationCity =
            hospitalization.getHospital().getCity().toLowerCase(Locale.ROOT);

        if (hospitalName.contains(search) ||
            hospitalizationCity.contains(search)) {
            foundHospitalizations.add(hospitalization);
        }
    }
    adapter.setHospitalizationsList(foundHospitalizations);
    adapter.notifyDataSetChanged();
}
    
```

Codice 9: Metodo filterHospitalizations()



Figura 77: Ricerca per nome e città ospedalizzazioni

La ricerca dell'ospedalizzazione può essere effettuata anche in base alla data di inizio dell'ospedalizzazione, nella parte inferiore è presente un bottone da cui è possibile selezionare una data tramite un pop-up. Appena è selezionata una data è richiamato il metodo *filterHospitalizationsDate()* che mostra tutte le ospedalizzazioni la cui data di inizio corrisponde alla data selezionata dall'utente.



Figura 78: Ricerca ospedalizzazioni per data

I bottoni di ogni card relativi alla modifica ed all'eliminazione sono mostrati sempre mentre il bottone relativo ai documenti clinici è nascosto in caso non fosse almeno un documento e, infine, il bottone relativo alla mappa è nascosto in caso non fosse associata una localizzazione precisa dell'ospedale.



Figura 79: Card per ogni ospedalizzazione

### HospitalizationMapsActivity

La visualizzazione sulla mappa dell'ospedale associato all'ospedalizzazione è possibile leggendo i dati della latitudine e della longitudine dell'ospedale passati a questa activity dall'adapter mediante un oggetto *Bundle* ovvero un dizionario chiave-valore per passare i dati da un'activity all'altra. La mappa è inserita tramite la libreria open-source OsmDroid: una soluzione completa e gratuita della classe MapView (API v1) di Android. La libreria include anche un sistema modulare di provider

di tile con supporto per numerose fonti di tile online e offline e supporto per overlay con overlay integrati per tracciare icone, tracciare la posizione e disegnare forme.

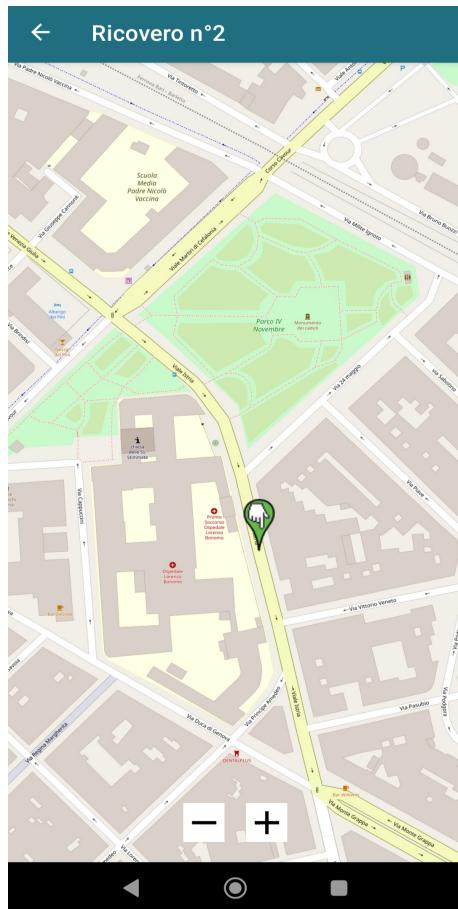


Figura 80: HospitalizationMapsActivity

In questa activity sono presenti due button per aumentare/diminuire lo zoom sulla mappa ed un marcatore sulla mappa che indica la posizione scelta.

```
public class HospitalizationMapsActivity extends AppCompatActivity {

    private MapView map = null;
    private Bundle bundle;

    private double latitude;
    private double longitude;
    private String title;

    private ActionBar actionBar;

    @Override
    protected void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_maps_hospitalization);
```

```

bundle = getIntent().getExtras();
latitude = bundle.getDouble("LATITUDE");
longitude = bundle.getDouble("LONGITUDE");
title = bundle.getString("TITLE");

actionBar = getSupportActionBar();
actionBar.setTitle(title);

Context ctx = getApplicationContext();
Configuration.getInstance().load(ctx,
    PreferenceManager.getDefaultSharedPreferences(ctx));

map = (MapView) findViewById(R.id.mapsHospitalization);
map.setTileSource(TileSourceFactory.MAPNIK);

map.setBuiltInZoomControls(true);
map.setMultiTouchControls(true);

IMapController mapController = map.getController();
mapController.setZoom(18);
GeoPoint startPoint = new GeoPoint(latitude, longitude);
mapController.setCenter(startPoint);

Marker startMarker = new Marker(map);
startMarker.setPosition(startPoint);
startMarker.setAnchor(Marker.ANCHOR_CENTER, Marker.ANCHOR_BOTTOM);
map.getOverlays().add(startMarker);
}

}
    
```

Codice 10: Codice HospitalizationMapsActivity

### **Classe Hospitalization**

La classe Hospitalization rappresenta l'ospedalizzazione memorizzata nel database, questa classe ha i seguenti attributi:

- key (Stringa)
- title (Stringa)
- startHospitalization (Stringa)
- endHospitalization (Stringa)
- hospital (Classe Hospital)
- hospitalizationCause (Stringa)
- ward (Stringa)
- doctor (Stringa)
- notes (Stringa)

- symptomsList (Lista di oggetti di classe ShownSymptom)

La classe implementa l’interfaccia *Serializable* in modo che gli oggetti di questa classe possano essere passati da un’activity all’altra mediante oggetti di classe *Bundle*.

---

```

public class Hospitalization implements Serializable {

    private String key;
    private String title;
    private String startHospitalization;
    private String endHospitalization;
    private Hospital hospital;
    private String hospitalizationCause;
    private String ward;
    private String doctor;
    private String notes;
    private ArrayList<ShownSymptom> symptomsList;

    public Hospitalization() {
        this.title = "";
        symptomsList = new ArrayList<>();
    }

    public Hospitalization(String startHospitalization, String endHospitalization,
                           Hospital hospital, String hospitalizationCause, String ward, String doctor,
                           String notes) {
        this.title = "";
        this.startHospitalization = startHospitalization;
        this.endHospitalization = endHospitalization;
        this.hospital = hospital;
        this.hospitalizationCause = hospitalizationCause;
        this.ward = ward;
        this.doctor = doctor;
        this.notes = notes;
    }
    ...
}

```

---

Codice 11: Classe Hospitalization

### **InputHospitalizationActivity**

Questa activity ha un ruolo fondamentale, infatti non consente soltanto l’inserimento di una ospedalizzazione nel database ma tramite due appositi bottoni presenti è possibile inserire una nuova struttura ospedaliera nel database oppure visualizzare tutti gli ospedali italiani presenti ordinati sia in base alla città di residenza dell’utente che in base alla posizione rilevata tramite gps. Un accorgimento importante adottato nello sviluppo dell’activity è l’utilizzo di variabili flag con cui è possibile mantenere lo stato dell’activity appena si avvia l’activity per l’inserimento di un ospedale o la visualizzazione. L’utilizzo dei flag consente di modificare anche il comportamento dell’activity stessa infatti per ridurre codice ridondante questa activity è utilizzata anche per la modifica delle ospedalizzazioni. I dati e i flag sono gestiti da specifici oggetti di classe *Bundle* tra le activity:

- ListHospitalizationsActivity

- InputHospitalizationActivity
- InputHospitalActivity
- SuggestHospitalActivity

Per spiegare al meglio questo meccanismo, prendo il caso in cui un utente mentre sta creando una nuova ospedalizzazione decide di creare un nuovo ospedale da associare. Premendo il bottone "new" l'activity InputHospitalizationActivity è chiusa ed è richiamata l'activity InputHospitalActivity, durante questo passaggio è creato un oggetto Hospitalization (*partialHospitalization*) contenente i dati inseriti fino a quel momento dall'utente. Questo oggetto insieme al flag che indica lo stato dell'activity InputHospitalizationActivity è inserito in un bundle e i dati sono passati alla nuova activity avviata, in questa activity l'ospedale creato dall'utente è inserito automaticamente nell'ospedalizzazione che l'utente sta creando modificando l'oggetto *partialHospitalization* ottenuto dall'activity precedente. A questo punto è creata una nuova activity InputHospitalizationActivity con i dati (parziali) dell'ospedalizzazione tra cui l'ospedale creato dall'utente e lo stato dalla InputHospitalizationActivity precedente. Discorso analogo per il bottone "esplora" in cui l'utente non inserisce l'ospedale ma può selezionarlo dall'activity SuggestHospitalActivity in cui sono mostrati sia gli ospedali inseriti dall'utente che gli ospedali italiani (presi da un file json presente in memoria).

```
createHospitalBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {

        Intent in = new Intent(InputHospitalizationActivity.this,
                               InputHospitalActivity.class);

        Bundle bundleStart = new Bundle();

        Hospitalization partialHospitalization = new Hospitalization();

        bundleStart.putInt("ACTIVITY_KEY", activityCode);
        partialHospitalization
            .setStartHospitalization(startHospitalizationBtn.getText().toString());
        partialHospitalization
            .setEndHospitalization(endHospitalizationBtn.getText().toString());
        partialHospitalization
            .setHospitalizationCause(hospitalizationCauseInput.getText().toString());
        partialHospitalization.setWard(wardInput.getText().toString());
        partialHospitalization
            .setDoctor(doctorInput.getText().toString());
        partialHospitalization.setNotes(notesInput.getText().toString());
        partialHospitalization.setSymptomsList(shownSymptomsList);

        if (activityCode == R1) {
            bundleStart.putString("MODIFIED_HOSPITALIZATION_KEY",
                bundle.getString("MODIFIED_HOSPITALIZATION_KEY",
                    Hospitalization.getDefaultKey()));
        }
    }
}
```

```

        bundleStart.putSerializable("PARTIAL_HOSPITALIZATION",
            partialHospitalization);
        bundleStart.putInt("STARTER_ACTIVITY",
            SuggestHospitalActivity.HOSPITALIZATION_CALL);

        in.putExtras(bundleStart);
        startActivity(in);
    }
});

```

Codice 12: Salvataggio dello stato dell'activity tramite Bundle

Nell'interfaccia dell'activity ogni campo preso in input ha una card associata dal design minimale, i campi "Reparto", "Medico" e "Note" sono a libera scelta scelta dell'utente mentre i campi "Causa del ricovero" e "Struttura" sono obbligatori per l'inserimento dell'ospedalizzazione.



Figura 81: InputHospitalizationActivity (parte superiore)

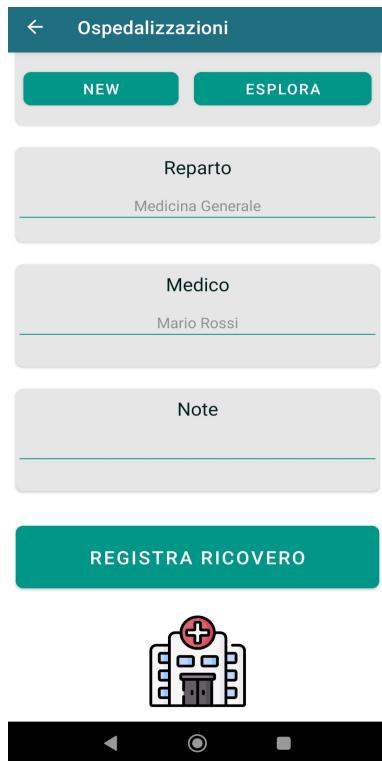


Figura 82: InputHospitalizationActivity (parte inferiore)

I due bottoni corrispondenti ai campi "Inizio ricovero" e "Fine ricovero" aprono un menù con cui è possibile selezionare la data di inizio e fine ricovero rispettivamente. Ovviamente la data di fine ricovero non può essere antecedente alla data di inizio ricovero ma se ciò avviene vi è un messaggio di errore che obbliga l'utente a inserire una data corretta.

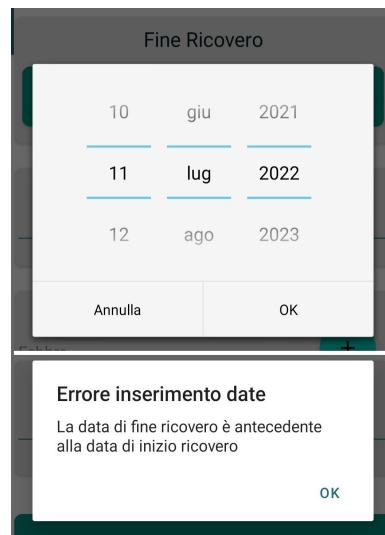


Figura 83: selezione date

L'inserimento della lista dei sintomi è a libera scelta per l'utente ma l'inserimento è guidato ovvero vi è già una lista di sintomi già predefinita ottenuta da un file json presente in memoria, sintomi che non sono in questa lista non possono essere aggiunti.



Figura 84: Inserimento di sintomi

Il file json relativo ai sintomi è letto mediante il metodo `getJsonData()` della classe `JsonDataController` di cui parlerò in seguito da cui si ottiene un oggetto json contenente la descrizione e il nome di ogni sintomo. Con il metodo `getListSymptomsNames()` sono inseriti in una lista di stringhe i nomi di tutti i sintomi letti dal file, questa lista è aggiunta alla `AutoCompleteTextView` che rappresenta il campo in cui è inserito il sintomo per l'autocompletamento.

```
private ArrayList<String> getListSymptomsNames() {
    Iterator<String> iterator = symptomsJsonData.keys();
    ArrayList<String> symptomsList = new ArrayList<>();

    while (iterator.hasNext()) {
        String symptomName = iterator.next();
        symptomsList.add(symptomName);
    }
    return symptomsList;
}
```

Codice 13: Metodo `getListSymptomsNames()`

### Classe `ShownSymptom`

Gli oggetti di classe `ShownSymptom` rappresentano i sintomi memorizzati nell'ospedalizzazione, ogni sintomo presenta due campi:

1. `name (Stringa)`
2. `description (Stringa)`

```
public class ShownSymptom implements Serializable, Parcelable{

    private String name;
    private String description;

    public ShownSymptom() {
```

}

---

```

private ShownSymptom(Parcel parcel){
    name = parcel.readString();
    description = parcel.readString();
}

public ShownSymptom(String name, String description) {
    this.name = name;
    this.description = description;
}
...
}

```

---

Codice 14: Classe ShowSymptom

La classe `ShownSymptom` implementa i metodi `describeContents()` e `writeToParcel()` dell’interfaccia `Parcelable`. L’interfaccia `Parcelable` descrive una modalità di registrazione di un oggetto con tutti i suoi dati primitivi tipo `int`, `boolean`, `String` - oppure un qualsiasi oggetto che a sua volta implementa `Parcelable` - in un oggetto `Parcel` associato ad esso. Nella classe vi è anche il campo statico `CREATOR` che è un oggetto che implementa l’interfaccia `Parcelable.Creator`, questo oggetto serve a ricostruire la classe che implementa `Parcelable` tramite i metodi `createFromParcel()` e `newArray()` ed il costruttore privato `ShownSymptom(Parcel parcel)`.

---

```

@Override
public int describeContents() {
    return 0;
}

@Override
public void writeToParcel(Parcel parcel, int i) {
    parcel.writeString(name);
    parcel.writeString(description);
}

public static final Parcelable.Creator<ShownSymptom> CREATOR = new
    Creator<ShownSymptom>() {
        @Override
        public ShownSymptom createFromParcel(Parcel parcel) {
            return new ShownSymptom(parcel);
        }

        @Override
        public ShownSymptom[] newArray(int i) {
            return new ShownSymptom[i];
        }
    };

```

---

Codice 15: Metodi `Parcelable`

L’implementazione di questa interfaccia nella classe `ShownSymptom` è necessaria affinché la lista dei sintomi presente nell’ospedalizzazione possa essere passata da un’activity all’altra in og-

getti Bundle secondo il meccanismo descritto in precedenza.

### Classe **ShownSymptomAdapter**

La classe **ShownSymptomAdapter** rende possibile la visualizzazione delle card per ogni sintomo inserito dall'utente, al suo interno vi è la classe **SymptomViewHolder** con cui popolare la RecycleView associata con gli elementi presenti nella lista dell'adapter. Appena l'utente preme il bottone "+" posto sulla destra è aggiunto un sintomo alla lista che può essere rimosso mediante l'apposito bottone presente in ogni card.

### Integrazione con la sezione "**Documenti clinici**"

Nella fase di creazione di una nuova ospedalizzazione, al contrario di quanto avviene per gli ospedali, non è possibile creare un nuovo documento clinico da associare all'ospedalizzazione, per farlo bisogna accedere alla sezione "**DOCUMENTI CLINICI**" dal menù dei monitoraggi e o modificare un documento già esistente oppure inserendo nel database un nuovo documento con l'ospedalizzazione associata ad esso.



Figura 85: Menù monitoraggi

La voce del menù associata apre l'activity **ListMonitoringActivity** in cui sono mostrati tutti i documenti clinici inseriti dall'utente ed è possibile effettuare una ricerca in base al nome e alla data del documento.



Figura 86: ListMonitoringActivity

Ogni documento è modellato dalla classe **Monitoring** i cui campi sono:

- nameMonitoring (Stringa)
- date (Stringa)
- key (Stringa)
- hospitalization (Classe Hospitalization)
- path (Lista di stringhe)

```
public class Monitoring implements Serializable {

    private String nameMonitoring;
    private String date;
    private String key;
    private Hospitalization hospitalization;

    private ArrayList<String> path;

    public Monitoring()
    {
        this.nameMonitoring = "";
        this.date = "";
    }
}
```

```

    this.path = new ArrayList<>();
    hospitalization = null;
}

public Monitoring(String name, String date, ArrayList<String> path)
{
    this.nameMonitoring = name;
    this.date = date;
    this.path = path;
    hospitalization = null;
}
...
}

```

Codice 16: Classe Monitoring

È possibile visualizzare tutte le informazioni relative ad un documento cliccando la card associata per passare all'activity **DetailsMonitoringActivity** in cui sono mostrati, oltre alla data e al nome del documento (già visibili dalla card), l'ospedalizzazione e le immagini associate.



| 1/2 ounces vodka

| 1/2 ounce peach schnapps

Figura 87: DetailsMonitoringActivity

L'activity **AddMonitoringActivity** consente l'inserimento e la modifica di un documento clinico con un meccanismo analogo a quello utilizzato nelle ospedalizzazioni in cui il comportamento dell'activity cambia in base al suo stato. L'utente può scrivere il nome del documento, che non può essere vuoto, selezionare la data di inserimento, le immagini e l'ospedalizzazione associate.



Figura 88: AddMonitoringActivity

Dal bottone "SELEZIONA UNA OSPEDALIZZAZIONE" si accede all'activity **HospitalizationMonitoringActivity** con cui è possibile associare un'ospedalizzazione da associare al documento, le card presenti in questa activity sono della classe *HospitalizationAdapter* la stessa utilizzata per mostrare le card nell'activity *ListHospitalizationActivity*. Ovviamente il comportamento della card è diverso in quanto non sono mostrati i comandi per la modifica, l'eliminazione e la visualizzazione dei documenti associati ma solo due bottoni: uno per l'aggiunta dell'ospedalizzazione al documento e l'altro per la visualizzazione dell'ospedalizzazione sulla mappa. Il comportamento della card può essere cambiato mediante dei flag predefiniti passati al costruttore della classe *HospitalizationAdapter*, le informazioni mostrate nella card restano inalterate ma cambiano soltanto i comandi mostrati con lo scopo di evitare codice ridondante.



Figura 89: HospitalizationMonitoringActivity

Anche in questa activity è possibile cercare una ospedalizzazione in base al nome dell'ospedale e alla città del ricovero oppure in base alla data di inizio dell'ospedalizzazione tramite il bottone apposito on basso a destra. La sezione "Documenti clinici" era già presente in Helena ma è stato migliorato l'aspetto grafico della sezione, sono state aggiunte nuove funzionalità ed è stata integrata la sezione "Ospedalizzazioni".

#### 4.4 Sviluppo sezione Ospedali

La sezione Ospedali permette all'utente di visualizzare tutti gli ospedali italiani e inserirne dei nuovi in un database , includendo le seguenti funzionalità:

- Suggerimento di ospedali in base alla residenza o alla posizione rilevata tramite gps
- Inserimento, modifica, eliminazione di ospedali creati dall'utente
- Visualizzazione sulla mappa dell' ospedale
- Ricerca per nome e città dell' ospedale

I dati degli ospedali italiani sono stati ottenuti da un dataset, aggiornato periodicamente, del Ministero della Salute accessibile al seguente link: [Opendata ospedali Ministero della salute](#) da cui è

possibile scaricare un file in formato csv. È stata effettuata una fase di preprocessing del dataset, mediante uno script esterno in Python, in cui sono state estratte soltanto le informazioni rilevanti ai nostri scopi quali:

- Nome dell'ospedale
- Città
- Provincia
- Regione
- Indirizzo
- Coordinate geografiche

Le coordinate geografiche sono state ottenute effettuando delle chiamate alle API di geolocation fornite da servizi quali: [Position Stack](#) e [Here maps](#) passando l'indirizzo, il nome della città e la provincia in cui l'ospedale è situato. A partire da queste informazioni è possibile sia la visualizzazione sulla mappa dell'ospedale che il calcolo della distanza in chilometri rispetto al punto selezionato. La sezione è legata strettamente alla sezione Ospedalizzazioni in quanto è possibile creare un nuovo ospedale o cercare un ospedale mentre si sta creando una nuova ospedalizzazione ma è comunque possibile accedere alla sezione ospedali tramite una voce presente nel menu principale di Helena.

### **Classi Hospital ed HealthcareFacility**

La classe Hospital modella ciascun ospedale inserito e visualizzato dall'utente nelle attività descritte nei prossimi paragrafi. La classe è sottoclassificata della classe HealthcareFacility da cui eredita tutti gli attributi quali:

- key (Stringa)
- name (Stringa)
- city (Stringa)
- province (Stringa)
- city (Stringa)
- region (Stringa)
- address (Stringa)
- latitude (reale)
- longitude (reale)
- localizationFlag (intero)
- distance (reale)

La classe implementa l'interfaccia *Serializable*.

```

public class HealthcareFacility implements Serializable {

    protected String key;
    protected String name;
    protected String city;
    protected String province;
    protected String region;
    protected String address;

    protected double latitude;
    protected double longitude;
    protected int localizationFlag;
    protected double distance;

    public HealthcareFacility(){

    }

    public HealthcareFacility(String name, String city, String province, String
        region, String address) {
        this.name = name;
        this.city = city;
        this.province = province;
        this.region = region;
        this.address = address;

        this.localizationFlag = 1;
        distance = 0;
    }
    ...
}

```

Codice 17: Classe HealthcareFacility

La classe inoltre include i metodi:

- *distanceKmCoordinates()* per il calcolo della distanza in kilometri
- *getShownDistance()* per stampare il valore della distanza

ed è presente l'attributo di classe *Comparator<HealthcareFacility>* *distanceComparator* per il confronto di due ospedali in base al valore della distanza grazie al metodo *compare()* ed è fondamentale per l'activity *SuggestHospitalActivity*.

```

public static double distanceKmCoordinates(double lat1, double lon1, double lat2,
    double lon2) {

    final int earthRadiusKm = 6371;
    double dLat = degreesToRadians(lat2 - lat1);
    double dLon = degreesToRadians(lon2 - lon1);

```

```

lat1 = degreesToRadians(lat1);
lat2 = degreesToRadians(lat2);

    double a = Math.sin(dLat / 2) * Math.sin(dLat / 2)
        + Math.sin(dLon / 2) * Math.sin(dLon / 2) * Math.cos(lat1) *
        Math.cos(lat2);
    double c = 2 * Math.atan2(Math.sqrt(a), Math.sqrt(1 - a));

    return earthRadiusKm * c;
}

public String getShownDistance() {

    int distanceInt = (int) Math.round(distance);

    if (distanceInt > 1) {
        return String.valueOf(distanceInt) + " " + "kilometri";
    }

    if (distanceInt == 1) {
        return "1 kilometro";
    }
    return "0,5 kilometri";
}

public static Comparator<HealthcareFacility> distanceComparator = new
Comparator<HealthcareFacility>() {
    @Override
    public int compare(HealthcareFacility healthcareFacility, HealthcareFacility
        t1) {
        double distance = healthcareFacility.getDistance();
        double distanceCompare = t1.getDistance();

        if(distance > distanceCompare){
            return 1;
        }

        if(distanceCompare > distance){
            return -1;
        }
        return 0;
    }
};

```

Codice 18: Codice per il calcolo la stampa ed il confronto della distanza in kilometri tra due strutture

### Activity InputHospitalActivity

L'inserimento e la modifica di una struttura ospedaliera avvengono in questa activity con un sistema analogo a quello adottato per l'inserimento/modifica delle ospedalizzazioni in cui un flag cambia il comportamento dell'activity. L'utente può inserire tre campi:

1. Nome della struttura ospedaliera (Obbligatorio)

2. Città della struttura ospedaliera (Obbligatorio)
3. Indirizzo della struttura ospedaliera (Opzionale)



Figura 90: Activity InputHospitalActivity

```

int localizationFlag = 1;

List<Address> addressList = new ArrayList<>();
try {
    addressList = geocoder.getFromLocationName(address + " " + city,
        1);
} catch (IOException e) {
    e.printStackTrace();
    localizationFlag = 0;
}

List<Address> citiesList = new ArrayList<>();

if (localizationFlag == 0) {
    try {
        citiesList = geocoder.getFromLocationName(city, 1);
    } catch (IOException e) {
        e.printStackTrace();
    }
}

```

Codice 19: Calcolo delle coordinate dell'ospedale

A partire dal campo città e indirizzo inseriti dall'utente, l'activity usando un oggetto di classe `Geocoder` contenente i metodi per il geocoding: un processo in cui a partire da un indirizzo si ottengono le coordinate geografiche con una certa precisione. Nel caso in cui non sia possibile ottenere la posizione esatta dell'ospedale, si impostano le coordinate dell'ospedale pari alle coordinate della città in cui è situato segnalando con un flag l'imprecisione del processo di geocoding. L'ospedale

creato è inserito in Firebase in una apposita sezione i cui dati sono privati per ogni utente ed è visualizzabile dall'activity SuggestHospitalActivity.

### SuggestHospitalActivity

La visualizzazione degli ospedali inseriti dall'utente e di tutti gli ospedali italiani, questi ultimi presi da un file Json presente in memoria creato in precedenza, avviene in questa activity. L'activity non si limita soltanto alla visualizzazione, in una apposita card per ciascun ospedale, di tutte le strutture ospedaliere ma è anche in grado di suggerire gli ospedali più vicini accedendo a informazioni quali: città di residenza dell'utente e posizione attuale dell'utente. La prima è già presente in Firebase e per velocizzarne la lettura è memorizzata in locale nelle *SharedPreferences* mentre la seconda è ottenuta mediante GPS. L'utente tramite un apposito menù ben visibile nell'activity può selezionare una delle due opzioni, se il GPS non è attivato l'applicazione apre l'apposita sezione per l'attivazione del GPS nel dispositivo. In entrambi i casi l'applicazione leggendo le coordinate geografiche di ciascun ospedale (latitudine e longitudine) calcola la distanza dell'ospedale in base alla città dell'utente, distanza usata per l'ordinamento crescente della lista degli ospedali. Il calcolo delle distanze degli ospedali in base alla residenza è effettuato appena l'activity è aperta in quanto finché l'utente è in questa activity la residenza non può cambiare (modificabile dal menu *Profilo utente*) sia sugli ospedali inseriti dell'utente letti da Firebase che sugli ospedali italiani presenti di default letti richiamando il metodo *getDeafaultHospitals()*.

Le funzioni per il calcolo della distanza ed il compratore degli ospedali in base al valore della distanza, ovvero *distanceComparator* sono presenti nella classe *HealthcareFacility* già descritti, comparatore preso in input dal metodo *sort()* della classe Collections per l'ordinamento della lista degli ospedali.

```
dbRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {

        for (DataSnapshot snap : snapshot.getChildren()) {
            Hospital hospital = snap.getValue(Hospital.class);
            double hospitalLatitude = hospital.getLatitude();
            double hospitalLongitude = hospital.getLongitude();
            double hospitalDistance =
                HealthcareFacility.distanceKmCoordinates(latitudeUserCity,
                    longitudeUserCity,
                    hospitalLatitude, hospitalLongitude);

            hospital.setDistance(hospitalDistance);
            hospitalList.add(hospital);
        }
        getDeafaultHospitals();

        hospitalListResidence.addAll(hospitalList);
        Collections.sort(hospitalListResidence,
            HealthcareFacility.distanceComparator);

        adapter.setHospitalList(hospitalList);
        adapter.notifyDataSetChanged();
    }
}
```

```

@Override
public void onCancelled(@NonNull DatabaseError error) {
}

});

```

Codice 20: Aggiunta degli ospedali alla liste

Se l'utente seleziona l'ordinamento degli ospedali in base alla posizione tramite l'apposito bottone, l'activity richiama il metodo *filterHospitalsGps()* in cui sono letti i campi *latitude* e *longitude* dell'oggetto *Location* utilizzato anche per ottenere il nome della città attuale mostrata in una apposita card.

```

private void filterHospitalsGps() throws SecurityException {

    if (locationManager.isProviderEnabled(LocationManager.GPS_PROVIDER)) {
        locationManager.requestLocationUpdates(LocationManager.GPS_PROVIDER,
            1000, 100, new LocationListener() {
                @Override
                public void onLocationChanged(@NonNull Location location) {
                    double gpsLatitude = location.getLatitude();
                    double gpsLongitude = location.getLongitude();

                    Geocoder geocoder = new Geocoder(SuggestHospitalActivity.this,
                        Locale.getDefault());
                    List<Address> listAddressGps = new ArrayList<>();
                    String gpsLocality = " ";

                    try {
                        listAddressGps = geocoder.getFromLocation(gpsLatitude,
                            gpsLongitude, 1);
                    } catch (IOException e) {
                        e.printStackTrace();
                    }

                    if (listAddressGps != null) {
                        if (listAddressGps.size() > 0) {
                            Address addr = listAddressGps.get(0);
                            gpsLocality = addr.getLocality();
                            citySearchTextView.setText(gpsLocality);
                        }
                    }
                    hospitalListGps.clear();

                    // se citt del gps e citt di residenza coincidono, non necessario
                    // ricalcolare le distanze
                    if (gpsLocality.equals(userCity)) {
                        adapter.setHospitalList(hospitalListResidence);
                    }
                }
            });
    }
}

```

```
        } else {
            for (Hospital hospital : hospitalList) {
                double hospitalLatitude = hospital.getLatitude();
                double hospitalLongitude = hospital.getLongitude();
                double hospitalDistance =
                    HealthcareFacility.distanceKmCoordinates(gpsLatitude,
                    gpsLongitude,
                    hospitalLatitude,
                    hospitalLongitude);

                hospital.setDistance(hospitalDistance);
                hospitalListGps.add(hospital);
            }
            Collections.sort(hospitalListGps,
                HealthcareFacility.distanceComparator);
            adapter.setHospitalList(hospitalListGps);
        }
        adapter.notifyDataSetChanged();
    }
});
```

Codice 21: Metodo per la ricerca e ordinamento degli ospedali

Non è necessario calcolare la distanza di ogni ospedale se la città di residenza dell'utente e la città rilevata tramite GPS coincidono.

Nella parte inferiore vi è la *RecyclerView* in cui sono visualizzate le card per ciascun ospedale con informazioni quali:

- Nome della struttura ospedaliera
  - Città
  - Provincia
  - Regione
  - Indirizzo
  - Distanza in base alla residenza (o alla posizione attuale)

e dei bottoni per:

- Cercare la struttura ospedaliera su internet aprendo automaticamente il browser di default.
  - Visualizzare la struttura ospedaliera sulla mappa richiamando una apposita activity.

Se l'activity è chiamata mentre si sta creando un'ospitalizzazione dall'activity InputHospitalizationActivity nella card compare un bottone per associare l'ospedale all'ospitalizzazione che si sta creando altrimenti se l'activity è chiamata dal menù principale, non essendoci alcuna ospedalizzazione in fase di creazione, questo bottone scompare e compare un bottone per l'inserimento di un nuovo ospedale.

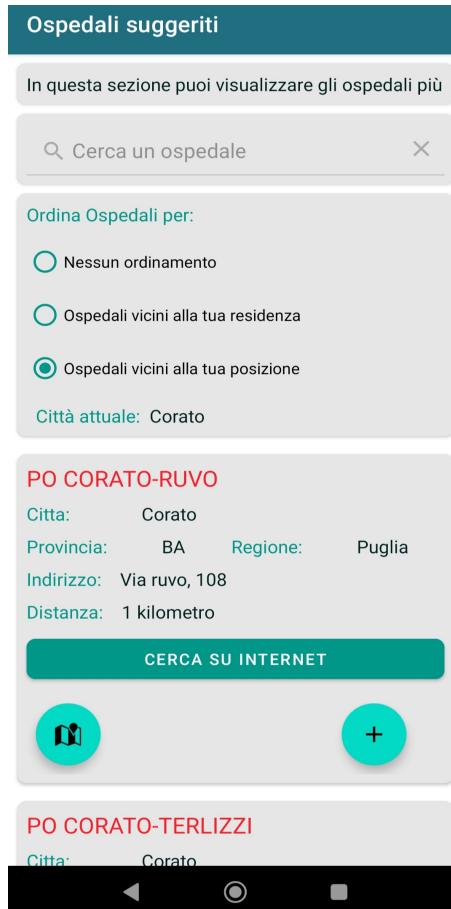


Figura 91: SuggestHospitalActivity, layout mentre si crea ospedalizzazione

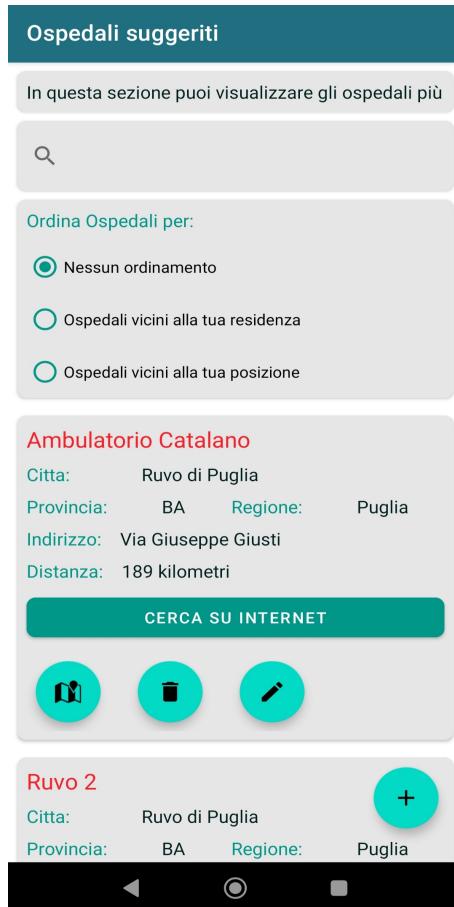


Figura 92: SuggestHospitalActivity

Per gli ospedali creati dall’utente è possibile tramite due bottoni aggiuntivi la modifica e l’eliminazione dell’ospedale sul database. Il bottone per la modifica dell’ospedale richiama l’activity *InputHospitalActivity* in cui un apposito flag ne modifica il comportamento secondo il meccanismo analogo presente per l’inserimento/modifica delle ospedalizzazioni. La visualizzazione della card è resa possibile dalla classe **HospitalAdapter** che collega il layout della card definita in *layout\_row\_hospital* ad ogni ospedale presente nella lista presa in input.

### **HospitalMapsActivity**

L’activity è molto simile all’activity HospitalizationMapsActivity, entrambe prendono in input latitudine, longitudine e nome dell’ospedale mostrato su una mappa creata usando le funzioni della libreria Osmdroid.

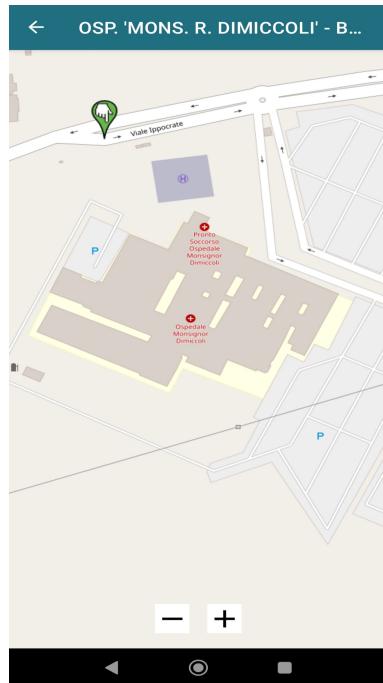


Figura 93: HospitalMapsActivity

## 4.5 Sviluppo sezione Farmacie

In questa sezione dell'applicazione, accessibile dal menù a tendina dell'activity iniziale, l'utente può visualizzare le farmacie e le parafarmacie presenti in Italia, ordinandole per vicinanza in base alla città di residenza o alla posizione, inserendo nel database una nuova farmacia nel database e cercando su una mappa farmacie e parafarmacie presenti in una città cercata dall'utente. I dati delle farmacie e parafarmacie italiane sono stati ottenuti da dataset, periodicamente aggiornati, del ministero della salute accessibili al seguente link: [Open data Ministero della salute](#), in formato csv. È stato eseguito un pre-processing da uno script python non presente nel codice dell'applicazione in fase preliminare con lo scopo di estrarre solo le informazioni di nostro interesse quali:

- Nome
- Città
- Provincia
- Regione
- Indirizzo
- Tipo (Farmacia o Parafarmacia)
- Latitudine
- Longitudine
- Flag geocoding

I campi latitudine e longitudine sono stati ottenuti effettuando delle chiamate alle api per la geolocation fornite da [Position Stack](#) e [Here maps](#) passando l'indirizzo, il nome della città e la provincia in cui l'attività è situata e i dati ottenuti sono stati salvati in un file json presente nella cartella /assets da cui è estratta una lista di oggetti *Pharmacy*.

### Classe Pharmacy

La classe **Pharmacy** è sottoclasse della classe **HealthCareFacility** ed eredita tutti gli attributi ed i metodi della superclasse aggiungendo un nuovo attributo ovvero type che specifica se l'oggetto si riferisce ad una farmacia oppure ad una parafarmacia.

---

```
public class Pharmacy extends HealthCareFacility {  
  
    private String type;  
  
    public Pharmacy(){  
  
    }  
  
    public Pharmacy(String name, String city, String province, String region, String  
        type, String address) {  
        super(name, city, province, region, address);  
        this.type = type;  
    }  
    ...  
}
```

---

Codice 22: Classe Pharmacy

### Activity PharmacyActivity

L'activity **PharmacyActivity** ha al suo interno due distinti fragments:

1. **ListPharmacyFragment** per la visualizzazione ed il suggerimento di farmacie/parafarmacie
2. **MapsPharmacyFragment** per la visualizzazione sulla mappa delle farmacie/parafarmacie presenti nella città cercata dall'utente.

È possibile passare da un fragment all'altro velocemente dal menù inferiore da cui è possibile avviare l'activity **InputPharmacyActivity** per l'inserimento di farmacie/ parafarmacie.

---

```
public class PharmacyActivity extends AppCompatActivity {  
  
    private BottomNavigationView bottomNavigationView;  
    private ListPharmacyFragment fragment1;  
    private MapsPharmacyFragment fragment2;  
  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_pharmacy);  
        bottomNavigationView = findViewById(R.id.bottomNavigationPharmacy);  
    }  
}
```

```
fragment1 = new ListPharmacyFragment();
fragment2 = new MapsPharmacyFragment();

bottomNavigationView.setOnItemSelectedListener(item -> {
    switch (item.getItemId()){
        case R.id.listPharmaciesMenu:
            getSupportFragmentManager()
                .beginTransaction()
                .replace(R.id.fragmentContainerPharmacy, fragment1)
                .commit();
            return true;

        case R.id.mapsPharmacyMenu:
            getSupportFragmentManager()
                .beginTransaction()
                .replace(R.id.fragmentContainerPharmacy, fragment2)
                .commit();
            return true;

        case R.id.addPharmacyMenu:
            Intent in = new Intent(this, InputPharmacyActivity.class);
            Bundle bundle = new Bundle();
            bundle.putInt("ACTIVITY_KEY", InputPharmacyActivity.APH);
            in.putExtras(bundle);
            startActivity(in);
    }
    return true;
});
}
```

---

Codice 23: Activity PharmacyActivity

### ListPharmacyFragment

In questo fragment l'utente può visualizzare tutte le farmacie e parafarmacie presenti sia sul file json che nell'apposita sezione del database in cui sono memorizzate farmacie e parafarmacie inserite da ciascun utente. È possibile ricercare farmacie/ parafarmacie per nome e città, filtrarle in base al tipo e ordinarle in base alla distanza sia della città di residenza che della posizione attuale con metodi simili a quelli presenti nell'activity SuggestHospitalActivity.



Figura 94: ListPharmacyFragment

Per ciascuna attività sono visualizzate informazioni quali:

- Nome
- Tipo
- Città
- Provincia
- Regione
- Indirizzo
- Distanza (rispetto alla residenza o alla posizione attuale)

sono presenti sempre due bottoni:

1. Ricerca su internet aprendo il browser di default
2. Visualizzazione sulla mappa

in aggiunta, per le farmacie/parafarmacie inserite dall'utente, sono presenti i bottoni per la modifica e l'eliminazione sul database. La visualizzazione di ogni card nella RecyclerView è possibile

grazie alla classe **PharmacyAdapter** che associa ad ogni elemento della lista delle farmacie/para-farmacie una card.

### **PharmacyMapsActivity**

L'activity è richiamata da un apposito bottone presente in ogni card delle farmacie visualizzate nel fragment descritto in precedenza e permette la visualizzazione della farmacia/parafarmacia sulla mappa, prendendo in input appena è chiamata il nome della farmacia, la latitudine e la longitudine. Come per la visualizzazione di ospedalizzazioni e ospedali, la mappa è stata creata mediante la libreria OsmDroid.

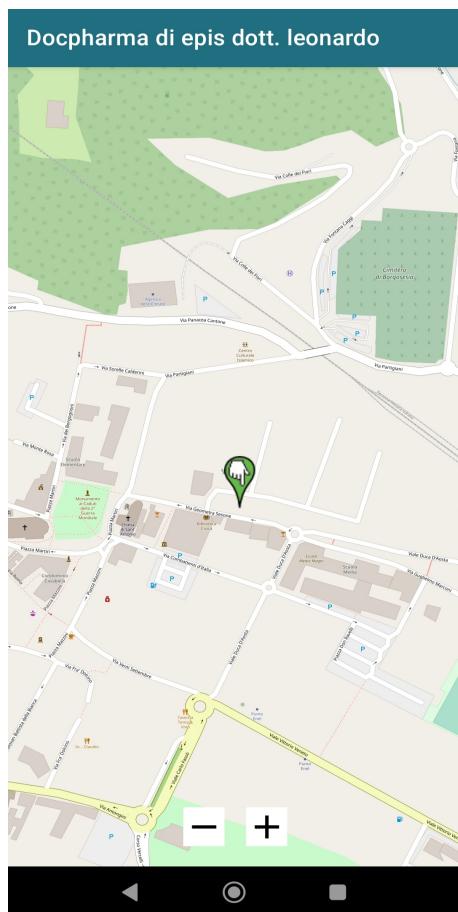


Figura 95: PharmacyMapsActivity

### **MapsPharmacyFragment**

Nel fragment descritto precedentemente, ListPharmacyFragment, il suggerimento delle farmacie e parafarmacie in base alla posizione o alla residenza dell'utente, in questo fragment l'utente può cercare una qualsiasi città italiana e visualizzare direttamente sulla mappa le farmacie e parafarmacie presenti nella città selezionata. Il fragment come prima cosa acquisisce le farmacie e le parafarmacie sia presenti in Firebase (inserite dall'utente) che presenti nel file json nella cartella /assets, creando una lista di oggetti Pharmacy.

```
dbRef = FirebaseDatabase.getInstance()
getReference("Farmacie_utenti")
.child(FirebaseAuth.getInstance().getCurrentUser().getUid());
```

```

dbRef.addValueEventListener(new ValueEventListener() {
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        pharmaciesList.clear();

        for (DataSnapshot snap : snapshot.getChildren()) {
            Pharmacy pharmacy = snap.getValue(Pharmacy.class);
            double pharmacyLatitude = pharmacy.getLatitude();
            double pharmacyLongitude = pharmacy.getLongitude();

            if (pharmacyLatitude != 0 && pharmacyLongitude != 0) {
                double pharmacyDistance =
                    HealthcareFacility.distanceKmCoordinates(latitudeUserCity,
                        longitudeUserCity, pharmacyLatitude, pharmacyLongitude);
                pharmacy.setDistance(pharmacyDistance);
                pharmaciesList.add(pharmacy);
            }
        }
        // prendo le farmacie dal file json
        getPharmaciesList();

        pharmaciesListResidence.addAll(pharmaciesList);
        Collections.sort(pharmaciesListResidence,
            HealthcareFacility.distanceComparator);

        adapter.setPharmacyArrayList(pharmaciesList);
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {
    }
});
```

Codice 24: Lettura delle farmacie e parafarmacie

Appena l'utente inserisce la città, il fragment inserisce in una lista le farmacie/parafarmacie presenti nella città selezionata e calcola le coordinate della città per posizionare la mappa e visualizzare i marcatori.

---

```

mapsCityInputBtn.setOnClickListener(new View.OnClickListener() {

    @Override
    public void onClick(View view) {
        String citySearch = mapsCityInput.getText().toString().trim();
        double latitude = 0;
        double longitude = 0;

        List<Address> addressList = new ArrayList<>();
```

```

try {
    addressList = geocoder.getFromLocationName(citySearch, 1);
} catch (IOException e) {
    e.printStackTrace();
}

if(addressList.size() > 0){
    Address address = addressList.get(0);
    latitude = address.getLatitude();
    longitude = address.getLongitude();
}

if(latitude != 0 && longitude != 0){
    ArrayList<OverlayItem> itemsMaps = new ArrayList<>();
    for(Pharmacy pharmacy : pharmaciesList){
        if(pharmacy.getCity().equals(citySearch)){

            double latitudePharmacy = pharmacy.getLatitude();
            double longitudePharmacy = pharmacy.getLongitude();
            Marker marker = new Marker(map);
            marker.setTitle(pharmacy.getName());
            marker.setPosition(new GeoPoint(latitudePharmacy,
                longitudePharmacy));

            itemsMaps.add(new OverlayItem(pharmacy.getName(),
                pharmacy.getType(), new GeoPoint(latitudePharmacy,
                longitudePharmacy)));
        }
    }

    IMapController mapController = map.getController();

    mapController.setZoom(18);

    GeoPoint startPoint = new GeoPoint(latitude, longitude);
    mapController.setCenter(startPoint);

    map.setTileSource(TileSourceFactory.MAPNIK);
    map.setBuiltInZoomControls(true);
    map.setMultiTouchControls(true);

    ItemizedIconOverlay<OverlayItem> overlayItemIconOverlay =
        new ItemizedIconOverlay<OverlayItem>(context, itemsMaps, null);
    map.getOverlays().add(overlayItemIconOverlay);
} else{
    new AlertDialog.Builder(context).setTitle("Attenzione!")
        .setMessage("La città che hai inserito non è stata trovata")
        .setPositiveButton("Ok", new
            DialogInterface.OnClickListener() {

                @Override

```

```

        public void onClick(DialogInterface dialogInterface,
            int i) {
            mapsCityInput.setText("");
        }
    }).show();
}
});

```

Codice 25: Codice per la visualizzazione sulla mappa delle farmacie e parafarmacie

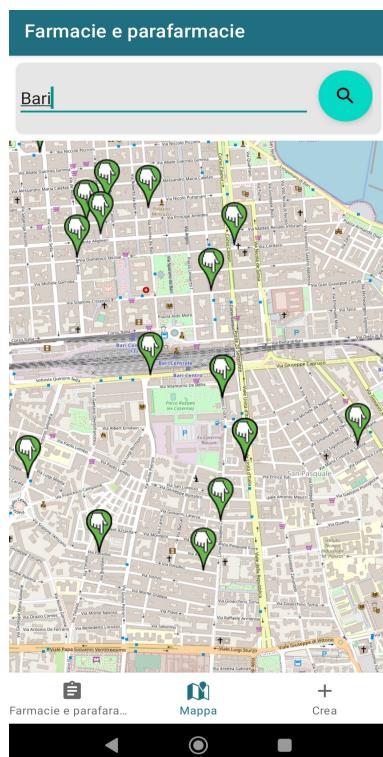


Figura 96: MapsPharmacyFragment

### **InputPharmacyActivity**

L'activity è accessibile dall'ultima voce del menù presente sulla destra e consente all'utente di inserire nel database una farmacia o parafarmacia.

L'utente deve inserire obbligatoriamente il nome della farmacia e la città necessaria per il geocoding ed il calcolo delle distanze nel fragment ListPharmacyFragment mentre l'indirizzo della farmacia è opzionale ma l'assenza dell'indirizzo non consente la visualizzazione della posizione esatta della farmacia sulla mappa. Anche in questa activity vi è un meccanismo analogo a quello presente nelle sezioni Ospedali e Ospedalizzazioni, grazie al quale è possibile usare la stessa activity sia per l'inserimento che per la modifica di una farmacia/parafarmacia inserita dall'utente in base ad un flag impostato alla creazione dell'activity.

Inserisci una farmacia o parafarmacia

Nome

---

Tipo

 Farmacia
 Parafarmacia

Città

---

Indirizzo

---

**INSERISCI**

[Visualizza farmacie e parafarmacie](#)

Figura 97: InputPharmacyActivity

## 4.6 Sviluppo sezione Stato Emotivo

Questa sezione, accessibile dal menù dei monitoraggi, da la possibilità all'utente di riportare il proprio stato emotivo sulla base delle cinque emozioni primarie ( più la neutralità ) quali:

- Neutralità
- Felicità
- Tristezza
- Rabbia
- Paura
- Disgusto

verbalizzando a parole sue le emozioni che prova. Anche se esistono tecniche più avanzate per il rilevamento/riconoscimento delle emozioni (o dello stato emotivo) lo scopo di questa sezione è fornire una funzionalità semplice ed intuitiva da usare anche come diario in cui l'utente può descrivere il suo stato d'animo in un determinato momento, associando un'emozione (con una sua intensità) che sta provando.



STATO EMOTIVO

Figura 98: Voce del menù per accedere alla sezione Stato Emotivo

### **Classe MentalStatus**

La classe modella lo stato emotivo inserito dall'utente e ha i seguenti attributi:

- key (Stringa)
- title (Stringa)
- emotion (Stringa)
- date (Stringa)
- hour (Stringa)
- intensity (intero)
- description (Stringa)

```
public class MentalStatus implements Serializable {

    private String key;
    private String title;
    private String emotion;
    private String date;
    private String hour;
    private int intensity;
    private String description;

    public MentalStatus(){

    }

    public MentalStatus(String title, String emotion, int intensity, String
        description) {
        this.title = title;
        this.emotion = emotion;
        this.intensity = intensity;
        this.description = description;
    }
    ...
}
```

Codice 26: Classe MentalStatus

### **Activity MentalHealthActivity**

L'activity comprende due distinti fragments:

1. InputMentalStatusFragment
2. MentalStatusFragment

il primo per l'inserimento di uno stato emotivo, il secondo per la visualizzazione dei vari stati emotivi presenti nel database.



Figura 99: InputMentalStatusFragment, card per l'intensità dell'emozione

### **InputMentalStatusFragment**

L'utente inserisce lo stato emotivo tramite questo fragment in cui è presente uno slider con cui l'utente può selezionare una delle sei emozioni di base. Se l'emozione selezionata è diversa dalla

neutralità, compare una card per inserire il valore di intensità dell'emozione in un intervallo compreso tra 1 e 10. Il campo titolo è obbligatorio mentre il campo "Diario" è opzionale. Tutte le informazioni sono salvate in Firebase.

### MentalStatusFragment

L'utente può visualizzare, eliminare e ricercare per data di inserimento e per titolo gli stati emotivi inseriti presenti nel database Firebase. Come prima cosa, non appena è creato il fragment, sono lette le informazioni dal database e ogni stato emotivo è inserito in una lista di oggetti di classe *MentalStatus*.

---

```
public class MentalStatusFragment extends Fragment {

    private InputMentalStatusFragment fragment;
    private Context context;
    private SearchView searchViewEmotion;
    private RecyclerView recyclerViewEmotion;
    private FloatingActionButton dateSearchBtn, addEmotionBtn;
    private MentalStatusAdapter adapter;

    private DatabaseReference dbRef;
    private String userID;
    private ArrayList<MentalStatus> mentalStatusList;

    private DateController dateController;

    public MentalStatusFragment() {
    }

    @Override
    public void onCreate(@Nullable Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
    }

    @Nullable
    @Override
    public View onCreateView(@NonNull LayoutInflater inflater, @Nullable ViewGroup container, @Nullable Bundle savedInstanceState) {
        View view = inflater.inflate(R.layout.fragment_mental_status, container, false);
        initInterface(view);
        context = getActivity();
        dateController = new DateController();

        fragment = new InputMentalStatusFragment();
        mentalStatusList = new ArrayList<>();
        adapter = new MentalStatusAdapter(context);
        recyclerViewEmotion.setLayoutManager(new LinearLayoutManager(context));
        recyclerViewEmotion.setAdapter(adapter);

        userID = FirebaseAuth.getInstance().getCurrentUser().getUid();
    }
}
```

```
dbRef =
    FirebaseDatabase.getInstance().getReference("Stato_mentale_utenti").child(userID);

dbRef.addValueEventListener(new ValueEventListener() {
    @SuppressLint("NotifyDataSetChanged")
    @Override
    public void onDataChange(@NonNull DataSnapshot snapshot) {
        mentalStatusList.clear();

        for(DataSnapshot snap : snapshot.getChildren()){
            mentalStatusList.add(snap.getValue(MentalStatus.class));
        }
        adapter.setMentalStatusArrayList(mentalStatusList);
        adapter.notifyDataSetChanged();
    }

    @Override
    public void onCancelled(@NonNull DatabaseError error) {

    }
});

...
final MaterialDatePicker materialDatePicker =
    MaterialDatePicker.Builder.datePicker().build();

dateSearchBtn.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        materialDatePicker.show(getParentFragmentManager(), "DATE_PICKER");
    }
});

materialDatePicker.addOnPositiveButtonClickListener(new
    MaterialPickerOnPositiveButtonClickListener() {
        @Override
        public void onPositiveButtonClick(Object selection) {
            String normalizedSelectedDate =
                dateController.getDataStringDatePicker(materialDatePicker);
            filterEmotionsDate(normalizedSelectedDate);
        }
    });
}

searchViewEmotion.setOnQueryTextListener(new SearchView.OnQueryTextListener()
{
    @Override
    public boolean onQueryTextSubmit(String s) {
        return false;
    }

    @Override
```

---

```

        public boolean onQueryTextChange(String s) {
            filterEmotions(s);
            return false;
        }
    });

    return view;
}
...
}

```

Codice 27: MentalStatusFragment

Ad ogni stato emotivo è associata una specifica card mediante la classe **MentalStatusAdapter**. Richiamando le funzioni *filterEmotionsDate()* e *filterEmotions()* è possibile la ricerca per nome e per data.

---

```

private void filterEmotionsDate(String date){
    ArrayList<MentalStatus> emotionsDate = new ArrayList<>();

    for(MentalStatus mentalStatus : mentalStatusList){
        if(mentalStatus.getDate().equals(date)){
            emotionsDate.add(mentalStatus);
        }
    }
    adapter.setMentalStatusArrayList(emotionsDate);
    adapter.notifyDataSetChanged();
}

private void filterEmotions(String title){
    ArrayList<MentalStatus> mentalStatusesTitle = new ArrayList<>();
    title = title.toLowerCase(Locale.ROOT);

    for(MentalStatus mentalStatus : mentalStatusList){
        if(mentalStatus.getTitle().toLowerCase(Locale.ROOT).contains(title)){
            mentalStatusesTitle.add(mentalStatus);
        }
    }
    adapter.setMentalStatusArrayList(mentalStatusesTitle);
    adapter.notifyDataSetChanged();
}

```

Codice 28: Metodi per la ricerca degli stati emotivi

È presente un apposito bottone con cui è possibile inserire un nuovo stato emotivo accedendo al fragment descritto nel paragrafo precedente.

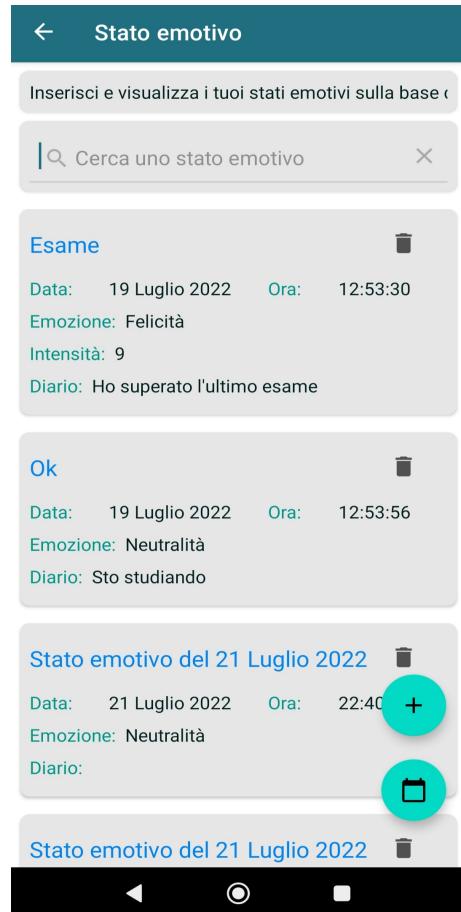


Figura 100: MentalStatusFragment

## 5 Sperimentazione

Dopo aver affrontato nei capitoli precedenti tutti gli aspetti relativi allo stato dell'arte, alla progettazione e all'implementazione delle nuove funzionalità in Helena, in questo capitolo discuteremo del comportamento degli utenti, discussione coadiuvata dall'analisi dei dati ottenuti tramite un questionario proposto. Ad ogni soggetto coinvolto nella sperimentazione è stato chiesto esplicitamente il consenso per l'utilizzo dei dati a scopo di ricerca. Tutti i dati raccolti da ogni utente sono tutelati dal [Regolamento UE 2016/679, art. 89](#) una normativa del GDPR (Regolamento Generale sulla Protezione dei Dati)

riguardante garanzie e deroghe relative al trattamento a fini di archiviazione nel pubblico interesse, di ricerca scientifica o storica o a fini statistici.

La sperimentazione è stata divisa in due fasi, nella prima è stata data la possibilità agli utenti di utilizzare, compiendo task specifici, le tre nuove sezioni principali implementate nell'applicazione ovvero:

1. Sezione Ospedalizzazioni
2. Sezione Farmacie
3. Sezione Stato emotivo

Nella seconda fase, agli utenti sono stati sottoposti, per ciascun task specifico ad ognuna delle funzionalità implementate, due questionari: un questionario NPS (*Net Promoter Score*) ed un questionario SUS (*System Usability Score*). In aggiunta è stato sottoposto un questionario, di carattere generale, per la raccolta di informazioni tra cui:

- Genere
- Fascia di età
- Grado di formazione
- Livello di conoscenza delle tecnologie

La sperimentazione avverrà su un campione di 30 persone tutti i dati, raccolti in maniera completamente anonima e non riguardante informazioni sensibili, saranno stati utilizzati con lo scopo di misurare il grado di usabilità di ciascuna sezione implementata.

### 5.1 Protocollo sperimentale

È stato necessario coinvolgere un significativo numero di persone affinché si potesse procedere alla sperimentazione, a ciascun utente sono stati spiegati in dettaglio gli obiettivi, le finalità della sperimentazione e sono stati forniti tre task:

1. Creazione di una ospedalizzazione in un ospedale nel raggio di 40 chilometri dalla residenza
2. Ricerca di parafarmacie entro 3 chilometri di distanza dalla posizione (rilevata tramite GPS)
3. Inserimento di uno stato emotivo a seguito di un evento negativo, ricerca di uno stato emotivo già inserito

task da completare in un minimo di tre giorni.

### **Modulo sottoposto agli utenti**

Il modulo sottoposto agli utenti comprendente i questionari *SUS* e *NPS* per ciascun task ed il questionario di carattere generale è stato creato con l'ausilio di [Google Moduli](#): uno strumento online gratuito per la creazione di sondaggi e questionari dalle numerose funzionalità.

Le caratteristiche del questionario sono le seguenti:

- Utilizzo di un approccio informale
- Porre domande in modo semplice
- Cercare di non influenzare in nessun modo l'utente
- Evitare l'uso di termini troppo tecnici
- Concedere all'utente tutto il tempo di cui ha bisogno per rispondere alle domande

A supporto dell'utente vi è l'interfaccia semplice ed intuitiva di Google Moduli che mostra chiaramente all'utente le domande e le opzioni selezionabili per ciascuna domanda, un foglio di calcolo riceve automaticamente in tempo reale i dati raccolti la cui analisi può essere effettuata tramite una apposita sezione di Google Moduli.

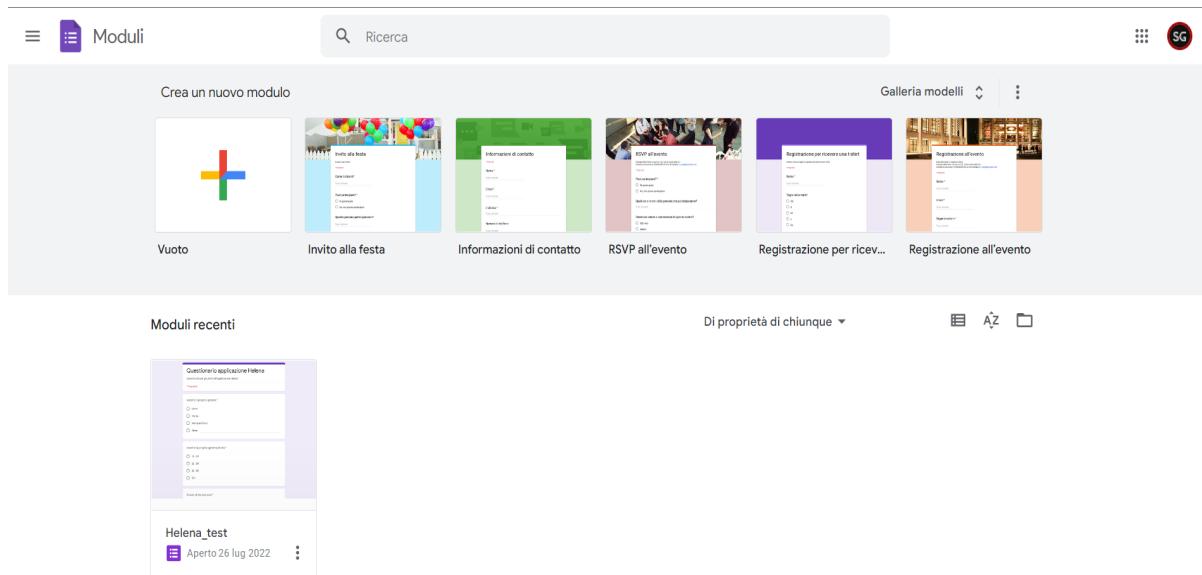


Figura 101: Google Moduli

### **Questionario Generale**

Tramite questo questionario sono raccolte informazioni, non sensibili, di carattere generale degli utenti con lo scopo di comprendere la dimestichezza che ha nell'utilizzo di app per monitorare la salute o comunque di app che includono funzionalità simili a quelle sviluppate in relazione all'età, al genere, al grado di formazione e all'attività sportiva dell'utente. In dettaglio le domande sono le seguenti:

- Inserire il proprio genere (Uomo/Donna/Non specificato)
- Inserire la propria gamma di età (18-24/ 25-34/ 35-49/ 50+)

- Grado di formazione (Diploma di scuola elementare/ Diploma di scuola media/ Diploma di scuola superiore/ Laurea/ Altro)
- Livello di conoscenza delle tecnologie (Base/ Intermedio/ Avanzato/ Molto avanzato)
- Pratichi sport? (Si/ No)
- Hai mai utilizzato app per monitorare lo stato di salute? (Si/ No/ Non so)
- Hai mai utilizzato app per cercare ospedali/farmacie? (Si/ No/ Non so)
- Hai mai utilizzato app per monitorare lo stato emotivo/psicologico? (Si/ No/ Non so)

### Questionario NPS

L'NPS, *Net Promoter Score*, è un parametro basato su una domanda molto semplice: "Con quale probabilità consiglierebbe questa azienda/prodotto/sito ad un amico o collega?". Gli utenti rispondono a questa domanda assegnando un punteggio su una scala da 0 a 10 secondo il gradimento o la soddisfazione nei confronti dell'azienda/prodotto/sito. In base al punteggio, l'utente è classificato come:

- Detrattore - punteggio da 0 a 6
- Passivo - punteggio da 7 a 8
- Promotore - punteggio da 9 a 10

Il questionario è sottoposto a più utenti e il valore dell'NPS è così calcolato:

$$NPS = \frac{(NumeroPromotori - NumeroDetrattori)}{(NumeroUtenti)} \cdot 100$$

I vantaggi del Net Promoter Score sono molteplici: il questionario è molto intuitivo e facile da usare offrendo un linguaggio comune per la classificazione degli utenti in tre categorie distinte ovvero: promotori, passivi e detrattori inoltre agevola l'analisi comparativa tra prodotti concorrenti in quanto è adottato in maniera diffusa dalla maggior parte delle aziende nel mondo. Tuttavia l'NPS non è un parametro specifico e non è in grado di mettere in evidenza problematiche presenti nel prodotto valutato, questo problema può essere ovviato utilizzando questionari più specifici come ad esempio il questionario SUS.

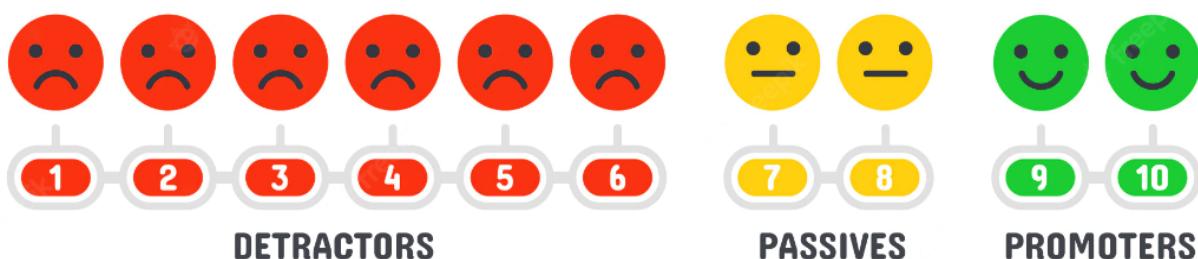


Figura 102: Net Promoter Score

### Questionario SUS

Il questionario SUS, *System Usability Scale*, ha lo scopo di misurare l'usabilità di un sistema. Il questionario include 10 domande, a ciascuna domanda l'utente può assegnare un punteggio da 1 a 5 in base al grado di accordo con l'affermazione, con 5 l'utente è completamente d'accordo mentre con 1 l'utente è completamente in disaccordo. Per ogni questionario, il SUS fornisce un punteggio da 0 a 100, calcolato nel seguente modo per tener conto delle polarità invertite:

- Per gli item dispari, è sottratto 1 dalla risposta dell'utente
- Per gli item pari, è sottratto 5 dalla risposta dell'utente

In questo modo, tutti i valori vanno da 0 a 4 la cui somma va da 0 a 40, tale punteggio è moltiplicato per 2.5 ottenendo un punteggio finale che varia da 0 a 100. Un sistema usabile deve avere un punteggio di almeno 67/100.

Il questionario SUS presenta molteplici vantaggi quali: facilità di somministrazione agli utenti, può essere utilizzato su campioni di piccole dimensioni con risultati affidabili infine è un questionario valido, in grado di discriminare sistemi usabili e sistemi non usabili.



Figura 103: System Usability Scale

## 5.2 Analisi dei dati raccolti

In questo paragrafo discuto dei dati raccolti mediante il modulo Google, calcolando i punteggi dei questionari NPS e SUS somministrati ciascun task relativo alle sezioni implementate in Helena.

### Domande di carattere generale

Inserire il proprio genere

30 risposte

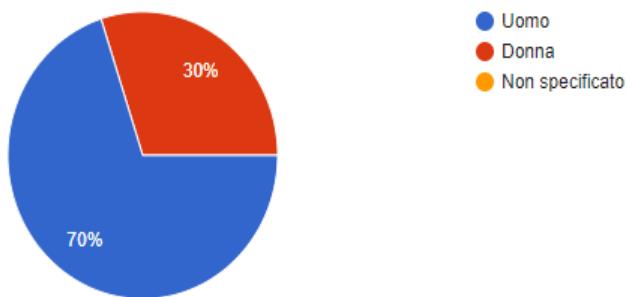


Figura 104: Grafico genere degli utenti

Gli utenti selezionati per il test sono per la maggior parte maschi, nonostante ci fosse l'opzione "Non specificato" non è stata selezionata da alcun utente.

Inserire la propria gamma di età

30 risposte

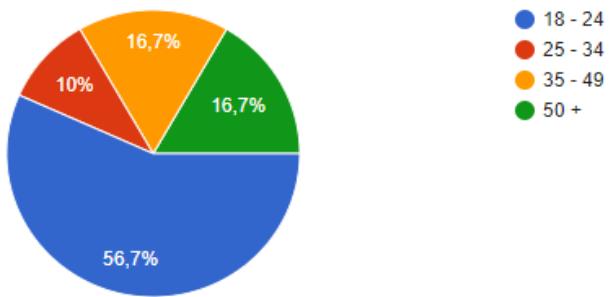


Figura 105: Grafico età degli utenti

Gli utenti che hanno risposto al questionario sono per oltre la metà di età compresa tra i 18 e i 24 anni, questo è possibile spiegarlo in quanto gran parte degli utenti a cui è stato somministrato il modulo sono miei colleghi del corso di laurea o amici di un gruppo di coetanei. Sono stati scelti molti utenti in questa fascia di età in quanto sono più propensi a praticare sport utilizzando applicazioni per il monitoraggio dell'attività motoria. Inoltre, nel campione di utenti considerato, il 33,4% ha più di 35 anni, in questa fascia di età gli utenti, con l'aumentare dei problemi di salute, sono più propensi a utilizzare applicazioni per la ricerca di farmacie/ospedali.

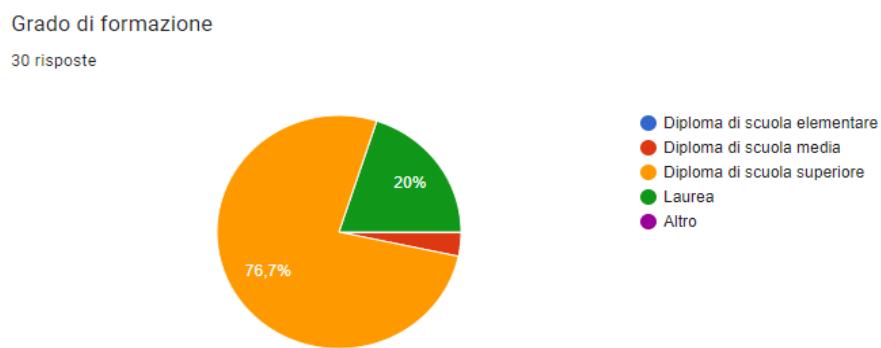


Figura 106: Grafico di formazione degli utenti

Come già detto, la maggior parte degli utenti sono ragazzi di età compresa tra i 18 e i 24 anni, ragazzi che dopo aver concluso il liceo hanno proseguito gli studi iscrivendosi all'università. Soltanto sei utenti su trenta hanno una laurea (triennale/magistrale/a ciclo unico) ed un solo utente ha dichiarato di essere in possesso del solo diploma di scuola media.

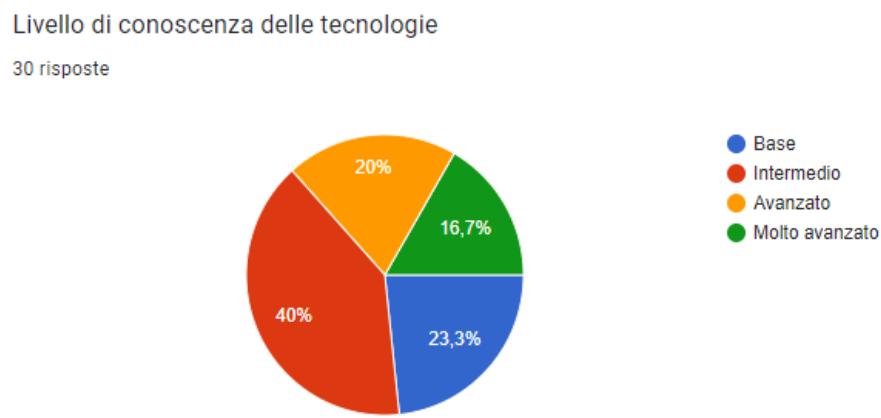


Figura 107: Livello di conoscenza delle tecnologie degli utenti

La maggior parte degli utenti ha un livello intermedio di conoscenze delle tecnologie, seguito dal 36.7% che dichiara una conoscenza avanzata e molto avanzata infine circa il 23% dichiara una conoscenza base limitata, probabilmente, alle funzionalità base offerte dai dispositivi elettronici.

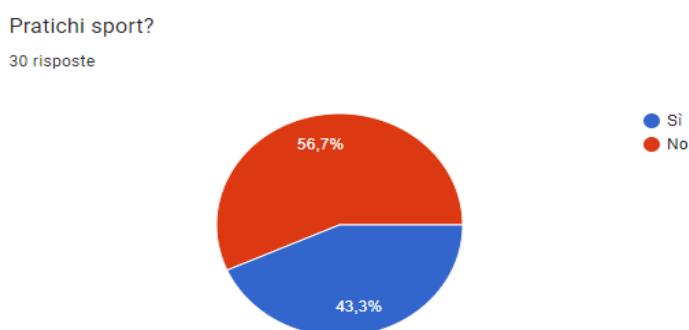


Figura 108: Statistiche di chi pratica sport

Nonostante la maggior parte degli utenti coinvolti è di giovane età e gode di buona salute, oltre la metà di loro ha dichiarato di non praticare alcun tipo di sport, percentuale che si riflette nell'utilizzo di app per il monitoraggio della salute.

Hai mai utilizzato app per monitorare lo stato di salute?

30 risposte

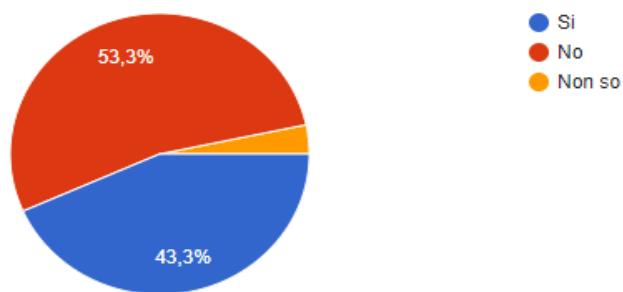


Figura 109: Statistiche di chi utilizza app per il monitoraggio della salute

Le percentuali di chi utilizza app per il monitoraggio della salute coincide con la percentuale di chi pratica sport ma oltre la metà degli utenti dichiara di non aver mai utilizzato app per il monitoraggio dello stato di salute e un solo utente dichiara di non sapere se ha mai utilizzato app di questo genere.

Hai mai utilizzato app per cercare ospedali/farmacie?

30 risposte

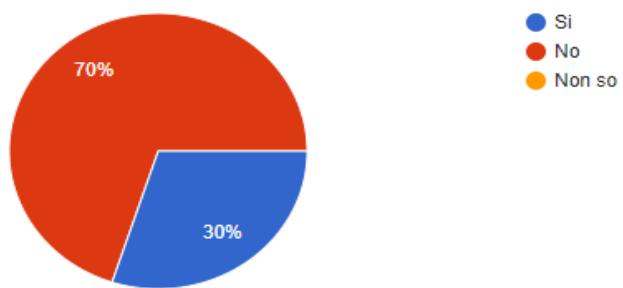


Figura 110: Statistiche di chi utilizza app per cercare ospedali/farmacie

La maggior parte degli utenti dichiara di non aver mai utilizzato applicazioni per la ricerca di ospedali/farmacie in quanto sono applicazioni poco diffuse e spesso non aggiornate ai dati più recenti.

Hai mai utilizzato app per monitorare lo stato emotivo/psicologico?

30 risposte

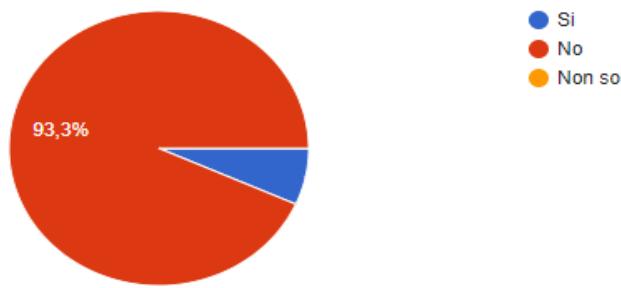


Figura 111: Statistiche di chi utilizza app per monitorare lo stato emotivo/psicologico

Nonostante con la pandemia è aumentata da parte dei cittadini di supporto psicologico, oltre il 90% degli utenti del campione selezionato dichiara di non aver mai utilizzato questo tipo di applicazioni, soltanto due utenti su trenta dichiarano di averle usate almeno una volta.

### **Valutazione sezione Ospedalizzazioni**

La sezione Ospedalizzazioni è stata testata assegnando agli utenti il seguente task:

*Inserimento di una ospedalizzazione con almeno un sintomo in un ospedale a 40 chilometri dalla propria posizione.*

La domanda posta agli utenti nel questionario NPS è stata la seguente: "Su una scala da 0 a 10, quanto consigliresti la sezione ospedalizzazioni ad un amico?"

#### Questionario NPS Task #1

Su una scala da 0 a 10, quanto consigliresti la sezione ospedalizzazioni ad un amico?

30 risposte

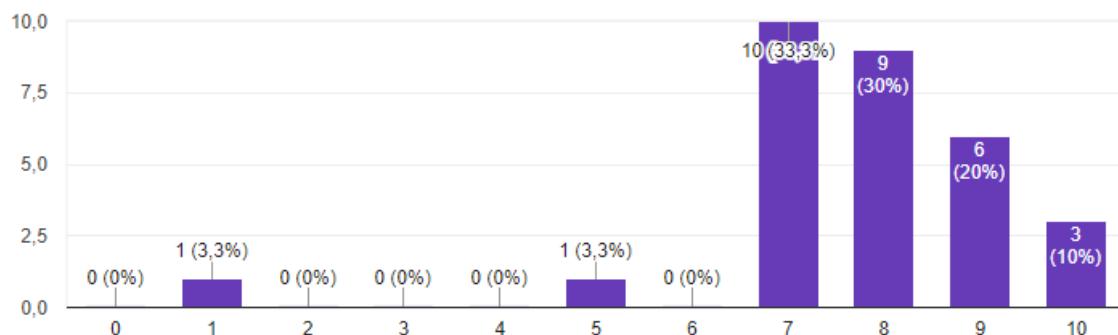


Figura 112: Questionario NPS sezione ospedalizzazioni

Dal grafico possiamo notare che il 30% degli utenti è classificabile come *promotore* mentre oltre il 60% è classificabile come *passivo* e soltanto due utenti su trenta sono classificabili come *detrattori*, quindi il Net Promoter Score risultante è il seguente:



Figura 113: Net Promoter Score ospedalizzazioni

Il Net Promoter Score è nella media, ciò è dovuto alla presenza di 9 promotori e soltanto 2 utenti detrattori tuttavia oltre il 60% degli utenti è classificabile come *passivo*.

Il *System Usability Scale* medio ottenuto dal questionario SUS relativo alla sezione ospedalizzazioni è pari a **68.08**, un punteggio mediamente buono influenzato sicuramente dalla poca propensione degli utenti ad utilizzare la sezione ospedalizzazioni frequentemente.

Penso che mi piacerebbe usare la sezione ospedalizzazioni frequentemente

30 risposte

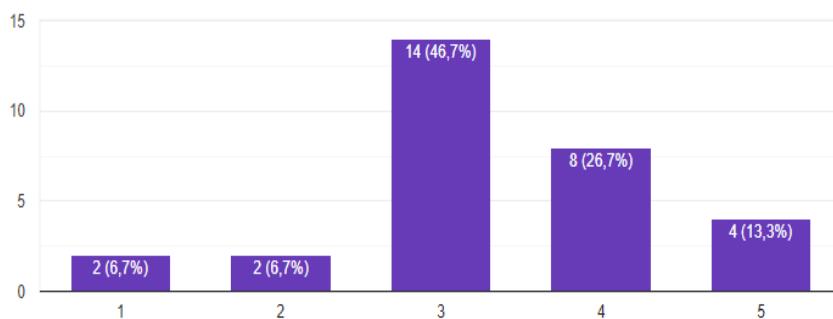


Figura 114: Domanda 1 questionario SUS

Riepilogando i punteggi ottenuti dai questionari SUS ed NPS sono i seguenti:

- Net Promoter Score: 23
- System Usability Scale: 68.08

### Valutazione sezione Farmacie

La sezione Farmacie è stata testata assegnando agli utenti il seguente task:

*Ricerca di una parafarmacia entro 3 chilometri dalla propria residenza.*

La domanda posta agli utenti nel questionario NPS è stata la seguente: "Su una scala da 0 a 10, quanto consigliresti la sezione farmacie ad un amico?"

Su una scala da 0 a 10, quanto consiglieresti la sezione farmacie ad un amico?

30 risposte

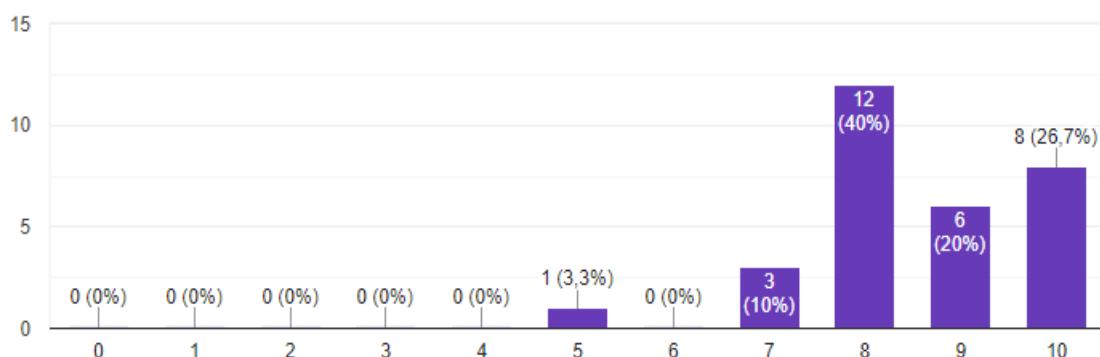


Figura 115: Questionario NPS sezione farmacie

Dal grafico possiamo notare che il 46,7% è classificabile come *promotore*, il 50% è *passivo* e un solo utente è *detrattore*, quindi il Net Promoter Score risultante è il seguente:



Figura 116: Net Promoter Score farmacie

Il Net Promoter Score è risultato molto buono, segno che la sezione è stata particolarmente apprezzata da parte degli utenti i quali hanno a disposizione nel Playstore poche app con funzionalità simili.

Il System Usability Scale medio ottenuto dal questionario SUS relativo alla sezione farmacie è pari a **75,8**, un punteggio molto buono che conferma l'apprezzamento degli utenti nei confronti di questa sezione risultata particolarmente utile e facile da usare come è possibile evincere dal grafico seguente relativo a una delle domande presenti nel questionario. In particolare nella domanda 3, l' 80% degli utenti ha ritenuto la sezione molto facile da usare, di questi il 43,3% degli utenti ha assegnato un punteggio pari a 5/5 mentre il 36,7% degli utenti ha assegnato un punteggio pari a 4/5. Soltanto un utente ha ritenuto abbastanza scarsa l'usabilità della sezione Farmacie.

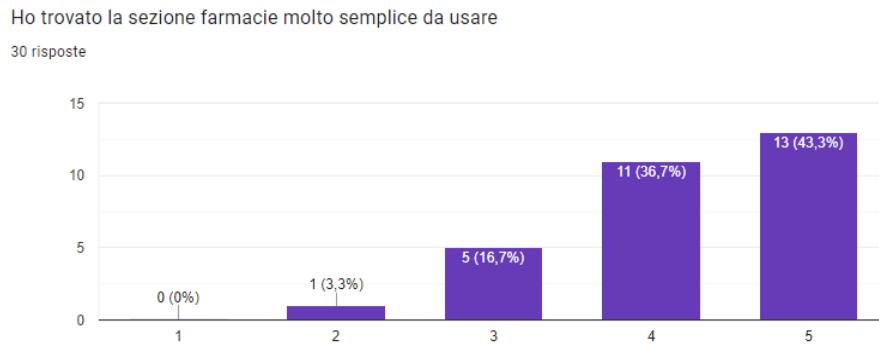


Figura 117: Domanda 3 questionario SUS

Riepilogando i punteggi ottenuti dai questionari SUS ed NPS sono i seguenti:

- Net Promoter Score: 43
- System Usability Scale: 75,8

### Valutazione sezione Stato Emotivo

La sezione Stato emotivo è stata testata assegnando agli utenti il seguente task:

*Inserimento di uno stato emotivo e ricerca di stati emotivi inseriti due giorni prima.*

La domanda posta agli utenti nel questionario NPS è stata la seguente: "Su una scala da 0 a 10, quanto consiglieresti la sezione Stato emotivo ad un amico?"

Su una scala da 0 a 10, quanto consiglieresti la sezione stato emotivo ad un amico?

30 risposte

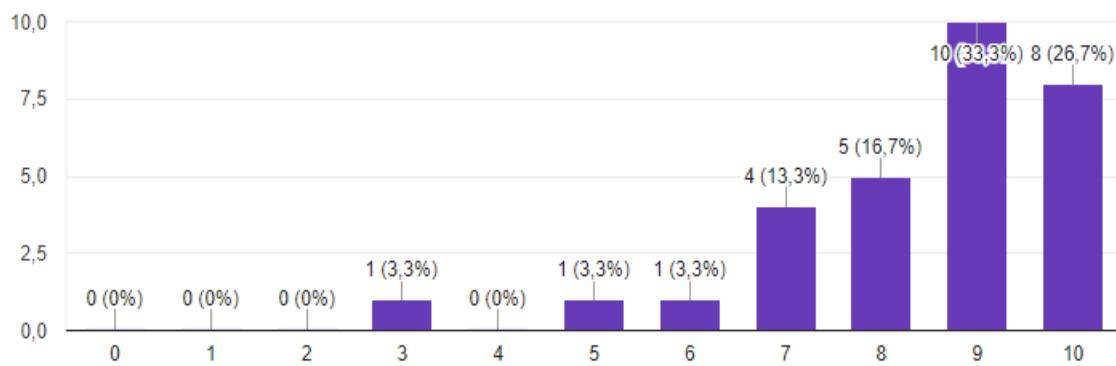


Figura 118: Questionario NPS sezione Stato Emotivo

Dal grafico possiamo notare che il 60% degli utenti è classificabile come *promotore*, il 30% è classificabile come *passivo* e soltanto tre utenti sono *detrattori*, quindi in Net Promoter Score risultante è il seguente:



Figura 119: Net Promoter Score Stato Emotivo

Il valore alto del Net Promoter Score dimostra l'apprezzamento degli utenti i quali probabilmente hanno l'esigenza di una app in cui dopo una dura giornata di lavoro o una delusione di qualsiasi tipo possano esprimere il proprio stato emotivo e tenerne traccia nel tempo.

Il System Usability Scale medio ottenuto dal questionario SUS relativo alla sezione Stato Emotivo è pari a **80.5** a dimostrazione che la sezione non solo è stata percepita molto utile da parte degli utenti ma è anche in grado di garantire una elevata usabilità.

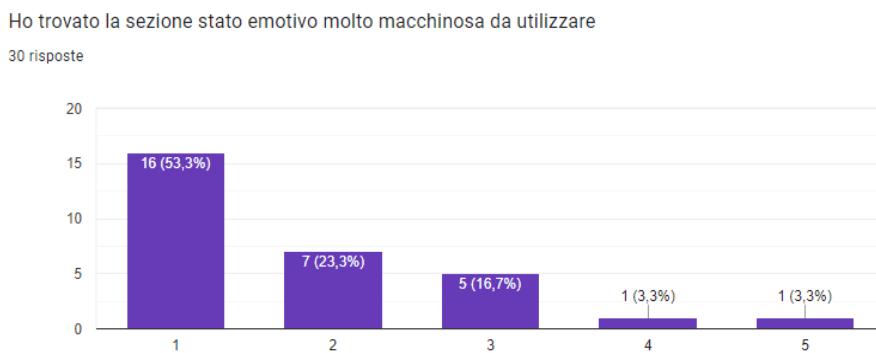


Figura 120: Domanda 8 questionario SUS

Soltanto due utenti hanno trovato la sezione molto macchinosa mentre il 76,6% degli utenti ha trovato la sezione poco macchinosa e intuitiva da utilizzare. Riepilogando i punteggi ottenuti dai questionari SUS ed NPS sono i seguenti:

- Net Promoter Score: 50
- System Usability Scale: 80.5

*L'unico modo di fare un ottimo lavoro è amare quello che fai. [Steve Jobs]*

## 6 Conclusioni e sviluppi futuri

Il lavoro di tesi ha introdotto funzionalità inedite nell'applicazione e-health Helena, funzionalità raramente presenti nelle applicazioni concorrenti. In particolare nella sezione **Ospedalizzazioni** l'utente può inserire un ricovero, specificando causa e sintomi e attraverso l'integrazione della sezione ospedali può cercare e visualizzare gli ospedali presenti vicini alla sua posizione, alla sua residenza (con la relativa distanza in kilometri) oppure inserire un nuovo ospedale nel database e visualizzando la posizione di ogni ospedale in una mappa integrata nell'applicazione.

La seconda sezione implementata è stata la sezione **Farmacie** in cui l'utente può visualizzare tutte le farmacie e parafarmacie italiane presenti sia nei pressi della sua posizione e della sua residenza, visualizzandole su una apposita mappa con la possibilità di inserire/ modificare/ eliminare una farmacia presente nel database. L'utente può cercare le farmacie sia per nome che per città inoltre è integrata la ricerca sul web tramite browser predefinito della farmacia selezionata.

Nelle funzionalità implementate è stato preso in considerazione l'aspetto strettamente emotivo degli utenti, i quali nella sezione **Stato emotivo** possono inserire il proprio stato emotivo, selezionando l'emozione provata (tra le sei emozioni di base presenti quali: neutralità, felicità, tristezza, rabbia, paura, disgusto) aggiungendo dei pensieri riguardo lo stato emotivo inserito. Inoltre l'utente può visualizzare, cercare ed eliminare gli stati emotivi inseriti nel database.

Un'ulteriore sezione introdotta è stata la sezione **Temperatura corporea** in cui l'utente può inserire manualmente nel database la propria temperatura corporea, visualizzando e cercando le misurazioni inserite i cui valori numerici sono riportati in un apposito grafico.

In aggiunta all'introduzione delle nuove funzionalità sono stati effettuati importanti bugfix che hanno coinvolto il login nell'applicazione, la sezione "Diario alimentare" e la sezione "Documenti clinici". Quest'ultima sezione è stata integrata con la sezione Ospedalizzazioni, è stata migliorata l'interfaccia e sono state aggiunte le funzionalità di ricerca per nome e per data del referto.

Le nuove sezioni introdotte possono essere estese e migliorate ulteriormente con funzionalità quali:

- Integrazione dei dati provenienti da un termometro smart nella sezione Temperatura corporea
- Estensione della sezione stato emotivo con una funzionalità in grado di misurare il livello di stress attraverso un apposito questionario psicologico
- Estensione della sezione Ospedali con una funzionalità per monitorare in tempo reale i posti letto disponibili nei vari reparti
- Integrazione della sezione Ospedalizzazioni con i sistemi informativi delle ASL locali
- Integrazione dei dati provenienti dal Fitbit con la sezione Ospedalizzazioni
- Estensione della sezione Farmacie con una chat in tempo reale tra utente e farmacista

In conclusione, le funzionalità sviluppate e le funzionalità che potrebbero essere ancora sviluppate rendono l'applicazione Helena un progetto innovativo e potenzialmente di successo tra gli utenti.

Ringrazio il prof. Pasquale Lops e il dott. Marco Polignano per l'indispensabile supporto in questo lavoro di tesi.

Ringrazio la mia famiglia per avermi supportato durante questo percorso.

Ringrazio i miei amici e colleghi conosciuti durante questi tre anni presso il Dipartimento di Informatica: Federico Canistro, Alba Basile, Leonardo Dipace, Matteo Inglese, Vincenzo La Bella, Antonio Capicotto, Michele Carnimeo, Fabio Abbondanza, Ivan De Cosmis.

Ringrazio i miei amici e compagni di sempre: Stefano Labianca, Ezio Di Bisceglie, Federico De Benedittis, Domenico Campanale, Mattia Cantatore.

## Riferimenti bibliografici

- [And] Developer Android. Platform architecture. <https://developer.android.com/guide/platform>. Accessed: 16-05-2022.
- [AWS<sup>+</sup>21] Emeka C Anyanwu, R Parker Ward, Atman Shah, Vineet Arora, and Craig A Umscheid. A mobile app to facilitate socially distanced hospital communication during covid-19: Implementation experience. *JMIR mHealth and uHealth*, 9(2):e24452, 2021.
- [BHH<sup>+</sup>05] Nuala M Byrne, Andrew P Hills, Gary R Hunter, Roland L Weinsier, and Yves Schutz. Metabolic equivalent: one size does not fit all. *Journal of Applied physiology*, 2005.
- [BLS<sup>+</sup>20] Jacob E Barkley, Andrew Lepp, Antonio Santo, Ellen Glickman, and Bryan Dowdell. The relationship between fitness app use and physical activity behavior is mediated by exercise identity. *Computers in Human Behavior*, 108:106313, 2020.
- [Bok21] Anthony Jnr Bokolo. Application of telemedicine and ehealth technology for clinical services in response to covid-19 pandemic. *Health and technology*, 11(2):359–366, 2021.
- [Cala] John Callaham. The history of android: The evolution of the biggest mobile os in the world. <https://www.androidauthority.com/history-android-os-name-789433/>. Accessed: 16-05-2022.
- [Calb] Giancarlo Calzetta. Cybersecurity: perché la sanità è difficile da proteggere. <https://www.ilsole24ore.com/art/cybersecurity-perche-sanita-e-difficile-proteggere-AEkI5YLE>.
- [coma] Quora community. What is the first mobile application? <https://www.quora.com/What-is-the-first-mobile-application>. Accessed: 08-08-2022.
- [comb] Iberdrola company. ehealth, when technology becomes the key to social well-being. <https://www.iberdrola.com/innovation/ehealth>. Accessed: 24-04-2022.
- [con] Ti consiglio. Bonus psicologo, troppe richieste: metà dei beneficiari esclusi. <https://www.ticonsiglio.com/bonus-psicologo-richieste/>. Accessed: 08-08-2022.
- [Cyb] Redazione Red Hot Cyber. Denunciato l'attacco ransomware all'ospedale di düsseldorf. <https://www.redhotcyber.com/post/denunciato-l-attacco-ransomware-all-ospedale-di-dusseldorf/>.
- [DMCM20] Courtney R Davis, Karen J Murphy, Rachel G Curtis, and Carol A Maher. A process evaluation examining the performance, adherence, and acceptability of a physical activity and diet artificial intelligence virtual health assistant. *International journal of environmental research and public health*, 17(23):9137, 2020.
- [dodmcedo] Federazione Nazionale degli ordini dei medici chirurghi e degli odontodentisti. Elenco dei medici caduti nel corso dell'epidemia di covid-19. <https://portale.fnomceo.it/elenco-dei-medici-caduti-nel-corso-dellepidemia-di-covid-19/>. Accessed: 04-08-2022.

- [E<sup>+</sup>01] Gunther Eysenbach et al. What is e-health? *Journal of medical Internet research*, 3(2):e833, 2001.
- [Eme] Emeteria. A brief history of android studio. <https://emeteria.com/learn/history-android-studio#:text=Android%20Studio%20was%20first%20released,was%20created%20in%20Accessed:08-07-2022>. Accessed: 08-07-2022.
- [GFM<sup>+</sup>21] Lorenzo Gios, Giulia Crema Falceri, Stefano Micocci, Luigi Patil, Sara Testa, Simona Sforzin, Ettore Turra, Diego Conforti, Giulia Malfatti, Monica Moz, et al. Use of ehealth platforms and apps to support monitoring and management of home-quarantined patients with covid-19 in the province of trento, italy: app development and implementation. *JMIR formative research*, 5(5):e25713, 2021.
- [GHL<sup>+</sup>19] Kyeonghye Guk, Gaon Han, Jaewoo Lim, Keunwon Jeong, Taejoon Kang, Eun-Kyung Lim, and Juyeon Jung. Evolution of wearable devices with real-time disease monitoring for personalized healthcare. *Nanomaterials*, 9(6):813, 2019.
- [Gim] Gimbe. Report 7/2019. il definanziamento 2010-2019 del ssn. <https://www.gimbe.org/pagine/1229/it/report-72019-il-definanziamento-20102019-del-ssn>. Accessed: 04-08-2022.
- [GRM15] Mouzhi Ge, Francesco Ricci, and David Massimo. Health-aware food recommender system. In *Proceedings of the 9th ACM Conference on Recommender Systems*, pages 333–334, 2015.
- [GT18] Akhil Gudivada and Nasseh Tabrizi. A literature review on machine learning based medical information retrieval systems. In *2018 IEEE symposium series on computational intelligence (SSCI)*, pages 250–257. IEEE, 2018.
- [Gur21] Dimitri Gurtner. Neuralink and beyond: Challenges of creating an enhanced human. Technical report, Université de Fribourg, 2021.
- [GZD<sup>+</sup>21] Eddy A Golden, Micol Zweig, Matteo Danieletto, Kyle Landell, Girish Nadkarni, Erwin Bottinger, Lindsay Katz, Ricardo Somarriba, Vansh Sharma, Craig L Katz, et al. A resilience-building app to support the mental health of health care workers in the covid-19 era: Design process, distribution, and evaluation. *JMIR Formative Research*, 5(5):e26590, 2021.
- [HKS<sup>+</sup>18] Insol Hwang, Hong Nam Kim, Minho Seong, Sang-Hyeon Lee, Minsu Kang, Hoon Yi, Won Gyu Bae, Moon Kyu Kwak, and Hoon Eui Jeong. Multifunctional smart skin adhesive patches for advanced health care. *Advanced healthcare materials*, 7(15):1800275, 2018.
- [Iye20] Sriram Iyengar. Mobile health (mhealth). In *Fundamentals of telemedicine and telehealth*, pages 277–294. Elsevier, 2020.
- [JS21] Jawahar Jagarapu and Rashmin C Savani. A brief history of telemedicine and the evolution of teleneonatology. In *Seminars in Perinatology*, volume 45, page 151416. Elsevier, 2021.
- [JTR<sup>+</sup>21] Beth K Jaworski, Katherine Taylor, Kelly M Ramsey, Adrienne Heinz, Sarah Steinmetz, Ian Pagano, Giovanni Moraja, and Jason E Owen. Exploring usage of covid coach, a

- public mental health app designed for the covid-19 pandemic: evaluation of analytics data. *Journal of medical Internet research*, 23(3):e26559, 2021.
- [Kin] Kinsta. Cosa è github? introduzione a github per principianti. <https://kinsta.com/it/knowledgebase/cosa-e-github/>. Accessed: 08-07-2022.
- [KK19] Angelina Kouroubali and Dimitrios G Katehakis. The new european interoperability framework as a facilitator of digital transformation for citizen empowerment. *Journal of biomedical informatics*, 94:103166, 2019.
- [KKL<sup>+</sup>20] Changhwan Kim, Gibaek Kim, Youngbin Lee, Giuk Lee, Seungyong Han, Daeshik Kang, Sumin Helen Koo, and Je-sung Koh. Shape memory alloy actuator-embedded smart clothes for ankle assistance. *Smart Materials and Structures*, 29(5):055003, 2020.
- [KZHOB17] Kourosh Kalantar-Zadeh, Nam Ha, Jian Zhen Ou, and Kyle J Berean. Ingestible sensors. *ACS sensors*, 2(4):468–483, 2017.
- [Lab] MIT Media Laboraotry. Wearable computing. [media.mit.edu/wearables/](http://media.mit.edu/wearables/).
- [LBL<sup>+</sup>20] Tony Luczak, Reuben Burch, Edwin Lewis, Harish Chander, and John Ball. State-of-the-art review of athletic wearable technology: What 113 strength and conditioning coaches and athletic trainers from the usa said about technology in sports. *International Journal of Sports Science & Coaching*, 15(1):26–40, 2020.
- [Leo] Giancarlo Leo. Le competenze digitali dei professionisti sanitari nell'epoca del coronavirus. <https://www.aidr.it/le-competenze-digitali-dei-professionisti-sanitari-nellepoca-del-coronavirus/>.
- [LTYW08] Gang Luo, Chunqiang Tang, Hao Yang, and Xing Wei. Medsearch: a specialized search engine for medical information retrieval. In *Proceedings of the 17th ACM conference on Information and knowledge management*, pages 143–152, 2008.
- [M<sup>+</sup>19] Elon Musk et al. An integrated brain-machine interface platform with thousands of channels. *Journal of medical Internet research*, 21(10):e16194, 2019.
- [MCGR03] Fabrizia Mantovani, Gianluca Castelnuovo, Andrea Gaggioli, and Giuseppe Riva. Virtual reality training for health-care professionals. *CyberPsychology & Behavior*, 6(4):389–395, 2003.
- [MCT13] Yung-Cheng Ma, Yi-Ping Chao, and Tzung-You Tsai. Smart-clothes—prototyping of a health monitoring platform. In *2013 IEEE Third International Conference on Consumer Electronics; Berlin (ICCE-Berlin)*, pages 60–63. IEEE, 2013.
- [Mor17] Laurence Moroney. The firebase realtime database. In *The Definitive Guide to Firebase*, pages 51–71. Springer, 2017.
- [Mus14] Benjamin Muschko. *Gradle in action*. Simon and Schuster, 2014.
- [NSZZA19] Nariman Noorbakhsh-Sabet, Ramin Zand, Yanfei Zhang, and Vida Abedi. Artificial intelligence transforms the future of health care. *The American journal of medicine*, 132(7):795–801, 2019.

- [OML15] Oluwaseun Ireti Obasola, Iyabo Mabawonku, and Ikeoluwa Lagunju. A review of e-health interventions for maternal and child health in sub-sahara africa. *Maternal and child health journal*, 19(8):1813–1824, 2015.
- [OMS] OMS. Oms: Covid-19 aumenta del 25% i casi di ansia e depressione. <https://unric.org/it/oms-covid-19-aumenta-del-25-i-casi-di-ansia-e-depressione/>. Accessed: 08-08-2022.
- [OSK<sup>+</sup>21] Aleksandr Ometov, Viktoriia Shubina, Lucie Klus, Justyna Skibińska, Salwa Saafi, Pavel Pascacio, Laura Flueratoru, Darwin Quezada Gaibor, Nadezhda Chukhno, Olga Chukhno, et al. A survey on wearable technology: History, state-of-the-art and current challenges. *Computer Networks*, 193:108074, 2021.
- [PNI<sup>+</sup>20] Marco Polignano, Fedelucio Narducci, Andrea Iovine, Cataldo Musto, Marco De Gemmis, and Giovanni Semeraro. Healthassistantbot: a personal health assistant for the italian language. *IEEE Access*, 8:107479–107497, 2020.
- [PWCL16] Kunwoo Park, Ingmar Weber, Meeyoung Cha, and Chul Lee. Persistent sharing of fitness app status on twitter. In *Proceedings of the 19th ACM Conference on Computer-Supported Cooperative Work & Social Computing*, pages 184–194, 2016.
- [SCSH<sup>+</sup>20] Herman Saksono, Carmen Castaneda-Sceppa, Jessica Hoffman, Vivien Morris, Magy Seif El-Nasr, and Andrea G Parker. Storywell: designing for family fitness app motivation by using social rewards and reflection. In *Proceedings of the 2020 CHI conference on human factors in computing systems*, pages 1–13, 2020.
- [SFdA<sup>+</sup>20] Joan B Soriano, Esteve Fernandez, Alvaro de Astorza, Luis A Pérez de Llano, Alberto Fernandez-Villar, Dolors Carnicer-Pont, Bernardino Alcazar-Navarrete, Arturo Garcia, Aurelio Morales, Maria Lobo, et al. Hospital epidemics tracker (hepitracker): Description and pilot study of a mobile app to track covid-19 in hospital workers. *JMIR Public Health and Surveillance*, 6(3):e21653, 2020.
- [SRE21] Jennifer Schindler-Ruwisch and Christa Palancia Esposito. “alexa, am i pregnant?”: A content analysis of a virtual assistant’s responses to prenatal health questions during the covid-19 pandemic. *Patient education and counseling*, 104(3):460–463, 2021.
- [Staa] Statcounter. Os mobile markket share. <https://gs.statcounter.com/os-market-share/mobile/worldwide>. Accessed: 16-05-2022.
- [Stab] Statista. Number of available applications in the google play store from december 2009 to march 2022. <https://www.statista.com/statistics/266210/number-of-available-applications-in-the-google-play-store/>. Accessed: 16-05-2022.
- [SZ] Luisa Regimenti Stefania Zambelli. Nuove rivelazioni circa l’aggiornamento del piano pandemico italiano e il report dell’oms. [https://www.europarl.europa.eu/doceo/document/P-9-2021-002082\\_IT.html](https://www.europarl.europa.eu/doceo/document/P-9-2021-002082_IT.html). Accessed: 04-08-2022.
- [Thoa] Barbara Thompson. Android architecture: Application layers, framework, component. <https://www.guru99.com/android-architecture.html>. Accessed: 16-05-2022.

- [Thob] Joe Thompson. A concise history of the smartwatch. <https://www.bloomberg.com/news/articles/2018-01-08/a-concise-history-of-the-smartwatch>.
- [Tos] Regione Toscana. App toscana salute. <https://www.uslcentro.toscana.it/index.php/mappa-del-sito/altro/1410-info-covid-19/28270-app-toscana-salute>. Accessed: 04-08-2022.
- [VJL<sup>+</sup>17] Katia Vega, Nan Jiang, Xin Liu, Viirj Kan, Nick Barry, Pattie Maes, Ali Yetisen, and Joe Paradiso. The dermal abyss: Interfacing with the skin by tattooing biosensors. In *Proceedings of the 2017 ACM International Symposium on Wearable Computers*, pages 138–145, 2017.
- [YZD<sup>+</sup>18] Shiqi Yang, Ping Zhou, Kui Duan, M Shamim Hossain, and Mohammed F Alhamid. em-health: towards emotion health through depression prediction and intelligent health recommender system. *Mobile Networks and Applications*, 23(2):216–226, 2018.